
NAVMAP: UM SISTEMA PARA NAVEGAÇÃO POR MAPEAMENTO DO ROBÔ MÓVEL NOMAD XR4000

Paulo R. C. Dutra¹, Márcio M. de Sousa², Gabriel F. Andriolli³,
Alberto J. Álvares⁴ e João C. E. Ferreira⁵

^{1,2,4}Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica,
Grupo de Automação e Controle (GRACO), CEP 70910-900, Brasília, DF

e-mail: ¹prcdutra@hotmail.com, ²marcio_ms79@hotmail.com, ⁴alvares@AlvaresTech.com

^{3,5}Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica,
GRIMA-GRUCON, Caixa Postal 476, CEP 88040-900, Florianópolis, SC

e-mail: ³gabriel@das.ufsc.br, ⁵jcf@grucon.ufsc.br

Resumo O desenvolvimento de robôs autônomos teleoperados aconteceu devido à existência de ambientes impróprios à ação humana. Estes robôs, por sua vez, necessitam normalmente de uma representação do ambiente (um mapa) e a habilidade de interpretar esta representação para que sejam capazes de gerar trajetórias e de se movimentar com segurança em ambientes sujeitos a variações de posicionamento de objetos. Este trabalho descreve um sistema de navegação com teleoperação através da Internet para o robô Nomad XR4000. O sistema constrói mapas bidimensionais do ambiente que circunda o robô através da fusão de sensores ultrasônicos, infravermelhos e na estimativa da posição do robô. A navegação é realizada utilizando-se os mapas ambientais armazenados. O usuário, através de uma interface gráfica acessada localmente ou via WEB, pode definir um objetivo a ser alcançado pelo robô. Um módulo de planejamento da trajetória baseado em inteligência artificial define um caminho e controla o movimento do robô até o objetivo sem que ocorram colisões.

Palavras-chave: Sistema de navegação, robô móvel teleoperação.

Abstract: *The development of teleoperated autonomous robots occurred due the existence of improper environments to the human action. These robots, in turn, usually need a representation of the environment (a map) and the ability to interpret that representation to be able to plan a path towards some goal location and to move safely in an environment where there may be variations in the position of the objects. This work describes a navigation system with remote monitoring through the Internet to the Nomad XR4000 mobile robot. The system builds bidimensional maps of the environment that surrounds the robot through the combination of sonar and infra-red sensors, and also the position estimate of the robot. The navigation will be carried through stored environment maps. The user, through a graphical interface in a Web browser, can define a goal to be reached by the robot. A path planner based in artificial intelligence defines a trajectory and controls the robot motion without collisions.*

Keywords: *Navigation system, mobile robot, teleoperation*

1 INTRODUÇÃO

A grande variedade de tarefas que podem ser realizadas por robôs móveis autônomos tem impulsionado o crescimento do interesse por esse campo do conhecimento nos últimos anos. Atualmente, os robôs móveis estão inseridos em aplicações que vão desde as associadas à substituição do homem em ambientes hostis até a condução de pessoas em museus (Thrun et al. 1999).

Para um robô móvel autônomo, o problema de controle se traduz no conceito de navegação. Segundo Nehmzow (2000), a navegação pode ser definida por três competências fundamentais:

- Autolocalização
- Planejamento de trajetória
- Construção e interpretação de mapas

Autolocalização denota a capacidade de determinar/estabelecer sua própria posição em relação a um sistema de referência. Planejamento de trajetória requer a determinação da posição de origem e da posição de destino em relação a um mesmo sistema de referência. Essas duas competências requerem, normalmente, o uso de uma representação do ambiente (mapa) e a habilidade de interpretar essa representação.

De uma forma geral, as representações do ambiente são divididas em mapas geométricos, topológicos e por decomposição em células (Júnior, 2002), sendo os dois últimos os mais comumente utilizados.

Os mapas topológicos consideram o ambiente como uma série de lugares e conexões entre esses lugares, sendo naturalmente

representados por um grafo. Uma das principais vantagens dessa abordagem é a facilidade de planejamento da trajetória.

