

- AUTOMAÇÃO E ROBÓTICA

VEÍCULO AUTÔNOMO PROGRAMÁVEL PARA UTILIZAÇÃO EM ESTUDOS DE VISÃO COMPUTACIONAL

Luiz Eduardo Sauerbronn

Marcelo Dreux

Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Departamento de Engenharia Mecânica
Rua Marquês de São Vicente, 225
CEP 22453-900 - Rio de Janeiro, RJ, Brasil
[sauer, dreux]@mec.puc-rio.br

Ricardo Zelenovsky

Pedro Paulo Levi Mateus Canazio

Instituto Militar de Engenharia (IME)
Departamento de Engenharia Elétrica
Praça General Tibúrcio 80, Praia Vermelha
CEP 22290-270 – Rio de Janeiro, RJ, Brasil
[zele,ppaulo]@aquarius.ime.eb.br

***Resumo.** O presente trabalho descreve a arquitetura de um veículo autônomo programável que possui um suporte mecânico, apoiado sobre três rodas e carrega sobre si dispositivos eletrônicos. Estes dispositivos são responsáveis pelo acionamento dos motores do veículo e pela comunicação (por infravermelho e por rádio) com outros equipamentos externos. A integração das partes mecânicas com as eletrônicas representa o cerne deste trabalho. Este veículo tem como objetivo servir de suporte para estudos em visão computacional, uma vez que sua principal forma de sensoriamento baseia-se, essencialmente, na análise de imagens capturadas por uma câmera acoplada ao veículo.*

***Palavras-chave:** Robótica, Visão, Infravermelho, Rádio.*

1. Introdução

O mundo no qual robôs se movem é tridimensional e dinâmico. Ele se altera em função do próprio movimento do robô e do movimento dos demais objetos e personagens presentes no mundo, que por sua vez estão ocupados cuidando de seus próprios movimentos.

Para interagirem suavemente com o ambiente que os circunda, robôs precisam possuir capacidade de sensoriamento. Dentre as diversas formas de sensoriamento, visão vem sendo reconhecida, há muito, como uma das, potencialmente, mais importantes. Parte deste reconhecimento, deve-se ao fato de que muitos sistemas biológicos utilizam visão como fonte primária na aquisição de informações acerca do ambiente onde se situam.

O projeto de pesquisa (Sauerbronn & Dreux, 1996), que originou o presente trabalho, visa demonstrar a integração das partes eletrônicas e mecânicas utilizadas na implementação de um robô, que servirá de suporte para testes de performance de algoritmos de processamento de imagens e visão robótica.

Para execução destes testes, o robô tem que ser capaz de:

1. Enviar, por rádio, as imagens capturadas pela sua câmera a um micro à distância.
2. Receber comandos provenientes deste mesmo micro à distância.
3. Deslocar-se conforme comando recebido do micro à distância.

O presente trabalho descreve um robô que atende às três características acima.

2. Esquema Geral do Robô

Uma imagem gerada pela câmera é transmitida do veículo para o microcomputador através do aparelho Video-Link (SETO Ind. e Com., 1997), destinado exclusivamente a esta tarefa. A informação transmitida por ele é captada pelo microcomputador através de antena que está diretamente ligada à placa de captura de imagem. Assim, o microcomputador recebe permanentemente as imagens captadas pela câmera e as envia diretamente para a tela do microcomputador.

Esta comunicação é unidirecional (veículo-microcomputador). Conforme o veículo se afasta do microcomputador, o sinal vai perdendo confiabilidade e os níveis de ruído na transmissão aumentam. Caso haja grandes obstáculos (paredes) entre o veículo e o microcomputador o sinal torna-se extremamente ruidoso. Assim sendo, futuramente deve-se procurar desenvolver (ou adquirir) sistema com maior potência de transmissão.

A outra forma de comunicação é utilizada para o microcomputador informar ao veículo que trajetória deve tomar. Para estabelecer-se esta comunicação, foram desenvolvidas 2 placas de infravermelho (uma emissora e outra receptora) especificamente para este fim. Uma destas placas fica acoplada ao microcomputador e sua tarefa é enviar os sinais enviados através da porta paralela do microcomputador para o veículo. A outra placa faz o inverso, capta o sinal do ambiente e o envia para a porta paralela do veículo. O veículo transporta uma placa-mãe de um microcomputador PC/AT486 e é a porta paralela desta placa que recebe o sinal digitalizado. A partir da recepção do sinal, a porta paralela aciona os motores de forma a gerar a trajetória desejada.

A parte mecânica é responsável pelo deslocamento do veículo. A parte de visão procura identificar a melhor trajetória a ser seguida. Por fim, a parte eletrônica é responsável pela conexão entre o resultado do processamento de imagem e o acionamento da parte mecânica.

Esquemáticamente, tem-se (figura 1):

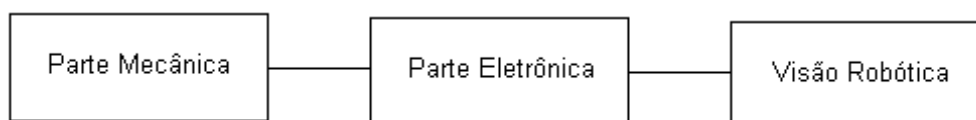


Figura 1 - Integração entre as partes do sistema robótico

3. Parte Mecânica

Há, na verdade, inúmeras configurações mecânicas possíveis para o robô desejado. Caso ele seja móvel, como é o caso do descrito neste trabalho, o veículo pode deslocar-se utilizando rodas, patas, esteiras, dentre outras. Cada possibilidade destas resulta em um enfoque específico em sua implementação. A princípio, deve-se analisar as condições em que o veículo irá trabalhar e, a partir daí, optar pela configuração mais apropriada.

Para a escolha da melhor configuração foram analisados inúmeros modelos de robôs. Foram tomados exemplos oriundos dos mais diferentes e distintos campos de atuação: campeonatos de robótica (FIRST, 1997), universidades (Brooks, 1997), indústrias (ISRobotics, 1997), sites da internet (Robotics Magazine, 1997), dentre outros. Após extensa pesquisa, determinou-se a configuração que melhor atenderia as exigências do projeto. A escolhida, por sua larga gama de aplicações, foi encontrada em robôs de diferentes áreas e propósitos, e utiliza rodas como forma de tração. Na figura 2, pode se ver a disposição dos 02 (dois) motores Gearbox (12V, 20 rpm) da RS do Brasil, 02 (dois) eixos fixos, 01 (uma) roda louca, bem como as correias que ligam os motores aos eixos e a placa de alumínio que agrega o sistema mecânico.

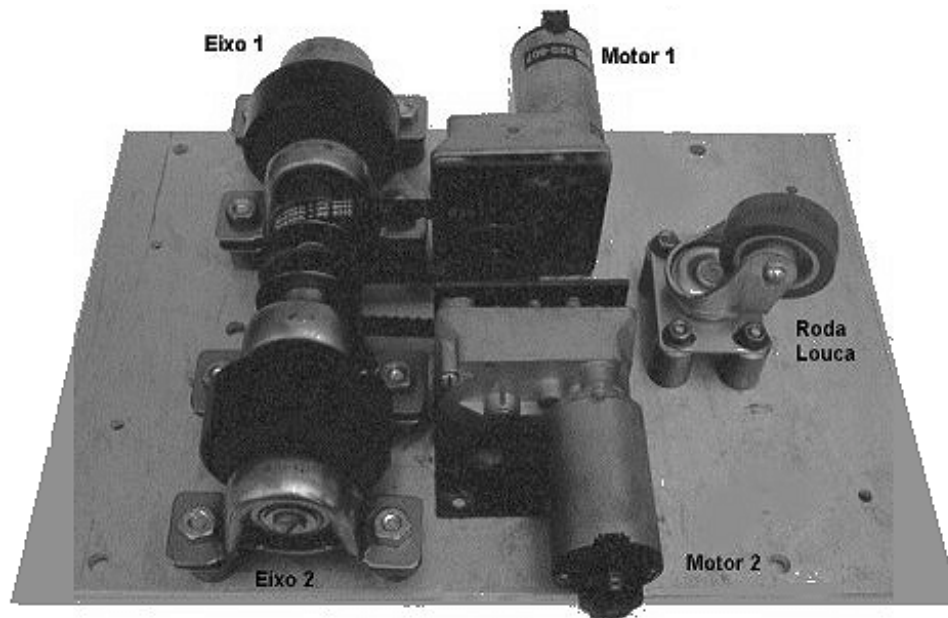


Figura 2 - Configuração mecânica escolhida.

Os motores possuem caixa de redução e possibilitam ao robô atingir uma velocidade de 10cm/seg.. Cada um dos motores mostrados acima (ao centro), são acionados de forma independente pela chamada "placa de interface", que será descrita na seção que segue abaixo.

4. Parte Eletrônica

Descreve-se abaixo, a implementação eletrônica utilizada no projeto. São discutidas as dificuldades encontradas no processo de boot do robô, a questão da autonomia (baterias) e o ítem eletrônico mais importante implementado para o projeto, a "placa de interface".

Nos sistemas robóticos, a eletrônica atua como interface entre os algoritmos de programação e as partes mecânicas envolvidas no processo. O que se espera é que um comando lógico possa ser traduzido em esforços mecânicos. O sistema a ser descrito abaixo foi idealizado a partir do Projeto Pebbles do Laboratório de Inteligência Artificial do MIT (Pebbles Project, 1997).

O primeiro arranjo da implementação eletrônica a ser testado consistiu na utilização de uma placa-mãe de micro PC/AT486, alimentado diretamente a partir de uma bateria de 6 Volts. A montagem do conjunto placa-mãe/bateria é demonstrada esquematicamente abaixo (figura 3):

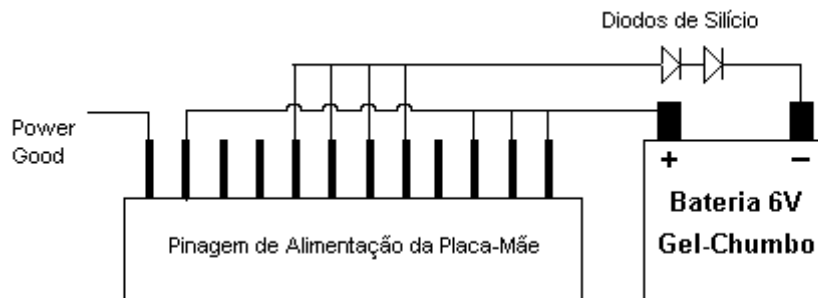


Figura 3 - Representação da alimentação da placa-mãe a partir da bateria de 6 Volts

O Pino de “Power Good” tem a função de permitir o funcionamento do computador somente quando os níveis de tensão nos demais pinos estejam estabilizados. Aterrando-se momentaneamente o pino Power Good, resseta-se a placa-mãe.

É válido ressaltar que a tensão de alimentação tem que ser capaz de polarizar os transistores contidos nos Circuitos Integrados (CI) e não pode ser superior a determinado valor sob pena de superaquecer os CI, queimando-os. Estes valores, para a placa mãe utilizada, foram respectivamente de 3,5Volts (mínimo) e 5,2Volts (máximo).

O processo de Boot

O processo de boot utilizado no corrente trabalho consiste, inicialmente, em ligar a bateria aos terminais de alimentação da placa-mãe. Após isto, nenhuma alteração é sentida até que se aterre momentaneamente o Power Good. Entra então em ação o processo de Reset da placa. Verificou-se que a placa-mãe consome algo na faixa de 3 Ampéres.

Com o Reset, o processo de Boot se inicia. Dentre as primeiras conseqüências, percebe-se uma queda de tensão na bateria e o início de aquecimento nos diodos, que dissipam aproximadamente 2 Watts cada (ver diodos de silício na figura 3).

A BIOS¹ (Bios, 1998) processa então um teste de auto-verificação. Neste teste, irá verificar a configuração do sistema, identificando possíveis situações onde o processo de boot não pode prosseguir (falta de memória RAM, chip da CPU ou placa de vídeo acoplada à placa mãe). Há necessidade também que haja um teclado (ou um terminador no conector do teclado) conectado à placa mãe durante o período de Boot.

¹ Basic Input Output System

Após executar sua auto-verificação, a BIOS inicia o processo de leitura e execução do sistema operacional. Para isto, verifica em sua configuração o dispositivo que contém o sistema (tipicamente dispositivo de disco flexível), acessa-o e inicia a leitura do mesmo.

Após o processo de boot, são desconectados do robô, o teclado e o dispositivo de disco flexível (todo o sistema operacional, bem como programa de gerenciamento dos processos, já foram carregados na memória). E assim, o robô está livre para trafegar pelo ambiente sem nenhuma limitação de fios que o conectem a quaisquer outros objetos.

A Placa de Interface

Parte do projeto original, a chamada Placa de Interface possui 3 objetivos distintos, sendo eles:

i) Controlar os motores DC a partir de sinal recebido pela porta paralela da placa-mãe. O circuito responsável por este controle é mostrado abaixo (figura 4):

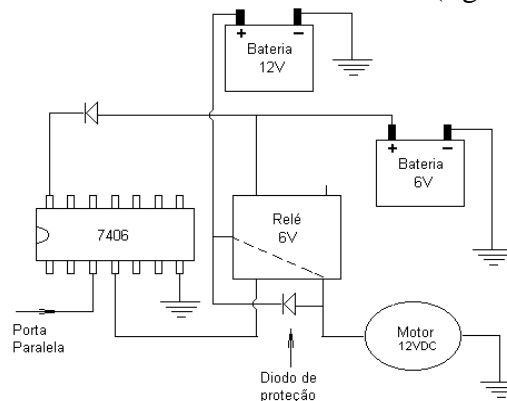


Figura 4 - Circuito responsável pelo acionamento dos Motores 12VDC a partir do sinal da Porta Paralela

Assim, quando a porta paralela envia sinal lógico 0 (equivalente ao terra), o relé fecha a conexão e o motor DC começa a girar. Foi também inserido ao conjunto um diodo com o intuito de evitar corrente reversa sobre o relé. A presença desta corrente reversa reduz a vida útil do relé, pois evita o centelhamento toda vez que há o chaveamento liga/desliga.

O fundamental no acionamento do relé consiste na tensão entre seus bornes e a corrente por ele requerida. Ou seja, há necessidade de uma potência mínima dissipada no relé para que ele permaneça acionado.

Assim, com este aparato é possível acionar os motores DC a partir de comandos enviados para a porta paralela. O comando *outportb*, da Linguagem C de programação é um bom exemplo de comando que envia sinais para a porta paralela. Através do comando *outportb* consegue-se definir os níveis de tensão em oito pinos (equivalentes a um byte) da porta paralela.

ii) Estabelecer comunicação com um micro à distância:

Como previsto no projeto original, o robô envia as imagens obtidas em sua câmera através do aparelho Video Link (SETO Ind. e Com., 1997). Esta imagem é capturada por um micro à distância através de antena sintonizada no canal 12, então, é digitalizada e processada.

O resultado do processamento é então retornado ao robô por infravermelho. Um dos objetivos funcionais da placa de interface é o de estabelecer esta comunicação por infravermelho.

iii) Utilizar ultra-som no reconhecimento de obstáculos:

Adquiriu-se uma placa de ultra-som para detecção de obstáculos na faixa de 30 cm a 1,5 m. O objetivo previsto em projeto consiste em aparelhar o robô com mais este dispositivo de sensoriamento. A partir do momento em que a placa detecta algum objeto aproximando-se, ela envia um sinal à placa de interface que, automaticamente, transfere este sinal ao micro remoto, para que este tome a decisão acerca da nova rota a ser tomada. Esta última função da placa de interface ainda está em fase de implementação.

Após a implementação das partes mecânica e eletrônica acima descritas, obteve-se a primeira versão do robô desejado, cujas fotos podem ser vistas abaixo:

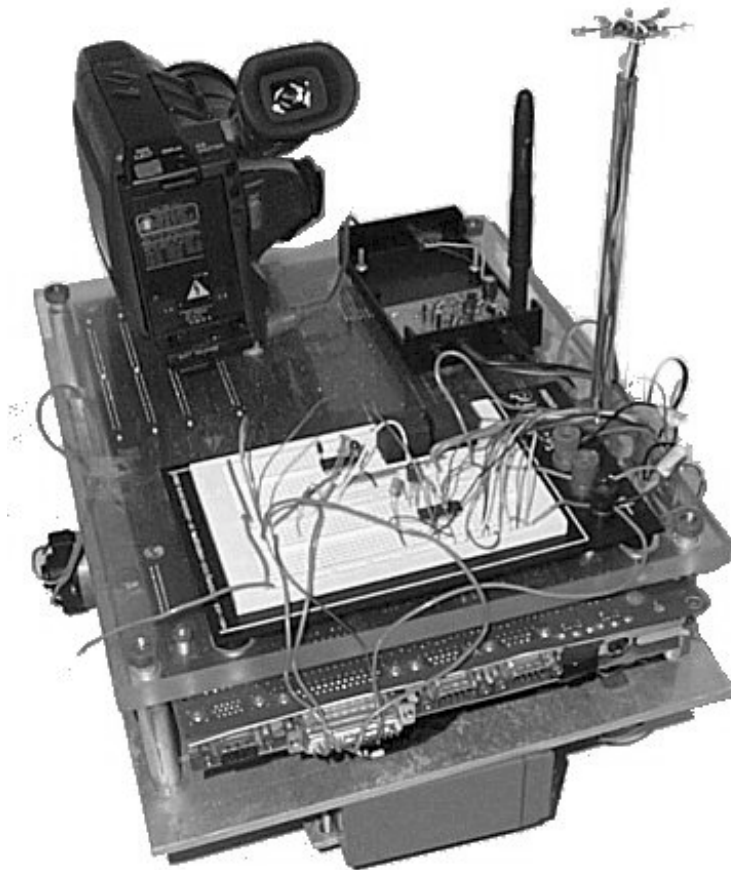


Figura 5 - Vista em perspectiva do robô

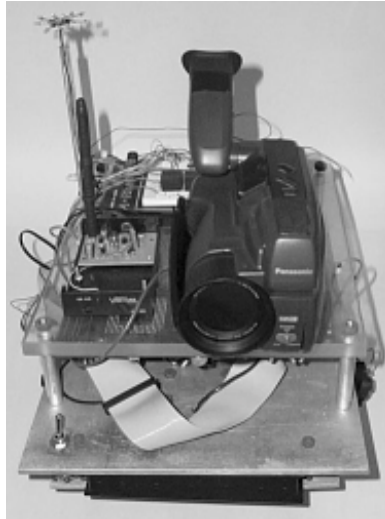


Figura 6 - Vista frontal do robô

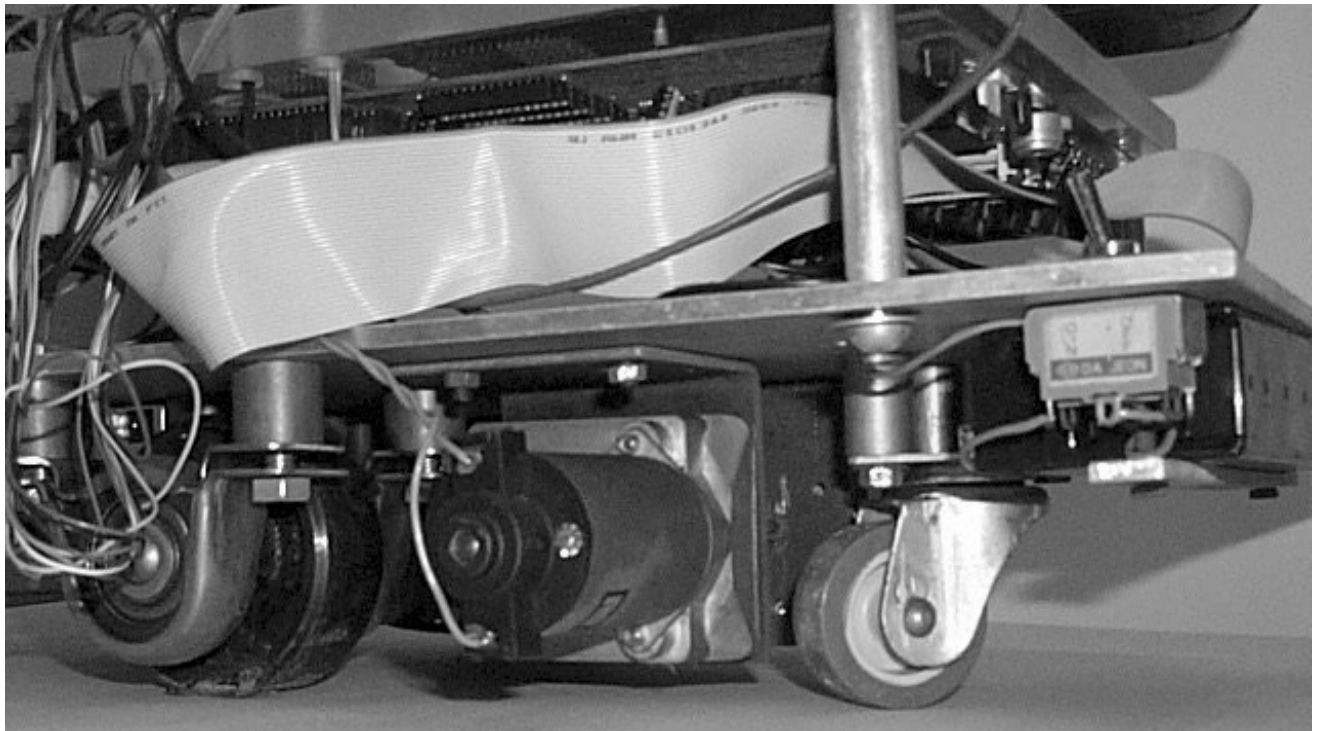


Figura 7 - Vista lateral do robô.

5. Conclusão

Após todo trabalho dispendido na implementação da parte eletrônica, verificou-se quanto o projeto ainda poderia ser simplificado. A única necessidade eletrônica do projeto consiste na implementação de uma placa que acione os motores a partir de comandos de rádio (ou infravermelho) dados por um micro à distância. Não há sequer necessidade da placa-mãe e de toda complexidade de seu acionamento.

Neste sentido, foram adquiridas placas com sistemas embutidos e barramento PC104. Estas placas possuem DOS implementado em hardware, são pequenas, leves, compactas; preenchem todos os requisitos necessários ao presente projeto, ou seja capacidade de processamento (CPU) e comunicação com a placa de interface (através de sua porta paralela) e ainda, não necessitam a conexão de quaisquer periféricos (dispositivo de disco flexível ou teclado) para seu acionamento. Sua alimentação também se dá através da bateria de 6Volts.

Assim, o próprio sistema mecânico poderia tornar-se menor e mais leve. A câmera utilizada também deve ser substituída por uma CCD ligada diretamente à porta paralela. O processamento de imagem passaria assim a ser realizado pelo próprio robô e sua comunicação com o microcomputador à distância teria apenas a finalidade de informar sua atual situação, bem como os resultados encontrados oriundos do processamento das imagens capturadas.

A placa responsável pelo sistema de reconhecimento de obstáculos por ultra-som também irá compor uma nova versão do presente projeto de pesquisa.

Informações bem mais abrangentes sobre o projeto acima mencionado podem ser encontradas na homepage (<http://www.mec.puc-rio.br/~sauer/trab/mest/meurobo/vision1.html>).

Agradecimentos

Aos funcionários da oficina mecânica do laboratório de Termociências da PUC-Rio, pelo carinho e atenção. Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, por ter financiado uma bolsa de estudos ao primeiro autor, cujo trabalho de pesquisa deu origem a este artigo. À Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro, FAPERJ, por ter financiado a implementação do robô. Ao Instituto Militar de Engenharia, IME, pelo apoio institucional.

REFERÊNCIAS

- Sauerbronn & Dreux, 1996, Robótica aplicada a reconhecimento de ambientes, Projeto de pesquisa, Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).
- SETO Ind. e Com., 1997, - Seto Indústria Comércio Produtos Eletrônicos Ltda., Rua Dr. Siqueira Campos, n^o 147, São Paulo, SP, C.G.C.45.922.465/0001-38, 1997.
- FIRST, 1997, - Organização “FIRST - For Inspiration and Recognition in Science and Technology”, <http://www.usfirst.org>, 1997.
- Brooks, 1997, - Prof. Rodney Brooks, diretor do laboratório de Inteligência Artificial, Instituto de Tecnologia de Massachusetts, Boston, Massachusetts, Estados Unidos, <http://www.ai.mit.edu/people/brooks/>, email: brooks@ai.mit.edu, 1997.
- ISRobotics, 1997, Sistemas Robóticos Inteligentes, Boston, Massachusetts, Estados Unidos, <http://www.isr.com>, 1997.
- Robotics Magazine, 1997, “RS&T Magazine for Educators, Students, Hobbyists & Enthusiasts”, <http://www.robotmag.com>, 1997.
- Peebles Project, 1997, Projeto Pebbles, Robôs Exploradores, Laboratório de Inteligência Artificial, Instituto de Tecnologia de Massachusetts, Boston, Massachusetts, Estados Unidos, <http://www.ai.mit.edu/projects/mars-rovers/pebbles.html>, 1997.
- Bios, 1998, MR Bios (Microid Research), <http://www.mrbios.com>, 1998.

AUTONOMOUS PROGRAMMABLE VEHICLE FOR USE IN COMPUTER VISION

***Abstract.** This paper describes the architecture of an autonomous programmable vehicle whose mechanical support rests on three wheels. There are two motors in order to transport some electronic devices. These devices are responsible for powering the vehicle motors, as well as for communication (radio and infrared) with external equipments that are monitoring the process. The integration of the mechanical parts with the electronic parts represents the core of this paper. This vehicle has the purpose of supporting an automated vision system, that works as its main form of sensing.*

***Keywords:** Robotics, Vision, Infrared, Radio.*

INFORMATIZAÇÃO DA ADMINISTRAÇÃO DE MATERIAIS NA FABRICAÇÃO DE *BLANK* SOLDADO

José Jean-Paul Zanlucchi de Souza Tavares

Gilmar Ferreira Batalha

José Reinaldo Silva

Escola Politécnica da USP - Dept. de Engenharia Mecatrônica e de Sistemas Mecânicos -
Av. Prof. Mello Moraes, 2231 05508.900 São Paulo - SP

Resumo

Procede-se inicialmente a uma revisão sobre modelagem de sistemas automatizados e projeto da informatização da área de administração de materiais na fabricação de *blank* soldado, utilizando uma metodologia orientada a objetos dedutiva. Aplicou-se a metodologia *Abstract View / Abstract Object* na concepção e desenvolvimento da representação do sistema, objetivando a reutilização de soluções e uma abordagem híbrida *Top-Down* e *Bottom-Up*. A área de administração de materiais na fabricação de *blank* soldado foi escolhida pelo fato da sua crescente adoção no processo de conformação de chapas e conseqüentes mudanças no processo de manufatura.

Palavras-chave: Manufatura; *Blank* Soldado; Design Orientado à Objeto; Sistema Integrado.

1. INTRODUÇÃO

Esse trabalho busca modelar a informatização da área de administração de materiais na fabricação de *blank* soldado através de técnicas da teoria de sistemas. A configuração de sistemas de manufatura, no caso da fabricação de *blank* soldado, é enfocada segundo dois aspectos: um modelo funcional relacionado com o planejamento da produção, e um modelo físico vinculado ao conjunto dos processos, máquinas e equipamentos. Estes dois modelos necessários para descrever o sistema como um todo, inseridos em contextos diferente, isto é, cada componente interage com um número restrito dos demais componentes, tornam a funcionalidade de seus componentes reflexos de suas interações. Neste sentido são utilizadas duas metodologias para análise da representação dessas funcionalidades e do contexto específico de cada componente, sendo seus formalismos implementados utilizando um banco de dados dedutivo. A representação consiste de objetos chamados de AV (*Abstract views*), isto é, uma síntese da interação dos objetos com o entorno, e outros chamados de AO (*Abstract objects*) que compõe o modelo do artefato em algum formalismo. A área de administração de materiais na fabricação de *blank* soldado foi escolhida pelo fato da crescente adoção de *blank* soldados no processo de conformação de chapas promover alguma mudanças nítidas no processo de manufatura, destacando-se no contexto deste trabalho: as mudanças logísticas (ex.: redução do estoque, diminuindo o número de peças antes separadas e agora passam a formar um *blank* único, somando-se ainda a grande importância dos sistemas atuais de entrega e armazenagem e sua influência sobre as propriedades das chapas e do produto

final, envolvendo centro produtor, centro prestador de serviços e indústria automobilística, e suas respectivas filiais) e as mudanças de produtividade, uma vez que a redução do número de operações de estampagem em prensa, conformando uma só peça em lugar de duas ou três, reduz o consumo de energia, otimiza a utilização de linhas, diminui o tempo de produção em prensa e a montagem no carro, reduzindo estoques intermediários, ao estocar um lote apenas ao invés de várias peças não unidas antes da montagem.

2. MODELAGEM DE SISTEMAS AUTOMATIZADOS

Sistemas automatizados são sistemas complexos compostos por vários subsistemas ou módulos especializados, o que pressupõe uma divisão do sistema original em vários subsistemas, criando uma estrutura hierárquica complexa onde as relações entre os subsistemas concentram a funcionalidade do sistema global integrado (Silva 1998).

A Figura 1 representa as relações de vários subsistemas em um sistema de informação de administração de materiais, para o processo de fabricação de blank soldado. É possível identificar várias interdependências entre o Estoque, que indica a existência do material, com Compras, Vendas, Contabilidade, Produção e Qualidade, no que tange os materiais necessários para fabricação e entrega dos produtos finais e seus custos respectivos; a área Tributária com Compras, Vendas, Contabilidade no que diz respeito os impostos devidos e suas contabilizações; Compras com Contabilidade, referente ao pagamento dos fornecedores; Vendas e Contabilidade, para recebimento dos clientes. Esses relacionamentos determinam o tipo de implementações possíveis e, em outros casos mais complexos, podem determinar inclusive a funcionalidade e exequibilidade do projeto. A relação do sistema com os subsistemas é tal que a funcionalidade pretendida deve ser consequência da integração dos subsistemas. A complexidade do sistema aumenta pois cada subsistema possui um domínio específico com relações e restrições particulares, e, para cada inter-relação podem possuir diferentes funcionalidades, flexibilizando assim o sistema total. O conhecimento dos sistemas também ser refinados, determinando um modelo interativo capaz de poder reconhecer novas inter-relações entre subsistemas e ampliar a integração entre eles.

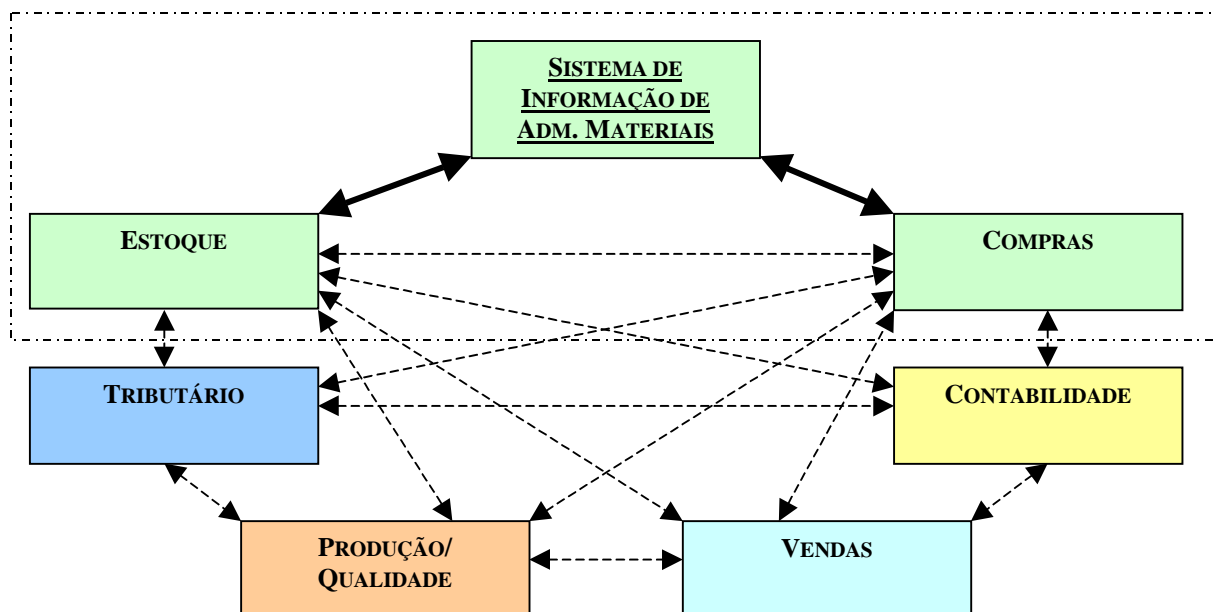


Figura 1. Esquema das inter-relações no sistema de informação de administração de materiais para o processo de fabricação de blank soldado.

Os sistemas automatizados podem ser representados através de três modelos, a saber, relacional, que apresenta as relações entre os subsistemas, funcional, o qual descreve as especificações funcionais dos sistemas e subsistemas, e dinâmico, responsável pelo processo a ser implementado (Tavares e Silva 1997). A Figura 2 apresenta uma solução para representação de cada um desses modelos separadamente.

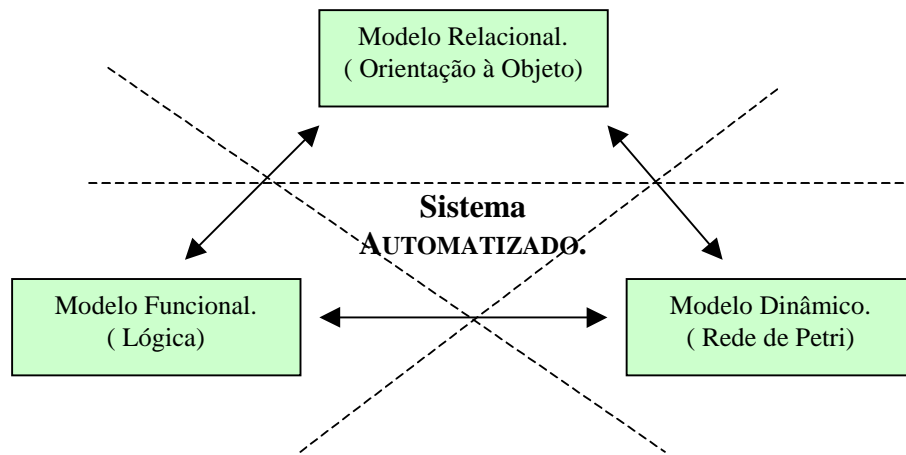


Figura 2. Representação de Sistemas Automatizados.

3. REPRESENTAÇÃO AV/AO

O modelo proposto para a modelagem conceitual de sistemas automatizados, e que pode também ser utilizado até o projeto detalhado é baseado em objetos abstratos AV/AO, que por sua vez foram adaptados do modelo ADV/ADO (Cowan e Lucena 1995). O *Abstract Data View* (ADV) foi criado para especificar claramente e formalmente a separação de funcionalidade (“*separation of concerns*”) entre uma componente de software e o seu domínio de aplicação. Isto é particularmente importante quando a componente de software se destina a vários usuários, todos com características específicas e utilizando a mesma componente com diferentes propósitos. Essa aproximação para especificação de interfaces claramente separa os componentes de aplicação dos demais dentro de uma abordagem cliente/servidor. Assim, os modelos das componentes de aplicação são chamados de *Abstract Data Objects* (ADO), que são projetados para minimizar o conhecimento do ambiente em que eles são usados e podem ser melhor reutilizados (Gray 1993).

A diferença entre a proposta de Cowan e Lucena e a que apresentamos neste artigo vai além da supressão do termo “Data”, característico das componentes de *software*. O modelo AV/AO, contém uma abordagem em lógica clássica para as relações entre componentes e para a conexão entre AV que é de fato a expressão da estrutura de uma rede de Petri do sistema integrado. Tem também uma proposta específica de análise de consistência entre cada modelo lógico parametrizado, AO, e seu respectivo comportamento AV. Apesar do formalismo proposto ser simples, é o suficiente para a maior parte dos projetos de design em engenharia, principalmente para os sistemas cujo processo de integração está intimamente associado a sistemas de controle supervisorio, como mostraremos em um exemplo adiante.

A estruturação do projeto está associada à abordagem orientada a objetos (ambos AV e AO são de fato objetos (Bittencourt 1996, Chan e Trinter 1993 e Panton et al. 1993). Além de responder pela parte funcional da abordagem, esta característica é um indicativo de que há uma boa possibilidade de reutilização de projetos representados por AV/AO, uma vez que os melhores resultados para a reutilização de projetos está associada a métodos orientados a

objetos. A relação entre AV e AO não é simétrica, uma vez que várias instâncias de AV podem estar associadas a um mesmo AO de forma a criar diferentes pontos de vista ou funcionalidades. Este relacionamento *muitos - para - um* significa que cada instância de AV deve ser coerente com o AO associado, o que é denominado de coerência vertical. Implica também que todas as instâncias dos AV devem ser consistentes entre si, o que é chamado de coerência horizontal.

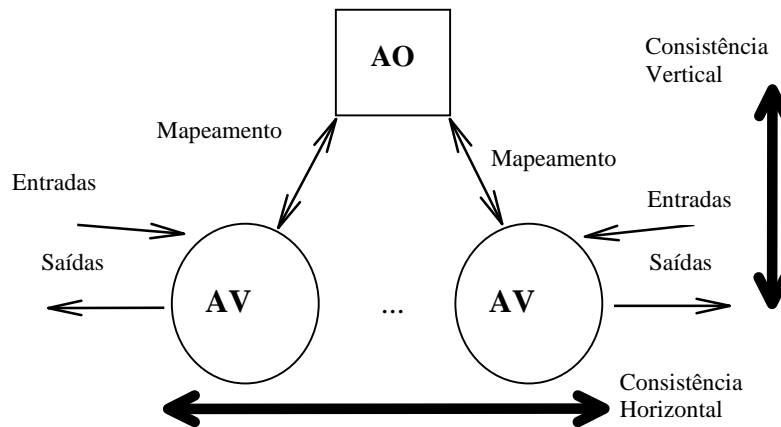


Figura 3. Relação *muitos - para - um* e propriedades de coerência do modelo AV/ AO.

O conceito da consistência garante a abstração correta de componentes de projeto e é fundamental no processo de reutilização, uma vez que as componentes reutilizadas devem ser adaptadas antes de serem incluídas em um novo projeto (Dieter 1991, Gray 1993, Medland 1992, Panton et al. 1993 e Reich 1995).

Nesse modelo, tanto o AV como o AO podem ser refinados, e a consistência horizontal e vertical deve ser mantida entre eles (Silva 1998).

Com um modelo seguindo a metodologia a ser indicada, consegue-se um sistema que armazena ao mesmo tempo o objeto projetado, AO, e possibilite a busca de objetos numa base de dados através de especificações por meio de AV.

Pode-se iniciar o projeto com as características funcionais, sem necessariamente especificar de início as características do objeto a ser projetado. Uma vez identificado um objeto a ser reutilizado (e as inter-relações do novo projeto com ele) através de AV, esse pode ser “integrado” ao projeto pelo AO correspondente, depois de verificada a consistência horizontal. Tudo se passa como esse novo AO fosse um elemento de base para uma composição (Silva 1998).

O processo de reutilização empregado é uma versão simplificada do método baseado em metáforas descrito em (Silva 1992).

Conceitualmente as relações de consistência possuem definições formais, a saber, a vertical indica que, caso exista um mapeamento de entradas e saídas de AV para um determinado AO, este deverá deduzir toda e qualquer entrada em uma saída. A horizontal, formaliza que, caso duas entradas indiquem a mesma saída de um AO, as entradas devem ser idênticas, gerando unicidade. A Figura 4 apresenta uma tela do RLOG (Liu 1998), um banco de dados orientado à objeto dedutivo da Universidade de Regina do Canadá, contendo as regras de consistência do modelo AV/AO.

```

Rlog[26]> open avao
Rlog[27]> display
There are the following types available:
Type Dados: String(16);

There are the following relations available:

Relation AO 's Attributes: (
Nome: String(16);
CjDados: Set.It has not maximal limitation
Type Dados:
)
The keys are the following attributes:
Nome
)

Relation AV 's Attributes: (
Entrada: Type Dados;
Saida: Type Dados;
AO_En: String(16);
AO_Sa: String(16);
The keys are the following attributes:
Entrada Saida AO_En AO_Sa
)

Relation AO 's facts:
Rules available are:
AV_En_Sa_AOEn_AOSa :- AO(AOEn,<En>), AO(AOSa,<Sa>)
AV_En1_Sa_AOEn_AOSa :- AV_En2_Sa_AOEn_AOSa, AO(AOEn,<En1>), AO(AOSa,<Sa>), _En1=_En2

```

Figura 4. Regras de consistência do modelo AV/AO no RLOG.

4. ADMINISTRAÇÃO DE MATERIAIS NA FABRICAÇÃO DE BLANK SOLDADO

A área de controle de sistemas de manufatura e CIM [*Computer Integrating Manufacturing*] apresentam soluções para o processo do ponto de vista operacional, entretanto, dados gerenciais são segregados. Por outro lado, os sistemas ERP [*Enterprise Resource Planning*], amplamente utilizados na atualidade, auxiliam os processos de administrativos internos e fornecem dados gerenciais, todavia, sem conexão direta com a área de controle e supervisão da manufatura. Através da aplicação do modelo AV/AO é possível modelarmos sistemas CIMM [*Computer Information Manufacturing Management*] pois, por meio da separação dos domínios e mapeamento de suas interfaces, é possível relacioná-los e os integrar num mesmo sistema.

O exemplo a ser apresentado advém de uma empresa siderúrgica real, que possui um sistema ERP instalado. O processo se caracteriza pela terceirização da fabricação do blank soldado, pois a empresa foca sua produção na fabricação de aço, mas tem interesse estratégico no mercado de blank soldado. Como exemplo de um blank soldado complexo tem-se a Figura 5. Trata-se da base do painel lateral externo do veículo do projeto ULSAB – *Ultra Light Steel Auto Body*, a qual reúne três diferentes graus de resistência mecânica, além, de cinco diferentes espessuras (Jaroni et al. 1997, Batalha et. al. 1999, Pereira et al. 1999 e Picinini Filho et al. 1999). Na Figura 6 apresenta-se esquematicamente a fabricação do Blank Soldado.

O processo real inicia-se através de uma ordem de venda de blank soldado para o cliente final. Nessa ordem já está especificado os materiais que irão compor o blank, bem como as quantidades e a data de entrega do produto. A área responsável pela compra do produto verifica, então, a existência de matéria prima, ou seja, bobina, no estoque para que seja cortada e soldada através da máquina de solda à laser pelo fornecedor. Caso não hajam bobinas necessárias para criação do lote de blank, é necessário criar uma solicitação de transferência de bobinas entre a matriz e a filial. Essa solicitação de transferência, por meio de

uma interface com o sistema produtivo, inicia a produção das bobinas na matriz. Terminada a fabricação, a bobina é pesada e embarcada para a filial, juntamente com uma nota fiscal de simples remessa. Ao chegar na filial é necessário verificar o peso do material, dar entrada em estoque e registrar a nota fiscal. As bobinas a serem processadas em blank, deverão ser remetidas ao fornecedor através de uma nota fiscal de beneficiamento. Uma ordem de compra de *blank* soldado é enviada ao fornecedor, informando qual *blank* soldado, em qual quantidade, qual preço e quando deverão ser entregues à filial.

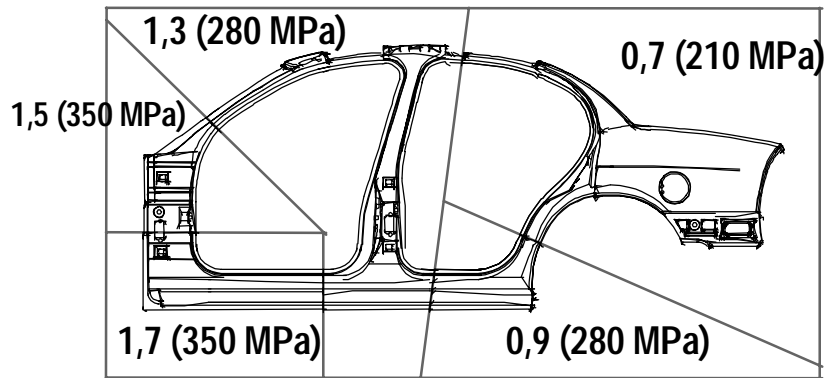


Figura 5. Painel lateral externo. - “*Tailor blank*” soldado a Laser

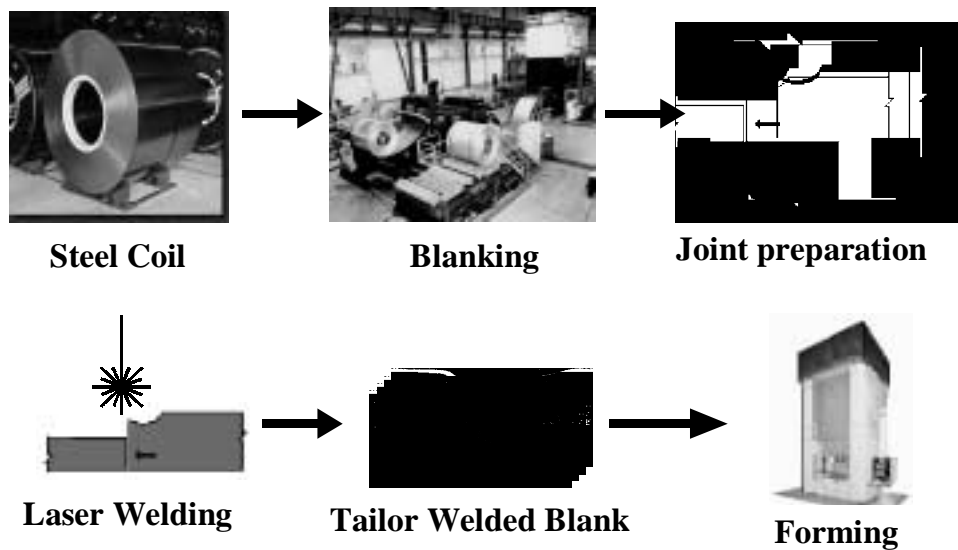


Figura 6. Processo de Fabricação do *Blank* Soldado.

No beneficiamento, as bobinas passam pelas operações de apara lateral, corte, solda e estampagem para, enfim, serem embalados e se tornarem o produto final especificado pela ordem de compra. No recebimento do *blank* soldado pela filial, dá-se baixa no estoque de bobina disponibilizada ao fornecedor, gera-se estoque de *blank* e se verifica o montante da fatura do fornecedor com relação à quantidade recebida e o preço indicado na ordem de compra.

Os processos básicos de Administração de Materiais do sistema ERP existente são: Processo de Compra que compreende Requisição de Compra, Solicitação de Cotação, Contrato Básico, Pedido/ Ordem de Compra, Entrada de Mercadoria e Verificação de Fatura; o Processo de Transferência que se separa em Requisição de Compra, Pedido, Entrada de Atividade/ Registro Fiscal; Processo de Administração de Materiais que se separa em Entrada de Mercadorias, Transferência de Mercadorias, Saída de Mercadorias, vinculados ou não à pedidos de compra e/ ou ordens de vendas.

O sistema implantado possui ordem de venda desvinculada da ordem de compra e ordem de compra desvinculada do pedido de transferência, pois apenas integra os passos internos dos processos administrativos principais, desprezando as possíveis inter-relações desses processos. Isso faz com que, operacionalmente, seja necessário criar ordens de compra em função do programa de remessas do cliente, bem como gerar pedidos de transferência em função da análise do estoque de bobinas e do programa de remessas fora do sistema ERP instalado.

Para otimizar esse processo se utilizará o modelo AV/AO como descrito em Tavares, Silva e Batalha 2000. Primeiramente é necessário identificar os AO's principais, a saber, AO_{Cliente}, AO_{Companhia} e AO_{Fornecedor}. Cada AO possui um domínio específico que se relaciona com um ou mais AO's por meio dos AV's. Na modelagem AV/AO não é necessário detalhá-lo, podendo apenas indicá-los e os tratar através de AV's, como é o caso dos AO_{Cliente} e AO_{Fornecedor}. A Figura 7 apresenta cada AO representando um domínio diferente e se relacionando com todos os outros domínios através dos AV's representados por setas unidirecionais.

Nesse caso, o AV₁ e o AV₄ representam as informações trocadas entre o Cliente e a Companhia. Optou-se por representá-los separadamente pois tratam de diferentes tipos de dados, a saber, o AV₁ informa o programa de entrega de blank do cliente e os pagamentos referentes aos recebimentos ocorridos, enquanto o AV₄ trata do envio do blank da Companhia ao Cliente, junto com a sua Nota Fiscal. Para efetuar a venda, esse blank soldado deve ser fabricado. Nesse caso, o AV₂ lida, respectivamente, com o envio de bobina a beneficiar e Nota Fiscal, pedidos de compra de *blank* e pagamento do fornecedor; enquanto o AV₃ representa o retorno do blank produzido, sua Nota Fiscal e retorno de sucata..

Pode-se refinar esse esquema separando o AO_{Companhia} como sendo dois AO's referentes aos AO_{Matriz} e AO_{Filial} com dois AV's correspondente ao relacionamento entre eles.

Assim sendo, o AV_J refere-se ao Pedido de Transferência de bobina do AO_{Filial} para o AO_{Matriz}, e o AV₁ diz respeito à remessa de bobina e sua nota fiscal de transferência. A consistência a ser cumprida é se gerar pedidos de transferência em função dos programas de remessa existentes advindos do AV₁. O envio de bobinas do AV₁ deve ser vinculado ao pedido de transferência indicado no AV_J.

As interfaces entre as áreas devem ser consistentes com respeito a todo o sistema. O programa de remessas do cliente é comunicado à área de Vendas, a qual informa à Compras quais os produtos devem ser adquiridos do fornecedor. Para se iniciar o serviço de corte a área de compras precisa verificar o estoque da(s) bobina(s) que compõe(m) o *blank*. Caso não haja bobina, a área de compras inicia o processo de transferência por meio de um pedido de transferência à matriz.

Ao se aplicar as regras de consistência, constata-se que para se criar um pedido de corte de chapa é necessário existir um programa de remessa e uma remessa de beneficiamento com nota fiscal. O pagamento dos serviços realizados só ocorre após a entrega do *blank*, nota fiscal dos serviços de corte, referenciados a um pedido de corte. O pagamento pelo produto entregue ao cliente ocorre após a remessa e faturamento do mesmo. Nos casos de um novo blank soldado, é necessária uma interação entre os processos de fabricação de *blank*, bobina e do processo de venda, para os testes e ensaios de viabilidade.

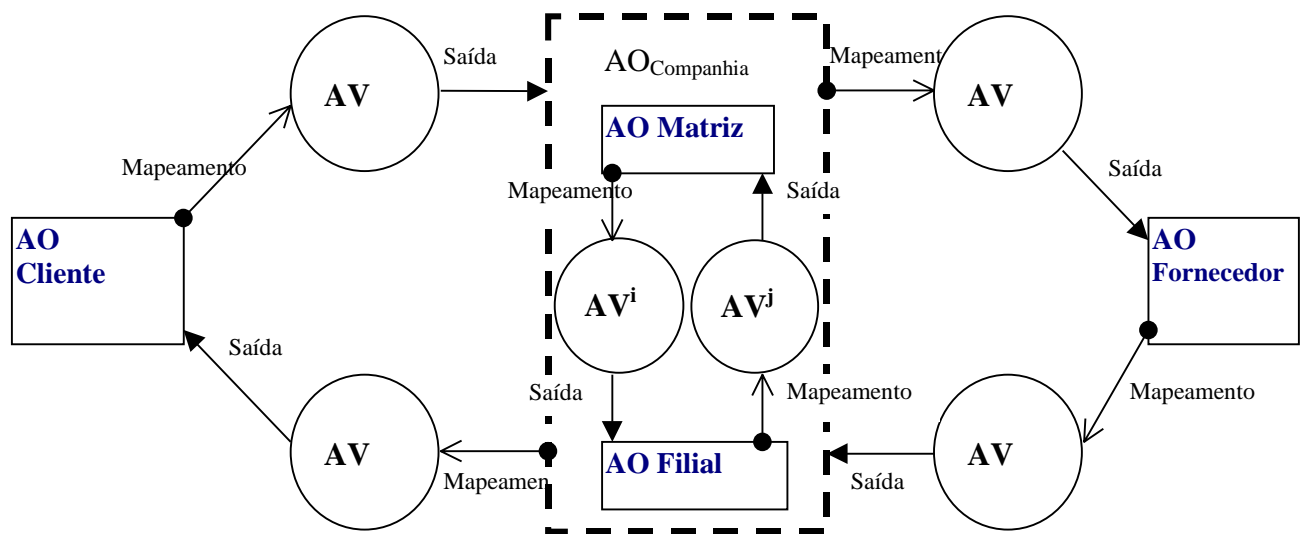


Figura 7. Representação do processo de administração de materiais modelada por AV/AO.

5. CONCLUSÃO E PERSPECTIVAS

Apesar da empresa siderúrgica possuir um processo automatizado, com a utilização do sistema ERP, esse processo apresenta falhas que podem acarretar grandes prejuízos junto ao cliente final, quando é produzido um *blank* com especificação, quantidade, preço e prazo diferente do informado pelo fato de não haver um vínculo direto entre o processo de vendas e compras.

Através do modelo AV/AO aplicado a esse caso, pode-se mostrar um sistema otimizado, o qual verifica a consistência da base existente e automaticamente processa pedidos de compra de *blank* e pedidos de transferência de bobinas de um dado programa de entrega do cliente.

6. BIBLIOGRAFIA

- Batalha, G. F. et al. 1999. Proc II Workshop FINEP/RECOPE - Desafios, Experiências e Expectativas sobre o Controle da Rugosidade em Chapas Metálicas para a Indústria Automotiva, 27 de abril de 1999, S. Paulo, FINEP/RECOPE, ISBN 85-86686-05-0.
- Bittencourt, G. 1996. Inteligência artificial: ferramentas e teorias. Campinas, Instituto de computação, Unicamp, 1996.
- Chan, D. K. C. e Trinter, P. W. 1993. Object comprehension: a query notation for object-oriented databases, Lecture notes in computer science, Springer, 752, pg. 55-72,
- Cowan, D. D. e Lucena, C. P. J. 1995. Abstract data view: An interface specification concept to enhance design for reuse, IEEE Trans. of software Engineering, v. 21, 3.
- Dieter, G. E. 1991. Engineering design: a materials and processing approach, 2 ed., Mc Graw Hill.
- Gero, J. S. 1989. Metamodel: An integrated modeling framework for intelligent CAD, Artificial Intelligence in Design, Computational Mechanisms Publications, Southampton, pg. 429-49.

- Gray, P. M. D. 1993 Knowledge reuse through networks of large KBS, Lecture notes in computer science, Springer, 752, pg. 55-72.
- Jaroni, U., Prange, W. e Schneider, C. 1997. Tailored blanks: In: Laser Assisted Net Shape Engineering 2 - Proc. of the 30th International CIRP Seminar on Manufacturing Systems Erlangen, 23 a 26 sept. 1997, Ed. CIRP-WGP-WLT, 61-8, ISBN 3-87525-090-7.
- Levesque, H. J. e Mylopoulosm J. 1996. Na overview of knowledge respresentation, On conceptual modeling, Springer Verlag, pp. 3-17.
- Liu, M. 1998 Relationlog: a Typed Extension to Datalog with Sets and Tuples, The Journal of Logic Programming, 1-30.
- Medland, A. J. 1992. The computer-base design process. 2 nd ed., Chapman & Hall, London.
- Panton,W.N., Al-Qaimani,G., Doan,K. 1993 On interface objects in object-oriented databases, Lecture notes in computer science, Springer, 752, pg. 55-72.
- Pereira, J. F. B., Andrade, S. L. e Rosa, L. K. 1999. Projeto ULSAB - Ajudando a Construir o Carro do Futuro, In: Proc II Workshop FINEP/RECOPE - Desafios, Experiências e Expectativas sobre o Controle da Rugosidade em Chapas Metálicas para a Indústria Automotiva, 27 de abril de 1999, S. Paulo, EPUSP/FINEP/RECOPE, ISBN 85-86686-05-0. Pp. 20-31.
- Picinini Filho, M., Sampaio, A , Azambuja, S., Campbel, C., Carvalho, C., Suzuki, L. , Santos, J. e Venturinnen, C.. Desenvolvimento Pioneiro de Blank soldado na América Latina: O projeto da CSN para a GMB e a FORD. Ibid, pp. 32-44.
- Silva, J. R. 1998 Interactive Design of Integrated Systems, In.: Intelligent Systems for Manufacturing: Multi-Agent Systems and Virtual Organizations, Camarinha-Matos, L., Afsarmanesh, H., Marik, V. (eds.), IFIP, -KAP, Londres, pp. 567-78.
- Silva, J. R. 1994. An object-oriented approach to design of flexible manufacturing systems, IFIP, pg. 91- 106
- Silva, J. R. 1992 Uma formalização para o processo de design baseado em metáforas: suas aplicações em automação de sistemas a eventos discretos, Tese de doutorado, Universidade de São Paulo, São Paulo.
- Tavares, J. J. P. Z. S. e Silva, J. R. 1997. Fusão Entre Objetos e IA na Modelagem e Design de Sistemas Automatizados, III Simpósio Brasileiro de Automação Inteligente, pg. 359-64.
- Tavares, J. J. P. Z. S. , Silva, J. R. e Batalha, G. F. 2000. Towards the Formalization of Information System for Material Management in Manufacturing Process. Proc. The Third World Congress on Intelligent Manufacturing Process & Systems, Massachussets Institute of Technology, Cambridge, MA-EUA, June 28-30, 2000 (Aceito p. publicação e apresentação)
- Reich, Y. 1995. The study of design methodology, J. of mechanical Design, 117, pg. 211-14.

UMA METODOLOGIA PARA GERAÇÃO DE TRAJETÓRIAS DE MANIPULADORES NO ESPAÇO CARTESIANO

Altamir Dias
Henrique Simas
Julio Feller Golin
Raul Guenther

Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica, Laboratório de Robótica, Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil. E-mail: simas,golin@lcmi.ufsc.br; altamir@emc.ufsc.br

Resumo

O presente trabalho apresenta um estudo realizado para geração de trajetórias sobre uma superfície determinada no espaço cartesiano. Curvas B-Splines são utilizadas para geração do caminho sobre a superfície e algoritmos matemáticos são usados para definir uma velocidade constante do efetuador final. O trabalho descreve a teoria envolvida para o cálculo teórico da trajetória e uma experimentação prática realizada com um manipulador industrial.

Palavras chaves : Robótica, controle de velocidade, algoritmos de controle

1.INTRODUÇÃO

Para realização de tarefas de manipuladores no espaço cartesiano é necessário a definição das coordenadas de posição e velocidade das juntas para os controladores. Estas referências podem ser dadas no espaço de juntas ou espaço cartesiano.

O controle no espaço de juntas não garante a aquisição de parâmetros da curva, tais como derivadas e raio de curvatura, entre outros. Para tarefas mais complexas há a necessidade de se formular a trajetória no espaço cartesiano – e através de algoritmos de cinemática inversa será possível obter os referidos parâmetros.

Trabalhos já apresentados abordam este problema aplicados a geração de trajetórias para máquinas ferramentas (Sarma & Dutta, 1997). Outras aplicações são na geração de conjuntos de caminhos paralelos em superfícies livres (Suresh & Yang, 1994, Dias, 1999a, 1999b).

O presente trabalho tem por objetivo apresentar uma metodologia para a geração de coordenadas de juntas de um manipulador, considerando especificações desejadas em um caminho parametrizado sobre uma superfície definida no espaço cartesiano.

Neste trabalho são analisados dois casos: no primeiro um caminho é definido sobre uma superfície plana onde o manipulador executa uma seqüência de circunferências. Na segunda análise, um caminho helicoidal é definido sobre uma superfície cilíndrica.

Para experimentação prática será utilizado um manipulador industrial SCARA. Para isto será realizado um levantamento das limitações físicas e tecnológicas do manipulador e do algoritmo de geração das trajetórias.

No corpo do trabalho são apresentadas as características das B-Splines, sua formulação e propriedades importantes para o cálculo de caminhos de tarefas, a técnica de discretização de um caminho contínuo e a aplicação destas teorias na determinação de perfis de referência para controle do manipulador sobre a trajetória desejada.

2. CURVAS B-SPLINE

2.1 Definição

As curvas B-spline são segmentadas e definidas no espaço Cartesiano a partir de um conjunto de pontos no espaço, chamados de vértices de controle \mathbf{V}_i . Estes vértices formam um polígono convexo – chamado polígono característico – que limita a curva. B-splines podem ser representadas e obtidas de diferentes formulações, como a fórmula recursiva de Cox e de Boor, ou por diferenças finitas, ou ainda por condições de continuidade. Neste trabalho trata-se somente de B-splines cúbicas parametrizadas num intervalo unitário $u \in [0,1]$, utilizando a representação matricial segundo (Qiulin 1987). Nesta representação, um segmento de curva B-spline cúbica $\mathbf{r}_i(\mathbf{u})$ é dada como uma combinação linear de quatro vértices de controle \mathbf{V} e das funções de ajuste $N_{4,j}(u)$, $j = 0, 1, 2, 3$. As funções de ajuste determinam o “peso” de cada vértice ao longo da curva e são sempre as mesmas para qualquer tipo de segmento de curva B-spline. Então, um segmento de curva B-spline pode ser definido por:

$$\mathbf{r}_i(\mathbf{u}) = [N_{4,0}(u) \ N_{4,1}(u) \ N_{4,2}(u) \ N_{4,3}(u)] \begin{bmatrix} \mathbf{V}_i \\ \mathbf{V}_{i+1} \\ \mathbf{V}_{i+2} \\ \mathbf{V}_{i+3} \end{bmatrix} = \mathbf{U} \mathbf{M}_b \mathbf{V} . \quad (1)$$

onde $\mathbf{U} = [1 \ u \ u^2 \ u^3]$ e:

$$\begin{aligned} N_{4,0}(u) &= \frac{1}{3!}(1 - 3u + 3u^2 - u^3) \\ N_{4,1}(u) &= \frac{1}{3!}(4 - 6u^2 + 3u^3) \\ N_{4,2}(u) &= \frac{1}{3!}(1 + 3u + 3u^2 - 3u^3) \\ N_{4,3}(u) &= \frac{1}{3!}(u^3) \end{aligned} \quad \mathbf{M}_b = \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \quad (2)$$

Estendendo a formulação de caminhos tridimensionais por curvas B-splines, pode-se formular superfícies parametrizadas ($\mathbf{r}(\mathbf{u}, \mathbf{w})$) a partir de uma nuvem de pontos no espaço cartesiano.

Desta forma, uma superfície B-spline é definida de acordo com a equação

$$\mathbf{r}_i(\mathbf{u}, \mathbf{w}) = \mathbf{U} \mathbf{M}_b \mathbf{V} \mathbf{M}_b^T \mathbf{W}^T \quad (3)$$

ou

$$\mathbf{r}(\mathbf{u}, \mathbf{w}) = \sum_{K=0}^3 \sum_{L=0}^3 N_{4,k}(u) N_{4,L}(w) \mathbf{V}_{K,L} \quad (4)$$

2.2 Propriedades

As curvas e superfícies B-splines, independente de sua ordem, apresentam importantes propriedades geométricas. A primeira delas é o controle local. Como cada segmento de curva B-spline cúbica é determinada por quatro vértices consecutivos $\mathbf{V}_i, \dots, \mathbf{V}_{i+3}$; a mudança de um vértice não exige o “recálculo” de toda a curva, mas dos quatro segmentos que dele dependem.

Esta propriedade foi a mais importante para a escolha do algoritmo B-spline para a formulação da trajetória. Isto porque as curvas de Bezier, bem como as curvas Splines, apesar de também ser possível fornecerem parâmetros para o cálculo de propriedades das curvas, como as derivadas (tangentes, normais e binormais) da curva, não se adequaram ao problema tratado neste trabalho.

A formulação matemática a partir de curva B-splines resultou numa curva suave para o caminho da tarefa, sem a presença de oscilações indesejadas e sem excessivo esforço computacional no cálculo de seus parâmetros.

3. METODOLOGIA PARA GERAÇÃO DE TRAJETÓRIA NO ESPAÇO CARTESIANO APLICADO AO MANIPULADOR INDUSTRIAL

3.1 Manipulador SCARA INTER

Para executar as experimentações de laboratório, foi empregado um manipulador industrial de configuração SCARA. Este manipulador possui arquitetura aberta, o que permite acesso ao seu ambiente de programação.

Através de um pós processamento, os arquivos de dados em formato padronizado ASCII podem ser enviados ao controlador do manipulador para tratamento das informações contidas.

Os pontos a serem interpolados para definição da trajetória são definidos livremente no espaço cartesiano ou sobre uma superfície parametrizada. Uma vez definidos, utilizou-se a formulação por B-splines para gerar a curva representativa do caminho e interpolar outros pontos entre estes pontos dados, definindo assim, uma curva de trajetória desejada no espaço cartesiano.

A partir da curva de trajetória, é realizada a discretização deste caminho, pois o controlador do manipulador é digital. Esta discretização consiste em caminhar sobre a curva através de cordas de comprimento L , respeitando-se uma tolerância δ previamente definida. A tolerância δ é o desvio da curva que o efetuador dará, ao dar um passo linear seguindo a trajetória, na execução da tarefa. Procura-se maximizar os comprimentos de corda ao longo da curva, para que o tempo de execução da tarefa seja minimizado

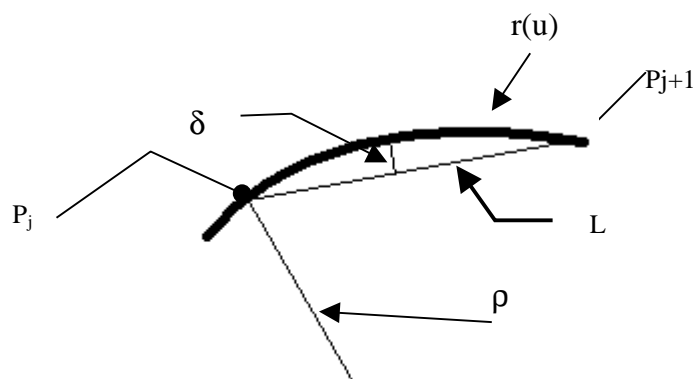


Figura 1. Discretização de um caminho contínuo

A Fig 1 ilustra as propriedades necessárias para o cálculo dos passos sobre a curva através da equação (5), onde ρ é o raio de curvatura local da curva no espaço cartesiano, δ a precisão para a execução da tarefa requerida e P é o conjunto de j pontos discretos sobre a curva.

Em (Qiulin, 1987) é apresentado um equacionamento para cálculo do passo L sobre uma curva parametrizada. Assim o passo L é mostrado na equação 5, como:

$$L^2=4\delta(2\rho-\delta) \quad (5)$$

Utilizando-se o software Matlab, implementaram-se rotinas para geração dos perfis de referência para as juntas do manipulador. Estas rotinas consistiram em montar matrizes de pontos para coordenadas das juntas. Estes pontos são determinados a partir de algoritmos de cinemática inversa do manipulador SCARA.

3.2 Limitações do manipulador SCARA

Na implementação e execução das rotinas Matlab, foram levantadas as limitações físicas e tecnológicas do manipulador SCARA, que são apresentadas a seguir:

- Limitações da orientação do efetuador final em $\pm 2,5$ rad, o que impede o manipulador de executar uma 1 volta completa em torno do seu eixo de orientação;
- Limitações de posição e velocidade (Golin *et al*, 1998);
- Limitações do controlador que possui um período de amostragem fixo (1ms).

Para executar as tarefas desejadas foram propostas as seguintes soluções:

- Fixar uma orientação em relação ao sistema de coordenada da base;
- Fixar uma posição do eixo de rotação do efetuador final;
- Implementar procedimentos matemáticos dentro das rotinas Matlab para evitando saturações de velocidade, saída da área de trabalho do manipulador e singularidades no Jacobiano.

4. EXPERIMENTAÇÃO PRÁTICA NO CONTROLE DE VELOCIDADE DO EFETUADOR FINAL SOBRE UMA TRAJETÓRIA NO ESPAÇO CARTESIANO

4.1 Descrição do problema

Utilizando-se as rotinas Matlab e considerando as limitações do manipulador, foram definidas duas superfícies para geração de diferentes trajetórias.

A primeira experimentação foi realizada sobre uma superfície paralela ao plano XY da base do manipulador. Nesta superfície foram definidos pontos de uma circunferência, cujo raio é de 7 cm e definida uma tarefa de fazer o manipulador percorrer uma trajetória sobre esta circunferência, realizando três voltas numa velocidade de 5 cm/s.

Na segunda experimentação foram amostrados pontos sobre uma superfície cilíndrica parametrizada no espaço cartesiano Este cilindro foi parametrizado em função do ângulo θ . Este conjunto de pontos definiu uma helicoidal com passo de 6 cm e raio de 7 cm. A velocidade definida para o efetuador percorrer a trajetória no espaço cartesiano foi de 5 cm/s.

4.2 Solução na execução da tarefa

Devido ao período de amostragem do controlador ser uma variável inteira, ou seja, de um 1 ms, é necessário um arredondamento dos tempos de cada passo. Este

arredondamento é devido à relação comprimentos de arco entre os pontos discretizados P e velocidade local de avanço desejada na trajetória não resultar num valor inteiro de tempo em milissegundos. Assim um passo pode ter um tempo de duração de 30,5 ms por exemplo. Para definir os valores de tempos foram utilizadas três metodologias.

A primeira solução foi arredondar os períodos de tempo e dividir cada segmento de trajetória entre os pontos P, em T-1 segmentos, onde T é a parte inteira do tempo em ms para que o manipulador percorra o passo em questão.

Esta solução resultou num perfil de referência das velocidades das juntas excessivamente oscilatório, gerando vibrações indesejadas na execução da tarefa.

Uma segunda implementação foi proposta para evitar as oscilações na velocidade das juntas. A solução foi aproximar o arco que une dois pontos consecutivos de P por uma reta e em seguida dividi-la em T – 1 segmentos como na solução anterior. Nesta proposta percebeu-se que o arredondamento do período de tempo T, de evolução entre os pontos P, tem influência no desempenho da execução da tarefa, pois também gerou excessivas variações na velocidade.

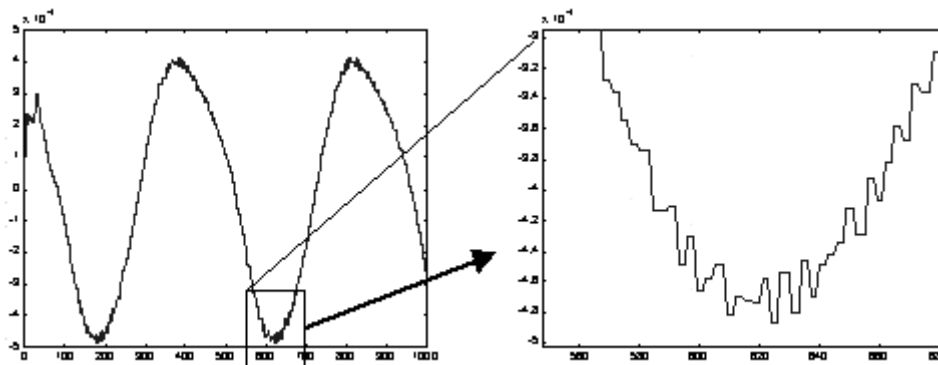


Figura 2. Resultado do perfil de velocidade para a junta 1 entre os tempos 550 e 680 ms

Para resolver o problema de vibrações e arredondamentos de tempo foi implementada uma terceira proposta. Esta solução consiste em reajustar o comprimento de arco entre dois pontos P através de um algoritmo de biseção. Este algoritmo busca, entre dois arcos consecutivos, um arco que aproxime a relação entre o período do passo, sem arredondamentos e a velocidade de avanço sobre o arco.

Na Fig 2, tem-se no detalhe o perfil de referência velocidade da junta 1 no período de tempo entre 550 e 680 ms.

Nesta solução o perfil de velocidade foi mais suave, apesar de pequenas oscilações ainda presentes. Na experimentação prática o manipulador se comportou de forma satisfatória, sem apresentar vibrações ou oscilações na evolução do efetuador final sobre a trajetória.

5. RESULTADOS EXPERIMENTAIS

As soluções discutidas acima foram implementada em rotinas Matlab para as duas tarefas propostas – circunferência sobre superfície plana e uma curva helicoidal sobre uma superfície cilíndrica. Nesta seção são apresentadas as experimentações utilizando a terceira proposta.

5.1 Circunferência sobre o plano XY

Nas realizações desta tarefa utilizou-se controlador PD com ajustes 400 para o ganho proporcional e 1 para o ganho derivativo.

A Fig 3 apresenta o perfil de referências para o controle da posição e velocidades das juntas, calculados pela rotina Matlab. Na Fig 4 é apresentado o erro de posição entre o valor real de posição no manipulador e a referência.

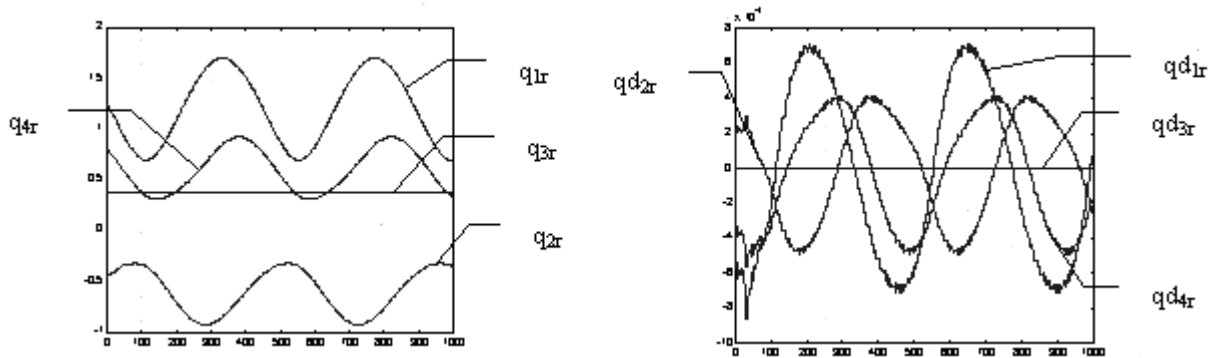


Figura 3. Referência de posição e velocidades das juntas para o controle do manipulador SCARA

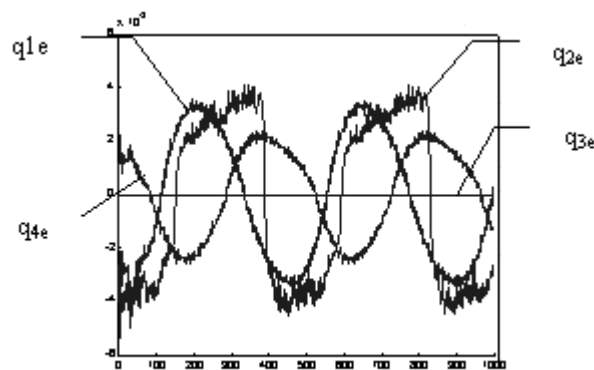


Figura 4. Erro de posição das juntas para a primeira tarefa

5.1 Execução de tarefa seguindo um caminho helicoidal sobre um cilindro

Para a tarefa da trajetória helicoidal foi ajustado o controlador com os mesmos parâmetros da tarefa anterior.

A Fig 5 apresenta o perfil de referências para o controle da posição e velocidades das juntas. Na Fig 6, o erro de posição das juntas.

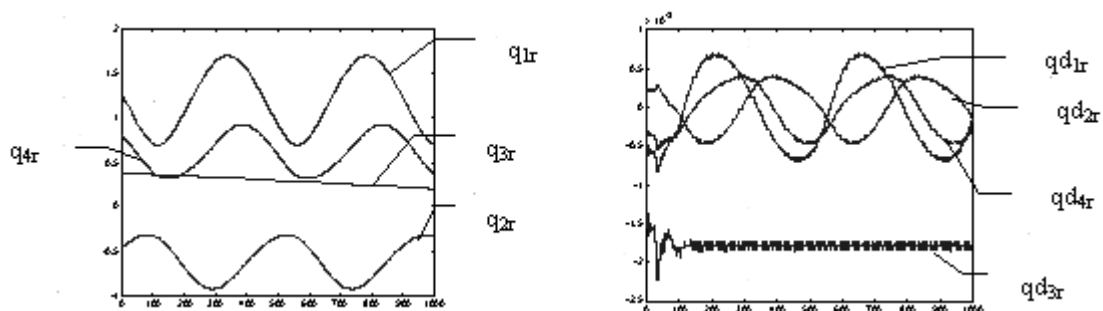


Figura 5. Referência de posição das juntas para o controle do manipulador SCARA

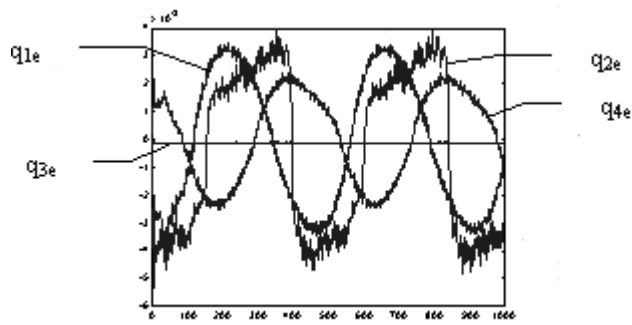


Figura 6. Erro de posição das juntas para a segunda tarefa

6. CONCLUSÃO

Este trabalho possibilitou a aplicação de uma metodologia à programação de tarefas no espaço cartesiano de um manipulador industrial. As trajetórias foram obtidas a partir de curvas B-spline modeladas sobre superfícies parametrizadas segundo suas propriedades geométricas, constituindo o caminho para o efetuador final. Considerou-se aqui o problema de trajetórias com velocidade constante no espaço cartesiano.

Para a experimentação prática destes resultados foram implementados métodos para discretização da trajetória cartesiana e às equivalentes trajetórias de juntas. Procedimentos para cálculos de velocidades de junta foram testados conforme as limitações do manipulador.

Os procedimentos implementados e testados no manipulador resultaram em curvas de referência de posição e velocidade suaves – transformadas para espaço das juntas. Para isso diferentes modelos de discretização foram estudados para se atender as características do controlador.

Como perspectivas para este trabalho pretende-se utilizá-lo no contexto do projeto Roboturb (UFSC-COPEL-PADCT), que está desenvolvendo um manipulador para recuperação de superfícies erodidas de turbinas hidráulicas para o processo de geração de trajetórias. Também pretende-se utilizar num estudo mais avançado, na modelagem de superfícies livres B-splines parametrizadas, nas quais podem realizadas tarefas mais complexas – por exemplo soldagem e pintura.

7. REFERÊNCIAS

- Angeles, Jorge ; 1997, “Fundamentals of Robotic Mechanical Systems”, Springer-Verlag.
- Chen, Y.D. ; Ni, J. and Wu, S. M. ; 1993, “Real-Time CNC Tool Path Generation for Machining IGES Surfaces”; Journal of Engineering for Industry, vol 115, pp-480-486.
- Dias, Altamir, Toledo, Leonardo B de, Deiseroth, Michael. , 1999, “ A CAD/Robotics System to "Off-Line" Programming of Industrial Robots.” , XV Congresso Brasileiro de Engenharia Mecânica, Aguas de Lindóia, SP.
- Dias, Altamir, Toledo, Leonardo B de, Deiseroth, Michael, 1999, “An off-line programming system for industrial robots”, In: FLEXIBLE AUTOMATION AND INTELLIGENT MANUFACTURING, Tilburg, the Netherlands. Proceedings of

the Ninth International Fair Conference. New York: begell house inc. v.1. p.835-846.

- Dragomatz, D. ; Mann, S. ; 1997 "A Classified Bibliography of Literature on NC Milling Path Generation" Computer-Aided Design, vol 29 No 3 pp. 239-247.
- Golin, J., Weihmann, L., Guenther, R. , 1998; "Manual do Usuário do Robô Inter"; documento do Laboratório de Robótica da UFSC.
- Qiulin, D. e Davies, B.J. , 1987; "Surface Engineering Geometry for Computer-Aided Design and Manufacturing", Ellis Horwood Limited.
- Sarma, R. and Dutta, D.; 1997, "The Geometry and Generation of NC Tool Paths"; Journal of Mechanical Design; vol 119, pp. 253-258.
- Suresh , D. C. H. ; 1994, "M Constant Scallop-height Machining of Free-form Surfaces"; Journal of Engineering for Industry, vol 115, pp- 253-259.

PROJETOS DE AUTOMAÇÃO DE SISTEMAS MECÂNICOS BINÁRIOS UTILIZANDO O GRAFCET

Vitor Ferreira Romano

Universidade Federal do Rio de Janeiro, Depto. de Engenharia Mecânica, Lab. de Robótica,
P.O. Box 68503, 21945-970 Rio de Janeiro, RJ, Brasil. E-mail: romano@serv.com.ufrj.br

Paulo Rogério Araújo Guimarães

Universidade Federal de Juiz de Fora, Colégio Técnico Universitário, Depto. De Mecânica,
R. Bernardo Mascarenhas, 1283, Juiz de Fora, MG, Brasil. E-mail: prag@mailctu.ufjf.br

Resumo

Este trabalho apresenta uma proposta metodológica a ser empregada na elaboração de projetos de automação de sistemas mecânicos binários, baseada na elaboração de um diagrama de função seqüencial denominado GRAFCET. Apresenta como exemplo um projeto conceitual de um alimentador automático de chapas para prensas de estamparia em que as tecnologias operativa e de comando baseiam-se essencialmente na lógica binária associada à álgebra de Boole.

Palavras-chave: Automação; GRAFCET; Diagrama de função seqüencial; Pneutrônica.

1. INTRODUÇÃO

Durante o processo de automação industrial pode-se ter a interação de componentes mecânicos, elétricos, eletrônicos, hidráulicos ou pneumáticos, gerenciados a partir de comandos lógicos. Esta interação de tecnologias busca uma solução otimizada para o problema que se apresenta, sendo que a cada aplicação deve-se enquadrar as tecnologias que melhor se adaptem.

Neste processo, surgem problemas na utilização simultânea destas tecnologias distintas. Diante disto, verificou-se a necessidade de se estudar métodos que favorecessem a melhoria da comunicação entre especialistas das diversas áreas envolvidas no processo de automação.

O desenvolvimento de ferramentas para análise e elaboração de projetos de sistemas automáticos busca a possibilidade de se executar todo o conceito e seqüência de funcionamento de um equipamento sem se ocupar da tecnologia a ser empregada. Neste contexto, as normas internacionais estabeleceram padrões gráficos para a representação de seqüências e funções lógicas.

Este trabalho propõe uma metodologia que facilite e otimize a elaboração de projetos de controle de sistemas mecânicos binários, auxiliando no processo decisório das tecnologias de comando a serem empregadas.

2. CENÁRIO DE APLICAÇÃO

Ao se propor um projeto de automação de um equipamento ou de um processo é preciso primeiro conhecer o cenário de aplicação do mesmo, para que a solução tecnológica a ser apresentada satisfaça plenamente os interesses da automação.

É preciso ainda, verificar se o sistema caracteriza-se como um sistema binário, ou seja, a cada etapa do processo, um elemento só pode assumir duas condições mutuamente exclusivas. Por exemplo, num processo de estampagem de metais, a prensa só deve ser acionada se a chapa estiver corretamente posicionada na ferramenta. No caso, a chapa está na posição correta ou não. Desta forma, classifica-se as condições de funcionamento como **binárias**. Então, pode-se dizer que o funcionamento do sistema segue a *lógica binária*.

3. PROJETO INTEGRADO

À medida que se conhece o problema a ser solucionado por uma automação, bem como as condições necessárias para seu funcionamento, inicia-se a concepção do projeto. A partir da concepção inicial do equipamento automático, define-se as tecnologias operativas e de comando do mesmo, além de elaborar-se um esquema de instalação e a descrição de seu funcionamento.

3.1 Tecnologia de Comando

Para definir-se qual tecnologia de comando empregar é preciso primeiro conhecer todas as etapas de funcionamento, assim como a complexidade do sistema. Por tratar-se de projetos que seguem a lógica binária, é possível elaborar-se todo o projeto lógico sem ocupar-se da tecnologia de comando a ser utilizada.

3.2 Tecnologia Operativa

Para automação do sistema mecânico proposto neste artigo, emprega-se atuadores pneumáticos, os quais atuam como comandos binários, uma vez que as grandezas envolvidas adquirem somente dois valores significativos, como por exemplo: o cilindro está **avançado** ou **recuado**. A utilização destes sistemas binários, associados à álgebra *booleana* proporciona a utilização de uma linguagem e metodologia unificadas para representação e projeto de comandos binários, propiciando a integração dos comandos pneumáticos a estruturas mais complexas de automação industrial. A definição da tecnologia operativa a ser empregada passa pela verificação das características do mecanismo, como cargas e velocidades envolvidas.

3.3 Esquema de Instalação

Definida a tecnologia operativa a ser empregada, parte-se para a elaboração da concepção construtiva do equipamento. Para isto, elabora-se um *esquema de instalação* do mesmo. Este esquema deve ser executado levando-se em consideração a seqüência de operações e as características e cursos a serem executados, de modo que a análise do funcionamento seja facilitada para o projetista. Nesta fase ainda não se deve ocupar com o dimensionamento estrutural e sim com as características funcionais.

A fig. 1 apresenta como exemplo o esquema de instalação de um alimentador pneumático de chapas, onde a manipulação das mesmas se dá pela sujeição à vácuo.

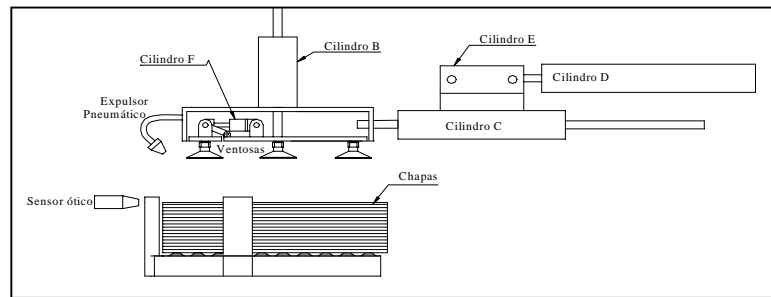


Figura 1. Esquema de Instalação de um Alimentador Pneumático de Chapas

3.4 Descrição do funcionamento

Baseados no esquema de instalação, elabora-se a descrição do funcionamento do mecanismo. A descrição deve ser sucinta e objetiva. Para o exemplo da fig. 1, o alimentador automático de chapas deve ter seu ciclo completado automaticamente na seguinte seqüência:

1. A partir de um sinal manual ou de um outro comando externo, inicia-se o ciclo com o avanço do cilindro B. O ciclo não deve ser iniciado sem que haja uma confirmação da existência de chapas no *magazine*.
2. Após tocar a chapa localizada no *magazine*, é aplicado vácuo ao conjunto de ventosas.
3. Cilindro B recua erguendo a chapa. Simultaneamente, o cilindro F é acionado flexionando a extremidade da chapa.
4. Expulsor emite o jato de ar para separação de chapas.
5. Cilindro F retorna à posição inicial.
6. Cilindro D avança, posicionando o carro alimentador em direção à prensa.
7. Cilindro C avança com a chapa em direção à matriz localizada na prensa, até que haja o posicionamento no sentido do eixo *x*.
8. Cilindro B avança e posiciona a chapa na ferramenta (eixo *y*).
9. Cilindro E avança para posicionamento transversal. (eixo *z*).
10. Cessa-se a aplicação do vácuo e as ventosas liberam a chapa.
11. Os cilindros B e E retornam simultaneamente.
12. Os cilindros C e D recuam simultaneamente.

3.5 Esquema de Montagem dos Atuadores

Baseado nas informações subtraídas do esquema de instalação e da descrição de funcionamento, elabora-se um esquema de montagem dos atuadores, associando a cada um dos atuadores os elementos de sinais responsáveis pela monitoração de seus estados. No exemplo da fig. 1, sendo o acionamento pneumático, elabora-se um esquema de circuito pneumático.

3.6 Representação Gráfica do Comando Seqüencial

A representação gráfica de um comando seqüencial tem por finalidade representar de uma forma sistemática a descrição do funcionamento do equipamento.

3.6.1 Diagrama Trajeto-Passo

O diagrama trajeto-passo apresenta a condição momentânea de cada atuador ao longo dos diversos passos que compõem o ciclo total do mecanismo. Por meio de linhas retas, indicamos se os atuadores permanecem recuados, avançados ou se estão em movimento num determinado instante do ciclo. Estes diagramas, usados tradicionalmente na pneumática e na eletropneumática são de simples elaboração e proporcionam uma visão clara da seqüência funcional do sistema. Porém, quando introduzimos uma série de condições para o funcionamento, como temporizações, saltos, funções simultâneas, este diagrama apresenta algumas limitações, tornando-se confuso e de difícil entendimento.

3.6.2 Diagrama de Função Seqüencial – GRAFCET

Na maioria dos casos, os projetos lógicos mecânicos são executados de forma intuitiva, empiricamente, apoiado basicamente em combinações lógicas padrões, onde o sucesso vai depender da experiência do projetista. Este método conduz normalmente a problemas no *start-up*, bem como a dificuldades de soluções de problemas durante manutenções.

Com o GRAFCET, as seqüências são descritas de acordo com um diagrama com símbolos normalizados que mostram claramente os controles a serem implementados. Então, através de técnicas simples como mostrado por Bollmann (1995), o diagrama de fluxo pode ser transformado em circuitos reais, sejam pneumáticos, hidráulicos ou em diagramas de contatos.

3.6.3 GRAFCET Descritivo

O diagrama de função seqüencial deve seguir a representação básica de um conjunto de retângulos que representam uma **etapa** de um comando seqüencial. Cada etapa deve seguir uma numeração seqüencial de identificação e opcionalmente uma descrição sucinta do mesmo. Os retângulos representativos das Etapas serão interligados por retas, ou **ligações de vínculo orientadas** (LVO), e transversalmente a essas LVOs pequenos traços indicam a **transição** entre as etapas. A cada transição estará associada uma condição lógica para prosseguimento da seqüência, chamada **receptividade**, como definido por Blanchard (1996). Ao lado de cada etapa, no campo **ações** faz-se uma descrição sucinta das ações associadas a esta etapa (fig. 2). No campo **a** indica-se uma propriedade do sinal que gera a ação, como exemplo “S” indica que o sinal deve ser memorizado ao longo de uma ou mais etapas. O campo **b** é reservado para a indicação do elemento de sinal que confirma a realização da ação.

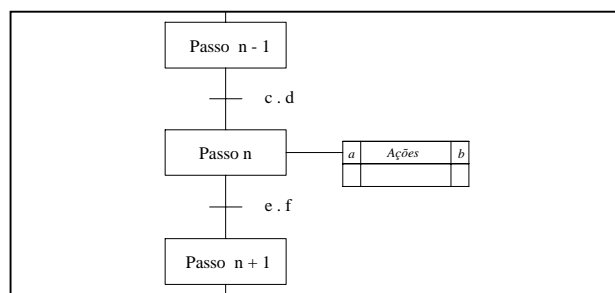


Figura 2. Bloco básico correspondente a uma etapa do GRAFCET, onde as equações booleanas *c . d* e *e . f* indicam a receptividade, ou condição lógica associada a cada transição

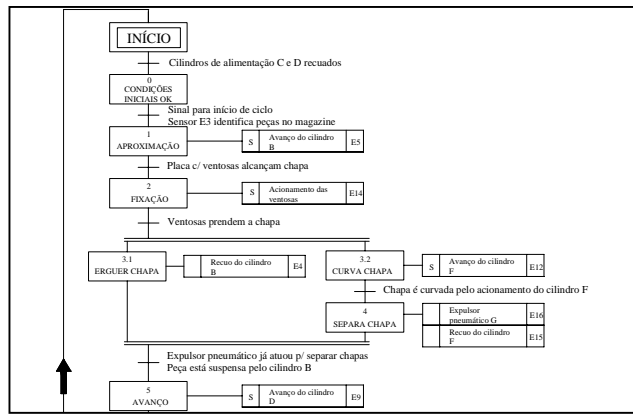


Figura 3. Detalhe do GRAFCET Descritivo

No GRAFCET descritivo insere-se a descrição de cada etapa na seqüência a realizar, além de associar-se a cada receptividade a descrição das condições lógicas para que ocorra a transição entre etapas. A fig. 3 apresenta o GRAFCET decriptivo para o exemplo considerado.

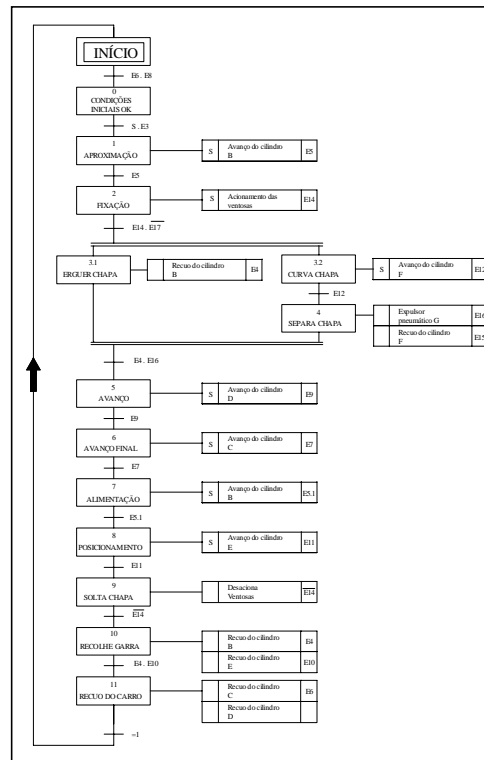


Figura 4. Gráfico de Função sequencial GRAFCET

3.6.4 GRAFCET

O GRAFCET descritivo facilita a intercomunicação entre as pessoas envolvidas no projeto, independente de sua área de formação, no entanto, para a execução técnica do projeto, necessita-se associar a cada condição lógica, um elemento de sinal. Daí, elabora-se outro diagrama (fig. 4) indicando os elementos de comando e sensores correspondentes às condições descritas no diagrama GRAFCET descritivo (fig.3).

3.7 Diagrama Lógico

De posse do GRAFCET, elabora-se o diagrama lógico, associando-se a cada etapa um módulo lógico correspondente.

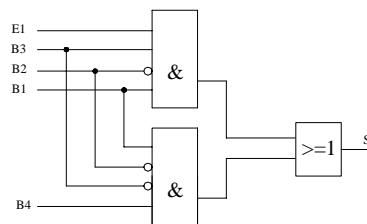
3.7.1 Comando Combinatório para Início de Ciclo

Os comandos binários podem ser **combinatórios**, onde a *saída ocorre unicamente em função da combinação lógica dos sinais de entrada*, ou **seqüenciais**, onde *os comandos produzem uma seqüência predeterminada de ações*, como mencionado por Bollmann (1995).

Verifica-se no diagrama (fig. 4), que para o início da etapa 1, existe uma condição lógica associada denominada *senal para início de ciclo*. Esta condição lógica está representada no diagrama pela letra **S**, o qual é resultante de um comando combinatório, que deve considerar as condições impostas para a partida do equipamento. Elaborar-se então, uma tabela de correspondência lógica para as variáveis de entrada e de saída (S), para posterior montagem da tabela verdade. Daí, tem-se a equação booleana, que pode ser simplificada pelas propriedades da álgebra de Boole ou pelo Diagrama de Karnaugh-Veitch. A equação booleana simplificada será então convertida num símbolo lógico. Do trabalho de Guimarães (1999) para o exemplo citado, tem-se a seguinte condição lógica representada por sua equação booleana.

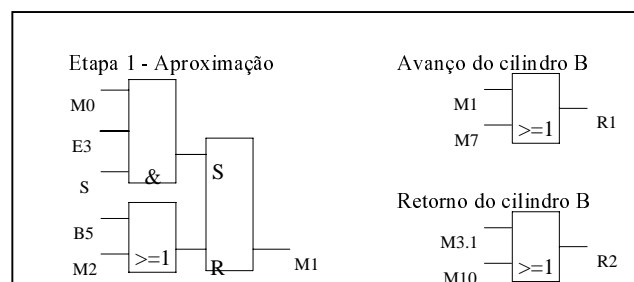
$$S = B1 \cdot \overline{B2} (E1 \cdot B3 + B4 \cdot \overline{B3})$$

O símbolo lógico será:



3.7.2 Comando Seqüencial

Os comandos combinatórios e seqüenciais comporão o diagrama lógico do sistema. Para execução dos comandos seqüenciais, associa-se a cada etapa um módulo lógico correspondente. Como exemplo, utilizando o método passo a passo em que cada passo está associado a uma memória do tipo RS (*Set-Reset*), o bloco lógico da etapa 1 do Grafcet (*Aproximação*) é representado na fig. 5, onde o diagrama lógico será elaborado dividido em dois blocos, o de processamento de sinais e o de atuação externa, que serão apresentados lado a lado. O bloco lógico à esquerda representa o *módulo de processamento de sinais* e o bloco lógico à direita representa o *módulo de atuação externa*.



Assim como o projeto lógico, o dimensionamento mecânico tem forte influência no resultado final do mecanismo.

4. CONCLUSÃO

A metodologia proposta permite por meio da elaboração de um diagrama de função seqüencial, Grafcet, a descrição funcional do equipamento, além de mostrar com clareza, quais os controles a serem implementados. Permite ainda, que seja desenvolvido todo o conceito de funcionamento do equipamento sem se ocupar da tecnologias de comando e operativa que serão empregadas. Esta metodologia foi aplicada no trabalho de Guimarães (1999) apresentando resultados bastante satisfatórios.

Por meio de símbolos lógicos correspondentes, o GRAFCET pode ser convertido em outros diagramas lógicos, como diagrama elétrico, diagrama pneumático ou diagrama de contatos (*ladder diagram*).

A linguagem do GRAFCET permite a fácil comunicação entre os profissionais de várias áreas envolvidos com o processo de automação (engenheiros mecânicos e especialistas em controle, por exemplo).

A utilização da lógica binária permite por meio das propriedades da álgebra booleana e do diagrama de Karnaugh-Veitch, a simplificação da montagem do sistema lógico, acarretando num projeto tão simples quanto ele possa ser.

REFERÊNCIAS BIBLIOGRÁFICAS

- Blanchard, M., 1994, *Comprendre, maitriser et appliquer le GRAFCET*, Paris , Cépaduès-Éditions.
- Bollmann, A., 1995, *Fundamentos da automação Industrial Pneutrônica*, São Paulo, ABHP – Associação Brasileira de Hidráulica e Pneumática.
- Guimarães, P. R. A., 1999, Projeto de um alimentador Pneumático Utilizando o Método GRAFCET para determinação do Funcionamento de um Sistema Lógico Automático, Tese de mestrado COPPE/UFRJ.
- Keeton, W., Vicente, C., 1995 “Escolha de Tecnologia de Controle em Aplicações Pneumáticas”, *Revista ABHP – Associação Brasileira de Hidráulica e Pneumática*, n. 95, pp. 10-14, Set/Out.

IMPLEMENTAÇÃO DE UM CONTROLADOR PARA UM ROBÔ COM SEIS GRAUS DE LIBERDADE

Fábio Luciano Demarchi

Raul Guenther

Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica, Laboratório de Robótica, Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil.

E-mail: fabio.demarchi@embraer.com.br

Resumo

Neste trabalho apresenta-se um sistema de controle para um robô industrial com seis graus de liberdade. A falha no controlador do robô em questão motivou o desenvolvimento de um novo sistema de controle para o robô, implementado num microcomputador PC conectado ao robô através de uma interface especialmente desenvolvida. O controlador é dotado de funções para programação ponto-a-ponto, execução de trajetórias e movimentação do robô. Apresenta-se a metodologia utilizada, a arquitetura do controlador, sua implementação e os resultados alcançados. Adicionalmente, foi implementado um controle para o seguimento de trajetórias e uma estratégia de controle PID, cuja análise e implementação também são discutidas.

Palavras-chave: Robótica, Controladores, Automação Industrial

1. INTRODUÇÃO

O atual processo de automatização nas empresas induz o crescimento do mercado de robótica e de manipuladores. No País, este crescente mercado é dominado por alguns fornecedores estrangeiros, que consideram o Brasil um mero comprador e usuário de tecnologia. Dentro deste contexto, este trabalho visa ser uma iniciativa no sentido do desenvolvimento da tecnologia de controladores para robôs industriais no Brasil.

O robô industrial Reis Rv-15 está instalado no Laboratório de Soldagem (Labsolda) da Universidade Federal de Santa Catarina (UFSC) desde o ano de 1989. Desde então tem sido utilizado para aplicações específicas na área de soldagem. Nos últimos anos, no entanto, seu uso tem sido prejudicado devido à constantes problemas no seu sistema de controle. As falhas freqüentes praticamente inviabilizavam sua utilização e o robô estava sendo sucateado. O desenvolvimento deste trabalho teve como objetivo o aproveitamento da estrutura do robô Reis Rv-15 para desenvolvimento e pesquisa de sistemas de controle para robôs.

A idéia básica é substituir o sistema de controle original por um sistema de controle baseado em um microcomputador PC. O sistema de controle é totalmente novo porque, como o sistema de controle original é fechado (não é possível se acessar o código-fonte do *software* de controle) não há possibilidade de modificá-lo. Dos componentes originais utiliza-se o mecanismo com seus motores e sensores, as placas de interface do controlador, e as placas de acionamento da fonte de potência.

Neste trabalho, inicialmente descreve-se o Robô Reis Rv-15. Depois apresenta-se um panorama dos sistemas de controle para robôs, a implementação da interface Robô x PC e finalmente, a implementação do controlador e os resultados obtidos.

2. O ROBÔ REIS RV-15

2.1 Descrição do Robô Reis Rv-15

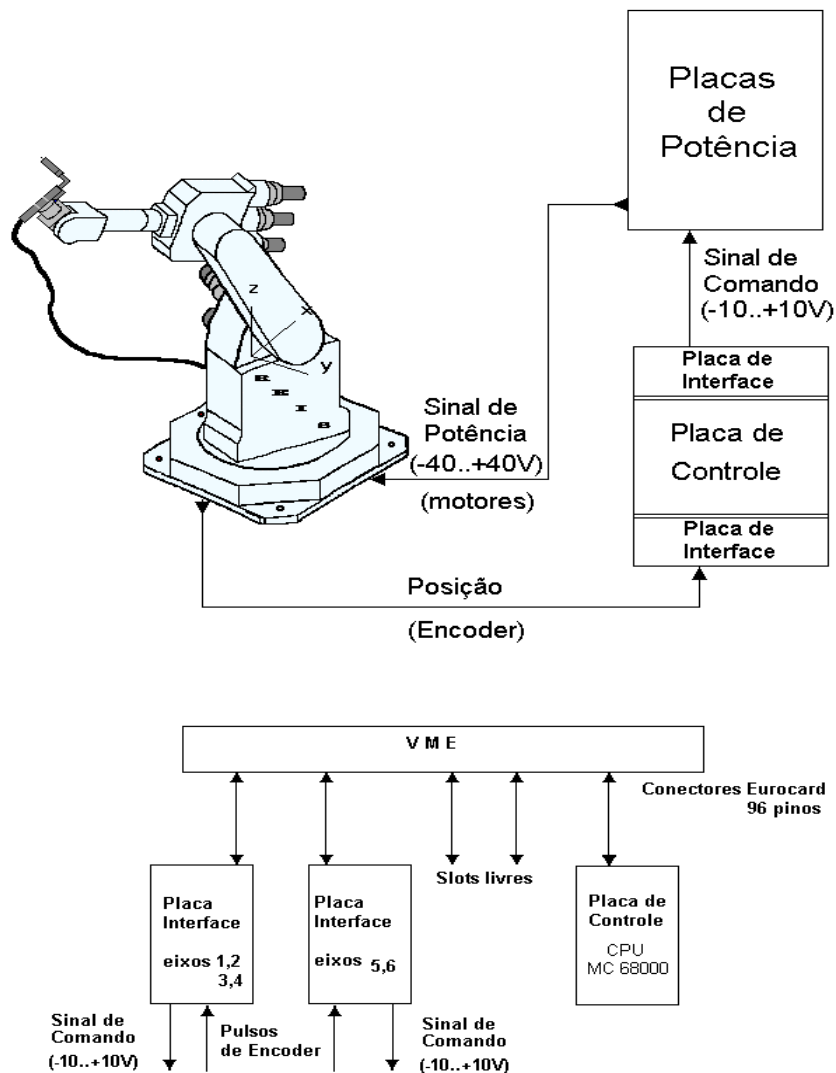


Figura 1. Sistema original de controle do robô Reis Rv-15

O robô Reis Rv-15 foi fabricado no Brasil pela empresa Ipso Automação, sob licença da empresa Reis (Alemanha). É um robô industrial que possui arquitetura de controle fechada, ou seja, permite apenas a programação de trajetórias pelo usuário, e não possibilita a programação e implementação de algoritmos de controle.

O robô tem seis graus de liberdade de rotação. Seu acionamento é elétrico, através de motores CC *brushless*. A medição da posição dos eixos é feita através de *encoders* incrementais, um para cada eixo. A capacidade de carga é de 15 Kg. Na indústria este robô é utilizado em várias aplicações, como soldagem, revestimento em camadas, montagem, manipulação e usinagem.

O robô é mecanicamente constituído de eixos, correias e engrenagens que acoplam os motores aos eixos. Para coordenar os motores dos eixos de modo que estes realizem um determinado movimento, existem circuitos eletrônicos constituídos de placas de acionamento dos motores (placas de potência) e placas de aquisição de dados e condicionamento de sinais dos sensores (placas de interface). O sistema é gerenciado por uma placa de controle, que contém o *software* de controle residente em memórias do tipo ROM.

2.2. Esquema original de funcionamento do sistema Reis Rv-15

O controlador recebe sinais de posição dos *encoders* através das placas de interface. O controlador calcula o torque a ser desenvolvido em cada eixo naquele momento e envia um sinal de comando para as placas de potência, através das placas de interface. As placas de potência recebem o sinal de baixa energia (sinal de comando) vindo da placa de controle e o convertem num sinal de potência a fim de acionar o motor. Este esquema de funcionamento é mostrado na figura 1.

3. SISTEMAS DE CONTROLE PARA ROBÔS

3.1 Introdução aos sistemas de controle para robôs

O sistema de controle de um robô é um sistema computacional que gerencia todas as suas atividades. Ele deve englobar a interface com o usuário, a linguagem de programação de robôs, o gerador de trajetória, o sistema de controle de movimento, entre outras funções como interface com periféricos e I/O. Os robôs utilizados nas indústrias são geralmente dotados de sistemas de controle “fechados”, que não permitem acesso ao código-fonte do sistema.

Os sistemas de controle podem também ser “abertos” e permitirem acesso ao código-fonte do *software* do sistema de controle. Estes sistemas geralmente são baseados em microcomputadores pessoais comuns (PC's) e caracterizam-se por facilitar a implementação do robô num ambiente industrial ou educacional.

Quanto à complexidade do sistema de controle, deve-se ressaltar se este possui somente controle ponto-a-ponto, ou se também há controle com seguimento de trajetória.

O controle ponto-a-ponto é a forma mais simples de controlar os eixos de um robô, no qual a sequência de pontos programados pelo usuário é diretamente utilizada como *set-point* (posição desejada) para os controles dos eixos. Durante a execução da trajetória, o controlador lê os pontos programados, e compensa o sistema no sentido de diminuir o erro entre o ponto desejado e o ponto real. Isto gera uma trajetória descontínua, com paradas nos pontos em que há mudança de ponto desejado. Esta característica limita a utilização do robô para aplicações do tipo “*pick-and-place*”.

No sistema com seguimento de trajetória, os pontos programados pelo usuário são interpolados sendo gerada uma sequência de pontos, com intervalo de tempo de interpolação T_t pré-definido. Os pontos gerados são utilizados como *set-points* pelo controlador, e a cada intervalo de tempo T_t , o *set-point* é atualizado. Isto gera uma trajetória suave e sem paradas, permitindo também o controle de velocidade, o que amplia a capacidade de utilização do robô nas aplicações industriais, como por exemplo, soldagem e pintura.

Outras características, como modelagem da cinemática e dinâmica do robô, podem estar presentes no sistema de controle. Em particular, a cinemática inversa permite que o usuário defina pontos no espaço cartesiano e o sistema gere as coordenadas correspondentes em espaço de junta. A modelagem dinâmica permite que o controlador de cada eixo compense forças gravitacionais, forças de atrito, forças de Coriolis, entre outras, resultando num controle mais preciso e robusto, porém com maior custo de processamento.

3.2 Projeto de um algoritmo de controle para PC

No presente trabalho, realiza-se um sistema de controle de maneira totalmente independente da placa de controle do sistema Reis Rv-15. Este sistema de controle deve ser capaz de receber os sinais dos *encoders* (realimentação de posição) e gerar uma referência para as placas de acionamento dos eixos. Assim, é necessário uma placa de aquisição de dados que transforme os pulsos de *encoder* numa palavra de dados interpretável por um microcomputador PC. O algoritmo de controle escrito em linguagem Pascal, reside no PC, e calcula uma referência para cada eixo, que é transmitida pela placa de aquisição de dados.

Esta abordagem de projeto permite o desenvolvimento completo de um sistema de controle para o robô, possibilitando a criação de uma base de conhecimentos acerca do desenvolvimento e implementação de sistema operacionais para robôs industriais.

Para este trabalho, aproveita-se as placas de aquisição de dados (placas de interface) disponíveis no sistema do robô, que convertem os pulsos de *encoder* (informação de posição angular de cada eixo) numa palavra de dados de 16 bits, colocando esta palavra disponível no barramento de dados. A placa de controle original, por sua vez, é substituída pelo PC. Na figura 2., mostra-se a configuração do sistema com a conexão do PC ao barramento VME.

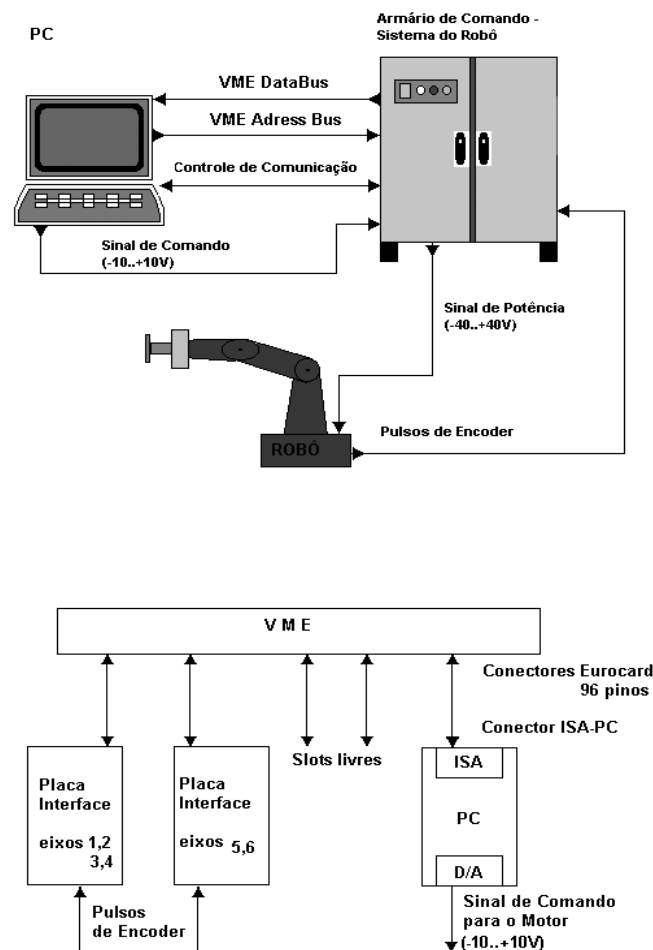


Figura 2. Sistema modificado, com a conexão do PC ao barramento VME (e retirada da placa de controle original da Reis)

Os *encoders* enviam pulsos para as placas de interface, de acordo com a movimentação dos eixos. As placas de interface convertem estes pulsos numa palavra de dados e disponibilizam esta informação no *bus* de dados de um barramento VME. O PC acessa este barramento VME através de uma interface apropriada constituída da placa de aquisição de dados INTERDATA e de uma conexão por cabos. Esta interface realiza o acesso ao endereço da palavra de dados que contém as informações dos *encoders*. O *software* de controle do sistema de controle lê em tempo real estes dados de posição angular de cada eixo e calcula como ação de controle, uma referência de torque para cada eixo. Esta referência é a saída do programa, e consiste de um sinal de torque para cada motor de eixo. Através dos conversores D/A da placa INTERDATA, os sinais de referência são enviados para as placas de acionamento. As placas de acionamento amplificam o sinal de referência e acionam os motores que movimentam os eixos. Este esquema pode ser visualizado na figura 2.

O *software* de controle é baseado num algoritmo PID, sendo que existe uma configuração distinta dos parâmetros proporcional, integral e derivativo para cada eixo. Da mesma maneira, o *software* possui subrotinas de leitura e controle específicas para cada eixo, que são executadas sequencialmente. O *software* de controle é apresentado em maiores detalhes no item 5 deste trabalho.

4. INTERFACE ROBÔ X PC

A fim de possibilitar o desenvolvimento de um sistema de controle para o robô no PC tornou-se necessário construir uma interface entre o PC e o robô que permita que um *software* escrito no PC possa controlar os eixos do robô.

A interface construída é constituída de três partes: placa de aquisição de dados, cabo de conexão PC x sistema do robô e circuito gerador de *clock*. Basicamente, esta interface conecta o PC ao barramento VME e às placas de potência do sistema Reis Rv-15 através da placa de aquisição de dados INTERDATA, conectada a um barramento ISA do PC.

A análise dos diagramas esquemáticos das placas de interface do sistema Reis Rv-15 permitiu determinar os códigos de endereçamento dos dados referentes à leitura da posição dos eixos e inicialização dos eixos. Para estas funções de leitura e inicialização dos eixos, utiliza-se a porta digital da placa INTERDATA, que possui 8 Entradas e 8 Saídas digitais, permitindo ler a palavra de 16 bits do barramento VME em dois ciclos de 8 bits.

O sinal de comando dos eixos é transmitido diretamente às placas de potência de cada eixo, sem passar pelo barramento VME. Para isto, utiliza-se seis saídas analógicas (uma saída para cada eixo) da placa INTERDATA, que geram o sinal de comando de -10 a $+10$ V.

O circuito gerador de *clock* serve unicamente para alimentar a linha *SYSCLOCK* (16 MHz) do barramento VME, utilizada por circuitos integrados específicos da placa de interface. O cabo de conexão liga as E/S digitais da placa INTERDATA com o barramento VME e conecta as saídas analógicas do conversor digital analógico (D/A) da placa INTERDATA com as placas de potência. Os sinais nestas linhas de conexão não passam por qualquer circuito de condicionamento, ou *driver* de linha. Portanto, a conexão entre a placa INTERDATA e o sistema do robô é feita de maneira direta, sem qualquer circuito intermediário de condicionamento de sinais, exceto para o gerador de sinal de *clock*.

Apesar da relativa simplicidade, a interface PC x sistema do robô construída não apresentou qualquer problema com relação à ruído ou falha de leitura, durante todo o período de realização deste projeto. Verificou-se também que o tempo de leitura é da ordem de alguns microssegundos, utilizando-se um PC-486 100 MHz. Como o tempo de amostragem total obtido com o *software* de controle é da ordem de dois a três milissegundos, observa-se que apesar de se realizar dois ciclos de leitura para cada leitura efetiva de posição de eixo, este atraso de leitura não é relevante em relação ao tempo de amostragem total. Deste modo, para

futuros desenvolvimentos, não há a necessidade de se modificar a interface construída com o objetivo de se melhorar o desempenho, mesmo sendo esta composta de simples cabos de conexão e um simples circuito gerador de *clock*. Uma descrição mais completa desta interface é encontrada em (Demarchi, 1999).

5. IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE

O objetivo deste sistema de controle é a movimentação dos eixos do robô através do teclado do PC, bem como programação de pontos (ângulos de rotação dos eixos) e execução de trajetórias. Como ponto de partida, realizou-se a análise do problema de controle dos eixos utilizando-se uma estrutura de controle proporcional para cada eixo individualmente. Desta forma, o *software* deve ler em tempo real a posição dos eixos do robô e comparar esta posição com um valor desejado (*set-point*) fornecido pelo usuário, gerando um erro de posição. O controle calcula então uma ação proporcional a este erro.

Uma subrotina de leitura e controle acessa os dados de posição dos eixos disponível no barramento VME através da interface construída, e calcula uma ação de controle para cada eixo individualmente. Esta subrotina é executada ciclicamente, e monitora constantemente a posição dos eixos. Uma subrotina de movimentação monitora o valor das teclas no teclado do PC e quando o usuário pressiona uma tecla específica, incrementa ou decrementa o valor do *set-point* de um eixo específico. A ação de controle age no sentido de diminuir o erro entre a posição atual e o *set-point*, movimentando o eixo, que pára quando o erro é nulo. Sub-rotinas para prevenção de colisão com os fim-de-cursos e inicialização dos eixos (pois se utiliza encoder incremental) também foram implementadas.

Na subrotina para programação de pontos, o usuário movimenta o robô (cada eixo individualmente) através do teclado do PC, e quando atinge um ponto específico, “grava” este ponto, acionando uma tecla específica. Este ponto (os ângulos de cada eixo) é gravado num arquivo texto de pontos programados. Este arquivo gera uma seqüência de pontos para cada eixo, que é utilizada por uma sub-rotina de execução. Quando o usuário decide executar o programa, esta sub-rotina lê cada ponto e iguala ao *set-point* de cada eixo. O robô então movimenta cada eixo até que o erro entre a posição atual e o *set-point* seja nulo. A sub-rotina é executada até que todos os pontos do arquivo de pontos programados sejam percorridos.

A fim de se obter controle de trajetória utilizou-se um gerador de trajetórias já utilizado em outros trabalhos do Laboratório de Robótica da UFSC, e descrito em Bier *et al.* (1998). O gerador de trajetórias tem a função de interpolar os pontos programados pelo usuário, e gerar um vetor de pontos que caracterizam uma trajetória no espaço. Este gerador, implementado em Matlab, foi modificado para utilização no sistema de controle em Pascal. A principal modificação realizada consiste em se gravar o vetor de pontos interpolados gerados no Matlab num arquivo que pode ser lido pelo *software* de controle em Pascal. Estes pontos são lidos a cada T ms (tempo de amostragem do *software* de controle), sendo que a cada T ms, o *set-point* de cada eixo é atualizado, e o eixo percorre uma trajetória suave através de uma trajetória descrita pelo arquivo de pontos, caracterizando o controle de trajetória.

Para o controle PID, a implementação da ação integral é feita aproximando-se a integral pelo somatório do erro de posição e multiplicando-se este somatório por um ganho K_i . Para a ação derivativa, utiliza-se aproximação de primeira ordem. Adiciona-se um filtro passa-baixa de primeira ordem, para atenuar a amplificação de ruído característico da ação derivativa. Na figura 3 mostra-se o esquema de controle PID implementado, onde $\theta_1, \theta_2, \dots, \theta_n$, são os pontos (ângulos de rotação do eixo) programados pelo usuário, θ_{sp} é o *set-point* fornecido pelo gerador de trajetórias, e θ é a posição angular real do eixo. O bloco “eixo” representa a placa de potência e o eixo mecânico propriamente dito. Para determinação dos ganhos K_p, K_i e K_d

utilizou-se um procedimento prático proposto em (Klafter, 1990), no qual não se necessita levantar o modelo do eixo do robô.

Na figura 4 compara-se a trajetória real seguida e a teórica gerada pelo gerador de trajetórias para o eixo 3. Como o erro é muito pequeno, mostra-se na mesma figura, o erro de trajetória para várias configurações dos ganhos PID. Em geral, nos testes realizados o erro máximo não ultrapassou 30 milésimos de grau, para todos os eixos e em várias outras velocidades e configurações dos braços do robô.

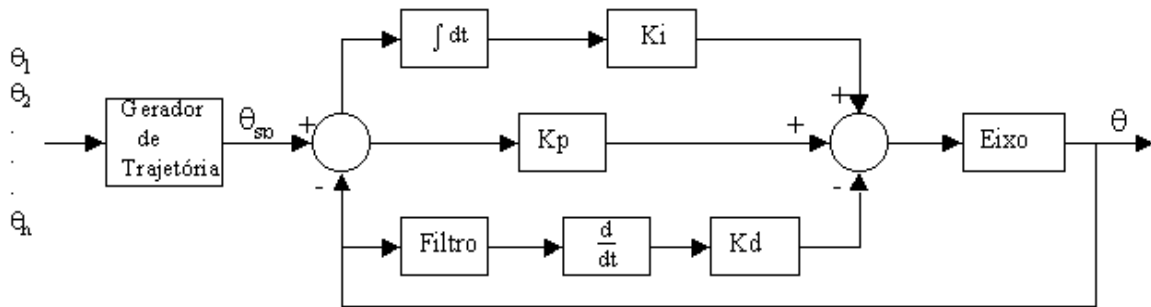


Figura 3. Esquema de controle implementado

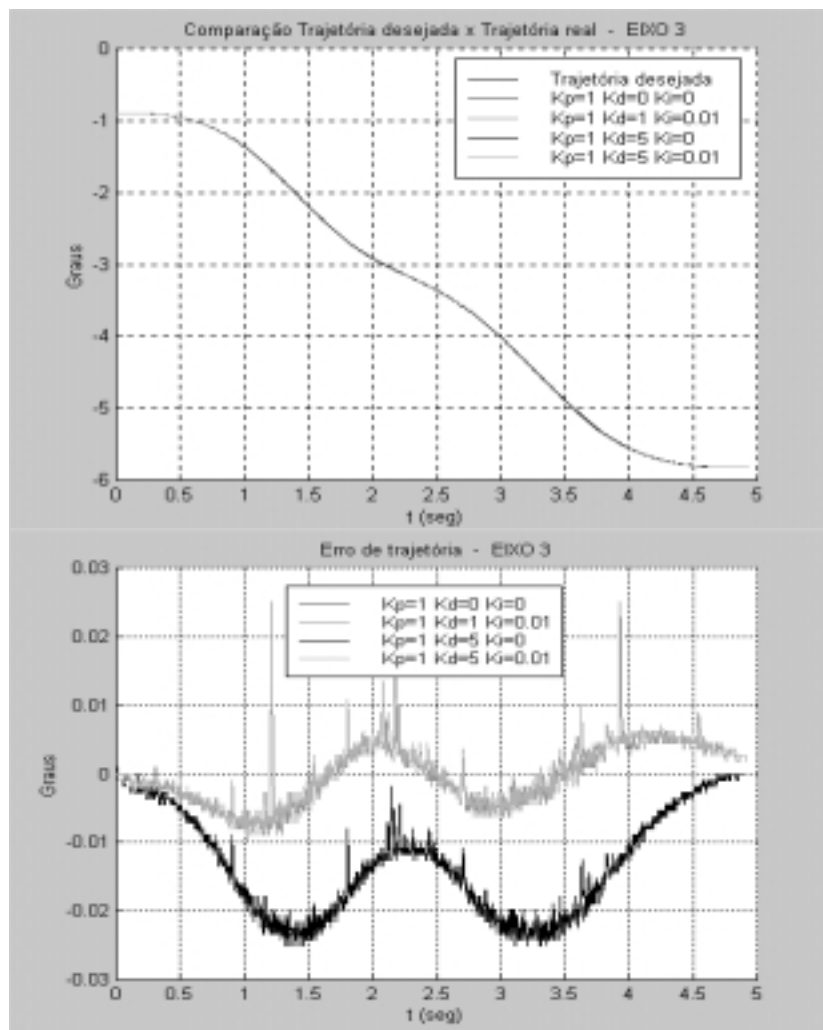


Figura 4. Comparação trajetória real x teórica e erro de trajetória para o eixo 3

Em geral, a ação integral diminuiu ainda mais o erro de trajetória obtido para o controle proporcional e a ação derivativa diminuiu o efeito de oscilações que ocorrem com o aumento do ganho integral. No entanto, mesmo utilizando-se um filtro, o ganho derivativo deve ser limitado, para não se amplificar demasiadamente o ruído.

A carga utilizada nos testes constituiu-se de uma tocha de soldagem Mig acoplada ao último eixo do robô (eixo 6), pesando aproximadamente 3 Kg. Lembrando que a capacidade nominal do robô é 15 Kg, uma hipótese para a relativamente boa resposta dos eixos pode ser o fato de os motores serem superdimensionados para a presente aplicação. No entanto, para a maioria das aplicações industriais que requerem controle de trajetória, a capacidade de carga não ultrapassa 5 Kg em média e portanto, esta abordagem de controle continua válida.

6. CONCLUSÕES

Este trabalho teve como objetivo o desenvolvimento de um sistema de controle para o robô industrial Reis Rv-15. O controlador original do robô foi desligado tendo sido implementado um controlador totalmente novo em um microcomputador PC.

Para tanto foi construída uma interface de hardware entre um PC e o sistema do robô. Esta interface permitiu o desenvolvimento de um *software* de controle, que permite a um usuário movimentar o robô, programar pontos no espaço em coordenadas de junta e executar trajetórias, totalmente a partir do teclado do PC.

Utilizou-se estratégia de controle PID individual para cada eixo. Para o controle com seguimento de trajetória implementou-se um algoritmo que gera trajetórias a partir de pontos programados pelo usuário. Os resultados obtidos foram relativamente satisfatórios considerando a simplicidade do controlador implementado. O sistema foi validado através da realização de várias trajetórias pelo robô, dentro do seu espaço de trabalho, com uma carga correspondente à uma tocha de soldagem de aproximadamente 3 Kg.

Desse modo, atingiu-se o objetivo de desenvolver um sistema de controle para o robô Reis Rv-15 e demonstrou-se a viabilidade prática de implementar um sistema de controle para um robô industrial baseado num microcomputador tipo PC. Dominando-se a tecnologia de controladores para robôs industriais, universidades e fabricantes nacionais podem obter capacitação tecnológica para o desenvolvimento e construção de robôs universais do tipo apresentado neste trabalho, que são largamente utilizados nas indústrias.

Como continuação natural do trabalho, prevê-se a inclusão do modelo cinemático ao gerador de trajetórias, a fim de permitir programação de pontos no espaço em coordenadas cartesianas. A realização de uma interface amigável com o usuário e a implementação de facilidades de programação para o usuário, são mais algumas sugestões no sentido de tornar o sistema básico desenvolvido neste trabalho, aplicável a nível industrial.

7. REFERÊNCIAS

- Sciavicco, L., Siciliano, B., 1996, "Modeling and Control of Robot Manipulators", McGraw-Hill, New York.
- Klafter, R.D., Chmielewski, T.A., Negin, M., 1990, "Robotic Engineering", Addison Wesley.
- Demarchi, F.L., 1999, "Desenvolvimento de Sistema de Controle Para um Robô Industrial de Seis Eixos", Relatório final da disciplina DAS 5511 - Projeto de Fim de Curso, Universidade Federal de Santa Catarina, Florianópolis, S.C., Brasil.
- Bier, C.C., Martins, A.E.C, Passold, D., 1998, "Planejamento de Trajetória", Documento interno do Laboratório de Robótica, Universidade Federal de Santa Catarina, Florianópolis, S.C., Brasil.

CONTROLE DE UM SERVOPOSICIONADOR PNEUMÁTICO POR MODOS DESLIZANTES

Eduardo A. Perondi

Raul Guenther

Laboratório de robótica – Departamento de Engenharia Mecânica – Universidade Federal de Santa Catarina - UFSC - 88040 – 900 - Florianópolis - SC

Resumo

Este artigo relata um estudo experimental de um esquema de controle por modos deslizantes (SLM) aplicado ao controle de posição de um servoposicionador pneumático. O algoritmo SLM utilizado caracteriza-se pela simplicidade e facilidade de implementação. A convergência do movimento à superfície de deslizamento é provada a partir do método de Lyapunov para o caso de uma superfície de chaveamento linear e invariante no tempo. Seus resultados são comparados com os fornecidos por um controlador por realimentação de estados convencional (PVA) e mostram que o controlador SLM é mais robusto que o PVA quanto a variações da inércia da carga deslocada.

Palavras-chave: Servoposicionadores pneumáticos, Controle a Estrutura Variável, Robótica.

1. INTRODUÇÃO

Apesar de os servoposicionadores pneumáticos apresentarem algumas vantagens relativas (são baratos, leves e limpos), eles têm a sua aplicação industrial limitada pela dificuldade do seu controle. Isto se deve principalmente à alta compressibilidade do ar e das altas não-linearidades inerentes ao seu funcionamento.

Este artigo dá seguimento ao trabalho de desenvolvimento de um controlador que supere as dificuldades inerentes aos servoposicionadores pneumáticos permitindo a sua aplicação em robótica. Dentro desta linha de pesquisa já foi investigado a possibilidade de utilização dos controladores por realimentação de variáveis de estado PVA e PΔP (Perondi e Guenther, 1999a) e da possibilidade de compensação do atrito para um controlador PVA através do uso de um observador de estados (Perondi e Guenther, 1999b). Uma dificuldade encontrada com o uso destes controladores é a falta de robustez quanto à variação da inércia deslocada, pois muitos mecanismos de precisão apresentam esta característica (a maioria dos robôs, por exemplo, tem a inércia dos seus braços dependente da sua configuração geométrica variável no tempo e trabalham com variação da massa da carga deslocada).

O controle por modos deslizantes (SLM - *slide mode control*) é um tipo de controle a estrutura variável que se caracteriza por apresentar baixa sensibilidade a perturbações e a variações nos parâmetros do sistema. Esta característica deve-se à capacidade do SLM de operar com altos ganhos, pois na vizinhança do modo deslizante a entrada do controlador é próxima de zero enquanto que a saída é um valor limitado. A utilização de altos ganhos é uma maneira usual de minimizar a influência das perturbações e incertezas. Vários autores têm proposto o uso de algoritmos de controle por modos deslizantes em servoposicionadores pneumáticos (Thomasset et al., 1993; Bouri et al., 1994; Tang e Walker, 1995; Surgenor et al., 1995; Pandian et al., 1997).

O presente artigo descreve os procedimentos de projeto e a aplicação prática de um esquema SLM proposto por Surgenor et al. (1995). Neste artigo a prova da convergência do movimento à superfície de deslizamento é realizada a partir do método de Lyapunov. Os resultados obtidos (equivalentes aos de Surgenor et al. (1995) e de Pandian et al. (1997)) comprovam que o controlador SLM apresenta performance bastante superior à do controlador tradicional por realimentação de estados PVA, principalmente no que diz respeito à robutez quanto à variação da massa deslocada.

2. O SERVOPOSICIONADOR PNEUMÁTICO

O sistema em estudo está ilustrado na Fig.1. Uma servoválvula proporcional 5/3 (MPYE-5-1/8 FESTO) aciona um cilindro de dupla ação com transmissão de movimento por fita com diâmetro interno de 0,025 m e curso 1 m (DGPL-1000 FESTO). A válvula possui vazão nominal (medida) $Q_N = 7.10^{-3} \text{ m}^3/\text{s}$ (420 l/min) independente da pressão de suprimento devido a uma realimentação interna. Este recurso garante também um comportamento aproximadamente linear de abertura da válvula em função da tensão aplicada. Um pequeno sobrepassamento (*overlap*) ocasiona uma zona-morta de cerca de 4%. O êmbolo móvel do cilindro é conectado a um sensor linear (Festo POT-1000-TFL) de curso igual ao do cilindro atuador (1,0 m). Três sensores piezoelétricos de pressão medem a pressão absoluta em cada câmara do cilindro e a pressão manométrica de suprimento. O aparato de controle e aquisição de dados consiste em uma placa de controle rápida (dSPACE GmbH – DS1102) acoplada a um microcomputador IBMPC DX266MHz. O tempo de amostragem utilizado é de 1 ms. Nas conversões AD e DA, a placa utilizada apresenta programas auxiliares para gerenciamento da aquisição de dados e módulos de acoplamento para o *MatLab/Simulink*. Assim, os algoritmos em tempo real de medições e controle estão programados via módulo *dSPACE* para o *MatLab/Simulink* (ver dSPACE GmbH, 1996).

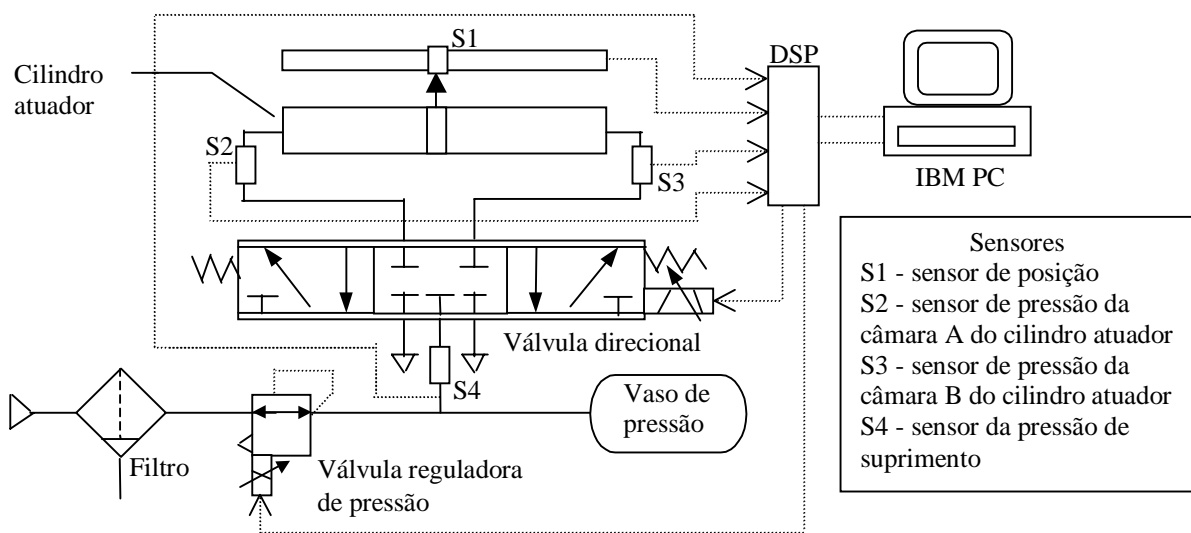


Figura 1. Esquema do servoposicionador pneumático.

3. O CONTROLE DO SERVOPOSICIONADOR

Os controladores apresentados neste trabalho foram desenvolvidos considerando o posicionador pneumático representado por uma função de transferência de terceira ordem

entre o sinal de controle em tensão da servoválvula (u – entrada) e o deslocamento do pistão (y – saída), com a forma

$$T(s) = \frac{Y(s)}{U(s)} = \frac{b_0}{s(s^2 + a_2s + a_1)} \quad (1)$$

cuja correspondente equação diferencial é dada por

$$\ddot{y} + a_2\dot{y} + a_1y = b_0u \quad (2)$$

onde s é a variável de Laplace e b_0 , a_1 e a_2 são parâmetros que dependem das características do sistema ($a_1 = 327$, $a_2 = 229$ e $b_0 = 4113$, calculados para o sistema experimental em estudo). Esta representação é utilizada por diversos autores (veja, por exemplo, Uebeling e Vaughan, 1997 e Perondi e Guenther, 1999a) e resulta da linearização do modelo obtido a partir da descrição da vazão mássica e evolução da variação da pressão em cada câmara do cilindro atuador e do equilíbrio de forças no êmbolo do pistão.

Com uma realimentação proporcional o modelo linear de terceira ordem (1) tem pólos em malha fechada que se deslocam para o semiplano complexo da direita à medida que o ganho aumenta. Desta forma, para que não haja risco de instabilidade, o ganho proporcional fica limitado a pequenos valores, o que limita o desempenho do posicionador. Performances melhores podem ser obtidas utilizando uma realimentação de estados, pois como o modelo linear representado pela Eq.(1) é controlável, pode-se desta forma alocar os pólos do sistema em qualquer posição do plano complexo. Assim, através do projeto adequado dos ganhos de um controlador de estados, teoricamente é possível obter a performance desejada para o sistema.

Diversos autores desenvolvem controladores com realimentação de estados tomando o modelo linear de terceira ordem como base, Wikander (1988), Virvalo, (1995). Estes controladores são referenciados como controladores PVA pois utilizam medidas de *posição* (P), *velocidade* (V) e *aceleração* (A). Sendo y_d o sinal de referência fixa para a posição, a lei de controle um controlador PVA para um sistema de 3ª ordem é dada por

$$u = K_p(y_d - y) - K_v\dot{y} - K_a\ddot{y} \quad (3)$$

Como a posição dos pólos do sistema dependem dos seus parâmetros, a variação destes afeta sua posição e, conseqüentemente, a performance dinâmica do sistema realimentado. Isto introduz a necessidade do uso de controladores robustos às variações dos parâmetros. Esta propriedade torna-se muito importante em sistemas pneumáticos de posicionamento preciso pois variações paramétricas são inerentes ao funcionamento de sistemas altamente não-lineares e também podem ocorrer em função de alterações de características durante do sistema durante o seu funcionamento (mudança no valor da inércia deslocada, por exemplo).

4. O CONTROLE POR MODOS DESLIZANTES APLICADO A UM SERVOPOSICIONADOR PNEUMÁTICO

No controle a estrutura variável a estrutura do controlador é trocada de acordo com uma determinada regra. Para isso utilizam-se leis de controle chaveadas com o objetivo de dirigir a trajetória do sistema no espaço de estados em direção a hipersuperfícies conhecidas como *superfícies de deslizamento*. Estas superfícies, definidas em projeto, uma vez atingidas devem

confinar a trajetória do sistema de forma que ela deslize sobre a superfície até o equilíbrio. Este comportamento caracteriza o regime de deslizamento ou de modo deslizante.

Utkin (1978), DeCarlo et al.(1988) e Slotine e Li (1991) abordam extensivamente a teoria e aplicações do controle a estrutura variável. No presente trabalho é apresentado o desenvolvimento de um controlador com estrutura variável aplicado ao controle de posição do servoposicionador pneumático.

4.1 Projeto de um controlador por modos deslizantes para o servoposicionador pneumático

Para um servo-sistema pneumático com dinâmica dada pela função de transferência da Eq.(1), Surgenor et al. (1995) sugerem o uso da seguinte superfície de deslizamento,

$$S = -c_1 e + c_2 \dot{y} + \ddot{y} = -c_1(y_d - y) + c_2 \dot{y} + \ddot{y} = -c_1 y_d + c_1 y + c_2 \dot{y} + \ddot{y} = 0 \quad (4)$$

onde $e \equiv y_d - y$. Quando a trajetória dos estados encontra a superfície de deslizamento $S = 0$, a sua dinâmica passa a ser dada por

$$\frac{y}{y_d} = \frac{c_1}{s^2 + c_2 s + c_1} \quad (5)$$

Esta dinâmica deve ser projetada de forma que $y \rightarrow y_d$ quando $t \rightarrow \infty$. Para tanto, uma condição suficiente é que os seus coeficientes c_1 e c_2 sejam estritamente positivos.

Uma condição de convergência da trajetória de estados em direção à superfície de deslizamento pode ser estabelecida escolhendo o sinal de controle u de forma que a distância quadrática (medida por S^2) decresça ao longo de qualquer trajetória do sistema. Para tanto, pode-se projetar o sinal de controle de forma que a função tipo de Lyapunov $V = S^2 / 2$ possua derivada primeira em relação ao tempo negativa ($\dot{V} = S\dot{S} < 0$), garantindo-se assim que $S \rightarrow 0$ quando $t \rightarrow \infty$, assintoticamente.

A convergência da trajetória em direção à superfície de deslizamento também pode ser obtida pela escolha da lei de controle de forma que

$$\frac{1}{2} \frac{d}{dt} S^2 = S\dot{S} \leq -\eta |S| \quad (6)$$

onde $\eta > 0$ pode ser interpretado com uma medida da velocidade de deslocamento da trajetória em direção à superfície de deslizamento. Slotine e Liu (1991) demonstram que se o estado inicial do sistema é tal que $S(t=0) \neq 0$ (ou seja, a trajetória se encontra inicialmente fora da superfície de deslizamento) e a condição de convergência da Eq.(6) é satisfeita, a superfície de deslizamento é alcançada pela trajetória do estado em um tempo finito t_a dado por $t_a < |S(t=0)| / \eta$.

O controle por modos deslizantes do sistema de 3ª ordem em estudo necessita do projeto de um sinal de controle para que a trajetória obedeça à condição de convergência (6). Para o projeto do sinal de controle, \dot{S} pode ser calculado a partir da Eq.(4), ou seja, $\dot{S} = c_1 \dot{y} + c_2 \ddot{y} + \dddot{y}$. Isolando \ddot{y} na Eq.(2) e substituindo na equação acima, resulta em $\dot{S} = (c_1 - a_1) \dot{y} + (c_2 - a_2) \ddot{y} + b_0 u$ e portanto $S\dot{S} = S[(c_1 - a_1) \dot{y} + (c_2 - a_2) \ddot{y} + b_0 u]$.

Existem diversas possibilidades para a escolha do sinal de controle para que a condição de convergência (6) seja satisfeita. Surgenor et al. (1995) propõem um controlador com sinal de controle dado por $u = -V_{\max} \text{sgn}(S)$ onde V_{\max} é o valor da tensão de controle máxima aplicada à servoválvula. Para este controlador Surgenor et al. (1995) não provam a convergência do movimento em direção à superfície de controle.

Um sinal de controle equivalente pode ser dado por:

$$u = -\frac{\eta}{\hat{b}_0} \text{sgn}(S) \quad (7)$$

onde \hat{b}_0 é uma estimativa do parâmetro b_0 .

Neste caso, $S\dot{S} = [(c_1 - a_1)\dot{y} + (c_2 - a_2)\ddot{y}]S - \eta|S|b_0/\hat{b}_0$. Assim, quando $S > 0$, a condição dada em (6) é satisfeita se $[(c_1 - a_1)\dot{y} + (c_2 - a_2)\ddot{y}] - \eta b_0/\hat{b}_0 < 0$, o que é verdade se

$$\eta > \frac{\hat{b}_0}{b_{0\min}} \left(\overline{a_1 c_1} |\dot{y}_{\max}| + \overline{a_2 c_2} |\ddot{y}_{\max}| \right) \quad (8)$$

onde $\overline{a_1 c_1} = \max(|c_1 - a_{1\min}|, |c_1 - a_{1\max}|)$ e $\overline{a_2 c_2} = \max(|c_2 - a_{2\min}|, |c_2 - a_{2\max}|)$. Os subíndices *min* e *max* referem-se aos valores extremos máximo e mínimo em módulo que os parâmetros ou estados podem assumir no sistema enquanto que a função $\max(x_1, x_2)$ é definida como $\max(x_1, x_2) = x_1$ para $x_1 \geq x_2$ e $\max(x_1, x_2) = x_2$ para $x_1 < x_2$. Com $S < 0$ a Eq.(8) também satisfaz a condição de convergência dada pela Eq.(6).

A Eq.(8) permite concluir que o controlador proposto necessita apenas do conhecimento do limite superior de η . Basta então que as incertezas paramétricas sejam limitadas para que a condição de deslizamento seja sempre satisfeita. Esta equação indica ainda que à medida em que a dinâmica desejada se afasta da dinâmica (natural) do sistema (o valor de c_1 se distancia do valor de a_1 e o valor de c_2 se distancia do valor de a_2) é necessário uma atividade mais intensa de controle, aumentando efeitos indesejáveis (como, por exemplo, o desgaste maior dos componentes a longo prazo). Além disso, como o chaveamento real ocorre em uma velocidade limitada, surgem sinais de alta frequência indesejáveis conhecidos como *chattering*.

4.2 Projeto de um controlador de posição por modos deslizantes para o servoposicionador pneumático

Lembrando que o erro de posição é definido como $e = y_d - y$ e que a função de transferência a malha fechada no modo de deslizamento ($S = 0$) é dada pela Eq.(6), pode-se definir $c_1 \equiv \omega_n^2$ e $c_2 = 2\zeta\omega_n = 2\zeta\sqrt{c_1}$, onde ω_n e ζ são a frequência natural e o índice de amortecimento da resposta de 2ª ordem com relação à posição desejada y_d . Os parâmetros ω_n e ζ podem ser especificados por qualquer critério de projeto baseado nas características estáticas e dinâmicas da resposta desejada.

Como o projeto de um controlador com modos deslizantes pode resultar em um controlador inadequado devido ao excessivo *chattering* no sinal de controle, uma alternativa é modificar a lei de controle para incluir a chamada camada limite (*boundary layer* - ver

Slotine e Li, 1991). Este procedimento consiste na introdução da seguinte modificação do sinal de controle

$$u = -\frac{\eta}{\hat{b}_0} \text{sat}(S) = -V_{\max} \text{sat}(S) \quad (9)$$

onde $\text{sat}(S)$ é definida como $\text{sgn}(S/\phi)$ quando $|S| > \phi$ e como S/ϕ quando $|S| \leq \phi$. $V_{\max} = \eta/\hat{b}_0$ é o valor da tensão de saturação (tensão máxima aplicada). Na condição $|S| \leq \phi$ (no interior da camada limite), o controlador se comporta como um controlador de estados com vetor de ganhos $(V_{\max}/\phi)[-c_1 \ c_2 \ 1]$ e vetor de estados $[e \ \dot{y} \ \ddot{y}]$. É importante salientar que existe um compromisso entre o *chattering* do sinal de controle e o erro de regime de posição pois ambos dependem da largura ϕ da camada limite. Assim, quanto maior ϕ , menor o *chattering* e maior o erro de regime.

A definição de ω_n e ζ pode ser realizada a partir do tempo de assentamento da resposta e da escolha de um sobrepasso máximo aceitável. A exemplo de Surgenor et al.(1995), neste trabalho foram definidos os valores máximos aceitáveis de 0,2s e 1,5% para o tempo de assentamento (critério de 2%) e sobrepasso, respectivamente. Estes valores implicam em uma frequência natural $\omega_n = 25 \text{ rad/s}$ e em um índice de amortecimento $\zeta = 0,8$ para a dinâmica do movimento deslizante, resultando em $c_1 = 625$ e $c_2 = 40$.

Para a implementação do controlador é necessário a definição dos fatores V_{\max} e ϕ .

Através de um estudo dos parâmetros do sistema linear foram obtidas estimativas para os valores extremos de b_0 , a_1 e a_2 . Por simulação foram determinados os valores aproximados para \dot{y}_{\max} e \ddot{y}_{\max} , o que, a partir da Eq.(8), resultou em $V_{\max} > 0,05 \text{ V}$. Testes experimentais mostraram que valores desta ordem para V_{\max} ocasionam atrasos muito grandes no momento do arranque inicial do movimento possivelmente devido ao efeito do atrito estático. Este atraso no arranque leva à necessidade de uso de valores maiores para V_{\max} . Optou-se assim pela utilização de $V_{\max} = 1,5 \text{ V}$. Com relação à largura da camada limite da superfície de deslizamento o valor utilizado é de $\phi = 35$. Este valor é resultante de um compromisso entre o nível de *chattering* mínimo e a precisão aceitável para o erro de posicionamento.

5. Resultados

Na figura 2 estão apresentadas as curvas referentes às respostas experimentais da posição y ao longo do tempo para dois casos de massas (0,3 e 1,4Kg) com o uso dos controladores PVA e SLM.

Os ensaios foram realizados considerando, a exemplo de Pandian et al.(1997), um curso de 0,3m em torno da posição central do cilindro. Os ganhos do controlador PVA foram ajustados para a resposta com a massa de 0,3Kg considerando a seguinte relação $K_p = 4$, $K_v = K_p/50$ e $K_a = K_v/1000$ (Perondi e Guenther, 1999a). O controlador SLM está de acordo com o projeto apresentado na seção 4.2 com os fatores V_{\max} e ϕ ajustados para uma melhor resposta com a massa de 0,3Kg. Em ambos os controladores o sinal de velocidade foi obtido pela derivação numérica do sinal medido de posição amostrado a 1ms. A velocidade foi filtrada digitalmente por um filtro *butterworth* passa baixa com frequência de corte a

100rad/s. O sinal de aceleração foi obtido pela derivação numérica do sinal de velocidade. Não foram utilizados filtros adicionais.

Para diminuição do erro de regime e dos efeitos da assimetria inerente ao funcionamento da válvula, a zona-morta de aproximadamente 4% foi compensada em ambos os controladores. Isto foi realizado através da soma de valores fixos de compensação aos sinais de controle de acordo com a direção de abertura da válvula, resultando nas seguintes expressões para o sinal de controle efetivo: $u_{efet} = u + zm_1$ para $u > 0$ e $u_{efet} = u - zm_2$ para $u < 0$, onde zm_1 e zm_2 são os valores de compensação da zona morta. Foram utilizados $zm_1 = 0,056$ V e $zm_2 = 0,02$ V.

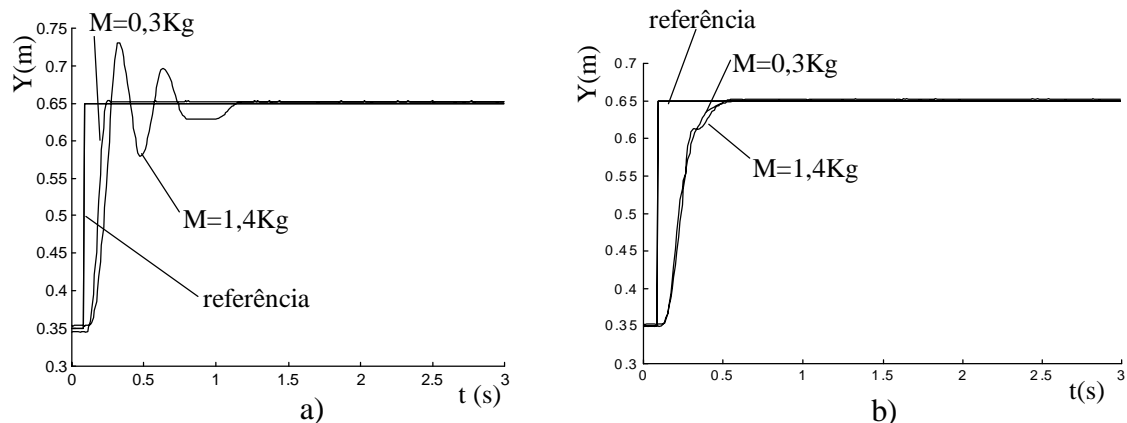


Figura 2. a) deslocamentos com controle PVA b) deslocamentos com controle SLM.

A figura 2 apresenta as respostas de posição dos controladores PVA e SLM para as duas massas ensaiadas. Os resultados indicam que para a massa de 0,3Kg ambos os controladores mostram bons resultados, com baixos tempos de assentamento e com erros de regime menores que $\pm 2,4$ mm e $\pm 0,2$ mm para os controladores PVA e SLM, respectivamente. Para o caso da massa de 1,4Kg percebe-se que o controlador SLM, apesar de apresentar uma perda de desempenho com tempo de assentamento e erro de regime ($\pm 1,6$ mm) maiores, mantém o comportamento da resposta relativamente próximo ao do caso anterior enquanto que a resposta para o controlador PVA é muito oscilatória, com tempo de assentamento muito maior, além de apresentar sobrepasso.

6. Conclusões

Os resultados obtidos experimentalmente comprovam que no controle de posição do servoposicionador pneumático o controle por modos deslizantes é mais robusto que o controlador de estados PVA quanto à variação da inércia.

Uma dificuldade prática encontrada na implementação que afeta a performance do SLM decorre do ruído proveniente da derivação do sinal de velocidade para obtenção da aceleração. O ruído soma-se ao *chattering* inerente ao SLM aumentando significativamente a intensidade de trabalho da válvula (o que pode resultar em desgaste dos seus componente a longo prazo) levando à necessidade de aumento da camada limite. Visando contornar este problema, a utilização de um controlador que não necessite da derivação do sinal de velocidade será objeto de futuros estudos.

REFERÊNCIAS

- Bouri, M., Thomasett, D., Richard, E., Scavarda, S., 1994, “Nonlinear Sliding Mode Control of an Electropneumatic Servodrive”, Proc.7th Bath Int.Fluid Power Workshop, Vol.6, Bath,UK,pp.501-504.
- DeCarlo, R. A., Zak, S. H., Matthews, G. P., 1988, “Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial”. Proceedings of the IEEE, vol. 76, nº.3.
- dSPACE digital signal processing and control engineering, 1996, DS1102 User’s Guides, Germany.
- Liu, S., Bobrow and J.E. ,1988, “An analysis of a pneumatic servo system and its application to a computer-controlled robot”, Transactions of the ASME, vol. 110, pp. 228-235.
- Pandian, S. R., Hayakawa,Y., Kanazawa,Y., Kamoyama, Y., Kawamura, S., 1997, “Practical Design of a Sliding Mode Controller for Pneumatic Actuators”. ASME Journal of Dynamic Systems, Measurement, and Control, vol.119, pp. 666-674.
- Perondi, E.A. e Guenther, R., 1999a, “Controle em Malha Fechada de um Servoposicionador Pneumático”, anais do XV Congr. Bras. de Eng. Mec. – COBEM 99 - Águas de Lindóia, SP, Brasil.
- Perondi, E.A. e Guenther, R., 1999b, “Controle com compensação de atrito de um servoposicionador Pneumático”, anais do IV Congreso Iberoamericano de Ingeniería Mecánica, CIDIM '99 Santiago, Chile.
- Slotine, J.J.E., Li, W., 1991, “Applied Nonlinear Control”, Prentice-Hall Inc..
- Surgenor, B. W., Vaughan, N. D. and Uebling, M., 1995, “Continuous sliding mode control of a pneumatic positioning”, System, Design and Performance, Proceedings of 8th Bath International Fluid Power Workshop, pp. 270-285.
- Thomasset, D., Richard, E., Scavarda, S., Lin, X.F., Sesmat, S., Bouhal, A., 1993, “Control of an electro-pneumatic servodrive: a state-affine or a sliding approach”. IFAC World Congress, Sydney.
- Uebling, M. and Vaughan, 1997, “On linear dynamic modeling of a pneumatic servo system”, The fifth Scand. Int. Conf. on Fluid Power, SICFP'97, Linköping, Sweden, pp. 363-378.
- Utkin, V.I., 1978, “Sliding Modes and Their Application in Variable Structure Systems”. MIR Publishers, Moscow.
- Virvalo, T. ,1995, “Modeling and design of a pneumatic position servo system realized with commercial components”, PhD Thesis, Tampere, Finland.
- Wikander, J., 1988, “Adaptive Control of Pneumatic Cylinder”. Dr. Thesis, Stockholm.

