

NAVEGAÇÃO REATIVA NA ROBÓTICA MÓVEL

Halana Santos Lisboa

halanalisboa17@gmail.com

Stephanie Kamarry Alves de Sousa

stephaniekamarryas@gmail.com

Bruno Donato dos Santos

donato@hotmail.com.br

Resumo: A navegação reativa é uma das arquiteturas aplicada a navegação de robôs móveis. Esse modelo traz consigo uma forte conexão entre a percepção e a atuação, ideal para aplicações em ambientes altamente dinâmico e não estruturado, ou seja, possui independência em relação ao modelo do ambiente de atuação. Em virtude do que foi mencionado, este trabalho coloca em evidência uma metodologia de ensino para navegação reativa através de simulação computacional utilizando o MATLAB e um código fonte de programação que utiliza métodos para contornar o obstáculo e detecta possíveis colisões, capazes de lidar com objetos dinâmicos e trajetória desconhecida sem sofrer limitações impostas pelo planejamento.

Palavras-Chave: Robô, simulação, metodologia, MATLAB, sensoriamento;

INTRODUÇÃO

O estudo da robótica móvel vem apresentando um crescimento acentuado no desenvolvimento de pesquisas nos últimos anos, com aplicação principalmente em áreas domésticas e industriais. A maneira como o robô escolhe a ação a ser executada é conhecida como arquitetura de navegação. Ao tratar-se de navegação de robôs móveis, segundo a literatura, existem arquiteturas que fornecem uma maneira segura do robô autônomo sair de um local a outro do ambiente tais como os modelos deliberativo, reativo e híbrido.

O presente trabalho tem foco voltado para a navegação reativa, que tem como a principal característica a forte conexão entre sensoriamento e atuação, refletindo o estado

atual do ambiente diretamente no processo de planejamento do robô, isto é justamente o oposto de outras técnicas de navegação, onde o conhecimento total sobre a geometria do ambiente precisa ser estabelecido antes de se iniciar o processo de planejamento, a exemplo da navegação deliberativa.

No modelo reativo, evita-se o alto custo computacional envolvido na elaboração e manutenção de exemplos confiáveis do ambiente de operação do robô, além disso, essas arquiteturas constituíram-se em simples métodos baseados em regras, assim sendo, tendem a ser rápidas e adequadas quando o ambiente é muito dinâmico, onde a velocidade de resposta é crucial para o funcionamento correto do sistema, porém não é indicado para tarefas que exijam um bom comportamento. Alguns obstáculos podem ter seus comportamentos caracterizados e previstos, muitos outros mostram-se totalmente ou parcialmente imprevisíveis (STÜRMER; HEINEN, 2004). Os métodos propostos para contornar tal problema baseiam-se em detecção de colisão, capazes de lidar com objetos dinâmicos e trajetória desconhecida sem sofrer limitações impostas pelo planejamento, características da navegação reativa.

Levando em consideração esses aspectos, o objetivo deste resumo é explicar uma metodologia de ensino para navegação reativa através de simulação computacional utilizando o MATLAB e um código fonte de programação, esse desenvolvido pelo grupo

de pesquisa em robótica da universidade federal de Sergipe (GPRUFS).

METODOLOGIA

A ferramenta utilizada para a simulação da navegação reativa consiste em um roteiro prático, onde o mapa é desconhecido e o robô deve chegar ao destino sem sofrer colisões e utilizando sensores para descobrir o que encontra.

Definições

- Ponto de partida (X_i, Y_i): é o ponto que se define como origem;
- Ponto de destino (X_d, Y_d): é o ponto que se define como chegada da trajetória;
- Posição do robô (X, Y): é o ponto em que o robô ocupa uma posição no plano;
- Obstáculo (X_o, Y_o): é o ponto onde existe um obstáculo conhecido pelo sensor do robô;
- Local livre: São todos os pontos onde o sensor do robô não detecta obstáculos.

Estratégias

Nesta etapa foram pontuadas observações consideráveis para um bom desempenho do robô, tais como:

- o sensoriamento, o qual foi observado que uma maior quantidade de sensores aplicados na lógica de programação aumentaria a chance de detecção de obstáculos, evitando assim colisões.
- o controle de velocidade linear (V_2), a qual diminui-se à medida que se aproxima do obstáculo.
- o controle de velocidade angular (W_2), a qual aumenta à medida que se aproxima do obstáculo.

