

UM MÉTODO DE GERAÇÃO DE TRAJETÓRIA PARA ROBÔS NÃO-HOLONÔMICOS COM ACIONAMENTO DIFERENCIAL

NOMES DOS AUTORES*

* afiliação

Email: e-mail

Resumo— A geração de trajetória é um dos blocos essenciais de um planejador de movimento de robôs autônomos. Ela consiste em determinar uma relação entre um caminho e o tempo de forma que esta informação possa ser usada pelo sistema de controle do robô. Este artigo apresenta um método de geração de trajetória para ser utilizado por robôs não-holonômicos com acionamento diferencial. Este método baseia-se em amostrar o caminho não-holonômico, levando em consideração a máxima distância que o robô pode percorrer entre passos consecutivos de amostragem do sistema de percepção. Ele se mostra simples e computacionalmente rápido, o que permite a sua utilização em sistemas reais.

Palavras-chave— Geração de Trajetória, restrições não-holonômicas, robôs com acionamento diferencial.

Abstract— Trajectory generation is one of essential modulus of an autonomous robot motion planner. It consists in finding a relation between the path and time in order to be used by the robot control system. This paper shows a trajectory generation method applied to differential drive nonholonomic robots. This method is based on sampling the feasible path taking into account the maximum distance covered by the robot between two consecutive sampling steps of the robot perception system. It is simple and computationally quick, which permits applying it in actual systems.

Keywords— Trajectory generation, nonholonomic constraints, differential drive robot.

1 Introdução

Um dos grandes problemas de robôs autônomos é a implementação de um sistema eficiente para planejamento de movimentos. Tal sistema é responsável por definir os movimentos necessários para que um robô execute uma determinada tarefa. É um problema não-trivial cuja solução mais empregada é a sua subdivisão em vários subproblemas de menor complexidade. A figura 1 apresenta um planejador de movimentos básico utilizado para robôs móveis com a tarefa de navegação.

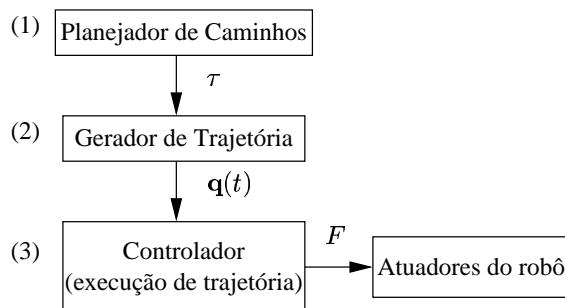


Figura 1: Diagrama de blocos de um planejador de movimentos para robôs móveis autônomos

O planejador de caminhos (nível 1) deve determinar um caminho τ entre duas configurações distintas de forma a evitar colisões com os obstáculos no espaço de trabalho. A maioria dos planejadores existentes foi compilada por Latombe (1991) e são denominados *planejadores geométricos* pois resumem-se a seqüências de semi-retas ligando a configuração inicial q_i até a final, q_f . O gerador de trajetória (nível 2) consiste em re-

lacionar as configurações do caminho obtido com o tempo ($q(t)$) ou, de outra forma, gerar perfis de velocidade para o robô. A geração de trajetória tem sido bastante abordada na literatura onde são propostos vários métodos aplicados para diferentes tipos de robôs (Fraichard, 1998; Elnagar and Hussein, 2000; Oriolo et al., 2000; Milam et al., 2000; Pledgie et al., 2002). Por fim, o nível 3 é o sistema de controle, responsável pela execução da trajetória e que gera um conjunto de forças e torques (F) para os atuadores do robô de modo que ele possa deslocar-se no ambiente.

Robôs móveis de uma maneira geral apresentam restrições não-holonômicas que restrigem alguns dos seus movimentos. Isto implica na adição de um novo bloco ao planejador de movimento, o qual deve situar-se entre o planejador de caminhos e o gerador de trajetórias. Ele é responsável por transformar um caminho puramente geométrico em outro que respeite às restrições cinemáticas existentes. Os caminhos de Reeds and Shepp (1990), compostos por concatenações de segmentos de reta e arcos de círculo de raio mínimo, são os exemplos clássicos de adaptadores. No caso de robôs com acionamento diferencial, como exemplificado na figura 2, tem-se que o raio mínimo de curvatura é zero, degenerando os caminhos de Reeds e Shepp nos seguintes movimentos:

1. uma rotação sobre o próprio eixo, até apontar para a posição final; e
2. um movimento em linha reta até alcançar a posição final; e
3. uma rotação para alcançar a orientação final.