Os mapas por decomposição em células, chamados de grades de ocupação (Martin, 1996 e Moravec, 1985), representam o ambiente mediante uma discretização em células, formando uma grade onde cada célula representa uma região quadrada do ambiente e armazena um valor que indica a probabilidade de ocupação desta área. O preenchimento de uma célula através de tratamento probabilístico tem o problema do custo computacional, que cresce muito quando o ambiente a ser mapeado é grande. Dessa forma, algumas aplicações em tempo real se tornam inviáveis (Dissanayake, 2001). A facilidade de representação do ambiente e a possibilidade de integração de leituras de diferentes sensores constituem as principais vantagens desses mapas.

O problema fundamental da construção e interpretação dos mapas é conhecer a localização do robô no ambiente. O método mais comum se faz através do sistema de odometria do robô (*dead reckoning*), no qual a posição e a orientação são estimadas somando-se cada movimento em relação a um ponto inicial. Em função do problema do erro odométrico, que são acumulados a todo tempo, é mais comum encontrar sistemas que acrescentam também informação sensorial, uma vez que este erro não pode ser corrigido sem uma percepção externa.

Outro problema que tem chamado atenção no campo da robótica é o monitoramento remoto de robôs. A existência de ambientes onde a ação humana não é viável levou ao desenvolvimento de robôs teleoperados. Em Byrd (1996), um robô móvel operado remotamente é utilizado para visualizar e monitorar lugares perigosos. Nehmzow et al. (1996) e Simmons (1998) utilizam a Internet como meio de comunicação para controlar remotamente robôs móveis.

Dentro desse contexto, este trabalho apresenta aspectos da metodologia e da implementação de um sistema para a navegação de um robô móvel. O sistema desenvolvido permite ainda o monitoramento remoto através da Internet.

Na seção 2 é apresentado o robô móvel utilizado nesse trabalho. A seção 3 detalha o sistema de navegação implementado apresentando alguns resultados simulados. A arquitetura voltada ao monitoramento remoto é descrita na seção 4. Na seção 5 são apresentadas os resultados experimentais. As conclusões são reportadas no capítulo 6.

2 O ROBO MÓVEL NOMAD XR4000

O robô móvel Nomad XR4000 (fig 1) é um sistema integrado destinado para aplicações industriais e de pesquisa. Entre suas características, destacam-se:

- Sistema de movimentação formado por 8 motores e 4 rodas;
- Sensores infravermelhos, ultrasônicos e de colisão;
- Dois PCs Pentium com sistema operacional Linux (programação em linguagem C) interconectados utilizando o protocolo TCP/IP;
- Câmera acoplada a um manipulador *pan-tilt*;
- Sistema de comunicação via rádio (*wireless*);
- Sistema de estimativa da posição e orientação (odômetro);



Figura 1 – Robô Móvel Nomad XR4000

O sistema motor do robô XR4000 baseia-se num sistema holonômico, provendo três graus de liberdade (X, Y e θ): dois de translação e um de rotação. Suas rodas possuem eixos de translação e rotação independentes, somando oito motores para as quatro rodas do sistema. Três DSP (*Digital Signal Processing*) e um microcontrolador dedicado realizam o controle dos oito eixos e o cálculo da estimativa da posição (*dead reckoning*), que é baseado no modelo cinemático do robô e em dados provenientes de encoders localizados em cada atuador (motores e rodas).

O sistema de sensores ultrasônicos, sistema *Sensus 250*, consiste em dois anéis contendo 24 sensores, provendo informações sobre a distância de objetos entre 150 e 7000 mm. O sistema *Sensus 350* consiste em dois anéis de 24 sensores por luz infravermelha capazes de prover informações de presença de objetos próximos (tipicamente entre 300 e 500 mm de distância).

O Nomad XR4000 dispõe ainda de sensores de colisão que retornam informações sobre contato físico com objetos no ambiente. Neste trabalho, estes sensores foram utilizados somente para casos de emergência, uma vez em que não é desejável que o robô sofra colisões.

A programação do Nomad XR4000 é feita usando a arquitetura XRDev fornecida pelo fabricante. Esta arquitetura multiprocessada é composta principalmente de três processos:

- Nrobot, o processo servidor que comunica-se com o hardware do robô.
- Ngui, um processo que oferece uma interface gráfica através da qual o usuário pode enviar comandos e receber dados de qualquer processo do robô.
- Processo do usuário, que comunica-se com o Nrobot para poder realizar várias tarefas definidas pelo usuário.

