

Técnicas de Navegação

Diego Pereira Rodrigues
Departamento de Engenharia da Computação e Automação Industrial
Universidade Estadual de Campinas - UNICAMP

Resumo—Este artigo trata da definição, classificação e comparação de alguns dos algoritmos utilizados para a Navegação Robótica, como solução para os problemas enfrentados na implementação desta.

Palavras Chave—Classificação, Navegação, Problemas, Técnicas.

I. INTRODUÇÃO

QUANDO nos referimos a navegação de robôs móveis autônomos, queremos descrever as técnicas que fornecem os meios para que um robô autônomo se mova de forma segura de um local a outro do ambiente. Encontrar um caminho de uma determinada posição até um destino é um dos problemas fundamentais da robótica móvel autônoma.

II. CLASSIFICAÇÃO DAS TÉCNICAS DE NAVEGAÇÃO

As técnicas utilizadas em navegação robótica podem ser classificadas em dois grandes grupos: (A) medidas relativas da posição e (B) medidas absolutas da posição (Borenstein, et al., 1995)[1].

A. Navegação baseada em medidas relativas da posição

- **Navegação Odométrica:** Este método usa encoders para medir a rotação das rodas e controlar a orientação do veículo. A vantagem deste sistema de navegação é sua autonomia e a capacidade de prover sempre ao veículo uma estimativa de sua posição. A desvantagem deste sistema é que o erro de posição aumenta sem limites, a menos que uma referência independente seja usada periodicamente para reduzir o erro.

- **Navegação Inercial:** Este método usa giroscópios e algumas vezes acelerômetros para tomar medidas das rotações e acelerações do veículo. As medidas são integradas para obter uma estimativa da velocidade e posição do robô. Este sistema também tem a vantagem de ser autônomo. Já uma desvantagem nestes sistemas é o alto custo dos sensores. Os giroscópios, com tecnologia recente, usam fibra ótica, conhecidos também como giroscópios a laser e são de muita precisão.

B. Navegação baseada em medidas absolutas da posição

- **Navegação por sinais ativos (“active beacons”):** Este método computa a posição absoluta do robô através da medida da direção de incidência de três ou mais sinais ativos

transmitidos. Os transmissores usualmente usam luzes ou frequências de radio e são localizados em lugares conhecidos do ambiente.

- **Navegação por reconhecimento de marcas artificiais:** Este método usa marcas artificiais distintas as quais são localizadas em lugares conhecidos do ambiente de navegação. A vantagem destas marcas artificiais é que podem ser projetadas para serem otimamente detectadas e reconhecidas, mesmo em ambientes com condições adversas.

- **Navegação por reconhecimento de marcas naturais:** Este método utiliza marcas com características próprias distintas do ambiente, o qual não precisa ser preparado. Este implica que o ambiente tem que ser muito bem conhecido. A confiabilidade deste método não é tão alta como o método onde se utilizam marcas artificiais.

- **Navegação por comparação de modelos (“Model Matching”):** Através deste método a informação adquirida pelos sensores do robô é comparada com um mapa modelo do ambiente. Se as características do mapa criado com base nos sensores do robô e o mapa modelo do ambiente coincidem, então a posição absoluta do robô pode ser estimada. Os mapas usados em navegação podem ser classificados em dois tipos: (1) mapas geométricos e (2) mapas topológicos. Mapas geométricos representam o espaço de navegação total, denominado na literatura inglesa como “world (W)”, em um sistema de coordenadas globais, em quanto os topológicos representam o espaço de navegação em uma rede de nós e arcos.

III. PROBLEMAS RELACIONADOS

Diversos problemas vem desafiando os pesquisadores da área de robótica móvel a muitos anos (localização do robô, obstáculos móveis, imprecisão do mapa, imprecisão dos sensores). Diversas técnicas foram desenvolvidas para tentar resolver estes problemas e tornar a navegação mais robusta e capaz de atuar de forma satisfatória em ambientes complexos.

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), pouquíssimos sistemas são capazes de controlar o robô de forma adequada. Este problema é complexo, e forma a base de um sistema de controle robótico.

IV. TÉCNICAS DE NAVEGAÇÃO

A seguir são apresentadas as três principais abordagens utilizadas pelas técnicas de navegação de robôs móveis autônomos (Sensoriais/Reativas, Roadmaps e Matriciais).

