

DEVELOPMENT OF A HYBRID ARCHITECTURE FOR THE AUTONOMOUS CONTROL OF MOBILE ROBOTS

Hugo Silva, hugodaluz@unb.br

Alberto Álvares, alvares@alvarestech.com

Department of Mechanical Engineering and Mechatronics - University of Brasilia

Abstract. *Um dos desafios mais significativos da robótica móvel está na área de navegação autônoma, onde o robô deve estar apto a sentir e navegar em um ambiente incerto e geometricamente restrito sem sofrer qualquer tipo de interferência externa. O desenvolvimento de uma arquitetura que permite a navegação autônoma surge como uma combinação única de assuntos como sensoramento, localização, mapeamento, planejamento de trajetória e execução de movimento. O objetivo desse trabalho é o desenvolvimento de uma arquitetura híbrida (combinando técnicas de planejamento e reatividade) para permitir a navegação autônoma de robôs móveis em um ambiente conhecido e estruturado. A arquitetura está sendo desenvolvida de forma modular permitindo o desenvolvimento e incorporação de várias técnicas de mapeamento, localização e planejamento de trajetória independentemente da estratégia reativa utilizada no robô. A arquitetura proposta visa integrar um módulo de planejamento de trajetória utilizando Diagrama de Voronoi e um módulo de comportamento reativo utilizando controladores fuzzy para controle de direção e velocidade através de um módulo de coordenação. O planejamento de trajetória será realizado utilizando um mapa criado a priori e o algoritmo de Dijkstra será usado para a busca do melhor caminho. Após a geração da trajetória, o módulo de coordenação é responsável por enviar as ações de controle ao robô utilizando informações de posicionamento para definir se o robô seguirá o caminho planejado ou executará ações reativas. Além disso, o módulo de coordenação monitora a execução da rota definida, através de um controlador nebuloso, decidindo se é necessário o re-planejamento. A arquitetura proposta será validada em um robô Nomad XR4000 e atualmente está sendo desenvolvida utilizando uma linguagem de programação orientada a objetos. Os primeiros resultados desse trabalho são: a interface gráfica desenvolvida para integrar os módulos e um protótipo do módulo de planejamento.*

Keywords: *Fuzzy Logic, Hybrid Architecture, Mobile Robotic, Path-Planning, Voronoi Diagram.*

1. INTRODUCTION

Os robôs móveis são utilizados como ferramentas em vários campos de aplicação, tais como plantas de fábricas, serviços médicos, exploração espacial e futebol de robôs. Essa versatilidade tem contribuído para o desenvolvimento de uma base científica e matemática através da pesquisa de questões teóricas e fundamentais que de outra forma poderiam não ser propostas. Um robô autônomo pode ser definido de forma abstrata como o mapeamento de uma seqüência de entradas sensoriais para uma ação apropriada em resposta a essas percepções (Duffy, 2000). Segundo Siegwart and Nourbakhsh (2004), a atuação do robô no mundo físico está sujeita às restrições geométricas e incertezas, dessa forma, para que a navegação autônoma seja bem sucedida é necessária a combinação de pesquisas em sensoramento, localização, mapeamento e planejamento de trajetória.

Esse artigo é parte de uma pesquisa em andamento do Grupo de Inovação e Automação Industrial (GIAI) da Universidade de Brasília e tem o objetivo de criar uma arquitetura híbrida para o controle de robôs móveis. É importante notar que essa pesquisa está direcionada a robôs físicos situados em um mundo real que influencia diretamente o comportamento desse sistema. O resultado esperado ao fim da pesquisa é uma arquitetura que permita o comportamento autônomo do robô, fazendo com que o mesmo navegue usando um planejador de trajetória, coordenando o caminho planejado com ações reativas para evitar colisões com obstáculos imprevistos. Nessa arquitetura o planejamento e a reação são tratados como tarefas complementares, pois cada uma é essencial para o sucesso da outra.

Embora existam várias arquiteturas propostas em publicações (Fernandez-Leon et al., 2008; Grassi, 2006; Barberá, 2001; Arkin, 1998; Brooks, 1986), a maioria delas considera desprezíveis as incertezas inerentes à tarefa de navegação em ambientes reais ou limita-se à aplicações em ambientes simulados em condições ideais ou propõe abordagens baseadas em comportamentos simples (seguir paredes, desviar de obstáculos). Os resultados experimentais indicam a utilidade dessas abordagens e assim esses algoritmos podem fornecer uma boa base para planejadores completos. Porém, essas abordagens nem sempre garantem provas de completude onde um caminho pode ser encontrado, nem fornecem princípios bem estabelecidos para quando esses algoritmos falham.