A conexão com o robô é feita através da interface *Proxim RangeLan 2* (<http://www.proxim.com>) e um rádio Ethernet. Esse sistema fornece acesso à rede local utilizando o protocolo TCP/IP.

3 SISTEMA DE NAVEGAÇÃO

A figura 2 mostra a arquitetura do sistema de navegação por mapeamento estático e dinâmico desenvolvido. Nesse item serão detalhados os aspectos dessa arquitetura e da sua implementação.

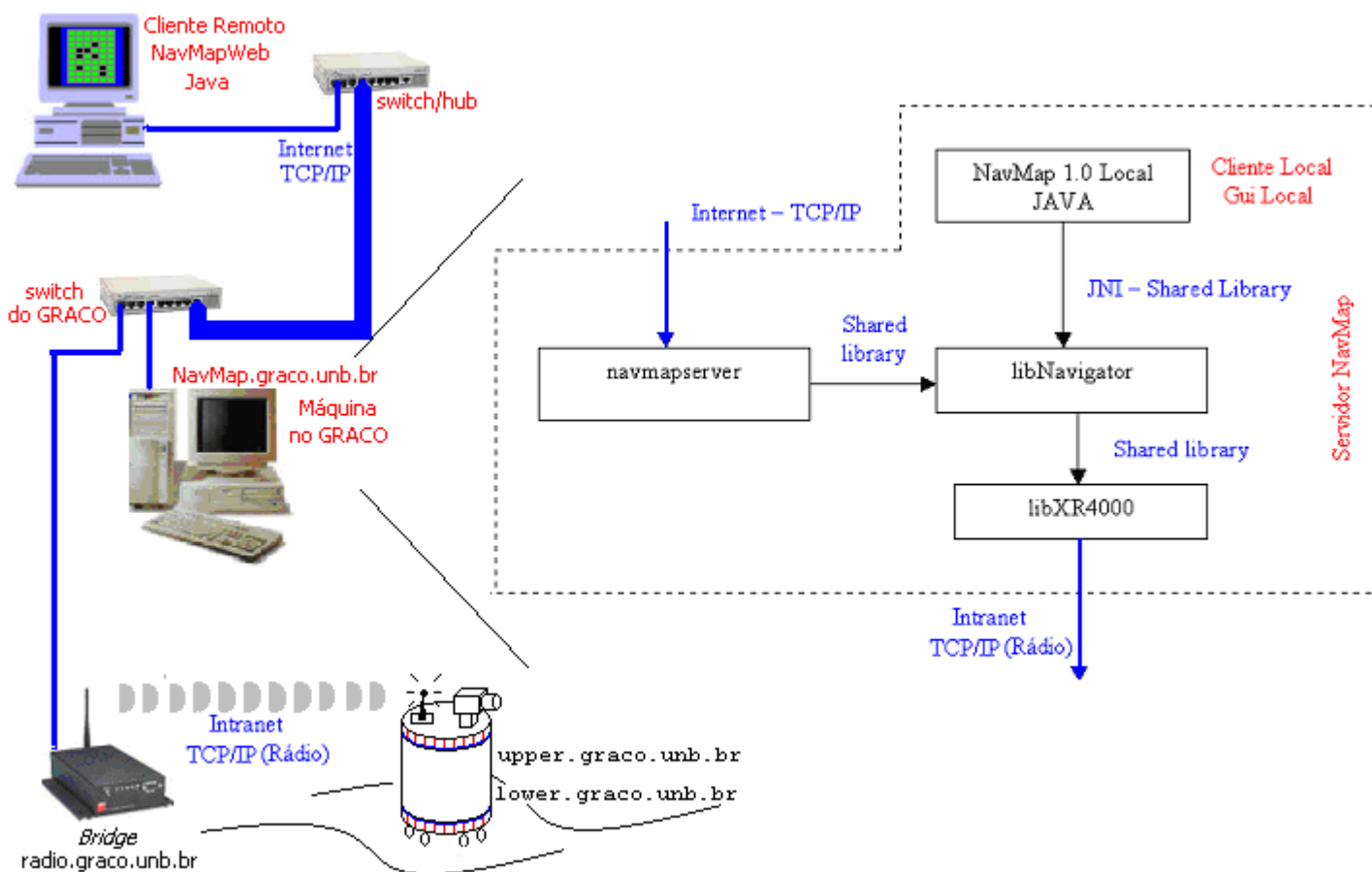


Figura 2 – Arquitetura do sistema de navegação teleoperado através da Internet.