Tais movimentos são apropriados para navegação em ambientes estáticos, porém podem re-

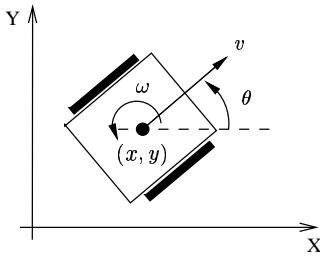


Figura 2: Exemplo de um robô com acionamento diferencial

presentar alguma desvantagem caso o robô atue em um ambiente dinâmico ou na perseguição a um alvo móvel. Alguns trabalhos foram propostos onde a geração de trajetória e a adaptação de caminhos são realizadas conjuntamente. Arakawa et al. (1995) e Khatib et al. (1997) utilizaram *splines*, como as Curvas de Bezier por exemplo. Outros utilizam clotóides geradas a partir das máximas velocidades linear e angular que podem ser desenvolvidas por um robô móvel, como foi proposto por Fleury et al. (1995), ou apresentaram idéias similares, como o trabalho de Choi and Kim (2001). Artigos mais recentes usam um método de geração de trajetória para robôs móveis baseado no cálculo da cinemática inversa de um manipulador (Mason and Burdick, 2002) ou propõe uma abordagem para a deformação de uma trajetória original com o objetivo de evitar colisões com obstáculos (Lamiraux and Bonnafous, 2002).

O enfoque principal deste artigo é a geração de trajetória onde é proposto um método para ser utilizado por robôs não-holonômicos com acionamento diferencial. A sua aplicação é voltada tanto para ambientes estáticos como para ambientes dinâmicos, atuando em conjunto com o adaptador de caminhos. Ele consiste em, primeiramente, obter um caminho adequado para o robô para, em seguida, amostrá-lo no tempo de acordo com a velocidade média desejada para cada período de amostragem do sistema de percepção do robô. Esta técnica é simples, computacionalmente rápida e gera configurações temporizadas para o sistema de controle empregado. Estas características são particularmente importantes para ambientes dinâmicos, onde a trajetória necessita ser constantemente recalculada, como um ambiente de futebol de robôs, por exemplo.

Uma explanação simplificada do método será apresentada na seção 2, seguindo-se uma abordagem teórica sobre o adaptador de caminhos utilizado (seção 3) e do gerador de trajetória proposto (seção 4). A seção 5 apresentará alguns resultados obtidos para um ambiente estático e, por fim, algumas conclusões e perspectivas para futuros trabalhos estão na seção 6.

2 Método Proposto

O método de geração de trajetória utilizado neste trabalho consiste na utilização conjunta de um adaptador de caminhos geométricos e de uma técnica que estabeleça a relação entre as configurações do caminho com o tempo. Geradores de caminhos geométricos são necessários especialmente para ambientes com obstáculos estáticos, entretanto tais caminhos isoladamente não são apropriados para um robô com acionamento diferencial, o qual apresenta uma restrição não-holonômica que força os seus movimentos a serem colineares com a direção das rodas. Considerando que a configuração do robô dada por $\mathbf{q} = (x, y, \theta)$, onde (x, y) denota a posição do centro de massa do robô e $\theta \in [-\pi, \pi]$ é o ângulo entre o vetor velocidade linear e o eixo horizontal, a restrição cinemática imposta é representada pela seguinte equação:

$$-\sin \theta \cdot \dot{x} + \cos \theta \cdot \dot{y} = 0 \quad (1)$$

Um caminho apropriado para um robô não-holonômico é definido como um conjunto de configurações ligando \mathbf{q}_i com \mathbf{q}_f , respeitando a equação 1. Um método bastante empregado consiste em gerar caminhos compostos por arcos de círculo de raio mínimo e segmentos de retas (Reeds and Shepp, 1990), que em robôs com acionamento diferencial implicam em movimentos de rotação em torno do próprio eixo seguidos por movimentos em linha reta. Isto implica que, em determinados instantes, as velocidades linear e angular do robô são simultaneamente iguais a zero. Para algumas situações este método não resolve completamente o problema da adaptação do caminho como, por exemplo, ambientes com obstáculos móveis. Um outro método alternativo foi proposto por Pedrosa et al. (2002). Ele tem a vantagem de ser simples e é determinado através de uma fórmula fechada, permitindo que um caminho entre dois pontos distintos possa ser calculado em tempo real além de impedir, sempre que possível, que as velocidades linear e angular do robô tornem-se nulas.

