

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.

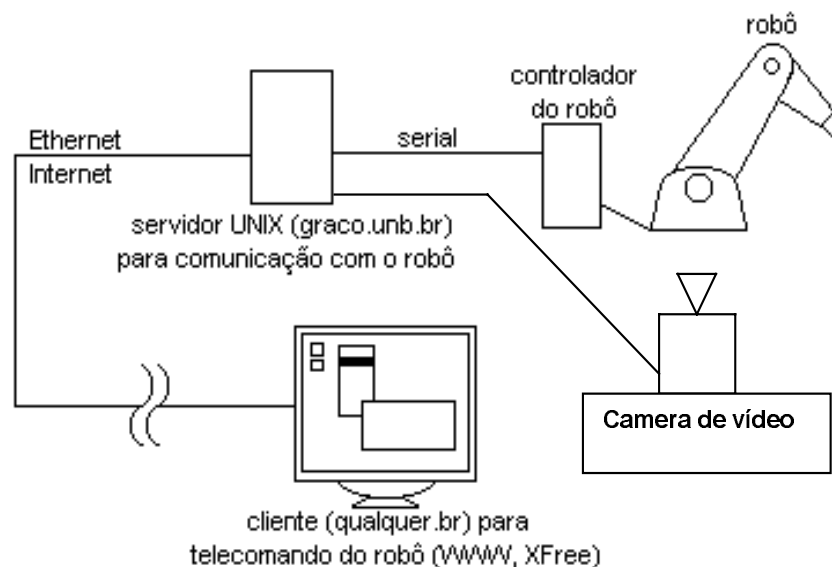


Figura 1. Arquitetura do sistema de teleoperação via Internet: WebCam e WebRobot.

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).

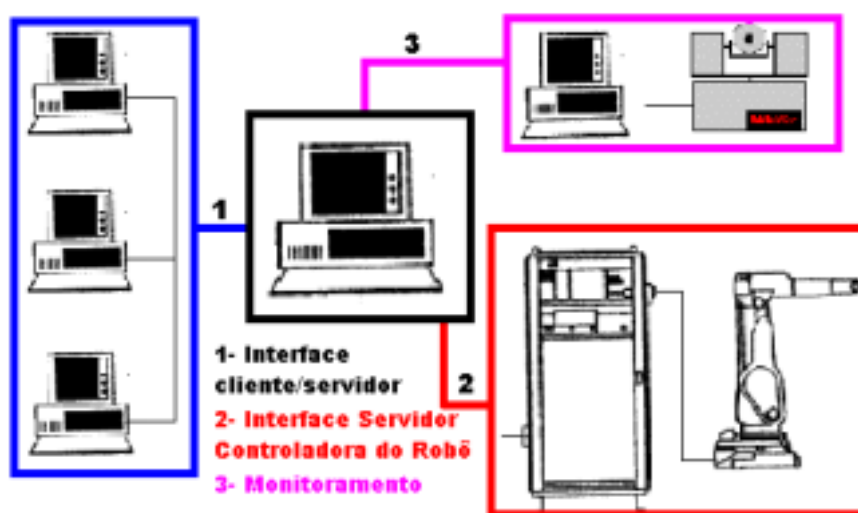


Figura 2. Subsistemas do sistema telerobótico RobWebLink (*Robot Web Link*).

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.

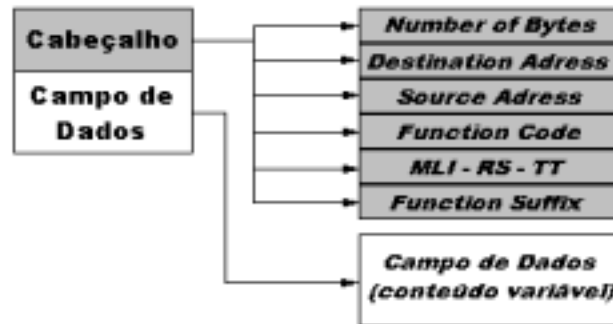


Figura 3. Estrutura do telegrama definido pela biblioteca ARAP.

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.

