

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.

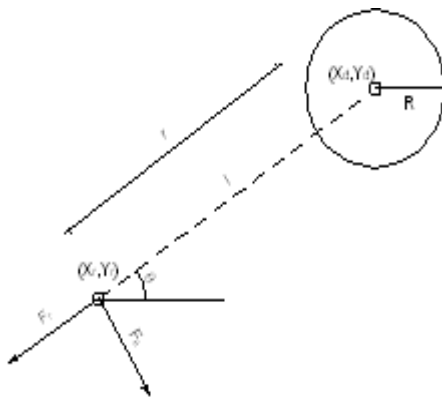


Figura 1. Modelo da força repulsiva.

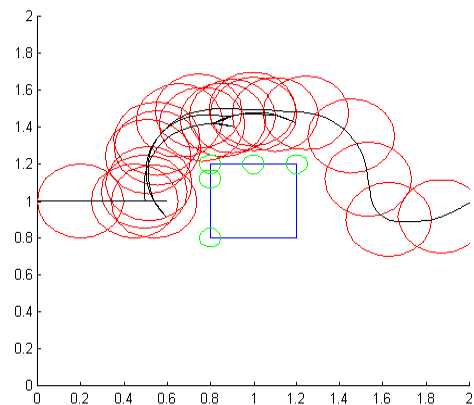


Figura 2. Trajetória obtida pelo algoritmo de navegação.

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).

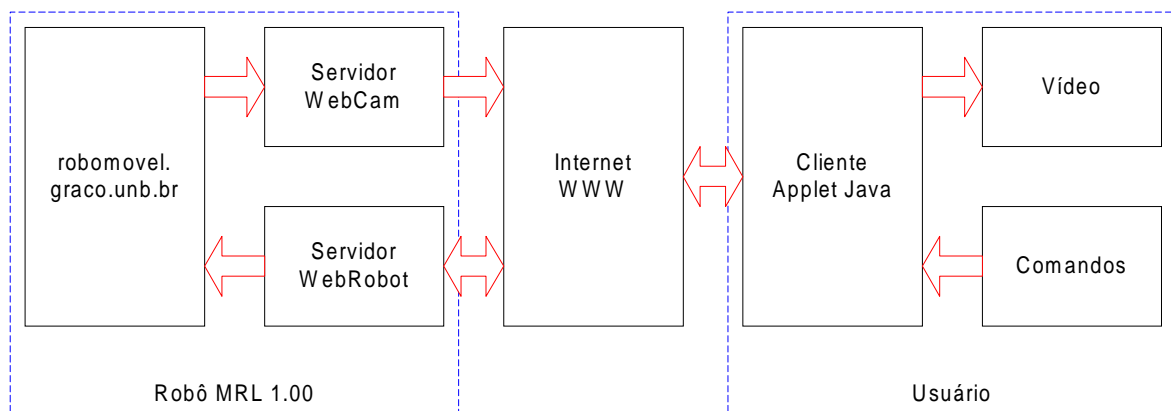


Figura 3. Arquitetura do sistema de teleoperação.

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).

