

Sistema de Controle Híbrido para Robôs Móveis Autônomos

Farlei J. Heinen, Fernando S. Osório

Unisinos, Mestrado em Computação Aplicada
Av. Unisinos 950, São Leopoldo – RS - Brasil
farlei@exatas.unisinos.br
osorio@exatas.unisinos.br
<http://nccg.unisinos.br/robotica/simulador>

Resumo. Neste trabalho foi desenvolvido um sistema de controle robusto para robôs móveis autônomos que é capaz de operar e de se adaptar a diferentes ambientes e condições. Para isso foi proposta uma arquitetura de controle híbrida, integrando através de uma abordagem de três camadas (nível vital, funcional e deliberativo), as principais técnicas de controle robótico (controle deliberativo e controle reativo). Para garantir a robustez do sistema de controle, foi integrado um módulo localizador utilizando a técnica de localização Monte Carlo. Este módulo localizador possui um papel de destaque no sistema de controle, fornecendo uma base sólida para o controle e navegação do robô móvel autônomo. Para a validação do sistema de controle proposto, foi implementado um simulador de robôs móveis (SimRob3D) que permite a utilização de modelos de ambiente tridimensionais, bem como diversos modelos sensoriais e cinemáticos. Os resultados obtidos demonstraram que o sistema de controle é capaz de localizar o robô móvel localmente, globalmente, e de se relocalizar. Nas tarefas de navegação o robô é capaz de planejar e seguir uma trajetória em um ambiente dinâmico, contendo obstáculos imprevistos (estáticos e móveis).

1 Introdução

A robótica móvel [1] é uma área de pesquisa que lida com o controle de veículos autônomos ou semiautônomos. O que diferencia a robótica móvel de outras áreas de pesquisa em robótica tais como a robótica de manipuladores, é a sua ênfase nos problemas relacionados com a operação (locomção) em ambientes complexos de larga escala, que se modificam dinamicamente, compostos tanto de obstáculos estáticos como de obstáculos móveis. Para operar neste tipo de ambiente o robô deve ser capaz de adquirir e utilizar os conhecimentos sobre o ambiente, estimar uma posição dentro deste ambiente, possuir a habilidade de reconhecer obstáculos, e responder em tempo real as diferentes situações que possam ocorrer neste ambiente. Além disso, todas estas funcionalidades devem operar em conjunto. As tarefas de perceber o ambiente, se localizar no ambiente, e se mover pelo ambiente evitando colisões, são problemas fundamentais no estudo dos robôs móveis autônomos.

As tarefas que permitem que um robô móvel se locomova de um ponto ao outro no ambiente são chamadas de tarefas de navegação robótica. Existem diversos problemas que dificultam a navegação em um ambiente real: ambientes dinâmicos que se alte-

ram com o passar do tempo (ex: mobília que é deslocada de posição), obstáculos móveis que se deslocam continuamente pelo ambiente, imprecisão dos sensores e erros de resposta aos comandos enviados aos atuadores, entre outros.

Na robótica móvel autônoma, o principal problema que dificulta em muito a navegação é o problema da localização. Sem saber a posição do robô em relação a sua representação do ambiente (mapa), um sistema de controle tem dificuldades de controlar o robô de forma adequada, e possui limitações na execução de tarefas mais complexas. Um sistema de controle robusto deve ter a capacidade de se localizar em um ambiente, utilizando as informações sensoriais e um mapa para estimar a posição do robô. Este problema é complexo, e forma a base de um sistema de controle robótico.

O principal objetivo deste trabalho é desenvolver um sistema de controle robusto para robôs móveis autônomos que seja capaz de operar e de se adaptar a diferentes ambientes e condições. O sistema de controle deve ser capaz de se localizar no ambiente utilizando um mapa e os dados sensoriais. O sistema deverá ser capaz de manter uma estimativa de posição correta a partir de uma posição conhecida (localização local), se localizar globalmente sem uma posição inicial conhecida (localização global), e se recuperar de possíveis erros de localização (relocalização). O sistema de controle deve ser capaz de navegar em um ambiente dinâmico evitando colisões com obstáculos estáticos e móveis. Para implementar este sistema de controle foi proposta uma arquitetura de controle híbrida.

2 Arquitetura de Controle

Neste trabalho foi proposta uma arquitetura de controle híbrida, integrando em uma única estrutura os principais métodos utilizados atualmente no controle de robôs móveis autônomos.