Uma vez que este caminho adequado tenha sido determinado, a etapa de geração de trajetória consiste em amostrá-lo no tempo onde leva-se em consideração a velocidade média desejada para o robô e o período de amostragem do sistema de percepção. Como resultado tem-se um gerador referências temporais para o sistema de controle. Uma apresentação formal das técnicas empregadas será realizada nas seções seguintes.

3 Caminhos Ponto-a-Ponto Não-Holonômicos

O adaptador de caminhos geométricos utilizado baseia-se em polinômios paramétricos de terceiro grau para representar a posição do centro de massa do robô. O ângulo de orientação é imposto

de forma a respeitar a restrição cinemática imposta (Pedrosa et al., 2002). Dessa forma, têm-se:

$$\begin{aligned} x(\lambda) &= a_0 + a_1 \lambda + a_2 \lambda^2 + a_3 \lambda^3 \\ y(\lambda) &= b_0 + b_1 \lambda + b_2 \lambda^2 + b_3 \lambda^3 \\ \theta(\lambda) &= \tan^{-1} \left(\frac{dy(\lambda)/d\lambda}{dx(\lambda)/d\lambda} \right) \end{aligned} \quad (2)$$

onde o parâmetro $\lambda \in [0, 1]$ é adimensional.

Considerando que $\mathbf{q}_i = (x_i, y_i, \theta_i)$ e $\mathbf{q}_f = (x_f, y_f, \theta_f)$ são conhecidas, pode-se determinar os valores dos coeficientes através da substituição destas condições de contorno nas equações 2. Como se obtém um sistema linear cujo número de incógnitas é maior que o de equações, arbitre-se dois valores dos coeficientes. Assim, a solução final para a maioria das situações é:

$$\begin{aligned} a_0 &= x_i \\ a_1 &\Rightarrow \text{arbitrado} \\ a_2 &\Rightarrow \text{arbitrado} \\ a_3 &= \Delta x - a_2 - a_1 \\ b_0 &= y_i \\ b_1 &= d_i a_1 \\ b_2 &= 3(\Delta y - d_f \Delta x) + d_f a_2 - 2(d_i - d_f)a_1 \\ b_3 &= 3d_f \Delta x - 2\Delta y - d_f a_2 - (2d_f - d_i)a_1 \end{aligned}$$

onde $d_i = \tan \theta_i$, $d_f = \tan \theta_f$, $\Delta x = x_f - x_i$ e $\Delta y = y_f - y_i$.

Caso seja adotado algum critério para melhorar o caminho entre \mathbf{q}_i e \mathbf{q}_f , pode-se definir um intervalo de valores para os coeficientes arbitrados. Assim, impondo que entre o intervalo aberto $\lambda \in (0, 1)$, o caminho seja regular sempre que possível, ou seja, sem pontos de valor máximo ou mínimo nas direções x e/ou y , dois sistemas de inequações envolvendo os coeficientes arbitrados são obtidos. Suas soluções garantem que o critério imposto seja atendido. Assim, para que o caminho seja regular na direção x , tem-se o sistema:

$$\begin{cases} a_1 \geq 0 \\ a_2 \geq -a_1 \\ a_2 \leq 3\Delta x - 2a_1 \end{cases}$$

que apresenta como uma possível solução:

$$a_1 = \Delta x \quad a_2 = 0$$

Para a direção y o sistema de inequações é:

$$\begin{cases} d_i a_1 \geq 0 \\ d_f a_2 \geq (d_i - 2d_f)a_1 + 3(d_f \Delta x - \Delta y) \\ d_f a_2 \leq 3d_f \Delta x - 2d_f a_1 \end{cases}$$

onde uma solução é:

$$a_1 = \frac{\Delta y}{d_i} \quad a_2 = 3\Delta x - \frac{(2d_f + d_i)\Delta y}{d_i d_f}$$

Para alguns arranjos de \mathbf{q}_i e \mathbf{q}_f estes dois sistemas têm soluções em comum, o que permite satisfazer o critério imposto em ambas as direções simultaneamente. Porém, para outros arranjos não há uma solução comum. Assim, deve-se escolher em qual direção o critério tem que ser atendido. Um modo de realizar esta escolha é verificar em qual direção o caminho terá menor comprimento. A figura 3 mostra alguns caminhos obtidos entre dois pontos distintos com orientações diversas.