3.1 Módulo de Mapeamento e Autolocalização

A abordagem utilizada para representar o ambiente que circunda o robô foi a de grades de ocupação. Essa representação foi utilizada principalmente por fornecer uma maneira simplificada de integrar diferentes percepções sensoriais para representar espaço no qual o robô está inserido.

Nessa abordagem, o ambiente é dividido em células, sendo que cada uma delas representa uma região quadrada do mundo real. Neste trabalho o estado de ocupação da célula é indicado através de um valor inteiro. Os valores utilizados e armazenados nas células e os seus respectivos significados são:

- (-1): Região não-explorada do espaço. No início do mapeamento todas as células do mapa são inicializadas com o valor -1;
- (0): Região livre do espaço;
- (8): Tolerância: Essa informação é utilizada para evitar colisão do robô com obstáculos devido à diferença entre o diâmetro do robô e a dimensão da célula.
- (9): Região ocupada do espaço: Identifica os obstáculos presentes no ambiente.

O tamanho da célula da grade de ocupação deve ser escolhido levando-se em conta aspectos como custo computacional, precisão e limitação do mapeamento do ambiente real. No sistema desenvolvido, a dimensão definida para cada célula

foi de 100 x 100 mm. A estrutura de armazenamento dos mapas é um vetor unidimensional com 90000 posições equivalente a uma matriz quadrada de ordem 300. Assim, um mapa pode representar uma região com dimensões máximas de 30 x 30 m ou 900 m².

As células são preenchidas a partir do processamento sensorial que consiste em integrar as leituras dos sensores ultrassônicos e infravermelhos com a posição estimada do robô no ambiente. Uma margem de segurança foi adicionada em torno dos obstáculos percebidos pelos sensores como medida de prevenção de colisões.

A posição e a orientação do robô são obtidas do sistema odométrico. O problema desse método é que qualquer erro na estimativa da posição do robô poderá afetar tanto a construção do mapa quanto a interpretação do mesmo.

A figura 3 mostra em detalhes a interface gráfica desenvolvida na linguagem de programação Java (Sun Microsystems, 2003a), chamada NavMapWeb, onde se pode visualizar o mapa do ambiente e a posição do robô sendo ambos constantemente atualizados. No mapa são representados os espaços vazios (branco), os obstáculos (preto), as zonas de segurança (verde) e os espaços ainda não explorados (cinza), além do robô na sua respectiva posição. Essa interface possibilita ainda salvar ou carregar um mapa obtido, ajustar a resolução das grades e a visualização do estado das baterias.

Observa-se na figura 2 que a interface NavMapWeb é executada remotamente e a comunicação com as outras partes do sistema se dá via Internet. Esta interface também pode executar localmente e, nesse caso, é referenciada por NavMap Local.