A. Abordagem Sensorial / Reativa

A navegação baseada em sensores incorpora as informações sensoriais, refletindo o estado atual do ambiente, diretamente no processo de planejamento do robô. Isto é justamente o oposto das técnicas de navegação clássicas, onde o conhecimento total sobre a geometria do ambiente precisa ser estabelecido antes de se iniciar o processo de planejamento. As principais vantagens da navegação baseada em sensores em relação aos métodos clássicos são:

- em muitos casos o robô não possui conhecimento antecipado sobre o ambiente onde ele irá atuar;
- o robô dispõe de um conhecimento parcial sobre o ambiente devido a limitação de memória;
- os modelos de ambiente geralmente não são precisos;
- no ambiente podem ocorrer situações inesperadas ou mudanças rápidas.

Existem diversas técnicas baseadas em sensores que podemos citar:

A.1 DistBug

O algoritmo Distbug tenta garantir que o destino seja atingido se possível, ou indicar se o destino for inacessível. O algoritmo é reativo no sentido de que ele depende dos dados sensoriais para tomar decisões locais, e não utiliza nenhuma forma de representação interna do ambiente.

O algoritmo consiste em dois modos de locomoção: se mover diretamente em direção ao alvo entre os obstáculos e seguir as bordas dos obstáculos. As bordas dos obstáculos são contornadas até que uma certa condição indique que ele deva novamente ir em direção ao alvo. A direção em que o modo de seguir as bordas decide se locomover irá influenciar o tamanho do caminho que será percorrido. O algoritmo decide em que direção ele se movimentará baseado em informações locais.

Esta técnica é simples, mas em um ambiente controlado é capaz de desviar de obstáculos e evitar situações de armadilha como os mínimos locais. Mas ela se limita a ambiente estáticos, que não possuam obstáculos móveis.

A.2 Navegação Baseada em Comportamentos

A navegação baseada em comportamentos utiliza uma ligação forte entre percepção e ação para se locomover de forma robusta na presença de incerteza no ambiente. O robô executa comportamentos como "abrir a porta" ou "entrar na sala". Para organizar estes comportamentos, geralmente se utilizam autômatos de estados finitos nos quais os estados correspondem aos comportamentos e os arcos correspondem aos gatilhos que indicam um novo comportamento. O estado atual indica o comportamento do robô. Quando o robô recebe as informações sensoriais, e estas informações correspondem a um determinado gatilho que o estado atual possui, o comportamento que o gatilho indica se torna o novo estado atual.

Já que os autômatos de estados finitos são baseados em comportamentos e gatilhos, o robô não necessita de um modelo do ambiente ou informações completas sobre o estado atual do mundo.

A.3 Mapas Neurais

Um mapa neural é uma "representação neural localizada dos sinais do mundo exterior" [Ama89]. Sinais provenientes de um espaço X são mapeados através de uma função $f(X)$ em um campo neural F . O campo neural F é uma rede neural recorrente, onde as dinâmicas definem as interações inibitórias e excitatórias entre as unidades.

O mapeamento assemelha-se ao espaço de configuração n -dimensional C de um robô e o sinal X é a informação atual sobre o ambiente (áreas livres, obstáculos, destino). As unidades da rede são distribuídas sobre C implementando uma representação discreta ordenada topologicamente de C . Assim, cada unidade i corresponde a uma única configuração representativa (ci) do robô e cada configuração possível de C é representada pelo ci mais próximo. O peso entre as unidades i e j refletem o custo de mover o robô de uma configuração ci para cj . Quanto maior o custo, menor é o peso. A dinâmica difusa do mapa é devido as interações excitatórias locais, que resultam em uma superfície de ativação estável sobre C , gerando um mapa de navegação completo até o destino. Um procedimento simples de subida passo-à-passo (steepest ascent) no equilíbrio da superfície de ativação de qualquer configuração inicial irá retornar um caminho livre de obstáculos até o destino mais próximo.

Para um robô móvel, C é um plano bidimensional. Em geral, as unidades do mapa são distribuídas de forma homogênea sobre C . A topologia e o número de conexões pode variar. As opções comuns incluem topologias retangulares e hexagonais com conexões de (e para) uma unidade i se estendendo para os seus vizinhos. O custo utilizado para determinar o peso das conexões é a distância euclidiana entre as diferentes configurações.