CONTROLADOR ROBUSTO H_∞ ASSOCIADO A TORQUE COMPUTADO PARA UM MANIPULADOR ROBÓTICO

Emerson de Góes
Alberto Adade Filho

Instituto Tecnológico de Aeronáutica, Divisão de Engenharia Mecânica-Aeronáutica, 12228-900, S. José dos Campos, SP, Brasil. E-mail: adade@mec.ita.cta.br

Resumo

Este trabalho aborda o projeto de um controlador robusto H_∞ , associado a um controle via método do torque computado, para um manipulador robótico de alto desempenho, visando suprimir efeitos de incertezas paramétricas - especialmente variações de massa na extremidade do manipulador - e dinâmicas não modeladas da planta. Ilustra-se que o método do torque computado só apresenta resultado aceitável quando não há incertezas consideráveis no modelo da planta. Um exemplo de aplicação em um robô de três graus de liberdade é realizado para ilustrar esta abordagem de controle robusto.

Palavras-chave: Controle Robusto; Otimização H_∞ ; Controle de Robôs.

1. INTRODUÇÃO

Uma técnica bastante investigada para o controle de movimento de manipuladores robóticos de alto desempenho é o controlador particionado ou tipo torque computado (Craig, 1986) cuja lei de controle tem duas componentes: uma baseada no modelo dinâmico da planta e outra responsável pelo ajuste baseado no erro entre o movimento desejado e o real.

O modelo dinâmico de um manipulador robótico está sujeito a incertezas, o que sugere basicamente duas abordagens para controle (Abdallah et al., 1991): uma a que usa o controle adaptativo e a outra, o controle robusto.

Este trabalho descreve o projeto de um controlador robusto para um manipulador robótico, utilizando a teoria de otimização H_∞ , associado a um controlador tipo torque computado.

2. CONFIGURAÇÃO E PROJETO DO CONTROLADOR ROBUSTO

2.1 Necessidade do Controle Robusto

O principal objetivo desta seção é demonstrar a necessidade de um controlador robusto para compensar os efeitos de dinâmicas não modeladas e variações no modelo do robô, quando este é controlado através do método do torque computado e é submetido a variações de trajetórias e variações na massa do objeto manipulado.

No método do torque computado (vide Fig. 1), a planta básica (robô) só é linearizada pelo controlador se o modelo de projeto, utilizado pelo controlador, for totalmente preciso. O modelo dinâmico de manipuladores robóticos, seriais, rígidos, pode ser representado na forma matricial,

$$\tau = M(\theta) \cdot \ddot{\theta} + C(\theta, \dot{\theta}) \cdot \dot{\theta} + G(\theta) \quad (1)$$

onde θ e τ são, respectivamente, o vetor de posição das juntas e o vetor de torques de controle, e M , C e G são a matriz de inércia, a matriz que contém os termos relacionados aos torques/forças de Coriolis e centrípeta e o vetor com os termos de torques/forças gravitacionais, respectivamente. Ainda na Fig. 1, θ_d é o vetor de posições desejadas e K_p e K_v são matrizes de ganhos de realimentação para a dinâmica do erro, no sistema compensado pelo modelo do manipulador (caixa denominada *inércia unitária* na Fig. 1).

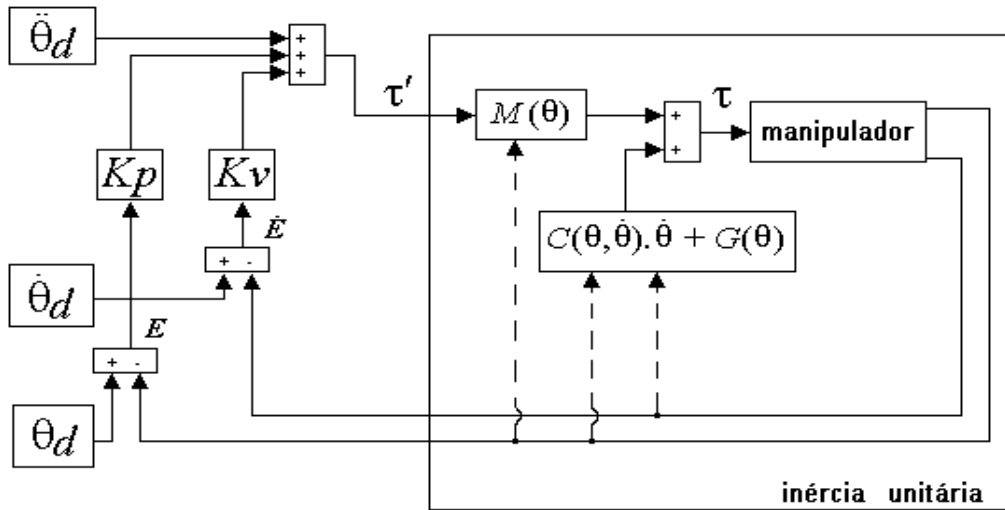


Figura 1. Método do torque computado aplicado a um manipulador robótico

Na modelagem realizada para uso no controlador tipo torque computado, tem-se

$$\tau^* = \hat{M}(\theta) \cdot \tau' + \hat{C}(\theta, \dot{\theta}) \cdot \dot{\theta} + \hat{G}(\theta) \quad (2)$$

pois o modelo utilizado, em geral apresentará diferenças relativas à Eq. (1). Isto implica no aparecimento de um erro de modelo nas equações dinâmicas de rastreamento de erro. Utilizando-se a Eq. (2) para controlar a Eq. (1), isto é, fazendo-se $\tau = \tau^*$, tem-se

$$\hat{M} \cdot \tau' + \hat{C} \cdot \dot{\theta} + \hat{G} = M \cdot \ddot{\theta} + C \cdot \dot{\theta} + G \quad (3)$$

e considerando a componente de ajuste baseado no erro,

$$\hat{M} \cdot (\ddot{\theta}_d + K_v \cdot \dot{E} + K_p \cdot E) + \hat{C} \cdot \dot{\theta} + \hat{G} = M \cdot \ddot{\theta} + C \cdot \dot{\theta} + G \quad (4)$$

$$\hat{M} \cdot (\ddot{E} + K_v \cdot \dot{E} + K_p \cdot E) = (M - \hat{M}) \cdot \ddot{\theta} + (C - \hat{C}) \cdot \dot{\theta} + (G - \hat{G}) \quad (5)$$

$$\ddot{E} + K_v \cdot \dot{E} + K_p \cdot E = \hat{M}^{-1} \cdot [(M - \hat{M}) \cdot \ddot{\theta} + (C - \hat{C}) \cdot \dot{\theta} + (G - \hat{G})] \quad (6)$$

Tem-se, assim, um novo conjunto de equações para a dinâmica do erro,

$$\ddot{E} + K_v \cdot \dot{E} + K_p \cdot E = \eta \quad (7)$$

onde η denota o lado direito da Eq. (6).

A Eq. (7) mostra que a dinâmica do erro de rastreamento é forçada por uma perturbação η , gerada pelas diferenças entre os parâmetros do modelo estimado e os parâmetros reais. A perturbação, decorrente das incertezas de modelo (por exemplo, variação de massa na ponta do manipulador quando este realiza operações tipo *pick-and-place*), pode degradar sensivelmente o desempenho do sistema e até desestabilizá-lo. Estudos de simulação realizados em (Góes, 2000), demonstraram que o controle via método do torque computado não garante desempenho satisfatório para manipuladores robóticos com sistema de acionamento direto sujeitos a perturbações no modelo (vide Fig. 2). Isto justifica o uso de uma técnica de controle mais apurada, no caso, controle robusto.

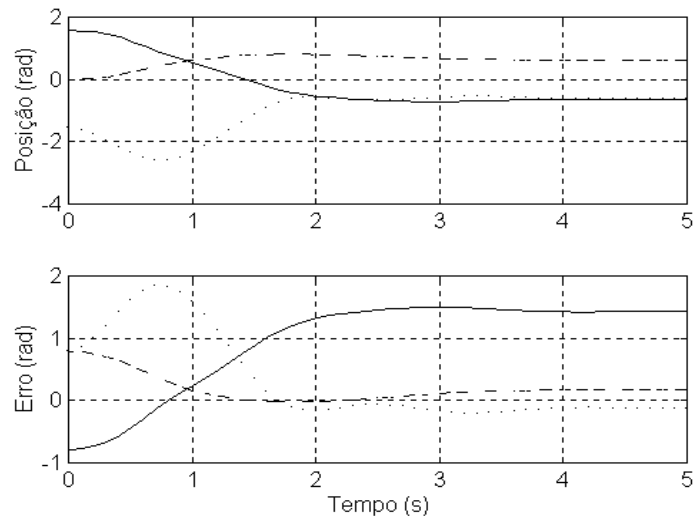


Figura 2. Resultados para a simulação de um sistema robótico (Robô ITA-IEMP) utilizando o método do torque computado com adição de massa de 100% ao último elo. Curva tracejada representa a junta 1 (base), curva cheia a junta 2 (braço) e pontilhada a junta 3 (antebraço)

No estudo da Fig. 2, o controle tipo torque computado foi projetado para não apresentar sobresinal além de proporcionar rapidez na estabilização para uma entrada tipo degrau, com posições inicial e final para cada junta dadas na Tabela 2. Nota-se na Fig. 2 que o desempenho dinâmico foi degradado pela variação da massa no último elo.

A Fig. 3 ilustra como η varia para uma trajetória cúbica no espaço de juntas determinada pela Tabela 1, para o modelo dinâmico do robô ITA-IEMP (descrito no Apêndice A) numa configuração de acionamento direto, submetido a uma variação de massa em sua extremidade.

Tabela 1. Posições e velocidades iniciais e finais para cada junta do robô ITA-IEMP

Tempo do trajeto = 2 s					
Nº da junta	Posição inicial (rad)	Posição final (rad)	Velocidade inicial (rad/s)	Velocidade final (rad/s)	Variação de massa de 0 Kg a 2,5 Kg
1	$-8\pi/9$	$8\pi/9$	0.0	0.0	
2	$\pi/6$	$\pi/6$	0.0	0.0	
3	$-\pi/4$	$-\pi/4$	0.0	0.0	

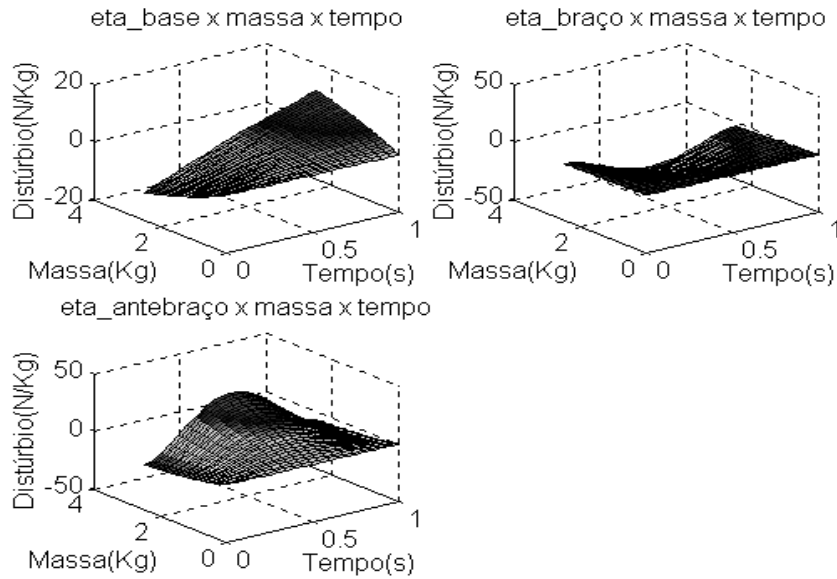


Figura 3. Variação de η para trajetória cúbica, com posições inicial e final dadas na Tabela 1

2.2 Projeto H_∞

2.2.1 Otimização H_∞

No controle H_∞ o objetivo de controle é minimizar a norma H_∞ da função de transferência de w (vetor de sinais exógenos) para z (vetor de saída associado aos requisitos de desempenho para o sistema), conforme ilustrado na Fig. 4.

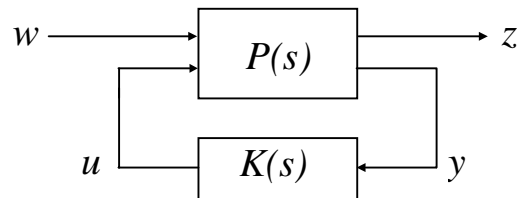


Figura 4. Configuração de controle sem modelagem de incertezas

Na Fig. 4, $P(s)$ representa a planta aumentada e $K(s)$ o controlador robusto. Seja $P(s)$ particionada da forma:

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \leftrightarrow \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right] \quad (8)$$

$$\begin{aligned} z &= P_{11}w + P_{12}u \\ y &= P_{21}w + P_{22}u \end{aligned} \quad (9)$$

A função de transferência de w para z fica:

$$T_{zw} = F_l(P, K) = P_{11} + P_{12} \cdot K \cdot (I - P_{22} \cdot K)^{-1} \cdot P_{21} \quad (10)$$

O controle ótimo H_∞ pode ser descrito matematicamente como a minimização de

$$\|F_l(P, K)\|_\infty = \max_{\omega} \bar{\sigma}(F_l(P, K)(j\omega)) \quad (11)$$

no espaço de todos os controladores $K(s)$ realizáveis que estabilizam o sistema.

2.2.2 Modelo Nominal da Planta

A representação nominal no espaço de estados (A,B,C,D) de um sistema robótico controlado via método do torque computado (robô+TC), com $x = [E' \ \dot{E}']'$ e $u = \tau$:

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}; B = \begin{bmatrix} 0 \\ I \end{bmatrix}; C = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}; D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (12)$$

K_p e K_v são as matrizes de ganhos proporcional e derivativo (vide Fig. 1).

2.2.3 Obtenção da Planta Aumentada

Note-se que o problema em questão, no espaço de erros, é um problema de regulação, um problema de sensibilidade mista: a função sensibilidade, $S = (I + GK)^{-1}$ é formatada juntamente com uma ou mais funções de malha fechada, tais como a função KS ou a função sensibilidade complementar $T = I - S$. No caso em que o ruído na medida é insignificante, procura-se formatar S e KS , pois S é a função de transferência entre a perturbação externa e a saída e KS entre a perturbação e os sinais de controle.

A planta P aumentada para o sistema, baseada em (Skogestad & Postlethwaite, 1996), está ilustrada na Fig. 5, onde G é a planta nominal (robô + TC), definida pela Eq. (12). Os vetores w , u , z_1 , z_2 e y têm dimensão 3. Neste projeto foram realimentados apenas os erros de

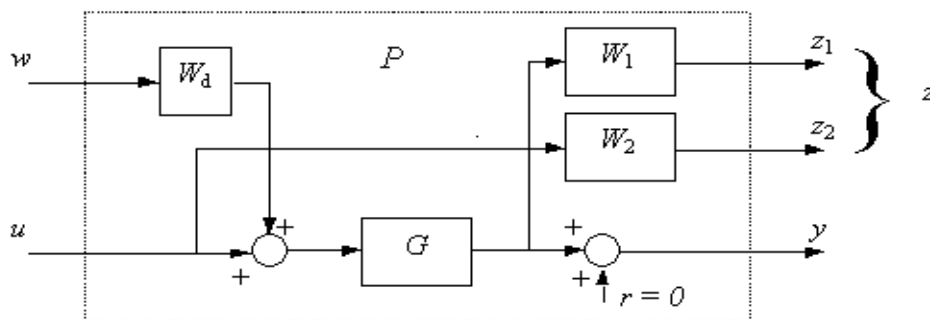


Figura 5. Planta aumentada para o sistema Robô + TC com incertezas

posição das juntas do manipulador, por dois motivos: dimensão do controlador, e problema de posto (*rank*) com a matriz da planta aumentada, para aplicação do algoritmo de Glover-Doyle dado em (Balas *et al.*, 1994).

O bloco W_d tem como função normalizar o nível de perturbação; isto limita os valores singulares do controlador que será projetado. A matriz de transferência W_1 trabalha como um filtro passa-baixas, limitando os valores singulares da função sensibilidade S , e W_2 tem como

função limitar os sinais de controle, atuando como um filtro passa-altas com frequência de corte aproximadamente igual a desejada para a largura de banda de malha fechada. As matrizes de transferência W_1 e W_2 utilizadas têm as seguintes estruturas:

$$W_1 = \begin{bmatrix} \frac{s/M_1 + w_{B1}^*}{s + w_{B1}^* A_1} & 0 & 0 \\ 0 & \frac{s/M_2 + w_{B2}^*}{s + w_{B2}^* A_2} & 0 \\ 0 & 0 & \frac{s/M_3 + w_{B3}^*}{s + w_{B3}^* A_3} \end{bmatrix}, \quad W_2 = I_{3 \times 3} \quad (13)$$

onde M_i , A_i , e w_{Bi}^* são, respectivamente, o máximo pico de magnitude de S , máximo erro de rastreamento de estado estacionário e mínima frequência de largura de banda (Skogestad & Postlethwaite, 1996). Como ponto de partida de projeto adotou-se, para todas as funções em (11), $M_1 = M_2 = M_3 = 2$, $A_1 = A_2 = A_3 = 0,0001$, $w_{B1}^* = 1,56$ rad/s, $w_{B2}^* = 3,23$ rad/s e $w_{B3}^* = 3,01$ rad/s. Foram escolhidas funções de primeira ordem como tentativa inicial, pois a dimensão do controlador depende do número de pólos destas funções.

2.2.4 Projeto do Controlador Robusto

Para reduzir a influência da perturbação η (oriunda das incertezas de modelo e definida na seção 2.1) no sistema, considerando a variação de massa no último elo como a principal fonte de incerteza, optou-se por utilizar a técnica de projeto de controle robusto que considera a teoria de otimização H_∞ (Maciejowski, 1989; Skogestad & Postlethwaite, 1996). Ao sistema controlado via método do torque computado é adicionada uma nova malha que contém o controlador robusto, cuja entrada é o vetor de erros de posição e cuja saída é um vetor v , como mostra a Fig. 6. O problema agora, é encontrar um controlador robusto que minimize a transferência de energia entre η e E (vide Eq. 7).

Uma solução de projeto de controlador robusto H_∞ para o sistema em questão foi obtida utilizando-se a função SYSIC, que auxilia na confecção da planta aumentada, e a função HINFSYN, que implementa o algoritmo de Glover-Doyle (Glover & Doyle, 1988), ambas funções do MATLAB μ -Analysis and Synthesis toolbox (Balas *et al.*, 1994). Resultados de simulações utilizando o projeto efetuado estão mostrados na seção 2.2.5.

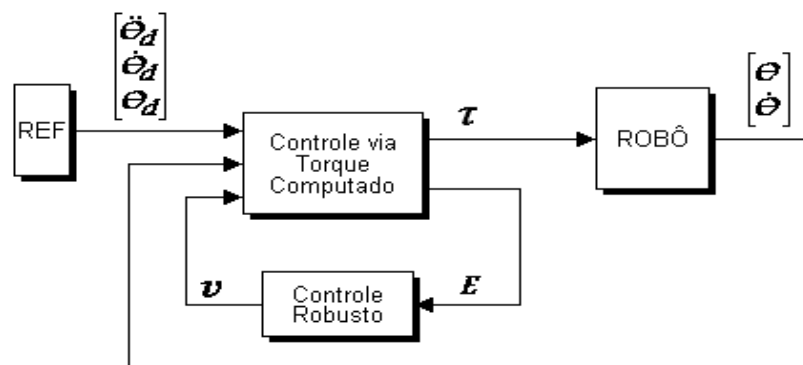


Figura 6. Adição do Controlador Robusto ao Sistema Robô + Torque Computado

Após algumas iterações as funções de ponderação foram escolhidas, de modo que os ganhos e ordem do controlador fossem os menores possíveis para o melhor atendimento dos

requisitos de desempenho: resposta sem sobresinal e tempo de estabilização menor que 1s. Para a obtenção da matriz W_l , Eq.13, utilizou-se os seguintes valores para as larguras de banda: $w_{B1}^* = 1.56 \text{ rad/s}$, $w_{B2}^* = 10 \text{ rad/s}$ e $w_{B3}^* = 100 \text{ rad/s}$. A matriz W_d adotada foi $W_d = \text{diag}(11,22,34)$. As outras ponderações não sofreram modificações. O controlador robusto obtido tem a descrição no espaço de estados mostrada no Apêndice B.

2.2.5 Resultados Obtidos

Os resultados apresentados nesta subseção foram obtidos através da utilização do simulador de manipuladores robóticos descrito em (Adade Filho & Góes, 1998). Foi aplicado um degrau de posição a cada junta do robô ITA-IEMP, segundo a Tabela 2. O sistema da Fig.6 foi simulado com o controlador robusto obtido na subseção 2.2.4 gerando as figuras 7 e 8. A Fig. 7 mostra as posições angulares e os erros de posição de cada junta quando não há adição de massa à sua extremidade. A Fig. 8 mostra as mesmas grandezas descritas na Fig. 7, com o robô submetido a um aumento de massa na extremidade equivalente a 100% da massa nominal do último elo. Em ambas as figuras, a curva tracejada representa a junta 1 (base), a curva cheia a junta 2 (braço) e a curva pontilhada a junta 3 (último elo).

Tabela 2. Posições iniciais e finais para as juntas e extremidade do robô ITA-IEMP

Espaço das Juntas			Espaço Cartesiano		
Nº da junta	Posição inicial (rad)	Posição final (rad)	Elemento do vetor	Posição inicial (m)	Posição final (m)
1	0.0	$\pi/4$	X	0.3320	0.3888
2	$\pi/2$	$\pi/4$	Y	0.0000	0.3888
3	$-\pi/2$	$-\pi/4$	Z	0.3080	0.2178

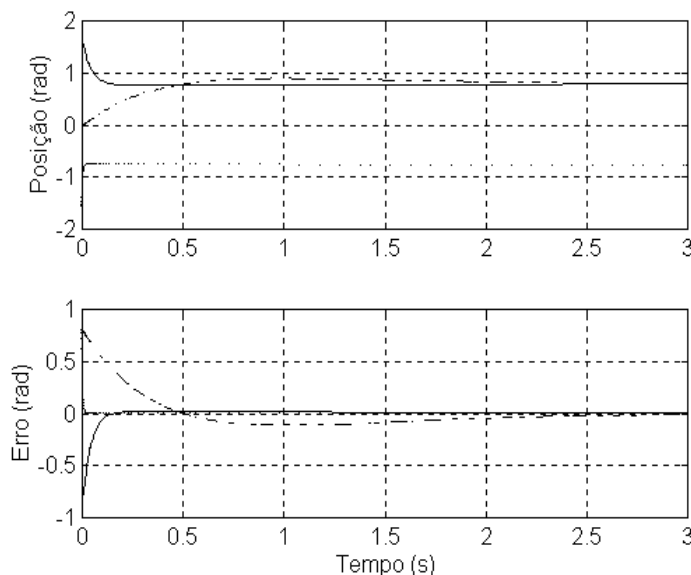


Figura 7. Posições e erros de posição para as juntas do robô ITA-IEMP considerando 0% de variação de massa no último elo e variação da trajetória dada pela Tabela 2

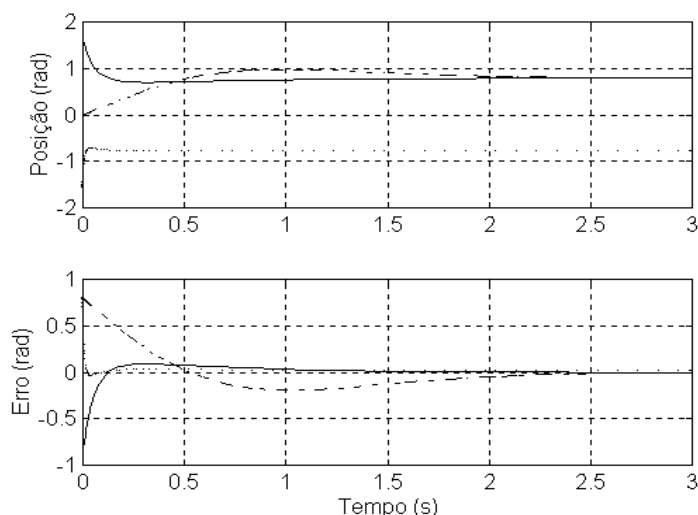


Figura 8. Posições e erros de posição para as juntas do robô ITA-IEMP com 100% de variação de massa no último elo e variação da trajetória dada pela Tabela 2

3. CONCLUSÃO

Este trabalho aborda o projeto de um controlador robusto H_∞ , que atua juntamente com um controlador via método do torque computado, utilizado para posicionar a extremidade de um manipulador robótico de elos rígidos seriais numa configuração de acionamento direto, quando submetido a variações de massa no órgão terminal.

Simulações utilizando técnicas de controle convencionais tipo PID e torque computado foram realizadas em (Góes, 2000), nas quais se constatou uma degeneração bastante grande do desempenho do sistema quando submetido a variações da massa na extremidade, especialmente para o controle via método do torque computado (vide Fig. 2). Como ilustra a Fig. 8 deste trabalho, para uma variação de massa de 100% na extremidade do manipulador, a associação com o controlador H_∞ torna a degeneração de desempenho bem menor.

4. REFERÊNCIAS

- Abdallah, C., Dawson, D., Dorato, P. and Jamshidi, M. (1991). Survey of Robust Control for Rigid Robots, *IEEE Control System Magazine*, february, pp. 24-30.
- Adade Filho, A. and Góes, E. (1998). Simulation of the dynamics and control of robotic manipulators using SIMULINK, *Proc. IASTED Int. Conf. on Modelling and Simulation*, Pittsburgh, Pennsylvania, pp. 236-240.
- Balas, G. J., Doyle, J. C., Glover, K., Packard, A. and Smith, R. (1994). *μ -Analysis and Synthesis Toolbox User's Guide*, The MathWorks, Natick, Mass.
- Craig, J. J. (1986). *Introduction to Robotics*, Addison-Wesley, Mass.
- Glover, K. and Doyle, J. C. (1988). State-space formulae for all stabilizing controllers that satisfy an H_∞ -norm bound and relations to risk sensitivity, *Systems & Control Letters*, North, Holland, Vol. 11, pp. 167-172.
- Góes, E. (2000). *Controle Robusto de um Manipulador Robótico*. Tese de Mestrado, Instituto Tecnológico de Aeronáutica (ITA).
- Maciejowski, J. M. (1989). *Multivariable Feedback Design*, Addison-Wesley, U.K.
- Skogestad, S. and Postlethwaite, I. (1996). *Multivariable Feedback Control - Analysis and Design*, John Wiley & Sons, Chichester, U.K.

APÊNDICE A

DESCRIÇÃO BÁSICA DO ROBÔ ITA-IEMP.

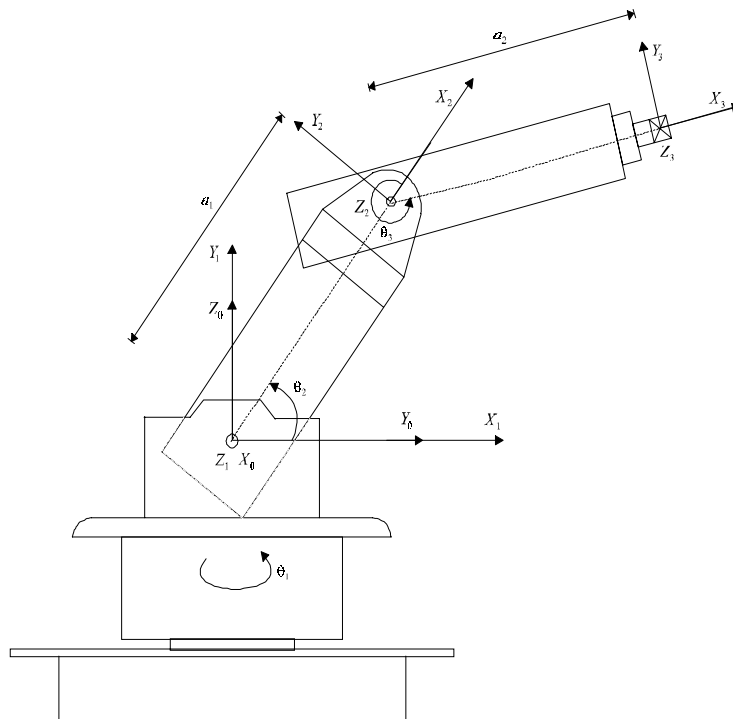


Figura 9 – Desenho Esquemático do robô ITA-IEMP

Tabela 3 - Parâmetros de Denavit-Hartenberg

Elo	α (graus)	a (m)	d	θ (var.)
1	90^0	0	0	θ_1
2	0^0	0.309	0	θ_2
3	0^0	0.332	0	θ_3

Tabela 4 - Propriedades de Massa

Parâmetros	Elo 1	Elo 2	Elo 3
CG (m)	(0, 0, -.074)	(-.28, 0, -.10)	(-.15, 0, 0)
I_{xx} (Kg*m ²)	0	0.2042	0
I_{yy} (Kg*m ²)	0.2629	0.9116	0.2385
I_{zz} (Kg*m ²)	0	0.7177	0.2363
I_{xy} (Kg*m ²)	0	0	0
I_{xz} (Kg*m ²)	0	-0.2431	0
I_{yz} (Kg*m ²)	0	0	0
Massa (Kg)	13.23	8.429	2.482

APÊNDICE B

DESCRIÇÃO NO ESPAÇO DE ESTADOS DO CONTROLADOR OBTIDO.

ak =									bk =		
1.0e+005 *									1.0e+007 *		
-0.0063	0	0	0	0	0	0	0	0	-0.0141	0.0000	0.0000
0	-0.0063	0	0	0	0	0	0	0	0.0000	-0.0141	0.0000
0	0	-0.0063	0	0	0	0	0	0	0.0000	0.0000	-0.0141
-0.9750	0	0	-0.0001	0	0	.0001	0	0	-2.1797	0.0000	0.0000
0	-0.9759	0	0	-0.0003	0	0	.0004	0	0.0000	-2.1797	0.0000
0	0	-1.17	0	0	-0.041	0	0	.027	0.0000	0.0000	-2.1797
0	0	0	0	0	0	0	0	0	0.0001	0.0000	0.0000
0	0	0	0	0	0	0	0	0	0.0000	0.0001	0.0000
0	0	.0001	0	0	0	0	0	0	0.0000	0.0000	0.0004

ck =									dk =		
-0.0710	0	0	-0.0177	0	0	.0573	0	0	0	0	0
0	-.495	0	0	-.12	0	0	.16	0	0	0	0
0	0	-87.1	0	0	-18.31	0	0	11.99	0	0	0

DESENVOLVIMENTO DE UM GERADOR DE TRAJETÓRIAS PARA UM ROBÔ INDUSTRIAL SCARA

Julio Feller Golin

Raul Guenther

Universidade Federal de Santa Catarina – Departamento de Engenharia Mecânica – CEP
88040-900 – Campus Universitário – Florianópolis – SC
golin@lcmi.ufsc.br, guenther@emc.ufsc.br

Resumo

Este trabalho trata do desenvolvimento de uma ferramenta para programação *offline* de um manipulador industrial. Foi desenvolvida uma interface gráfica para a geração de trajetórias de referência e a integração das informações ali obtidas com o ambiente de controle do robô através da transmissão de um arquivo de dados.

Palavras-chave: robótica, programação *offline*, geração de trajetórias.

1. INTRODUÇÃO

O Laboratório de Robótica da UFSC possui um robô industrial SCARA, para fins de pesquisa, cuja principal característica é ter uma arquitetura aberta. Com isso, tem-se a possibilidade de implementação e testes de algoritmos diversos para controle de posição, força, aquisição de sinais e comunicação de dados, ao contrário da maioria dos robôs industriais, em que o usuário não tem acesso ao ambiente de controle do manipulador. O ambiente computacional utilizado como interface com o robô e para sua programação é o XOberon.

Este robô chegou ao laboratório somente com as implementações dos módulos (softwares) imprescindíveis ao seu funcionamento básico. No caso da execução de trajetórias, estes módulos implementavam somente trajetórias ponto-a-ponto do tipo trapezoidal, sendo executadas *online* conforme a seqüência de comandos do usuário. Por ser textual, esta programação exige um usuário “experiente” e familiarizado com o espaço de trabalho do manipulador. Como cada comando corresponde a uma movimentação, quanto maior o número de pontos a atingir ou de trajetórias a executar, maior o número destes comandos.

Assim, para movimentar o manipulador, foi necessário o desenvolvimento de um planejador ou gerador de trajetórias, responsável por calcular a história temporal das posições, velocidades e acelerações desejadas (no espaço de juntas ou no espaço cartesiano). Este problema inclui as questões da formulação utilizada para geração de trajetórias de referência, que podem ou não ser em tempo real de movimentação, e da interface com o usuário. Elas serão discutidas respectivamente nas seções 2 e 3.

A geração de trajetórias trata da “estratégia” utilizada para o cálculo das trajetórias de referência. Para que uma trajetória possa ser realizada fisicamente pelo manipulador, ela deve obedecer às restrições impostas pelas características cinemáticas e dinâmicas do mesmo.

As trajetórias de referência podem ser classificadas em dois grandes grupos: as ponto-a-ponto e as por caminho contínuo ou *path motion* (Sciavicco & Siciliano 96). As trajetórias ponto-a-ponto caracterizam-se quando se deseja que o manipulador efetue o movimento apenas de uma posição inicial para uma posição final, como na movimentação de materiais. Já as trajetórias por caminho contínuo são necessárias em aplicações onde deseja-se que o manipulador passe por diversos pontos e sem velocidades nulas intermediárias, como por exemplo na soldagem a arco e no desvio de obstáculos.

Posto isso, busca-se neste trabalho desenvolver uma ferramenta gráfica que auxilie o usuário na tarefa da programação de trajetórias de referência para o manipulador mostrado na figura 1.

A programação é feita a partir de um conjunto reduzido de dados, como posições desejadas amostradas no espaço cartesiano e seus respectivos tempos, e então o gerador de trajetórias calcula as trajetórias de referência no espaço das juntas do manipulador. Estes resultados são posteriormente enviados ao controlador do robô na forma de um arquivo de dados.

O desenvolvimento desta ferramenta foi feito integrando o aplicativo Matlab (Windows) e o ambiente de programação XOberon (compilado para PowerPC). Em Matlab foi desenvolvido o aplicativo, isto é, o gerador de trajetórias. Em XOberon desenvolveram-se rotinas para leitura e conversão do arquivo de dados e execução das trajetórias de referência.

2. TRAJETÓRIAS DE REFERÊNCIA

Em diferentes aplicações na robótica a quantidade de pontos pelos quais o efetuador final deve passar é muito maior que apenas os pontos inicial e final das trajetórias ponto-a-ponto. Com isso, o problema a resolver é: dados n pontos pelos quais o manipulador deve passar – chamados de caminho ou *path points* – gerar um polinômio interpolador de ordem $(n - 1)$ ou gerar uma série de curvas de baixa ordem que, concatenadas nos pontos do caminho, produzam uma trajetória suave e contínua. Optamos pela segunda alternativa já que, no primeiro caso, não é possível especificarem-se as velocidades inicial e final desejadas bem como tem-se comportamentos mais oscilatórios e grande demanda computacional para polinômios de alta ordem. Assim, utilizamos em nossa formulação polinômios cúbicos pois são o de ordem mínima para que se possa garantir continuidade de posição e velocidade nos pontos do caminho. No gerador de trajetórias deste trabalho foram feitas duas implementações utilizando as formulações de polinômios cúbicos. Uma delas, chamada “solução natural”, é descrita em Chapra & Canale (1992), Sedgewick (1983) e Qiulin (1987). Esta é a implementação mais usual de polinômios cúbicos, que garante continuidade de posição, velocidade e aceleração. As acelerações no início e no final do movimento são nulas. Entretanto, esta formulação não garante velocidades inicial e final nulas. Uma outra formulação, apresentada em Sciavicco & Siciliano (1996) e chamada “solução por pontos virtuais”, atende esta condição.

Estas implementações são bastante conhecidas e não são reapresentadas. Apresenta-se somente o sistema sistema de equações da “solução por pontos virtuais” desenvolvido em Sciavicco & Siciliano (1996) e sua solução.



Figura 1. Robô SCARA

Na formulação de Sciavicco & Siciliano (1996), são estabelecidas as condições de continuidade de posição, velocidade e aceleração entre segmentos vizinhos através de polinômios de posição para o k -ésimo segmento¹ dados por

$$P_k(t) = p_{3k}t^3 + p_{2k}t^2 + p_{1k}t + p_{0k} \quad (1)$$

suas derivadas primeira e segunda e condições iniciais de posição (q_i e q_f), velocidade e aceleração.

Para um conjunto de n pontos, estas condições vão resultar num sistema acoplado com infinitas soluções. Para resolver a indeterminação são dadas duas condições a mais no tempo – os chamados “pontos virtuais”. Estes valores temporais são inseridos entre os dois primeiros (t_a) e os dois últimos (t_b) pontos da seqüência original, resultando em nova indexação destes valores². Com isso o sistema resulta com $4n + 4$ equações para $4n + 4$ incógnitas³. Conforme apresentado em Sciavicco & Siciliano (1996), a forma numericamente eficiente de se determinar os coeficientes dos $n+1$ polinômios interpolantes é a partir do cálculo das acelerações $\ddot{P}_k(t)$. Como o polinômio genérico $P_k(t)$ é uma cúbica, sua segunda derivada é uma função linear no tempo que pode ser escrita como

$$\ddot{P}_k(t) = \frac{\ddot{P}_k(t_k)}{\Delta t_k}(t_{k+1} - t) + \frac{\ddot{P}_k(t_{k+1})}{\Delta t_k}(t - t_{k+1}), \quad k = 2, \dots, n+1 \quad (2)$$

com $\Delta t_k = (t_{k+1} - t_k)$, onde os valores a determinar são as acelerações.

Neste trabalho, a partir de (2), foi construído um sistema de equações lineares $\mathbf{Ax} = \mathbf{b}$, sendo \mathbf{x} o vetor com as acelerações intermediárias a determinar. A matriz \mathbf{A} é tridiagonal e idêntica para todas as juntas do manipulador, pois seus coeficientes dependem apenas dos intervalos de tempo especificados pelo usuário. O vetor \mathbf{b} é dado por termos conhecidos, associados aos tempos de cada posição. Os elementos destas matrizes são:

Diagonal principal de \mathbf{A} :

$$a_{11} = 2(\Delta t_2 + \Delta t_1) + \frac{\Delta t_1^2}{\Delta t_2} + \Delta t_1, \quad a_{kk} = 2(\Delta t_{k+1} + \Delta t_k) \quad \text{para } k = 2, \dots, n-1 \text{ e}$$

$$a_{nn} = 2(\Delta t_{n+1} + \Delta t_n) + \frac{\Delta t_{n+1}^2}{\Delta t_n} + \Delta t_{n+1}.$$

$$\text{Diagonal inferior de } \mathbf{A}: a_{21} = \Delta t_2 - \frac{\Delta t_1^2}{\Delta t_2} \quad \text{e} \quad a_{k,k-1} = \Delta t_k \quad \text{para } k = 3, \dots, n.$$

$$\text{Diagonal superior de } \mathbf{A}: a_{k-1,k} = \Delta t_{k+1} \quad \text{para } k = 1, \dots, n-2 \quad \text{e} \quad a_{n-1,n} = \Delta t_n - \frac{\Delta t_{n+1}^2}{\Delta t_n}.$$

¹ Para os n pontos dados do caminho têm-se $(n - 1)$ segmentos interpoladores.

² A seqüência original t_1, t_2, \dots, t_n é reindexada como $t_1, t_2 \equiv t_a, t_3, \dots, t_n, t_{n+1} \equiv t_b, t_{n+2}$.

³ Notar que os $n - 2$ pontos intermediários, nomeados $k = 3, \dots, n$, resultam $4(n - 2)$ equações análogas a (1).

Elementos de **b**:

$$b_1 = 6 \left[\frac{q_3 - q_1 - \dot{q}_1 \Delta t_1 - (\ddot{q}_1 \Delta t_1^2)/3}{\Delta t_2} - q_1 - \frac{\ddot{q}_1 \Delta t_1}{2} \right]$$

$$b_2 = 6 \left[\frac{q_4 - q_3}{\Delta t_3} - \frac{q_3 - q_1 - \dot{q}_1 \Delta t_1 - (\ddot{q}_1 \Delta t_1^2)/3}{\Delta t_2} \right]$$

$$b_k = 6 \left[\frac{q_{k+2} - q_{k+1}}{\Delta t_{k+1}} - \frac{q_{k+1} - q_k}{\Delta t_k} \right] \quad \text{para } k = 3, \dots, n-2$$

$$b_{n-1} = 6 \left[\frac{q_{n+2} - q_n - \dot{q}_{n+2} \Delta t_{n+1} - (\ddot{q}_{n+2} \Delta t_{n+1}^2)/3}{\Delta t_n} - \frac{q_n - q_{n-1}}{\Delta t_{n-1}} \right]$$

$$b_n = 6 \left[q_{n+2} - \frac{\ddot{q}_{n+2} \Delta t_{n+1}}{2} - \frac{q_{n+2} - q_n - \dot{q}_{n+2} \Delta t_{n+1} - (\ddot{q}_{n+2} \Delta t_{n+1}^2)/3}{\Delta t_n} \right].$$

Resolvendo este sistema linear, obtêm-se os valores das acelerações intermediárias. Com isso as posições dos pontos virtuais são: $q_2 = q_1 + \frac{1}{6} \ddot{P}_1(t_2) \Delta t_1^2$ e $q_{n+1} = q_{n+2} + \frac{1}{6} \ddot{P}_n(t_{n+1}) \Delta t_{n+1}^2$. Integrando (2) duas vezes obtêm-se os polinômios de velocidade (3) e posição (4) para cada segmento da trajetória que deve ser executada por uma junta:

$$\dot{P}_k(t) = -3\alpha_k(t_{k+1} - t)^2 + 3\beta_k(t - t_k)^2 - \gamma_k + \delta_k \quad (3)$$

$$P_k(t) = \alpha_k(t_{k+1} - t)^3 + \beta_k(t - t_k)^3 + \gamma_k(t_{k+1} - t) + \delta_k(t - t_k) \quad (4)$$

onde $\alpha_k = \ddot{P}_k(t_k)/6\Delta t_k$, $\beta_k = \ddot{P}_k(t_{k+1})/6\Delta t_k$, $\gamma_k = P_k(t_k)/\Delta t_k - \frac{1}{6} \ddot{P}_k(t_k) \Delta t_k$ e $\delta_k = P_k(t_{k+1})/\Delta t_k - \frac{1}{6} \ddot{P}_k(t_{k+1}) \Delta t_k$.

3. IMPLEMENTAÇÃO DE UMA INTERFACE PARA PROGRAMAÇÃO OFFLINE

O desenvolvimento da interface trata de como o usuário pode especificar, de maneira simples, um conjunto mínimo de dados que é utilizado no cálculo das trajetórias de referência. Estas informações incluem o caminho desejado – isto é, os pontos espaciais que o manipulador deve seguir – podendo também especificar parâmetros como tempo, orientação, velocidade e/ou aceleração em cada ponto do caminho. Tipicamente, esta descrição é feita no espaço cartesiano onde a tarefa a ser executada pode ser representada de forma mais “natural”.

Neste trabalho foi desenvolvida uma interface no aplicativo Matlab onde o usuário insere uma seqüência de pontos, através do *mouse*, na área de trabalho do robô, bem como os respectivos instantes de tempo e outras informações conforme a técnica de interpolação escolhida. A escolha por Matlab deve-se a dois fatores: a programação de trajetórias em XOberson requer capacidades gráficas que o sistema não dispõe e a programação *offline* permite uma análise prévia dos resultados gerados e deixe o robô livre para outras

atividades, enquanto esta programação é feita. Além disso, outra necessidade nesta etapa de especificação da tarefa é a comparação e análise de resultados teóricos, sua visualização e testes, e o Matlab atende bem estes requisitos.

A figura 2 ilustra a inserção dos pontos no espaço de trabalho do robô no plano xy e eixo z . Os valores de referência da orientação são inseridos de forma semelhante. Esta seqüência corresponde aos pontos no espaço operacional que se deseja que o efetuador final atinja. Através da cinemática inversa estes pontos são convertidos em posições de cada junta. A partir daí, conforme o tipo de trajetória escolhido – “natural” ou por “pontos virtuais” –, são calculados os respectivos perfis de posição, velocidade e aceleração e gerados dois arquivos de saída com os resultados. Um arquivo contém os perfis de posição e velocidade amostrados a cada 1 ms para cada junta, período equivalente ao *clock* do controlador. O outro contém os coeficientes dos respectivos polinômios de posição para cada segmento da trajetória de cada junta. Um destes arquivos é posteriormente enviado ao controlador do robô para execução dos movimentos. A definição destes arquivos é discutida adiante.

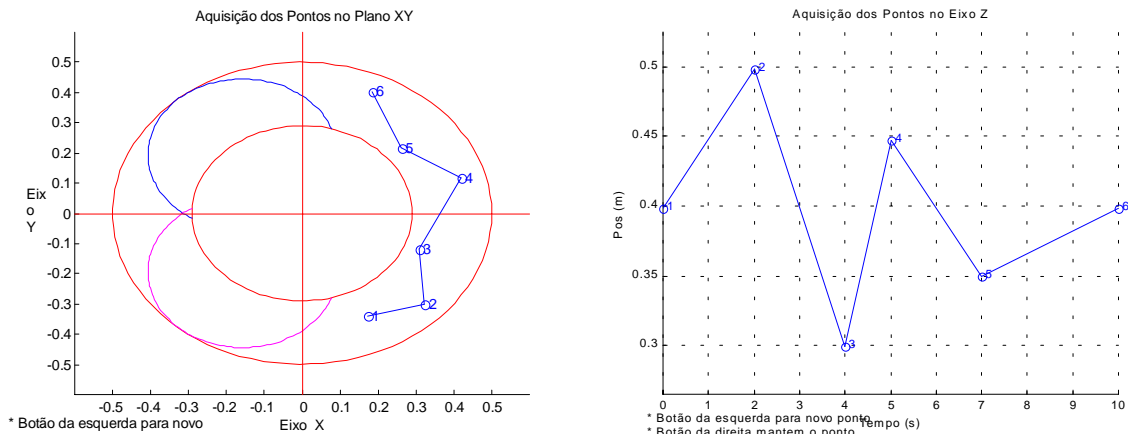
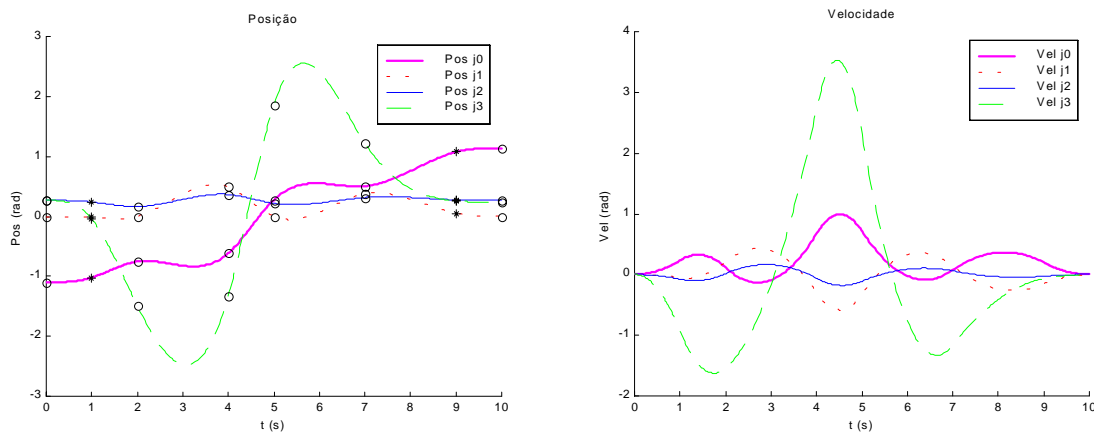


Figura 2. Representação do espaço de trabalho do robô Inter no plano xy (esq.) e no eixo z (dir.) para aquisição dos pontos do caminho.

Na figura 3 são mostrados os perfis de posição, velocidade e aceleração no espaço de juntas do manipulador, para o caminho definido acima, utilizando a formulação dos polinômios cúbicos por “pontos virtuais” bem como a trajetória de referência do efetuador final em seu espaço de trabalho.



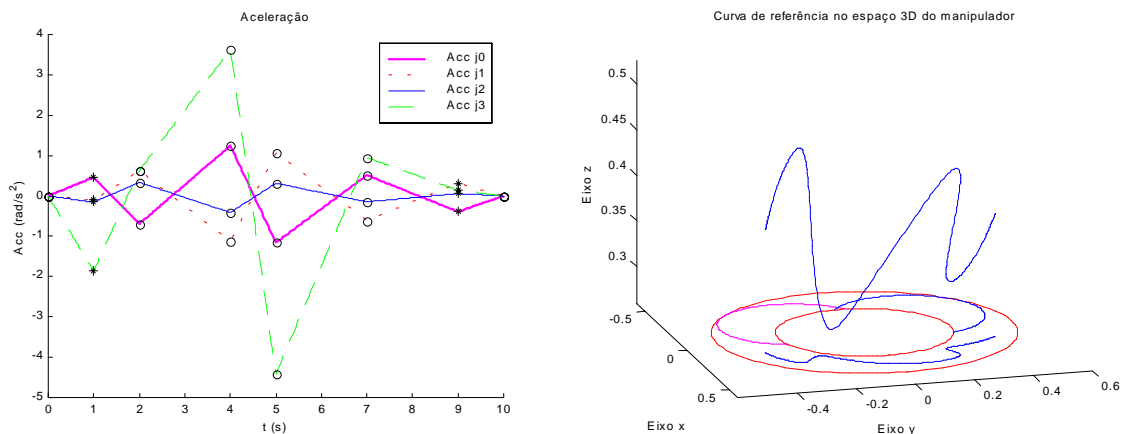


Figura 3. Perfis de posição, velocidade e aceleração no espaço de juntas calculados pelo gerador *offline* e trajetória de referência resultante no espaço operacional.

4. COMUNICAÇÃO XOBERON – MATLAB

Como o aplicativo Matlab não faz parte do sistema XOberon, é preciso estabelecer uma comunicação entre estes dois ambientes de modo que se possa movimentar o robô de acordo com os resultados obtidos na fase da especificação. Uma solução proposta e implementada para este caso é a criação de um arquivo padronizado e compatível para ambos os sistemas (Matlab e XOberon), que quando carregado para o controlador do robô possa ser corretamente lido e interpretado. As informações deste arquivo são utilizadas para a posterior execução dos movimentos desejados. A transferência deste arquivo para o controlador do robô é feita através de TFTP (*trivial file transfer protocol*), em que o computador local executa o processo servidor e o armário de controle do robô o cliente.

No XOberon é possível definir eventos (tarefas) e o tempo de execução destes eventos. Os dois tipos de eventos mais utilizados na programação de sistemas em tempo real são os que chamamos *main event* e *every event*. O *main event* é utilizado para implementação de tarefas que são executadas com alta prioridade porém somente quando houver tempo computacional disponível, isto é, quando não houver uma tarefa de tempo crítico sendo executada – por isto são consideradas tarefas de tempo não crítico. O *every event* implementa tarefas de tempo crítico e com a maior prioridade – neste caso a tarefa tem de ser obrigatoriamente executada e os resultados disponibilizados no tempo especificado.

No nosso caso, o objetivo é desenvolver módulos que leiam um arquivo pré-definido e já carregado para a memória do manipulador, processe adequadamente estes dados e envie sinais de referência de posição e velocidade de cada junta ao controlador do robô. Isto envolve basicamente a instalação de duas tarefas, a partir de um procedimento responsável pelo gerenciamento destas:

- uma tarefa *every event*, para o cálculo dos valores de referência e envio ao controlador, e
- uma tarefa *main event*, para supervisão dos estados do robô e a desinstalação do *every event* quando finalizada a trajetória desejada.

Para leitura e interpretação do arquivo de referência foram desenvolvidas uma série de rotinas em XOberon que executam as seguintes tarefas, mostradas de forma genérica na

figura 4: a) leitura e verificação da integridade do arquivo de dados; b) conversão dos dados de caracteres para representações numéricas reais; c) instalação do processo com o controlador escolhido; d) posicionamento do manipulador para o início da trajetória; e) envio dos valores de referência para o controlador; f) procedimentos de segurança e supervisão de processos.

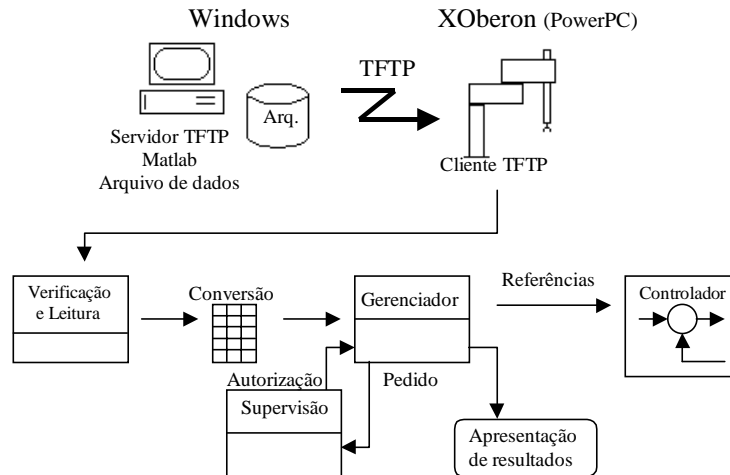


Figura 4. Diagrama de relacionamento das rotinas do gerador de trajetórias, implementadas em XOberon.

Como são gerados dois arquivos de dados para uma mesma trajetória, conforme mencionado na seção 3, e um deles é transmitido ao controlador do robô, eles devem seguir uma padronização. Esta padronização deve ser tanto no conteúdo, isto é, em como os dados são armazenados (no nosso caso foi utilizado o padrão ASCII) como na sua organização dentro do arquivo, para que se possa saber o que cada conjunto de números representa. No caso da leitura direta de valores de referência, foi adotado o seguinte padrão:

$$\begin{bmatrix} \theta_0 & \dots & \theta_3 & \dot{\theta}_0 & \dots & \dot{\theta}_3 \\ \vdots & & \vdots & \vdots & & \vdots \\ \theta_0 & \dots & \theta_3 & \dot{\theta}_0 & \dots & \dot{\theta}_3 \end{bmatrix}$$

onde cada linha representa os valores a serem enviados ao controlador a cada 1ms. É interessante ressaltar que este padrão pode ser adotado para a geração de trajetórias de qualquer perfil, ou seja, independente do método utilizado para gerá-las, desde que se utilize o controlador PD originalmente implementado ou um equivalente a este. Para o arquivo de dados composto pelos coeficientes dos polinômios de posição para cada junta, em cada segmento foi adotado o seguinte padrão:

$$\begin{bmatrix} a_{31k} & a_{21k} & a_{11k} & a_{01k} & \dots & a_{34k} & \dots & a_{04k} & t_k \\ & & & & \vdots & & & & \\ a_{31n-1} & a_{21n-1} & a_{11n-1} & a_{01n-1} & \dots & a_{34n-1} & \dots & a_{04n-1} & t_n \end{bmatrix}$$

onde a_{31k} representa o coeficiente a_3 da junta 1 no k -ésimo segmento. Assim, cada linha representa um segmento de duração de t_k segundos, para as quatro juntas.

Uma potencial limitação do uso do arquivo com valores de referência é o seu tamanho, já que para cada 1s de trajetória são necessárias 1000 linhas de dados. Assim,

quanto mais demorada for a trajetória de referência, mais memória disponível deve-se ter no robô. Já para o arquivo de coeficientes pode-se considerar esta limitação como inexistente, já que os arquivos gerados são bastante pequenos.

5. RESULTADOS

Foram testadas trajetórias em diferentes regiões da área de trabalho do manipulador. O controle é independente por juntas, do tipo PD (Sciavicco & Siciliano 1996). Os ganhos destes controladores e as implementações podem ser vistos em Golin *et al.* (1998). Com este controlador verificou-se que a transmissão e tratamento dos dados foram feitos corretamente e que houve seguimento de trajetória. Isto comprova experimentalmente a validade das implementações.

6. CONCLUSÕES

Neste trabalho foi desenvolvida uma ferramenta gráfica para a programação *offline* de um manipulador industrial SCARA. Esta ferramenta possibilita a geração de trajetórias suaves e contínuas no espaço das juntas do manipulador, interpolando e atingindo os pontos do caminho desejado, especificados pelo usuário no espaço cartesiano. Desenvolveu-se uma metodologia para a transmissão de dados entre o Matlab e o ambiente de controle do robô (XOberon).

Por fim, este trabalho vem possibilitando a realização de outras atividades práticas com este manipulador que necessitam de resultados obtidos em aplicativos como o Matlab.

7. REFERÊNCIAS

- Angeles, Jorge, 1997, “Fundamentals of Robotic Mechanical Systems”, Springer-Verlag.
- Bier, C.C., Cunha, A.E.C., Martins, D., Passold, F., 1998, “Planejamento de Trajetória”, Relatório Interno – Laboratório de Robótica da UFSC.
- Chapra, S. e Canale, R., 1992, “Numerical Methods for Engineers”, McGraw-Hill, New York.
- Craig, John J., 1986, “Introduction to robotics mechanics & control”, Addison-Wesley, Reading, MA.
- Dias, A., Toledo, L. e Guenther, R., 1998, “Um Sistema CAD/CAM para a Programação Fora de Linha de Manipuladores”, V CEM-NNE (Congresso de Eng. Mecânica do Norte-Nordeste)
- Golin, J., Weihmann, L. e Guenther, R., 1998, “Manual do Usuário do Robô Inter”, documento interno do Laboratório de Robótica da UFSC.
- The Mathworks, Inc., 1997, “Matlab – Versão do Estudante – Guia do Usuário”, Makron Books, São Paulo.
- Qiulin, D. e Davies, B.J., 1987, “Surface Engineering Geometry for Computer-Aided Design and Manufacturing”, Ellis Horwood Limited.
- Sciavicco, L. e Siciliano, B., 1996, “Modeling and Control of Robot Manipulators”, McGraw-Hill, New York.
- Sedgewick, R., 1983, “Algorithms”, Addison-Wesley, Reading, MA.
- Vestli, S., 1997, “XOberon”, IfR, ETH, Zurique.

ARQUITETURA PARA PEQUENAS E MÉDIAS EMPRESAS ORGANIZADAS HOLONICAMENTE

Gustavo Nucci Franco
Antonio Batocchio

Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Departamento de Engenharia de Fabricação, Campinas – SP – Brasil, email: nfranco@fem.unicamp.br

Resumo

Para que empresas sobrevivam em um ambiente de crescente competitividade, essas devem se adaptar às exigências de seu ambiente externo. Isso significa rápida incorporação de novas tecnologias, novos produtos, novas estruturas, etc., de acordo com as necessidades do mercado. Surge, então, a questão de como reagirão as pequenas e médias empresas (PMEs) frente a concorrência das grandes empresas globais. Assim, este trabalho propõe uma arquitetura que possibilite maior agilidade para PMEs. Essas empresas serão modeladas através do paradigma holônico, proporcionando-lhes diferenciais competitivos.

Palavras chave. Paradigma holônico, sistema baseado em agentes, pequena e média empresa, modelagem de processos de negócio.

1. INTRODUÇÃO

Indústrias de diversos setores têm passado por mudanças nas últimas duas décadas. Os mercados estão se tornando cada vez mais voláteis e imprevisíveis, levando a necessidade de lotes menores e demanda crescente de produtos customizados de baixo custo. Esses fatores se agravam a medida que a globalização da economia amplia o espaço de competição, aumentando a já acirrada concorrência entre as empresas. Fica evidente que aqueles que não alavancarem sua vantagem competitiva verão diluídas suas aspirações e alternativas de sobrevivência. Diante desse cenário, resta a questão de como as pequenas e médias empresas (PMEs) se posicionarão em um futuro próximo.

Novos sistemas devem ser buscados, passando as estratégias de desenvolvimento a ter uma importância cada vez maior na competitividade industrial. Mais do que um fator diferenciador, a competência tecnológica é qualificadora para as empresas que desejarem obter sucesso em um futuro próximo.

Uma resposta a esses desafios é o projeto internacional denominado Sistema Holônico de Manufatura (HMS), que vem desenvolvendo sistemas de manufatura através da integração de unidades modulares altamente flexíveis, ágeis e reusáveis. O objetivo do HMS a longo prazo é desenvolver sistemas flexíveis e adaptativos para manufatura. Essa geração de sistemas de manufatura formará fábricas distribuídas, reconfiguráveis e virtuais, nas quais homem, máquina e computador formarão blocos dinâmicos [HMS, 2000].

A partir de conclusões tiradas sobre a função dos países que ficarem a margem do processo de desenvolvimento, em 1998 formou-se no Brasil o Grupo de Pesquisa em Sistemas

Holônicos de Manufatura [GPHMS, 2000]. Trata-se de um grupo de pesquisa que visa investigar e traduzir para a realidade brasileira as tecnologias desenvolvidas a partir do paradigma holônico. Parte dos trabalhos do GPHMS diz respeito a soluções para PMEs.

Este trabalho propõe uma arquitetura para PMEs desenvolvida do estudo dos princípios desenvolvidos para o Sistema Holônico de Manufatura (HMS). Para isso, os principais princípios serão inicialmente apresentados. Os sistemas baseados em agentes serão mostrados como uma forma de se garantir a cooperação entre os elementos da arquitetura. Em seguida, o paradigma holônico será apresentado, dando bases fundamentais para que uma arquitetura para PMEs seja desenvolvida.

2. SISTEMAS BASEADOS EM AGENTES

Agentes são um novo paradigma para o desenvolvimento de aplicativos, podendo levar a uma revolução na engenharia de software. Atualmente o estudo de sistemas baseados em agentes tem sido foco da ciência da computação, sobretudo em seu campo da inteligência artificial (AI), mas seu interesse vêm crescendo em outras áreas.

Existem várias utilizações para o termo agente, cada uma levando a uma noção substancialmente diferente de agência. De acordo com Shardlow (1990):

‘Agentes fazem coisas, agem: é por isso que se chamam agentes’.

Trata-se de uma tarefa ativa que origina ações que afetam o ambiente, ao invés de permitirem passivamente que o ambiente os afete.

Uma característica importante no comportamento de agentes é sua autonomia. Agentes devem ser capazes de executar tarefas sem a intervenção direta de outros agentes, além de controlar suas ações e estados internos. Para isso, os agentes são dotados de uma flexibilidade que lhes permite selecionar ações que atendam aos objetivos do sistema. Assim, além de autonomia, os agentes possuem:

- *correspondência*: devem ter percepção sobre seu ambiente (o mundo físico, um usuário, um conjunto de agentes, etc.) e devem responder em tempo hábil às mudanças que ocorrem nesse ambiente.
- *proatividade*: não agem apenas em resposta a seu ambiente, devendo ser capazes de mostrar oportunismo, ser direcionados a objetivos e tomar iniciativas quando necessário.
- *sociabilidade*: devem ser capazes de interagir com outros agentes, possuindo comportamento humanos para solucionar seus problemas e ajudar outros agentes em suas atividades.

Incorporando essas características, pode-se formar sistemas com desempenho melhor, verdadeiras sociedades chamadas de sistemas baseados em agentes. A questão chave nesses é a conciliação da autonomia e da cooperação, minimizando os conflitos e chegando-se aos objetivos determinados.

2.1 Agentes como sistemas intencionais

Até aqui, os agentes foram associados às ações que realizam. Embora ações autônomas estejam diretamente envolvidas na sua definição, essas dificilmente podem ser critério único para definir o comportamento de um agente, uma vez que qualquer ação pode ser descrita de diversas formas. Por exemplo, ao acenar para um amigo, o braço é levantado, músculos são contraídos, neurônios são ativados, ... A análise de um agente meramente por suas ações aparentemente não é prática. Assim, deve-se buscar outras formas para definir o comportamento de um agente.

Tomando como objeto de comparação o comportamento humano, esse pode ser previsto e explicado pela atribuição de atitudes como acreditar, querer, escolher, temer, etc. Por

exemplo [Wooldridge & Jennings, 1994]: “*Janine levou seu guarda chuva porque acreditava que iria chover*” ou “*Michael trabalhou duro pois ele queria seu título de PhD*”. Essas atitudes são as noções intencionais e as entidades cujo comportamento pode ser previsto por essa atribuição foram chamadas por Dennett (1987) de sistemas intencionais.

Se um sistema intencional pode ser uma condição necessária e suficiente para descrever o comportamento humano, sua utilização na modelagem de agentes pode ser uma solução. No entanto, atribuir intenções a agentes artificiais como máquinas é questionável, merecendo uma análise e fundamentação antes de ser realizada. McCarthy (1978) argumenta que é legítimo atribuir desejos, vontade própria, intenções, consciência ou habilidades a uma máquina quando essa atribuição expressa a mesma informação sobre a máquina que se expressa sobre uma pessoa. Ainda, essa atribuição se torna útil quando ajuda no entendimento da estrutura da máquina, de seu comportamento passado e futuro.

2.2 O modelo BDI

As noções intencionais são ferramentas de abstração que provêm uma maneira conveniente e familiar de descrever, explicar e prever o comportamento de sistemas. Sua atribuição é mais correta para sistemas de estrutura conhecida, mas é mais útil para estruturas complexas e completamente desconhecidas. Surgem, assim, vários modelos que tentam explicar de maneira formal a racionalização dos agentes. Um dos modelos de maior aceitação, estudo e implementação é o BDI (*belief-desire-intention model*) [Bratman, 1987].

O modelo BDI foi introduzido em meados dos anos 80, sendo mantido inalterado desde então. Desde seu desenvolvimento, houve muitas alterações em áreas de interesse como a AI, fazendo com que surgissem muitas outras propostas. Ainda assim, esse modelo continua sendo muito utilizado. Segundo Georgeff et al. (1998) isso se deve a combinação de um respeitável modelo filosófico sobre o comportamento humano, várias implementações, aplicações bem sucedidas e uma semântica lógica abstrata e elegante.

O modelo BDI utiliza termos que se relacionam a crença, desejo e intenção para representar componentes habituais (estados, objetivos e ações), uma vez que os agentes, ao contrário dos objetos, consideram o trabalho com informações imperfeitas. Assim, os componentes que formam o modelo BDI são:

- *Crenças (beliefs)*: representam o componente informativo do ambiente. Em termos computacionais, informam o estado do ambiente através de valores de variáveis, bancos de dados relacionais, expressões lógicas em cálculo predicativo ou outra estrutura de dados. Devido à dinâmica do ambiente, seu não determinismo e a visão local dos agentes, esse atributo deve ser alterado de acordo com ações sensíveis.
- *Desejos (desires)*: são os objetivos que indicam o estado final a ser alcançado. Os processos em sistemas baseado em agentes são orientados a objetivo, e não a tarefas, sendo, assim, realizados de uma forma consciente e de acordo com o estado desejado. Desse modo, cria-se um processo mais flexível, capaz, inclusive, de recuperar falhas e aproveitar oportunidades não inseridas explicitamente no processo.
- *Intenções (intentions)*: formam o componente deliberativo do sistema. Uma vez que o sistema deve agir, as ações ou procedimentos apropriados devem ser selecionados entre diversas possibilidades. Trata-se de uma função, projetada de modo a permitir que o sistema atinja seus objetivos, dados os recursos disponíveis e o estado do ambiente. Suas respostas são seqüências de comandos.

Pode-se, então, pensar em um agente como um elemento que consiste de crenças sobre o mundo real, desejos que precisam ser satisfeitos e intenções a serem adotadas em resposta a eventos externos ou objetivos internos. Devido a limitação de recursos, um sistema pode armazenar planos genéricos e parametrizados para serem utilizados em situações futuras (tal

qual o ocorre com as intenções). Trata-se de uma biblioteca de *planos* semanticamente idênticos às intenções. Cada *plano* é uma especificação abstrata de seqüências possíveis de ações que podem ser utilizadas em ocasiões pré definidas. O conjunto de *planos* pode, ainda, limitar as ações de uma agente.

3. SISTEMAS HOLÔNICOS

O neologismo holon surgiu do trabalho de Arthur Koestler que, há mais de 30 anos, observou e descreveu propriedades importantes de sistemas complexos, a estrutura e as relações entre seus componentes. Em suas obras, Koestler discute numa visão sociológica a natureza de sistemas como o cérebro humano, sociedades e organismos vivos.

Koestler (1967) introduziu a palavra holon, combinação de *holos* (do grego todo) com o sufixo *on* (parte) para indicar *partes* de um sistema que possuem o comportamento de *todos*. Seu trabalho parte do princípio de que sistemas podem ser mais eficazes se forem compostos por subsistemas intermediários estáveis [Simon, 1966]. Naturalmente, sistemas complexos, como os animais, são sempre estruturados como uma hierarquia com vários níveis de formas estáveis. São células que formam órgãos, que por sua vez se agrupam para formar sistemas com fisiologias bem definidas, levando a um ser humano. Em seguida, percebeu que nessas hierarquias, mesmo sendo fácil de se identificar os subsistemas e seus elementos, *todos* e *partes* em um senso absoluto não existem. Assim, os holons enfatizam a natureza híbrida dos sistemas reais: são ao mesmo tempo *todos* auto assertivos e *partes* dependentes vistas de um nível superior.

Outro termo introduzido por Koestler foi holarquia, o qual representa o sistema formado por holons cooperativos, organizados de tal forma que estabelecem ambientes operacionais e interfaces externas. Etimologicamente, hierarquia é uma palavra com significado voltado a relações de subordinação, o que leva a estruturas rígidas. As holarquias são estruturas abertas, onde um todo global pode dinamicamente fazer parte de uma ou várias outras estruturas.

O paradigma holônico vem sendo estudado para que novas tecnologias possam ser desenvolvidas. Por potencialmente proverem sistemas inteligentes, flexíveis e reconfiguráveis, muitas aplicações podem surgir com a utilização dessas tecnologias.

3.1 Estática de sistemas complexos

A análise de Koestler (1967) está baseada em uma descrição estática de sistemas complexos. A princípio, os holons possuem uma tendência a manter sua individualidade, uma vez que são *todos*. No entanto, devem funcionar como *partes* integrantes de um sistema maior. Desse modo, os holons possuem duas características vitais:

- *Autonomia*: a capacidade de uma entidade de criar e controlar a execução de seus próprios planos e/ou estratégias.
- *Cooperação*: o processo pelo qual um conjunto de entidades desenvolvem mutualmente planos exeqüíveis e os executam.

Assim como os agentes, os holons estão sujeitos a conflitos entre autonomia e cooperação. O equilíbrio é encontrado quando um holon balanceia suas tendências auto assertivas e integrativas. Para isso, Wyns (1999) fala na existência de um conjunto de regras que governam os holons, dando-lhes maior ou menor liberdade de ação. Ainda assim, os holons possuem flexibilidade para escolher, dentro de limites, a estratégia a ser adotada para atender aos objetivos do sistema. De acordo com o modelo BDI, isso equívale a dizer que os holons possuem um conjunto de *planos* que limitam (e ajudam) suas *intenções* de atender seus *desejos* de acordo com sua *crença* no sistema. Conclui-se que autonomia e cooperação em sistemas holônicos podem ser obtidas através de conceitos baseados em agentes.

Um dos diferenciais de um sistema holônico, o qual lhe garante aspectos próprios em relação aos sistemas baseados em agentes, é sua estrutura hierárquica. Os holons pertencem naturalmente a múltiplas hierarquias ao mesmo tempo. As holarquias são estruturas verticais em forma de árvore cujos ramos se interligam com outras holarquias, formando redes horizontais.

Sistemas holônicos devem ser capazes de associar alta eficiência com alta adaptatividade. Embora essa associação pareça ser contraditória, a solução pode ser obtida através de propriedades dinâmicas desse sistema como auto similaridade e auto organização. Assim, esse sistema se torna extremamente ágil.

3.2 Dinâmica de sistemas adaptativos

A capacidade de reconfiguração com agilidade vem estabelecendo um dos grandes diferenciais de alguns modernos sistemas de manufatura: a agilidade. Em sistemas holônicos, reconfigurar significa movimentar holons entre as diversas holarquias, introduzir novos holons e extrair holons desnecessários. A reconfigurabilidade está associada às propriedades de auto similaridade e auto configuração [Wyns, 1999].

Auto similaridade é uma característica importante que reduz a complexidade do sistema como um todo, facilitando a integração de diversos elementos. Os sistemas holônicos devem possuir componentes auto similares, ou seja, devem ser formados por conjuntos de holons do mesmo tipo (básico) que possuam comportamento e interfaces semelhantes. Essa tendência a auto similaridade em sistemas complexos está relacionada à tomada de decisões e ao processamento distribuído de informações. A identidade própria de cada holon dentro de um mesmo tipo básico é conseguida através da agregação de holons relacionados, habitando suas funções específicas (especializações). Defini-se, assim, auto similaridade horizontal, como sendo aquela dentro de um mesmo nível de agregação, e similaridade vertical, entre diferentes níveis de agregação.

Enquanto um sistema distribuído, deve haver mecanismos que garantam o alcance dos objetivos globais do sistema a partir dos objetivos locais dos holons. Além disso, os holons precisam construir e manter modelos de partes do sistema que sejam relevantes. A engenharia da computação, com o desenvolvimento de arquiteturas de computadores cada vez mais complexos, indicam algumas estratégias onde indivíduos cooperativos em ambientes competitivos proporcionem uma performance global superior a de sistemas formados por indivíduos não cooperativos. Alguns desses modelos direcionam para a reconfiguração do sistema, levando à propriedade de auto organização. Um exemplo são as cada vez mais avançadas arquiteturas *plug-and-play* que se tornaram um verdadeiro fator de marketing em resposta aos árduos trabalhos de instalação de novos sistemas (periféricos, placas, aplicativos, etc.) aos antigos modelos de computadores pessoais.

4. MODELO PARA PMES

O estudo do paradigma holônico pode gerar tecnologias que garantam vantagens competitivas para pequenas e médias empresas (PMEs). Uma típica PME pode ser descrita como uma estrutura casualmente holônica, onde a execução dos processos de negócio está baseada na colaboração e na autonomia de seus trabalhadores.

Toh et al. (1998) representam essa PME através de uma holarquia onde as interações são normalmente estabelecidas entre os recursos humanos, não sendo inclusos os recursos computacionais e operacionais (máquinas, equipamentos, etc.)

Um grande passo é dado com a transformação de uma PME em um sistema onde os holons são formados integrando-se homem-máquina-computador, estando esses conectados a um meio comum de comunicação. Para isso, deve ser definida uma arquitetura para o sistema,

especificando a estrutura holônica e indicando componentes, responsabilidades, dependências, interfaces, dados, interações e restrições.

4.1 Estrutura do sistema

A estrutura de uma arquitetura holônica para PMEs parte da identificação das funções de negócio da empresa. Basicamente, uma PME pode ser representada em seu nível mais macro pelas funções executivas, administrativas e manufatura. Esse é, então, tomado como o primeiro nível de agregação dos holons da PME. Traduzindo essa estrutura para uma linguagem formal chega-se a figura 1, para a qual foi utilizada a UML (*Unified Modelling Language*) de Rumbaugh et al. (1997) que padroniza diagramas para orientação a objetos. O diagrama indica, de acordo com o tipo de terminação das linhas ligando os blocos e os valores associados a essas, que uma PME é formada por pelo menos um holon executivo, pelo menos um holon administrativo e pelo menos um holon de manufatura. Pode-se verificar, ainda, as interações entre os diversos holons.

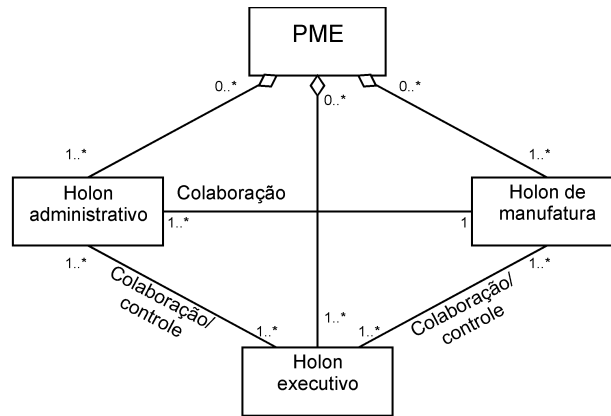


Figura 1 – Estrutura básica de uma PME holônica vista em modelo orientado a objeto

Para cada um dos holons que formam a PME, deve ser especificado um conjunto de holons básicos. Por exemplo, para o holon de manufatura, foi adotada a estrutura da arquitetura de referência PROSA (*Product-Resource-Order-Staff Architecture*) descrita por Wyns (1999) e cuja estrutura básica está representada na figura 2. De acordo com esse

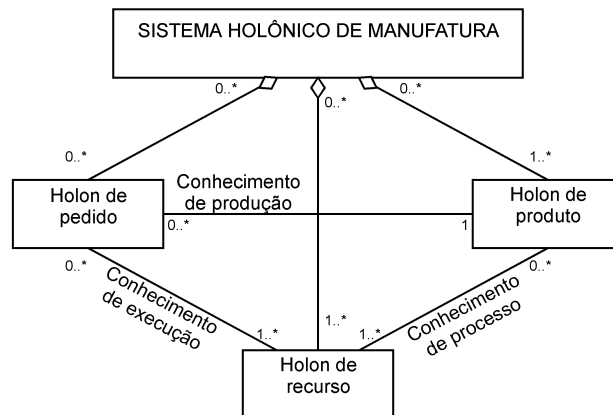


Figura 2 – Estrutura básica de um holon de manufatura segundo a PROSA [Wyns, 1999]

diagrama UML, a PROSA é construída sobre três tipos básicos de holons: holon de recurso, holon de produto e holon de pedido. As interações entre esses holons indica a base de conhecimento trocada entre esses: produção, execução e processo.

A arquitetura holônica para a PME prevê, ainda, a existência de holons de suporte, cuja função é a de auxiliar os holons básicos em funções específicas. Ainda assim, a função de tomada de decisão continua a ser de responsabilidade dos holons básicos, sendo o holon de suporte um especialista externo que provê informações suficientes. Um exemplo pode ser o holon de programação definido pela PROSA, que se comunica com os holons de pedido e os holons de recurso. O holon de programação recebe as informações dos holons de pedido, fazendo uma programação inicial que, a princípio, será seguida pelos holons de recurso. No entanto, quando a programação sugerida se torna inviável, como no caso das paradas das máquinas, os holons de recurso e pedido buscam um acordo que gere uma programação mais eficiente para a situação encontrada, retornando as informações ao holon de programação.

Para diminuir a complexidade do comportamento do sistema, os holons devem ser estruturados em hierarquias apropriadas, ou seja, devem ser criadas as agregações de acordo com as especialidades de algumas partes do sistema. Por exemplo, a PROSA define a agregação da figura 3, onde todos os holons do diagrama UML são holons de recurso (a seta indica uma relação de especialização).

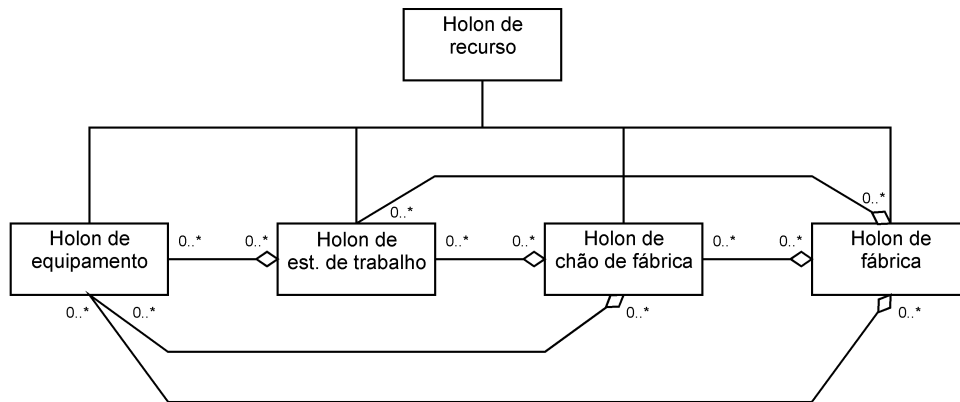


Figura 3 – Agregações do holon de recurso segundo a PROSA [Wyns, 1999]

4.2 Comunicação entre os holons

Uma vez definida a estrutura do sistema, os holons básicos devem ser detalhados através de modelos que determinem autonomia e cooperação entre esses e os demais holons. Dentro de uma arquitetura holônica, processos como negociação de utilização de recursos, planejamento de materiais e cálculo de custos envolvem a participação de vários holons trocando informações e serviços a fim de cumprir objetivos. Para isso, cada holon deve ter a habilidade de se comunicar com outros holons através de algum mecanismo.

Para um holon participar de uma holarquia é necessário que esse se comunique através de uma linguagem. Essa linguagem deve consistir de um protocolo e uma sintaxe para expressar as informações. Além disso, para um holon interpretar as intenções de outros holons, esse deve compartilhar um mesmo modelo de informação, ou conseguir traduzir informações entre diversos modelos, ou seja, cada linguagem deve possuir uma semântica.

A interoperabilidade é um dos pontos críticos para a implementação de sistemas distribuídos. Assim, negociar, cooperar e comunicar é um desafio a ser superado. Para isso, a teoria de sistemas baseados em agentes provê algumas soluções. Por exemplo, uma estrutura dos dados necessária pode ser definida aplicando-se o modelo BDI [Rao & Georgeff, 1995].

4. CONCLUSÕES

As indústrias têm observado mudanças no ambiente econômico que colocam em risco a sobrevivência de muitas empresas. Parece óbvio concluir que a atual estrutura econômica mundial cria um ambiente muito mais competitivo, o que exige mudanças organizacionais para adequar a empresa ao novo cenário.

O paradigma holônico mostra grandes potenciais para o desenvolvimento de novas tecnologias, inclusive para PMEs. Assim, um campo de muitas aplicações surge. Uma delas é o desenvolvimento de um sistema de informação que melhore o desempenho da empresa através de um ganho de agilidade. Desenvolvendo-se redes de informações a partir dos conceitos de autonomia e cooperação, pode-se chegar a sistemas realmente flexíveis, reconfiguráveis e ágeis, melhorando a produtividade alcançada.

5. AGRADECIMENTOS

Este trabalho foi possível com a ajuda da Fapesp (Fundação de Amparo à Pesquisa do Estado de São Paulo), que vem financiando esta pesquisa (Fapesp #98/10840-4), e ao Recope/Finep/BID pelo financiamento dos recursos computacionais utilizados neste trabalho.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- Bratman, M.E., 1987, *Intentions, plans and practical reason*. Harvard University Press.
- Dennett, D.C., 1987, *The Intentional Stance*. The MIT Press.
- GPHMS, 2000, Grupo de Pesquisa em Sistemas Holônicos de Manufatura. In www.def.fem.unicamp.br/~defhp/index.htm.
- HMS, 2000, Welcome to the Holonic Manufacturing System (HMS) Web Site. In hms.ifw.uni-hannover.de.
- Koestler, A. *The ghost in the machine*. London: Arkana Books, 1967.
- McCarthy, J., 1978, *Ascribing mental qualities to machines*. Technical report, Stanford AI Lab.
- Rao, A.S., Georgeff, M.P., 1995, BDI Agentes: from theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems – ICMAS-95*, San Francisco (CA), USA.
- Rumbaugh, Booch & Jacobsen, 1997, *The Unified Modelling Language (UML) ver 1.0*. Rational Co., California, USA.
- Shardlow, N., 1990, Action and agency in cognitive science. Master's thesis, Department of Psychology, University of Manchester, Manchester, UK.
- Toh, K.T.K., Newman, S.T., Bell, R., 1998, An information system architecture for small metal-working companies. In *Proc. Instn. Mech. Engrs.* v.212, part B, pp. 87-103.
- Wyls, J., 1999, *Reference Architecture for Holonic Manufacturing Systems: the key to support evolution and reconfiguration*. PhD thesis, Departement Werktuigkunde Afdeling Productietechnieken, Katholieke Universiteit Leuven, Leuven, Belgium.
- Wooldridge, M.J., Jennings, N.R., 1994, Agent Theories, Architectures, and Languages: a survey. In *Proc. ECAI-Workshop on Agent Theories, Architectures and Languages*, Amsterdam, The Netherlands, p.1-32.

UMA NOVA TECNOLOGIA PARA ACIONAMENTO DE MOTORES DE PASSO OPERANDO EM MICROPASSO

Solly A . Segenreich

Juan Paniagua Vargas

Pontifícia Universidade Católica do Rio de Janeiro – Depto. de Engenharia Mecânica- Rua Marq. São Vicente 225, 22.453-900 Rio de Janeiro – RJ

e-mail: solly@mec.puc-rio.br

Resumo

A crescente capacidade de regulação e a possibilidade de chaveamentos precisos das correntes de fase dos motores de passo, tem possibilitado o surgimento de acionamentos em micropasso cada vez mais eficientes. Com isto, a nociva, e, muitas vezes, limitadora trepidação dos motores de passo operando em passo inteiro ou meio passo, foi grandemente reduzida, fazendo com que este tipo de motores sejam uma solução viável em quase todos os projetos de automação e robótica.

Nos acionamentos convencionais, é fundamental o acompanhamento do nível da corrente na fase do motor. Este acompanhamento, por ser realizado em níveis baixos de tensão e na presença de ruído, acaba sendo um sério inconveniente exigindo uma montagem elaborada e eventuais filtros. O presente trabalho apresenta uma proposta de um novo acionamento em que a necessidade do acompanhamento da corrente é eliminado. Os primeiros testes indicam a viabilidade do novo conceito, apresentando resultados extremamente animadores.

Palavras-chave: Motores de passo, Acionamentos, Micropassos.

1. INTRODUÇÃO

A utilização de acionamentos com ceifamento de tensão, conhecidos como acionamentos *Chopper*, termo que iremos usar neste trabalho pois está incorporado ao jargão mecatrônico, e, mais recentemente, a possibilidade de atuação em regime de micropasso, fizeram com que os motores de passo passassem a ser utilizados numa ampla gama de aplicações em robótica e automação industrial, antes reservada aos servomotores CC de custo muito mais elevado.

Como é sabido, o motor de passo é um atuador do tipo digital. Na operação mais típica, a cada pulso recebido da unidade de controle, as correntes em suas fases são chaveadas e o rotor do motor avança, ou recua, um passo.

O ângulo do passo varia de motor a motor, mas, tipicamente, é de $1,8^\circ$. Este modo de operação confere ao motor de passo sua ampla versatilidade, tornando-o atraente para aplicações em malha aberta. Velocidade e posição são diretamente controladas pela frequência e número de pulsos enviados. Com a introdução das técnicas de micropasso, obtém-se resoluções de até $0,09^\circ$ o que resulta ótimas qualidades de posicionamento e uma operação suave livre de trepidações. Das muitas referências sobre motores de passo destacamos algumas (Kuo,1974), (Da Silva,1989) e (Parker Co,1997).

Um problema central na tecnologia de acionamento de motores de passo, consiste na necessidade de uma precisa regulação da corrente de fase e na capacidade de chaveá-la

em alta velocidade. Isto é, em princípio, complicado, pois como as fases são enrolamentos elétricos, representam cargas indutivas que não reagem linearmente a mudança de tensão.

A limitação da corrente de fase usando simplesmente um resistor em série, mostrou-se uma solução ruim para aplicações onde a corrente deve ultrapassar 1A. Assim, os acionamentos chopperizados, antes restritos aos motores de corrente contínua, passaram a ser utilizados, no contexto de motores de passo.

Uma dificuldade inerente aos acionamentos chopperizados convencionais, é a necessidade de monitoramento da corrente de fase. Este monitoramento é feito mediante a introdução de um resistor de baixíssima resistência ($0,2 \Omega$) em série com a fase, detectando-se a queda de tensão através dele. Esta tensão, proporcional à corrente, serve como informação para o corte da tensão principal.

Como o circuito chaveia correntes elevadas, existe o perigo do ruído eletromagnético gerado, contaminar a tensão de monitoramento, provocando chaveamentos em falso e fazendo com que o acionamento não opere de forma adequada.

Este problema tem sido contornado, nas placas comerciais, através de uma construção compacta e a introdução de alguns capacitores de filtragem.

O presente trabalho propõe uma alternativa em que o monitoramento da corrente é totalmente dispensado. Os primeiros testes foram realizados em janeiro de 2000, com um acionamento programado para operação em 1600 passos por volta. Os resultados obtidos, comparados com testes realizados anteriormente utilizando a tecnologia mais convencional, foram extremamente animadores.

Citamos alguns trabalhos anteriores (Ferreira et al.,1993), (Segenreich et al.,1997) e (Lima, 1997) desenvolvidos com este objetivo e que proporcionam uma leitura mais completa sobre o tema.

2. A TECNOLOGIA CONVENCIONAL DE ACIONAMENTO CHOPPERIZADO

O problema fundamental no acionamento de motores de passo, capazes de produzir torques em níveis desejados em automação e robótica (10 a 500 N.cm), é a necessidade de ligar e desligar correntes de fase em fração de segundos. No caso de utilização do acionamento em regime de micropasso, o problema se complica ainda mais, pois devemos produzir níveis intermediários de corrente em intervalos de tempo ainda menores.

A dificuldade neste controle decorre do fato de os enrolamentos do motor representarem cargas indutivas, o que faz com que a corrente seja uma função não linear da tensão, seguindo-a com atraso. As expressões (1) e (2) representam de forma aproximada o comportamento da corrente.

$$i = \frac{V}{R} (1 - e^{-\frac{t}{\tau}}) \quad (1)$$

$$\tau = \frac{L}{R} \quad (2)$$

onde:

- i - corrente
- V- tensão aplicada
- R- resistência ôhmica total
- τ - constante de tempo
- L - indutância da fase do motor

Existem duas técnicas básicas para energizar as fases. Em ambas, aplica-se uma tensão várias vezes superior à tensão nominal do motor, com o objetivo de estabelecer a corrente desejada no menor intervalo de tempo possível.

a- Limitação de Corrente por Resistência-

Consiste na colocação de um resistor em série com a fase. Este resistor tem o papel de limitar a corrente a seu valor nominal, já que a resistência ôhmica do enrolamento da fase é, em geral, baixa.

Esta técnica, a mais antiga em motores de passo, só tem sido aplicada com sucesso, para motores com pouca exigência de torque (impressoras, drivers, etc.). Nas aplicações típicas em automação ou robótica, onde a corrente de fase é superior a 1A, o resistor acaba dissipando muita potência inviabilizando sua utilização.

b- Limitação da Corrente por Ceifamento da Tensão (*chopper*)-

Neste caso, aplica-se a tensão elevada diretamente sobre a fase do motor. Quando a corrente alcança o valor desejado, a tensão é desligada fazendo com que a corrente passe a decrescer. Quando a corrente alcança um valor mínimo pré-determinado, ou após um tempo pré-fixado, a tensão é religada e o processo se repete. O fato da tensão ser ligada e desligada continuamente é que origina o nome *chopper* em inglês.

O acompanhamento da intensidade da corrente na fase é feito através da colocação de um resistor de sensoramento em série com a fase. Este resistor, tipicamente na faixa de $0,2\Omega$, gera uma pequena queda de potencial V_{sense} , suficiente para avaliar a corrente de fase segundo a expressão

$$i = V_{sense} / R_{sense} \quad (3)$$

Como a resistência deste resistor é muito baixa, a potência nele dissipada é desprezível.

Um acionamento chopperizado típico é mostrado na figura 1.

Imaginemos que num primeiro instante o transistor T1 está ligado. A corrente passa a circular pela fase com intensidade crescente. V_{sense} aumenta de forma proporcional a corrente até alcançar o valor máximo $V_{sense} = V_{ref}$. Neste ponto, o comparador A1 inverte o sinal, ressetando o flip flop FF1 e desligando o transistor. O transistor será religado na passagem da próxima frente de onda e o processo se repete. O nível médio da corrente desejada é determinado pelo de V_{ref}

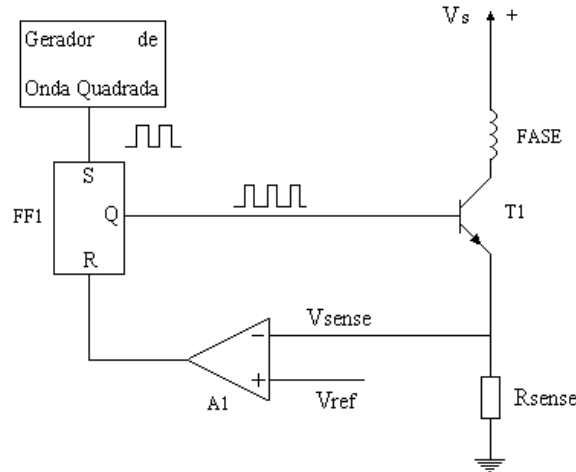


Figura 1 – Representação esquemática de um chopper convencional.

O circuito real é mais complexo pois ao invés de um único transistor, utilizam-se quatro, formando uma ponte em H, o que permite a inversão da corrente, mas o princípio de chopperização é o mesmo.

O chopper clássico aqui ilustrado, foi utilizado, originalmente, para o controle de servomotores de corrente contínua. A voltagem V_{sense} era controlada à partir de um sinal de erro representando a diferença entre a velocidade angular programada e a velocidade angular real medida por um sensor de velocidade, numa típica configuração de malha fechada.

Há cerca de vinte anos, este tipo de circuito limitador de corrente passou a ser usado com motores de passo para superar os inconvenientes da limitação de corrente via resistor em série. O nível da corrente é pré programado via V_{ref} num valor próximo ao valor nominal. Mais recentemente, com a introdução de valores de passo fracionários ou micropassos, os valores intermediários de corrente passaram a ser obtidos via uma tabela de valores para V_{ref} que podem ser gerados em tempo real por software (Lima,1997) ou armazenados em EPROMS (Paniagua,2000).

Apesar de representar um grande avanço, o chopper descrito apresenta alguns inconvenientes no acionamento de motores de passo. Os dois principais inconvenientes são os seguintes:

- a - O resistor R_{sense} deve ser de indutância nula para não interferir no processo.
- b - O chaveamento das correntes nas fases gera um ruído eletromagnético que pode contaminar V_{sense} . Consequentemente, a linha de realimentação de V_{sense} deve ser curta e bem isolada, pois o ruído eletromagnético gerado pelo chaveamento das fases, pode gerar valores falsos de V_{sense} na entrada do comparador e consequentemente desligar a tensão prematuramente.

Tendo em vista estes problemas inerentes ao chopper clássico, e, mais, o fato de que no acionamento de motores de passo, a programação dos valores de V_{ref} é definida à priori, independentemente da real operação do motor, os autores passaram a examinar possibilidade de uma alteração conceitual no arranjo do chopper eliminando, por completo, R_{sense} e, consequentemente, a necessidade de V_{sense} .

3. UM NOVO ACIONAMENTO CHOPPERIZADO PARA MOTORES DE PASSO SEM A UTILIZAÇÃO DE R_{sense}

Nesta seção descreve-se o novo acionamento em que foi eliminada a linha de realimentação de V_{sense} . A figura abaixo apresenta de forma sucinta o novo circuito.

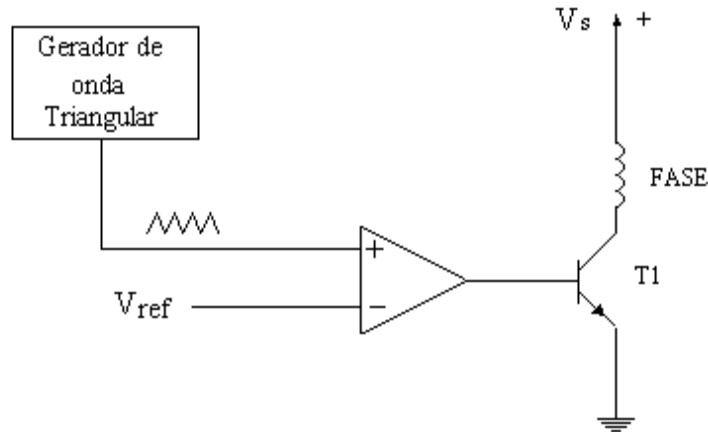


Figura 2 – Representação esquemática do chopper modificado sem R_{sense}

Segundo este arranjo, a largura de pulso durante a qual o transistor conduz é determinada diretamente pelo nível de V_{ref} . A figura abaixo mostra, com maior detalhe, a geração e o controle da largura do pulso.

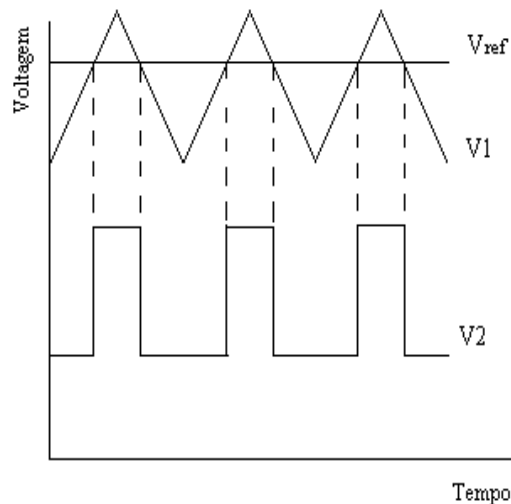


Figura 3 – Detalhe da geração dos pulsos com largura modulada por V_{ref} .

Para se testar o novo conceito foi montado um acionamento em micropasso com a resolução de 1600 passos por volta, já que dispunha-se de resultados para esta resolução

obtidos num acionamento clássico (com R_{sense}) permitindo uma comparação. O diagrama do circuito completo é mostrado na figura 4.

O sinal de sentido e os pulsos são enviados para um contador circular. A totalização da contagem, que pode ser para frente ou para traz dependendo do sinal de sentido, gera os valores dos endereços de entrada nas duas eproms. Os valores digitais de corrente para cada fase são transferidos das eproms para dois conversores D/A que geram as duas voltagens de referência

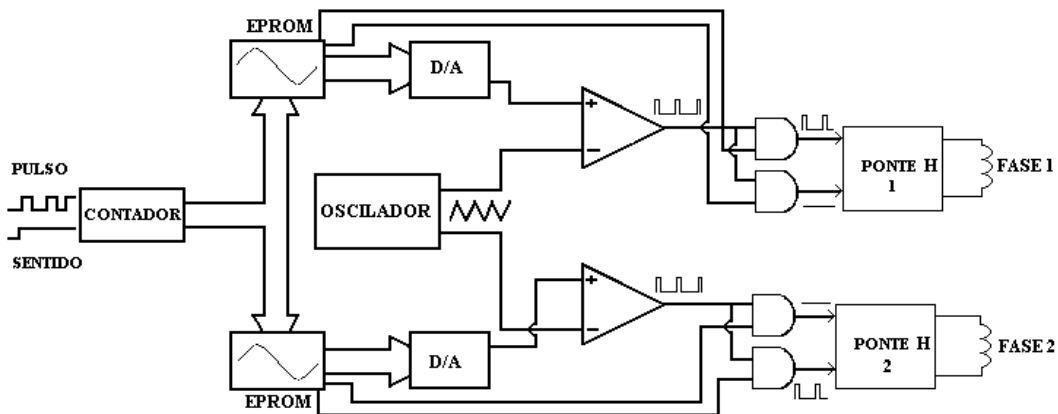


Figura 4 – Diagrama completo do chopper modificado proposto.

4. RESULTADOS EXPERIMENTAIS

Os testes realizados para avaliar o desempenho do novo acionamento visaram ao levantamento da curva torque limite versus velocidade do motor.

O arranjo experimental consistiu em pendurar uma massa, via corda de nylon, a uma polia acoplada ao eixo do motor, gerando um torque resistente conforme mostrado na figura abaixo.

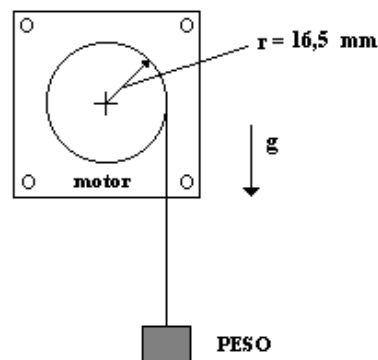


Figura 5 – Esquema de aplicação do peso para geração do torque resistente.

Para cada valor de massa, a velocidade angular era incrementada em cada corrida até detectar-se uma perda de passo (a posição angular do eixo ao final da corrida não mais coincidia com a posição inicial). Este valor de velocidade, representava o limite para aquela massa (torque).

Para se obter corridas mais confiáveis, o sistema de controle foi programado para executar quatro revoluções em cada sentido, repetindo-se o processo de forma contínua, por cinco vezes, num total de 40 revoluções completas.

Apresentamos, abaixo, os resultados para uma resolução de 1600 passos/volta para o acionamento convencional e o acionamento proposto.

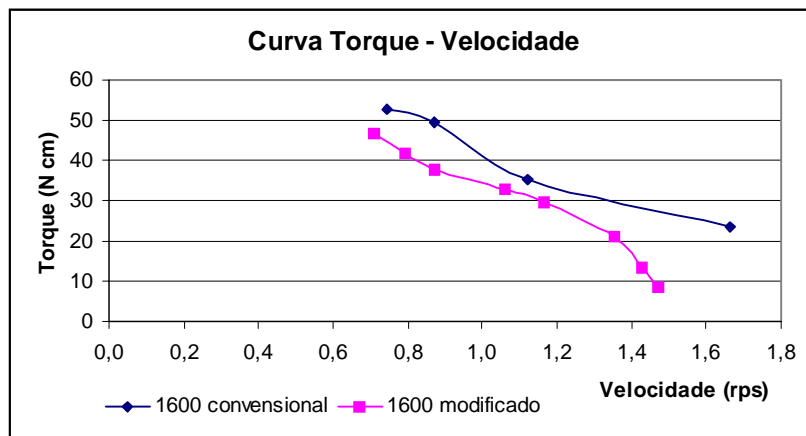


Figura 6 – Gráfico das curvas torque x velocidade angular para os dois acionamentos.

5. CONCLUSÕES

A análise dos resultados até agora obtidos, indicam um funcionamento suave com níveis de torque compatíveis com a tecnologia convencional. Os níveis de torque um pouco mais baixos obtidos com a nova placa, devem-se ao fato de termos limitado a largura do pulso a valores bem conservativos para não correr risco de queima de ponte o que comprometeria a obtenção dos primeiros resultados a tempo. Verificação dos sinais na entrada das pontes em H, feitas com auxílio de osciloscópio, mostram pulsos limpos sem contaminação de ruído. Cabe ressaltar que a placa resultou em menos componentes e com uma construção bem mais simples.

Testes adicionais estão sendo programados para avaliar todos os aspectos desta nova tecnologia e para otimizar os parâmetros objetivando uma melhoria adicional nas curvas torque x velocidade.

6. REFERÊNCIAS

- De Silva, C.W.,1989, “Control Sensors and Actuators”, Prentice Hall, New York.
- Ferreira, C.G. e Segenreich,S. A . ,1993, “Motor de Passo- Técnica de Acionamento Anti-ressonância”, Proceedings of the 12th Brazilian Congress of Mechanical Engineering, Rio de Janeiro, Brazil, Vol 1, pp-417-420.
- Kuo, B.K., 1974, “Theory and Applications of Step Motors”, West Publishing Co.
- Lima, R.V., 1997, “Análise e Desenvolvimento Experimental de uma Placa de Acionamento em Micropasso”,Tese de Mestrado, Depto. de Eng. Mecânica, PUC-Rio, Rio de Janeiro, Brasil.
- Parker Compumotor Division, 1997, ”Step Motors & Sevo Motors – Systems and Control Catalog, USA.
- Segenreich, S.A ., Lima,R.V. and Barros, L.A .,1997, ” Experimental Development of a Microstepping Driver for Increased Resolution and Low Vibration”, Proceedings of DINAME 97, Angara dos Reis , Brazil. Ed. by ABCM.
- Paniagua,J.V.,2000,” Acionamento de Motores de Passo em Micropasso : Duas Tecnologias”, Versão preliminar de Tese de Mestrado a ser Defendida em março de 2000, Depto. de Eng. Mecânica, PUC-Rio, Rio de Janeiro, Brasil.

UM SISTEMA DE VISÃO PARA RECONHECIMENTO DE PEÇAS EM UM TABULEIRO DE XADREZ

Sérgio Faustino Ribeiro
Juracy Emanuel M. da França
Marcelo Alves de Barros
José Homero Feitosa Cavalcanti
Universidade Federal da Paraíba – CCT/COPIN/NEUROLAB-CT/DTM
Av. Aprígio Veloso,882 – Campus – 58.109-970 Campina Grande, PB
E-mail: homero@dsc.ufpb.br

Resumo

Neste trabalho apresenta-se o projeto e o desenvolvimento de um sistema de visão para detectar as peças em um tabuleiro de xadrez. A posição das peças (presença, ou ausência de peças em cada casa do tabuleiro) é transformada numa matriz simbólica que será utilizada por um sistema de controle robótico. O sistema de visão utiliza como ferramenta Redes Neurais Artificiais.

Palavras-chave: Visão Computacional, Inteligência Artificial, Redes Neurais Artificiais.

1. INTRODUÇÃO

O NEUROLAB (Laboratório de Redes Neurais Artificiais) está desenvolvendo um sistema robótico inteligente para o movimento de peças sobre um tabuleiro de xadrez, denominado AutoXad (Ferreira et ali, 1999). Na figura 1 apresenta-se a fotografia do AutoXad. O AutoXad é composto de um braço robótico, uma câmara digital QuickCam P/B (QuickCam User Guide, 1995) (ver esfera apontada por uma seta), um tabuleiro de xadrez, e um microcomputador PENTIUM utilizado para executar as tarefas de aquisição dos dados da câmara e de controle do sistema utilizando um Sistema Inteligente (Ferreira et ali, 1999).

O sistema de visão utiliza um software, denominado NEUROMORFO, que realiza operações básicas de processamento de imagens tais como: cálculo de histograma, equalização, filtragem, binarização, extração de contorno (morfologia), etc (Harilack and Shapiro, 1992) (Pandya, 1995). Tais operações são realizadas sobre imagens armazenadas em arquivos do tipo BMP de 8 bits. Além disso, o NEUROMORFO é capaz de detectar as peças no tabuleiro de xadrez, utilizando redes neurais artificiais, e transferir esta informação (presença, ou ausência de peças em cada casa do tabuleiro), na forma de uma matriz simbólica, para o sistema de controle do robô.

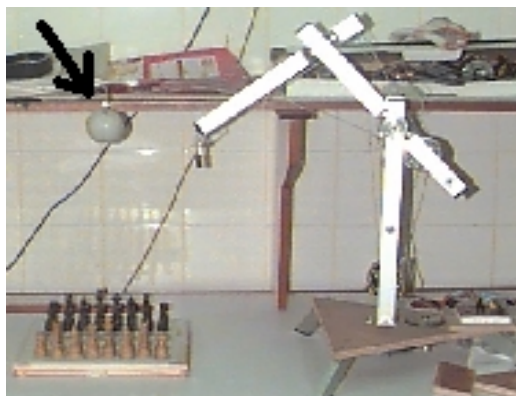


Figura 1. AutoXad.

2. DESCRIÇÃO DO NEUROMORFO

A QuickCam possui foco de 45,5 cm para o infinito e está a uma altura de 45,7 cm em relação à superfície do tabuleiro. Cada casa do tabuleiro possui uma dimensão de 2,7cm x 2,7 cm, de tal forma que uma casa na imagem pode ser enquadrada por uma moldura de 27 x 27 pixels, havendo uma correspondência, aproximada, de 1 pixel na imagem capturada pela câmara para 1 mm na superfície do tabuleiro. Na figura 2 apresenta-se a tela principal do NEUROMORFO com uma imagem do tabuleiro de xadrez (formato BMP) obtida pela QuickCam.

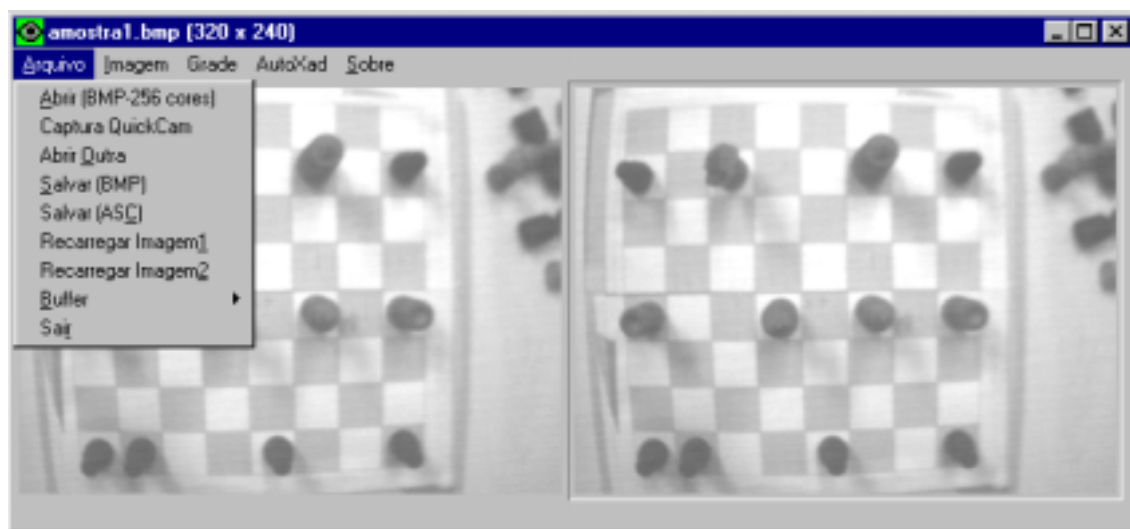


Figura 2. Tela principal do NEUROMORFO

O laboratório é iluminado por três lâmpadas fluorescentes. A disposição do tabuleiro em relação às lâmpadas gera uma variação da luminosidade ao longo do mesmo tal que, as casas que estão próximas à parede no fundo da imagem, na figura 3, apresentam níveis de cinza menores que as que estão na frente. A equalização da imagem é feita através do espalhamento dos níveis de cinza, multiplicando-se cada nível por um fator obtido pela divisão entre 255 e o maior nível encontrado na imagem. Na figura 3 pode-se observar a sombra das peças ao longo do tabuleiro.

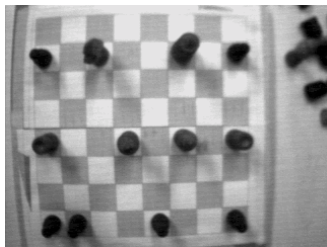


Figura 3. Imagem equalizada do tabuleiro de xadrez.

A imagem do tabuleiro está contida num quadrado de 216mm de lado, e a imagem da QuickCam tem a dimensão 320 x 240 pixel, isto é, a imagem contém informações redundantes. Necessitou-se isolar o tabuleiro do restante da imagem. Para isto, desenhou-se (virtualmente no computador) uma grade com as dimensões aproximadas do tabuleiro (8 x 8 casas de 27 x 27 pixels cada) e ajustada manualmente sobre o mesmo. A grade possibilita ao NEUROMORFO reconhecer as coordenadas das casas do tabuleiro de xadrez. Na figura 4 é apresentada a imagem do tabuleiro de xadrez juntamente com a grade. Observe-se que, devido à distorção na imagem gerada pela QuickCam, a grade não fica perfeitamente ajustada sobre o tabuleiro (em particular nas bordas). Observe-se também que a região central das casas do tabuleiro é a que apresenta a menor distorção, isto é, possui as informações mais relevantes.

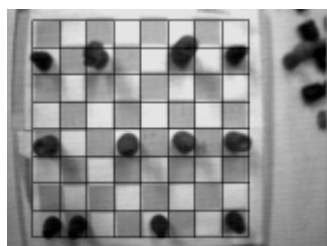


Figura 4. O tabuleiro de xadrez com a grade

Na figura 5 apresenta-se a imagem da ampliação ("zoom" de 300%) sobre a imagem de uma peça do tabuleiro localizada na posição 4D (quarta coluna a partir da esquerda e quarta linha a partir da base do tabuleiro de xadrez da figura 3). Convencionou-se como região central da casa do tabuleiro uma área de 9 x 9 pixels (cada casa possui uma dimensão de 27 x 27) deslocada em 9 pixels a partir das coordenadas da casa (ver quadrado central na figura 5).

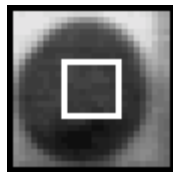


Figura 5. Ampliação da imagem de uma casa do tabuleiro

Na figura 6 apresenta-se a representação visual do conjunto de amostras extraídas do tabuleiro da figura 4, sendo que a ordem na qual a extração ocorre é da esquerda para a direita, e de cima para baixo.

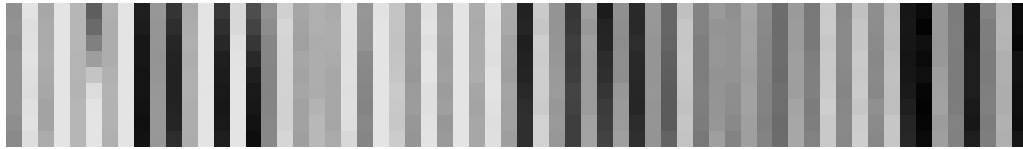


Figura 6. Amostras extraídas do tabuleiro

3. ANÁLISE DA IMAGEM USANDO REDES NEURAIS ARTIFICIAIS

A análise da imagem da casa do tabuleiro feita na região central com 9 x 9 pixel (81 pixels) do quadrado da figura 5 foi feita utilizando redes neurais artificiais. Devido às características da rede neural utilizada na análise da imagem, decidiu-se reduzir a dimensão dos dados (81 pixels) a serem analisados. Calculou-se a média aritmética de cada linha da amostra de 9 X 9 pixel e obteve-se um vetor com nove valores. Este vetor foi aplicado à entrada da rede neural.

Na figura 7 apresenta-se o conteúdo do vetor (valor numérico entre 0 e 255) das casas representadas por 8TD, 8CD e 8BD. Essas casas não contêm peça e portanto são claras, apresentando níveis altos de intensidade, principalmente por estarem numa das zonas mais iluminadas do tabuleiro.

138	142	146	147	147	146	143	146	144
222	226	225	220	224	226	225	224	227
172	164	163	168	169	172	170	171	167
Sem Peça								

Figura 7. Valores dos pixels de três casas sem peças do tabuleiro

Na figura 8 mostra-se o conteúdo correspondente às três casas representadas por 7TD, 7CD e 7BD. As casas 7TD e 7BD estão ocupadas por uma peça, os níveis correspondentes à peça possuem valores muito baixos. A casa 7CD não é ocupada por uma peça.

20	18	19	21	21	24	24	29	30
152	153	153	152	150	149	150	150	152
46	38	36	34	34	36	39	46	52
Com Peça								

Figura 8. Valores dos pixels de três casas do tabuleiro

Utilizou-se uma rede neural multicamada (RNMC) (Hummelhart, 1986) com 3 camadas (figura 9). A RNMC possui 9 neurônios na camada de entrada, 8 neurônios na camada escondida, e 1 neurônio na camada de saída. Os 9 neurônios da camada de entrada possuem função de ativação linear, servindo apenas para transformar os valores da entrada em partes por unidade (peso igual a 255). Os demais neurônios da RNMC possuem função de ativação do tipo sigmóide. O treinamento dos parâmetros da rede (pesos, β , θ , σ) foi realizado usando o algoritmo da propagação retroativa do erro (APR). Tal função de ativação sigmóide é representada pela equação 1.

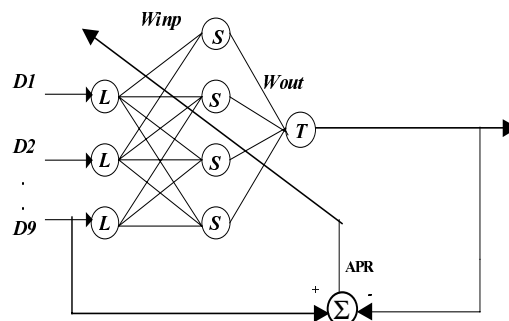


Figura 9. Rede Neural Artificial Multicamada.

$$S(X_i, \Theta, \beta, \delta) = \frac{\Theta}{[1 + \exp(-\beta(\sum X_i * Win_i + \delta))]} \quad (1)$$

Atribuíram-se os valores 1 (um) quando há uma peça sobre a casa e 0 quando não há peça (ver Figura 10). O conjunto de valores utilizados no treinamento da RNMC foram escolhidos a partir das casas centrais do tabuleiro. A rede aprenderá através do conjunto de treinamento apresentado à mesma (conjunto de vetores representando padrões). A escolha de um conjunto de treinamento que represente de forma adequada o problema visado, e tenha um tamanho adequado, é de fundamental importância para o bom desempenho da rede. Considerando tal importância, foi utilizado um conjunto de quatro imagens básicas para realizar o treinamento da rede. Escolheram-se 4 imagens de casas como conjunto básico de treinamento. As casas foram: uma casa branca sem peça, uma casa branca com peça, uma casa preta sem peça, e uma casa preta com peça. A RNMC foi treinada durante 1000 iterações utilizando as 4 amostras básicas. A seguir, foi feito o teste com a imagem do tabuleiro de xadrez apresentado na figura 4. Na figura 10 apresenta-se o resultado da classificação das casas do tabuleiro.



Figura 10. Classificação do tabuleiro efetuada pela RNMC

4. CONCLUSÃO

A atual implementação do NEUROMORFO foi capaz de realizar a detecção das peças no tabuleiro com um grau de acerto em torno de 98%. Os erros se devem, possivelmente, às variações de iluminação no ambiente. Como se deseja um sistema robusto, está sendo analisado o uso do Sistema de Previsão Inteligente (SPI) (Melo et al, 1999) para atribuir tal robustez ao sistema. Esse novo sistema de visão poderá ser empregado em: orientação de robôs móveis, futebol de robôs, análise de imagens clínicas, etc. Futuramente, será utilizada uma versão adaptada do SPI para o reconhecimento do tipo das peças (pião, torre, cavalo, rei

e dama) sobre as casas do tabuleiro de xadrez. O NEUROMORFO foi escrito na linguagem C++ utilizando o C++ Builder da Borland.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- Ferreira, C.N.M.A., Ferreira, J.R.S., Cavalcanti, J.H.F., Alsina, P.J., e Franca, J.E., 1999, “Posicionamento Inteligente de um Braço Robótico” COBEM’99 - Congresso Brasileiro de Engenharia Mecânica, Águas de Lindóia, SP.
- Harilack, R.M., e Shapiro, L.G., 1992, “Computer and Robot Vision”, Addison-Wesley Publishing Company.
- Humelhart, D.E. e Hinton, G.E. & Williams, R.J. 1986, “Learning Internal Representations by Error Propagation, Parallel Distributed Processing: Explorations in the Microstructures of Cognition”. Vol 1, D.E. Humelhart & J.L. McClelland, Cambridge, MA: MIT Press, pp. 318-362.
- Melo, H., Cavalcanti, J.H.F., Silva, J.F., e Silva, V.P.R., 1999, “Sistema de Previsão Inteligente”, CBRN - IV Congresso Brasileiro de Redes Neurais, São José dos Campos – SP.
- Pandya, A.S., e Macy, R.B., 1995, “Pattern Recognition with Neural Networks in C++”, CRC Press, e IEEE Press.
- QuickCam User Guide, 1995, Connectix Corporation, November.

CONCEPÇÃO CINEMÁTICA DE UM MANIPULADOR PARA VOLUMES DE TRABALHO RESTRITOS

Raul Guenther

Henrique Simas

Edson Roberto de Pieri

Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica, Laboratório de Robótica, Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil. E-mail: simas@lcmi.ufsc.br; guenther@emc.ufsc.br

Resumo

O presente artigo apresenta a concepção cinemática de um manipulador para a realização de tarefas dentro de um volume de trabalho restrito. Especificamente ele será utilizado na recuperação de regiões erodidas por cavitação em turbinas hidráulicas de grande porte. A estrutura cinemática é criada a partir da caracterização do problema, da proposição de alternativas e da análise das mesmas. Para a análise utiliza-se uma combinação de simulações numéricas com simulações em maquetes.

Palavras chaves : Robótica, cinemática de manipuladores.

1. INTRODUÇÃO

Unidades hidráulicas usadas para geração de energia elétrica são frequentemente máquinas de grande porte. Seus rotores são construídos em aço de elevada resistência, podendo apresentar mais de 9 m de diâmetro (turbinas Francis) e pesar até 200 t.

O perfil dos rotores é projetado para apresentar grande eficiência hidráulica e condições de escoamento de forma a evitar o desenvolvimento de regiões de baixa pressão onde ocorra cavitação, fenômeno que pode provocar severa erosão nas pás do rotor reduzindo sua eficiência hidráulica e comprometendo a integridade estrutural do conjunto.

Apesar dos grandes avanços na mecânica de fluídos ainda não se consegue evitar totalmente a cavitação nas turbinas. Assim, mesmo as turbinas modernas apresentam cavitação, fenômeno mais agudo nas turbinas mais antigas.

Sendo tecnicamente complexa e economicamente inviável a substituição de rotores, a erosão por cavitação precisa ser mantida sob controle. Isto é feito através do seu monitoramento e da deposição “in loco” de ligas de aço inoxidável especiais nas regiões erodidas, com a finalidade de recompor o perfil original sempre que necessário. Essa deposição é feita mediante soldagem manual.

As regiões erodidas estão dentro do canal de condução do rotor da turbina e o operador realiza a soldagem num ambiente restrito e fortemente insalubre. Além disso, as erosões ocorrem em posições bastante desfavoráveis para a soldagem, tanto do ponto de vista do soldador como para a adesão do metal fundido.

O projeto ROBOTURB tem por objetivo desenvolver um sistema automatizado para a recuperação das regiões erodidas nos rotores de turbinas hidráulicas. Pretende-se, assim, eliminar o trabalho insalubre, obter uma melhor qualidade na recuperação e com isso

prolongar sensivelmente os períodos entre as manutenções, e reduzir o custo das recuperações.

O projeto integra pesquisas na áreas de metrologia, soldagem, acionamentos e automação e está sendo desenvolvido através de uma parceria entre a Universidade Federal de Santa Catarina (UFSC) e a Companhia Paranaense de Energia Elétrica (COPEL). Financiado pelo PADCT, o projeto envolve os departamentos de engenharia mecânica (EMC) e de Automação e Sistemas (DAS) da UFSC, e o Laboratório Central de Pesquisa e Desenvolvimento (LACTEC) da COPEL.

Utilizando o sistema automatizado, além de eliminar a exposição de pessoas ao ambiente insalubre, pretende-se alcançar a melhoria e a uniformização na qualidade da soldagem e, portanto, do acabamento superficial, e reduzir a quantidade de material depositada e o tempo gasto nesta operação.

O elemento central do sistema é um manipulador integrado ao processo de soldagem, com capacidade para operar no pequeno espaço restrito do canal do rotor. Esse manipulador desenvolverá também a operação de sensoramento através da qual será feita a avaliação da forma da região erodida.

Neste trabalho apresenta-se a concepção cinemática para esse manipulador a partir da caracterização do problema realizada na seção 2 e da metodologia descrita na seção 3. As alternativas para a estrutura cinemática estudadas estão apontadas na seção 4. Os resultados das simulações estão apresentados na seção 5 e a conclusão está na seção 6.

2. CARACTERIZAÇÃO DO PROBLEMA CINEMÁTICO

De acordo com dados levantados pelo LACTEC ao longo de diversas recuperações de turbinas, as cavitações ocorrem no canal de condução do rotor formado pelas superfícies de ataque e de sucção e limitado pela coroa (veja Fig.1).

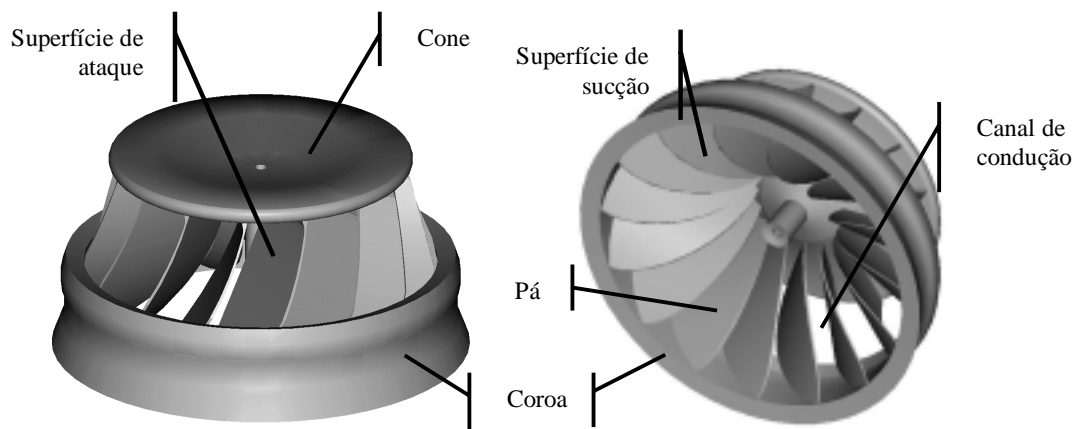


Figura 1. Rotor de uma turbina hidráulica

Adicionalmente existe também uma região com cavitações no tubo de sucção da turbina (Fig.2).

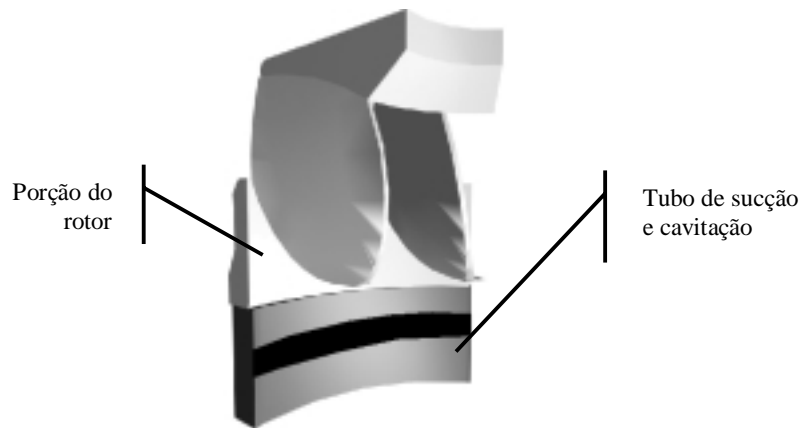


Figura 2. Vista do rotor com o tubo de sucção de uma turbina hidráulica

No canal de condução as cavitações ocorrem na superfície de sucção e na ligação desta superfície com a coroa, principalmente nas áreas mostradas na Fig.3.

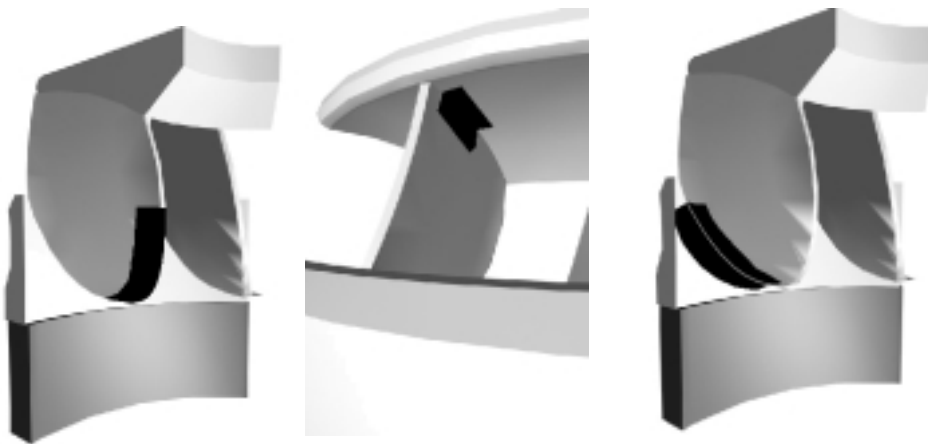


Figura 3. Principais áreas de cavitação no canal de condução

A movimentação da tocha de soldagem ao longo da cavitação no tubo de sucção (Fig.2) é simples porque o movimento é praticamente livre da possibilidade de colisões. Nas cavitações da superfície de sucção o movimento da tocha é dentro do canal de condução e, portanto, dentro de um espaço restrito e passível de colisões.

Para executar a recuperação das cavitações o manipulador funcionará dentro do canal de condução, que pode ser caracterizado como um volume de trabalho curvo e afunilado.

Os dados fornecidos pela LACTEC permitiram verificar que as regiões cavitadas tem uma forma caracterizada pela preponderância de uma dimensão, cujo comprimento chega a 0.70 m e cuja largura é de aproximadamente 0.30 m.

O processo de soldagem para a recuperação está sendo desenvolvido pelo Labsolda-UFSC e prevê a utilização de uma tocha com duplo arame. Por isso o processo exige que o manipulador tenha seis graus de liberdade para que a tocha possa atingir todas as posições e orientações dentro do espaço de trabalho.

O sensor laser que fará o mapeamento da superfície cavitada impõem a mesma exigência cinemática.

Fica caracterizado, portanto, que o robô deve ser projetado para realizar movimentos com seis graus de liberdade ao longo de cavitações longas em um espaço de trabalho restrito.

3. METODOLOGIA DE ANÁLISE CINEMÁTICA

O projeto cinemático do robô foi realizado a partir do exame de diversas propostas de estruturas cinemáticas concebidas para realizar movimentos com seis graus de liberdade em um espaço de trabalho curvo e afunilado.

A análise dessas alternativas incluiu o estudo dos movimentos em cada uma das juntas, a avaliação da manipulabilidade (Sciavicco e Siciliano, 1996), a verificação das possibilidades de colisão do manipulador com as superfícies do canal de condução, e as possibilidades de fixação do robô.

Esta análise foi realizada combinando simulações numéricas em computador com simulações em maquetes em tamanho natural (1:1) das alternativas de estruturas dos robôs e do canal de condução do rotor de uma turbina característica.

As simulações nas maquetes possibilitaram o estudo das alternativas de fixação de cada proposta de robô, e ensejaram a verificação da possibilidade de colisões. Além disso, auxiliaram na determinação das posições que as juntas do robô assumem durante o movimento ao longo das regiões cavitadas.

As simulações em computador foram preponderantes no estudo dos movimentos nas juntas.

3.1. A maquete do canal de condução

A construção desta maquete foi realizada a partir da escolha de uma turbina com características representativas de uma série de instalações. Esta escolha recaiu na usina de Foz de Areia, no sul do estado do Paraná, de propriedade da COPEL.

Feita a escolha, o LACTEC levantou o perfil do canal de condução e com esses dados construiu-se a maquete do canal em fibra de vidro, apresentada na Fig.4.

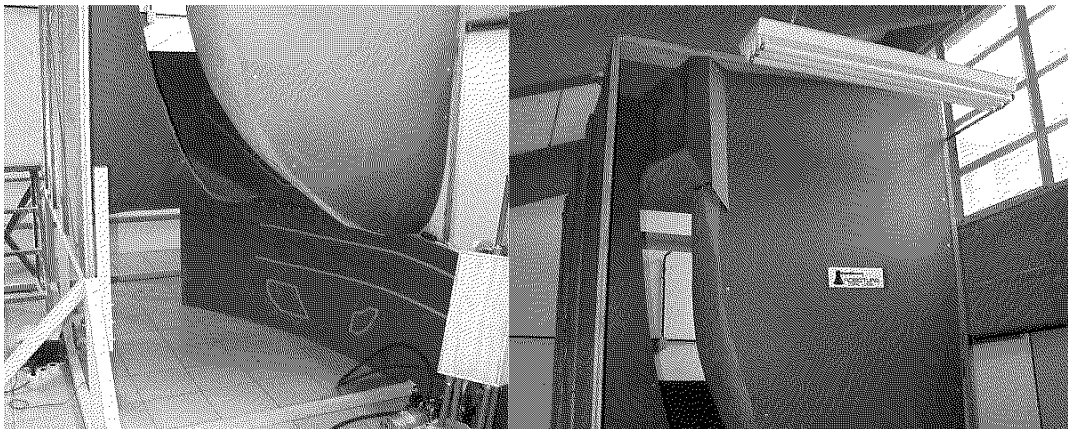


Figura 4. Maquete do canal de condução

3.2. Simulações numéricas

As simulações numéricas em computador foram desenvolvidas através da utilização combinada dos programas Matlab[®] 5.1, Mathematica[®] 3.01, 3D Studio Max[®], Autocad 13[®].

Com o programa Matlab[®] 5.1 realizou-se a simulação da cinemática direta, utilizando a metodologia de Denavit-Hartenberg (Fu et al, 1987) através da qual foram analisados os movimentos nas juntas a partir das suas posições iniciais e finais obtidas em experimentações nas maquetes.

O programa Mathematica[®] 3.01 foi empregado no desenvolvimento dos modelos matemáticos dos manipuladores.

O desenho dos manipuladores foi realizado com auxílio do programa Autocad 13[®] e o programa 3D Studio Max[®] possibilitou a visualização dos movimentos em três dimensões.

4. ALTERNATIVAS PARA A ESTRUTURA CINEMÁTICA

4.1. Estrutura articulada

A estrutura articulada estudada é formada por seis juntas de rotação dispostas de na forma apresentada na Fig.5. São três juntas de rotação interligando os três primeiros elos e um punho esférico(Craig, 1986). A Fig.5 mostra também um protótipo da tocha de soldagem e do sensor acoplados à extremidade do manipulador.

Como pode ser observado da figura, a disposição dos três primeiros elos foi projetada de forma a propiciar a realização de movimentos em uma região estreita. Os elos 2 e 3 tem a função de desenvolver o movimento ao longo do comprimento da cavitação.

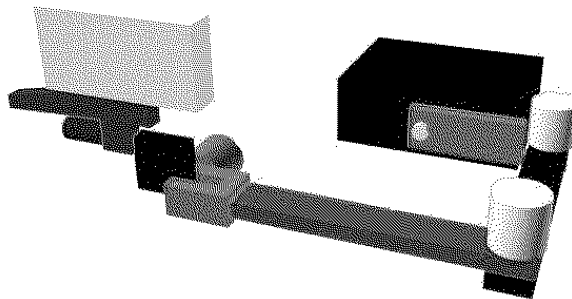


Figura 5 – Estrutura articulada

4.2. Estrutura com junta telescópica

A estrutura com junta telescópica estudada é formada por duas juntas de rotação seguidas de uma junta telescópica cujo corpo constitui o terceiro elo (Fig.6). À extremidade deste terceiro elo é acoplado um punho esférico.

Também neste caso a concepção da forma dos três primeiros elos buscou contemplar a necessidade de realizar movimentos em uma região restrita. Neste caso a junta telescópica tem a função de efetuar o movimento ao longo do comprimento da cavitação.

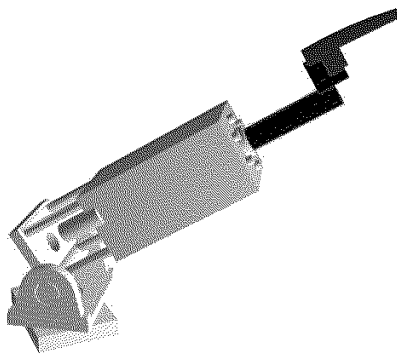


Figura 6 – Estrutura com junta telescópica

Cabe ressaltar que a junta telescópica para realizar movimentos no espaço de trabalho restrito imposto pela tarefa foi objeto de um projeto mecânico especialmente desenvolvido para aplicação no projeto Roboturb. Trata-se de um mecanismo composto por três conjuntos de fusos de esferas recirculantes que funcionam de forma acoplada. Devido à limitações dos componentes mecânicos disponíveis no mercado, essa junta telescópica resultou com dimensões que a fazem ocupar um grande volume dentro da região restrita do canal (veja Fig.8). Maiores detalhes podem ser encontrados em (Albertazzi, 1999).

4.3. Estrutura articulada com trilho

Neste caso a estrutura articulada é a mesma descrita acima, com seis juntas de rotação, acrescentando-se um movimento de translação ao longo de um trilho (Fig.7).

O trilho acrescenta um grau de liberdade, torna o manipulador redundante com sete graus de liberdade, e possibilita que o movimento ao longo do comprimento da cavitação seja realizado prioritariamente pela translação ao longo do trilho.

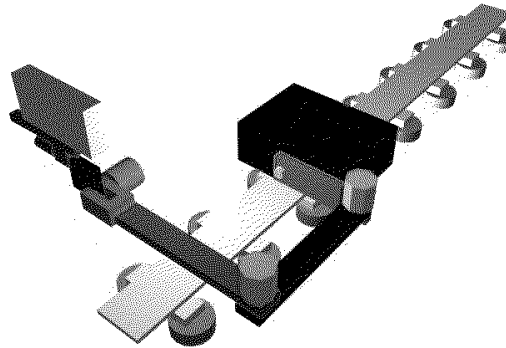


Figura 7 – Estrutura articulada com trilho

5. RESULTADOS DAS SIMULAÇÕES

Para cada uma das alternativas apresentadas na seção anterior foram construídas maquetes em madeira com movimentos semelhantes aos desejados para o manipulador.

Essas réplicas utilizadas na maquete do canal de condução permitiram a análise dos melhores pontos de fixação do manipulador, dos seus movimentos e das possibilidades de colisão.

Em todos os casos a fixação do manipulador ocorreu no tubo de sucção da turbina ou nas superfícies de sucção e de ataque do canal de condução do rotor. A escolha da localização para a fixação foi realizada de tal forma que o manipulador alcançasse toda a região a ser recuperada.

Para cada alternativa a maquete do manipulador foi fixada medindo-se as posições de suas juntas no início da operação de recuperação e no final. Esses dados foram então utilizados na simulação do movimento em computador.

Apresenta-se a seguir os resultados das simulações em maquete.

5.1. Estrutura articulada

A estrutura articulada exige uma escolha judiciosa dos locais de fixação, o que dificulta sua aplicação pelo fato dessa escolha cuidadosa nem sempre ser possível “in loco”.

De acordo com o esperado e verificado nas simulações na maquete, as simulações numéricas mostraram que a movimentação maior ocorre nos elos 2 e 3. Devido a isso eles necessitam ter grandes dimensões para poder alcançar toda a região erodida.

A grande dimensão desses elos amplia a possibilidade de ocorrerem colisões do “cotovelo” que eles formam com as superfícies do canal de condução da turbina.

5.2. Estrutura com junta telescópica

Essa estrutura também exige uma escolha criteriosa dos locais de fixação. O projeto mecânico da junta telescópica resultou em um mecanismo com dimensões externas que ocasionam colisões com as superfícies do canal de condução, conforme pode ser observado na Fig.8. Esse fato inviabiliza sua utilização.

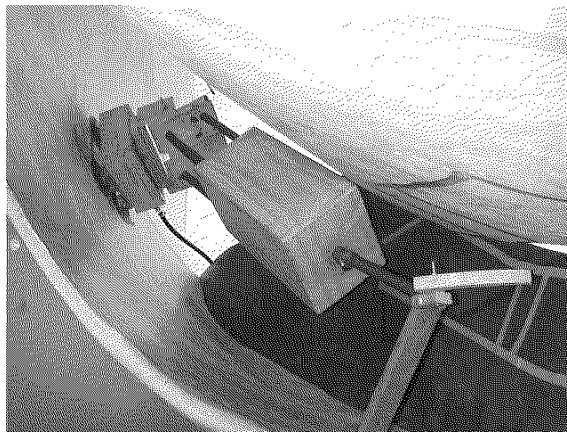


Figura 8 – Manipulador com junta telescópica no interior do canal de condução

5.3 – Estrutura articulada com trilho

A fixação do trilho pode ser feita na superfície de sucção e de ataque do canal de condução do rotor. A Fig.9 mostra o trilho fixado através de ventosas na superfície de ataque.

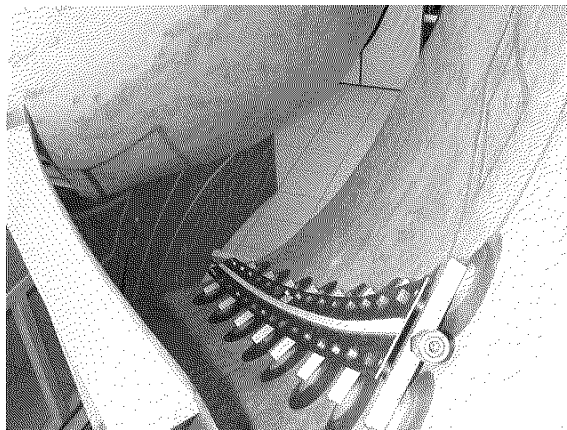


Figura 9 -Trilho fixado na superfície de ataque da turbina hidráulica

A experimentação mostrou que essa alternativa não exige maiores cuidados na escolha do local de fixação do trilho, facilitando seu emprego “in loco”.

A utilização do trilho para a realização do movimento ao longo do comprimento da região erodida adequa a estrutura cinemática à forma da erosão. Com isso, o movimento a ser realizado pelos elos é menor, fazendo com que eles possam ter menores dimensões. Isso faz com que a possibilidade de colisões seja sensivelmente reduzida.

Uma vantagem adicional é que com sua fixação à uma das superfícies do canal, o trilho (flexível) assume a forma curva do volume de trabalho, facilitando a movimentação da tocha ao longo do região cavitada.

Devido a esses fatores essa é a estrutura cinemática escolhida.

6. CONCLUSÃO

Este artigo descreve a concepção cinemática de um manipulador a ser empregado na recuperação de regiões erodidas pela cavitação em turbinas hidráulicas de grande porte. Apresenta-se também uma metodologia desenvolvida para proporcionar a análise cinemática e a verificação de colisões em um manipulador cujos movimentos ocorrem em um volume de trabalho restrito. Essa metodologia combina simulações numéricas com simulações em maquete. Essa combinação mostrou-se adequada, e possibilitou a definição da estrutura cinemática do manipulador.

7. REFERÊNCIAS

- Sciavicco, L., Siciliano, B., 1996, “Modeling and Control of Robot Manipulators”, McGraw-Hill, New York.
- Fu, K. S., Gonzalez, R.C. , Lee, C.S.G, 1987, “Robotics, Control, Sensing, Vision, and Intelligence”, McGraw-Hill, Singapore.
- Craig, J.J. 1986, “Introduction to Robotics Mechanics & Control”, Addison-Wesley Publishing Company
- Albertazzi, A., 1999, “Relatório Projeto Roboturb” . Florianópolis.

AGRADECIMENTOS

A realização desse trabalho contou com o apoio financeiro da FINEP e da COPEL e com a colaboração técnico-científica dos laboratórios de Metrologia e de Soldagem do departamento de engenharia mecânica da UFSC, e do Laboratório Central de Pesquisa e Desenvolvimento (LACTEC) da COPEL. Os autores agradecem todas as contribuições.

POSICIONAMENTO INTELIGENTE DE UM BRAÇO ROBÓTICO

Cristiany N. Moscoso do Amaral Ferreira

Everaldo Fernandes Monteiro

José Homero Feitosa Cavalcanti

Universidade Federal da Paraíba – CCT/COPIN/NEUROLAB-CT/DTM

Av. Aprígio Veloso,882 – Campus – 58.109-970 Campina Grande, PB

E-mail: homero@dsc.ufpb.br

Resumo

Neste trabalho descreve-se o projeto, a implementação, a montagem de um protótipo, e os resultados experimentais obtidos de um Sistema de Posicionamento Inteligente de um braço mecânico utilizado para mover as peças sobre um tabuleiro de xadrez. O Sistema é baseado em técnicas de Inteligência Artificial, utilizando Lógica Fuzzy cujas regras e axiomas foram obtidas a partir da observação (analogia) dos movimentos das mãos de um jogador (simetria) durante a manipulação das peças numa partida de xadrez. Serão apresentados alguns detalhes da construção do braço mecânico e os resultados experimentais obtidos do movimento de peças no tabuleiro de xadrez.

Palavras-chave: Sistemas Inteligentes, Robótica e Lógica Fuzzy.

1. INTRODUÇÃO

Atualmente os robôs possuem liberdade para realizar determinadas tarefas de acordo com o meio a que pertencem, como por exemplo, na exploração espacial, na manipulação de amostras (ampolas) em análises biomédica e outros. Esses robôs devem perceber o meio em que estão inseridos e modificar suas ações para realizar tarefas específicas, através de um comportamento inteligente. Um dos temas de grande interesse na robótica, é o posicionamento de braços robóticos durante a execução de uma tarefa de manipulação de objetos. Muitas tarefas exigem que os movimentos sejam executados dentro de um tempo e uma trajetória rigorosamente precisos. Porém, algumas vezes as tarefas a serem executadas não necessitam de tal precisão, mas por outro lado necessitam de um grau de adaptabilidade e flexibilidade próximo aos encontrados nos seres humanos. Portanto, o estudo de técnicas que empregam a adaptabilidade em sistemas robóticos se mostra interessante principalmente quando necessita-se que esses sistemas desempenhem tarefas nocivas aos seres humanos e que apresentem um comportamento variável. A manipulação de amostras em laboratórios de análise clínica se enquadra neste tipo de tarefa. Entretanto é possível apresentar um estudo de sistemas robóticos através de problemas semelhantes com fácil adaptação para esses ambientes atípicos e que pelas próprias características, não suportam a presença de terceiros.

No nosso laboratório foi desenvolvido um manipulador robótico com três graus de liberdade que está sendo utilizado para testes experimentais de sistemas de controle inteligente (Moscoso et ali, 1999).

Apresenta-se um sistema robótico projetado para o posicionamento de peças de xadrez em um tabuleiro, descrevendo-se todas as partes do protótipo do manipulador robótico com três graus de liberdade e uma garra (ver figura 1), bem como o Sistema Inteligente (SI) responsável pelo controle inteligente do braço. Apresenta-se também uma análise dos resultados experimentais obtidos.

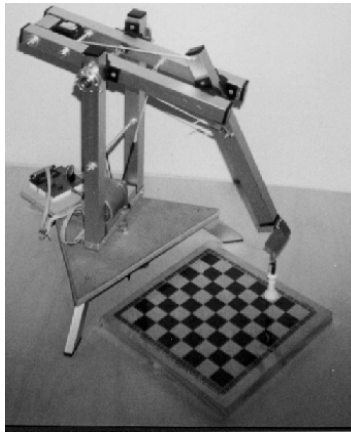


Figura 1. Fotografia do robô AUTOXAD

2. ARQUITETURA DO AUTOXAD

O AUTOXAD é composto de uma base, três braços (S1,S2 e S3) e uma garra (ver Craig, 1986), conforme ilustrado na figura 2. As juntas dos braços são acionadas por motores de passo, sendo que a junta 1, responsável pelo movimento do braço S1, gira em torno da normal ao Eixo Base (ver figura 3), executando movimentos para a direita e esquerda. A junta 2, responsável pelo movimento do braço S2, executando movimentos para cima e para baixo e a junta 3, responsável pelo movimento do braço S3, executando movimentos também para cima e para baixo. A localização dos motores (M1,M2 e M3) é ilustrada na figura 2. O motor M1 movimenta o braço S1 através de um conjunto de roldanas. O motor M2 preso a base do braço S1 movimenta o braço S2 com um sistema de transmissão por cabos, e o motor M3 preso ao braço S2 movimenta o braço S3 também com um sistema de transmissão por cabos. O braço S3 possui uma garra eletromagnética (eletroímã) em sua extremidade.

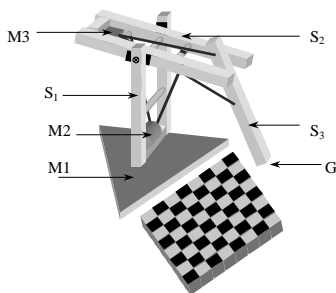


Figura 2. Desenho do AUTOXAD

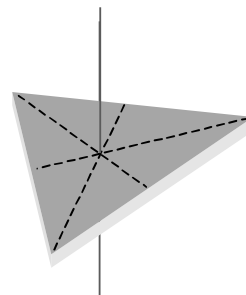


Figura 3. O Eixo Base

3. O AMBIENTE DE TRABALHO DO AUTOXAD

Convencionou-se que a frente do tabuleiro deve ser paralela ao lado frontal do triângulo equilátero da base (ver figura 4). Além disso, a reta perpendicular a esse lado deve passar pelo centro do tabuleiro dividindo-o em dois lados (lado da dama e lado do rei). Esses posicionamentos são necessários devido a análise de simetria que será feita para especificar os estados do sistema.

O tabuleiro é formado por oito colunas e oito linhas, ou seja sessenta e quatro casas. As colunas do tabuleiro foram nomeadas como T_D , C_D , B_D , D , R , B_R , C_R e T_R , onde T representa a coluna da torre, C representa a coluna do cavalo, B representa a coluna do bispo, D representa a coluna da dama e R representa a coluna do rei. O subscritor $_D$ e $_R$ indica a representação dos lados do tabuleiro, ou seja, lado da dama e lado do rei. As linhas foram enumeradas de um a oito. Cada casa é identificada pelo número da linha e da coluna, por exemplo, $3C_D$, $5R$, $8T_R$.

Analisando-se o tabuleiro pode-se chegar a algumas conclusões, sobre, a simetria entre os dois lados. Por exemplo, se a garra do AUTOXAD estiver posicionada no ponto X, marcado na figura 5 (no encontro das linhas tracejadas), o movimento para a coluna D é simétrico ao movimento para a coluna R, pois o manipulador utilizará apenas o motor M1 (da base) para deslocar a garra a uma das duas colunas. Partindo-se dessa análise é possível observar que todas as colunas de um lado do tabuleiro possuem uma coluna simétrica. Fazendo-se, agora, uma análise nas linhas do tabuleiro observa-se que existem linhas simétricas entre os dois lados, de cada jogador, no tabuleiro. Porém, essa simetria é obtida com o movimento dos motores M2 e M3, o que dificulta a utilização desta simetria em uma implementação prática. Supondo-se que o manipulador está posicionado na casa 4R e que deseja-se movimenta-lo para a casa 4B_R, então será necessário modificar o posicionamento dos braços S2 e S3, ou seja, para executar este movimento na mesma linha será necessário acionar o motor M2 para cima e M3 e para baixo. Considera-se o movimento igual para todas as colunas ($R \rightarrow B \rightarrow C \rightarrow T$).

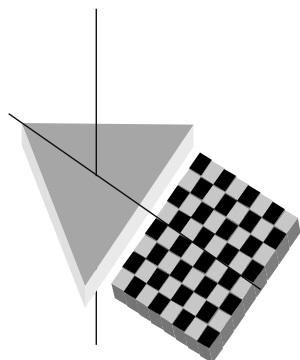


Figura 4. Posição das Estruturas

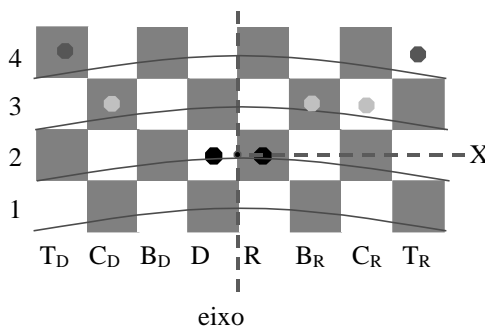


Figura 5. Casas simétricas na mesma linha

4. O SISTEMA INTELIGENTE (SI)

O SI é composto de uma base de conhecimento, onde estão armazenadas as informações, em forma de regras, sobre a localização de todas as casas do tabuleiro de xadrez; um módulo Fuzzy [ZAD88] que permite a inferência sobre as regras; e um módulo ATUADOR encarregado do movimento dos braços do robô. O SI foi implementado utilizando a linguagem de programação C++ Builder da Borland. A cada movimento do AUTOXAD foi associada uma meta ou objetivo. A meta do sistema foi planejada, a partir da analogia de uma

situação do mundo real, isto é, baseou-se na observação dos movimentos realizados por um jogador de xadrez. Com isso, foi possível predefinir uma seqüência de movimentos necessários para a realização da meta desejada. Definiu-se quatro estratégias: IDENTIFICA, PEGA, MOVE E SOLTA. A estratégia IDENTIFICA, localiza a posição da peça; PEGA, movimenta o robô para a posição desejada e pega a peça; MOVE, movimenta a peça para a casa desejada; e SOLTA, a garra solta a peça no lugar determinado.

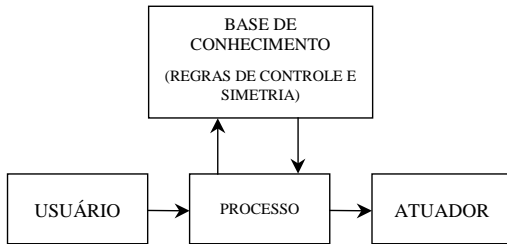


Figura 6. O Diagrama do SI

Essas estratégias foram implementadas em forma de tarefa para que fosse possível executá-las através de um escalonamento em tempo real. A cada tarefa foi associado um descritor. Os descritores são representados pela declaração $descr(id, st, temp, freq)$ onde, id é a identificação da tarefa; st , indica o estado da tarefa (executando=-1, bloqueada=0, e pronta=1); tempo, o tempo de ativação da tarefa em ms; e $freq$, intervalo de tempo em que a tarefa será novamente ativada (Cavalcanti e Alsina, 1997).