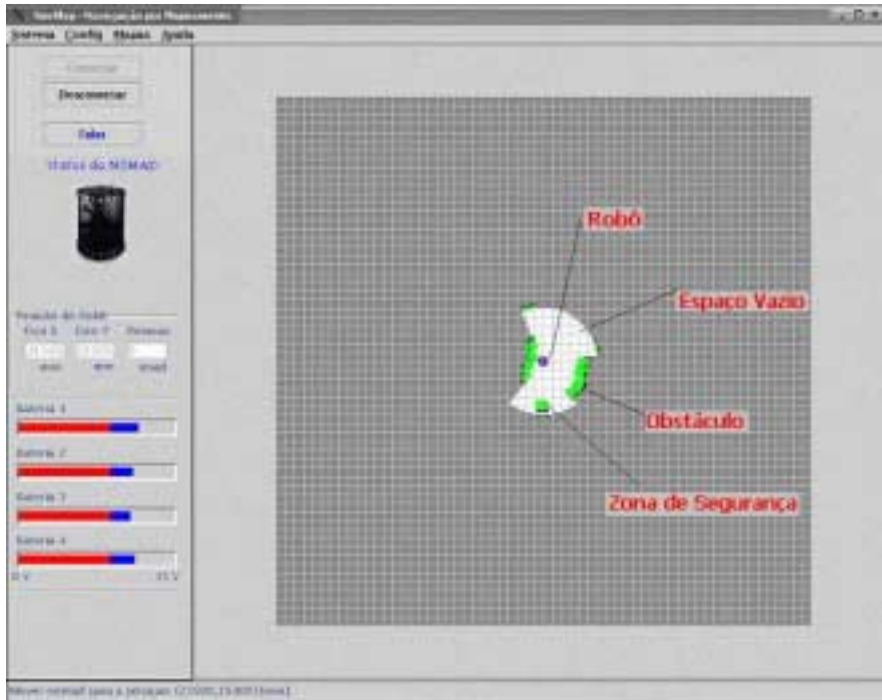


Figura 3 – Interface gráfica NavMapWeb com a aplicação em funcionamento

3.2 Módulo de Planejamento de Trajetória

O planejamento de trajetória (*Path Planning*) requer a determinação da posição de origem e da posição de destino em relação a uma mesma referência. A partir delas e de um mapa, tem a função de determinar uma trajetória que ligue os pontos de origem e destino.

Uma vez que na maioria dos casos existem vários caminhos realizáveis que levariam o robô de um ponto inicial a um ponto final, pode-se definir algumas características desejáveis ao trajeto. Por exemplo, alguns trajetos podem ser mais curtos enquanto outros mais seguros. Essas características implicam diretamente na escolha do método de planejamento de trajetória.

A maior parte das abordagens de planejamento de trajetória abstraem o espaço de busca em um grafo de caminhos possíveis, como, por exemplo, as abordagens por diagrama de Voronoi e grafos de visibilidade (Latombe, 1991). Essa aproximação pode levar naturalmente ao aprendizado do ambiente, onde o robô pode passar por vários caminhos para mapear o ambiente.

Para selecionar o melhor caminho em um grafo de vários caminhos possíveis dispõe-se, também, de várias técnicas de inteligência artificial (IA) bem conhecidas, como o algoritmo de busca *branch and bound* (Winston, 1984).

No sistema desenvolvido utilizou-se o algoritmo de busca A* (Winston, 1984), que é um refinamento do algoritmo *branch and bound*. Nesse algoritmo, o espaço de busca é reduzido excluindo-se múltiplos caminhos para um determinado sub-nó e deixando somente o caminho de menor custo. A escolha do caminho de menor custo para expansão em cada estágio é melhorada adicionando-se ao custo atual um custo estimado para os caminhos restantes. Se o custo estimado for sempre menor que o custo atual, a busca A* produzirá uma solução ótima. Uma função típica utilizada na estimativa desse custo no planejamento de trajetória é a distância de uma linha reta

entre a posição representada pelo nodo atual e a posição objetivo.

A figura 4 mostra o resultado simulado do módulo de planejamento de trajetória utilizando-se um mapa imaginário. As células não ocupadas são as células em branco. Os obstáculos estão representados nas células em preto. A letra R indica a posição inicial do robô e M a meta a ser alcançada. O caminho encontrado pelo algoritmo A* pode ser observado pelas células marcadas com a cor cinza. Nessa simulação a solução encontrada já prevê uma margem de segurança como medida preventiva de colisões.

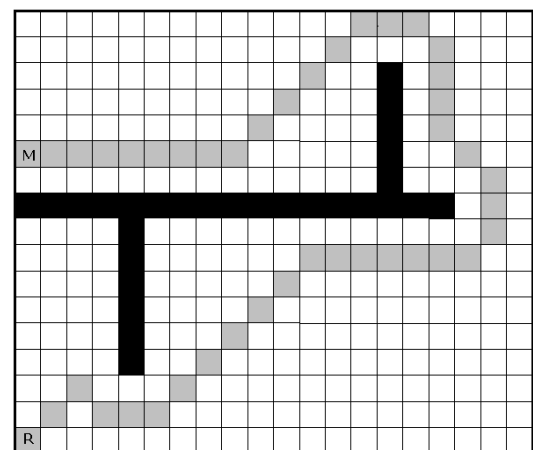


Figura 4 – Resultado da simulação do algoritmo A*

No sistema implementado, o módulo de planejamento de trajetória é acionado toda vez que um movimento é requisitado. O usuário pode especificar o destino a ser alcançado pelo robô, ou seja, as coordenadas de destino, com um simples toque do *mouse* sobre o mapa da interface gráfica (figura 3). Quanto esse evento ocorre, o módulo responsável pelo planejamento de trajetória verifica imediatamente a existência de um caminho sem colisões entre os pontos de origem e destino e, caso exista, o robô é comandado pelo sistema de navegação até atingir o objetivo. O controle de todo o movimento é feito autonomamente.