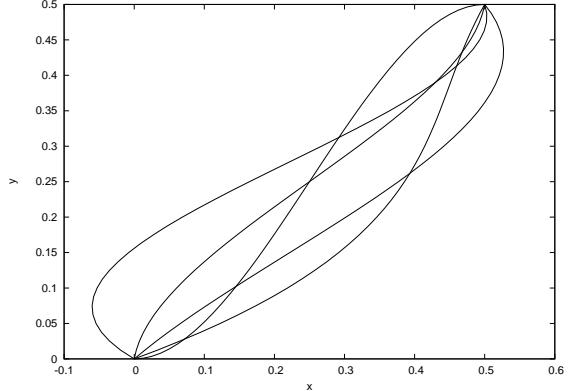


Figura 3: Exemplos de caminhos não-holonômicos entre duas configurações distintas

4 Geração de Trajetória

Uma vez obtido o caminho não-holonômico, deve-se utilizar algum método para expressar as configurações encontradas em função do tempo. Nesta etapa é necessário observar em que tipo de ambiente o robô efetuará os seus movimentos e se a trajetória deve ser calculada em tempo real, ou não. Para ambientes estáticos ou ambientes livres de obstáculos mas com configuração final fixa, normalmente calcula-se previamente a trajetória antes de enviá-la para o sistema de controle, porém não se exclui a possibilidade de realizar este cálculo em tempo real. Para ambientes dinâmicos ou em tarefas de perseguição de um alvo móvel, a trajetória deve, necessariamente, ser recalculada em períodos constantes de tempo. Neste trabalho, a trajetória é calculada em tempo real tanto em ambientes estáticos quanto em ambientes dinâmicos, cuja abordagem será apresentada a seguir.

4.1 Cálculo da Trajetória em Tempo Real

A necessidade da geração da trajetória ser calculada em tempo real se dá pela dinamicidade de ambientes povoados por obstáculos móveis ou pela perseguição de um alvo móvel pelo robô. Uma trajetória estática não é desejável porque ela pode ser interceptada por algum obstáculo, provocando uma colisão indesejada, ou no caso de uma perseguição, porque a distância entre o robô e o alvo

sempre é variável. Em ambos os casos torna-se imprescindível que a trajetória seja periodicamente calculada a cada passo de amostragem do sistema de percepção do robô. O método proposto para esta geração em tempo real é simples e baseia-se em duas características principais:

1. o caminho não-holonômico entre a configuração atual do robô (\mathbf{q}_r) e \mathbf{q}_f deve ser recalculada periodicamente; e
2. a distância percorrida pelo robô entre dois passos consecutivos é conhecida.

O cálculo, também em tempo real, do caminho não-holonômico não sofre qualquer tipo de restrição pois o método é simples e possui uma fórmula fechada para a sua determinação, tornando-o bastante apropriado. Para a distância percorrida durante um intervalo de amostragem, deve-se considerar a velocidade média desejada v_d para o movimento e o período de amostragem Δt do sistema de percepção. Assim, determina-se a distância percorrida α através de $\alpha = v_d \Delta t$.

Um maneira de mapear λ e t é determinar o valor desejado do parâmetro para que, no próximo passo de amostragem, o robô esteja a uma distância α de sua posição atual. Para isso, considera-se que a função parametrizada do comprimento do caminho é dada por $s(\lambda)$ através de:

$$s(\lambda) = \int_0^\lambda \sqrt{\frac{dx(\gamma)}{d\gamma}^2 + \frac{dy(\gamma)}{d\gamma}^2} d\gamma \quad (3)$$