Observou-se que para que o SI possa movimentar corretamente o braço, o sistema dependerá de um conhecimento prévio de seu ambiente de trabalho, isto é, deverá ter o conhecimento do tabuleiro e suas casas, as peças utilizadas no jogo, a posição inicial da garra no tabuleiro (centro), e outras informações. Isto significa que o sistema necessita de uma base de conhecimento composta por dados e regras para que o manipulador consiga alcançar a sua meta com êxito. A base de conhecimento deverá também receber as posições simétricas existentes.

A cada movimento das peças no tabuleiro de xadrez foi associado uma casa fonte e uma casa destino. Por exemplo, $3C_D \rightarrow 3B_R$, indicando o movimento de uma peça vinda da terceira linha da coluna do cavalo da dama (fonte) para a terceira linha da coluna do bispo do rei (destino). Foram associados variáveis Fuzzy a essas especificações.

Para as colunas foram associadas duas variáveis, uma discreta (θ_c) e uma variável fuzzy (θ_{cf}) (variável lingüística). O conhecimento da posição de cada coluna no tabuleiro é baseada no conhecimento do tabuleiro e na função de pertinência (ver figura 7). O mesmo foi feito para o conhecimento das linhas.

5. Posicionamento do Eixo do Motor Base (elo L1)

O posicionamento da garra e das peças sobre o tabuleiro é verificado por um sistema de visão não descrito neste trabalho. Foram desenvolvidas algumas regras nebulosas (ver quadro 1) para o reconhecimento das casas fontes e destino durante o movimento de peças no tabuleiro.

No posicionamento da garra do ROBOXAD utilizando SIMETRIA, inicialmente deve-se especificar o posicionamento das casas fonte e destino. Por exemplo, sabendo-se que a peça_fonte está em $3C_D$ e a casa_destino é $3B_R$, o AUTOXAD detecta que o lado_fonte=D, o

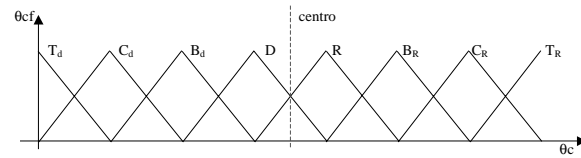


Figura 7. Função de pertinência da posição da coluna

lado_destino=R, a linha_fonte=3, a linha_destino=3, a coluna_fonte=C e a coluna_destino=B. O módulo que analisa a SIMETRIA no movimento da peça da sua casa fonte para a casa destino é apresentado na tabela 1.

Tabela 1. Determinação da simetria

```

SIMETRIA=0
if (posicionado) //Garra posicionada sobre uma casa
then
  if(linha_fonte == linha_destino) //Casas fonte e destino na mesma linha
  then
    if(lado_fonte != lado_destino) //Lados da dama e do rei
    then
      if(coluna_fonte==coluna_destino || coluna_fonte==R || coluna_fonte==D)
      then SIMETRIA = 1

```

Conhecendo-se a existência de SIMETRIA no movimento da garra a seguinte regra Fuzzy e sua tarefa associada representam o movimento do motor da base do AUTOXAD.

```

if (SIMETRIA==1)
then motor base (M1)

```

No caso de SIMETRIA=0 (não há simetria), o movimento do AUTOXAD deverá ser feito pelos motores M1, M2 e M3. Na figura 5 ilustra-se um possível movimento de peças entre as casas $3B_R$ para $3C_R$. O AUTOXAD movimenta os três motores para ajustar o posicionamento da peça. Ele desloca o motor base (M1) na direção da casa_destino. A seguir, ele movimenta o braço S2 para cima e o braço S3 para baixo, posicionando a garra sobre a casa $3C_R$. O mesmo tipo de movimento é feito para o posicionamento nas casas da mesma linha sem simetria.

Na figura 8 ilustra-se a representação do posicionamento da garra do manipulador em relação ao centro das casas do tabuleiro de xadrez. Denomina-se R a distância entre o eixo base e a primeira linha do tabuleiro. A variável x representa a metade do comprimento da linha ou da coluna de cada casa do tabuleiro. Supondo-se que inicialmente a garra esteja sobre o centro da casa 2R com deslocamento x no eixo X e 3x, no eixo Y. A distância da garra ao eixo base do manipulador é representada por $L1 = \sqrt{(R + 3x)^2 + x^2}$.

O ângulo do plano formado pelos braços L2 e L3 em relação ao eixo base é dado pela eq.1. Observe-se que se deve conhecer R e x para se encontrar o ângulo θ_1 . O motor de passo da base do AUTOXAD é conectado, via uma correia, a um disco rígido que está acoplado ao eixo da base. A relação dos raios dos círculos do motor de passo e do disco é igual a 9. Sabendo-se que o motor de passo possui um passo de 7.5 graus, necessitando de 48 passos para uma rotação, calculam-se $D=9*48=432$ passos para uma rotação do disco (ou 360°). Calculou-se 1,2 passos/grau. Por exemplo, para $\theta=15^\circ$, o número de passos é $np=1,2*15=18$ passos.

$$\theta_1 = \arctg\left(\frac{x}{(R + 3x)}\right) \quad (1)$$

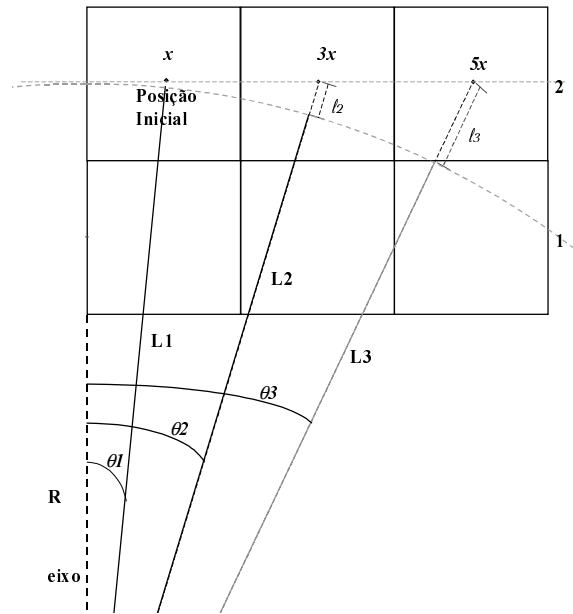


Figura 8. Representação do posicionamento do manipulador

6. Posicionamento dos Eixos dos Motores dos Elos L2 E L3

Na figura 9 apresentam-se dois desenhos no plano cartesiano dos elos L2 e L3. O plano cartesiano é perpendicular ao eixo base do AUTOXAD. Observe-se que J2 está na origem dos eixos X e Y. Considere-se que o braço L2 faz um ângulo θ_2 em relação ao eixo X e L3 faz um ângulo θ_3 em relação ao eixo X, e que a garra do braço L3 está sobre o eixo X. Na figura 9a os elos L2 e L3 possuem o mesmo comprimento, enquanto na figura 9b, eles são diferentes. Utilizando-se geometria básica consegue-se calcular os ângulos θ_2 e θ_3 sendo conhecido o ponto em que a garra encosta no eixo X. Os valores dos senos dos ângulos θ_2 e θ_3 são mostrados na eq.2 e eq.2. A partir das eq.2 e eq.2 obtém-se a relação entre os senos dos ângulos θ_2 e θ_3 mostrados na eq.2. Supondo-se conhecido θ_2 , o valor de θ_3 pode ser calculado utilizando a eq.5.

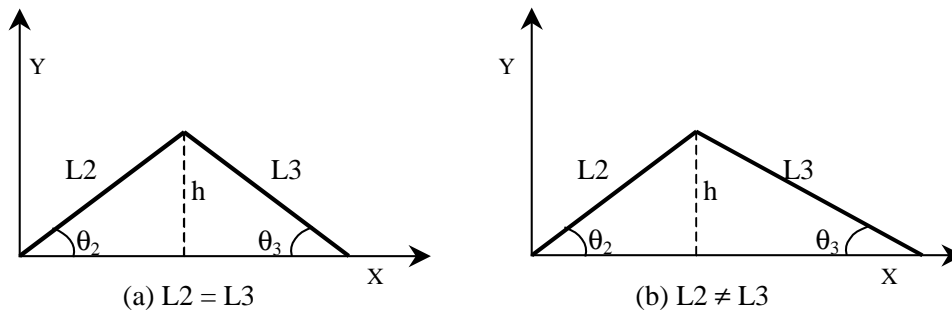


Figura 9. Representação do posicionamento dos elos L2 e L3 no plano cartesiano

$$\text{sen } \theta_2 = \frac{h}{L_2} \tag{2}$$

$$\text{sen } \theta_3 = \frac{h}{L_3} \tag{3}$$

$$\frac{\text{sen } \theta_2}{\text{sen } \theta_3} = \frac{L3}{L2} \quad (4)$$

$$\text{sen } \theta_3 = \frac{L2}{L3} \text{sen } \theta_2 \quad (5)$$

Para que a garra seja movida sobre o eixo X, sobre o plano em que está o tabuleiro de xadrez, os valores dos ângulos θ_2 e θ_3 devem ser iguais. Esta condição é implicitamente especificada na forma de regra Fuzzy quando $\theta_2 = \theta_3$. Na figura 10 apresenta-se o desenho no plano cartesiano de L2 e L3 quando a garra é movimentada sobre o plano do tabuleiro de xadrez. Se os elos L2 e L3 possuírem o mesmo comprimento, a partir da eq.5 obtém-se $\theta_3 = \theta_2$. A regra fuzzy mostrada na eq.6, utilizada no Sistema Inteligente do AUTOXAD, ilustra a movimentação da garra sobre o eixo X quando $\theta_3 = \theta_2$. A variável booleana *garra_sobre_eixo_X* indica que a garra está posicionada sobre o eixo X das coordenadas cartesianas. O AUTOXAD possui braços com comprimentos iguais ($L2=L3$).

$$\text{if } L2==L3 \text{ and } \theta_2==\theta_3 \text{ then } \text{garra_sobre_eixo_X} \quad (6)$$

$$\text{if } \text{lado_fonte}==\text{lado_destino} \text{ then } \theta=\theta_destino-\theta_fonte \quad (7)$$

$$\text{if } \text{lado_fonte}!=\text{lado_destino} \text{ then } \theta=\theta_destino+\theta_fonte \quad (8)$$

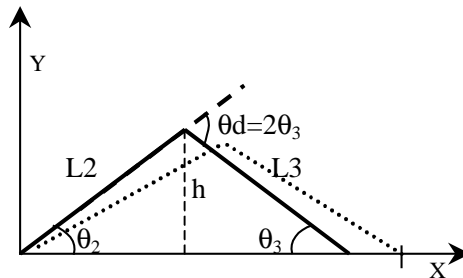


Figura 10. Movimento da garra sobre o plano

7. Resultados Experimentais

A figura 11 ilustra a representação do posicionamento da garra do manipulador sobre as posições fonte ($2R$) e destino ($2C_R$). O primeiro passo do AUTOXAD é verificar, utilizando as regras do quadro 1, se a posição_fonte= $2R$ é simétrica em relação a posição_destino= $2C_R$. Após a confirmação de que as posições fonte e destino não são simétricas deve-se calcular a distância a ser percorrida pelos elos L2 e L3 e o ângulo θ a ser rotacionado pelo elo L1 na direção da posição destino. Na tabela 2 apresentam-se os dados de posicionamento de cada posição, ou seja, os dados de posicionamento da casa fonte e os dados de posicionamento da casa destino. Com esses dados pode-se calcular o deslocamento que deve ser feito pelos elos L1, L2 e L3.

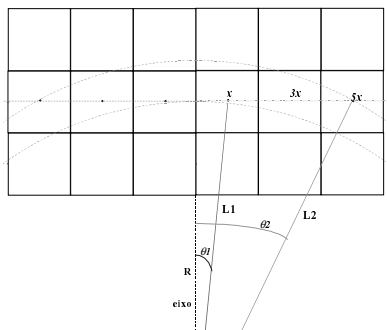


Figura 11. Representação do posicionamento da garra nas posições $2R \rightarrow 2C_R$.

Tabela 2. Coordenadas das posições 2R e 2C_R.

Posição	R	x	y
Fonte	15	2	6
Destino	15	10	6

O cálculo dos deslocamentos que devem ser feito pelos elos L2 e L3 é feito com a fórmula $\sqrt{(R+y)^2 + x^2}$. Obtem-se como resultado: L1=21,09 cm e L2=23,26 cm. A distância a ser percorrida pelos elos L2 e L3 é dada por $l=L2-L1$. Obtem-se $l=2,17$. Sabendo-se que o deslocamento em passos de L3 é duas vezes o deslocamento de L2. Então a quantidade de passos a serem percorridos pelos motores de L2 e L3 são 4 e 8 passos respectivamente. O ângulo rotacional θ_1 é calculado pela regra fuzzy mostrada na eq.7 ($\theta=\theta_{\text{DESTINO}}-\theta_{\text{FONTE}}$), então $\theta = 20,02^\circ$, que corresponde a 24 passos a ser percorrido pelo elo da base (ou L1).

8. CONCLUSÃO

Neste trabalho apresentou-se o projeto, implementação e resultados experimentais obtidos do protótipo de um robô de três graus de liberdade desenvolvido para o posicionamento de um manipulador robótico a ser aplicado em laboratórios de análise clínicas. Utilizou-se um tabuleiro de jogo de xadrez como plataforma de testes do robô. Apresentou-se detalhes do Sistema Inteligente que foi desenvolvido para o controle do robô. Utilizando-se o Sistema Inteligente mostrou-se que é possível o posicionamento do braço do robô baseado na simetria das casas do tabuleiro de xadrez e no posicionamento especial dos dois últimos elos do robô. A utilização da lógica fuzzy possibilitou a representação qualitativa e quantitativa da posição dos braços do robô, permitindo o desenvolvimento de um sistema inteligente para o controle do robô. Atualmente desenvolve-se um sistema inteligente multi agente para controle deste robô.

9. REFERÊNCIAS

- Ferreira, C. N. M. do Amaral e Ferreira, J. . da Silva e Cavalcanti, J. H. F. e Alsina, P. J. e Franca, J. E., 1999, “Posicionamento Inteligente De Um Braço Robótico”, COBEM’99 - Congresso Brasileiro de Mecânica, Águas de Lindóia, SP.
- Cavalcanti, J. H. F. e Alsina, P. J., 1997, “Load Exchange Beteween two Robots”, Workshop on Intelligent Robotcs – WRI’97, Brasília, Agosto de 1997, Pp. 80-88.
- Craig, John J., 1986, “Introduction to Robotic Mechanics & Control”, Addison-Wesley Publishing Company.
- McComb, Gordon, 1987, “Robot Builder’s Bonanza 99 Inexpensive Robotics Projects”, McGrawHill. .
- Zadeh L.A., 1988, “Fuzzy Logic”, IEEE Computer Mag., pp83-93. Publicado também In: Anderson, J.A.; Rosenfeld, E. (Ed.) Neurocomputing Foundations of Researchs, pp.177-194, The MIT Press, Cambridge, USA.

- O trabalho mostra ser viável aplicar a estratégia adotada dentro do conceito de fábrica virtual.

6. AGRADECIMENTOS

Os autores agradecem a FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo, ao NUMA – Núcleo de Manufatura Avançada (PRONEX), e a SandVik do Brasil S.A.

7. REFERÊNCIAS

- Beard, T. “Servicio CNC de larga distancia”. MetalMecanica-International, Colômbia, v. 4, ed. 4, p. 22-24, Ago./Set. 1999.
- Bremer, C. F. “Um sistema para apoio à formação de empresas virtuais baseada em recursos de chão-de-fábrica”. Gestão&Produção, São Carlos, SP, Brasil, v. 6, n.2, p. 79-86, ago. 1999.
- Coppini, N, L., Baptista, E, A. “Alternativas de otimização dos parâmetros de corte para cenários de fabricação em usinagem”. In.: Encontro Nacional de Engenharia de Produção - ENEGEP’97, Niterói, RJ, Brasil, Anais... Out. 1998.
- Coppini, N, L., Coelho, R, T., Baptista, E, A. “Tool Optimization Expert System”. In.: Integrity – Reliability – Failure an International Conference, Porto, Portugal, Anais... July 1999.
- Diniz, A, E., Coppini, N. L., Vilella, R. C., Rodrigues, A, C. S. “Otimização das condições de usinagem em células”. Máquinas e Metais, Brasil, n.281, p. 48-54, Jun. 1989.
- DOMINGUES, M. A., NAZZONI, R. “Como a variação da velocidade de corte em tornos automáticos beneficia a produção”. Máquinas e Metais, Brasil, n.399, p. 186-192, Abril 1999.
- Elbestawi, M. A., Teltz, R. “Intelligent Machining Systems: Opportunities and Challenges”. In.: Integrity – Reliability – Failure an International Conference, Porto, Portugal, Anais... July 1999.
- Ferreira, J. C. E., Stradiotto, C. R. K., Butzke, A. U. “Parâmetros de usinagem e geração do programa NC em CAD/CAPP/CAM”. Máquinas e Metais, Brasil, n.396, p. 54-63, Jan. 1999.
- Hogarth, S. “Remote Possibilities: Machine Diagnostics at a distance”. Manufacturing Engineering, USA, v. 122, n.2, p. 70-78, Feb. 1999.
- Neves, M., Proença, A. “Evolução dos perfis de automação no Brasil”. Máquinas e Metais, Brasil, n.399, p. 83-91, Abril 1999.

A Figura 3 apresenta o fluxograma utilizado pelo *software* gerenciador e pelo sistema especialista na troca de informações através dos arquivos “PARAM.ATD” e “BACK.ATD”.

4. SIMULAÇÃO DE OTIMIZAÇÃO

O ensaio realizado para validação da estratégia de comunicação adotada foi realizado no NUMA em São Carlos. Apesar de os computadores utilizados estarem localizados dentro do mesmo prédio, utilizou-se uma configuração na qual os microcomputadores estavam conectados via Internet.

O envio e o recebimento dos arquivos, "PARAM.ATD" e "BACK.ATD", foram realizados com êxito em apenas alguns segundos, apesar deste depender da velocidade do sistema de rede utilizado. No torno, simulou-se a usinagem de peças, dando início aos ciclos de usinagem para que o *software* gerenciador realizasse a contagem do número de peças usinadas e atualizasse o arquivo "PARAM.ATD". Editou-se manualmente o valor da variável responsável pela medição da peça, com o objetivo de simular o final de vida da ferramenta.

Observa-se que o sistema de monitoramento adotado consistiu em um sensor de emissão acústica, juntamente com um apalpador montado na torre de ferramentas da máquina, já existente e em utilização nos Laboratórios do NUMA. Através deste sensor é possível avaliar o progresso do desgaste da ferramenta pela variação dimensional do diâmetro do corpo de prova.

Graças as características do CNC da máquina-ferramenta utilizada, uma condição do tipo “IF...THEN...” foi possível de ser programada dentro do programa “G” para definir o momento de troca da ferramenta, como sendo aquele para o qual o sensor identificou o limite superior estabelecido para a variação dimensional do diâmetro.

Na outra extremidade, vide Figura 1, o sistema especialista TOES recebeu as informações corretamente e também não apresentou problemas no envio do arquivo "BACK.ATD" alterado. Desta forma, esta simulação permitiu-nos verificar o bom funcionamento da estratégia desenvolvida.

Propositalmente, o cabo de rede que fazia a conexão entre os microcomputadores foi desconectado com a finalidade de simular a queda da linha telefônica e com ela, a interrupção da comunicação via Internet, como resultado o torno interrompeu a fabricação aguardando a resposta do *software* gerenciador. Isto demonstrou que a fragilidade do sistema de comunicação adotado, como uma linha telefônica ou mesmo um cabo de rede, pode comprometer o sistema produtivo a ser otimizado, entretanto, este problema está sendo resolvido através do desenvolvimento de uma rotina em que a máquina-ferramenta terá um tempo máximo para o recebimento de informações do *software* gerenciador. Após este tempo limite, que deverá ser configurado em função do ambiente de rede utilizado, o *software* gerenciador liberará a máquina-ferramenta para a usinagem e somente após o restabelecimento da comunicação via rede ou Internet, da-se-á seqüência ao processo de otimização que fora interrompido, sem entretanto, comprometer a produção de peças

5. CONCLUSÕES

O presente trabalho permite-nos concluir:

- A estratégia de comunicação desenvolvida permite obter a completa automação em otimização de processos de usinagem;
- A metodologia de otimização se mostrou adequada para sistemas automatizados;
- O processo produtivo não corre o risco de sofrer interrupções, em função da fragilidade do sistema de comunicação adotado.

O sistema especialista TOES interage com o CNC da máquina ferramenta através de dois arquivos previamente definidos, que são o arquivo “PARAM.ATD” e o arquivo “BACK.ATD”. O primeiro “PARAM.ATD” é gerado pelo *software* gerenciador e é lido pelo sistema especialista TOES. Já o TOES ao receber as informações existentes no arquivo “PARAM.ATD”, realiza os cálculos necessários e devolve os resultados gravados no arquivo “BACK.ATD”.

Ambos arquivos possuem o mesmo conteúdo, sendo que a diferença entre os nomes adotados foi determinada para evitar possíveis problemas em função de interrupções do sistema de comunicação utilizado. A estrutura do arquivo adotado, é mostrada na Figura 2.

R0	Ordem se serviço	Utilizada pelo CNC da máquina-ferramenta, no armazenamento do nome do programa CNC para a usinagem da peça.
R1	Fim de vida da ferramenta	Possui valor <i>default</i> = 0. No momento em que é decretado o final da vida da ferramenta, este valor é alterado para 1.
R2	Diferença	Armazena o valor obtido na medição da peça. A máquina utilizada possui sistema de medição através de um apalpador em conjunto com sistema de monitoramento por emissão acústica. Após a usinagem de cada peça, é realizada sua medição, através do toque do referido apalpador que encosta na peça, e cujo momento do toque é acusado pela emissão acústica. Neste momento, é armazenado o valor dimensional obtido.
R3	Reservado	Reservado para eventual uso de outros sistemas de monitoramento.
R4	Reservado	Reservado para eventual uso de outros sistemas de monitoramento.
R5	Velocidade de Corte (Vc)	Armazena o valor da velocidade de corte.
R6	Avanço de corte (fn)	Armazena o avanço de corte.
R7	Profundidade de corte (ap)	Armazena o valor da profundidade de corte.
R8	Número de peças usinadas	Armazena o número de peças usinadas. A cada início de ciclo de usinagem da peça, soma-se 1 ao valor inicial (<i>default</i> = 0)
R9	Limite de rotação	Armazena o valor da rotação adotada como limite para a operação.
R10	Medição da peça	Armazena a dimensão limite para o decreto do final da vida da ferramenta.
R11	Reservado	Reservado para eventual uso de outros sistemas de monitoramento.
R12	Controle de liberação	Possui valor <i>default</i> = 0, neste caso, o sistema especialista TOES, ou o <i>software</i> gerenciador, lê as informações do arquivo, mas não dá prosseguimento ao processo de otimização. No momento em que o final de vida da ferramenta é decretado, este valor é alterado para 1, liberando assim, o <i>software</i> gerenciador, ou o TOES, para prosseguir com a otimização.

Figura 2. Estrutura dos arquivos PARAM.ATD e BACK.ATD.

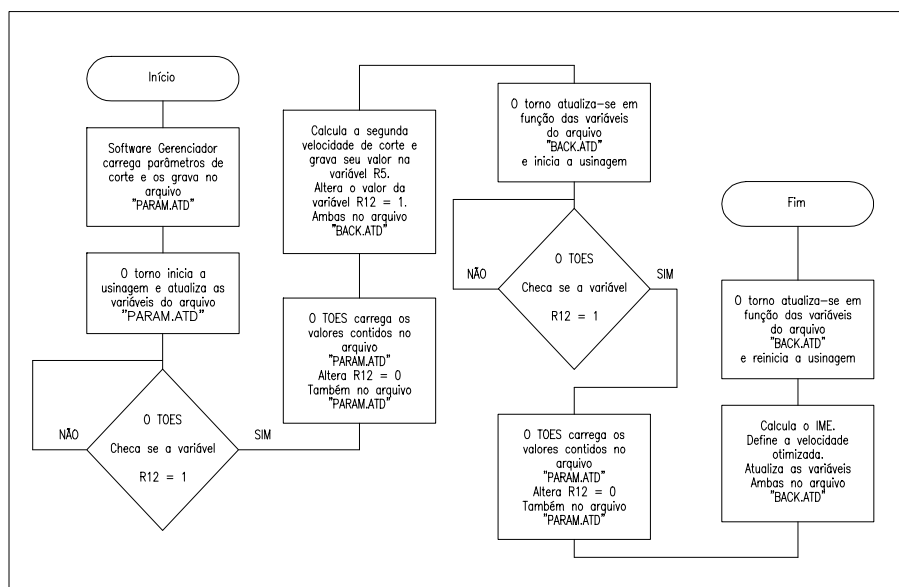


Figura 3. Fluxograma da estratégia de comunicação.

3. ESTRATÉGIA DE COMUNICAÇÃO ENTRE TOES E CNC

A comunicação entre o sistema especialista TOES e o CNC da máquina ferramenta pode ser realizada basicamente via uma rede do tipo LAN (*Local Area Network*) ou uma rede do tipo WAN (*Wide Area Network*).

A rede do tipo LAN permite a conexão e compartilhamento de dados entre microcomputadores em um mesmo ambiente, como uma fábrica ou um escritório. Neste caso, o sistema especialista TOES pode ser instalado em uma máquina localizada em qualquer ponto de conexão da rede, próxima ou não da máquina ou máquinas a serem otimizadas.

Para que a otimização de um processo de usinagem seja possível de ser realizada via Internet, deve-se utilizar uma rede do tipo WAN, neste caso, o microcomputador, no qual reside o sistema especialista TOES, pode estar localizado em qualquer parte onde se tenha acesso a Internet.

As características da máquina-ferramenta utilizada, um torno CNC INDEX pertencente ao NUMA – Núcleo de Manufatura Avançada, da Escola de Engenharia - USP de São Carlos, facilitou o desenvolvimento dos trabalhos devido ao fato de o mesmo possuir um microcomputador do tipo padrão IBM-PC, que é responsável pelo controle de seu CNC. Desta forma, uma placa de rede foi instalada neste microcomputador, permitindo assim, o compartilhamento de sua unidade de disco rígido em um ambiente de rede, LAN ou WAN.

Outra característica importante desta máquina é a possibilidade de uso de variáveis de programação dentro dos programas de usinagem desenvolvidos em linguagem “G”. Assim, é possível reservar uma variável dentro do programa “G”, sendo que o valor desta variável é definido em um arquivo externo, que possui formato texto (ASCII).

Este arquivo pode ser lido, alterado e gravado enquanto o torno realiza operações de usinagem, considerando que somente a cada início de ciclo de usinagem o CNC do torno carrega as informações nele contidas. Este arquivo também é importante para a contagem do número de peças devido ao fato de que uma rotina foi desenvolvida, em linguagem de programação disponível pelo comando numérico, para adicionar o valor 1 (um) a cada início de ciclo de trabalho, em uma variável predefinida que será apresentada mais adiante.

Um outro computador foi utilizado como servidor, também está localizado no NUMA, e faz a conexão entre o sistema especialista TOES e o CNC da máquina-ferramenta. Este servidor foi empregado para possibilitar maior segurança ao sistema. Neste foi instalado um *software* gerenciador, desenvolvido no LabView® – *software* que apresenta recursos para o desenvolvimento de aplicações de aquisição de dados.

Este gerenciador é responsável pela aquisição/recebimento de informações existentes no CNC da máquina ferramenta e no sistema especialista TOES, finalmente, tem-se o computador em que esta instalado o sistema especialista TOES. Um esquema deste sistema é mostrado em detalhes na Figura 1.

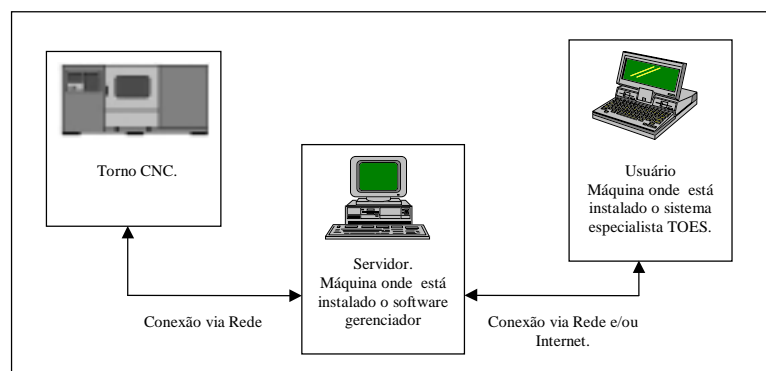


Figura 1. Esquema do sistema utilizado para otimização.

$$k = Zt_1 * (Vc_1)^x * tc_1 \quad (3)$$

$$x = \frac{\log\left(\frac{T_1}{T_2}\right)}{\log\left(\frac{Vc_2}{Vc_1}\right)} \quad (4)$$

Onde tc é tempo de corte, Zt é o número de peças, Vc é a velocidade de corte e T é a vida da ferramenta em minutos. O índice $_1$ representa uma situação diferente do índice $_2$.

E) Calcular : a velocidade de corte de mínimo custo, $V_{c_{mc}}$, velocidade de corte de mínimo custo limite, $V_{c_{mcLim}}$, e a velocidade de corte de máxima produção, $V_{c_{mxp}}$. A $V_{c_{mc}}$, juntamente com a $V_{c_{mxp}}$, compõem os extremos do IME. Já a $V_{c_{mcLim}}$, sempre pertencerá ao IME. Para o cálculo destas velocidades, com conseqüente determinação do IME, devem-se utilizar as equações 5, 6 e 7 descritas logo abaixo :

$$V_{c_{mc}} = \left(\frac{k * (Sh + Sm)}{60 * (x - 1) * [K_{ft} + (Sh + Sm) * T_{ft}]} \right)^{\frac{1}{x}} \quad (5)$$

$$V_{c_{mcLim}} = \left(\frac{k * (Sh + Sm)}{60 * (x - 1) * K_{ft}} \right)^{\frac{1}{x}} \quad (6)$$

$$V_{c_{mxp}} = \left(\frac{k}{(x - 1) * T_{ft}} \right)^{\frac{1}{x}} \quad (7)$$

Onde Sh é o salário homem, Sm é o salário máquina, K_{ft} é o custo do ferramental e T_{ft} é o tempo de troca da ferramenta.

F) Após definição do IME, pode-se adotar a velocidade de corte otimizada para o sistema máquina-ferramenta-peça, isto pode ser realizado respeitando-se o sistema produtivo envolvido. Esta tarefa é realizada pelo TOES e não será abordada aqui por não possuir envolvimento direto com a estratégia de comunicação desenvolvida, tema principal deste trabalho.

Como pode ser observado na metodologia apresentada acima, será necessário anotar os resultados obtidos na usinagem dos lotes de peças quando varia-se a velocidade de corte. Esta tarefa pode ser realizada pelo próprio operador da máquina. Deste modo, o TOES trabalha no modo interativo.

Todos os cálculos necessários são realizados pelo TOES, entretanto, no modo interativo cabe ao operador ou responsável pela otimização do processo a alteração da velocidade de corte na máquina CNC, e também a introdução dos resultados obtidos no TOES. Porém, estas tarefas são realizadas sem interferência humana no modo de otimização automático. Portanto, a rotina de comunicação entre os microcomputadores deve apresentar confiabilidade, evitando-se assim, possíveis acidentes e prejuízos ao sistema produtivo.

O uso da Internet para realização de diagnósticos a distância, em problemas ocorridos em máquinas-ferramenta, vem ocorrendo em empresas Norte Americanas, (Hogarth, 1999) e (Beard, 1999). Nestes casos, utilizou-se de recursos de vídeo conferência para que o responsável pelo suporte técnico orientasse os clientes na outra extremidade, onde, a máquina-ferramenta apresenta problemas. Com este recurso obteve-se economia em relação aos dispêndios ocasionados por viagens, que são realizadas apenas nos casos em que o problema da máquina-ferramenta não apresentou solução adequada com o uso da nova ferramenta.

O presente trabalho objetiva apresentar a estratégia de comunicação desenvolvida que possibilita a otimização da velocidade de corte pelo TOES dentro do conceito de fábrica virtual. Para isto, apresentar-se-á a metodologia adotada para otimização do processo de usinagem, em seguida a estratégia de comunicação desenvolvida e os resultados obtidos em uma simulação de otimização em que a mesma foi testada. Finalmente, apresentam-se as conclusões que demonstram a possibilidade de seu uso na otimização do processo de usinagem em modo completamente automático.

2. METODOLOGIA PARA OTIMIZAÇÃO

A otimização do processo de usinagem no sistema especialista TOES baseia-se na metodologia proposta por Diniz *et al.* (1989). Esta metodologia consiste na determinação, em ambiente fabril, das constantes x e k da equação de Taylor, e conseqüentemente do IME, conforme descrito abaixo.

- A) Determinar os parâmetros de corte de acordo com os procedimentos em uso por parte do responsável pela elaboração do processo de usinagem, como a adoção de valores sugeridos por catálogos de fabricantes de ferramentas, com base na experiência do operador, ou até mesmo, recuperando-se valores armazenados em banco de dados oriundos de experimentos anteriores. Observa-se que os valores da profundidade de corte, ap , e do avanço de corte, fn , adotados, devem possuir os valores máximos possíveis em relação às restrições inerentes ao sistema máquina-ferramenta-peça.
- B) Iniciar a usinagem do primeiro lote de peças até que o final de vida da aresta da ferramenta seja decretado por um critério previamente estabelecido. Anotar o número total de peças, Z_{t1} , usinadas para a primeira aresta da ferramenta.
- C) Alterar o valor da velocidade de corte em $\pm 20\%$ em relação a velocidade de corte utilizada para a usinagem do primeiro lote de peças. Em seguida, iniciar a usinagem do segundo lote de peças até que seja decretado o final de vida da aresta da ferramenta, pelo mesmo critério anterior. Neste momento deve-se anotar o número de peças usinadas, Z_{t2} , para a segunda aresta da ferramenta.
- D) Após a realização da usinagem obtêm-se os valores da vida da ferramenta expressos em número de peças, em que para V_{c1} , obteve-se Z_{t1} , e para V_{c2} , obteve-se Z_{t2} . Assim é possível a determinação das constantes x e k através das equações 2 e 3. É importante observar que a vida da ferramenta também pode ser expressa em comprimento de corte, em metros, ou em tempo, em minutos. Nos casos em que a vida da ferramenta for expressa em comprimento de corte, pode-se também utilizar as equações 2 e 3. Entretanto, em situações em que a vida da ferramenta for expressa em tempo, deve-se utilizar a equação 4 para o cálculo da constante x , juntamente com a equação 3 para o cálculo da constante k .

$$x = \frac{\log\left(\frac{Z_{t1}}{Z_{t2}}\right)}{\log\left(\frac{V_{c2}}{V_{c1}}\right)} + 1 \quad (2)$$

processistas e/ou engenheiros de processo, buscando-se assim, reduzir principalmente os tempos de corte com conseqüente redução do tempo de fabricação da peça. Destaca-se que o emprego de novas ferramentas de corte normalmente respeita as condições sugeridas pelo catálogo do fabricante. São pouco frequentes os relatos sobre a otimização dos parâmetros de corte para a ferramenta de corte em uso.

Tratando-se de parâmetros de corte, pode-se otimizar a profundidade de corte, ap , o avanço, fn , e a velocidade de corte, vc . A profundidade de corte é adotada em função de restrições físicas inerentes à peça, ferramenta e/ou máquina-ferramenta tais como: a camada de sobremetal a ser retirada, o fio de corte da ferramenta, a potência da máquina-ferramenta e outros limitantes referentes aos dispositivos de fixação da peça. Assim sendo, procura-se empregar a maior profundidade de corte possível entre estas restrições.

O avanço é definido basicamente em função do acabamento superficial desejado, desta forma, sua utilização está limitada à faixas de avanço que atendam a este requisito.

Já a velocidade de corte apresenta grande flexibilidade com relação à sua otimização, apesar de seu efeito negativo em relação à vida da ferramenta. Entretanto, otimizando-se a velocidade de corte com base no intervalo de máxima eficiência, IME, podem-se obter melhores resultados com relação aos custos e tempos produtivos (Coppini & Baptista, 1998).

Logo, pode-se otimizar o processo de usinagem do sistema máquina-ferramenta-peça em uso, através de seus parâmetros de corte. Para que a otimização apresente resultados coerentes com o sistema envolvido, as constantes x e k da equação de vida de Taylor devem ser determinadas em ambiente fabril (equação 1).

$$T * v^x = K \quad (1)$$

Onde T é a vida de ferramenta, v é a velocidade de corte.

Entretanto, a constante necessidade de obtenção de dados relativos ao processo de usinagem, aliada à necessidade de resolução de cálculos complexos para a obtenção destas constantes, tem afastado o emprego desta técnica por parte de processistas. Este problema pode ser resolvido através do emprego de sistemas especialistas de usinagem.

Um sistema especialista, o TOES – *Tool Optimization Expert System*, foi especialmente desenvolvido para a otimização da velocidade de corte em ambiente fabril, respeitando-se o cenário produtivo em que é executado o processo alvo da otimização (Coppini *et al*, 1999).

Este sistema pode ser utilizado de modo interativo ou de modo automático. O modo interativo é aquele em que o operador, ou pessoa responsável pela otimização do processo, alimenta o sistema com os resultados oriundos da usinagem e aguarda o resultado calculado pelo TOES.

Visando o aproveitamento dos avanços tecnológicos obtidos na área de tecnologia da informação, principalmente no que diz respeito à Internet, desenvolveu-se uma estratégia que possibilita a interação entre o TOES e o CNC. Esta foi definida de modo que o TOES e o CNC podem estar situados em localidades diferentes, numa mesma fábrica, ou até mesmo, separados por milhares de quilômetros. Deste modo, é possível realizar a otimização de um processo de usinagem através da Internet dentro do conceito de fábrica virtual, de maneira completamente automática.

O modo automático é aquele em que a otimização é realizada sem qualquer intervenção humana, pois, os dados sobre o processo produtivo são previamente cadastrados, assim como no modo interativo, mas neste caso, o sistema especialista recebe as informações complementares necessárias à otimização do processo de usinagem diretamente do CNC da máquina-ferramenta.

OTIMIZAÇÃO DO PROCESSO DE USINAGEM APLICADO DENTRO DO CONCEITO DE FÁBRICA VIRTUAL

Elesandro Antonio Baptista ⁽¹⁾

Reginaldo Teixeira Coelho ⁽²⁾

Nivaldo Lemos Coppini ⁽¹⁾

⁽¹⁾ Faculdade de Engenharia Mecânica e de Produção - FEMP

⁽¹⁾ Universidade Metodista de Piracicaba – UNIMEP - E-mail: m9803057@unimep.br

⁽²⁾ NUMA - Núcleo de manufatura avançada - Escola de Engenharia de São Carlos

⁽²⁾ Universidade de São Paulo - USP - E-mail: rtcoelho@sc.usp.br

RESUMO

A otimização dos processos produtivos tem recebido especial atenção por parte das indústrias nos dias de hoje, visto o aumento da concorrência propiciado, principalmente, pela globalização. Com o objetivo de proporcionar ganho de competitividade em processos de usinagem, desenvolveu-se o TOES – *Tool Optimization Expert System*. Um sistema especialista que possibilita a otimização da velocidade de corte em ambiente fabril, respeitando-se o cenário produtivo e o sistema máquina-ferramenta-peça envolvidos na otimização. Para possibilitar o emprego do TOES em fábricas virtuais, desenvolveu-se uma estratégia para comunicação entre TOES e CNC (Comando Numérico Computadorizado). O presente trabalho objetiva apresentar as estratégias utilizadas para o interfaceamento entre sistema especialista e CNC, que possibilitaram a otimização dos parâmetros de corte de maneira automática, inclusive via rede ou Internet, dentro do conceito de fábrica virtual. Os resultados de testes preliminares permitem concluir que a estratégia utilizada foi adequada, apesar da fragilidade do sistema de comunicação utilizado.

Palavras-chave: Otimização, Sistemas Especialistas, Fábrica Virtual

1. INTRODUÇÃO

Os recentes avanços tecnológicos propiciaram considerável ganho de competitividade para as indústrias manufatureiras, principalmente para as indústrias dos países desenvolvidos que possuem acesso, normalmente mais rápido e em primeira mão, às novas tecnologias. Somando-se a este fato o advento da globalização, tem-se atualmente no Brasil um mercado altamente agressivo, e em muitos casos predatório, com relação à competitividade imposta por produtos importados.

Este novo cenário desencadeou uma corrida, principalmente por parte das indústrias nacionais, em direção à atualização tecnológica. Buscando-se, assim, diminuir o atraso tecnológico existente em relação às indústrias dos países desenvolvidos, e garantir sua sobrevivência (Neves & Proença, 1999).

Em indústrias manufatureiras que possuem, entre seus processos produtivos, processos de usinagem, sempre buscou-se a otimização do processo de fabricação. O emprego de novos dispositivos de fixação, de sujeição, de novas ferramentas de corte, ou até mesmo, o emprego de máquinas mais evoluídas tecnologicamente, são recursos normalmente empregados por

MEDIÇÃO DA DIREÇÃO E INTENSIDADE DO VENTO UTILIZANDO UM ANEMÔMETRO INTELIGENTE

José Homero Feitosa Cavalcanti

Joelson Nogueira de Carvalho

Everaldo Fernandes Monteiro

José Felício da Silva

Universidade Federal da Paraíba - CCT/DSC/COPIN/NEUROLAB – CT/DTM

Rua Aprígio Veloso, 882 - Campus II - 58109-790 - Campina Grande - PB

E-mail: homero@dsc.ufpb.br

Resumo

Este artigo apresenta o projeto e detalhes obtidos da construção de um anemômetro projetado para detectar as variações da direção e da intensidade dos ventos. Também são apresentados resultados obtidos da aferição do anemômetro utilizando Lógica Fuzzy.

Palavras-chave: Anemômetro, Sistemas Inteligente e Lógica Fuzzy.

1. INTRODUÇÃO

No mundo atual são importantes as informações sobre o tempo e principalmente sobre a direção e a intensidade dos ventos. A energia eólica é a energia que se pode obter a partir da força do vento. Para bem utiliza-la deve-se conhecer a direção, intensidade e a regularidade dos ventos utilizando anemômetros. Existem diversos tipos de anemômetros desenvolvidos para aplicações específicas que fornecem a direção (figura 1.a) e intensidade (figura 1.b) dos ventos. Esses anemômetros são de difícil utilização e geralmente exigem placas de aquisição de dados e softwares em tempo real, recurso geralmente de alto custo e difíceis de serem utilizados.



Figura1. Anemômetro, direção e intensidade do vento

Diversos pesquisadores têm desenvolvido sistemas de aquisição de dados e análise utilizando microcomputadores. Por exemplo, Araújo (1989) apresentou o projeto e a implementação de um sistema de aquisição, construção de uma base de dados, e processamento de dados meteorológicos utilizando microcomputador. Ele monitorou, utilizando microcomputador, as características de regime do vento disponível em uma determinada região.

O nosso laboratório está engajado em pesquisas que necessitam de instrumentos de medição de baixo custo e que utilizem as interfaces padrão dos microcomputadores do tipo IBM PC. A partir dos resultados obtidos do desenvolvimento de um sistema estabilizador de planos (Cavalcanti et al, 1999), desenvolveu-se um novo modelo de anemômetro, denominado BIRUTA, que é um sistema experimental para medição da direção e intensidade do vento usando Lógica Fuzzy e microcomputadores. Na figura 2 apresenta-se a fotografia do sistema BIRUTA.

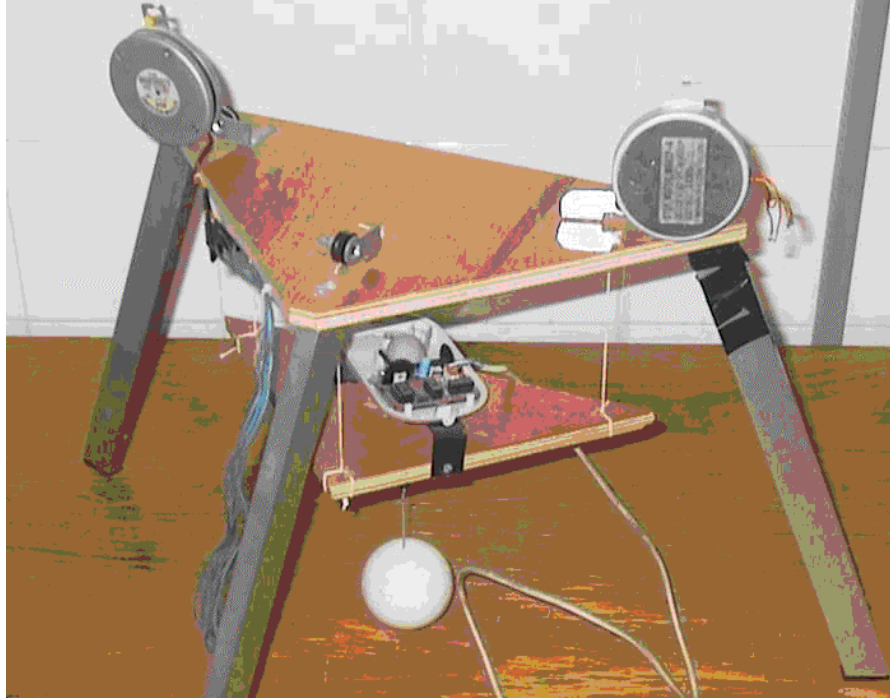


Figura 2. O sistema BIRUTA

Neste trabalho inicialmente descreve-se sucintamente o sistema BIRUTA. A seguir, apresenta-se o software desenvolvido para obtenção dos dados da direção e intensidade do vento e se apresentam resultados preliminares da aferição deste instrumento utilizando a Lógica Fuzzy (Zadeh, 1988, 1965). Por último, conclui-se este trabalho com comentários e sugestões para trabalhos futuros.

2. O SISTEMA BIRUTA

A detecção da intensidade e da direção do vento foi feita com um pêndulo com uma bola de isopor acoplada à extremidade do seu braço. No lado direito da figura 3 apresenta-se o desenho do protótipo do sistema BIRUTA, no seu lado esquerdo mostra-se o desenho do sistema de eixos cartesianos definidos pelo pêndulo.

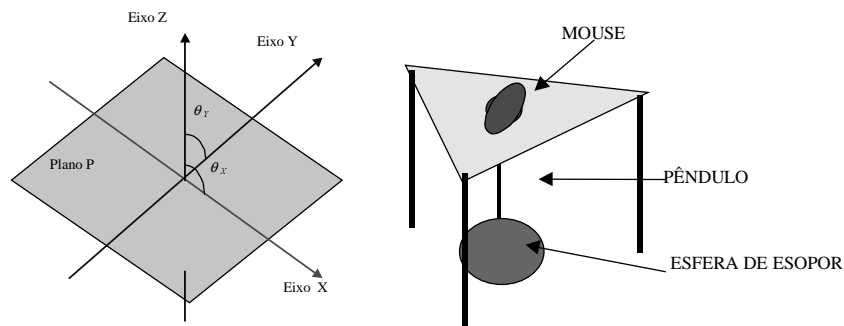


Figura 3. O Sistema BIRUTA

O pêndulo está localizado onde passa o eixo imaginário Z, considera-se que o braço do pêndulo tem comprimento fixo e apenas os valores dos eixos X e Y podem variar. Havendo uma mudança na intensidade ou direção do vento, o pêndulo se move. O movimento do pêndulo é feito sobre os eixos X e Y do plano. Utilizou-se o periférico "mouse" dos microcomputadores como o registrador de coordenadas X e Y do pêndulo. O pêndulo permanece sempre no sentido vertical. Quando o vento muda, o pêndulo sofre a sua ação, o dispositivo giratório do mouse registra em tempo real a magnitude da variação dos eixos em termos de unidades de coordenadas. A partir das coordenadas X e Y fornecidas pelo "mouse" o software gera o vetor intensidade e direção do vento.

3. ANÁLISE DOS DADOS

Na figura 4 apresenta-se esquematicamente um pêndulo simples cujo comportamento dinâmico é descrito matematicamente pela equação diferencial não linear mostrada na eq.1. Na eq.1 t é o tempo na forma contínua, Tl(t) é torque de carga, P é o peso do pêndulo, L é o comprimento do pêndulo, $\theta(t)$ é o ângulo entre o pêndulo e a vertical, g é a aceleração da gravidade. Observe-se que o pêndulo necessita de um torque externo ($T_e > T_l$) para se movimentar. Considerando-se a aceleração do pêndulo igual a zero, o torque para movimentar o pêndulo é apresentado na eq.2.

$$Tl(t) = \frac{1}{2}(P/g)L \frac{d^2\theta(t)}{dt^2} + PL\text{sen}(\theta(t)) \quad (1)$$

$$Tl(t) = PL\text{sen}(\theta(t)) \quad (2)$$

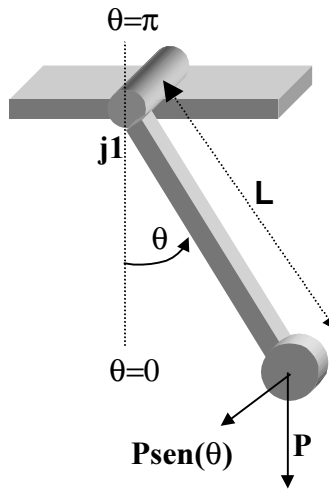


Figura 4. O Pêndulo Simples

Os dados obtidos do mouse indicam movimentos do pêndulo nos eixos X e Y. Esses dados devem ser transformados em intensidade e direção do vento. O movimento do pêndulo no sistema BIRUTA está representado na figura 5, na área formada pela semi-esfera de raio L, igual ao comprimento do pêndulo. A distância da bola presa ao pêndulo ao eixo Z será dada pela eq.3, e o ângulo entre o pêndulo e os eixos X e Y é dado pela eq.4. O eixo Y do mouse foi orientado na direção norte e sul, e o eixo X foi orientado na direção oeste e leste.

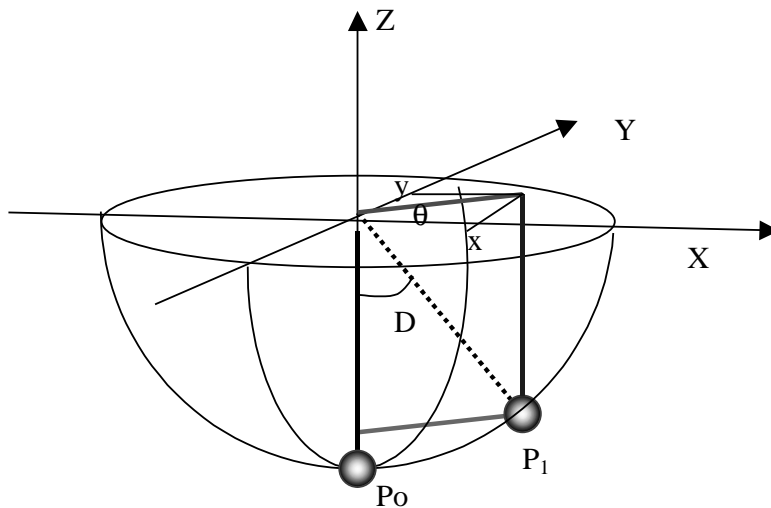


Figura 5. Detecção de intensidade e direção do vento

$$D = \sqrt{X^2 + Y^2} \quad (3)$$

$$\theta = \arctg(Y/X) \quad (4)$$

4. O SISTEMA DETECTOR DE POSIÇÃO DO BIRUTA

Foi implementada uma interface visual gráfica para permitir, em um monitor de vídeo, o acompanhamento de todo o processo de estabilização do plano. Na figura 6 é apresentada a tela inicial do BIRUTA.

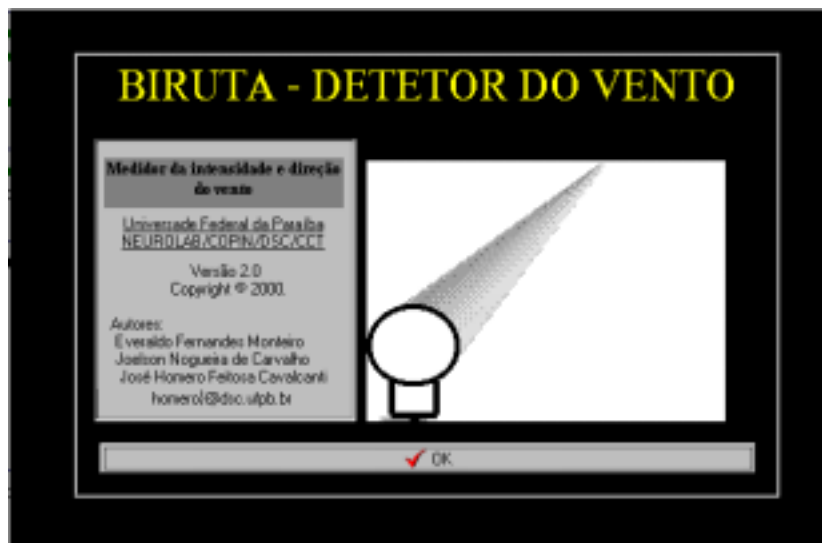


Figura 6. Tela inicial do BIRUTA

Na figura 7 apresenta-se a tela de trabalho do BIRUTA. Na sua parte superior esquerda são colocadas as informações para controle dos motores de passo, escolha das portas paralelas do microcomputador, e posicionamento do mouse (bloco Refer Atual). Os motores de passo serão usados para correção da posição ou proteção do anemômetro. Na parte superior central está localizado o eixo de coordenadas em que se movimenta o indicador de posição do mouse. Na parte inferior direita da figura 7 localiza-se um globo com um pêndulo que indica, em tempo real, a posição atual do pêndulo em que está acoplada a bola de isopor. A faixa inferior da figura é utilizada para traçar as curvas das posições X e Y do mouse obtidos experimentalmente. Os resultados apresentados na faixa inferior indicam que o pêndulo inicialmente estava em repouso, a seguir, o pêndulo, impulsionado pelo vento, se movimenta na direção do eixo X sem quase movimento na direção do eixo Y. O pêndulo permanece

alguns instantes posicionado sobre o eixo Y e começa a se movimentar na direção do eixo X até que fique posicionado como mostrado na figura 7.

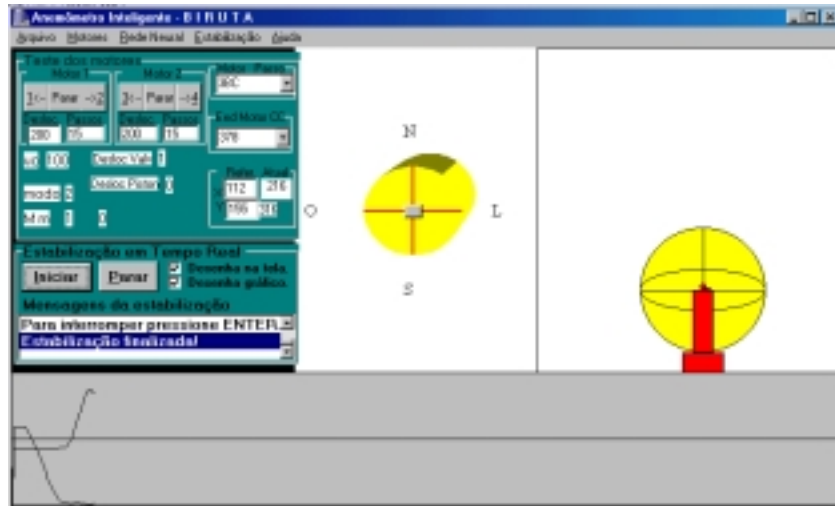


Figura 7. Tela principal do Sistema BIRUTA

5. TRANSFORMAÇÃO DOS DADOS

O NEUROLAB, o laboratório que desenvolveu o sistema BIRUTA, ainda não dispõe de equipamentos padrões (Silva et al, 1999) para a aferição do anemômetro. Decidiu-se fazer uma aferição quantitativa e qualitativa utilizando a Lógica Fuzzy (Zadeh, 1965 e 1988). Decidiu-se testar o anemômetro utilizando resultados obtidos da sua exposição ao ar deslocado por um carro, numa estrada em linha reta, em diferentes velocidades. Supôs-se que era zero a velocidade do vento na estrada. Utilizou-se um pêndulo com 10cm de comprimento.

Tabela 1. Velocidade do vento e deslocamento do pêndulo

#	X(cm)	Y(cm)	D(cm)	V(km/h)
1	0.0	0.0	0.0	0.0
2	6.0	8.0	10.0	7.00
3	9.0	12.0	15.0	10.00
4	18.0	24.0	30.0	23.00
5	27.0	36.0	45.0	39.00
6	33.0	44.0	55.0	55.00
7	35.0	54.0	65.0	75.00
8	48.0	64.0	80.0	103.00
9	54.0	72.0	90.0	118.00

Na Tabela 1 apresentam-se 9 amostras obtidas experimentalmente com um microcomputador “Lap Top” acoplado ao sistema BIRUTA. Na coluna 1 apresentam-se os números das amostras, nas colunas 2 e 3 apresentam-se os deslocamentos no eixo X e Y detectados pelo mouse,, na coluna 4 apresentam-se o valor da inclinação do pêndulo dado pela eq.3, na coluna 5 apresentam-se as velocidades em que foram observados os deslocamentos do pêndulo. Na figura 8, baseado na Tabela 1, apresenta-se a o gráfico da velocidade do vento (velocidade do carro) em função da distância D indicada pela eq.3.

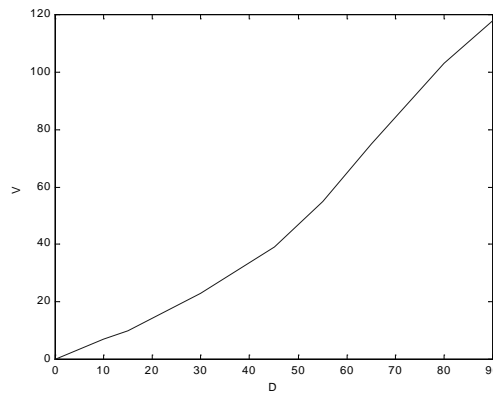


Figura 8. Velocidade do vento (V) em função da distância (D)

5.1 Análise e aferição do instrumento - qualitativo

Atribuiu-se uma variável fuzzy (D_f) para representar a intensidade do vento. Associaram-se diferentes valores lingüísticos para D_f (zero (ZE), pequeno (PP), médio (PM) e grande (PG)). Na figura 9 são apresentadas as curvas das funções de pertinência (escolhidas heurísticamente) da variável fuzzy D_f . Na figura 9 o valor de D foi denominado alvo (valor 0.7). No cálculo qualitativo de D, inicialmente D é calculado usando a eq.3. A seguir, o valor de D é transformado em por unidade (p.u.) com peso base 100 (comprimento do braço do pêndulo em milímetro). O valor qualitativo fuzzy (variáveis lingüísticas) de D_f é calculado usando a operação união do conjunto fuzzy. A função de pertinência de D_f vencedora é a que tem maior valor. Na figura 9 o alvo é uma reta perpendicular à abscissa no ponto $D=0.7$. Neste ponto a função de pertinência vencedora é a PM (maior valor). Portanto, $D_f=PM$ para $D=0.7$.

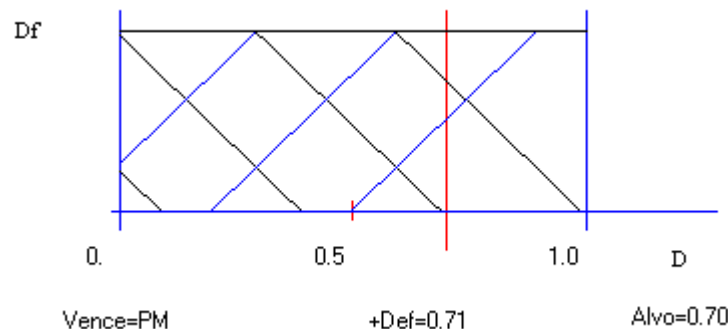


Figura 9. Funções de pertencimento de D_f .

5.2 Análise e aferição do instrumento - quantitativo

No cálculo quantitativo de D, utilizando a “defuzzyficação” da variável fuzzy da velocidade do vento, inicialmente D é calculado usando a eq.3. A seguir ele é normalizado entre zero e um com peso 100. Na Tabela 2 são mostrados os valores discretos atribuídos às variáveis lingüísticas de D_f utilizados na “defuzzyficação” de D_f . A seguir, a partir da figura 9, utilizando o centro de gravidade como mostrado na eq.5 com μ_i representado as funções de pertencimento e w_i representando os pesos. Usando-se os pesos da tabela 2, calculou-se o valor “defuzzyficado” de D_f , denominado D_d , que é mostrado como $+Def=0.71$ na figura 9. Para $D=0.7$, obteve-se $\mu_1=0.0$, $\mu_2=0.3$, $\mu_3=0.8$ e $\mu_4=1.0$. A partir da Tabela 2 obtém-se $w_1=0.0$; $w_2=0.3$; $w_3=0.7$ e $w_4=1.0$. Obteve-se para a intensidade do vento $D_d=+Def=0.71$.

$$I = \frac{\sum_{i=1}^4 \mu_i w_i}{\sum_{i=1}^4 \mu_i} \quad (5)$$

Tabela 2. Pesos das variáveis lingüísticas

ZE	PP	PM	PG
0.	0.3	0.7	1.0

Os valores da velocidade do vento obtidos experimentalmente foi normalizado e desenhados na figura 10 (retas verdes perpendiculares à abscissa com altura proporcional à velocidade do vento). A seguir, fez-se a “defuzzyficação” dos valores de Dd para 20 pontos regularmente espaçados na abscissa da figura 10 (ver curva em vermelho).

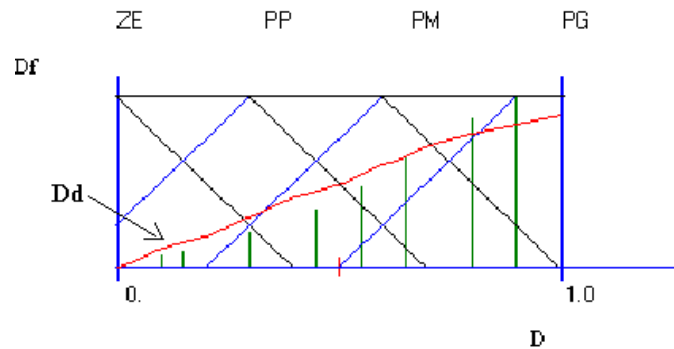


Figura 10. Variável fuzzy Df

Observe-se, na figura 10, que os valores de Dd (curva vermelha) não coincidem (em alguns pontos) com os pontos da velocidade do vento V (retas de cor verde). Definido-se $E=V-Dd$, desenvolveu-se um algoritmo, mostrado abaixo, para a adaptação das funções de pertinência da velocidade do vento, no intuito de tornar a curva de Dd igual a curva de V. Na tabela 3 apresentam-se os resultados finais da adaptação dos pesos das funções de pertinência da variável fuzzy da velocidade do vento. Na figura 11 apresenta-se o gráfico da variável fuzzy velocidade em função do valor de deslocamento do pêndulo obtido após a adaptação dos pesos das funções de pertinência. Observe-se que é mínima a diferença entre os pontos reais e os fornecidos utilizando a “defuzzyficação”.

Faça 1000 vezes

```

{
  Para cada ponto V na abscissa
  {
    calcule Dd;//Valor discreto de D (eq.3)
    E=V-Dd;
    Calcule a função vencedora  $\mu_i$  usando a operação união;
    Adapte o peso da função de pertinência vencedora com  $W_{ik+1}=W_{ik}+hE$ 
  }
}

```

Tabela 3. Pesos das variáveis lingüísticas após adaptação

ZE	PP	PM	PG
0.	0.2	0.5	1.1

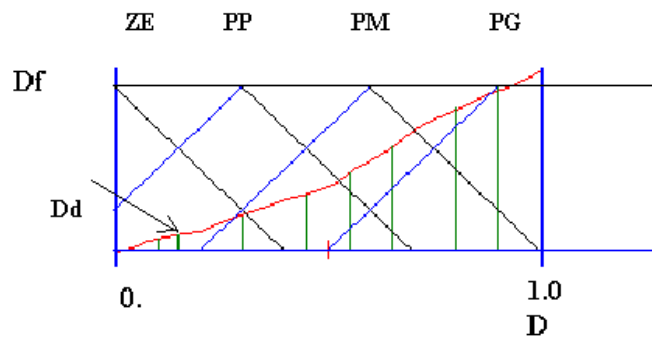


Figura 11. Resultado da aferição do anemômetro

6. CONCLUSÃO

Apresentou-se o projeto, implementação e resultados experimentais de um novo anemômetro utilizando dispositivos de interface padrão de microcomputadores. Mostrou-se a aferição deste novo tipo anemômetro utilizando Lógica Fuzzy o que permitiu a representação qualitativa e quantitativa da velocidade do vento. Futuramente será desenvolvido um software que permitirá o direcionamento do anemômetro (utilizando motores de passo) que o protegerá de falhas devido à intensidade do vento. Também será melhorado o algoritmo de adaptação do anemômetro pela inclusão da modificação do posicionamento e da declividade das funções de pertinência da variável fuzzy que representa a velocidade do vento.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- Ferreira, J. R. da Silva, Carvalho, J. N. de, Cavalcanti, J. H. F., e Monteiro, E. F., 1999 “Sistema Estabilizador de Planos – SESPLA”, COBEM’99 - Congresso Brasileiro de Mecânica, Águas de Lindóia, SP.
- Silva, T.M. & Lima, W.S. & Célere, S.W., 1999 “**Proposta de um Tipo Particular de Anemômetro de Fio Quente: Projeto e Calibração**”, COBEM’99 - Congresso Brasileiro de Mecânica, Águas de Lindóia, SP.
- Araújo, G. M. B. de, 1989, **Aquisição e processamento de dados para sistema eólico**, Dissertação de Mestrado, COPELE (Coordenação de Pós-Graduação em Engenharia Elétrica), Departamento de Engenharia Elétrica da UFPB.
- Zadeh, Lotfi A., 1988, **Fuzzy Logic**, IEEE Computer Mag. April 1988, pp83-93. Republicado por J.A. Anderson & E. Rosenfeld, **Neurocomputing Foundations of Researchs**, MIT Press, Cambridge, Massachusets, USA, 1988. Pp.177-194.
- Zadeh, L.A., 1965, **Fuzzy Sets**, Information and Control, 1965, 8, pp.28-44.

BIBLIOTECA DE CD

José Homero Feitosa. Cavalcanti
Marcelo Cavalcanti Rodrigues
Everaldo Fernandes Monteiro
Universidade Federal da Paraíba - CCT/DSC/COPIN/NEUROLAB – CT/DTM
Rua Aprígio Veloso, 882 - Campus II - 58109-790 - Campina Grande - PB
E-mail: homero@dsc.ufpb.br

Resumo

Neste trabalho apresenta-se o projeto e detalhes de um protótipo de um sistema robótico utilizado na automação de uma Biblioteca de *Compact Disk*. O sistema é constituído de dois manipuladores robóticos, um para movimentar a torre de CD e o outro para movimentar os CD.

Palavras-chave: Sistemas Inteligente, Robótica, Robôs Cooperantes.

1. INTRODUÇÃO

A automação atualmente é uma realidade que está presente nos mais variados processos, desde o processo de fabricação de veículos, empregando altíssima tecnologia, até processos menores como o controle de acervos bibliográficos através de sistemas computacionais. As bibliotecas e os setores de arquivos das empresas possuem microcomputadores que são normalmente utilizados para o controle cadastral dos seus acervos, mas que quase sempre nunca são utilizados no arquivamento e resgate deles.

O estudo de aplicações para desempenhar essas tarefas se faz importante, pois uma grande parte das revistas técnicas e científicas, assim como os congressos, simpósios, conferências e outros, estarem publicando seus periódicos e anais em *compact disks* (CD). Além disso, os microcomputadores, utilizados nas bibliotecas, geralmente são equipados com monitores coloridos, discos rígidos de grande capacidade de armazenamento, interface para acesso a internet, kit multimídia e outros, o que facilita a implantação do sistema robótico.

O volume de CD nas bibliotecas atualmente atinge uma quantidade significativa. Eles geralmente são armazenados precariamente em torres de CD de baixo custo ou estantes sem padronização, o que dificulta o arquivamento e resgate dos CD. A necessidade de um processo automatizado para uma biblioteca de CD (CDTECA) conduziu ao desenvolvimento de um sistema robótico para desempenhar tal função. Esse sistema robótico pode ser construído de uma forma que permita a interação entre dois robôs ou entre o robô e um homem.

Diversos pesquisadores têm apresentado resultados experimentais obtidos com sistemas em que existe a cooperação entre dois robôs e entre um homem e um robô. Por exemplo, Fujisawa et al (1994) apresentam uma sucinta relação de trabalhos referentes a interação entre o homem e o robô manipulador. Osumi & Arai (1994) apresentam diversos tipos de algoritmos desenvolvidos para controle de robôs cooperativos

O robô proposto neste trabalho, denominado MOVECD, é composto de dois manipuladores, um deles é utilizado para movimentar a torre de CD (robô POSIÇÃO) e o outro para manipular o CD (robô PEGACD). O sistema de controle inteligente (SCI) controla o robô através de técnicas de Inteligência Artificial.

A seguir apresenta-se a metodologia de desenvolvimento do sistema MOVECD, as características dos dois robôs, o desenvolvimento do SCI para o MOVECD, a operação do MOVECD sem o robô POSIÇÃO, a operação do MOVECD com o usuário simulando o robô POSIÇÃO, e a operação do MOVECD com o robô POSIÇÃO. Finalmente, conclui-se este trabalho com comentários sobre os resultados experimentais obtidos das experimentações realizadas e se sugerem melhoramentos para futuras implementações.

2. METODOLOGIA DE DESENVOLVIMENTO DO SISTEMA

Na figura 1 apresenta-se a fotografia do protótipo do MOVECD. Nessa figura, a torre de CD está no lado esquerdo, o elevador na parte central, e o trilho na parte inferior da figura. Na figura 2 apresenta-se a fotografia da garra do MOVECD.

O projeto, o desenvolvimento e a construção do protótipo do MOVECD foram feitos a partir da observação (analogia) da forma como um usuário manipula (coloca/retira) um CD numa torre de CD. Observaram-se três diferentes estratégias de manipulação dos CD:

a)O usuário, além de olhar, utiliza as suas duas mãos para manipular o CD. Ele apóia uma das mãos sobre a torre de CD (robô POSIÇÃO) e com a outra mão (robô PEGACD) - utilizando os dedos polegar, indicador e médio- posiciona o CD e o coloca/retira da sua locação;

b)O usuário, sem olhar para a torre de CD, utiliza as suas duas mãos para manipular o CD, ele apóia uma das mãos sobre a torre de CD (robô POSIÇÃO) e com a outra mão (robô PEGACD) -utilizando os dedos polegar, indicador e médio- posiciona o CD e o coloca/retira da sua locação;

c)O usuário, sem olhar para a torre de CD, utiliza uma das mãos para manipular o CD (robô PEGACD) -utilizando os dedos polegar, indicador e médio- posiciona o CD e o coloca/retira da sua locação;



Figura1. Protótipo do MOVECD

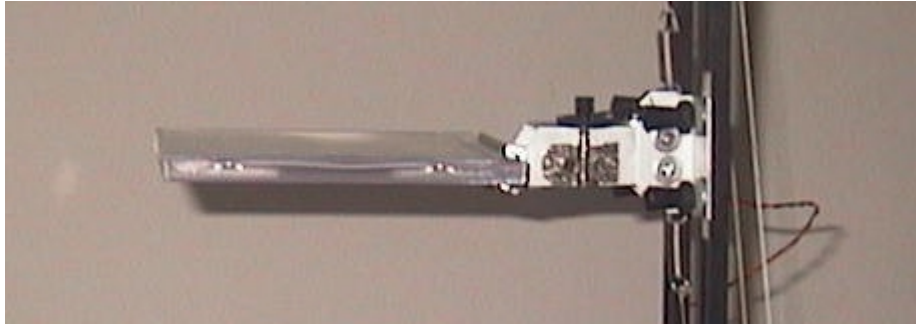


Figura 2. A garra do MOVECD

3. CARACTERÍSTICAS DO MOVECD

Na figura 3 apresenta-se o desenho estilizado do PEGACD composto de um manipulador de CD, denominado PEGACD, e de uma torre de CD com um manipulador, denominado POSIÇÃO. Na torre de CD, desenhada à direita na figura 3, existem locações apropriadas para o armazenamento dos CD, inclusive o topo da torre.

O robô POSIÇÃO possui somente um grau de liberdade, que pode ser representado por um operador ou por um motor elétrico que gira a torre de CD e um detector de posição. O movimento do robô POSIÇÃO deve ser sincronizado com os do robô PEGACD.

O PEGACD possui uma garra que permite segurar o CD, um mini elevador que é utilizado para movimentar para cima e para baixo a garra do PEGACD, e um trilho para aproximação do elevador à torre dos CD. A garra do PEGACD é movimentada por um motor elétrico de passo (MOTOR1) que pode estar no estado ativo (segurar o CD) ou inativo (soltar o CD). O elevador do PEGACD é movido por um motor elétrico de passo (MOTOR2), ele se encarrega de posicionar a garra na altura da locação do CD na torre. O elevador do PEGACD é montado sobre uma plataforma que se move sobre trilhos. A plataforma é movimentada por um motor elétrico de passo (MOTOR3).

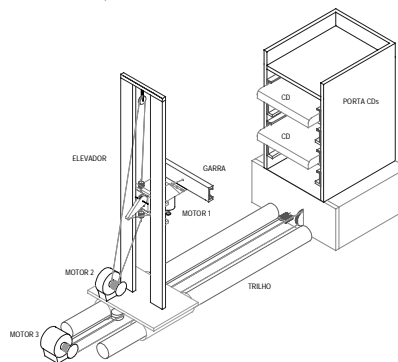


Figura 3. O PEGACD

4. O SISTEMA DE CONTROLE INTELIGENTE

Para o MOVECD foram desenvolvidas operações gerais de movimentos dos CD. Por exemplo, retirar um CD da sua locação na torre e colocá-lo sobre o topo da torre (TORRE-TOPO), retirar um CD do topo da torre e colocá-lo sobre uma locação na torre (TOPOTORRE). Para ilustrar a operação básica do MOVECD escolheu-se a tarefa de retirar um CD da sua locação na torre e a seguir, colocá-lo sobre a torre (TORRETOPO). Essa operação pode ser feita de três maneiras: 1) sem utilizar o robô POSIÇÃO; 2) o usuário simulando o robô POSIÇÃO; 3) utilizando o robô POSIÇÃO.

Para o SCI inicialmente desenvolveu-se um executivo em tempo real semelhante ao apresentado em Cavalcanti & Alsina (1997), foram implementadas diversas tarefas para se-

rem executadas em tempo real durante a fase de operação do PEGACD. Algumas tarefas são descritas na Tabela 1.

Tabela 1. Tarefas desenvolvidas para o MOVECD

Nome	Descrição da Tarefa
MOTOR1(S)	Pega (S=1) ou Solta (S=0) o CD
MOTOR2(S,n)	Sobe (S=1) ou Desce (S=0) a garra n cm
MOTOR3(S,n)	Aproxima (S=1) ou Recua (S=0) o elevador
ESPERA(n)	Espera n segundos pela acomodação do CD.

5. O MOVECD SEM O ROBÔ POSIÇÃO

Nessa estratégia de movimento considera-se que não existe o robô POSIÇÃO e a torre de CD permaneceu fixa durante o experimento em tempo real. Inicialmente foram feitos vários testes experimentais, em malha aberta, com o MOVECD. A partir desses testes definiu-se a seqüência de acionamento dos motores do MOVECD apresentadas no fluxograma da figura 4.

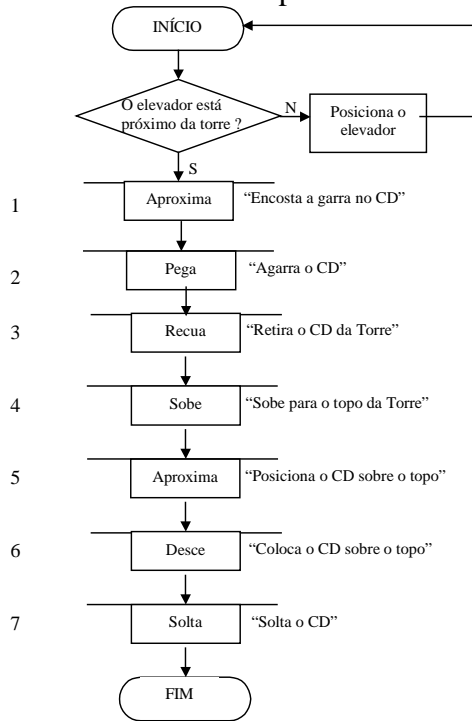


Figura 4. Operação TORRETOPO

Mesmo utilizando motores de passo nas juntas do robô PEGACD o índice de acerto foi baixíssimo, com um valor abaixo de 10%. A falha no posicionamento deveu-se a construção da garra, folgas nos elos dos motores, etc. A tela de trabalho do MOVECD, apresentada na figura 5, foi obtida durante o posicionamento do CD na operação TORRETOPO. A seqüência de ativação dos motores está apresentada no fluxograma da figura 4. No início da operação a garra deve ser posicionada a 1 cm da torre. Estando posicionada, confirmada pela pergunta “O elevador está próximo da torre?”, é iniciada a operação TORRETOPO descrita no fluxograma da figura 4. Cada etapa da figura 4 está representada na figura 5 numeradas de 1 a 7.

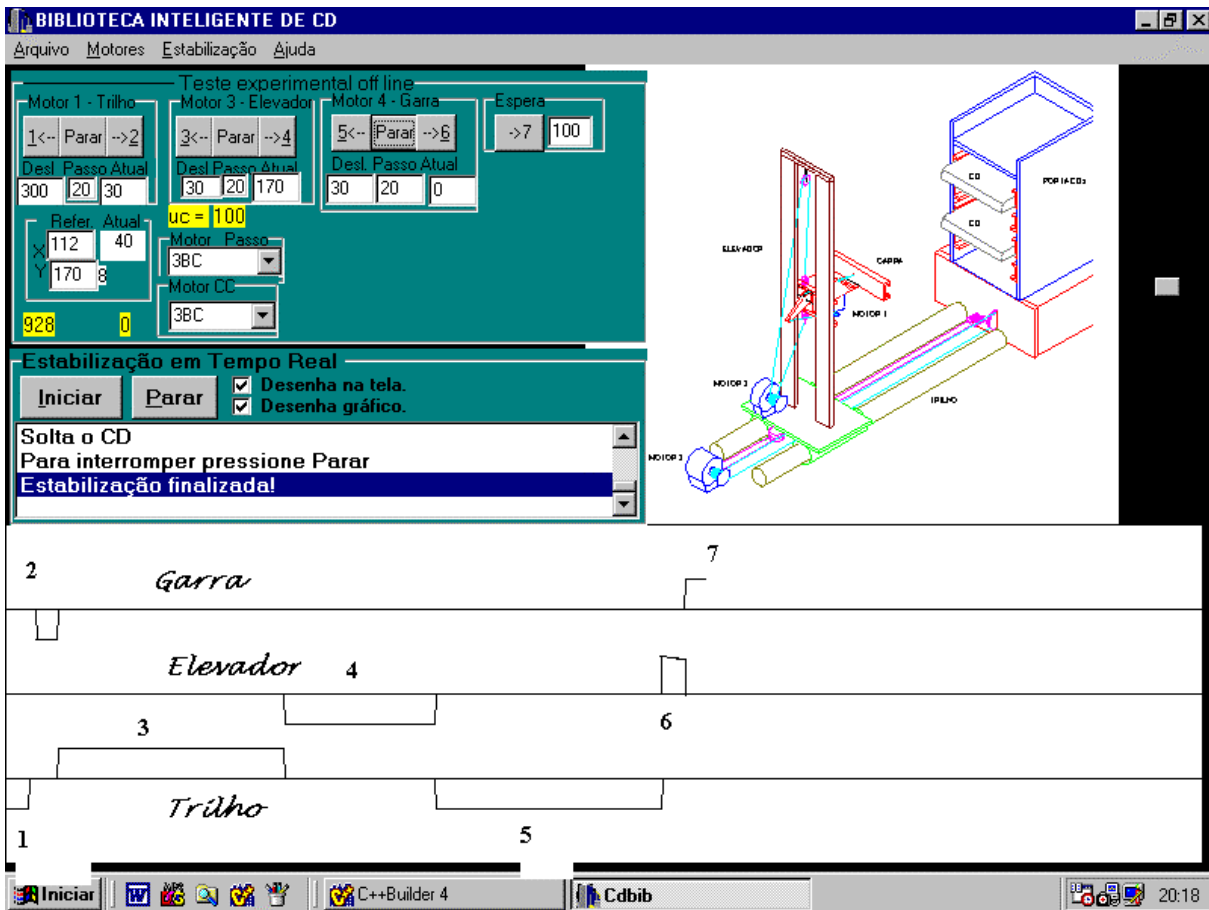


Figura 5. Tela principal obtida na operação TORRETOPO com torre fixa

6. O MOVECD COM O USUÁRIO SIMULANDO O ROBÔ POSIÇÃO

A estratégia de manipulação dos CD que utiliza a mão do usuário exige a cooperação “offline” entre o PEGACD e o operador. Essa característica de operação exigiu o desenvolvimento de um Sistema Controle Inteligente Híbrido (SCIH) que opera em tempo real e “off line”. O fluxograma da figura 6 ilustra o funcionamento do SCIH. A comunicação entre o SCIH e o usuário (simulando o robô POSIÇÃO) é feita por meio de perguntas e respostas, enquanto que as operações do PEGACD são executadas em tempo real por meio de tarefas.

No início o PEGACD está em repouso, o elevador se encontra próximo (1cm) da torre, a garra está na altura da locação fonte do CD (bloco Início). A seguir, o SCIH pergunta ao usuário se o PEGACD está convenientemente posicionado: se não estiver é ativada a operação de posicionamento do elevador; se estiver, são ativadas as tarefas e a seqüência de movimentos do PEGACD. A tarefa Aproxima é representada pela tarefa MOTOR3(1,1cm) que movimenta o elevador sobre o trilho para a aproximação. A seguir, para esperar o posicionamento correto (equilíbrio dinâmico) da torre -posicionamento feito pelo usuário-, é acionada a tarefa Espera (1s). Estando o elevador corretamente posicionado e a garra encostada ao CD, é acionada a tarefa Pega, que é representada pela tarefa MOTOR1(1). A seguir, mudando de operação em tempo real para operação “offline”, o SCIH pergunta ao usuário se o robô PEGACD agarrou corretamente o CD: se a resposta for não, será ativada a operação para o posicionamento e manipulação correta do CD; se a resposta for sim, o SCIH ativará uma seqüência de tarefas em tempo real para transportar o CD ao topo da torre (seqüência semelhante a já anali-

sada). Os resultados experimentais obtidos nesta estratégia de movimento são semelhantes aos resultados experimentais obtidos na estratégia sem o robô POSIÇÃO cujos resultados estão apresentados na figura 5 (exceto a tarefa Espera). O índice de acerto foi baixo, com um valor em torno de 50% no movimento TORRETOPO com esta estratégia.

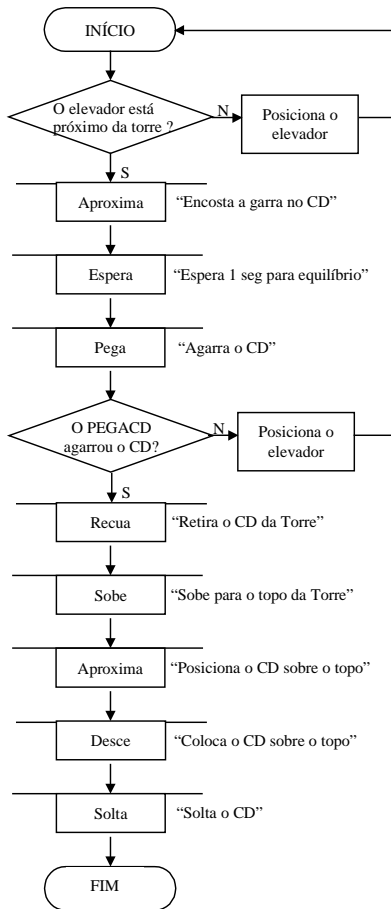


Figura 6. Operação geral TORRETOPO

7. O MOVECD COM OS DOIS ROBÔS

O PEGACD deve retirar um CD da sua locação na torre e a seguir, colocá-lo sobre a torre, e vice-versa. Essas operações devem ser feitas em tempo real e de uma forma cooperante pelos dois manipuladores (PEGACD e POSIÇÃO). Essa característica de operação utiliza um Sistema Controle Inteligente (SCI) que opera totalmente em tempo real. A seqüência de execução e interação entre os dois robôs é semelhante à apresentada no fluxograma da Figura 6 (trocando-se o usuário pelo robô POSIÇÃO). Neste trabalho não foram apresentados os resultados dos testes experimentais da operação conjunta dos dois robôs manipuladores.

8. CONCLUSÃO

Apresentou-se o projeto, implementação e resultados experimentais obtidos de um protótipo de robô a ser aplicado numa biblioteca de CD. Apresentaram-se duas novas abordagens de robôs cooperantes, uma que utiliza a interação com o operador e a outra composta de dois robôs que interagem para colocar/retirar CD de uma biblioteca de CD. Apresentou-se o projeto e implementação do protótipo do robô, descrevendo-se o funcionamento do sistema inteligente de controle do MOVECD. Devido a garra que sustenta o CD as operações de colo-

car/retirar CD da sua locação na torre nem sempre foi feito com sucesso com as duas estratégias apresentadas neste trabalho.

Atualmente está sendo testada a estratégia de movimento do MOVECD com os dois robôs manipuladores. No futuro será utilizada uma câmara digital para a visualização dos movimentos do PEGACD. Além disso, está sendo desenvolvida uma nova garra que possibilitará o funcionamento, sem erro de posicionamento, do PEGACD.

9. AGRADECIMENTOS

Agradecemos à FINEP pelo financiamento em parte desta pesquisa. Subprojeto “Construção de um robô para controle de uma biblioteca de robôs”, FINEP/FNDCT/PROGRAMA/BID/CCT/UFPB, referência 1798/98.

10. REFERÊNCIAS BIBLIOGRÁFICAS

- Cavalcanti, J.H.F. & Alsina, P.J., (1997) “Load Exchange Between Two Robots”, Workshop on Intelligent Robotics – WRI’97, Brasília, Pp. 80-88.
- Craig, J.J., (1986) “Introduction to Robotics – Mechanics & Control”, Addison Wesley Publishing Company, USA.
- Groover, M.P. & Weiss, M. & Nagel, R.N. & Oldrey, N.G. (1989) “Robótica”, MacGraw-Hill, São Paulo.
- Fujisawa, Y. & Kosuge, K. (1994), “Manipulator for Man-Robot Cooperation”, Fuzzy Logic Technology and Applications, Ed. Robert Marks II, pp 168-172.
- Osumi, H. & Arai, T. (1994) “Cooperative Control Between Two Position-controlled Manipulators”, Proceedings of 1994 IEEE International Conference On Robotics and Automation, Vol.2, San Diego, California, USA, pp1509-1514.

NEW ADAPTIVE METHOD OF PARAMETERS ESTIMATION FOR INDUCTION MOTOR DRIVES USING FUZZY LOGIC

Sergio Shimura

Waldir Po

**Universidade de São Paulo – Escola Politécnica da Universidade de São Paulo,
Departamento de Engenharia Eletrônica, São Paulo, SP, Brasil. E-mail:
sshimura@usp.br**

Abstract

A new approach to estimating induction motor parameters from measured quantities for speed-sensorless control is described. The reference model comprises two parts: in the first section, the rotor speed is determined. Next, the stator currents are calculated and compared to the real measures. The error is used in a Fuzzy system to correct the rotor and stator resistances. The method also includes an off-line estimation of the initial parameters and estimation of load torque and moment of inertia. Computer simulations are compared with experimental results to demonstrate the effectiveness of this method.

Keywords: Parameter Estimation , Induction Motor, Fuzzy Logic.

1. INTRODUCTION

High-performance motion control applications require exact knowledge of the electrical and mechanical motor parameters such as resistances, inductances and moment of inertia. Combined with measured voltages and currents the motor driver is able to calculate the rotor and stator fluxes, estimate rotor speed and adjust the control parameters so the optimum performance is maintained. Rotor and stator resistance are very sensitive to temperature variations and errors in these resistances highly affect speed and fluxes estimation in open loop (Kanmachi & Takahashi, 1995) and closed loop approaches (Tamai et al., 1987; Schauder, 1989). Different implementations and techniques have been proposed to obtain a more robust system like MRAS (Peng; Fukao 1994), and EKF (Wade et al., 1997). The main inconvenience of these systems is the need of high speed A/D converters due to high frequency injection in torque control and PWM operation.

Fuzzy Logic and Neural Networks were recently introduced in speed estimation (Krzeminski, 1995; Ben-Brahim, 1995) but these methods still presents high estimation errors at low speeds. To overcome this problem, this paper presents a new adaptive method of electrical and mechanical parameter estimation with R_s (stator resistance) and R_r (rotor resistance) correction by a Fuzzy Logic system. By using two equations, one little sensitive to rotor resistance variations and the other little sensitive to stator resistance variations, the estimator is robust and it is not restricted to PWM operation or high frequency injection in torque control as other methods are.

Experimental tests are performed in a 1/3 hp induction motor drive to validate the models.

2. PROPOSED METHOD

The proposed method has a stator current estimator and a Fuzzy Logic corrector in the adaptive mechanism. The current estimator has two parts: the first estimates the rotor speed and the second, the stator currents. The difference between estimated and measured currents is used to correct the stator and rotor resistances for the adjustable model.

2.1 Proposed scheme

Two equations are used by the rotor speed estimator: one is derived from the steady state equations and the other derived from the dynamic equations of the motor model. The equations are obtained so that the steady state equation output is not affected by R_s variations and the transient equation output is not affected by R_r variations. These characteristics are used by the system to determine the correction of the resistance parameters.

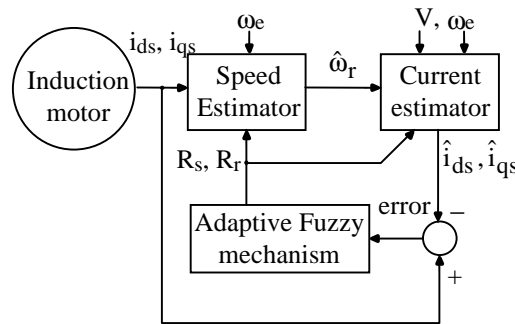


Figure 2.1: Proposed adaptive estimator

2.2 Stator current estimator

The following equations of the induction motor dynamic model in synchronous reference frame is used to estimate the stator currents, using as inputs, the motor parameters, synchronous and estimated rotor speeds, and stator voltages.

$$\lambda_{ds} = L_s \cdot i_{ds} + L_m \cdot i_{dr} \quad (2.1)$$

$$\lambda_{qs} = L_s \cdot i_{qs} + L_m \cdot i_{qr} \quad (2.2)$$

$$\lambda_{dr} = L_m \cdot i_{ds} + L_r \cdot i_{dr} \quad (2.3)$$

$$\lambda_{qr} = L_m \cdot i_{qs} + L_r \cdot i_{qr} \quad (2.4)$$

$$\lambda_{ds} = 1/p \cdot [v_{ds} - R_s \cdot i_{ds} + \omega_e \cdot \lambda_{qs}] \quad (2.5)$$

$$\lambda_{dr} = 1/p \cdot [v_{dr} - R_r \cdot i_{dr} + (\omega_e - \omega_r) \cdot \lambda_{qr}] \quad (2.6)$$

$$\lambda_{qs} = 1/p \cdot [v_{qs} - R_s \cdot i_{qs} - \omega_e \cdot \lambda_{ds}] \quad (2.7)$$

$$\lambda_{qr} = 1/p \cdot [v_{qr} - R_r \cdot i_{qr} - (\omega_e - \omega_r) \cdot \lambda_{dr}] \quad (2.8)$$

$$C_e = 3/2 \cdot P \cdot L_m \cdot (i_{qs} \cdot i_{dr} - i_{ds} \cdot i_{qr}) \quad (2.9)$$

$$\sigma = 1 - L_m^2 / (L_r \cdot L_s) \quad (2.10)$$

where: $\lambda_{ds}, \lambda_{qs}$ = d- and q-axis stator fluxes
 $\lambda_{dr}, \lambda_{qr}$ = d- and q-axis rotor fluxes
 i_{ds}, i_{qs} = d- and q-axis stator currents
 i_{dr}, i_{qr} = d- and q-axis rotor currents
 v_{ds}, v_{qs} = d- and q-axis stator voltages
 v_{dr}, v_{qr} = d- and q-axis rotor voltages
 σ = leakage coefficient

L_s, L_m, L_r = stator, magnetizing and rotor inductances
 R_r, R_s = rotor and stator resistances
 P = number of poles
 p = differential operator
 ω_e, ω_r = synchronous and rotor speed
 T_e = electromagnetic torque

2.3 Rotor speed estimator

2.3.1 Steady state equation

By eliminating i_{qr} from (2.4) and (2.8) and assuming $v_{qr} = 0$ for squirrel-cage induction motor:

$$(\omega_e - \omega_r) = \frac{1}{\lambda_{dr}} \cdot \left[-p\lambda_{qr} - R_r \cdot \frac{\lambda_{qr}}{L_r} + R_r \cdot \frac{L_m}{L_r} \cdot i_{qs} \right] \quad (2.11)$$

In steady state, $p\lambda_{qr} \approx 0$ and $|\lambda_{qr}| \ll |L_m \cdot i_{qs}|$ can be assumed. Then (2.11) can be rewritten as:

$$(\omega_e - \omega_r) = \frac{R_r \cdot L_m}{L_r} \cdot \frac{i_{qs}}{\lambda_{dr}} \quad (2.12)$$

Solving for λ_{dr} from (2.1) and (2.3) and substituting in (2.12):

$$\omega_r = \omega_e - \frac{K_1 \cdot i_{qs}}{(K_2 - i_{ds})} \quad (2.13)$$

where K_1 e K_2 are fixed values and given by $\frac{L_m^2 \cdot R_r}{L_r^2 \cdot L_s \cdot \sigma}$ and $\frac{1}{L_s \cdot \sigma} \cdot \lambda_{ds}$, respectively.

From (2.7), assuming $p\lambda_{qs} \approx 0$ for the steady state condition and $R_s \cdot i_{qs} \ll v_{qs}$ (negligible ohmic losses), then the flux λ_{ds} is calculated as:

$$\lambda_{ds} = v_{qs} / \omega_e \quad (2.14)$$

The equation (2.13) estimates rotor speed directly from the measured currents and integrators are not required, thus making it easy to implement.

2.3.2 Transient equation

Solving (2.6) and (2.8) for ω_r and eliminating R_r , the transient equation is obtained (Kanmachi & Takahashi, 1995):

$$\omega_r = \omega_e - \left(\frac{p\lambda_{dr} \cdot i_{qr} - p\lambda_{qr} \cdot i_{dr}}{\lambda_{dr} \cdot i_{dr} + \lambda_{qr} \cdot i_{qr}} \right) \quad (2.15)$$

The rotor fluxes (λ_{dr} and λ_{qr}) and rotor currents (i_{dr} and i_{qr}) can be calculated from measured voltages and currents.

2.4 Fuzzy estimator of R_s and R_r

The Fuzzy estimator uses the stator current error to determine the correction ΔR_s or ΔR_r added to the parameters R_s and R_r , respectively. ΔR_s is calculated in the steady state condition and ΔR_r , in the transient condition. The feedback mechanism is used to minimize the stator current error.

Each input and output variable has 3 Fuzzy sets: positive (P), negative (N) and zero (Z).

2.4.1 R_s estimator

Four input variables (e_1 , e_2 , Δe_1 and Δe_2) and one output variable (ΔR_s) are defined for the R_s estimator. Figure 2.2 shows the membership functions of these Fuzzy variables,

defined as follows: e_1 and Δe_1 are the i_{ds} current error and its derivative, respectively; e_2 and Δe_2 are the i_{qs} current error and its derivative, respectively; and ΔR_s is the correction added to the variable R_s .

Tables 2.1 and 2.2 summarizes the rule sets used by the Fuzzy estimator.

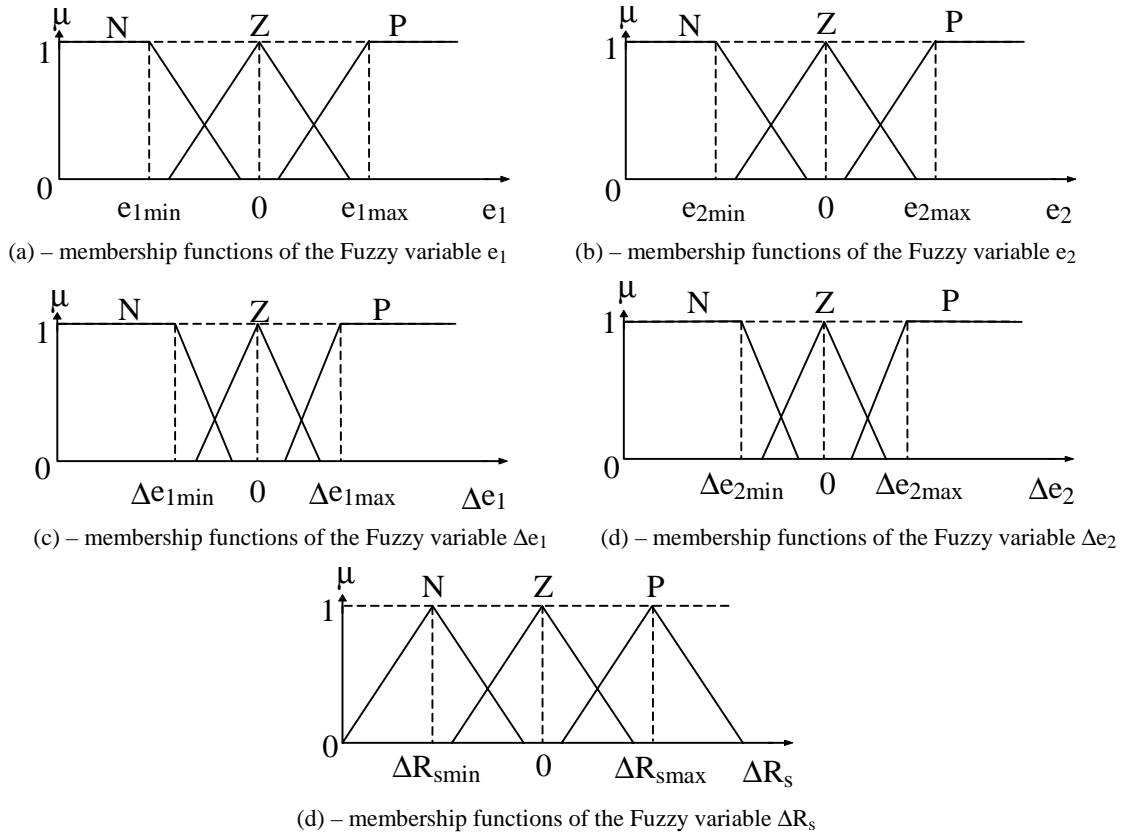


Figure 2.2: membership functions of the Fuzzy variables e_1 , e_2 , Δe_1 , Δe_2 and ΔR_s

Table 2.1: Rule set table for the inputs e_1 and e_2

$e_2 \backslash e_1$	N	Z	P
N	–	–	P
Z	–	Z	–
P	N	–	–

Table 2.2: Rule set table for the inputs Δe_1 and Δe_2

$\Delta e_2 \backslash \Delta e_1$	N	Z	P
N	–	–	P
Z	–	Z	–
P	N	–	–

2.4.2 R_r estimator

Table 2.3: Rule set table for the inputs ITAE and Δ ITAE

Δ ITAE \ ITAE	N	Z	P
N	–	–	P
Z	–	Z	–
P	N	–	–

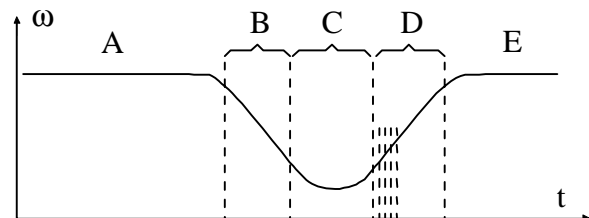


Figure 2.4: Moment of inertia estimation

The process to estimate de R_r is similar to the one used to estimate R_s . The difference is that this system uses as input variables an error function ITAE and its derivative, $\Delta ITAE$. Figure 2.3 shows the correspondent membership functions and table 2.3, the rule sets.

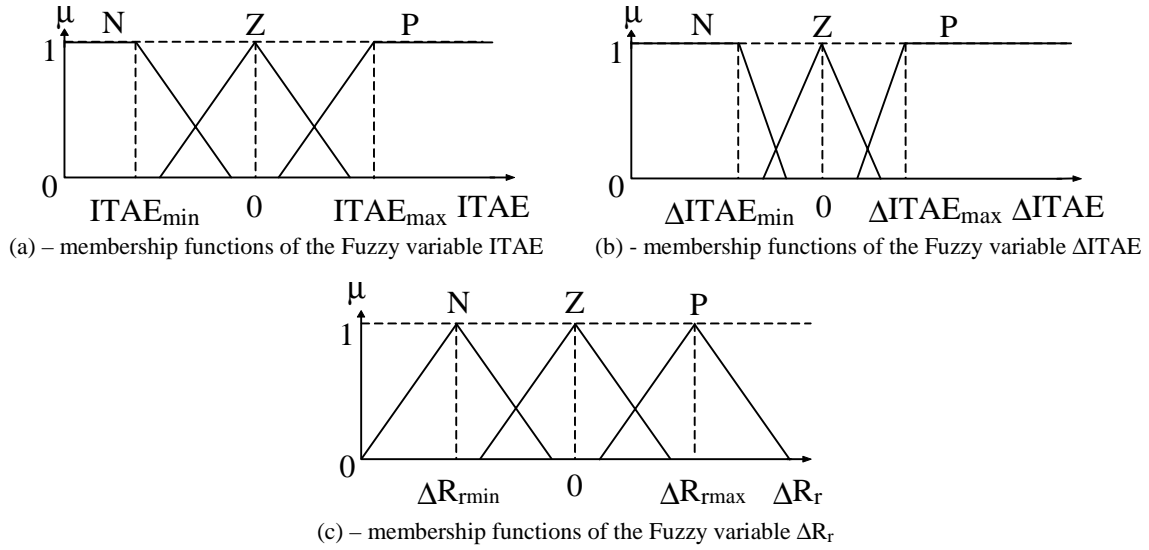


Figure 2.3: membership functions of the Fuzzy variables ITAE, $\Delta ITAE$ and ΔR_r

2.5 Estimation of moment of inertia and load torque

From the general electromechanical equation at time (k):

$$T_e(k) = I \cdot \dot{\omega}(k) + T_l(k) \quad (2.16)$$

where T_e is the electromagnetic torque, $\dot{\omega}$ is the angular acceleration, I is the moment of inertia and T_l , the load torque.

And at time (k+1):

$$T_e(k+1) = I \cdot \dot{\omega}(k+1) + T_l(k+1) \quad (2.17)$$

Subtracting (2.16) from (2.17) and assuming that there is no variation in load torque, the following equation is obtained for the moment of inertia I :

$$I = \frac{T_e(k+1) - T_e(k)}{\dot{\omega}(k+1) - \dot{\omega}(k)} \quad (2.18)$$

In the proposed method, a variation in produced torque is necessary in order to calculate the result of equation (2.18). In interval time B, $T_e(k) = 0$ and $\dot{\omega}(k)$ is measured. Electromagnetic torque is reapplied after B (C, D and E). $T_e(k+1)$ and $\dot{\omega}(k+1)$ are measured in interval D.

Once the moment of inertia is known, the load torque can be calculated using (2.19):

$$T_l = T_e(k) - I \cdot \dot{\omega}(k) \quad (2.19)$$

To attenuate the effects of torsional vibrations and make the acceleration signal usable, it must be heavily filtered. The electromagnetic torque must also be filtered so the same time delay is created. The amount of filtering is not crucial as long as they are in phase with each other and adequately filtered from the effects of torsional vibrations.

2.6 Initial parameters calculation

The initial parameters are calculated using the usual free run, rotor locked and direct current tests. The free run and rotor locked tests gives the stator currents i_{ds0} and i_{qs0} and i_{dsL} and i_{qsL} , respectively. The electric parameters of the dynamic model are calculated by (2.20), (2.21) and (2.22), assuming a known relation between X_s and X_r . The stator resistance is obtained in the direct current test.

$$R_r = \frac{v_{qs} \cdot i_{qs}}{(i_{ds}^2 + i_{qs}^2)} - R_s \quad (2.20)$$

$$X_m = \frac{v_{qs0}^2}{v_{qs0} \cdot i_{ds0} - (i_{ds0}^2 + i_{qs0}^2) \cdot X_s} \cdot \frac{1}{1 + \frac{X_s}{X_m}} \quad (2.21)$$

$$X_s = \frac{v_{qsL} \cdot i_{dsL}}{(i_{dsL}^2 + i_{qsL}^2) \cdot \left(1 + \frac{X_s}{X_r} + \frac{X_s}{X_m}\right)} \cdot \left(\frac{X_s}{X_r} + \frac{X_s}{X_m}\right) \quad (2.22)$$

3. SIMULATION RESULTS

3.2 Response to R_s and R_r variation

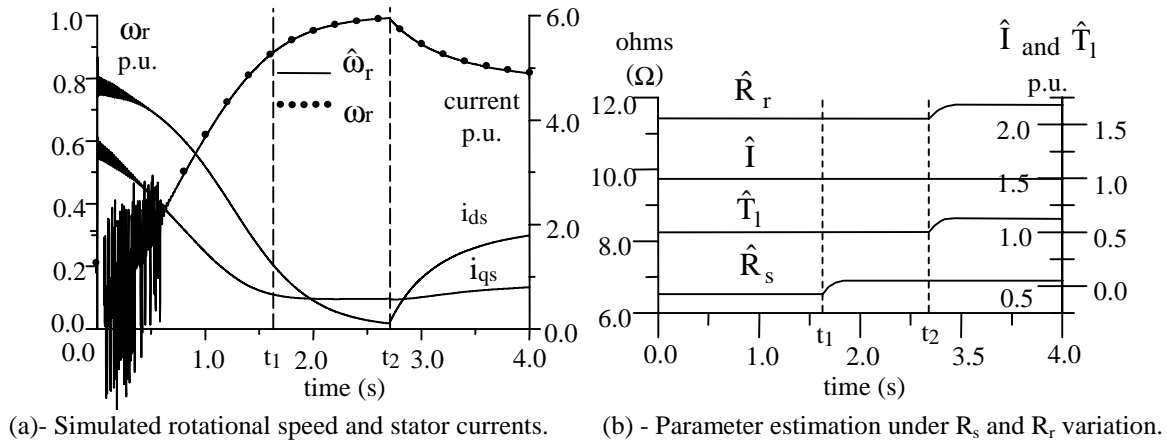


Figure 3.2: System response to R_s and R_r variation.

Figure 3.2 shows the estimation results of +20% variation in both resistances simultaneously. The test conditions were $v_s = 0.22$ p.u., $\omega_s = 1$ p.u. and torque step of $T_1 = 0.37$ p.u. at $t_2 = 2.7$ s. The stator resistance, is updated during interval time t_1 to t_2 , when equation (2.13) is used; and rotor resistance is updated after time t_2 when (2.15) is used.

Under these simulated conditions, the estimated speed error is less than 1% in the steady state condition and less than 0.3% in the transient condition.

3.3 Estimation of moment of inertia

Figure 3.3 shows how the estimated parameters behave under the estimation of moment of inertia. At $t_1 = 1.5$ s, the load torque of 0.37 p.u. magnitude is applied. The produced

electromagnetic torque is removed at $t_2 = 2.7\text{s}$ and reapplied at $t_3 = 3.0\text{s}$. During the interval time between t_2 and t_3 , $\hat{\omega}(k)$ is sampled and after t_3 , $\hat{\omega}(k+1)$ is sampled.

The transient equation is used in the interval time 0s to t_1 . Between t_2 and t_3 , the error in the speed estimation is higher than the error observed in other intervals because the stators currents are close to zero. If this interval is not considered, the maximum error in speed estimation is less than 0.10% during steady state and less than 0.01% during transient.

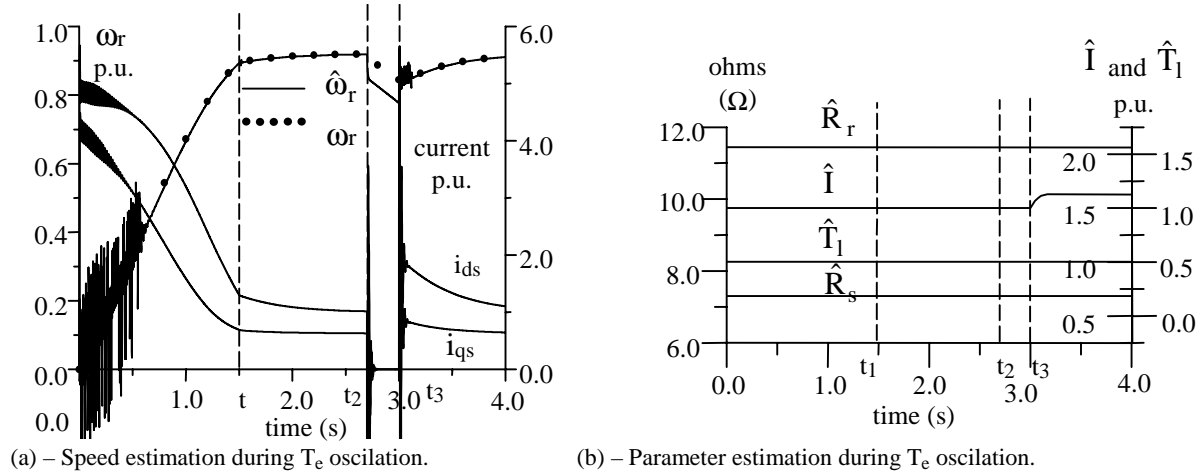


Figure 3.3: estimation of moment of inertia.

4. EXPERIMENTAL RESULTS

Table 4.1: Electrical and mechanical parameters of the induction motor - 220V – 1/3 hp

$R_s = 6.500 \Omega$	$L_m = 546.184547 \text{ mH}$	Nominal current = 0.376 A
$R_r = 9.137 \Omega$	$Z_b = 584.15 \Omega$	Number of poles = 2
$L_s = 556.400662 \text{ mH}$	$L_{ls} = 10.21612 \text{ mH}$	Nominal speed = 3500 rpm
$L_r = 569.942954 \text{ mH}$	$L_{lr} = 23.75841 \text{ mH}$	I (motor and load) = $7.72 \times 10^{-4} \text{ kg.m}^2$

4.1 Simulation and experimental results comparison

Figure 4.1 shows both the real and estimated speed, for comparison, along with the estimated produced torque \hat{T}_e and the real load torque T_1 . The data were obtained under free run. The actual voltages, currents and torque were measured in the test bench.

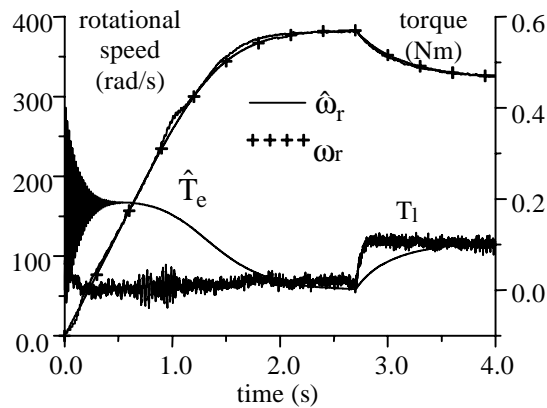


Figure 4.1: Measured and estimated rotational speed, and produced torque \hat{T}_e and load torque T_1 .

4.2 Response of the new speed estimator

The new speed estimator for steady state (2.13) was tested in free-run and in variable speed (figure 4.7).

The results of the proposed estimator are close to the real measures for speeds above 0.6 p.u. as shown on figure 4.6. The measured speed errors are 0.3% during no-load condition and 0.9% after the load torque is applied.

In the variable speed test, shown on figure 4.7, the estimated speed was also close to the real measures: the speed error was between 0.3%, with $\omega_e = 405$ rad/s, and 2.0%, with $\omega_e = 257$ rad/s.

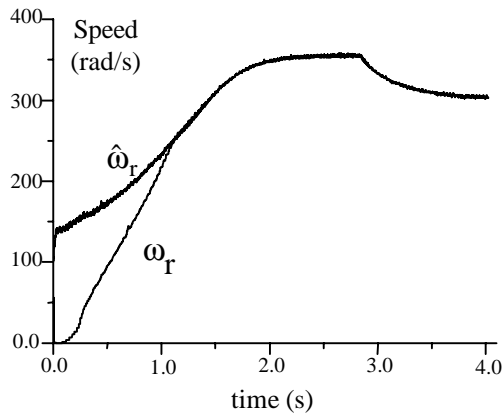


Figure 4.6: Actual and estimated rotational speed

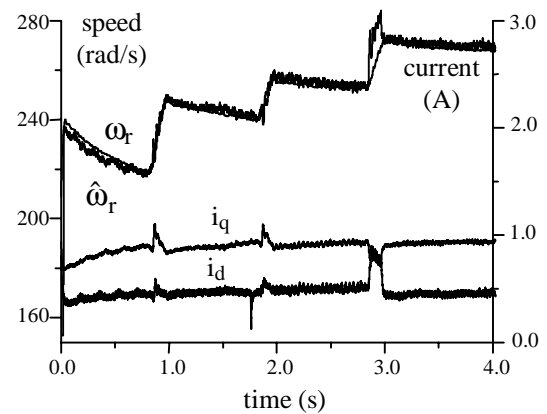


Figure 4.7: Actual and estimated speeds for various ω_e

5. CONCLUSIONS AND FINAL ANALYSIS

A new parameter estimation method using Fuzzy Logic correction of R_s and R_r was presented. This work also includes the initial parameter determination and estimation of speed, moment of inertia and load torque. Using direct current, free run and rotor lock tests, the initial parameters can be calculated. The estimation and correction of parameters is possible using simple algorithms.

The new system was evaluated through computer simulations and validated by experimental results. In the computer simulations the maximum observed error in the steady state condition was 0.3% and in the transient condition was 0.2% for 50% variation in both R_r and R_s . The experimental results using a 1/3 hp three-phase induction motor showed error between 0.3% with no load and $\omega_e = 405$ rad/s, and 2.0% with load and $\omega_e = 257$ rad/s.

This new method can be simplified for using in low cost induction motor drivers through the implementation of the steady state equation (2.13) alone, which proved to give good results over a wide range of speeds.

BIBLIOGRAPHY

- Ben-Brahim, L., 1995, "Motor speed identification via neural networks". IEEE Industry Applications Society Magazine, pp. 28-32.
- Kanmachi, T.; Takahashi, I., 1995, "Sensor-less speed control of an induction motor". IEEE Industry Applications Magazine, pp. 22-27.
- Krzeminski, Z., 1995, "Estimation of induction motor speed in fuzzy logic system". EPE '95. European Conference on Power Electronics and Applications, 6., Sevilha. S.I., EPFL, v.3, pp. 563-568.

- Peng, F.; Fukao, T., 1994, "Robust speed identification for speed-sensorless vector control of induction motors". IEEE Transactions on Industry Applications, v.30, n.5, pp. 1234-1240.
- Schauder, C., 1989, "Adaptive speed identification for vector control of induction motors without rotational transducers". Proceedings IEEE/IAS Annual Meeting, 24. San Diego, S.I., IEEE, pp. 493-499.
- Tamai, S.; Sugimoto, H.; Yano, M., 1987, "Speed sensorless vector control of induction motor with model reference adaptive system". IEEE/IAS Annual Meeting, 22. Atlanta. S.I., IEEE, pp. 189-195.
- Wade, S.; Dunnigan, M. W.; Williams, B. W., 1997, "Improving the accuracy of the rotor resistance estimate for vector-controlled induction machines". IEE Proceedings - Electric Power Applications. v.144, n.5, pp. 285-294.

AVALIAÇÃO DE PARÂMETROS OPERACIONAIS DO CICLO TÉRMICO DE UMA PLANTA FRIGORÍFICA DE GRANDE PORTE

Orozimbo A. A Rego (Universidade Federal de Uberlândia)
Cassius Ricardo Nasc. Ferreira (aluno mestrado UFU)

Resumo

No presente trabalho é analisado o comportamento do COP de uma planta frigorífica de grande porte em função de parâmetros operacionais da instalação. A avaliação é feita através da simulação numérica do ciclo térmico, que opera com amônia como fluido frigorífico, realiza a compressão em estágios com resfriamento intermediário e subresfriamento e produz frio em três níveis de temperatura de evaporação. São apresentados e analisados resultados que avaliam os processos de compressão e de condensação e da distribuição da carga térmica em diferentes temperaturas.

Palavras-chave: Refrigeração, Simulação Numérica.

1 . INTRODUÇÃO

No presente trabalho são apresentados resultado para o COP, Coeficiente de Performance, da planta frigorífica de uma industria de carnes de grande porte instalada no município de Uberlândia - MG .

O estudo do COP , obtido através de um modelo que simula a planta , tem por finalidade estabelecer parâmetros de operação para as propriedades termodinâmicas do ciclo frigorifico que opera com amônia .

O modelo permite também estabelecer estratégias de operação da fabrica , que produz embutidos , defumados , alimentos “in natura” e outros produtos , em diversos níveis de produção e temperaturas , visando diminuir os custos para a produção do frio necessário , através de valores adequados para o COP da instalação .

A planta frigorifica foi concebida originalmente para atender uma carga térmica de aproximadamente 4000 KW para uma fabrica que fazia abate de bovinos e suínos para industrialização e de comercialização das carnes mais nobres. Devido a alteração no mercado a fabrica tem abatido apenas suínos , não processando carnes “in natura”, fazendo com que a demanda de frio diminua , apesar das instalações atualmente também estarem sendo utilizadas para o congelamento de carcaças de frango . Essa demanda corresponde atualmente a em torno de 60% da capacidade de produção de frio instalada.

2 . DESCRIÇÃO DA PLANTA ESTUDADA

A Figura 1 apresenta de maneira esquemática a planta frigorifica que opera com amônia num ciclo de compressão em estágios com resfriamento intermediário e subresfriamento e com três níveis de temperatura de evaporação nos separadores de liquido . A condensação é feita num único nível de pressão através de uma bateria de condensadores evaporativos.

Os separadores de líquido são montados cada um com dois vasos em paralelo , que permite a operação e a manutenção adequada dos mesmos .

No separador de líquido SL1 , a amônia opera a uma temperatura que oscila em torno dos -45°C e é bombeada para atender a demanda de frio dos túneis de choque térmico de carcaças, nos túneis de congelamento helicoidais e nos túneis contínuos longitudinais, que operam com temperaturas internas em torno dos -35°C .

O segundo separador SL2 , opera a -35°C aproximadamente e fornece amônia a esta temperatura para as câmaras de estocagem de produtos , essas câmaras , em número de 100 aproximadamente na fabrica , operam em diversos níveis de temperatura variando de -30°C até 5°C .

O separador de líquido SL3 opera a -12°C e a amônia produzida é utilizada basicamente para a produção de água gelada com glicol a aproximadamente 0°C .

Essa água gelada é utilizada em “ fan - coils” , para o condicionamento do ar dos ambientes da fabrica (abate , desossa , industrialização e circulação) que operam com temperatura e umidade controladas.

Nos dois primeiros níveis de evaporação (-45°C e -35°C) , a compressão é feita em dois estágios , primeiro dos separadores de líquidos para o resfriador intermediário , e deste para o condensador e depósito de refrigerante .

No nível de evaporação de (-12°C) a compressão é feita num único estágio do separador de líquido para o condensador .

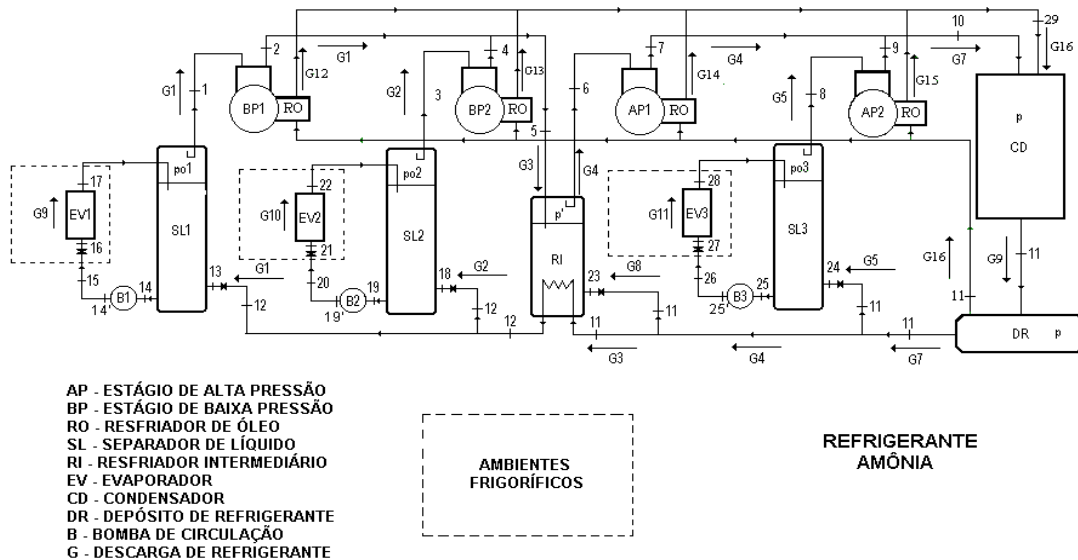


Figura 1 – Esquema da instalação

A amônia que alimenta os dois primeiros separadores de líquido sofre um processo de sub-resfriamento no resfriador intermediário, 11 para 12, o que não ocorre no terceiro separador .

Os estágios de compressão são realizados por compressores de parafuso , instalados em paralelo com a refrigeração do óleo lubrificante sendo feita por amônia a alta pressão que é depois recondensada .

A instalação esta devidamente instrumentalizada em termos industriais, para atender os aspectos de controle de operação do sistema, sendo que a carga de refrigerante nos compressores é regulada , de acordo com a demanda de frio , pelo nível de pressão nos separadores de líquido.

3. O MODELO

O modelo desenvolvido permite avaliar para o ciclo, o coeficiente de performance teórico, COP, definido por:

$$\text{COP} = (CT_1 + CT_2 + CT_3) / (PBP_1 + PBP_2 + PAP_1 + PAP_2)$$

onde:

CT_i = Carga térmica atendida pelo separador de líquido SL_i.

PBP_i = Potência teórica consumida no estágio de baixa pressão BP_i.

PAP_i = Potência teórica consumida no estágio de alta pressão AP_i.

O modelo foi desenvolvido através do balanço de energia nos diversos componentes do ciclo para regime permanente de onde foram obtidos também os valores das potencias consumidas , calores trocados nos equipamentos e fluxos de refrigerante nos diferentes pontos do ciclo .

Não estão apresentados resultados neste trabalho , mas o modelo permite também estimar a potência consumida nos processos de bombeamento de amônia com base na perda de carga das linhas e do fluxo de amônia necessário para se executar o processo de resfriamento do óleo lubrificante dos compressores.

4. RESULTADOS

Com a finalidade de se visualizar os níveis de operação do ciclo, a Tabela 1 apresenta os valores das propriedades termodinâmicas dos diversos pontos do ciclo. Os valores das pressões e temperaturas foram obtidos através de instrumentação existente na planta e as demais, entalpia, entropia, volume específico e título através de software apropriado. Como ilustração, foram estimadas as perdas de carga nas linhas de circulação de amônia.

A Tabela 2 fornece os valores das descargas de refrigerante nos diversos pontos do ciclo, obtidos considerando-se as cargas térmicas originais de projetos nos separadores de líquido, SL1, SL2 e SL3, respectivamente de 300, 300 e 400 Toneladas de Refrigeração (TR). Para esta situação, o COP teórico calculado do ciclo foi de 3,904.

Os valores das descargas de circulação de amônia nos evaporadores, EV1, EV2 e EV3, foram obtidos admitindo-se uma taxa de evaporação nos mesmos respectivamente de 10%, 12% e 5%.

Tabela 1- Propriedades termodinâmicas do ciclo da instalação

PONTO	P (BAR)	T (°C)	H (KJ/KG)	S (KJ/KG.°C)	V (M3/KG)	X
1	0,544	-45	1383	6,061	2,011	1
2	2,681	40	1553	5,93	0,5566	-
3	0,9298	-35	1399	5,874	1,219	1
4	2,681	31	1532	5,864	0,539	-
5	2,681	35,53	1542	5,897	0,5479	-
6	2,681	-12	1432	5,508	0,4516	1
7	11,71	80	1610	5,413	0,1375	-
8	2,681	-12	1432	5,508	0,4516	1
9	11,71	80	1610	5,413	0,1375	-
10	11,71	80	1610	5,413	0,1375	-
11	11,71	24	296,1	1,113	0,001655	-
12	11,71	14	247,7	0,9469	0,001616	-
13	0,544	-45	247,7	1,087	0,3864	0,1915
14	0,544	-45	-21,05	-0,09116	0,001438	0
15	2,044	-45	-20,84	-0,09116	0,001438	-
16	0,544	-45	-20,84	-0,09021	0,001747	0,000154
17	0,544	-45	119,3	0,524	0,2024	0,1
18	0,9298	-35	247,7	0,2015	0,1643	-
19	0,9298	-35	21,4	0,09066	0,001463	0
20	2,13	-35	21,57	0,09066	0,001463	-
21	0,9298	-35	21,57	0,0914	0,001618	0,000128
22	0,9298	-35	186,7	0,7847	0,1476	0,12
23	2,681	-12	296,1	1,16	0,06077	0,1316
24	2,681	-12	296,1	1,16	0,06077	0,1316
25	2,681	-12	124	0,5007	0,001528	0
26	3,781	-12	124,2	0,5007	0,001528	-
27	2,681	-12	124,2	0,5013	0,001586	0,000129
28	2,681	-12	189,4	0,7511	0,02403	0,05
29	11,71	30	1469	0,5213	0,1175	-

As descargas de amônia nos resfriadores de óleo foram estimadas através de balanço térmico e determinado o fluxo de amônia necessário para retirar o calor durante o processo de compressão atingindo-se as temperaturas de recalque medidas.

A partir deste ciclo básico e utilizando-se o modelo desenvolvido, apresenta-se a seguir uma análise da influência de alguns parâmetros de operação no COP da instalação.

Um importante fator a ser analisado são as condições de resfriamento do processo de compressão. A intensidade deste resfriamento é influenciada por condições de projeto, de operação e, fica diretamente caracterizada pela temperatura de descarga do compressor. Quanto mais baixos esses valores, mais intenso terá sido o processo de refrigeração. As Figuras 2 e 3 a seguir, apresentam respectivamente as variações do COP da instalação, em função das temperaturas de descarga dos compressores de alta e de baixa.

Tabela 2 – Vazão de amônia nos pontos da instalação de acordo com a Figura 1

NÚMERO	VAZÃO (KG/HR)
1	3342
2	3294
3	6636
4	7694
5	4453
7	12147
8	929,8
9	27058
10	22965
11	77529
12	119,6
13	9,217
14	224,1
15	129,7
16	482,6

Pode-se constatar através da Figura 2 que uma variação de 30°C na temperatura de descarga dos compressores de alta pode significar teoricamente, uma variação de 12% no COP. A variação imposta no processo de resfriamento do estágio de baixa pressão, Figura 3, como esperado, influi de maneira menos significativa naquele coeficiente.

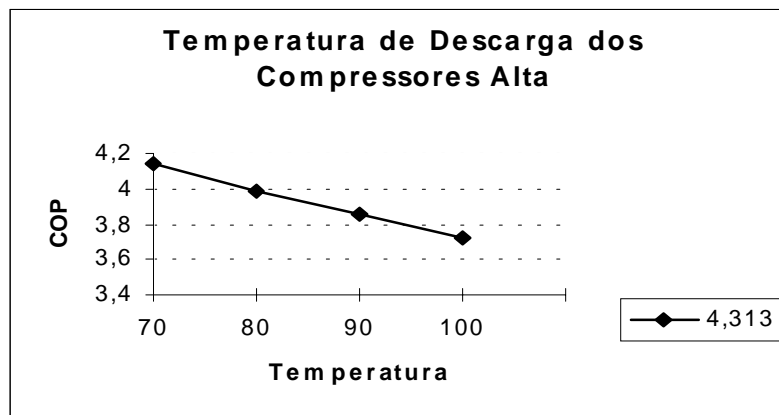


Figura 2 – Influência do resfriamento nos estágios de alta pressão AP1 e AP2.

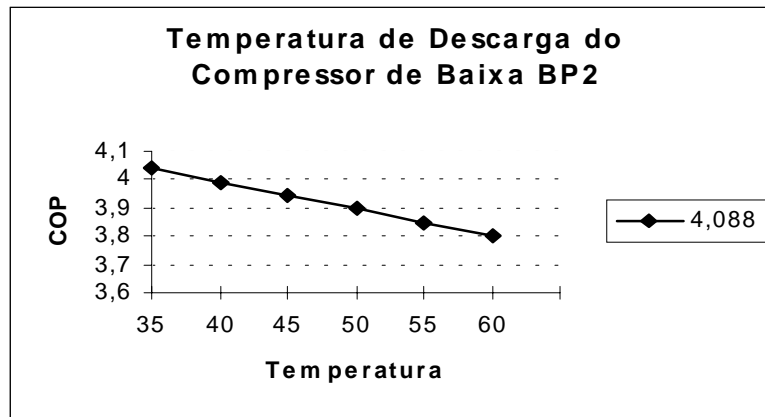


Figura 3 – Influência do resfriamento no estágio de baixa pressão BP2

As condições climáticas, temperatura e umidade relativa, além de influir de maneira bastante significativa na carga térmica da instalação, provocam alterações significativas no processo de condensação do ciclo frigorífico por operar com condensadores evaporativos.

A Figura 4 ilustra a variação no COP da instalação como função do grau de subresfriamento no condensador causado por possíveis condições de operação.

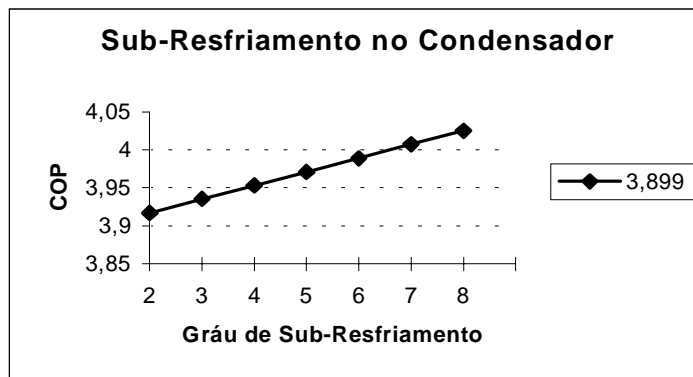


Figura 4 – Influência do subresfriamento no condensador

Já a Figura 5, mostra a variação do COP com a temperatura de condensação que pode ser conseguida através de regulação no sistema.

Pode-se constatar que, para as faixas de variação de temperaturas possíveis de serem conseguidas, as condições de condensação influem de maneira menos significativa no COP do que a maneira como é realizada o processo de resfriamento da compressão.

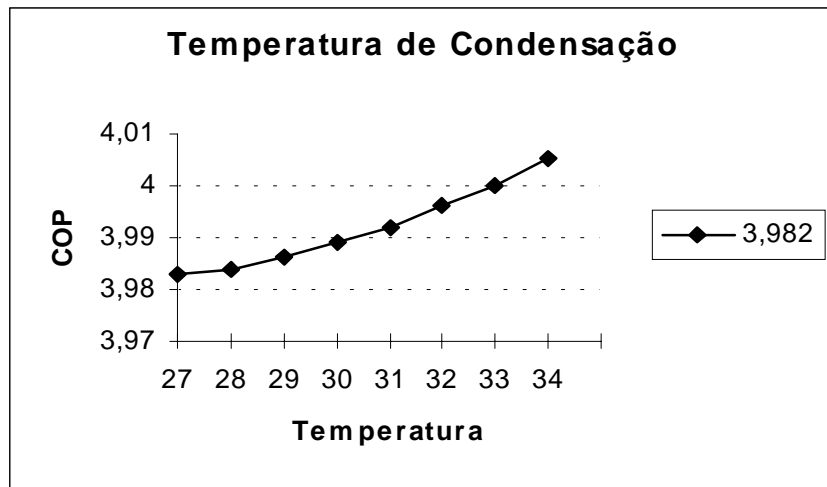


Figura 5 – Influência da temperatura de condensação

A variação das condições de subresfriamento no resfriador intermediário influem pouco no COP como pode ser constatado através da Figura 6 onde uma variação de 18°C no valor do grau de subresfriamento, implica numa variação de 2% no COP.

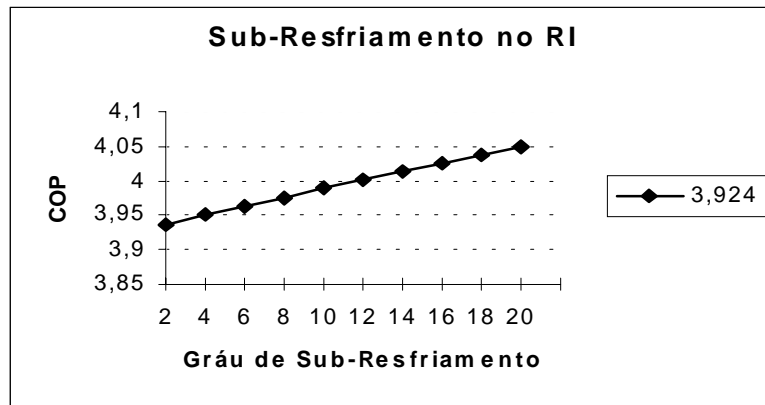


Figura 6 – Influência do subresfriamento no resfriador intermediário

Uma outra análise que pode ser feita utilizando o modelo, é da possível transferência de carga térmica de um nível de temperatura de evaporação para outro. Isto pode ser feito, por exemplo, transferindo o local de processamento de produtos quando possível.

A figura 7 mostra o ganho no COP da instalação ao transferirmos gradativamente a carga térmica do separador de líquido SL1 para o SL3 que opera com temperatura de evaporação mais alta.

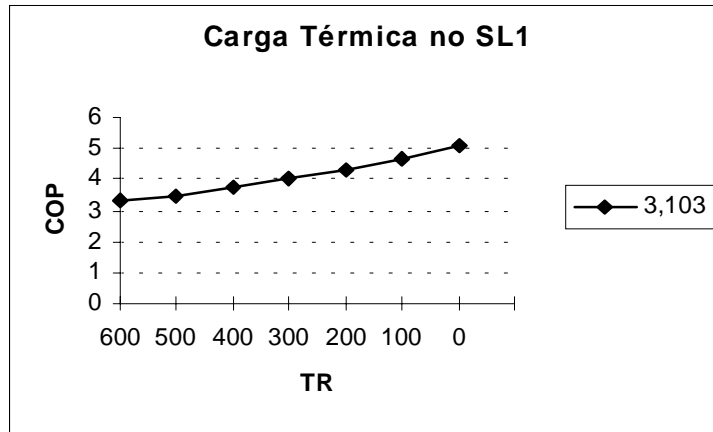


Figura 7 – Influência dos níveis de temperatura da carga térmica

Esta influência é bastante significativa ao constatarmos através da Figura 7 que o valor do COP pode variar até 30%.

5. CONCLUSÃO

Apesar da simplicidade do modelo, desenvolvido para uma situação de regime permanente, o mesmo permite avaliações preliminares que podem contribuir na tomada de decisões no que diz respeito a parâmetros térmicos de operação e de estratégia de produção da fábrica visando diminuir os custos com energia elétrica na produção do frio. Esta análise preliminar permite também constatar a importância da refrigeração no processo de compressão. O modelo está sendo aprimorado considerando o regime não regular de operação do ciclo e incluindo no mesmo outros equipamentos, sistemas e a carga térmica nos diversos pontos de consumo de frio.

6. REFERÊNCIAS

- Stoecker, W.F. e Jones, W.J.; “Refrigeração e Ar Condicionado”; Ed. McGraw-Hill, São Paulo, 1996
- Van Wylen, G.J., “Fundamentos da Termodinâmica Clássica, Ed. McGraw-Hill, São Paulo, 1996
- EES, Engineering Equation Solver, 1999.
- ASHRAE Handbook, Fundamentals Volume, “American Society of Heating, Refrigerating and Air Conditioning Engineers, Atlanta GA., 1991.
- Stoecker, W.F., “Design of Thermal Systems”, 2nd ed., McGraw-Hill, NY, 1989.
- Sonntag, R., Borgnakke, C. e Van Wylen, G., “Fundamental of Thermodynamics”, 5th ed., John Wiley & Sons, NY, 1999.

METODOLOGIA SISTEMÁTICA PARA ESPECIFICAÇÃO DE ROBÔS

Vanderley de Vasconcelos
Silvério Ferreira da Silva Júnior

Comissão Nacional de Energia Nuclear – CNEN, Centro de Desenvolvimento da Tecnologia Nuclear – CDTN, C.P. 941, 30.161-970, Belo Horizonte, MG, Brasil.

E-mail: vasconv@urano.cdtm.br

Resumo

A especificação de robôs envolve o conhecimento de inúmeras áreas, tais como mecânica, eletrônica, computação, controle, instrumentação, cinemática, dinâmica, aquisição de sinais, simulação e segurança. Tudo isto sem falar dos conhecimentos específicos das aplicações a que se destinam. A dificuldade de encontrar profissionais com estas qualificações levou ao desenvolvimento de uma metodologia sistemática para a especificação de robôs, baseada em técnicas de análise e projeto estruturado de sistemas informatizados, bem como modelagem e simulação de processos. A filosofia da metodologia é dividir o processo de especificação do robô em seis etapas: *projeto pré-conceitual*; *projeto conceitual*; *projeto lógico*; *projeto físico*; *simulação computacional*; e *definição dos parâmetros do sistema*. Neste trabalho apresenta-se a utilização desta metodologia para a especificação de um robô para operar em um sistema de ensaio automatizado idealizado para realizar a inspeção ultra-sônica das soldas de um vaso de pressão de classe nuclear.

Palavras chaves: robôs, especificação, metodologia, análise de sistemas, simulação

1. INTRODUÇÃO

A especificação de robôs é uma tarefa multidisciplinar que envolve o conhecimento de inúmeras áreas, tais como mecânica, eletrônica, computação, controle, instrumentação, cinemática, dinâmica, aquisição de sinais, simulação e segurança, além dos conhecimentos específicos das aplicações a que se destinam. A dificuldade de encontrar profissionais, ou mesmo equipes, com todas estas qualificações, levou ao desenvolvimento de uma metodologia sistemática para a especificação de robôs, baseada em técnicas de análise e projeto estruturado de sistemas informatizados, bem como modelagem e simulação de processos. Separando-se a complexa tarefa de especificação em etapas mais simples, iniciando-se pela identificação do QUE o sistema irá fazer e detalhando-se, em etapas subsequentes, COMO ele irá realizar suas tarefas, aumentam-se as chances de definição de um robô mais adequado à aplicação. A contínua interação da equipe de especificação com os clientes e usuários finais, através do uso de ferramentas gráficas, como Diagrama de Fluxo de Dados e modelagem de sólidos, além de auxiliar na elaboração de documentos conjuntos, é fundamental para a correta identificação dos conceitos de projeto, ambiente de trabalho e parâmetros técnicos do robô.

2. METODOLOGIA DESENVOLVIDA

A filosofia básica da metodologia desenvolvida é dividir o processo de especificação em seis etapas: *projeto pré-conceitual; projeto conceitual; projeto lógico; projeto físico; simulação computacional; e definição de parâmetros do sistema.*

2.1 Projeto pré-conceitual

Esta etapa deve retratar as necessidades específicas do usuário ou cliente do sistema. Deve reunir informações que suportem as decisões do grupo de especificação e permitam a definição dos dispositivos, equipamentos e *softwares* necessários para realizar as funções propostas para o sistema. É de fundamental importância que os usuários do sistema, estejam conscientizados de que eles, como interessados no sistema, devem desenvolver todos os esforços para que a pré-conceituação caracterize perfeitamente as suas reais necessidades, uma vez que ela será o elemento de suporte para as decisões do grupo de especificação.

Nesta pré-conceituação, deve ser editada uma documentação, a ser aprovada pelo usuário, contendo os seguintes itens: *objetivos; benefícios; dificuldades; escopo e abrangência; principais saídas; interligações com outros sistemas; principais arquivos; e estimativas.*

Nesta fase, é bastante útil a elaboração de um Diagrama de Contexto (Gane & Sarson, 1984), identificando as principais entradas e saídas, bem como as interfaces externas do sistema. O Diagrama de Contexto é uma primeira representação de um sistema, mostrado como um grande processo, interagindo com o ambiente em que está inserido. O ambiente de um sistema, meio externo ou em torno do sistema, é delimitado pelos elementos externos que exercem influência sobre o comportamento do sistema (operador, equipamentos, *hardware* e *software* externos, etc.), fornecendo dados ou utilizando suas informações. Neste tipo de diagrama, as entidades externas, que estabelecem uma fronteira entre o sistema e o ambiente externo, estão ligadas ao sistema através dos fluxos de dados.

2.2 Projeto conceitual

Nesta etapa, o que se pretende é produzir um documento contendo os conceitos, segundo a ótica dos usuários, isto é, contendo os detalhes necessários a uma caracterização, a mais completa possível, dos objetivos do sistema. O conteúdo deste documento, gerado através de uma contínua interação entre o grupo que fará a especificação do sistema e os usuários, passa a servir como compromisso contratual assumido entre as partes.

Os documentos consolidados e redigidos nesta etapa devem ser aprovados pelos usuários e conter os seguintes itens: *objetivos; conceitos básicos; modelo conceitual; dicionário de dados; e elos de comunicação.*

Nesta fase, os *objetivos* devem ser apresentados de acordo com uma visão funcional da área interessada, isto é, devem ser identificadas as funções que o sistema deverá realizar, procurando-se definir o **QUE** fazer e não **COMO** fazer, descrevendo, portanto, as funções e não seus detalhes de operação.

Deve ser feito o detalhamento do modelo conceitual do sistema, isto é, os Diagramas de Fluxo de Dados de primeiro nível – DFDs (Gane, 1990)(Gane & Sarson, 1984)(Mershon & Dendyna, 1997) e, em alguns casos, os DFDs de segundo nível. Os DFDs contêm funções, fluxos, depósitos de dados e entidades externas, detalhando, em níveis crescentes, o processo e os fluxos de dados identificados no Diagrama de Contexto. As funções identificadas representam as

ações que o sistema deverá executar, independente de como elas serão executadas, inclusive aquelas que serão executadas de forma manual ou interativa. Em um DFD, uma função pode ser entendida como um componente do sistema onde somente os dados de entrada e os dados de saída são conhecidos, sem ainda um conhecimento explícito a respeito do processamento interno de transformação dos dados de entrada em dados de saída (Christóvão, 1999). Nestes diagramas, os depósitos de dados se referem às estruturas de dados lógicos e não aos meios físicos para armazenamento e recuperação das informações. Os fluxos de dados são utilizados para mostrar o movimento de conjuntos de informações de um ponto a outro do sistema.

2.3 Projeto lógico

Nesta fase, a partir dos DFDs já obtidos no projeto conceitual, detalha-se, por meio de refinamentos sucessivos, um conjunto de DFDs até um nível onde as funções básicas do sistema estejam totalmente identificadas. Neste conjunto, deverá ser o mais nivelado possível, isto é, as funções deverão ser apresentadas com o mesmo nível de detalhe. Neste ponto, mantém-se a visão lógica, mas deve-se especificar integralmente o sistema.

Com base no projeto conceitual do sistema, deve ser detalhado, no dicionário de dados, o conteúdo dos arquivos e fluxos identificados no refinamento do projeto lógico. Nesta fase, se necessário, deve-se definir os dados em termos de entidades e relacionamentos entre elas, normalizar as definições de arquivos, obtendo-se em termos lógicos, a definição da base de dados requerida pelo sistema.

Devem, também, ser analisadas as diversas alternativas de automatização do sistema, estabelecendo-se as etapas que deverão ser totalmente automatizadas e aquelas que deverão ser realizadas interativamente, bem como as interfaces entre os vários sistemas, automatizados ou não. Deve-se levar em conta os recursos humanos e materiais envolvidos em cada uma das alternativas apresentadas.

O documento gerado no final desta etapa, contendo o *modelo lógico*, a *definição dos dados*, as *definições das funções* e as *definições de fronteiras com outros sistemas*, deve ser aprovado em reunião entre a equipe de especificação e o usuário.

2.4 Projeto físico

Nesta etapa, deve ser caracterizada a dependência entre as funções identificadas no projeto lógico e definidos os esquemas de controle, recuperação (em caso de falhas) e segurança. O objetivo é transformar-se o modelo lógico em um modelo físico, determinando **COMO** o sistema deverá funcionar. O foco é a eficiência do sistema, tendo por base o seu desempenho. O modelo físico idealizado deve incorporar a solução proposta nas fases anteriores, de acordo com os recursos tecnológicos disponíveis. Inclui a identificação de módulos e a sua integração, bem como uma descrição formal das partes do sistema que serão manuais e a identificação de como os usuários irão interagir com a parte automatizada do sistema. O modelo físico dos dados deve corresponder à forma com que os dados serão implementados nos dispositivos de armazenamento existentes.

A especificação da interface homem-máquina é de grande importância para satisfazer as necessidades dos usuários em relação ao sistema e a participação destes, na sua especificação, é fundamental. Entre as atividades envolvidas, destacam-se a escolha dos dispositivos de entrada e saída, o formato de todas as formas de entradas e saídas (telas, relatórios, etc.) e a apresentação da seqüência e temporização das entradas e saídas.

É também importante a definição dos critérios de *backup* e da consistência dos dados de entrada e saída do sistema. No detalhamento do dicionário de dados e dos arquivos, devem ser definidas as informações a serem obtidas, manipuladas e armazenadas pelo sistema. Deve ser realizada a identificação dos programas e equipamentos necessários para a implementação do sistema, que realizam as funções lógicas identificadas. Nesta etapa, deve ser feito o detalhamento de como executar as funções lógicas identificadas no projeto lógico.

O documento gerado no final desta etapa, contendo o *modelo físico*, a *interligação entre os sistemas envolvidos*, e os *sistemas de controle, recuperação e segurança*, deve ser aprovado, em reunião entre a equipe de especificação e o usuário.

2.5 Simulação computacional do sistema

A solução proposta no projeto físico do sistema automatizado utilizando robô deve ser avaliada através da sua simulação computacional utilizando-se, por exemplo, o *software* de modelagem de robôs Workspace[®] (Robot Simulation LTD., 1998).

2.6 Definição dos parâmetros de especificação do sistema

Devem ser definidos os parâmetros de especificação do robô, os quais são importantes para a aplicação a que se destina. Deverão ser definidos, por exemplo: número de graus de liberdade; número de braços; dimensões dos braços (comprimentos e seções transversais); espaço de trabalho (“*workspace*”); capacidade (carga máxima suportada e número de elementos que podem ser acoplados no atuador final); peso máximo do sistema completo; mecanismos de fixação e movimentação dos pontos de apoio do robô; velocidade máxima de movimentação do atuador final; velocidade de aquisição de dados sobre as trajetórias em tempo real; aceleração máxima; precisão e exatidão do sistema de posicionamento; tipos dos mecanismos de acionamento das juntas, adaptados às condições de trabalho e requisitos de nível de ruído; métodos de controle de posição do robô; capacidade de armazenamento de dados do sistema de controle; capacidade da fonte de alimentação; linguagens de programação; interface com programas de simulação de robôs, como o Workspace[®], permitindo programação “*off-line*”; interface do robô com o computador; temperatura e ambiente de operação; tempo de inspeção do sistema automatizado; tolerância do sistema a ruídos; dispositivos para evitar a entrada de água no manipulador e sistemas associados, quando for o caso; empuxo dos sistemas que ficarão submersos, se for o caso (cabos de comunicação, energia, sinais de vídeo, tubulações de ar comprimido, etc.); garantia, em termos de número de horas de operação, e procedimentos de testes e manutenção periódica. A aplicabilidade dos parâmetros listados e a necessidade de especificação de parâmetros adicionais deverão ser definidas para cada caso.

3. EXEMPLO DE APLICAÇÃO DA METODOLOGIA

Na Figura 1, é apresentado um exemplo fictício de um Diagrama de Contexto, parte do *projeto pré-conceitual*, para um sistema automatizado de inspeção ultra-sônica de vasos de pressão de classe nuclear. Como possíveis informações que deverão ser obtidas junto ao usuário, neste exemplo, para elaborar a documentação necessária para esta fase, podem ser mencionadas: grau de automação necessário para as tarefas de inspeção; necessidade de análise automatizada dos sinais de ultra-som pelo sistema, gerando automaticamente um relatório de defeitos ou não conformidades; procedimentos genéricos de inspeção; principais problemas operacionais e de

segurança radiológica e convencional encontrados durante a inspeção; tipos de equipamentos, componentes e regiões a serem inspecionados (vasos de pressão, bocais, soldas, etc.); tipos de inspeções a serem realizadas (em tempo real ou não, interna e/ou externa, etc.); dificuldade de acesso dos equipamentos de inspeção; disponibilidade de recursos no local de inspeção (energia elétrica, ar comprimido, instalações para descontaminação, etc.); flexibilidade necessária para atender demandas de inspeção (inspeções durante a montagem dos equipamentos, inspeções em serviço, inspeções em reatores de testes ou de submarinos, etc.); necessidade de interface com sistemas já existentes (sistemas de aquisição de dados, sistemas de posicionamento, instrumentação de ultra-som, bem como, *hardware* e *software* de uma maneira geral); filosofia do sistema de posicionamento dos transdutores de ultra-som (se visual, interativo, automatizado, com pontos de controle, etc.). Neste exemplo, este tipo de informação, de caráter geral, influenciará na definição do conceito do sistema de inspeção a ser especificado, e deverá ser obtido o quanto antes possível.

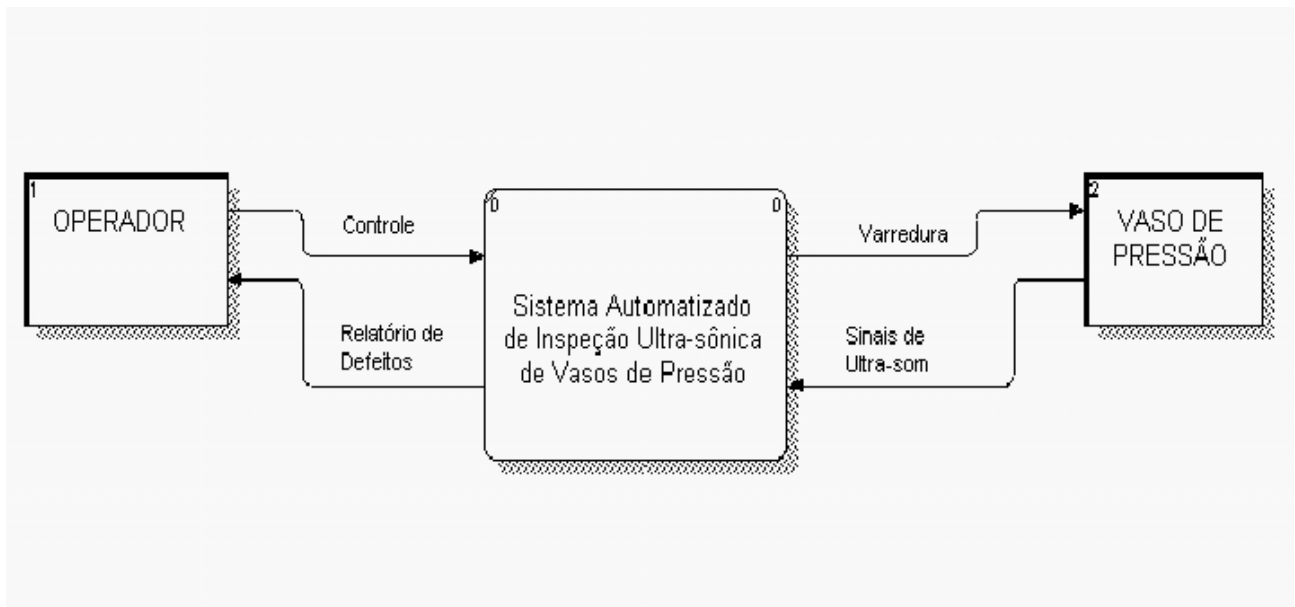


Figura 1 - Exemplo fictício de um diagrama de contexto para um sistema automatizado para inspeção ultra-sônica de vaso de pressão.

Na Figura 2 é mostrado um DFD de primeiro nível, parte do projeto conceitual do exemplo analisado. O que se procura obter, com a utilização deste tipo de diagrama, é uma visão geral do sistema, de modo a permitir que o usuário analise se os objetivos do sistema serão atingidos, sem no entanto se perder nos aspectos operacionais. Nesta fase, um dicionário de dados deve ser inicializado, através da descrição do fluxo de dados e arquivos, até o nível de elementos (dados individuais que não podem mais ser decompostos em outros conjuntos de dados), sem entretanto descrever, ainda, estes elementos individualmente.

O sistema deverá se comunicar com o ambiente externo por meio dos fluxos de dados identificados no diagrama de contexto, que darão origem aos *elos de comunicação*, que se caracterizam como dados, sinais, relatórios, telas, documentos, arquivos, etc. Nesta etapa, devem ser definidos os arquivos e fluxos de dados que farão as interfaces com outros sistemas.