O sistema de navegação, à medida que conduz o robô até o objetivo, executa o sensoriamento de obstáculos não previstos no mapa. Caso seja detectado algum obstáculo muito próximo ao robô (distâncias iguais ou inferiores a 200 mm), uma nova trajetória é calculada de forma a desviar do obstáculo, evitando assim possíveis colisões.

4 TELEOPERAÇÃO ATRAVÉS DA INTERNET

A teleoperação através da Internet implementada no sistema é baseada na arquitetura cliente-servidor definida por Álvares (1998, 1999a, 1999b), que define as seguintes partes:

- O servidor, representado pelos programas localizados no robô;
- O cliente, representado por *applets* Java localizadas no *browser* do usuário.

Uma arquitetura similar foi utilizada em outro trabalho no laboratório, no qual Tourino (2002) faz a condução do robô Nomad XR4000 através da Internet.

No sistema implementado, o servidor desenvolvido, chamado NavMap, é executado no microcomputador Pentium II 600Mhz com sistema operacional Linux Suse 8.1, localizado no laboratório. Esse servidor se comunica com o servidor Nrobot do robô através da rede local *ethernet* utilizando o protocolo TCP/IP, como mostrado na figura 2.

O servidor NavMap foi implementado a partir de duas bibliotecas desenvolvidas na linguagem de programação C++: *libXR4000* e *libNavigator*. A biblioteca *libXR4000* faz um mapeamento das funções, variáveis e estruturas disponibilizadas na biblioteca do fabricante (*Nclient_host*) para um modelo orientado a objetos mais simples de ser manipulado dentro de outras aplicações. A *libNavigator* contém a implementação dos módulos de mapeamento e planejamento de trajetória. A comunicação entre a interface em Java e o sistema escrito em C++ utiliza a tecnologia JNI (*Java Native Interface*, Sun Microsystems, 2003b), que permite aplicações em Java de fazerem chamadas para bibliotecas nativas.

O cliente executa a aplicação, cuja interface gráfica é mostrada na figura 3, no *browser* (Netscape, MS Explorer, etc.) do usuário ou em uma GUI (*Graphical User Interface*) para ambiente Xfree Linux. Este pode estar localizado em qualquer lugar do mundo. São disponibilizadas também ao usuário as imagens de algumas câmeras existentes no laboratório e da câmera do próprio robô. A aplicação pode ser acessada no endereço <http://NavMap.graco.unb.br>. O código fonte, binários e documentação também estão disponíveis nesta URL.

A comunicação entre cliente e servidor é realizada através de canais de comunicação (*sockets*) utilizando o protocolo TCP. Essa conexão é realizada especificando-se o endereço IP da máquina remota servidora e a porta de comunicação a ser utilizada.

5 EXPERIMENTOS E RESULTADOS

Com o objetivo de observar e testar o sistema desenvolvido, realizou-se o mapeamento do laboratório do Graco (Grupo de Automação e Controle) utilizando-se a aplicação

desenvolvida. O robô foi inicialmente posicionado em um ponto fixo no qual as coordenadas em relação ao referencial global eram conhecidas. Essas coordenadas foram passadas ao sistema para localizar o robô no ambiente. O robô foi então guiado através do ambiente indicando-se a posição de destino e o mapa foi paralelamente sendo gerado.

O resultado pode ser observado na Figura 5 (a). As figuras 5 (b) e (c) representam, respectivamente, o mapa do Graco feito manualmente e a superposição dos mapas para fins de comparação.

Observou-se que algumas células acusaram obstáculos inexistentes (circuladas em vermelho na figura 5) e que esse erro ocorre com mais frequência nas células mais distantes do robô, indicando a necessidade de uma reavaliação do modelo adotado para os sensores e/ou de um melhor processamento entre os diversos sensores.