O artigo está organizado da seguinte maneira, na seção 2 são apresentados os tipos de arquiteturas para controle robótico, a seção 3 aborda a arquitetura proposta, na seção 4 é apresentado o estado atual do desenvolvimento da arquitetura e o robô móvel utilizado na validação, o Nomad XR4000 e a seção 5 traz as conclusões do trabalho em andamento.

2. ARCHITECTURES FOR ROBOT CONTROL

Uma arquitetura faz com que as percepções dos sensores disponíveis sejam fornecidas a um programa que irá alimentar os atuadores com as ações geradas com base nas percepções. Muitos pesquisadores têm apresentado propostas de arquiteturas ao longo do tempo, cada qual adaptada às tarefas a serem executadas na pesquisa e nas características do robô utilizado. As arquiteturas podem ser classificadas com relação ao nível de deliberação e reatividade no sistema. As classificações descritas por Arkin (1998) são: deliberativas, reativas, e híbridas (deliberativas e reativas).

2.1 Reactive Architecture

Os robôs reativos também são ditos baseados em comportamento, pois são instruídos a atuar através da ativação de uma coleção de comportamentos primitivos de baixo nível (Arkin, 1998). Comportamentos físicos complexos surgem através da interação do conjunto comportamental e das complexidades do ambiente no qual o robô se encontra (Brooks, 1986). Para o paradigma reativo, a representação do ambiente em que o robô atua é desnecessária e as ações devem ser mapeadas diretamente das percepções obtidas dos sensores. Essa metodologia provê respostas mais rápidas e flexíveis que as alcançadas pelos métodos tradicionais de controle robótico.

2.2 Deliberative Architecture

Segundo Duffy (2000), abordagens meramente deliberativas derivam da inteligência artificial clássica onde o objetivo era construir um cérebro que podia exibir certo grau de comportamento inteligente. A abordagem deliberativa pode ser utilizada em situações onde o mundo pode ser modelado, onde as incertezas são restritas e onde existem algumas garantias de que não haverá mudanças significativas no mundo durante a execução, desde que um plano possa ser executado. Tais sistemas são naturalmente lentos, seriais, baseados em conhecimento e com recursos limitados.

2.3 Hybrid Architecture

A arquitetura híbrida une deliberação e reação, tendo como objetivo reduzir o tempo de resposta do robô a mudanças ambientais, ao mesmo tempo em que realiza planejamentos que possuem alto custo computacional. É considerada ideal para robôs que precisam percorrer ambientes conhecidos e com o mínimo de mudança na sua configuração. Além disso, permitem a adaptação do comportamento baseado no modelo de mundo disponível, flexibilizando e otimizando sistemas meramente reativos.

3. PROPOSED ARCHITECTURE

A arquitetura proposta nesse trabalho visa fornecer comportamento autônomo em ambientes conhecidos considerando as incertezas dos sensores do robô e principalmente a possibilidade de existência de obstáculos móveis ou estacionários que não estejam previstos no plano de navegação. A arquitetura proposta nesse trabalho permitirá que o robô:

- Formule, execute e modifique os planos de navegação de acordo com o estado do mundo.
- Perceba e reaja a mudanças no ambiente, navegando sem colisões com obstáculos móveis e/ou estacionários.

Os módulos propostos para compor a arquitetura são: Sensoriamento, Mapeamento e Localização, Planejamento, Coordenação e Ação. Esses módulos serão executados de forma independente, mas compartilharão informações durante a navegação autônoma. A interação entre os módulos pode ser vista na figura 1 e será descrita durante essa seção.

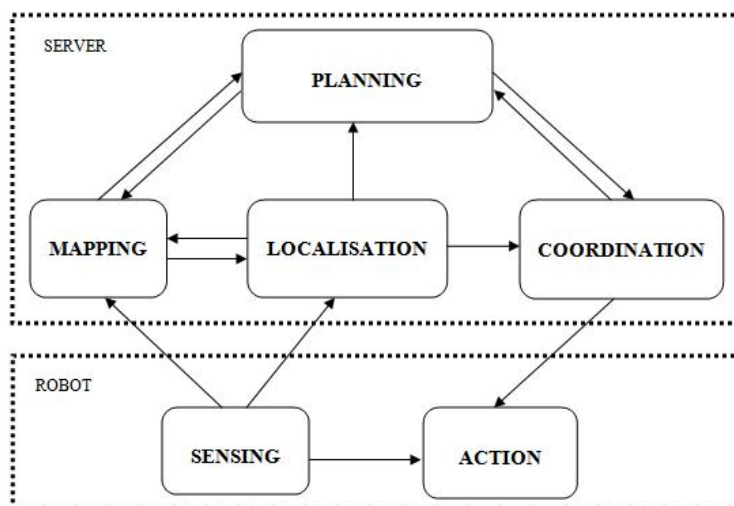


Figura 1: A interação entre os módulos da arquitetura, onde as setas representam fluxo de informações.