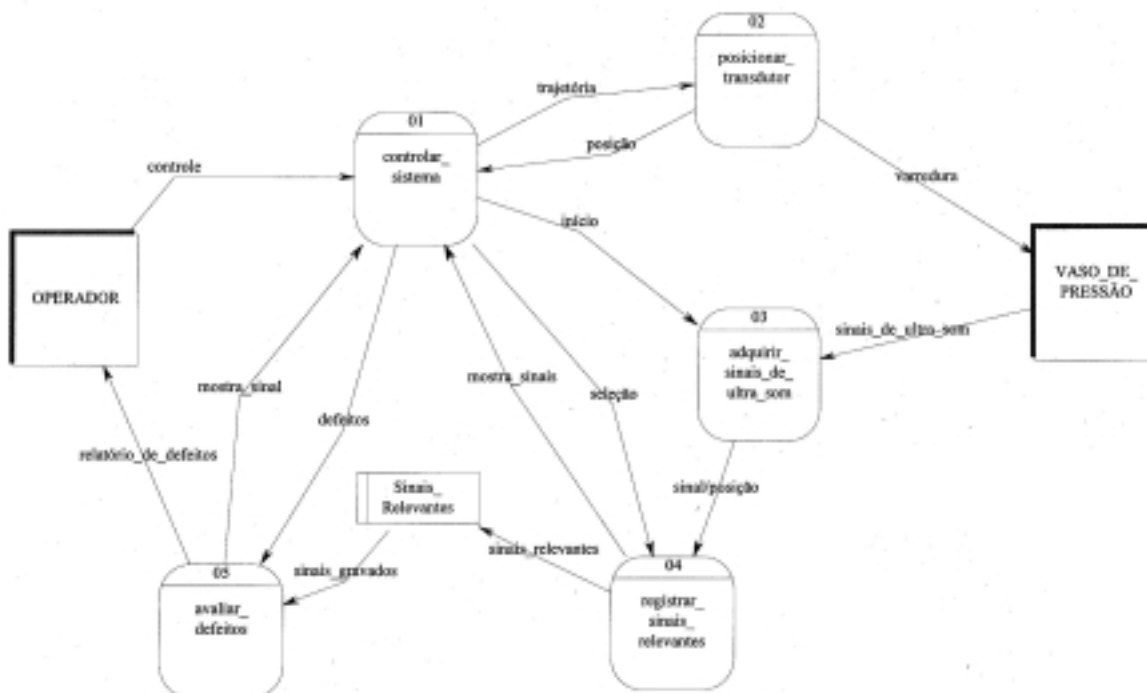


Figura 2 - Exemplo fictício de um DFD, de primeiro nível, para um sistema automatizado de inspeção ultra-sônica de vasos de pressão.

No *projeto conceitual* devem também ser definidos os conceitos básicos resumindo o conjunto de termos e idéias que são fundamentais para o conhecimento dos requisitos para o sistema. Nos parâmetros básicos, devem constar a definição de fatores condicionantes ou medidas que influenciem a especificação ou a operação do sistema. Deve ser definido, por exemplo, se as inspeções serão submersas e se deverão ser acompanhadas de um sistema de visualização com câmera de vídeo para auxiliar no posicionamento dos transdutores de ultra-som, evitar obstáculos e realizar os ensaios. Deverá, também, ser definida se haverá a necessidade de exame visual das regiões internas do vaso de pressão e de equipamentos específicos para esta tarefa. Os requisitos de inspeção para cada região, bem como os acessórios internos que dificultam o acesso a essas regiões, devem ser identificados. Devem ser relacionadas as condições de realização dos ensaios que influenciem na especificação dos materiais, componentes e equipamentos, como, por exemplo, a presença de radiação gama e nêutrons no ambiente de trabalho do robô.

Devem ser levantadas, nesta etapa do projeto, a legislação e as normas relacionadas, por exemplo, com os critérios de segurança referentes ao uso de robótica e à inspeção de instalações nucleares (códigos ASME – “The American Society of Mechanical Engineers”, normas da NRC – “Nuclear Regulatory Commission”, CFR – “Code of Federal Regulations”, CNEN – Comissão Nacional de Energia Nuclear, manuais de garantia da qualidade, programas, manuais e procedimentos de inspeção, etc.).

Deve ser definido se os usuários do sistema trabalham, por exemplo, basicamente com pesquisa e desenvolvimento e, além das tarefas rotineiras de inspeções para monitoração e reparos, deverão realizar ensaios fora de rotina para verificar influências de parâmetros de funcionamento específicos no aparecimento e evolução de trincas no vaso de pressão, bocais e suas soldas. Devem ser levantados os conceitos e os parâmetros básicos do sistema, por meio de entrevistas e reuniões com os usuários finais e com os gerentes responsáveis, bem como através da análise de informações técnicas de sistemas semelhantes existentes.

No *projeto lógico* do exemplo apresentado deve ser verificado se existe a necessidade de manter arquivos contendo informações de inspeções anteriores, possibilitando, por exemplo, o acompanhamento da evolução de trincas. Deve ser verificado se é necessário ter-se usuários com diferentes níveis de acesso, controlados através de senhas. Em caso afirmativo, devem ser estabelecidos, nesta etapa, os diferentes comandos, dados de entrada e dados de saída aplicáveis a cada nível de usuário. As entradas necessárias podem ser, por exemplo, as opções de controle, as informações relativas a destino, trajetória e programação do manipulador robótico. As possíveis saídas do sistema automatizado podem ser o movimento para o destino selecionado pelo usuário, as imagens obtidas por eventuais câmeras de vídeo, as informações periódicas sobre a posição dos transdutores de ultra-som e os sinais de ultra-som. A descrição detalhada do dicionário de dados nesta etapa será fundamental para as definições do projeto físico a seguir.

No *projeto físico*, devem ser abordados os aspectos de segurança relativos aos operadores e à instalação que esteja sendo inspecionada, bem como à integridade dos equipamentos utilizados, tanto durante a instalação e retirada do sistema de inspeção no vaso, quanto durante as inspeções. Como o sistema constituído pelo manipulador robótico e os equipamentos de ultra-som deverão trabalhar em inspeção dentro do vaso do reator, as preocupações com a segurança convencional, durante as inspeções, estarão restritas aos possíveis danos causados ao vaso do reator, seus componentes e ao próprio sistema. Assim, a segurança da operação do manipulador robótico deverá ser garantida, por exemplo, através de: limitação de tamanho, carga e velocidade do manipulador robótico; sistema de monitoração via câmera de vídeo; parada de emergência; consideração das condições de trabalho na especificação do sistema de inspeção (temperatura, radiação, ambiente corrosivo, etc.); sistema para evitar choques com obstáculos; redundância no sistema, para fixação do manipulador robótico ao vaso e projeto dos sistemas, levando-se em conta o critério de falha segura.

São também importantes, no projeto físico, informações como a forma de fixação da base do manipulador na parte superior das paredes cilíndricas do vaso, com proteção para evitar a queda do sistema no vaso; valor do empuxo dos componentes do braço, da base e câmara de controle para evitar cargas no sistema de fixação da base; e pressão positiva no interior destes componentes para evitar entrada de água, em caso de falhas.

As informações adicionais, listadas a seguir, constituem partes importantes do projeto físico e devem auxiliar na especificação do sistema: necessidade de vedação para os transdutores, sistemas elétricos e mecânicos; tempo máximo de inspeção, para não danificar os equipamentos e minimizar os riscos de exposição radiológica aos operadores; percentual das regiões do vaso de pressão a ser inspecionado; frequência de inspeção; necessidade de dispositivos para transporte e armazenamento do sistema de inspeção; relação entre velocidade de varredura, precisão e exatidão do sistema de posicionamento, características dos transdutores e programa de inspeção e interface entre sistema de posicionamento e transdutores.

Na etapa de *simulação computacional* do sistema devem ser modelados, neste exemplo, o manipulador robótico (ou sistema de posicionamento), os transdutores de ultra-som, o vaso de pressão, os bocais e as soldas a serem inspecionadas.

A integração dos modelos geométricos dos sistemas envolvidos e a simulação computacional dos movimentos necessários à inspeção ultra-sônica, das diversas regiões e soldas definidas, terá importante papel na definição do conceito do sistema de movimentação dos transdutores de ultrassom e dos diversos parâmetros de especificação do sistema.

4. CONCLUSÕES

A metodologia apresentada neste trabalho para especificação de robôs foi utilizada para a especificação de um robô para operar em um sistema fictício de ensaio automatizado de inspeção ultra-sônica de soldas de um vaso de pressão de classe nuclear. A subdivisão da complexa tarefa de especificação em etapas mais simples e a documentação dos compromissos assumidos entre a equipe de especificação e o usuário é uma contribuição importante no sentido de se obter a melhor alternativa para o sistema automatizado. Componentes como vasos de pressão de reatores nucleares, tubos dos geradores de vapor e tubulações de seus circuitos primários, são inspecionados por métodos de ensaios não destrutivos, com o auxílio de sistemas de ensaio automatizados. Para estas aplicações, a tarefa de especificação de robôs adquire uma grande importância.

5. AGRADECIMENTOS

Os autores agradecem ao CDTN/CNEM – Centro de Desenvolvimento da Tecnologia Nuclear/Comissão Nacional de Energia Nuclear e ao CTMSP – Centro Tecnológico da Marinha, São Paulo, o apoio ao desenvolvimento deste trabalho.

6. REFERÊNCIAS

- Christóvão, M.T., 1999, “Conceitos Básicos de Análise Essencial”, CDTN/CNEN, Belo Horizonte, MG (Nota Interna NI-AL3-02/99).
- Christóvão, M.T., Batista, J.R. e Jabace, A.C., 1999, “Metodologia de Desenvolvimento de Sistemas”, CDTN/CNEN, Belo Horizonte, MG (Nota Interna NI-AL3-01/99).
- Gane, C., 1990, “O Relatório Gane”, Livros Técnicos e Científicos Editora S. A., Rio de Janeiro.
- Gane, C. e Sarson, T., 1984, “Análise Estruturada de Sistemas”, Livros Técnicos e Científicos Editora S.A., Rio de Janeiro.
- Kiebel, G.R., 1993, “An approach to software quality assurance for robotic inspection systems”. Proceedings, 41st Conference on Robotics and Remote Systems, p-51-55.
- Mershon, J.D. and Bendyna, N., 1997, “BPWin Methods Guide”, Logic Works, Berkely, California.
- Oliveira, J.E.L., Maestrelli, N.C. e Miguel, P.A.C., 1999, Sistema de classificação e codificação para seleção de robôs. “Metodologia”, Maio, pp 24-31.
- Robot Simulation LTD., 1998, “Workspace 4 - Educational user guide manual”, Newcastle-upon-Tyne, United Kingdom.
- Shakeri, C., Brown, D.C. and Noori, M.N., 1997, “A knowledge design tool for the concurrent design of robots”. Wisconsin University, ([http://me.wpi.edu/~cirrus/Publicat... Engineering/](http://me.wpi.edu/~cirrus/Publicat...))

METODOLOGIA PARA A MODELAGEM DE SISTEMAS DE ELEVADORES EM EDIFÍCIOS INTELIGENTES

Gladys Bastidas Gustin
Paulo Eigi Miyagi
Diolino Dos Santos Filho

Universidade de São Paulo, Escola Politécnica, Departamento de Engenharia Mecatrônica e de Sistemas Dinâmicos, 05508-900 São Paulo, SP, Brasil. E-mail: gladysbg@usp.br

Resumo

O sistema de elevadores é um recurso fundamental para o transporte vertical de pessoas e bens e, constitui um componente imprescindível dos edifícios inteligentes. Neste contexto, sua integração com outros sistemas prediais, sua disponibilidade e a eficiência do seu serviço, devem ser objetos de cuidadosa análise em função de sua influência no desempenho e produtividade de todo o edifício e das atividades dos usuários do mesmo. Considerando estes aspectos e o fato destes sistemas terem um comportamento dinâmico definido principalmente pela ocorrência de eventos discretos, o presente trabalho introduz uma metodologia para a modelagem destes sistemas baseada em redes de Petri interpretadas.

Palavras-chave: Redes de Petri, Edifícios inteligentes, Sistemas de elevadores, Modelagem de sistemas a eventos discretos.

1. INTRODUÇÃO

As demandas atuais de maior eficiência, produtividade e segurança, têm-se convertido em considerações cada vez mais relevantes no projeto de sistemas. Neste contexto, observa-se que para o caso específico dos edifícios inteligentes, novas metodologias estão sendo desenvolvidas no sentido de considerar peculiaridades relacionadas à funcionalidade que se deseja associar ao edifício. De acordo com estas abordagens, o comportamento e interação dos usuários que utilizam um determinado edifício devem ser considerados, permitindo incrementar a produtividade destes neste ambiente, uma vez que o prédio estaria assim, mais apto a atender às suas necessidades (Finley et al., 1991).

Nos projetos mais recentes para construção de edifícios inteligentes procura-se a instalação de sistemas automatizados, cuja função é a realização autônoma e integrada das diversas funções e serviços existentes em seus ambientes. A maior parte destes sistemas possuem como característica um comportamento dinâmico definido através das mudanças de estados devido à ocorrência de eventos discretos. A classe de sistemas definida por esse comportamento é denominada Sistemas a Eventos Discretos – SED (Ramadge & Wonham, 1989) para os quais têm sido desenvolvidas várias técnicas (Redes de Petri, Cadeias de Markov, Teoria de filas, Álgebra mini-max, Máquinas de estados, etc.) para sua modelagem, análise, controle e projeto. Entre estas técnicas uma de comprovada eficiência para sistemas desta natureza é a baseada na teoria de redes de Petri (Peterson, 1981). Os principais sistemas onde esta técnica é aplicada são: sistemas de comunicações, sistemas de software, sistemas de

manufatura e mais recentemente nos sistemas prediais (Gomes, 1997). Esta técnica consiste em utilizar as redes de Petri e suas extensões, para a modelagem gráfica da estrutura do sistema e do seu comportamento dinâmico. Entretanto, na modelagem de sistemas complexos e com diferentes níveis hierárquicos se evidencia um detrimento em uma de suas principais características: sua visualização gráfica. Assim é adequado considerar que a modelagem seja realizada usando interpretações não formais, seguida de procedimentos de detalhamento gradativo e com interpretações formais. É neste contexto que as técnicas do PFS (*Production Flow Schema*) e do MFG (*Mark Flow Graph*) (Miyagi, 1996) foram desenvolvidas. O PFS e o MFG são extensões interpretadas das redes de Petri, próprias para aplicação em diferentes níveis de modelagem, análise e controle de SED. Este procedimento é denominado metodologia PFS/MFG.

Assim, o objetivo deste trabalho é introduzir uma metodologia de modelagem de sistemas de elevadores em edifícios inteligentes, mediante o uso do PFS/MFG, que considere a integração deste com outros sistemas prediais, e sirva de auxílio para o projeto destes sistemas.

2. REDES DE PETRI

2.1 Metodologia PFS/MFG

A metodologia PFS/MFG consiste num procedimento sistemático e racional baseado na interpretação da abordagem hierárquica de redes de Petri. Inicialmente o modelo conceitual (representando um alto nível de abstração do sistema e sem consideração de sua dinâmica) do sistema é desenvolvido através do PFS. Nesta etapa o propósito é modelar as principais características das funções que serão consideradas no sistema, isto é, a ênfase está na identificação dos componentes ativos e passivos do sistema, assim como do fluxo de itens (pessoas, material e/ou informação) entre estes elementos. Para a modelagem do comportamento dinâmico do sistema, o modelo em PFS é convertido progressivamente em um modelo em MFG, isto é, uma classe de rede de Petri interpretada que detalha a dinâmica das diversas partes do sistema (Miyagi, 1996).

Os elementos estruturais de um PFS estão representados na fig. 1. São eles: a) *atividade* que representa os componentes ativos; b) *Inter-atividade ou distribuidor* que representa os componentes passivos e c) *arco orientado* que indica a relação lógica entre os componentes do sistema. O PFS não possui o conceito de marcação.

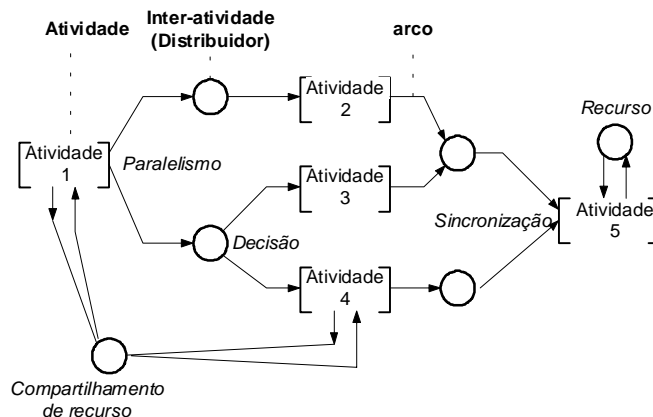


Figura 1. Exemplo de um modelo em PFS com a identificação de seus elementos estruturais e algumas das características de SEDs que é capaz de modelar.

Para a descrição funcional (da dinâmica) do sistema, os elementos do grafo PFS são então detalhados. Este detalhamento pode gerar sub-grafos totalmente em PFS ou sub-grafos híbridos (PFS/MFG) com alguns elementos em PFS e outros em MFG ou, sub-grafos inteiramente em MFG. Os elementos estruturais do MFG (ilustrados na fig. 2) são: a) *marcas*: que indicam a manutenção de condições; b) *boxes*: que representam condições (pré e pós-condições) que podem estar associadas ao modo de operação ou à disponibilidade de um recurso do sistema. Estes são chamados *boxes temporizados* quando consideram um intervalo de tempo de manutenção de marcas em seu interior; c) *transições*: correspondem a um evento que causa a mudança de estado do sistema. Estas podem ter inscrições que representam regras adicionais de disparo. Quando nesta é considerado o decorrer do tempo, a transição é chamada de *transição temporizada*; d) *arcos*: os quais estabelecem relações causais entre os eventos e as condições ou vice-versa; f). *portas*: que habilitam ou inibem a ocorrência dos eventos correspondentes as transições, sendo denominadas *habilitadoras* ou *inibidoras* segundo seja o caso; g) *arco de sinal de saída*: permitem a representação explícita da interação do grafo com elementos externos do modelo. Destaca-se que no MFG, é usado o conceito de marcação (distribuição dinâmica das *marcas*), e é através deste que se reflete a dinâmica dos estados do sistema.

Além disso, o MFG apresenta extensões que incrementam o seu poder de modelagem e adicionam novos elementos a este, algumas destas extensões são: o E-MFG (Santos, 1993) que possui uma hierarquia no processo de disparo das *transições*, e é composto por a) *marcas com atributos*; b) *boxes controladores*: os quais executam atribuições de valores aos atributos de uma marca; c) *regras de produção nas transições* e d) *filtros dos atributos nos arcos* e o MFG auto-modificável (Gustin, 1999) que permite alterar dinamicamente a estrutura do modelo em função do estado atual do sistema, através do uso de *arcos auto-modificáveis* cujos pesos, dependem da marcação em lugares específicos definidos previamente (vide fig. 2).

Esta metodologia permite assim que, através de uma visão macro (genérica) e conceitual do sistema, os diferentes sub-sistemas e suas funções sejam detalhados até o nível de interface com os dispositivos físicos instalados no edifício. Este nível de detalhe facilita o controle (nível de hardware e conexões) do sistema e explicita sua implementação computacional (controle programável).

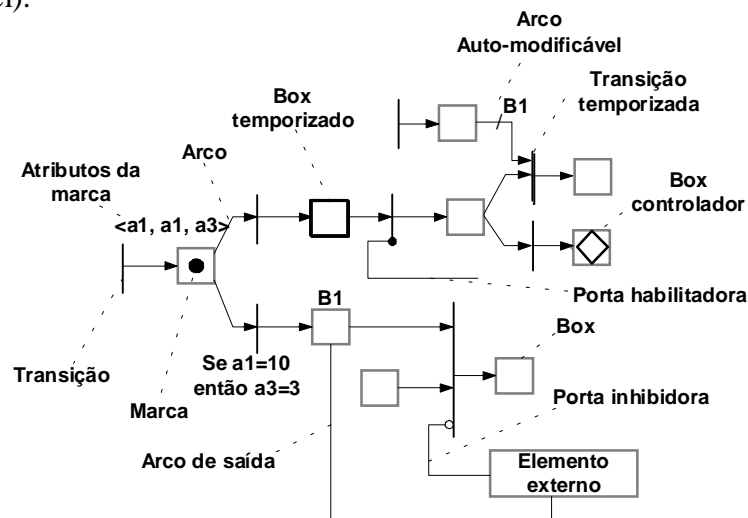


Figura 2. Exemplo de um modelo em MFG e algumas das suas extensões com a identificação de seus elementos estruturais.

3. METODOLOGIA DE MODELAGEM

A metodologia para a modelagem de sistemas de elevadores pode ser dividida nas etapas gerais ilustradas na fig. 3. Em cada etapa é considerada a avaliação da mesma e o retorno a uma etapa anterior de ser necessário.

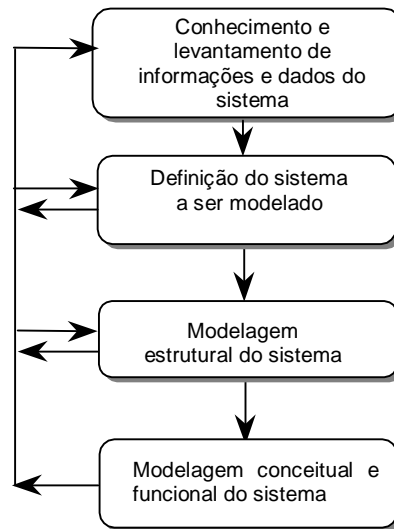


Figura 3. Metodologia para a modelagem de sistemas de elevadores em edifícios inteligentes.

Etapa 1: Conhecimento e levantamento de informações e dados do sistema

Esta etapa tem como finalidade identificar as características do sistema que permitam a sua definição. Aqui devem considerar-se os requisitos e funções necessárias do sistema.

Nesta etapa procura-se conhecer o sistema de elevadores, através do levantamento de informações (tanto do edifício, como do sistema propriamente dito), para realizar uma análise preliminar e identificar os dados relevantes ao sistema, pertinentes para a modelagem do mesmo e que permitam caracterizar quantitativamente os elementos deste, tanto para o caso de um edifício existente ou para um em fase de projeto.

Etapa 2: Definição do sistema a ser modelado

Nesta etapa, as informações levantadas são analisadas e avalia-se a validade e viabilidade técnica do atendimento das necessidades, considerando as normas relacionadas com os sistemas de elevadores e o tipo e função do prédio onde são instalados, além da tecnologia disponível, para se ter uma definição precisa do que é requerido.

Estas considerações permitem também que nesta etapa, sejam definidas as estratégias de controle a serem implementadas no sistema.

Etapa 3: Modelagem estrutural do sistema

Esta etapa consiste no desenvolvimento do modelo estrutural do sistema de elevadores, onde são identificadas as partes que o compõem, considerando-o dentro do contexto dos edifícios inteligentes e levando em conta a arquitetura de controle adotada para o sistema.

Nesta etapa o sistema é abordado de forma bipartida, dado que este pode ser representado mediante a inter-conexão de dois subsistemas fundamentais que se comunicam cooperativamente, estes subsistemas representam a parte operativa relacionada ao objeto de controle (cabins dos elevadores) e a parte de controle relacionada ao sistema de controle.

Esta decomposição permite tratar mais adequadamente a complexidade do sistema de elevadores para a construção do modelo global do sistema.

Etapa 4: Modelagem conceitual e funcional do sistema

Nesta etapa a modelagem de cada um dos elementos do modelo estrutural do sistema é realizada de forma sistemática e hierárquica. Inicialmente, aborda-se a modelagem conceitual

para então através de um refinamento do mesmo, obter seu modelo funcional. Assim, primeiramente aborda-se o sistema em um nível de abstração que não considera o seu comportamento dinâmico e, posteriormente realiza-se o detalhamento dos modelos.

Para esta etapa é adotada como ferramenta a metodologia PFS/MFG, pelas suas características e eficiência no desenvolvimento de modelos.

4. ESTUDO DE CASO

4.1 Prédio dos ambulatórios (PAMB-HC)

O estudo de caso considerado neste trabalho refere-se à modelagem do sistema de elevadores do prédio dos ambulatórios (PAMB) do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HC-FMUSP). Este é formado por 6 institutos (Central, Coração, Pediatria, Ortopedia, Psiquiatria e Medicina Nuclear) e mais 6 hospitais auxiliares. O prédio do PAMB faz parte do Instituto Central.

A seguir é apresentada a modelagem do sistema de elevadores do PAMB, resultante da aplicação da metodologia descrita.

Etapa 1: Conhecimento e levantamento de informações e dados do sistema

Inicialmente foi identificada a função do sistema de elevadores dentro do PAMB: possibilitar o transporte vertical de pacientes, visitantes, funcionários da área de saúde e funcionários em geral, além evidentemente de equipamentos e outras cargas móveis.

Algumas das informações levantadas foram:

- O edifício do PAMB possui uma área construída de cerca de 116.000 m² e, é composto por 11 andares e mais 7 andares técnicos (interandares).

- O PAMB conta com 12 elevadores de uso geral (5 grupos) e 2 elevadores para manutenção, distribuídos em quatro blocos de circulação (A, B, C, D)

Etapa 2: Definição do sistema a ser modelado

Segundo a norma brasileira para este tipo de edifícios (hospitais), deve-se ter em conta entre outras as seguintes especificações:

- Os elevadores de pacientes servindo a mais de quatro andares devem ter comando automático coletivo.

- Deve existir pelo menos, um elevador adaptável para as manobras do corpo de bombeiros.

- O sistema de elevadores deverá contar com estratégias adequadas em caso de incêndio e falta de energia.

A análise das informações levantadas do sistema e, a consideração das normas que existem para hospitais, resultaram na especificação e descrição da seqüência de eventos requerida para cada uma das seguintes estratégias (detalhes das estratégias podem ser encontrados em (Barney, 1985; Elevator world, 1990)) de funcionamento e integração do sistema de elevadores com outros sistemas prediais:

Estratégia duplex/triplex (de atendimento aos usuários);

Estratégia para tráfego de pico de subida (de atendimento aos usuários);

Estratégia para tráfego de pico de descida (de atendimento aos usuários);

Estratégia em caso de incêndio;

Estratégia em caso de falta de energia;

Estratégia para manutenção preventiva;

Estratégia em caso de falha.

Etapa 3: Modelagem estrutural do sistema

O modelo estrutural do PAMB é apresentado na fig. 4. Nele são identificados (linha tracejada) tanto os elementos que pertencem a um grupo (grupo D) como as partes que correspondem a cada cabina (cabina 1) do grupo (relacionados com os blocos de circulação A,B,C e D).

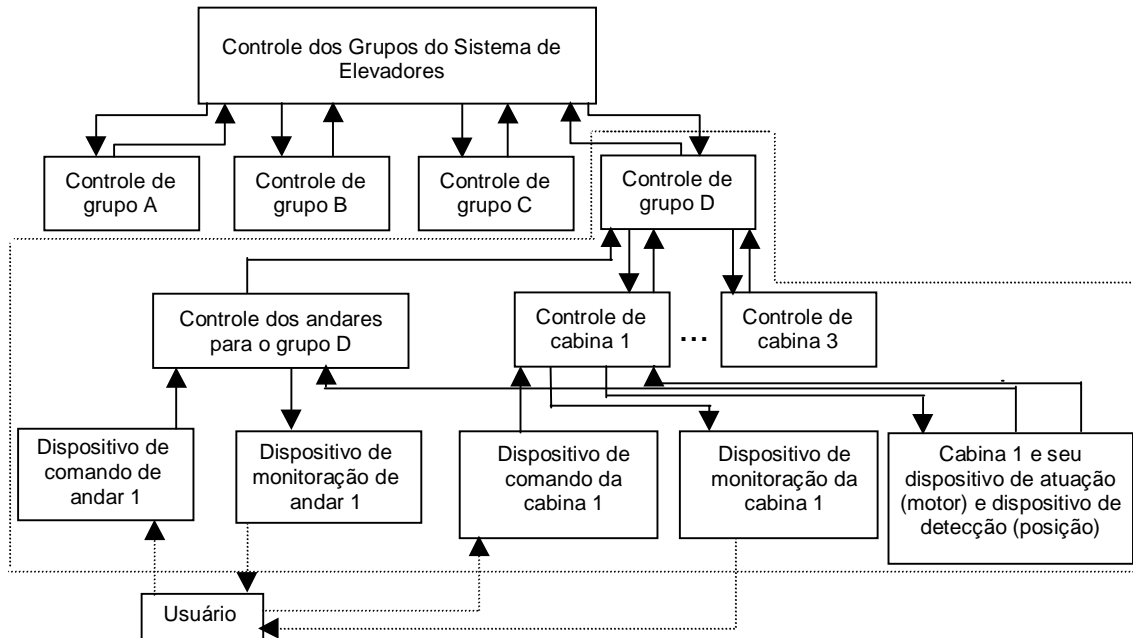


Figura 4. Modelo estrutural do sistema de elevadores do PAMB.

Etapa 4: Modelagem conceitual e funcional do sistema de elevadores

Os modelos conceitual e funcional do sistema de elevadores do PAMB são desenvolvidos de acordo com as informações e a definição do sistema e segundo o modelo estrutural. Assim a modelagem conceitual considera cada um dos elementos estruturais do sistema em uma abordagem *top - down*. Esta modelagem é realizada usando o PFS e o MFG e suas extensões.

Considerando a complexidade do modelo completo do sistema de elevadores do PAMB, devido ao número de elevadores e andares envolvidos e, com a finalidade de exemplificar modelos mais compactos e de fácil entendimento, serão apresentados aqui apenas alguns dos modelos mais gerais e representativos do sistema.

- Modelagem do controle de grupo

Neste nível são consideradas duas partes: as estratégias a serem executadas pelo grupo, e as operações sobre os controles de cabina. (vide fig. 5).

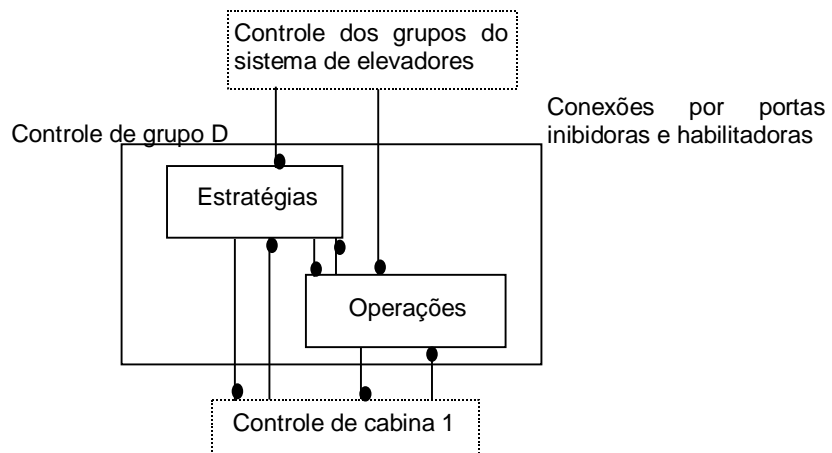


Figura 5. Esquema das atividades realizadas pelo controle de grupo D.

Um modelo desta parte é apresentado a seguir:

Modelagem das estratégias

Um exemplo da modelagem das estratégias correspondente à estratégia em caso de incêndio, é apresentado na fig. 6.

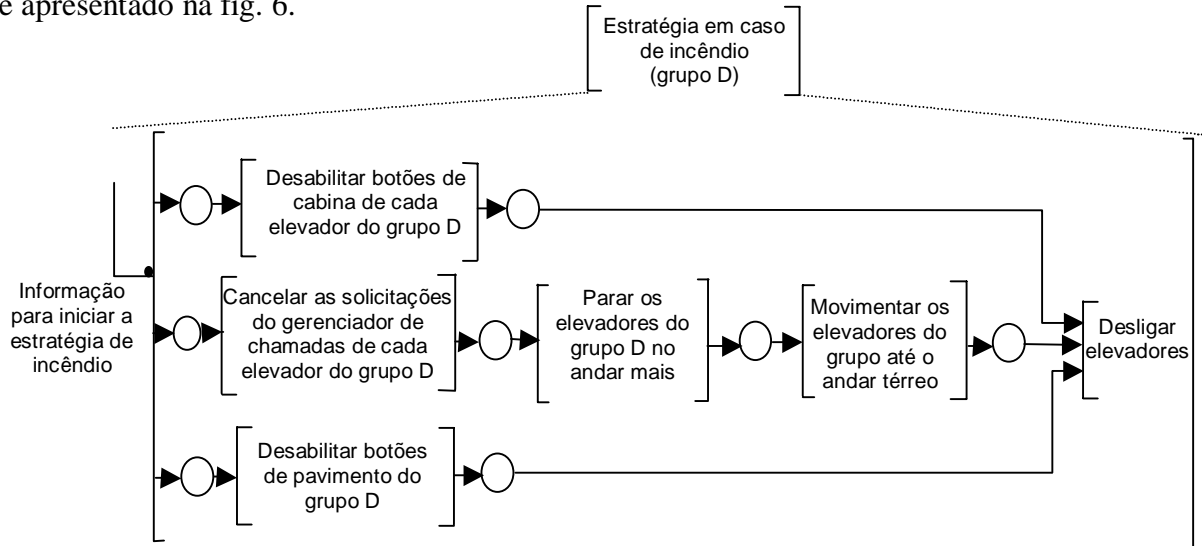


Figura 6. Modelo em PFS da estratégia em caso de incêndio para o grupo D.

- Modelagem do objeto de controle (cabina e dispositivo de atuação)

No nível do objeto de controle, são identificadas duas *atividades* básicas para cada elevador: [subir], [descer] e um elemento *inter-atividade* que representa a parada nos andares (vide fig. 7).

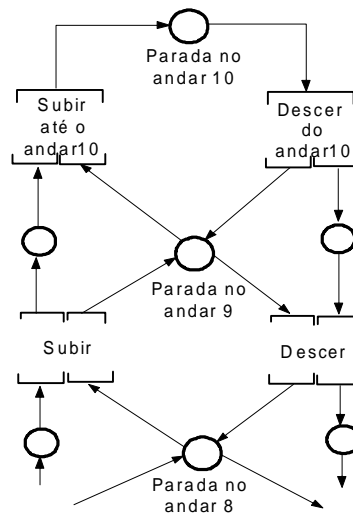


Figura 7. Modelo em MFG do objeto de controle (cabina 1).

A partir do modelo conceitual em PFS desenvolve-se um modelo MFG através de refinamentos das atividades consideradas. Este modelo é apresentado na fig. 8.

Como ilustrado nos modelos apresentados o detalhamento sucessivo das atividades permite a construção do modelo completo do sistema, ao explicitar as interações entre os diferentes dispositivos dos sistemas, e identificar os sinais de controle do sistema e os parâmetros de entrada/saída da programação. Desta forma a presente abordagem confirma ser efetiva para o projeto de integração dos sistemas de elevadores com outros sistemas prediais.

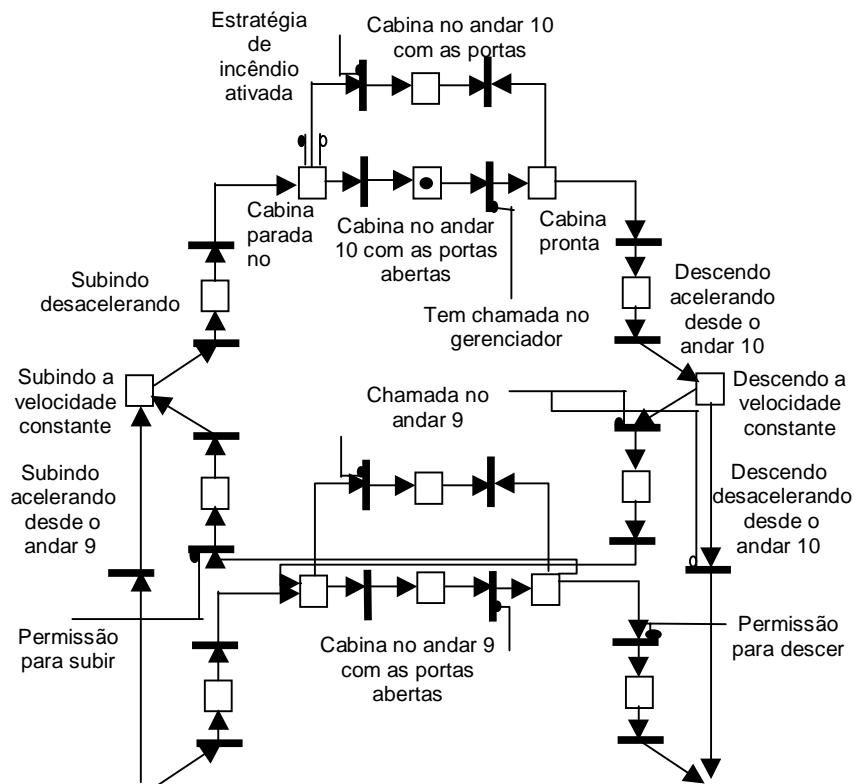


Figura 8. Modelo em MFG do objeto de controle (cabina 1).

5. CONCLUSÕES

No trabalho foi proposta uma metodologia para a modelagem de sistemas de elevadores através do uso das redes de Petri e suas extensões, mediante a utilização da metodologia PFS/MFG, visando o apoio ao projeto destes sistemas em edifícios inteligentes. Para demonstrar a efetividade da metodologia, foi considerado como estudo de caso um prédio hospitalar (PAMB), no qual a modelagem resultou satisfatória ao permitir o estudo da dinâmica e a integração do sistema de elevadores com outros sistemas prediais, a geração da especificação funcional de todo o sistema (objeto e controle e sistema de controle) e explicitar as interações entre os diferentes dispositivos deste.

A metodologia PFS/MFG também confirmou através da especificação do modelo funcional do sistema, ser efetiva para definir as funções de controle para o projeto de controladores programáveis, facilitando desta forma a implementação do controle, além da integração com outros sistemas.

Agradecimentos

Os autores agradecem as entidades que auxiliam diferentes partes desta pesquisa: HC-FMUSP, FSP-USP, CNPq e FAPESP.

6. REFERÊNCIAS

- Barney, G. C; Dos Santos, S. M., 1985, Elevator Traffic Analysis, Design and Control. Ellis Horwood Limited, Chichester.

- ELEVATOR WORLD, 1990, *Educational Package and Reference Library*, v. 1, Mobile, Editor Elevator World Educational Division.
- Finley, M.R.; Karakura, A.; Nbogni, R, 1991, Survey of intelligent buildings concepts, *IEEE Communications Magazine*, vol 29, n. 4, pp. 18-23.
- Gomes, L.F., 1997, Redes de Petri reactivas e hierárquicas - integração de formalismos no projecto de sistemas reactivos de tempo-real, Tese de Doutoramento, Universidade Nova de Lisboa, Lisboa, Portugal.
- Gustin, G. B., 1999, Aplicação de redes de Petri interpretadas na modelagem de sistemas de elevadores em edifícios inteligentes. *Dissertação de mestrado*, Universidade de São Paulo, São Paulo.
- Miyagi, P.E., 1996, Controle Programável, Editora Edgard Blücher, São Paulo.
- Peterson, J.L., 1981, Petri Net Theory and The Modeling of Systems, Prentice Hall, Englewood Cliffs.
- Ramadge, P.J. & Wonham, W.M.,1989, The control of discrete event systems, *Proceedings of the IEEE*, vol.77, n. 1.
- Santos, D., 1993, Proposta do Mark Flow Graph estendido para a modelagem e controle de sistemas integrados de manufatura. *Dissertação de mestrado*, Universidade de São Paulo, São Paulo.

MODELAGEM DE SISTEMA DE MOVIMENTAÇÃO DE MATERIAIS UTILIZANDO-SE UMA ABORDAGEM HÍBRIDA (ORIENTADA A PROCESSOS E A OBJETOS)

Fabrcio Junqueira

Paulo Eigi Miyagi

Diolino José dos Santos Filho

Universidade de São Paulo, Escola Politécnica, Depto. de Eng. Mecatrônica e de Sistemas Mecânicos, São Paulo, SP, Brasil. E-mail: fabri@usp.br

Resumo

Considera-se os sistemas de movimentação de materiais (tanto interno como de distribuição) em empresas de manufatura um dos elementos chave para a sua competitividade, pois dela depende a flexibilidade da produção e uma distribuição eficiente dos produtos. Desta forma, procura-se estudar ferramentas que permitam a modelagem e análise destes sistemas. Uma das ferramentas que tem se demonstrado efetiva para esta finalidade são as Redes de Petri Interpretadas como o E-MFG (Enhanced Mark Flow Graph). Assim, procura-se, neste trabalho, abordar o E-MFG de uma forma híbrida (orientada a processo e a objeto), visando-se aumentar a sua capacidade de modelagem de sistemas de movimentação de materiais.

Palavras-chave: E-MFG, Redes de Petri, Modelagem de Processos, Sistemas de Movimentação de Materiais.

1. INTRODUÇÃO

Muitas empresas estão procurando ferramentas que as ajudem a melhorar a eficiência de seus sistemas produtivos, ou seja, torná-los eficazes e permanecerem competitivas. Dentre as várias ferramentas aplicáveis para esta finalidade, pode-se destacar o E-MFG (Enhanced Mark Flow Graph) – uma Rede de Petri Interpretada para aplicação em sistemas de manufatura – que vem demonstrando ser eficaz tanto na modelagem de sistemas produtivos como na especificação do controle destes sistemas, possibilitando a execução de simulações para análise de desempenho e implementação do controle (Santos Filho, 1993; 1995).

Dentre os requisitos mais importantes para ser competitivo estão a agilidade na resposta às necessidades do mercado – responder num curto espaço de tempo – e a flexibilidade na produção, ou seja, poder alterar o seu mix de produtos em produção com rapidez. Para tanto, as linhas de produção não podem ser rígidas, ou seja, dedicadas a apenas um produto ou família de produtos, é necessário mobilidade e intercambialidade de recursos. Para se garantir esta mobilidade utilizam-se transportadores de alta flexibilidade tais como VATs (Veículos Autônomos de Transporte) que incluem os veículos guiados pelo homem e cujas características de autonomia e versatilidade não têm sido adequadamente modeladas quando o enfoque é orientado apenas ao processo (Inamasu, 1995; Santos Filho, 1998). Desta forma, o que se apresenta aqui é uma contribuição à solução deste problema.

2. CONCEITOS FUNDAMENTAIS

Antes de introduzir o conceito de modelagem híbrida (orientada a objeto e orientada para processos) utilizando-se Redes de Petri interpretadas, tal como o E-MFG, apresenta-se os conceitos de objetos e do E-MFG.

2.1 Objetos

A definição de objetos baseia-se em três conceitos fundamentais (Cantù, 1995):

- classes – é um tipo¹ de dado definido pelo usuário, o qual tem um estado², uma representação³ ou algumas operações⁴ ou comportamentos. Uma classe possui alguns dados internos e alguns métodos⁵, na forma de procedimentos ou funções. Uma classe usualmente descreve as características genéricas e o comportamento de uma série de objetos⁶ muito semelhantes;
- heranças – é a possibilidade de se definir um novo tipo a partir de uma já existente;
- polimorfismo – indica a capacidade de um objeto assumir várias formas, ou seja, permite-se que se referencie objetos de classes diferentes por meio da mesma variável.

Destes, os mais relevantes para este estudo são os dois primeiros por possibilitarem, respectivamente, o encapsulamento de “dados” e por possibilitar a utilização dos modelos existentes na confecção de outros modelos mais complexos.

Esta abordagem possibilita definir os elementos apenas uma vez, evitando-se duplicação de definições de elementos e facilitando futuras alterações dos modelos pois quando se altera, por exemplo, a estrutura de uma classe, todos os objetos que forem baseados nesta – que herdam as suas características – são atualizadas automaticamente.

2.2 O E-MFG

O E-MFG (Santos Filho, 1993; 1995) é uma extensão do MFG (Hasegawa, 1988; Miyagi, 1985; 1996). Este, assim com o MFG, é uma ferramenta de modelagem do comportamento dinâmico de sistemas de controle de equipamentos e sistemas de manufatura, com capacidade para representar assincronismo, seqüencialização, paralelismo, concorrência e intertravamentos. Este é composto pelos seguintes elementos:

- elementos do MFG (Miyagi, 1985):
 - *box*: indica uma condição e é representado por um bloco quadrado (figura 1 (a));

¹ Os tipos determinam os valores que uma variável pode assumir (Cantù, 1995).

² Focando-se um componente do sistema de movimentação, o transportador, teria-se os seguintes estados: parado, movimentando-se, carregando ou descarregando.

³ Como representação, focando-se o transportador, poderia-se ter a sua representação em E-MFG, indicando os seus possíveis estados e dinâmica (ver figura 9 como exemplo).

⁴ Ainda com relação ao transportador, poderia-se considerar as operações de carga e descarga de materiais.

⁵ Como exemplo de método, pode-se citar as respostas às instruções para parar / movimentar e instruções para carregar / descarregar materiais.

⁶ Objeto é uma instância de uma classe, em outras palavras, é uma variável do tipo de dados definido pela classe (Cantù, 1995).

- *transição*: representa a ocorrência de eventos e admite inscrições que representam regras adicionais restritivas para a evolução do estado do sistema (figura 1 (g));
- *marcas*: indicam a manutenção de uma condição (figura 1 (i)), podem ser individualizadas por *atributos* (figura 1 (j)), onde pode-se associar diversas informações referentes ao produto, ao processo, etc., podendo este assumir qualquer tipo;
- *arcos orientados*: estabelecem uma relação causal entre os eventos e as condições, podem conter inscrições variáveis que manipulam a transmissão dos *atributos* das *marcas* individuais (figura 1 (k));
- *arcos de sinal de saída*: podem transmitir informações aos dispositivos externos relativas ao estado dos *boxes*. Esta informação é derivada dos *atributos* da *marca* existente no respectivo *box*, através da especificação de inscrições variáveis nestes *arcos* (figura 1 (n)).
- elementos derivados de macro-representações:
 - *box capacidade*: é um macro elemento correspondendo a um grafo MFG com N elementos *box*, onde o símbolo N indica a sua capacidade, ou seja, quantos elementos *box* foram combinados para gerar o *box capacidade* (figura 1 (b));
 - *box agrupador*: um macro elemento correspondendo a um grafo MFG de uma operação de montagem (figura 1 (c));
 - *box dispersor*: um macro elemento correspondendo a um grafo MFG de uma operação de desmontagem (figura 1 (d)).
- elementos derivados de Redes de Petri Coloridas e outras extensões (Santos Filho, 1993):
 - *box temporizador*: quando uma *marca* aparece neste tipo de *box*, ela deve necessariamente permanecer neste *box* um intervalo de tempo, o qual pode ser determinado a partir de alguns dos *atributos* (figura 1 (f));
 - *transição temporizada*: uma vez que todas as condições de disparo estejam satisfeitas, esta *transição* só dispara após decorrido um intervalo de tempo. Se durante este tempo uma das condições deixa de ser satisfeita, a contagem do tempo é anulada. Será reiniciada somente após todas as condições estarem novamente satisfeitas (figura 1 (h));
 - *marca individual composta*: é uma *marca* simples que contém os *atributos* referentes à composição de várias *marcas* individuais simples;
 - *box controlador*: *box* onde se associa um conjunto de regras para a atualização dos *atributos* das *marcas*, as quais são regras de produção do tipo “se...então” referentes ao estado dos *atributos* e a atualização destes (figura 1 (e));
 - *portas habilitadoras / inibidoras*: portas que habilitam (figura 1 (l)) ou inibem (figura 1 (m)), respectivamente, a ocorrência de eventos. Esta condição adicional é especificada através de inscrições fixas relacionadas aos *atributos* agregados às *marcas*.

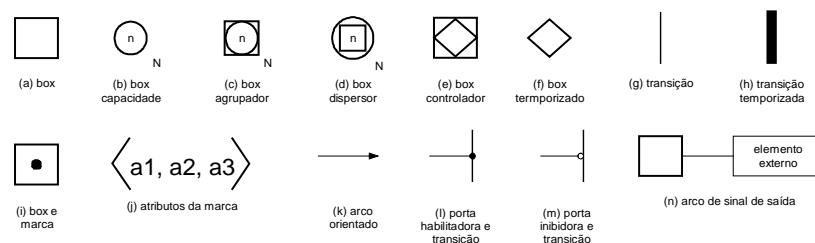


Figura 1. Elementos do E-MFG

Para efetuar a modelagem utilizando o E-MFG, devem ser observados os seguintes pontos:

- os *boxes* e as *transições* são necessariamente conectados de forma intercalada através dos *arcos orientados*.
- não existe limite para o número de *arcos* que entram ou saem dos *boxes* e das *transições*. Mas, em um par *transição-box* ou *transição-origem* da porta externa, pode existir apenas um *arco*.
- as *portas* e *arcos de sinal de saída* são formas de representação explícita das condições de controle.

A dinâmica de um sistema modelado em E-MFG é descrita através das regras de disparo das transições que alteram as marcações no grafo. Para mais detalhes destas regras consultar Miyagi [1996] ou Santos Filho [1993].

3. MODELAGEM HÍBRIDA E ESTUDO DE CASO

Afim de se apresentar a abordagem de modelagem híbrida (orientada a processo e a objeto), apresenta-se o exemplo de um sistema de movimentação de materiais utilizando-se inicialmente a metodologia orientada para processo na modelagem e em seguida a metodologia híbrida (orientada para processos e para objetos).

3.1 Formulação do problema

O problema enfoca a movimentação de materiais entre unidades produtivas (que podem ser máquinas, prédios de uma mesma empresa ou entre empresas) – que serão designadas como *nós* – através de transportadores de alta flexibilidade – VATs. Sempre que existe a necessidade de transporte, o sistema produtivo (que não está sendo abordado neste trabalho) comunica esta necessidade a uma unidade de controle, informando o *nó* de origem e o *nó* de destino do material, e esta designa um transportador para realizar a movimentação do material (figura 2). Na figura 3 ilustra-se uma representação de várias unidades produtivas – *nós* – e os possíveis caminhos entre elas – *arcos*.

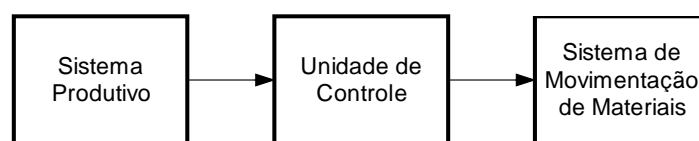


Figura 2. Fluxo de informações.

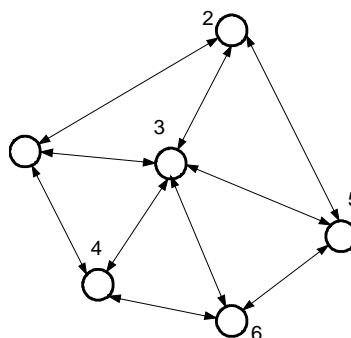


Figura 3. Modelo da malha de movimentação.

Nas próximas seções serão abordados os modelos dos elementos constituintes deste sistema: os *nós* e os transportadores.

3.2 Modelagem orientada para processos

Nesta modelagem, os modelos dos *nós* e dos transportadores são tradicionalmente desenvolvidos separadamente e interligados através de *portas* e *arcos de sinal de saída* (do E-MFG). Os transportadores são modelados por *marcas* que percorrem a malha de movimentação e, conseqüentemente, passam pelos *nós*, e possuem *atributos* para a sua identificação.

O modelo da figura 4 ilustra a representação de um *nó* da malha de transporte, onde o transportador pode estar:

- passando pelo *nó*;
- carregando um material no *nó* para ser transportado;
- descarregando um material que acabou de ser transportado.

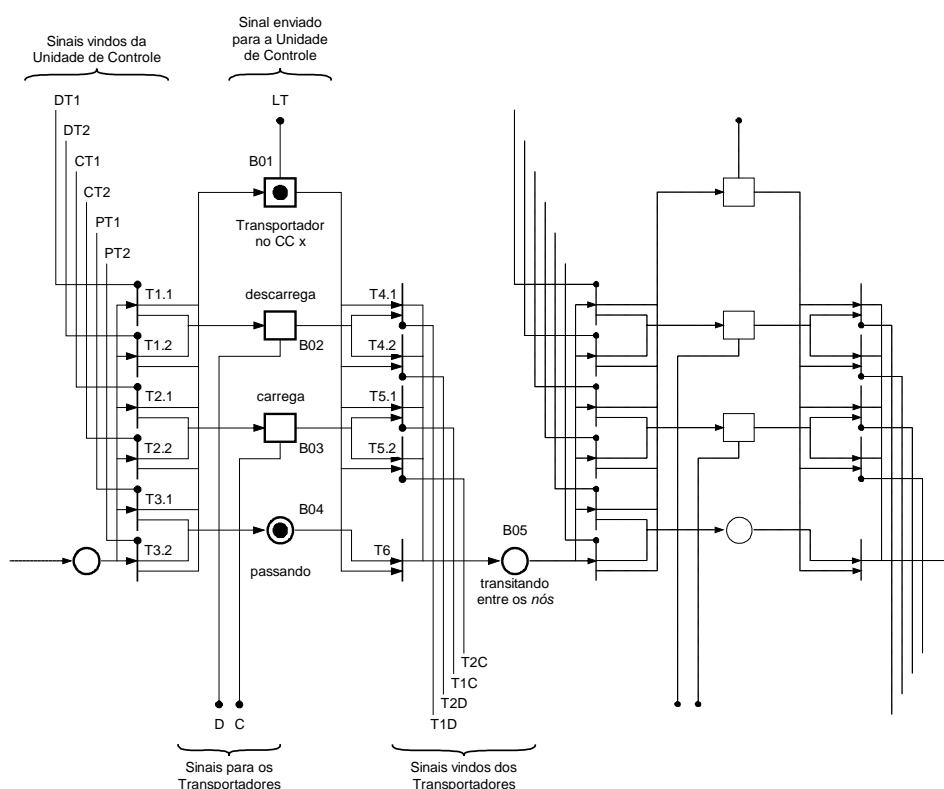


Figura 4. Representação da malha de movimentação utilizando-se o E-MFG, destacando-se a representação de um *nó* da malha.

Na ilustração da figura 4 as *marcas* indicam para a unidade de controle a localização do transportador e o que ele está fazendo – neste caso, passando pelo *nó*. As *transições* T1.i, T2.i e T3.i (onde *i* indica o número de transportador, ou seja, para *n* transportadores existirá *n* *transições* do tipo T1, *n* do tipo T2 e *n* do tipo T3) são habilitadas pela unidade de controle e são responsáveis, respectivamente, por: operação em que o transportador deixa o material sendo transportado; operação que o transportador carrega um material para ser transportado e operação que transportador passa direto pelo *nó*. Já as *transições* T4.i e T5.i são *transições* que disparam quando as respectivas operações de descarga e carga forem concluídas (mais

uma vez, n transportadores implicam em n transições T4 e n transições T5). Esta característica pode ser observada na figura 6.

O mesmo acontece no modelo dos transportadores (figura 5)– o número de transições T01.i, T02.i, T03.i, T04.i, T05.i e T09.i é proporcional ao número de nós presentes na malha de movimentação (figura 7).

Esta abordagem permite que as inter-relações entre os modelos sejam evidenciadas, porém, sempre que se necessitar de uma mudança como a inclusão de um novo transportador ou mesmo um novo nó à malha, é necessário alterar todas as relações entre os componentes. Este inconveniente motivou a adoção da modelagem híbrida a seguir descrita.

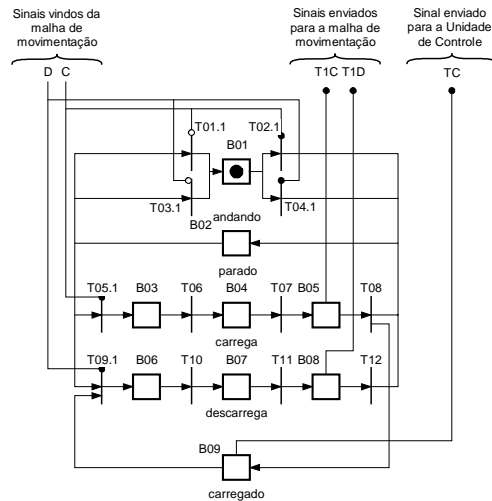


Figura 5. Representação do elemento transportador.

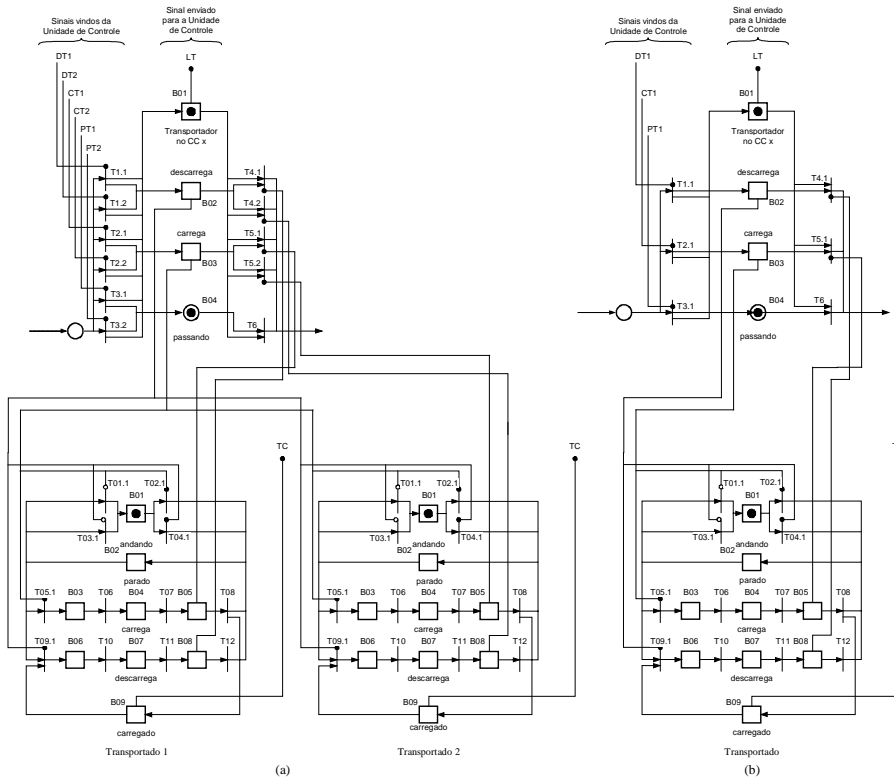


Figura 6. Exemplo de mudanças necessárias na representação dos nós ao se variar o número de transportadores: (a) o modelo com 2 transportadores; (b) o modelo com um transportador.

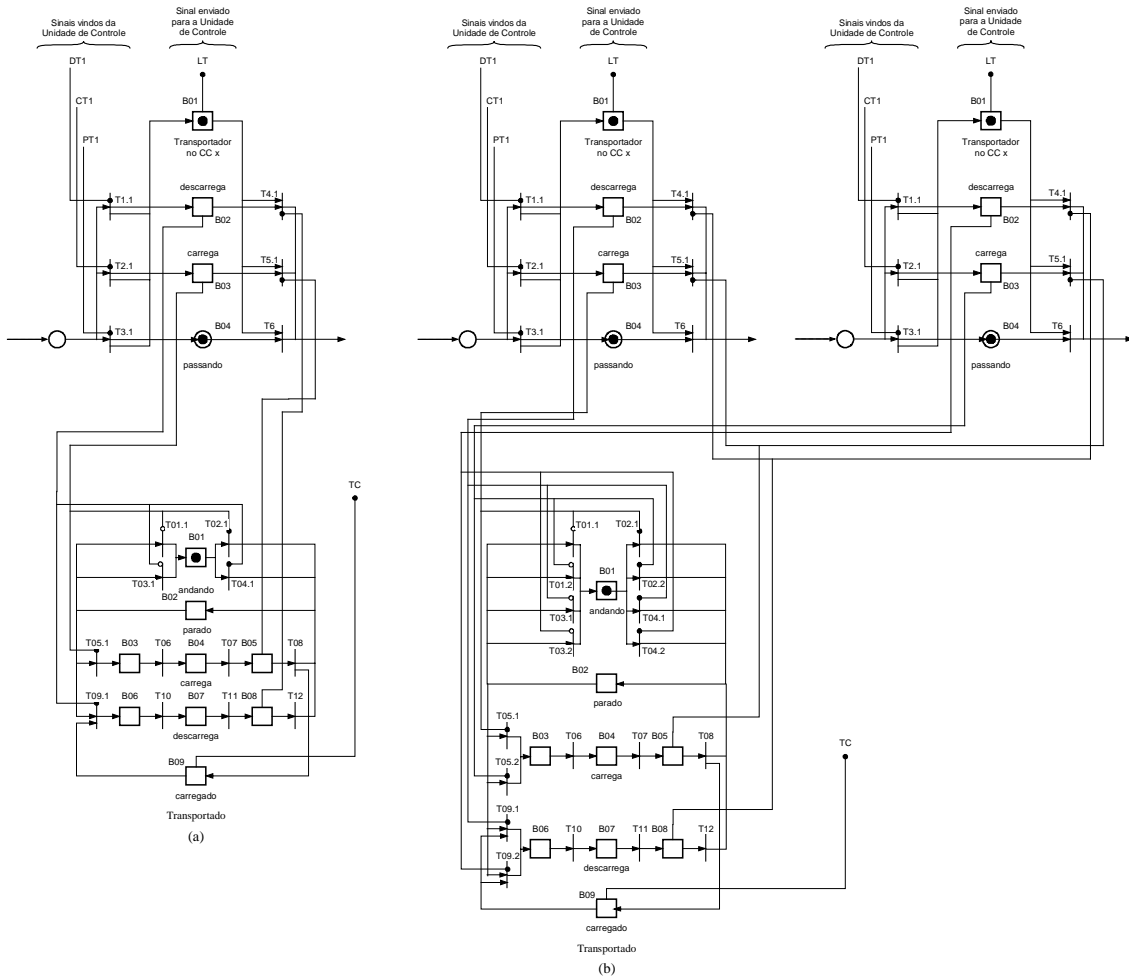


Figura 7. Exemplo de mudanças necessárias na representação dos transportadores ao se variar o número de nós: (a) o modelo com um nó e (b) o modelo com dois nós.

3.3 Modelagem híbrida (orientada para processos e para objetos)

Nesta abordagem apenas as interações entre a unidade de controle e os nós é feita através de *arcos de sinal de saída* e *portas*. As interações entre os nós e os transportadores passam a ser realizadas através de *box controladores* que alteram os *atributos* da *marca* e, conseqüentemente, interagem com o modelo do transportador.

Para tanto, a *marca* (figura 8), ao invés de apenas identificar o transportador, passa a ser composta de dois *macro atributos*: “instrução” e “transportador”.

$$\langle \text{marca} \rangle \equiv \langle \langle \text{instrução} \rangle, \langle \text{transportador} \rangle \rangle$$

Estes *atributos* em conjunto com filtros nos *arcos* orientados eliminam a necessidade de um novo conjunto de *transições* para cada novo transportador inserido no modelo como acontecia na modelagem orientada para processos (figura 4).

O *macro atributo* “instrução” é composto de 3 *atributos*:

- origem – indica o nó onde o transportador deverá pegar o material;
- destino – indica o nó onde se deve deixar o material;

- rota a ser tomada – é uma lista de *nós* indicando a trajetória que o transportador deve realizar.

$\langle \text{instrução} \rangle \equiv \langle \text{origem, destino, rota a ser tomada} \rangle$

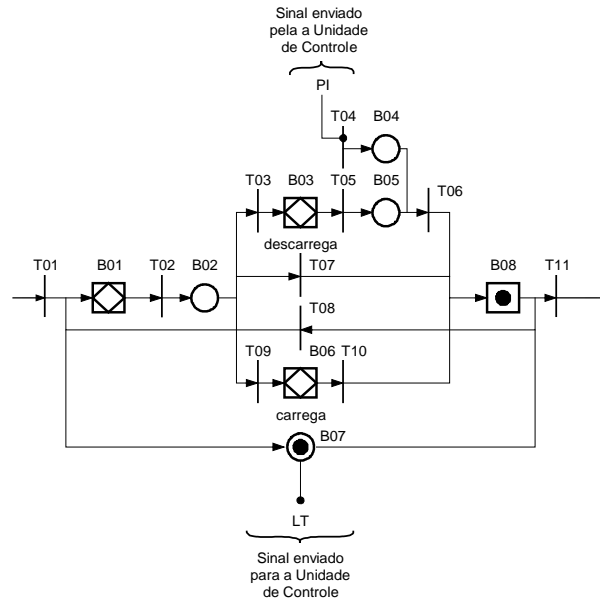


Figura 8. Nova representação de um *nó* da malha de movimentação.

Sempre que um transportador descarrega um material, este fica aguardando, em B05 (figura 8), uma nova requisição de transporte. O *macro atributo* “instrução” é passado pela unidade de controle para o *nó* através de uma *porta* – PI – e deste para o transportador através de um dos filtros nos *arcos orientados* que chegam na forma de porta habilitadora à *transição* T06 (figura 8), fazendo com que haja a combinação dos *atributos* da *marcas* presentes nos *boxes* B04 e B05, originando como resultado uma *marca* em B08.

O modelo do transportador, visto na figura 5, também sofre alterações (figura 9).

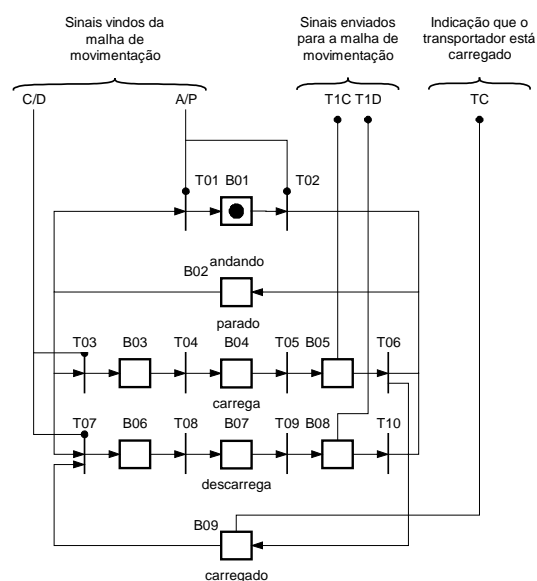


Figura 9. Nova representação do elemento transportador.

Assim, não mais é necessário um conjunto de *transições* para cada *nó* da malha de movimentação – como no modelo da figura 5 – pois o grafo (figura 9) passa a ser encapsulado em uma *marca* e seus *arcos de sinal de saída* e *portas* passam a ser vinculados aos *atributos* do *macro atributo* “transportador”. Desta forma, as *portas* (C/D e A/P) são habilitadas pelos *box-controladores* B03 ou B06 (figura 8).

$$\langle \text{transportador} \rangle \equiv \langle \text{Transp. i, C/D, A/P, T1C, T1D, TC, grafo} \rangle$$

4. COMENTÁRIOS FINAIS

A proposta de encapsular modelos E-MFG (tem-se neste caso o transportador) e associar seus *arcos de sinal de saída* e *portas* (*gates*) com os *atributos* de uma *marca*, ou seja, inserir um grafo E-MFG dentro da *marca* faz com que as conexões anteriormente necessárias através de *arcos de sinal de saída* e *portas* passem a ser associadas à alteração de *atributos* da *marca* utilizando-se *box controladores* e o fluxo das *marcas* passa a ser direcionado pelos *atributos* e filtros contidos nos *arcos orientados*.

Esta nova abordagem permite que vários transportadores sejam incluídos ao modelo sem a necessidade de se incluir novos *arcos de sinal de saída* e/ou *portas*, bastando para isto definir uma nova *marca* contendo o objeto previamente encapsulado. Algo semelhante pode ser feito com os *nós* que passam a ter que ser conectados com os demais *nós* e com o sistema de controle. Disponibiliza-se assim uma técnica onde os usuários podem ter maior flexibilidade e agilidade no trato com o modelo, possibilitando, por exemplo, a simulação de desempenho do sistemas com *n* transportadores, sem se ter a necessidade de grandes alterações do modelo – apenas a inclusão de uma nova *marca*.

Outra característica que o modelo passa a apresentar é o conceito de herança das linguagens orientadas a objeto pois se necessitar considerar outras características para o transportador como sua capacidade de reboque, basta inserir um novo atributo à classe transportador definindo-se esta capacidade.

5. AGRADECIMENTOS

Os autores agradecem o apoio dado pela FAPESP e pelo CNPq ao presente trabalho.

6. REFERÊNCIAS

- Cantù, M., 1995, Dominando o Delphi – “A Bíblia”. São Paulo. Makron Books do Brasil Editora Ltda.
- Hasegawa, K.; et al., 1988, Application of the Mark Flow Graph to Represent discrete Event Production Systems and System Control. Transactions of the Society of Instrument and Control Engineers, SICE, Tokyo, Japão, v.24, n.1, p.69-75, ISSN 0453-4654.
- Inamasu, R. Y., 1995, Modelo de FMS: Uma Plataforma para Simulação e Planejamento. Tese de Doutorado, Escola de Engenharia de São Carlos da Universidade de São Paulo.
- Miyagi, P.E., 1985, A Study on MFG Based Programming Method for Robots. Dissertação de Mestrado, Tokyo Institute of Technology, Tokyo.
- Miyagi, P.E., 1996, Controle Programável - Fundamentos do Controle de Sistemas a Eventos Discretos. Editora Edgard Blücher, São Paulo.
- Reisig, W., 1992, A Primer in Petri Design. Springer-Verlag, Berlin.
- Santos Filho, D.J., 1993, Proposta do Mark Flow Graph Estendido para a Modelagem e Controle de Sistemas Integrados de Manufatura. Dissertação de Mestrado, Escola Politécnica da USP, São Paulo.

- Santos Filho, D. J. & Miyagi, P. E., 1995, Enhanced Mark Flow Graph to Control Flexible Manufacturing Systems. *Jornal of The Brazilian Society of Mechanical Sciences*, ABCM, Rio de Janeiro, RJ, v.17, n.2, p.232-248, ISSN 0100-7386.
- Santos Filho, D.J., 1998, *Sistemas Antropocêntricos de Produção Baseado em Redes de Petri Interpretadas*. Tese de Doutorado, Escola Politécnica da USP, São Paulo.

TRANSFERÊNCIA DE CALOR E PERDA DE CARGA EM FLUIDOS PSEUDOPLÁSTICOS TERMODEPENDENTES

Ivan Azevedo Cardoso

Michel Lebouche

Rene Devienne

Pontifícia Universidade Católica do Paraná

Faculdade de Engenharia Mecânica

Caixa Postal 5030; CEP 80061-990 Curitiba PR

E-mail: ivanac@rla01.pucpr.br

Resumo

Apresentamos resultados da transferência de calor e perda de carga para fluidos pseudoplásticos em escoamento laminar no interior de tubos cilíndricos. Foi utilizado o modelo da lei de potência de Ostwald para caracterizar os fluidos. Estes fluidos são resfriados em um trocador de calor bitubular por um outro fluido em escoamento turbulento externo em contracorrente e a baixa temperatura. Foi desenvolvido um método de forma a determinar os valores dos coeficientes locais de troca de calor. As variações dos números de Nusselt associados podem ser aproximadas usando a abscissa adimensional $X_+ = 2z/(D.Pe)$. Os dados experimentais mostram que os números de Nusselt são fortemente dependentes da diferença entre a temperatura de entrada do fluido pseudoplástico T_e e as temperaturas da parede $T_w(z)$. Este último fenômeno, atribuído às variações dos índices de consistência K e reológico n com a temperatura, pode ser mostrado pelas evoluções dos perfis de velocidades axiais que são determinados pelo efeito Doppler pela anemometria laser. Uma correlação muito simples aparece como aceitável em nossa faixa de variação experimental. Comparações com um código numérico também são propostas.

Palavras chave: Fluidos não newtonianos; Reologia; Transferência de Calor; Termodependência.

1. INTRODUÇÃO

É bem conhecido que quando há envolvimento de fluidos não-newtonianos o projeto de trocadores de calor não é uma operação simples. Este problema está primariamente ligado às indústrias de alimentos ou químicas quando os produtos a serem tratados podem exibir comportamento reológico complexo. Além do mais, tais fluidos como extrato de tomate e purês de frutas geralmente apresentam alta viscosidade aparente de tal forma que aparecem situações para as quais o regime térmico nunca é estabelecido dentro do trocador de calor. Isto é particularmente válido no caso do resfriamento com um aumento da complexidade. Projetistas de plantas térmicas podem achar muitos estudos experimentais e teóricos na literatura. Ver por exemplo, os seguintes artigos: Oliver e Jenson (1964), Christiansen *et al.*(1966), Forest e Wilkinson (1977), Popovska e Wilkinson (1977). Estes autores analisaram o efeito de tal

fenômeno como:

- (1) termodependência, também extensivamente tratada no caso de aquecimento, ver, por exemplo, Sieder e Tate (1936), Metzner *et al.*(1957), Joshi e Bergles (1982);
- (2) dissipação viscosa, que pode ser de grande importância em processos de resfriamento devido ao fato de que nessas situações a transferência de energia é geralmente fraca;
- (3) convecção natural.

Apresentamos alguns resultados ligados aos coeficientes de transferência de calor entre um fluido pseudoplástico, chamado de fluido de trabalho, e uma parede resfriada por um fluido newtoniano. A geometria testada corresponde aos escoamentos em contracorrente no interior de um tubo horizontal para o fluido de trabalho e dentro de um espaço anular concêntrico para o fluido de resfriamento. O fluxo de calor e a temperatura de parede variam ao longo do tubo. Este estudo pode fornecer resultados práticos para os trocadores de calor.

2. DESCRIÇÃO DO EXPERIMENTO

2.1 Instalação experimental

É constituída essencialmente de um sistema que permita o escoamento de fluido quente em um circuito fechado (veja fig. 1). Este escoamento é impulsionado por uma bomba helicoidal, que foi escolhida para evitar excessivo cisalhamento. Um tubo de seção inicial de 47 diâmetros foi colocado à entrada, de forma a permitir a estabilização do perfil de velocidade para qualquer vazão, esta sendo medida por um medidor eletromagnético. O comprimento de entrada hidrodinâmica pode ser avaliado pela equação: $L/D \approx Re_g/10$ e, de fato, nossos consistentes produtos levam a baixos números de Reynolds (<30) de tal forma que o campo e velocidades se desenvolve muito rapidamente. Re_g é o número de Reynolds generalizado definido por $Re_g = 1/\Delta^n \rho V_d^{2-n} D^n / 8^{n-1} K$ onde Δ é a correção do gradiente de velocidade na parede definido por $\Delta = (3n + 1)/4n$, V_d é a velocidade média axial e ρ é a massa específica.

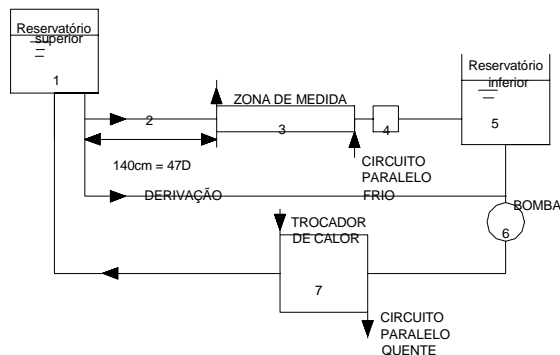


Figura 1. Esquema Geral da Instalação

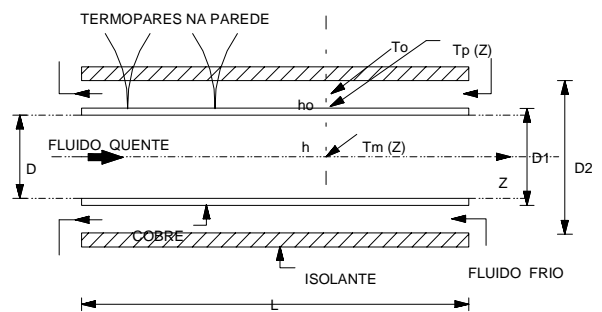


Figura 2. Detalhes da Zona de Medida

A temperatura média de entrada T_e pode ser controlada por um trocador de calor, e é medida por um termopar com precisão de $\pm 0,1$ K. A região observada é composta essencialmente de um tubo de cobre horizontal de 3,2 m de comprimento, através do qual flui o líquido quente (ver fig. 2). Foram inseridos 40 termopares Chromel-Alumel ao longo da parede do tubo, que foram então conectados a uma central de medidas, alguns dos quais foram fixados acima e abaixo na mesma seção para controlar a eventual ocorrência de convecção natural. O

líquido de resfriamento externo (uma mistura de água e etileno glicol) flui em uma seção anular ($D_1 = 35$ mm, $D_2 = 72$ mm). Devido a sua composição química a temperatura média do fluido frio pode chegar a -10 °C. A entrada e a saída deste fluido é feita por três guias radiais. As temperaturas de entrada e saída do líquido frio foram medidas com termômetros de platina capazes de medir essas diferenças tipicamente pequenas de temperaturas. A medida da vazão deste líquido frio permite-nos estimar a potência total trocada. De fato, a temperatura média de entrada e a vazão permanecem constantes (ver tabela 1, que indica também as características principais da instalação).

Os perfis de velocidade axial na entrada e na saída são medidos por anemometria Doppler.

Tabela 1. Principais Características da Instalação

Diâmetro interno do tubo	$D = 0,034$ m	Número total de termopares	41
Comprimento da zona resfriada	106.D	Número de Reynolds	$0 \rightarrow 30$
Vazão do fluido quente	$Q = 0 \rightarrow 2,2$ m ³ /h	Vazão do fluido frio	2,2 m ³ /s
Velocidade média	$V_d = 0 \rightarrow 0,47$ m/s	Temperatura de entrada do fluido quente	$18 \rightarrow 52$ °C
Potência total trocada	$\phi = 0 \rightarrow 3,1$ kW	Temperatura de entrada do fluido frio	$-6,5$ °C

2.2 Fluidos testados

Foram utilizadas soluções aquosas de carboximetilcelulose (c.m.c.) com uma concentração de 4 % em peso. O comportamento reológico foi medido por meio de um reômetro com tensão controlada equipado com geometria cone-prato (diâmetro = 4 cm, ângulo = 4°). Os diagramas da figura 3 mostram claramente que a lei de potência $\tau = K \dot{\gamma}^n$, onde τ é a tensão de cisalhamento, somente pode ser aplicada em certas faixas de gradientes de velocidades $\dot{\gamma}$ que devem ser especificadas - uma lei de potência exata corresponderia a linhas retas. Na presente situação, restringimos nossa faixa para: $10 - 150$ s⁻¹, que serão os valores limites considerados em nosso experimento.

As dependências da temperatura do índice de consistência K e do índice reológico n foram medidas com exatidão (ver fig. 4). No domínio de temperatura considerado aqui, a lei exponencial parece ser suficiente (ver fig. 4), que é: $K = a \exp(-bT) = 42,2 \exp(-0,049T)$; $n = a' \exp(b'T) = 0,43 \exp(0,0096 T)$. Note que, contrariamente ao caso usual, no presente estudo, é necessário considerar a variação de n com a temperatura, devido a que as formas dos perfis de velocidade na entrada dependem do índice n (ver na subsecção comparação com resultados numéricos).

2.3 Avaliação dos coeficientes locais de troca de calor

O coeficiente de troca de calor para o fluido pseudoplástico $h = h(z)$ é definido por:

$$h(z) = \varphi(z) / \{T_m(z) - T_w(z)\} \quad (1)$$

onde $\varphi(z)$ é a densidade de fluxo local de calor para a superfície interna do tubo de cobre. $T_m(z)$, $T_w(z)$ são, respectivamente, as temperaturas média e de parede (ver fig 2) com:

$$T_m(z) = T_e + \frac{1}{\rho \cdot c_p Q_0} \int_0^z \varphi(z') \cdot \pi \cdot D \cdot dz' \quad (2)$$

Para o fluido de resfriamento externo, podemos escrever:

$$h_0 = \varphi'(z) / \{T_w(z) - T_0(z)\} \quad (3)$$

sendo $\varphi'(z)$ a densidade de fluxo de calor local para a superfície externa do tubo de cobre.

O fluxo de calor axial no interior da parede pode ser negligenciado, de tal forma que $D_1 \cdot \varphi'(z) = D \cdot \varphi(z)$. Aqui, $T_0(z)$ é a temperatura média do fluido frio. Isto conduz a:

$$h(z) = h_0 \cdot (D_1/D) \cdot [T_w(z) - T_0(z)] / [T_m(z) - T_w(z)] \quad (4)$$

O fluido de resfriamento sofre apenas pequenas variações de temperatura, e devido ao regime turbulento de escoamento, o coeficiente h_0 pode ser assumido como constante ao longo do tubo e pode ser estimado de um balanço de energia na potência total de calor trocada ϕ :

$$\phi = \int_0^L \varphi'(z) \cdot \pi \cdot D_1 \cdot dz = h_0 \int_0^L \{T_w(z) - T_0(z)\} \pi \cdot D_1 \cdot dz \quad (5)$$

com $\phi = \rho \cdot C_{p0} \cdot Q_0 \cdot (T_{o,0} - T_{i,0})$, obtido por um balanço de calor da água fria. Aqui Q_0 , $T_{o,0}$, $T_{i,0}$ são, respectivamente, a vazão e as temperaturas de entrada e saída para o fluido frio: $T_{o,0} = T_0(0)$, $T_{i,0} = T_0(L)$. Os resultados são representados por meio do número de Nusselt local, definido como $Nu = h \cdot D / \lambda$. λ , a condutividade térmica, sendo tomada como igual à da água. C_p é o calor específico do fluido de trabalho, e C_{p0} é o calor específico do fluido frio.

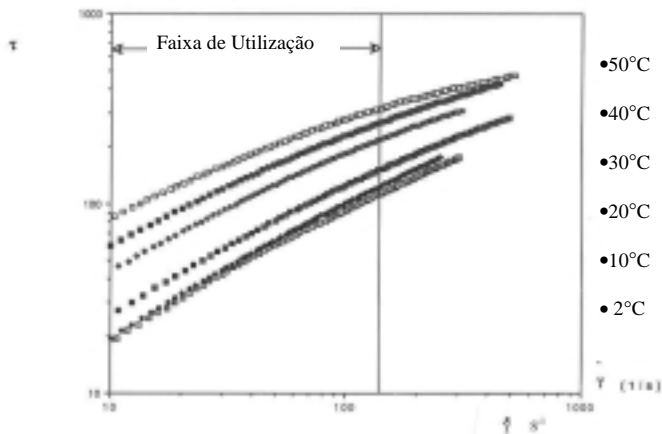


Figura 3. Variações da tensão de cisalhamento τ em função do gradiente de velocidade $\dot{\gamma}$ para diferentes temperaturas

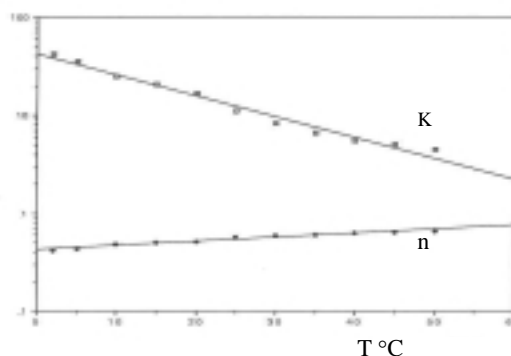


Figura 4. Evolução dos índices de consistência K e n com a temperatura.

3. RESULTADOS EXPERIMENTAIS

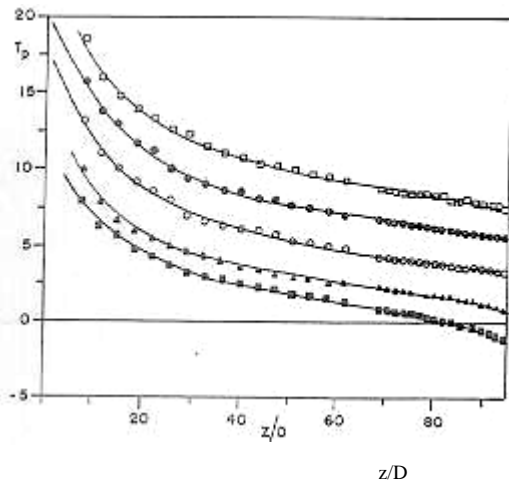


Figura 5. Temperatura de parede T_w em função da abscissa reduzida z/D para vazões diferentes e uma temperatura média de entrada $T_e = 35 \text{ C}$ (\square 1200 l/h $Re_g=12,1$; \bullet 750 l/h $Re_g=6,2$; \circ 450 l/h $Re_g= 3,1$; \blacktriangle 150 l/h $Re_g=0,7$; \blacksquare 75 l/h $Re_g= 0,2$)

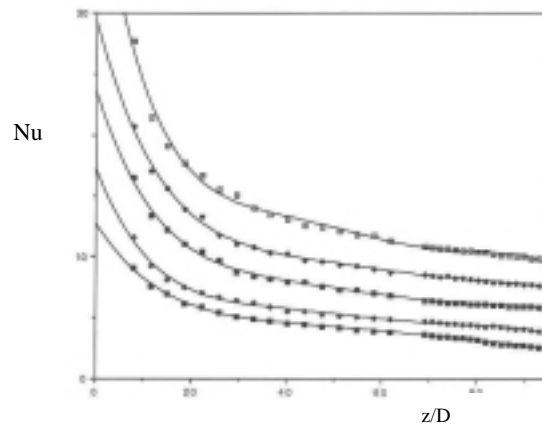


Figura 6. Número de Nusselt em função da abscissa axial z/D para vazões diferentes e uma temperatura média de entrada $T_e= 52 \text{ C}$ (\square 1200 l/h $Re_g= 18,7$; \bullet 750 l/h $Re_g=10$; \blacksquare 450 l/h $Re_g= 5,2$; \triangle 150 l/h $Re_g= 1,3$; \blacktriangle 75 l/h $Re_g= 0,5$)

A fig. 5 ilustra um exemplo de evolução da temperatura de parede; estas variações parecem ser grandes; além disso, é possível distinguir um decréscimo da inclinação quando a temperatura cai abaixo do ponto de congelamento. A fig. 6 mostra um exemplo das variações dos números de Nusselt, e indica claramente que estamos numa situação de comportamento assintótico. Podemos considerar as variações da posição axial e da vazão utilizando a variável usual $X_+ = 2.z / (D.Pe)$, uma abscissa axial adimensional (ver fig. 7). Note que o número de Peclet ($Pe = \rho V_d D C_p / \lambda$) não contém informação a respeito do comportamento reológico e que esta análise permanece razoável para todos os valores das temperaturas de entrada T_e .

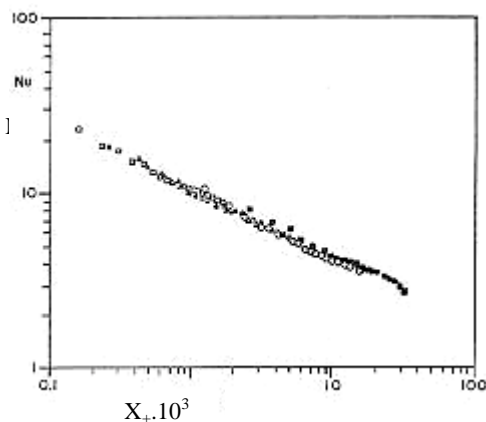


Figura 7. Número de Nusselt em função de $X_+=2z/D/Pe$ (representação logarítmica) para uma temperatura média de entrada $T_e=52^\circ\text{C}$ (\square 1200 l/h $Re_g=18,7$; \blacktriangle 750 l/h $Re_g=10$; \circ 150 l/h $Re_g=1,3$; \blacksquare 75 l/h $Re_g=0,5$).
Por outro lado, deve ser notado que as v

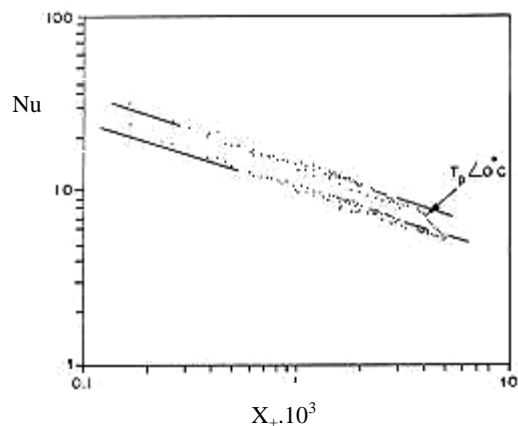


Figura 8: Número de Nusselt em função de $X_+=2z/D/Pe$ (representação logarítmica) para diferentes temperaturas médias de entrada T_e e uma vazão fixa (1200 l/h) $\blacksquare T_e= 52^\circ \text{ C}$, $n= 0,68$; $\circ T_e= 35^\circ \text{ C}$, $n= 0,60$; $\bullet T_e= 18^\circ \text{ C}$, $n=0,51$

dependentes desta temperatura de entrada T_e (ver fig. 9). Este efeito pode ser atribuído à termodependência. O valor do expoente de X_+ foi achado como sendo 0,36, sendo este o valor médio para todos os pontos experimentais. Ele é muito próximo do valor de $1/3$ achado no trabalho de Levêque (1928), $Nu = 1,41 (Gz)^{1/3}$ onde Gz é o número de Graetz definido por $Gz = \pi/4 Pe/z/D$.

3.2 O modelo.

Como usual, a correlação acima foi ajustada multiplicando-a por dois fatores de correção. O primeiro fator, escrito como $\Delta^{1/3} = \{(3n+1)/(4n)\}^{1/3}$ descreve o caráter não-newtoniano do fluido e foi introduzido por Pigford (1955). Para obter uma equação prática, nós escolhemos n para corresponder à temperatura de entrada. Esta correção está na maioria em 7%, que é, portanto, insuficiente para reagrupar todas as curvas de Nusselt em uma. Isto é explicado pelo fato que o efeito negativo ($T_e \uparrow, n \uparrow, \dot{\gamma}_w \downarrow$) adiciona a outro efeito negativo causado pelo aumento progressivo da viscosidade aparente. Este aumento é causado pelo resfriamento ao longo da parede, que conduz a uma variação no gradiente de velocidade (como mostrado posteriormente na subsecção Comparação com os resultados numéricos). Esta variação torna-se maior à medida que a temperatura de entrada T_e aumenta.

Torna-se necessário, portanto, um segundo fator que é relacionado com as variações da temperatura de parede e mais especificamente com a termodependência. No presente estudo, não temos nem o fluxo parietal constante nem uma temperatura de parede constante. Podemos considerar que a temperatura média não varia muito ao longo de nosso conduto. Escreve-se: $K_m/K_w \approx K_e/K_w$, onde o índice de consistência K_e é à temperatura de entrada do fluido. Um raciocínio similar já foi utilizado, por exemplo, por Metzner e Gluck (1959), Mizushina *et al.* (1967), Mahalingham *et al.* (1975). Das observações acima, propomos uma correlação única que engloba toda nossa gama de resultados (total de 450 pontos) excluindo os pontos em que existem os efeitos da congelção:

$$Nu = 1,15 [(3n + 1)/4n]^{1/3} \cdot [K_m / K_w]^{0,16} (X_+)^{-0,36} \quad (4)$$

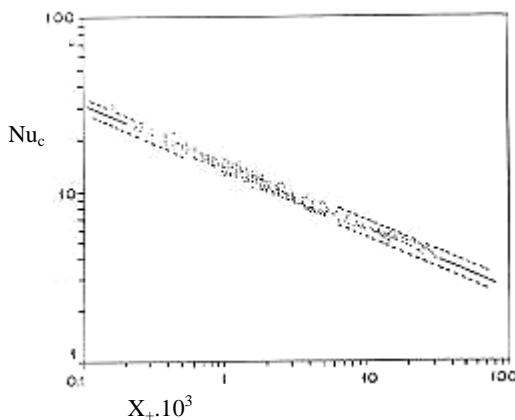


Figura 9. Variações globais do número de Nusselt corrigido $Nu_c = Nu / \{ [K_m/K_w]^{0,16} \}$ em função de $X_+ = 2z/D/Pe$; todos os pontos experimentais; - - correlação proposta (inclinação de $-0,36$); largura da faixa =

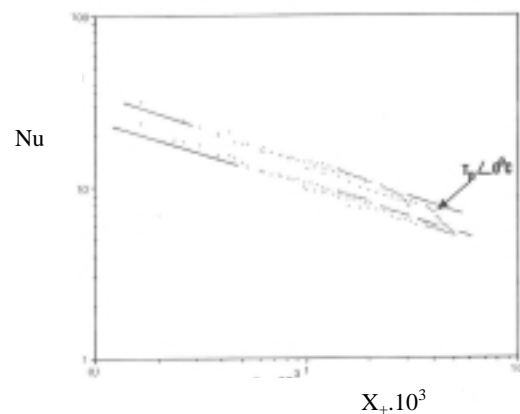


Figura 10. Comparação entre os resultados experimentais e numéricos; . pontos experimentais; - - curvas numéricas (na superior $T_e = 52$ C, a inferior $T_e = 18$ C)

3.3 Comparação com os resultados numéricos

Foi utilizado um programa numérico desenvolvido por El Ouardighi *et al.*(1989), que resolve o sistema de equações dos campos dinâmico e térmico e levou-se em consideração as variações dos parâmetros com a temperatura. Foi possível introduzir a distribuição de temperaturas de parede resultante dos experimentos. Neste caso, o programa fornece a densidade de fluxo de calor local $\phi(z)$. Uma comparação pode ser feita com as curvas de Nusselt obtidas experimentalmente. A concordância parece ser boa (ver fig. 10), com exceção dos pontos correspondentes a $T_w < 0$ °C para os quais aparece o congelamento. A determinação dos perfis axiais de velocidade na saída leva-nos a uma segunda comparação e mostra o seguinte efeito já observado por Scirocco *et al.*(1985): a queda na temperatura de parede causa um aumento na viscosidade aparente perto da parede, que portanto muda a estrutura do escoamento pela introdução da componente da velocidade radial na direção do centro do cilindro.

As figs. 11 e 12 mostram que esta variação é muito mais importante para altas temperaturas de parede, que explica a influência negativa da temperatura de entrada T_e nos valores dos números de Nusselt.

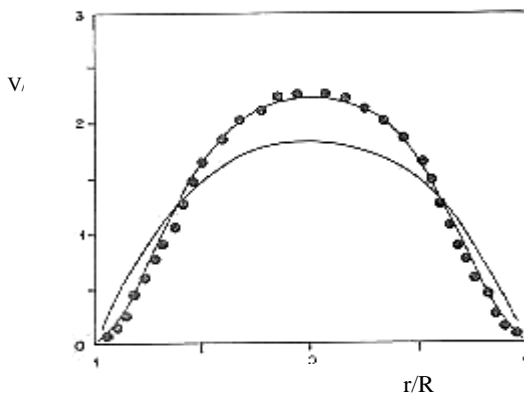


Figura 11. Evolução da velocidade axial V/V_d (para $T_e = 52$ C, 1200 l/h); determinações numéricas; • valores experimentais na saída

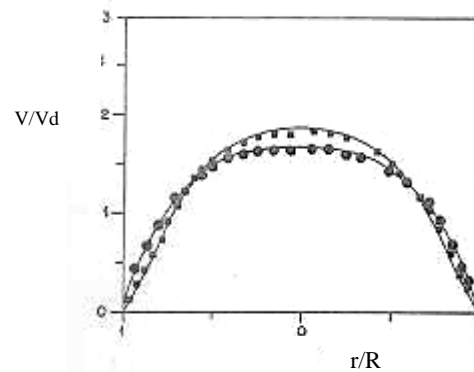


Figura 12. Evolução da velocidade axial V/V_d (para $T_e = 35$ °C, 1200l/h); determinações numéricas; • valores experimentais na entrada; ■ valores experimentais na saída

3.1 Perdas da Carga

O conhecimento das leis de perda de carga nos condutos industriais é muito importante para o dimensionamento das instalações. Para os fluidos não-termodependentes, o cálculo das perdas de carga em tubos se obtém formalmente, ao menos para o escoamento estabelecido. O mesmo não acontece para os fluidos termodependentes. Nesta parte veremos uma aplicação de nosso programa numérico estudando a influência da termodependência sobre a perda de carga.

Em geral associamos ao gradiente de pressão local o coeficiente de atrito local Cf_{iso} (caso isotérmico) definido por: $Cf_{iso} = 2 |\tau_p| / [\rho (V_d)^2]$. Utilizando o perfil de velocidade para um fluido de Ostwald em regime dinâmico estabelecido para um determinado gradiente de pressão deduz-se facilmente que $Cf_{iso} = 16 / Re_g$.

Em presença de resfriamento, os fluidos termodependentes apresentam um aumento de suas viscosidades ou de suas consistências. A resistência ao escoamento é então mais forte e resulta uma perda de carga superior àquela que teríamos se o fluido não fosse termodependente. Então

a fração entre o gradiente de pressão e o gradiente de pressão no caso isotérmico é superior a 1. Como no caso isotérmico, definindo o coeficiente de atrito local não-isotérmico C_f , representa-se então a evolução da fração ($C_f/C_{f_{iso}}$) ao longo do conduto: $C_f/C_{f_{iso}} = (dp/dx)/(dp/dx)_{iso} > 1$. Nas figuras 13 e 14 apresentamos os resultados numéricos da perda de carga considerando os dois casos de condições limites puras: fluxo de calor constante na parede ou temperatura de parede constante. As figuras mostram a evolução do coeficiente de atrito ao longo do conduto. Nota-se à medida que se penetra na zona resfriada, as perdas de carga aumentam. Com efeito, uma diminuição da temperatura parietal aumenta a viscosidade aparente, aumentando a resistência ao escoamento, provocando um crescimento do coeficiente local de atrito. Nota-se também que este efeito é mais pronunciado no início da zona resfriada; este resultado esperado se explica pela presença de fortes gradientes térmicos nesta zona. As mesmas figuras mostram também a influência da potência de resfriamento sobre as perdas de carga (curva a, b). Assim a um dado número de Reynolds, um aumento da densidade de fluxo (fig. 13) ou uma diminuição da temperatura de parede (fig. 14) provoca um aumento do atrito e conseqüentemente perdas de carga mais importantes. Isto se explica pelo crescimento da viscosidade aparente. Finalmente, as mesmas figuras mostram a influência do número de Reynolds sobre as perdas de carga (curvas b,c). Assim à densidade de fluxo constante (fig.13) ou à temperatura de parede constante (fig.14), um aumento do número de Reynolds leva a uma diminuição da perda de carga; isto se explica pelo aumento da componente da velocidade axial que reduz o efeito do resfriamento.

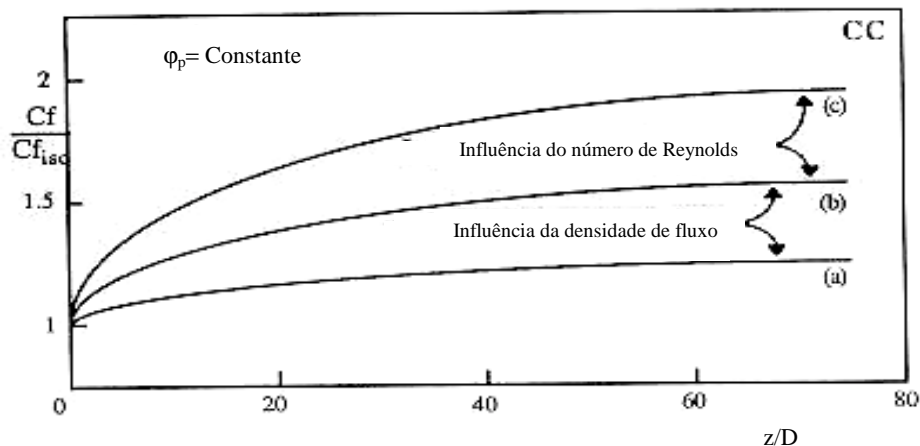


Figura 13. Evolução do coeficiente de atrito ao longo do conduto.

1°) Influência da densidade de fluxo $Re_g = 100$

a) $\phi_p = 5 \text{ KW/m}^2$

b) $\phi_p = 10 \text{ KW/m}^2$

2°) Influência do número de Reynolds:

$\phi_p = 10 \text{ KW/m}^2$

b) $Re_g = 100$

c) $Re_g = 50$

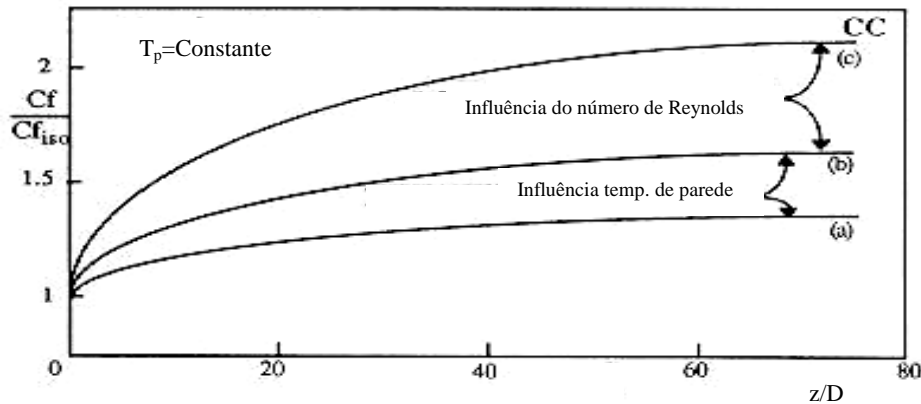


Figura 14 . Evolução do coeficiente de atrito ao longo do conduto ($T_e = 50^\circ\text{C}$)
 1°) Influência da temperatura de parede $Re_g = 50$ 2°) Influência do número de Reynolds

- a) $T_p - T_e = -20^\circ\text{C}$
 b) $T_p - T_e = -40^\circ\text{C}$

2°) Influência do número de Reynolds

- $T_p - T_e = -40^\circ\text{C}$
 b) $Re_g = 50$
 c) $Re_g = 25$

4. CONCLUSÃO

Estudamos o escoamento de um material pseudoplástico resfriado por um fluxo turbulento em contracorrente, que é de considerável interesse prático. Este estudo mostrou que é possível determinar os valores locais do coeficiente de transferência de calor. A evolução da temperatura medida e o fluxo calculado indicam que estamos longe da hipótese comum de que estas quantidades permanecem constantes. Além do mais, parece necessário levar em consideração a interação entre os campos térmico e dinâmico, que conduz a uma análise mais complexa. Usando um código numérico existente no laboratório pudemos comparar os resultados experimentais com os numéricos e estender nosso estudo para a perda de carga.

5. REFERÊNCIAS

- Christiansen, E. B., Jensen, G. E., and Tao, F. S. 1966. "Laminar flow heat transfer". AICHE J., 12, pp.1196-1202
- El Ouardighi, A., Van Tuan, N. et Lebouché, M. 1989. "Ecoulement et transfert de chaleur par convection forcée pour un fluide non Newtonien thermodépendant". C.R. Acad. Sci. Paris, pp. 91-99
- Forest, G. and Wilkinson, W. L. 1977. "Laminar Heat Transfer to Power Law Fluids in Tubes with Constant Wall Temperature". Trans. Inst. Chem. Eng., 51, pp. 331-337
- Joshi, S. D. and Bergles, a. e. 1982. "Heat transfer Laminar Flow of non-newtonian Fluids in Tubes. J. Heat Transfer", 3, pp. 51-56
- Mahalingham, R., Tilton, L. O. and Coulson, J. M. 1975. "Heat Transfer in Laminar Flow of non-Newtonian Fluids". Chem. Eng. Sci., 30, pp. 921-929

- Metzner, A. B. and Gluck, D. F. 1959. "Heat Transfer to non-Newtonian fluids under Laminar-Flow Conditions". Chem. Eng. Sci., 12, pp. 185-190
- Metzner, A. B., Vaughn, R. D. and Houghton, G. L. 1957. "Heat Transfer to non-Newtonian Fluids". AIChE J., 3, pp. 92-100
- Mizushima, T., Ito, R., Kuriwake, Y. and Yahikagawa, K. 1967. "Boundary-layer heat transfer in a circular tube to Newtonian and non-Newtonian Fluids". Kagaku-Kogaku Ronbunshu, 31, pp. 250-256
- Oliver, D. R. and Jenson, V. G. 1964. "Heat transfer to pseudoplastic fluids in laminar flow in horizontal Tubes". Chem, Eng. Sci., 19, pp. 115-129
- Popovska, F. and Wilkinson, W. L, 1977. "Laminar heat transfer to Newtonian and non-Newtonian fluids in Tubes". Chem. Eng. Sci., 32, pp. 1155-1164
- Scirocco, V., Devienne, R. and Lebouché, M. 1985. "Ecoulement et transfert de chaleur pour un fluide pseudoplastique dans la zone d'entrée d'un tube". Int J. Heat Mass Transfer, 28, pp. 91-99
- Sieder, E. N. and Tate, G. E. 1936. "Heat Transfer and Pressure Drop of Liquids in Tubes". Ind. Eng. Chem. Research. 28, pp. 1429-1435.
- Levêque, 1928, J. Ann. Mines, Paris, Vol. 13, pp. 305-381.

SOBRE A MODELAGEM E O CONTROLE DE ROBÔS MANIPULADORES

Sebastião Cícero Pinheiro Gomes

Fundação Universidade Federal do Rio Grande

Departamento de Matemática

e-mail: dmtscpg@super.furg.br; <http://www.dmat.furg.br/numa>

Resumo

A dinâmica de robôs manipuladores com vários graus de liberdade é não-linear e bastante complexa. Além dos acoplamentos entre as diversas partes (elos) do manipulador, existem efeitos dinâmicos internos aos atuadores, tais como elasticidade e atritos não-lineares, os quais contribuem sensivelmente para o acréscimo da complexidade do modelo, dificultando também a eficiência das leis de controle. A fim de avaliar estes problemas, três diferentes técnicas de controle são testadas, emitindo-se ao final, sobre cada uma delas, conclusões sobre desempenho, estabilidade e complexidade computacional.

Palavras chave: Modelagem, Atritos, Atuadores, Controle, Robustez.

1. INTRODUÇÃO

Existe uma grande quantidade de trabalhos científicos sobre teoria de controle em geral, com aplicações diversas, entre as quais a robôs manipuladores (Ferretti *et. al.*, 1998; Park *et. al.*, 1998). Em sua maioria, os modelos dinâmicos em geral são simplificados, principalmente com relação à dinâmica dos atuadores. Isto resulta num fato relativamente comum: quando se parte para alguma implementação prática, os resultados de simulação das leis de controle diferem por vezes sensivelmente dos encontrados na experimentação, sendo necessários ajustes nos parâmetros de controle, os quais são conseguidos em exaustivos testes experimentais (Armstrong, 1988). Com o objetivo de contornar este problema, desenvolveu-se um simulador baseado em modelos dinâmicos bastante realistas e comprovados experimentalmente. Trata-se do Simulador com Interface para Manipuladores (SIM, versão 1.0, <http://www.dmat.furg.br/numa>). Objetiva-se com isso diminuir diferenças entre simulações e experimentos, fato que contribui também para o desenvolvimento de leis de controle com condições reais de implementação. Aproveitando a existência do simulador, três técnicas de controle foram então implementadas e testadas, ressaltando-se ao final as principais conclusões resultantes destas simulações.

2. MODELAGEM DINÂMICA

O conhecimento de modelos dinâmicos realistas é fundamental para testar, em simulação, a eficiência de leis de controle para robôs manipuladores. O modelo dinâmico global do manipulador, considerado no presente trabalho, é constituído de uma parte relativa à dinâmica dos atuadores, e outra relativa à estrutura, conforme será explicitado a seguir.

2.1 Modelo dos atuadores

Um atuador robótico do tipo moto-reductor, pode ser considerado como um elemento transmissor de movimento com uma certa elasticidade interna, conforme a figura abaixo.

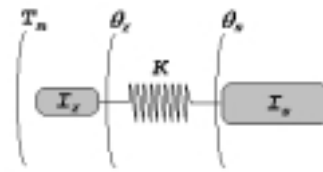


Figura 1. Atuador com um modo flexível.

I_r é a inércia do rotor, K a constante elástica interna e I_s a inércia da carga acoplada à saída do reductor. T_m , θ_r e θ_s são o torque motor e as posições angulares do rotor e carga respectivamente. Uma equação diferencial representando a dinâmica deste atuador pode ser facilmente obtida, ficando a mesma sob a forma (considera-se o atuador isoladamente, de forma que não há torques atuando sobre a saída do reductor):

$$I_r \ddot{\theta}_r + \frac{K}{n^2} (\theta_r - n\theta_s) = T_m - T_{at} \quad (1)$$

$$I_s \ddot{\theta}_s - K \left(\frac{\theta_r}{n} - \theta_s \right) = 0$$

T_{at} corresponde ao torque de atrito não-linear, considerado aqui como atuante totalmente no lado rotor e escrito sob a forma:

$$T_{at} = f_v^* \dot{\theta}_r \quad (2)$$

f_v^* é o coeficiente de atrito viscoso variável (Gomes *et al.*, 1992; Gomes, 1995), o qual é função das variáveis de estado e também do torque motor. Este modelo representa bem o fenômeno conhecido em literatura inglesa pelos nomes *stick* e *slip*, muito presente em mecanismos de transmissão de movimento. A figura 2 mostra uma simulação sobreposta a um resultado experimental. Foi aplicado um torque motor senoidal de pequena amplitude, pouco superior aos torques de atrito estático, positivo e negativo, representados por retas horizontais na figura. Pode-se visualizar claramente o modo *stick*: a velocidade rotor vai a zero e só abandona esta situação (modo *slip*) quando o torque motor torna-se superior ao atrito estático.

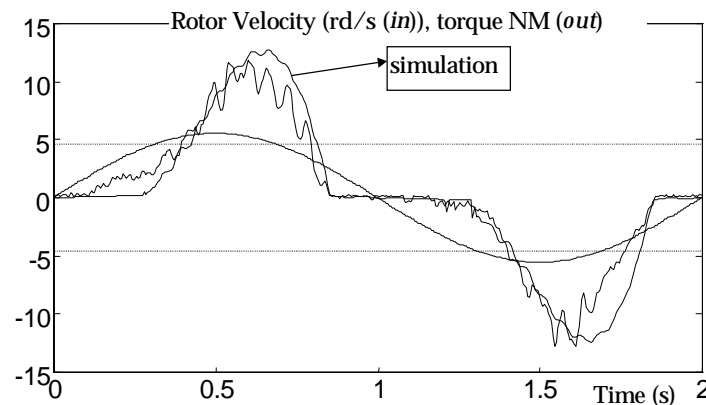


Figura 2. Simulação e resultado experimental com o *harmonic-drive* e torque motor senoidal.

2.2 Modelo estrutural

Idealizou-se um manipulador a três graus de liberdade para a realização dos testes, conforme esquema mostrado na figura seguinte .

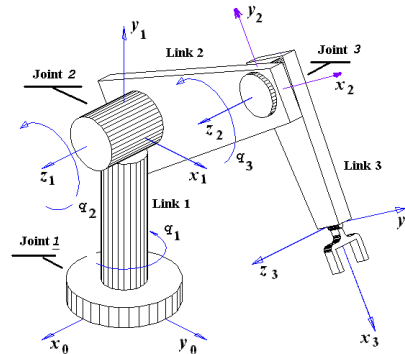


Figura 3. Manipulador com três graus de liberdade idealizado para a realização de simulações.

O processo de modelagem de uma estrutura rígida poli-articulada, como é o caso do manipulador da figura 3, já é bem conhecido e não oferece nenhum problema (Grover *et al.*, 1988, Craig, *et al.*, 1986). Utiliza-se em geral o formalismo de Euler-Lagrange (Meirovith, 1970), que necessita das energias cinética e potencial do sistema. Além das massas das articulações e carga terminal, considerou-se também as massas dos elos, levando-se em conta inclusive a energia cinética rotacional de cada massa do sistema. A equação diferencial matricial representando o modelo global (atuadores e estrutura) possui então a seguinte forma :

$$[I(\vec{X})]\ddot{\vec{X}} + \vec{f}(\vec{X}, \dot{\vec{X}}) + [f_v(\vec{X}, \dot{\vec{X}}, \vec{T}_m)]\dot{\vec{X}} + \vec{g}(\vec{X}) + [K]\vec{X} = \vec{T}_m \quad (3)$$

onde $\vec{X} = (\theta_{r1} \quad q_1 \quad \theta_{r2} \quad q_2 \quad \theta_{r3} \quad q_3)^T$, θ_{r1} , θ_{r2} e θ_{r3} sendo as posições angulares dos rotores dos moto-redutores 1, 2 e 3 respectivamente, assim como q_1 , q_2 e q_3 as posições angulares dos respectivos elos (posições angulares nas saídas dos redutores de velocidade). O vetor com os torques motores possui a forma: $\vec{T}_m = (T_{m1} \quad 0 \quad T_{m2} \quad 0 \quad T_{m3} \quad 0)^T$, onde T_{m1} , T_{m2} e T_{m3} são os torques aplicados nos rotores 1, 2 e 3. Os outros termos que aparecem na equação (3) possuem a seguinte definição:

- $[I(\vec{X})]$ é a matriz de inércia do sistema;
- $\vec{f}(\vec{X}, \dot{\vec{X}})$ o vetor com os torques de natureza Coriolis - centrífugos;
- $[f_v(\vec{X}, \dot{\vec{X}}, \vec{T}_m)]$ a matriz diagonal com os coeficientes de atrito viscoso variáveis (Gomes, 1995);
- $\vec{g}(\vec{X})$ o vetor com os torques de natureza gravitacional;
- $[K]$ a matriz com as constantes elásticas dos moto - redutores;

A equação (3) deixa claro que ao modelo estrutural acrescentaram-se as equações dos modelos das três articulações, incluindo-se assim os modos de vibração dos rotores, com toda a implicação de atritos não-lineares considerados nos modelos.

3. TRAJETÓRIAS DE REFERÊNCIA

Adotou-se, neste trabalho, as trajetórias de referência exponenciais, propostas em (Gomes *et al.*, 1998). A posição, velocidade e aceleração angulares desejadas são funções do tempo (t), das posições angulares inicial e final (q_i e q_f) de cada elo do manipulador, e do tempo final (t_f), sob a forma:

$$q^d(t) = (q_f - q_i) \left[1 - e^{-\left(\frac{bt^N}{(t_f + \varepsilon - t)^M}\right)} \right] + q_i \quad (4)$$

Considerando-se $N = M = b = 1$ e $\varepsilon = 0$, tem-se a velocidade e a aceleração sob as formas:

$$\dot{q}^d(t) = (q_f - q_i) e^{-\left(\frac{t}{t_f - t}\right)} \left(\frac{t_f}{(t_f - t)^2} \right) \quad (5)$$

$$\ddot{q}^d(t) = \frac{(q_f - q_i) \left(-\frac{1}{(t_f - t)} - \frac{t}{(t_f - t)^2} \right) e^{-\left(\frac{t}{t_f - t}\right)} t_f}{(t_f - t)^2} + \frac{2(q_f - q_i) e^{-\left(\frac{t}{t_f - t}\right)} t_f}{(t_f - t)^3} \quad (6)$$

Estas equações são válidas para $0 \leq t < t_f$ (o tempo inicial é suposto nulo). Para $t \geq t_f$, assume-se que $q^d = q_f$ e $\dot{q}^d = \ddot{q}^d = 0$.

4. LEIS DE CONTROLE

A seguir são apresentadas três técnicas de controle (Spong *et al.*, 1999): uma conhecida como dinâmica inversa, sendo não-linear e multivariável; duas outras baseadas na estratégia de controle independente das juntas.

4.1 Controle independente das juntas (*feedforward disturbance cancellation*)

Negligenciando as flexibilidades internas aos atuadores, a equação (3) pode ser desmembrada em duas, uma para a dinâmica estrutural e outra para a dinâmica dos atuadores:

$$\begin{aligned} D(\vec{q})\ddot{\vec{q}} + \vec{H}(\vec{q}, \dot{\vec{q}}) &= \vec{\tau} \\ I_r \ddot{\vec{\theta}}_r + C_r \dot{\vec{\theta}}_r &= \vec{T}_m - \vec{\tau} \end{aligned} \quad (7)$$

onde $H(\vec{q}, \dot{\vec{q}})$ corresponde aos torques coriolis-centrífugos e gravitacionais. C_r é uma matriz diagonal com coeficientes constantes de atrito viscoso (o atrito é considerado linear para o projeto dos ganhos do controlador). Escrevendo o sistema (7) sob a forma de $2n$ equações ($k = 1, 2, \dots, n$), onde n é o número de graus de liberdade, tem-se:

$$\begin{aligned} \sum_{j=1}^n d_{jk}(\vec{q}) \ddot{q}_j + h_k(\vec{q}, \dot{\vec{q}}) &= \tau_k \\ I_{rk} \ddot{\theta}_{rk} + c_{rk} \dot{\theta}_{rk} &= T_{mk} - \frac{1}{n_k} \tau_k \end{aligned} \quad (8)$$

Como a elasticidade interna foi negligenciada, $q_k = \theta_{rk}/n_k$, onde n_k corresponde ao índice de redução da k -ésima articulação. As equações (8) então permitem que se escreva a seguinte equação:

$$\left[I_{rk} + \frac{1}{n_k^2} d_{kk}(q) \right] \ddot{\theta}_{rk} + c_{rk} \dot{\theta}_{rk} = T_{mk} - \frac{1}{n_k} \underbrace{\left[\sum_{\substack{j=1 \\ j \neq k}}^n d_{jk}(\vec{q}) \ddot{q}_j + h_k(\vec{q}, \dot{\vec{q}}) \right]}_{p_k} \quad (9)$$

Tomando uma aproximação média para o termo

$$\left[I_{rk} + \frac{1}{n_k^2} d_{kk}(\vec{q}) \right] = I_k = \text{constante}, \quad (10)$$

a equação (9) torna-se:

$$I_k \ddot{\theta}_{rk} + c_{rk} \dot{\theta}_{rk} = T_{mk} - p_k \quad (11)$$

O torque de controle da k -ésima articulação possui então a seguinte forma:

$$T_{mk} = K_{pk} (\theta_k^d - \theta_{rk}) + K_{dk} (\dot{\theta}_k^d - \dot{\theta}_{rk}) + K_{ik} \int_0^t (\theta_k^d - \theta_{rk}) dt + \hat{p}_k \quad (12)$$

onde

$$\hat{p}_k = \frac{1}{n_k} \left[\sum_{\substack{j=1 \\ j \neq k}}^n d_{jk}(\vec{q}) \ddot{q}_j^d + h_k(\vec{q}, \dot{\vec{q}}) \right] \quad (13)$$

Supondo que $\hat{p}_k \approx p_k$, a substituição das equações (12) e (13) na equação (11) torna esta última linear, facilitando assim o projeto dos ganhos K_{pk} , K_{dk} e K_{ik} .

No caso do controle a altos ganhos implementado neste trabalho, considerou-se que o \hat{p}_k contém apenas os termos relativos aos torques gravitacionais:

$$T_{mk} = K_{pk} (\theta_k^d - \theta_{rk}) + K_{dk} (\dot{\theta}_k^d - \dot{\theta}_{rk}) + K_{ik} \int_0^t (\theta_k^d - \theta_{rk}) dt + \underbrace{\frac{1}{n_k} g_k(\vec{q})}_{\hat{p}_k} \quad (14)$$

Neste caso, os ganhos realmente necessitam ser maiores a fim de compensar tanto a aproximação da equação (10), quanto as diferenças mais significativas entre \hat{p}_k (equação (14)) e p_k (equação (9)).

4.2 Controle multivariável (dinâmica inversa)

Negligenciando os atritos não-lineares e as elasticidades internas dos atuadores, a equação (3) pode ser escrita sob a forma (Spong *et al.*, 1989):

$$D[\vec{q}] \ddot{\vec{q}} + \vec{H}(\vec{q}, \dot{\vec{q}}) = \vec{T}_m \quad (15)$$

onde o $H(\vec{q}, \dot{\vec{q}})$ contém os termos Coriolis-centrífugos e gravitacionais. Nesta técnica de controle, o vetor com os torques motores possui a forma:

$$\vec{T}_m = \vec{H}(\vec{q}, \dot{\vec{q}}) + \vec{T}_c \quad (16)$$

onde

$$T_c = [D(\vec{q})] \ddot{\vec{q}}^d + [K_p] (\vec{q}^d - \vec{q}) + [K_d] (\dot{\vec{q}}^d - \dot{\vec{q}}) + K_i \int_0^t (\vec{q}^d - \vec{q}) dt \quad (17)$$

Substituindo as equações (16) e (17) na (15), obtém-se uma equação matricial linear de segunda ordem para o erro $\vec{e} = \vec{q}^d - \vec{q}$, podendo-se então facilmente projetar as matrizes de ganhos $[K_p]$, $[K_d]$ e $[K_i]$ de forma que o erro tenda para zero com a velocidade e o amortecimento desejados.

5. SIMULAÇÕES

A primeira simulação mostra resultados (figura 4) com o controle por dinâmica inversa. Mostram-se as posições de referência (curvas tracejadas) e as posições de saída, nas situações com (gráficos do lado esquerdo) e sem (gráficos do lado direito) componente integral na lei de controle. Percebe-se portanto que, sem a utilização da componente integral, existem importantes erros de estado estacionário (o resultado para as outras técnicas de controle é semelhante), devido principalmente aos atritos não lineares nos atuadores.

Na figura 5 vê-se a velocidade rotor, desejada e de saída, bem como o torque motor, em duas situações: 5 ms e 30 ms como passo de discretização. Comportamento com tendência à instabilidade acontece a partir de 30 ms de passo. Ressalta-se, portanto, a importância de uma escolha criteriosa para o passo de discretização da lei de controle, principalmente em função de modos de alta frequência devidos à elasticidade interna aos atuadores.

As figuras 6, 7 e 8 têm o objetivo de avaliar o desempenho das técnicas de controle em presença de uma diminuição substancial do tempo final de trajetória (t_f), que de 1 segundo passa para 0.5 segundos. Nesta situação, as velocidades aumentam, fazendo crescer os torques inerciais e Coriolis-centrífugos. Foram adotados os seguintes códigos nas figuras para identificar as leis de controle: cont. 1 (dinâmica inversa); cont. 2 (*feedforward cancellation*); cont. 3 (altos ganhos e compensação gravitacional). Em todas vêem-se a posição angular da base (referência (curva tracejada) e de saída) e a velocidade rotor da base (referência (curva tracejada) e de saída). Como os ganhos foram ajustados para um bom desempenho com 1 segundo de tempo final de trajetória, há uma perda de desempenho, principalmente no cont. 2, sendo que o melhor resultado foi obtido com o cont. 3.

É muito comum, principalmente em aplicações industriais, manipuladores com garras na extremidade e que suspendem objetos, transportando-os para outras posições. Portanto, a carga terminal acaba variando a sua massa, sendo necessário que o controlador seja robusto em estabilidade e desempenho para com esta variação. As figuras 9, 10 e 11 mostram resultados de simulações abordando este aspecto. O tempo final t_f continuou 0.5 s, enquanto que a massa da carga terminal foi multiplicada por dez. Pode-se perceber uma perda geral de desempenho, uma vez que os ganhos foram projetados para um valor específico da massa da carga terminal, e um tempo final de trajetória de 1 s. O pior resultado foi obtido com o controle *feedforward cancellation* (cont. 2). Nos casos em que o tempo final de trajetória é o de projeto (1 s), a ampliação da massa provoca apenas pequenas perdas em termos de desempenho, conforme pode ser verificado na figura 12.

6. CONCLUSÕES

Resumidamente, podem ser extraídas as seguintes conclusões do presente trabalho:

- É fundamental que o modelo dinâmico do manipulador leve em conta os atritos não-lineares existentes nas articulações, sendo este um fator determinante, que afeta sensivelmente o grau de realismo e o caráter preditivo das simulações. O modelo para os torques de atrito utilizado no presente trabalho foi bastante testado experimentalmente, mostrando-se sempre muito realista;
- A tabela seguinte resume as conclusões sobre comparações entre as três técnicas de controle avaliadas. Levou-se em conta os seguintes critérios: robustez ao desempenho (precisão das respostas, mesmo em presença de diminuição do tempo final de trajetória e de variações paramétricas); estabilidade a variações paramétricas e a dinâmicas negligenciadas nos modelos nominais de base para as leis de controle (deve ser lembrado que as simulações foram realizadas com o modelo global da equação (3)); complexidade para efeito de implementação computacional em tempo real.;

Tabela 1. Resumo das conclusões sobre a comparação entre as técnicas de controle.

	Cont. 1	Cont. 2	Cont. 3
Desempenho	bom	regular	bom
Estabilidade	bom	bom	bom
Complexidade	alta	média	baixa

- O que garante a boa robustez a estabilidade é o retorno colocalizado das velocidades dos rotores, nas componentes derivativas presentes em todas as técnicas de controle;

REFERÊNCIAS

- Armstrong, B. S. R., 1988. Dynamics for robot control: friction modeling and ensuring excitation during parameter identification. PhD thesis, Stanford University.
- Craig, J. J., 1986, Introduction to robotics, mechanics and control. Addison Wesley.
- Ferretti, G.; Magnani, G.; Rocco, P., 1998. Compensation of motor torque disturbances in industrial robots. In IEEE Robotics and Automation Conference, pg. 2995-3001.
- Gomes, S. C. P. and Chrétien, J.P., 1992, Dynamic modeling and friction compensated control of a robot manipulator joint. In IEEE Robotics and Automation Conference, pg. 1450-55.
- Gomes, S. C. P. e Bier, C. C., 1998. Estudo sobre trajetórias de referência para o controle de robôs manipuladores. Congresso Brasileiro de Automática, CD-ROM, Uberlândia, MG.
- Gomes, S. C. P., 1995, Modelagem de atritos internos às articulações de robôs manipuladores. In XIII COBEM, CD-ROM, Belo Horizonte.
- Groover, M. P.; Weiss, M.; Nagel, R. N. and Odrey, N. G., 1989, Robótica, tecnologia e programação. McGraw-Hill.
- Meirovith, M., 1970, Methods of analytical dynamics. McGraw-Hill.
- Park, J.; Chung, W. and Youm, Y., 1998. Analytic nonlinear H^∞ optimal control for robotic manipulators. In IEEE Robotics and Automation Conference, pg. 2709-15.
- Spong, M. W. and Vidyasagar, M., 1989, Robot dynamic and control. John Wiley and Sons.

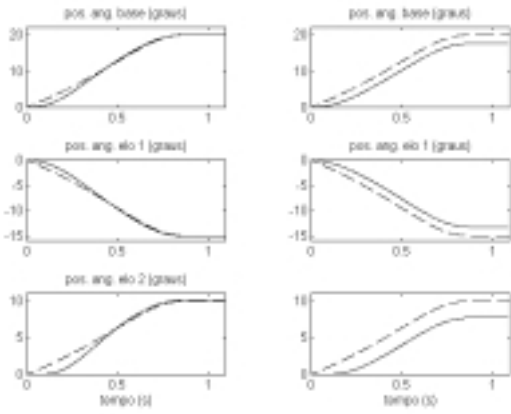


Figura 4. Simulações com e sem a utilização de componente de controle integral.

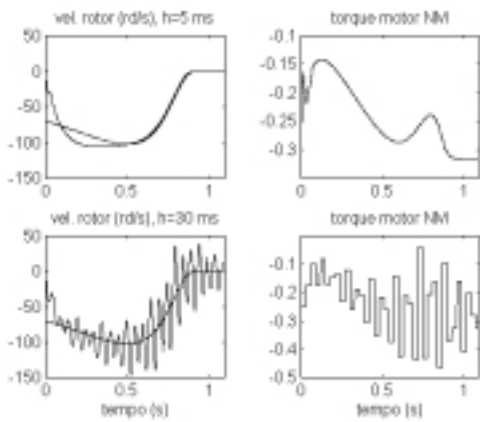


Figura 5. Simulações com a dinâmica inversa (resultados para o elo 2), com diferentes passos de discretização.

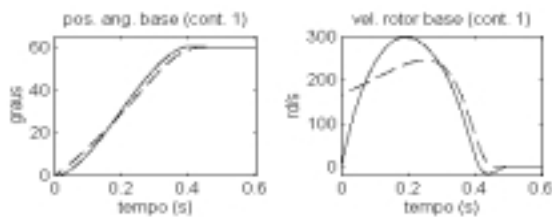


Figura 6. Controle por dinâmica inversa, e tempo final de trajetória $t_f = 0.5$ s.

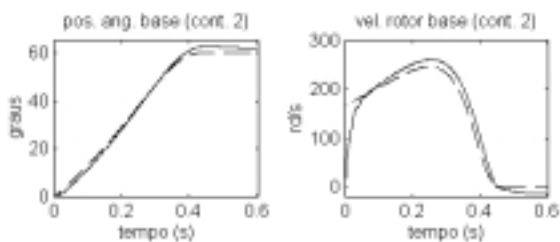


Figura 7. Controle *feedforward cancellation*, e tempo final de trajetória $t_f = 0.5$ s.

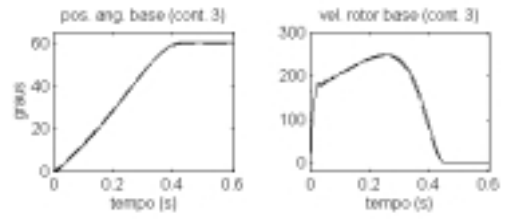


Figura 8. Altos ganhos e compensação gravitacional, com $t_f = 0.5$ s.

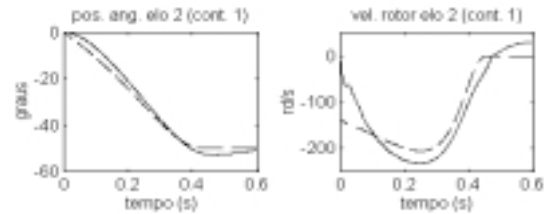


Figura 9. Dinâmica inversa, com $t_f = 0.5$ s e a massa da carga terminal dez vezes maior do que a nominal ($M_c = 10M_{CN}$).

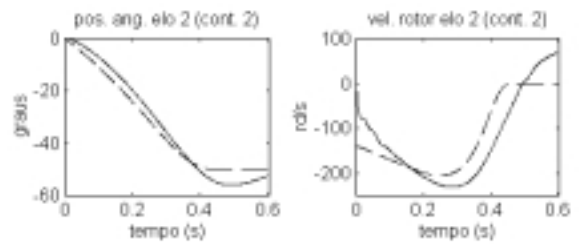


Figura 10. *Feedforward cancellation*, $t_f = 0.5$ s e $M_c = 10M_{CN}$.

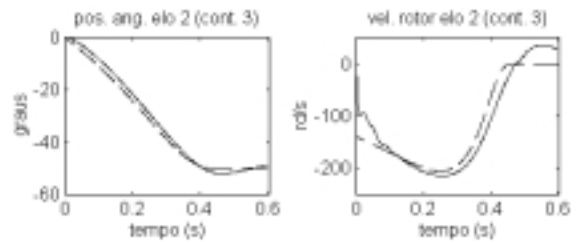


Figura 11. Altos ganhos e compensação gravitacional, $t_f = 0.5$ s e $M_c = 10M_{CN}$.

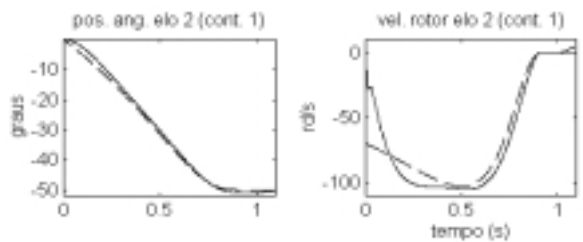


Figura 12. Dinâmica inversa, com $t_f = 1$ s, e $M_c = 10M_{CN}$.

MULTIVARIABLE CONTROL FOR MULTI INPUT - MULTI OUTPUT COOLING MACHINES BASED ON VAPOR COMPRESSION

José Maria Gálvez, Ph.D.

Agostinho Gomes da Silva, M.Sc.

Department of Mechanical Engineering, Federal University of Minas Gerais, Brazil.

Av. Antônio Carlos 6627, Pampulha, 31.270-901 Belo Horizonte, MG, Brazil.

Phone: +55 31 499-5236. Fax: +55 31 443-3783. E-mail: jmgm@dedalus.lcc.ufmg.br.

Abstract

Traditionally, cooling machines are usually operated as open loop systems. With the just arrived world-wide energy crisis, the search for new energy-saving cooling systems has been intensified. Recently, several improvements have been proposed by the technical community but most of these new techniques are related with new device technology. Variable speed compressors and electronic expansion valves are among the most successful device improvements. The researcher's expectancy points to new automatic control schemes for the next generation of cooling machines, however, some difficulties must be solved before new schemes can be used in practice. The feedback control design is a difficult task due to the cross coupling among the process variables. This paper presents a new control scheme that allows to control the freezing power and the superheating in an independent way. It explains a control design procedure performed in the frequency domain. Finally, it shows simulation results, which validate the design procedure.

Keywords: Cooling Systems, Multivariable Control, Frequency Domain.

1. INTRODUCTION

The improvement of power consumption efficiency of industrial devices is one of the main issues for the incoming century. In the first century of the industrial age, the world population has virtually exploded, nature has been almost destroyed and energy resources have been almost depleted. In spite of that, the human living comfort has become a necessity for most of the whole world population even for the third world people. It is a fact that the next decades are going to testify a continuous and strenuous search for new devices and technologies to save energy resources.

The energy consumption by heating and cooling systems in commercial and industrial buildings corresponds to 50% of the world energy consumption (Imbabi, 1990). The operation of heating and cooling systems in commercial and industrial buildings still is an inefficient and high-energy consumption process (Shoureshi, 1993).

It is already known that the solution for the poor operation of heating and cooling systems relies on the proper choice and design of automatic controllers. Low cost controllers such as On/Off control and PID control are currently used as the standard controllers in the heating, ventilation and air conditioning (HVAC) industry. However, their low energy efficiency causes an extra undesired energy burning.

An important fact that causes the control low energy efficiency is that most designs are only capable of dealing with constant thermal loads, which is not the general case. In practice, thermal loads are time varying. The temperature sensor location is another difficult in the control field of heating and cooling devices. The natural position for the temperature sensor is close to, or even inside, the target environment. In practice, to avoid the inclusion of time delays in the control loop, the temperature sensor is usually located close to the heat source (sink). Furthermore, conventional single-input single-output (SISO) control of cooling machines is not capable of controlling the freezing power and super-heating independently (I/O cross coupling).

To deal with the control problem of time varying processes, time delays and I/O cross coupling, several control strategies have been proposed by the control community. Among them, robust control, adaptive control and intelligent control are the most important. A drawback of these sophisticated alternatives is that they are usually expensive and required advanced computational resources.

To face time-varying thermal loads, control loop time delays and I/O cross coupling, new-low cost multi-input multi-output (MIMO) control strategies must be explored. This paper introduces a MIMO control scheme that permits the independent control of the output variables of cooling machines based on the vapor compression cycle. Figure 1 shows the schematic diagram of a cooling machine of this type.

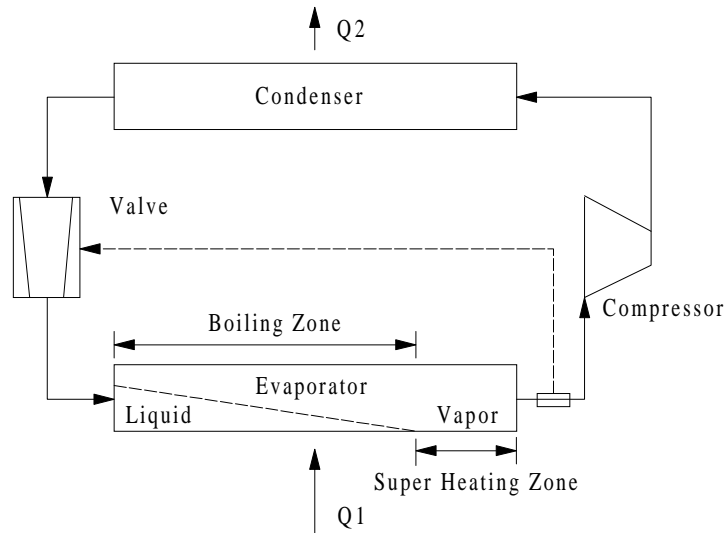


Figure 1. The Cooling System.

2. THE COOLING SYSTEM

This paper focus in the control of the system constituted by the expansion valve, evaporator and compressor (Figure 1). In this case, the system dynamics is defined by a matrix transfer function of the form:

$$\begin{bmatrix} \Delta T(s) \\ Q_1(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} MFR(s) \\ VFR(s) \end{bmatrix} \quad (1)$$

The system inputs are the expansion valve opening position, which defines the mass flow rate (MFR) and the compressor speed, which controls the volume flow rate (VFR). The system outputs are the super heating, ΔT , and the freezing power, Q_1 , (Figure 2).

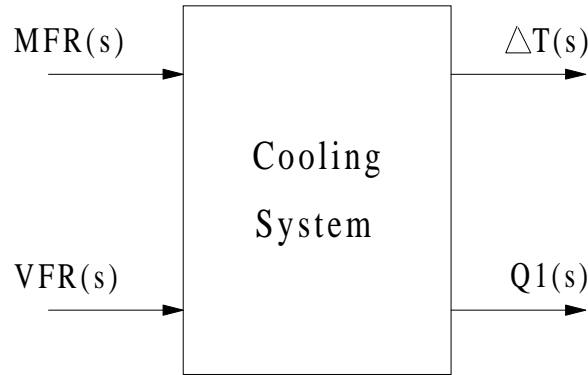


Figure 2. The Open Loop System.

Equation 1 can be written as:

$$[Y(s)] = [G(s)][U(s)] \quad (2)$$

with

$$[G(s)] = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix}$$

and

$$[Y(s)] = \begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \Delta T(s) \\ Q_1(s) \end{bmatrix}$$

$$[U(s)] = \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} = \begin{bmatrix} MFR(s) \\ VFR(s) \end{bmatrix}$$

Ideally, the opening of the expansion valve would be used to regulate the super heating and the velocity of the variable-speed compressor would be used to control the generation of freezing power (Figure 3 with $G_{12}(s) = G_{21}(s) = 0$), unfortunately, this is not the case.

A strong cross coupling interaction among inputs and outputs is a characteristic of these systems, which means that $G_{12}(s)$ and $G_{21}(s)$ can not be neglected in practice. This can be easily

understood since each output is a function of both inputs (the valve opening position and the compressor velocity), as shown in Figure 3.

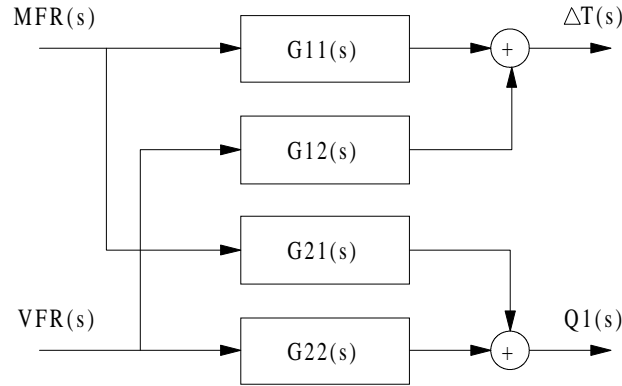


Figure 3. The Cooling System Cross Coupling.

3. MODEL STRUCTURE AND VALIDATION

Several linear and non-linear computational models can be found in the technical literature (Koury, 1998, Machado, 1996, Rocha, 1995 & Outtagarts, 1994). Considering the work by Machado (1986) a 2x2 MIMO model structure was chosen for the cooling machine:

$$[Y(s)] = [G(s)] [U(s)] = \begin{bmatrix} \Delta T(s) \\ Q_1(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} MFR(s) \\ VFR(s) \end{bmatrix} \quad (3)$$

where

$$G_{11}(s) = K_{11} \frac{1}{T_{I_{11}} s + 1} \quad G_{12}(s) = K_{12} \frac{(T_{2_{12}} s + 1)}{(T_{I_{12}} s + 1)}$$

$$G_{21}(s) = K_{21} \frac{(T_{3_{21}} s + 1)}{(T_{I_{21}} s + 1)(T_{2_{21}} s + 1)} \quad G_{22}(s) = K_{22} \frac{(T_{3_{22}} s + 1)}{(T_{I_{22}} s + 1)(T_{2_{22}} s + 1)}$$

The model identification procedure (Machado, 1996) led to:

$$[G(s)] = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{-5.62}{(45s+1)} & \frac{-2.49(-70s+1)}{(59.52s+1)} \\ \frac{33.89(-36.37s+1)}{(25.65s+1)(67.79s+1)} & \frac{22.20(630s+1)}{(80s+1)(90s+1)} \end{bmatrix}$$

4. THE MIMO CONTROLLER

The structure of a MIMO output feedback control law was defined as:

$$[U(s)] = [K(s)] [E(s)] = [K(s)] [R(s) - Y(s)] \quad (4)$$

where

$$[K(s)] = \begin{bmatrix} K_{11}(s) & K_{12}(s) \\ K_{21}(s) & K_{22}(s) \end{bmatrix}$$

and

$$[R(s)] = \begin{bmatrix} \Delta T(s) \text{ Setpoint} \\ Q_1(s) \text{ Setpoint} \end{bmatrix}$$

Figure 4 shows the MIMO controller structure:

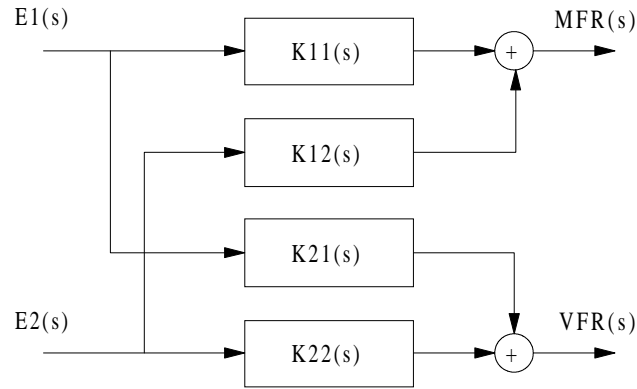


Figure 4. The MIMO Controller Implementation.

The controller tuning procedure led to the following control law:

$$[K(s)] = \begin{bmatrix} K_{11}(s) & K_{12}(s) \\ K_{21}(s) & K_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{-1.0952s - 0.0219}{s} & \frac{-0.6133s - 0.0123}{s} \\ \frac{1.6719s + 0.0334}{s} & \frac{1.3867s + 0.0277}{s} \end{bmatrix}$$

Finally, the closed loop transfer function can be determine by (Francis, 1987):

$$T(s) = [I + G(s)K(s)]^{-1}G(s)K(s) \quad (5)$$

It should be noticed that, the entry $K_{ij}(s)$ has the general form of the of a SISO classical PI controller. The designed controller leads to a closed loop transfer function that can be approximated (at low frequencies) by the diagonal matrix transfer function of the form:

$$[Y(s)] \cong \begin{bmatrix} T_{11}(s) & 0 \\ 0 & T_{22}(s) \end{bmatrix} [R(s)] \Leftrightarrow \begin{bmatrix} \Delta T(s) \\ Q_1(s) \end{bmatrix} \cong \begin{bmatrix} T_{11}(s) & 0 \\ 0 & T_{22}(s) \end{bmatrix} \begin{bmatrix} \Delta T(s) \text{ Setpoint} \\ Q_1(s) \text{ Setpoint} \end{bmatrix} \quad (6)$$

Equation 6 shows the main feature of the proposed controller (due to the space limitation the full form of Equation 6 was omitted), the close loop system is diagonal dominant at low frequency. This fact allows the control of the superheating and freezing power to be independent. Figure 5 shows the block diagram for the closed loop system.

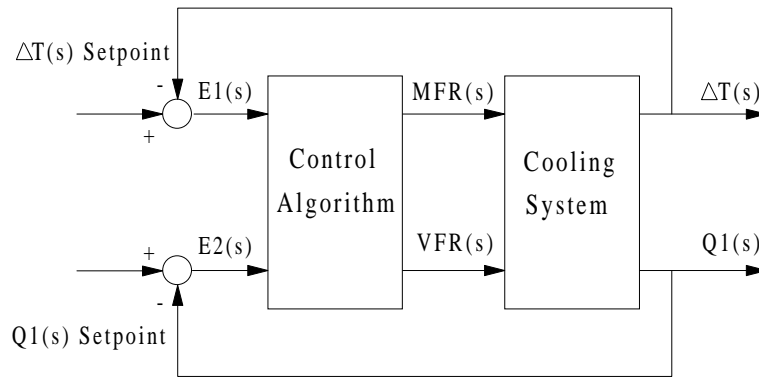


Figure 5. The Closed Loop System.

5. SIMULATION RESULTS

Simulation results are presented here to illustrated the controller performance. Figure 6 presents the system open loop responses. It shows the super-heating (quadrant II) and freezing power (quadrant IV) time responses. It can be observed (in quadrants I and III) the strong effect of the I/O cross coupling. In the ideal case, the time responses shown in quadrants I and III should remain in zero for all time or at least return to zero at steady state. Figure 7 shows the system closed loop responses. It shows the super-heating (quadrant II) and freezing power (quadrant IV) closed loop time responses to a unit step. In Figure 7 also shows (in quadrants I and III) that the effects of the I/O cross coupling were eliminated by the proposed control law.

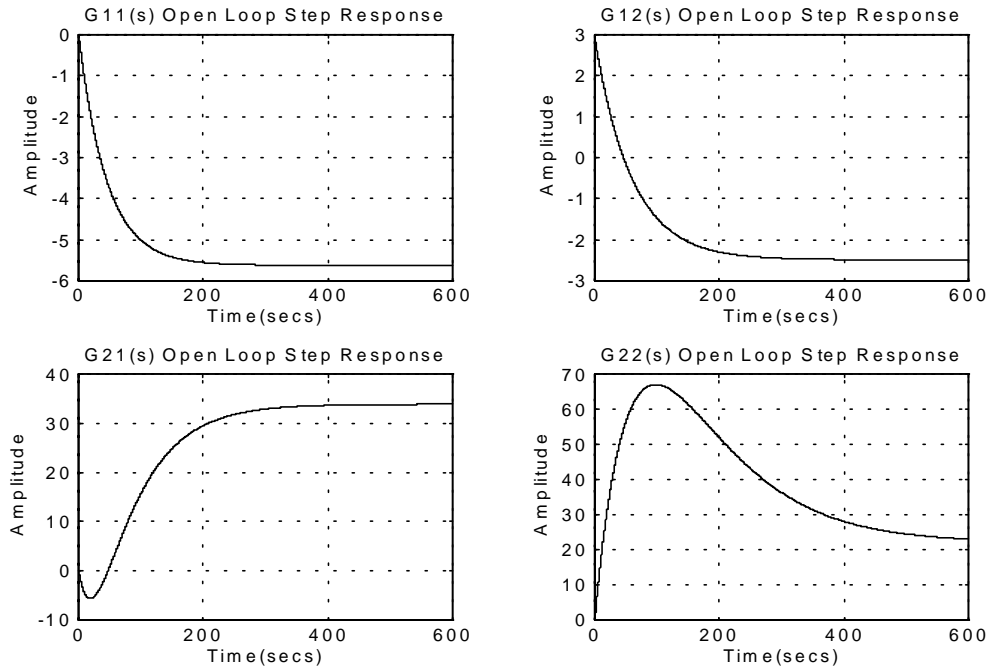


Figure 6. Open Loop Step Responses.

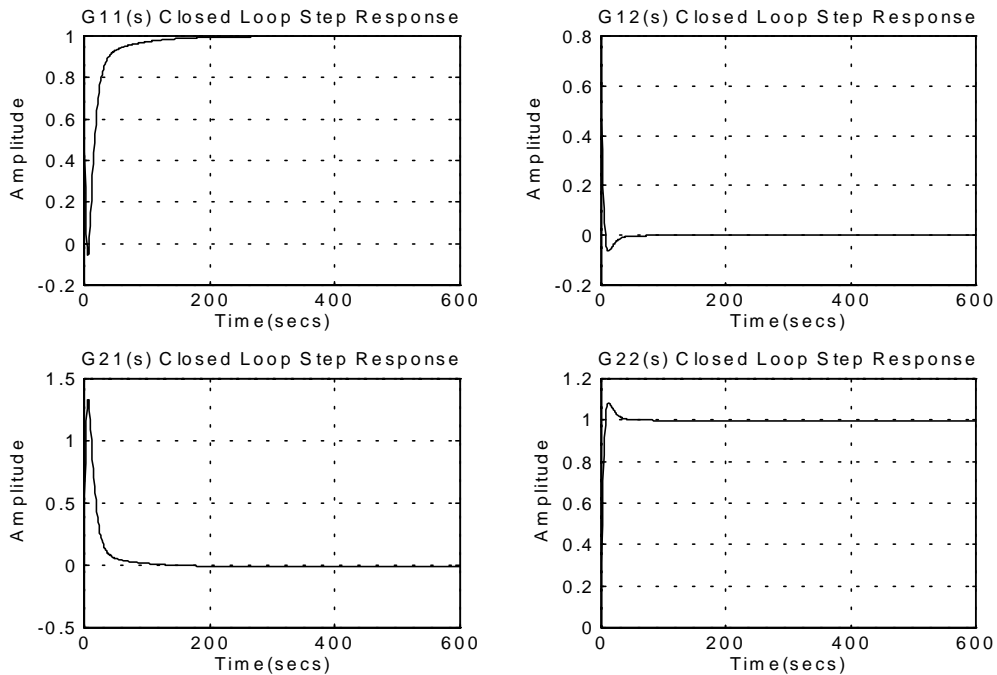


Figure 7. Closed Loop Step Responses.

6. FINAL COMMENTS

Traditionally, classical on-off controllers for cooling machines have shown to be inefficient for energy saving purposes (Rocha, 1995; Machado, 1996). Variable compressor speed operation has recently emerged as the solution for the energy consumption minimization problem. The searching for an inexpensive compressor speed controller is currently on the focus of the attention of the control community and although some fine results can be found in the technical literature the final solution is still under investigation.

This paper shown that independent control of superheating and freezing power in a cooling system based on vapor compression is a feasible task. A MIMO controller for a cooling machine was designed and implemented in computer simulation. The potential of the proposed MIMO controller for saving energy and keeping comfort was verified through simulation.

Further research has to be done including the plant non linearity to assess the full capability of the proposed scheme.

7. REFERENCES

- Arguello-Serrano, B. & Vélez-Reyes, A., “Nonlinear Control of a Heating, Ventilating, and Air Conditioning System with Thermal Load Estimation”, *IEEE Transactions on Control Systems Technology*, vol. 7, pp. 56-63, January 1999.
- Grosdidier, P. and Morari, M., “Interactions Measurements for Systems under Decentralized Control”, *Automatica*, vol. 22, pp. 309-319, 1986.
- Imbabi, M.S., “Computer Validation of Scale Model Test for Building Energy Simulation”, *Int. J. Energy Res.*, vol. 14, pp. 727-736, 1990.
- Machado, L., *Modele de Simulation et Etude Experimentale d’un Evaporateur de Machine Frigorifique en Regime Transitoire*, L’Institut National des Sciences Appliquees de Lyon, France, 1996.
- McAvoy, T. J., “Interaction Analysis - Principles and Applications”, Instrument Society of America, 1983.
- Mees, A.I., “Achieving Diagonal Dominance”, *Systems and Control Letters*, vol. 1, pp. 155-158, 1981.
- Outtagarts, A., *Comportement Dynamique d’un Evaporateur de Machine Frigorifique Alimente par un Detendeur Electronique*, L’Institut National des Sciences Appliquees de Lyon, France, 1994.
- Rocha, R., *Contribuição ao Estudo de um Sistema de Refrigeração com Compressor Operando a Velocidade Variável*, Dissertação de Mestrado, PPGEE-UFGM, Brasil, 1995.
- Stoecker, W.F. & Jones, J.W., *Refrigeration and Air Conditioning*, McGraw-Hill Book Company, 1982.

MOBILE ROBOT NAVIGATION BASED ON POTENTIAL FIELD

Allan Aminadab André Freire Soares

Pablo Javier Alsina

Universidade Federal do Rio Grande do Norte – Centro de Tecnologia – Departamento de Engenharia Elétrica

59072-970 - Natal - RN – Brasil

E-mail: aminadab@leca.ufrn.br , pablo@leca.ufrn.br

Abstract

This paper presents a case study of mobile robot obstacle avoidance based on the potential field method. Attractive and Repulsive Potential field functions, which are associated to the robot goal and the environment obstacles respectively, are described. The motion planning and obstacle avoidance of a soccer player robot are studied and simulation results are shown. The involved problems are discussed and future work perspectives are proposed.

Keywords: Mobile Robot, Navigation, Potential Field.

1. INTRODUCTION

The capability of planning its own motion in order to achieve a given task is a fundamental condition for any implementation of an autonomous robot system.

Let the robot be a single rigid body \mathbf{A} moving in a Euclidean work space $\mathbf{W} \in \mathbf{R}^2$ populated of fixed rigid objects (obstacles) $\mathbf{B}_1, \dots, \mathbf{B}_q$. In its simplest form, the motion planning problem can be stated as (Latombe, 1998):

Given an initial configuration (position and orientation) $q_{ini} = (x_{ini}, y_{ini}, \theta_{ini})$ and a goal configuration $q_{goal} = (x_{goal}, y_{goal}, \theta_{goal})$ of \mathbf{A} in \mathbf{W} , generate a path τ specifying a continuous sequence of configurations q of the robot \mathbf{A} , starting at q_{ini} and terminating at q_{goal} , avoiding contact with the obstacles \mathbf{B}_i 's. Return this path, if it exists, or report failure otherwise.