Assim, para o próximo passo de amostragem, o λ desejado é determinado através da resolução da seguinte equação:

$$s(\lambda) - \alpha = 0 \quad (4)$$

Porém, a equação 3 não tem uma função elementar primitiva conhecida, considerando que $x(\lambda)$ e $y(\lambda)$ são dados pelas equações 2, mas a evolução do comprimento do caminho é monotonicamente crescente entre o intervalo $\lambda \in [0, 1]$. Esta característica permite aproximar a função $s(\lambda)$ nas vizinhanças da origem pela fórmula expandida de Taylor, dada por:

$$s(\lambda) = s(0) + s'(0)\lambda + \frac{s''(0)}{2!}\lambda^2 + \frac{s'''(0)}{3!}\lambda^3 + \dots \quad (5)$$

onde:

$$s^i(0) = \left. \frac{d^i s(\lambda)}{d\lambda^i} \right|_{\lambda=0}$$

para $i = 1, 2, 3, \dots$. Sem perda de generalidade, pode-se truncar esta equação em λ^3 , pois para exponents maiores os termos tendem a zero. Também sempre considera-se que $s(0) = 0$. Derivando a equação 3 sucessivamente, os coeficientes

da equação 5 podem ser determinados completamente. Assim, têm-se:

$$\begin{aligned} \beta_1 &= s'(0) = \sqrt{a_1^2 + b_1^2} \\ \beta_2 &= s''(0) = \frac{2(a_1 a_2 + b_1 b_2)}{\beta_1} \\ \beta_3 &= s'''(0) = \frac{4(a_2^2 + b_2^2) + 6(a_1 a_3 + b_1 b_3)}{\beta_1} - \frac{\beta_2^2}{\beta_1} \end{aligned}$$

Reescrevendo a equação 4, tem-se:

$$s(\lambda) = \beta_1 \lambda + \frac{\beta_2}{2!} \lambda^2 + \frac{\beta_3}{3!} \lambda^3 \quad (6)$$

Como esta equação descreve a evolução da distância com relação à origem e é uma aproximação da equação 3 nas vizinhanças de $\lambda = 0$, seu desempenho será melhor ou pior dependendo dos arranjos de \mathbf{q}_i e \mathbf{q}_f . Isto pode ser visualizado nos gráficos da figura 4 que mostram uma comparação entre as evoluções das equação 6 e equação 3, por integração numérica. Em ambos os gráficos há uma aproximação das curvas nas vizinhanças de $\lambda = 0$, mas no gráfico superior as curvas distoam à medida que o valor de λ cresce.

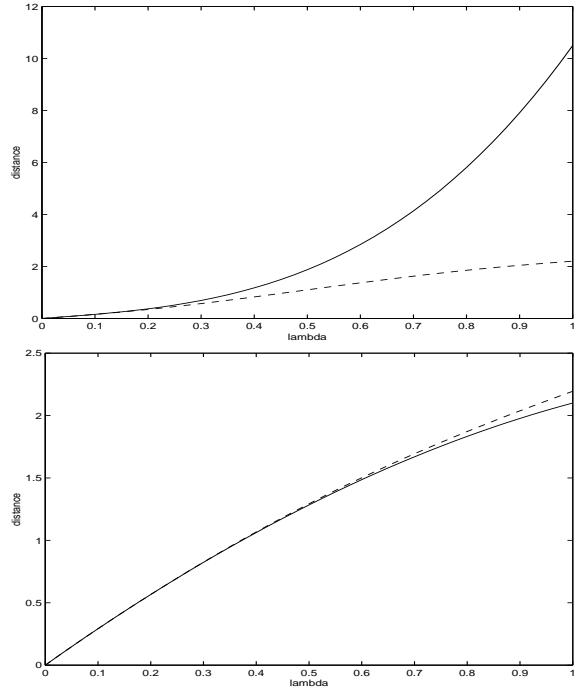


Figura 4: Exemplos de similaridade entre o comprimento do caminho definido por polinômio cúbico (—) e comprimento do caminho calculado por integração numérica (- -)

Como a equação 6 aproxima-se da equação 3 de maneira apropriada na origem, então a equação 4 pode ser solucionada uma vez que a distância percorrida é suficientemente curta permitindo que variações de λ próximas de zero sejam capazes de representá-la. Neste trabalho optou-se pela

resolução numérica utilizando o método da Bissecção. Desse modo, a cada passo de amostragem, é possível calcular o valor desejado do parâmetro e calcular a próxima configuração que será enviada para o sistema de controle do robô como a nova referência. Explicitando o método de forma algorítmica, tem-se que:

1. calcular os coeficientes de $x(\lambda)$ e $y(\lambda)$ levando em consideração as configurações \mathbf{q}_r e \mathbf{q}_f ;
2. calcular os coeficientes da equação 6;
3. calcular o valor desejado do parâmetro (λ_d) para o próximo passo de amostragem através da resolução da equação 4; e
4. por fim, calcular a próxima configuração desejada $\mathbf{q}_d = [x(\lambda_d), y(\lambda_d), \theta(\lambda_d)]$, através das equações 2, para ser utilizada como referência pelo sistema de controle do robô.