3.1 Sensing

O módulo de Sensoriamento é o responsável pelas aquisições e armazenamento dos dados dos sensores do robô. A percepção é o processo pelo qual os robôs mapeiam as medições dos sensores em representações internas do ambiente e é uma tarefa difícil porque em geral os sensores são ruidosos e o ambiente é parcialmente observável, imprevisível e geralmente dinâmico. Por isso, nesse módulo as representações internas devem possuir três propriedades citadas por Russel and Norvig (2003): conter informação suficiente para o robô tomar as decisões corretas, estar bem estruturada para ser atualizada de forma eficiente e ser natural e possuir variáveis internas com valor correspondente ao estado natural das variáveis no mundo físico.

Além de armazenar os dados dos sensores seguindo as propriedades citadas, esse módulo deverá acrescentar o erro nas leituras, o erro de posicionamento do robô e aplicar técnicas de filtragem para descartar leituras inválidas. Essas informações serão utilizadas pelos módulos de Localização, Mapeamento e Ação.

3.2 Localisation

A localização é um dos problemas mais importantes na robótica móvel, pois é o ponto principal em qualquer interação física de sucesso. Os robôs móveis devem saber onde estão para que possam encontrar a localização do alvo. O módulo de Localização é o responsável por definir o posicionamento do robô no ambiente da forma mais precisa possível. A utilização de dados odométricos para a localização do robô tem como principal desvantagem o acúmulo do erro que aumenta à medida que o robô se desloca, a menos que uma referência independente seja usada periodicamente para reduzi-lo (Borenstein et al., 1996). Apesar de trabalhos anteriores considerarem esse tipo de erro desprezível quando o robô percorre pequenas distâncias, a arquitetura proposta prevê rotinas para correções no posicionamento através da combinação de informações dos módulos de Mapeamento, Sensoriamento e Localização.

3.3 Mapping

O módulo de Mapeamento é o responsável por criar, armazenar, atualizar e fornecer acesso aos mapas e informações globais e locais do ambiente. Os mapas gerados por métodos de mapeamento automático possuem várias vantagens sobre a geração manual quando consideradas as mudanças do ambiente, as características que devem ser reconhecíveis aos sensores e o nível de detalhamento oferecido (Choset et al., 2005) e essa construção pode ser feita antes ou durante a navegação (Elfes, 1987). Esse módulo permitirá a construção de mapas identificando as características e a posição dos obstáculos no ambiente (com uma incerteza associada).

Esse módulo terá dois comportamentos, de forma off-line permitirá a construção dos mapas dos ambientes através dos sensores de distância e durante a navegação criará os mapas locais para que o módulo de Localização possa identificar as regiões que possuem similaridade entre o mapa local e o mapa global calculando a sua posição atual. A figura 2 descreve com detalhes os módulos envolvidos e a dinâmica do cálculo da posição através das informações sensoriais e a correspondência entre os mapas locais e globais.

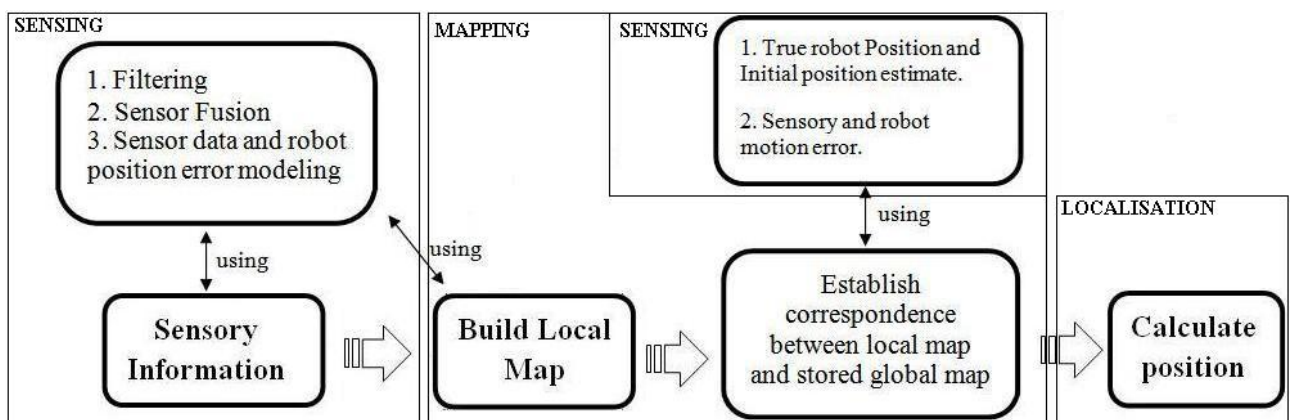


Figura 2: Interação entre os módulos para determinação da posição do robô (Adaptado de Borenstein et al. (1996)).

