

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

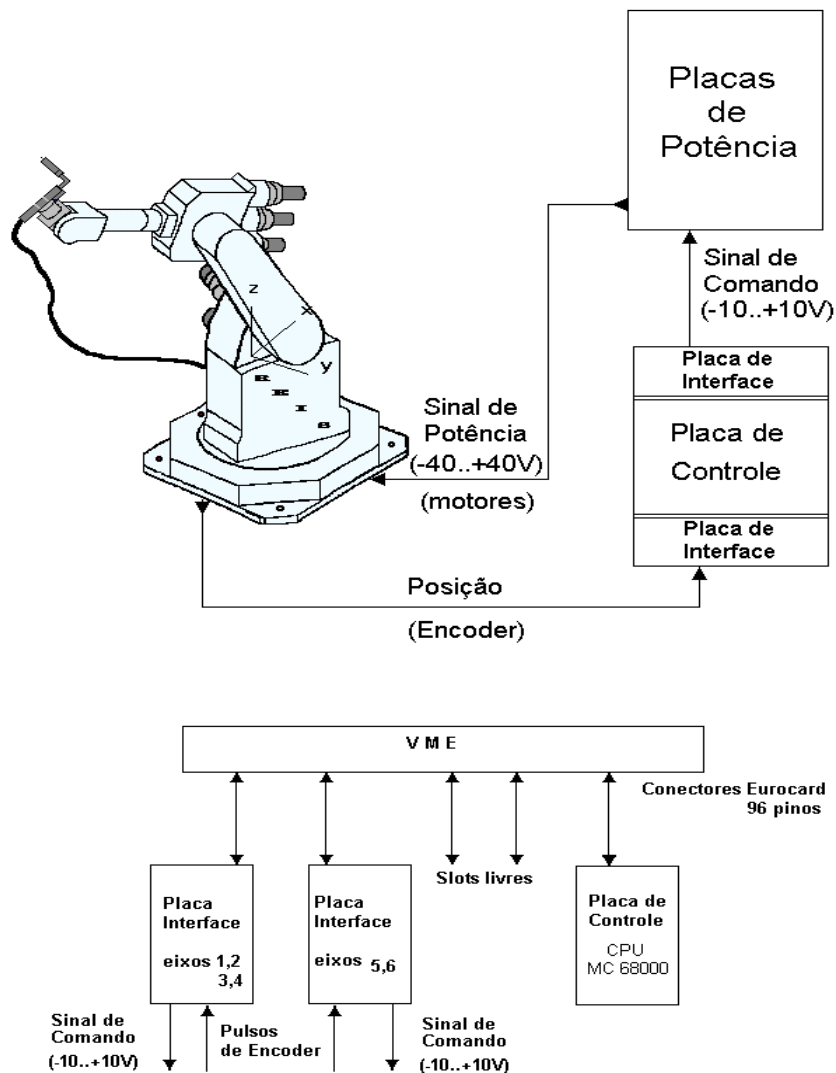


Figura 1. Sistema original de controle do robô Reis Rv-15

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.

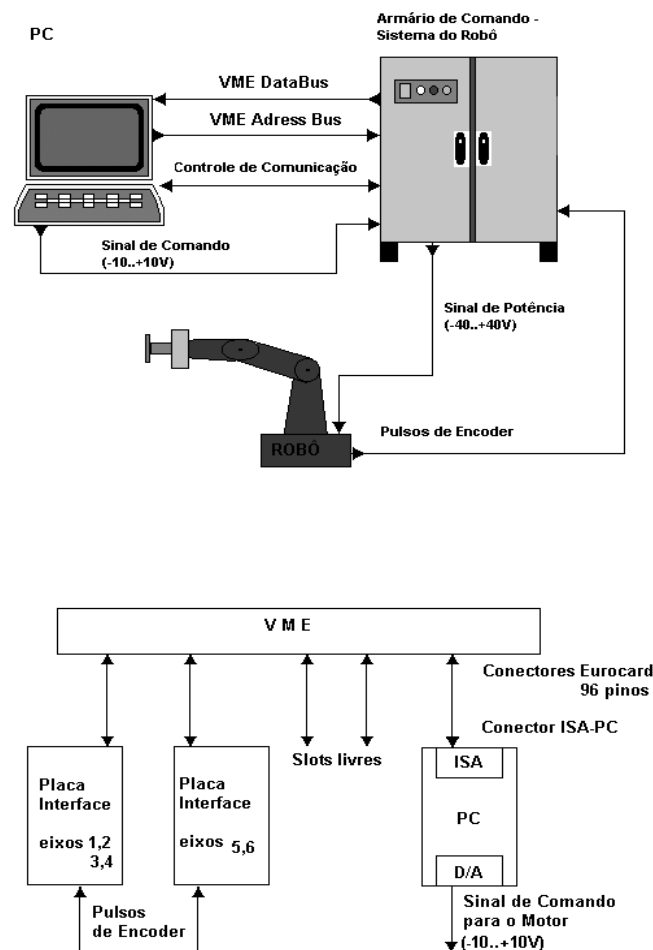


Figura 2. Sistema modificado, com a conexão do PC ao barramento VME (e retirada da placa de controle original da Reis)

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.