B. Abordagem Roadmap

A abordagem roadmap para a geração de caminhos

consiste em reduzir as informações ambientais em um grafo representando os possíveis caminhos existentes. Uma vez que um roadmap foi construído, um caminho pode ser calculado conectando a posição atual e o objetivo ao grafo e encontrando uma trajetória dentro deste grafo.

Em geral os métodos roadmap são rápidos e simples de se implementar, mas eles não fornecem uma boa forma de representação das informações do ambiente.

A seguir serão apresentados alguns métodos que utilizam a abordagem roadmap, mas antes é necessário apresentar algumas definições:

- Espaço de Configuração: Este conceito é utilizado quando o planejamento de trajetória é executado com objetos estacionários e conhecidos. O espaço de configuração é uma transformação do espaço físico onde o robô tem um tamanho bem definido, para um espaço onde o robô é tratado como um ponto.

- Espaço Livre: O espaço livre de um espaço de configuração simplesmente consiste das áreas que não são ocupadas por obstáculos.

- Caminho Livre: O caminho livre entre uma posição inicial e um objetivo é o caminho que pertence inteiramente ao espaço livre e não entra em contato com nenhum obstáculo.

B.1 Grafos de Visibilidade

Um grafo de visibilidade é obtido gerando-se segmentos de reta entre os pares de vértices dos obstáculos. Todo o segmento de reta que estiver inteiramente na região do espaço livre é adicionada ao grafo (Fig. 1). Para executar o planejamento de trajetória, a posição atual e o objetivo são tratados como vértices, isso gera um grafo de conectividade onde utiliza-se algoritmos de procura para se encontrar um caminho livre. Os métodos mais simples utilizados para construir um grafo de visibilidade tem complexidade $O(n^3)$.

O caminho mais curto que for encontrado no grafo de visibilidade é o caminho ótimo para o problema especificado. Mas os caminhos encontrados no grafo “tocam” os obstáculos. Isso não é aceitável em um problema de navegação robótica, por esse motivo os obstáculos devem ser modificados, criando-se um espaço de configuração. Para a utilizar um método de navegação com grafo de visibilidade é necessário quer o mapa seja completo e bem definido. Não é possível navegar em um ambiente que possua obstáculos móveis e a localização do robô móvel deve ser conhecida com precisão durante toda a navegação.

B.2 Diagramas de Voronoi

Um diagrama de Voronoi é um estrutura geométrica que representa informações de proximidade sobre uma série de pontos ou objetos. Dada uma série de sites ou objetos, o plano

é particionado atribuindo para cada ponto o seu site mais próximo. Os pontos que não possuem um único site mais próximo formam o diagrama de Voronoi. Isto é, os pontos no diagrama de Voronoi são equidistantes de dois ou mais sites.

No caso do planejamento de trajetória, a região

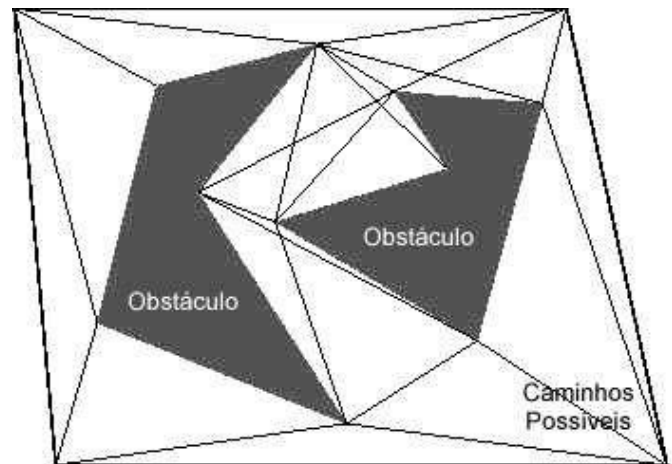


Fig. 2. Grafos de Visibilidade [2]