3.4 Planning

O planejamento é uma atividade onde um objetivo é especificado em linguagem de alto-nível e um robô de posse dessas especificações as transforma em um conjunto de primitivas de movimento de baixo-nível para cumprir o objetivo (Siegwart and Nourbakhsh, 2004). O resultado pode ser um plano de ação descrito por uma seqüência de subtarefas compostas de comandos básicos disponíveis no sistema e pode ser feito em diferentes níveis de abstração, variando desde um nível mais alto, como no caso do planejamento de tarefas, até um nível mais específico, como é o caso do planejamento de trajetória (Grassi, 2006). Na arquitetura proposta, o módulo de Planejamento é o responsável por gerar a trajetória para o alcance do objetivo utilizando a representação do ambiente e a localização do robô, fornecidas pelos módulos de Mapeamento e Localização respectivamente.

Segundo Choset et al. (2005), o problema do planejamento de trajetória pode ser descrito como: dado um robô com uma configuração inicial, uma configuração objetivo, sua forma e um conjunto de obstáculos localizados no espaço de trabalho, encontrar para ele um caminho livre de colisões da configuração inicial à configuração objetivo. Métodos de planejamento de movimento podem ser utilizados para determinar quais movimentos o robô deve realizar de forma a alcançar uma posição ou configuração desejada no ambiente livre de colisões com obstáculos, levando em consideração aspectos da sua dinâmica e suas restrições de movimento (Grassi, 2006).

Esse problema, aparentemente trivial, tem se mostrado de grande dificuldade principalmente na aplicação de técnicas que funcionem eficientemente na presença de múltiplos obstáculos. Existem vários métodos para lidar com o problema de planejamento de trajetória e suas extensões tais como: decomposição em células, mapa de rotas, campos potenciais, representação poligonal e planejamento baseado em amostragem. Algumas das razões para a existência das várias abordagens desenvolvidas são as suposições feitas a respeito da forma do robô e dos obstáculos e as restrições impostas pela estrutura mecânica do robô. Uma discussão sistemática desses métodos pode ser encontrada nas referências (Choset et al. 2005; Siegwart and Nourbakhsh 2004; Russel and Norvig 2003; Latombe 1991). Esse problema conta com muita informação que deve ser conhecida do ambiente, sendo este estático, bem estruturado e completamente conhecido, existe pouca dificuldade nos problemas de navegação, podendo ser resolvidos com métodos existentes (Choset et al., 2005). Porém, em ambientes reais a navegação do robô deve considerar as incertezas associadas aos sensores e ao ambiente e também à dinâmica ambiental.

Nesse trabalho será utilizado um método de mapa de rotas, que é uma decomposição do espaço de configuração do robô baseado especificamente na geometria do obstáculo (Choset *et al.*, 2005). Essa a abordagem captura a conectividade do espaço livre do robô em uma rede de curvas e linhas unidimensionais. O desafio é construir um conjunto de rotas que juntas permitam o robô ir a qualquer lugar no seu espaço livre, enquanto minimiza o número total de rotas. Esse conjunto de rotas é uma estrutura geométrica que o robô usa para planejar um caminho entre dois pontos em um ambiente. Esse mapa é usado como uma rede de segmentos de trajetória para o planejamento de movimento do robô, que fica reduzido a uma busca por uma série de rotas que ligam a posição inicial do robô até o objetivo (Choset et al., 2005).

O tipo de mapa de rotas utilizado para o planejamento será o Diagrama de Voronoi (exibido na figura 3), um método completo que tende a maximizar a distância entre o robô e os obstáculos no mapa. Algoritmos que encontram caminhos no mapa de rotas de Voronoi são completos porque a existência de um caminho no espaço livre implica na existência de um no diagrama de Voronoi. Os algoritmos para cálculo de diagramas de Voronoi podem ser divididos em três classes de acordo com a técnica empregada (Berger): incremental, divisão e conquista e varredura do plano.

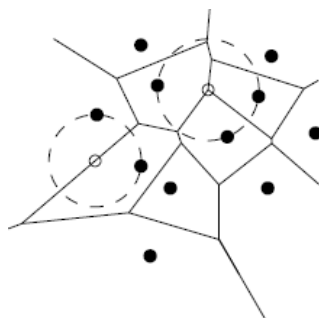


Figura 3: Diagrama de Voronoi de um conjunto de pontos. Onde os vértices são pontos que estão próximos de três ou mais geradores. E as arestas são pontos que estão próximos de dois geradores (de Berger et al, 2008).