Este procedimento faz com que o adaptador de caminhos e o gerador de trajetórias trabalhem em um laço fechado, juntamente com o sistema de controle. A figura 5 apresenta uma trajetória calculada em tempo real durante a perseguição de um alvo móvel, representado pela linha tracejada, onde a posição do robô é representada por \circ .

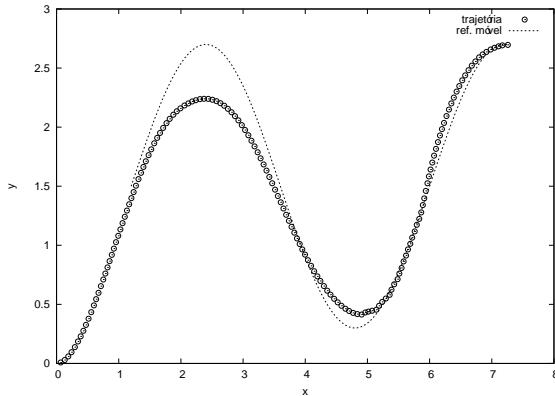


Figura 5: Exemplo de trajetória calculada em tempo real na perseguição de um alvo móvel

5 Exemplo de Trajetória

Para testar a aplicabilidade do método de geração de trajetória apresentado foi utilizado um exemplo de navegação de um robô móvel em um ambiente estático fechado. Foram executadas as etapas de planejamento de caminho, adaptação de caminho e geração de trajetória. A fase de execução de trajetória não foi abordada neste trabalho. O ambiente de trabalho em questão é povoado com quatro obstáculos poligonais. Para a determinação do caminho geométrico utilizou-se o método da decomposição exata de células (Latombe, 1991). A figura 6 mostra o caminho obtido.

O passo seguinte foi a adaptação do caminho geométrico em um não-holonômico (Pedrosa et al., 2002). O resultado é mostrado na figura 7.

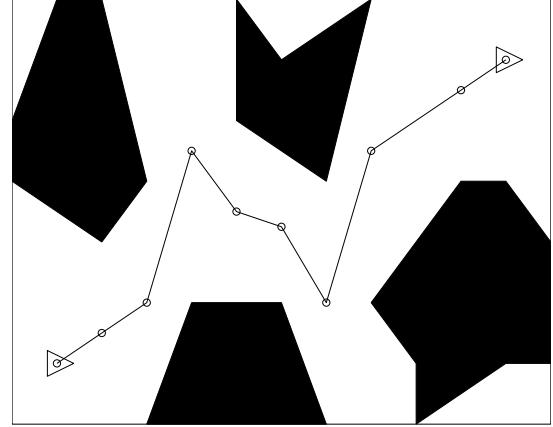


Figura 6: Caminho geométrico obtido com a decomposição do ambiente em células exatas

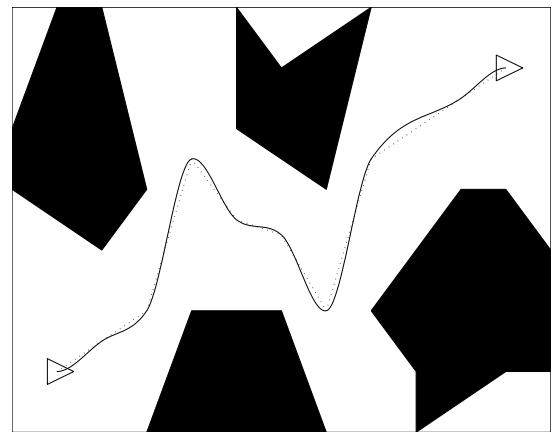


Figura 7: Caminho não-holonômico obtido com o método dos polinômios cúbicos

Por fim, a última etapa foi a geração de trajetória. Como valor da velocidade média desejada foi adotado $v_d = 2m/s$ e para o intervalo de amostragem adotou-se $\Delta t = 0,033s$, que é o valor empregado nos sistemas de visão de futebol de robôs. O resultado é apresentado na figura 8.