Em Lora (1998) é utilizado um filtro de Kalman para integrar a informação de uma bússola digital com a informação do sistema de odometria. Várias outras aplicações utilizando-se filtros de Kalman para integrar informação sensorial são citados em Lora (1998).

Apesar dos problemas observados, o mapa adquirido pode ser utilizado como uma primeira aproximação do ambiente. O controle dos movimentos foram feitos automaticamente pelo sistema desenvolvido e não houve nenhuma colisão enquanto a exploração do ambiente era realizada, mostrando a capacidade do robô em contornar obstáculos de maneira autônoma e eficiente.

Observou-se que o erro odométrico não afetou significativamente os resultados obtidos devido à baixa velocidade de deslocamento do robô (limitada em 300mm/s) e ao fato de as dimensões do ambiente explorado serem relativamente pequenas. Porém, futuras melhorias do sistema deverão incluir a implementação de algum meio de recalibragem do sistema de coordenadas do robô com base no sistema de referência do mapa do ambiente.

Em Yamauchi (1996), por exemplo, o sistema faz robô retornar para uma posição de calibração da odometria a cada 10 minutos. Em Thrun (1999) o robô calibra seu sistema de odometria com um erro menor do que 100 mm.

6 CONCLUSÃO

Foi avaliado neste trabalho a implementação de um sistema de navegação para o robô móvel Nomad XR4000. Para a representação do ambiente utilizou-se a abordagem de grades de ocupação devido à possibilidade de se integrar facilmente diferentes percepções sensoriais para representar espaço. Para a localização do robô utilizou-se o sistema de odometria (*dead reckoning*).

O sistema desenvolvido possui uma interface gráfica onde o usuário pode visualizar o mapa sendo construído à medida em que o robô se movimenta pelo ambiente. Ela permite, através de comandos de alto nível efetuados pelo usuário, o envio de coordenadas de destino ao sistema de navegação. A interface apresenta ainda o estado das baterias e as imagens colhidas pela câmera embarcada no robô.

O planejamento de trajetória é feito de maneira automática pelo sistema toda vez que uma movimentação é requisitada. Esta utiliza uma técnica de IA para traçar o caminho que o

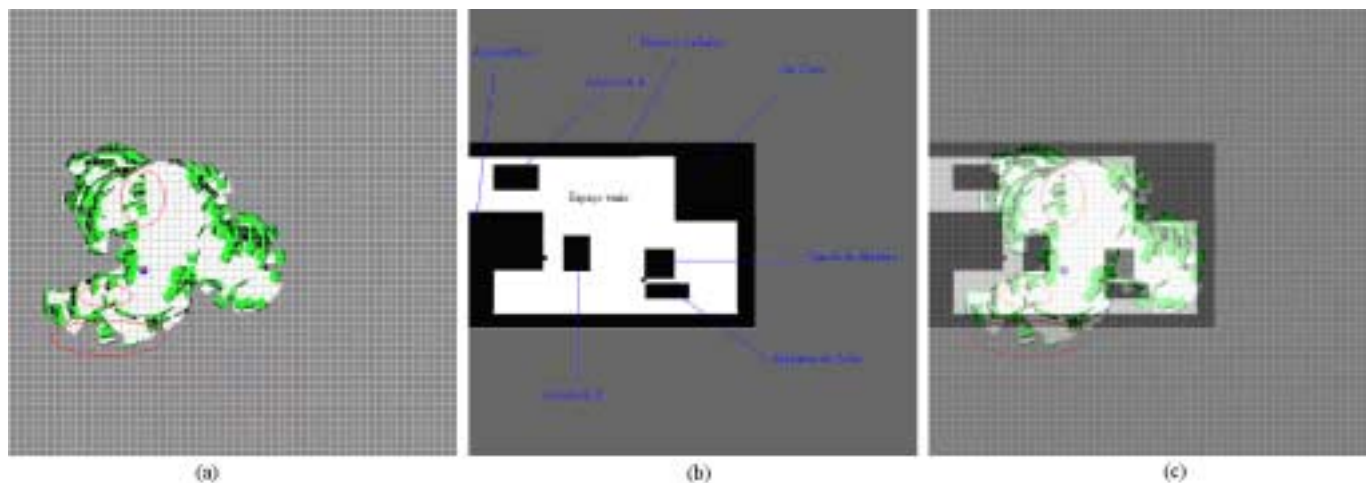


Figura 5 – (a) Mapa do Graco obtido com o sistema desenvolvido, (b) mapa do Graco feito manualmente e (c) superposição dos mapas em escala

robô deve seguir. O módulo de planejamento de trajetória mostrou-se capaz de guiar o robô através dos obstáculos do ambiente, contornando-os sem que ocorressem colisões.