Para esse trabalho foi utilizado o algoritmo de Fortune (de Berger et al, 2008), também conhecido como Método de Varredura. A estratégia desse algoritmo é deslocar uma linha horizontal - a linha de varredura- do topo até a base do plano (comportamento exibido na figura 4). Enquanto a varredura é realizada, informações são mantidas a respeito das interseções da estrutura com a linha de varredura. Enquanto a linha de varredura se move para baixo a informação não muda, exceto em certos pontos especiais, os pontos de evento (de Berger et al, 2008).

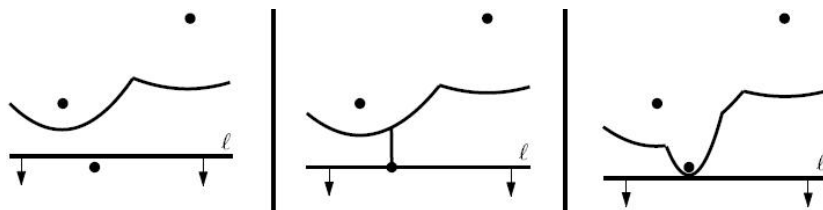


Figura 4: O método de varrimento, onde pode ser vista a linha de varredura (l) passando pelos sítios (de Berger et al, 2008).

No diagrama gerado por essa técnica, as arestas são partes dos bissetores de pares de pontos de eventos e os vértices são pontos de interseção entre esses pontos. Após a geração desse grafo, é preciso que ocorra a “poda” para eliminar os vértices e arestas que estão em posições inválidas, que impossibilitam a movimentação do robô no ambiente. Esse Diagrama de Voronoi será calculado antes da navegação a partir de um mapa global do ambiente fornecido a priori pelo módulo de Mapeamento. Choset e Burdick (2000) apresentam em seu trabalho um método de planejamento de trajetória criando um Diagrama de Voronoi Generalizado Hierárquico de forma incremental em ambientes desconhecidos com base em informações sensoriais. Essa abordagem pode ser utilizada em situações onde não há mapa do ambiente ou o surgiram novos obstáculos durante a navegação.

Para realizar a busca no grafo de Voronoi, foi escolhido o algoritmo de Dijkstra, utilizado para calcular o menor caminho entre a origem e o alvo, e que requer que cada aresta no grafo seja associada a um peso. Assim, o problema original de planejamento de trajetória é reduzido a um problema de busca de caminho em um diagrama de Voronoi, que é um problema de busca em um grafo discreto (Choset *et al.*, 2005).

A vantagem dessa técnica é que um robô com sensores de ultra-som pode seguir a extremidade de Voronoi usando as mesmas regras da criação do diagrama (Choset *et al.*, 2005) e os caminhos resultantes maximizam a distância até os obstáculos. Dessa forma, o movimento baseado em Voronoi pode minimizar a imprecisão dos sensores. A desvantagem é a produção de caminhos longos que o caminho ótimo no sentido de tamanho total quando o espaço de configuração é muito aberto (Russel and Norvig, 2003). A seqüência de ações resultante pode ser vista na figura 5, desde a criação do Diagrama de Voronoi até o envio da trajetória ao robô.

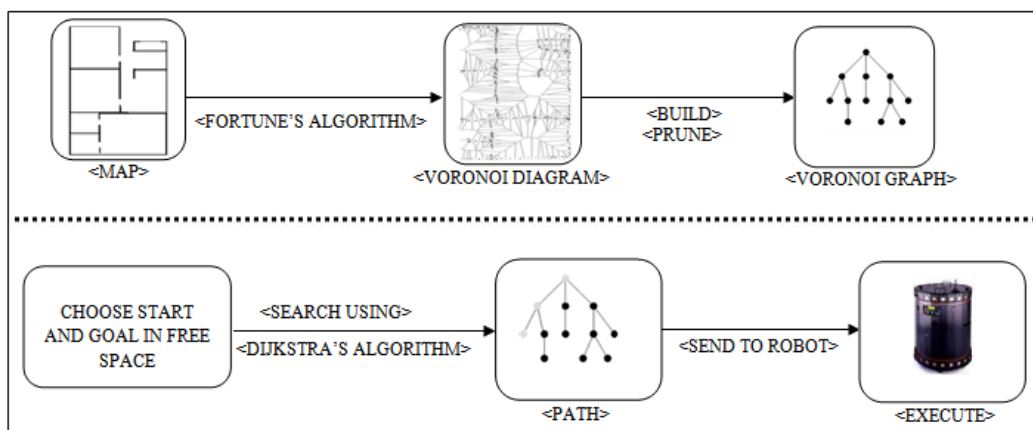


Figura 5: Seqüência de ações que envolvem a geração do Diagrama de Voronoi e o posterior envio do caminho encontrado para o robô.