There are several techniques that provide means of solving this problem, however, not all of them solve the problem in its general case. Despite of particular differences, all these methods can be grouped in three general approaches: *Roadmap* (Nilsson, 1969), (Ó'Dúnlaing & et al, 1983), *Cell Decomposition* (Schwartz & et al, 1987), (Brooks & Lozano-Pérez, 1983) and *Potential Field* (Khatib, 1986), (Koditschek, 1989). In the first two approaches, the motion planning problem is reduced to the search of a shortest path in a *connectivity graph*, which captures the global topological properties of the workspace. On the other hand, potential field methods are based on a completely different idea. The robot movement is guided by the influence of an *artificial potential field* $U(q)$, function of the robot configuration q , whose local variations reflect in some degree the structure of the free space. In most implementations, the potential function is defined as a sum of an attractive potential,

which pulls the robot toward the goal, and a repulsive potential, which pushes the robot away from the obstacles.

In this paper, a case study of mobile robot motion planning and obstacle avoidance based on the potential field method is presented. The studied case is the motion planning and obstacle avoidance of a soccer player robot. Section 2 gives a brief description of the problem. Section 3 describes the implementation issues of the potential field method. Simulation results are shown and analyzed in Section 4. The final conclusions and perspectives for future works are given in Section 5.

2. DESCRIPTION OF THE PROBLEM

The problem under study is the motion planning and obstacle avoidance of a soccer player robot according to the FIRA robot competition rules. In this competition, two teams are in contend. Each team consists in three soccer players, which are mobile robots of cube format, with a size of $7.5 \times 7.5 \times 7.5 \text{ cm}^3$. The workspace is a rectangle of $130 \times 150 \text{ cm}^2$, as shown in Figure 1. Thus, the navigation problem can be considered as a two dimensional one. The robot configuration can be parameterized as $q = (x, y, \theta)$, where (x, y) represents the position of the origin of the robot coordinate frame F_A (which is fixed on the center of the axis of its drive wheels), expressed with respect to the workspace coordinate frame F_W . Similarly, the coordinate θ represents the orientation angle of the robot frame with respect to the workspace frame. Due to cinematic constraints of the robot, the orientation θ must always be tangent to the trajectory (differential drive model (Borenstein et al., 1996)). The configurations of the robots and the ball are captured by a PC-based vision system, by means of a camera localized over the workspace.

In a first approach, the navigation problem to be solved by the robot is the following: to find a path from its actual configuration until the goal configuration, avoiding the obstacles (the other five robots). In this case, the goal configuration will be the ball position. In a simplified initial version of the problem, the other robots will be considered as obstacles in fixed positions.

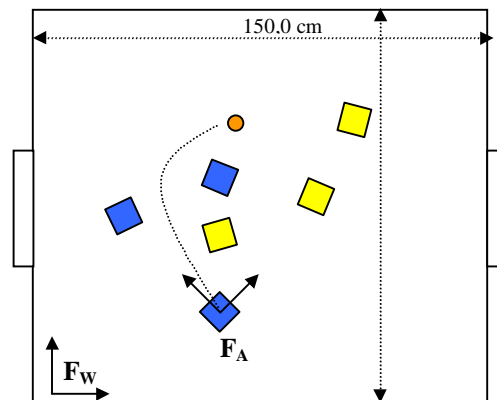


Figure 1. The robot and its Workspace

3. THE POTENTIAL FIELD METHOD

The potential field method was originally proposed for on-line collision avoidance in unstructured environments, when the robots does not have a prior model of the obstacles, sensing them on-line, as in the case of robot soccer. Navigation methods based in this approach are quite efficient, reasonably reliable and fast in a wide range of situations. In this

method, the robot is treated as particle under the influence of an artificial potential field. The potential function is expressed as the sum of an attractive potential $U_{at}(q)$, pulling the robot to the goal, and a repulsive potential $U_{rep}(q)$, pushing the robot away from the obstacles:

$$U(q) = U_{at}(q) + U_{rep}(q) \quad (1)$$

The attractive potential function can be considered as a "valley" of the potential surface. It is defined as a parabolic surface with a global minimum of the potential function at q_g :

$$U_{at}(q) = \frac{1}{2} \hat{\xi} \left\| q - q_{goal} \right\|^2 \quad (2)$$

where ξ is a positive gain. The attractive potential function is shown in Figure 2.

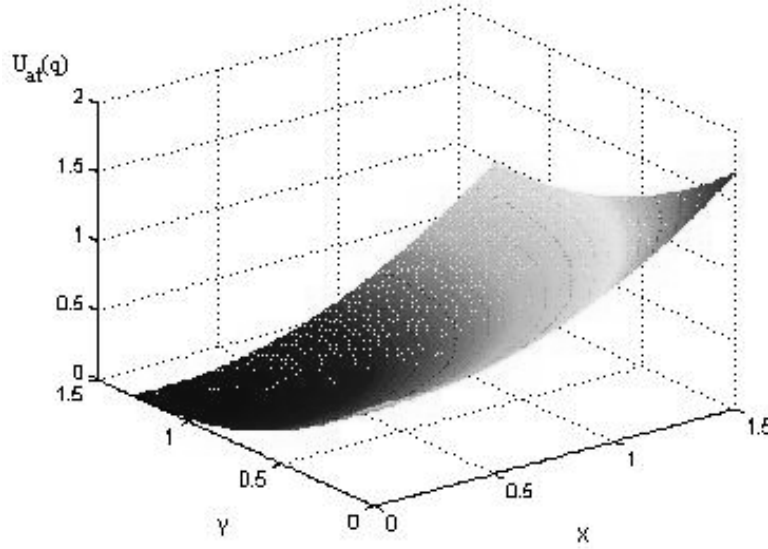


Figure 2. The attractive potential surface

The repulsive potential forms a potential barrier around the obstacles, which impedes the robot to collide with them. The repulsive potential created by each obstacle B_i is defined by:

$$U_{B_i}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{d_B(q)} - \frac{1}{d_{B0}} \right)^2 & \text{if } d_B(q) \leq d_{B0} \\ 0 & \text{if } d_B(q) > d_{B0} \end{cases} \quad (3)$$

where η is a positive gain and $d_B(q)$ is the minimum distance from the robot (at configuration q) to the obstacle (at configuration q_B). The positive parameter d_{B0} is the *distance of influence* of the obstacle. Beyond this distance, the potential created by the obstacle is null, having no effect on the robot. For this reason, the potential field methods are often referred to as local methods. The potential function grows in the neighborhood of the obstacle and tends to infinity as the robot gets closer to its boundaries. The repulsive potential acting over the robot is the sum of the potentials created by each of the obstacles:

$$U_{rep}(q) = \sum U_{B_i}(q) \quad (4)$$

The Repulsive potential function is shown in Figure 3.

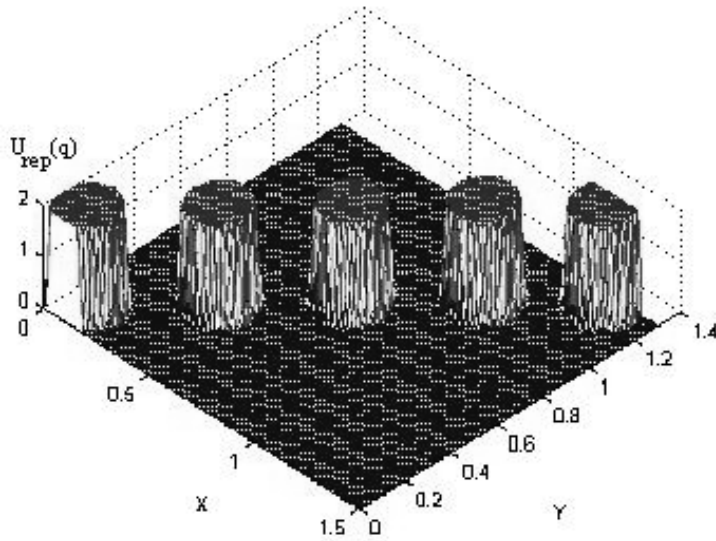


Figure 3. The repulsive potential surface

The potential field approach belongs to the family of *steepest descent* optimization methods. At each iteration, the most promising direction for robot motion is the direction of the artificial force induced by the potential function:

$$\vec{F}(q) = -\vec{\nabla}U(q) \quad (5)$$

Thus, the robot follows the negated gradient of the potential function, descending from the "mountains", which are created by the obstacles, to the "valley" created by the goal in the potential surface. This potential function is depicted in Figure 4.

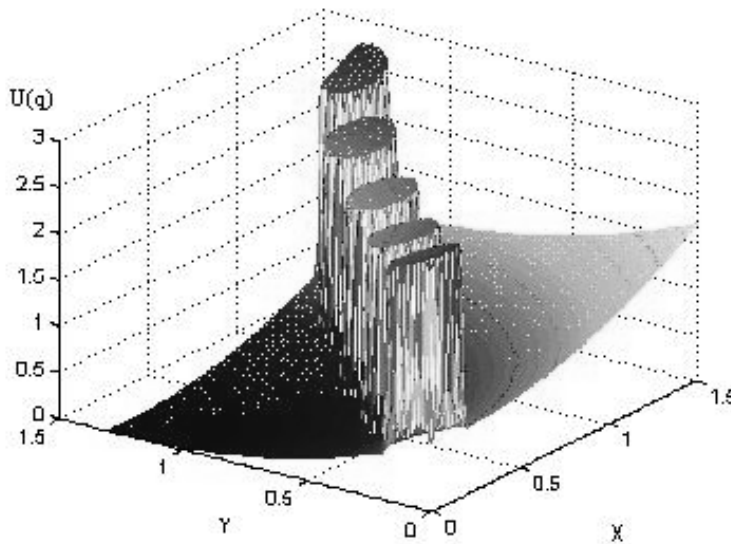


Figure 4. The potential surface

At each iteration i , the new configuration, q_{i+1} , is obtained from the previous one, q_i , through a small displacement δ_i , in the direction of the artificial force:

$$q_{i+1} = q_i + \ddot{a}_i \cdot \frac{\bar{F}(q)}{\|\bar{F}(q)\|} \quad (6)$$

Since, due to cinematic constraints, the robot orientation must be tangent to the path, the expression above is applied only to the position coordinates. The robot orientation at each iteration was obtained from the direction of the segment lying the actual position (x_i, y_i) to the next position (x_{i+1}, y_{i+1}) .

4. SIMULATION RESULTS

In order to investigate the feasibility of the potential field method in motion planning and obstacle avoidance of the soccer player robot, simulation experiments were performed. The parameters were obtained by trial-and-error, and their best found values are shown in Table I.

Table I. Control Parameters

Parameter	Value
ξ	1.00
η	2.82
d_{B0}	0.14 m
δ_i	0.0008 m

Experiment I:

In the first experiment, the robot, in the configuration $(1.1m, 0.1m, 100^\circ)$, must reach the ball, in the position $(0.1m, 1.2m)$, avoiding the obstacles located at positions: $(0.2, 0.2)$, $(0.4, 0.4)$, $(0.6, 0.6)$, $(0.8, 0.8)$ and $(1.0, 1.0)$. In this situation, the obstacles are between the robot and the ball, but there is enough space to allow the robot pass through them. The obtained results for this experiment are shown in Figure 5.

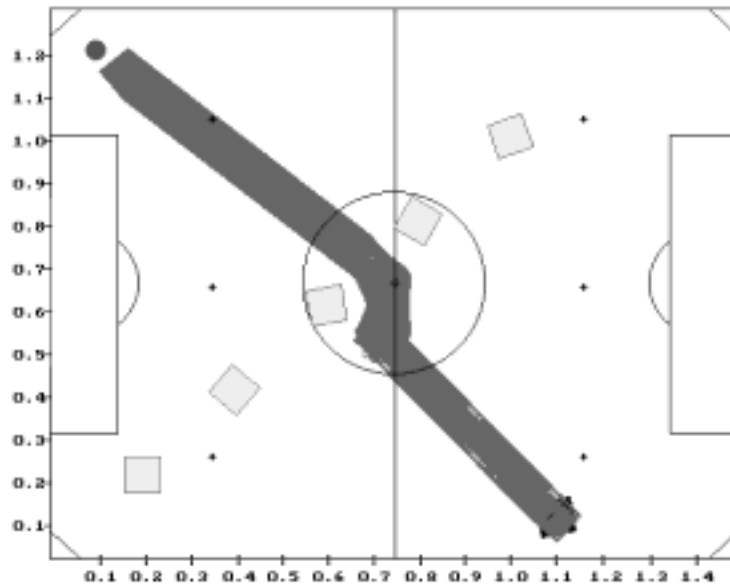


Figure 5. Experiment I

It can be noted from Fig. 5 that the robot is attracted by the goal until reach the distance of influence of the obstacle, being repelled by it. In this case, the robot contours the obstacle without any problem, reaching the goal.

Experiment II:

A more difficult situation was simulated in experiment II. The initial position of the robot, ball and obstacle are respectively: (1.15,0.25), (0.15,0.95), (0.95,0.30), (1.00,0.75), (0.75,0.65), (0.40,0.60) and (0.40,0.95). In this situation, the robot has several obstacles between it and the ball. Also, the ball is not visible from the robot initial position, as shown in Figure 6.

Again, although this obstacle configuration is more unfavorable than the previous one, the robot found a free path to the goal, despite of the multiple obstacles between its initial position and the ball. When the robot approaches the distance of influence of an obstacle, the repulsive potential impede the collision, and the robot passes around its neighborhood.

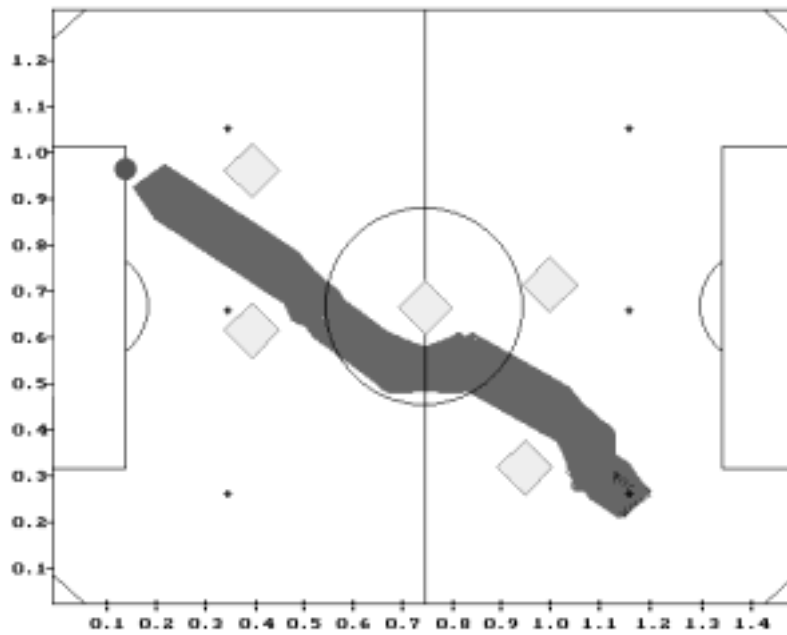


Figure 6. Experiment II

Experiment III

In this case, a worst case situation was studied. The Robot, in the initial position (0.85,0.45), is surrounded by a barrier formed by the other five robots, which are in the positions (0.95,0.75), (0.70,0.70), (0.60,0.60), (0.65,0.50) and (0.70,0.35). The ball is in the position (0.40,0.80), behind the obstacle barrier. The corresponding simulation results are shown in Figure 7.

As shown in Figure 7, the robot cannot pass through the potential barrier created by the obstacles and gets stuck in the resulting local minimum. This problem was expected, since the potential field approach is, in fact, a steepest descent optimization method and no technique was utilized to escape the local minimum.

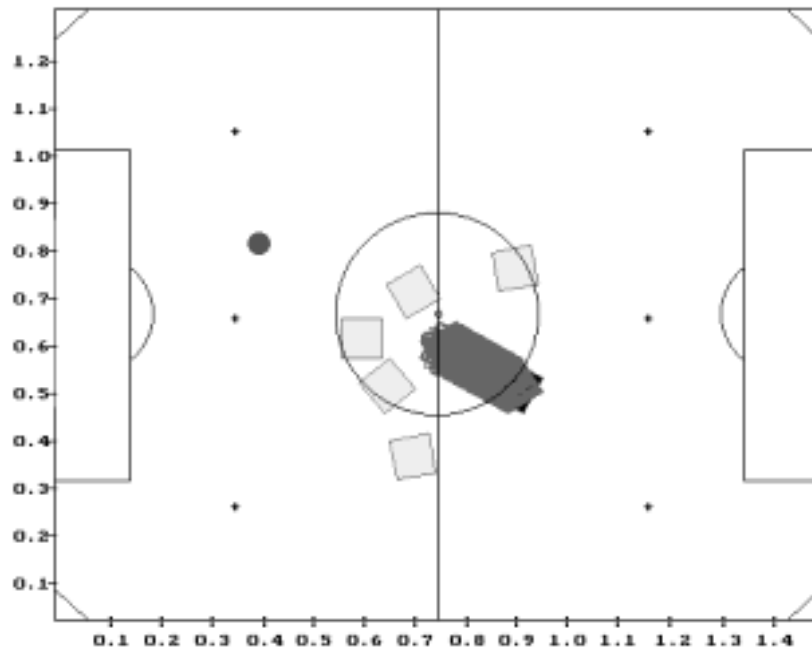


Figure 7. Experiment III

Although local minima are a serious problem in steepest descent-based optimization methods, there is not the case in present situation. In fact, the potential field method won't be utilized alone. A higher hierarchy control instance will be utilized to generate intermediate goals according to a predefined game strategy. The function of the studied potential field technique in the robot navigation will be essentially the obstacle avoidance and, according to these previous simulation results, the method seems to achieve this objective in an efficient and reliable way.

For simplicity, in this previous study, the direction in which the robot approaches to the goal was not taken in consideration. Currently, a modification of the potential function in order to treat with the approaching direction is under study. The idea is to include preferential directions in the attractive potential function by establishing a "ditch" in the potential surface on the approaching side of the goal and a potential barrier in the opposite side. It is expected that, with this artifice, the robot will contour the goal and will approach it from the right side.

5. CONCLUSION

In this paper, the application of the potential field method to the motion planning of a soccer player robot was studied. The obtained simulation results suggest that this method can be applied to implement a reliable and efficient robot obstacle avoidance capability. In extreme worst case situations, problems with local minima were observed, but this is not really a great impediment for the application of this method to the case under study, since it will be utilized together with a higher control level that will define intermediate goals according to a game strategy. So, it is expected that mainly of the difficulties due to local minima will be solved. A modification on the potential function is under study in order to include the goal approaching direction on the motion planning.

For simplicity, the movement of the other robots was not taken in consideration. In future works, the method will be modified to take into account the predicted movements of the obstacles.

Currently, a robot soccer team is under construction in the Robot Laboratory of the Electrical Engineering Department of the UFRN and the studied technique will be implemented in the real robots.

6. REFERENCES

- Borenstein, J., Everett, H. R. & Feng, L., 1996, "Navigating Mobile Robots - System and Techniques". A. K. Peters, Ltd. USA.
- Brooks, R. A. & Lozano-Pérez, T., 1983, "A Subdivision Algorithm in Configuration Space for Find Path with Rotation". Proceedings of the 8th International Conference on Artificial Intelligence. Karlsruhe, FRG, pp. 799-806.
- Khatib, O, 1986, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". International Journal of Robotic Research. Vol. 5, No. 1, pp. 90-98.
- Koditschek, 1989, D. E., "Robot Planning and Control via Potential Functions". in The Robotics Review 1, MIT Press.
- Latombe, J. C., 1998, "Robot Motion Planning". Kluwer Academic Publishers, USA, 5th printing.
- Nilsson, N. J., 1969, "A Mobile Automaton: An Application of Artificial Intelligent Techniques". Proceedings of the 1th International Joint Conference on Artificial Intelligence. Washington D.C., USA, pp. 509-520.
- Ó'Dúnlaing, C., Sharir, M. & Yap, C. K., 1983, "Retraction: A New Approach to Motion Planning". Proceedings of the 15th ACM Symposium on the Theory of Computing. Boston, USA, pp. 207-220.
- Schwartz, J. T., Sharir, M & Hopcroft, J., 1987, "Planning, Geometry and Complexity of Robot Motion ". Ablex, Norwood, NJ, USA.

DESENVOLVIMENTO DE SOFTWARE PARA O APOIO DO ENSINO DE METROLOGIA DIMENSIONAL MECÂNICA

Eduardo Carlos Bianchi

Robson Cristiano de Campos

Universidade Estadual Paulista, Departamento de Engenharia Mecânica, 17033-360, Bauru, SP, Brasil. E-mail: bianchi@bauru.unesp.br.

Resumo

Neste trabalho foi desenvolvido um software que possibilita o ensino da medição dos erros dimensionais mecânicos, sem a utilização física de instrumentos de medição. O software procura estimular o usuário a ser autodidata, onde o avanço da informática permitirá que profissionais/alunos possam aprender através de computadores, utilizando o professor apenas para esclarecer dúvidas que o programa não esclareça.

Palavras-chave: Metrologia, ensino, software

1. INTRODUÇÃO

Atualmente, têm-se dado cada vez mais atenção ao estudo da prática de metrologia, devido à tendência no mercado em busca de qualidade e competitividade. A metrologia é uma ciência exata que tem como meta quantificar o valor de uma medida dimensional.

A metrologia dimensional mecânica pode ser dividida em dois grandes ramos, sendo que um é responsável pelas medidas muito precisas, também conhecida de mecânica fina, que se refere a microgeometria. Esta parte da metrologia utiliza aparelhos sofisticados em geral caros, como o rugosímetros, medidores de circularidade, dentre outros.

O outro ramo da metrologia é de medidas macrogeométricas. Para a obtenção destas medidas, os instrumentos utilizados nestas práticas, são normalmente menos sofisticados e menos caros que os utilizados nas medições microgeométricas. Nas medições macrogeométricas, os instrumentos de medição mais utilizados são: paquímetros, micrômetros, relógios comparadores, blocos padrões etc.

No entanto, a simples obtenção destes equipamentos não significa necessariamente que a realização da quantificação de uma medida dimensional seja fácil. Na verdade o profissional que irá executar estas medidas, deverá possuir bons conhecimentos teóricos e práticos sobre metrologia.

Nas instituições de ensino e em pequenas empresas tem-se dificuldade para a realização de práticas de laboratório devido ao elevado custo dos equipamentos e instrumentos de medição, além de serem delicados.

Neste sentido, foi desenvolvido um software capaz de realizar a simulação destas práticas, bem como avaliar o usuário, permitindo que este interaja com o software e possa testar os

seus conhecimentos. Além disso o software desenvolvido apresenta páginas gráficas com exemplos práticos e resumos da teoria sobre o erro a ser medido, permitindo que o usuário possa esclarecer as suas dúvidas.

2. Descrição do Software

O software foi desenvolvido na Linguagem Visual Basic 3.0 podendo ser utilizado tanto no ambiente Windows 3.1 como no Windows 95. O programa contém explicações teóricas, simulações de medições e exercícios práticos para avaliar os conhecimentos do usuário com relação aos erros de forma, posição, orientação e batida.

A seguir serão apresentadas as principais telas deste software, contendo explicações sucintas a respeito de seu funcionamento e interação gráfica com o usuário.

2.1 Tela de Apresentação

A tela de apresentação, mostrada na figura 1, é um menu de escolha. Nela encontram-se pequenos menus com erros de forma, orientação, batida, posição, bibliografia e saída.



Figura 1 - Tela de apresentação dos submenus

Para o usuário aprender ou esclarecer as possíveis dúvidas sobre qualquer um dos erros geométricos, proposto pelo software, basta o usuário clicar com o mouse sobre o assunto referente. Por exemplo: um usuário que deseja obter informações sobre o erro de circularidade, basta clicar o submenu, doravante denominado botão, de circularidade no menu de erros de forma.

Caso o usuário deseje sair do software, basta apenas clicar o ícone de saída, que permanece sempre posicionado ao lado direito da tela de apresentação (vide figura 1).

3 Erros de Forma

O menu de escolha dos erros de forma, apresenta algumas práticas que podem ser acionadas pelo usuário na seqüência que for mais conveniente. Clicando-se qualquer um dos

botões deste menu, o usuário terá acesso imediato à simulação, teoria e prática da referida medição.

O menu de erros de forma é composto por retilidade (ou diferença de reta), circularidade, cilindridade, planicidade, linha e superfície qualquer.

3.1 Tela de Retilidade

A tela de retilidade, apresentada na figura 2, é composta por um pequeno menu de escolha, uma animação gráfica e, na parte superior, uma explicação sucinta do procedimento da medição utilizando-se como instrumentos de medição um relógio comparador e um desempenho.

Uma animação gráfica mostra o procedimento prático para realizar a medição de retilidade. Isto é feito, colocando-se a peça a ser medida sobre o desempenho e em seguida passa-se o relógio comparador sobre a peça. Durante esta operação os valores do erro, ponto a ponto, são registrados, para delimitar o campo de tolerância.

No menu de escolha denominado de "Controles", o usuário poderá escolher três opções para a realização da prática de um exercício aleatório monitorado pelo computador, aprofundar-se mais os conceitos teóricos sobre este tipo de erro ou voltar para a tela de apresentação.

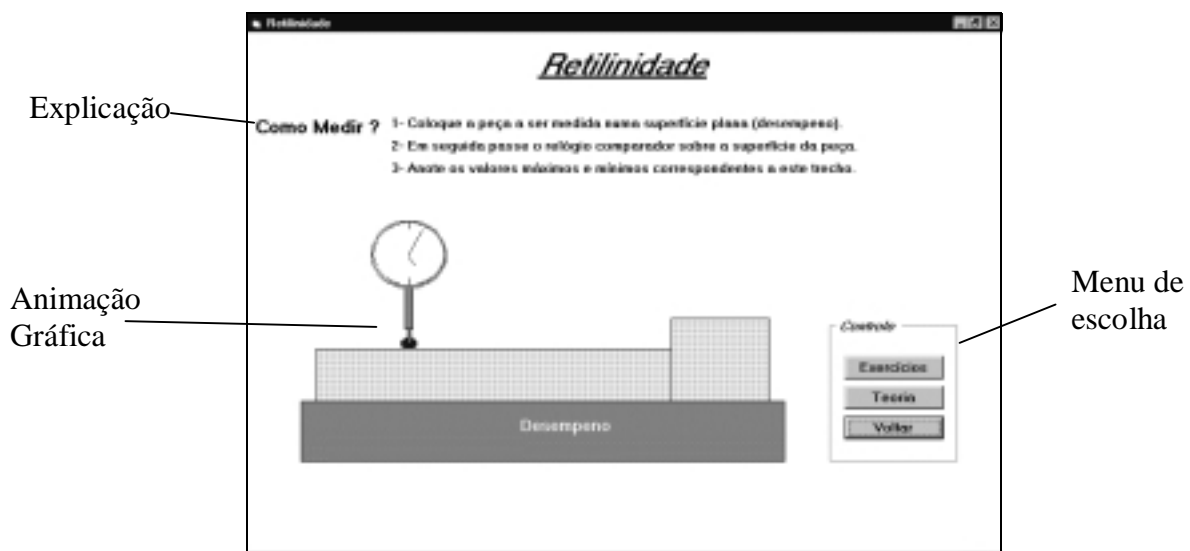


Figura 2 - Tela de retilidade.

3.2 Tela de Planicidade

Clicando-se sobre o botão planicidade aparecerá a tela de planicidade, apresentada na figura 3, que é composta por animação gráfica, teoria e exercícios.

A animação gráfica mostra o relógio comparador deslocando-se sobre a superfície de uma peça. Esta deve estar posicionada sobre três apoios num mesmo plano. Estes apoios não são alinhados numa mesma direção e também não são coincidentes afim de se obter um plano de referência. Isto é obtido com a utilização de três "macaquinhos" posicionados numa mesma altura, com relação a superfície que se deseja medir. É importante ressaltar que devido as limitações gráficas do Visual Basic 3.0, não foi possível realizar uma animação espacial, ou seja, com profundidade. Portanto o usuário deverá considerar que os três "macaquinhos" não estão localizados numa mesma linha.

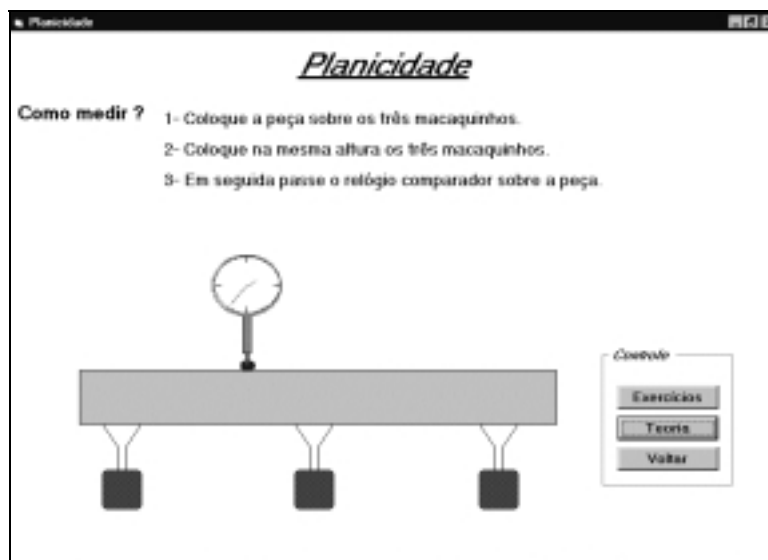


Figura 3 - Tela de planicidade

3.3 Tela de Circularidade

Acionando-se a tela de circularidade, o usuário irá se deparar com a animação gráfica de uma medição de erro de circularidade.

A tela de circularidade, apresentada na figura 4, mostra o relógio comparador posicionado na parte superior de uma peça (com geometria cilíndrica), vista de frente. A prática é realizada girando-se a peça em torno de seu centro, ao mesmo tempo em que um relógio comparador registra o perfil geométrico desta. Também é oferecida neste caso a opção do usuário acessar a parte de teoria e/ou exercício proposto.

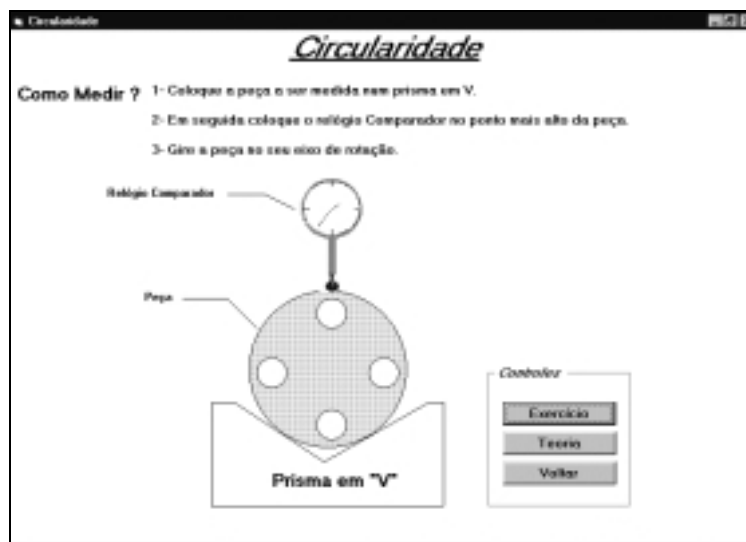


Figura 4 - Tela de circularidade.

3.4 Tela de Cilindricidade

A tela de cilindridade mostra uma peça em rotação, posicionada entre eixos. Durante o movimento de rotação, um relógio comparador translada sobre a peça no sentido longitudinal,

registrando os pontos que representam o perfil geométrico da peça para neste caso. Isto resulta num campo de tolerância cilíndrico.

Devido aos recursos gráficos disponíveis e a posição em que a peça se encontra, torna-se difícil de se demonstrar que a peça está rotacionando em torno do seu eixo central. Para evitar possíveis confusões, colocou-se uma observação na parte inferior da peça alertando sobre este fato e uma seta indicando a rotação da peça, conforme mostra a figura 5.

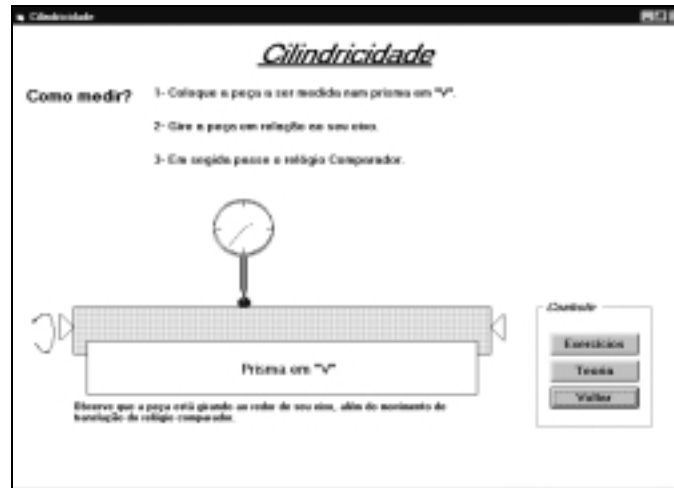


Figura 5 - Tela de cilindricidade.

3.5 Telas de Linha e Superfície Qualquer

Estas duas práticas ficaram substancialmente comprometidas na parte de sua animação gráfica e de seus respectivos exercícios devido a limitação gráfica na representação tridimensional dos instrumentos de medida, pelas limitações da linguagem Visual Basic versão 3.0, em animar o relógio comparador para percorrer as direções horizontal, vertical e diagonal.

Como as práticas de linha e superfície qualquer requerem trajetórias diferentes das convencionais, não foi possível a sua realização. Espera-se para um futuro breve o surgimento de uma nova versão do Visual Basic, onde isto se torne possível.

No entanto, o usuário poderá ter acesso às teorias de linha qualquer e superfície qualquer. Cada uma destas apresentará informações teóricas referentes a cada caso, do procedimento para a obtenção do campo de tolerância e um exemplo de cada, conforme pode ser observado nas figuras 6 e 7.

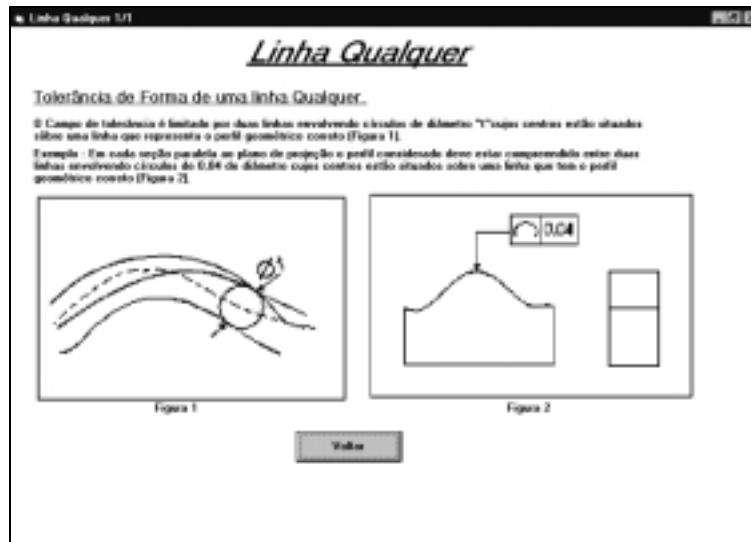


Figura 6 - Tela de linha qualquer



Figura 7 - Tela de superfície qualquer

4 Erros de Orientação

De forma similar ao que foi realizado no item que trata dos erros de forma, o usuário também poderá melhorar os seus conhecimentos sobre os erros de orientação. Este tipo de desvio de posição é definido para elementos (linhas ou superfícies) que têm pontos em comum através de intersecção de suas linhas ou superfícies, os quais são classificados em: paralelismo, perpendicularismo e inclinação.

4.1 Tela de Paralelismo

Ao entrar na tela de paralelismo, o usuário assistirá os passos principais para a realização da medição deste erro. Para a realização desta prática, inicialmente, uma peça cilíndrica vazada é encaixada no pino que está paralelo ao desempenho, conforme é apresentado nas figuras 8 e 9.

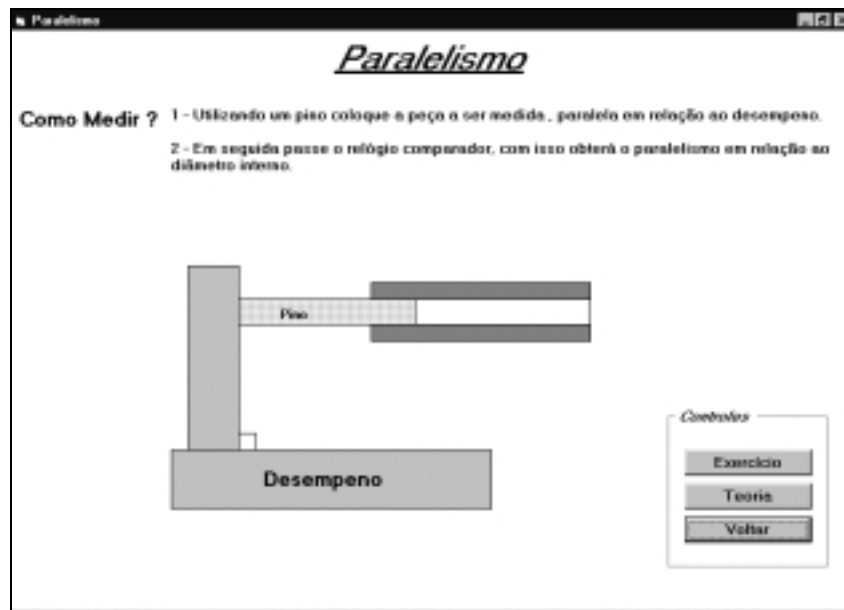


Figura 8 - Encaixando a peça no pino

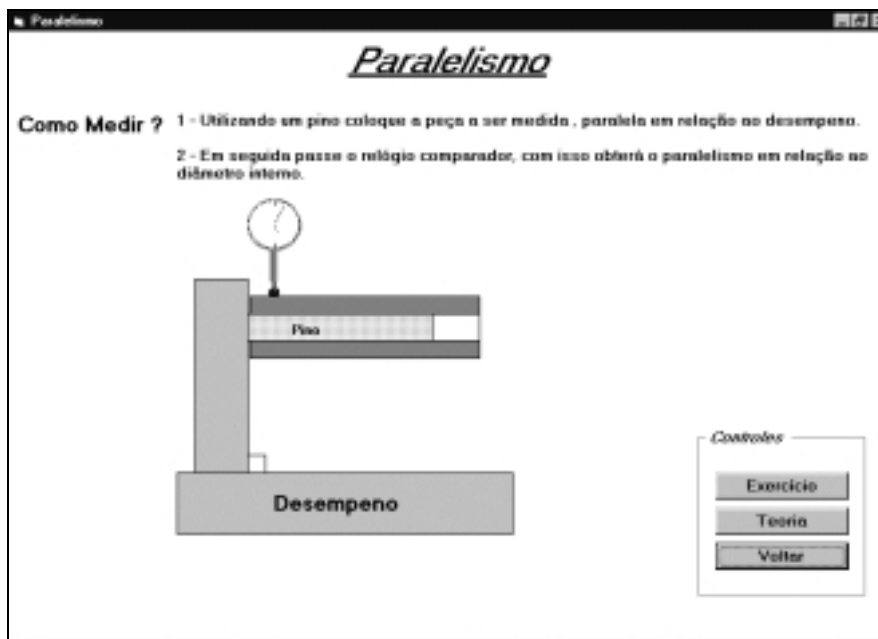


Figura 9 - Medindo o erro de paralelismo

Em seguida, passe-se o relógio comparador na superfície externa da peça, e assim terá realizado a medição de paralelismo da superfície externa em comparação a superfície interna.

4.2 Tela de Perpendicularidade

Na tela de perpendicularidade, o usuário terá acesso aos principais passos para realizar uma medição de perpendicularismo.

Inicialmente deve-se posicionar a peça de forma cilíndrica vazada no pino, conforme mostra a figura 10, e em seguida deve-se passar o relógio comparador na superfície que deseja verificar o erro de perpendicularismo, apresentado na figura 11.

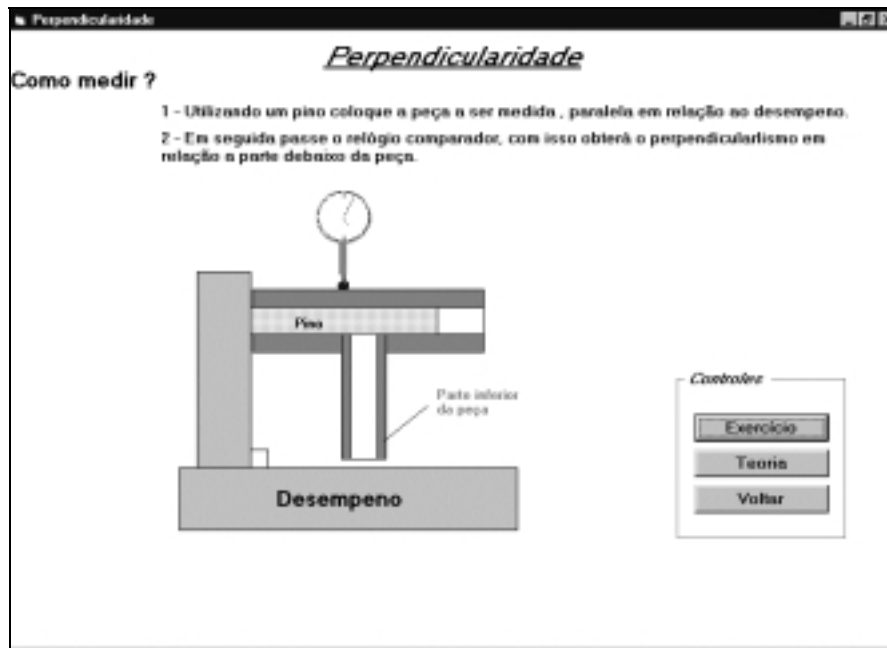


Figura 11 - Relógio comparador passando sobre a peça

5 - CONCLUSÕES

Este software é um protótipo autodidático, que tem como diretriz principal o ensino de metrologia relacionado as práticas de erros de forma, orientação, batida e posição nas instituições educacionais técnicas.

O software apresenta as seguintes vantagens:

- Proteção contra a pirataria, devido ao sistema de senha;
- Possui todas as telas e imagens dos erros de forma, posição, orientação e batida;
- Permite e obriga que o usuário obedeça a ordem dos passos preestabelecida para a realização da prática, com a necessidade de ativar e desativar os botões necessários;
- Monitora os passos do usuário e o avalia;

A interação entre software e usuário é amigável, tornando-se este software um “video-game educativo”, facilitando a aprendizagem.

7 - REFERÊNCIAS

AGOSTINHO, O. L., RODRIGUES, A. C. S., LIRANI, J. Tolerância, ajuste, desvios e análise de dimensões. São Paulo: Edgard Blücher, 1981. SCHNEIDER, C.A.; FLESCHE, C. A.; Apostila metrológica dimensional básica. Universidade Federal de Santa Catarina, 1985.

ABNT NBR ISO 10012-1. Requisitos de garantia da qualidade para equipamento de medição. 1993.

FURUKAWA, S., BIANCHI, E.C. Aspectos teóricos e práticos em metrologia dimensional mecânica e de forma. Bauru: Faculdade de Engenharia e Tecnologia, UNESP, Câmpus de Bauru, 1995, 89p. (Mimeogr.).

LIRANI, João; Introdução à metrologia industrial. Universidade de São Paulo, São Carlos, 1985.

WHITEHOUSE, D.J. Hand book of surface metrology. Bristol: Institute of Physics, 1994.

ARQUITETURA DOS SISTEMAS DE MANUFATURA

Eng. Marcelo Moretti Fioroni¹

Prof. Dr. Antônio Batocchio²

Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Departamento de Engenharia de Fabricação, Caixa Postal 6122, CEP: 13083-970, Campinas/SP, Brasil;
Fone: (019) 788-3290; FAX: (019) 289-3722. E-mail: mfioroni@mii.zaz.com.br¹,
batocchi@fem.unicamp.br²

Resumo

A comunidade acadêmica e industrial vem obtendo resultados expressivos no desenvolvimento dos sistemas de manufatura, em uma busca constante para satisfazer os crescentes anseios do público consumidor. Propostas como o Sistema Ágil e o Sistema Holônico, buscam conjugar grande personalização de produtos com boa velocidade de resposta e baixo custo. O sistema Ágil engloba várias tecnologias, como agentes inteligentes na manufatura, algoritmos genéticos e inteligência artificial. Já os Sistemas Holônicos buscam atingir um novo patamar de flexibilidade e agilidade, trabalhando com uma organização semelhante à encontrada nos sistemas orgânicos, onde as células se apresentam como estruturas estáveis e auto-suficientes, que se agrupam formando estruturas maiores, os tecidos, de estabilidade equivalente. Na mesma velocidade em que avançam a tecnologia e os sistemas de manufatura, novas arquiteturas para estes sistemas são rapidamente criadas. O presente artigo reúne as propostas mais recentes ou relevantes e discute suas características, tecendo uma comparação entre elas e oferecendo uma visão abrangente. Indica também as tendências atualmente perceptíveis na proposição de novas arquiteturas capazes de atender à manufatura do próximo milênio.

Palavras-chave: Sistemas de Manufatura, Arquitetura dos Sistemas de Manufatura, Manufatura Ágil, Sistema Holônico de Manufatura

1. INTRODUÇÃO

É fato notório que a globalização está impondo novas exigências de produção para as empresas. O mercado está cada vez mais exigente quanto à qualidade dos produtos e exige também uma crescente personalização. Cada consumidor deseja que o produto esteja adaptado às suas próprias necessidades. Neste ambiente, a inovação deixa de ser algo apenas desejável para tornar-se regra. As conseqüências disto para a indústria é que seus sistemas de manufatura precisam ser capazes de fabricar produtos personalizados, em lotes pequenos, rapidamente, a um baixo custo e mantendo os níveis de qualidade. Para conseguir tal resultado, novos sistemas de manufatura tem sido propostos pelas instituições de pesquisa com o apoio da indústria.

Cada vez mais automatizados, os sistemas de manufatura atuais lançam mão do que há de mais avançado em termos de tecnologia, fazendo uso intensivo de computadores, redes de comunicação e procedimentos robotizados. A arquitetura dos sistemas de manufatura reflete fielmente as características do mercado atendido por ele. Um exemplo disso é o modelo de

linha de produção criado por Henry Ford para a fabricação do “Ford T”, cujo objetivo era produzir o mais rapidamente possível, para um mercado que não se importava com a personalização e onde havia grande demanda pelo produto (automóvel).

Neste trabalho, pretende-se mostrar uma retrospectiva sobre a arquitetura dos sistemas de manufatura até os tempos atuais, acompanhando seu gradativo avanço na busca constante de atender à sociedade de consumo da época. Nota-se claramente que a tarefa de atender ao mercado foi ficando cada vez mais complexa e árdua, exigindo um desenvolvimento crescente de novos sistemas. Nos tempos atuais, onde o mercado atingiu níveis de exigência nunca vistos antes, a comunidade científica e empresarial se desdobra para criar sistemas de manufatura capazes de atender a demanda atual e futura. Para enfrentar este grande desafio algumas iniciativas tem sido criadas, como o IMS - Intelligent Manufacturing Systems Consortium (IMS, 1999), que pretende criar o estado da arte em sistemas de manufatura. Apresenta-se aqui algumas propostas já elaboradas para estes sistemas e suas respectivas arquiteturas, como os sistemas de manufatura ágil e o sistema holônico de manufatura, uma das propostas do supracitado IMS, e que se encontra atualmente em estudo no Brasil pelo GPHMS – Grupo de Pesquisa em Sistema Holônico de Manufatura, sediado na Unicamp (Batocchio & Fioroni, 1999).

2. DEFINIÇÃO DE ARQUITETURA PARA SISTEMAS DE MANUFATURA

Conforme explicado por Wyns *et al.*(1996) a arquitetura de um sistema de manufatura é o resultado do projeto do sistema, onde são especificadas as funções dos componentes, suas interfaces, interações e restrições. Tem por objetivo descrever de forma resumida toda a complexidade dinâmica de um sistema através de modelos simples. Entre as vantagens da definição de uma arquitetura, cita-se (Wyns, 1996):

- Fornecendo uma abstração do sistema complexo de forma simples, a arquitetura auxilia o projetista a definir as interfaces e interações entre os diversos componentes;
- Quando é necessária uma alteração no processo, reduz-se o impacto das mudanças pela possibilidade de focalizar apenas as áreas que necessitam de maiores mudanças;
- A arquitetura indica os componentes vitais para o sistema, os quais não deverão ser alterados quando da adaptação deste sistema para novos usos, caso em que se estaria aplicando uma arquitetura diferente. Comparando com a arquitetura de edificações, seria como mudar as paredes de sustentação de uma casa;
- A arquitetura é um meio de comunicação importante durante o processo de projetar ou reprojeter o sistema, fornecendo base para discussões entre as partes envolvidas.

À semelhança da arquitetura de edificações, a arquitetura de sistemas de manufatura também pode ser apresentada de várias maneiras. No caso de um prédio, sua arquitetura é apresentada aos clientes na forma de uma maquete, mas quando apresentada à construtora, são fornecidas várias outras informações, como materiais utilizados, locais de passagem de fios, etc. Assim, de uma arquitetura de sistema de manufatura, espera-se conseguir:

- Uma terminologia unificada, sem ambigüidades e conhecida por todos os envolvidos;
- Simplicidade no projeto do sistema, permitindo um desenvolvimento fácil e rápido da sua arquitetura;
- Maior qualidade no desenvolvimento dos sistemas, por se basearem em conceitos confiáveis, já comprovados pela arquitetura;
- Interfaceamento e a possibilidade de reaproveitamento de módulos da arquitetura em diferentes projetos ou gerações de sistemas;
- Desenvolvimento ou implementação de tarefas que podem ser divididas entre diferentes times, permitindo que cada um aplique suas melhores especialidades ou equipamentos;

- Fácil identificação das soluções utilizadas. A arquitetura precisa indicar e justificar claramente quando e como cada estágio do desenvolvimento recebeu implementações de engenharia.

A arquitetura de um sistema pode ser formulada no estilo descritivo ou prescritivo. O estilo descritivo define uma codificação particular dos elementos de projeto e é usado durante as discussões entre o projetista e o cliente. O estilo prescritivo se limita a apresentar os elementos e suas configurações. Este estilo é aquele usado nos planos de construção de um edifício, nos quais o construtor se baseia para realizar seu trabalho.

Neste estudo, as arquiteturas são apresentadas no estilo descritivo, de modo a permitir seu fácil entendimento e viabilizar sua abordagem neste estudo.

3. OS SISTEMAS DE MANUFATURA PROPOSTOS ATÉ A ATUALIDADE

Os sistemas de fabricação existem desde os primórdios da humanidade, quando o homem começou a perceber que poderia realizar melhor suas tarefas com o auxílio de instrumentos. Prova disso é a descoberta arqueológica de uma "fábrica" de ferramentas no Quênia com a idade de 2,34 milhões de anos, conforme noticiado na Folha de São Paulo (1999). O homem evoluiu muito desde então, e a busca por melhores meios de realizar seus trabalhos permaneceu.

3.1 O Sistema Funcional

Um dos primeiros sistemas de fabricação elaborados foi o chamado "Sistema Funcional", onde os recursos fabris da empresa com a mesma função são agrupados em setores. Desta forma, a matéria prima é transportada em lotes ao longo dos diferentes setores, onde sofre operações específicas em cada um deles. É caracterizado pela sua grande flexibilidade, já que qualquer operação pode ser feita em qualquer ordem. Também é pouco suscetível a paradas de produção devido a quebras de máquinas, e o seu agrupamento facilita o treinamento da mão de obra. O sistema funcional apresenta tempo de fabricação longo, dificuldade na manutenção da qualidade do produto e no gerenciamento do sistema, que se torna muito complexo. Uma grande quantidade de estoque intermediário é necessário para o seu funcionamento, o que exige um investimento considerável (Batocchio, 1992).

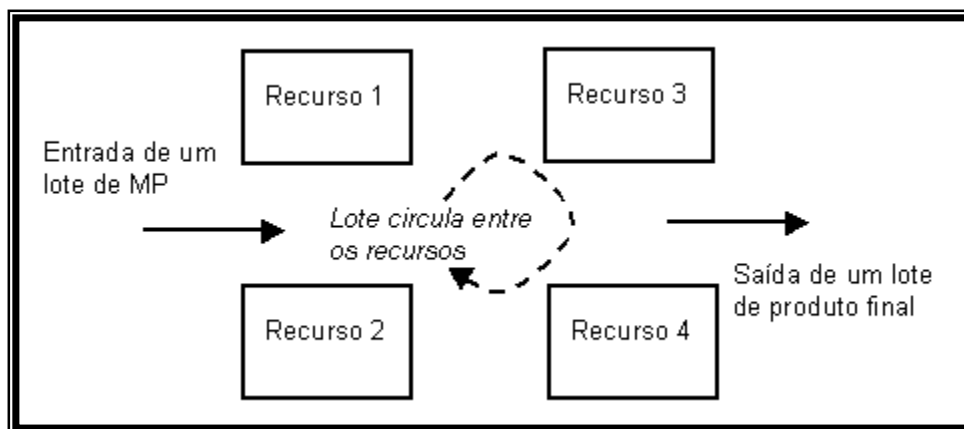


Figura 1. Funcionamento de um Sistema Funcional

Trata-se do sistema adequado para a fabricação de lotes pequenos, e prazos de entrega curtos. É o sistema ideal quando os produtos a serem fabricados são muito diversificados, ou são alterados freqüentemente. A Figura 1 apresenta esquematicamente o funcionamento deste

sistema, onde os recursos representam uma ou mais máquinas de mesmo tipo ou pessoas, onde ocorre uma das etapas de fabricação

3.2 Sistema em Linha

Quando os produtos a serem fabricados tem pequena variação e exigem muitas operações, surgem empecilhos para a fabricação pelo sistema funcional. Para este caso, foi desenvolvido o sistema de fabricação em linha, no qual os diferentes equipamentos permanecem dispostos na seqüência das operações exigidas para a fabricação do produto. Desta forma, obtêm-se uma boa velocidade de fabricação, permitindo atender rapidamente aos pedidos para grandes quantidades (Batocchio, 1992).

Os pontos fracos deste sistema residem na sua grande sensibilidade a problemas na linha. Uma máquina quebrada pode interromper completamente a produção. Outro ponto negativo é a sua baixa flexibilidade e alto custo de implantação, que o tornam aplicável apenas quando a empresa possui vários produtos similares com tempo de vida alto. A representação esquemática deste sistema é apresentada na Figura 2.

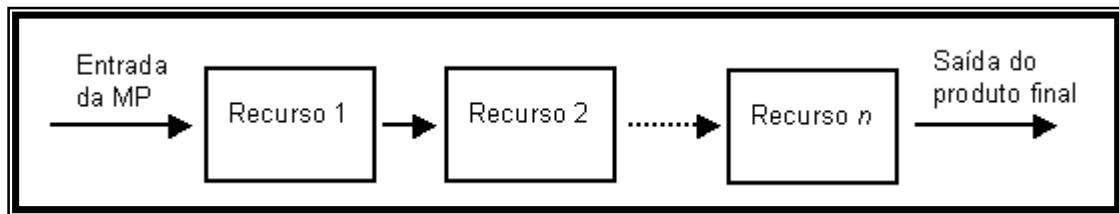


Figura 2. Funcionamento de um Sistema em Linha

3.3 Sistema Celular

O sistema celular destina-se à fabricação de peças que tenham certa similaridade, em lotes pequenos. Caracteriza-se pelo agrupamento de todas as máquinas usadas para a fabricação de um determinado "grupo" ou "família" de peças, as quais sofrem operações em todas ou algumas destas máquinas. O agrupamento de máquinas é chamado "célula" (Batocchio, 1992).

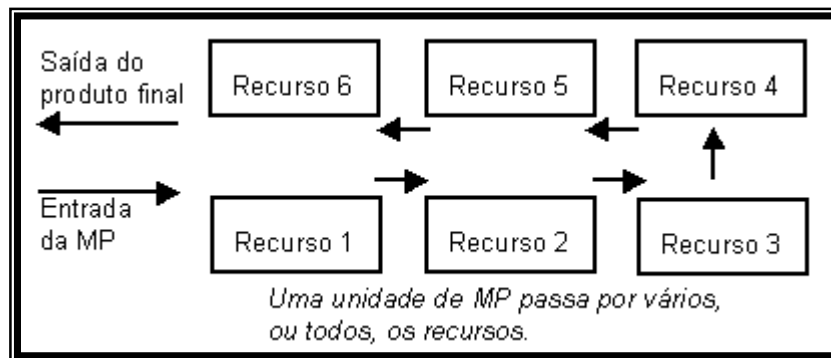


Figura 3. Funcionamento de um Sistema Celular

O sistema celular apresenta as vantagens de reduzir os estoques intermediários, permitir a fabricação de lotes pequenos, aproveitamento ótimo da mão de obra e equipamentos, e permitir a manutenção de altos níveis de qualidade. Suas desvantagens ficam por conta de sua também grande sensibilidade a imprevistos como quebras de máquina, paradas para manutenção, ou acontecimentos semelhantes; flexibilidade relativamente baixa, pois se

houver uma variação muito grande no desenho da peça a ser fabricada, a célula passa a não mais ser capaz de produzi-la, seja pela ausência de uma determinada máquina, ou um processo que passa a ser muito demorado, prejudicando a fabricação das outras peças do grupo. A Figura 3 ilustra o funcionamento de um sistema celular. A matéria-prima, em seu processo de fabricação, passa por várias ou todas as máquinas. Este sistema é aplicado atualmente na “manufatura enxuta” (Lean Manufacturing), principalmente quando o produto é pequeno e manipulável por operadores.

3.4 FMS – Sistemas de Manufatura Flexível

O avanço da tecnologia, que proporcionou novos recursos fabris como robôs, centros de usinagem automatizados e etc. O conceito de FMS envolve um alto nível de automação, onde a fabricação é realizada por centros de usinagem multifuncionais e robôs dotados de ferramentas. A transferência entre as máquinas é feita por veículos robotizados (AGVs) e o posicionamento/setup das máquinas é realizado por robôs manipuladores ou similares. A matéria prima, produtos acabados, ferramentas a até produtos em estágio intermediário são armazenados em um armazém automatizado. Todos estes elementos são ligados a um computador central que coordena todas as ações e recebe ordens de fabricação diretamente do escritório central (Asai & Takashima, 1994).

Desta forma, a produção segue rigorosamente as tendências detectadas pela pesquisa de mercado e é capaz de mudar rapidamente, passando a fabricar um produto diferente. Isto caracteriza sua flexibilidade. A Figura 5 mostra um esquema do funcionamento de um sistema FMS.

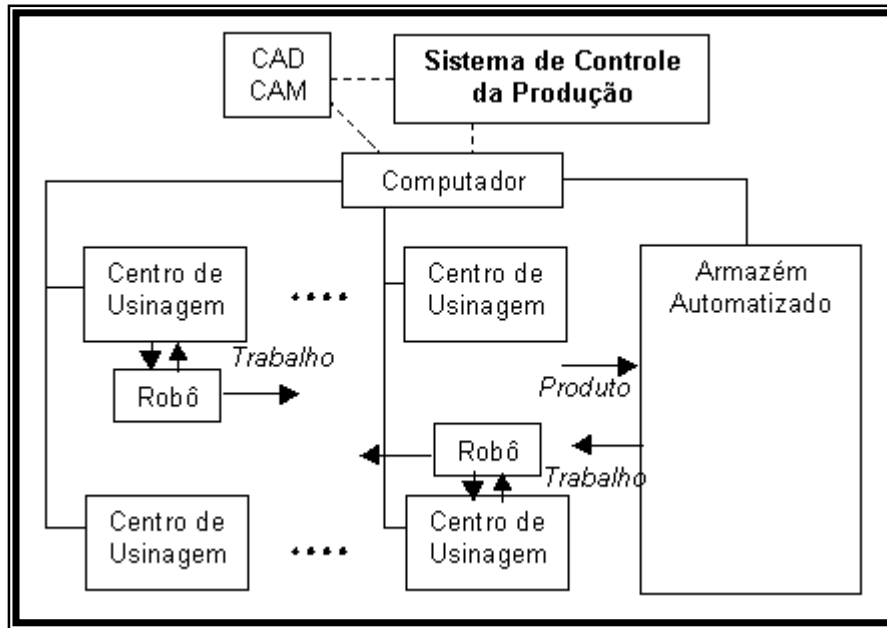


Figura 5. Funcionamento de um sistema FMS (Asai & Takashima, 1994)

3.5 CIM – Manufatura Integrada por Computador

Com o avanço em constante aceleração da informática e a disponibilidade cada vez maior dos computadores, estes começaram a encontrar aplicação nas empresas. O desenvolvimento de softwares para CAE, CAD/CAM acelerou ainda mais este processo.

O uso dos computadores permitiu uma maior integração entre os diversos departamentos das empresas através de redes informatizadas. Otimizou processos e permitiu um

gerenciamento mais eficiente dos recursos da empresa. Segundo Asai & Takashima (1994), existem dois tipos de CIM:

- “CIM in series”, que descreve uma aplicação vertical dos computadores, cobrindo operações de pesquisa de mercado, planejamento de produto e projeto, planejamento da produção e controle, teste e distribuição;
- “Cross CIM”, que descreve uma aplicação horizontal dos computadores, que se constitui em ligações bidirecionais entre todas as fábricas e unidades da empresa com o escritório central automatizado, que centraliza as informações e decisões.

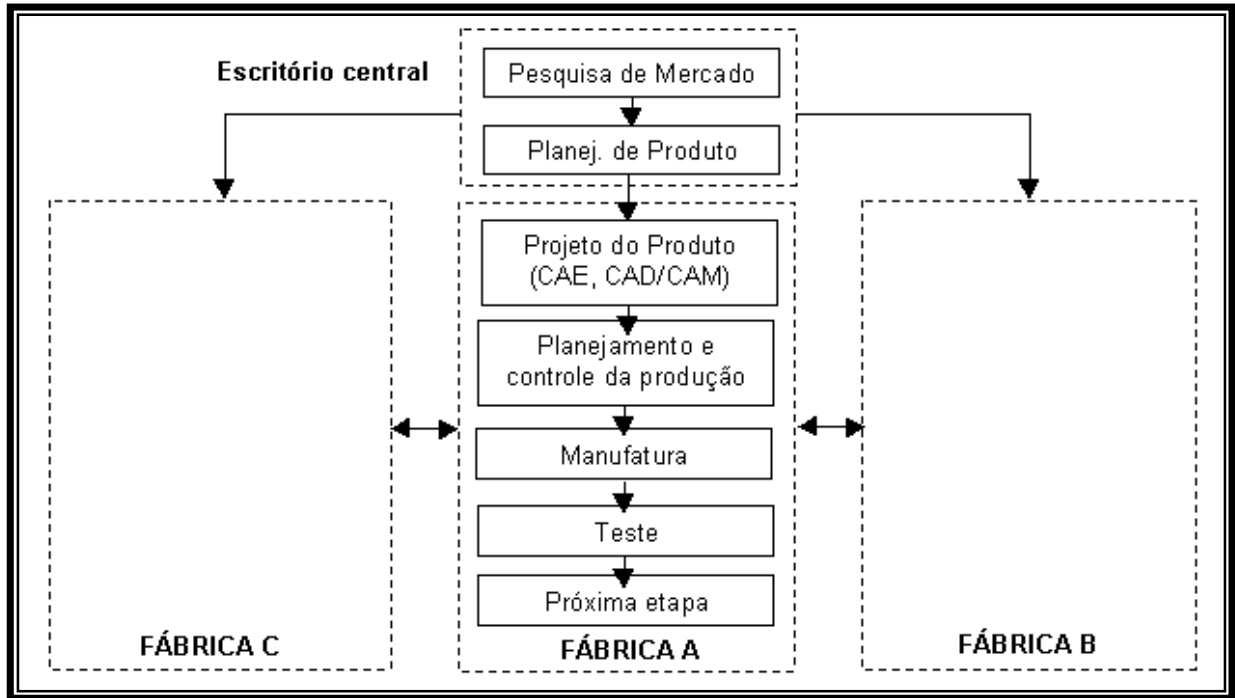


Figura 4. Funcionamento de um sistema CIM (Asai & Takashima, 1994)

A Figura 4 apresenta um diagrama do funcionamento de uma empresa utilizando o CIM. O escritório central coordena as ações das fábricas, de acordo com as tendências detectadas pela pesquisa de mercado.

4. AS PROPOSTAS PARA O PRÓXIMO SÉCULO

Na busca pelo sistema de manufatura capaz de atender aos crescentes anseios do mercado, novos sistemas tem sido desenvolvidos ou propostos. A grande variedade de propostas torna inviável a apresentação de todas neste espaço. Devido a isto, apresenta-se a proposta mais “amadurecida” e já aplicada em algumas empresas, que é o Sistema de Manufatura Ágil, e uma das propostas mais promissoras para o próximo século, que é o Sistema Holônico de Manufatura.

4.1 Sistema de Manufatura Ágil

O conceito de Manufatura Ágil é bastante abrangente e envolve todos os aspectos da empresa. A estrutura da manufatura Ágil é suportada por três recursos principais: gerenciamento inovador de organizações, recursos humanos altamente capacitados e com poder de decisão, e tecnologias inteligentes e flexíveis (Kidd, 1995). A agilidade é alcançada através da integração destes três recursos em um sistema coordenado e independente. Em

termos simples, a manufatura ágil pode ser considerada como a integração entre a organização, pessoal altamente capacitado e tecnologias avançadas, visando alcançar cooperação e inovação para fornecer aos seus clientes produtos personalizados e de alta qualidade. Este conceito é ilustrado na Figura 6.

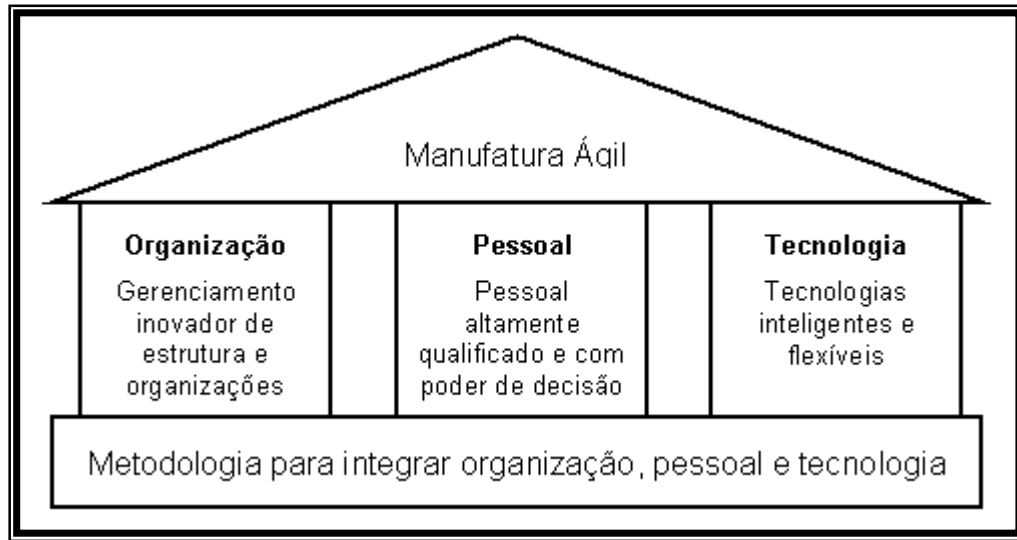


Figura 6. Estrutura da manufatura Ágil (Kidd, 1994)

Outra definição de manufatura Ágil é fornecida por Wandmacher (1994): “A habilidade de prosperar em um ambiente de contínua e inflexível mudança”. Os “três recursos” citados por Kidd (1994) tem sido desenvolvidos separadamente para atingir o conceito de agilidade, e várias empresas os tem aplicado com sucesso. Atualmente, a maioria das empresas de tecnologia aplicam o conceito de Manufatura Ágil. No Brasil existem exemplos bastante claros entre as empresas provedoras de acesso e serviços para Internet. Neste mercado o ambiente de mudanças é evidente. No início, o serviço era cobrado por um número fixo de horas, cujo excesso incidia em taxas maiores. Logo a seguir veio o acesso ilimitado, onde o cliente não tinha mais limite de horas pagando um preço fixo. O ponto culminante foi o acesso gratuito, onde simplesmente não há mais cobrança do serviço de acesso. Tudo isso em um espaço de apenas 7 anos.

4.2 Sistema Holônico de Manufatura

O HMS – Holonic Manufacturing System é um dos temas em estudo pelo IMS (1999). Trata-se de um sistema que busca atingir a estabilidade e capacidade de reação encontrada nos sistemas orgânicos e sociais. É baseado nas observações do filósofo alemão Arthur Koestler, que propôs a palavra “holon” como a representação de uma estrutura que é independente e estável, mas é parte integrante de uma estrutura maior, também estável e independente, como as células orgânicas que se unem, formando os tecidos (Wyns, 1996).

Uma arquitetura de referência foi proposta por Wyns (1999), chamada PROSA, que divide o sistema de manufatura em três “holons” principais: um holon de produto, responsável pela informação sobre o produto (método de fabricação, materiais e ferramentas necessários, nível de qualidade, etc.), um holon de recurso, que abrange os recursos à disposição da empresa (pessoal, máquinas, energia elétrica, meios de transporte, etc.) e um holon de pedido, que representa o pedido do cliente dentro do sistema (prazo de entrega, tipo de produto, etc.). Um diagrama explicativo da arquitetura PROSA está apresentado na Figura 7

Adicionalmente a estes três holons, são agregados holons secundários, chamados de especialistas, que assumem tarefas específicas. Por exemplo, um holon especialista de seqüenciamento poderia fornecer informações para os holons de pedido e recurso, de modo a capacitá-los a tomar as melhores decisões.

O sistema holônico agrega as características de um controle hierárquico e de um heterárquico. No exemplo anterior, os holons trabalham de forma independente, recebendo as informações do holon especialista de seqüenciamento como “conselhos”, não como um comando, de forma que quando ocorre um evento imprevisto, este ignora os “conselhos” e toma suas próprias decisões.

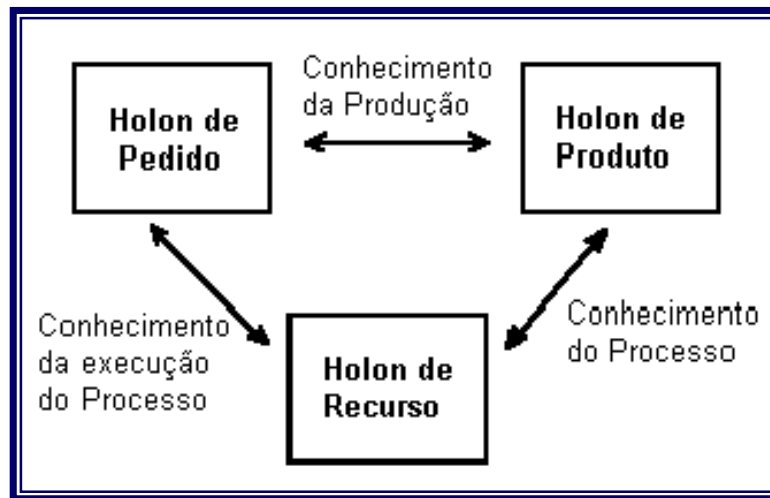


Figura 7. Estrutura da arquitetura PROSA (Wyns, 1999)

5. COMPARAÇÕES E CONSIDERAÇÕES

Analisando-se todos os sistemas apresentados de forma seqüencial, nota-se que cresceu ao longo do tempo a preocupação em atender aos anseios do cliente. Há um grande contraste entre o sistema em linha, usado por Ford para fabricar o Ford “T”, que era um produto sem qualquer variação ou personalização, e os sistemas aplicados atualmente, buscando fabricar um produto diferente para cada cliente, de forma a atender seus anseios pessoais.

Nos sistemas mais antigos, o componente cliente nem mesmo aparece. Este começa a aparecer no sistema CIM, como o elemento “pesquisa de mercado”, que busca encontrar uma tendência. Esta tendência tem a intenção de determinar grupos de pessoas com as mesmas necessidades, o que iria gerar a criação de um novo produto. Já no sistema Ágil, a ligação com o cliente já é mais estreita, colocando toda a empresa a seu serviço, buscando atender ao seu desejo individual e não mais de um grupo de pessoas consideradas “iguais”.

No sistema holônico, este conceito vai além, colocando o desejo do cliente como parte essencial do próprio sistema de manufatura (“holon de pedido”). Como nos explicam Batocchio *et al.* (1999), o Sistema Holônico de Manufatura reúne componentes importantes do sistema Ágil e de outros sistemas, podendo ser considerado em um nível mais elevado do que o da manufatura ágil.

6. CONCLUSÕES

Esta análise torna claro que os sistemas de manufatura a serem desenvolvidos futuramente deverão ter sua estrutura cada vez mais centralizada no cliente, buscando

estabelecer com este uma ligação cada vez mais estreita e permitindo a ele um controle cada vez maior sobre o resultado final do sistema de manufatura, que é o produto.

Os sistemas que não tiverem foco no cliente certamente estarão fadados ao fracasso. Está claro que o mercado assumiu uma tendência irreversível.

As novas tecnologias disponibilizadas e popularizadas atualmente, como a Internet, são capazes de estreitar o relacionamento empresa-cliente, e devem ser aproveitadas nos novos sistemas de manufatura.

7. AGRADECIMENTOS

Os agradecimentos dos autores à FAPESP pela manutenção de uma bolsa de mestrado a um dos autores e ao RECOPE/FINEP/BID pelos recursos computacionais disponibilizados a este trabalho.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- "Arqueologia" - Folha de São Paulo, Caderno Mais!, p.11 - 09/05/1999.
- Asai, K. and Takashima, S., "Manufacturing, Automation Systems and CIM Factories", Chapman & Hall, Japan, 1994.
- Batocchio, A. – "Tecnologia de Grupo", Apostila do curso IM190 - Unicamp - SP. 1992.
- Batocchio, A. and Fioroni, M. M., "GPHMS Home Page – Grupo de Pesquisa em Sistema Holônico de Manufatura" [online]. Disponível na Internet via WWW. URL: <http://www.fem.unicamp.br/~defhp/index.htm> . Arquivo consultado em 11 de dezembro de 1999.
- Batocchio, A., Fioroni, M. M.; Georges, M. R. R.; Souza, A. C.; Rosa, A. B. and Franco, G. N., "Manufatura Ágil X Sistema Holônico de Manufatura", Apresentado no IV SBAI - Simpósio Brasileiro de Automação Inteligente, São Paulo, SP, 08-10 de Setembro de 1999.
- IMS, "IMS – Intelligent Manufacturing Systems Home Page" [online]. Disponível na Internet via WWW. URL: <http://www.ims.org> . Arquivo consultado em 11 de dezembro de 1999.
- Kidd, P. T., "Agile Manufacturing – Forging New Frontiers", Addison-Wesley, England, 1994.
- Wandmacher, R. R., "Information Management as a Key to Agile Manufacturing", presented at Manufacturing Information Systems, Detroit, October 24, 1994.
- Wyns, J., Brussel, H. V., Valkenaers, P. and Bongaerts, L., "Workstation Architecture in Holonic Manufacturing Systems", presented at 28th CIRP International Seminar in Manufacturing Systems, may 15-17, 1996, Johannesburg, South Africa, p220-231. "Cirp Journal on Manufacturing Systems", Vol.26, No 4.
- Wyns, J., "Reference Architecture for Holonic Manufacturing Systems: PROSA", Doctoral Thesis, Katholieke Universiteit Leuven, Mechanical Engineering Dept., Belgium, 1999.

UM ESTUDO SOBRE CONTROLE ROBUSTO DE ESTRUTURAS FLEXÍVEIS ATIVADAS POR ESCOAMENTO DE FLUIDOS

Celiane Costa Machado

Sebastião Cícero Pinheiro Gomes

Álvaro Luiz De Bortoli

Fundação Universidade Federal do Rio Grande, Departamento de Matemática, 96201-900, Rio Grande, RS, Brasil, E-mail:mmaccm@super.furg.br, E-mail:dmtscpg@super.furg.br

Universidade Federal do Rio Grande do Sul, PPGMAp, 91509-900, Porto Alegre, RS, Brasil, E-mail:dbortoli@mat.ufrgs.br.

Resumo

O presente trabalho apresenta uma abordagem sobre o desenvolvimento de controladores para atenuar de forma ativa as vibrações de uma estrutura flexível sujeita a uma perturbação externa, oriunda do escoamento de um fluido em contato com a estrutura. Desenvolveu-se, inicialmente, um controlador robusto com a técnica LQG/LTR (*Linear Quadratic Gaussian / Loop Transfer Recovery*). Simulações foram realizadas, verificando-se características de robustez na rejeição da perturbação externa devida ao fluido e na rejeição de ruídos de medida, considerando a planta nominal igual à real. A mesma robustez não se verificou quando foram realizadas variações paramétricas entre as plantas nominal e real. Foi desenvolvido, ainda, um outro controlador por alocação de pólos do regulador e do filtro. Este último controlador foi projetado de forma que a malha direta obedeça às barreiras de robustez da estabilidade e do desempenho. Os resultados foram bons, tendo-se verificado robustez da estabilidade em presença de significativas variações paramétricas entre as plantas nominal e real.

Palavras-chave: Modelagem, Manipulador, Flexível, Controle, Robustez.

1. INTRODUÇÃO

O controle de estruturas flexíveis tem sido um assunto bastante pesquisado nos últimos vinte anos, devendo-se este interesse principalmente às aplicações espaciais (Kim *et al.*, 1997) e (Xie *et al.*, 1997). A maioria das aplicações concentra-se no controle de robôs manipuladores com elos flexíveis (Cannon *et al.*, 1984), (Kojima, 1990) e (Schmitz, 1985). Mais recentemente, trabalhos já indicam para a necessidade de se desenvolver controladores robustos para estruturas flexíveis (Soares, 1997) e (Vincent, 1996). Os trabalhos citados anteriormente utilizam controle ativo, embora também existam estudos sobre a utilização de controle passivo (Furusho *et al.*, 1997). Apesar de todas as pesquisas já realizadas, trata-se de um assunto que está longe de ser

totalmente dominado, existindo ainda diversos desafios científicos que se tornam objetivos a serem vencidos pela comunidade que pesquisa este assunto. Este trabalho pretende abordar um destes desafios: controlar uma estrutura flexível sujeita a perturbações externas. Estas perturbações são consideradas aqui como provenientes do escoamento de um fluido interagindo com a estrutura. Situações práticas semelhantes a esta aqui tratada em teoria podem acontecer, por exemplo, em robótica sub-aquática (Watonabe *et al.*, 1998). Em aplicações práticas também há, geralmente, dificuldades técnicas em instrumentalizar a planta física com muitos sensores, além de evidentemente encarecer o projeto. Considerou-se então no presente trabalho, que apenas a posição angular do rotor ou cubo (atuador na Fig. 1) era observada, fato que impõe dificuldades extras ao projeto de controladores que idealmente deveriam ter o retorno do estado completo para o caso de estruturas flexíveis. Desenvolveu-se controladores baseados na técnica LQG/LTR (Cruz, 1996) e ainda, baseados em alocação de pólos com estimação de estado. Os controladores foram desenvolvidos considerando-se o modelo nominal da planta sem a presença do fluido, que entra nas simulações como uma perturbação externa. Além desta perturbação, realizou-se também simulações considerando-se variações em parâmetros tais como massa específica da estrutura e módulo de elasticidade, variações estas com relação ao modelo nominal que serviu de base para o projeto dos controladores. O objetivo final foi o de testar tanto a robustez ao desempenho quanto à estabilidade.

2. MODELAGEM DINÂMICA FLUIDO-ESTRUTURA

A estrutura considerada no presente artigo é constituída de uma lâmina flexível articulada em uma extremidade (atuador) e livre na outra. Esta estrutura ainda está sujeita ao contato com um fluido escoando a uma determinada velocidade.

2.1 Modelo Estrutural

Existem alguns formalismos para modelagem de estruturas flexíveis (Boyer *et al.*, 1998). Podem ser citados por exemplo, o formalismo modal analítico, o discreto e o que assume formas polinomiais para as deformações elásticas. Em (Pereira, 1999) existe um estudo comparativo entre diversos formalismos, sendo uma das conclusões o fato do formalismo discreto ser uma abordagem simples e bem representativa. Neste trabalho considerou-se portanto o formalismo discreto, que recebe o nome em idioma inglês de *lumped mass approach*. A flexibilidade contínua é aproximada por uma discreta a partir de duas articulações fictícias, conforme indicado na Fig. 1. Neste caso, os primeiros dois modos flexíveis estarão presentes. Em (Gomes & Chrétien, 1992) existem mais detalhes sobre este tipo de modelagem estrutural.

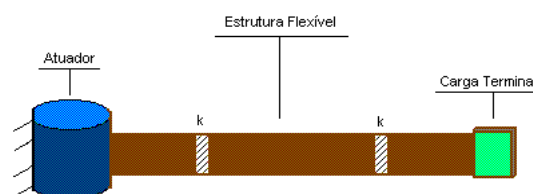


Figura 1. Estrutura flexível e sua aproximação discreta.

Sendo L o comprimento do braço, a primeira articulação fictícia encontra-se a $L/4$ do atuador e a segunda a $L/2$ da primeira, conforme a Fig. 1. A constante k é obtida a partir da expressão: $k = nEI/L$, onde n é o número de articulações fictícias, enquanto que EI é o módulo

de elasticidade vezes o momento de inércia da seção reta. O lagrangeano do sistema é determinado e com ele obtém-se as equações da dinâmica a partir do formalismo de Euler-Lagrange. Considerando-se pequenas deformações angulares nas articulações fictícias, obtém-se então o modelo linearizado, o qual escrito na forma de estado é representado pela seguinte equação:

$$\dot{x} = Ax + BTm \quad (1)$$

O vetor de estado é composto pelas três posições angulares referentes ao rotor e ângulos na primeira e segunda articulação fictícia, bem como das respectivas velocidades angulares. Portanto, são seis variáveis de estado. Tm é o torque aplicado no rotor. A única observação considerada é a posição angular do rotor:

$$y = Cx; \quad C = [1 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (2)$$

2.2 Modelo do Fluido

Considerou-se um modelo bastante simplificado para o fluido uma vez que o objetivo não era o de se conseguir extrema precisão para a interação fluido estrutura e sim, o de averiguar se o sistema se manteria estável em presença de uma perturbação externa não prevista no projeto do controlador. Admitindo que a estrutura forma um ângulo de ataque com a direção do fluxo, o fluido provoca uma força que atua perpendicularmente à mesma. Esta força pode ser decomposta em duas componentes, F_x e F_y . A primeira, F_x , age na direção do escoamento (força de arrasto) e a segunda, F_y , age perpendicularmente a esta direção (força de sustentação). Considerando-se um ângulo de ataque pequeno F_x é, então, aproximadamente nula. Assim, segundo o modelo considerado, (Hughes & Brighton, 1974) e (Schlichting, 1960), as forças exercidas pelo fluido na estrutura possuem a seguinte forma:

$$\begin{aligned} F_y &= \frac{1}{2} \rho v^2 A_p C_l \\ F_x &= 0.05 F_y \approx 0 \end{aligned} \quad (3)$$

$C_l = 2\pi \sin(\alpha)$, onde α é o ângulo de ataque e C_l o coeficiente de sustentação. ρ é a massa específica do fluido, v a velocidade de escoamento e A_p a área projetada (corda por comprimento). A força F_y causa torques nas articulações gerando um vetor de perturbações p , ficando então a dinâmica descrita sob a forma:

$$\dot{x} = Ax + BTm + Dp \quad (4)$$

A Figura 2 mostra uma simulação em malha aberta. O torque motor, durante os primeiros 0,05 s foi de 8 Nm, nos próximos 0,05 s de -8 Nm, sendo zero no restante do tempo. A velocidade do fluido e o ângulo de ataque foram 10 m/s e 5°, respectivamente, valores mantidos para todas as simulações realizadas no trabalho. Observa-se que, devido a presença do fluido, as velocidades não se estabilizam no zero e a estrutura toda se desloca uma vez que a mesma é articulada em uma extremidade e livre na outra.

Um outro modelo para o cálculo da perturbação externa devido ao fluido também foi desenvolvido, baseado nas equações do escoamento potencial e os torques obtidos são ligeiramente diferentes daqueles obtidos pelo modelo aqui apresentado, não provocando diferenças significativas no comportamento das vibrações da estrutura.

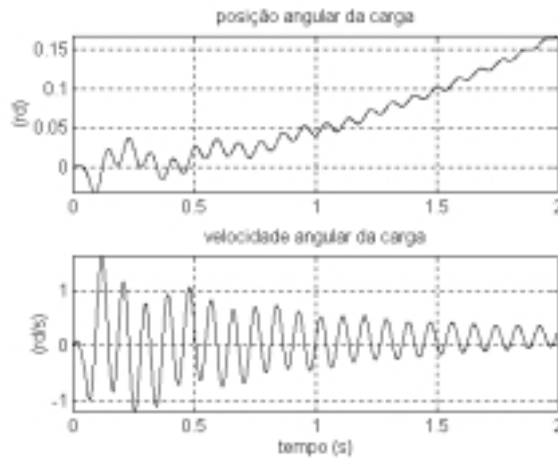


Figura 2. Simulação em malha aberta (velocidade do fluido = 10 m/s).

3. PROJETO DE CONTROLE

Inicialmente projetou-se um controlador LQG/LTR, adotando-se como critério uma largura de faixa do controlador aproximadamente duas vezes o primeiro modo de vibração do sistema, conforme mostrado na Fig. 3.

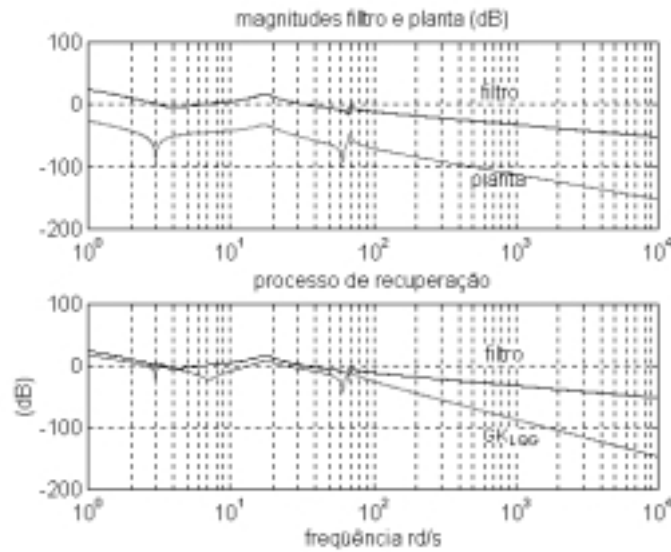


Figura 3. Respostas em frequência relativas ao projeto LQG/LTR (processo de recuperação com $\rho=10^{-6}$).

Um segundo controlador foi projetado por meio de alocação de pólos do regulador e do filtro de forma que a malha direta obedeça às especificações, na forma de barreiras de robustez do desempenho e da estabilidade. Esta técnica de alocação de pólos encontra-se descrita em maiores detalhes em (Franklin & Powell, 1995). Resumidamente, o controle é dado pela expressão:

$$Tm = -K_r \hat{x} \tag{5}$$

onde \hat{x} corresponde ao estado estimado a partir da equação:

$$\dot{\hat{x}} = A\hat{x} + BTm + K_f(y - C\hat{x}). \quad (6)$$

Assumindo-se que s_1, s_2, \dots, s_6 são os pólos do regulador (raízes de $[A - BK_r]$) e $\alpha_c(s) = (s - s_1)(s - s_2) \dots (s - s_6)$ o respectivo polinômio característico, o vetor de ganhos do regulador pode ser obtido a partir da expressão:

$$K_r = [0 \ 0 \ \dots \ 0 \ 1] M_c^{-1} \alpha_c(A) \quad (7)$$

onde $M_c = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$ corresponde à matriz de controlabilidade (M_c deve ser inversível). Processo semelhante pode ser utilizado para a obtenção dos ganhos do estimador. Assumindo-se que $\alpha_e(s) = (s - \beta_1)(s - \beta_2) \dots (s - \beta_6)$ é polinômio característico de $[A - K_f C]$, os ganhos do estimador são obtidos a partir da seguinte expressão:

$$K_f = \alpha_e(A) \vartheta^{-1} [0 \ 0 \ \dots \ 1]^T \quad (8)$$

onde $\vartheta = [C \ CA \ CA^2 \ \dots \ CA^{n-1}]$ corresponde à matriz de observabilidade (também deve ser inversível).

Um projeto foi então desenvolvido por meio da técnica de alocação de pólos, objetivando boa robustez do compensador como um todo. A Fig. 4 mostra a resposta em freqüência, na qual pode-se constatar que o decaimento $|GK|$ começou em aproximadamente 25 rd/s , acentuando-se definitivamente em aproximadamente 100 rd/s permitindo que as barreiras de robustez da estabilidade e do desempenho sejam obedecidas.

A Tabela 1 mostra os pólos da planta, do filtro e do regulador. Pode-se observar que houve no projeto uma preocupação específica com a baixa freqüência, fato que propicia que se obtenha respostas como a mostrada na Fig. 4.

Tabela 1. Pólos da planta $[\lambda_i(A)]$, do regulador $[\lambda_i(A - BK_r)]$ e do filtro $[\lambda_i(A - K_f C)]$.

Sistema	Pólos
Planta	$-1,816 \pm 69,5678i$; $-2,3636 \pm 17,4717i$; $0,0000$; $-0,1426$
Regulador	$-5,1054 \pm 97,0057i$; $-3,0167 \pm 23,9909i$; $-2,0005 \pm 0,3491i$
Filtro	$-3,7390 \pm 69,5655i$; $-9,4267 \pm 17,5048i$; $-3,9852 \pm 0,4192i$

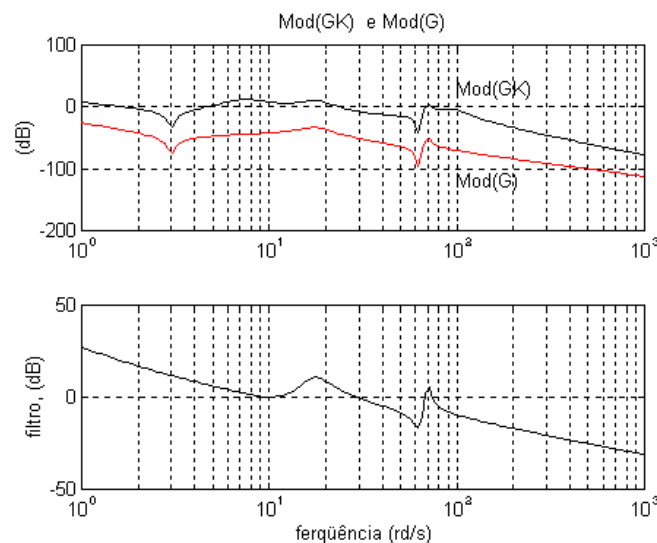


Figura 4. Projeto por alocação de pólos com características de robustez.

4. SIMULAÇÕES E ANÁLISE DE RESULTADOS

As simulações mostradas a seguir foram feitas com a mesma ativação em malha aberta seguida da aplicação do controle vista na Fig. 2. A única diferença agora é que a partir desta ativação aplicou-se a lei de controle obtida por meio dos projetos apresentados anteriormente.

As Figs. 5, (a)-(d), referem-se a resultados de simulações obtidos por meio do projeto LQG/LTR, considerando a perturbação externa devida ao fluido, com exceção do resultado mostrado na Fig. 5 (a). A Fig. 5 (a), mostra resultados de uma simulação da planta nominal, indicando que o controlador mostrou-se robusto à perturbação externa causada pelo fluido, já que a estabilidade foi mantida. A simulação da Fig. 5 (b) contém uma variação de -5% no parâmetro módulo de elasticidade estrutural e evidencia um significativo acréscimo na amplitude das oscilações, indicadoras de perda de desempenho.

Na Fig. 5 (c), que corresponde a uma variação de -20% na inércia do cubo, pode-se observar que a estabilidade e o desempenho foram mantidos. A Fig. 5 (d) é o resultado de uma simulação considerando um ruído de medida na posição angular do rotor, sem levar em conta a perturbação devida ao fluido. Pode-se perceber que o sistema se mantém estável aparecendo oscilações de pequena amplitude na velocidade da carga, oriundas das oscilações no torque de controle em razão da presença do ruído de medida.

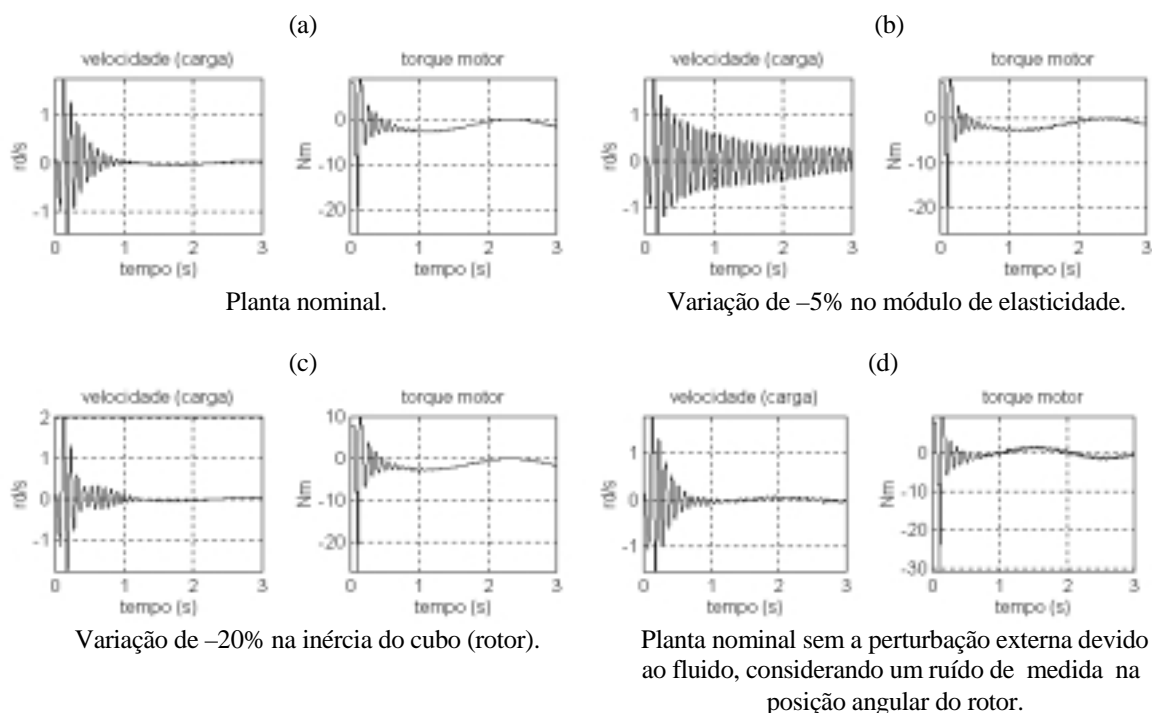
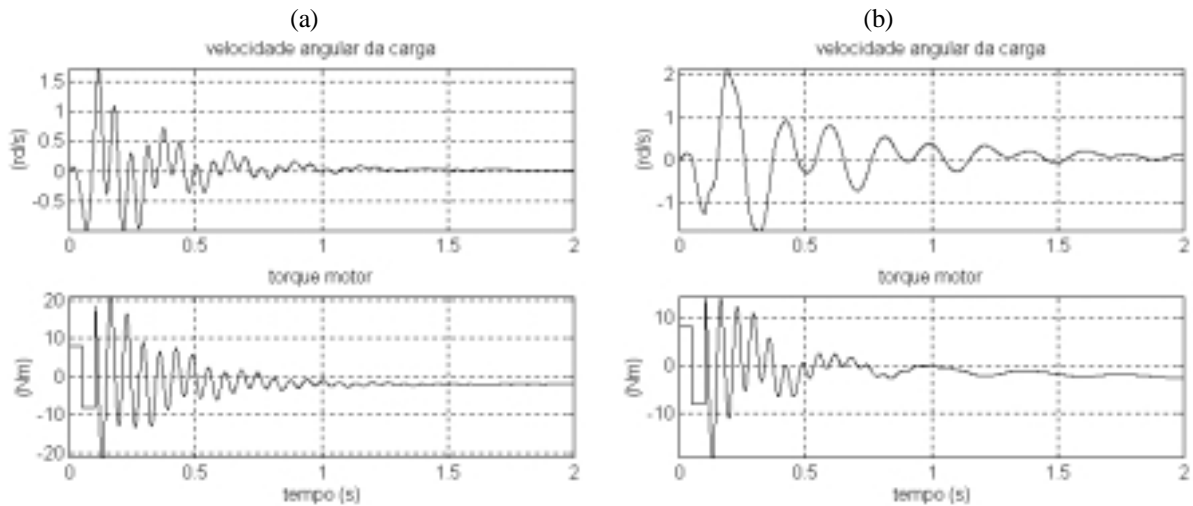


Figura 5. Simulações com o projeto LQG/LTR.

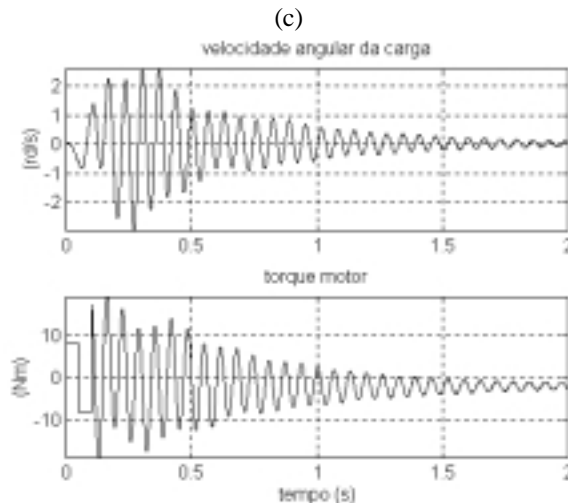
As Figs. 6, (a)-(c), referem-se a resultados de simulações obtidos por meio do projeto por alocação de pólos. A Fig. 6 (a), apresenta resultados considerando-se a planta real igual à nominal. Pode-se perceber que as vibrações da carga são atenuadas em aproximadamente 1 segundo e o torque de controle se estabiliza em um nível constante abaixo de zero para compensar os torques oriundos do contato do fluido com a estrutura.

As Figs. 6 (b) e 6 (c) são resultados de simulações com variações de -80% no módulo de elasticidade e -40% na massa específica da estrutura, respectivamente. Pode-se notar que em ambos os casos o sistema se mantém estável e os torques tendem a se estabilizar a um nível constante.



Planta nominal.

Varição de -80% no módulo de elasticidade.



Varição de -40% na massa específica da estrutura.

Figura 6. Simulações com o projeto por alocação de pólos.

5. CONCLUSÕES

Investigou-se no presente trabalho, diferentes estratégias de controle para atenuar vibrações em uma estrutura flexível ativada continuamente pelo escoamento de um fluido. Impôs-se ao problema a restrição importante de que apenas a posição angular do rotor era observada. Pode-se extrair as seguintes observações como principais conclusões:

- o projeto LQG/LTR apresentou bons resultados quanto à robustez na rejeição da perturbação externa e na rejeição de ruídos de medida considerando a planta nominal igual à real, porém não se conseguiu bons resultados quando foram consideradas variações em alguns dos parâmetros, conclusão evidentemente válida apenas para esta aplicação;
- utilizando-se alocação de pólos para o regulador e para o filtro e ainda, respeitando-se barreiras de robustez ao desempenho e à estabilidade, é possível a obtenção de bons resultados em presença de significativas variações paramétricas.

Indicações de trabalhos futuros seriam a investigação de outras técnicas de controle robusto aplicadas ao mesmo problema e ainda, a investigação sobre o efeito de não linearidades dos atuadores como possíveis diminuidoras de desempenho. Além disso, uma análise mais detalhada

da perturbação causada pelo fluido pode ser obtida via solução das equações de Navier-Stokes.

REFERÊNCIAS

- Boyer, F., Glandais, N. and Khalil, W., 1998, “Consistent first and second Order Dynamic Model of Flexible Manipulators”, IEEE Robotics and Automation Conference, May, Leuven, Belgium.
- Cannon, R. H. Jr. and Schmitz, E., 1984, “Initial Experiments on the End-Point Control of a Flexible One-Link Robot”, The International of Robotics Research, Stanford, California.
- Chrétien, J. P. et Alazard, D., 1991, “Modélisation des chaînes polyarticulées flexibles: comparaison de trois techniques de mise en équation”. Rapport D.R.E.T./D.E.R.A. n.-2/7735, CERT, Toulouse.
- Cruz, J. J., 1996, “Controle Linear Robusto Multivariável”, EDUSP.
- Franklin, G. F., Powell, J. D., 1995, “Feedback Control of Dynamic Systems”, Addison-Wesley Publishing Company.
- Furusho, J., Zhang, G. and Sakaguchi, M., 1997, “Vibration Suppression Control of Robot Arms Using a Homogeneous-Type Electrorheological Fluid”, IEEE Robotics and Automation Conference, April, Albuquerque, New México.
- Gomes, S. C. P. and Chrétien, J. P., 1992, “Dynamic modeling and friction compensated control of a 1 d.o.f. rigid or flexible manipulator”. In proceedings of Microdynamic et Pointage de Grande Précision, CNES, 1er Symposium International, November, Nice, France.
- Hughes, W. F., Brighton, J., 1974, “Dinâmica de Fluidos”, McGraw-Hill do Brasil, Coleção Shaum.
- Kim, J-S, Suzuki, K., Yamano, M. and Uchiyama, M., 1997, “Vibration Suppression Control of Constrained Spatial Flexible Manipulators”, IEEE Robotics and Automation Conference, April, Albuquerque, New México.
- Kojima, H., 1990, “Dynamic Finite Element Analysis of the Position Control System of a Two-Link Horizontal Flexible Robot”, Journal of Robotics & Mechatronics, Tenjin-cho, Japan.
- Pereira, A. E. L., 1999, “Um Estudo Sobre Modelagem Matemática de Estruturas Flexíveis”, Dissertação de Mestrado, UFRGS, Porto Alegre, Brasil.
- Schlichting, H., 1960, “Boundary Layer Theory”, McGraw-Hill, New York.
- Schmitz, E., 1985, “Experiments on the End-Point Control of a very Flexible One-link Manipulator”, Ph.D. Thesis, Stanford, California, Unite States.
- Soares, A. M. S., 1997, “Modelagem e identificação experimental de estruturas flexíveis”. Tese de Doutorado, ITA, São José dos Campos, Brasil.
- Vincent, S., 1996, “Complémentarité d’actionneurs pour la commande active des structures flexibles. Study of actuators harmony for active control of flexible structures”.Thèse de Doctorat ENSAE, Spécialité Automatique, CERT, Toulouse.
- Xie, H-P, Kalaycioglu, S. and Patel R. V., 1997, “Control Shuttle Remote Manipulator Systems”, IEEE Robotics and Automation Conference, April, Albuquerque, New México.
- Watanabe, K., Suzuki, H., Tao, Q. and Yoshida, K., 1998, “Basic Research on Underwater Docking of Flexible Structures”, IEEE Robotics and Automation Conference, May, Leuven, Belgium.

UM ESTUDO SOBRE MODELAGEM MATEMÁTICA DE ESTRUTURAS FLEXÍVEIS (PARTE II)

Adriana Elisa Ladeira Pereira
Sebastião Cícero Pinheiro Gomes
Álvaro Luiz De Bortoli

Fundação Universidade Federal do Rio Grande, Dep. de Matemática, CEP: 96201-900, Rio Grande, RS, Brasil, E-mail: mmaaelp@super.furg.br, dmtscpg@super.furg.br
Universidade Federal do Rio Grande do Sul, PPGMAp, CEP: 91509-900, Porto Alegre, RS, Brasil, E-mail: dbortoli@mat.ufrgs.br

Resumo

Este trabalho é uma continuação da Parte I, constituindo-se num estudo sobre técnicas de modelagem de estruturas do tipo manipulador mecânico com elo flexível. A motivação para a sua realização é o conhecimento de um modelo dinâmico preditivo e realista que pode ser de grande auxílio ao projeto de controladores com condições reais de implementação prática. Este artigo contém a segunda parte do estudo sobre diversas técnicas de modelagem dinâmica de uma estrutura do tipo manipulador flexível. Duas técnicas de modelagem dinâmica utilizando o formalismo dos Modos Assumidos são desenvolvidas e analisadas, tendo-se utilizado funções de formas polinomiais e oriundas dos modos exatos da estrutura. Comparações de respostas em frequência entre as funções de transferência analíticas (Parte I) e as funções de transferência obtidas com as duas técnicas citadas são realizadas, extraíndo-se ao final conclusões sobre estes formalismos de modelagem.

Palavras Chave: Robótica, flexível, modelagem, dinâmica, controle.

1. INTRODUÇÃO

No presente artigo são desenvolvidas e analisadas mais duas técnicas de modelagem dinâmica para o manipulador flexível considerado na Parte I. Trata-se do Método dos Modos Assumidos com duas diferentes funções de forma: analíticas (modos exatos do problema) e polinomiais, confundindo-se neste último caso com o Método dos Elementos Finitos. No Método dos Modos Assumidos com funções de forma exatas, inicialmente toda a teoria é desenvolvida detalhadamente até a determinação das formas próprias para cada modo de vibração. O conhecimento das frequências naturais e das formas de cada modo possibilita a determinação das equações da dinâmica do sistema. As funções de transferência, nos casos colocado e não colocado, são obtidas e suas respostas frequenciais são, então, comparadas com as respostas obtidas das funções de transferência analíticas (Parte I). Ao utilizar-se funções de forma polinomiais faz-se, inicialmente, o desenvolvimento teórico. Posteriormente são realizadas comparações também com o estudo analítico apresentado anteriormente.

2. O MÉTODO DOS MODOS ASSUMIDOS COM FUNÇÕES DE FORMAS EXATAS

Considere-se as equações diferenciais obtidas após a transformação de variáveis sugerida por Brakwel (Schmitz, 1985).

$$I_r \ddot{\theta}(t) + \rho \int_0^l x \frac{\partial^2 y}{\partial t^2} dx + m_c l \ddot{y}(l, t) = T_m \quad (1)$$

$$EI \frac{\partial^4 y}{\partial x^4}(x, t) + \rho \frac{\partial^2 y}{\partial t^2}(x, t) = 0 \quad (2)$$

com as respectivas condições de contorno:

$$\begin{aligned} EI \frac{\partial^2 y}{\partial x^2}(x, t) \Big|_{x=0} + T_m - I_r \ddot{\theta} &= 0 \\ y(x, t) \Big|_{x=0} = y(0, t) = w(0, t) + 0\theta(t) &= 0 \\ EI \frac{\partial^2 y}{\partial x^2}(x, t) \Big|_{x=l} = EI \frac{\partial^2 w}{\partial x^2}(l, t) &= 0 \\ EI \frac{\partial^3 y}{\partial x^3}(x, t) \Big|_{x=l} = m_c \frac{\partial^2 y}{\partial t^2}(x, t) \Big|_{x=l}. \end{aligned} \quad (3)$$

Uma forma de resolver as equações (1) e (2) é a partir da expansão da função $y(x, t)$ como uma série finita na forma (Meirovitch, 1967):

$$y(x, t) = \sum_{i=0}^n \phi_i(x) q_i(t), \quad (4)$$

sendo

$$\theta(t) = \sum_{i=0}^n \theta_i q_i(t); \quad \theta_i = \frac{d\phi_i}{dx} \Big|_{x=0} \quad (5)$$

Aqui $\phi_i(x)$ são as funções de forma exatas, $q_i(t)$ são as coordenadas generalizadas dependentes do tempo, $\theta_i(t)$ são as amplitudes modais do movimento rotacional da estrutura e n é um número finito igual ao número dos modos de vibração que se deseja incluir no modelo.

As funções $\phi_i(x)$ utilizadas nesta parte do trabalho devem satisfazer às condições de contorno e à equação diferencial associada ao problema. Com a equação (4), a equação (2) pode ser escrita na forma:

$$\frac{l^4}{\phi_i(x)} \frac{d^4 \phi_i}{dx^4} = - \frac{\rho l^4}{EI} \frac{1}{q_i(t)} \frac{d^2 q_i}{dt^2} = \lambda_i^4. \quad (6)$$

Dessa forma, separa-se o problema original em dois problemas desacoplados, um somente temporal e o outro espacial.

O Problema temporal pode ser escrito como:

$$\frac{d^2 q_i}{dt^2} + w_i^2 q_i(t) = 0, \quad (7)$$

onde $w_i^2 = \frac{EI}{\rho l^4} \lambda_i^4$, sendo que este problema admite solução na forma:

$$q_i(t) = C e^{j w_i t}, \quad (8)$$

sendo C uma constante.

O problema espacial pode ser escrito como:

$$\frac{d^4 \phi_i}{dx^4} - \beta_i^4 \phi_i(x) = 0, \quad (9)$$

onde $\beta_i = \frac{\lambda_i}{l}$, sendo que este problema admite solução na forma:

$$\phi_i(x) = C_{1i} \text{sen}(\beta_i x) + C_{2i} \text{cos}(\beta_i x) + C_{3i} \text{senh}(\beta_i x) + C_{4i} \text{cosh}(\beta_i x), \quad (10)$$

sendo as constantes C_{1i} , C_{2i} , C_{3i} e C_{4i} determinadas pelas condições de contorno (3), expressas agora na variável $\phi_i(x)$.

Das condições de contorno, obtém-se:

$$\begin{cases} EI\beta_i(C_{4i} - C_{2i}) + I_r w_i^2(C_{1i} + C_{3i}) = 0 \\ C_{2i} + C_{4i} = 0 \\ -C_{1i} \sin(\beta_i l) - C_{2i} \cos(\beta_i l) + C_{3i} \sinh(\beta_i l) + C_{4i} \cosh(\beta_i l) = 0 \\ C_{1i}(-EI\beta_i^3 \cos(\beta_i l) + m_c w_i^2 \sin(\beta_i l)) + C_{2i}(EI\beta_i^3 \sin(\beta_i l) + m_c w_i^2 \cos(\beta_i l)) + \\ + C_{3i}(EI\beta_i^3 \cosh(\beta_i l) + m_c w_i^2 \sinh(\beta_i l)) + C_{4i}(EI\beta_i^3 \sinh(\beta_i l) + m_c w_i^2 \cosh(\beta_i l)) = 0. \end{cases} \quad (11)$$

Resolvendo-se este sistema, encontram-se as constantes C_{1i} , C_{2i} e C_{3i} em função da constante C_{4i} , resultando na equação:

$$\begin{aligned} & \left[\left(\frac{2EI\beta_i \sin(\beta_i l) + I_r w_i^2 (\cos(\beta_i l) + \cosh(\beta_i l))}{I_r w_i^2 (\sin(\beta_i l) + \sinh(\beta_i l))} - \frac{2EI\beta_i}{I_r w_i^2} \right) (-EI\beta_i^3 \cos(\beta_i l) + m_c w_i^2 \sin(\beta_i l)) + \right. \\ & \left. - \left(\frac{2EI\beta_i \sin(\beta_i l) + I_r w_i^2 (\cos(\beta_i l) + \cosh(\beta_i l))}{I_r w_i^2 (\sin(\beta_i l) + \sinh(\beta_i l))} \right) (EI\beta_i^3 \cosh(\beta_i l) + m_c w_i^2 \sinh(\beta_i l)) + \right. \\ & \left. + EI\beta_i^3 (\sinh(\beta_i l) - \sin(\beta_i l)) + m_c w_i^2 (\cosh(\beta_i l) - \cos(\beta_i l)) \right] C_{4i} = \Phi(\beta_i) C_{4i} = 0. \end{aligned} \quad (12)$$

Portanto, para que C_{4i} seja diferente de zero (solução não trivial) é necessário que os termos entre colchetes sejam nulos, ou seja:

$$\Phi(\beta_i) = 0, \quad (13)$$

sendo esta equação usada para determinar os valores dos β_i e, a partir destes valores, determinam-se os λ_i , pois $\lambda_i = \beta_i l$. As primeiras três raízes não nulas desta equação foram encontradas utilizando-se o Método Numérico da Bissecção, com oito dígitos significativos de precisão:

$$\beta_1 = 1.522638; \beta_2 = 2.615733; \beta_3 = 4.432305,$$

o que implica em

$$\lambda_1 = 2.45144718; \lambda_2 = 4.21133013; \lambda_3 = 7.13601105.$$

As demais constantes presentes na equação (10) são relativas aos parâmetros estruturais, tendo-se utilizado os mesmos valores numéricos da Parte I. Como

$$w_i = \pm \sqrt{\frac{EI}{\rho l^4}} \lambda_i^2, \quad (14)$$

as frequências dos três primeiros modos de vibração são:

$$w_1 = 17.50942188 \text{ rd/s}; w_2 = 51.67329338 \text{ rd/s}; w_3 = 148.36715 \text{ rd/s}.$$

Pode-se escrever então:

$$\phi_i(x) = C_{4i} [M \sin(\beta_i x) - \cos(\beta_i x) + N \sinh(\beta_i x) + \cosh(\beta_i x)] \quad (15)$$

sendo:

$$\begin{aligned} M &= \frac{(2EI\beta_i \sin(\beta_i l) + I_r w_i^2 (\cos(\beta_i l) + \cosh(\beta_i l)))}{I_r w_i^2 (\sin(\beta_i l) + \sinh(\beta_i l))} - \frac{2EI\beta_i}{I_r w_i^2} \\ N &= -\frac{(2EI\beta_i \sin(\beta_i l) + I_r w_i^2 (\cos(\beta_i l) + \cosh(\beta_i l)))}{I_r w_i^2 (\sin(\beta_i l) + \sinh(\beta_i l))}. \end{aligned}$$

Porém, $\phi_i(x)$ deve satisfazer à seguinte condição de ortogonalidade (Pereira, 1999):

$$\int_0^l [\phi_i(x)]^2 \rho dx + I_r \left[\frac{d\phi_i}{dx}(0) \right]^2 + m_c [\phi_i(l)]^2 = I_T, \quad (16)$$

onde :

$$I_T = I_r + I_B + m_c l^2 \text{ e } I_B = \frac{\rho l^3}{3},$$

sendo I_T correspondente à inércia total de corpo rígido da estrutura. Portanto, aplicando-se a condição (16), chega-se à equação:

$$\begin{aligned}
C_{4i}^2 \left\{ \rho \left[M^2 \left(\frac{l}{2} - \frac{1}{4\beta_i} \operatorname{sen}(2\beta_i l) \right) - \frac{M}{2\beta_i} (1 - \cos(2\beta_i l)) + \frac{l}{2} + \frac{1}{4\beta_i} \operatorname{sen}(2\beta_i l) + \right. \right. \\
+ \frac{MN}{\beta_i} (\cosh(\beta_i l) \operatorname{sen}(\beta_i l) - \operatorname{senh}(\beta_i l) \cos(\beta_i l)) + \frac{M}{\beta_i} (1 + \operatorname{senh}(\beta_i l) \operatorname{sen}(\beta_i l) - \cosh(\beta_i l) \cos(\beta_i l)) + \\
- \frac{N}{\beta_i} (\operatorname{senh}(\beta_i l) \operatorname{sen}(\beta_i l) + \cos(\beta_i l) \cosh(\beta_i l) - 1) - \frac{1}{\beta_i} (\operatorname{sen}(\beta_i l) \cosh(\beta_i l) + \operatorname{senh}(\beta_i l) \cos(\beta_i l)) + \\
+ N^2 \left(-\frac{l}{2} + \frac{1}{4\beta_i} \operatorname{senh}(2\beta_i l) \right) + \frac{N}{2\beta_i} (\cosh(2\beta_i l) - 1) + \frac{l}{2} + \frac{1}{4\beta_i} \operatorname{senh}(2\beta_i l) \left. \right] + I_T \beta_i^2 (M + N)^2 + \\
+ m_c (M \operatorname{sen}(\beta_i l) - \cos(\beta_i l) + N \operatorname{senh}(\beta_i l) + \cosh(\beta_i l))^2 \left. \right\} = C_{4i}^2 K_i(\beta_i) = I_T
\end{aligned} \quad (17)$$

Portanto, a partir da equação acima, pode-se determinar analiticamente o valor da constante C_{4i} na forma:

$$C_{4i}^2 K_i(\beta_i) = I_T \Rightarrow C_{4i} = \pm \sqrt{\frac{I_T}{K_i(\beta_i)}}, \quad (18)$$

ou seja, para cada β_i tem-se um K_i e uma conseqüente constante C_{4i} .

A partir destas constantes e dos valores dos $\beta_i, i = 1, 2, 3$, são determinadas as funções de forma, cujos gráficos são mostrados na Figura 1.

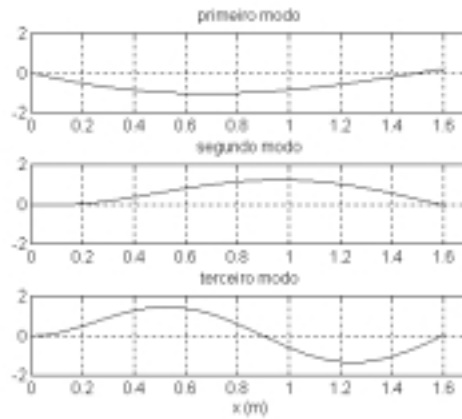


Figura 1 Gráficos das funções de forma

2.1 Obtenção do modelo dinâmico

O Lagrangeano do sistema pode ser escrito sob a forma (Pereira, 1999):

$$L = \frac{1}{2} \sum_{i=0}^3 I_T \dot{q}_i^2 - \frac{1}{2} \sum_{i=0}^3 I_T w_i^2 q_i^2 + T_m \left(\sum_{i=0}^3 \frac{d\phi_i(0)}{dx} q_i \right), \quad (19)$$

aplicando-se então as equações de Euler-Lagrange, chega-se ao sistema:

$$\dot{\vec{X}} = A\vec{X} + BT_m, \quad (20)$$

onde

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -w_1^2 & 0 & 0 & 0 & -2\xi_1 w_1 & 0 & 0 \\ 0 & 0 & -w_2^2 & 0 & 0 & 0 & -2\xi_2 w_2 & 0 \\ 0 & 0 & 0 & -w_3^2 & 0 & 0 & 0 & -2\xi_3 w_3 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & \frac{d\phi_1}{dx}(0) & \frac{d\phi_2}{dx}(0) & \frac{d\phi_3}{dx}(0) \end{bmatrix}^T,$$

$$X = [q_0 \quad q_1 \quad q_2 \quad q_3 \quad \dot{q}_0 \quad \dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T$$

e $\xi_i, i=1,2,3$, correspondem às razões de amortecimento, as quais serão diferentes de zero caso se considere o amortecimento natural da estrutura (estes valores podem ser identificados a partir de experimentos em malha aberta).

A matriz de observação, para o caso colocado, possui a seguinte forma (observa-se a velocidade do rotor):

$$C_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & \frac{d\phi_1}{dx}(0) & \frac{d\phi_2}{dx}(0) & \frac{d\phi_3}{dx}(0) \end{bmatrix},$$

Para o caso não colocado, observa-se a posição $y(l,t)$ da carga terminal e portanto, a matriz de observação será:

$$C_{nc} = [l \quad \phi_1(l) \quad \phi_2(l) \quad \phi_3(l) \quad 0 \quad 0 \quad 0 \quad 0]$$

A Figura 2 contém as respostas em frequência das funções de transferência analítica (Parte I) e por Modos Assumidos com Funções de Forma Exatas, para o caso colocado. A fim de garantir uma comparação mais efetiva com o resultado analítico, o amortecimento estrutural foi negligenciado, assumindo-se as razões de amortecimento ($\xi_i, i=1,2,3$) nulas. Analisando-se esta figura, observa-se a perfeita coincidência com o resultado analítico, ou seja, os três primeiros zeros e pólos são idênticos nos dois resultados. Deve-se ressaltar que se utilizou, como funções de forma, os modos exatos do problema e apenas neste caso, o método dos modos assumidos se confunde com a solução exata apresentada na Parte I.

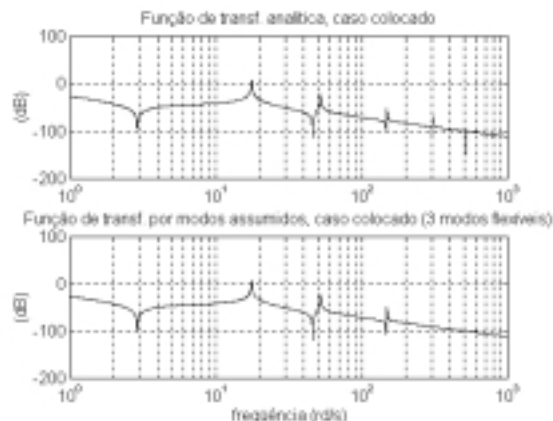


Figura 2 Funções de Transferência Analítica e por Modos Assumidos (caso colocado).

3. O MÉTODO DOS MODOS ASSUMIDOS (FUNÇÕES DE FORMA POLINOMIAIS)

Neste caso, as funções de forma são polinômios, confundindo-se assim com o método dos elementos finitos. No caso clássico, o elemento geralmente é de terceira ordem. Esta técnica é inconveniente por necessitar de um número grande de elementos, a fim de minimizar o efeito de uma modelagem incorreta das deformações correspondentes ao primeiro elemento finito (conectado ao rotor) e ao último (conectado à carga), ampliando assim a complexidade do modelo (Pereira, 1999).

3.1 Utilização de uma aproximação polinomial do quinto grau

Para superar as dificuldades do método clássico citado anteriormente, Chrétien (Chrétien, 1990) propôs a utilização de um único elemento, mas com uma deformação aproximada por um polinômio do quinto grau. Considere-se uma estrutura elementar, conforme mostrada na Figura 3. Este elemento possui o comprimento total da estrutura flexível, pois se considera um único elemento finito. Dois parâmetros cinemáticos são introduzidos:

$$q_3(t) = \frac{M_{fA}}{EI}; q_6(t) = \frac{M_{fB}}{EI},$$

onde M_{fA} e M_{fB} são os momentos fletores nas extremidades da estrutura elementar.

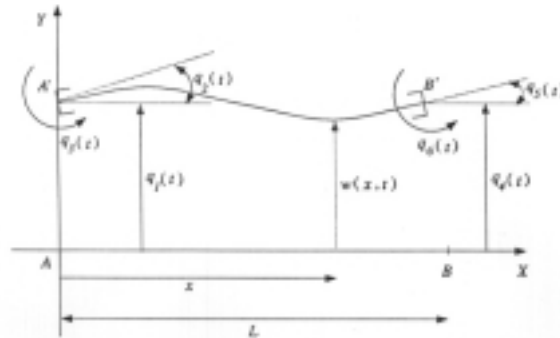


Figura 3 Estrutura elementar com a representação de seis parâmetros cinemáticos.

A expressão para a deformação da estrutura possui a forma:

$$w(x,t) = a_0(t) + a_1(t)x + a_2(t)x^2 + a_3(t)x^3 + a_4(t)x^4 + a_5(t)x^5, \quad (21)$$

com as seguintes condições de contorno:

$$a) w(0,t) = q_1(t); b) \frac{\partial w}{\partial x}(0,t) = q_2(t);$$

$$c) \frac{\partial^2 w}{\partial x^2}(0,t) = q_3(t); d) w(l,t) = q_4(t);$$

$$e) \frac{\partial w}{\partial x}(l,t) = q_5(t); f) \frac{\partial^2 w}{\partial x^2}(l,t) = q_6(t).$$

Aplicando-se as condições de contorno, obtém-se a deformação na forma:

$$w(x,t) = \sum_{i=1}^6 \phi_i(x)q_i(t), \quad (22)$$

onde

$$\phi_1(x) = 1 - \frac{10x^3}{l^3} + \frac{15x^4}{l^4} - \frac{6x^5}{l^5}; \phi_2(x) = x - \frac{6x^3}{l^2} + \frac{8x^4}{l^3} - \frac{3x^5}{l^4}; \phi_3(x) = \frac{x^2}{2} - \frac{3x^3}{2l} + \frac{3x^4}{2l^2} - \frac{x^5}{2l^3};$$

$$\phi_4(x) = \frac{10x^3}{l^3} - \frac{15x^4}{l^4} + \frac{6x^5}{l^5}; \phi_5(x) = -\frac{4x^3}{l^2} + \frac{7x^4}{l^3} - \frac{3x^5}{l^4}; \phi_6(x) = \frac{x^3}{2l} - \frac{x^4}{l^2} + \frac{x^5}{2l^3}.$$

As expressões para calcular as energias cinética e potencial são as mesmas do caso analítico (Parte I), mas considerando que $\theta = q_2$. A aplicação das equações de Euler-Lagrange possibilita a obtenção do modelo dinâmico na forma:

$$[m][\dot{q}] + [k][q] = [S], \quad (23)$$

onde os elementos da matriz $[m]$ são dados por (Pereira, 1999):

$$m_{ij} = \int_0^l \rho \phi_i(x) \phi_j(x) dx + m_c \phi_i(l) \phi_j(l), (i, j) \neq 2;$$

$$m_{i2} = m_{2i} = \int_0^l \rho [\phi_2(x) + x] \phi_i(x) dx + m_c [\phi_2(l) + l] \phi_i(l), i \neq 2;$$

$$m_{22} = \int_0^l \rho \{[\phi_2(x) + 2x] \phi_2(x) + x^2\} dx + m_c \{[\phi_2(l) + 2l] \phi_2(l) + l^2\} + I_T.$$

os elementos da matriz $[k]$ são:

$$k_{ij} = \int_0^l EI \frac{d^2 \phi_i}{dx^2} \frac{d^2 \phi_j}{dx^2} dx, i = 1, \dots, 6 \quad j = 1, \dots, 6.$$

e $[S]$ é o vetor com os esforços externos.

Na Figura 4 mostra-se os três primeiros modos de vibração no gráfico da função de transferência do Método dos Modos Assumidos com funções de forma polinomiais. Verifica-se ainda que os zeros são muito próximos dos três primeiros zeros encontrados na função de transferência analítica. Percebe-se também que os dois primeiros pólos encontram-se deslocados para a direita com relação aos dois primeiros pólos analíticos e que o terceiro pólo coincide aproximadamente com o terceiro pólo analítico.

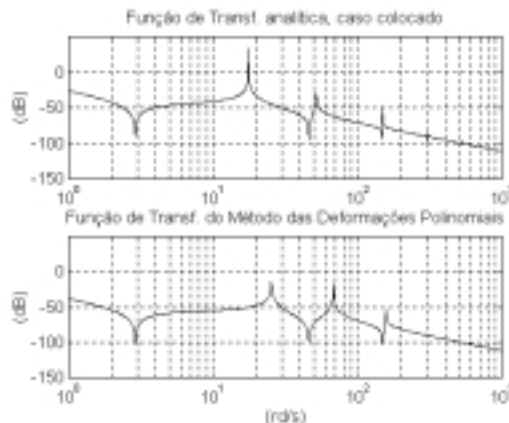


Figura 4 Funções de Transferência Analítica e do Método dos Modos Assumidos com Funções de Forma Polinomiais (caso colocado).

4. CONCLUSÕES

Sobre o Método dos Modos Assumidos (modos exatos)

- Trata-se da técnica que apresentou o melhor desempenho pois, conforme já esperado, não há erros com relação ao resultado exato (funções de transferência analíticas);
- Encontrar as funções de forma, neste caso dos modos exatos, não é um processo exatamente simples, principalmente porque o mesmo não depende somente das condições geométricas de contorno e há a necessidade também da utilização de uma condição de ortogonalidade;
- Ao contrário do formalismo discreto, este método deve apresentar grande complexidade na determinação do modelo de estruturas com vários elos flexíveis, principalmente em razão de dificuldades na obtenção das funções de forma, dificuldades estas oriundas da complexidade envolvida no trabalho com as condições de contorno.

Sobre o Método dos Modos Assumidos (funções de forma polinomiais)

- Esta técnica apresenta um bom desempenho quando um grande número de elementos clássicos de terceira ordem é considerado. Porém, este elevado número de elementos amplia em muito a complexidade do modelo;
- Testou-se, então, qual seria o desempenho de um modelo obtido a partir de um único elemento de quinta ordem. Os resultados mostraram erros muito pequenos nos zeros, mas erros significativos nos pólos;
- Algum erro já era esperado uma vez que as funções de forma (analiticamente constituídas de funções trigonométricas do tipo seno, seno hiperbólico, etc.) são aproximadas por funções polinomiais.

5. REFERÊNCIAS

- Chrétien, J. P. 1990 “Modélisation des Chaînes Polyarticulés Flexibles: Synthèse Bibliographique”, Rapport D. R. E. T./ D. E. R. ^a n-1/7735, Toulouse, França.
- Meirovitch, L., 1967 “Analytical Methods in Vibrations”, The Macmillan Company, New York.
- Pereira, A. E. L., 1999, “Um Estudo Sobre Modelagem Matemática de Estruturas Flexíveis”, Dissertação de Mestrado, UFRGS, Porto Alegre, RS, Brasil.
- Schimtz, E., 1985, “Experiments on the End-Point Control of a very Flexible One-Link Manipulator”, PhD. Thesis, Satnford, california, Unite States.

UM ESTUDO SOBRE A MODELAGEM MATEMÁTICA DE ESTRUTURAS FLEXÍVEIS (PARTE I)

Adriana Elisa Ladeira Pereira
Sebastião Cícero Pinheiro Gomes
Álvaro Luiz De Bortoli

Fundação Universidade Federal do Rio Grande, Dep. de Matemática, CEP 96201-900, Rio Grande, RS, Brasil, E-mail: mmaaelp@super.furg.br, dmtscpg@super.furg.br
Universidade Federal do Rio Grande do Sul, PPGMAp, CEP 91509-900, Porto Alegre, RS, Brasil, E-mail: dbortoli@mat.ufrgs.br

Resumo

Controlar estruturas flexíveis, com o intuito de atenuar vibrações de forma ativa, é um problema complexo. Isto deve-se principalmente ao atuador estar colocado em uma extremidade, enquanto se deseja atenuar vibrações na outra extremidade da estrutura, existindo toda uma dinâmica flexível entre o atuador e o sensor a partir do qual a vibração é medida. Neste sentido, o conhecimento de um modelo dinâmico preditivo e realista pode ser de grande auxílio ao projeto de controladores com condições reais de implementação prática. Estes problemas motivaram a realização deste trabalho, o qual contém um estudo analítico detalhado do problema de modelagem dinâmica de uma estrutura do tipo manipulador flexível. Nesta primeira parte, uma técnica de modelagem é desenvolvida e analisada, tratando-se do Formalismo Discreto. Comparações de respostas em frequência entre as funções de transferência analítica e do modelo discreto são realizadas, concluindo-se ao final sobre a precisão e simplicidade deste primeiro formalismo de modelagem dinâmica estudado.

Palavras-chave: Modelagem, Manipulador, Flexível, Vibrações, Robótica.

1. INTRODUÇÃO

A partir dos anos oitenta, o estudo sobre o controle de estruturas flexíveis tem se intensificado, principalmente motivado por aplicações na robótica flexível (De Luca et al., 1998), (Isogai et al., 1999), (Kim, et al., 1997). Neste artigo serão desenvolvidas as equações dinâmicas governantes do movimento de uma estrutura que consiste em uma lâmina flexível, acoplada em uma extremidade a um atuador do tipo motor elétrico, a partir do qual é aplicado um torque de controle (Figura 1). Na outra extremidade existe uma massa constante, denominada de carga terminal. Serão ainda deduzidas as funções de transferência analíticas (exatas) e as funções de transferência a partir da abordagem discreta, para que se possa fazer comparações entre as mesmas.



Figura 1. Estrutura flexível

2. ESTUDO ANALÍTICO

Considera-se, inicialmente, uma estrutura flexível de comprimento l (Figura 2), fixa na extremidade onde se localiza o rotor, no qual é aplicado um torque T_m , e livre na outra extremidade na qual é colocada uma carga, sendo I_r a inércia do rotor e m_c a massa da carga (de momento de inércia rotacional desprezível).

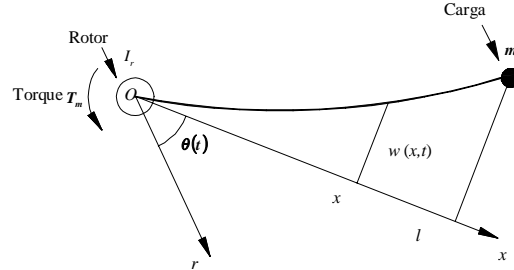


Figura 2. Estrutura flexível.

2.1 Equações da Dinâmica

As equações das Energias Cinética e Potencial são dadas por (Schmitz, 1985):

$$E_c = \frac{1}{2} I_r \dot{\theta}^2 + \frac{1}{2} \int_0^l \left(\frac{\partial w}{\partial t} + x \dot{\theta} \right)^2 \rho dx + \frac{m_c}{2} \left(\frac{\partial w}{\partial t} + x \dot{\theta} \right)^2 \Big|_{x=l}, \quad (1)$$

$$E_p = \frac{1}{2} \int_0^l EI \left(\frac{\partial^2 w}{\partial x^2}(x,t) \right)^2 dx, \quad (2)$$

onde ρ é a massa por unidade de comprimento da linha elástica, E é o Módulo d'Young e I é o momento de inércia da área da secção transversal da estrutura. Foram negligenciados esforços de cisalhamento na estrutura quando esta se deforma, hipótese não restritiva para estruturas delgadas conforme a considerada neste trabalho.

Pelo Princípio de Hamilton (Meirovitch, 1967), obtém-se as seguintes equações do movimento:

$$(I_r + I_B) \ddot{\theta}(t) + \rho \int_0^l x \frac{\partial^2 w}{\partial t^2} dx + m_c l (\ddot{w}(l,t) + l \ddot{\theta}(t)) = T_m, \quad (3)$$

$$EI \frac{\partial^4 w}{\partial x^4}(x,t) + \rho \frac{\partial^2 w}{\partial t^2}(x,t) + \rho x \ddot{\theta}(t) = 0, \quad (4)$$

com as seguintes condições de contorno (Schmitz, 1985):

$$\begin{aligned} EI \frac{\partial^2 w}{\partial x^2}(x,t) \Big|_{x=0} + T_m - I_r \ddot{\theta} &= 0 \\ w(x,t) \Big|_{x=0} = w(0,t) &= 0 \\ EI \frac{\partial^2 w}{\partial x^2}(x,t) \Big|_{x=l} = EI \frac{\partial^2 w}{\partial x^2}(l,t) &= 0 \\ EI \frac{\partial^3 w}{\partial x^3}(x,t) \Big|_{x=l} = m_c \left(\frac{\partial^2 w}{\partial t^2} + x \ddot{\theta} \right) \Big|_{x=l} & \end{aligned} \quad (5)$$

Considerando-se a transformação de variável sugerida por Brakwel (Soares, 1997):

$$y(x,t) = w(x,t) + x \theta(t), \quad (6)$$

resulta o sistema em função da variável $y(x,t)$:

$$EI \frac{\partial^4 y}{\partial x^4}(x,t) + \rho \frac{\partial^2 y}{\partial t^2}(x,t) = 0, \quad (7)$$

$$I_r \ddot{\theta}(t) + \rho \int_0^l x \frac{\partial^2 y}{\partial t^2} dx + m_c l \ddot{y}(l,t) = T_m, \quad (8)$$

com as respectivas condições de contorno:

$$EI \frac{\partial^2 y}{\partial x^2}(x,t) \Big|_{x=0} + T_m - I_r \ddot{\theta} = 0$$

$$y(x,t) \Big|_{x=0} = y(0,t) = w(0,t) + 0.\theta(t) = 0 \quad . \quad (9)$$

$$EI \frac{\partial^2 y}{\partial x^2}(x,t) \Big|_{x=l} = EI \frac{\partial^2 w}{\partial x^2}(l,t) = 0$$

$$EI \frac{\partial^3 y}{\partial x^3}(x,t) \Big|_{x=l} = m_c \frac{\partial^2 y}{\partial t^2}(x,t) \Big|_{x=l}$$

Aplicando-se a Transformada de Laplace na equação (7), resulta em:

$$\frac{d^4 \bar{y}}{dx^4} - \beta^4 \bar{y} = 0 \quad , \quad (10)$$

sendo $\beta^4 = -\frac{\rho}{EI} s^2$ e $\lambda = \beta l$, com as seguintes condições de contorno:

$$\bar{y}(0,s) = 0$$

$$EI \frac{d^2 \bar{y}}{dx^2} \Big|_{x=0} + \bar{T}(s) = I_r s^2 \frac{d\bar{y}}{dx} \Big|_{x=0} \quad , \quad (11)$$

$$EI \frac{d^2 \bar{y}}{dx^2} \Big|_{x=l} = 0$$

$$EI \frac{d^3 \bar{y}}{dx^3} \Big|_{x=l} = m_c s^2 \bar{y} \Big|_{x=l}$$

onde $\bar{T}(s)$ é a transformada de Laplace de $T_m(t)$. A solução geral de (10) é dada na forma:

$$\bar{y} = A \operatorname{sen}(\beta x) + B \operatorname{senh}(\beta x) + C \cos(\beta x) + D \operatorname{cosh}(\beta x) \quad (12)$$

É fácil demonstrar ainda que (Pereira, 1999):

$$\bar{\theta}(s) = \frac{\partial \bar{y}(0,s)}{\partial x} \quad (13)$$

e

$$y_i(s) = \bar{y}(l,s) \quad (14)$$

onde $y_i(s)$ é a transformada de Laplace da posição da carga.

Aplicando-se à equação (12) as condições de contorno (11), resulta no sistema:

$$M(\lambda) \begin{bmatrix} A \\ B \\ C \end{bmatrix} = -\frac{1}{EI\beta^2} \begin{bmatrix} \bar{T} \\ 0 \\ 0 \end{bmatrix} \quad , \quad (15)$$

onde:

$$M(\lambda) = \begin{bmatrix} \varepsilon \lambda^3 & \varepsilon \lambda^3 & -2 \\ -\operatorname{sen} \lambda & \operatorname{senh} \lambda & -(\cos \lambda + \operatorname{cosh} \lambda) \\ -(\cos \lambda + \eta \lambda \operatorname{sen} \lambda) & \operatorname{cosh} \lambda + \eta \lambda \operatorname{senh} \lambda & \operatorname{sen} \lambda - \operatorname{senh} \lambda + \eta \lambda (\cos \lambda - \operatorname{cosh} \lambda) \end{bmatrix} \quad ,$$

$D = -C$, $\varepsilon = \frac{I_r}{3I_B}$ e $\eta = \frac{m_c l^2}{3I_B}$, sendo I_r o momento de inércia do rotor e I_B o momento de inércia

da lâmina. As constantes A , B , C e D são obtidas da solução de (15), sendo funções de λ e de $\bar{T}(s)$. Substituindo-as na solução geral (12), obtém-se $\bar{y}(s)$ em função de λ e de $\bar{T}(s)$.

2.2 Funções de Transferência Analíticas

Os pólos das funções de transferência são as raízes da equação $D(\lambda)=0$, sendo $D(\lambda)$ obtido pelo determinante da matriz $M(\lambda)$, que em série de Taylor pode ser escrito como (Pereira, 1999):

$$D(\lambda) = \frac{4}{3} \lambda^3 (1 + 3\varepsilon + 3\eta) \prod_{i=0}^{i=\infty} \left(1 - \frac{\lambda^4}{\lambda_i^4} \right) = \frac{4}{3} \lambda^3 (1 + 3\varepsilon + 3\eta) \prod_{i=1}^{i=\infty} \left(1 + \frac{w_i^2}{w_i^2} \right) \quad (16)$$

sendo $w_i = \sqrt{\frac{EI}{\rho l^4}} \lambda_i, i = 1, 2, \dots, \infty$ e $\lambda_i, i = 1, 2, \dots, \infty$ as raízes de $D(\lambda) = 0$.

2.2.1 Função de Transferência no caso colocado

No caso colocado, observa-se a posição angular do rotor, ou seja, a observação é feita no mesmo local onde está sendo aplicado o torque motor. As equações (12) e (13) permitem a obtenção do numerador da função de transferência neste caso (Pereira, 1999), o qual possui a forma:

$$N_\theta(\lambda) = -\frac{2}{EI\beta} [1 + \cos \lambda \sinh \lambda + n\lambda(\sinh \lambda \cos \lambda - \sin \lambda \cosh \lambda)] \quad (17)$$

A função de transferência pode então ser escrita como:

$$\frac{\bar{\theta}(s)}{\bar{T}(s)} = \frac{N_\theta(\lambda)}{D(\lambda)} = \frac{1}{I_T s^2} \prod_{i=1}^{i=\infty} \left(\frac{1 + \frac{s^2}{\Omega_i^2}}{1 + \frac{s^2}{w_i^2}} \right), \quad (18)$$

onde $I_T = I_B + I_r + m_c l^2$ é o momento de inércia de corpo rígido.

2.2.2 Função de Transferência do caso não colocado

No caso não colocado, observa-se a posição angular da carga, ou seja, a observação é feita num local diferente do qual está sendo aplicado o torque motor. As equações (12) e (14) permitem a obtenção do numerador da função de transferência neste caso, sob a forma [Pereira, 1999]:

$$N_{y_i}(\lambda) = -\frac{2}{EI\beta^2} (\sin \lambda + \sinh \lambda) \quad (19)$$

A função de transferência pode então ser escrita como:

$$\frac{\bar{y}_i(s)}{\bar{T}(s)} = \frac{N_{y_i}(\lambda)}{D(\lambda)} = \frac{l}{I_T s^2} \prod_{i=1}^{i=\infty} \left(\frac{1 - \frac{s^2}{\alpha_i^2}}{1 + \frac{s^2}{w_i^2}} \right). \quad (20)$$

2.3 Gráficos das Funções de Transferência

Considerando-se os parâmetros de um manipulador flexível real apresentado em (Gomes e Chrétien, 1992a), foram traçadas as respostas em frequência das funções de transferência analíticas nos casos colocado e não colocado, vistas nas figuras 3 e 4 respectivamente.

Na Figura 3 verifica-se a presença de picos que representam os valores nos quais a função de transferência do caso colocado tende a mais infinito, equivalentes, portanto, aos pares de pólos complexos e conjugados em malha aberta. Percebe-se também a existência de pontos nos quais a função de transferência tende a menos infinito, correspondentes aos pares de zeros complexos e conjugados em malha aberta. Trata-se, portanto, de um sistema de fase mínima.

Na Figura 4 identifica-se a presença de picos, representando os pares de pólos complexos e conjugados em malha aberta e verifica-se a ausência de zeros complexos uma vez que, no caso não colocado, o sistema é de fase não mínima.

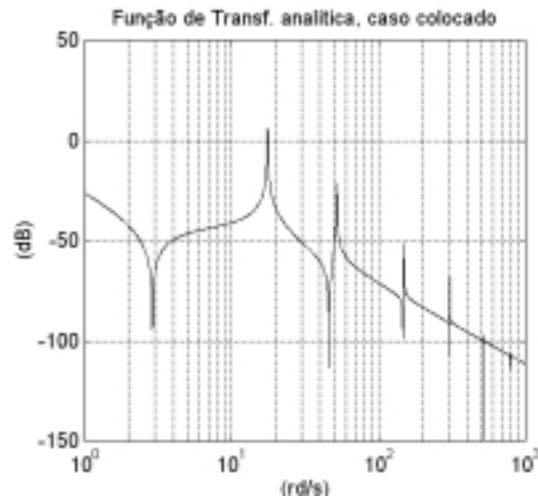


Figura 3. Gráfico da Função de Transferência Analítica (caso colocado).

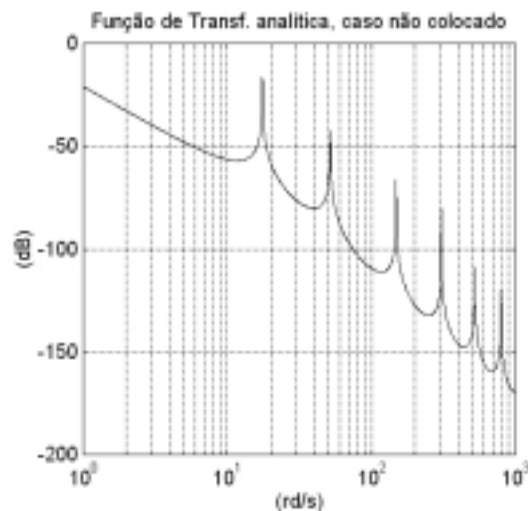


Figura 4. Gráfico da Função de Transferência Analítica (Caso não colocado).

3. ABORDAGEM DISCRETA

Na Abordagem Discreta, a estrutura de flexibilidade contínua é dividida em partes rígidas conectadas por elementos flexíveis chamados de articulações fictícias, sendo que neste artigo será considerado o caso em que a estrutura de comprimento l é dividida em quatro partes rígidas, ou seja, são colocadas três articulações fictícias, conforme mostra a Figura 5.

As partes rígidas (elos) têm massas concentradas nos centros de massa, ou seja, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) e (x_4, y_4) (coordenadas dos centros de massa dos elos de massas m_1 , m_2 , m_3 e m_4 respectivamente), enquanto (x_c, y_c) são as coordenadas do centro de massa

da carga de massa m_c . Conforme observa-se na Figura 5, k é a constante elástica dada por (Gomes e Chrétien, 1992 b): $k = \frac{nEI}{l}$, onde n é o número de articulações fictícias, EI é o módulo d'Young multiplicado pela inércia da seção reta.

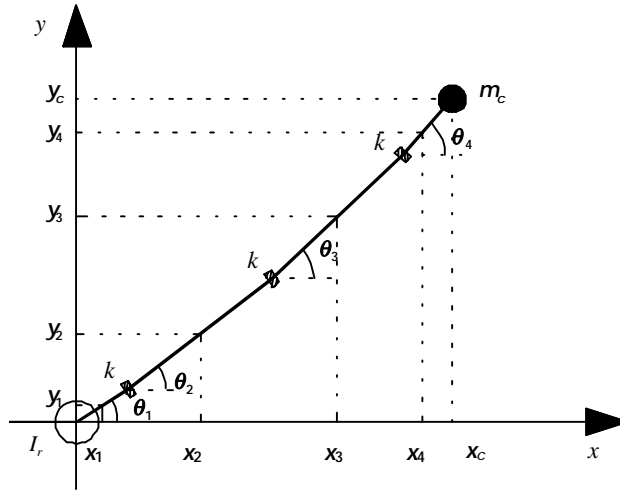


Figura 5. Estrutura dividida em quatro partes rígidas.

3.1 Modelo com três modos flexíveis

Será desenvolvida a modelagem matemática para o caso citado, ou seja, para uma estrutura flexível dividida em quatro partes rígidas com três articulações fictícias (ver Figura 5).

Neste caso, as Energias Cinética e Potencial são escritas na forma:

$$E_c = \frac{1}{2} I_r \dot{\theta}_1^2 + \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2) + \frac{1}{2} m_4 (\dot{x}_4^2 + \dot{y}_4^2) + \frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2) \quad (21)$$

$$E_p = \frac{1}{2} k (\theta_2 - \theta_1)^2 + \frac{1}{2} k (\theta_3 - \theta_2)^2 + \frac{1}{2} k (\theta_4 - \theta_3)^2. \quad (22)$$

Aplicando-se as equações de Euler-Lagrange, obtém-se o seguinte sistema:

$$[I] \ddot{\vec{\theta}} + [C_{at}] \dot{\vec{\theta}} + [K] \vec{\theta} = \vec{T}_m, \quad (23)$$

onde:

$$I = \begin{bmatrix} I_r + \frac{l^2}{64} \left(\frac{m_1}{4} + m_2 + m_3 + m_4 + m_c \right) & \frac{3l^2}{64} \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) & \frac{3l^2}{64} \left(\frac{m_3}{2} + m_4 + m_c \right) & \frac{l^2}{64} \left(\frac{m_4}{2} + m_c \right) \\ \frac{3l^2}{64} \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) & \frac{9l^2}{64} \left(\frac{m_2}{4} + m_3 + m_4 + m_c \right) & \frac{9l^2}{64} \left(\frac{m_3}{2} + m_4 + m_c \right) & \frac{3l^2}{64} \left(\frac{m_4}{2} + m_c \right) \\ \frac{3l^2}{64} \left(\frac{m_3}{2} + m_4 + m_c \right) & \frac{9l^2}{64} \left(\frac{m_3}{2} + m_4 + m_c \right) & \frac{9l^2}{64} \left(\frac{m_3}{4} + m_4 + m_c \right) & \frac{3l^2}{64} \left(\frac{m_4}{2} + m_c \right) \\ \frac{l^2}{64} \left(\frac{m_4}{2} + m_c \right) & \frac{3l^2}{64} \left(\frac{m_4}{2} + m_c \right) & \frac{3l^2}{64} \left(\frac{m_4}{2} + m_c \right) & \frac{l^2}{64} \left(\frac{m_4}{4} + m_c \right) \end{bmatrix}$$

$$C_{at} = \begin{bmatrix} c_r + c_1 & -c_1 & 0 & 0 \\ -c_1 & c_1 + c_2 & -c_2 & 0 \\ 0 & -c_2 & c_2 + c_3 & -c_3 \\ 0 & 0 & -c_3 & c_3 \end{bmatrix}, \quad K = \begin{bmatrix} k & -k & 0 & 0 \\ -k & 2k & -k & 0 \\ 0 & -k & 2k & -k \\ 0 & 0 & -k & k \end{bmatrix}, \quad \vec{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \text{ e } \vec{T}_m = \begin{bmatrix} T_m \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

3.2 Funções de Transferência (sistema 3af)

Considerando-se o sistema de equações diferenciais na forma de estado e um vetor de observações

$$\dot{X} = AX + Bu; Y = CX \quad (24)$$

e tomando-se a Transformada de Laplace deste sistema, obtém-se:

$$G(s) = C(sI - A)^{-1}B \quad (25)$$

onde $G(s)$ é a função de transferência, sendo o numerador e o denominador desta função dois polinômios em s , onde as raízes do denominador são os autovalores da matriz A (pólos de $G(s)$), sendo as raízes do numerador os zeros de $G(s)$.

A matriz $C = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ permite a observação da posição angular do rotor, tratando-se portanto do caso colocado, para o qual a função de transferência possui a forma:

$$\frac{\theta_1}{T_m}(s) = \frac{\sum_{i=0}^6 \chi_i s^i}{\sum_{j=0}^8 \gamma_j s^j} \approx \frac{1}{I_T s^2} \frac{\left(1 + \frac{s^2}{\Omega_1^2}\right) \left(1 + \frac{s^2}{\Omega_2^2}\right) \left(1 + \frac{s^2}{\Omega_3^2}\right)}{\left(1 + \frac{s^2}{\omega_1^2}\right) \left(1 + \frac{s^2}{\omega_2^2}\right) \left(1 + \frac{s^2}{\omega_3^2}\right)} \quad (26)$$

Considerando-se a matriz de observação $C = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$, observa-se a posição da carga terminal, sendo a função de transferência, neste caso não colocado:

$$\frac{\theta_4}{T_m}(s) = \frac{\sum_{i=0}^6 \xi_i s^i}{\sum_{j=0}^8 \psi_j s^j} \approx \frac{1}{I_T s^2} \frac{\left(1 - \frac{s^2}{\Omega_1^2}\right) \left(1 - \frac{s^2}{\Omega_2^2}\right) \left(1 - \frac{s^2}{\Omega_3^2}\right)}{\left(1 + \frac{s^2}{\omega_1^2}\right) \left(1 + \frac{s^2}{\omega_2^2}\right) \left(1 + \frac{s^2}{\omega_3^2}\right)} \quad (27)$$

Na Figura 6 observa-se a função de transferência do modelo discreto sobreposta ao resultado analítico (Figura 3), para o caso colocado. Verifica-se que os dois primeiros pares de pólos e zeros se aproximam bem dos dois primeiros pares de pólos e zeros observados na Função de Transferência Analítica. Já o terceiro par pólo-zero do caso Discreto é intermediário aos terceiro e quarto pares de pólos e zeros do caso analítico.

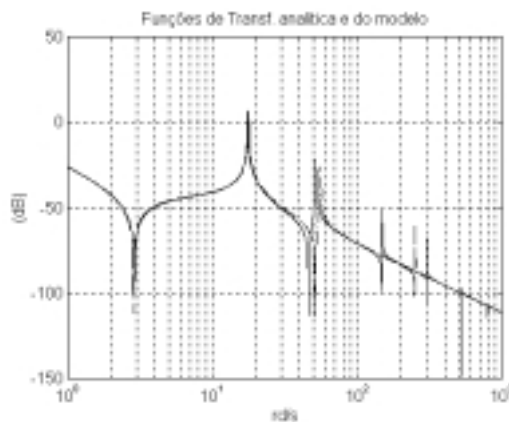


Figura 6 Funções de Transferência Analítica e do Modelo Discreto com 3 modos flexíveis(caso colocado-3af), sendo negligenciados os torques de atrito do Modelo Discreto.

4. CONCLUSÕES

Sobre o Estudo Analítico

- As funções de transferência analíticas são fundamentais para a aferição dos modelos dinâmicos encontrados a partir de diferentes técnicas;
- Com estas funções de transferência obtém-se as respostas frequenciais as quais revelam os pólos (comum aos casos colocado e não colocado) e os zeros (específicos para cada caso);

Sobre o Formalismo Discreto

- Trata-se da mais simples das técnicas em termos matemáticos uma vez que, não há a necessidade de se trabalhar com equações diferenciais parciais, nem, obviamente, com as conseqüentes condições de contorno;
- Esta técnica mostrou ser bem representativa até os primeiros dois modos de vibração, sendo que o mesmo não acontece a partir do terceiro modo, isto para valores iguais das constantes elásticas das articulações fictícias (conforme considerado neste trabalho);
- Devido a sua simplicidade, o Formalismo Discreto apresenta facilidades na obtenção de modelos dinâmicos e cinemáticos para manipuladores com vários elos flexíveis, o mesmo não acontecendo com os demais formalismos vistos na continuação deste trabalho (parte II).