O sistema apresenta ainda a característica da arquitetura cliente-servidor que permite o monitoramento remoto do robô através da Internet.

Os problemas relatados quanto ao mapeamento podem ser amenizados através de um melhor tratamento dos dados, como o uso de filtros de Kalman ou algoritmos genéticos. A questão da correção do erro odométrico é também necessária para se realizar a navegação de forma mais eficiente e para se elevar a autonomia do robô.

REFERÊNCIA BIBLIOGRÁFICA

Álvares, A. J. Romariz, L. S. 1998. "Desenvolvimento de um manipulador com dois graus de liberdade controlado remotamente via Internet". V CEM-NNE98. Fortaleza.

Álvares, A. J. 1999a. "Telerobótica: Metodologia Para o Desenvolvimento De Sistemas Robóticos Teleoperados Via Internet", Águas de Lindóia, SP, XV COBEM.

Álvares, A. J. 1999b. "Telerobotics: A Methodology for the Development of Through-the-Internet Teleoperated Robotic Systems" Telemanipulator and Telepresence Technologies V – SPIE Boston, USA.

Bilgiç, T. e Turksen, I.B., "Model-based Localization for an Autonomous Mobile Robot Equipped with SonarSensors", IEEE Conference on Systems, Man and Cybernetics, Oct. 22-25, Vancouver, B.C. Canada, 1995.

Byrd, J. S. "An Intelligent Inspection and Survey Robot", University of South Carolina. Columbia. USA. 1996.

Dissanayake, M. W. M. G. et al. "A solution to the simultaneous localization and map building problem", IEEE Transaction on Robotics and Automation, v.17, n.3, p.229-241, 2001.

Júnior, E. P. S. Navegação exploratória baseada em problemas de Valores de Contorno. 2002. Tese (Doutorado em Ciência da Computação) – Universidade Federal do

Rio Grande do Sul, Instituto de Informática, Porto Alegre, Brasil, 2002.

Latombe, J.C., "Robot Motion Planning". Kluwer Academic Publishers, 1991

Lora, F.A.S., Hemeryly, E.M. e Lages,W.F., "Sistema para Navegação e Guiagem de Robôs Móveis Autônomos", Controle & Automação, Vol. 9, No. 3, pp. 107-118, 1998.

Martin, M. C. Moravec, H. "Robot Evidence Grids", Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.

Moravec, H. P. Elfes, A. "High resolution maps from wide angle sonar", IN: IEEE International Conference on Robotics & Automation (ICRA), 1985.

Nehmzow, U. Bühlmeier, A. Dürer, H. "Remote Control of Mobile Robot via Internet", Manchester, 1996.

Nehmzow, U. , "Mobile Robotics: A Practical Introduction", Springer, London, 2000.

Simmons, R. "Xavier: An Autonomous Mobile Robot on the Web". Carnegie Mellon University. USA, 1998.

Sun Microsystems, 2003a, "The Java Tutorial", Disponível na Internet através do endereço: <http://java.sun.com/docs/books/tutorial/index.html>.

Sun Microsystems, 2003b, "Java Native Interface: Programmer's Guide and Specification".

Thrun, S., et alli, "MINERVA: A second generation mobile tour-guide robot". In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1999.

Tourino, S. R. G. Álvares, A. J. "Teleoperation Architecture for the Navigation through WEB of the Nomad XR4000 Mobile Robot", João Pessoa, PB, II CONEM, 2002.

Winston, P. H. "Artificial Intelligence", Addison-Wesley, 1984.

Yamauchi, B. "Mobile Robot Localization in Dynamic Enviroments Using Dead Reckoning and Evidence Grids". Journal of Robotics Systems, Special Issue on Mobile Robots, Vol. 14, No. 2, pp. 107-120, 1996.