3.5 Coordination

O módulo de Coordenação é o responsável por coordenar a interação entre o planejamento e a reatividade, gerenciar a execução da trajetória planejada monitorando o robô. Para isso, deve gerar um conjunto de ações com base na trajetória planejada, enviar para o módulo de Ação no robô e monitorar sua execução em tempo real, solicitando o re-planejamento da trajetória sempre que necessário.

Em caso de erro pela existência de novos obstáculos móveis ou estacionários, esse módulo deverá interromper a execução da trajetória atual e se for identificada a necessidade de re-planejamento, o obstáculo detectado deve ser incluído no mapa pelo módulo de mapeamento e uma nova trajetória deve ser calculada. Se a geometria do obstáculo não for identificada completamente pelo robô, o mapa será atualizado parcialmente e a trajetória planejada pode ser modificada à medida que o robô possui mais detalhes sobre a forma do obstáculo. A figura 6 apresenta o ciclo de re-planejamento que pode ocorrer durante a navegação.

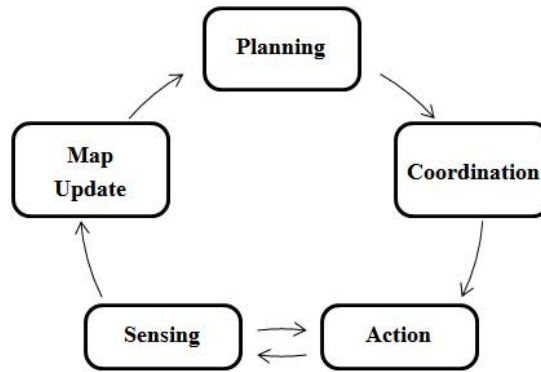


Figura 6: O ciclo que envolve percepção, atualização do mapa, planejamento, coordenação e ação deve ser repetido durante a navegação autônoma.

Para realizar a monitoração da trajetória percorrida pelo robô, será criado um controlador fuzzy que determina se o robô está seguindo a trajetória corretamente ou se existe a necessidade de criação de uma nova trajetória entre a posição atual e a posição alvo. Esse controlador nebuloso utilizará as informações de posicionamento do robô, a trajetória definida pelo módulo de planejamento e um conjunto de regras nebulosas.

3.6 Action

O módulo de Ação é o responsável por gerar os movimentos do robô com base no planejamento efetuado e comportamentos reativos de acordo com as percepções do ambiente. Esse módulo é composto por controladores Fuzzy responsáveis por ajustes na direção do robô e controle de velocidade e por um conjunto de classes que controlam o movimento do robô. Dessa forma, o robô poderá evitar colisões durante a navegação enquanto não ocorre o re-planejamento.

O controlador fuzzy para navegação deverá controlar os desvios de obstáculos do robô de forma reativa. Para isso um conjunto de regras será criado para relacionar os estímulos sensoriais (infravermelho e ultra-som) e sinais de atuação sobre a direção atual do robô. As regras propostas serão criadas com o objetivo de manter o robô na direção do alvo e ao mesmo tempo desviar dos obstáculos para que não haja grande desvio de trajetória e que o robô possa identificar, mesmo que parcialmente o obstáculo que está impedindo a navegação. As tarefas de desvio de obstáculos e navegação ao alvo podem ser concorrentes e as regras devem priorizar a segurança do robô.

O controlador fuzzy para ajuste de velocidade permitirá o controle da velocidade do robô, acelerando quando distante de obstáculos e desacelerando quando próximo, permitindo que haja tempo para ajustes na direção, utilizando para isso as informações sensoriais.

4. IMPLEMENTATION

A arquitetura proposta começou a ser desenvolvida a partir de uma interface gráfica para a integração dos módulos para comunicação com o robô. Essa interface permite a escolha da forma do robô para representação e o mapa do ambiente que será utilizado. Na figura 7 é possível visualizar o protótipo da interface gráfica, após a escolha do robô e do mapa, sendo este exibido em duas dimensões. Essa ferramenta, conta ainda com um módulo de conexão com o robô, que permite a disponibilização dos dados sensoriais para os outros módulos.

O robô utilizado para a avaliação da arquitetura é o Nomad XR4000, um sistema robótico holonômico que utiliza técnicas de controle distribuído e possui um robusto design mecânico e modular oferecendo três graus de liberdade: dois de translação e um de rotação (Nomadic, 1999). Esse robô possui sistemas de sensores de colisão, de ultra-som, infravermelhos e integradores (odômetros).