Todas as alterações nos valores das variáveis foram utilizadas na realização de testes em diversos ambientes, além disso foi utilizado uma outra estratégia em conjunto, que se refere ao controle de posição final, ou

seja, após o robô ter conseguido desviar de todos os obstáculos e ao passo que se aproxima do destino final a valor da velocidade linear (V_1) é reduzida gradativamente.

```
V2 = 0;
if d_obs_min < 150 && d > 140
    Wmax2 = Wmax;
    Kw = 0.5;
    W2 = Wmax2*tanh(Kw*theta_e_obs);
    W1 = 0;
    V1 = V1/2;
    if v_sensor(1) || v_sensor(16) < 150 && d > 140
        Wmax2 = Wmax;
        Kw = 0.5;
        W2 = -Wmax2*tanh(Kw*theta_e_obs);
        W1 = 0;
        V1 = V1/2;
    end
end
```

Figura 1 - Introdução da estratégia de programação para o desvio de obstáculo.

A figura 1 retrata uma das lógicas aplicadas, em que os comportamentos eram analisados de acordo com o mapa, onde as constantes expressas recebem valores caso a distância mínima entre o robô e o obstáculo (d_obs_min) alcance um valor menor que 150, distância essa pré definida na lógica, além disso era condicional que o ponto de destino (d) estivesse a um valor menor que 140 centímetros. Nessas condições a velocidade angular (W_1) e linear (V_1) recebem o valor de zero e $V_1/2$, respectivamente, ambas referentes ao destino final. Visto que o robô possui 16 sensores, é possível, através da variável $v_sensor(x)$, definir qual sensor será utilizado no momento. Na figura 1, é possível ver a utilização dos sensores 1 e 16.

```
if v_sensor(8) || v_sensor(9) < 150 && d > 140
    Wmax2 = Wmax;
    Kw = 0.5;
    W2 = -Wmax2*tanh(Kw*theta_e_obs);
    W1 = 0;
    V1 = V1/2;
end
else
    V2 = V1;
    W2 = W1;
end
```

Figura 2 - Implementação da estratégia de programação para o desvio de obstáculo - Parte II

Continuando a implementação lógica buscou-se aplicar sensores ($v_sensor(x)$) fazendo uma subdivisão por quadrantes, ou seja, dividindo as regiões de visão do robô, dado como exemplo o da figura 2 na qual a condição trata-se do intervalo dos sensores 8 e 9, correspondendo a lateral do robô, convencionalmente escolhido.

RESULTADOS E DISCUSSÃO

Os resultados obtidos a partir da metodologia de simulação, utilizando o MATLAB, especificamente para a navegação reativa de robôs foram bastantes positivos, pois foram analisados importantes dados antes despercebidos e com esses fazer correções para melhorar o desempenho do robô.

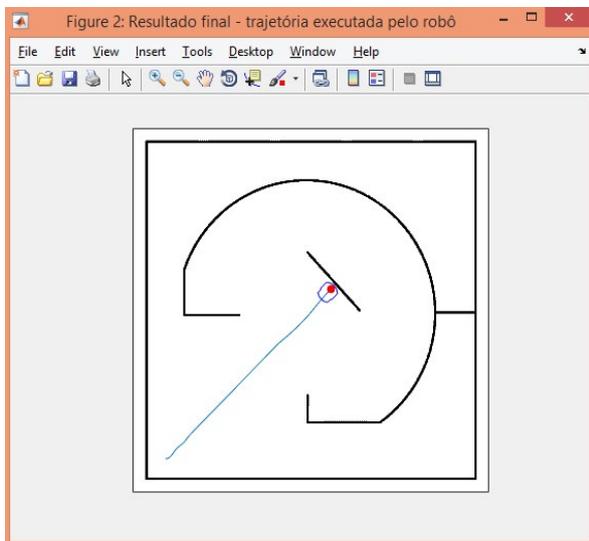


Figura 3 - Resultado final de trajeto - mapa 1

A partir do simulador percebe-se o comportamento do robô em relação a sua lógica, o percurso realizado por ele, pois o próprio sistema traça uma linha em azul dos movimentos até a chegada ao seu ponto de destino (X_d, Y_d), esse sinalizado por um ponto vermelho no mapa. Além da trajetória, foram examinados os valores de velocidade angular e linear ao longo do tempo, como ilustra a figura a seguir.