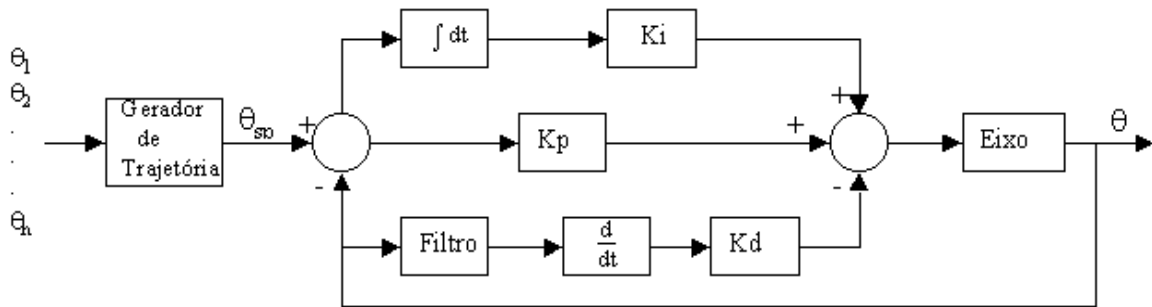


Figura 3. Esquema de controle implementado

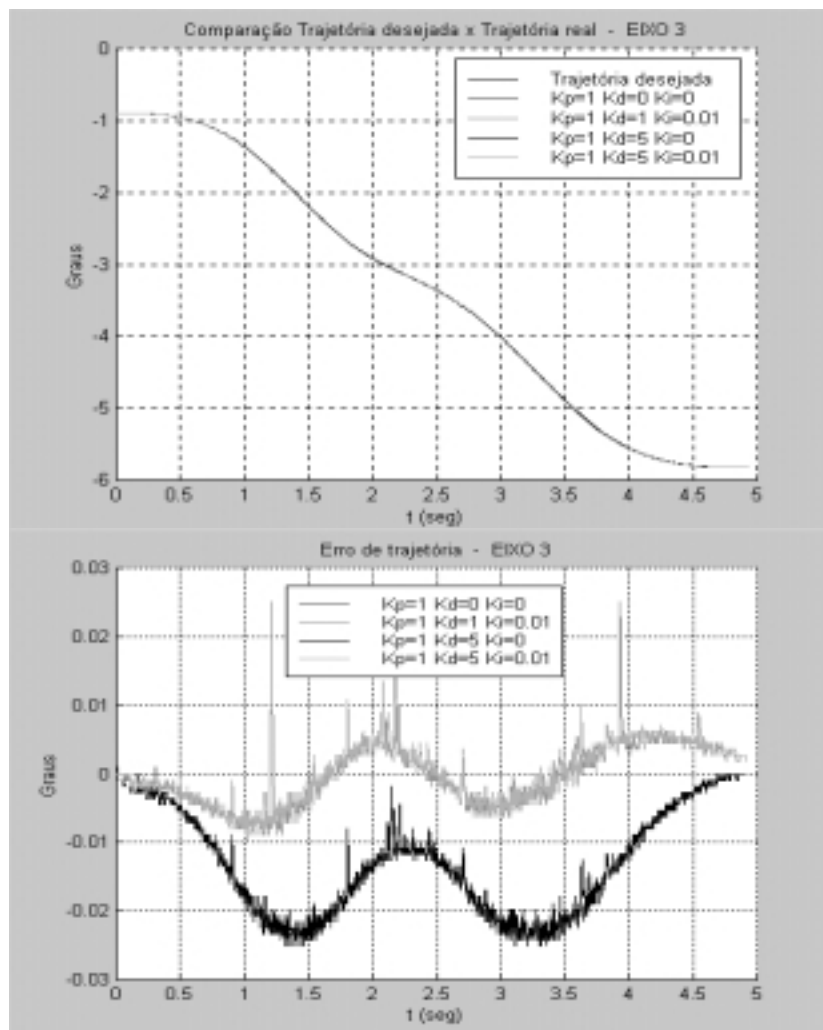


Figura 4. Comparação trajetória real x teórica e erro de trajetória para o eixo 3

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.