O controle do robô móvel foi separado em 3 camadas: camada vital, camada funcional e camada deliberativa (Fig. 1), sendo cada uma delas responsável pelo controle reativo, controle de execução de planos, e tarefas de planejamento a longo prazo, respectivamente. Este tipo de sistema também foi adotado nos trabalhos realizados por Gat [2] com a arquitetura ATLANTIS, e por Bonasso et al. [3] com a arquitetura 3T.

Para fornecer uma base sólida para a execução das tarefas desempenhadas pelas camadas de controle, foi integrado na arquitetura um módulo localizador. Este módulo localizador deve fornecer uma estimativa da posição do robô em relação a um mapa utilizando os dados sensoriais. Um módulo localizador é uma parte essencial de uma arquitetura de controle para robôs móveis autônomos, e na arquitetura proposta neste trabalho ele desempenha um papel de destaque.

A forma como o ambiente é representado internamente no sistema de controle, determina a sua precisão e performance. Cada uma das principais abordagens para o controle de robôs móveis autônomos utiliza, e melhor se adapta, a uma determinada representação de ambiente. Como a arquitetura proposta neste trabalho integra diversas abordagens de uma forma híbrida, a representação de ambiente utilizada é composta de diversas camadas: camada poligonal, camada matricial e camada topológica/semântica (Fig. 1).

Para permitir a comunicação entre os vários componentes da arquitetura de controle é disponibilizada uma área de memória compartilhada. Através do uso deste depósito central de informações, os diversos módulos podem trocar informações vitais para o funcionamento do robô móvel autônomo.

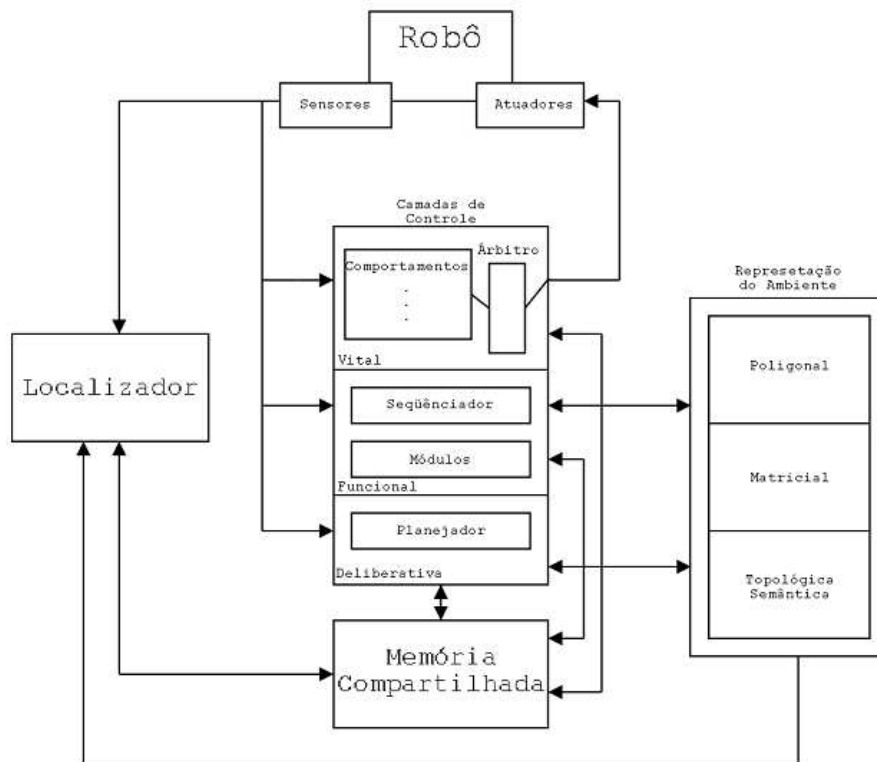


Fig. 1. Diagrama da arquitetura de controle, onde são visualizados os seus principais componentes: Camadas de controle; Localizador; Representação do Ambiente; e Memória Compartilhada.

2.1 Sistema de controle COHBRA

Com base na arquitetura proposta foi implementado o sistema de controle **COHBRA** (Controle Híbrido de Robôs Autônomos). Foi dada uma atenção especial no desenvolvimento do módulo localizador, que consideramos o componente principal do sistema de controle e um dos focos principais deste trabalho. Uma vez que o robô móvel autônomo possua uma boa estimativa da sua posição real no ambiente, as tarefas de navegação se tornam mais simples e precisas. A seguir os principais componentes do sistema de controle serão descritos.