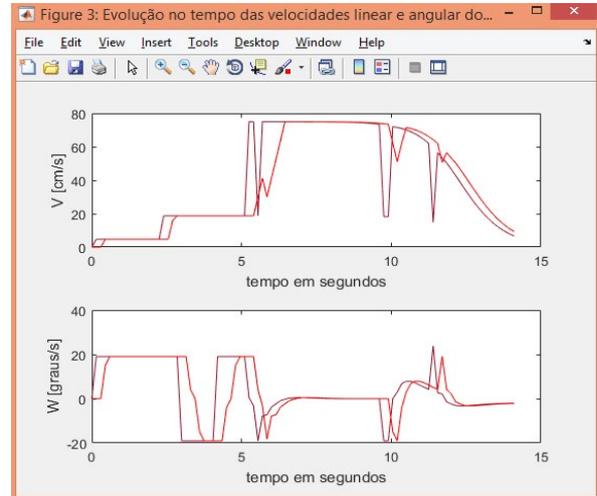


Figura 4 - Resultados da velocidade linear e angular, respectivamente (V e W) - mapa 01

Cada gráfico exibido acima apresenta duas linhas que correspondem à desvio de obstáculo e ao ponto de destino. Com o resultado alcançado pelo simulador, do tempo de resposta das velocidades pode-se perceber que em ambos os casos as velocidades se assemelham ao decorrer do tempo, até chegar no (X_d, Y_d), apresentado em alguns períodos uma pequena desconformidade, essa gerada quando o robô através dos sensores percebe o sinal de obstáculo.

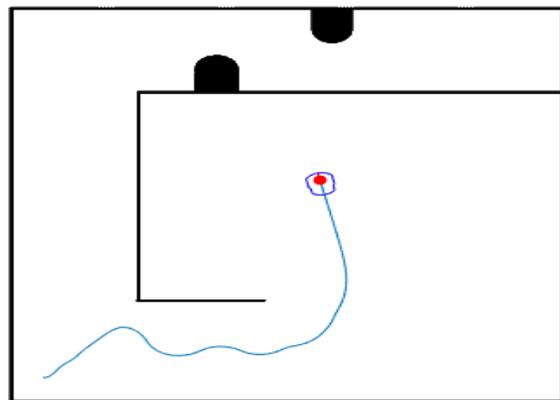


Figura 5 - Resultado final de trajeto - mapa2

Utilizando a mesma lógica inicial os efeitos no segundo ambiente também foram positivos, já que o objetivo foi alcançado, além disso percebeu-se que neste mapa o comportamento de trajetória apresentou

variações em razão do robô precisar realizar o contorno do obstáculo. Em outros mapas testados os resultados de percurso foram alcançados, inclusive foi realizado simulações com obstáculos móveis em alguns deles.

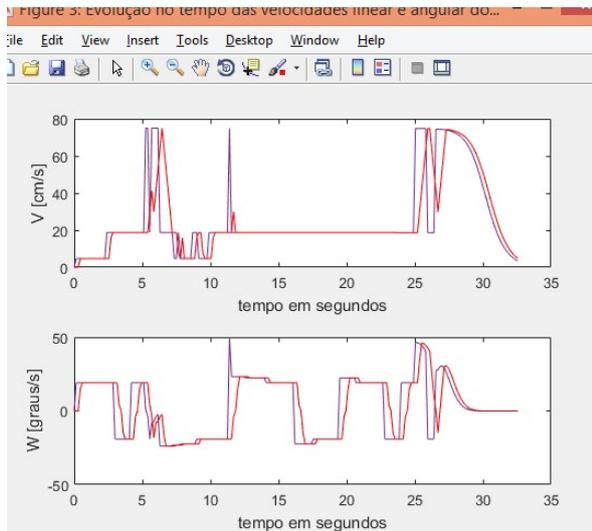


Figura 6 - Resultados da velocidade linear e angular, respectivamente (V e W) - mapa 02.

Semelhante ao comportamento gráfico de velocidades anterior, nota-se que realmente as linhas representativas de velocidade em relação aos pontos (X_o, Y_o) e (X_d, Y_d) tendem a ter um movimento bastante parecidos, tanto para o gráfico da velocidade angular quanto para velocidade linear. E por fim como descrito na programação ambas as velocidades começam a reduzir ao passo que o robô chega próximo ponto de destino, essa atitude ficou presente em todos os resultados obtidos nos ambientes.

CONCLUSÕES

Visto o que foi exposto, pode-se concluir que a metodologia de ensino para navegação reativa através de simulação computacional utilizando o MATLAB proporciona o aprendizado estratégico e funcional por meio de manipulações de variáveis, possibilitando assim, a visualização real do comportamento reativo do robô.

REFERÊNCIAS

STÜRMER, M.; HEINEN, F. J. Sistema de Controle e Navegação para Robôs Móveis Autônomos em Ambientes de Armazenagem. [S.l.], 2004.