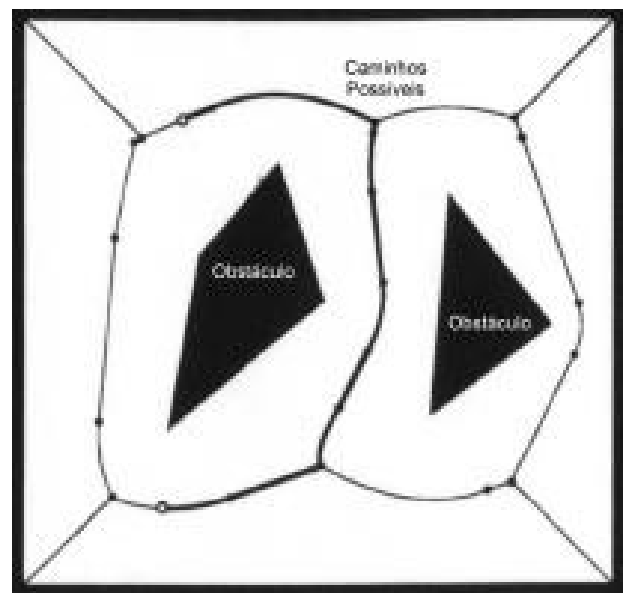


Fig. 2. Diagrama de Voronoi [2]

bidimensional em que o robô se locomove geralmente contém obstáculos, cada um destes obstáculos pode ser representado por polígonos côncavos ou convexas. Para encontrar o diagrama generalizado de Voronoi para esta coleção de polígonos, podemos calcular o diagrama através de uma aproximação, convertendo os obstáculos em uma série de pontos. Primeiramente as faces dos polígonos são subdivididas em uma série de pontos. O próximo passo é calcular o diagrama de Voronoi para esta coleção de pontos. Após o diagrama de Voronoi ser calculado, os segmentos do diagrama que intersectam algum obstáculo são eliminados.

Este método gera uma rota que na sua maior parte permanece equidistante dos obstáculos (Fig. 2), criando um caminho seguro para o robô se locomover, apesar de não ser o caminho mais curto.

B.3 Decomposição Celular

A idéia básica por trás do método de decomposição celular é que um caminho entre a posição inicial e o objetivo pode ser determinado subdividindo-se o espaço livre em regiões menores chamadas células. Após esta decomposição, um grafo de conectividade é construído de acordo com as adjacências entre as células, onde os nodos representam as células no espaço livre, e os arcos entre os nodos indicam as adjacências. Através do grafo de conectividade, um caminho contínuo, ou canal, pode ser determinado simplesmente seguindo as células livres adjacentes. Estes passos são ilustrados abaixo utilizando-se os métodos de decomposição celular exata e aproximada.

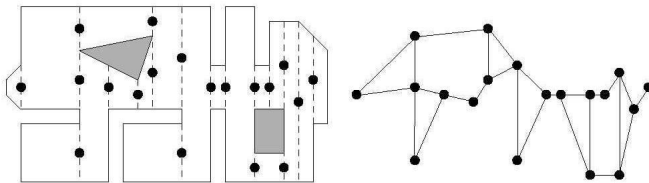


Fig. 3. Decomposição Celular [2]

* Decomposição Celular Exata

O primeiro passo neste tipo de decomposição celular é subdividir o espaço livre, que é limitado tanto externamente como internamente por polígonos, em células triangulares e trapezoidais traçando segmentos de reta paralelos a cada vértice interno até o polígono limitador externo. Então cada célula é numerada e representada como um nodo no grafo de conectividade. Nodos que são adjacentes no espaço de configuração são ligados no grafo (Fig. 3). Um caminho neste grafo corresponde a um canal no espaço livre. Este canal é então transformado em um caminho livre conectando a posição inicial e o objetivo através dos pontos médios das interseções das células adjacentes do canal.

C. Abordagem utilizando Matrizes

A abordagem utilizando matrizes, representam o ambiente utilizando uma matriz de tamanho fixo. Cada célula da matriz pode conter diferentes atributos que auxiliam na navegação do robô. Cada célula, por exemplo, pode indicar a presença de um obstáculo na região correspondente do ambiente, ou indicar a probabilidade desta determinada região estar ou não ocupada por um obstáculo.