2.1.1 Representação Interna

Para a **representação interna do ambiente** foram utilizadas todas as camadas definidas na arquitetura de controle: camada poligonal, camada matricial e a camada topológica/semântica. A camada poligonal é utilizada principalmente pelo módulo localizador para estimar a posição do robô, e é fornecida inicialmente pelo usuário em formato DXF como uma planta baixa do ambiente. A camada matricial é utilizada pela camada deliberativa para planejar uma trajetória até o objetivo. A camada matricial é gerada a partir da camada poligonal discretizando o ambiente em células. A camada topológica / semântica tem como principal função auxiliar o planejamento, otimizando a velocidade do cálculo de trajetória.

2.1.2 Módulo Localizador

O **módulo localizador** foi implementado utilizando-se a técnica de localização Monte Carlo [4]. A localização Monte Carlo possui uma série de vantagens: ela utiliza menos recursos computacionais do que a maioria das outras técnicas; concentra os recursos utilizados nas áreas de maior interesse para a localização; por ser uma técnica probabilista fornece mais informação além da localização do robô (a certeza sobre a localização, por exemplo).

Um dos principais motivos da escolha do método de localização Monte Carlo para a implementação do módulo localizador do sistema de controle COHBRA, é a sua capacidade de resolver os 3 grandes problemas da localização: localização local, localização global, e relocalização.

Na localização local, o localizador Monte Carlo é capaz de manter uma posição correta a partir de uma posição inicial com um nível aceitável de erro, sem a necessidade de uma função externa de recalibragem. O localizador Monte Carlo é capaz de se localizar globalmente, não necessitando de uma informação inicial sobre a sua posição. Esta técnica de localização também é capaz de se relocalizar, conseguindo detectar quando a posição atual, aparentemente correta, não reflete a posição real do robô.

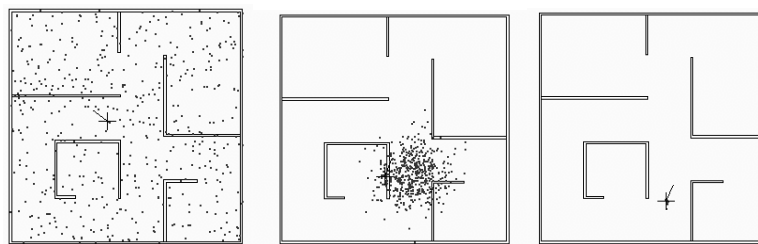


Fig. 2. Seqüência de imagens mostrando a evolução da distribuição das partículas durante a localização do robô móvel utilizando o algoritmo Monte Carlo.

O algoritmo de localização Monte Carlo é dividido em duas fases: fase de movimentação e fase de leitura sensorial. Inicialmente são geradas N amostras (partículas) distribuídas uniformemente em todo o mapa do ambiente (Fig. 2), ou distribuídas em

torno da posição mais provável do robô. Uma amostra é composta da posição (x,y,direção) e de uma certeza associada. Na **fase de movimentação** do robô, o MCL gera N novas amostras que aproximam a posição do robô após que ele tenha se movimentado. Cada amostra é gerada aleatoriamente retirando-se uma amostra do conjunto de amostras anterior, com a chance sendo determinada pela certeza da amostra antiga. Sendo l' a posição da amostra antiga, a posição l da nova amostra é gerada utilizando $P(l | l', a)$, com a ação de movimento ' a ' observada. Na **fase de leitura sensorial**, as informações são incorporadas redimensionando o peso da série de amostras, utilizando-se os valores sensoriais para estimar a certeza de uma determinada amostra com relação a sua posição.

Para determinar se o robô está ou não localizado é utilizada a melhor amostra (com a maior certeza), e uma medida de dispersão que indica se a maioria das partículas estão concentradas em torno desta posição. A dispersão é a média da distância aproximada entre todas as partículas com uma certeza maior do que um limiar preestabelecido. Quanto maior a dispersão, mais ambíguo será o conjunto de partículas. As camadas de controle levam em conta esta informação na hora de comandar o robô.

Um dos problemas da localização Monte Carlo é que se assume que o ambiente é estático, que a representação do ambiente corresponde ao ambiente real e que não existem obstáculos móveis. Uma situação deste tipo dificilmente é encontrada em um ambiente real, por esse motivo foi necessária a utilização de certos recursos para contornar este problema.