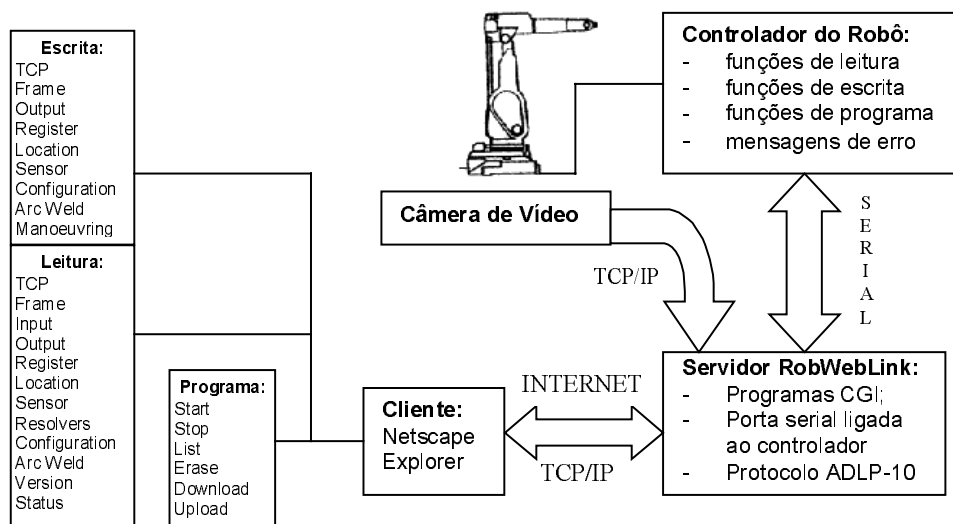


Figura 4. Arquitetura, fluxo de dados e funções disponibilizadas pelo RobWebLink.

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.

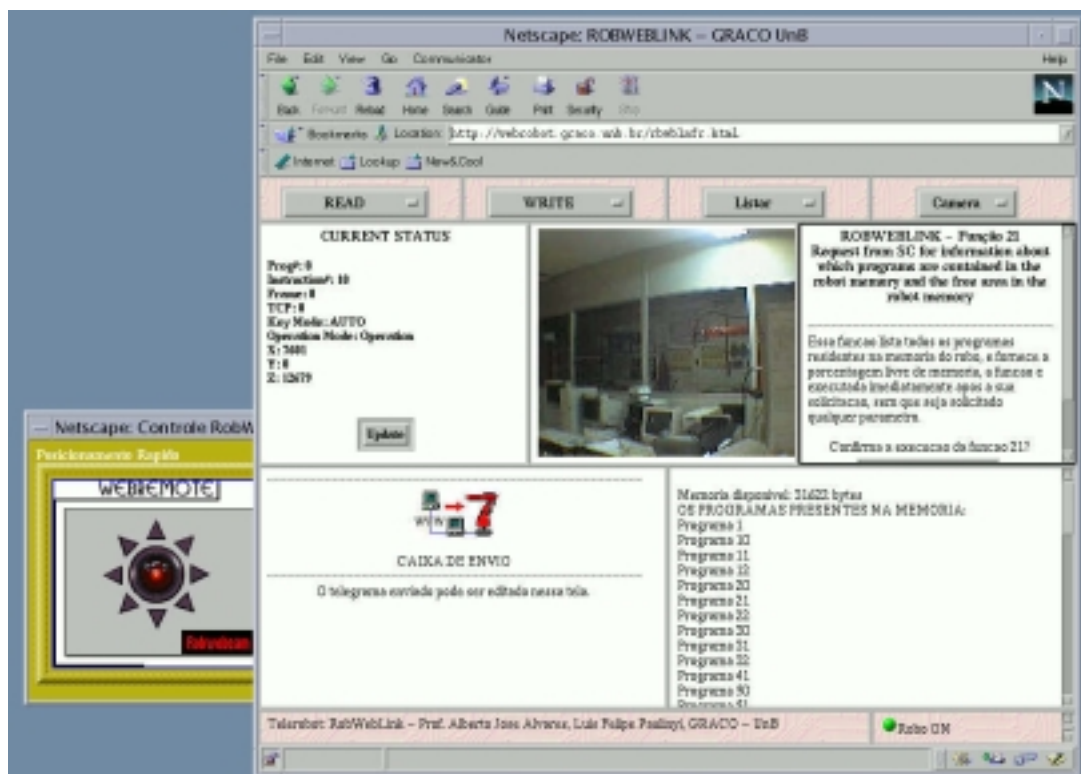


Figura 5. Sistema RobWebLink: interface com o usuário (<http://webrobot.graco.unb.br>).

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.