As técnicas que utilizam matrizes (Fig. 4), também chamadas de matrizes de ocupação, são simples de se construir, representar e manter em ambientes de larga escala. O uso de matrizes facilita a determinação de caminhos até o destino.

No entanto as matrizes consomem muita memória e requerem muito tempo para atualizações globais. Isto ocorre pois a

resolução da matriz deve ser precisa o suficiente para capturar todos os detalhes importantes do ambiente.

C.1 Transformada da Distância

Neste método, descrito em [Len90], a célula do destino é marcada com uma distância de valor 0. Todas as outras posições são marcadas com um valor muito alto, ou “infinito”. O algoritmo inicia o cálculo no destino, e em cada passo ele visita todas as posições adjacentes às posições visitadas no passo anterior. O valor de distância para a posição “i” adjacente a posição “j” é atualizado da seguinte forma:

Se $map(i)$ estiver ocupado então $d(i) = \text{infinito}$;

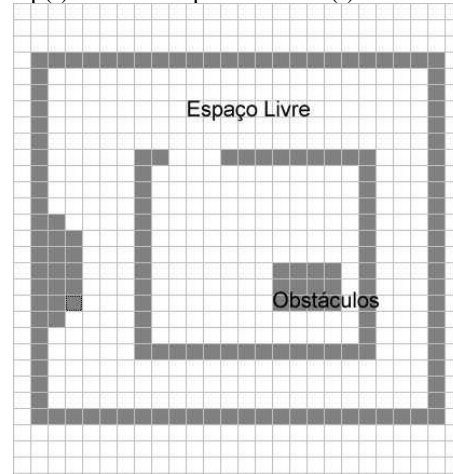


Fig. 4. Decomposição Celular [2]

Senão $d(i)$ recebe o menor entre $d(i)$ ou $(d(j) + c(i,j))$;

Onde $c(i,j)$ é o custo associado para se mover da posição j para a posição i.

A transformada de distância se expande em torno do destino como uma onda, se propagando em torno dos obstáculos. O caminho mais curto de qualquer posição até o destino pode ser encontrado seguindo os vizinhos com o menor $d(i)$ até que $d(i)$ seja igual a 0 (zero).

Cada vez que a posição do destino for alterada, a transformada de distância para todo o mapa deve ser recalculada. Outro problema associado com este método, é o fato de que os caminhos são gerados muito próximos dos obstáculos, não aceita obstáculos móveis e o mapa deve ser bem definido.

C.2 Campos Potenciais

Os métodos de campos potenciais para a navegação de robôs móveis são muito populares entre os pesquisadores da área. Os obstáculos exercem uma força repulsiva no robô, enquanto o objetivo aplica uma força atrativa no robô. A soma de todas as forças, a força resultante R, determina a direção e a velocidade do movimento. Uma das razões para a popularidade deste método é a sua simplicidade e elegância. Um método de campos potenciais simples pode ser

implementado rapidamente e produz um resultado inicial aceitável sem a necessidade de muitos refinamentos.

Simultaneamente, o conceito de campo potencial é aplicado ao histogram grid. A medida que o robô se move, uma janela de ($W_s \times W_s$) células o acompanha, se sobrepondo a região C. Esta região é chamada de região ativa (C^*), e as células que momentaneamente pertencem a esta região são chamadas de células ativas (c_{ij}^*).

Cada célula ativa exerce uma força repulsiva virtual F_{ij} contra o robô. A magnitude desta força é proporcional a c_{ij}^* e inversamente proporcional a d_n , onde d é a distância entre a célula e o centro do robô, e n é um número positivo.

Na implementação de Borenstein & Koren [Bor89] foi assumido um valor de $n=2$. A soma de todas as forças repulsivas gera a força repulsiva resultante F_r . Ao mesmo tempo, uma força atrativa virtual F_t de magnitude constante é aplicada no robô, puxando-o em direção ao objetivo. A soma de F_r e F_t gera o vetor de força resultante R . A direção de R é utilizada como referência para determinar a velocidade de rotação do robô.