Uma abordagem para tentar solucionar este problema é proposta por Fox [6], que utiliza uma técnica de filtragem para ignorar certas leituras dos sensores quando estas não representarem uma leitura esperada. Fox propôs duas técnicas, filtro de entropia e filtro de distância. A técnica utilizada na implementação do localizador Monte Carlo foi a de filtro de distância, pois ela se adapta melhor com sensores de distância utilizados no robô.

Além de permitir que o robô móvel autônomo se localize em um ambiente dinâmico, o filtro de distância auxilia aos outros módulos do sistema de controle. Ele disponibiliza na área de memória compartilhada as informações sobre os sensores filtrados, permitindo assim que se detecte novos obstáculos estáticos, pois um sensor filtrado indica uma diferença entre a representação interna e o ambiente real. Estes dados são então utilizados pela camada de controle funcional para atualizar a representação interna do ambiente.

2.1.3 Camada Vital

A **camada vital** é responsável pelo controle reativo do robô móvel autônomo, sendo composta de diversos processos simples executados em paralelo chamados de "comportamentos". Estes comportamentos relacionam as entradas sensoriais com as saídas para os atuadores. Cada comportamento pode ser visto como uma "reação motora", reagindo diretamente aos estímulos do ambiente.

Na camada vital foram implementados 5 comportamentos primitivos, tendo como objetivo capacitar o robô a seguir as trajetórias calculadas pela camada deliberativa; auxiliar o módulo localizador; e manter a integridade física do robô. Ao seguir a trajetória pré-estabelecida o robô móvel autônomo deve ser capaz de desviar dos obstáculos inesperados, tanto estáticos como móveis. Os 5 comportamentos da camada vital

são: Parar, Vagar, Desviar de Obstáculos, Ir em direção ao Alvo, Inverter Direção. O comportamento de Desviar de Obstáculos é baseado no método de campos potenciais proposto por Borenstein & Koren [5].

As relações entre estes comportamentos são gerenciadas através de um **árbitro**. O árbitro tem a função de ativar ou inibir certos comportamentos, dependendo dos comandos recebidos pelo seqüenciador da camada funcional. O árbitro também pode ser programado com regras de fusão de saídas, para unificar saídas ambíguas dos comportamentos.

2.1.4 Camada Funcional

A **camada funcional** é composta de diversos módulos que interagem entre si, executando diversas funções de integração entre os componentes do sistema de controle.

Uma das funções da camada funcional é selecionar quais comportamentos primitivos a camada vital executará em um determinado instante de tempo, e fornecer parâmetros para estes comportamentos. Fornecendo informações e alterando a seqüência de execução destes comportamentos, o robô consegue executar as tarefas de alto nível planejadas pela camada deliberativa.

O seqüenciamento é executado através da inibição das saídas dos comportamentos. Nenhum módulo da camada vital deve atuar diretamente no controle dos atuadores do robô. Ao invés disso, os módulos da camada funcional fornecem informações para os comportamentos da camada vital que estiverem ativos, e inibem as saídas dos comportamentos que precisam ser desativados.

O seqüenciador da camada funcional foi implementado na forma de um autômato finito. Cada estado deste autômato indica para o árbitro da camada vital quais comportamentos devem ser acionados ou inibidos.

A camada funcional também é composta por módulos funcionais que desempenham diversas tarefas auxiliares no processo de controle do robô móvel autônomo. No sistema de controle COHBRA foram utilizados uma série de módulos funcionais. Um destes módulos é responsável pelo monitoramento e atualização da representação interna do ambiente, e os demais fornecem parâmetros que auxiliam outros módulos nas diferentes camadas de controle.

2.1.5 Camada Deliberativa

A **camada deliberativa** tem a função exclusiva de planejar a trajetória até um objetivo indicado pelo usuário, ou requisitado através de uma interrupção da camada funcional.

O planejamento é processado em duas fases. Na primeira fase, utilizando as informações topológicas, é executado um pré-planejamento que determina a seqüência de regiões topológicas que farão parte do caminho final, o algoritmo utilizado é o de Dijkstra [7]. Esta informação otimiza a fase final de planejamento. Na fase final, o algoritmo A* [8] é utilizado para calcular a trajetória definitiva. A representação matricial, otimizada com o pré-planejamento, é utilizada nesta fase.

O planejamento de trajetória utilizando o algoritmo A* produz um caminho que é composto da seqüência de células que devem ser seguidas para se chegar até o objeti-

vo. Esta sequência de células é convertida para uma sequência de pontos na representação poligonal utilizando como base o centro da célula. O plano final é a trajetória a ser seguida pelo robô móvel autônomo, da sua posição atual até o seu objetivo, composta de uma sequência de pontos na camada de representação poligonal. Este plano é disponibilizado na memória compartilhada para a ser utilizado pela camada funcional.