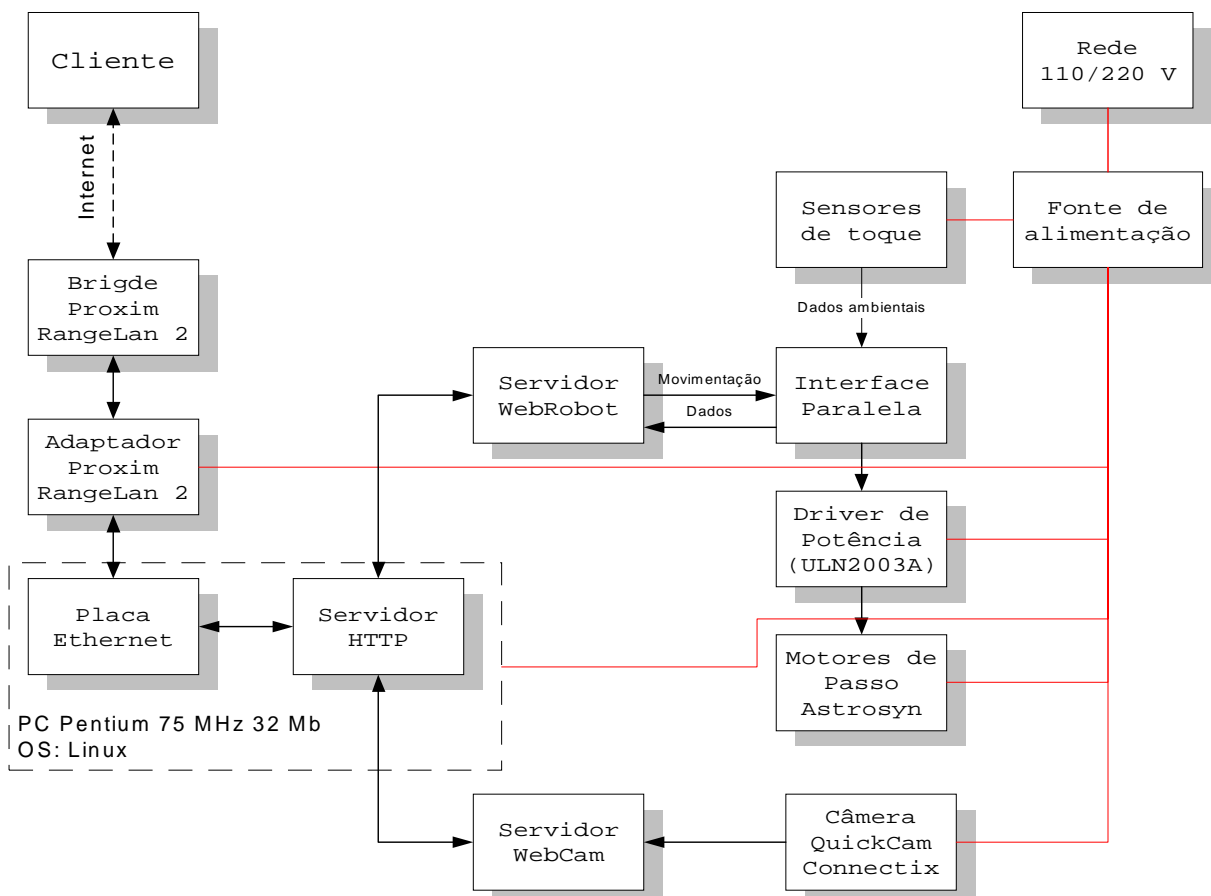


Figura 4. Módulos integrantes do sistema.

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.

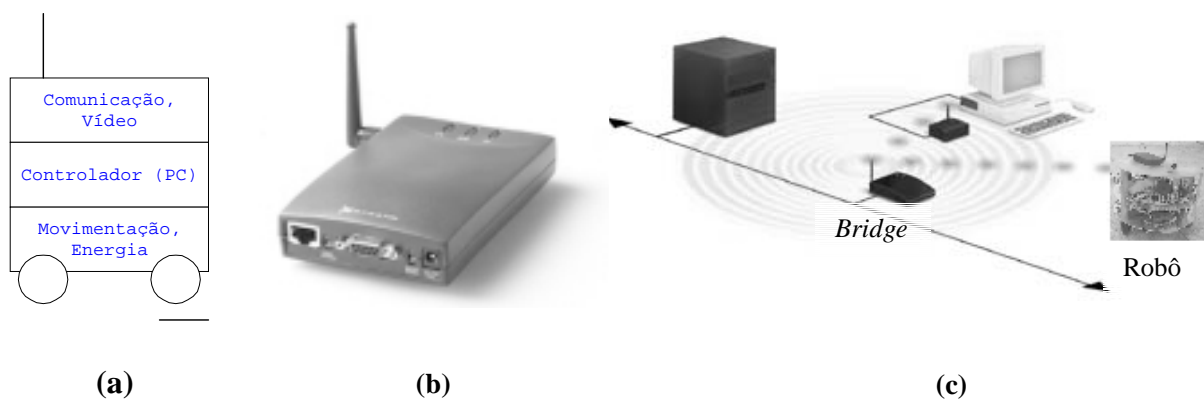


Figura 5. (a)Arquitetura física do robô desenvolvido ; (b) Adaptador Ethernet utilizado no robô móvel (RangeLan 2 7921, Proxim). (c) Arquitetura de comunicação utilizada no robô móvel.

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.