5. REFERÊNCIAS

- De Luca, A., Panzieri, S. and Ulivi, G., 1998, "Stable Inversion Control for Flexible Link Manipulators", IEEE-International Conference on Robotics and Automation, Leuven, Belgium.
- Gomes, S. C. P. and Chrétien, J. P., 1992 a, "Dynamic modeling and friction compensated control of a robot manipulator joint", IEEE-International Conference on Robotic and Automation.
- Gomes, S. C. P. and Chrétien, J. P., 1992 b, "Dynamic modeling and friction compensated control of 1 d. o. f. rigid or flexible manipulator", In Microdynamic et Pointage de Grande Précision, CNES, 1er Symposium International.
- Isogai, M., Arai, F., and Fukuda, T., 1999, "Modeling and Vibration Control with Neural Network for Flexible Multi-Link Structures", IEEE-International Conference on Robotics and Automation, Nice, France.
- Kim, J., Suzuki, K., Yamano, M. and Uchiyama M., 1997, "Vibration Suppression Control of Constrained Spatial Flexible Manipulators", IEEE-International Conference on Robotics and Automation, Albuquerque, New México.
- Meirovitch, L., 1967, "Analytical Methods in Vibrations", The Macmillan Company, New York.
- Pereira, A. E. L., 1999, "Um Estudo Sobre Modelagem Matemática de Estruturas Flexíveis", Dissertação de Mestrado, UFRGS, Porto Alegre, RS, Brasil.
- Schmitz, E., 1985, "Experiments on the End-Point Control of a very Flexible One-link Manipulator", Ph.D. Thesis, Stanford, California, Unite States.
- Soares, A. M. S., 1997, "Modelagem e Identificação Experimental de Estruturas Flexíveis", Tese de Doutorado, ITA, São José dos Campos, Brasil.

TELEROBÓTICA : TELEOPERAÇÃO DO ROBÔ ABB IRB 2000 VIA WWW

Alberto José Álvares

Luís Felipe de Aguiar Paulinyi

GRACO - Grupo de Automação e Controle, Departamento de Engenharia Mecânica, Faculdade de Tecnologia, UnB, CEP 70910-900 Brasília - DF, Brasil, E-mail: alvares@graco.unb.br

Resumo

A telerobótica permite que sistemas robóticos sejam controlados remotamente, eliminando a necessidade da presença humana no local. Este trabalho apresenta o desenvolvimento de um sistema para controlar remotamente o Robô Industrial com seis graus de liberdade IRB 2000 da Asea Brown Boveri utilizando a Internet como via de controle. O controlador do Robô ABB IRB 2000 tem incorporado um sistema de controle remoto através da interface serial RS-232 baseado em 42 funções. Utiliza-se desta capacidade de comunicação remota, interligando o sistema em rede de comunicação local utilizando o protocolo TCP/IP (Transport Control Protocol/Internet Protocol). O sistema é baseado em uma arquitetura Cliente/Servidor, tendo como servidor uma plataforma UNIX (Linux), que disponibiliza o serviço WWW (World Wide Web), que será acessado pelo cliente utilizando um browser (Netscape ou Explorer). O sistema de teleoperação desenvolvido está disponível no endereço <http://webrobot.graco.unb.br>. São abordados aspectos referentes à segurança e ao tempo de resposta do Robô ao Servidor WWW. Por outro lado, serviços como inicialização do Robô, iniciar programa, ligar e desligar Robô, download e upload de programas, entre outros, são plenamente satisfeitos.

Palavras-chave: TeleRobótica, Internet, Teleoperação, Robótica, Mecatrônica

1. INTRODUÇÃO

Sistemas robóticos teleoperados têm sido amplamente aplicados em automação de fábricas, exploração espacial, exploração submarina, aplicações militares, etc., podendo realizar uma série de tarefas sem a presença e interferência humana. Sistemas como o Mars Pathfinder (Goodall, 1997), são capazes de tomar algumas decisões independentemente. O controle desses sistemas é feito remotamente, e comandos são transmitidos por uma grande diversidade de meios tais como o rádio, microondas e redes de computadores. A TeleRobótica utilizando a Internet como *link* de comunicação é um novo campo de pesquisa que se desenvolve na área da Teleoperação tendo muitos grupos de pesquisa atuando nesta promissora área da Telemática (Monteiro *et al.*, 1997), em função, principalmente, dos baixos custos de acesso à Internet.

As vantagens de se utilizar a rede de comunicação Internet como via de controle deve-se a sua versatilidade, pois disponibiliza ao usuário uma série de serviços, tais como FTP (*File Transport Protocol*), TELNET, WWW (*World Wide Web*), *Electronic Mail*, permitindo a transmissão de informações de maneira simples, além de ter uma vasta distribuição em todo mundo, sendo um meio bastante acessível e possuindo uma interface com muitos recursos multimídia, de fácil manipulação e baixo custo. Todas essas características têm sido favoráveis ao aparecimento de vários sistemas robóticos controlados via Internet. O Telerobot da Austrália (<http://telerobot.mech.uwa.edu.au>) é um sistema robótico teleoperado pela internet que permite a manipulação de objetos utilizando-se de uma garra (Taylor & Trevelyan, 1995). O RobWebCam do GRACO - Grupo de Automação e Controle da Universidade de Brasília (<http://www.graco.unb.br/robwebcam>), permite o controle remoto de um manipulador para posicionar uma câmera de vídeo transmitindo imagem de vídeo em tempo real (Álvares & Romariz, 1998).

A Telerobótica pode ser definida como sendo uma área da Telemática e da Robótica voltada à teleoperação de sistemas robóticos utilizando-se de um *link* de comunicação (Klauss *et al.*, 1997). Uma nova terminologia esta sendo empregada no caso de se utilizar a Rede de Comunicação Internet como *link* de telecomunicações. Neste caso, designa-se de sistemas *World Wide Web Robots*, *WebRobots* ou simplesmente

Internet Robots. Esta terminologia é válida para aplicações que utilizam Robôs Industriais, Manipuladores, *Pan-Tilt*, Máquinas de Comando Numérico e outros equipamentos industriais similares.

Este trabalho tem como objetivo a implementação de uma interface utilizando os recursos do ambiente WWW para controlar remotamente o Robô ABB IRB 2000, tendo a Internet como via de controle.

2. ARQUITETURA DO SISTEMA

O controlador do Robô ABB IRB 2000 tem incorporado um sistema de controle remoto via interface serial RS-232 baseado em 42 funções. Este trabalho utiliza-se desta capacidade de comunicação remota interligando o sistema, em rede local de comunicação, utilizando o protocolo TCP/IP (*Transport Control Protocol/Internet Protocol*). O sistema concebido é baseado na arquitetura cliente/servidor, tendo como servidor uma plataforma UNIX (Linux) que disponibiliza o serviço WWW, que será acessado pelo cliente utilizando um *browser* (Netscape ou Explorer). O servidor WWW conectado ao Robô via interface serial disponibiliza as funções de controle. A teleoperação é monitorada através da captura de imagem em tempo real e informações/status do Robô são enviadas pelo seu controlador. A metodologia utilizada para implementação do sistema de teleoperação via Internet é baseada em Álvares & Romariz (1999).

2.1 Módulos WebCam e WebRobot: requisitos para teleoperação

Sistemas teleoperados remotamente necessitam de dados e/ou imagens do objeto de controle, bem como, da transmissão de comandos através de um *link* de comunicação, que neste caso é baseado em conexão via Rede de Comunicação (Schilling *et al.*, 1997), através do Protocolo Internet (IP). A figura 1 apresenta a arquitetura do sistema, onde o Servidor Unix HTTP (*Hypertext Transfer Protocol*) disponibiliza dois tipos de serviços básicos, que estão representados na figura 1 através de dois módulos:

- . WebCam: Visualização do objeto teleoperado, através de vídeo e/ou imagem;
- . WebRobot: Disponibilização de funções de controle remoto do objeto teleoperado.

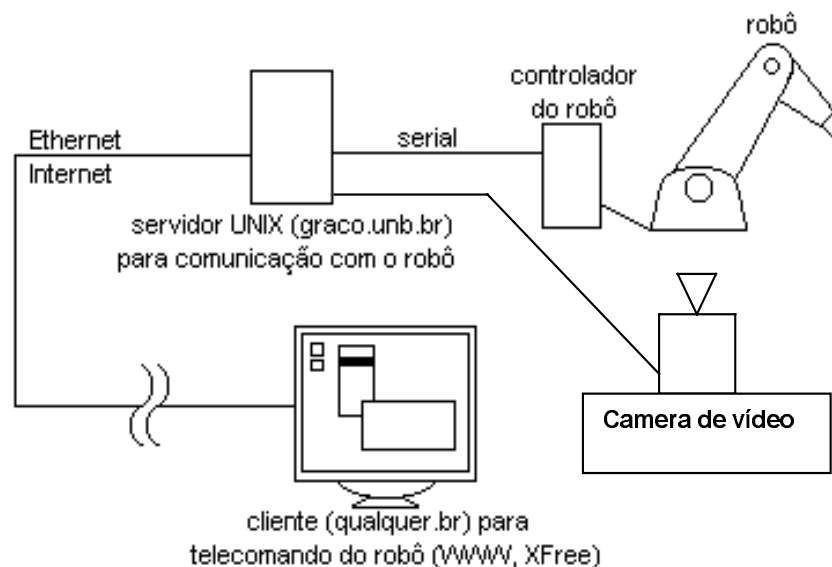


Figura 1. Arquitetura do sistema de teleoperação via Internet: WebCam e WebRobot.

Servidor WebCam: visualização do objeto teleoperado. Este módulo é responsável pela captura de imagens em tempo real do Robô, utilizando-se da tecnologia WebCam ou WebVideo (Wolf & Froitzheim, 1997). É utilizado o sistema RobWebCam desenvolvido no GRACO em 1997 (Álvares & Romariz, 1998) como servidor WebCam.

O sistema de vídeo é baseado na SunVideo (Sun, 1994) que possibilita a captura, digitalização e compressão em formato JPEG (Joint Photographic Experts Group) de um sinal de vídeo NTSC ou PAL não modularizado (composto ou S-Vídeo). Assim, utilizou-se a plataforma Solaris como WebCam. Cabe destacar que o sistema RobWebCam tem um manipulador (*Pan-tilt*) que pode posicionar a câmera para possibilitar uma melhor monitoração do ambiente (Álvares & Romariz, 1998 e 1999).

Servidor WebRobot: funções de controle remoto do objeto teleoperado. A fim de teleoperar o Robô utiliza-se o protocolo de comunicação ADLP-10 da ABB através de uma interface de comunicação serial conectado ao Servidor HTTP - WebRobot. O servidor WebRobot é implementado no sistema operacional Linux.

O mecanismo de acesso às funções do objeto teleoperado é baseado em programas CGI (*Common Gateway Interface*) e HTML (*Hypertext Markup Language*). Para cada função disponibilizada pelo protocolo de comunicação do Robô existe um programa CGI que é acessado no servidor WebRobot pelo cliente utilizando-se de um *browser* WWW.

2.2 Definição dos subsistemas para a interface de teleoperação

A partir da arquitetura definida para o sistema implementou-se o ambiente de teleoperação em uma plataforma multiusuário utilizando o Sistema Operacional Linux e Solaris, desenvolvendo-se três subsistemas de interface com o usuário (fig. 2):

- interface cliente/servidor (Páginas visualizadas com *Browser*);
- interface de comunicação servidor/controlador do Robô (Servidor HTTP - WebRobot);
- interface de monitoração visual do sistema (Servidor HTTP - WebCam).

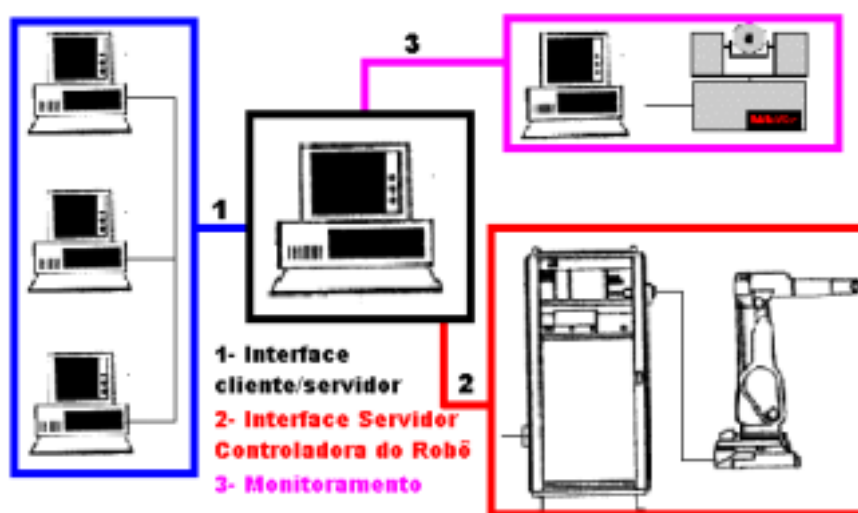


Figura 2. Subsistemas do sistema telerobótico RobWebLink (*Robot Web Link*).

Interface cliente e servidor. A utilização do protocolo TCP/IP, Internet/Intranet (Eckel & Hare, 1995), para a comunicação entre o cliente e o servidor é bastante atrativa devido a sua presença no mundo inteiro, através da Internet. Há alguns problemas quando se utiliza o protocolo TCP/IP, pois a aplicação pode sofrer atrasos (*delays*) não apresentando um tempo determinístico de acesso, o que não é desejável para aplicações em tempo real.

A utilização de servidor WWW apresenta uma interface rica e simples de ser utilizada. Os clientes/navegadores apresentam texto, gráficos, vídeo e áudio, permitindo uma comunicação multimídia entre cliente e servidor. O cliente acessa o servidor UNIX via WWW, e de maneira interativa define as tarefas a serem executadas pelo Robô, que retornará uma confirmação ou negação da execução para o cliente, bem como, uma imagem em tempo real do ambiente onde se encontra o Robô.

Servidor e controlador. O Robô ABB IRB 2000 dispõe de uma porta serial para a comunicação com um computador pessoal (*Computer Link*) através do protocolo ADLP-10 (*ABB Data Link Protocol*) (ABB Robotics, 1993), utilizando comunicação assíncrona entre o servidor UNIX e o controlador do Robô, com palavras de oito *bits*, paridade e *stop bit*. A função básica do protocolo é estabelecer a comunicação, coordenando o envio e recebimento de dados, enviados em bloco. Cada bloco de dados é composto de um telegrama, que por sua vez é descrito pela biblioteca ARAP (*ABB Robot Aplicacion Protocol*). O

procedimento de comunicação é somente aplicável ponto-a-ponto, ou seja, a comunicação de duas unidades de equipamento onde uma é superior a outra (*master/slave*).

A biblioteca ARAP é composta de 42 funções de troca de dados entre o Robô e o computador, descrevendo a forma como os dados e respostas do Robô são enviados e códigos de erro que podem ocorrer em uma transmissão de dados.

Monitoramento. Para garantir a segurança na execução das funções de comando, além das respostas enviadas pelo controlador do Robô, é necessário o monitoramento através da captura de imagens em tempo real. Outro fator importante para a segurança é verificar quais são os usuários que estão acessando o servidor e controla-lo por meio de senhas.

O monitoramento por vídeo é feito por uma câmera (Sun, 1994) conectada a um servidor que envia as imagens via WWW. A realimentação gráfica é viável quando se utiliza velocidade de transmissão acima de 64 Kbps.

3. SISTEMA ROBWEBLINK (*ROBOT WEB LINK*): IMPLEMENTAÇÃO

3.1 Especificação do hardware: WebRobot

O hardware utilizado para o sistema é composto pelo Robô ABB IRB 2000, pelos servidores WWW (Linux e Solaris) e pelo cliente que acessará os servidores que controlam a teleoperação do Robô, a captura de imagem e posicionamento da câmera que possibilita a realimentação visual.

O modelo IRB 2000 da Asea Brown Boveri, é um manipulador industrial equipado com funções para soldagem a arco, com seis graus de liberdade, capacidade máxima de carga de 10 kg e sua linguagem de programação nativa é a ARLA.

Os servidores UNIX são: um computador pessoal com o sistema operacional (*OS*) Linux e uma Workstation Sparc20 com OS Solaris 2.6.

3.2 Descrição do protocolo ADLP-10/ARAP

O protocolo ADLP-10 é um procedimento para comunicação assíncrona entre duas estações com um sistema de hierarquia, transmissão assíncrona serial no modo *half-duplex*, possibilidade de início de transmissão por ambos os lados (superior e subordinado). O quadro de dados é composto por: palavra de 8 *bits* + 1 *bit* de paridade + 1 *stop bit*. A função básica do ADLP-10 é estabelecer a comunicação verificando o envio e recebimento de mensagens, que são divididas em blocos, garantindo a integridade dos dados transmitidos. A verificação da integridade é feita através da paridade e pela soma de checagem do bloco. Cada bloco de dados é composto de um telegrama, descrito pela biblioteca ARAP.

O protocolo ADLP-10. O protocolo de comunicação define alguns sinais que podem ser utilizados por qualquer uma das estações (computador ou Robô) para iniciar uma comunicação. Cada sinal tem uma função durante o processo de comunicação. Se os sinais não forem enviados de maneira correta a comunicação é interrompida.

Para iniciar a comunicação a estação envia um ENQ (*enquiry*) dessa forma ela espera uma resposta da outra estação para verificar se ela está pronta para continuar o processo de comunicação. As respostas possíveis são: ACK (*acknowledge*) que a estação receptora responde quando está pronta para comunicar-se; WACK (*wait and acknoeledge*) indica que a estação receptora está funcionando mas não está pronta para a comunicação; RVI (*reverse interrupt*) indica que a estação deseja enviar a mensagem e não recebê-la; NAK (*negative acknowledgement*) indica que a estação receptora não reconhece a comunicação.

Para o controle do telegrama existem outros sinais: DLE (*Data Link Escape*) indica quando inicia e quando termina o campo de dados do telegrama, STX (*Start of text*) verifica a paridade do envio de mensagens podendo ser par ou ímpar e EOT (*End of Transmission*) que finaliza a comunicação entre as estações.

A biblioteca ARAP. A Biblioteca ARAP descreve 42 funções de troca de dados entre o Robô e o

computador e a formatação dos telegramas de envio de dados e de recebimento, bem como, os códigos de erro respondidos pelo Robô.

Um telegrama sempre consiste de um Cabeçalho e um Campo de Dados. Todos os dados enviados pelo telegrama são enviados na forma binária. O Cabeçalho sempre possui o mesmo número de *bytes* e é composto por campos que estão sempre na mesma posição. O campo de dados possui tamanho e conteúdo variável dependendo da função a ser executada. O cabeçalho contém informações a respeito do tamanho do telegrama e do tipo do telegrama. O código da função a ser executada, pode ser visualizado na estrutura do telegrama na figura 3.

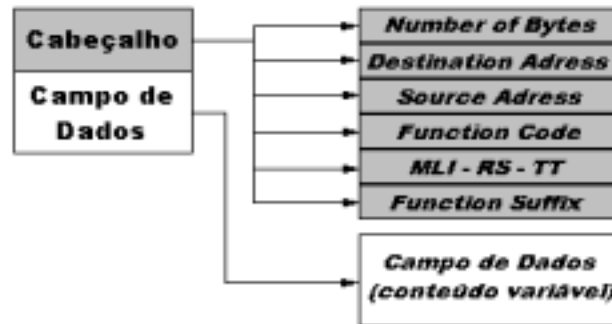


Figura 3. Estrutura do telegrama definido pela biblioteca ARAP.

O *Number of Bytes* ou NOB determina o número de *bytes* pelos quais o telegrama é composto. Este ocupa dois *bytes* do cabeçalho.

O *Destination Adress* determina quem será o a estação receptora do telegrama, utiliza-se normalmente 1 para o Robô e zero para o computador, esse *byte* (*byte 2*) é semelhante ao *Source Adress* que determina a estação que enviou o telegrama.

O *Function Code* especifica o número da função a ser executada. As funções foram divididas em três grupos de operações básicas: *as funções de programa, as funções de leitura e as funções de escrita.*

Os valores para *MLI* (Message Length Indicator), *RS* (Response Status) e *TT* (Telegram Type) são alinhados e enviados como um único *byte*. O *MLI* refere-se ao comprimento do telegrama (se possui ou não uma continuação). O *RS* comunica se a resposta é ou não um código de erro. O *TT* indica se o telegrama é um comando, uma resposta ou uma mensagem espontânea enviada pelo Robô. O *Function Suffix* é específico para cada função dando opções para o uso daquela função.

A tabela 1 apresenta todas as funções de programa da biblioteca ARAP que tem como função o gerenciamento de programas da memória do Robô.

Tabela 1. Funções de programas da biblioteca ARAP

Funções de Programa	Saída
Transfer of program/block SC→IRB	Transfere um programa ou bloco do computador para a memória do Robô
Start of robot program	Inicia um programa presente na memória do Robô
Stop of robot program	Para imediatamente a execução de um programa
Request from SC which programs are contained in the robot memory	Lista os programas existentes na memória do Robô e mostra o espaço livre de memória
Erase a specific program in robot memory from SC	Apaga um programa da memória do Robô
Request from SC for loading a program from floppy	Carrega um programa da unidade de disco flexível do Robô
Transfer of program from IRB to SC	Transfere um programa da memória do Robô para o computador

As funções de leitura, descritas na tabela 2, têm por objetivo reportar ao usuário as configurações do Robô.

Observa-se que os valores são enviados em binário e o limite é 256, no exemplo em que o programa executado é o programa 2000, que é obtido pela justaposição do *byte* 8 e 9 formando o binário 11111010000 que é o número 2000 na base 2. O valor do *function suffix* pode ser 1 ou 0. 0 para iniciar um programa do começo e 1 para iniciar um programa do ponto onde a última execução havia parado.

Funções de escrita, descritas na tabela 3, permitem que se modifiquem as configurações do Robô.

Tabela 2. Funções de Leitura da biblioteca ARAP

Funções de Leitura	Saída
Reading of TCP register data	Coordenadas de um TCP específico
Reading of location register	Valores de um Location Register específico
Reading of register data	Valores de um Register Data específico
Reading of sensor register data	Valor de Sensor Register específico
Reading of digital inputs	Entradas digitais
Reading of digital outputs	Saídas digitais
Reading of configuration data	Dados de configuração do Robô
Reading of frame register data	Valores de um Frame específico
Status request	Posição do Robô e modo de operação
Reading of arc-weld data	Leitura dos dados de soldagem
Reading of analogue inputs	Entradas Analógicas do Robô
Reading of analogue outputs	Saídas Analógicas do Robô
Reading of ARAP version	Versão da Biblioteca ARAP do Robô
Reading of resolver offsets	Offsets do Resolver

Tabela 3. Funções de escrita da biblioteca ARAP

Função de Escrita	Saída
Writing of TCP register data	Modifica coordenadas de um TCP específico
Writing of location register data	Modifica o valor de um Location Register específico
Writing of register data	Modifica o valor de um Register específico
Writing of sensor register data	Modifica o valor de um Sensor específico
Writing of digital output value	Modifica o valor de uma saída digital específico
Writing of configuration data	Modifica a configuração
Writing of frame register data	Modifica o valor de um frame específico
Change of robot operation mode	Modifica o modo de operação do Robô
Manoeuvring of IRB	Movimenta o Robô
Writing of arc weld data	Modifica os dados de soldagem
Writing of analogue output value	Modifica a saída analógica

3.3 Especificação do software de teleoperação do Robô: interface RobWebLink

As funções especificadas pela biblioteca ARAP foram divididas em três grupos, conforme apresentado na figura 4.

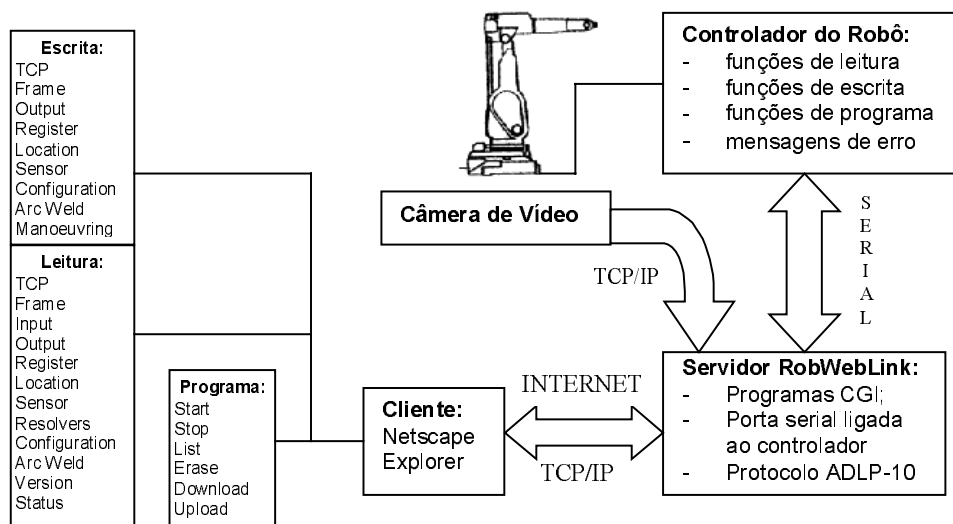


Figura 4. Arquitetura, fluxo de dados e funções disponibilizadas pelo RobWebLink.

Através das *funções de leitura* são obtidos os dados de configuração e memória do Robô. Com as *funções de escrita* o usuário pode alterar as configurações. Com as *funções de programa* o usuário pode fazer

uploads e *downloads* de programas, bem como executar programas que estejam na memória do Robô.

O cliente acessa a *homepage* com uma tela de comando dividida em *frames* correspondentes as funções, bem como às várias informações referentes ao Robô. Cada função possui uma página WWW correspondente, que descreve o funcionamento da função e pede para que o usuário defina os parâmetros da função. A leitura desses parâmetros é feita por um CGI, programado em C da GNU. A interface entre o servidor e o controlador do Robô estabelece a comunicação segundo o protocolo ADLP-10, enviando via RS232C os telegramas e recebendo suas respectivas respostas ou mensagens de erro, que são apresentadas no formato html e visualizadas na tela de comando.

4. IMPLEMENTAÇÃO DO SISTEMA ROBWEBLINK (<http://webrobot.graco.unb.br>)

O sistema implementado é baseado em uma arquitetura cliente/servidor em plataforma UNIX, utilizando o Sistema Operacional Linux e um servidor HTTP para a disponibilizar o serviço WWW. Para o monitoramento utiliza-se o sistema RobWebCam (Álvares & Romariz, 1998).

4.1 Interface cliente/servidor

Foi desenvolvida uma Interface com o usuário onde são controladas todas as funções (Figura 5). Através dos menus pode-se acessar os arquivos correspondentes às funções da biblioteca ARAP. Foi criado um arquivo HTML para cada função que pode ser executada pelo Robô, que aparece na *tela de funções*. Há um *menu de opções* que permite ativar a realimentação gráfica. A *tela de status* permite que o estado do Robô apareça na tela (posição, modo de operação e configurações principais). A *tela de envio* (caixa de envio) apresenta o telegrama a ser enviado, podendo ser corrigido ou editado antes do envio. A tela de recebimento mostra o campo de dados do telegrama recebido do Robô. A figura 6 apresenta a tela principal do Sistema RobWebLink, onde podem ser visualizados nos frames superiores os menus onde o usuário seleciona as funções. Na segunda linha de *frames* as três janelas são: a janela de status, a janela de realimentação gráfica e a janela de função. Na terceira linha estão as janelas de envio e recebimento de mensagens.

Utilizaram-se os formulários eletrônicos do HTML para permitir o envio de parâmetros necessários em cada uma das diferentes funções. Cada função é tratada individualmente por um CGI, de forma modular.

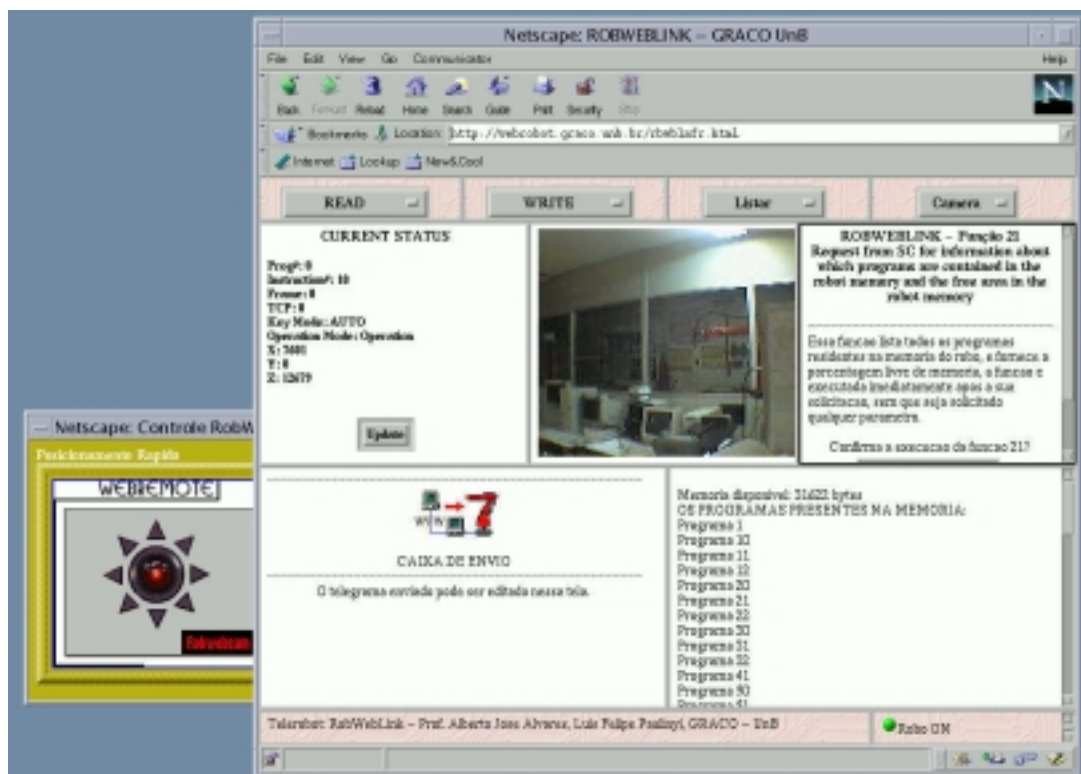


Figura 5. Sistema RobWebLink: interface com o usuário (<http://webrobot.graco.unb.br>).

4.2 Servidor e Controlador

O sistema foi desenvolvido em C++ da GNU para Linux. Desenvolveram-se os programas para serem executados via WWW/HTTP por programas CGI, que implementam o protocolo ADLP/10. Desta forma cada uma das funções tem o seu programa correspondente. Cada programa CGI possui módulos que realizam as seguintes tarefas: *Leitura dos dados enviados pelo formulário; Montagem do Telegrama a ser enviado; Inicialização da Porta Serial; Estabelecimento da comunicação com o Robô; Envio e recebimento de Telegramas; Verificação do telegrama recebido; Montagem da Homepage que será apresentada ao usuário com a resposta.*

4.3 Monitoramento - RobWebCam (<http://www.graco.unb.br/robwebcam>)

Foi utilizado o sistema RobWebCam (Álvares & Romariz, 1998) para monitoração visual da operação do sistema. O usuário pode acessar o menu de opções e movimentar a câmera através de um *joystick* virtual (figura 5). O ideal seria ter várias câmeras em várias posições diferentes para que o usuário escolhesse a melhor visão, detalhada e/ou global.

5. RESTRIÇÕES DO PROTOCOLO INTERNET (TCP/IP) E SEGURANÇA

Como a Internet apresenta limitação de largura de banda, aplicações em tempo real para captura de vídeo apresentam sérias restrições. Para vencer estas restrições é necessário utilizar compressão de dados e conexão de grande velocidade à Internet.

Outra limitação é o *delay* (atraso) inerente ao protocolo TCP, pois os pacotes enviados não necessariamente estão na mesma ordem dos pacotes recebidos pelo cliente, o que não é desejável para aplicações em tempo real.

Esta restrição pode ser resolvida adicionando algum grau de autonomia para a aplicação, no caso um Robô, de tal forma a contornar situações de emergência, bem como, situações perigosas. Este nível de autonomia é obtido localmente no Servidor WebRobot, que deve monitorar estas condições limites (Hirzinger *et al.*, 1997).

Apesar dos datagramas com informações de controle enviados para o Robô pelo cliente via Protocolo TCP/IP serem pequenos, da ordem de 2 a 3 Kbytes, o Protocolo TCP não garante aplicações em tempo real. Uma alternativa para o futuro é utilizar o Protocolo RTP (*Real-Time Transport Protocol*) para aplicações em tempo real (Melchioris, 1996). A partir do momento que o usuário execute um programa o sistema exige uma verificação de senhas.

6. CONCLUSÕES

O trabalho situa-se na área de teleoperação apresentando o desenvolvimento do sistema RobWebLink. O sistema apresentou bons resultados no que concernem as funções diretamente relacionadas ao controle do Robô através do seu protocolo de comunicação, por requerer arquivos de 1 a 2 Kbytes de tamanho para implementar estas funções. Entretanto, a realimentação através de vídeo exige maior largura de banda para aplicações em tempo real. Utilizou-se o sistema RobWebCam como servidor para captura de vídeo (WebCam). Uma opção seria a utilização de Realidade Virtual ao invés de se utilizar sinal de vídeo, através de VRML (Melchioris, 1996).

O sistema comprovou sua operacionalidade, conectando o cliente ao servidor WWW e este ao Robô, permitindo o envio e o recebimento de informações. É necessário levar em conta os aspectos referentes à segurança. Uma resposta do Robô ao servidor pode ser degradada em função de atrasos de comunicação entre o servidor e o cliente. Assim, os aspectos de tempo real têm grande importância de acordo com o nível de segurança exigido. A evolução do sistema é prevista para integração a uma Célula Flexível de Soldagem.

7. REFERÊNCIAS

Álvares, A. J. & Romariz, L. J., 1999, "TeleRobótica: Metodologia Para o Desenvolvimento de Sistemas Robóticos Teleoperados Via Internet", XV Congresso Brasileiro de Engenharia Mecânica, Águas de

- Lindóia, S.P., 22-26 de Novembro.
- Álvares, A. J. & Romariz, L. J., 1998, “Desenvolvimento de um Manipulador com Dois Graus de Liberdade Controlado Remotamente Via Internet”, V Congresso de Engenharia Mecânica Norte e Nordeste, Fortaleza, 27-30 de Outubro, pp. 529-536.
- Eckel, G. & Hare, C., 1995, “Building a Linux Internet Server”, New Riders Publishing, Indianapolis.
- Goodall, K., “Mars Pathfinder - Welcome to Mars, <http://mars.jpl.nasa.gov/default>.
- Hirzinger, G. et al, 1997, “Teleoperating Space Robots - Impact for the Design of Industrial Robots”, IEEE International Symposium on Industrial Electronics, Portugal, July 12-16, pp. 250-256.
- Melchior, C., 1996, “Sistemas Interpessoais de Videoconferência (mbone)”, <http://www.penta.ufrgs.br/~crisina/mbone/ti/indiceti.htm>.
- Monteiro F. et al, 1997, “Teleoperating a Mobile Robot - a Solution Based on Java Language”, IEEE International Symposium on Industrial Electronics, Portugal, July 12-16, pp. 263-267.
- Robotics, ABB., 1993, “Computer Link/ADLP-10/ARAP S3”, Reference Manual.
- Schilling, K., Roth, H. and Lieb, R., 1997, “Teleoperations of Rovers - From Mars to Education”, IEEE International Symposium on Industrial Electronics, Portugal, July 12-16, pp. 257-262.
- Sun, 1994, “Sun Video User’s Guide”.
- Taylor, K. & Trevelyan, J., 1995, “A Telerobot on the World Wide Web”, National Conference of the Australian Robot Association, Melbourne, July 5-7, <http://telerobot.mech.uwa.edu.au>.
- Wolf, H. & Froitzheim, K., 1997, “Webvideo - A Tool for WWW Based Teleoperation”, IEEE International Symposium on Industrial Electronics, Portugal, July 12-16, pp. 268-273.

DESENVOLVIMENTO DE UM ROBÔ MÓVEL AUTÔNOMO TELEOPERADO VIA INTERNET

Sérgio Roberto Gonsalves Tourino

Alberto José Álvares

Universidade de Brasília, Departamento de Engenharia Mecânica, Grupo de Automação e Controle, 70910-900, Brasília, DF., Brasil. E-mail: tourino@graco.unb.br.

Resumo

Este trabalho apresenta o desenvolvimento de um Robô Móvel Autônomo controlado remotamente via *Internet* utilizando o sistema operacional *Linux* como plataforma de controle e comunicação. A locomoção é realizada através de motores de passo, controlados por um *driver* de potência e programas em linguagem C. Sua estrutura foi fabricada em alumínio, consistindo em um chassi com três níveis: o primeiro contendo motores e alimentação elétrica (baterias ou alimentação externa). O segundo abrigando uma placa-mãe *Pentium* (responsável pelo controle do sistema) e uma placa de Rede *Ethernet*. O terceiro nível é constituído por um *link* de comunicação via rádio/*Internet* (Adaptador *Ethernet*) e uma câmera CCD (*Charge Coupled Device*). O controle via *Internet* é realizado através de uma interface Java/CGI (*Common Gateway Interface*) que envia os comandos do usuário para o sistema de navegação do robô (via rádio) tendo uma realimentação através de um sistema de captura de vídeo baseado em uma *WebCam*.

Palavras-chave: Robótica Móvel, Mecatrônica, Teleoperação, Internet, Telerobótica.

1. INTRODUÇÃO

A existência de ambientes insalubres ou impróprios à ação humana, como centrais nucleares, levou ao desenvolvimento de sistemas remotos de operação. Entre esses sistemas encontra-se grande aplicação na soldagem remota, onde tochas TIG (*Tungsten Inert Gas*) são utilizadas para a soldagem de tubulações em ambientes radioativos (Parmar, 1996). Com uma outra abordagem tecnológica, a utilização de robôs móveis para visualização e monitoramento de ambientes perigosos também alcança grande desenvolvimento. Recentemente, pesquisadores (Nehmzow, 1996) estudam o controle de robôs móveis através da utilização da *Internet* como veículo de transmissão de dados, com fins de pesquisa acadêmica.

Este artigo apresenta os resultados do desenvolvimento de um sistema robótico móvel e sua interface de teleoperação.

2. MODELAGEM DO SISTEMA

São descritos neste tópico o sistema de navegação desenvolvido para o robô, assim como a arquitetura de teleoperação utilizada no projeto baseada na metodologia de desenvolvimento

de sistemas robóticos teleoperados via Internet proposta por (Álvares, 1999).

2.1 Algoritmo de Navegação

A função do sistema de navegação de um robô móvel é a geração de trajetórias, permitindo que o mesmo seja capaz de realizar os objetivos definidos pelo usuário. Para isso o robô obtém dados externos do sistema de sensores e, por comparação com um mapa interno, o melhor trajeto é obtido, sendo então usado pelo robô para alcançar seus objetivos.

Diversas metodologias são usadas para a geração de trajetórias (Jones, 1999), destacando-se as redes neurais artificiais, a lógica *fuzzy* ou nebulosa, e o método dos campos potenciais (Benporad, 1996).

A analogia de campos potenciais define forças fictícias de campo atuando sobre o robô (Tourino, 1999). O movimento do robô é então direcionado como resposta a essas forças. Pretende-se, com esta metodologia, obter trajetórias capazes de desviar o robô de obstáculos conhecidos pelo sistema e alcançar um objetivo pré-definido pelo controlador ou pelo usuário. Foram definidas duas forças fictícias gerais que atuam sobre o robô:

- uma força atrativa, gerada pelo objetivo, denominada F_d ;
- forças repulsivas, geradas pelos obstáculos, denominada de F_o ;

Definidas as forças, a trajetória do robô é obtida seguindo-se a direção da força fictícia resultante em um deslocamento de comprimento d . Neste novo ponto é então calculada a nova resultante das forças e o robô sofre novo deslocamento, reiniciando o processo até que o objetivo seja alcançado.

Forças Atrativas

A força atrativa é modelada como um escoamento elementar do tipo sorvedouro, gerando uma força no sentido do robô para o destino.

Forças Repulsivas

As forças repulsivas (figura 1) são modeladas com base em dois escoamentos elementares: fontes, responsáveis pela repulsão, e vórtices, responsáveis pelo movimento do robô ao redor do obstáculo.

Força Resultante sobre o Robô

Para um ambiente com diversos obstáculos tem-se o efeito combinado das forças fictícias. Os componentes F_x e F_y da força são então calculados através das expressões:

$$\begin{aligned} F_x &= S.r_d \cdot \cos \theta_d + \sum_{i=1}^n \left(\frac{f \cdot \cos \theta_i}{r_i^2} - \frac{v \cdot \sin \theta_i}{r_i^2} \right) \\ F_y &= S.r_d \cdot \sin \theta_d + \sum_{i=1}^n \left(\frac{f \cdot \sin \theta_i}{r_i^2} + \frac{v \cdot \cos \theta_i}{r_i^2} \right) \end{aligned} \quad (1)$$

onde tem-se as seguintes variáveis:

S Intensidade atrativa

- f Intensidade repulsiva
- v Intensidade de vórtice
- r_d, r_i Distância ao destino / obstáculo
- θ_d, θ_i Ângulo em relação ao destino / obstáculo

A nova posição do robô é então obtida a partir da posição antiga, da direção de movimento e da distância deslocada.

A figura 2 mostra um resultado simulado obtido através do algoritmo, para um caso de obstáculo reconhecido por um sistema de sensores de toque. O retângulo representa o obstáculo real no ambiente, os pequenos círculos na superfície do mesmo representam os obstáculos virtuais, e os grandes círculos representam as posições ocupadas pelo robô durante o seu deslocamento.

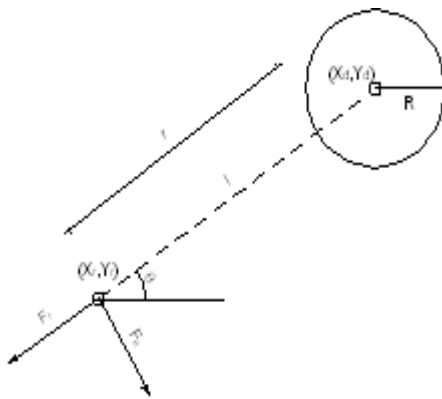


Figura 1. Modelo da força repulsiva.

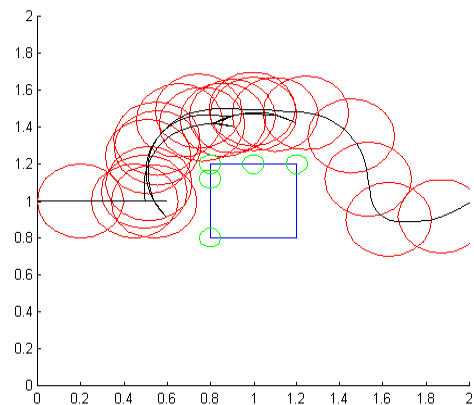


Figura 2. Trajetória obtida pelo algoritmo de navegação.

2.2 Arquitetura de Teleoperação

A arquitetura do sistema baseia-se no modelo cliente-servidor, que divide as atribuições do sistema em um servidor, representado pela página HTML (*Hiper Text Markup Language*) e programas CGI localizados no robô móvel; e o cliente, representado pelo *applet* Java, funcionando no *browser* do usuário. Entre o cliente e o servidor encontra-se a rede *Internet*, que permite a comunicação entre os módulos (figura 3).

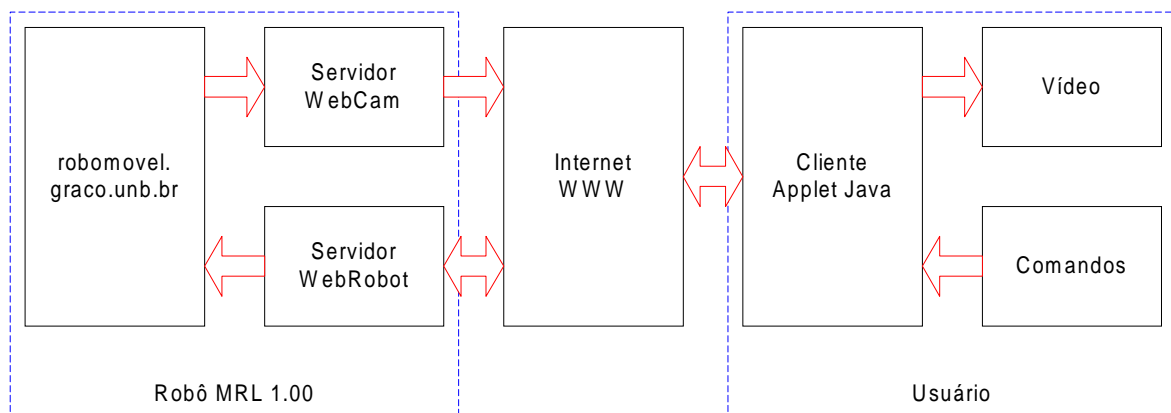


Figura 3. Arquitetura do sistema de teleoperação.

O servidor MRL é composto de dois módulos principais (Álvares, 1999):

- WebCam: visualização do ambiente remoto através de vídeo;
- WebRobot: controle do sistema teleoperado.

O *applet* Java, carregado no *browser* do usuário no instante que o mesmo acessa a página HTML do sistema (<http://robomovel.graco.unb.br>), é responsável pelo envio de comandos de movimentação do robô, através de requisições a programas CGI-BIN, desenvolvidos em linguagem C, localizados no servidor. O servidor, além de processar os comandos CGI, é responsável pela captura de vídeo *WebCam* (Connectix, 1986) e o envio das imagens para o *browser*. Tem-se, assim, uma interação entre o usuário e o ambiente em que o robô se encontra.

3. RESULTADOS OBTIDOS

3.1 Implementação Física

A estrutura desenvolvida visa a modularização (figura 4), sendo organizada em três níveis: na parte inferior (primeiro nível) encontram-se os sistemas de movimentação, sensorial e energia; no segundo nível tem-se o controlador (placa mãe de um *Pentium* de 75 Mhz); e no terceiro nível o sistema de comunicação e vídeo (figura 5a).

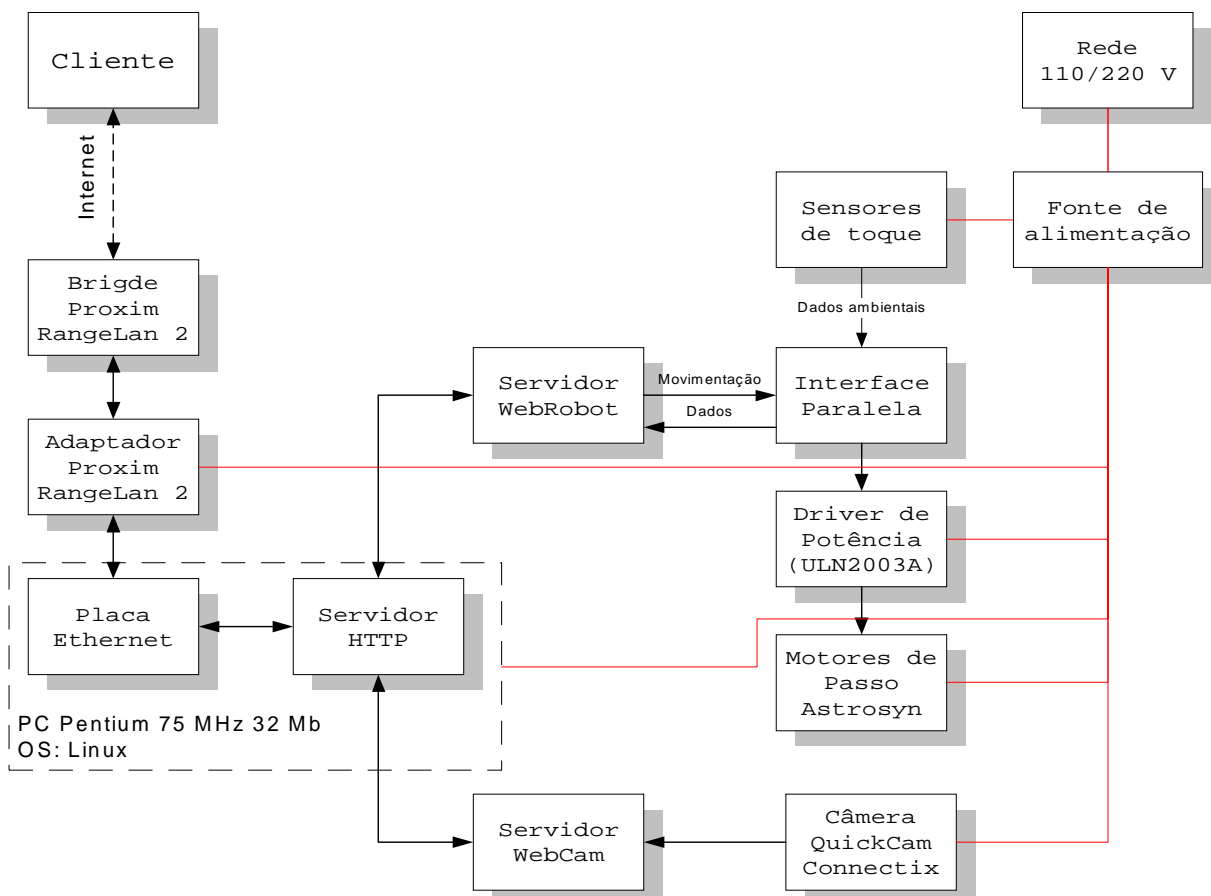


Figura 4. Módulos integrantes do sistema.

O sistema de locomoção é composto por um conjunto de motores de passo, que controlam os movimentos do robô através do deslocamento diferencial de duas rodas. Uma terceira roda de borracha é utilizada como apoio do conjunto. Os motores são alimentados

através de um *driver* de potência, baseado no *chip* ULN 2003A (Gadre, 1998) e no transistor TIP 120. Este *driver* é controlado através da interface paralela do módulo controlador.

O sistema de alimentação é representado por uma fonte de tensão de computador ligada à rede elétrica, colocada na parte posterior do nível inferior. Uma implementação futura incluirá a utilização de baterias recarregáveis para permitir uma maior autonomia do sistema. Ainda no primeiro nível estão presentes duas “antenas” ou *whiskers* que funcionam como sensores de toque para o reconhecimento de obstáculos pelo robô. São utilizados sensores binários, tipo liga-desliga, conectados às entradas da interface paralela do controlador. No futuro serão utilizados sensores para medição de distâncias ultrasônico e infravermelho com saída TTL (*Transistor-Transistor Logic*) conectados diretamente à interface paralela.

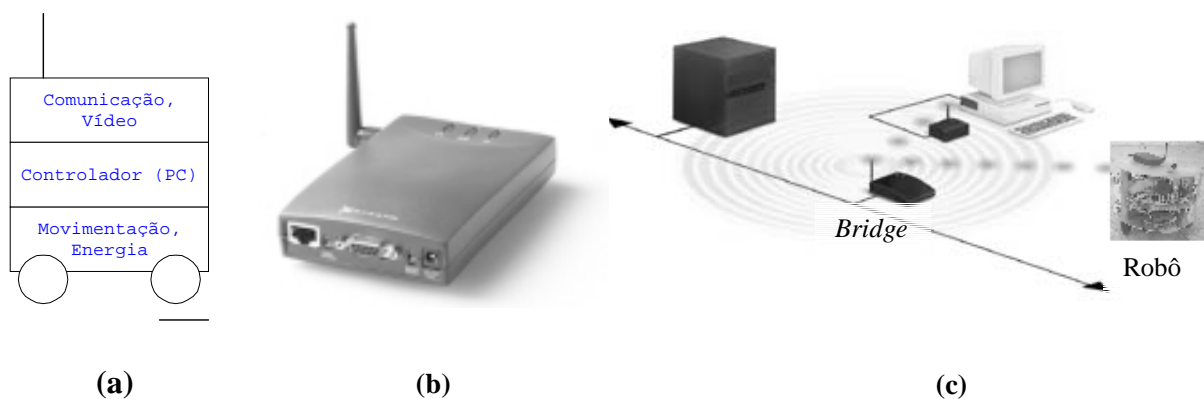


Figura 5. (a)Arquitetura física do robô desenvolvido ; (b) Adaptador Ethernet utilizado no robô móvel (RangeLan 2 7921, Proxim). (c) Arquitetura de comunicação utilizada no robô móvel.

O controlador é representado por um computador pessoal tipo PC (*Personal Computer*) *Pentium*, funcionando sob o sistema operacional *Linux* e programado através da linguagem C.

O sistema de comunicação utilizado (terceiro nível) baseia-se em um rádio adaptador *Ethernet* da Proxim (figura 5b), que conecta-se a um *bridge* (RangeLan2 7510, Proxim) presente no laboratório, interligando o sistema à rede *Internet*. A utilização de um sistema de rádio para a comunicação com o robô deu-se devido aos seguintes fatores: mobilidade, ou seja, um sistema robótico móvel não deve possuir cabos ligando-o a uma base fixa; e sistemas a rádio são adequados para pequenas distâncias (300 m), entre o adaptador e o *bridge*, ou para maiores distâncias utilizando antenas mais potentes e vários repetidores/*bridge* em um ambiente de células, similar à telefonia celular.

A figura 5c apresenta a infraestrutura de comunicação utilizada no laboratório do GRACO, incluindo o servidor, o robô móvel desenvolvido e o robô Nomad XR4000 (Nomadic, 1999), também integrante do ambiente do laboratório de robótica móvel. Observa-se que o *bridge* é a parte responsável pela interconexão entre os diversos adaptadores móveis à rede local do laboratório e a *Internet*.

A captura de imagens é realizada através de uma câmera tipo *WebCam* (Álvares, 1998), que envia suas imagens através de um *driver* para páginas HTML (Hiper Text Markup Language) ou aplicativos Java. A compressão das imagens no formato JPEG é realizada via software da câmera, que envia as imagens diretamente para o *browser*, onde é feita automaticamente a animação das imagens recebidas.

A figura 6 apresenta fotos do robô desenvolvido, onde podem ser observados os módulos descritos anteriormente.

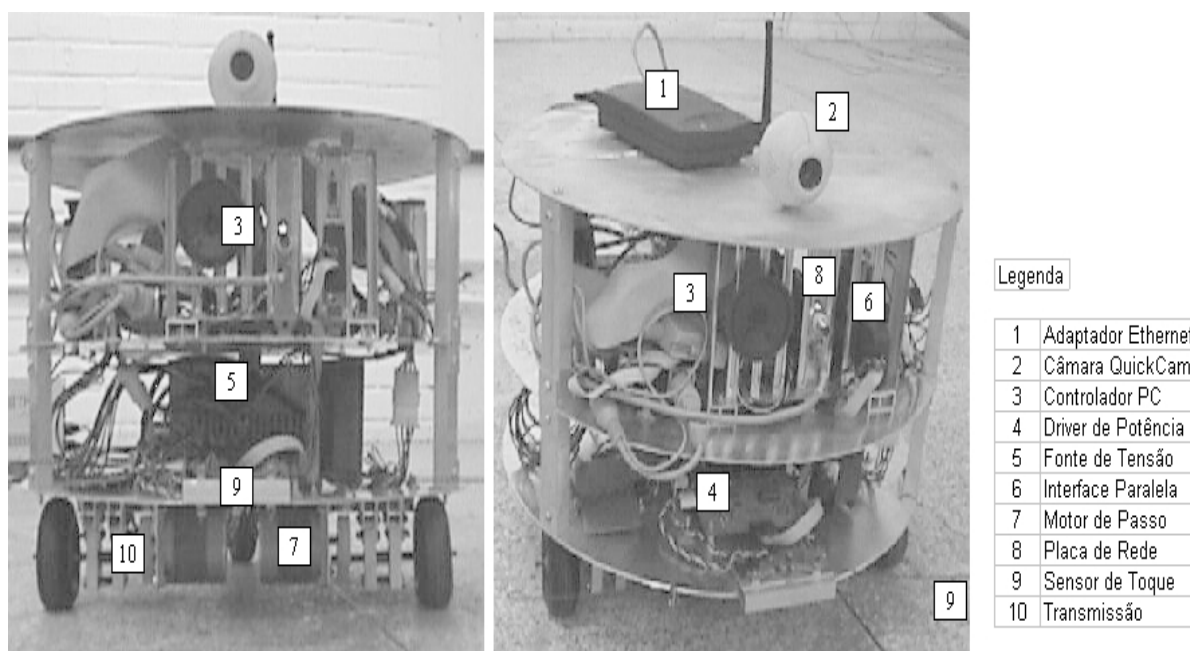


Figura 6. Robô móvel desenvolvido no GRACO (MRL 1.00).

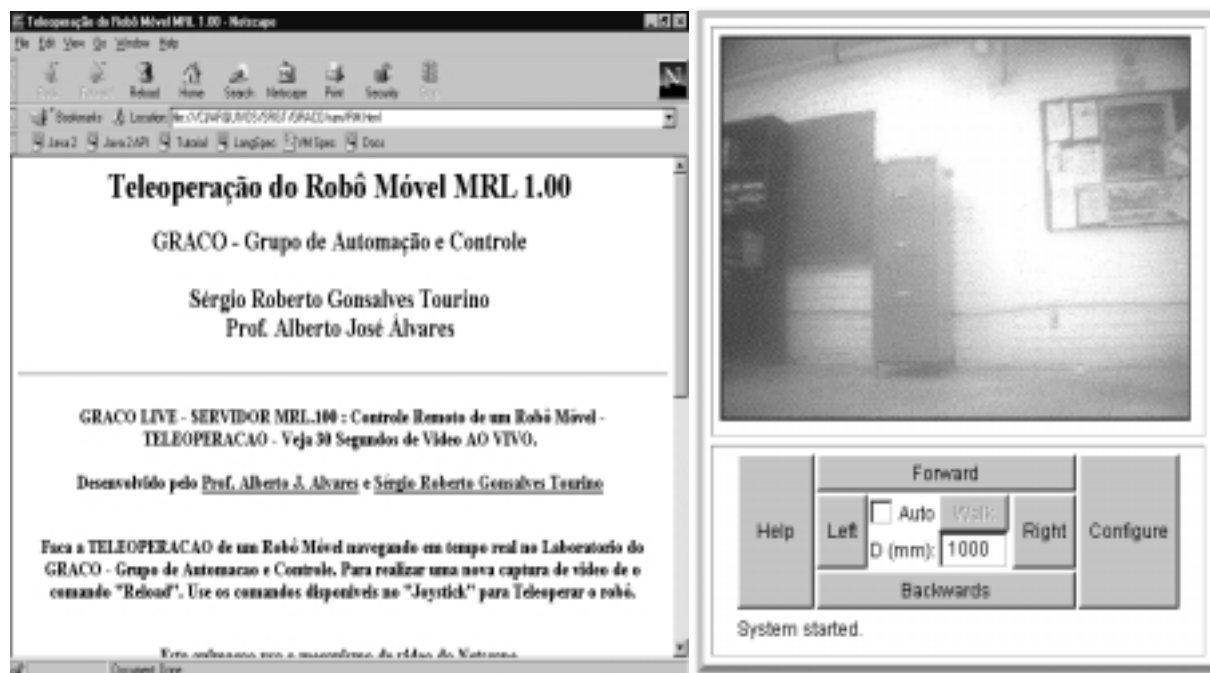
3.2 GUI (Graphical User Interface) Para Teleoperação

A interface de teleoperação (GUI) desenvolvida é baseada na linguagem de programação Java, permitindo uma boa interação usuário-robô. São utilizados quatro botões que controlam a translação e a rotação do robô, no plano XY, e um quinto botão responsável pela implementação do algoritmo de navegação autônoma. A figura 7a apresenta a página de entrada do sistema na Internet.

Na figura 7b pode ser vista a interface desenvolvida. Na parte superior do console encontra-se a imagem recebida pela câmera localizada no robô, o que possibilita a visualização do ambiente remoto.

A parte inferior do console é responsável pelo controle posicional do robô. Destaca-se nesta parte o botão *Walk*, responsável pela navegação autônoma do sistema. Assim, se o usuário deseja que o robô caminhe 3 metros adiante, evitando possíveis obstáculos, deve ajustar a distância D para 3000 mm, ativar o sistema autônomo através do *checkbox* Auto, e enviar o comando através do botão *Walk*. O robô tentará então, utilizando seus sensores de toque, navegar de forma autônoma até o ponto final desejado, a 3 metros a frente.

Ainda na parte inferior são localizados dois botões, *Help* e *Configure*, responsáveis, respectivamente, pela abertura de uma janela de ajuda ao usuário, contendo instruções sobre o funcionamento do sistema; e pela configuração de aspectos relativos à recepção da imagem, como luminosidade e qualidade, permitindo assim uma melhor interação usuário-ambiente remoto.



(a)

(b)

Figura 7. (a) Página HTML de teleoperação; (b) *Applet* Java desenvolvido para a teleoperação.

4. ANÁLISE DOS RESULTADOS

A aplicação de sistemas robóticos de teleoperação implica a utilização de um sistema em tempo real. Entretanto, o ambiente WWW e o protocolo TCP/IP não são adequados para aplicações em tempo real, devido à sua limitação de largura de banda (para transmissão de vídeo) e o *delay* (atraso) do envio dos pacotes TCP entre servidor e cliente. Parte desses problemas podem ser reduzidos através do uso de imagens JPG (compactadas) e a implementação de um certo grau de autonomia no robô (sensoriamento e navegação autônoma), inseridos no sistema desenvolvido.

A utilização do *Linux* como sistema operacional foi guiada pela capacidade deste de prover serviços gerais (como conexão Internet, HTTP) assim como permitir, através da linguagem C, controle sobre dispositivos como sensores e motores. A sua robustez e confiabilidade, quando comparados a outros sistemas operacionais, tornam-no adequado a aplicações em robótica.

A necessidade de aplicação em tempo real, internamente ao robô, para gerenciar simultaneamente captura de vídeo e navegação, foi implementada através da definição de prioridades para os processos: assim, o processo responsável pela navegação é disparado com maior prioridade que os demais processos, garantindo a segurança do sistema, embora reduzindo a transmissão das imagens capturadas. Uma outra abordagem possível para o problema seria a utilização da extensão de tempo real para o Linux (Barabanov, 1996), o que permitiria uma melhor aplicação do sistema ao controle do robô.

O algoritmo de navegação implementado foi simulado através do *software* MatLab, sendo realizados testes em diversas configurações de obstáculos, com a simulação de um sistema de sensores de toque. Verificou-se uma boa estabilidade do método, embora em alguns casos o sistema demore a convergir. A futura substituição dos sensores de toque para sensores do tipo *sonar* no robô é compatível com o algoritmo implementado, já tendo sido inclusive realizados testes do mesmo nessa nova configuração.

5. CONCLUSÕES

Embora composto por diversos subsistemas interconectados, o robô móvel desenvolvido mostrou-se adequado para a proposta de desenvolvimento de um sistema simples e barato de robótica móvel voltado para aplicações didáticas e de pesquisa na área de Teleoperação. A sua tecnologia pode ser aplicada para utilizações diversas, como inspeção de sistemas industriais, ou mesmo aplicações em televigilância. A utilização da *Internet* como meio de transmissão de dados entre a GUI (em Java/html) e o robô mostra-se adequada, e permite que o sistema seja amplamente disponível para usuários, sem limitações geográficas. Os tempos de resposta do sistema estão associados à largura de banda do acesso a *Internet* tendo melhor desempenho para conexões mais rápidas. A evolução do sistema está prevista e será utilizado um *Kernel Linux* em Tempo Real (Barabanov, 1996) para permitir um controle mais adequado dos acionamentos.

6. AGRADECIMENTOS

Os autores agradecem ao apoio financeiro dado pela Fundação de Amparo à Pesquisa do Distrito Federal (FAP-DF), que permitiu que o trabalho fosse realizado, e aos alunos do curso de Engenharia Mecatrônica Thiago Belardi e Gustavo Fleury, pelo auxílio prestado durante o desenvolvimento do projeto.

7. REFERÊNCIAS

- Álvares, A. J. & Romariz, L. J., 1999, “TeleRobótica: Metodologia Para o Desenvolvimento de Sistemas Robóticos Teleoperados Via Internet”, XV Congresso Brasileiro de Engenharia Mecânica, Águas de Lindóia, S.P., 22-26 de Novembro.
- Álvares, A. J. & Romariz, L. J., 1998, “Desenvolvimento de um Manipulador com Dois Graus de Liberdade Controlado Remotamente Via Internet”, V Congresso de Engenharia Mecânica Norte e Nordeste, Fortaleza, 27-30 de Outubro, pp. 529-536.
- Barabanov, M., 1996, “Real-Time Linux” Disponível na Internet via WWW. URL: <http://www.rtlinux.org>
- Benporad, A. De Luca, A. Oriolo, G. 1996. “Local Incremental Planning for a Car-Like Robot Navigating among Obstacles”, Proceedings of the 1996 IEEE, Mineapolis.
- Connectix, 1996, “QuickCam Color – Guia do Usuário”.
- Everett, H. R., 1995, “Sensors for Mobile Robots, Theory and Application”, A. K. Peters, Massachusetts. USA.
- Gadre, D. 1998. “Programming the Parallel Port: Interfacing The PC for Data Acquisition and Process Control”, R&D Publishing.
- Jones, J. L., Seiger, B. A., Flynn, A. M. 1999. “Mobile Robots: Inspiration to Implementation.” A K Peters. Massachusetts. USA.
- Nehmzow, U. Bühlmeier, A. Dürer, H. 1996. “Remote Control of Mobile Robot via Internet”, Manchester.
- Nomadic Technologies, 1999, “XR4000 Mobile Robot”, Disponível na Internet via WWW, URL: <http://www.robots.com/products.htm>.
- Parmar, R. S., 1995, “Welding Processes and Technology”, Khanna Publishers, Delhi.
- Tourino, S. R. G. Álvares, A. J. 1999. “Algoritmo de Navegação para Robôs Autônomos Móveis”, VI Congresso Nacional de Estudantes de Engenharia Mecânica, Brasília.
- Zhai, S. Milgram, P., 1991, “A Telerobotic Virtual Control System”, Proc. SPIE. Boston.

DESENVOLVIMENTO DE UM ROBÔ MÓVEL AUTÔNOMO TELEOPERADO VIA INTERNET

Sérgio Roberto Gonsalves Tourino

Alberto José Álvares

Universidade de Brasília, Departamento de Engenharia Mecânica, Grupo de Automação e Controle, 70910-900, Brasília, DF., Brasil. E-mail: tourino@graco.unb.br.

Resumo

Este trabalho apresenta o desenvolvimento de um Robô Móvel Autônomo controlado remotamente via *Internet* utilizando o sistema operacional *Linux* como plataforma de controle e comunicação. A locomoção é realizada através de motores de passo, controlados por um *driver* de potência e programas em linguagem C. Sua estrutura foi fabricada em alumínio, consistindo em um chassi com três níveis: o primeiro contendo motores e alimentação elétrica (baterias ou alimentação externa). O segundo abrigando uma placa-mãe *Pentium* (responsável pelo controle do sistema) e uma placa de Rede *Ethernet*. O terceiro nível é constituído por um *link* de comunicação via rádio/*Internet* (Adaptador *Ethernet*) e uma câmera CCD (*Charge Coupled Device*). O controle via *Internet* é realizado através de uma interface Java/CGI (*Common Gateway Interface*) que envia os comandos do usuário para o sistema de navegação do robô (via rádio) tendo uma realimentação através de um sistema de captura de vídeo baseado em uma *WebCam*.

Palavras-chave: Robótica Móvel, Mecatrônica, Teleoperação, Internet, Telerobótica.

1. INTRODUÇÃO

A existência de ambientes insalubres ou impróprios à ação humana, como centrais nucleares, levou ao desenvolvimento de sistemas remotos de operação. Entre esses sistemas encontra-se grande aplicação na soldagem remota, onde tochas TIG (*Tungsten Inert Gas*) são utilizadas para a soldagem de tubulações em ambientes radioativos (Parmar, 1996). Com uma outra abordagem tecnológica, a utilização de robôs móveis para visualização e monitoramento de ambientes perigosos também alcança grande desenvolvimento. Recentemente, pesquisadores (Nehmzow, 1996) estudam o controle de robôs móveis através da utilização da *Internet* como veículo de transmissão de dados, com fins de pesquisa acadêmica.

Este artigo apresenta os resultados do desenvolvimento de um sistema robótico móvel e sua interface de teleoperação.

2. MODELAGEM DO SISTEMA

São descritos neste tópico o sistema de navegação desenvolvido para o robô, assim como a arquitetura de teleoperação utilizada no projeto baseada na metodologia de desenvolvimento

de sistemas robóticos teleoperados via Internet proposta por (Álvares, 1999).

2.1 Algoritmo de Navegação

A função do sistema de navegação de um robô móvel é a geração de trajetórias, permitindo que o mesmo seja capaz de realizar os objetivos definidos pelo usuário. Para isso o robô obtém dados externos do sistema de sensores e, por comparação com um mapa interno, o melhor trajeto é obtido, sendo então usado pelo robô para alcançar seus objetivos.

Diversas metodologias são usadas para a geração de trajetórias (Jones, 1999), destacando-se as redes neurais artificiais, a lógica *fuzzy* ou nebulosa, e o método dos campos potenciais (Benporad, 1996).

A analogia de campos potenciais define forças fictícias de campo atuando sobre o robô (Tourino, 1999). O movimento do robô é então direcionado como resposta a essas forças. Pretende-se, com esta metodologia, obter trajetórias capazes de desviar o robô de obstáculos conhecidos pelo sistema e alcançar um objetivo pré-definido pelo controlador ou pelo usuário. Foram definidas duas forças fictícias gerais que atuam sobre o robô:

- uma força atrativa, gerada pelo objetivo, denominada F_d ;
- forças repulsivas, geradas pelos obstáculos, denominada de F_o ;

Definidas as forças, a trajetória do robô é obtida seguindo-se a direção da força fictícia resultante em um deslocamento de comprimento d . Neste novo ponto é então calculada a nova resultante das forças e o robô sofre novo deslocamento, reiniciando o processo até que o objetivo seja alcançado.

Forças Atrativas

A força atrativa é modelada como um escoamento elementar do tipo sorvedouro, gerando uma força no sentido do robô para o destino.

Forças Repulsivas

As forças repulsivas (figura 1) são modeladas com base em dois escoamentos elementares: fontes, responsáveis pela repulsão, e vórtices, responsáveis pelo movimento do robô ao redor do obstáculo.

Força Resultante sobre o Robô

Para um ambiente com diversos obstáculos tem-se o efeito combinado das forças fictícias. Os componentes F_x e F_y da força são então calculados através das expressões:

$$\begin{aligned} F_x &= S.r_d \cdot \cos \theta_d + \sum_{i=1}^n \left(\frac{f \cdot \cos \theta_i}{r_i^2} - \frac{v \cdot \sin \theta_i}{r_i^2} \right) \\ F_y &= S.r_d \cdot \sin \theta_d + \sum_{i=1}^n \left(\frac{f \cdot \sin \theta_i}{r_i^2} + \frac{v \cdot \cos \theta_i}{r_i^2} \right) \end{aligned} \quad (1)$$

onde tem-se as seguintes variáveis:

S Intensidade atrativa

- f Intensidade repulsiva
- v Intensidade de vórtice
- r_d, r_i Distância ao destino / obstáculo
- θ_d, θ_i Ângulo em relação ao destino / obstáculo

A nova posição do robô é então obtida a partir da posição antiga, da direção de movimento e da distância deslocada.

A figura 2 mostra um resultado simulado obtido através do algoritmo, para um caso de obstáculo reconhecido por um sistema de sensores de toque. O retângulo representa o obstáculo real no ambiente, os pequenos círculos na superfície do mesmo representam os obstáculos virtuais, e os grandes círculos representam as posições ocupadas pelo robô durante o seu deslocamento.

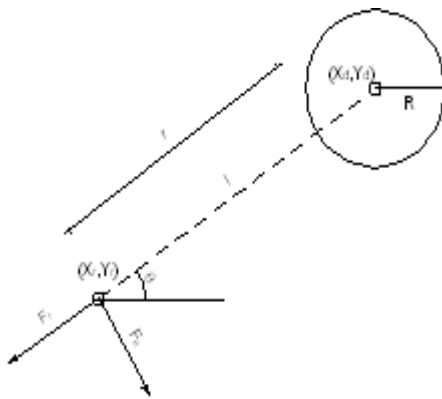


Figura 1. Modelo da força repulsiva.

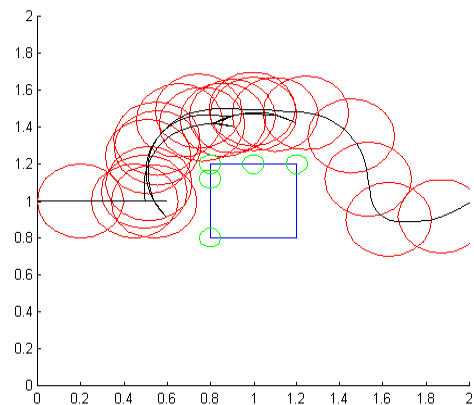


Figura 2. Trajetória obtida pelo algoritmo de navegação.

2.2 Arquitetura de Teleoperação

A arquitetura do sistema baseia-se no modelo cliente-servidor, que divide as atribuições do sistema em um servidor, representado pela página HTML (*Hiper Text Markup Language*) e programas CGI localizados no robô móvel; e o cliente, representado pelo *applet* Java, funcionando no *browser* do usuário. Entre o cliente e o servidor encontra-se a rede *Internet*, que permite a comunicação entre os módulos (figura 3).

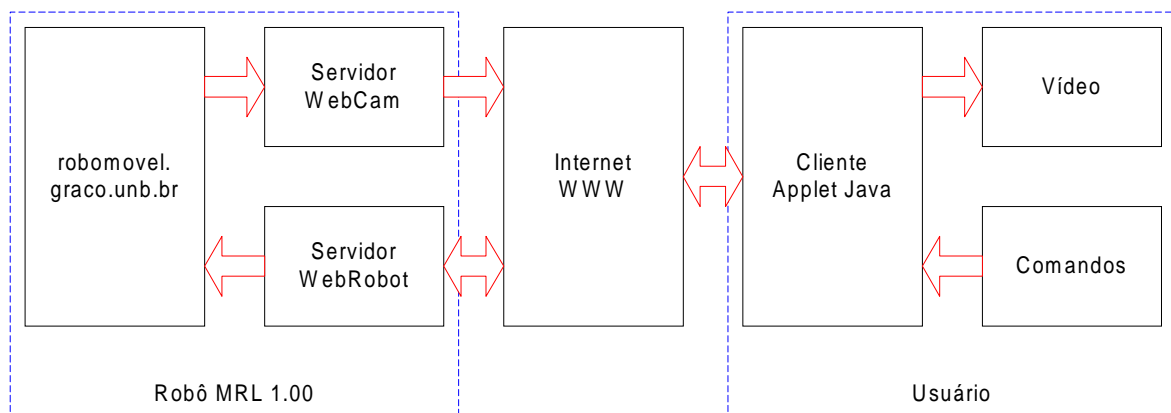


Figura 3. Arquitetura do sistema de teleoperação.

O servidor MRL é composto de dois módulos principais (Álvares, 1999):

- WebCam: visualização do ambiente remoto através de vídeo;
- WebRobot: controle do sistema teleoperado.

O *applet* Java, carregado no *browser* do usuário no instante que o mesmo acessa a página HTML do sistema (<http://robomovel.graco.unb.br>), é responsável pelo envio de comandos de movimentação do robô, através de requisições a programas CGI-BIN, desenvolvidos em linguagem C, localizados no servidor. O servidor, além de processar os comandos CGI, é responsável pela captura de vídeo *WebCam* (Connectix, 1986) e o envio das imagens para o *browser*. Tem-se, assim, uma interação entre o usuário e o ambiente em que o robô se encontra.

3. RESULTADOS OBTIDOS

3.1 Implementação Física

A estrutura desenvolvida visa a modularização (figura 4), sendo organizada em três níveis: na parte inferior (primeiro nível) encontram-se os sistemas de movimentação, sensorial e energia; no segundo nível tem-se o controlador (placa mãe de um *Pentium* de 75 Mhz); e no terceiro nível o sistema de comunicação e vídeo (figura 5a).

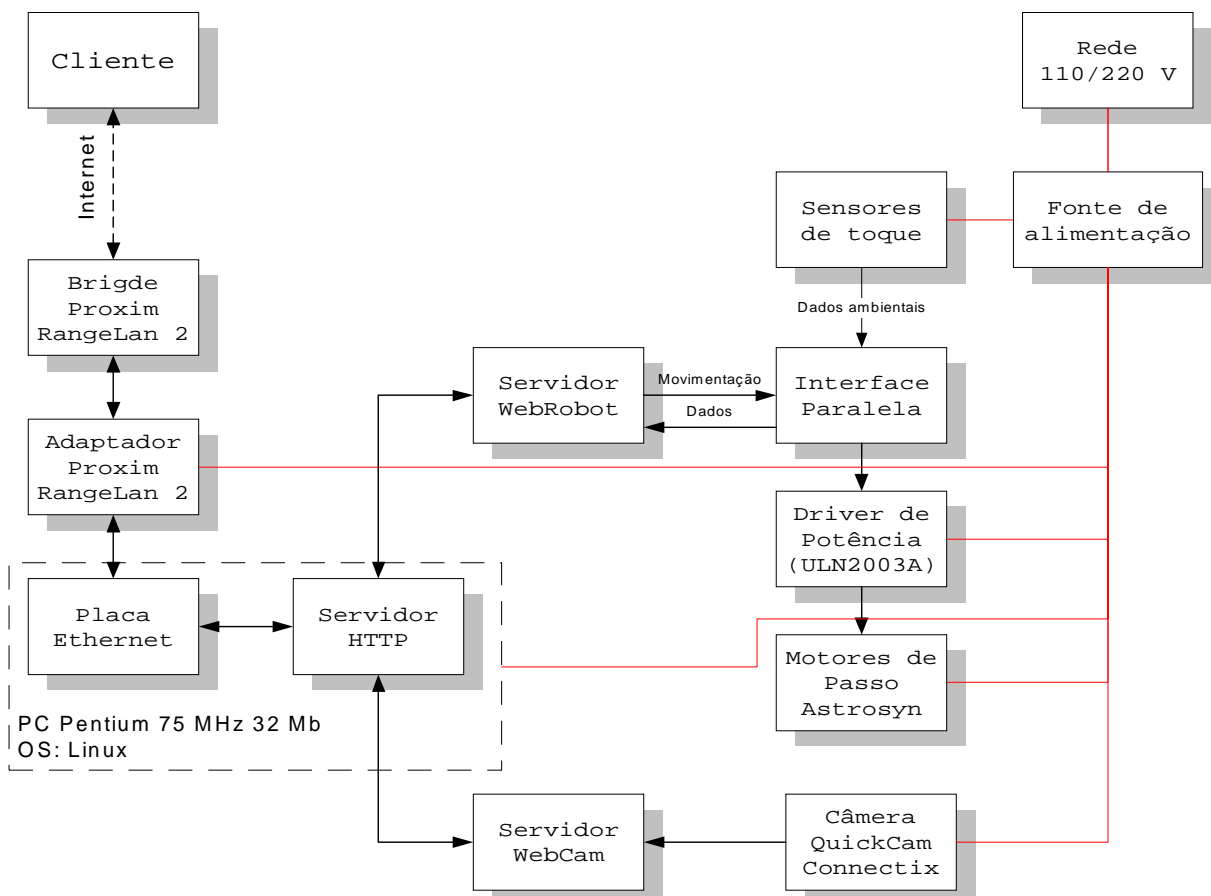


Figura 4. Módulos integrantes do sistema.

O sistema de locomoção é composto por um conjunto de motores de passo, que controlam os movimentos do robô através do deslocamento diferencial de duas rodas. Uma terceira roda de borracha é utilizada como apoio do conjunto. Os motores são alimentados

através de um *driver* de potência, baseado no *chip* ULN 2003A (Gadre, 1998) e no transistor TIP 120. Este *driver* é controlado através da interface paralela do módulo controlador.

O sistema de alimentação é representado por uma fonte de tensão de computador ligada à rede elétrica, colocada na parte posterior do nível inferior. Uma implementação futura incluirá a utilização de baterias recarregáveis para permitir uma maior autonomia do sistema. Ainda no primeiro nível estão presentes duas “antenas” ou *whiskers* que funcionam como sensores de toque para o reconhecimento de obstáculos pelo robô. São utilizados sensores binários, tipo liga-desliga, conectados às entradas da interface paralela do controlador. No futuro serão utilizados sensores para medição de distâncias ultrasônico e infravermelho com saída TTL (*Transistor-Transistor Logic*) conectados diretamente à interface paralela.

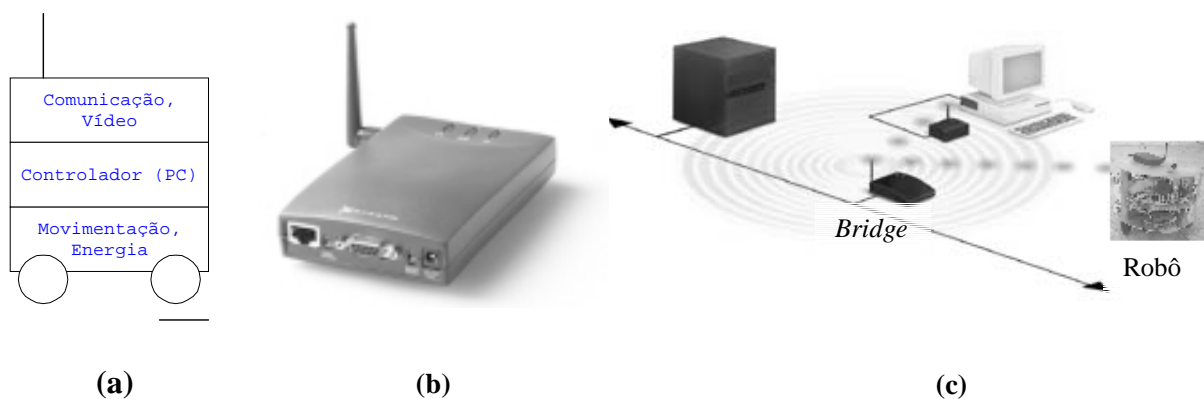


Figura 5. (a)Arquitetura física do robô desenvolvido ; (b) Adaptador Ethernet utilizado no robô móvel (RangeLan 2 7921, Proxim). (c) Arquitetura de comunicação utilizada no robô móvel.

O controlador é representado por um computador pessoal tipo PC (*Personal Computer*) *Pentium*, funcionando sob o sistema operacional *Linux* e programado através da linguagem C.

O sistema de comunicação utilizado (terceiro nível) baseia-se em um rádio adaptador *Ethernet* da Proxim (figura 5b), que conecta-se a um *bridge* (RangeLan2 7510, Proxim) presente no laboratório, interligando o sistema à rede *Internet*. A utilização de um sistema de rádio para a comunicação com o robô deu-se devido aos seguintes fatores: mobilidade, ou seja, um sistema robótico móvel não deve possuir cabos ligando-o a uma base fixa; e sistemas a rádio são adequados para pequenas distâncias (300 m), entre o adaptador e o *bridge*, ou para maiores distâncias utilizando antenas mais potentes e vários repetidores/*bridge* em um ambiente de células, similar à telefonia celular.

A figura 5c apresenta a infraestrutura de comunicação utilizada no laboratório do GRACO, incluindo o servidor, o robô móvel desenvolvido e o robô Nomad XR4000 (Nomadic, 1999), também integrante do ambiente do laboratório de robótica móvel. Observa-se que o *bridge* é a parte responsável pela interconexão entre os diversos adaptadores móveis à rede local do laboratório e a *Internet*.

A captura de imagens é realizada através de uma câmera tipo *WebCam* (Álvares, 1998), que envia suas imagens através de um *driver* para páginas HTML (Hiper Text Markup Language) ou aplicativos Java. A compressão das imagens no formato JPEG é realizada via software da câmera, que envia as imagens diretamente para o *browser*, onde é feita automaticamente a animação das imagens recebidas.

A figura 6 apresenta fotos do robô desenvolvido, onde podem ser observados os módulos descritos anteriormente.

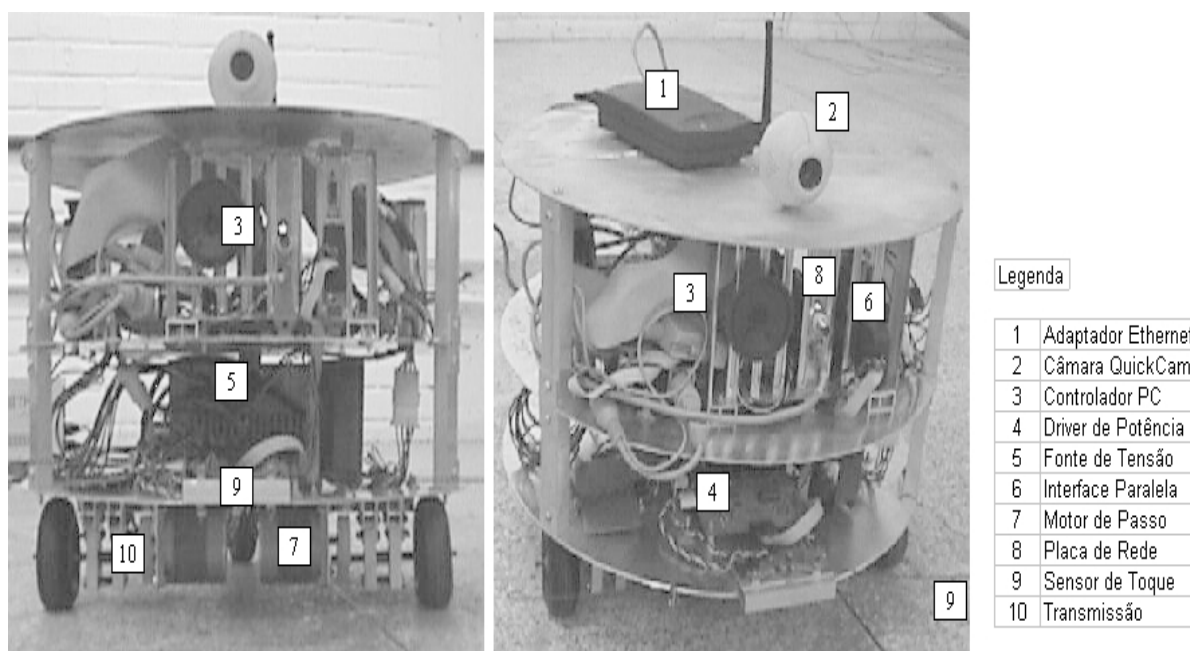


Figura 6. Robô móvel desenvolvido no GRACO (MRL 1.00).

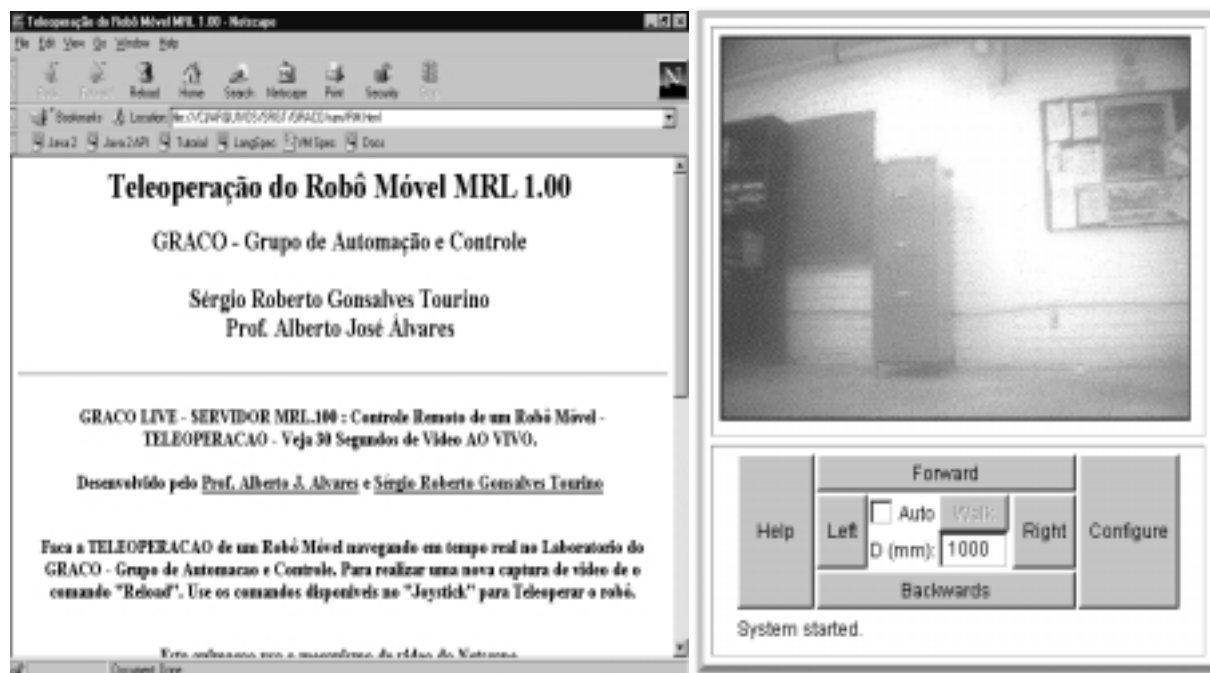
3.2 GUI (Graphical User Interface) Para Teleoperação

A interface de teleoperação (GUI) desenvolvida é baseada na linguagem de programação Java, permitindo uma boa interação usuário-robô. São utilizados quatro botões que controlam a translação e a rotação do robô, no plano XY, e um quinto botão responsável pela implementação do algoritmo de navegação autônoma. A figura 7a apresenta a página de entrada do sistema na Internet.

Na figura 7b pode ser vista a interface desenvolvida. Na parte superior do console encontra-se a imagem recebida pela câmera localizada no robô, o que possibilita a visualização do ambiente remoto.

A parte inferior do console é responsável pelo controle posicional do robô. Destaca-se nesta parte o botão *Walk*, responsável pela navegação autônoma do sistema. Assim, se o usuário deseja que o robô caminhe 3 metros adiante, evitando possíveis obstáculos, deve ajustar a distância D para 3000 mm, ativar o sistema autônomo através do *checkbox* Auto, e enviar o comando através do botão *Walk*. O robô tentará então, utilizando seus sensores de toque, navegar de forma autônoma até o ponto final desejado, a 3 metros a frente.

Ainda na parte inferior são localizados dois botões, *Help* e *Configure*, responsáveis, respectivamente, pela abertura de uma janela de ajuda ao usuário, contendo instruções sobre o funcionamento do sistema; e pela configuração de aspectos relativos à recepção da imagem, como luminosidade e qualidade, permitindo assim uma melhor interação usuário-ambiente remoto.



(a)

(b)

Figura 7. (a) Página HTML de teleoperação; (b) *Applet* Java desenvolvido para a teleoperação.

4. ANÁLISE DOS RESULTADOS

A aplicação de sistemas robóticos de teleoperação implica a utilização de um sistema em tempo real. Entretanto, o ambiente WWW e o protocolo TCP/IP não são adequados para aplicações em tempo real, devido à sua limitação de largura de banda (para transmissão de vídeo) e o *delay* (atraso) do envio dos pacotes TCP entre servidor e cliente. Parte desses problemas podem ser reduzidos através do uso de imagens JPG (compactadas) e a implementação de um certo grau de autonomia no robô (sensoriamento e navegação autônoma), inseridos no sistema desenvolvido.

A utilização do *Linux* como sistema operacional foi guiada pela capacidade deste de prover serviços gerais (como conexão Internet, HTTP) assim como permitir, através da linguagem C, controle sobre dispositivos como sensores e motores. A sua robustez e confiabilidade, quando comparados a outros sistemas operacionais, tornam-no adequado a aplicações em robótica.

A necessidade de aplicação em tempo real, internamente ao robô, para gerenciar simultaneamente captura de vídeo e navegação, foi implementada através da definição de prioridades para os processos: assim, o processo responsável pela navegação é disparado com maior prioridade que os demais processos, garantindo a segurança do sistema, embora reduzindo a transmissão das imagens capturadas. Uma outra abordagem possível para o problema seria a utilização da extensão de tempo real para o Linux (Barabanov, 1996), o que permitiria uma melhor aplicação do sistema ao controle do robô.

O algoritmo de navegação implementado foi simulado através do *software* MatLab, sendo realizados testes em diversas configurações de obstáculos, com a simulação de um sistema de sensores de toque. Verificou-se uma boa estabilidade do método, embora em alguns casos o sistema demore a convergir. A futura substituição dos sensores de toque para sensores do tipo *sonar* no robô é compatível com o algoritmo implementado, já tendo sido inclusive realizados testes do mesmo nessa nova configuração.

5. CONCLUSÕES

Embora composto por diversos subsistemas interconectados, o robô móvel desenvolvido mostrou-se adequado para a proposta de desenvolvimento de um sistema simples e barato de robótica móvel voltado para aplicações didáticas e de pesquisa na área de Teleoperação. A sua tecnologia pode ser aplicada para utilizações diversas, como inspeção de sistemas industriais, ou mesmo aplicações em televigilância. A utilização da *Internet* como meio de transmissão de dados entre a GUI (em Java/html) e o robô mostra-se adequada, e permite que o sistema seja amplamente disponível para usuários, sem limitações geográficas. Os tempos de resposta do sistema estão associados à largura de banda do acesso a *Internet* tendo melhor desempenho para conexões mais rápidas. A evolução do sistema está prevista e será utilizado um *Kernel Linux* em Tempo Real (Barabanov, 1996) para permitir um controle mais adequado dos acionamentos.

6. AGRADECIMENTOS

Os autores agradecem ao apoio financeiro dado pela Fundação de Amparo à Pesquisa do Distrito Federal (FAP-DF), que permitiu que o trabalho fosse realizado, e aos alunos do curso de Engenharia Mecatrônica Thiago Belardi e Gustavo Fleury, pelo auxílio prestado durante o desenvolvimento do projeto.

7. REFERÊNCIAS

- Álvares, A. J. & Romariz, L. J., 1999, “TeleRobótica: Metodologia Para o Desenvolvimento de Sistemas Robóticos Teleoperados Via Internet”, XV Congresso Brasileiro de Engenharia Mecânica, Águas de Lindóia, S.P., 22-26 de Novembro.
- Álvares, A. J. & Romariz, L. J., 1998, “Desenvolvimento de um Manipulador com Dois Graus de Liberdade Controlado Remotamente Via Internet”, V Congresso de Engenharia Mecânica Norte e Nordeste, Fortaleza, 27-30 de Outubro, pp. 529-536.
- Barabanov, M., 1996, “Real-Time Linux” Disponível na Internet via WWW. URL: <http://www.rtlinux.org>
- Benporad, A. De Luca, A. Oriolo, G. 1996. “Local Incremental Planning for a Car-Like Robot Navigating among Obstacles”, Proceedings of the 1996 IEEE, Mineapolis.
- Connectix, 1996, “QuickCam Color – Guia do Usuário”.
- Everett, H. R., 1995, “Sensors for Mobile Robots, Theory and Application”, A. K. Peters, Massachusetts. USA.
- Gadre, D. 1998. “Programming the Parallel Port: Interfacing The PC for Data Acquisition and Process Control”, R&D Publishing.
- Jones, J. L., Seiger, B. A., Flynn, A. M. 1999. “Mobile Robots: Inspiration to Implementation.” A K Peters. Massachusetts. USA.
- Nehmzow, U. Bühlmeier, A. Dürer, H. 1996. “Remote Control of Mobile Robot via Internet”, Manchester.
- Nomadic Technologies, 1999, “XR4000 Mobile Robot”, Disponível na Internet via WWW, URL: <http://www.robots.com/products.htm>.
- Parmar, R. S., 1995, “Welding Processes and Technology”, Khanna Publishers, Delhi.
- Tourino, S. R. G. Álvares, A. J. 1999. “Algoritmo de Navegação para Robôs Autônomos Móveis”, VI Congresso Nacional de Estudantes de Engenharia Mecânica, Brasília.
- Zhai, S. Milgram, P., 1991, “A Telerobotic Virtual Control System”, Proc. SPIE. Boston.

ESTUDO EXPERIMENTAL DE UM CONTROLADOR A ESTRUTURA VARIÁVEL E COMPENSAÇÃO DE ATRITO APLICADOS EM UM ROBÔ INDUSTRIAL

Alejandro Rafael Garcia Ramirez

Francisco Javier Triveño Vargas

Edson Roberto de Pieri

Universidade Federal de Santa Catarina, Laboratório de Controle e Microinformática, Depto. de Engenharia Elétrica, P.O. Box 476, 88045-900 Florianópolis, SC, BRASIL.
E-mails: ramirez@lcmi.ufsc.br, vargas@lcmi.ufsc.br, edson@lcmi.ufsc.br

Resumo

Neste trabalho é realizado um estudo experimental sobre o controle a estrutura variável por modos deslizantes e compensação de atrito aplicado em um robô industrial tipo SCARA. Resultados experimentais que gerencia e controla os módulos de operação Os resultados experimentais alcançados são a principal contribuição deste trabalho.

Palavras chave: Robot control, sliding mode dynamics, sliding control, variable structure controllers, joint friction.

1 Introdução

Existe uma exigência natural visando aprimorar a tecnologia de construção e controle de robôs manipuladores, de forma que eles possam desempenhar tarefas mais complexas. Assim, o estudo das leis de controle de robôs manipuladores constitui uma área ainda aberta à pesquisa. Uma técnica que tem se mostrado eficiente é o Controle a Estrutura Variável (Utkin, 1977). Esta técnica garante robustez ante perturbações e incertezas nos parâmetros do modelo do robô. Os primeiros trabalhos surgiram na antiga União Soviética e as primeiras aplicações no ocidente aconteceram na década de 70. Diversos pesquisadores realizaram o estudo e aplicação desta técnica em diversas áreas, incluindo a robótica, ver por exemplo, (Bailey and Arapostathis, 1987; De Carlo et al., 1988; Guenther and Hsu, 1993; Amaral et al., 1998).

Um problema a ressaltar nos sistemas de controle de robôs manipuladores, é que eles seriam mais simples se os atuadores fossem lineares. Na realidade, os atritos presentes nos redutores de velocidade fazem com que o torque aplicado no rotor não seja transmitido à estrutura de forma linear. O conhecimento do modelo do atrito é então essencial para o desenvolvimento das leis de controle (Gomes, 1995; Kircanski and Goldenberg, 1997).

Neste trabalho é realizado um estudo experimental sobre o controle de posição a estrutura variável de robôs manipuladores baseado na metodologia desenvolvida em (Bailey and Arapostathis, 1987). Experimentos são realizados em um robô manipulador industrial tipo SCARA, produzido pelo ETH-Suíça. Através dos resultados demonstra-se a importância da compensação do atrito, a qual constitui um aporte à lei de controle apresentada.

2 Controle a Estrutura Variável

O controle por modos deslizantes utiliza uma lei de controle chaveada a alta velocidade cujo objetivo é dirigir a trajetória de estados do sistema a uma superfície especificada durante a fase do projeto, chamada de superfície de deslizamento. Uma vez que a trajetória dos estados do sistema alcança esta superfície, mantém-se nela para todos os tempos subsequentes, deslizando sobre ela até o equilíbrio (Utkin, 1977).

O controle aplicado faz com que os vetores tangentes à trajetória do sistema apontem à superfície de deslizamento, conduzindo e confinando os estados sobre a superfície. Se diz que os estados deslizam nesta superfície (Utkin, 1977).

Um sistema não-linear de ordem n expresso na sua forma regular, pode ser controlado por uma lei de controle a estrutura variável de ordem m . Isto é representado através da seguinte equação (Utkin, 1977):

$$\dot{x} = f(x, t, u), \quad x \in \mathcal{R}^n, \quad u \in \mathcal{R}^m \quad (1)$$

e a lei de controle chaveada é dada por:

$$u_i(x, t) = \begin{cases} u_i^+, & s_i(x, t) < 0 \\ u_i^-, & s_i(x, t) > 0 \end{cases} \quad (2)$$

$$i = 1 \dots m$$

onde $s_i(x, t)$ é uma função escalar do vetor de estado, sendo que o conjunto de funções $s_i(x, t) = 0$ definem uma hipersuperfície no espaço de estados para onde a trajetória do sistema deve ser conduzida (Utkin, 1977).

2.1 Camada Limite e Chaveamento

Devido ao chaveamento extremamente rápido da lei de controle, é comum a ocorrência de um fenômeno denominado *chattering*. No caso ideal, uma vez que a trajetória de estados atinja a superfície $S(x, t) = 0$ em um instante t_i ela permanece deslizando nesta superfície para todos os tempos subsequentes (Utkin, 1977).

Entretanto, nos casos reais, a limitação física da velocidade de resposta dos atuadores, tais como atraso, zona morta etc., implicam que a condição ideal para o modo deslizante não é alcançada. Ocorre então uma oscilação da trajetória de estados em torno da superfície de deslizamento. com *chattering* (De Carlo et al., 1988).

Para reduzir o efeito do *chattering* tem-se adotado o conceito de *camada limite*. Tal conceito considera não somente o deslizamento ocorrendo na superfície, mas ao longo de uma vizinhança em torno da superfície de deslizamento (Utkin, 1977).

O que deseja-se obter através da aplicação da camada limite é um chaveamento *mais suave*, em relação ao chaveamento extremamente rápido da lei de controle. Assim, a equação (2) é alterada para adotar a seguinte forma (De Carlo et al., 1988):

$$\text{sat}\left(\frac{s}{\epsilon}\right) = \begin{cases} \text{sign}(s), & \text{se } |s| \geq \epsilon \\ \frac{s}{\epsilon}, & \text{se } |s| < \epsilon \end{cases} \quad (3)$$

3 Modelo do Robô

As equações que descrevem o comportamento dinâmico de um robô manipulador aparecem descritas em diversos trabalhos, ver por exemplo, (Asada and Slotine, 1986; Yoshikawa,

1990).

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = u \quad (4)$$

onde θ é o vetor das posições angulares de dimensão $n \times 1$; $\dot{\theta}$ é o vetor das velocidades angulares de dimensão $n \times 1$; $M(\theta)$, $C(\theta, \dot{\theta})$ e $G(\theta)$ são as matrizes de inércia, força centrípeta e de coriolis, e a força gravitacional, respectivamente. Sendo u o vetor de forças generalizadas aplicadas em cada uma das juntas.

O robô usado nos experimentos (ver figura 1) tem 4 graus de liberdade, do tipo SCARA. Foi construído pelo ETH de Suíça para fins de pesquisa em controle de força-posição.



Figure 1: Robô SCARA

Neste trabalho estamos interessados apenas nas duas primeiras juntas do robô, as quais têm uma dinâmica similar a de um robô planar.

A tabela 1 mostra os valores nominais dos parâmetros do robô:

Table 1: Parâmetros do robô

Parâmetros	<i>Elo 0</i>	<i>Elo 1</i>
Inércia (kgm^2)	0.23	0.16
Massa (kg)	15	12
Longitude (m)	0.25	0.25
Centro de massa (m)	0.10	0.18

4 Projeto do Controlador

A seguir são definidas as etapas para a implementação do controlador a estrutura variável. Os passos do projeto estão baseados na metodologia apresentada em (Bailey and Arapostathis, 1987).

4.1 Projeto da superfície de deslizamento

A superfície de deslizamento é uma função linear dos estados do sistema:

$$s_i(x, t) = s_i(\theta, \dot{\theta}, t) = c_i[\theta_i - \theta_{d_i}] + [\dot{\theta}_i - \dot{\theta}_{d_i}] \quad (5)$$

tal que $c_i > 0$ são constantes; e θ_i , $\dot{\theta}_i$, θ_{d_i} , $\dot{\theta}_{d_i}$ são as posições obtidas e desejadas em cada uma das juntas. Os erros de posição e velocidade são definidos como:

$$\begin{aligned} \tilde{\theta}_i &= \theta_i(t) - \theta_{d_i}(t) \\ \tilde{\dot{\theta}}_i &= \dot{\theta}_i(t) - \dot{\theta}_{d_i}(t) \end{aligned} \quad (6)$$

4.2 Projeto da lei de controle

Os ganhos do controlador são dados pela seguinte expressão:

$$K_i = \sum_{j=1}^2 [\bar{M}_{ij} | \ddot{\theta}_{d_j} - c_j \tilde{\dot{\theta}} | + \bar{C}_{ij} | \dot{\theta}_j - s_j |] + \xi_i \quad (7)$$

onde ξ_i é um vetor de projeto, \bar{M}_{ij} e \bar{C}_{ij} são as componentes das matrizes de inércia e coriolis, obtidas a partir do modelo dinâmico do robô (Bailey and Arapostathis, 1987):

$$\bar{M}_{ij} = \max M_{ij}(\theta) \quad | \quad \bar{C}_{ij} = \max \bar{C}_{ij}(\theta, \dot{\theta}) \quad (8)$$

4.2.1 Lei de controle

A lei de controle é:

$$u_i = -K_i \text{sign}(s_i) \quad (9)$$

onde u_i são os torques de controle aplicados em cada uma das juntas do robô.

4.2.2 Lei de controle com camada limite

Para a redução do *chattering* a lei de controle é modificada pela seguinte função saturação:

$$u_i = -K_i \text{sat}\left(\frac{s_i}{\epsilon}\right) \quad (10)$$

tal que K_i são os ganhos definidos na equação (7); e s_i é definida como em (5).

A equação (10) é modificada para incluir um termo para a compensação do atrito nas juntas do robô:

$$u_i = -K_i \text{sat}\left(\frac{s_i}{\epsilon}\right) + F_i(x) \quad (11)$$

tal que $F_i(x)$ representa o torque devido ao atrito.

5 Modelagem do atrito

A função $F_i(x)$ da equação (11) usualmente tem duas componentes, sendo que uma representa o atrito estático, e a outra, o atrito viscoso.

$$F(x) = B \dot{\theta} + C \text{sign}(\dot{\theta}) \quad (12)$$

onde B e C são os coeficientes de atrito viscoso e estático respectivamente.

Em particular, neste trabalho a expressão (13) permite determinar os torques para a compensação do atrito. Cabe ressaltar que a função F e os coeficientes a , b , c são diferentes para cada tipo de robô e, em geral, são conhecidos no módulo de controle do robô.

$$F = offset + sign(\dot{\theta}) (a + b abs(\dot{\theta}))(1 - e^{-c abs(\dot{\theta})}) \quad (13)$$

A tabela 2 mostra os valores dos coeficientes utilizados na compensação de atrito. Os coeficientes a e b acompanham a parcela viscosa do atrito, e o c , a estática. Também são mostrados os valores de c_i (ganhos da superfície) e de ϵ_i para a redução do chattering.

Table 2: Coeficientes da lei de controle

Coeficientes	<i>Junta0</i>	<i>Junta1</i>
Offset	0.3	-0.5
a	12	5.9
b	6	2.3
c	10	10
c_i	10	10
ϵ_i	0.3	0.5

Cabe ressaltar que na literatura são descritas outras formas de modelagem de atrito, ver por exemplo (Gomes, 1995). Isto será objeto de estudo em trabalhos futuros.

6 Resultados experimentais

A figura 2 mostra de forma simplificada a instalação experimental utilizada.

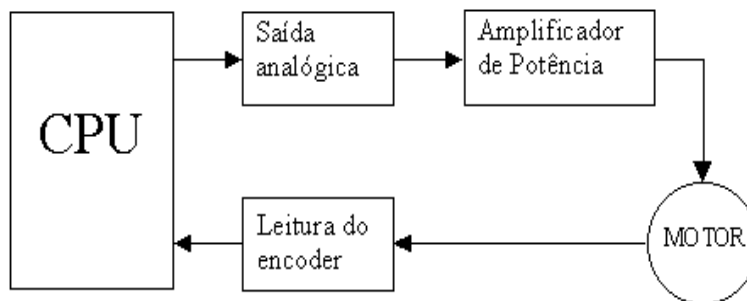


Figure 2: Estrutura de controle

O controle é implementado na forma de algoritmo no bloco CPU (Unidade Central de Processamento, neste caso o PC), sendo que a linguagem de programação do robô é XOberon¹ (Reiser, 1991). Existem diversos módulos que acompanham o robô e que permitem a geração de trajetórias, leituras das posições e velocidades nas juntas,

¹XOberon é uma extensão da linguagem Oberon. Permite uma programação em estilo orientado a objeto com característica de tempo real.

compensação do atrito, transformações de coordenadas, leituras a partir de um sensor de força, entre outras funções. Estes módulos são abertos e podem ser reprogramados (Weihmann, 1999).

Basicamente, o algoritmo de controle gera um torque para cada junta com o objetivo de compensar os erros de posição e velocidade na trajetória desejada. Estes torques são então convertidos em tensão e aplicados nos motores do robô. A leitura das posições das juntas é realizada através de codificadores angulares (*encoders*), os quais entregam o valor lido ao algoritmo de controle, onde é determinado o torque que vai ser aplicado para reduzir o erro de seguimento, repetindo-se o processo.

O período de amostragem é de $1ms$, que é suficientemente pequeno em relação às velocidades de operação projetadas, e em relação aos ganhos do controlador e da superfície. Estes ganhos foram ajustados experimentalmente. O valor da camada limite foi obtido considerando as limitações físicas dos atuadores, como mostra a tabela 3.

Table 3: Constantes de tempo dos motores

Constantes	<i>Junta0</i>	<i>Juntas1, 2, 3</i>
Elétrica (ms)	1.1	2.1
Mecânica (ms)	1.4	1.7

Na figura 3 mostra-se a aplicação da lei de controle (10) (Bailey and Arapostathis, 1987) para uma trajetória *spline* de quinta ordem projetada para a junta 0 do robô, enquanto a junta 1 permanece parada. Observa-se a presença de erros de seguimento, pois o robô não acompanha a trajetória desejada.

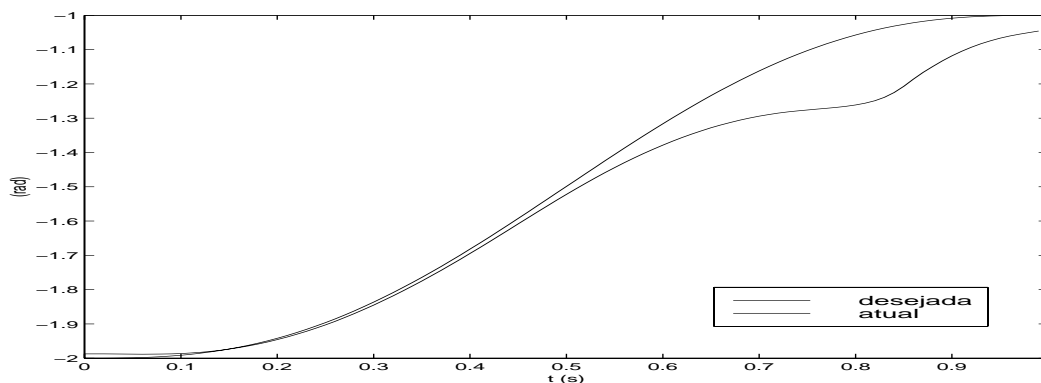


Figure 3: Trajetória de posição e posição desejada

Na figura 4 mostra-se a aplicação da lei de controle (10) (Bailey and Arapostathis, 1987) para uma trajetória *spline* de quinta ordem projetada agora para a junta 1 do robô, enquanto a junta 0 permanece parada. Observa-se a presença de erros de seguimento consideráveis.

Nas figuras 5 e 6 mostra-se o resultado da aplicação da lei de controle (11), nas mesmas trajetórias projetadas para cada junta de maneira independente. Neste caso foi

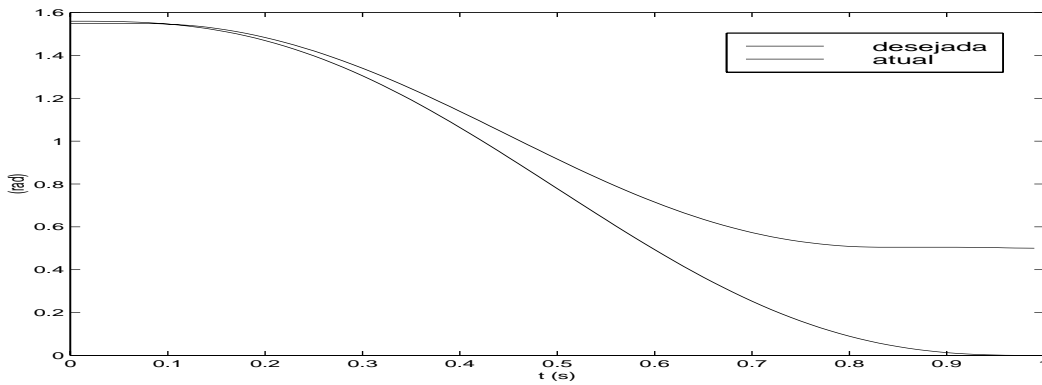


Figure 4: Trajetória de posição e posição desejada

modificada a lei de controle proposta em (Bailey and Arapostathis, 1987) para incluir o termo da compensação de atrito considerando a equação (13). Observa-se a diminuição significativa dos erros de seguimento em ambas as juntas, pois o robô acompanha as trajetórias desejadas.

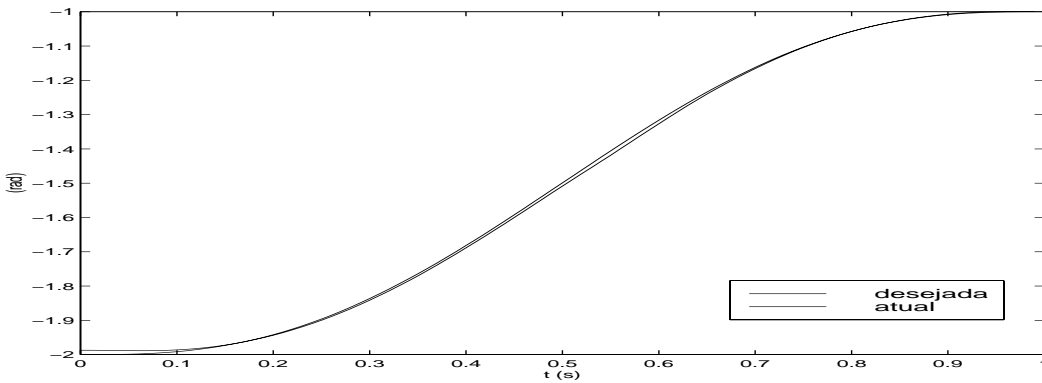


Figure 5: Trajetória de posição e posição desejada

Nas figuras 7 e 8 mostra-se a aplicação da lei de controle (10) considerando as mesmas trajetórias dos testes anteriores, só que agora projetadas simultaneamente nas juntas do robô. Observa-se, novamente, a presença de erros de seguimento quando não é compensado o atrito presente nas juntas.

Nas figuras 9 e 10 mostra-se a aplicação da lei de controle (11) considerando as mesmas trajetórias das figuras 7 e 8, projetadas simultaneamente nas juntas 0 e 1 do robô.

Observa-se a redução sinificativa dos erros de seguimento quando é compensado o atrito presente nas juntas

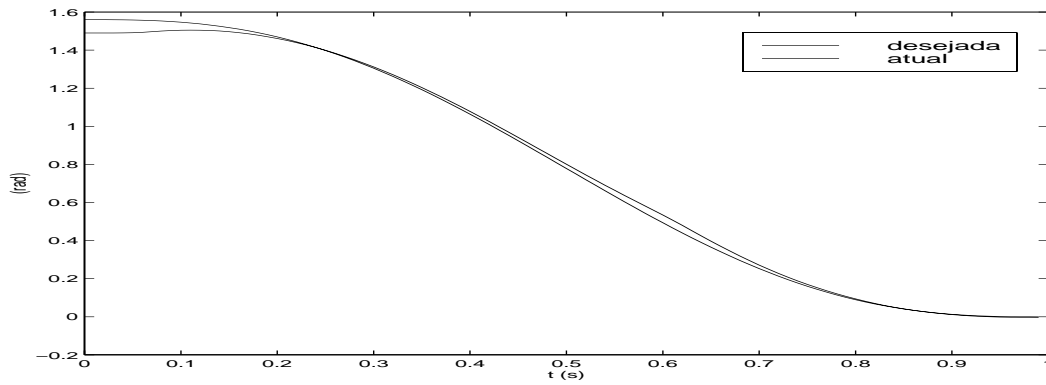


Figure 6: Trajetória de posição e posição desejada

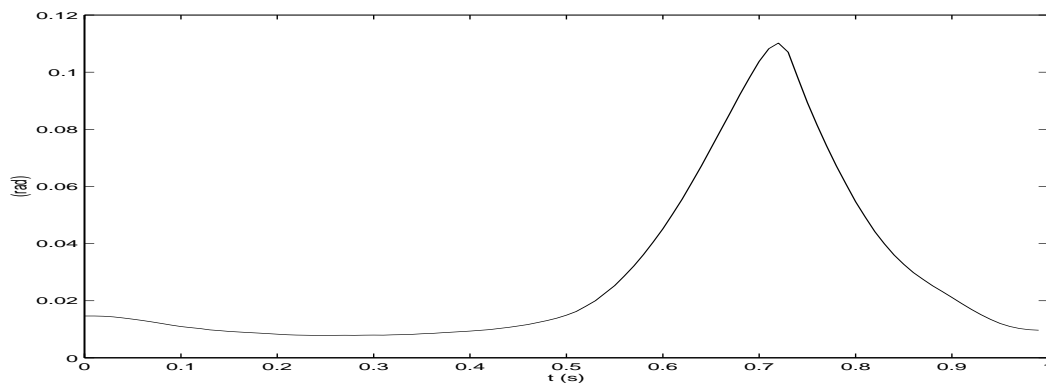


Figure 7: Erros de posição na junta 0

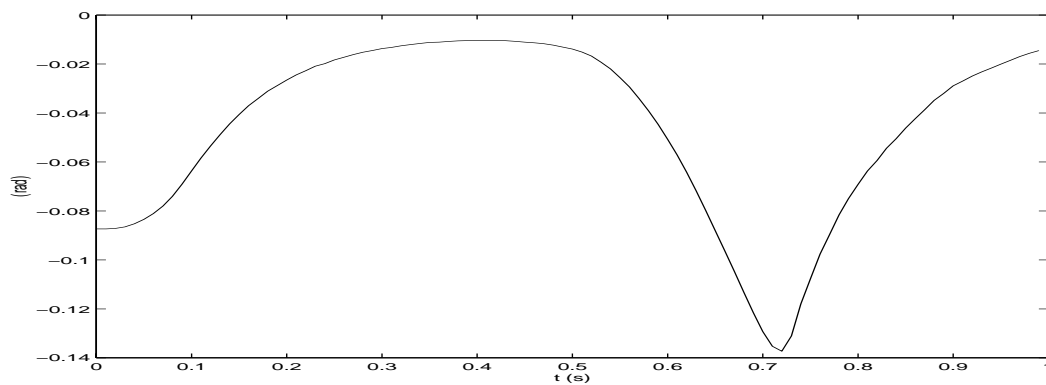


Figure 8: Erros de posição na junta 1

7 Conclusões

Foi avaliada experimentalmente a estratégia de controle descrita em (Bailey and Arapostathis, 1987) nas duas primeiras juntas de um robô industrial tipo SCARA. Considerando os erros de posição que surgiram na realização dos testes, foi aperfeiçoada esta metodologia incluindo o termo para compensação de atrito (estático e dinâmico), o qual implicou numa considerável melhoria no desempenho do robô. Este trabalho mostrou a importância desta dinâmica, a qual usualmente não é considerada no modelo do robô nem nas leis de controle, mas que pode afetar a qualidade do controle e o desempenho do robô industrial. Trabalhos futuros estão dirigidos ao estudo e implementação de outras formas de modelagem de atrito, incluindo a extensão destes resultados em robôs com transmissões flexíveis.

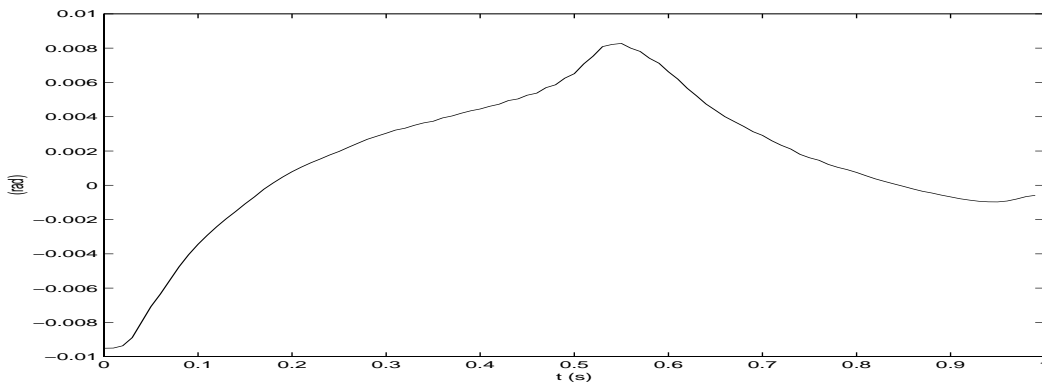


Figure 9: Erros de posição na junta 0

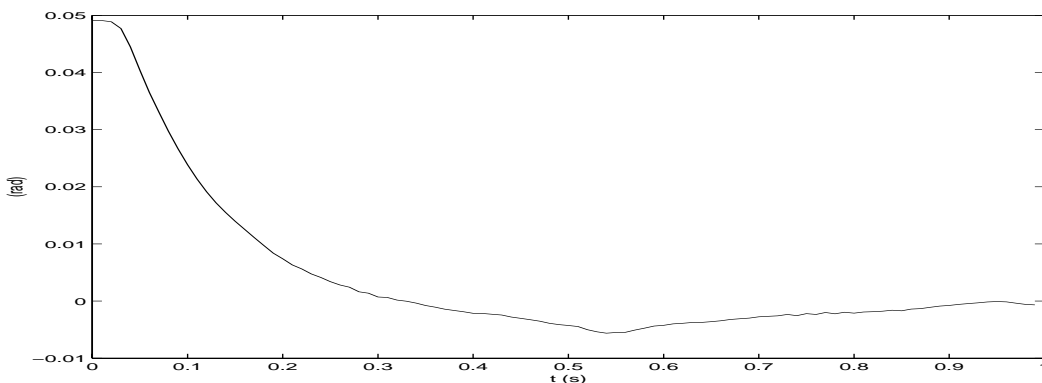


Figure 10: Erros de posição na junta 1

8 Agradecimentos

Agradecemos à CAPES pelo suporte material e financeiro, o qual possibilitou a realização deste trabalho.

References

- Amaral, S. D., De Pieri, E. R. and Guenther, R. (1998). Controle a estrutura variável de um robô manipulador em ambiente cinemático, *Congreso Brasileiro de Automática* **3**(1): 1113–1118.
- Asada, H. and Slotine, J. E. (1986). *Robot Analysis and Control*, Jonh Willey and Sons, New York.
- Bailey, E. and Arapostathis, A. (1987). Simple sliding mode control scheme applied to robot manipulators, *International Journal of Control* **45**(4): 1197–1209.
- De Carlo, R. A., Stanislaw, H. Z. and Mathews, G. P. (1988). Variable structure control of nonlinear multivariable systems a tutorial, *IEEE Proceedings* **76**(4): 212–232.
- Gomes, S. C. P. (1995). An experimental study of nonlinear stiffness, hysteresis, and friction effects in robot joints with harmonic drives and torque sensors, *Congreso Brasileiro de Engenharia Mecânica* pp. 1–10.
- Guenther, R. and Hsu, L. (1993). Variable struture adaptative cascade control of rigid-link electrically driven robot manipulators, *Proceedings of IEEE Conference on Decision and Control*.
- Kircanski, N. M. and Goldenberg, A. A. (1997). Modelagem de atritos internos às articulações de robôs manipuladores, *The International Journal of Robotics Research* **16**(2): 214–239.
- Reiser, M. (1991). *The Oberon System: User Guide and Programer's Manual*, ACC Press, Inc., New York.
- Utkin, V. I. (1977). Variable structure with sliding modes, *IEEE Transactions on Automatic Control* **1**(1): 212–222.
- Weihmann, L. (1999). *Módulos de operação do Robô Inter e linguagem de programação XOberon*, Disertação submetida à Universidade Federal de Santa Catarina, Brasil.
- Yoshikawa, T. (1990). *Foundations of Robotics Analysis and Control*, The MIT Press, EUA.

MODELAGEM E SIMULAÇÃO DE UM ROBÔ BÍPEDE

Daniel Carmona de Campos

Universidade Estadual de Campinas – Departamento de Mecânica Computacional.

Carlos André Dias Bezerra

Universidade Federal do Ceará – Departamento de Engenharia Mecânica. cadb@ufc.br

Douglas Eduardo Zampieri

André Mendeleck

Universidade Estadual de Campinas – Departamento de Mecânica Computacional.

Resumo

A necessidade de locomoção em ambientes sujeitos a alterações dinâmicas é uma realidade. O robô bípede vem atender a estas necessidades, pois tem a capacidade de locomoção em ambiente tridimensional, onde os robôs móveis com rodas geralmente não podem acessar. O objetivo deste trabalho é a concepção, modelagem cinemática e simulação de um robô móvel do tipo bípede, semelhante à uma estrutura antropomórfica com três rotações por perna. Metodologicamente o trabalho será dividido em duas etapas: determinação das equações do modelo cinemático, através das restrições do sistema mecânico e simulação do movimento do robô bípede em código MATLAB[®].

Palavras-chave: Robô Bípede, Modelo Cinemático, Estrutura Híbrida, Simulação Computacional.

1. INTRODUÇÃO

A substituição de mão de obra humana por sistemas automatizados e autômatos junto à sociedade é uma tendência decorrente da evolução tecnológica no mundo moderno. O estudo da locomoção sobre pernas é um dos mais desafiantes problemas relacionados com a robótica, pois os robôs que se locomovem sobre duas pernas são menos estáveis do que os robôs móveis com mais de 2 pernas, quadrúpedes e hexápodes, por exemplo.

Ao longo dos últimos anos, muitos pesquisadores têm sido atraídos para a área de sistemas robóticos bípedes (Raibert, 1986) devido a sua semelhança antropomórfica e ao potencial de mobilidade similar a do ser humano, que o permite subir escadas, andar em áreas contendo obstáculos, superfícies inclinadas e rugosas e até mesmo andar sem atuadores (McGeer, 1990). Além disso, os robôs bípedes apresentam potencial para uso em áreas perigosas ao ser humano, uso militar, reabilitação de deficientes físicos entre outros. Todavia, nenhum dos robôs bípedes até agora estudados apresentam a mobilidade e destreza humana (Lum *et al.*, 1999), pois os modelos usados são simplificados através da redução do número de graus de liberdade, da substituição das juntas por uniões de rotação e ao se considerar os segmentos como corpos rígidos (Popovic *et al.*, 1999).

Desde 1968 (Frank, 1968) várias técnicas de modelagem e controle têm sido propostas (Hemami and Wyman, 1979; Raibert; 1986; Furusho and Masubuchi, 1986 e Vukobratovic *et al.*, 1990) para solucionar o principal problema inerente ao robô bípede, que é a sua instabilidade durante a fase em que somente um pé está em contato com o chão. Porém, foi

somente em 1973 que o primeiro robô bípede, o WL-5, foi projetado, construído e citado na literatura científica por Kato e seus colegas (Kato *et al.*, 1974).

O modo de andar de um robô bípede é obtido através do movimento alternado de suas pernas, excetuando-se os robôs que pulam (Raibert, 1986). Pode-se dizer que o modo de andar (*gait*) é constituído por duas etapas distintas: bi-apoiada (*Stance fase*) quando os dois pés do robô bípede estão em contato com o chão e balanço (*Balance fase*) quando somente um dos pés está apoiado no chão. Nesta última etapa a perna que não se encontra apoiada é denominada de perna em balanço (*Swinging leg*).

Propõe-se neste trabalho obter o modelo cinemático de um robô bípede, bem como simular seu modo de andar, quantificando as variações de suas grandezas cinemáticas. A partir das equações que definem a cinemática da estrutura do robô bípede, será obtida a posição, a velocidade e a aceleração do centro de gravidade de cada segmento que compõe o robô. A simulação será realizada no software MATLAB[®]. Através de gráficos será analisado o movimento do robô, utilizando como parâmetros a posição dos segmentos ao longo do tempo, bem como suas velocidades e acelerações e a variação dos ângulos de cada junta.

2. MODELAGEM CINEMÁTICA DO ROBÔ BÍPEDE

O robô bípede (figura 1), a ser simulado computacionalmente, é constituído por 7 corpos: pé direito – segmento 1, canela direita – segmento 2, coxa direita – segmento 3, bacia – segmento 4, coxa esquerda – segmento 5, canela esquerda – segmento 6 e pé esquerdo – segmento 7. Todas 6 juntas de união entre os segmentos são de rotação, com 1 grau de liberdade, sendo desprezado o atrito entre elas. Cada junta é movida por um atuador independente.

O modo de andar do robô bípede neste trabalho será considerado simétrico e periódico. O robô sairá da posição inicial, que corresponde aos pés juntos e pernas eretas e após o segundo passo entrará em regime permanente.

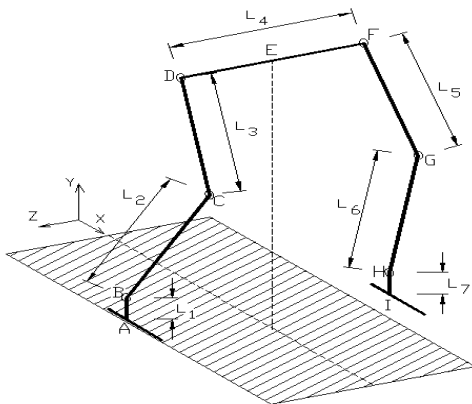


Figura 1. Vista 3D do robô bípede

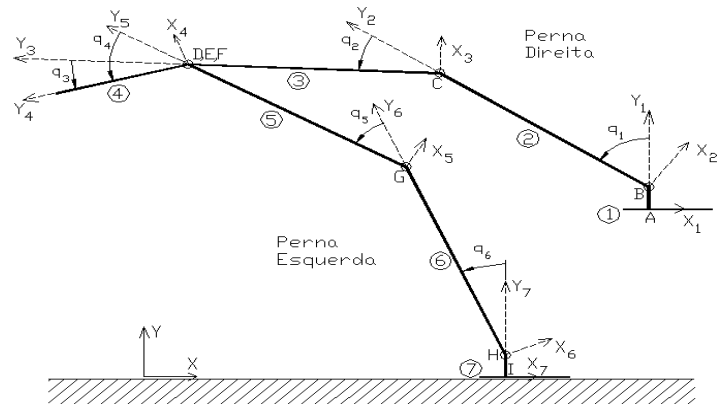


Figura 2. Bases móveis do robô bípede (Mita *et al.*, 1984)

O modelo matemático que descreve o movimento do robô bípede é obtido a partir dos seus parâmetros geométricos de acordo com as figuras 1 e 2. A bacia, representada pelo segmento D, E, F, é articulada (q_3 , q_4) em relação às duas pernas, servindo de suporte às mesmas.

Neste trabalho os centros de gravidade de cada segmento estão posicionados nas juntas, pois a massa dos atuadores é relativamente grande em relação a massa de cada segmento.

Com relação aos pés, estes sempre se mantêm paralelos ao solo, ou seja, não sofrem nenhuma rotação ao longo de todo movimento. Mesmo em balanço, o pé permanecerá paralelo através de controle local (*local feedback*, Mita *et al.*, 1984) de um atuador.

As coordenadas generalizadas neste modelo são representadas pelos ângulos relativos q_i ($i=1,\dots,6$), todos definidos como positivos na direção de Z inercial (figura 2). Estes ângulos são dados por rotações das base móveis B_i ($i=1,\dots,7$), uma em relação a outra (q_1 é a rotação da base B2 em Z em relação a B1, por exemplo). Cada base é solidária a seu respectivo segmento. Assim a base B3 é solidária ao corpo 3 (coxa direita) e assim por diante.

Quando o robô bípede realiza um passo, o sistema mecânico que o descreve possui 5 graus de liberdade, cuja equação de restrição é dada por

$$q_1 + q_2 + q_3 - q_4 - q_5 - q_6 = 0 \quad (1)$$

Para o caso do pé direito fixo ao chão, q_6 é determinado em função das outras coordenadas, enquanto que no caso do pé esquerdo fixo ao chão, a incógnita será q_1 . Esta simplificação é possível, já que os pés sempre estão paralelos entre si.

Considera-se na simulação que o robô bípede se locomove em uma superfície plana e horizontal, que o impacto entre o pé e o chão é perfeitamente inelástico e que não existe deslizamento entre os pés do robô e a superfície de locomoção. Além disso, considera-se o movimento somente no plano sagital, que é o plano perpendicular ao eixo Z formado pelos eixos XY.

A partir dos parâmetros das figuras 1 e 2, pode-se obter as coordenadas do centro de gravidade das juntas de cada segmento em relação ao referencial inercial.

Considerando-se o pé esquerdo em balanço (pé direito como suporte), obtém-se:

$$\begin{aligned} x_{AI} &= -l_2 \sin(q_1) - l_3 \sin(q_1 + q_2) + l_5 \sin(q_1 + q_2 + q_3 - q_4) + l_6 \sin(q_1 + q_2 + q_3 - q_4 - q_5) \\ y_{AI} &= l_1 + l_2 \cos(q_1) + l_3 \cos(q_1 + q_2) - l_5 \cos(q_1 + q_2 + q_3 - q_4) - l_6 \sin(q_1 + q_2 + q_3 - q_4 - q_5) - l_7 \\ z_{AI} &= -l_4 \end{aligned} \quad (2)$$

A equação representa a posição do ponto I a partir do ponto A na base inercial. Mas também descreve a posição dos outros pontos, já que o robô nesta situação é uma cadeia cinemática aberta, bastando eliminar coerentemente alguns termos da equação.

Analogamente, considerando-se o pé direito em balanço (pé esquerdo como suporte), obtém-se:

$$\begin{aligned} x_{IA} &= -l_6 \sin(q_6) - l_5 \sin(q_5 + q_6) + l_3 \sin(-q_3 + q_4 + q_5 + q_6) + l_2 \sin(-q_2 - q_3 + q_4 + q_5 + q_6) \\ y_{IA} &= l_7 + l_6 \cos(q_6) + l_5 \cos(q_5 + q_6) - l_3 \cos(-q_3 + q_4 + q_5 + q_6) - l_2 \cos(-q_2 - q_3 + q_4 + q_5 + q_6) - l_1 \\ z_{IA} &= l_4 \end{aligned} \quad (3)$$

Observa-se que o sistema é simétrico, pois as equações tem o mesmo formato, variando-se apenas os índices.

Para se obter as velocidades e as acelerações, basta derivar sucessivamente as equações (2) e (3) em relação ao tempo. Estas passagens serão omitidas neste trabalho.

3. SIMULAÇÃO

As simulações foram realizadas em código MATLAB[®] e representam a trajetória dos centros de gravidade de cada segmento. Como não existe rotação de nenhum segmento em relação ao eixo X ou Y, o movimento se limita ao plano XY.

Foi adotada a cinemática inversa como metodologia para movimentação do robô, ou seja, foi estabelecida a trajetória dos pés (pontos A e I) e da cintura (pontos D, E e F). Uma vez estabelecida a trajetória do pé em balanço e da bacia e o tempo de cada, pode-se calcular a posição dos joelhos e de todos os ângulos q_i em cada instante de tempo, pois como o robô bípede em balanço é um sistema de 5 graus de liberdade, ao se estabelecer a posição do pé e da bacia, 4 restrições foram estabelecidas (coordenadas X e Y de cada um) e, assumindo-se que a bacia não sofre rotação em relação ao sistema inercial, o sistema se torna totalmente determinado.

Para a determinação da melhor trajetória a ser seguida em função do acionamento das juntas, Vukobratovic *et al.* (1990) adotaram o modelo humano, ou seja, foram obtidos dados através de medições do modo de andar de pessoas. Bay e Hemami (1987) utilizaram o Gerador Central de Padrões (CPG) para gerar movimentos rítmicos das pernas. Katoh e Mori (1984) usaram um par de osciladores de van der Pol para gerar padrões de passos para o robô bípede BIPMAN. Cheng e Lin (1997) propuseram um algoritmo genético para a determinação da trajetória ótima.

Neste trabalho, foi estabelecida por trajetória dos pés em cada passo (*step*) meia senóide ou meia elipse. Especificando-se o ponto de início, a amplitude (altura do chão) e a distância em X de cada passo, a trajetória do pé se torna totalmente conhecida, tanto para a função senoidal, quanto para função elíptica (figuras 3 e 4). Considerou-se também que tanto a bacia quanto o pés possuem velocidades lineares constantes na direção X e que a fase bi-apoiada é instantânea.

Dos diversos robôs bípedes já construídos, observou-se que a bacia em alguns casos mantém-se com uma distância constante em relação ao solo. No entanto ao se observar o modo de andar humano, nota-se que existe uma variação de altura periódica em relação ao chão. Assim neste trabalho foi estabelecido que a bacia percorre uma trajetória senoidal em Y de pequena amplitude e, portanto, mantém uma altura média constante em relação ao solo. O mínimo desta função senoidal ocorre quando os dois pés estão apoiados no solo e seu máximo ocorre durante o balanço quando a bacia está na mesma vertical do centro do pé de apoio, semelhante ao modo de andar humano, conforme pode-se observar nas figuras 3 e 4.

4. RESULTADOS

Nas simulações utilizou-se os seguintes valores:

- *pernas* $l_2 = l_3 = l_5 = l_6 = 100$ mm
- *pés* $l_1 = l_7 = 10$ mm
comprimento da ponta dos pés = 15 mm
- *bacia* altura mínima da bacia em relação ao solo = 200 mm
amplitude de oscilação da bacia = 2 mm
- *obstáculos* 7 por 7 mm
- *passos* amplitude dos passos (altura máxima do solo) = 15 mm
distância de um passo em X (regime permanente) = 50 mm
distância de um passo em X (regime transitório) = 25 mm
período de 1 passo = 1 s
- *iterações* 30 por passo

Substituindo-se os parâmetros acima nas equações anteriores obtém-se as figuras 3 e 4.

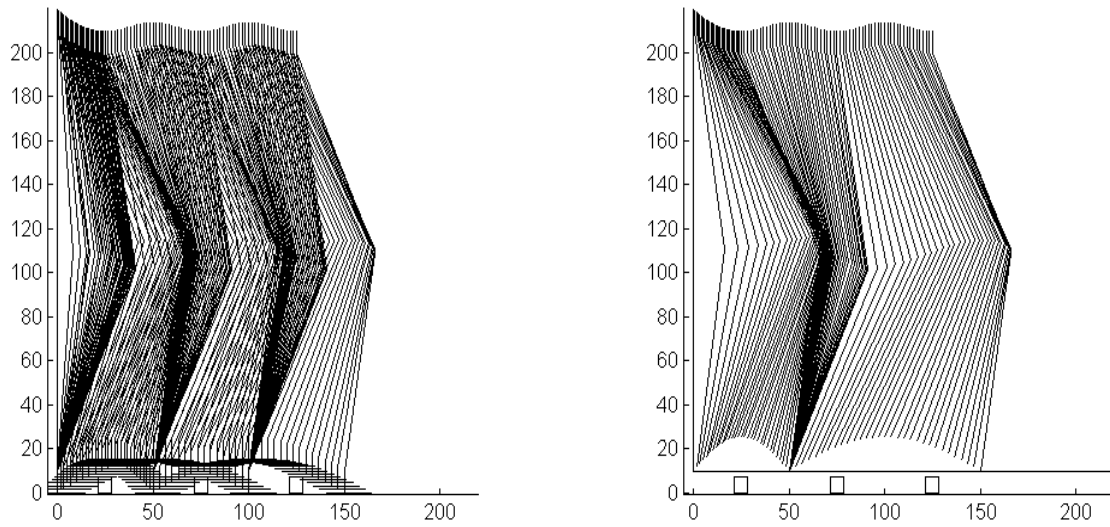


Figura 3. Modo de andar, trajetória Senoidal do pé (Figura 3b perna esquerda) [mm]

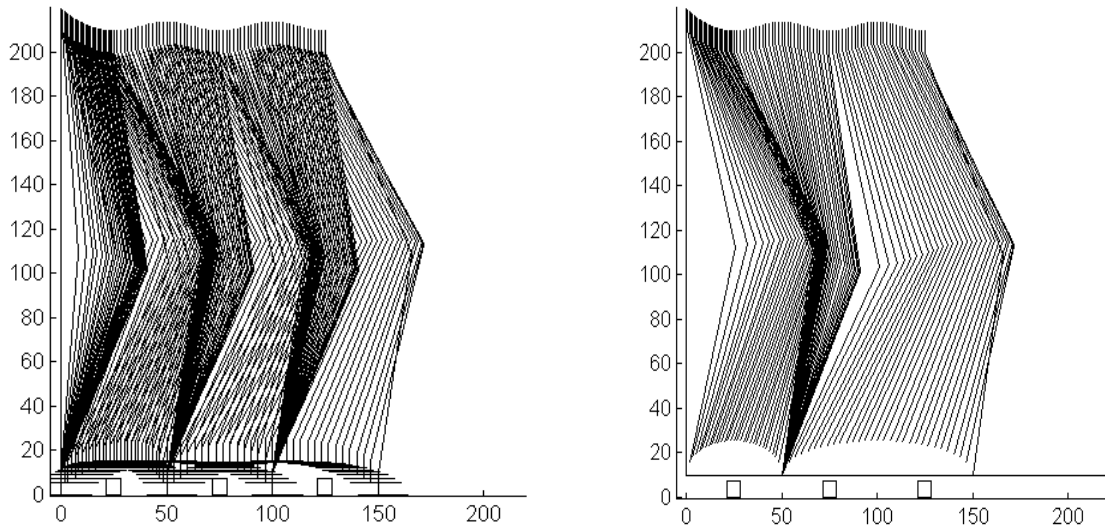


Figura 4. Modo de andar, trajetória Elíptica do pé (Figura 4b perna esquerda) [mm]

Como pode ser observado nas figuras 3b e 4b, para ambas as curvas a amplitude em Y e a distância de cada passo em X são as mesmas.

Nota-se, também, nas figuras 5 e 6 que para ambas trajetórias ocorre a mesma periodicidade para os ângulos de juntas simétricas, como era de se esperar. Além disso, verifica-se não existir grandes diferenças entre ambas trajetórias.

As figuras 7 e 10 mostram o deslocamento das juntas em função do tempo. O deslocamento de cada junta j ($j=A, \dots, I$) é representado por P_{jx} para deslocamento na direção inercial X e P_{jy} , para deslocamento na direção Y. Mais uma vez, pode-se observar a simetria das juntas B e H (tornozelos) e C e G (joelhos). As juntas A e I também são simétricas, mas não foram apresentadas pois seus gráficos são semelhantes a B e H, respectivamente.

De modo análogo, nas figuras 8 e 11 as velocidades são representadas por V_{jx} e V_{jy} e nas figuras 9 e 12 as acelerações por A_{jx} e A_{jy} .

Durante a transição que a perna em balanço passa da perna direita para a perna esquerda, ou vice-versa, ocorrem impactos que causam descontinuidades nas equações de movimento do modelo físico (Lum *et al*, 1999).

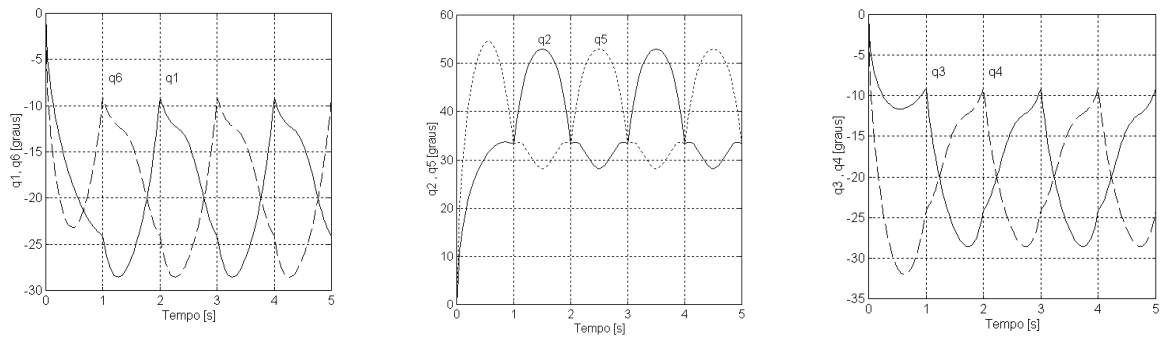


Figura 5. Ângulos q_i x tempo (pés com trajetória Senoidal)

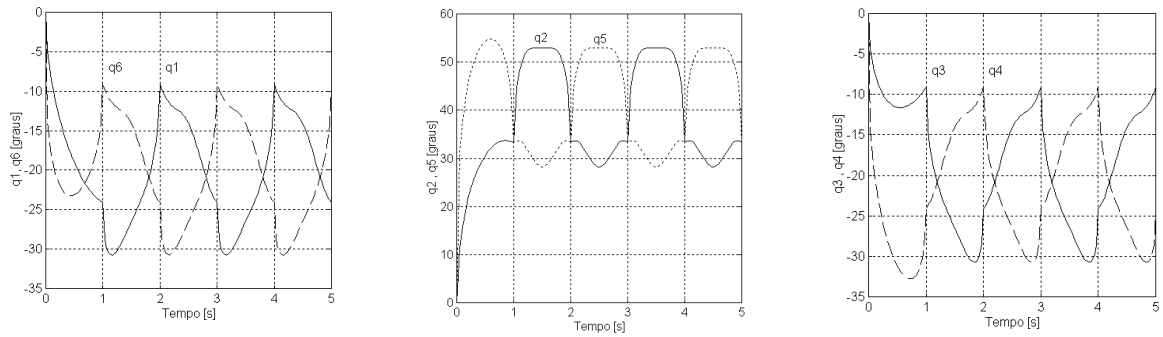


Figura 6. Ângulos q_i x tempo (pés com trajetória Elíptica)

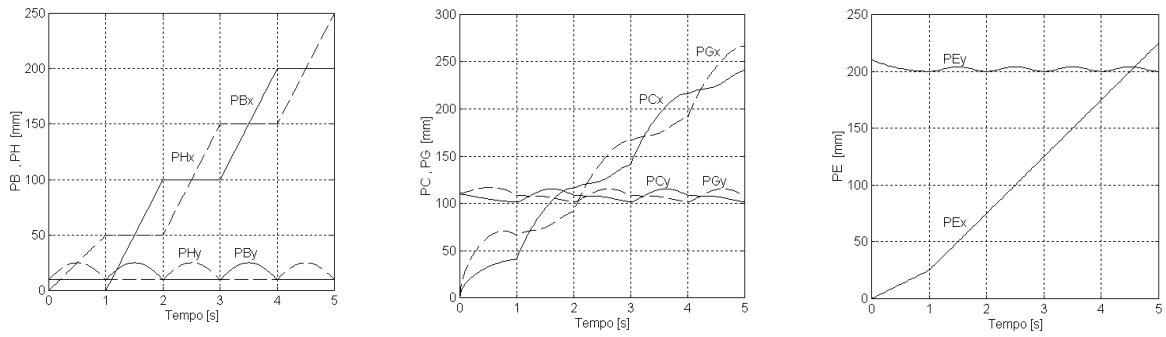


Figura 7. Deslocamento das juntas x tempo (pés com trajetória Senoidal)

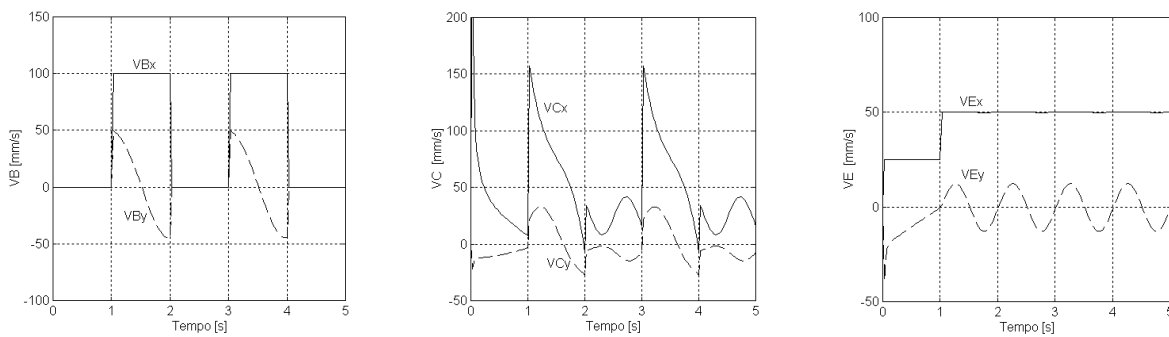


Figura 8. Velocidade das juntas x tempo (pés com trajetória Senoidal)

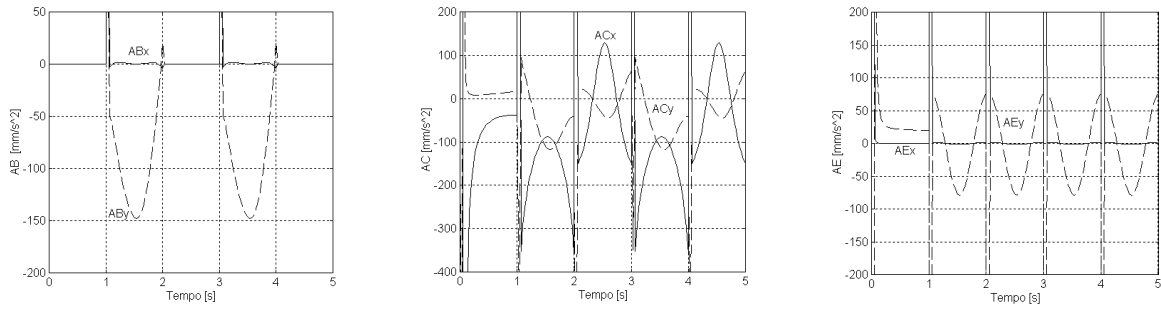


Figura 9. Aceleração das juntas x tempo (pés com trajetória Senoidal)

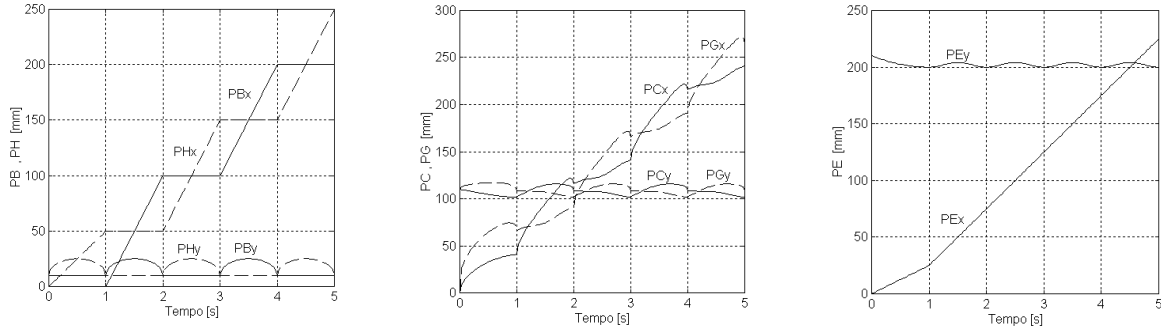


Figura 10. Deslocamento das juntas x tempo (pés com trajetória Elíptica)

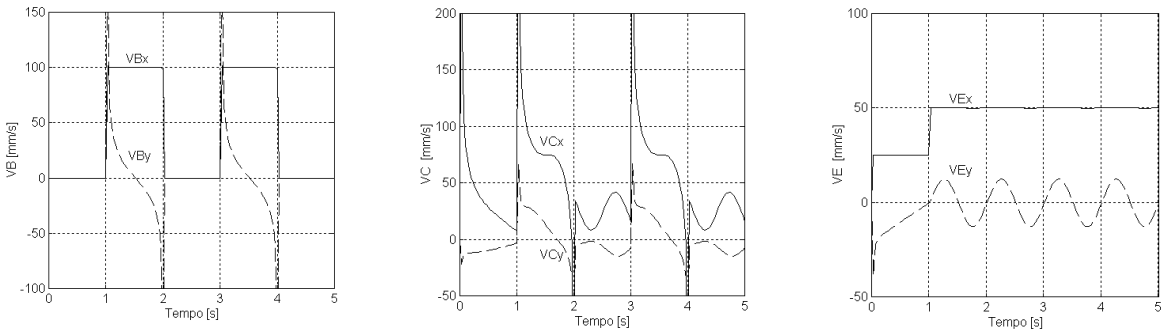


Figura 11. Velocidade das juntas x tempo (pés com trajetória Elíptica)

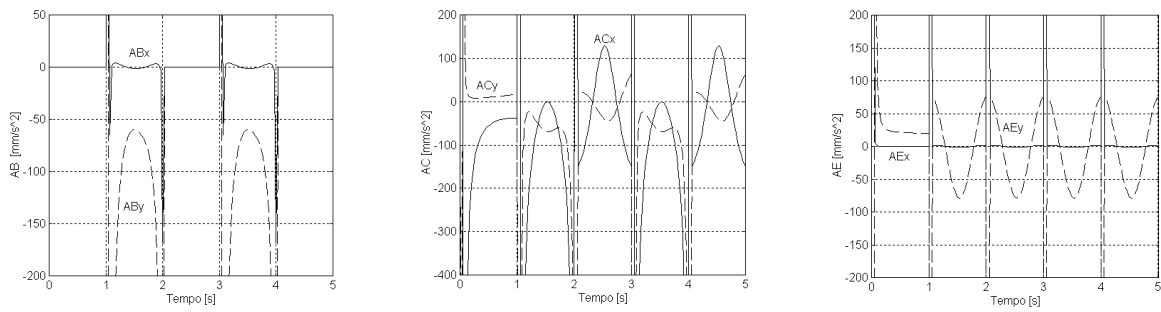


Figura 12. Aceleração das juntas x tempo (pés com trajetória Elíptica)¹

Como os pontos B e H são simétricos, tanto a velocidade quanto a aceleração só foram apresentadas para a junta B, já que as curvas de H são idênticas apenas deslocadas em fase no tempo em regime permanente. O raciocínio é similar para as juntas C e G.