3 Simulador

Para validar o sistema de controle proposto, optou-se por implementar um simulador que possuisse todos os recursos necessários para a realização de experimentos com robôs móveis autônomos em ambiente dinâmicos.

O simulador, chamado de **SimRob3D** (Simulador de Robôs Móveis em Ambiente Tridimensional), tem como principal característica o fato de implementar a arquitetura de controle proposta e de se utilizar um ambiente tridimensional para a navegação dos robôs móveis simulados. Este ambiente pode ser modelado em diversos softwares de modelagem tridimensional existentes no mercado (AutoCad, 3D Studio, entre outros), pois o simulador utiliza o formato de arquivo “.3DS” que é um formato bastante conhecido na área de computação gráfica. Este formato de arquivo permite que sejam especificados os diversos elementos de um ambiente (objetos, luzes, texturas), o que resulta em um ambiente com um nível de realismo muito superior aos ambientes utilizados nos simuladores bidimensionais.

O simulador permite a configuração dos obstáculos, permitindo que eles sejam movimentados pelo usuário em tempo real, ou que os obstáculos sejam programados com trajetórias cíclicas.

O simulador possui diversos modelos sensoriais e cinemáticos (atuadores), permitindo a configuração de diversos tipos de robôs. É importante destacar que todos os sensores e atuadores interagem com o ambiente tridimensional, e são adicionados **ruídos** sensoriais e **erros** de resposta, tornando a simulação mais realista.

Um característica importante do simulador é a sua modularidade. O controlador é programado separadamente como uma biblioteca dinâmica. O controlador é carregado em tempo de execução, e pode ser implementado na linguagem de preferência do pesquisador.

4 Resultados

Foram realizados experimentos de **localização** em ambientes estáticos, dinâmicos, e dinâmicos com obstáculos móveis para avaliar a capacidade de localização local, global e de relocalização. Os experimentos foram realizados no ambiente Trinity [9] (figura 3), com as alterações necessárias para avaliar cada tipo de localização.

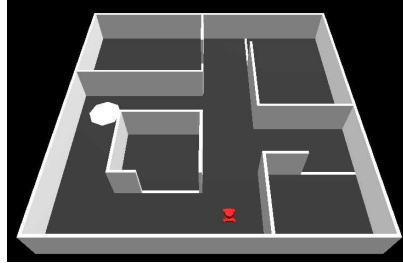


Fig. 3. Ambiente tridimensional utilizado nos experimentos de localização.

A tabela 1 apresenta os resultados obtidos, com o módulo localizador configurado para utilizar uma amostragem de 1000 partículas.

Tabela 1. Resultados obtidos nos experimentos de localização. Experimentos realizados seguindo-se uma trajetória pré-estabelecida, com um total de 2700 ciclos. Cada ciclo corresponde a um conjunto de leituras sensoriais recebidas do robô e utilizadas pelo módulo localizador para estimar a posição. Os valores foram obtidos a partir da média de 10 simulações para cada experimento

Experimento	Quant. de Ciclos para se Localizar	Quant. de Ciclos Localizado
Localização Local	0	2700
Localização Global	674,4	2025,5
Relocalização	681,8	2018,2

Foram realizados experimentos de **navegação** em ambientes estáticos, dinâmicos, e dinâmicos com obstáculos móveis, com o objetivo de avaliar a capacidade do sistema de controle em se locomover nestes ambientes, seguindo os planos fornecidos pela camada deliberativa.

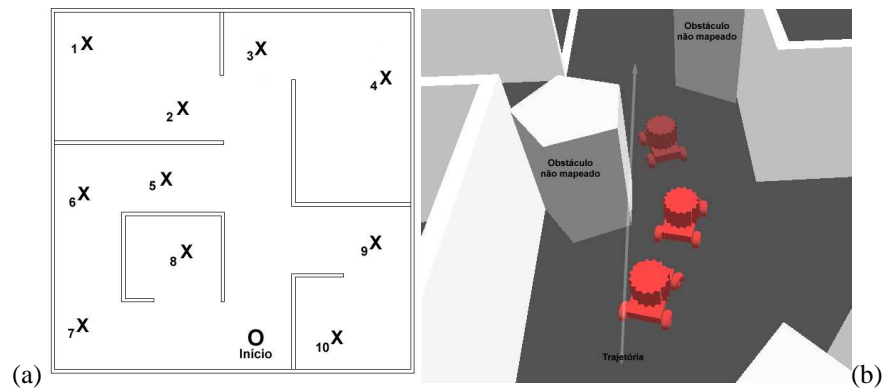


Fig. 4 Mapa do ambiente Trinity mostrando os pontos de destino dos experimentos de navegação (a). Seqüência de movimento do robô para se desviar de um obstáculo inesperado (b).