Apesar dos bons resultados fornecidos pelos métodos de campos potenciais, eles possuem uma série de problemas inerentes, que independem de uma implementação em particular: - Situações de armadilha devido a mínimos locais (comportamento cíclico);

- Não conseguem passar através de obstáculos muito próximos;

- Oscilação na presença de obstáculos ou em corredores estreitos;

O problema mais conhecido e mais citado dos campos potenciais são os mínimos locais ou situações de armadilha (Andrew & Hogan [And83], Tilove [Til89]). Um mínimo local pode ocorrer quando um robô entra em uma área sem saída (por exemplo, um obstáculo em forma de U). Estas armadilhas podem ser criadas por diversas configurações de obstáculos. No entanto, estas situações de armadilha podem ser solucionadas através de heurísticas ou através de um replanejador global.

A grande vantagem deste método é que ele não necessita de uma representação interna do ambiente (mapa), e pode navegar por um ambiente que possua obstáculos móveis.

C.3 AStar(A^*)

Apesar do algoritmo de procura AStar(A^*) não necessitar que o espaço de estados seja representado como uma matriz, na maioria das aplicações de navegação para robôs móveis em que se utiliza AStar o ambiente é representado desta forma. Por este motivo esses algoritmos serão descritos aqui.

Praticamente qualquer problema pode ser solucionado (assumindo que existem recursos infinitos) utilizando-se um algoritmo de procura. Primeiro é necessário encontrar uma maneira de descrever o problema como uma coleção de estados, e regras que possam ser aplicadas para se alterar estes estados. Pode-se imaginar o espaço de estados como sendo uma árvore com o estado inicial do problema sendo a raiz, e se expandindo para baixo criando um galho em cada ponto, dependendo de quantas regras possam ser aplicadas nesses pontos.

Para solucionar um problema como esse é necessário encontrar um estado de destino nesta árvore e então traçar o seu caminho de volta ao estado inicial.

V. TABELA COMPARATIVA

Para que fosse possível comparar todos os métodos de uma forma mais estruturada, eles foram organizados em uma tabela. Nesta tabela são comparados os seguintes atributos:

- representação: indica o tipo de representação de ambiente que este método utiliza;

- custo computacional: indica o custo computacional do método;

- mínimos locais: indica se o método possui o problema de mínimos locais;

- global: indica se o método tem a capacidade de navegar de forma global no ambiente;

- ambiente dinâmico: indica se o método tem a capacidade de navegar em um ambiente dinâmico;

- posição precisa: indica se o método necessita da posição precisa do robô no ambiente;

Apesar de alguns dos métodos apresentados não necessitarem de uma posição precisa para executar a navegação do robô móvel autônomo, a maioria deles necessita saber a localização do robô. Um dos principais problemas relacionados com o controle e navegação de robôs móveis autônomos é a localização.

VI. CONCLUSÃO

Este trabalho foi importante pois mostra de forma resumida os princípios de algumas das técnicas de navegação mais utilizadas, tornando o leitor apto a decidir qual se adéqua melhor à sua aplicação. Para isto, o artigo traz a tabela comparativa das técnicas.

Método	Representação	Custo Computacional	Mínimos Locais	Global	Ambiente Dinâmico
DistBug	Não Utiliza	Baixo	Não	Não	Não
Nav. Baseada em Comportamentos	Não Utiliza	Baixo	Sim	Não	Sim
Mapas Neurais	Campo Neural	Alto	Sim	Sim	Não
Grafo de Visibilidade	Poligonal	Alto	Não	Sim	Não
Voronoi	Poligonal	Alto	Não	Sim	Não
Decomposição Celular	Poligonal Topológica	Médio	Não	Sim	Não
Transformada de Distância	Grade	Médio	Não	Sim	Não
Campos Potenciais	Grade, Poligonal	Médio	Sim	Não	Sim
Astar	Grade	Baixo	Não	Sim	Não
Dstar	Grade	Médio	Não	Sim	Sim

Fig. 5. Tabela Comparativa [2]

REFERENCES

- [1] Carlos Enrique Villanueva Cano. Técnica De Navegação De Um Robô Móvel Baseado Em Um Sistema De Visão Para Integrá-Lo A Uma Célula Flexível De Manufatura. PhD thesis, Universidade De Brasília, 2006.
- [2] Farlei José Heinen. Sistema de Controle Híbrido para Robôs Móveis Autônomos. PhD thesis, UNISINOS, 2002.
- [3] Roland Siegwart and Illah R. Nourbakhsh. Intelligent