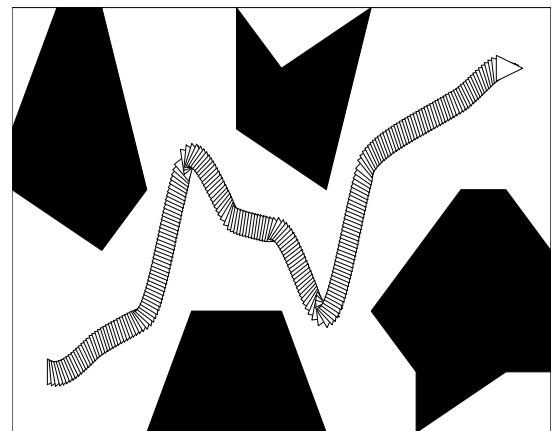


Figura 8: Trajetória calculada em tempo real

6 Conclusões

Este trabalho apresentou um método de geração de trajetória que pode ser utilizado por robôs não-holonômicos com acionamento diferencial. A técnica proposta atua em conjunto com um adaptador de caminhos geométricos. A idéia principal é amostrar, no tempo, o caminho não-holonômico levando em consideração a velocidade média desejada para que o robô execute todo o movimento planejado e repetir este procedimento a cada passo de amostragem do sistema de percepção do robô. Sua aplicabilidade é extensível para ambientes estáticos e dinâmicos. A abordagem mostrou-se bastante simples, computacionalmente rápida e, portanto, adequada para sistemas reais. Exemplos de trajetória geradas foram apresentados na perseguição de um alvo móvel e em um ambiente estático. Futuros trabalhos concentrarão-se em novos geradores de trajetória que possam ser aplicados para outros tipos de robôs (semelhantes a carros, robôs com *trailers*, etc.) além do projeto de controladores apropriados para sistemas com restrições não-holonômicas, permitindo assim a construção completa de um planejador de movimento.

Agradecimentos

Diogo Pedrosa agradece à Capes pelo suporte financeiro.

Referências

- Arakawa, A., Hiyama, M., Emura, T. and Kagaumi, Y. (1995). Trajectory generation for wheeled mobile robot based on landmarks, *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, Canada.
- Choi, J.-S. and Kim, B. K. (2001). Near-time-optimal trajectory planning for wheeled mobile robots with translational and rotacional sections, *IEEE Transactions on Robotics and Automation* **17**(1).
- Elnagar, A. and Hussein, A. M. (2000). On optimal constrained trajectory planning in 3d environments, *Robotics and Autonomous Systems* **33**.
- Fleury, S., Souères, P., Laumond, J.-P. and Chatila, R. (1995). Primitives for smoothing mobile robot trajectories, *IEEE Transactions on Robotics and Automation* **11**(3).
- Fraichard, T. (1998). Trajectory planning amidst moving obstacles: Path-velocity decomposition revisited, *Journal of the Brazilian Computer Society* **4**(3). Special issue on robotics.
- Khatib, M., Jaouni, H., Chatila, R. and Laumond, J.-P. (1997). Dynamic path modification for car-like nonholonomic mobile robots, *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, USA.
- Lamiraux, F. and Bonnafous, D. (2002). Reactive trajectory deformation for nonholonomic systems: Application to mobile robots, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA.
- Latombe, J.-C. (1991). *Robot Motion Planning*, Kluwer Academic Press.
- Mason, R. and Burdick, J. W. (2002). Trajectory planning using reachable-state density functions, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA.
- Milam, M. B., Mushambi, K. and Murray, R. M. (2000). A new computational approach to real-time trajectory generation for constrained mechanical systems, *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia.
- Oriolo, G., Panzieri, S. and Ulivi, G. (2000). Learning optimal trajectories for nonholonomic systems, *International Journal of Control* **73**(10).
- Pedrosa, D. P. F., Medeiros, A. A. D. and Alsina, P. J. (2002). Geração de caminhos ponto a ponto para robôs móveis com rodas, *Anais do XIV Congresso Brasileiro de Automática - CBA*, Natal, RN, Brasil.
- Pledgie, S. T., Hao, Y., Ferreira, A. M., Agrawal, S. K. and Murphey, R. (2002). Groups of unmanned vehicles: Differential flatness, trajectory planning, and control, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA.
- Reeds, J. A. and Shepp, L. A. (1990). Optimal paths for a car that goes both forwards and backwards, *Pacific Journal of Mathematics* **145**(2).