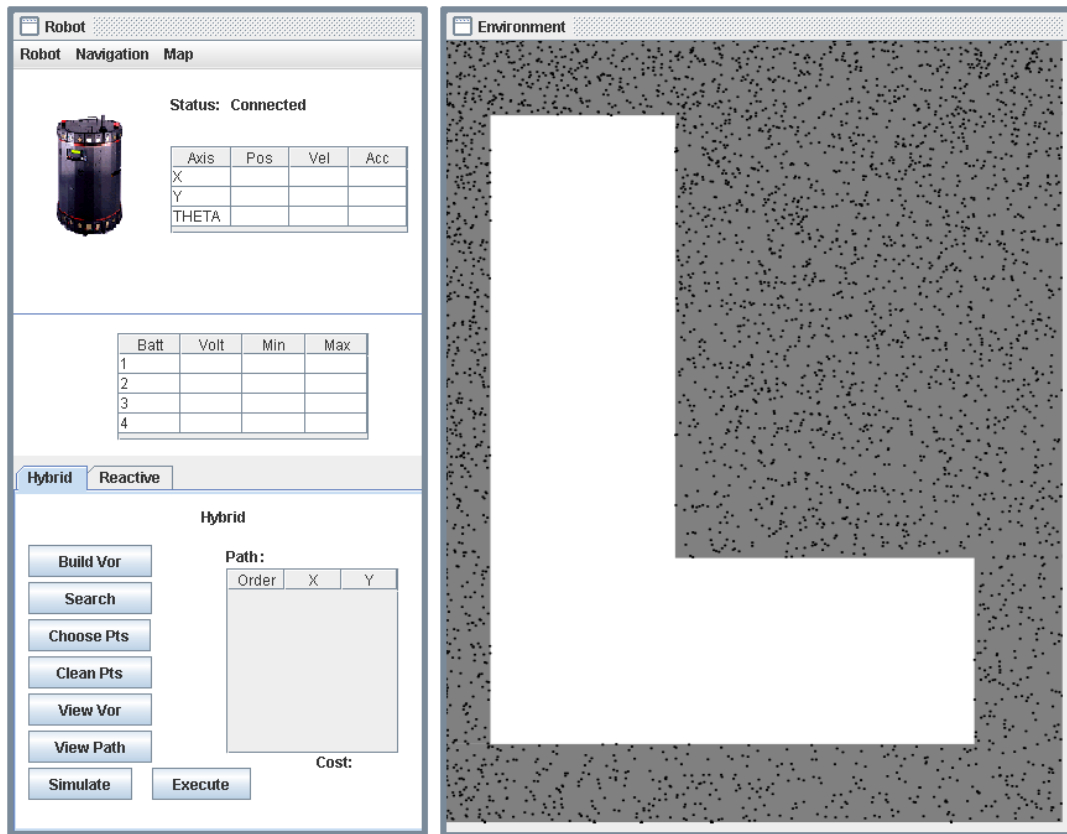


Figura 7: Interface gráfica do protótipo desenvolvido.

A figura 8 mostra os resultados do início do desenvolvimento do módulo de Planejamento. Utilizando um mapa simples, criado de forma manual, foi possível gerar o Diagrama de Voronoi a partir do conjunto de pontos do mapa. O resultado é um mapa de rotas que permite o movimento seguro do robô. A partir da definição dos pontos de início e objetivo, o algoritmo de Dijkstra foi utilizado para buscar o melhor caminho dentro do grafo de Voronoi, resultando na trajetória do robô.