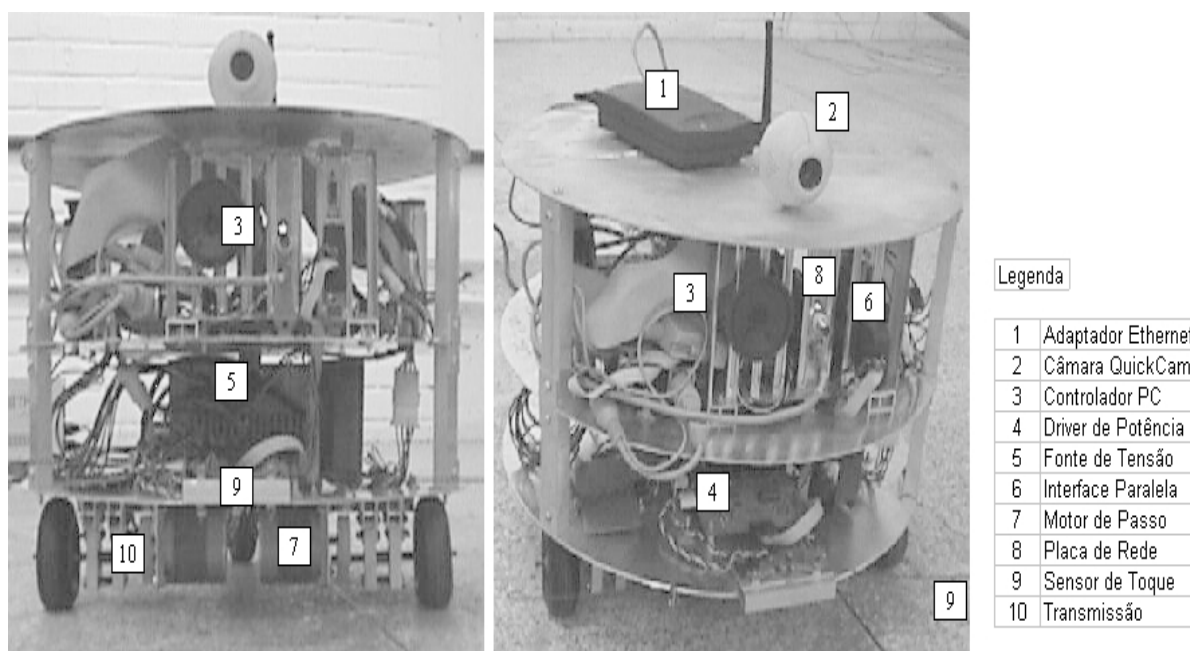


Figura 6. Robô móvel desenvolvido no GRACO (MRL 1.00).

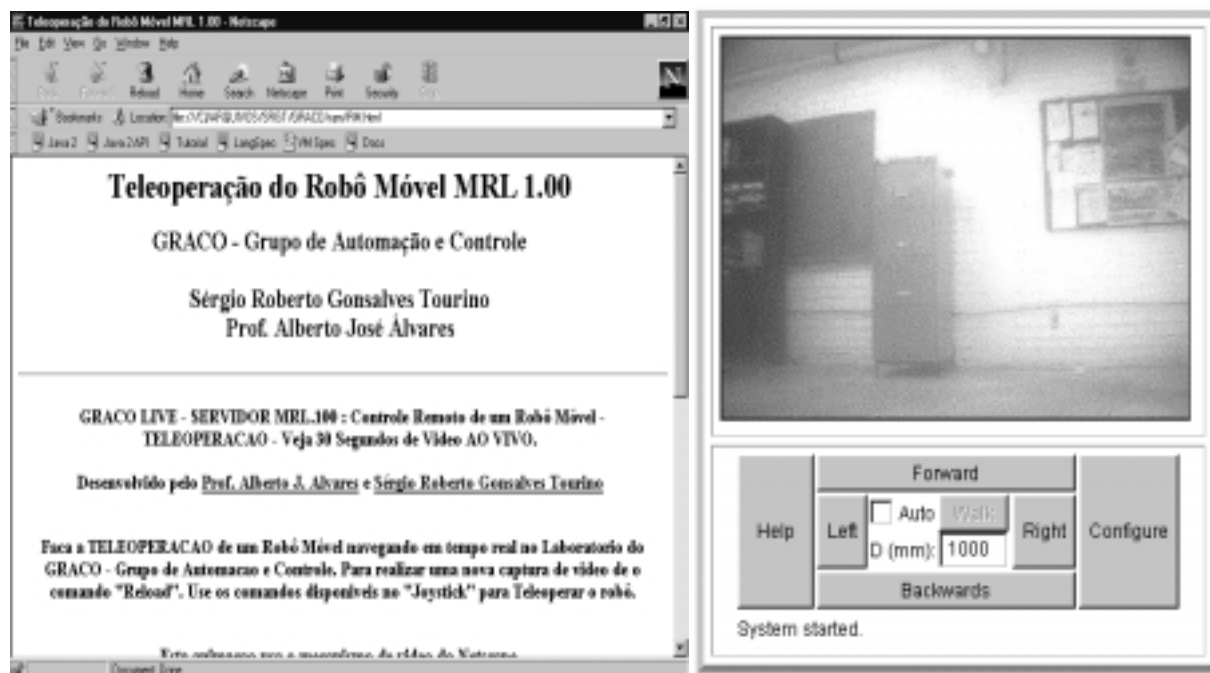
3.2 GUI (Graphical User Interface) Para Teleoperação

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.