¹ As variações de ABx, por exemplo, entre os instantes 1 e 2, 3 e 4 devem-se ao pequeno número de iterações ($ABx_{teórico} = 0$), já que tanto as acelerações como as velocidades são obtidas a partir de cálculos discretos dos deslocamentos.

A velocidade do ponto E (bacia) na direção X em regime permanente é constante em 50 mm/s e a velocidade do pé (ponto B ou H) varia entre 0 e 100 mm/s dependendo do instante de tempo, mas a cada 2 s (ou dois passos) a velocidade média de B ou H também é igual a 50 mm/s, mostrando que tanto o pé como a bacia apresentam mesma velocidade média.

5. CONCLUSÃO

A simulação apresentada permitiu validar o modelo proposto para o modo de andar do robô bípede com 7 segmentos no plano sagital. Através da mesma pôde-se visualizar a trajetória realizada por cada junta do robô, bem como obter a velocidade e aceleração das mesmas.

A função que gerou o padrão para o modo de andar não é única (trajetória dos pés Senoidal ou Elíptica). Esta função pode ser modificada para capacitar ao robô bípede transpor obstáculos específicos, ou desenvolver maiores velocidades. A função Senoidal demonstrou ser mais suave que a elíptica, pois tanto os gráficos de velocidade como de aceleração demonstraram ter menores variações e, portanto, menores impactos envolvidos.

6. AGRADECIMENTOS

À FAPESP (Proc. 99/09602-4) e à CAPES pelas bolsas concedidas.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- Bay, J.S. and Hemami, H., 1987, "Modeling of a neural pattern generator with coupled nonlinear oscillators". IEEE Transactions on Biomedical Engineering, Vol. 34, no. 4
- Cheng, M. Y. and Lin, C. S., 1997, "Genetic algorithm for control design of biped locomotion", Journal of Robotics Systems, 14(5), pp. 365-373.
- Frank, A. A., 1968, "Automatic control systems for legged locomotion machines", Ph.D. Thesis, University of Southern California.
- Furusho, J. and Masubuchi, M., 1986, "Control of a dynamical biped locomotion system for a steady walking", J. of Dyn. Systems Meas. And Control. Transaction of ASME, 108, 111-118.
- Lum, H.K., Zribi, M. And Soh, Y.C., 1999, "Planning and control of a biped robot". International Journal of Engineering Science. Vol. 37, pp. 1319-1349.
- Kato, I., *et al.*, 1974, "Information-power machine with senses and limbs" in 1st CISM-IFTToMM Symp. On Theory and Practice of Robots and Manipulators, Springer – Verlag.
- Kato, R. and Mori, M., 1984, "Control method of biped locomotion giving asymptotic stability of trajectory". Automatica, vol. 20, no. 4, pp. 405-414.
- McGeer, T., 1990, "Passive walking with knees". Proceedings of the IEEE Conference on Robotics and Automation, vol. 2, pp. 1640-1645.
- Mita, T., *et al.*, 1984, "Realization of high speed biped using modern control theory". International J. Control, vol. 40, no. 1, 107-119
- Popovic, D., *et al.*, 1999, "Optimal control of walking with functional electrical stimulation study". IEEE Transactions on Rehabilitation Engineering, Vol. 7, no. 1, March.
- Raibert, M.H., 1996, "Legged Robots that balance". MIT Press, Cambridge, MA.
- Vukobratovic, M., *et al.*, 1990, "Scientific fundamentals of robotics 7. Biped locomotion: Dynamics stability, control and application", Springer – Verlag.

AUTOMAÇÃO E CONTROLE DE UM SISTEMA DE POSICIONAMENTO SUBMICROMÉTRICO

Gilva Altair Rossi de Jesus

Carlos Alberto Martin

Universidade Federal de Minas Gerais, Departamento de Engenharia Mecânica, 31270-901,
Belo Horizonte-MG, Brasil. E-mail : gilva@vesper.demec.ufmg.br

Resumo

Este artigo apresenta a automação e a técnica de controle de posição empregada em um sistema de posicionamento que utiliza um módulo CNC (comando numérico computadorizado) de arquitetura semi-aberta. É empregada a estratégia de controle PI-D+*feedforward* de velocidade e aceleração no controle da posição e a técnica de posicionamento de pólos é empregada na sintonia do controlador de tal modo a se obter um erro de posicionamento na ordem submicrométrica. A automação de importantes eventos na planta é realizada de forma que se tenha uma operação segura do sistema de posicionamento, o qual, futuramente, irá compor um torno de ultra-precisão. Ensaios de seguimento de trajetória realizados no sistema de posicionamento são apresentados

Palavras-chave: Sistema de controle, controlador PID, controlador *feedforward*, posicionamento de pólos, posicionamento de precisão.

1. INTRODUÇÃO

A crescente necessidade de componentes com qualidade superficial nanométrica e incerteza dimensional submicrométrica vem impulsionando pesquisas na área de desenvolvimento de máquinas-ferramentas de ultra-precisão. Com o emprego da tecnologia de ultra-precisão obtém-se componentes tais como cilindros para fotocopiadoras, lentes especiais para câmaras fotográficas, espelhos planos utilizados em sistemas de medição laser, etc., os quais devem possuir incerteza dimensional entre 0,2 μm até 10 nm e rugosidade menor ou igual a 25 nm (Schroeter, 1997). A usinagem de ultra-precisão, capaz de produzir componentes com tais características, emprega ferramentas de diamante monocristalino de elevada precisão em máquinas-ferramenta especialmente projetadas para esta função e sob condições controladas de temperatura, umidade e meio ambiente.

Uma máquina-ferramenta de ultra-precisão requer um sistema de posicionamento com uma estrutura eletro-mecânica capaz de proporcionar erro de posicionamento (posição desejada menos posição real - quando o posicionar está parado) na ordem submicrométrica e erro de seguimento (posição desejada menos posição real - quando o posicionador está em

movimento) também submicrométrico. Na seção 2 deste artigo é apresentada a estrutura eletro-mecânica do sistema de posicionamento que, futuramente, irá compor um torno de ultra-precisão. Na seção 3 são apresentados o módulo de controle CNC, a implementação dos itens de segurança e o sistema de controle de posição. A técnica de posicionamento de pólos empregado em sintonia do controlador é mostrada na seção 4 e por fim, alguns ensaios são apresentados.

2. O SISTEMA DE POSICIONAMENTO ZX

O sistema de posicionamento é composto por dois eixos com estrutura eletro-mecânicas diferentes. No eixo Z do posicionador, mostrado na figura 1(a), são utilizadas duas guias cilíndricas aerostáticas e um servo-motor translativo CC sem escovas (De Jesus, 1999). Com isso, obtém-se um eixo cujo movimento será sem contato mecânico entre parte móvel (carro) e fixa, e conseqüentemente, com atrito desprezível. Além disso, dispensa o uso de conversores de movimento rotacional para translacional e de acoplamentos, de modo que o eixo fica livre de problemas como folgas e elasticidade e com isso a rigidez entre motor e eixo é maior, o que permite obter respostas mais rápidas e melhor precisão. Como sistema de medição utilizou-se uma escala opto-eletrônica com resolução de 50 nm. Quando energizado, o motor gera uma força que irá fazer com que o carro se deslize sobre as guias cilíndricas aerostáticas. A escala opto-eletrônica, presa ao carro, irá realimentar o valor medido de posição para o controlador de posição.

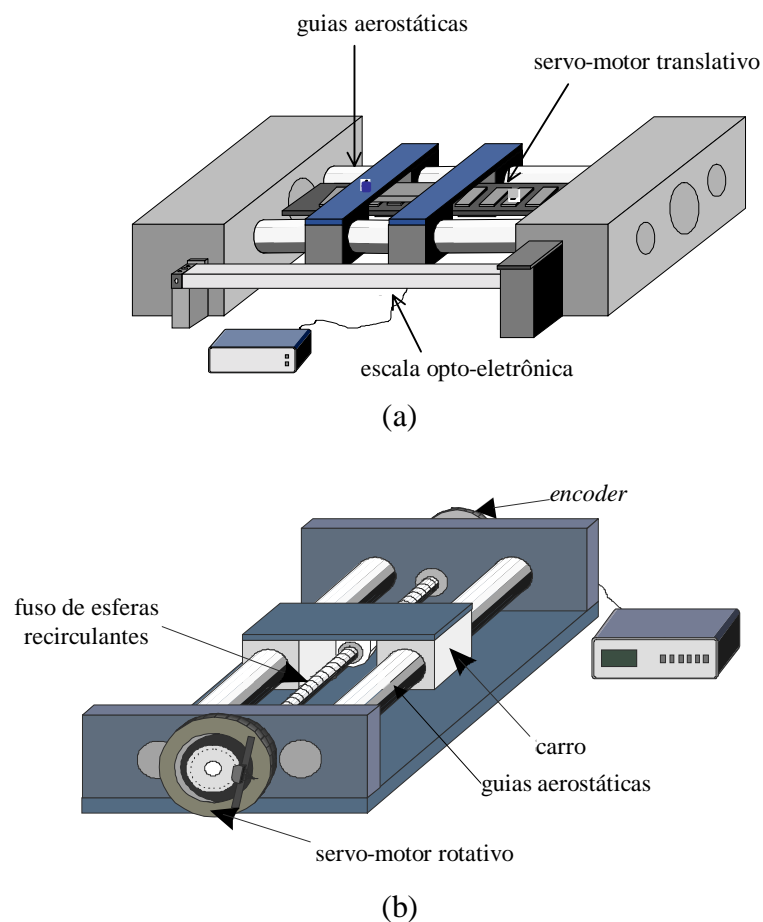


Figura 1 : (a) Eixo Z e (b) Eixo X

No eixo X, mostrado na figura 1(b), também foram utilizadas as guias aerostáticas, no entanto, optou-se por um servo-motor sem escovas CC, rotativo e sem carcaça, montado diretamente no fuso de esferas recirculantes de precisão. O servo-motor aciona o fuso, que através da castanha, transmite o movimento translativo a um carro sustentado por um par de guias aerostáticas. É utilizado um *encoder* que provê uma resolução linear de 0,77 nm. A figura 2 apresenta o esquema dos dois eixos sobrepostos. A estrutura eletro-mecânica do eixo Z permite avaliar a eficiência de servo-motores translativos neste tipo de aplicação e comparar com uma estrutura mais conhecida, ou seja, eixos que empregam servo-motores rotativos juntamente com fusos de esferas recirculantes (Slocum, 1992).

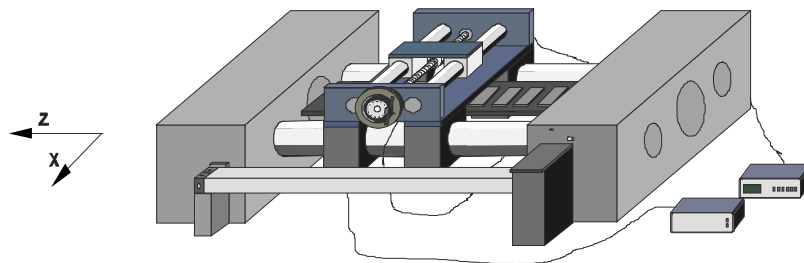


Figura 2 : Sistema de Posicionamento - Eixo Z e X

Para monitorar o sistema de posicionamento e controlar a posição de cada um dos eixos é utilizado um módulo de controle CNC, o qual possui o seu próprio processador aritmético (DSP – *Digital Signal Processor*) e banco de memórias.

3. O MÓDULO CNC

O módulo CNC tem como tarefas principais : (1) converter as informações de geometria espacial (comandos de entrada) em comandos para os dispositivos de acionamento (comandos de saída); (2) controlar a posição e velocidade dos eixos; (3) monitorar variáveis medidas na máquina; (4) e gerar perfis de trajetórias com diferentes níveis de complexidade (interpolação linear, circular, cúbica e spline).

A figura 3 mostra o fluxo principal de sinais entre o módulo CNC e a planta, e ilustra suas principais funções. O programa-peça, desenvolvido pelo usuário de acordo com a trajetória desejada, ao ser executado, passa ao algoritmo de controle o valor da referência de posição desejada. O algoritmo de controle simultaneamente recebe o valor da posição medida. Baseado na diferença entre estes dois valores (erro de posição), o algoritmo de controle irá calcular o valor do sinal de comando para o respectivo acionamento. Simultaneamente, o módulo CNC monitora a pressão das guias aerostáticas, limites de curso, emergência e falha do servo-conversor e poderá desabilitar o acionamento do eixo Z ou X se houver necessidade.

É importante lembrar que os comandos de posição somente serão executados com exatidão se o sistema de controle do eixo envolvido estiver corretamente ajustado (ou seja, de forma a minimizar o erro de posicionamento), pois é função do algoritmo de controle de posição fazer com que a posição medida seja igual à posição de referência. No item 4 será mostrado uma técnica de ajuste do sistema de controle.

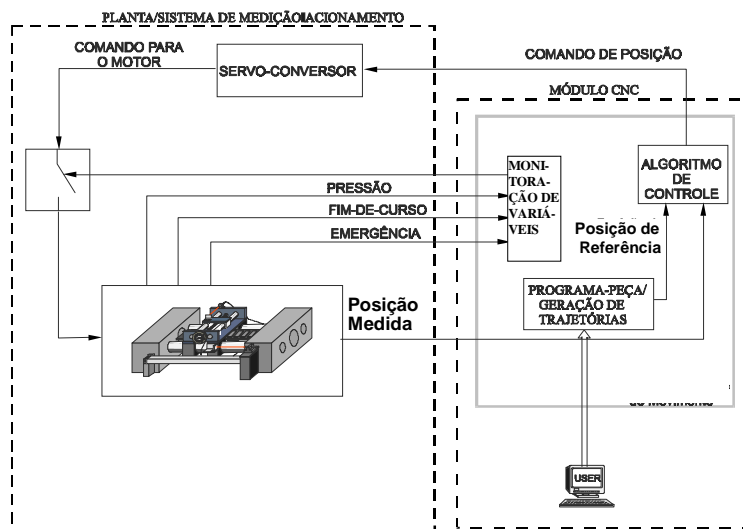


Figura 3 : Fluxo de Sinais entre Módulo CNC e Posicionador

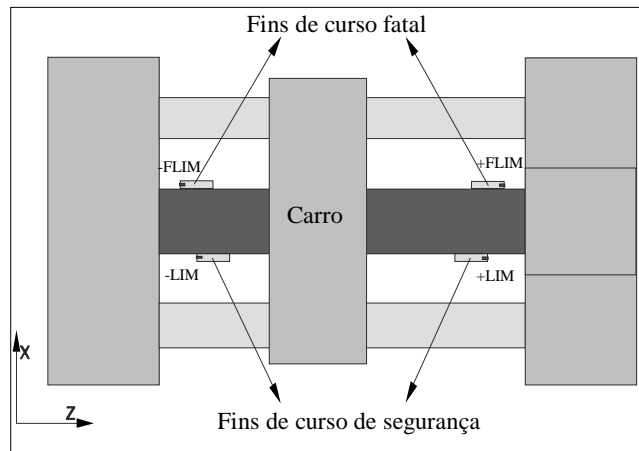
3.1 - Monitoração de Variáveis

Antes de se colocar um eixo em operação é necessário implementar alguns itens de segurança a fim de proteger os elementos que compõem o sistema de posicionamento e garantir o seu funcionamento adequado. Os programas de monitoração supervisionam sinais provenientes do sistema de posicionamento e executam medidas de segurança sempre que necessário. A seguir é apresentada a implementação da monitoração dos fins-de-curso do eixo Z.

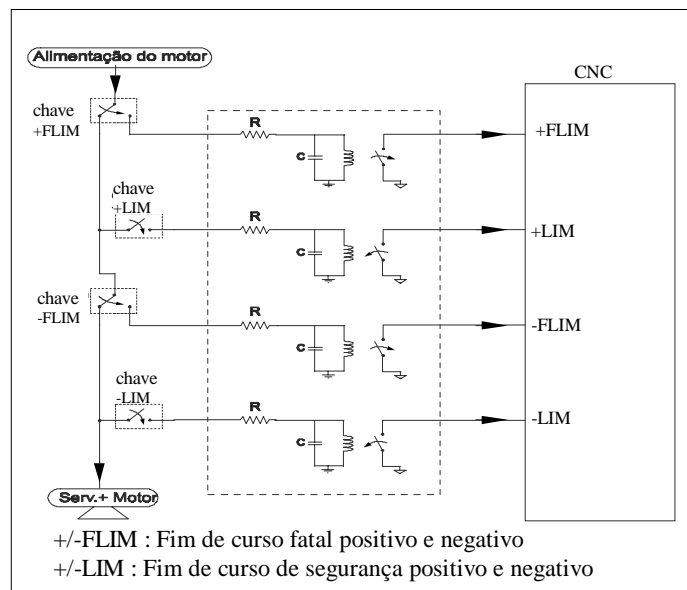
Em cada eixo foram instalados quatro interruptores de movimento, dois em cada extremidade do eixo, que irão atuar como fins-de-curso fatal e de segurança. A figura 4(a) mostra a disposição de tais interruptores no eixo Z, e um esquema semelhante à este é implementado no eixo X. Os sinais elétricos provenientes dos interruptores são conectados ao módulo CNC, através de um circuito elétrico envolvendo relés (figura 4(b)). No CNC, um programa especialmente desenvolvido monitora estes sinais.

Durante a movimentação, assim que o carro do posicionador atingir uns dos fins de curso de segurança, positivo ou negativo, o CNC irá interromper o movimento, mantendo o carro do posicionador fixo naquela posição. O operador é avisado, através de uma mensagem na tela do computador, que o fim-de-curso de segurança foi atingido. Caso ocorra do primeiro fim-de-curso ser atingido pelo carro e mesmo assim, por alguma falha, o movimento continuar, o carro irá tocar o segundo interruptor, isto é, o fim-de-curso fatal. Quando isso ocorrer, a alimentação do motor é cortada (figura 4(b)).

A pressão do ar comprimido utilizado nas guias aerostáticas também é monitorada. Quando a pressão do ar cair abaixo do valor permitido (5 bar) é acionado um circuito semelhante ao mostrado na figura 4(b), que interrompe o funcionamento do motor, além de avisar ao operador, através de uma mensagem na tela. Também para o interruptor de emergência um esquema semelhante é implementado. Desta forma, garante-se que somente quando todas as condições forem favoráveis, o posicionador irá operar. Este mesmo procedimento é repetido para os dois eixos envolvidos.



(a)



(b)

Figura 4 : (a) Fins-de-curso localizados no Eixo Z e (b) Diagrama Esquemático do Circuito de Relés

3.2 Algoritmo de Controle

A figura 5 mostra o diagrama de blocos da malha de controle de posição do eixo Z. A malha de controle do eixo X é constituída por subsistemas similares, no entanto, com outras características, conforme apresentado na seção anterior. O comportamento estático e dinâmico da resposta do sistema (posição medida), avaliado através do erro de posicionamento, rapidez de resposta e padrão de oscilação dependerá da lei de controle aplicada à planta e dos ajustes de seus coeficientes. No caso em questão, o algoritmo de controle de posição é do tipo proporcional-integral-derivativo juntamente com a ação *feedforward* (FF) de velocidade e aceleração (PI-D+FF).

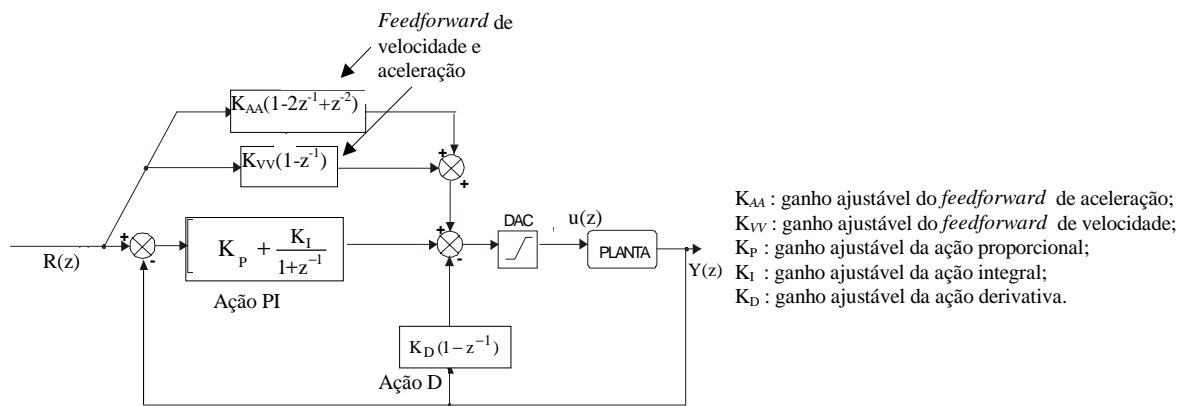


Figura 5 : Diagrama de Blocos da Malha Fechada com Controlador PI-D+FF

O ganho proporcional (K_p) influi na rapidez e conseqüentemente na rigidez e estabilidade da resposta, um ganho muito alto aumenta a velocidade de resposta do sistema, a rigidez aumenta, e o sistema fica mais próximo à instabilidade. Já o termo integral (K_i) permite reduzir o erro de regime estacionário e o termo derivativo (K_d) atua sobre o amortecimento do sistema. Uma ação derivativa muito forte poderá amplificar o ruído do sistema, e uma ação integral muito forte poderá causar um aumento na sobre-passagem passagem (Delta Tau, 1991). Além de erro de posicionamento mínimo e sobre-passagem nula, deseja-se que o sistema de posicionamento apresente uma alta rigidez, de forma a reduzir o máximo possível o aumento no erro de posicionamento devido à variação dos esforços de usinagem introduzidos pela peça (durante a usinagem) e devido às perturbações provenientes da própria máquina e do meio externo.

Na ação FF de velocidade, o ganho K_{vv} atua na redução de erros de seguimento (de posição) proporcionais à velocidade comandada, acarretados pelo amortecimento viscoso do sistema. Já a ação FF de aceleração é necessária para se reduzir os erros de seguimento proporcionais à aceleração comandada, acarretados pela inércia do sistema. Conforme mostrado na figura 5, as ações FF de velocidade e de aceleração são somadas ao comando de torque do motor, as quais são proporcionais à velocidade e aceleração comandadas, respectivamente. A velocidade comandada é obtida através da derivada da posição comandada ($R(z)$). A aceleração comandada é obtida através da segunda derivada da posição comandada.

4. SINTONIA ATRAVÉS DE POSICIONAMENTO DE PÓLOS

Obtido o modelo matemático da planta, constituída pelo servo-motor, pela mecânica do posicionador (transmissão, guias, etc.) e pelo sistema de medição de posição, é possível aplicar a teoria de controle clássico juntamente com a técnica de posicionamento de pólos a fim de se obter os ajustes do controlador ((Santina, 1996), (Ogata, 1995) e (Vásquez, 1994)). O modelo matemático da planta é obtido através do método de identificação dos mínimos quadrados não recursivos, o qual utiliza dados obtidos em ensaios realizados na planta. Para isso, o sistema em malha aberta é excitado com um sinal randômico e então, a entrada (referência de posição) e os dados de saída (posição) são usados no algoritmo dos mínimos quadrados não recursivos para se obter os parâmetros do modelo matemático (De Jesus, 1999). O modelo de cada eixo é de segunda ordem, e pode ser representado pela seguinte função de transferência :

$$G(z) = \frac{b_0 z}{z^2 + a_1 z + a_2} \quad (1)$$

onde, $G(z)$ é a função de transferência da planta, no domínio z ; b_0 , a_1 e a_2 são os parâmetros do modelo matemático identificado; e z é o operador da transformada z . A função de transferência em malha fechada é obtida através da figura 5. A equação característica, dada pelo denominador da função de transferência em malha fechada, fornece os pólos do sistema em malha fechada, ou seja :

$$\text{Eq. Caract.} = Z^3 + (a_1 + b_0 K_D - 1)z^2 + (a_2 - a_1 - b_0 K_D - b_0 K_P)z - (a_2 - b_0 K_D) \quad (2)$$

Empregando a técnica de posicionamento de pólos, pode-se impor os valores dos pólos da função de transferência da malha fechada e em seguida calcula-se quais valores de ajustes do PI-D fazem com que aqueles pólos se verifiquem. Como a equação característica (eq. 2) é de terceiro grau, o sistema possui três pólos. Sejam $(\lambda_1, \lambda_2$ e $\lambda_3)$ o conjunto de pólos desejados. Logo, a equação característica desejada pode ser escrita como :

$$\Delta(\lambda) = \lambda^3 - (\lambda_1 + \lambda_2 + \lambda_3)\lambda^2 + (\lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3)\lambda - \lambda_1\lambda_2\lambda_3 \quad (3)$$

Comparando as equações (2) e (3), tem-se que :

$$K_P = \frac{-(a_1 + b_0 K_D - a_2 + \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3)}{b_0} \quad (4)$$

$$K_I = \frac{-(a_1 - b_0 K_D + 1 + \lambda_1 + \lambda_2 + \lambda_3)}{b_0} - K_P \quad (5)$$

$$K_D = \frac{a_2 - (\lambda_1\lambda_2\lambda_3)}{b_0} \quad (6)$$

Através das equações 4, 5 e 6, calcula-se os valores dos ajustes do PI-D de forma a se obter pólos do sistema em malha fechada iguais a λ_1, λ_2 e λ_3 . Os ajustes da ação FF são feitos baseando-se na função de transferência que relaciona erro e o sinal de saída do sistema em malha fechada, a qual é obtida através da figura 5 :

$$E(z) = R(z) - Y(z) \quad (7)$$

O sinal de referência, $R(z)$, será uma rampa ou parábola, pois tais entradas compõem grande parte das possíveis trajetórias que um posicionador deve ser capaz de realizar. Aplicando o teorema do valor final na equação (7), obtém o valor do erro em regime permanente, isto é:

$$e_{SS} = \lim_{Z \rightarrow 1} (z - 1)E(z) \quad (8)$$

Para se obter e_{SS} nulo sem interferir na dinâmica do sistema, considera-se que o sistema de controle não possua ação integral e calcula-se os ganhos do FF que eliminam o erro de seguimento devido a entrada rampa e parábola. Portanto, para o controlador P-D+FF, aplicando o teorema do valor final, obtém-se o seguinte resultado para entrada rampa :

$$e_{SS} = \frac{T_S (1 - a_2 + b_0 K_D - b_0 K_{VV})}{b_0 K_I} \quad (9)$$

onde T_S é o tempo de amostragem do sistema de controle. Para se ter $e_{SS}=0$, tem-se que :

$$K_{VV} = \frac{1 - a_2 + b_0 K_D}{b_0} \quad (10)$$

Repetindo o procedimento para entrada parabólica, obtém-se :

$$K_{AA} = \frac{1}{b_0} \quad (11)$$

Em resumo, no procedimento adotado aqui, primeiro determina-se os ganhos do PI-D através da técnica de posicionamento de pólos (equações 4,5 e 6) e em seguida, determina-se os ganhos do FF que eliminam o erro devido à rampa e parábola (equações 10 e 11).

5. ENSAIOS DE POSICIONAMENTO

Inicialmente são apresentados os ensaios de posicionamento realizados no eixo Z. No ensaio mostrado na figura 6(a), o eixo Z se desloca 2 mm, permanece parado por alguns segundos e depois retorna ao ponto de partida. O gráfico mostrado na figura 6(b) mostra o erro (trajetória comandada menos a trajetória medida), o qual atinge o valor máximo de $\pm 4\mu\text{m}$. Analisando a curva de erro (figura 6(b)), nota-se que o sistema apresenta um comportamento oscilatório na partida do movimento (no tempo zero e no tempo próximo à 14 s). Tal comportamento é causado pelo ajuste inadequado do controlador PI-D+FF. A seguir são apresentados outros resultados com novos ajustes do controlador.

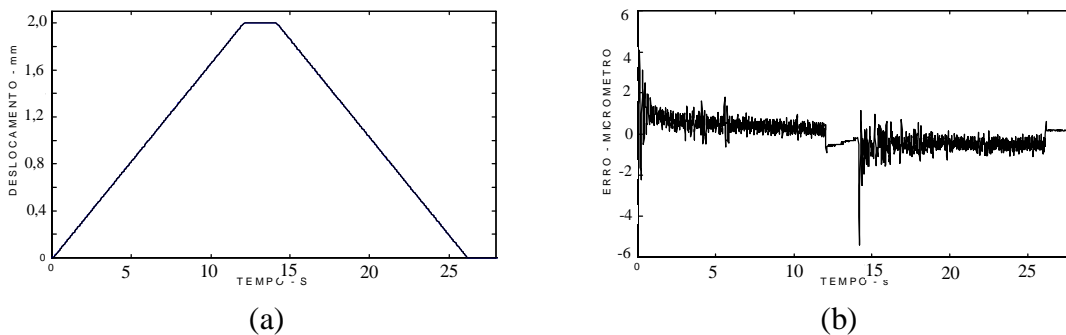


Figura 6 : Ensaio de posicionamento - Eixo Z - Velocidade de 10 mm/min

Sintonizando o controlador PI-D+FF conforme mostrado na seção 4 e repetindo o ensaio anterior, obteve-se um erro consideravelmente menor (figura 7). Os pólos são escolhidos baseado nos valores desejados de ζ (fator de amortecimento) e ω_N (frequência natural). Escolhendo o pólo dominante de tal forma que a frequência natural seja igual a 7 vezes a

frequência natural do sistema em malha aberta e ζ igual a 0,9 obteve-se um erro máximo de $\pm 0,4 \mu\text{m}$ (figura 7(b)).

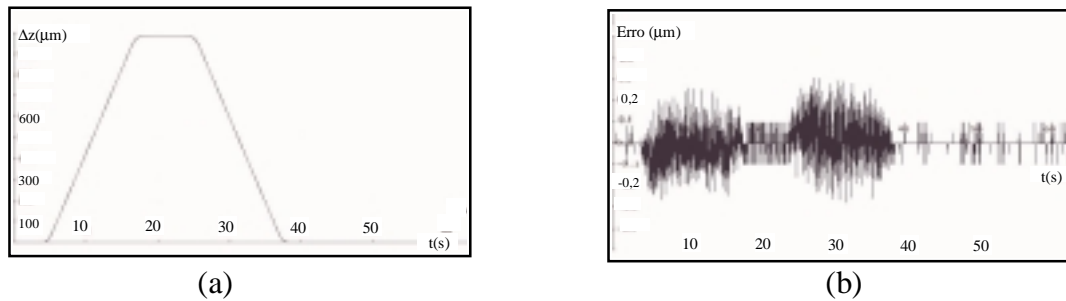


Figura 7 : Ensaio de posicionamento - Eixo Z - Velocidade de 10 mm/min

Os ensaios realizados no eixo X são apresentados na figura 8. A figura 8(a) mostra as curvas de deslocamento comandado sobreposta à curva de deslocamento medido (1) e também mostra o erro de seguimento (2). No ensaio mostrado na figura 8(a) não foi incluída a ação de controle FF (o controlador é somente PI-D) e nota-se a presença de erros elevados na partida e chegada. A figura 8(b) mostra o mesmo ensaio quando se inclui a ação de controle FF. Nota-se que o erro na partida e chegada foram reduzidos. E portanto, o erro máximo que atingia $\pm 100 \text{ nm}$, caiu para $\pm 30 \text{ nm}$ após a inclusão do FF. Nos dois casos, os pólos são escolhidos de tal forma que ζ seja igual a 0,9 e a frequência natural seja igual a 7 vezes a frequência natural do sistema em malha aberta.

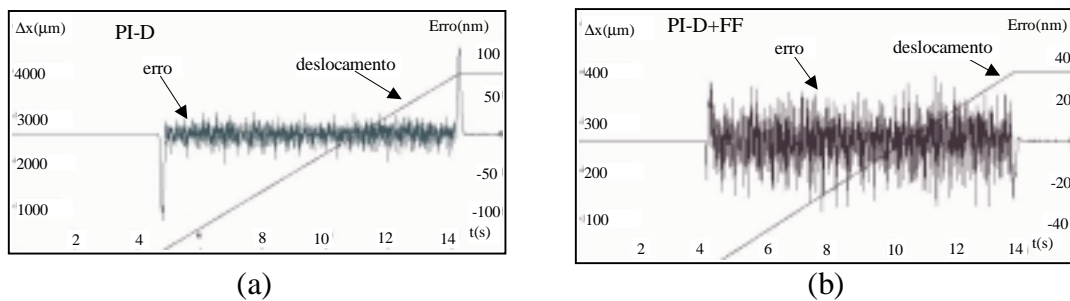


Figura 8 : Ensaio de posicionamento - Eixo X - Velocidade de 25 mm/min

6. CONCLUSÃO

Neste artigo foi apresentada a automação de itens de segurança de um sistema de posicionamento e o sistema de controle de posição do mesmo. Os ensaios apresentados foram realizados após a aplicação da teoria de controle clássico juntamente com a técnica de posicionamento de pólos na sintonia do controlador de posição tipo PI-D+FF. Utilizando este procedimento foi possível alcançar erros de posicionamento na ordem submicrométrica, além de permitir verificar, experimentalmente, o efeito de diferentes escolhas de pólos no erro de posicionamento.

7. REFERÊNCIAS

- Delta Tau Inc., “Programmable Motion Advanced Controller User’s Guide”, USA, 1991

- De Jesus, G. A. R., “Uma Contribuição para o Desenvolvimento e Controle de Sistemas de Posicionamento Submicrométricos.”, Tese de Doutorado, Depto. Eng. Mec., Universidade Federal de Santa Catarina, 1999.
- Ljung L., “The Control Handbook “, - System Identification - Editor William S. Levine - CRC Press /IEEE Press - 1996.
- Ogata, K; “Discrete-Time Control Systems”- Prentice Hall International Editions – 1995.
- Purquério, B. M., Porto, A. J. V., Jasinevicius, R. G., “O Torneamento de Ultraprecisão já é possível também no Brasil” - Máquinas e Metais, 1994, pp 106-114.
- Rao, G. S., Ro, P. I., “Submicrometer Control of a traction drive using state feedback and estimation “, Precision Engineering, 17, pp 124-130, 1995.
- Santana, M.S.; Stubberud A.R.; Hostetter, G.H., “Design Methods for Discrete-Time, Linear Time-Invariant Systems “ - The Control Handbook - Editor William S. Levine - CRC Press /IEEE Press - 1996.
- Schei, T. S., “Automatic Tuning of PID Controllers Based on Transfer Function Estimation”, Automatica, Vol. 30, No.12, 1994.
- Schroeter, R. B.; “Usinagem de Ultraprecisão de Elementos Ópticos Transmissivos no Espectro Infravermelho” - Tese de Doutorado - UFSC - 1997.
- Slocum, A. H. . “Precision Machine Design”. Prentice Hall, 1992.
- Vásquez, J. A. R., “Desenvolvimento de um Posicionador de Ultra-Precisão”. Dissertação de Mestrado - UFSC - 1994.

THEORY AND EXPERIMENTAL RESULTS ON MAP BUILDING USING LASER SCANNERS AND AN AUTONOMOUS MOBILE ROBOT – SmartROB-2

Marcelo Becker

State University of Campinas, School of Mechanical Engineering, Department of Mechanical Design, PO Box 6051, 13089-970, Campinas – SP, Brazil, Email: becker@fem.unicamp.br

Felix Wullschleger

Swiss Federal Institute of Technology, Institute of Robotics, ETH Center - IfR, CLA H-15.1, CH-8092, Zurich, Switzerland, Email: wullschl@ifr.mavt.ethz.ch

Abstract:

This article relates researches on Map-Building technique for mobile robots in unknown environments. Map-Building consists on: through sensors data, extract features from the environment and build a map that will be used by the robot to self-localize, plan paths and avoid obstacles in this environment. Our approach uses an algorithm for line and segment extraction and a simplified EKF (Extended Kalman Filter) for localization. We implemented this approach on a real platform, the mobile robot SmartROB-2, equipped with two SICK Laser Scanners and the real-time system XOberon. The results obtained were encouraging, nevertheless some problems were found due to the simplifications adopted. Solutions have been discussed with other researchers and a new approach is in progress.

Keywords: Map-Building, Stochastic Map, Laser Scanners, and Mobile Robots.

1. INTRODUCTION

Autonomous vehicles are programmable systems performing a multitude of tasks. Today, they are intended for material handling, transportation, decontamination, fire fighting, rescuing and many other hazardous activities. Therefore, the basic problem to produce an autonomous vehicle is a robot navigation problem that can be summarized by the following three questions: “Where is the vehicle?”, “Where is it going?” and “How should it get there?” (Leonard & Durrant-Whyte, 1992). The first question is one of localization, the second and third ones are essentially those of specifying a goal and being able to plan a path that results in achieving this goal, avoiding any obstacle that is in its path (path planning and obstacle avoidance). As the answer of the first question is a precursor to obtain the path planning and obstacle avoidance behaviors in autonomous vehicles, our paper focus this problem.

A priori model maps are rarely available, costly to obtain, and when they are available, they usually introduce inaccuracies in the planning tasks (Castellanos *et al.*, 1999). An automatic construction of the map of the environment, in which the robot navigates, using the robot itself would be desirable. Previous work in navigation for mobile robots has tended to treat the problems of localization, obstacle avoidance, and map building in isolation. Nevertheless, since the beginning of 90's years, one of the major motivations for the researchers in this area is the unified approach to navigation, i.e. simultaneous map building and localization. Such procedure is desirable because avoid rely on odometry or hand measuring of sensing locations, nevertheless has a high computational price, due to the necessity of maintain all environment previous observations during the exploration phase.

2. SIMULTANEOUS MAP-BUILDING AND SELF-LOCALIZATION

The aim of Map-Building is to produce and update autonomously maps of environments. This map will be used by the robot for self-localization in the environment without the need of modifying the environment (e.g. use of artificial beacon systems, code bars on the walls, etc.). To build autonomously a map of an environment consists basically on the following steps: Get data of the environment through some kind of sensor (e.g. optical sensors, ultrasonic, sonar, and laser); extract features from the sensory data; match observed features to the best fitting equivalent from the map; localize the robot with the matched features; match features with their re-prediction based on the robot new position; and, update the map features with their matched re-observation and store unmatched features as new ones (see Figure 1).

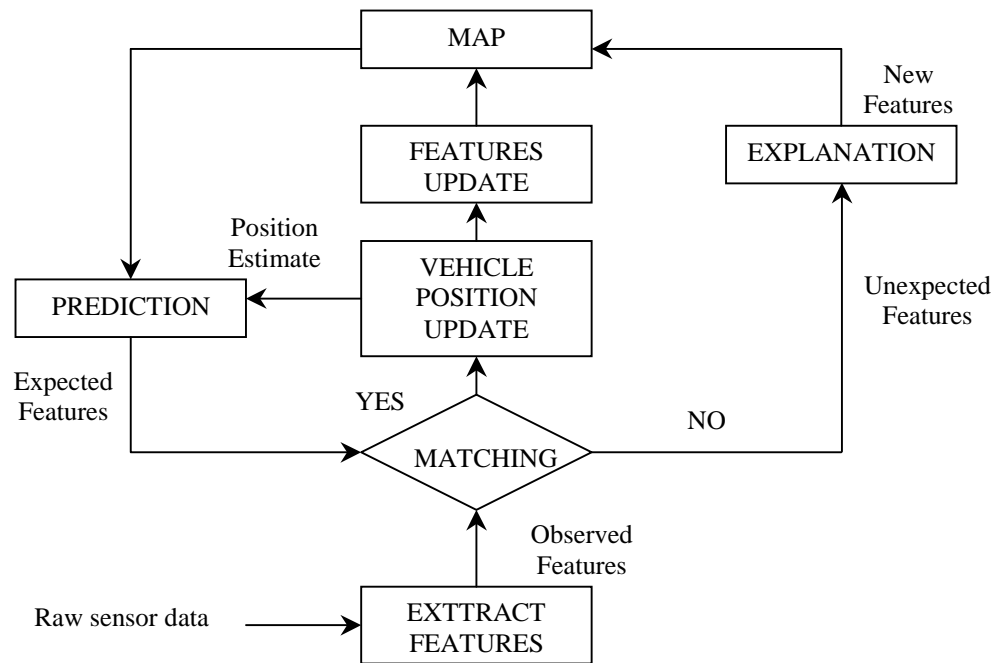


Figure 1 – Block diagram for simultaneous map building and localization.

There are many different approaches used to estimate the position of the robot during the map-building procedure. In this work, the basic tool that we used to localize the robot while it is building the map was the extended Kalman filter (EKF). Kalman filtering techniques have been used extensively to solve problems in localization, not only in robotic applications, but also, for other applications of navigation (for more details about EKF, see: Crowley, 1989, Leonard & Durrant-Whyte, 1992, and Grewal & Andrews, 1993).

The representation of the spatial information adopted was the called *stochastic map*. This representation is an exact mathematical approach to the simultaneous localization and map building problem, originally addressed by Smith *et al.* (1990), and contains the estimates of relationships among objects in the map, their uncertainties, and their inter-dependencies, given by all the available information provided by the sensors. In order to aid the comprehension of the EKF, the stochastic map, and the procedure and simplifications adopted in this work, the following terms are defined. A *spatial relationship* is represented by the vector of its *spatial variables*, x . Then, the position of the robot can be described by its coordinates, x and y , in a two dimensional Cartesian reference frame and by its orientation, ϕ , given as a rotation about the z axis:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} \quad (1)$$

An *uncertain* spatial relationship can be represented by a *probabilistic distribution*:

$$P(\mathbf{x}) = f(\mathbf{x}) d\mathbf{x} \quad (2)$$

The uncertain spatial relationships are modeled by estimating the first two moments of its probability distribution, i.e. the mean, $\hat{\mathbf{x}}$, and the covariance, $C(\mathbf{x})$, defined as:

$$\begin{aligned} \hat{\mathbf{x}} &\stackrel{\Delta}{=} E(\mathbf{x}) \\ \tilde{\mathbf{x}} &\stackrel{\Delta}{=} \mathbf{x} - \hat{\mathbf{x}} \\ C(\mathbf{x}) &\stackrel{\Delta}{=} E(\tilde{\mathbf{x}} \tilde{\mathbf{x}}^T) \end{aligned} \quad (3)$$

Where E is the expectation operator, and $\tilde{\mathbf{x}}$ is the derivation from the mean. For the mobile robot, these terms are:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\phi} \end{bmatrix} \quad C(\mathbf{x}) = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\phi} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\phi} \\ \sigma_{x\phi} & \sigma_{y\phi} & \sigma_\phi^2 \end{bmatrix} \quad (4)$$

The diagonal elements of the covariance matrix are just the variances of the spatial variables, while the off-diagonal elements are the covariances between the spatial variables. To model a system of n uncertain spatial relationships, it is necessary to construct the vector of all spatial variables, which is called as the *system state vector*. As before, the mean of the mean of the state vector, $\hat{\mathbf{x}}$, and the *system covariance matrix*, $C(\mathbf{x})$, are estimated.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix} \quad (5)$$

$$\begin{aligned} \text{Defining: } C(x_i, x_j) &\stackrel{\Delta}{=} E(\tilde{x}_i \tilde{x}_j^T) \\ C(x_j, x_i) &= C(x_i, x_j)^T \end{aligned} \quad (6)$$

$$C(\mathbf{x}) = \begin{bmatrix} C(x_1) & C(x_1, x_2) & \cdots & C(x_1, x_n) \\ C(x_2, x_1) & C(x_2) & \cdots & C(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ C(x_n, x_1) & C(x_n, x_2) & \cdots & C(x_n) \end{bmatrix} \quad (7)$$

Here, the x_i 's are the vectors of the spatial variables of the individual uncertain spatial relationships, and the $C(x_i)$'s are the associated covariance matrices. The $C(x_i, x_j)$'s are the cross-variance matrices between the uncertain spatial relationships. These off-diagonal sub-matrices encode the dependencies between the estimates of the different spatial relationships.

To implement the map-building procedure, we adopted some simplifications, which reduced the computational requirements. The results rely on the so-called 2-D assumption, that is, the environment can be sufficiently described in 2-D since all observed structures keep their forms in the vertical dimension. Supposing that every horizontal slice of an obstacle has a closed contour, the goal of the exploration is to build a closed chain of extracted features around each contour in the environment. Open ends of a chain, i.e. the end of a feature, where the successive one is still missing, indicate a frontier where new information can be gathered. The final map is a collection of all chains of features observed during the exploration of the environment.

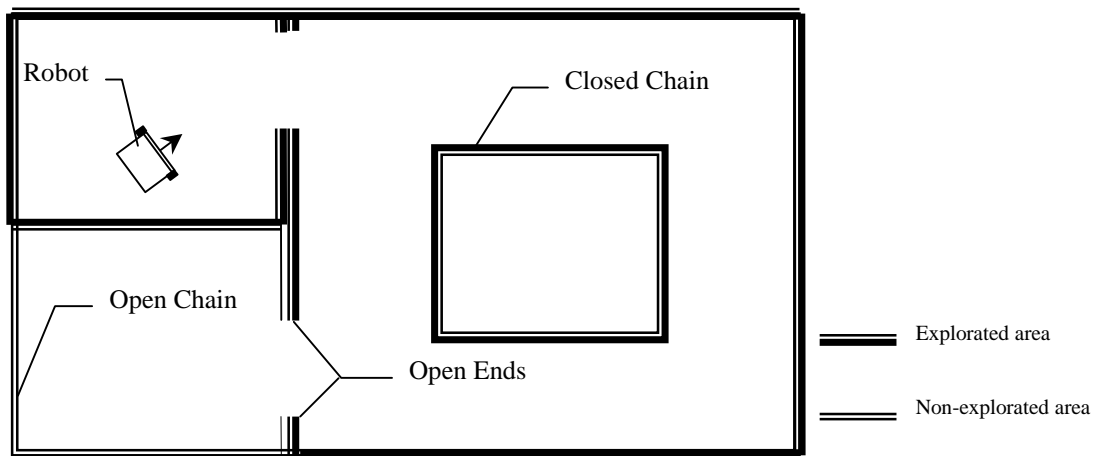


Figure 2 – Simple scene showing two chains (a closed and an open one) and the robot during the exploration phase. Observe that the Open Ends indicate the borders of the known environment.

The line extraction method (Arras, 1996, Wulschleger, 1999, and Wulschleger *et al.* 1999) is also used. This procedure delivers lines and segments together with their first order covariance estimate. The distinction of lines and segment is done as follow: A line is represented according to the Hessian model:

$$\rho \cdot \cos(\varphi - \alpha) - r = 0 \quad (8)$$

Where (ρ, φ) is the raw measurement in polar coordinates and (α, r) the model parameters. They come along with their second moments, which hold the propagated uncertainty from the raw data level. Segments have a 4-D representation. Either by Cartesian coordinates of their endpoints or by a position, an inclination, and a length. In a hierarchical order, segments are below lines since they lie on a line (the supporting line) and thus have the same (α, r) pair. In all comparisons, matchings, and updates of map entities pairs of (α, r) with their covariance matrix are used. Although this representation provides consistent treatment of all feature model parameters and thus permits blind comparison and manipulation of features, reducing the computational requirements, this choice is a sub-optimal one. This occur because we are simplifying the EKF, i.e. all the cross-variance matrices were neglected (the cross-correlation's between robot to feature and feature to feature are not considered). If in a way it was possible to save computational time, in the other way, some problems during the matching and updating procedures happened.

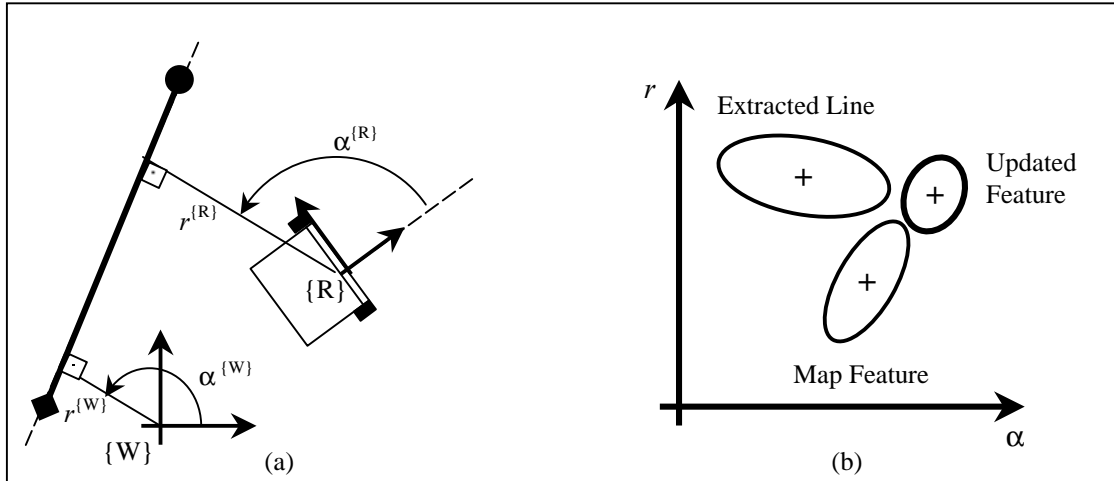
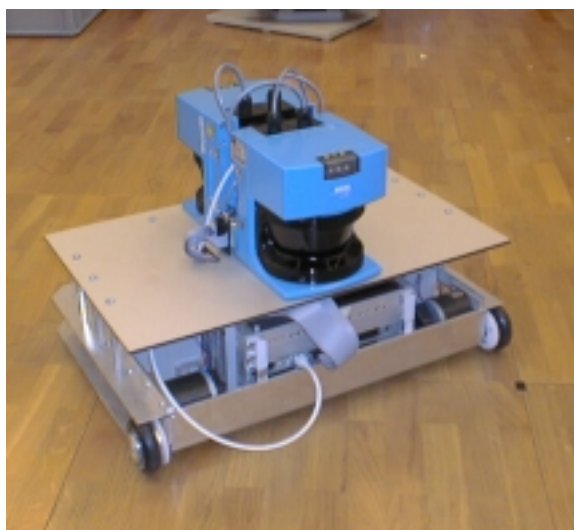


Figure 3 – A line (a) is represented by a Gaussian distribution in model space (b) with heading α and distance r as first moments.

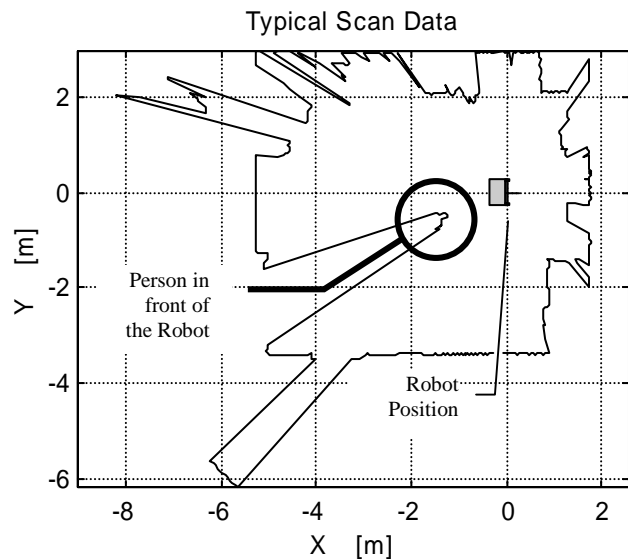
As all operations (e.g. matching, fusing, update, etc) are done in the model space (Figure 3-b), the use of a simplified EKF causes an increment on the position uncertainties, deformation, rotation, and translation of the observed features, when compared with the real ones (Castellanos *et al.*, 1999), and even creates “ghost features”.

3. IMPLEMENTATION AND RESULTS

The implementation of the Map-Building procedure used a real platform, a differential drive mobile robot - SmartROB-2, whose dimensions were 0.60 m width x 0.35 m length x 0.40 m height (see Figure 4-a and Badreddin, 1992). The mobile robot was equipped with two SICK LMS200 Laser Scanners covering 360° with a resolution of 2° and radial error measurement less than ± 20 mm. The entire code was written in XOboron, a deadline-driven hard real-time operating system. For more informations about Laser Ssensors and data processing, see Adams (1999).



(a)



(b)

Figure 4 – (a) Photo of the mobile robot SmartROB-2 equipped with two SICK LMS200 Laser Scanners and (b) plot of a typical laser scan data obtained by the robot during the experimental tests.

The sequence presented on Figure 5 shows the real exploration of the SmartROB-2 in an environment built in the laboratory. The lines “leaving” the SICKs laser scanners represent the matched laser scanner data and, the walls, the extracted features (remember the 2-D assumption). The robot assumed that its initial position is the position of the global reference frame $\{W\}$ and started to move around to explore the environment, following the wall and trying to “close” all chains of the environment. The exploration ended on Step 18. It is possible to observe that on Step 15 there is a big area of non-matched data, i.e. signals of the rear laser scanner that the EKF could not match with previous observations.

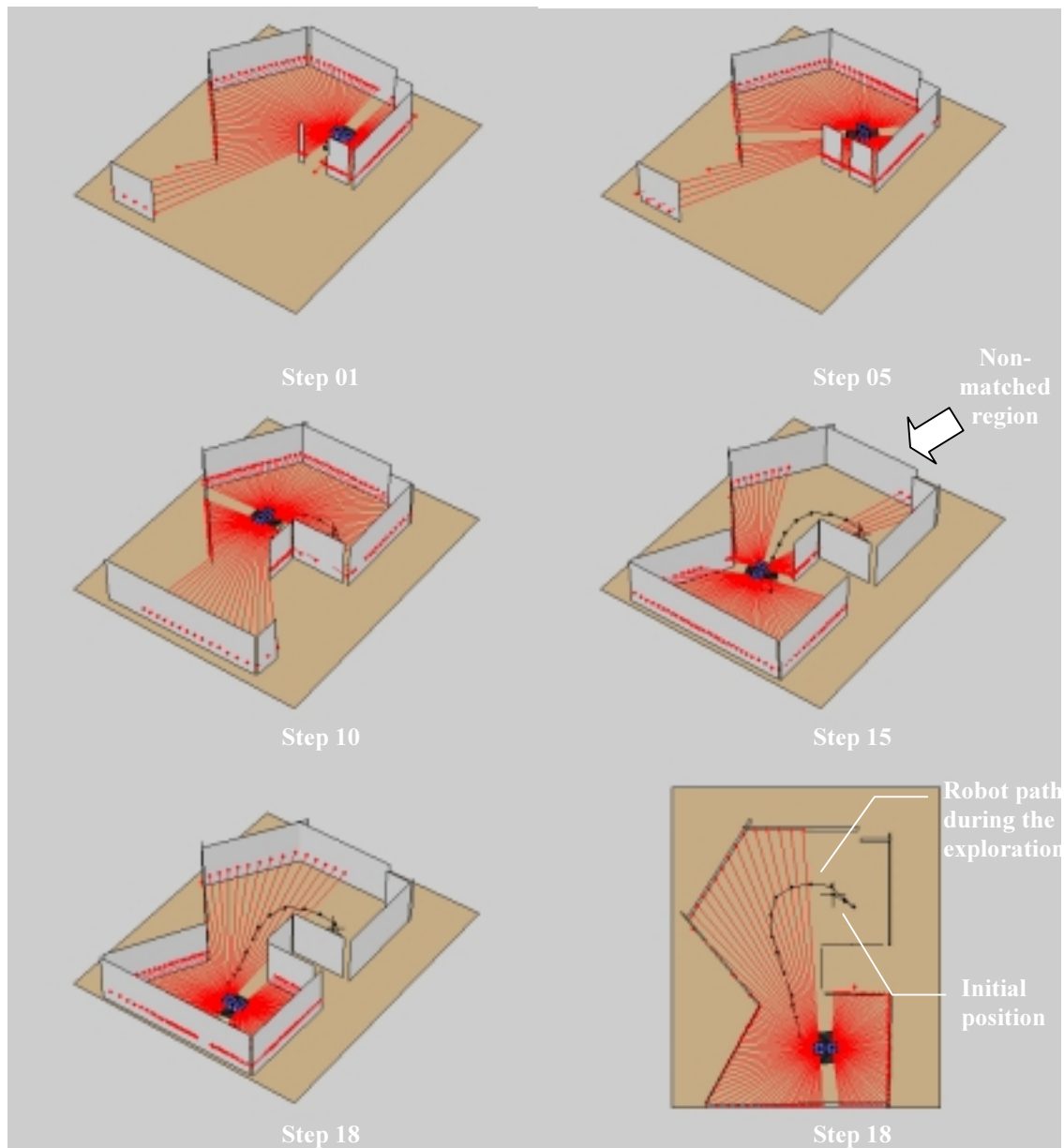


Figure 5 – Sequence of steps during a real exploration of an environment. See that in step 15, there is a big non-matched region behind the robot.

The Figure 6 shows an example of a more complex environment built in the laboratory, where the simplifications of the EKF produced “ghost features”. Figure 7 shows the matched walls and the position uncertainty of the robot during the exploration phase. Observe in Figure 7 that only 3 walls were matched (18, 40, and 43). The use of the complete EKF would solve these problems.

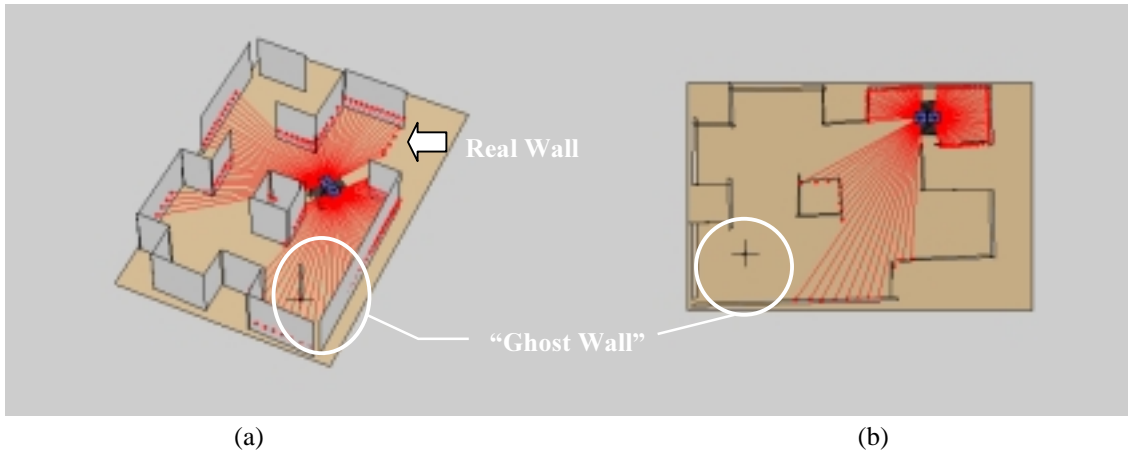


Figure 6 – (a) Detail of the “ghost wall” during the exploration of the environment and (b) the final map of the environment, the “ghost wall” was not deleted from the map.

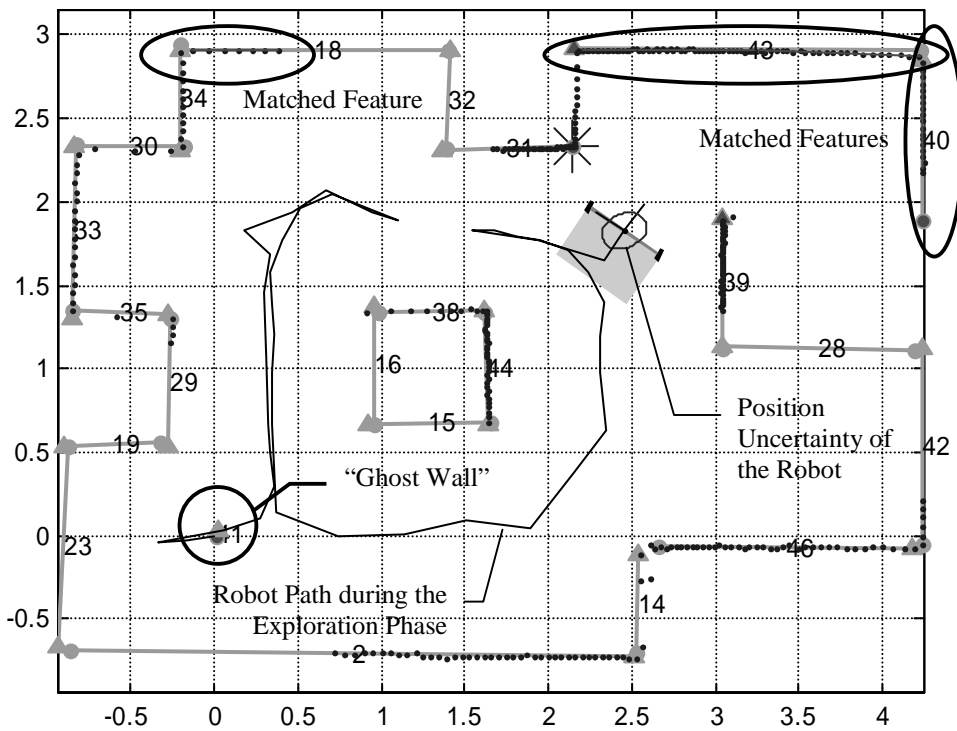


Figure 7 – Details of an exploration step using the simplified EKF when the robot was driving to explore the non-closed area. The dots represent the raw data (scan data).

5. CONCLUSIONS AND OUTLOOK

Researches on Map-Building approach for mobile robots were developed. Our approach uses an algorithm for line and segment extraction and a simplified EKF (Extended Kalman Filter) for localization. We implemented this approach on a real platform, the mobile robot SmartROB-2, equipped with two SICK Laser Scanners and the real-time system XOberon. The entire framework was developed such that a fully autonomous system can execute the Map-Building procedure without any off-board, off-line or post-processing procedures to get a practical environment model.

The results obtained were encouraging, nevertheless some problems were found due to the simplifications adopted. To save computational time we used a simplified EKF and all the cross-variance matrices were neglected. This option caused an increment on the position uncertainties and thus, difficulties to match the sensor data. Solutions have been discussed with other researchers and we have some options to overcome the problems found: the implementation of the complete EKF, or the use of other kind of Filter for localization (e.g. the Condensation Algorithm, Meier & Ade, 1999).

ACKNOWLEDGMENTS

The authors wish to express their thanks to Prof. Gerhard Schweitzer and all the staff of the Institute of Robotics, ETH Zurich (Switzerland) for their contribution to this paper. The CNPq - National Research Council (Brazil) and the ICSC - World Laboratory (Switzerland) funded this study.

REFERENCES

- Adams, M. D., 1999, "Sensor Modeling, Design and Data Processing for Autonomous Navigation", World Scientific Series in Robotics and Intelligent Systems, Vol. 13, p.232.
- Arras, K. O., 1996, "Map Building I", Diplomarbeit WS'95/96: IfR – ETH Zurich, 1996.
- Badreddin, E; 1992, "Mobile Roboter – Einführung in den Aufbau und die Funktionsweise Mobiler Roboter", Kurs, June 15 – 16, 1992, IfR – ETH Zurich, Switzerland.
- Castellanos *et al.*, 1999, "The SPMAP: A probabilistic framework for simultaneous localization and map building", IEEE Trans. on Robotics and Automation, Vol. 15, n. 5, pp. 948 – 952.
- Crowley, J: L., 1989, "World modeling and position estimation for a mobile robot using ultrasonic ranging", IEEE J. Robotics and Automation, pp. 674 – 680.
- Grewal, M. S. & Andrews, A. P., 1993, "Kalman filtering: theory and practice", Prentice Hall, 1993.
- Leonard, J. J. & Durrant-Whyte, H. F., 1992, "Directed sonar sensing for mobile robot navigation", Kluwer Academic Publishers, 1992.
- Meier, E.B. & Ade, F., 1999, "Using the condensation algorithm to implement tracking for mobile robots", Proc. of the 3rd European Workshop on Advanced Mobile Robots (Eurobot'99), September 6 - 8, 1999, Zurich, Switzerland, pp. 73 - 80.
- Smith, R. *et al.*, 1990, "Estimating uncertain spatial relationships in robotics", In: Autonomous Robotics Vehicles, editors: I. J. Cox and G. T. Wilfons, pp. 167 – 193.
- Wullschleger, F. H., 1999, "Map Building III", Diplomarbeit WS'98/99: IfR – ETH Zurich, 1999.
- Wullschleger, F.H. *et al.*, 1999, "A flexible exploration framework for map building", Proc. of the 3rd European Workshop on Advanced Mobile Robots (Eurobot'99), September 6 - 8, 1999, Zurich, Switzerland, pp. 49 - 56.

PATH PLANNING APPROACH FOR AUTONOMOUS AND ASSISTED NAVIGATION USING VELOCITY OBSTACLE PRINCIPLES

Marcelo Becker

State University of Campinas, School of Mechanical Engineering, Department of Mechanical Design, PO Box 6051, 13089-970, Campinas – SP, Brazil, Email: becker@fem.unicamp.br

Felix Wullschleger

Swiss Federal Institute of Technology, Institute of Robotics, ETH Center - IfR, CLA H-15.1, CH-8092, Zurich, Switzerland, Email: wullschl@ifr.mavt.ethz.ch

Abstract:

This article relates researches on Path Planning and Obstacle Avoidance techniques for autonomous and assisted navigation in real environments. To test our approach, simulations, using the software Matlab, and experimental tests, using the mobile robot SmartROB-2 equipped with two SICK Laser Scanners and the real-time system XOberon, were done. The results obtained were excellent, a new approach using the velocity obstacle technique was developed, and real tests using the mobile robot, and path planning controller are in progress.

Keywords: Path planning, obstacle avoidance, velocity obstacle, and autonomous and assisted navigation.

1. INTRODUCTION

Path Planning and Obstacle Avoidance approaches consist, respectively, on an algorithm that plans a path that results in achieving a given goal, and another one that allows the controller to detect and track obstacles, generating a new path. Actually, these techniques are being applied in real applications like: robot manipulators avoiding moving obstacles, intelligent vehicles negotiating freeway traffic, etc. This paper focus the problems found in applications similar to intelligent vehicles.

In the case of autonomous navigation of a vehicle, the obstacle avoidance algorithm generates the steering and velocity commands for the vehicle, guiding it to its goal, taking any necessary action to avoid collisions during the path. In the case of assisted navigation, the final decision is always took by the user and the controller should aid the user, providing suggestions of alternative paths to avoid crashes with obstacles. This kind of control is found mainly in applications where the interaction between man-machine is very close and the independence feeling of the user is very important (e.g. intelligent wheelchairs, assisted navigation systems for automobiles, etc.).

In a general manner, Path Planning can be divided into two main approaches: global path planning, based on a priori complete information about the environment, and local path planning, based on sensory informations in uncertain environment where the size, shape and location of obstacles are unknown (Beom & Cho, 1995). Global path planning methods can solve the path planning problems for completely known environments. But, they can not be used for navigation in complex and dynamically changing environments, where unknown obstacles may be located an *a priori* planned path. To overcome these difficulties, methods considering real-time environment informations from sensors must be used. Local path

planning methods uses informations of the sensors to provide environmental information for the vehicle's navigator.

The navigator is a navigation and obstacle avoidance controller that generates the steering and velocity commands for the autonomous vehicle. Based on sensor readings, the vehicle should be able to perform local path planning and to take appropriate control actions. Conflict can appear, e.g.: go to goal position vs. obstacle avoidance. There are many approaches used to solve the Path Planning problem: Visibility Graph (Fu & Liu, 1990); Pattern Recognition (Wang & Tsai, 1991); Feedback Solutions (Feng & Krogh, 1991); Deformable Virtual Zones (Zapata *et al.*, 1994); Fuzzy Sets and Neural Networks (Beom & Cho, 1995 and Baxter & Bumby, 1995); Step-by-Step Planning (Bobyry & Lumelsky, 1999), and many others.

Our approach is based on the Velocity Obstacle approach (Prassler *et al.*, 1999a, b and Fiorini & Shiller, 1998). Nevertheless, we opted for to keep the modulus of the velocity constant during the path, changing its value only if a predetermined maximum steering angle is exceed. This choice generates a new approach that can be used in autonomous or assisted applications.

2. LASER SCANNERS

We decided to use laser range finders (or simply Laser Scanners), although the high costs of this kind of sensor, because actually no other sensor is equally suited for detecting and tracking a large number of moving objects in real time. This function is essential for navigating in real world environments, always subjected to rapid changes. The Laser Scanners provide an accurate 2-D picture of the environment because of theirs high angular and range resolution. Nevertheless, even using a high precision kind of sensor, the presence of differing surface reflectivity's, textures, relative orientations, etc can produce erroneous signals. These problems should be analyzed during the design of the sensor (Adams, 1999) or later, by the user, when processing the data. For the data acquisition procedure we used a real platform, a differential drive mobile robot, SmartROB-2, (see Figure 1-a and Badreddin, 1992) equipped with two SICK LMS200 Laser Scanners covering 360° with a resolution of 2° and radial error measurement less than $\pm 20\text{mm}$.

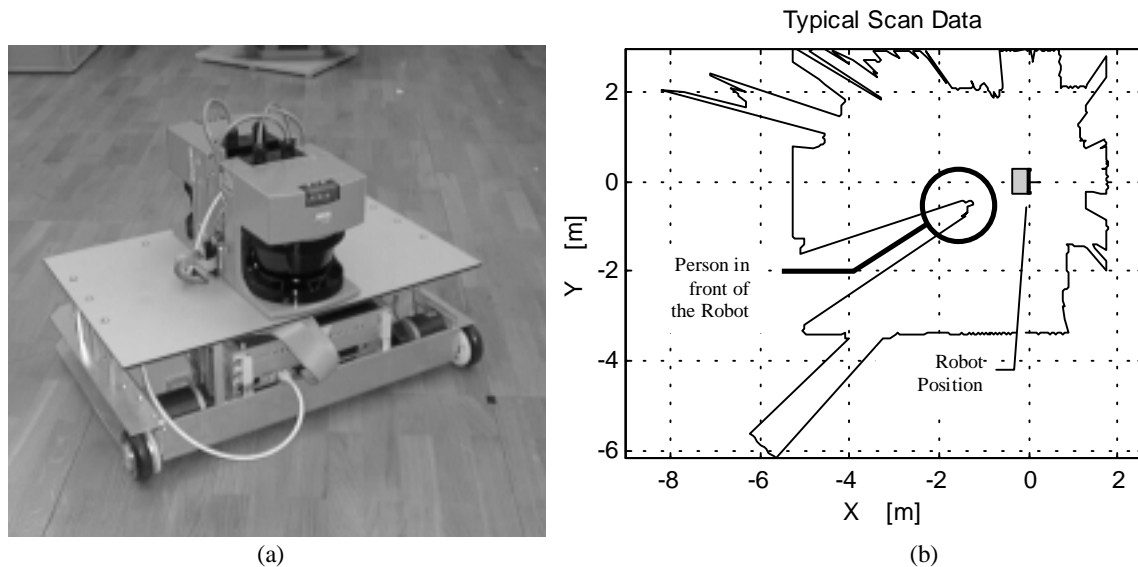


Figure 1 – (a) Photo of the mobile robot SmartROB-2 equipped with two SICK LMS200 Laser Scanners and (b) plot of a typical laser scan data obtained by the robot during the experimental tests.

3. NAVIGATION CONTROLLER

The basic navigation controller used was previously described in Becker (1999). Nevertheless, a new Obstacle Avoidance procedure was adopted. It essentially consists of three components: an algorithm for Obstacle Detection, one for Motion Detection and Tracking (or simply Motion Estimation), and another one for computing Evasive Maneuvers, which is based on the *Velocity Obstacle* approach, Prassler *et al.*, 1999a, b (Figure 2).

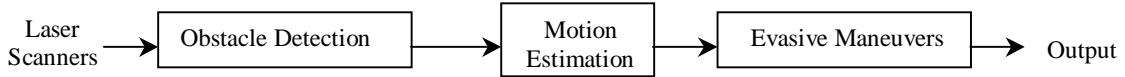


Figure 2 - Block diagram of the algorithms used for Obstacle Avoidance Controller.

3.1 Obstacle Detection and Motion Estimation

Based on the scan data, the robot can build a model of the environment and use this model to detect obstacles, motion, and to track the motion. However, how to represent and use these data in a real-time application? An efficient scheme for mapping scan data is the *Occupancy Grid Representation* (Elfes, 1989). In this approach, all the scan data are represented on a 2-D rectangular grid, where each grid element (cell) describes a small region of the real world. But, due to the necessity to completely initialize the grid and set each cell to some default value at step time t , this representation is too expensive in computer time. To avoid this problem and save computer time, a variation of this representation, called *Time Stamp Map* (Fiorini & Shiller, 1998), is used. In this case, only the occupied cells observed at each time step t are mapped, all the other cells in this grid remain untouched. This procedure allows the implementation in real-time, avoiding spend time mapping free space.

The easiest way to identify changes in the environment is to consider a sequence of the *Time Stamp Map*'s and to investigate where the steps of this sequence differ. A discrepancy between two subsequent steps is a strong indication of a potential modification in the environment. An important parameter to be considered is the distance from the bottom to the Laser Scanners position (i.e. the height of the sensor). Changing this position, the characteristics of the same objects is also changed. In the Figure 4 it is possible to observe the differences between two different positions of the Scanners (a- sensors positioned close to the ground and b- one meter of height). The method proposed to treat this problem is shown in the block diagram bellow (the double line blocks consider more than one *Time Stamp Map*):

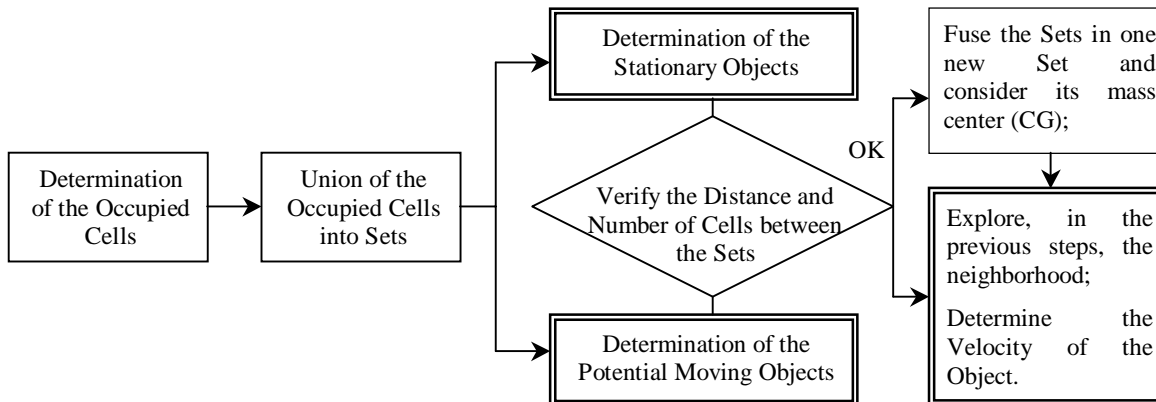


Figure 3 – Simplified Block Diagram of the method used to detect and track moving obstacles.

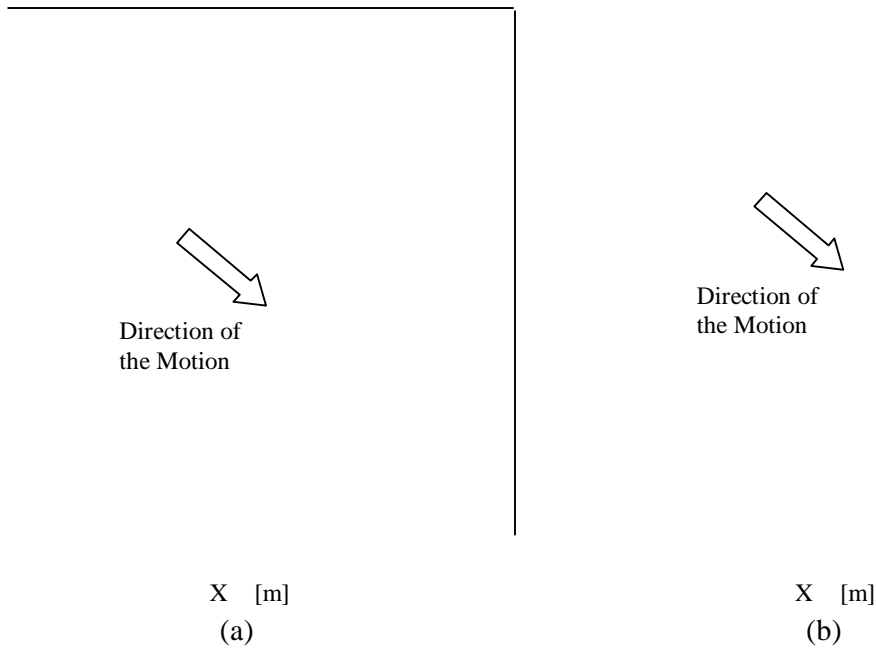


Figure 4 - Sequence of Scans of one person walking in a diagonal path: scanners located (a) one meter of height and (b) close to the ground. In both cases the robot was kept stationary.

Using the ideas proposed above, we obtained the following results:

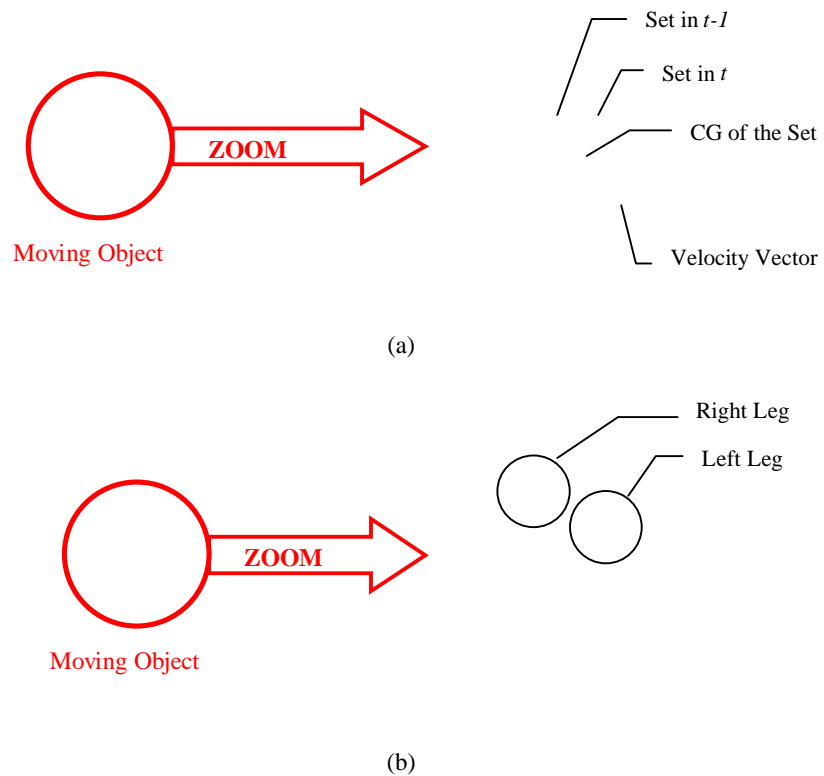


Figure 5 – Detection and tracking of one person walking in a diagonal path: scanners located (a) one meter of height and (b) close to the ground. In both cases the robot was kept stationary.

As shown in Figures 4 and 5, it is not easy to detect and track people walking (Kleitsch, 1998). Many times, in real environments, the informations provided by the Laser Scanners are not enough to avoid collisions with a lower height objects or objects that do not reflect the laser beam properly (e.g. glass doors).

3.2 Evasive Maneuvers

The basic idea of this algorithm is to obtain a “collision cone” and an equivalent condition on the absolute velocity of the robot (“velocity obstacle”), specific to a particular pair of robot / obstacle. The velocities of the robot inside of the *velocity obstacle* would cause collision between the robot and the obstacle, velocities outside of velocity obstacle would avoid collisions, and velocities on the boundaries of the velocity obstacle would result in the robot grazing the obstacle. Combining the results for each obstacle and the output of a navigation controller, it is possible to choose the better control output, which results in the robot going to the goal position and avoiding the obstacles. It is possible to see in Figure 6-b that for the actual velocity of the robot (*vector velocity* inside of the velocity obstacle area, see circle) the collision is imminent. Nevertheless, any velocity orientation that provides a position outside of the filled area would avoid the collision.

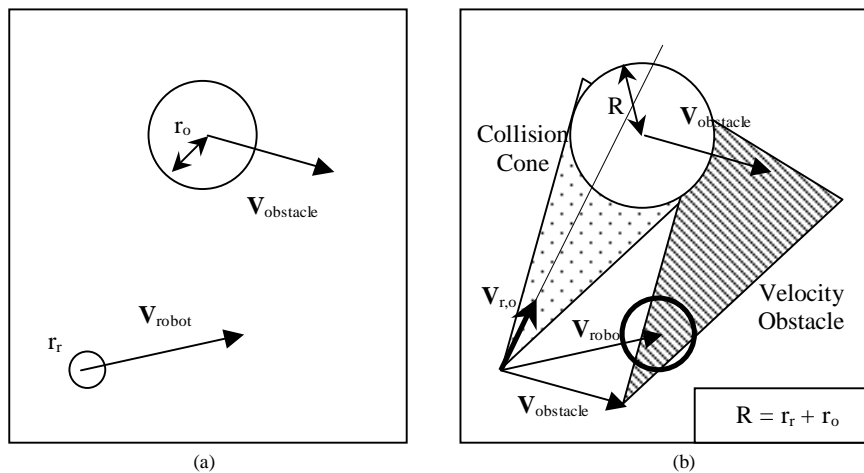


Figure 6 - The relative velocity ($\mathbf{V}_{r,o}$), the Collision Cone and the Velocity Obstacle.

There are two options to avoid the collision: the first one is to change the orientation of the *velocity vector* (\mathbf{V}_{robot}), steering the vehicle and, the second one, to change the modulus of the *velocity vector*, accelerating the vehicle (positive or negative acceleration). As one of the principles adopted for the controller behavior is to use a cruiser velocity during the path, the modulus of the *velocity vector* is changed only if the maximum steering angle is exceeded.

Based on the above premises, we used a circle with radius equal to the desired cruiser velocity to represent the set of possible orientation angles of the robot. Then, for each obstacle observed in the neighborhood, the *velocity obstacle* and the intersections between it and the circle are calculated. The results obtained for all obstacles are combined and result in the *Sets of Dangerous Orientation Angles* for the robot (Figure 7). In other words, if the orientation angle of the robot is inside of one of these sets, a collision between the robot and the obstacle will occur. Nevertheless, if the maximum steering angle is exceeded, the algorithm searches a new value for the modulus of the *velocity vector* that avoids the crash and does not exceed the maximum steering angle.

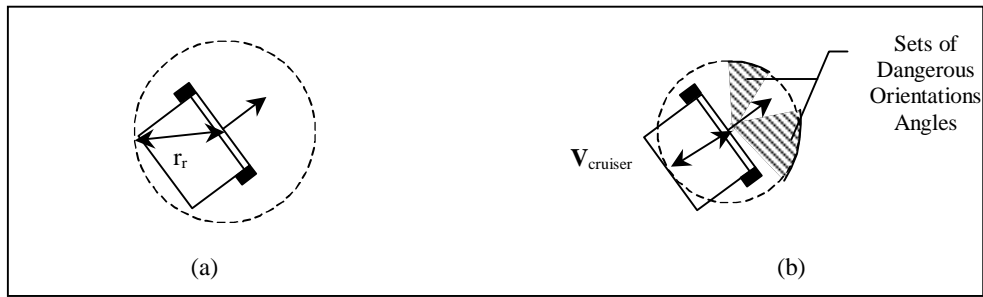


Figure 7 – (a) Detail of the robot radius and, (b) Sets of Dangerous Orientation Angles for the robot on the circle with radius equal to $V_{cruiser}$.

To simulate the vehicle behavior in complex environments (with moving and fixed obstacles), the software Matlab was used. Simulations are made in an environment with 10 obstacles using the technique described above: motion planning using velocity obstacle. A cruise velocity is used during the path. The Figure 8 shows the environment used: the moving obstacles (O_1 , O_2 , O_4 , O_5 and O_6), the fixed obstacles (O_3 , O_7 , O_8 , O_9 and O_{10}), the initial and final positions, and orientations.

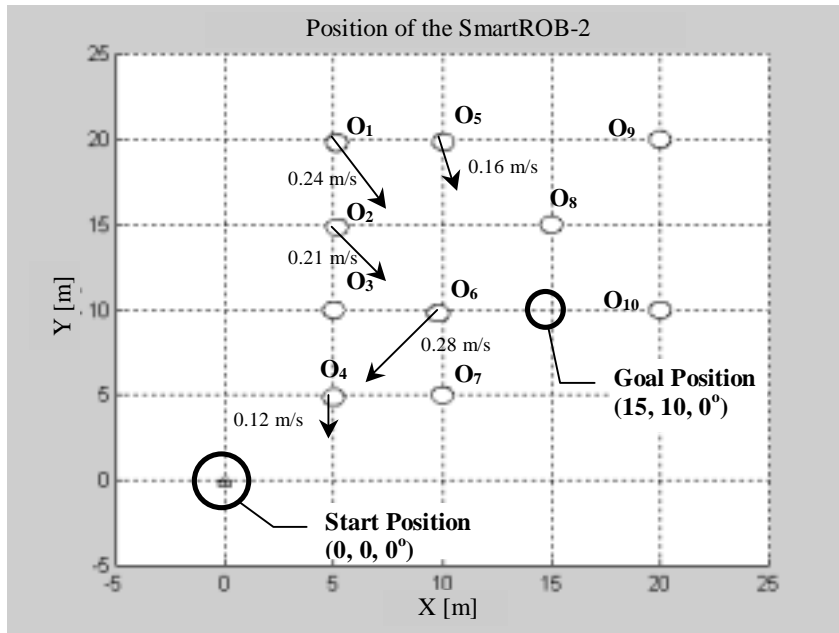


Figure 8 - Beginning of the simulation – environment with ten obstacles (five moving obstacles, their velocities are represented by vectors).

The results obtained using the obstacle avoidance controller were excellent. The capacity of the controller to predict and determine collision courses with several moving obstacles and compute an avoidance maneuver, which is close to its original heading as possible, allows the controller to avoid undesired and unnecessary accelerations and steering angles. It is possible to observe in Figures 9 and 10 that the vehicle moves close to the obstacles, grazing some of them. Only when the maximum steering angle is exceeded, the controller changes the cruise velocity to compute a new output angle. The behavior of the vehicle grazing the obstacles reduces the total displacement and the consumption of energy of the vehicle and increases its range.

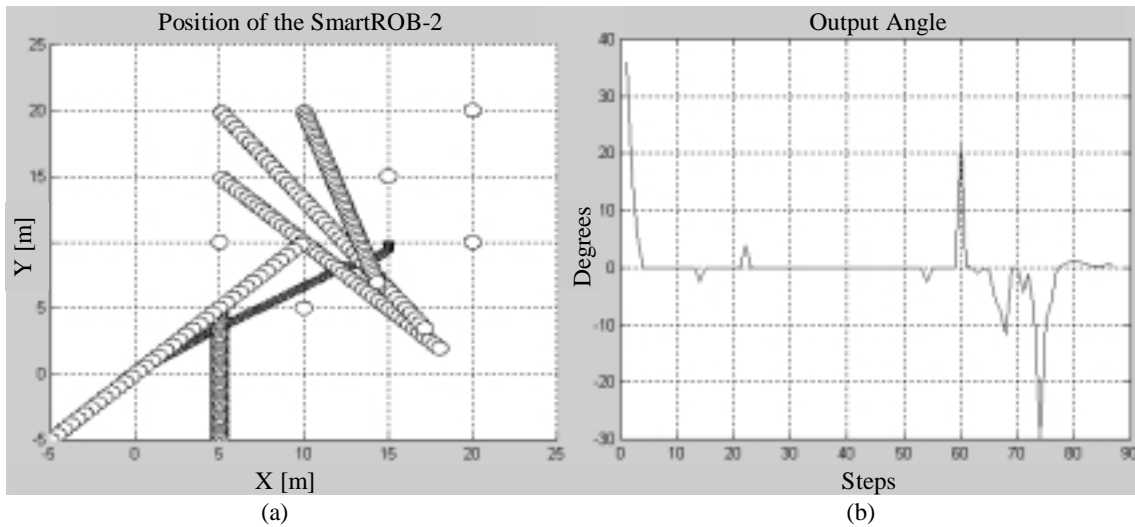


Figure 9 - (a) Paths simulated for the robot and the obstacles and (b) controller output angle.

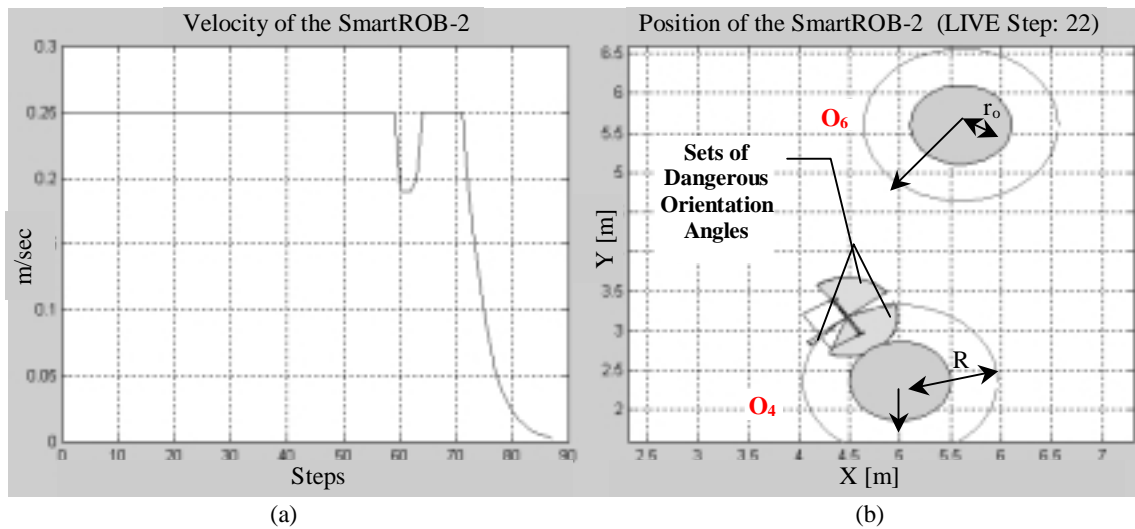


Figure 10 - (a) Velocity simulated for the robot and (b) detail of the sets of dangerous orientation angles (if the orientation angle of the robot is inside of these sets, the collision is imminent).

4. CONCLUSIONS AND OUTLOOK

We developed researches on Obstacle Avoidance for autonomous and assisted vehicles using a real sensor data and a mobile robot (SmartROB-2). The researches included simulations using the software Matlab, experimental tests using the mobile robot, and the real-time system XOberon.

The results obtained using the variation of the *Velocity Obstacle* approach were excellent, nevertheless it is necessary to do experimental tests to evaluate the behavior of the controller in real world environments. Tests about the position of the sensors on the robot, i.e., the height of the sensors, and its influence in the performance of the motion-tracking algorithm shown the efficiency of the algorithm developed. Due to the situations found in real world environments, we recommend the use of the Laser Scanners mainly to track moving obstacles and add Sonar or Vision sensors to detect objects not recognized by the Laser Scanners (e.g. lower height objects, glass doors, etc). The use sensor fusion would improve the performance of the algorithm.

ACKNOWLEDGMENTS

The authors wish to express their thanks to Prof. Gerhard Schweitzer and all the staff of the Institute of Robotics, ETH Zurich (Switzerland) for their contribution to this paper. The CNPq - National Research Council (Brazil) and the ICSC - World Laboratory (Switzerland) funded this study.

REFERENCES

- Adams, M. D., 1999, "Sensor Modeling, Design and Data Processing for Autonomous Navigation", World Scientific Series in Robotics and Intelligent Systems, Vol. 13, p.232.
- Badreddin, E; 1992, "Mobile Roboter – Einführung in den Aufbau und die Funktionsweise Mobiler Roboter", Kurs, June 15 – 16, 1992, IfR – ETH Zurich, Switzerland.
- Baxter, J. W. & Bumby, J. R., 1995, "Fuzzy control of a robotic vehicle", Proc. of Instn. Mech. Engrs. - Part I: Journal of Systems and Control Engineering, Vol. 209, pp. 79 - 91.
- Becker, M., 1999, "Simulation of a fuzzy controller for autonomous vehicles in complex and dynamic environments", Proc. of COBEM'99 – XV Brazilian Congress on Mechanical Engineering, Águas de Lindóia – SP, Brazil, Nov 22 to 26, 1999, CD-ROM.
- Beom, H. R. & Cho, H. S., 1995, "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning", IEEE Trans. on Syst., Man, and Cyber., Vol. 25, n. 3, pp. 464 - 477.
- Boby, S. & Lumelsky, V., 1999, "Control of dynamics and sensor based motion planning for a differential drive robot", Proc. of the 1999 IEEE International Conference on Robotics and Automation, May 1999, Detroit – MI - USA, pp. 157 – 162.
- Elfes, A., 1989, "Occupancy grids: a probabilistic framework for robot perception and navigation", Ph.D. Thesis, Electrical and Computer Engineering Department / Robotics Institute, Carnegie-Mellon University, USA.
- Feng, D. & Krogh, B. H., 1991, "Dynamic steering control of conventionally steered mobile robots", Journal of Robotic Systems, Vol. 8, n. 5, pp. 699 - 721.
- Fiorini, P. & Shiller, Z., 1998, "Motion planning in dynamic environments using velocity obstacles", International Journal of Robotics Research, Vol. 17, pp. 760 - 772.
- Fu, L.-C. & Liu, D.-Y., 1990, "An efficient algorithm for finding a collision-free path among polyhedral obstacles", Journal of Robotic Systems, Vol. 7, n. 1, pp. 129 - 137.
- Kleitsch, C., 1998, "Objekte / Menschen extrahieren aus Distanzdaten", Semesterarbeit SS'98, IfR – ETH Zurich, Switzerland.
- Prassler, E., Scholz, J. & Fiorini, P., 1999a, "Navigating a robotic wheelchair in a railway station during rush hour", International Journal of Robotics Research, Vol. 18, pp. 711 - 727.
- Prassler, E., Scholz, J. & Elfes, A., 1999b, "Tracking people in a railway station during rush hour", Proc. of 1st Int. Conf. on Computer Vision Systems (ICVS'99), Gran Canaria, Spain, pp. 162 – 179.
- Wang, L.-L. & Tsai, W.-H., 1991, "Collision avoidance by a modified least-mean-square error classification scheme for indoor autonomous land vehicle navigation", Journal of Robotic Systems, Vol. 8, n. 5, pp. 677 - 698.
- Zapata, R., Lépinay, P. & Thompson, P., 1994, "Reactive behaviors of fast mobile robots", J. of Robotic Syst., vol. 11, n. 1, pp. 13 - 20.

CÁLCULO DO DESVIO MÉDIO ARITMÉTICO R_a EM FUNÇÃO DO AVANÇO

Jorge Antônio Chaves de Oliveira

Noemia Gomes de Mattos de Mesquita

Hugo Marcelo Bezerra de Carvalho

Sérgio Murilo Verissimo de Andrade

Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Departamento de engenharia Mecânica, Grupo de Fabricação, 50740-530, Recife, PE, Brasil. E-mail: ngmm@npd.ufpe.br

Resumo

Neste trabalho foi calculado o avanço para o qual se terá o acabamento superficial desejado, medido através do desvio médio aritmético (R_a). Para tal foi feita uma análise geométrica do perfil da superfície e deduzida uma função a qual foi aplicada o método numérico de Newton-Raphson.

Palavras-chaves: acabamento superficial, rugosidade, torneamento e condições de corte.

1. INTRODUÇÃO

Nas operações de usinagem, um dos fatores que determinam se a peça manufaturada tem o acabamento superficial adequado para desempenhar a função a qual ela se destina, é a rugosidade da sua superfície. O termo rugosidade será utilizado para designar as amplitudes da rugosidade, tal como a “Profundidade de Rugosidade” (R_t) ou “Desvio Médio Aritmético de Rugosidade” (R_a).

Em alguns trabalhos encontrados na literatura os autores relacionam os parâmetros de usinagem (avanço) com a profundidade de rugosidade (R_t) (König, 1984; Ferraresi, 1970; Sandvik, 1994; Mitsubishi, 1999). No entanto a dispersão que o parâmetro (R_t) apresenta faz com que a confiabilidade deste parâmetro seja baixa.

Neste trabalho a variação da rugosidade, medida através de do parâmetro (R_a) (desvio médio aritmético de rugosidade), parâmetro este mais confiável que o (R_t), em função do avanço é obtida através de um método numérico, de modo que, para um determinado acabamento superficial desejado na peça pode-se obter qual o avanço máximo com o qual pode-se usinar a mesma e que ainda garantirá o acabamento exigido. Tanto neste trabalho como em outros (Bet, 1999; Cassier, 1986), na relação entre o R_a e o avanço é levado em consideração a influência da geometria da ferramenta.

2. PARÂMETROS DE RUGOSIDADE

As superfícies técnicas podem ser medidas ou avaliadas através de diferentes equipamentos, tais como, rugosímetros, perfilômetros, perfilógrafos e outros. Bet em seu

trabalho relaciona vinte e dois parâmetros verticais de rugosidade. A definição dos diferentes parâmetros de rugosidade pode ser encontrado nas normas técnicas internacionais e de diferentes países, tais como: DIN 140 - 4777; ISO 468 - 3274; VDI 2001 – 3402; ABNT NB 1964; NF 1961; BS 1134 – 1961; UNI 3963 – 1960; (appud Ferraresi, 1970)

2.1 Profundidade de rugosidade R_t

Segundo a norma VDI/VDE 2601 (1977) R_t foi no passado usado com os seguintes significados: $R_t = R_z$; $R_t = W_t + R_z$; $R_t = R_{max}$; $R_t = P_t \cong W_t + R_{max}$.

R_t deverá, por esta razão, não mais ser usado. O parâmetro R_t , o qual foi utilizado neste trabalho, tem o mesmo significado de R_y , quando definido da seguinte forma: R_t é a distância vertical entre o ponto mais elevado e o mais profundo do perfil de rugosidade, dentro do trecho de medição (L_m). L_m , que é chamado percurso de medição é o trecho do perfil de rugosidade dentro do qual são medidos os parâmetros de superfície (DIN 4762, 1960).

Já existe uma equação teórica que fornece uma relação entre o avanço e o R_t , equação 1 (König, 1984).

$$f = \sqrt{8 \times r_\epsilon \times R_t} \quad (1)$$

Mas o R_t , como dito anteriormente não é um parâmetro muito confiável para medição da rugosidade, pois qualquer sujeira na superfície da peça, vibração na mesa de medição ou variação da tensão da rede elétrica causam grande variação no valor do R_t . Por esta razão, deve-se usar o R_a que é menos influenciado por fatores como os acima citados.

2.2 Desvio médio aritmético de rugosidade R_a

R_a é a média aritmética dos valores absolutos das distâncias “y” da linha média do perfil de rugosidade dentro do trecho de medição, figura 1 e equação 2. R_a tem o mesmo significado da altura do retângulo cujo comprimento é igual ao comprimento total de medição L_m e cuja a área é igual entre as áreas limitadas pela linha média ao perfil de rugosidade (DIN 4768, 1974).

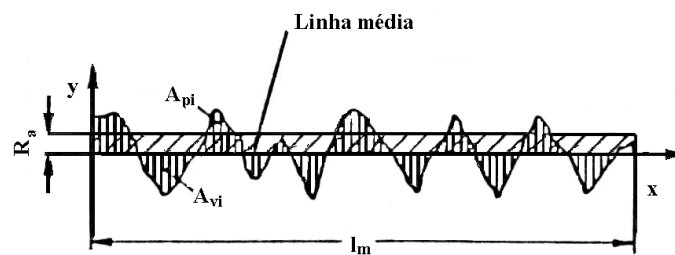


Figura 1. Desvio médio aritmético R_a

$$R_a \times L_m = \sum A_p + \sum A_v \quad (2)$$

Na equação 2, A_p (área de pico) é a área acima da linha média e A_v (área de vale) é a área abaixo da linha média. A linha média é a linha com a mesma forma do perfil geométrico, disposta paralelamente a direção geral do perfil. Ela divide o perfil de rugosidade, de modo que, a soma das áreas superiores seja igual a soma das áreas inferiores, figura 1.

3 DETERMINAÇÃO DO MÁXIMO AVANÇO, A PARTIR DA GEOMETRIA DA FERRAMENTA, QUE GERA O R_a DESEJADO

O torneamento é um processo que gera uma rugosidade cinemática periódica, o perfil da sua superfície é como o da figura 2. Para a obtenção do R_a , escolhe-se o L_m segundo a tabela 1, (NB-93 da ABNT, appud Ferraresi, 1970).

Tabela 1 - Comprimento de medição em função do R_a

Rugosidade, R_a (μm)	Mínimo comprimento de amostragem, L_m (mm)
De 0 até 0.3	0.25
Entre 0.3 e 3	0.8
Maior que 3	2.5

Nada garante que dentro do comprimento de medição, encontre-se um número inteiro de períodos do perfil da superfície, no entanto neste trabalho foi considerado que o L_m contém um número inteiro de períodos, ou seja um número inteiro de avanços, figura 2.

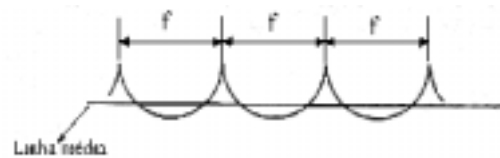


Figura 2- Perfil teórico de uma superfície torneada.

Para efeito de simplificação de cálculos, da figura 2 tem-se que:

$$L_m = n \times f \quad (3)$$

onde n é o número de períodos.

Pela definição de R_a e conforme a figura 2 tem-se:

$$R_a \times n \times f = \sum A_p + \sum A_v \quad (4)$$

A figura 3 mostra um período do perfil da figura 2, onde r_ϵ é o raio de arredondamento da ponta da ferramenta.

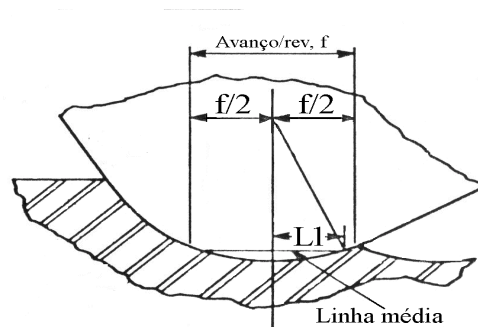


Figura 3. Perfil gerado pela ferramenta de corte.

Na figura 3, A_{p1} e A_{p2} são parcelas das áreas de pico A_{v1} e A_{v2} são parcelas das áreas de vale.

Observando-se a simetria da figura 3 constata-se que :

- as áreas de pico são iguais

$$A_{p1}=A_{p2}=A_p \quad (5)$$

- as áreas se vale são iguais

$$A_{v1}= A_{v2}=A_v \quad (6)$$

Uma vez que cada período contém duas áreas de pico e duas áreas de vale , podendo-se simplificar a equação 4

$$R_a \times n \times f = 2 \times n \times A_p + 2 \times n \times A_v \quad (7)$$

A figura 4 é uma ampliação de um trecho da figura 3.

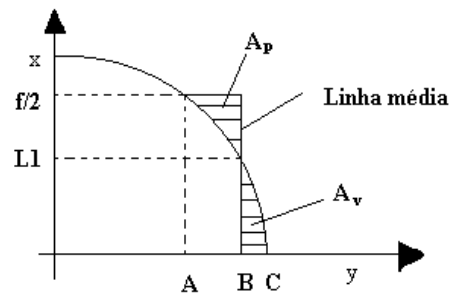


Figura 4. Detalhe da rugosidade cinemática.

Comparando-se as figuras 3 e 4, tem-se que OC corresponde ao raio de arredondamento da ferramenta de corte. A equação da circunferência definida pelo raio r_ϵ é:

$$x^2 + y^2 = r_\epsilon^2 \quad (8)$$

$$x = \sqrt{r_\epsilon^2 - y^2} \quad (9)$$

Calculando a área entre a curva $x = \sqrt{r_\epsilon^2 - y^2}$ e a reta $x = 0$ temos

$$A = \int_{y_1}^{y_2} \int_{x=0}^{x=\sqrt{r_\epsilon^2 - y^2}} dx dy = \int_{y_1}^{y_2} \sqrt{r_\epsilon^2 - y^2} dy \quad (10)$$

Pela figura 4, para $y_1 = B$, x_1 corresponde a L1. Fazendo-se $x_1 = L1$, na equação (8), pode-se calcular o limite inferior (y_1) da equação (10)

$$L1^2 + y_1^2 = r_\epsilon^2 \therefore y_1 = \sqrt{r_\epsilon^2 - L1^2} \quad (11)$$

Pela figura 4, $y_2 = C$ neste, x_2 é 0. O limite superior (y_2) da equação (10), é obtido fazendo-se $x_2 = 0$ na equação (8)

$$0^2 + y_2^2 = r_\epsilon^2 \therefore y_2 = r_\epsilon \quad (12)$$

A área do vale (A_v) é dada pela equação (10) calculada entre os limites y_1 e y_2 dados respectivamente pelas (11) e (12). Pela figura 4, a área de vale corresponde a área abaixo da curva entre os pontos BC.

$$A_v = \int_{\sqrt{r_\epsilon^2 - L1^2}}^{r_\epsilon} \sqrt{r_\epsilon^2 - y^2} dy = \frac{1}{4} \times r_\epsilon^2 - \frac{1}{2} \times L1 \times \sqrt{r_\epsilon^2 - L1^2} - \frac{1}{2} \times r_\epsilon^2 \times \arcsen\left(\frac{\sqrt{r_\epsilon^2 - L1^2}}{r_\epsilon}\right) \quad (13)$$

Pela figura 4, para $y_1 = A$, x_1 corresponde a $f/2$. Fazendo-se $x_1 = f/2$, na equação (8), pode-se calcular o limite inferior (y_1) da equação (10)

$$\left(\frac{f}{2}\right)^2 + y_1^2 = r_\epsilon^2 \therefore y_1 = \sqrt{r_\epsilon^2 - \left(\frac{f}{2}\right)^2} \quad (14)$$

Pela figura 4, para $y_2 = B$, x_1 corresponde a $L1$. Fazendo-se $x_2 = L1$, na equação (8), pode-se calcular o limite superior (y_2) da equação (10)

$$L1^2 + y_2^2 = r_\epsilon^2 \therefore y_2 = \sqrt{r_\epsilon^2 - L1^2} \quad (15)$$

Pela figura 4, a área de pico (A_p) corresponde a área do retângulo de altura $f/2$ e base ($B - A$) menos a área abaixo da curva entre os pontos AB. Esta área da curva é dada pela equação (10) calculada entre nos limites y_1 e y_2 dados respectivamente pelas (14) e (15). O ponto A é y_1 e o ponto B é y_2 .

$$A_p = \left[\frac{f}{2} \times \left(\sqrt{r_\epsilon^2 - L1^2} - \sqrt{r_\epsilon^2 - \left(\frac{f}{2}\right)^2} \right) \right] - \int_{\sqrt{r_\epsilon^2 - \left(\frac{f}{2}\right)^2}}^{\sqrt{r_\epsilon^2 - L1^2}} \sqrt{r_\epsilon^2 - y^2} dy \quad (16)$$

onde a integral é dada pela equação (17), chamada de S.

$$\begin{aligned} S &= \frac{1}{2} \times L1 \times \sqrt{r_\epsilon^2 - L1^2} + \frac{1}{2} \times r_\epsilon^2 \times \arcsen\left(\frac{\sqrt{r_\epsilon^2 - L1^2}}{r_\epsilon}\right) - \dots \\ &\dots - \frac{1}{8} \times f \times \sqrt{4 \times r_\epsilon^2 - f^2} - \frac{1}{2} \times r_\epsilon^2 \times \arcsen\left(\frac{\sqrt{4 \times r_\epsilon^2 - f^2}}{r_\epsilon} \times \frac{1}{2}\right) \end{aligned} \quad (17)$$

Definindo-se uma função M :

$$M = A_v - A_p \quad (18)$$

Conforme a definição de linha média as áreas acima da linha média são iguais as áreas abaixo da linha média, daí tem-se:

$$2 \times A_p = 2 \times A_v \quad (19)$$

$$A_v = A_p \quad (20)$$

E sabendo que $A_v = A_p$ temos que :

$$M = A_v - A_p = 0 \quad (21)$$

Usando

$$R_a \times f = 2 \times A_p + 2 \times A_v \quad (22)$$

Como pela equação (21) $A_v = A_p$ tem-se

$$R_a \times f = 4 \times A_v \quad (23)$$

ou seja:

$$f = \frac{4 \times A_v}{R_a} \quad (24)$$

Substituindo-se (23) na equação (16) e esta juntamente com a (13) em (20), M passa a ter como única incógnita L1, assim sendo usando-se o método numérico de Newton-Raphson nesta equação (20) pode-se determiná-lo.

Aplicando-se o método numérico de Newton-Raphson tem-se

$$L1_{i+1} = L1_i - M(L1_i)/M'(L1_i) \quad (25)$$

onde $M' = \frac{dM}{dL1}$.

Fazendo algumas iterações obtém-se um valor para L1, que substituído em (13) e esta na equação (23), obtém-se o avanço desejado.

4 RESULTADOS

Os resultados deste trabalho comprovam os resultados obtidos e publicados anteriormente por outros autores. Segundo Cassier (1986) o avanço é dado por:

$$f = \sqrt{31 \times R_a \times r_\epsilon^2} \quad (26)$$

Segundo Prasad (1997) o Avanço é dado por:

$$f = \sqrt{18 \times \sqrt{3} \times R_a \times r_\epsilon^2} \quad (27)$$

Segundo Drozda (1983) o avanço é dado por

$$f = \frac{2 \times (r_\epsilon^2 \times B - H \times G)}{R_a} \quad (28)$$

onde os parâmetros H, G, B(rad) são mostrados na figura 5.

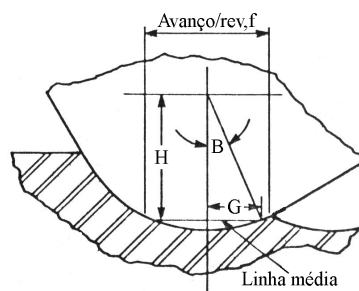


Figura 5. Modelo de rugosidade segundo (Drozda 1983).

Comparando as figuras 5 e 4 temos que :

$$G = L1 \quad (29)$$

$$H = \sqrt{r_\epsilon^2 - L1} \quad (30)$$

$$B = \arccos \frac{H}{r_\epsilon} \quad (31)$$

Na tabela 2 são apresentados os resultados comparativos entre o método desenvolvido e os outros métodos encontrados na literatura.

Tabela 2 - Comparação dos resultados

	Métodos			
	Cassier	Prasad	Drozda	Presente trabalho
Avanço $r_\epsilon = 1,2 \text{ mm}$ $R_a = 6,5 \mu\text{m}$	0,491732	0,493133	0,491047	0,491044
Avanço $r_\epsilon = 0,6 \text{ mm}$ $R_a = 8 \mu\text{m}$	0,121984	0,122331	0,122204	0,122209
Avanço $r_\epsilon = 0,8 \text{ mm}$ $R_a = 3,2 \mu\text{m}$	0,281709	0,2825112	0,28163	0,281634
Avanço $r_\epsilon = 0,4 \text{ mm}$ $R_a = 1,88 \mu\text{m}$	0,152683	0,153118	0,1525556	0,152553

5. CONCLUSÃO

Os valores calculados por este método apresentam, comparado com os trabalhos de Cassier, Prasad e Drozda, um desvio percentual que varia de 0.002% a 0.7%. A expressão para R_a apresentada por Drozda foi quase 100% comprovado pelo método matemático desenvolvido neste trabalho. A vantagem deste método é que o valor do avanço para uma R_a pré-determinado, pelas exigências funcionais da peça, pode ser obtido exclusivamente em função do raio de arredondamento da ponta da ferramenta de corte. Na expressão apresentada por Dozda os parâmetros B, H e G não estão relacionados diretamente com os parâmetros do processo de usinagem. Neste trabalho foram deduzidas as equações (29), (30) e (31) que possibilitam o cálculo de B, H e G.

Sugere-se, portanto, que em trabalhos futuros seja verificada se experimentalmente os resultados aqui apresentados são comprovados e qual a faixa de avanços para a qual as equações apresentadas e desenvolvidas são válidas. Em relação ao trabalho de Cassier e Prasad é interessante ter-se também as faixas de avanço dentro das quais as expressões por eles apresentadas são válidas.

Agradecimentos

Agradecemos ao CNPq/PIBIC pelo apoio financeiro para a realização deste trabalho.

REFERÊNCIAS

Bet, Luciano, 1999, “Estudo da Medição da Textura de Superfícies com Sondas Mecânicas e Sondas Ópticas Tipo Seguidor”, Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, S.C., apêndice 2.

Cassier, Zulay, Departamento de Mécanica, Universidad Simón Bolívar, Caracas, Venezuela.

Catálogo Sandvik, 1993/1994, “Herramientas de tornear”, 560p.

Din 4762 Blatt1, Erfassung der Gestaltabweichungen 2 . bis 5 Ordnung an Oberflächenschnitten, Beuth Verlag, Ago 1960

Din 4768 Blatt1, Ermittlung der Ra, Rz, Rmax, mit elektrischen Tastschnittgeräten , Beuth Verlag, Ago 1974

Drozda, Thomas J., 1983, “Tool and Manufacturing Engineers Book”, Ed. TMEH, USA, volume 1, chapter 1, 24-25p.

Ferraresi, D, 1970, “Fundamentos da Usinagem dos Metais”, Ed Blucher Ltda, S. Paulo, volume 1, 601-602p e 749p.

König, W, 1984, “Fertigungsverfahren”, VDI-Verlog, Dusseldorf, vol1, 179p

DIN 4762 Blatt1, 1960,

Mitsubishi Carbide, 1999, “Herramientas e Incertors, 487p.

Prasad, Avsrk, “Optimal Selection of Process for Turning Operation in CAPP Sistem Taylor and Francis Ltd , International Journal of Product Research 1997, Vol 3 , Iss 6, 1495-1522p.

ROBOT CALIBRATION USING A 3D VISION-BASED MEASUREMENT SYSTEM

José Maurício S. T. Motta

Universidade de Brasília, Departamento de Engenharia Mecânica, 70910-900 - Brasília, DF, Brasil. E-mail: jmmotta@unb.br

R. S. McMaster

Cranfield University, School of Industrial and Manufacturing Science, Building 62, Cranfield, MK43 OAL, England, UK. E-mail: R.S.McMaster@cranfield.ac.uk

Abstract

This work presents techniques for modeling and performing robot calibration processes under off-line programming using a 3-D vision-based measurement system. Kinematic modeling follows a singularity-free concept. The measurement system consists of a single CCD camera mounted on the robot tool flange and uses space resection models to measure the robot end-effector pose relative to a world coordinate system. A wide-angle lens is used and lens radial distortions are included in the model. Experimentation is performed on a PUMA-500 robot to test its accuracy improvement using the calibration system proposed. The robot was calibrated in many different regions and volumes within its workspace, achieving accuracy between three and six times better when comparing errors before and after calibration. The proposed off-line robot calibration system is fast, accurate and ease to setup.

Keywords: Kinematic model, Robot calibration, Absolute accuracy, Camera calibration.

1. INTRODUCTION

Most currently used industrial robots are still programmed by a teach pendant, especially in the automotive industry. However, the importance of off-line programming in industry as an alternative to teach-in programming is steadily increasing. The main reason for this trend is the need to minimize machine downtime and thus to improve the rate of robot utilization. A typical welding line with 30 robots and 40 welding spots per robot takes about 400 hours for robot teaching (Bernhardt, 1997). Nonetheless, for a successful accomplishment of off-line programming the robots need to be not only repeatable but also accurate.

Robot repeatability is unaffected by the method of programming since it is due to random errors (e.g. due to the finite resolution of joint encoders). In contrast, the systematic errors in absolute position are due almost entirely to programming the robot off-line. One of the leading sources of lack of accuracy is the mismatch between the prediction made by the kinematic model and the actual system. Robot constant pose errors are attributed to several sources, including errors in geometrical parameters (e.g. link lengths and joint offsets) and deviations which vary predictably with position (e.g. compliance or gear transmission errors). Robot calibration is an integrated process of modeling, measurement, numeric identification of actual physical characteristics of a robot, and implementation of a new model. By calibration, an improved kinematic model of the physical robot is generated and used in conjunction with simulation and off-line programming systems. This eliminates most of the

systematic errors in position and allows accurate off-line programs to be generated and task completion to be reliably achieved without on-line position editing.

The calibration procedure first involves the development of a kinematic model whose parameters represent accurately the actual robot. Next, robot characteristics specifically chosen are measured with measurement instruments with known accuracy. Then a parameter identification procedure is used to compute the parameter values set which, when introduced in the robot nominal model, accurately represents the measured robot behavior. Finally, the model in the position control software is corrected.

In spite of the industrial needs cited above, no calibration system has been accepted and used generally in industry so far. Notwithstanding many calibration systems being available currently in the market, none of them combine low price, accuracy, ease of use, and speed of setup and implementation (Schröer, 1994).

The objectives of this research are to investigate theoretical aspects involved in robot calibration methods and systems, to develop a feasible low cost vision-based measurement system using a single camera and, finally, to construct a prototype of a robot calibration system. More specific goals can be listed as: to achieve robot position accuracy below 1mm after calibration; to investigate practical aspects and achievable accuracy using common off-the-shelf CCD cameras as a 3-D measurement system using only a single camera; to build an off-line robot calibration system aiming at low cost, ease of use, flexibility within robot environments, and with an acceptable accuracy. The robot used to test the system was a PUMA-500.

2. KINEMATIC MODELING AND PARAMETER IDENTIFICATION

The first step to kinematic modeling is the proper assignment of coordinate frames to each link. Coordinate frames to joints are assigned such that the z-axis is coincident with the joint axis. This convention is used by many authors and in many robot controllers (McKerrow, 1995, Paul, 1981). The x-axis or the y-axis have their direction according to the convention used to parameterize the transformations between links. For either perpendicular or parallel joint axes the Denavit-Hartenberg or Hayati modeling convention were used respectively. The requirements of a singularity-free parameter identification model prevents the use of a single minimal modeling convention that can be applied uniformly to all possible robot geometries (Schröer *et al.*, 1997, Baker, 1990). At this point the homogeneous transformations between joints must have been already determined. The other axis (x or y) can be determined using the right-hand rule.

The kinematic equation of the robot manipulator is obtained by consecutive homogeneous transformations from the base frame to the last frame. Thus,

$$\hat{T}_N^0 = \hat{T}_N^0(k) = T_{1}^0 \cdot T_{2}^1 \dots T_{N}^{N-1} = \prod_{i=1}^N T_i^{i-1} \quad (1)$$

where N is the number of joints (or coordinate frames), $p = [p_1^T \ p_2^T \ \dots \ p_n^T]^T$ is the parameter vector for the manipulator, and p_i is the link parameter vector for the joint i, including the joint errors. The exact link transformation A^{i-1}_i is (Driels & Pathre, 1990):

$$A^{i-1}_i = T^{i-1}_i + \Delta T_i \quad , \quad \Delta T_i = \Delta T_i(\Delta p_i) \quad (2)$$

where Δp_i is the link parameter error vector for the joint i.

The exact manipulator transformation \hat{A}_{N-1}^0 is

$$\hat{\mathbf{A}}^0_N = \prod_{i=1}^N (\mathbf{T}^{i-1}_i + \Delta \mathbf{T}_i) = \prod_{i=1}^N \mathbf{A}^{i-1}_i \quad (3)$$

Thus,

$$\hat{\mathbf{A}}^0_N = \hat{\mathbf{T}}^0_N + \Delta \hat{\mathbf{T}} \quad , \quad \Delta \hat{\mathbf{T}} = \Delta \hat{\mathbf{T}}(\mathbf{q}, \Delta \mathbf{p}) \quad (4)$$

where $\Delta \mathbf{p} = [\Delta p_1^T \ \Delta p_2^T \ \dots \ \Delta p_n^T]^T$ is the manipulator parameter error vector and \mathbf{q} is the vector of joint variables $[\theta_1^T \ \theta_2^T \ \dots \ \theta_N^T]^T$. It must be stated here that $\Delta \hat{\mathbf{T}}$ is a non-linear function of the manipulator parameter error vector $\Delta \mathbf{p}$.

Considering m the number of measure positions it can be stated that

$$\hat{\mathbf{A}} = \hat{\mathbf{A}}^0_N = \hat{\mathbf{A}}(\mathbf{q}, \mathbf{p}) = (\hat{\mathbf{A}}(\mathbf{q}_1, \mathbf{p}), \dots, \hat{\mathbf{A}}(\mathbf{q}_m, \mathbf{p}))^T \quad : \mathfrak{R}^n \times \mathfrak{R}^{mN} \quad (5)$$

where $\hat{\mathbf{A}}: \mathfrak{R}^n \times \mathfrak{R}^{mN}$ is function of two vectors with n and mN dimensions, n is the number of parameters and N is the number of joints (including the tool), and

$$\Delta \hat{\mathbf{T}} = \Delta \hat{\mathbf{T}}(\mathbf{q}, \Delta \mathbf{p}) = (\Delta \hat{\mathbf{T}}(\mathbf{q}_1, \Delta \mathbf{p}), \dots, \Delta \hat{\mathbf{T}}(\mathbf{q}_m, \Delta \mathbf{p}))^T \quad : \mathfrak{R}^n \times \mathfrak{R}^{mN} \quad (6)$$

All matrices or vectors in bold are functions of m . The identification itself is the computation of those model parameter values $\mathbf{p}^* = \mathbf{p} + \Delta \mathbf{p}$ which result in an optimal fit between the actual measured positions and those computed by the model, i.e., the solution of the non-linear equation system (Motta & McMaster, 1999a)

$$\mathbf{J} \cdot \mathbf{x} = \mathbf{b} \quad (7)$$

where the following notation can be used

$$\mathbf{b} = \mathbf{M}(\mathbf{q}) - \mathbf{B}(\mathbf{q}, \mathbf{p}) \in \mathfrak{R}^{\phi m} \quad (8)$$

$$\mathbf{J} = \mathbf{J}(\mathbf{q}, \Delta \mathbf{p}) \in \mathfrak{R}^{\phi m \times n} \quad (9)$$

$$\mathbf{x} = \Delta \mathbf{p} \in \mathfrak{R}^n \quad (10)$$

$$\mathbf{r} = \mathbf{J} \cdot \mathbf{x} - \mathbf{b} \in \mathfrak{R}^{\phi m} \quad (11)$$

where \mathbf{B} is a vector formed with position and orientation components of $\hat{\mathbf{A}}$, $\mathbf{M}(\mathbf{q})$ are all measured components and ϕ is the number of measurement equations provided by each measured pose. \mathbf{J} is the identification Jacobean and \mathbf{r} is the residue to be minimized. If orientation measurement can be provided by the measurement system then 6 measurement equations can be formulated per each pose. If the measurement system can only measure position, each pose measurement can supply data for 3 measurement equations per pose and then \mathbf{B} includes only the position components of $\hat{\mathbf{A}}$.

One method to solve non-linear least-square problems proved to be very successful in practice and then recommended for general solutions is the algorithm proposed by Levenberg-Marquardt (Dennis & Schnabel, 1983). Several algorithms versions of the L.M. algorithm have been proved to be successful (globally convergent). From eq. (7) the method can be formulated as

$$x_{j+1} = x_j - \left[\mathbf{J}(x_j)^T \cdot \mathbf{J}(x_j) + \mu_j \cdot \mathbf{I} \right]^{-1} \cdot \mathbf{J}(x_j) \cdot \mathbf{b}(x_j) \quad (12)$$

where, according to Marquardt suggestion, $\mu_j = 0.001$ if x_j is the initial guess, $\mu_j = \lambda(0.001)$ if $\|\mathbf{b}(x_{j+1})\| \geq \|\mathbf{b}(x_j)\|$, $\mu_j = 0.001/\lambda$ if $\|\mathbf{b}(x_{j+1})\| \leq \|\mathbf{b}(x_j)\|$ and λ is a constant valid in the range of $2.5 < \lambda < 10$ (Press et al, 1994).

3. VISION-BASED MEASUREMENT SYSTEM

Robot Calibration using a camera system is potentially fast, automated and easy to use. Cameras can also provide full pose measuring capability (position and orientation). The system configurations can vary from stationary cameras to stereo or single moving cameras. Due to the needs of accuracy and large range of motion during robot calibration procedures, stationary cameras cannot fulfill the requirements, and stereo moving cameras are restricted to local calibration within small volumes since the field-of-view is limited by the distance between the cameras. A single moving camera presents the advantages of a large field-of-view with a potential large depth of field, and a considerably reduced hardware and software complexity of the system. One disadvantage is the need of camera re-calibration at each pose (position and orientation).

The camera model is at first assumed to be as the standard distortion-free “pin-hole” by which every real object point is connected to its corresponding image point through a straight line that passes through the focal point of the lens (Fig.1). The transformation from the world coordinates (x_w, y_w, z_w) to the camera coordinates (x, y, z) is:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \cdot \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \mathbf{T} \quad (13)$$

where the rotation matrix \mathbf{R} and translation vector \mathbf{T} can be written as:

$$\mathbf{R} = \begin{bmatrix} r1 & r2 & r3 \\ r4 & r5 & r6 \\ r7 & r8 & r9 \end{bmatrix} \quad \text{and} \quad \mathbf{T} = [\mathbf{T}_x \quad \mathbf{T}_y \quad \mathbf{T}_z]^T \quad (14)$$

The transformation from the 3-D camera coordinates to the distorted or true image coordinates (X, Y) , Pd in Fig. 1, can be achieved using the Radial Alignment Constraint algorithm (RAC) (Tsai, 1987, Zhuang & Roth, 1996, Lenz & Tsai, 1987).

$$\frac{X}{1 - k \cdot r^2} \cong f \cdot \frac{r1 \cdot x_w + r2 \cdot y_w + r3 \cdot z_w + \mathbf{T}_x}{r7 \cdot x_w + r8 \cdot y_w + r9 \cdot z_w + \mathbf{T}_z} \quad (15)$$

$$\frac{Y}{1 - k \cdot r^2} \cong f \cdot \frac{r4 \cdot x_w + r5 \cdot y_w + r6 \cdot z_w + \mathbf{T}_y}{r7 \cdot x_w + r8 \cdot y_w + r9 \cdot z_w + \mathbf{T}_z} \quad (16)$$

where k is the radial distortion coefficient and $r = (X^2 + Y^2)^{1/2}$. This system can be solved using the Singular Value Decomposition (SVD) method (Zhuang & Roth, 1996) through the system of linear equations below once \mathbf{R} , \mathbf{T}_x and \mathbf{T}_y are taking to be known and z_w is null (coplanar points):

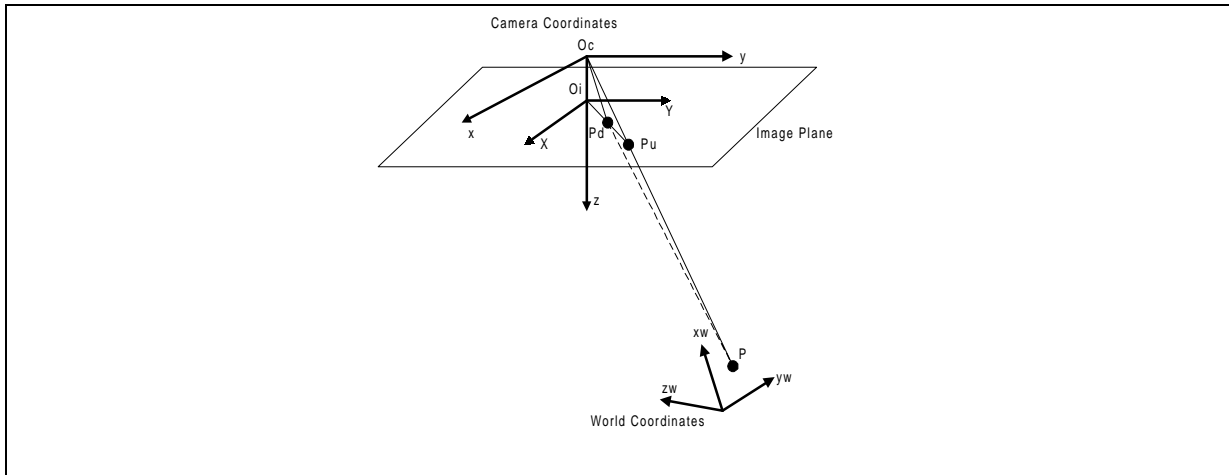


Figure 1 - Camera "pin-hole" model.

$$\begin{bmatrix} -X_i & x_i & -x_i \cdot r_i^2 \end{bmatrix} \begin{bmatrix} Tz \\ f \\ k \cdot f \end{bmatrix} = X_i \cdot w_i \quad (17)$$

where $x_i = r1.xw_i + r2.yw_i + Tx$, $w_i = r7.xw_i + r8.yw_i$, and i is the index correspondent to each calibration point in a grid of points. Details about the photogrammetric model, scale factor and image center calibration can be seen in the paper published by Motta & McMaster (1999b). The vision system used consists of a small CCD camera, 752 x 582 pixels, a 12.5 mm focus length lens, a software to process images to sub-pixel accuracy, and a target of calibration points as shown in Fig. 2. The disposition of two planes in 45 degrees from the horizontal plane avoids angles smaller than 20 degrees between the camera optical axis and the target plane, which may produce ill-conditioned solutions. The measurement system accuracy was assessed experimentally (Motta & McMaster, 1999b) showing to vary from 0.2mm to 0.4mm at distances from the target from 600mm to 1000mm.



Figure 2 - Calibration board and Robot with camera.

4. Experimental Results

Within the PUMA-500 robot workspace three calibration Regions were defined to collect data. In each Region several Volumes were defined with different dimensions. Once two different Volumes had the same volume they had also the same dimensions, whatever Region they were in.

Figure 3 represents graphically all Regions and Volumes within the PUMA-500 workspace. It can be observed that in Region 1 there were two Volumes equal in volume and dimensions, but at different locations (V1 & V1a). The reason for that was to observe the influence of the distance from the camera optical center to the target center point in the calibration overall accuracy, and also how different manipulator configurations within the same workspace Region (different joint motion ranges) could affect results, keeping constant volumes of motion.

4.1 Comparison between the Error Before and After Calibration

The average errors of the PUMA-500 calculated in each of the Volumes within the three measurement Regions are shown in this section. For calculated and measured data in different coordinate systems to be compared to each other, the robot base coordinate frame was moved to coincide with the world coordinate system at the measurement target plate. This procedure was carried out through a recalibration of the robot base in each Volume.

The results from the calculation of average errors and their standard deviation in each Volume of a Region can be seen in the graphs shown in Figures 4, 5 and 6, calculated before and after the calibration.

The results show that the error before and after the calibration tended to increase with the increase of the calibration volume in all Regions. The exception is the calibration volume V1a in Region 1, which presented larger errors than Volume V2 and V3.

The results suggest that V1a may have been placed in a workspace region where the robot arm has larger position errors due to the arm geometry configuration. This observation can be strengthened by the fact that after the calibration all Volumes in Region 1 had little differences in accuracy between each other.

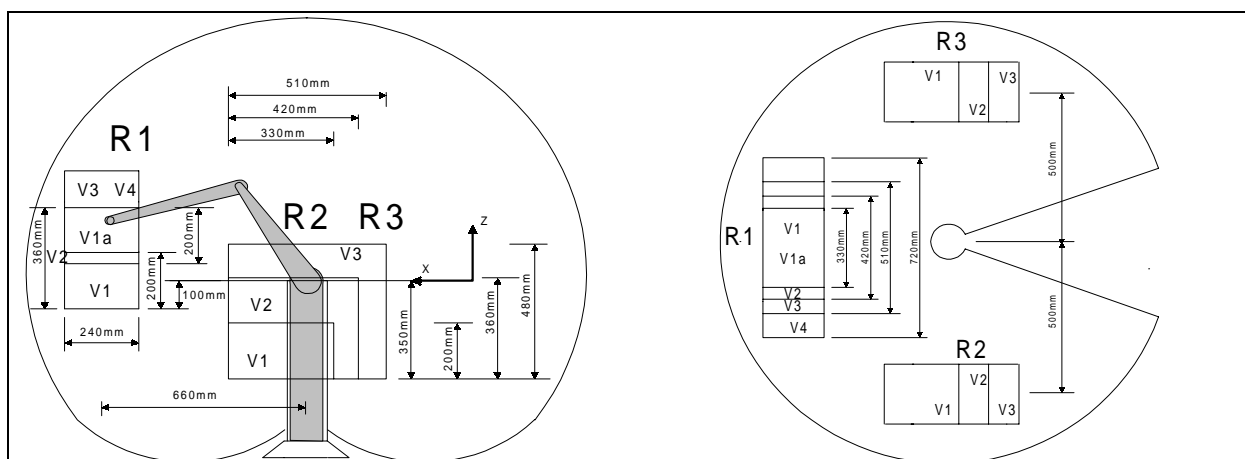


Figure 3 - Side and top view of the PUMA-500 Robot workspace showing Regions, Volumes and their dimensions and locations.

Another observation that stands out is the high average and standard deviation of the errors before the calibration in V4, Region 1, changing to be very small after the calibration.

That breaks the tendency observed in other Volumes. However, the average values of the errors for all Volumes are within an equivalent range of reliability shown by the standard deviations.

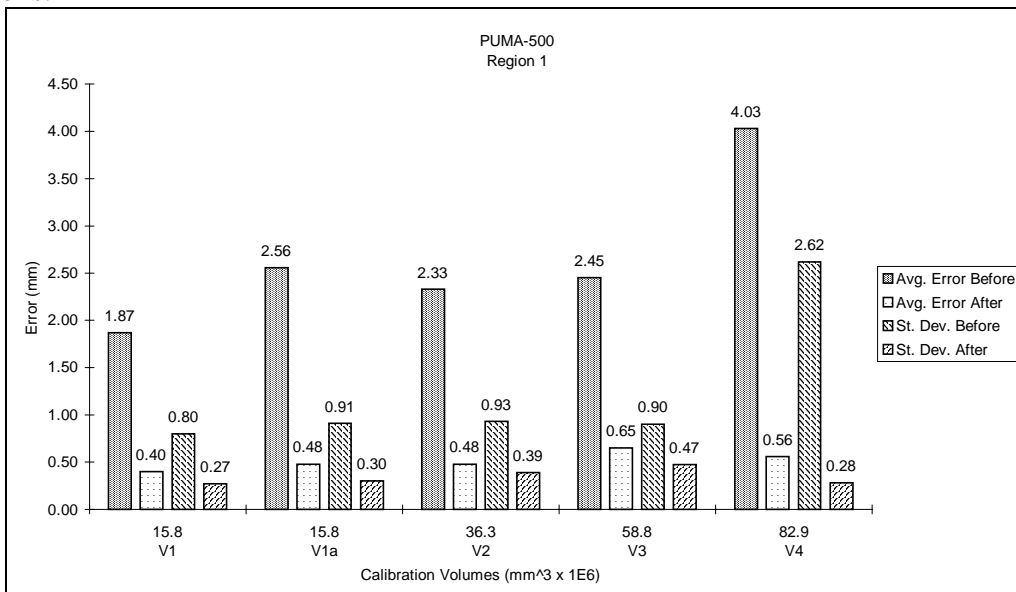


Figure 4 - Average Error and Standard Deviation calculated before and after calibration in each Volume in Region 1.

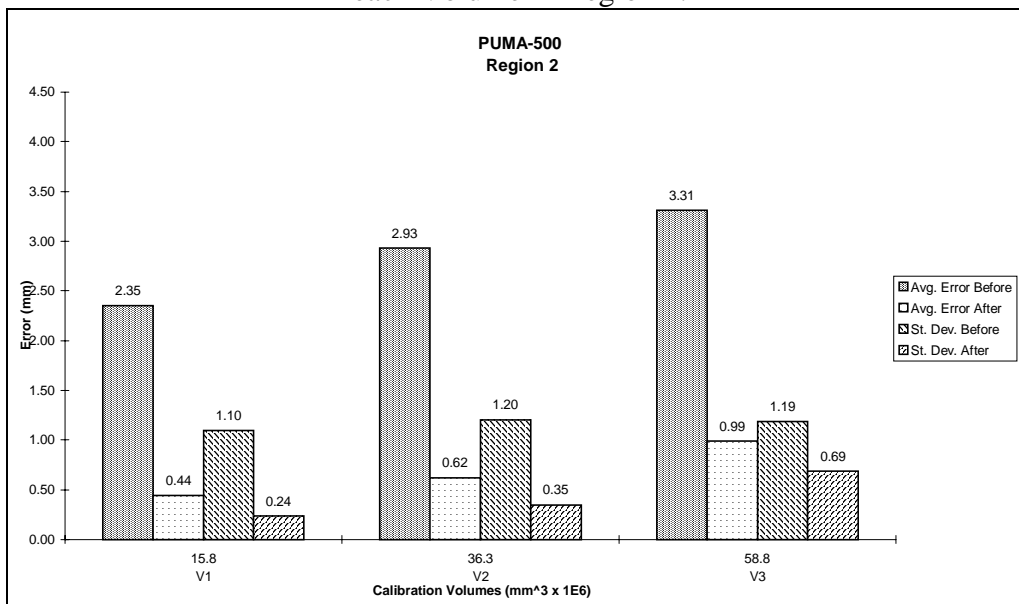


Figure 5 - Average Error and Standard Deviation calculated before and after calibration in each Volume in Region 2.

5. CONCLUSIONS

The calibration system proposed showed to improve the robot accuracy to well below 1mm. The system allows a large variation in robot configurations, which is essential to proper calibration. The robot calibration system approach proposed here stood out to be a feasible alternative to the expensive and complex systems available today in the market, using a single camera and showing good accuracy and ease of use and setup.

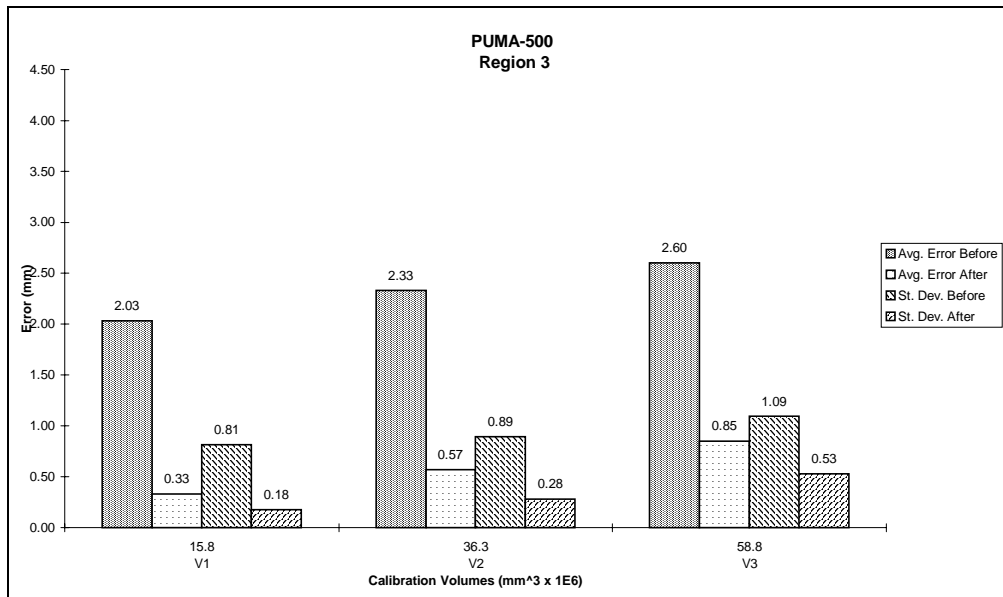


Figure 6 - Average Error and Standard Deviation calculated before and after calibration in each Volume in Region 3.

REFERENCES

- Baker, D. R., 1990, "Some Topological Problems in Robotics", The Mathematical Intelicencer, Vol. 12, No. 1, pp. 66-76.
- Bernhardt, R. , 1997, "Approaches for commissioning time reduction", Industrial Robot, Vol. 24, No. 1, pp. 62-71.
- Dennis, J. E. & Schnabel, R. B. , 1983, "Numerical Methods for Unconstrained Optimization and Nonlinear Equations", 1st ed., Prentice-Hall, New Jersey, USA.
- Driels, M. R. & Pathre, U. S., 1990, "Significance of Observation Strategy on the Design of Robot Calibration Experiments". Journal of Robotic Systems , Vol. 7, No. 2, pp. 197-223.
- Lenz, R. K. & Tsai, R. Y. , 1987, "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology", Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, NC, pp. 68-75.
- McKerrow, P. J. ,1995, "Introduction to Robotics". 1st ed. Addison Wesley, Singapore.
- Motta, J.M. & McMaster, R.S., 1999a, "Modeling, Optimizing and Simulating Robot Calibration with Accuracy Improvement", Journal of the Brazilian Society of Mechanical Sciences, Vol. 3, Sep. 1999, pp. 386-402.
- Motta, J.M. & McMaster, R. S., 1999b, "A 3-D Vision-based Measurement System for Robot Calibration", XV Brazilian Congress of Mechanical Engineering – COBEM99, Nov. 22-26, 1999, Águas de Lindóia, São-Paulo, Brazil.
- Paul, R. P. ,1981, "Robot Manipulators - Mathematics, Programming, and Control", Boston, MIT Press, Massachusetts, USA.
- Press W. H., Teukolsky, S. A., Flannery, B. P. And Vetterling, W. T., 1994, "Numerical Recipes in Pascal – The Art Of Scientific Computer", 1st Ed., Cambridge University Press, New York, USA.
- Schröer, K. ,1994, "Robot Calibration - Closing the Gap Between Model and Reality", Industrial Robot, Vol. 21, No. 6, pp. 3-5.

- Schröder, K.; Albright, S. L. & Grethlein, M. , 1997, “Complete, Minimal and Model-Continuous Kinematic Models for Robot Calibration”, *Robotics & Computer-Integrated Manufacturing*, Vol. 13, No.1, pp. 73-85.
- Tsai, R. Y. , 1987, “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the Shelf TV Cameras and Lenses”, *IEEE International Journal of Robotics and Automation*, Vol. RA-3, No. 4, pp. 323-344.
- Zhuang, H. & Roth, Z. S. , 1996, “Camera-Aided Robot Calibration”, CRC Press, USA.

MONITORAMENTO SIMULTÂNEO DA DEFORMAÇÃO E DA VARIAÇÃO DE RESISTÊNCIA ELÉTRICA VISANDO O CONTRÔLE DE ATUADORES A MEMÓRIA DE FORMA

Carlos José de Araújo

Universidade Federal da Paraíba, Departamento de Engenharia Mecânica

Caixa Postal: 10069

Campina Grande – PB BRASIL

e-mail: carlos@dem.ufpb.br

Michel Morin

Gérard Guénin

INSA de Lyon, Laboratoire GEMPPM, Bat. 502, 1^{er} étage

20, Av. Albert Einstein – 69621 Villeurbanne FRANÇA

Resumo:

O efeito memória de forma (EMF) que algumas ligas metálicas apresentam permitem que esses materiais sejam considerados intrinsecamente como atuadores que podem produzir um trabalho mecânico externo. A curva característica destes atuadores, deformação (ϵ) versus temperatura (T), apresenta uma histerese térmica (H_T) que dificulta o controle do deslocamento destes elementos. Este trabalho tem por objetivo verificar a possibilidade de controlar tais atuadores através de medidas simultâneas da deformação e da variação de resistência elétrica ($\Delta R/R$) em função da temperatura. Os resultados obtidos mostram que obtém-se uma relação linear entre ϵ e $\Delta R/R$ quando se elimina a temperatura. No entanto, a curva $\epsilon \times \Delta R/R$ obtida também apresenta uma histerese que é função da carga externa. Demonstra-se experimentalmente que essa histerese pode ser completamente eliminada após alguns ciclos de aquecimento e resfriamento do atuador, viabilizando a idéia do controle de seu deslocamento através do monitoramento de sua resistência elétrica.

Palavras chave: Fios, efeito memória de forma, atuadores, controle, deformação, resistência elétrica.

1. INTRODUÇÃO

As ligas com efeito memória de forma (EMF) representam uma nova classe de materiais capazes de produzir um deslocamento considerável sob carga quando submetidas a uma variação de temperatura. Este fenômeno pode ser reproduzido de maneira cíclica e portanto tais materiais podem ser usados como atuadores termomecânicos para aplicações

na robótica (Furuya & Shimada, 1990; Melton, 1998). A recuperação da deformação associada a um dado carregamento mecânico está diretamente relacionada a uma transformação de fase do tipo martensítica reversível ocorrendo no interior do material durante aquecimento e resfriamento (Funakubo, 1987). Infelizmente, a curva deslocamento versus temperatura característica desses atuadores não é linear e apresenta uma histerese térmica que dificulta o seu controle. Esta variação macroscópica da forma em função da temperatura é acompanhada da variação simultânea de uma série de propriedades físicas associadas a transformação de fase no material (Harrison, 1990), dentre elas a resistividade elétrica. Por essa razão, alguns pesquisadores (Honma *et al.*, 1984; Pruski & Kihl, 1993) sugerem que nas aplicações destes materiais como atuadores termomecânicos, um sistema de controle realimentado (*feedback control system*) pode ser desenvolvido através do acompanhamento da variação de resistividade elétrica observada durante a transformação (deslocamento) do material com a temperatura. Mesmo que essa proposta seja extremamente interessante, poucos testes foram efetivamente realizados e a possibilidade de controle do atuador com EMF depende bastante do material e dos tratamentos termomecânicos aplicados (Hesselbach *et al.*, 1994; Reynaerts *et al.*, 1995 e Mertmann & Hornbogen, 1997).

Em termos gerais, este trabalho se propõe principalmente a obter resultados experimentais do acoplamento entre a deformação associada ao EMF sob carga e a variação de resistência elétrica de fios com memória de forma submetidos a ciclagem térmica com o objetivo de verificar as potencialidades deste procedimento para o controle efetivo de elementos a memória de forma.

2. CONDIÇÕES EXPERIMENTAIS

A figura 1 mostra um esquema do sistema atuador com EMF e o resultado teoricamente esperado de um acoplamento entre as medidas de deslocamento, expresso pela deformação (ϵ), e variação de resistência elétrica do atuador ($\Delta R/R$).

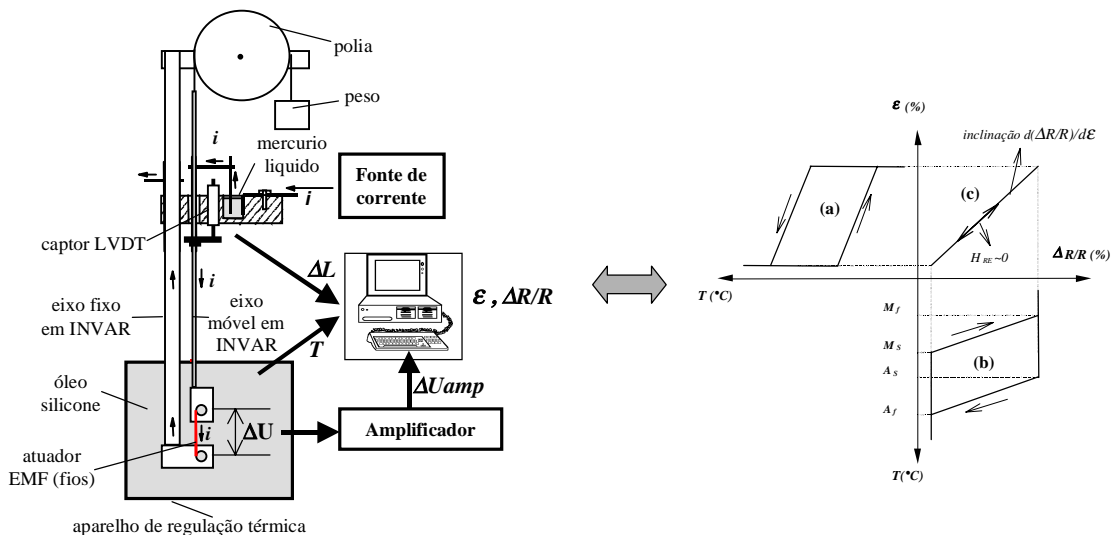


Figura 1 – Representação esquemática do sistema atuador com EMF para monitoramento simultâneo do deslocamento (ϵ) e da variação de resistência elétrica ($\Delta R/R$).

Observa-se que o sistema montado consiste basicamente de um fio atuador com EMF carregado em tração uniaxial constante por intermédio de pesos e podendo ser continuamente aquecido e resfriado através de um banho de óleo silicone termicamente regulado, enquanto que uma fonte de corrente contínua faz passar uma corrente elétrica constante (i) diretamente no fio atuador. O deslocamento vertical do fio (ΔL) é medido por um captor LVDT (*Linear Variable Differential Transducer*) enquanto que o sinal de tensão elétrica (ΔU) originado pela passagem da corrente é medido e amplificado (ΔU_{amp}). Uma placa de aquisição de 12bits, instalada em um computador, armazena os dados de temperatura (T), ΔL e ΔU_{amp} e um programa calcula os valores de ϵ e $\Delta R/R$. Verifica-se na ilustração da figura 1 a presença de uma histerese em temperatura na curva característica ϵ - T (curva a) assim como na curva $\Delta R/R$ - T (curva b), a qual pode ser supostamente contornada quando se elimina a temperatura e obtém-se o comportamento ϵ - $\Delta R/R$ (curva c), linear, de inclinação $d(\Delta R/R)/d\epsilon$ e com uma histerese (H_{RE}) aproximadamente nula.

Os fios atuadores usados no equipamento da figura 1 tiveram comprimentos entre 27 e 32mm. São fios de uma liga Ti-Ni-Cu com EMF fornecidos pela empresa belga *Advanced Materials Technologies Inc.* Todas as amostras apresentam a composição nominal Ti-45.0Ni-5.0Cu (at%), diâmetros de 0.55 mm e foram obtidas por trefilação à frio com redução total de área de 40% durante a fabricação. Os fios foram recozidos à 425°C durante 1h e posteriormente resfriados ao ar ambiente.

Os valores do carregamento mecânico aplicado aos atuadores corresponderam a tensões situadas entre 75MPa e 400MPa. A ciclagem térmica foi realizada entre 10°C e 154°C, independentemente da carga aplicada.

3. RESULTADOS E DISCUSSÕES

Um comportamento experimental típico, obtido com o sistema da figura 1 durante o primeiro ciclo térmico, é mostrado na figura 2 para uma carga equivalente a 175MPa. A figura 2(a) mostra a curva correspondente a deformação sob carga do atuador a memória de forma. Verifica-se que a contração durante o aquecimento permite uma recuperação por EMF da ordem de 6% e o atuador não retorna completamente a posição original a alta temperatura devido a existência de uma deformação acumulada de aproximadamente 1% após o aquecimento. Estes valores de deformação dependem da carga aplicada (De Araújo, 1999). A curva característica da evolução da resistência elétrica com a temperatura é mostrada na figura 2(b). As temperaturas de transformação do atuador, assim como sua histerese térmica (H_T), são determinadas pela aplicação do método das tangentes as curvas das figuras 2(a) e 2(b). Estas figuras revelam que os valores de histerese são praticamente idênticos nas curvas ϵ - T e $(\Delta R/R)$ - T , mas isso não é suficiente para eliminar a presença de uma histerese (H_{RE}) entre ϵ e $(\Delta R/R)$ quando estas medidas são acopladas, como mostra a figura 2(c). No entanto, a relação entre estas duas propriedades é linear e de inclinação $d(\Delta R/R)/d\epsilon$ praticamente igual no aquecimento e no resfriamento. Na realidade a presença de H_{RE} depende da forma das curvas das figuras 2(a) e 2(b), mas Pruski & Kihl (1993) sugerem um método para eliminá-la eletronicamente durante o emprego desses atuadores. A dependência de H_{RE} e da inclinação $d(\Delta R/R)/d\epsilon$ com relação a carga aplicada é mostrada na figura 3.

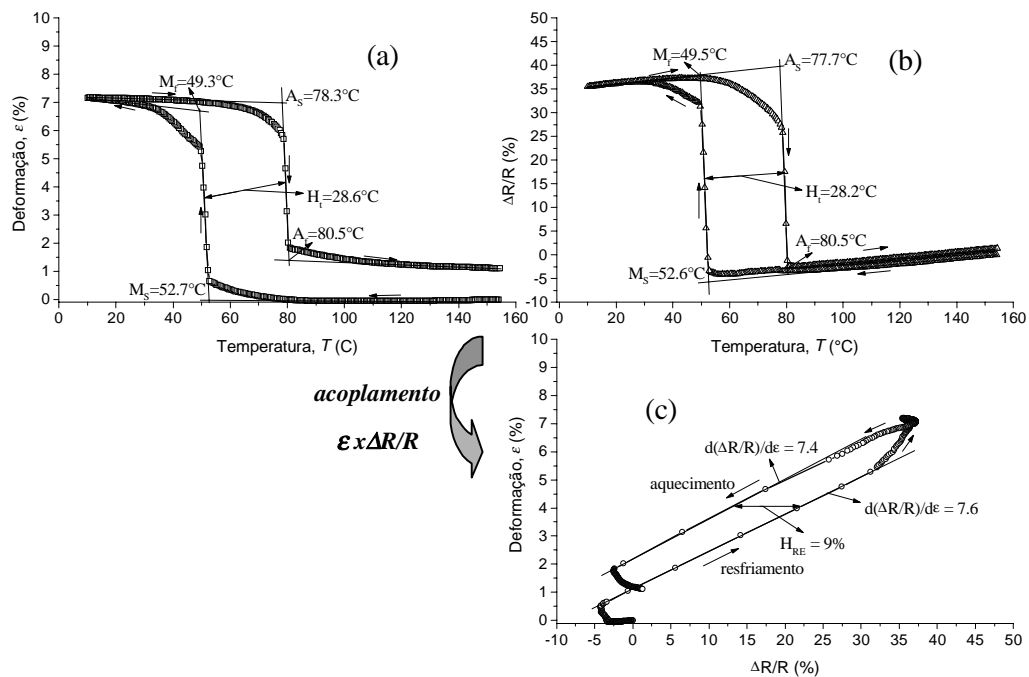


Figura 2 – Acoplamento entre as medidas de deformação e resistência elétrica durante um ciclo térmico simples para uma carga aplicada equivalente a 175MPa. (a) curva ϵ -T. (b) curva $(\Delta R/R)$ -T. (c) curva ϵ - $(\Delta R/R)$.