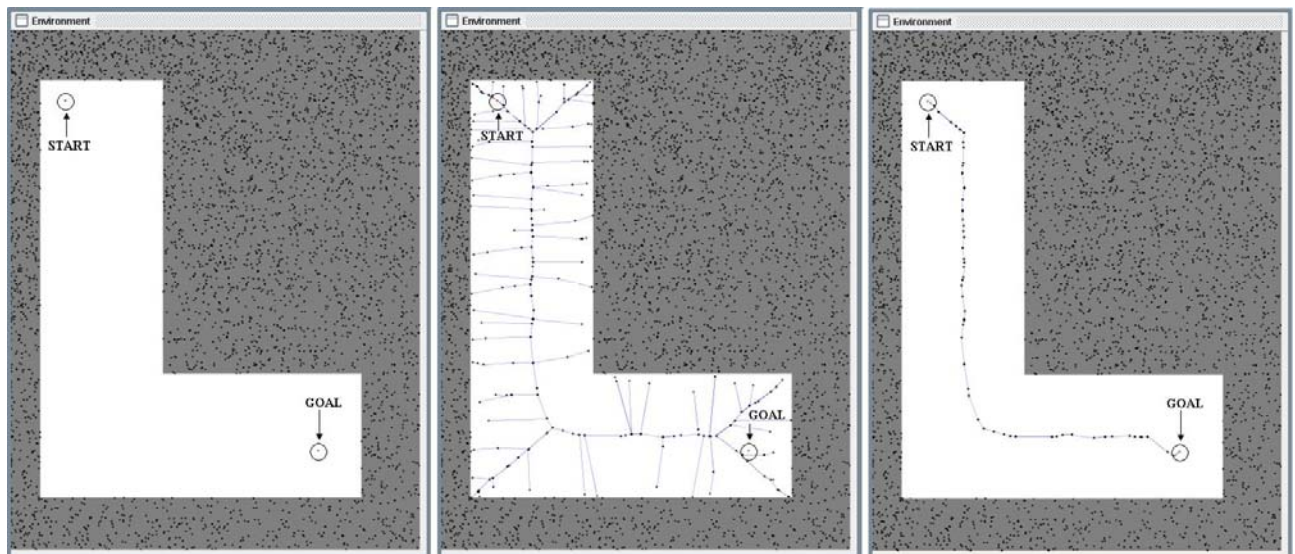


Figura 8: Sequência de ações: Escolha dos pontos inicial e final, visualização do Diagrama de Voronoi criado e busca pelo melhor caminho.

Conclusions

A pesquisa em torno da navegação autônoma de robôs móveis tem apresentado diversos trabalhos com uma grande variedade de abordagens distintas. A proposta desse trabalho envolve o desenvolvimento de uma arquitetura híbrida coordenando planejamento e reatividade. Essa arquitetura deverá lidar situações imprevistas, adaptando a trajetória do robô em tempo real, evitando colisões com obstáculos móveis ou estacionários, percebendo e reagindo às mudanças ambientais.

Apesar de ser um trabalho ainda em curso já é possível avaliar a viabilidade da arquitetura proposta com protótipo desenvolvido. O módulo de planejamento já está pronto para gerar um plano de navegação para o robô através de um mapa criado à priori. Apesar do algoritmo utilizado ainda não considerar a incerteza associada aos obstáculos no mapa, em implementações subseqüentes essa característica será incorporada para aumentar sua robustez. Em trabalhos futuros, o desenvolvimento e integração dos outros módulos da arquitetura proposta serão suficientes para garantir a navegação autônoma de robôs móveis.

REFERENCES

- Arkin, R. C., 1998, "Reactive robotic systems", 793--796.
- Barberá, H. M., 2001, 'A Distributed Architecture for Intelligent Control in Autonomous Mobile Robots - An Applied Approach to the Development of the Quaky-Ant Platform', PhD thesis, Dept. of Communications and Information Engineering, University of Murcia.
- Brooks, R., 1986, 'A robust layered control system for a mobile robot', *Robotics and Automation, IEEE Journal of* 2(1), 14-23.
- Choset, H.; Lynch, K. M.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L. E. & Thrun, S., 2005, *Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents)*, The MIT Press.
- de Berg, M.; Cheong, O.; van Kreveld, M. & Overmars, M., 2008, *Computational Geometry: Algorithms and Applications*, Springer, Berlin.
- Duffy, B., 2000, 'The Social Robot, Ph.D Thesis', PhD thesis, Department of Computer Science.
- Elfes, A., 1990, 'Sonar-based real-world mapping and navigation', 233--249.
- Feng, L.; Borenstein, J. & Everett, B., 1994, 'Where am I? Sensors and Methods for Autonomous Mobile Robot Localization', Technical report, University of Michigan.
- Fernandez-Leon, J. A.; Acosta, G. G. & Mayosky, M. A., 2008, 'Behavioral control through evolutionary neurocontrollers for autonomous mobile robot navigation', *Robot. Auton. Syst.* 57(4), 411--419.
- Grassi, V., 2006, 'Hybrid Architecture for Mobile Robots Based on Navigational functions with human interaction', PhD thesis, Polytechnic School, USP.
- Jain, L. C. & Martin, N. M., 1998, *Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms: Industrial Applications*, CRC Press, Inc., Boca Raton, FL, USA.
- Latombe, J.-C., 1990, *Robot Motion Planning (The Springer International Series in Engineering and Computer Science)*, Springer.
- Nomadic, 1999, 'Nomad XR4000 Hardware Manual Release 1.0', Nomadic Technologies, California, USA.
- Russell, S. J. & Norvig, P., 2003, *Artificial Intelligence: A Modern Approach*, Pearson Education.
- Siegwart, R. & Nourbakhsh, I. R., 2004, *Introduction to Autonomous Mobile Robots*, Bradford Book.

5. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.