Os resultados obtidos nos experimentos de navegação mostraram que, em todas as simulações (10 simulações por experimento: a partir da posição inicial, se locomover até os pontos de destino 1-10; ver Fig. 4a), o sistema de controle foi capaz de guiar o robô móvel até o seu destino desviando dos obstáculos não mapeados ou móveis (Fig. 4b), e o módulo localizador foi capaz de manter a posição correta durante todo o percurso. O módulo localizador manteve a posição estimada com uma probabilidade média de 98% e uma dispersão de 0,73cm com 91% das partículas agrupadas. O erro real de posição ficou em 2,16cm em média, em um ambiente com 6,15m².

Estes resultados demonstram a boa integração entre a camada deliberativa e a camada vital (reativa). Também pode-se observar a capacidade que o módulo localizador possui em manter uma posição correta, mesmo em um ambiente com várias alterações em relação ao mapa.

Os experimentos demonstraram que o sistema de controle foi capaz de manter uma estimativa de posição correta em todas as configurações, possuindo uma capacidade de localização local excelente. Os experimentos com localização global e relocalização obtiveram bons resultados na maioria dos ambientes, somente tendo dificuldades nos ambientes com grandes alterações. Na tarefa de navegação o sistema de controle foi capaz de locomover o robô até os objetivos determinados em todos os tipos de ambiente utilizados nos experimentos.

5 Conclusão

Os resultados mostraram que o sistema de controle foi capaz de controlar o robô móvel autônomo e executar uma tarefa de navegação tanto em ambientes estáticos como em ambientes dinâmicos com obstáculos móveis. O robô móvel foi capaz de seguir as trajetórias globais calculadas sem colidir com nenhum obstáculo ou ficar preso em mínimos locais, mesmo em um ambiente dinâmico. A arquitetura proposta atingiu plenamente os objetivos de realizar uma navegação global com um baixo custo computacional, evitando os mínimos locais em um ambiente dinâmico. Isto demonstra que foi possível através da combinação dos diferentes métodos obter um sistema de controle que integra as melhores características de cada um destes métodos. Os experimentos revelaram que o robô consegue manter uma boa estimativa de posição durante as tarefas de navegação, mantendo uma posição bastante precisa na maior parte do tempo.

A principal contribuição deste trabalho foi a proposta de uma nova arquitetura de controle para robôs móveis autônomos, que foi validada através de experimentos e se mostrou robusta e capaz de operar em ambientes dinâmicos. O sistema COHBRA se mostrou capaz de operar um robô móvel autônomo em ambientes em constante alteração e na presença de obstáculos móveis, foi capaz de localizar o robô nestes ambientes mesmo quando o ambiente não refletia por completo o modelo armazenado na representação interna (mapa).

Referências

1. G. Dudek and M. Jenkin. **Computational Principles of Mobile Robotics**. Cambridge University Press, Cambridge, UK. 2000.
2. E. Gat. **Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots**. AAAI-92 Proceedings, AAAI Press, 1992.
3. R. P. Bonasso., et al. **Experiences with an Architecture for Intelligent Reactive Agents**. Journal of Experimental and Theoretical AI, 9(2). 1997.
4. D. Fox, W. Burgard, F. Dellaert, and S. Thrun. **Monte Carlo localization: Efficient position estimation for mobile robots**. In Proc. of the National Conference on Artificial Intelligence (AAAI). 1999.
5. Borenstein, J. and Koren, Y. **Real-time Obstacle Avoidance for Fast Mobile Robots**. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, pp. 1179-1187. 1989.
6. D. Fox. **Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation**. Institute of Computer Science III, University of Bonn, Germany. Doctoral Thesis. 1998.
7. T. Cormen, C. Leiserson, R. Rivest. **Introduction to Algorithms**. MIT Electrical Engineering and Computer Science Series. MIT Press. 1990.
8. Nilsson N. J. **Principles of Artificial Intelligence**. Tioga Publishing Company. 1980.
9. Trinity College. **Fire-Fighting Home Robot Contest Website**. Última Atualização: Abr 2002. "<http://www.trincoll.edu/events/robot/>".