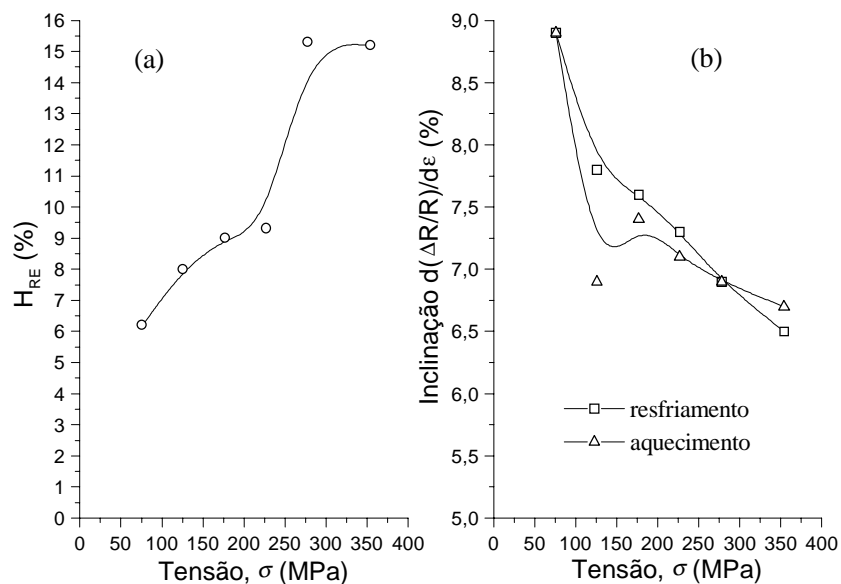


Figura 3 – Variação dos parâmetros do acoplamento ϵ - $(\Delta R/R)$ com a carga aplicada. (a) histerese H_{RE} . (b) inclinação $d(\Delta R/R)/d\epsilon$.

Verifica-se um aumento de H_{RE} e uma diminuição da inclinação $d(\Delta R/R)/d\varepsilon$ com a carga aplicada. Este aumento de H_{RE} torna-se mais pronunciado a partir de 250MPa, onde observa-se uma anomalia na curva ε -T que não existe sobre a curva $(\Delta R/R)$ -T (De Araújo, 1999). Este comportamento de H_{RE} é prejudicial ao controle do deslocamento do fio atuador, mas em teoria poderia ser artificialmente contornado pelo método sugerido por Pruski & Kihl (1993).

Tendo em vista que uma eventual aplicação destes atuadores na robótica implica na realização de operações sucessivas de aquecimento e resfriamento sob carga, os fios com EMF montados no sistema da figura 1 foram também submetidos a ciclagem térmica. Os resultados obtidos durante uma quinzena de ciclos para uma carga equivalente a 175MPa são importantes, como revela a figura 4.

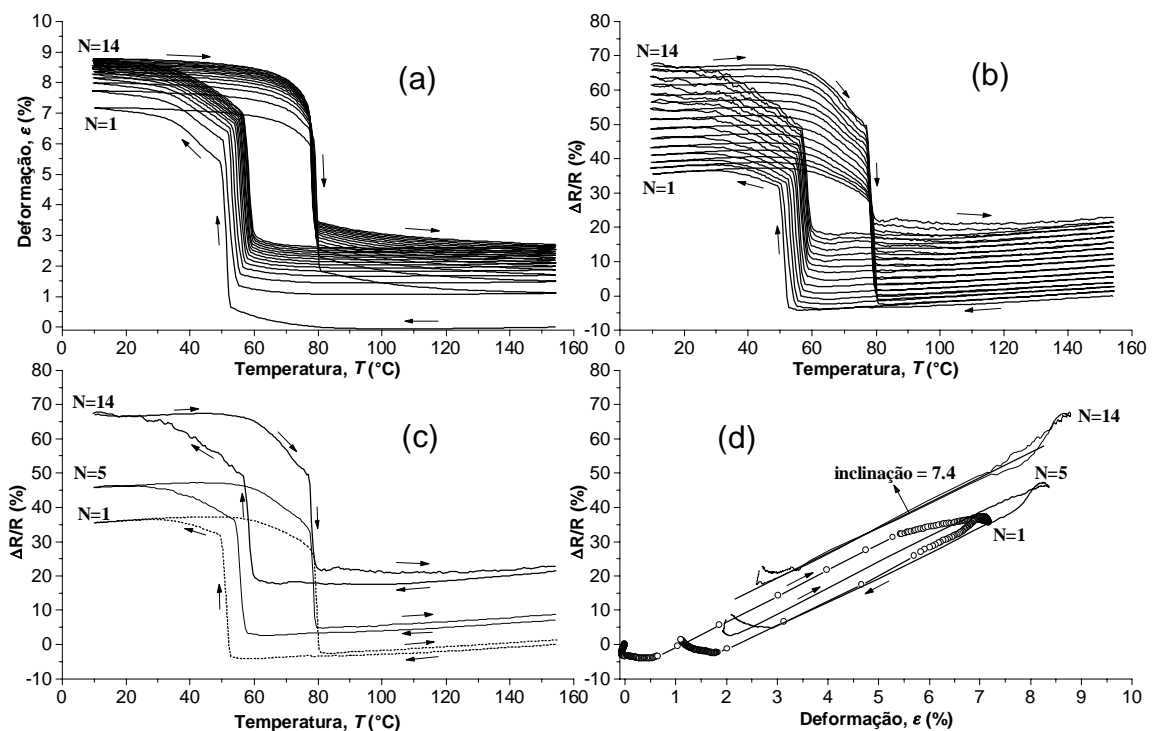


Figura 4 – Evolução do comportamento ε -T- $\Delta R/R$ para 175MPa durante aproximadamente 15 ciclos térmicos. (a) curvas ε -T. (b) curvas $(\Delta R/R)$ -T. (c) algumas curvas $(\Delta R/R)$ -T selecionadas em (b). (d) algumas curvas do acoplamento ε - $(\Delta R/R)$.

Observa-se nas curvas ε -T da figura 4(a) uma diminuição da histerese em temperatura H_T combinada com um deslocamento vertical destas curvas que corresponde a um aumento permanente de comprimento do fio atuador. Este fenômeno tende a saturar-se após os primeiros ciclos térmicos. O comportamento das curvas $(\Delta R/R)$ -T é idêntico em termos de H_T , porém não existe uma tendência clara de saturação da evolução de $\Delta R/R$, como mostra a figura 4(b). Estas evoluções são resultantes da introdução de defeitos microestruturais no interior do material, os quais afetam a transformação martensítica

responsável pelo EMF. A figura 4(c) revela claramente uma mudança na forma das curvas $(\Delta R/R)$ -T durante a ciclagem, fazendo com que a histerese H_{RE} existente no acoplamento ε - $(\Delta R/R)$ seja continuamente reduzida e finalmente anulada após alguns ciclos, como mostra a figura 4(d). Além disso, comparando as figuras 2(c) e 4(d) constata-se que a inclinação $d(\Delta R/R)/dT$ praticamente não é afetada pela ciclagem. Este comportamento traduzido pela figura 4 existe para todos os níveis de tensão estudados (entre 75MPa e 400MPa). No que diz respeito ao controle do fio atuador com EMF, a ausência de histerese entre as duas propriedades monitoradas, assim como a relação linear entre elas, são aspectos fundamentais que revelam a importância do procedimento de ciclagem sob carga, alternativamente a metodologia proposta por Pruski & Kihl (1993). Talvez o único inconveniente seja o deslocamento da curva ε - $(\Delta R/R)$ evidenciado pela figura 4(d), mas esse fenômeno tende a saturar-se para um número de ciclos ligeiramente superior ao realizado nesse trabalho.

Vale ressaltar que o comportamento dos atuadores à memória de forma depende bastante da sua história termomecânica e não existem garantias de que fios semelhantes com EMF (mesma composição química, por exemplo), submetidos a tratamentos termomecânicos diferentes daquele mencionado no item 2, apresentem o mesmo comportamento mostrado na figura 4. Assim sendo, torna-se importante dirigir estudos semelhantes a estes para outros tipos de ligas com EMF e diversos tratamentos termomecânicos, visando determinar as condições ótimas para o comportamento ε - $(\Delta R/R)$.

4. CONCLUSÕES

A realização de medidas simultâneas de deformação e variação de resistência elétrica em função da temperatura, em fios de uma liga com EMF trabalhando sob carga, permitiu demonstrar experimentalmente a viabilidade da idéia de controlar o deslocamento destes atuadores através do monitoramento de sua resistência elétrica. Para o primeiro ciclo térmico, a curva ε - $(\Delta R/R)$ obtida apresenta uma relação linear entre as duas propriedades tanto no resfriamento quanto durante o aquecimento, porém existe ainda uma histerese entre as duas medidas. Para todos os níveis de carga aplicada ao fio atuador, o comportamento ideal para o controle, isto é, relação linear e ausência de histerese entre as duas grandezas durante resfriamento e aquecimento, é obtido após a realização de alguns ciclos térmicos sucessivos, eliminando a necessidade de contornar artificialmente o problema da histerese na curva característica ε - $(\Delta R/R)$. Esta relação unívoca entre estas duas propriedades permitiria determinar a posição do fio atuador a memória de forma simplesmente através da medida de sua resistência elétrica, independentemente do fato do mesmo está sendo aquecido ou resfriado no instante considerado.

Agradecimentos

C. J. De Araújo agradece o apoio da Capes durante seu doutoramento no INSA de Lyon – França (Proc. nº 0073/95-13).

REFERÊNCIAS

- De Araújo, C. J., 1999, Comportement cyclique de fils en alliage à mémoire de forme Ti-Ni-Cu: analyse electro-thermomécanique, dégradation et fatigue par cyclage thermique sous contrainte, Thèse de Doctorat, INSA de Lyon, France, 177p.
- Funakubo, H., 1987, Shape Memory Alloys, Gordon & Breach Publishers, 270p.
- Furuya, Y. & Shimada, H., 1990, Shape memory actuators for robotic applications. In Engineering Aspects of Shape Memory Alloys. Edited by T. W. Duerig. Butterworth-Heinemann, London, pp. 338-354.
- Harrison, J. D., 1990, Measurable changes concomitant with the shape memory effect transformation. In Engineering Aspects of Shape Memory Alloys. Edited by T. W. Duerig. Butterworth-Heinemann, London, pp. 106-111.
- Hesselbach, J., Pittschellis, R., Stork, H., Hornbogen, E. and Mertmann, M., 1994, Optimization and control of electrically heated shape memory actuators. In Proceedings of 4th International Conference on New Actuators, Bremen, Germany, pp. 337-340.
- Honma, D., Miwa, Y. and Iguchi, N., 1984, Application of shape memory effect to digital control actuator, Bulletin of JSME, Vol.27, n°.30, pp. 1737-1742.
- Melton, K. N., 1998, General applications of SMA's and smart materials. In Shape Memory Materials, Edited by K. Otsuka and C. M. Wayman, Cambridge University Press, pp. 220-239.
- Mertmann, M. & Hornbogen, E., 1997, Grippers for the micro assembly containing shape memory actuators and sensors, Journal de physique IV, colloque C5, supplément au Journal de Physique III, Vol 7, n°11, pp. C5-621 - C5-626.
- Pruski, A. & Kihl, H., 1993, Shape memory alloy hysteresis, Sensors and Actuators A, Vol.36, pp. 29-35.
- Reynaerts, D., Peirs, J. and Van Brussel, H., 1995, Production of shape memory alloys for microactuation, Journal of Micromech. and Microeng., Vol.5, pp. 150-152.

APLICAÇÃO DE MODELOS HÍBRIDOS NA INTEGRAÇÃO DE SISTEMAS DE AR CONDICIONADO EM EDIFÍCIOS INTELIGENTES

Emília Villani

Paulo Eigi Miyagi

Universidade de São Paulo, Escola Politécnica, Depto. de Engenharia Mecatrônica e de Sistemas Mecânicos.

São Paulo, SP, Brasil.

e-mail: emiliav@usp.br

RESUMO

Considerando sistemas de ar condicionado em edifícios inteligentes como uma classe de sistemas híbridos, este trabalho introduz e analisa a aplicação de diferentes técnicas para a modelagem de sistemas híbridos derivadas de Redes de Petri. O objetivo é evidenciar as principais vantagens e desvantagens da utilização de cada técnica na modelagem e integração do sistema de ar condicionado com os demais sistemas prediais em um edifício inteligente.

Palavras-chave: Sistemas Híbridos, Redes de Petri, Edifícios Inteligentes, Ar condicionado.

1. INTRODUÇÃO

Um "Edifício Inteligente" (EI) pode ser definido como aquele onde a introdução de novas tecnologias e a integração de sistemas são exploradas de modo a maximizar a produtividade dos ocupantes, com utilização racional de recursos e minimização do custo [Becker, 1995; Arkin & Paciuk, 1995; Fujie & Mikami, 1991].

Em particular, em um EI o sistema de ar condicionado tem como objetivo a garantia de conforto térmico aos usuários e operadores do edifício, fator essencial no aumento da potencialidade de trabalho e otimização de desempenho [Jabardo, 1984; Fanger, 1970].

No edifício tradicional, o sistema de ar condicionado opera independentemente de outros sistemas prediais, como o de iluminação, de controle de acesso, etc. Entretanto, considerando edifícios inteligentes, um Sistema de Gerenciamento do Edifício (Building Management System - BMS) deve realizar a integração entre diversos subsistemas, permitindo o compartilhamento eficiente de informações. Para o sistema ar condicionado, esta integração possibilita o acesso às informações relativas a fatores diretamente ligados à produção de calor¹. Assim, através da utilização e processamento destas informações, pode-se, por exemplo, determinar qual a melhor atitude a ser tomada em relação ao sistema de ar condicionado.

Considerando esta possibilidade de integração, o presente trabalho introduz o conceito de sistemas híbridos para a modelagem de sistemas de ar condicionado em EI.

¹ Exemplos: número de ocupantes, número de máquinas em funcionamento, etc.

A utilização de abordagens híbridas é um tema de pesquisa relativamente recente. Enquanto a teoria de controle clássico fornece ferramentas para análise e síntese de sistemas contínuos, e abordagens equivalentes tratam de sistemas a eventos discretos, não existe uma abordagem formalizada e consolidada para modelagem e análise de sistemas com comportamento dinâmico discreto e contínuo simultaneamente [Lemmon, He & Markovsky, 1999]. Assim, este trabalho analisa diferentes propostas para modelagem de sistemas híbridos baseando-se na sua aplicação para a modelagem do sistema de ar condicionado em EI.

Este artigo está organizado da seguinte forma. A seção 2 apresenta a definição de sistemas híbridos e justifica a necessidade de se caracterizar o sistema de ar condicionado como tal. A seção 3 aborda o problema de modelagem de sistemas híbridos. São apresentadas diferentes técnicas de representação de sistemas híbridos, aplicando-as para modelagem de um exemplo de sistema de ar condicionado. A seção 4 faz uma comparação entre as ferramentas apresentadas, ressaltando suas principais vantagens e desvantagens na modelagem de sistemas de ar condicionado. Finalmente, a seção 5 apresenta algumas considerações finais.

2. SISTEMAS HÍBRIDOS

Os sistemas podem ser classificados de acordo com diversos critérios [Ho, 1987]. Uma possível dicotomia é a divisão dos sistemas em Sistemas de Variáveis Contínuas (SVC), cuja principal característica é a continuidade no tempo de suas variáveis, e Sistemas a Eventos Discretos (SED), cuja modificação do estado, ocorre em função de eventos discretos, como início do funcionamento de uma máquina, chegada de uma mensagem, etc. [Miyagi, 1996]. Dentro desta mesma classificação, foram também definidos os sistemas híbridos, onde coexistem simultaneamente característica de SVC e de SED [Antsaklis & Nerode, 1998].

O sistema de ar condicionado envolve o controle de variáveis como a temperatura, que variam de forma contínua no tempo, caracterizando o sistema como SVC. Na maioria dos casos, o controle do sistema de ar condicionado também é contínuo, sendo comumente adotadas estratégias de controle do tipo proporcional (P) e proporcional-integral (PI).

No entanto, considerando o contexto de edifícios inteligentes, torna-se imprescindível a interação com os demais sistemas prediais com características predominantemente discretas, como sistema de controle de acesso, sistema de iluminação, etc. Este sistemas são representados por variáveis discretas (ex. número de pessoas em uma sala, número de lâmpadas acesas, etc.), que variam de acordo com a ocorrência de eventos discretos (ex.: entrada de uma pessoa, acionamento de um equipamento, etc.). Isto implica em considerar-se o sistema também como SED.

O sistema resultante apresenta assim características contínuas e discretas simultaneamente, interagindo entre si, onde os eventos discretos influenciam diretamente as variáveis contínuas que afetam diretamente o conforto térmico de um ambiente. Nestes termos, o sistema deve ser representado como um sistema híbrido. A Figura 1 ilustra o que foi apresentado.

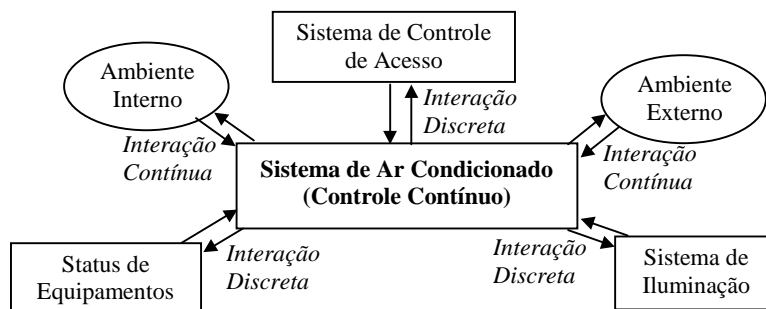


Figura 1 - Concepção híbrida de um sistema de ar condicionado de um EI.

2.1 Modelagem

Têm sido propostas diferentes abordagens para a modelagem, análise e controle de sistemas híbridos. Genericamente, as abordagens diferem entre si no que se refere à complexidade da parte contínua e discreta e à finalidade, se para análise ou controle [Antsaklis & Nerode, 1998].

Algumas abordagens consistem em extensões de modelos contínuos, nas quais são incluídas variáveis que podem ser modificadas de forma descontínua no tempo. Outras consistem na modificação de técnicas de modelagem utilizadas em SED, onde são introduzidos novos elementos que permitem a representação da dinâmica contínua.

Para o escopo do presente trabalho optou-se pelas técnicas derivadas das propostas para modelagem de SED, uma vez que o enfoque do presente trabalho está na integração do sistema de ar condicionado com os demais sistemas do edifício, que é de carácter predominantemente discreto. Entre os modelos para SED as Redes de Petri se destacam no que se refere a facilidade de interpretação do modelo e a representação de forma hierárquica. Assim, as técnicas apresentadas a seguir são derivadas das Redes de Petri.

3. MODELAGEM DE SISTEMAS DE AR CONDICIONADO

Para o estudo dos diversos tipos de redes de Petri considerou-se o sistema da Figura 2. Neste sistema um ventilador é responsável pela retirada do ar do ambiente e pela sua condução através da serpentina de água fria, onde o ar é resfriado, sendo posteriormente insuflado no ambiente. A temperatura do ambiente é controlada através da variação da temperatura do ar insuflado. Esta variação é realizada através de uma válvula de três vias que desvia parte do fluxo de água enviado à serpentina. A temperatura da água fornecida é considerada constante. O posicionamento da válvula é realizado através de um controlador do tipo PI.

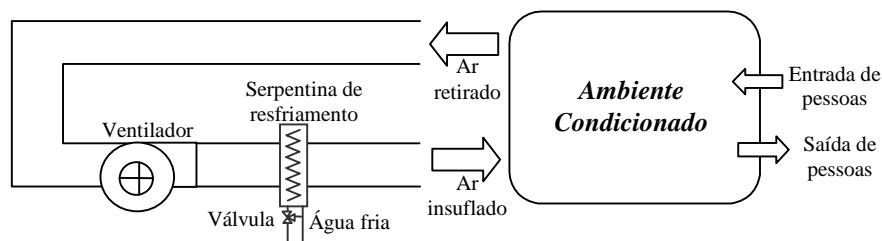


Figura 2 – Sistema de ar condicionado.

Considerou-se o ambiente condicionado como adiabático, tendo como fatores que influenciam na carga térmica o ar insuflado e a presença de pessoas no ambiente.

Obviamente o exemplo é bastante simplificado, uma vez que não considera renovação de ar, trocas de calor através de paredes, infiltração, calor liberado por iluminação e equipamentos, etc. No entanto, a utilização deste exemplo foi adotada pois, como o escopo do presente trabalho é a análise de técnicas para modelagem de sistemas híbridos, a consideração de sistemas mais complexos agregaria pouca informação.

A seguir apresenta-se os modelos do sistema de ar condicionado considerado utilizando-se três das técnicas mais representativas derivadas de redes de Petri para sistemas híbridos.

3.1 Redes de Petri Híbridas

As Redes de Petri Híbridas [Alla & David, 98] foram definidas através da adição, aos lugares e transições discretos (Figura 3 a e c), dos seguintes elementos:

- lugares contínuos: cuja marcação é um número real² (Figura 3 b);
- transições contínuas: as quais são associadas taxas de disparo representando um fluxo contínuo de marcas (Figura 3 d);

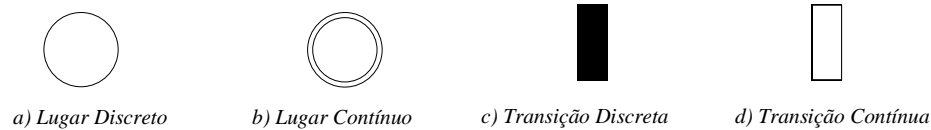


Figura 3 - Elementos da Rede de Petri Híbrida.

Observa-se que a taxa de disparo de uma transição contínua pode ser constante ou ser uma função que utiliza como variáveis a marcação de lugares discretos e contínuos. A Figura 4 ilustra a evolução no tempo (t) da marcação (m_{P_1}) de um lugar contínuo (P_1) conectado a uma transição contínua (T_1).

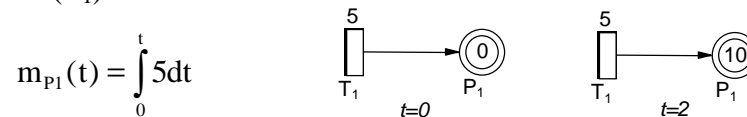


Figura 4 – Evolução no tempo da marcação de um lugar contínuo.

A modelagem do sistema da Figura 2 através das Redes de Petri Híbrida é apresentada a seguir.

Modelo do ventilador (Figura 5): o ventilador pode assumir dois estados discretos: ligado ou desligado. Quando o ventilador é ligado, a vazão de ar que circula pelo sistema de condicionamento do ar (v_{ar}) assume um determinado valor constante (γ). Quando o ventilador é desligado a vazão é zero.

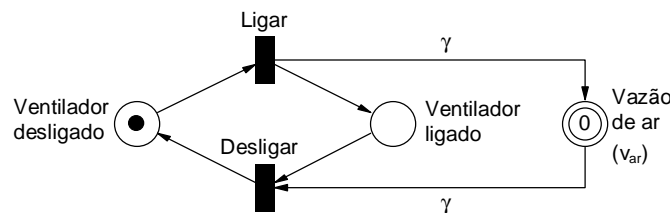


Figura 5 – Modelo em Rede de Petri Híbrida do ventilador.

Modelo da serpentina (Figura 6): a taxa de disparo (\dot{T}_{ar}) da transição contínua é função da temperatura do ar na entrada da serpentina, que é igual a temperatura no ambiente (T_{amb}), da vazão de ar (v_{ar}), da temperatura e da vazão da água fornecida a serpentina ($T_{água}$ e $v_{água}$), consideradas constante, da posição da válvula de três vias (P) e da constante de tempo (τ) do sistema.

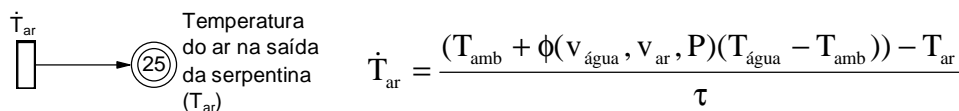


Figura 6 – Modelo em Rede de Petri Híbrida da serpentina.

² Originalmente a marcação do lugar contínuo foi definida como um número real não negativo [Alla & David, 1998], mas diversos autores a redefiniram para número real, de modo a possibilitar a representação de equações diferenciais de primeira ordem, como em [Drath, 1998], e [Demongodin & Koussoulas, 1998].

A função ϕ da expressão de \dot{T}_{ar} é determinada a partir das características da serpentina [Salsbury, 1996], e baseia-se no conceito de condutância global, onde a troca de calor é calculada através do coeficiente global de transferência de calor [Incropera, 1990].

Modelo do controlador PI (Figura 7): como o ventilador, o controlador também pode assumir dois estados discretos (ligado e desligado). Quando desligado o controlador coloca a válvula na posição fechada ($P=0$). Quando aberta a posição da válvula é determinada através da Equação 1, onde K_M , K_P , M_I são as constantes do controlador e λ é a temperatura desejada no ambiente.

$$\begin{aligned}
 P &= K_P * e + K_I * Ie + K_M \\
 e &= (T_{amb} - \lambda) \\
 Ie &= \int (T_{amb} - \lambda).dt
 \end{aligned}
 \tag{1}$$

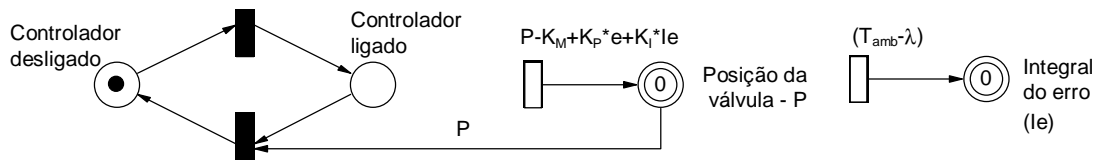


Figura 7 – Modelo em Rede de Petri Híbrida do controlador.

Modelo do ambiente (Figura 8): a evolução da temperatura do ambiente (T_{amb}) no tempo é determinada pela taxa de disparo da transição contínua (\dot{T}_{amb}), apresentada na Equação 2. Esta taxa é calculada a partir do calor introduzido/retirado do ambiente (Q_{total}), em função do número de pessoas (N_P), da carga térmica produzida por uma pessoa (K_P), considerada constante, da vazão e temperatura do ar insuflado pelo sistema de ar condicionado, do volume de ar no ambiente (vol) e da densidade (ρ) e do calor específico do ar a pressão constante (c_p).

$$\dot{T}_{amb} = \frac{Q_{total}}{vol * \rho * c_p} = \frac{(v_{ar} * \rho * c_p * (T_{ar} - T_{amb}) + N_P * C_P)}{vol * \rho * c_p}
 \tag{2}$$

Observa-se que esta expressão considera comportamento de gás perfeito e despreza diferenças de umidade entre o ar insuflado e o ar no ambiente.

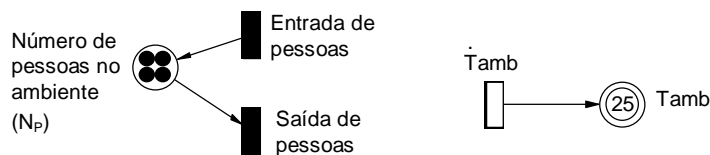


Figura 8 – Modelo em Rede de Petri Híbrida do ambiente.

3.2 Rede de Petri Híbridas de Alto Nível (HyNet)

Esta rede foi desenvolvida a partir de uma classe especial de redes de Petri de alto nível orientada a objetos chamada Thorns (Timed Hierarchical Object-Related Nets) e é definida em [Wieting, 1996].

Nesta rede utilizam-se conceitos de modelagem orientada a objetos [Meyer, 1997]. *Marcas* são “objetos” pertencentes a “classes” elementares ou a “classes” definidas. A linguagem utilizada na definição de objetos e classes é baseada em C++. Por classes

elementares entende-se as classes já definidas em C++, como “int”, “bool”, “char”, entre outras. A definição de uma nova classe é realizada através da definição de seus “atributos” e “métodos”. Os “atributos” são definidos a partir das classes elementares. Os “métodos” são definidos através de operações matemáticas.

Para a modelagem do sistema da Figura 2 são definidas 3 classes:

```

Class Fluxo_de_Ar {float var; float Tar;}
Class Ambiente {int Np; float Tamb;}
Class Controlador {float P;}
    
```

onde as variáveis entre chaves representam os “atributos” da “classe”.

Os elementos presentes na rede são *transições* discretas e contínuas e *lugares*. Aos *lugares* associa-se uma “classe”. A “classe” pode ser uma “classe” elementar (“int”, “char”, etc.) ou um “classe” definida.

Arcos têm um nome de variável a ele associado. O nome é utilizado para referenciar em uma transição os objetos de um lugar. Os arcos podem ser arcos orientados ou arcos contínuos (não orientados). Os arcos contínuos não retiram/adicionam objetos aos lugares, apenas modificam seus atributos.

Às transições são associadas *condições de ativação*, que impõem condições adicionais para o disparo da transição, e *ações de disparo*, que determinam os objetos introduzidos nos lugares de saída a partir dos objetos retirados dos lugares de entrada. Para transições contínuas a *ação de disparo* é constituída por um sistema de equações algébricas diferenciais.

O modelo em Rede de Petri de Alto Nível para o sistema da Figura 2 é apresentado na Figura 9.

As ações de disparo associadas às transições contínuas são:

$$\begin{aligned}
 Ad_1 : \frac{d(x.T_{ar})}{dt} &= \frac{(n.T_{amb} + \phi(v_{\acute{a}gua}, x.v_{ar}, o.P)(T_{\acute{a}gua} - o.T_{amb})) - x.T_{ar}}{\tau} \\
 Ad_2 : l.P &= K_P * (m.T_{amb} - \lambda) + K_I * \int (m.T_{amb} - \lambda).dt + K_M \\
 Ad_3 : \frac{d(c.T_{amb})}{dt} &= \frac{(a.v_{ar} * \rho * c_p (a.T_{ar} - c.T_{amb}) + c.N_P * C_P)}{vol * \rho * c_p}
 \end{aligned}$$

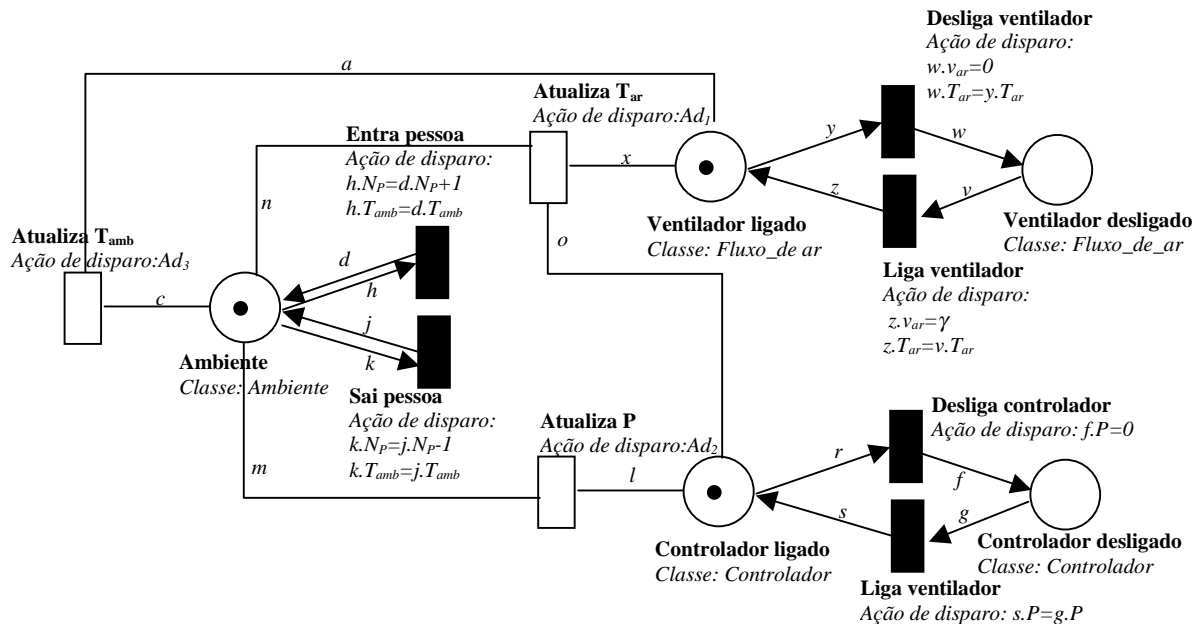


Figura 9 – Modelo em Rede de Petri de Alto Nível.

3.3 Rede de Petri Predicado/Transição Diferenciais (rede PTD)

Nas redes PTD os *lugares* representam configurações do sistema. A cada rede associa-se um vetor (V) com as variáveis contínuas a serem modeladas. A cada *lugar* (P_X) é associado um sistema de equações diferenciais (S_{P_X}) para a evolução das variáveis contínuas no tempo quando o sistema está naquela configuração. Quando um *lugar* está marcado o sistema de equações a ele associado determina a evolução no tempo das variáveis contínuas. Através do *disparo* das *transições* chaveia-se os sistemas de equações [Champagnat, 1998].

A Figura 10 apresenta um modelo em rede PTD do sistema da Figura 2.

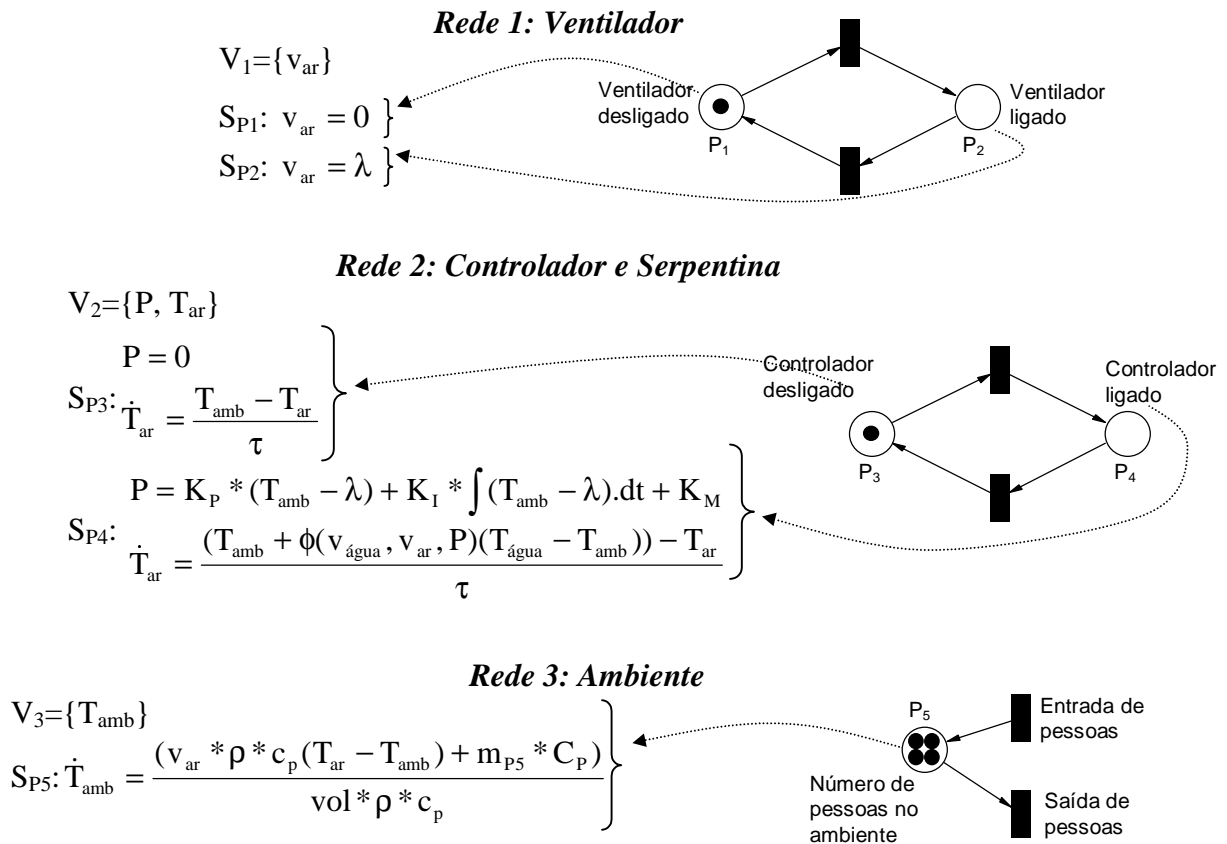


Figura 10 – Modelo em rede PTD.

4. ANÁLISE DAS TÉCNICAS DERIVADAS DE REDES DE PETRI

Para os três casos apresentados, observa-se que o modelo do sistema de ar condicionado é construído a partir da modelagem matemática da relação entre as diversas variáveis. Entretanto, no caso da Rede de Petri Híbrida, a representação da equação na forma de um par transição-lugar restringe o tipo de equação a ser representado para equações diferenciais de primeira ordem, e acrescenta pouca ou nenhuma informação ao modelo. Em contrapartida a rede PTD e a HyNet podem ser utilizadas para modelagem de sistemas mais complexos, uma vez que não existe a necessidade de transformar equações em elementos da rede, buscando-se apenas associar as equações aos elementos.

Quanto a estrutura da rede resultante, para a rede de Petri Híbrida e para a rede PTD a ligação entre as diversas sub-redes é realizada através das variáveis, por exemplo, a variável v_{ar} determinada no modelo do ventilador (Figura 5) é utilizada no modelo do ambiente (Figura

8). Esta característica faz com que esta relação entre os modelos não esteja explícita na estrutura da rede. Na HyNet esta relação fica explícita através dos arcos contínuos, como por exemplo os arcos “a”, “m”, “n” e “o” da Figura 9.

Quanto à construção do modelo híbrido, a rede PTD é aquela que apresenta maior facilidade de construção a partir da modelagem matemática. No caso da Hynet, a sua utilização, simulação e interpretação gráfica não é trivial, uma vez que o estado do sistema é representado por “atributos” das *marcas* e não está explícito no estado da rede.

A Rede de Petri Híbrida apresenta vantagens principalmente para a modelagem de sistemas onde deseja-se representar um fluxo contínuo de material, modelados através de transições contínuas, que é armazenado em pontos de sistema (ex.: tanques), modelados através de lugares contínuos. Neste caso a construção da rede híbrida é realizada principalmente a partir da observação do sistema e o modelo obtido tem um forte relacionamento físico com a estrutura do sistema, conservando a característica das Redes de Petri de ser uma descrição gráfica efetiva para análise e simulação. No entanto, de acordo com o apresentado anteriormente, este não é o caso do sistema de ar condicionado.

Quanto a utilização do conceito de orientação a objetos, observa-se que este auxilia na especificação de sistemas complexos. Em especial, para o caso do ar condicionado a adição de uma nova zona, de um novo “chiller”, etc. é simplificada pelo “instanciamento” de objetos.

5. CONSIDERAÇÕES FINAIS

Este trabalho abordou o problema de modelagem do sistema de ar condicionado considerando-o como um sistema híbrido. Esta abordagem é baseada na necessidade de representação de eventos e estados discretos e de variáveis contínuas, de forma a possibilitar a integração do sistema de ar condicionado com outros sistemas do edifício.

São apresentadas e comparadas diferentes técnicas para representação de sistemas híbridos, aplicando-as para modelagem de um exemplo de sistema de ar condicionado. De uma forma genérica observa-se que as ferramentas definidas a partir da especificação de uma interface entre a ferramenta de modelagem discreta (ex. Redes de Petri) e a ferramenta de modelagem contínua (ex. equações diferenciais) apresentam maior flexibilidade para utilização. Comparando as três redes apresentadas para o caso específico de modelagem de sistemas de ar condicionado, considerou-se a rede PTD como a mais adequada, devido a sua maior flexibilidade e maior facilidade de construção do modelo.

Os modelos aqui apresentados são uma simplificação dos modelos desenvolvidos para o Prédio dos Ambulatórios (PAMB) do Hospital das Clínicas, da Faculdade de Medicina da Universidade de São Paulo (HC – FMUSP). O PAMB inclui ambulatórios, centro cirúrgico, farmácia industrial, entre outras instalações, em cerca de 10.000m² de área construída. O seu sistema de ar condicionado apresenta possibilidade de aquecimento e resfriamento. As diversas zonas nas quais o PAMB é dividido apresentam diferentes características quanto ao tipo de carga térmica e as exigências de condicionamento.

Atualmente, seu sistema de controle de ar condicionado atua de forma pouco eficaz, devido à ausência de um sistema de controle que interaja com os demais sistemas do edifício. A proposta de uma solução para este problema baseia-se fundamentalmente em desenvolver módulos que possibilitem a interação dos diversos sistemas entre si.

Para a integração do sistema, estão sendo desenvolvidos modelos em redes de Petri dos diversos sistemas do edifício. Estes modelos são então integrados entre si através do BMS. Em particular, para o ar condicionado é desenvolvido um sistema de gerenciamento discreto que é responsável por chavear estratégias de controle e modificar o estado de equipamentos em função, entre outras coisas, de eventos ocorridos em outros sistemas. O modelo deste

sistema é desenvolvido em redes de Petri e interage diretamente com os modelos híbridos do sistema de ar condicionado através de arcos habilitadores e inibidores.

6. AGRADECIMENTOS

Os autores agradecem o Grupo de Administração dos Sistemas de Saúde da FSP-USP e a Superintendência do HC-FMUSP, que auxiliam no desenvolvimento desta pesquisa. Agradecemos também o apoio das seguintes entidades: FAPESP, CYTED, CNPq e CAPES.

7. REFERÊNCIAS

- Alla, H. & David, R., 1998, "Continuous and Hybrid Petri Nets" *Journal of Circuits, Systems and Computers*, vol.8, n.1
- Antsaklis, P. J. & Nerode, A., 1998, "Hybrid Control Systems" *IEEE Transactions on Automatic Control*, vol 43, n.4, pp 457-459.
- Arkin, H. & Paciuk, M., 1995, "Service Systems Integration in Intelligent Buildings "; In: *Proceedings of IB/IC Intelligent Buildings Congress*, pp 19-30, Telaviv, Israel.
- Becker, R., 1995, "What is an "Intelligent Building"; In: *Proceedings of IB/IC Intelligent Buildings Congress*", pp 320, Telaviv, Israel.
- Champagnat, R., 1998, "Supervision des Systèmes Discontinus: Definition d'un Modèle Hybride et Pilotage en Temps-réel" PhD Thesis, Univ. Paul Sabatier, Toulouse, France.
- Demongodin, I. & Koussoulas, N. T., 1998, " Differential Petri Nets: Representing Continuous Systems in a Discrete-Event World". *IEEE Transactions on Automatic Control*, vol. 43, n.4, pp 573-579.
- Drath, R., 1998, "Hybrid Object Nets: An Object Oriented Concept for Modelling Complex Hybrid Systems" *ADMP'98 3rd International Conference on Automation of Mixed Processes*, Reims, France.
- Fanger, P. O., 1970, "Thermal Confort" McGraw-Hill, New York, USA.
- Flax, B. M., 1991, "Intelligent Buildings"; In: *IEEE Communications Magazine*, pp 24-26, vol. 29 n. 4.
- Fujie, S. & Mikami, Y., 1991, "Construction Aspects of Intelligent Buildings"; In: *IEEE Communications Magazine*, pp 50-57, vol. 29 n. 4.
- Ho, Y. C., 1987, "Basic research, manufacturing automation, and putting the cart before the horse". *IEEE Transactions on Automatic Control*, vol.AC-32, n.12, pp.1042-1043.
- Incropera, F. P., 1990, "Fundamentals of heat and mass transfer", Wiley, New York.
- Jabardo, J. M. S., 1984, "Conforto Térmico das Habitações", Eternit/IPT, São Paulo, SP.
- Lemmon, M. D., He, K. X. & Markovsky, I., 1999, "Supervisory Hybrid Systems", *IEEE Control System*, vol.8 n.1, pp 42-55.
- Meyer, B., 1997, "Object-oriented software construction", Prentice Hall, New York, USA.
- Miyagi, P.E., 1996, "Controle Programável - fundamentos do controle de sistemas a eventos discretos" Editora Edgard Blücher Ltda, S. Paulo, SP.
- Salsbury, T. I., 1996, "Fault Detection and Diagnosis in HVAC Systems using Analytical Models", PhD Thesis, Dep. of Civil and Building Engineering, Loughborough University, Loughbrough, England.
- Wieting, R., 1996, "Hybrid High-Level Nets" *Proceedings of the 1996 Winter Simulation Conference*, Coronado.

DETERMINAÇÃO DOS PARÂMETROS E SEQÜÊNCIA DE CORTE PARA A MÁXIMA PRODUÇÃO EM TORNOS CNC.

Patrick Leloup

Noemia Gomes de Mattos de Mesquita

Hugo Marcelo Bezerra de Carvalho

Sérgio Murilo Veríssimo de Andrade

Universidade Federal de Pernambuco, Centro de Tecnologia e de Geociências, Departamento de Engenharia Mecânica, 50740-530, Recife, PE, Brasil. E-mail: ngmm@npd.ufpe.br

Resumo

Neste trabalho foi desenvolvido um método para a determinação dos parâmetros otimizados, na a máxima produção, utilizando a fórmula expandida de Taylor, assim como, para a obtenção dos tempos de usinagem, tanto no torneamento axial como no radial, em peças cilíndricas e cônicas. Dando-se assim uma contribuição para a automatização do programa em linguagem CN com parâmetros de corte otimizados.

Palavras-chave: Máxima produção, Seqüência de corte, Torno CNC.

1. INTRODUÇÃO

O planejamento da usinagem é uma das áreas mais deficitárias na fabricação, pois este setor não tem acompanhado o ritmo de modernização e desenvolvimento dos demais segmentos que compõem o sistema de produção.

O ponto primordial para a programação de uma máquina operatriz com comando numérico (CN) é a determinação dos parâmetros otimizados de usinagem (velocidade de corte, avanço e profundidade) assim como a melhor seqüência de usinagem (Prasad et al, 1997; Kayacan et al, 1996; Younis et al, 1997).

Neste trabalho está sendo desenvolvido um método para a obtenção dos parâmetros e da seqüência de corte para a máxima produção a fim de que possa ser integrado no desenvolvimento de um sistema CAM.

2. OBJETIVOS E METODOLOGIA UTILIZADA

Este trabalho está sendo desenvolvido em parceria com o SENAI/PE, que adquiriu recentemente um torno CNC modelo CENTUR-30 da empresa ROMI. Inicialmente, a determinação dos parâmetros de corte para a máxima produção, assim como as equações para a determinação do tempo de usinagem de uma peça, deverão ser obtidas para atender às necessidades específicas desta máquina, podendo posteriormente ser estendidas à qualquer máquina CNC (N.N., 1999).

O objetivo deste trabalho é determinar o tempo de usinagem para diferentes seqüências de corte, torneamentos axial e radial, de forma a se estabelecer a seqüência mais rápida.

Para o torno especificado temos como dados essenciais a potência máxima de 10 CV com um rendimento de 0.8, rotação máxima de 3500 rpm, velocidade de avanço e retorno

máximas de 10000 e 7500 mm/s respectivamente. As velocidades de rotação não assumem valores decimais, a sua variação é de uma rotação por minuto, só assumindo valores inteiros.

O trabalho está sendo desenvolvido com o auxílio do programa Mathcad, mas pretende-se desenvolver um software em linguagem C++ (Pohl, 1991).

Os resultados obtidos a partir do programa deverão ser comparados com os tempos reais de usinagem medidos na máquina CNC. Será possível assim, efetuar certas correções no equacionamento pois há sempre perdas de potência, através de, por exemplo, atrito, deflexões ou desgastes, aumentando o tempo real de usinagem.

3. DETERMINAÇÃO DOS PARÂMETROS DE CORTE PARA A MÁXIMA PRODUÇÃO

Na determinação dos parâmetros de corte, para a condição de máxima produção, enfrenta-se o seguinte problema: a velocidade de máxima produção é função dos parâmetros de Taylor x e K (veja equação 01) que variam com o avanço e a profundidade de corte.

$$V_c = \left[\frac{K}{(x-1) \times T_t} \right]^{\frac{1}{x}} \quad (01)$$

onde V_c é a velocidade de corte, x e K são constantes que dependem do avanço e da profundidade de corte e T_t é o tempo de troca da ferramenta, estimado em 2 minutos.

Por outro lado a profundidade de corte, calculada através da equação de Kienzle depende do valor da velocidade de corte, como mostra a equação (02)

$$p_{max} = \frac{N_c \times 60 \times 75 \times \eta \times \sin(\chi)}{K_s \times V_c \times (f \times \sin(\chi))^{1-z}} \quad (02)$$

onde N_c é a potência da máquina, η é o seu rendimento, χ é o ângulo de posição da ferramenta, K_s é a resistência específica de corte f o avanço e V_c a velocidade de corte.

Para solucionar este problema procedeu-se da seguinte maneira: as constantes K e x podem ser identificadas comparando-se a fórmula expandida de Taylor (04) com a equação (03)

$$V_c = K^{\frac{1}{x}} \times T^{\frac{-1}{x}} \quad (03)$$

$$V_c = C \times f^E \times p^F \times V_b^H \times T^G \quad (04)$$

obtendo-se assim (Mesquita, 1980):

$$x = \frac{-1}{G} \quad (05)$$

$$K = [C \times f^E \times p^F \times V_b^H]^x \quad (06)$$

onde C, E, F, H, G são constantes da fórmula expandida de Taylor e V_b é o desgaste de flanco que para operações de desbaste pode variar até 1.2 mm e para operações de acabamento é calculado em função da tolerância desejada.

Para um dado avanço, faz-se variar a profundidade de corte de 0.001 até a profundidade máxima permitida pela ferramenta. Calcula-se a velocidade de corte para cada profundidade, calculando em seguida a profundidade de corte máxima. Compara-se estas duas profundidades de corte, deve-se escolher a profundidade que seja igual à profundidade máxima. Faz-se variar o avanço e calcula-se a profundidade e a velocidade de corte. Calcula-se então o tempo de usinagem para esses parâmetros, escolhendo-se os que derem o menor resultado.

A tabela 1 apresenta os valores parciais das profundidades de corte p_i , das profundidades de corte máximas p_{max_i} e das respectivas velocidades de corte V_{c_i} , obtidas através do Mathcad, para um avanço de mm/volta, um ângulo de posição da ferramenta de 95° , desgaste de flanco igual a 0.8 mm e constantes de Taylor determinadas para um aço ABNT 1038 dadas por: $C=910.64$, $E=-0.3297$, $F=-0.0366$, $H=0.8936$, $x=2.6261$ e para $1/h=0.85$, $K_s=180.612$ N/mm^2 .

Tabela 1. Valores de p_i , p_{max_i} e V_{c_i}

p_i (mm)	p_{max_i} (mm)	V_{c_i} (m/s)
⋮	⋮	⋮
0.337	0.339	533.281
0.338	0.339	533.223
0.339	0.339	533.165
0.340	0.339	533.108
0.341	0.339	533.05
⋮	⋮	⋮

4. CÁLCULO DOS TEMPOS DE USINAGEM

4.1 Usinagem Cilíndrica

Para a usinagem de uma peça como a mostrada abaixo, devemos identificar os três pontos (Z_1, X_1) , (Z_2, X_2) , (Z_3, X_3) , onde X é dado em diâmetro.

O número de passes (ndp) é dado pela parte inteira da divisão da diferença dos diâmetros pela profundidade de corte (p).

$$ndp = \text{floor} \left(\frac{X_3 - X_1}{2 \times p} \right) \quad (07)$$

O material que deixará de ser usinado será retirado com um passe pela mesma ferramenta de desbaste para um pré-acabamento. A sua espessura pode ser calculada por

$$e = \frac{X_3 - X_1}{2} - \left[\text{floor} \left(\frac{X_3 - X_1}{2 \times p} \right) \right] \times p \quad (08)$$

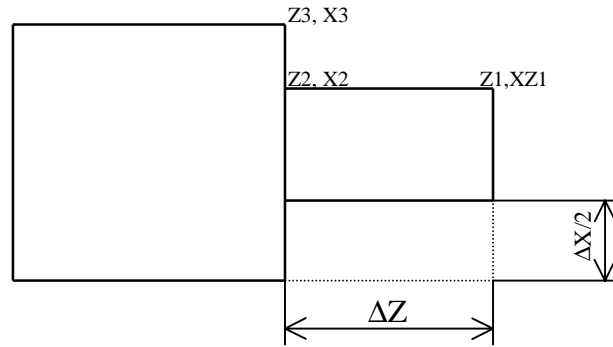


Figura 1. Peça a ser torneada

Para i variando de 1 à ndp , a velocidade de rotação (n) da peça em rotações por minuto será dada por

$$n_i = \text{floor} \left(\frac{1000 \times Vc}{\pi \times (180 - i \times 2 \times p)} \right) \quad (09)$$

A velocidade de avanço da ferramenta (Va) será dada por

$$Va_i = n_i \times f \quad (10)$$

Deve-se verificar se a rotação máxima de 3500 rpm não é ultrapassada. Caso $n_i > 3500$ então adotar como rotação $n_i = 3500$

O tempo de um passe será:

$$T_i = \frac{Z_2 - Z_1 + 2}{Va_i} + \frac{2}{7500} + \frac{Z_2 - Z_1 + 2 - 2}{10000} + \frac{p + 2}{7500} \quad (11)$$

Onde o primeiro termo é o tempo de corte com uma folga de 2 milímetros da peça, o segundo é o tempo de afastamento da ferramenta com uma inclinação de 45° até se distanciar de 2 milímetros da peça, o terceiro é o tempo de recuo da ferramenta e o quarto é o tempo de posicionamento da peça para o próximo corte. Sendo então o tempo total dado por:

$$\text{Tempo} = \sum_{i=1}^{ndp} T_i - \frac{2 + p}{7500} \quad (12)$$

Para a usinagem de uma peça cônica devemos identificar também três pontos, (Z_1, X_1) , (Z_2, X_2) , (Z_3, X_3) , como mostra a figura 2.

A velocidade de avanço será dada pela mesma expressão. A diferença será a distância a ser usinada assim como a distância do recuo da ferramenta que mudarão a cada passe. Tem-se que a distância a ser usinada para a parte cônica a cada passe é dada por

$$dist_i = \frac{Z_3 - Z_2}{X_3 - X_2} \times \left[\left(\frac{X_3 - X_2}{2} \right) - i \times p \right] \quad (13)$$

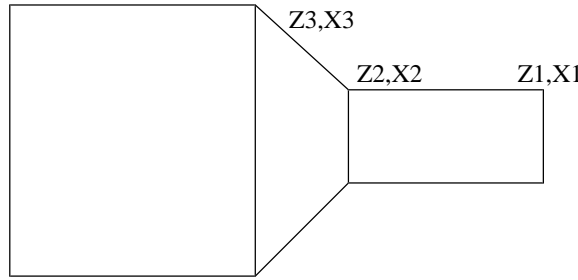


Figura 2. Peça cilíndrica

A velocidade de avanço será dada pela mesma expressão. A diferença será a distância a ser usinada assim como a distância do recuo da ferramenta que mudarão a cada passe. Tem-se que a distância a ser usinada para a parte cônica a cada passe é dada por

$$dist_i = \frac{Z_3 - Z_2}{X_3 - X_2} \times \left[\left(\frac{X_3 - X_2}{2} \right) - i \times p \right] \quad (13)$$

O tempo de um passe será dado por

$$T_i = \frac{dist_i + Z_2 - Z_1 + 2}{Va_i} + \frac{2}{7500} + \frac{dist_i + Z_2 - Z_1 + 2 - 2}{10000} + \frac{2 + p}{7500} \quad (14)$$

O tempo total é dado então pela equação (12).

4.2-Usinagem Radial

Para o desbaste radial, deve-se ter a usinagem dividida em pequenos intervalos nos quais a rotação é constante, já que as rotações só assumem números inteiros.

Tomando-se como exemplo a peça da figura 1, tem-se que a rotação inicial é dada por

$$n_{inic} = floor \left(\frac{1000 \times Vc}{\pi \times (X_3 + 4)} \right) \quad (15)$$

O número de intervalos (N) nos quais a rotação é constante é dado por

$$N = ceil \left(\frac{1000 \times Vc}{\pi \times X_2} \right) - floor \left(\frac{1000 \times Vc}{\pi \times (X_3 + 4)} \right) \quad (16)$$

Deve-se aqui levar em conta a rotação máxima da máquina que é de 3.500 rpm. Logo devemos inicialmente calcular o diâmetro para o qual a rotação máxima é atingida e comparar com X₂.

$$D_{max} = \frac{1000 \times Vc}{\pi \times 3500} \quad (17)$$

Se D_{max} estiver compreendido entre (X_3+2) e X_2 , no cálculo do número de intervalos devemos substituir X_2 por D_{max} e haverá um outro intervalo entre D_{max} e X_2 com rotação constante de 3.500 rpm. Se (X_3+2) for menor que D_{max} a usinagem será toda com rotação constante de 3.500 rpm. Se X_2 for maior que D_{max} vale a equação dada.

Deve-se ainda determinar os diâmetros que limitam cada intervalo

$$d_1 = X_3 + 4 \quad (18)$$

$$d_{N+1} = X_2 \quad (19)$$

$$d_i = \frac{1000 \times Vc}{\pi \times (n_{mic} + i - 1)} \quad (20)$$

Para i variando de 2 à N.

O número de passes é dado por

$$ndp = floor\left(\frac{Z_1 - Z_2}{P}\right) \quad (21)$$

O tempo de um passe é:

$$T_i = \sum_{j=1}^n \left(\frac{d_j - d_{j+1}}{2} \times \frac{1}{(n_{mic} + i - 1) \times f} \right) + \frac{2}{7500} + \left(\frac{d_1 - d_{N+1}}{2} - 2 \right) \times \frac{1}{7500} + \frac{2 + P}{10000} \quad (22)$$

O tempo total de usinagem é:

$$Tempo = \sum_{i=1}^{ndp} T_i - \frac{2 + p}{10000} \quad (23)$$

Para o caso de uma peça cônica, como a mostrada na figura 2, a parte cônica é a única que nos interessa pois a outra parte é resolvida como mostrado acima.

Para este caso a distância a ser usinada variará a cada passe.

$$dist_i = \frac{X_3 - X_2}{2} - \left(\frac{X_3 - X_2}{2} \right) \times i \times \frac{p}{Z_3 - Z_2} \quad (24)$$

O número de intervalos nos quais a rotação é constante dependerá também do passe que estamos efetuando.

$$N_i = ceil\left(\frac{1000 \times Vc}{\pi \times (d_1 - 2 \times dist_i)}\right) - floor\left(\frac{1000 \times Vc}{\pi \times d_1}\right) \quad (25)$$

O tempo de um passe é dado por:

$$T_i = \sum \left(\frac{d_j - d_{j+1}}{2} \times \frac{1}{(n_{mic} + i - 1) \times f} \right) + \frac{2}{7500} + \left(\frac{d_1 - d_{N_i+1}}{2} - 2 \right) \times \frac{1}{7500} + \frac{2+p}{10000} \quad (26)$$

O tempo total é dado pela equação (23).

5. COMPARAÇÃO ENTRE USINAGEM AXIAL E RADIAL

É de grande interesse encontrar uma relação entre as dimensões da peça a ser usinada que pudesse determinar qual o modo de usinagem mais rápido, o axial ou o radial.

Quanto menor for a relação $\Delta X/\Delta Z$ mais propícia a usinagem axial. Por exemplo: deseje-se desbastar uma peça a fim de obter o perfil da figura 1, dado pelos pontos $(Z_1, X_1)=(100,80)$, $(Z_2, X_2)=(60,80)$, $(Z_3, X_3)=(60,120)$.

Tem-se $\Delta X/\Delta Z=1$. Testando para $p=1.5$ mm, $f=0.4$ mm/rot e $V_c=400$ m/s encontramos 1.188 min para o desbaste axial e 1.319 min para o desbaste radial.

Fazendo-se agora $(Z_3, X_3)=(60,140)$, $\Delta X/\Delta Z=1.5$, encontra-se os seguintes resultados: 1.984 min para o desbaste axial e 1.829 min para o desbaste radial. Sendo o desbaste radial mais vantajoso.

Para se estabelecer uma relação $\Delta X/\Delta Z$ para a qual seria mais interessante a usinagem radial ou axial deve-se levar em consideração a que distância do eixo da peça está se usinando. Quanto mais excêntrico for o torneamento, menor será a velocidade de rotação e mais propício ao torneamento axial será, como mostra o seguinte exemplo: seja a peça $(Z_1, X_1)=(100,120)$, $(Z_2, X_2)=(60,120)$, $(Z_3, X_3)=(60,180)$, os tempos encontrados são: 2.644 min para o torneamento axial e 2.715 para o radial. Portanto comparando-se com o exemplo anterior, onde tem-se a mesma relação $\Delta X/\Delta Z=1.5$, neste exemplo o torneamento axial passa a ser melhor, só pelo fato de estar mais distante do eixo de rotação da peça.

Observa-se também que o avanço escolhido também influencia. Quanto menor o avanço, mais rápido o torneamento axial em relação ao radial. Tomando-se o exemplo $(Z_3, X_3)=(60,140)$ agora com um avanço de 0.2 mm/rot, obtêm-se: 1.354 min para o torneamento axial e 1.493 min para o radial. Sendo agora o desbaste axial mais vantajoso.

6-CONCLUSÃO

No estágio atual do trabalho, ainda não se encontrou uma relação exata entre as dimensões da peça para se determinar a melhor direção de torneamento. No entanto, já se pode observar que quanto menor a relação $\Delta X/\Delta Z$, mais propício ao torneamento axial, porém, quanto maior o avanço e quanto mais próximo o material a ser retirado estiver do centro de rotação, mais propício ao torneamento radial. As equações encontradas neste trabalho podem ser associadas para calcular os tempos de usinagem das diferentes seqüências possíveis para o desbaste de uma peça, utilizando os parâmetros de máxima produção, podendo ser facilmente aproveitado em um programador de máquinas CN.

Certos ajustes, porém, devem ser feitos nas equações pois não foi considerada nenhuma perda, como, por exemplo, atrito. Comparações com medidas experimentais mostraram um erro de 6.9% para o desbaste radial e 4% para o axial. Esta disparidade relativa às medições experimentais, todavia, não interferem em nada nas conclusões tiradas neste trabalho.

Estas diferenças podem ser devido a um pequeno intervalo de tempo na mudança de direção da ferramenta, a uma aceleração da ferramenta até atingir sua velocidade máxima ou

então a velocidade máxima da ferramenta não é atingida. Novas medições serão efetuadas para adequar as equações da melhor maneira possível aos resultados práticos.

Na programação de tornos CN, em operações de desbaste tem-se a opção de se retirar o sobremetal movimentando-se a ferramenta de corte no sentido axial ou radial. Este trabalho contribui para se escolher qual das duas opções é a melhor para diferentes tipos de peças, diminuindo-se desta forma o tempo necessário para a programação.

7 REFERÊNCIAS

- Ferraresi, D., 1977, “Fundamentos da Usinagem dos Metais”, São Paulo, Brasil.
- Kayacan M.C., Filiz I.H., Sonnez A.I., Baykasoglu A. and Dereli T., 1996, “OPPSROT: An Optimised Process Planning System For Rotational Parts”, Computers in Industry, vol. 32, pp.181 – 195.
- Mesquita, N.G.M., 1980, “Determinação dos Parâmetros da Fórmula de Taylor Expandida”, Tese de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, S.C., Brasil.
- N.N., 1999, “Manual de Programação e Operação CNC Mach 9 R 9001D”, São Paulo.
- Pohl, I., 1991, “C++ Para Programadores de PASCAL”, Ed. Berkeley, Rio de Janeiro, Brasil.
- Prasad, A.V.S.R.K., Rao P.N. and Rao U.R.K., 1997, “Computer Optimization of Multi-Pass Turning”, International Journal of Production Research, vol. 9, nº 2, Taylor & Francis LTD, London, England, pp. 247 – 259.
- Younis M.A. and Wahab M.A.A., 1997, “A CAPP ExpertSystem For Rotational Components”, Computers & Industrial Engineering, vol. 33, pp. 505 – 512.

ANÁLISE DA ESTABILIDADE DE AGARRAMENTO EM MANIPULADORES ROBÓTICOS

Max Suell Dutra

Marco Hiroshi Naka

Universidade Federal do Rio de Janeiro, COPPE-Programa de Engenharia Mecânica

C. P. 68503, CEP 21945-970, Rio de Janeiro, Brasil - E-mail: {max, mnaka}@serv.com.ufrj.br

Resumo

As tarefas de um manipulador robótico se resume em agarrar objetos e alterar a sua orientação ou posição através de um sistema articulado. No ato do agarramento, algumas considerações devem ser feitas, sendo que uma das principais é a questão da segurança do agarramento. Entende-se como segurança no agarramento, a não ocorrência do escorregamento do objeto e a não destruição do mesmo através de forças excessivas. Na primeira abordagem, relacionada com o objeto, é considerada a questão da estabilidade no agarramento, e as variáveis que a regem. Na segunda abordagem é observada a questão da estabilidade do contato, as forças resultantes e o comportamento do manipulador quando submetido à ação de uma força perturbadora durante o agarramento. Neste trabalho, busca-se apresentar a análise física e matemática do agarramento, destacando a importância da determinação da força resultante nos mecanismos múltiplos e da força interna entre os mecanismos.

Palavras-chave: manipuladores, estabilidade, cone de atrito, agarramento e elipsóide de contato

1. INTRODUÇÃO

O ser humano pode de olhos vendados, perceber e diferenciar uma laranja de um ovo e manipulá-los de forma adequada, sem danificá-los. Já nos sistemas robóticos normais, tem-se esse tipo de problema, pois tais sistemas não possuem capacidade de diferenciar os objetos manipulados e nem de aplicar uma força adequada sem provocar danos (Dario & De Rossi, 1985). O sistema de visão com reconhecimento dos objetos ainda não é confiável sem uma grande quantidade de informação. Com a finalidade de reconhecer objetos, são utilizados sensores tácteis nestes sistemas. Para que este sistema de reconhecimento funcione é imprescindível que haja um controle de forças associados a essa rede de sensores, de tal forma que as forças exercidas sejam adequadas e assegurem um agarramento estável. Para que este controle seja adequado é necessário avaliar física e matematicamente o aperto executado por cada garra mecânica a ser projetada. Segundo Jameson et al. (1986) a estabilidade do aperto tem pelo menos dois aspectos relevantes a serem observados: o primeiro é a capacidade de retorno para a posição de equilíbrio estático quando a posição do objeto é modificada. Isto pode ser chamado de estabilidade de aperto e está associada a uma restauração das forças aplicadas (Hanafusa & Asada, 1977) (Cherif & Gupta, 1999). O segundo aspecto importante é a capacidade de manter o contato sobre o objeto, quando o mesmo é submetido a ação de forças perturbadoras, isto pode ser chamado de estabilidade de contato. Esta concepção pode ser associada a resultante das forças

externas (equação 2), a questão das forças internas e ao atrito entre a garra mecânica e o objeto manipulado (Nakamura et al., 1989). E é sob essa concepção que este trabalho será desenvolvido.

Voltando ao primeiro aspecto, relativo a estabilidade do aperto, é observado que quando um objeto, sem restrições à sua movimentação, é agarrado e os dedos não definem uma posição de equilíbrio estático adequado, o dedo se movimentará até atingir a posição de equilíbrio estático, conforme esquematizado na figura 1 e 2.

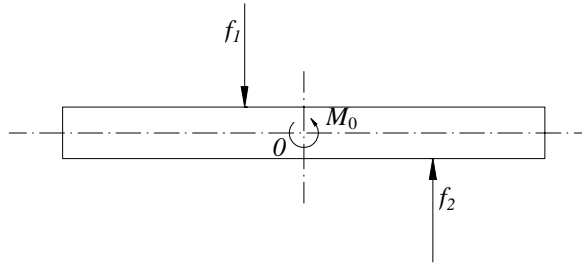


Figura 1. Ausência de Estabilidade Estática do Objeto

A resultante das forças externas F_0 e o momento M_0 , das forças externas aplicadas sobre o objeto pode ser dado por:

$$F_0 = \sum_{i=1}^n f_i + m_0 g \quad (1)$$

$$M_0 = \sum r_i \times f_i \quad (2)$$

onde, f_i é força externa aplicada no i -ésimo contato, m_0 é a massa do objeto e g , a aceleração da gravidade, r_i é a distância do centróide O até a força aplicada no i -ésimo contato.

Para que o objeto encontre uma posição de equilíbrio estático, é necessário que a equação (2) seja nula, o que não ocorre no exemplo da figura 1, se for considerado que todas as forças de contato exercidas sobre o objeto são iguais.

Observa-se também na figura 1 que, quando é analisada a estabilidade do aperto, é necessário que além da condição de livre movimentação do objeto seja também permitido o livre deslizamento dos dedos até atingir o ponto de equilíbrio estático (figura 2).

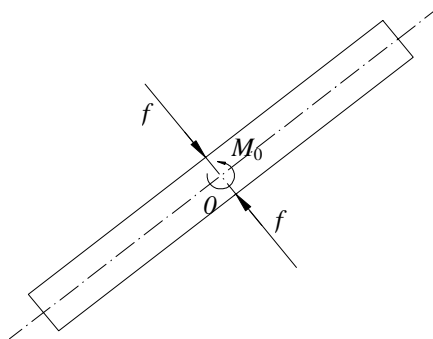


Figura 2. Objeto com Estabilidade Estática

O fator relevante dessa análise está na concepção do projeto da rigidez do dedo, de tal maneira que a rigidez total do sistema do dedo possa permitir a geração de forças de restauração para qualquer tipo de perturbação de posição do objeto.

Neste trabalho serão apresentados também, uma análise do mecanismo de agarramento sob uma visão matemática e a questão da influência do atrito como fator fundamental na estabilidade do agarramento.

2. ESTABILIDADE DO CONTATO

Visando simplificar o equacionamento matemático, serão feitas duas suposições:

- O contato dos dedos com o objeto é realizado através de pontos e não de superfícies;
- Os pontos de contato não se movimentam quando há uma mudança de orientação entre o objeto e o manipulador.

A primeira suposição é razoável considerando que as forças são pequenas, e tem como consequência que o objeto e os dedos em questão são rígidos (Salisbury, 1982). A segunda hipótese pode aparentemente não trazer nenhuma simplificação em nosso trabalho, mas serve para eliminar o problema das restrições não-holonômicas entre o objeto e a garra, devido a alteração dos pontos de contato com o movimento do objeto (Murray et al., 1993).

O modelo de contato adotado é o do ponto de contato com atrito que pode ser observado na figura 3, bem como a interpretação geométrica do modelo de atrito de Coulomb. Esse modelo indica a magnitude da força de contato que pode ser aplicada nas direções tangenciais da superfície como uma função da força normal (f_n) aplicada. O modelo de atrito de Coulomb é um modelo empírico o qual indica que a força tangencial (f_t) admissível é proporcional a força normal aplicada, e a constante de proporcionalidade é função dos materiais em contato.

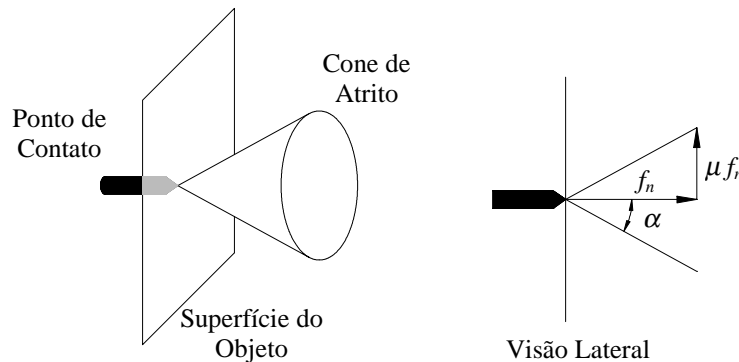


Figura 3. Ponto de contato e interpretação geométrica do modelo de atrito de Coulomb.

A condição de Coulomb para que não haja deslizamento é:

$$|f_t| \leq \mu f_n \quad (3)$$

onde $\mu > 0$, é o coeficiente de atrito estático.

O conjunto de forças que pode ser aplicado deve se manter no interior do cone sobre a superfície normal (Dutra & Naka, 1999). Este cone é chamado cone de atrito, e o ângulo do cone com relação a normal é dado por:

$$\alpha = \tan^{-1} \mu \quad (4)$$

Os valores típicos de μ são menores do que 1, e conseqüentemente, o ângulo do cone de atrito é geralmente menor que 45° .

O modelo de ponto de contato com atrito é usado quando existe atrito entre as pontas de dedo e o objeto, onde as forças exercidas em qualquer direção devem estar dentro do cone de atrito.

Uma força generalizada atuando sobre um corpo rígido consiste de uma componente linear (força) e uma componente angular (momento) atuando em um ponto. Esse tensor força/momento é conhecido na literatura como *wrench* (Murray et al., 1993):

$$W = \begin{bmatrix} F \\ M \end{bmatrix} \quad \begin{array}{l} F \in R^3 \text{ componente linear} \\ M \in R^3 \text{ componente angular} \end{array} \quad (5)$$

Para o caso de um ponto de contato com atrito, tem-se a seguinte matriz base *wrench*:

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6)$$

Logo a resultante das forças externas Q pode ser definida como:

$$Q = Wf_i \quad (7)$$

Sendo que f_i deve estar contida no cone de atrito, logo:

$$f_i \in FC_i = \left\{ f \in R^3 : \sqrt{f_1^2 + f_2^2} \leq \mu f_3, f_3 \geq 0 \right\} \quad (8)$$

onde f_1 e f_2 são as componentes da força resultante no plano e f_3 equivale a força normal.

Assim sendo, a determinação da força resultante é feita de tal forma que é considerado um problema estático e não dinâmico com a finalidade de simplificar o problema.

3. FORÇA INTERNA

A força interna no agarramento e na manipulação por garras robóticas têm sido discutida com base na equação (7) e especificado pelo elemento W (*wrench*). Nesta seção é explanado, de uma forma mais clara, o significado físico das forças internas e a sua definição alternativa.

Supondo que são aplicadas as forças f_i ($i = 1, 2, \dots, n$) no objeto, e considerando que um deslocamento virtual do objeto pode ser representado por:

$$\delta\phi = (\delta r^T \quad \delta\Omega^T)^T \quad (9)$$

onde, $\delta\Omega$ significa a orientação do deslocamento virtual e δr é a distância do centro de massa ao mecanismo robótico.

A somatória total dos trabalhos virtuais realizados por todos os mecanismos robóticos é:

$$\delta\omega = \sum_{i=1}^n (\delta\Omega \times p_i + \delta r)^T f_i \quad (10)$$

onde, p_i é a posição do i -ésimo contato

Simplificando, têm-se:

$$\delta\omega = \delta\phi^T WF_0 \quad (11)$$

A equação seguinte é uma condição necessária e suficiente para que $\delta\omega$ seja igual a zero para qualquer $\delta\phi$:

$$WF_0 = 0 \quad (12)$$

Desta forma, as forças internas são um conjunto de f_i , de tal forma que a somatória dos trabalhos virtuais executados pelos dedos são zero. A definição clássica (Murray et al., 1993) é de que as forças internas são um grupo de forças representadas com relação a uma base de *wrench* para cada contato. A existência da força interna é uma condição necessária para o agarramento.

Kerr e Roth (1986) propuseram que a força interna ótima está relacionada com os máximos valores de f_i e os máximos torques. A primeira vista, de fato, forças de agarramento maiores proporcionariam agarramentos mais seguros, assegurando uma boa estabilidade no contato (Cutkosky & Wright, 1986). Entretanto, caso haja um erro no posicionamento dos dedos, por menor que seja, devido a grande intensidade de forças de agarramentos, ter-se-ia grandes distúrbios de força e momento com relação ao centro de massa do objeto. Surge então, a importância de determinar uma força interna que seja mínima, mas suficiente para gerar uma razoável força resultante externa, que por sua vez assegurará um bom agarramento. A minimização das forças internas tem sido realizada por diversos pesquisadores utilizando algoritmos computacionais, através de programações não-lineares (Nakamura et al, 1989).

4. AÇÃO DE UMA FORÇA PERTURBADORA E O ELIPSÓIDE DE CONTATO

O grande problema relacionado com a estabilidade de contato é com o comportamento do mecanismo robótico quando o objeto agarrado é submetido a ação de uma força perturbadora, conforme visto na figura 4. A ação dessa aceleração perturbadora sobre o objeto altera o seu equilíbrio e torna-se necessário a restauração desse equilíbrio com a adição de novas forças.

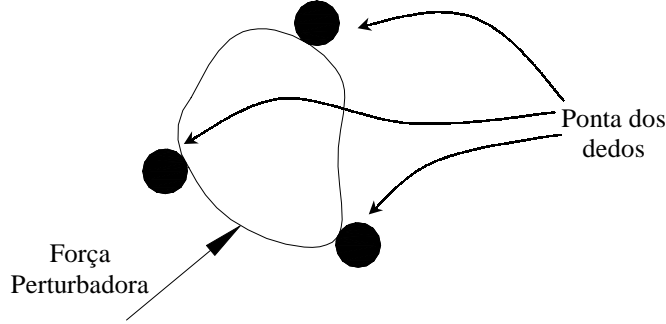


Figura 4. Ação de uma força perturbadora sob um agarramento estável

Desde que o equilíbrio dinâmico é mantido pela aplicação de um novo momento M_{ip} , este problema é reduzido para um problema estático, onde pode se supor que os dedos permaneçam imóveis no momento que são submetidos a ação da força perturbadora ($\dot{\theta}_i = 0$), e não há influência da ação da gravidade. Assim sendo, a dinâmica do i -ésimo dedo é descrito como:

$$M_{if} = I_i \ddot{\theta}_i + J_i^T F_0 \quad (13)$$

onde $I_i \in R^{n_i \times n_i}$ é a matriz de inércia, $\theta_i \in R^{n_i}$ são as coordenadas generalizadas, e $J_i = \partial r_i / \partial \theta_i \in R^{3 \times n_i}$ é a matriz Jacobiana do i -ésimo dedo.

Pode-se assumir também que:

$$M_{if} = J_i^T f_{id} \quad (14)$$

que é gerado pela juntas e no tempo inicial a garra exerce $F_0 = f_{id}$, o qual está incluso no cone de atrito. Onde f_{id} é a força aplicada pelo dedo sob o objeto.

Assumindo que a aceleração do objeto no ponto de contato, causado pela força de perturbação é representado por \ddot{r}_i , pode se avaliar a ação da aceleração. Supondo também que a garra pode seguir o movimento do ponto de contato, tem-se que $\dot{\theta}_i = 0$, logo a seguinte equação é satisfeita:

$$\ddot{r}_i = J_i \ddot{\theta}_i + \dot{J}_i \dot{\theta}_i = J_i \ddot{\theta}_i \quad (15)$$

Considerando a matriz de inércia I_i e as equações (14) e (15), as seguintes equações são derivadas da equação (13):

$$F_0 = f_{id} + f_{ia} \quad (16)$$

$$f_{ia} = (J_i I_i^{-1} J_i^T)^{-1} \ddot{r}_i \quad (17)$$

onde f_{ia} é a força gerado pela perturbação, $rank J_i = 3$ é assumido, I_i é positivo, e logo, $J_i I_i^{-1} J_i^T$ torna-se positivo.

A equação (16) mostra a força exercida sobre o objeto quando o ponto de contato é acelerado de \ddot{r}_i . A partir do momento que a ponta dos dedos começa a escorregar, F_0 está no limite do cone de atrito. A relação entre a estabilidade de contato e \ddot{r}_i é mostrada esquematicamente na figura 5.

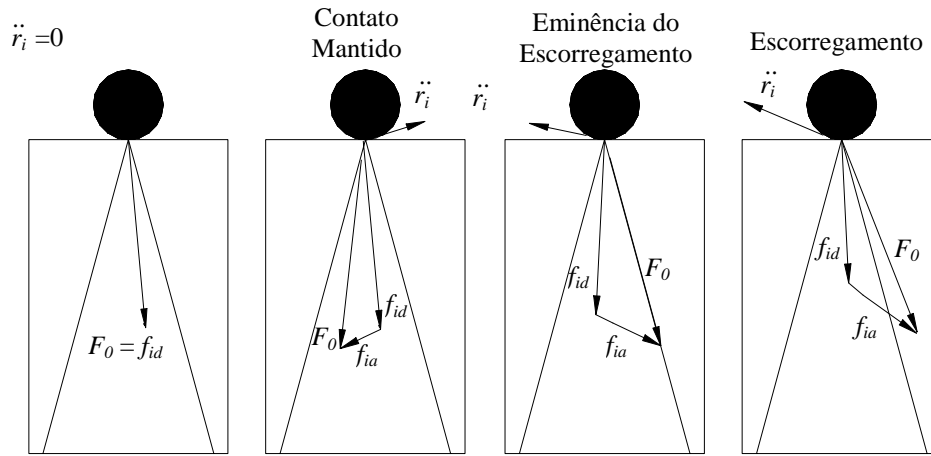


Figura 5. Estabilidade de Contato e Aceleração do Ponto de Contato

Assim, pode-se considerar um conjunto de esferas de pontos de contato com aceleração:

$$\left\{ \ddot{r}_i; \|\ddot{r}_i\| \leq a \right\} \quad (18)$$

onde a é um constante escalar positiva.

Assim, tem-se o conjunto de f_{ia} para todos os possíveis \ddot{r}_i :

$$\left\{ f_{ia}; f_{ia}^T (J_i I_i^{-1} J_i^T)^T (J_i I_i^{-1} J_i^T) f_{ia} \leq a^2 \right\} \quad (19)$$

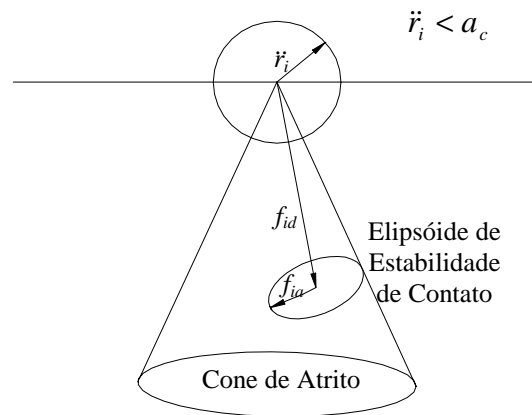


Figura 6. Elipsóide de Contato

Como $J_i I_i^{-1} J_i^T$ é definido como positivo, das equações (16) e (19), tem-se que o conjunto de pontos finais de f_i forma um elipsóide com centro em f_{id} e eixo principal $2a/M_{ij}$ ($j=1,2,3$) de comprimento, onde $M_{ij} > 0$ é o resultado de $J_i I_i^{-1} J_i^T$ (Golub & Van Loan, 1983). Considerando que a_c é o valor mínimo de a para produzir um elipsóide de contato (figura 6) com o cone de atrito, tem-se que a_c implica no máximo raio da esfera de \tilde{r}_i que não causa o escorregamento.

5. CONCLUSÃO

Neste trabalho foi discutida a importância da análise das forças de agarramento para a estabilidade do aperto. Foram também avaliados a questão da estabilidade do objeto que depende da rigidez e do atrito entre o manipulador robótico e o objeto a ser agarrado, e a estabilidade do contato que depende das forças resultantes que por sua vez, são dependentes das forças internas. A avaliação da ação de uma força perturbadora, dentro de um elipsóide de forças de contato, permite que não haja escorregamento do objeto, foi realizada. A questão de agarramento e manipulação de maneira segura requer análises mais profundas, onde a avaliação das interações entre os manipuladores robóticos e os objetos a serem agarrados devem ser o próximo passo dessa pesquisa.

6. REFERÊNCIAS

- CHERIF, M., GUPTA, K.K., “*Global Planning for Dexterous Re-orientation of Rigid Objects: Finger Tracking with Rolling and Sliding*”, Technical Report, INRIA – Institut National de Recherche en Informatique et en Automatique, France, 1999;
- CUTKOSKY, M.R., WRIGHT, P.K., “*Friction, Stability and the Design of Robotic Fingers*”, International Journal Robotics Researches, 1986;
- DARIO, P., DE ROSSI, D., “*Tactile Sensors and the Gripping Challenge*”, IEEE Spectrum, August, 1985;
- DUTRA, M. S., NAKA, M.H., “*Study of Kinematical Redundancy of a Mechanical Hand with Five Fingers*”, XV COBEM, Águas de Lindóia, Brasil, 1999;
- GOLUB, G.H., VAN LOAN, C.F., “*Matrix Computations*”, Baltimore: John Hopkins University Press, 1983;
- HANAFUSA, H., ASADA, H., “*Stable Prehension by a Robot Hand with Elastic Fingers*”, 7th Int. Symp. Industrial Robots, 1977;
- JAMERSON, J.W., LEIFER, L.J., “*Quasi-static Analysis: A Method for Predicting Grasp Stability*”, Proc. of IEEE Int. Conf. on Robotics and Automation, 1986;
- KERR, J., ROTH, B., “*Analysis of Multifingered Hands*”, International Journal of Robotics Researches, 1986;
- MURRAY, R.M., ZEXIANG, L., SASTRY, S. S., “*A Mathematical Introduction to Robotic Manipulation*”, CRC Press, 1st Edition, London, 1993;
- NAKAMURA, Y., NAGAI, K., YOSHIKAWA, T., “*Dynamics and Stability in Coordination of Multiple Robotic Mechanisms*”, The International Journal of Robotics Research, Vol. 8, no. 2, April 1989;
- SALISBURY, J.K., “*Kinematic and Force Analysis of Articulated Hands*”, Ph.D. Thesis, Stanford University, Dept. of Mechanical Engineering, 1982.

APLICAÇÃO DE REDES NEURAIS NO MOVIMENTO DE UM ROBÔ BÍPEDE

Max Suell Dutra

Armando Carlos de Pina Filho

Taltibio Araujo Ezequiel

Universidade Federal do Rio de Janeiro, Departamento de Engenharia Mecânica, Centro de Tecnologia, COPPE/PEM, C.P. 68503 - CEP. 21945-970, Rio de Janeiro, RJ, Brasil. E-mails: max@serv.com.ufrj.br, pina-filho@mailbr.com.br.

Resumo

A locomoção de seres humanos e animais, como a caminhada e a corrida, é desenvolvida através de movimentos rítmicos e sincronizados. Admitindo-se o ângulo formado pelas pernas quando realizamos um passo (chamado aqui de ângulo da bacia) como o principal parâmetro responsável pelo posicionamento da perna, o objetivo deste trabalho é determinar seu comportamento utilizando uma rede neural, de maneira a minimizar o tempo de cálculo para envio do sinal de posição. Através do emprego de uma rede neural, pode-se garantir que esse cálculo será realizado em “tempo real”, a partir do comprimento do passo desejado.

Palavras-chave: *Backpropagation*, Bípede, Controle, Rede Neural.

1. INTRODUÇÃO

Quando realizamos o movimento das pernas numa caminhada, um grande número de graus de liberdade está envolvido, tornando-se essencial uma boa coordenação entre eles. A maior parte dessa coordenação é realizada pelo sistema nervoso central, gerando sinais de acordo com o passo desejado. O processo de geração de sinais pode ser modelado por um gerador de padrões, sendo projetado como uma rede de osciladores não lineares mutuamente acoplados.

Uma outra maneira de gerar esses sinais é através de uma rede neural artificial, a qual consiste de elementos computacionais simples (neurônios), organizados em camadas e funcionando em paralelo. A rede neural tem habilidade para “aprender” relações complicadas de não-linearidades e tem sido amplamente utilizada na modelagem de sistemas complexos, proporcionando o controle de sistemas altamente não-lineares através do método clássico de *Backpropagation* (Wasserman, 1989).

Um dos maiores problemas (talvez o maior) para a elaboração de uma máquina para locomoção bípede é o controle, este sendo responsável pelo movimento, estabilidade da máquina e manutenção de certos requisitos adicionais, como por exemplo, baixo consumo de energia. Essas atividades dividem-se em tarefas como:

- controle simultâneo de atuadores;
- cálculo da trajetória das pernas e coordenação entre elas;
- posicionamento final dos pés;
- e adaptação global do movimento ao ambiente.

Todas essas tarefas precisam ser conduzidas em tempo real, sendo a rede neural a alternativa mais promissora ao controle tradicional, pois proporciona o aprendizado e tem alta adaptabilidade. A arquitetura de controle é compilada através do treinamento de uma rede neural em *Backpropagation*, modelagem também adotada por Lopes Júnior *et al.* (1999).

2. FATORES DETERMINANTES DO PASSO DE UM BÍPEDE

O movimento natural de um bípede envolve mais de 200 graus de liberdade, tornando-se a sua modelagem mais viável através da redução do número de graus de liberdade, analisando-se aqueles que serão os mais importantes para a determinação do passo, a saber: o compasso, que se apresenta com as pernas esticadas, como num pêndulo invertido; a rotação pélvica sobre o eixo vertical; a flexão do joelho; a flexão da planta do pé; o movimento lateral da pélvis; e a inclinação da pélvis (Saunders *et al.*, 1953).

Para uma melhor percepção desses fatores determinantes do passo, um modelo tridimensional com 15 graus de liberdade é necessário; entretanto, de maneira a simplificar o estudo técnico, admite-se adotar um modelo bidimensional (Dutra, 1995), que desempenha seu movimento em um plano, com 3 determinantes do passo (Figura 1).

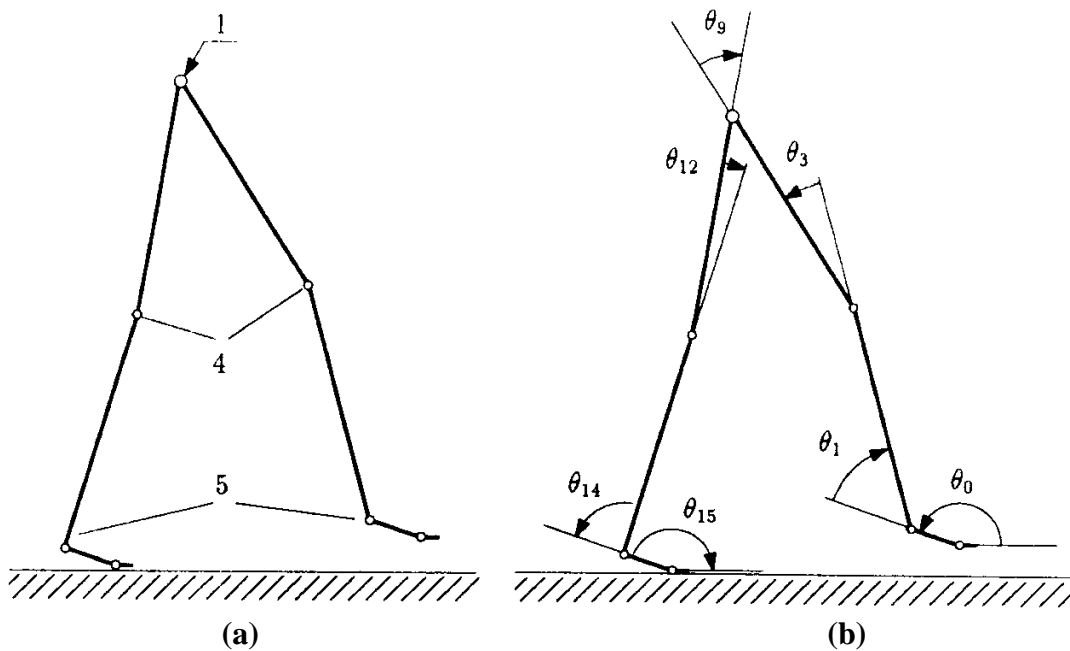


Figura 1. Modelo bidimensional de um robô bípede. (a) Determinantes principais do modo de andar: compasso (1), flexão do joelho (4) e flexão da planta do pé (5); (b) ângulos necessários para a descrição do movimento.

Uma vez que a locomoção de um bípede exige um movimento sincronizado, é necessário que o gerador de padrões proporcione ao sistema as referências adequadas para o movimento dos *links* (articulações). No presente estudo, o gerador de padrões fornece ao sistema de locomoção sinais de referência angular (θ_9) através de uma rede neural em *Backpropagation*.

3. TRABALHO EXPERIMENTAL E RESULTADOS

Estudos do movimento humano, bem como a análise de Fourier (Braune, 1987), levam aos resultados apresentados na Figura 2 para θ_1 , θ_3 , θ_9 , θ_{12} , e θ_{14} . A síntese desses dados mostra que o movimento dos ângulos dos joelhos θ_3 e θ_{12} , bem como o ângulo θ_9 , podem ser descritos com grande precisão por seu harmônico fundamental, estando o bípede nas fases de suporte duplo ou simples (Figura 3).

As funções harmônicas têm as seguintes formas:

$$\theta_3 = \theta_{30} + \hat{\theta}_3 \cos 2\omega t \quad (1)$$

$$\theta_9 = \theta_{90} + \hat{\theta}_9 \cos (\omega t - \alpha_9) \quad (2)$$

$$\theta_{12} = \theta_{120} + \hat{\theta}_{12} \cos 2\omega t \quad (3)$$

onde: θ_{30} , θ_{90} , e θ_{120} se referem aos valores iniciais dos ângulos; $\hat{\theta}_3$, $\hat{\theta}_9$, e $\hat{\theta}_{12}$ se referem às amplitudes; ω é a velocidade angular; t é o tempo; e α_9 corresponde ao ângulo de defasagem para determinação de θ_9 .

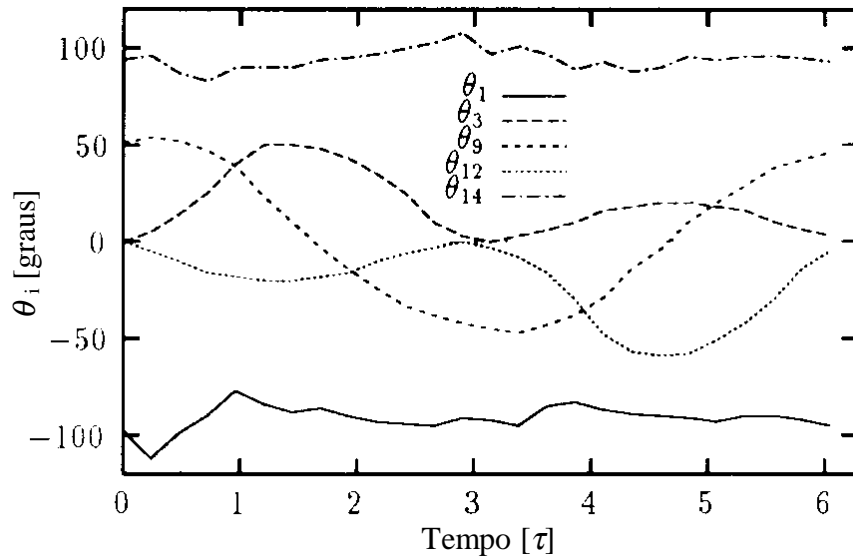


Figura 2. Comportamento dos ângulos das pernas em função do tempo.

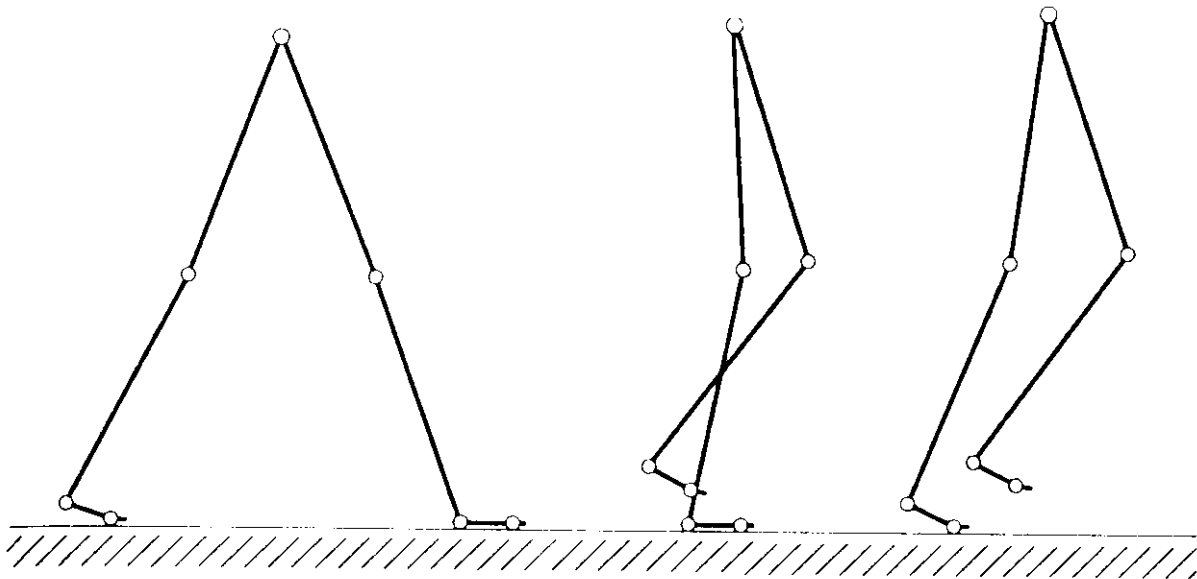


Figura 3. Movimentação das pernas, mostrando as fases de suporte duplo e simples.

A equação para geração dos comprimentos do passo é a seguinte (Dutra, 1995):

$$S = l_f (1 + \cos \theta_0) + l_u \cos (\theta_0 + \theta_1) + l_o \cos (\theta_0 + \theta_1 + \theta_3) - l_o \cos (\theta_0 + \theta_1 + \theta_3 + \theta_9) - l_u \cos (\theta_0 + \theta_1 + \theta_3 + \theta_9 + \theta_{12}) \quad (4)$$

onde: l_f é o comprimento do pé; l_u é o comprimento da tíbia; e l_o é o comprimento do fêmur.

No entanto, como o interesse é especificar o ângulo θ_9 a partir de um comprimento do passo desejado, deve-se encontrar a expressão de θ_9 em função do passo. Usando a expressão anterior e fazendo as devidas manipulações algébricas, chega-se ao seguinte resultado:

$$\theta_9 = \arccos \left[\frac{B + \sqrt{B^2 - 4AC}}{2A} \right] \quad (5)$$

onde: $A = (\ell_o \cos x + \ell_u \cos y)^2 + (\ell_o \sin x + \ell_u \sin y)^2$;

$B = 2z (\ell_o \cos x + \ell_u \cos y)$;

$C = z^2 - (\ell_o \sin x + \ell_u \sin y)^2$;

$x = \theta_0 + \theta_1 + \theta_3$;

$y = \theta_0 + \theta_1 + \theta_3 + \theta_{12}$;

$z = -s + \ell_f (1 + \cos \theta_0) + \ell_u \cos(\theta_0 + \theta_1) + \ell_o \cos(\theta_0 + \theta_1 + \theta_3)$.

Segundo medidas experimentais, tem-se normalmente durante uma caminhada: $\theta_0 = 170^\circ$, $\theta_1 = -95^\circ$, $\theta_3 = 3^\circ$ e $\theta_{12} = -3^\circ$, que são, respectivamente, o ângulo da planta do pé com o solo, o ângulo entre a articulação inferior da perna (tornozelo) e a parte posterior do pé, e os ângulos referentes à flexão dos joelhos. Além disso, pode-se adotar as seguintes medidas para os comprimentos: $\ell_f = 0,11$ m, $\ell_u = 0,37$ m e $\ell_o = 0,37$ m. Com estas medidas pode-se então configurar a rede para gerar os resultados desejados.

A rede foi configurada para treinamento com os seguintes parâmetros:

- comprimento mínimo do passo: 0,165 m;
- comprimento máximo do passo: 0,715 m;
- camada intermediária com 5 neurônios do tipo sigmóide;
- camada de saída com 1 neurônio do tipo linear;
- função utilizada: *Levenberg-Marquardt backpropagation*;
- número máximo de épocas para treino: 1000;
- erro médio quadrático (MSE): 1×10^{-5} .

Após o processo de treinamento, verifica-se a capacidade de generalização da rede, escolhendo-se diferentes valores de passo e especificando-se θ_9 (fase de simulação da rede). Em conjunção com a simulação da rede realiza-se o cálculo teórico de θ_9 com o objetivo de comparar os resultados e avaliar o erro percentual dos valores fornecidos pela rede.

A rede neural foi programada no software MATLAB utilizando as seguintes funções (já existentes no próprio programa):

- NEWFF: cria uma rede do tipo “*feed-forward backpropagation*”;
- TRAINLM: usa a otimização de Levenberg-Marquardt para o treinamento;
- TANSIG: especifica o(s) neurônio(s) do tipo sigmóide;
- PURELIN: especifica o(s) neurônio(s) do tipo linear;
- TRAIN: efetua o treinamento da rede;
- SIM: efetua a simulação da rede.

Executando o programa pode-se observar então os resultados encontrados por meio dos gráficos apresentados pelo próprio programa (Figuras 4, 5, 6, e 7), possibilitando uma avaliação do desempenho da rede.

Além dos gráficos, com os valores fornecidos pelo programa foi criada a Tabela 1 contendo os dados de entrada (comprimento do passo), os dados de saída (ângulo da bacia), tanto teóricos como obtidos pela rede, e também o erro percentual (note que o mesmo se encontra abaixo de 0,1% em todos os pontos, o que significa que a rede apresentou um desempenho excelente).

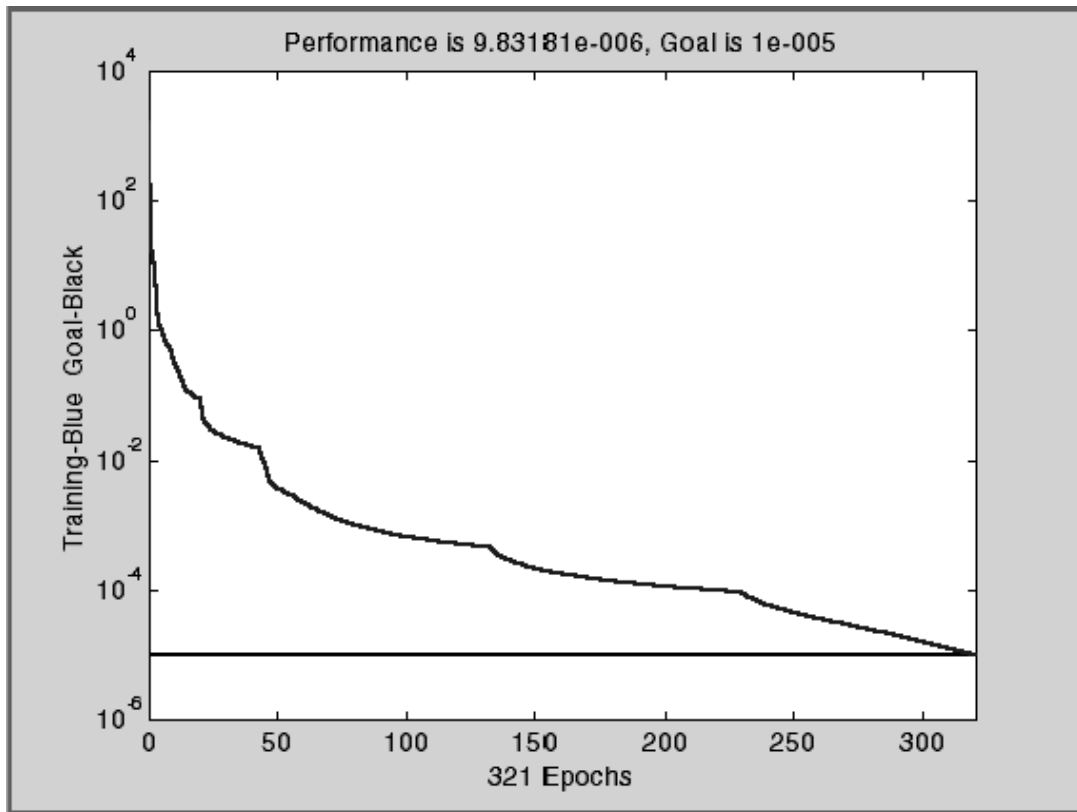


Figura 4. Treinamento da rede (erro médio quadrático: 1×10^{-5}).

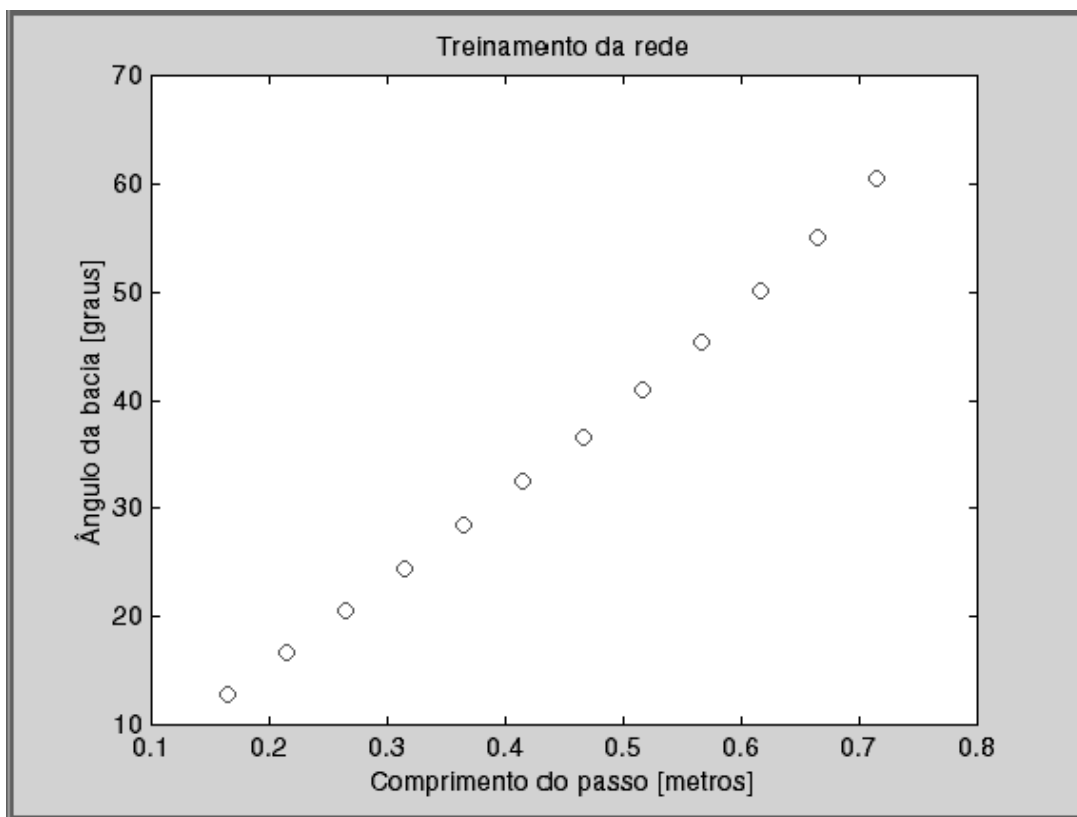


Figura 5. Resultados do treinamento (ângulo da bacia em função do passo).

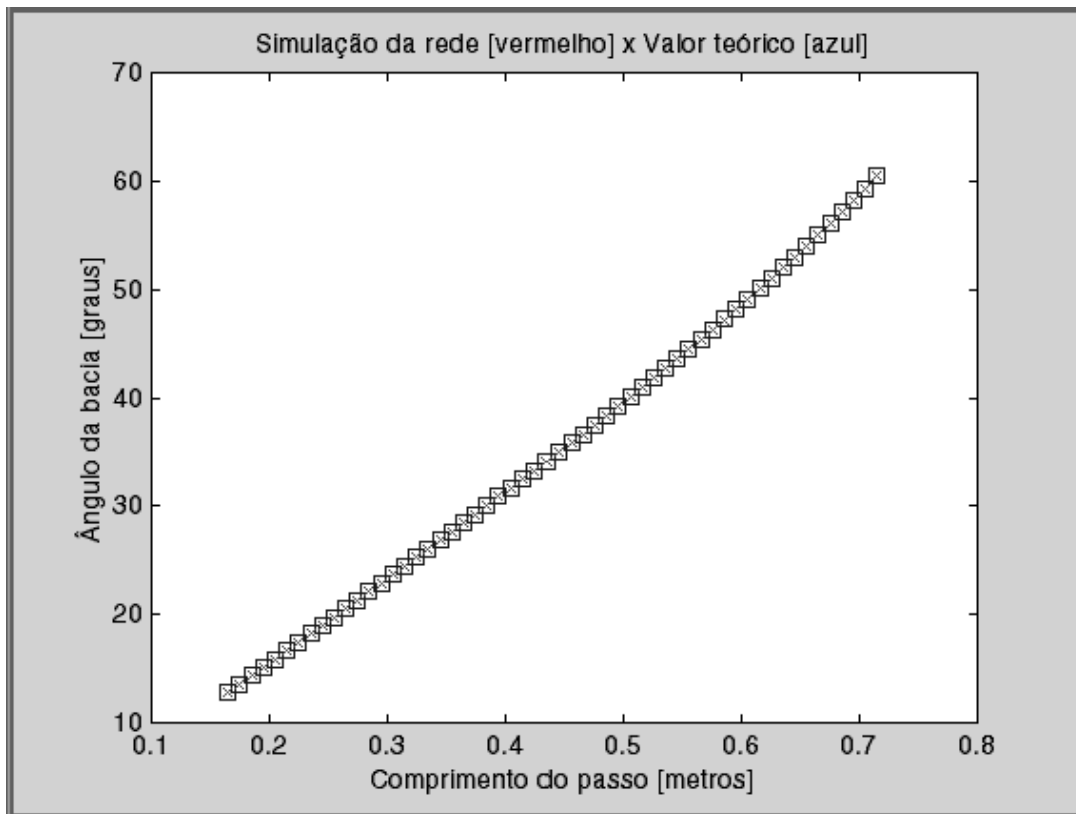


Figura 6. Comparação entre os valores da rede (x) e valores teóricos ().

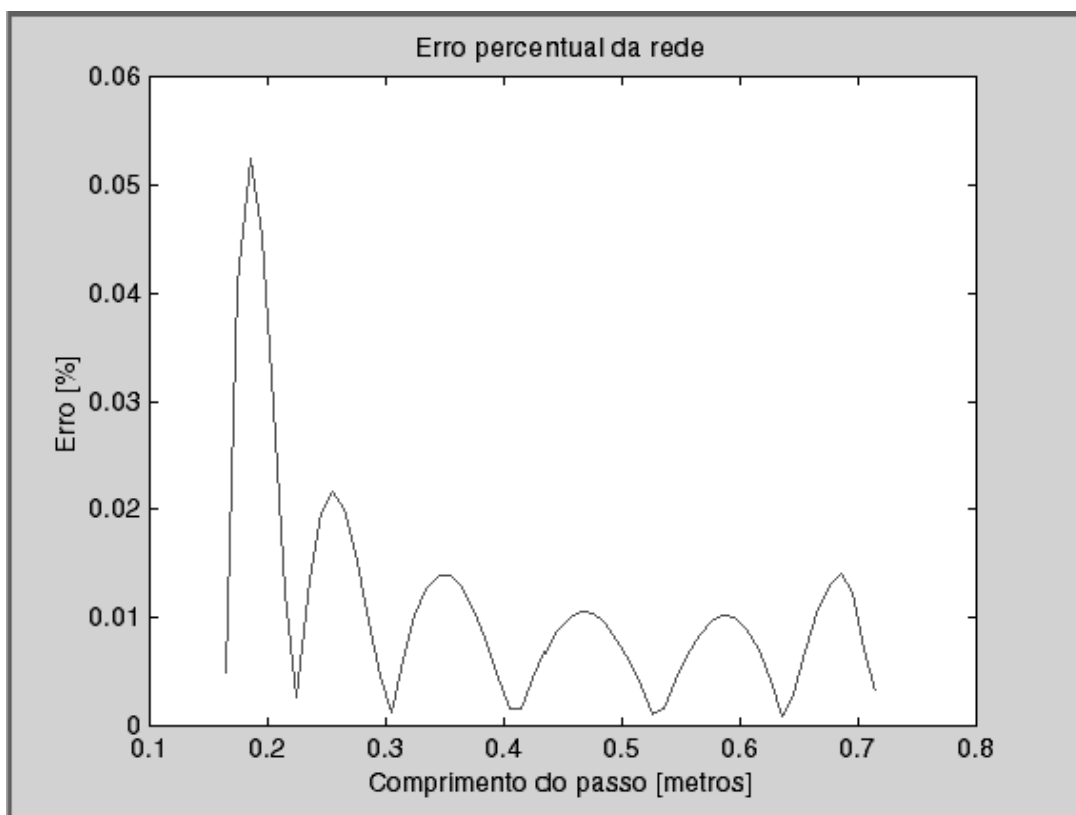


Figura 7. Avaliação do erro percentual dos resultados da rede.

Tabela 1. Alguns resultados experimentais finais.

Comp. do passo [m]	θ , teórico [graus]	θ , da rede [graus]	Erro da rede [%]
0,1650	12,7749	12,7755	0,0049
0,1850	14,3240	14,3165	0,0525
0,2050	15,8737	15,8690	0,0296
0,2250	17,4252	17,4257	0,0027
0,2450	18,9795	18,9833	0,0197
0,2650	20,5379	20,5420	0,0200
0,2850	22,1016	22,1039	0,0105
0,3050	23,6717	23,6714	0,0012
0,3250	25,2496	25,2471	0,0101
0,3450	26,8366	26,8329	0,0140
0,3650	28,4341	28,4305	0,0128
0,3850	30,0436	30,0411	0,0080
0,4050	31,6666	31,6660	0,0017
0,4250	33,3048	33,3063	0,0044
0,4450	34,9601	34,9632	0,0088
0,4650	36,6344	36,6383	0,0105
0,4850	38,3299	38,3335	0,0095
0,5050	40,0489	40,0513	0,0060
0,5250	41,7941	41,7946	0,0010
0,5450	43,5685	43,5666	0,0042
0,5650	45,3752	45,3714	0,0084
0,5850	47,2181	47,2132	0,0103
0,6050	49,1014	49,0970	0,0090
0,6250	51,0302	51,0280	0,0043
0,6450	53,0102	53,0118	0,0029
0,6650	55,0484	55,0541	0,0104
0,6850	57,1530	57,1610	0,0140
0,7050	59,3341	59,3383	0,0071
0,7150	60,4571	60,4552	0,0032

4. CONCLUSÕES

Este estudo demonstra a possibilidade do emprego de redes neurais para determinação dos ângulos de posicionamento para um sistema mecânico de locomoção bípede, fornecendo resultados excelentes, onde os erros apresentados pela rede se encontram abaixo de 0,1% em todos os pontos.

Vale ressaltar a grande importância da fase de treinamento, onde a rede consegue assimilar as informações relativas ao problema, possibilitando um desempenho ótimo na simulação e consequentemente apresentando os resultados desejados ao final do processo. A Figura 8 mostra o que aconteceria no caso de um treinamento mal sucedido.

A utilização de redes neurais para locomoção em velocidades normais (caminhada) se mostrou muito eficiente, porém não foram analisados casos onde o ângulo da bacia seja maior que 70° (como na subida de uma escada), ou mesmo em situações onde ambos os pés deixam de ter contato com o solo por alguns instantes (como na corrida).

Logo, verifica-se que o estudo aqui apresentado é o ponto de partida para elaboração de redes mais complexas, aplicáveis não somente ao caso de uma caminhada, mas também a situações como a corrida e subida de uma escada.

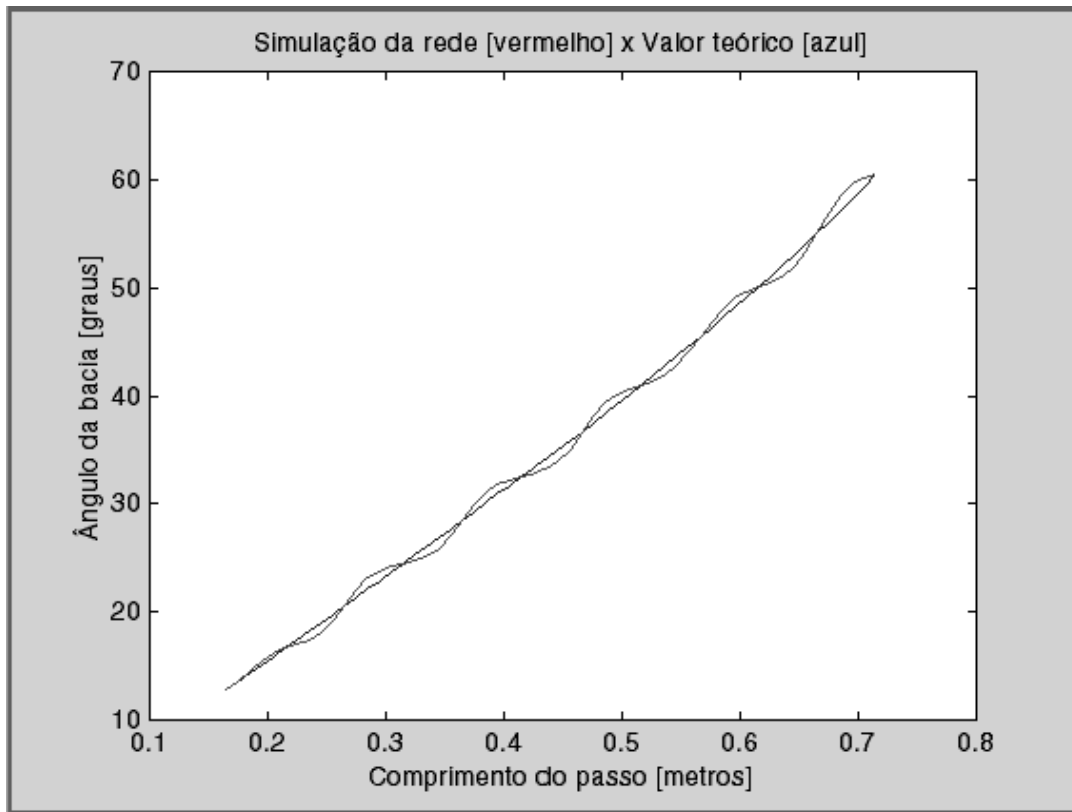


Figura 8. Treinamento mal sucedido (os resultados fornecidos pela rede apresentam discrepâncias, oscilando em torno dos valores teóricos).

5. REFERÊNCIAS BIBLIOGRÁFICAS

- Braune, W., Fischer, O., 1987, “The Human Gait”. Springer Verlag, Berlin, Germany. Translated from Der Gang des Menschen, B.G. Teubner, 1904.
- Dutra, M. S., 1995, “Bewegungskoordination und Steuerung einer zweibeinigen Gehmaschine”, Shaker Verlag, Aachen, Germany.
- Lopes Júnior, V., Dutra, M. S., Nascimento, Á. O. do, 1999, “Modeling of a Bipedal Locomotor Using Neural Networks”, Applied Mechanics in the Americas, Vol. 8, AAM and ABCM, Rio de Janeiro, Brazil.
- Saunders, J. B., Inman, V., Eberhart, H. D., 1953, “The Major Determinants in Normal and Pathological Gait”, J. Bone Jt. Surgery 35 A.
- Wasserman, P. D., 1989, “Neural Computing: Theory and Practice”, Van Nostrand Reinhold, New York, USA.

UTILIZAÇÃO DO CONCEITO DE REDES NEURAIIS COM CONTATOS SINÁPTICOS MÚLTIPLOS NO CONTROLE DE TRAJETÓRIA DE UM MANIPULADOR BI-ARTICULADO

Benedito Dias Baptista Filho

Comissão Nacional de Energia Nuclear

Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN/SP

Centro de Engenharia Nuclear – Divisão de Termo-Hidráulica

E-mail: bdbfilho@net.ipen.br

Travessa “R” nº 400 – CEP 05508-900 – São Paulo – SP – Brasil

Resumo

Um novo conceito de redes neurais artificiais baseado na adaptação de circuitos neurais biológicos com múltiplos contatos sinápticos, considerando novas funções de transferência e utilizando conceitos de aprendizado baseados nos princípios da memória reflexiva, foi utilizado no controle de um manipulador bi-articulado no plano para comandar a extremidade do manipulador sobre uma trajetória desejada. Este trabalho apresenta os novos conceitos, resume as principais características dessa classe de redes neurais, e apresenta resultados preliminares do controle do manipulador sobre trajetórias comandadas por mudanças nos sinais de posição desejada. Os resultados apresentados foram obtidos utilizando-se diferentes séries de intervalos de tempo para comandar as mudanças de posição desejadas, orientando o controlador de posição para seguir duas trajetórias retilíneas. Os resultados do teste apresentado sugerem a viabilidade de utilização de uma nova camada responsável pelo aprendizado da função da trajetória e pela coordenação da mudança dos sinais de posição.

Palavras-chave: Redes Neurais, Manipuladores, Controle, Robótica

1. INTRODUÇÃO

O objetivo do artigo é apresentar o mais recente trabalho de aplicação de uma nova classe de redes neurais artificiais batizada de *MULSY N-N (MULti-SYnaptic Neural Network)*, introduzidas por Baptista F. (1998). Essa nova categoria de redes neurais considera o projeto de redes especializadas por tarefa, baseadas na arquitetura de sistemas neurais biológicos, como apresentado por Baptista F. e Cabral (1999a e 1999b). Este trabalho consiste de seis seções. A primeira seção é esta introdução. A segunda seção resume os conceitos dessa rede. A terceira seção apresenta os modelos utilizados para aplicação no controle de posição do manipulador. A quarta seção apresenta a metodologia utilizada para controle de trajetória e os principais resultados. A quinta seção apresenta as principais conclusões abordando novas propostas.

2. O CONCEITO DE REDES *MULSY*

O conceito de redes *MULSY* foi introduzido por Baptista F. (1998). Esses conceitos foram baseados em circuitos e funções neurológicas. Sugere-se uma consulta a Kandel *et al.* (1991), que é uma das melhores referências para compreensão dos conceitos da neurofisiologia abordados nesses trabalhos. A principal base dos novos conceitos está no uso de múltiplos contatos sinápticos modelados por funções que imitam uma Gauseana. A Figura 1 exhibe graficamente uma dessas funções, generalizada por:

$$S = \frac{T}{1 + a(I - I_0)^2}, \quad (1)$$

onde, T é a “força de ligação” da sinapse, que pode assumir qualquer valor positivo (excitatória) ou qualquer valor negativo (inibitória), a é uma constante que pode ser escolhida adequadamente para produzir funções sem variações abruptas, de acordo com o número de terminais sinápticos, I é o valor do sinal transmitido no axônio, e, I_0 é o valor de I que maximiza S , o valor de saída para a célula alvo. Esse valor I_0 é chamado de “limiar”.

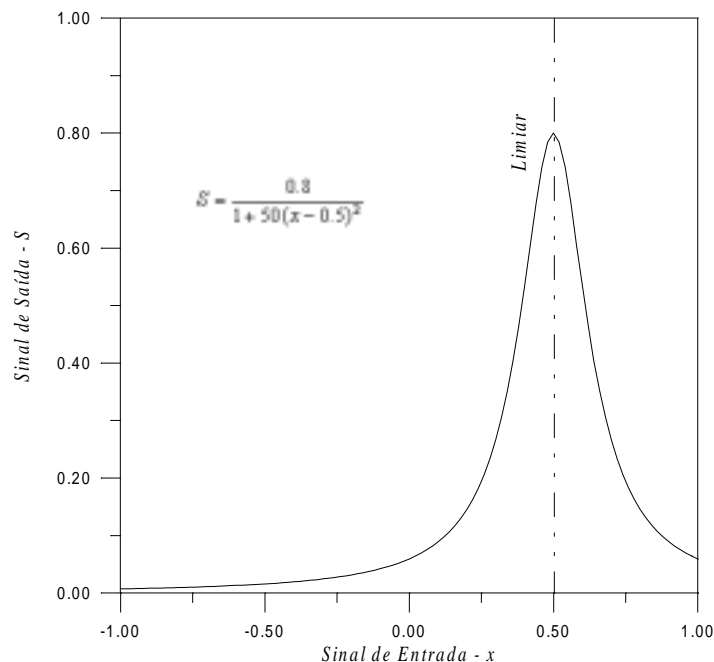


Figura 1. Função de Transferência Sináptica

A Eq. (1) representa um único terminal sináptico. Uma característica dessa função é que permite amplificação e resposta seletiva, aprimorando a função de transferência global do neurônio, sendo muito mais “leve”, em termos de processamento, do que uma função sigmóide. Com forças e limiares convenientes, um conjunto dessas funções pode reproduzir qualquer tipo de função contínua.

Em sistemas neurais biológicos, o conhecimento da habilidade adquirida é armazenado por meio de mecanismos de memória reflexiva, envolvendo alterações plásticas em sinapses químicas. As sinapses podem sofrer diferentes modificações em várias formas de aprendizado, por exemplo, podem ser deprimidas pela habituação ou podem ser reforçadas pela sensibilização. Formas de aprendizado mais complexas são o condicionamento clássico e a prática. Os processos envolvidos nos mecanismos de memória reflexiva sugeriram o desenvolvimento de

um circuito que se mostrou adequado para a implementação de um processo de aprendizado. Esse circuito foi implementado no que se denominou “*unidade de controle motor.*”

As redes *MULSY* utilizam “*unidades de controle motor*” cuja arquitetura básica está representada na Fig. 2. Nessa arquitetura são reunidas características adequadas para um controlador, recebendo sinais de dois caminhos: um do sistema sensório (p. ex., a posição atual); e, um de um nível de controle hierarquicamente superior (p. ex., uma posição desejada). Os sinais do sistema sensório e do nível de controle superior convergem para uma unidade de saída (neurônio motor), sendo transmitidos através de dois conjuntos de contatos sinápticos, simétricos em termos de limiar e força de ligação. Essas sinapses possuem características plásticas e têm suas “*forças de ligação*” ajustadas pelo aprendizado. Os modelos desenvolvidos para essas sinapses, representados pelas equações indicadas na Fig. 2, são detalhados por Baptista F., Cabral, e Soares (1998). Antes de qualquer treinamento, essas sinapses são *inativas*, i.e., $T_{j(k)} = 0$. O surgimento de um sinal de erro ε gera um sinal δ diferente de zero que age no sentido de aumentar ou diminuir um fator de disparo de mudanças de longa duração, o fator C . Esse fator comanda as mudanças plásticas, que representam o processo de aprendizado e que ocorrem somente nas sinapses da unidade de saída. O sinal do “desejo” é a referência para o ajuste dessas sinapses plásticas, mesmo aquelas do caminho do sistema sensório. Unidades responsáveis pela geração de sinais proporcionais às taxas de mudança dos sinais sensórios também participam desse processo, sendo interligadas por conexões “rígidas” (sem plasticidade). Os sinais de saída dessas unidades, combinados com o sinal de erro, além de participarem do processo de treinamento, controlam a dinâmica do sistema em um modelo que equivale à soma de $a_0\varepsilon + a_1d\varepsilon/dt + a_2d^2\varepsilon/dt^2 + \dots$. O coeficiente a_0 do erro é implementado por funções sinápticas que resultam em uma função de transferência linear:

$$S_{eps} = \frac{1}{N} \left(\frac{T_\varepsilon}{1 + 0.25(I - 2)^2} \right); \quad (2)$$

$$S_{ips} = \frac{1}{N} \left(\frac{-T_\varepsilon}{1 + 0.25(I + 2)^2} \right); \quad (3)$$

onde T_ε é força de ligação das sinapses de erro.

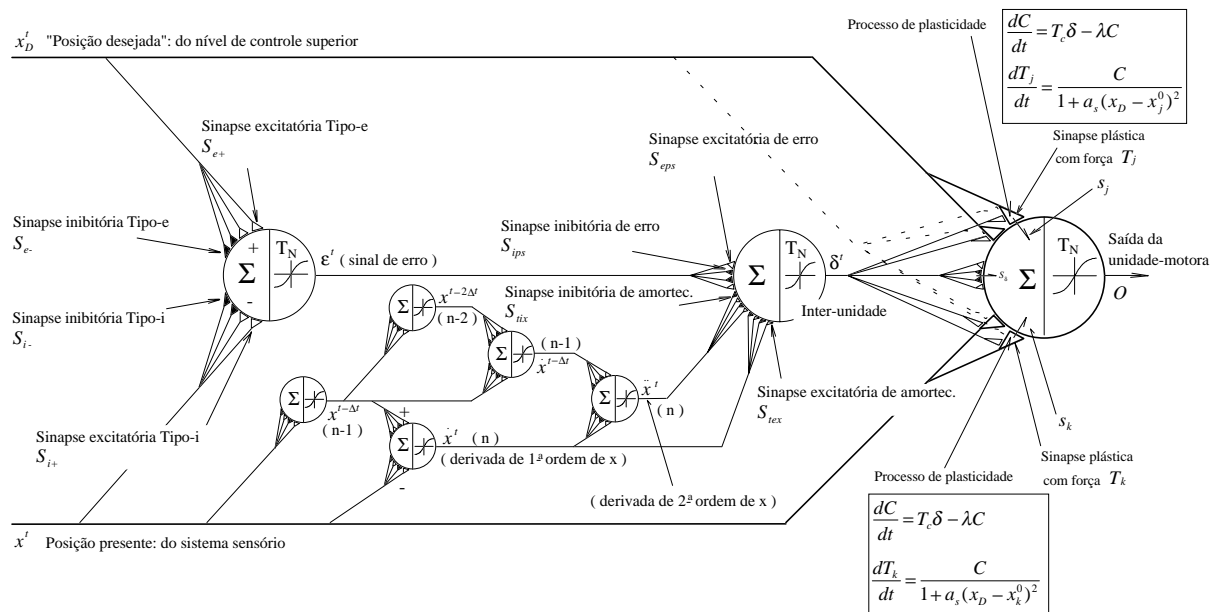


Figura 2. Conceito de Unidade de Controle Motor

As funções de transferência sináptica nas conexões dos sinais de taxas com a unidade facilitadora modelam características de amortecimento do tipo $x/x/$. Isso permite a atenuação de oscilações tornando o processo estável mesmo na presença de altas taxas de variação. Esse amortecimento é encontrado em sistemas biológicos, tanto na formação de circuitos de neurônios quanto nas próprias características físicas dos músculos:

$$S_{tex} = \frac{1}{N} \left(\frac{T_r}{1+11(I-1)^2} \right); \quad (4)$$

$$S_{tix} = \frac{1}{N} \left(\frac{-T_r}{1+11(I+1)^2} \right); \quad (5)$$

onde T_r é a força de ligação das sinapses de mudanças.

A função de transferência nas unidades que representam os neurônios é uma *tangente hiperbólica modificada*:

$$O = T_N \tanh(\alpha \sum S), \quad (6)$$

onde O é o sinal de saída, T_N representa o “tamanho” da unidade, α é um ganho, e $\sum S$ é a soma de todas as entradas sinápticas na unidade. O “tamanho” pode ser ajustado a valores convenientes para, por exemplo, melhorar a linearidade de resposta em uma determinada faixa de interesse, ou para amplificar ou atenuar a relação entrada/saída.

3. MODELOS DE SIMULAÇÃO

A. Modelo da Dinâmica do Processo

O manipulador bi-articulado é um problema não-linear com dois graus de liberdade. As variáveis consideradas no modelo são apresentadas na Fig. 3.

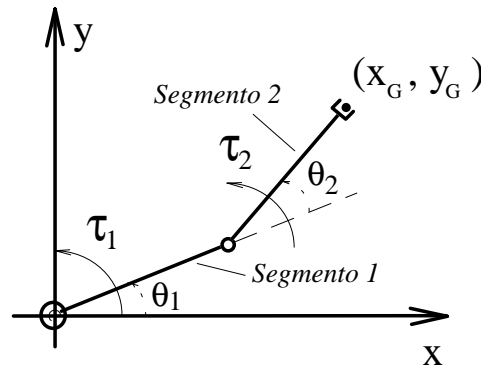


Figura 3. Modelo do Manipulador bi-articulado

A dinâmica desse sistema é representada por:

$$\tau_1 = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 + h_{122}\dot{\theta}_2^2 + h_{121}\dot{\theta}_1\dot{\theta}_2 + G_1; \quad (7)$$

$$\tau_2 = H_{22}\ddot{\theta}_2 + H_{12}\ddot{\theta}_1 + h_{211}\dot{\theta}_1^2 + G_2; \quad (8)$$

onde θ_1 é o ângulo entre o primeiro segmento e o eixo-x, θ_2 é o ângulo entre o segundo e o primeiro segmentos, τ_1 e τ_2 são os torques nas articulações 1 e 2 respectivamente, e os outros termos são:

$$\begin{aligned}
H_{11} &= m_1 l_{c1}^2 + I_1 + [m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) + I_2]; \\
H_{12} &= m_2 l_1 l_{c2} \cos \theta_2 + m_2 l_{c2}^2 + I_2]; \\
H_{22} &= m_2 l_{c2}^2 + I_2; \\
h_{122} &= -m_2 l_1 l_{c2} \sin \theta_2; \\
h_{121} &= -2m_2 l_1 l_{c2} \sin \theta_2; \\
h_{211} &= m_2 l_1 l_{c2} \sin \theta_2; \\
G_1 &= m_1 g l_{c1} \cos \theta_1 + m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)); \\
G_2 &= m_2 g l_{c2} \cos(\theta_1 + \theta_2).
\end{aligned}$$

Os subscritos 1 e 2 se referem ao i -ésimo segmento do manipulador com massa m_i , comprimento total l_i , distância da articulação ao centro de massa l_{ci} e momento de inércia I_i , e g é a aceleração da gravidade.

Os modelos da dinâmica dos motores acoplados ao manipulador, que são atuados pelos sinais de saída do controlador neural, está detalhada em Baptista F., Cabral, e Soares (1998). Nesses modelos foram consideradas as características dos motores e dos mancais.

B. Controle de Posição com a Nova Rede Neural

Foram utilizadas duas unidades de controle. Os sinais de entrada são aos ângulos, θ_1 e θ_2 , desejados e atuais. A Figura 4 representa a rede que modela os dois controladores acoplados ao processo. Considerando que a posição do segundo segmento afeta a carga no primeiro, o atuador do primeiro motor é alimentado com a soma das saídas dos dois controladores. Levando em conta que o ângulo θ_2 é relativo à direção do primeiro segmento, θ_1 e θ_2 são somados para alimentar o segundo controlador. Destaca-se que essas simplificações não são essenciais, sendo possível um sistema genérico, porém à custa de um maior número de unidades.

Na Figura 4, θ_{1D} e θ_{2D} representam os comandos de posição desejada, O_1 e O_2 são os sinais de saída das unidades de controle motor que irão alimentar os atuadores, D_1 e D_2 , IN^e são as inter-unidades responsáveis pela avaliação dos erros, IN^{RI} são as inter-unidades responsáveis pela avaliação da primeira derivada das taxas de variação (neste problema não são necessárias ordens mais elevadas), IN^F são as inter-unidades facilitadoras do aprendizado, e IN^D são as inter-unidades usadas para gerar os sinais necessários para avaliar as taxas de mudança dos sinais. Observa-se que inter-unidades de entrada, IN^I , foram acrescentadas para desempenhar a função de soma de entradas. Os parâmetros utilizados na simulação, o processo de treinamento e a avaliação de desempenho são apresentados e discutidos na próxima seção.

4. RESULTADOS

A Tabela 1 contém os parâmetros do manipulador. Os parâmetros dos componentes da rede neural estão na Tabela 2. O treinamento é desenvolvido durante a execução de comandos para mudança de posição que constituem 28 posições alvo. Esse modo de treinamento, representado na Fig. 5, caracteriza um processo de aprendizado não-supervisionado. O manipulador parte da posição de repouso, -90° , totalmente estirado, segue no sentido horário para a posição -185° , retorna para a posição de repouso, segue no sentido anti-horário para a posição $+185^\circ$, retornando novamente para a posição de repouso. Neste trabalho, esse conjunto de alvos foi submetido à rede por sete vezes, dando-se 10 segundos para cada mudança de alvo. Após as sete repetições, o sistema já está apto a atingir qualquer posição com suficiente precisão. As forças de ligação dos dois conjuntos de sinapses plásticas, inicialmente iguais a *zero*,

atingem os valores mostrados na Figura 6. Esse processo representa uma sessão de treinamento de 1960 segundos, consumindo apenas 56 segundos de CPU em um computador com processador Celeron 300 MHz. Essa rapidez mostra que o processo de aprendizado é computacionalmente eficiente.

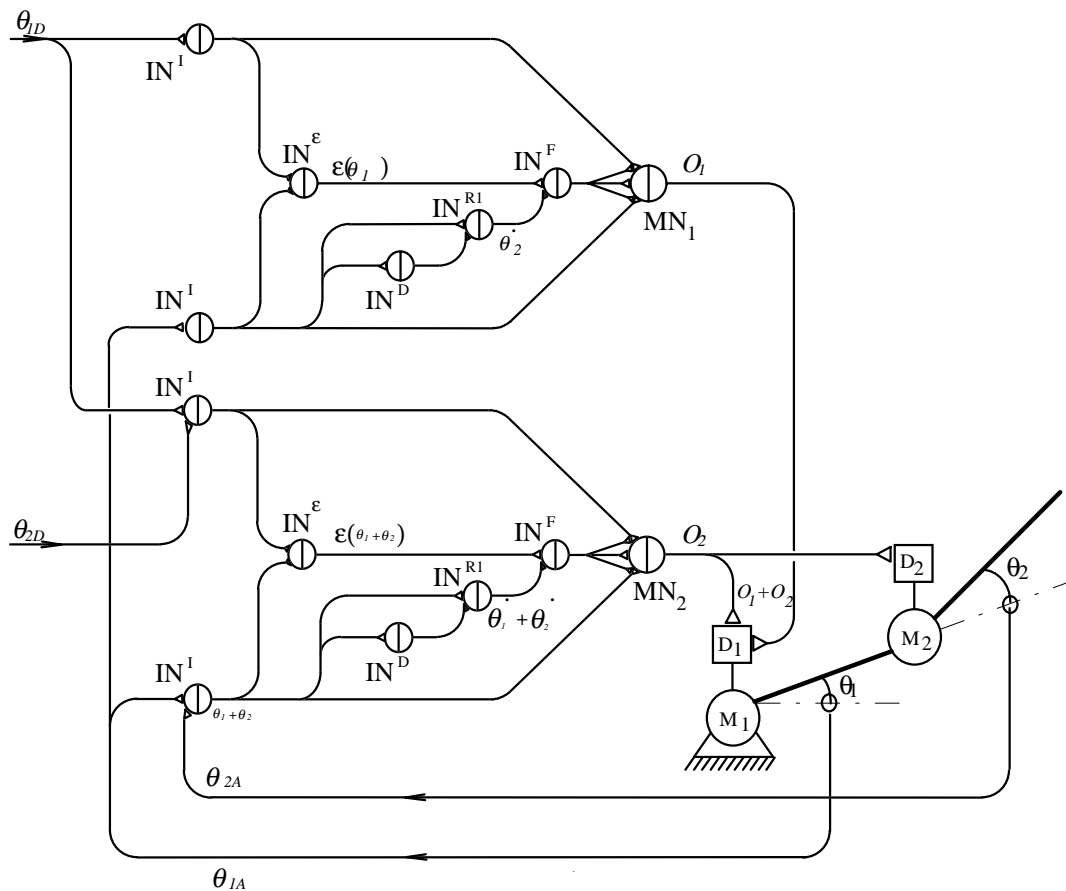


Figura 4. Esquema simplificado da rede para o controle do manipulador

Tabela 1. Parâmetros para o modelo do Manipulador

Parâmetro	Segmento	
	1	2
Comprimento - L (mm)	707	707
Massa - m (kg)	3.0	2.0
Momento de Inércia - I (kg m ²)	0.041	0.027
Ganho de Torque Motor - K_T (N M)	60	30
Momento de inércia do rotor - J_M (kg m ²)	0.0013	0.0013
Constante de perdas nos mancais - K_{LB}	1.0	1.0
Constante de perdas no Motor - K_{LM}	50.1	50.1

Tabela 2. Parâmetros para a Rede Neural

Parâmetro	Valor
Tamanho das unidades - T_N (Eq. 1)	2.1
Constante de ganho das unidades - α (Eq. 1)	0.5
Constante das sinapses plásticas - a (Eq. 5, 6)	20
Número de sinapses plásticas - caminho sensorio	13
Número de sinapses plásticas - caminho "desejo"	13
Intervalo de limiares consecutivos ($I_{a,j} - I_{a,j+1}$)	0.2
Força das sinapses de erro - T_e (Eqs. 7, 8)	2.5
Força das sinapses de taxas - T_r (Eqs. 9, 10)	0.09
Força das sinapses de facilitação - T_c (Eq. 3)	0.1
Constante de decaimento - λ (Eq. 3)	10
Constante de plasticidade sináptica - a_s (Eq. 4)	100

Os testes apresentados neste trabalho consistiram na orientação da extremidade do manipulador para seguir uma trajetória retilínea da posição de repouso (posição inicial da Fig. 5) à posição 11, indo em seguida da posição 11 à posição 14, segundo outra trajetória retilínea. Nesses testes foram consideradas 208 mudanças de posição com intervalos de tempo variando de 0,1 s a 0,3 s para cada mudança, com os resultados mostrados nas Figuras 7 a 9. A Figura 10 mostra instantâneos da posição do manipulador em intervalos de 3 segundos. Nas Figuras

7 a 9 pode ser observada a melhor aproximação às trajetórias desejadas com o aumento do tempo para cada posição intermediária, porém isso é obtido à custa de um maior tempo de trajeto, que nesses casos variou de 21 a 62 segundos.

A partir desse ponto foi iniciada uma pesquisa para se encontrar uma distribuição de intervalos de tempo que pudesse oferecer um melhor desempenho na relação *Erro de Trajeto vs. Tempo de Percurso*. Uma solução foi baseada em aproximações parabólicas que fazem com que os intervalos de tempo iniciais (fase de aceleração) e finais (desaceleração) sejam maiores. Foram utilizadas duas funções, uma para cada trecho do trajeto, como mostra a Figura 11. Os resultados estão na Figura 12. Observam-se erros de trajetória aproximadamente iguais àqueles da Figura 9 ($\Delta t = 0,3$ s.), tendo sido consumido um tempo total de percurso de 43 segundos contra 67 segundos daquele com intervalo de tempo constante.

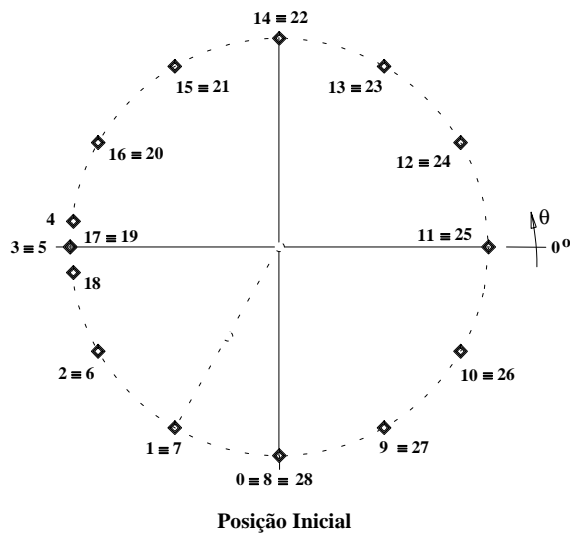


Figura 5. Posições do treinamento

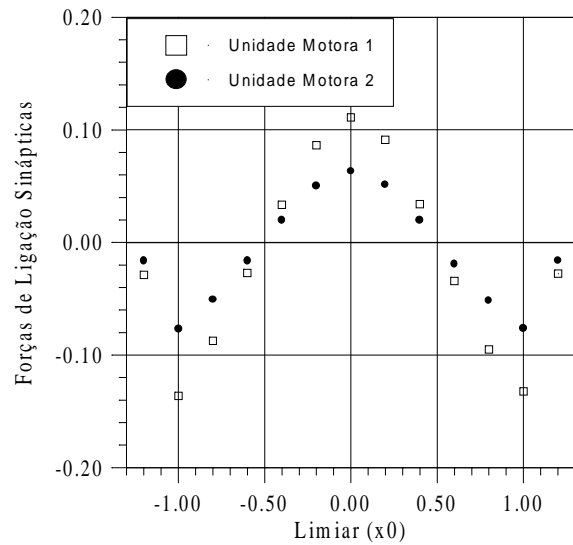


Figura 6. Forças de ligação sinápticas

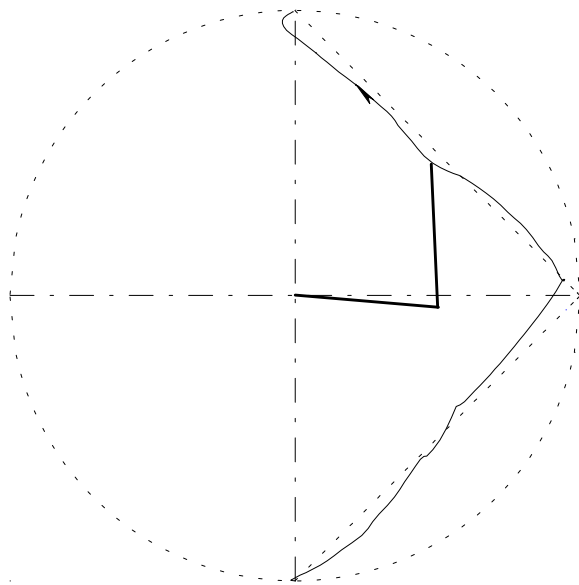


Figura 7. Trajetória da extremidade do manipulador (com $\Delta t = 0,1$ s)

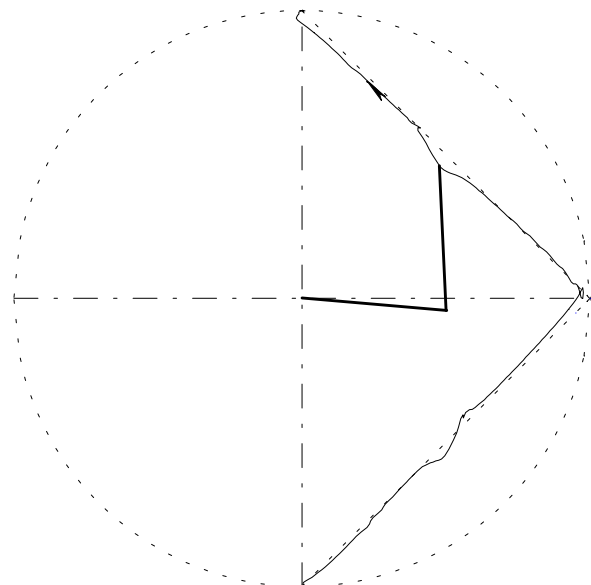


Figura 8. Trajetória da extremidade do manipulador ($\Delta t = 0,2$ s)

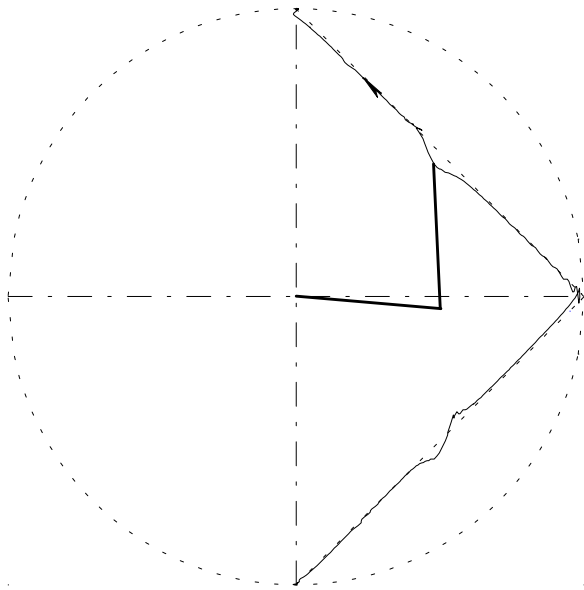


Figura 9. Trajetória da extremidade do manipulador ($\Delta t = 0,3$ s)

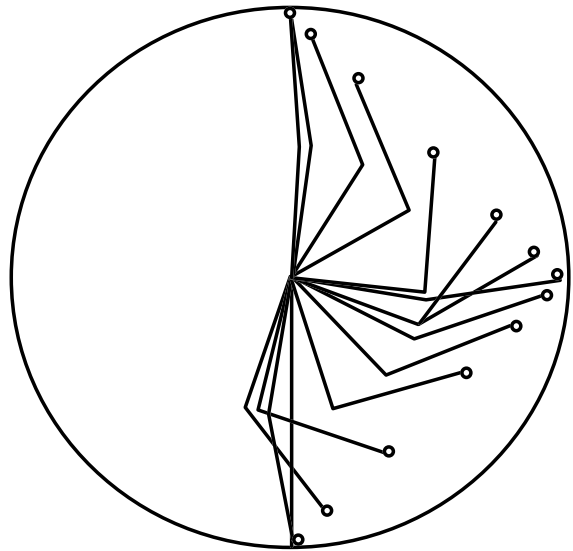


Figura 10. Instantâneos a cada 3 s

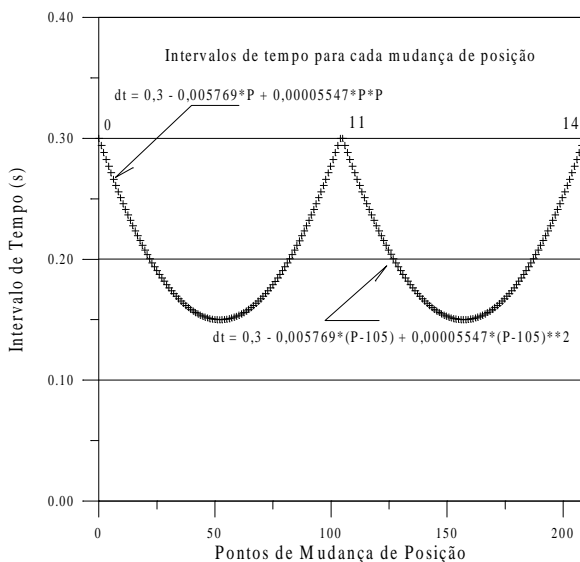


Figura 11. Funções para Δt variável

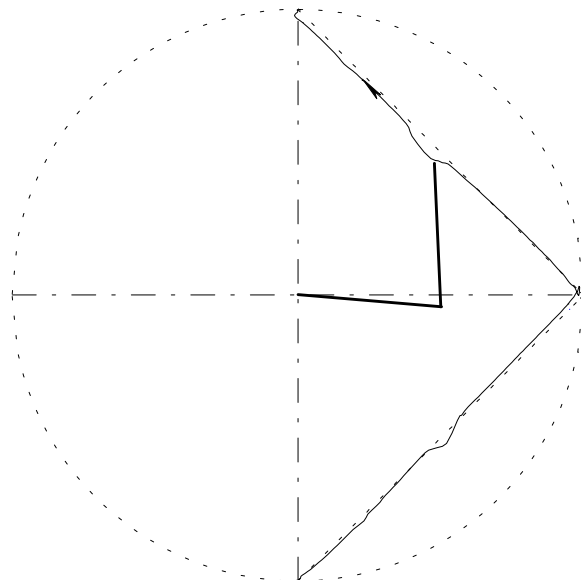


Figura 12. Trajetória com Δt variável

Observa-se que neste trabalho foi utilizada uma rede com um número menor de sinapses do que na rede apresentada em trabalhos anteriores, mantendo ainda uma precisão de posicionamento razoável exceto em algumas posições nas quais o manipulador forma ângulos retos.

5. CONCLUSÕES

Os testes demonstraram que o controlador de posição baseado nas redes MULSY tem capacidade para controle de trajetória desde que sejam gerados comandos de posição desejada em intervalos de tempo adequados. Os tempos de percurso foram relativamente grandes porém, os resultados do teste mostraram que pode haver uma otimização na geração de comandos, sugerindo a utilização de uma nova camada (em nível hierarquicamente superior) constituída por uma ou mais unidades, que sejam treinadas para gerar funções de trajetória e para

coordenar a mudança dos sinais de posição. O desempenho do processo de treinamento e a capacidade de generalização continuam sendo características de destaque desse novo conceito de redes neurais.

6. REFERÊNCIAS

- Baptista F., B.D., Cabral, E.L.L., 1999a, “Planar Two-Link Manipulator Control with Multiple-Synapse Neural Network Controller”, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM'99)*, pp. 647-652, September 19-23, Atlanta, USA.
- Baptista F., B.D., Cabral, E.L.L., 1999b, “Controle de um Manipulador Bi-articulado com uma Rede Neural de Contatos Multi-Sinápticos”, *Anais do 4º Simpósio Brasileiro de Automação Inteligente (4º SBAI)*, pp. 119-124, São Paulo, 08-10 de setembro de 1999.
- Baptista F., B.D., Cabral, E.L.L., Soares, A.J., 1998, “A New Approach to Artificial Neural Networks,” *IEEE Transactions on Neural Networks*, Vol. 9, No. 6, pp. 1167-1179.
- Baptista F., B.D., 1998, “*Redes Neurais para Controle de Sistemas de Reatores*,” São Paulo: Tese de Doutorado – USP/ IPEN-CNEN/SP.
- Kandel, E. R.; Schwartz, J. H.; Jessel, T.M. (Eds.), 1991, *Principles of Neural Science*. Prentice-Hall International Inc., Third Edition.