

IMPLEMENTAÇÃO DE ESTRATÉGIAS DE CONTROLE E PLANEJAMENTO DE TRAJETÓRIA EM ROBÔS MÓVEIS TERRESTRES BASEADAS EM CAMPOS VETORIAIS ARTIFICIAIS

LÍVIA DALFIOR SOSSAI*, TALES ARGOLO JESUS*

* *Departamento de Computação - DECOM Centro Federal de Educação Tecnológica de Minas Gerais
Avenida Amazonas, n.7675, CEP: 30510-000 Belo Horizonte - Minas Gerais, Brasil*

Emails: liviadalfior@gmail.com , talesargolo@decom.cefetmg.br

Abstract— This work consists of the implementation of a navigation system using some of the main trajectory planning strategies based on artificial potential fields on LEGO® Mindstorms NXT platform. The planning strategies covered by this work are: Potential Fields, Navigation Functions and Wavefront Planner. In order to implement these strategies on a differential wheeled robot, two cascaded closed-loop controllers were necessary. The first, external loop is responsible for correcting the steering error, whose reference signals are generated by the planning strategy. The second, inner and faster loop controls the wheels' velocity. The reference values originate from the first control loop. The robot's localization was estimated via odometry. The strategies were then compared using a set of metrics. They were utilized to evaluate the navigation system's behavior and the trajectory performed by each of the planning strategies. Besides, numerical issues involving the Navigation Functions lead to limitations when implementing such trajectory planning technique.

Keywords— mobile robots, navigation functions, control, trajectory planning, differential drive

Resumo— Neste trabalho são implementadas algumas das principais técnicas de planejamento de trajetória baseadas em campos vetoriais artificiais na plataforma LEGO® Mindstorms NXT. São elas: Campos Potenciais, Funções de Navegação e *Wavefront Planner*. O robô utilizado é constituído de duas rodas com tração diferencial e o sistema de controle consiste de duas malhas de controle em cascata: a primeira para controle de velocidade das rodas e, a segunda, para o controle de orientação do robô. O sinal de referência é definido pela estratégia de navegação, cujo cálculo é realizado utilizando-se a posição do robô, que é estimada via odometria. As estratégias foram comparadas segundo um conjunto de métricas específicas e em diferentes cenários, que permitiram avaliar o comportamento do robô ao realizar a trajetória planejada por cada uma das técnicas. Além disso, foram evidenciados problemas relacionados com precisão numérica, ao se implementar as Funções de Navegação na plataforma experimental.

Palavras-chave— robôs móveis, navegação, controle, planejamento de trajetórias, robô diferencial

1 Introdução

A locomoção autônoma de robôs terrestres é uma característica desejada principalmente quando se trata de ambientes que apresentam riscos a seres humanos. Navegar com relativa autonomia, neste contexto, significa partir de um ponto de origem e chegar a um destino sem colisões com obstáculos e considerando irregularidades presentes no ambiente. Além disso, entre a origem e o destino podem existir inúmeras possibilidades de percurso.

O presente trabalho apresenta um estudo comparativo entre três técnicas de planejamento de trajetória do ponto de vista da implementação em um robô não-holonômico de tração diferencial na plataforma comercial LEGO Mindstorms® NXT. Comparações entre técnicas de planejamento de trajetória são realizadas em (Gómez et al., 2013), em que Campos Potenciais são comparados com o Diagrama de Voronoi. Neste trabalho tem-se o objetivo de explorar a implementação e controle de algumas das estratégias clássicas de planejamento de trajetória, comparando-as em relação a um conjunto de métricas que envolve a complexidade do cenário, esforço de controle, tempo e espaço percorrido para realizar a trajetória até o alvo. A questão do controle será uma métrica fundamental, uma vez que as estratégias

focam em guiar o robô até o alvo, mas podem acabar por deixar a qualidade do rastreamento em plano secundário (Park and Kuipers, 2011).

A primeira técnica a ser estudada são os Campos Potenciais (CPs) inicialmente descritas por (Khatib, 1986). Apesar de simples e largamente utilizadas, a técnica apresenta limitações que são tratadas experimentalmente por (Koren and Borinstein, 1991). Além do conhecido mínimo local, responsável pelo comportamento cíclico, o estudo evidencia questões como as dificuldades de o robô passar por entre obstáculos próximos, bem como oscilações causadas nessas situações. No trabalho de (Urakubo, 2015), os CPs são estudados por meio da implementação em um robô diferencial. Existem, ainda, trabalhos que implementam os CPs em uma plataforma experimental por meio do uso de visão computacional para localização do robô (Poomarin et al., 2016).

A segunda técnica de planejamento a ser explorada neste trabalho são as Funções de Navegação (FN). Elas constituem uma extensão dos campos potenciais de forma a contornar a questão dos mínimos locais. Além disso, o uso de Funções de Navegação constitui uma maneira elegante de se projetar controladores (Colledanchise et al., 2013).

A última estratégia a ser explorada é o *Wave-*

front Planner que consiste em uma heurística responsável por discretizar o espaço de trabalho em um *grid* e propagar direções em cada célula. Em (Alves et al., 2013) o WF, associado a técnicas de localização baseadas em filtros, provê um sistema em tempo real para que a plataforma experimental possa se deslocar em um ambiente como um escritório. Em (Arantes et al., 2010) a técnica WF é associada à teoria de Autômatos Celulares e sensores para a navegação de um robô com tração diferencial, bem como o mapeamento de novas áreas.

2 Explorando a não-holonomia em um robô diferencial

Robôs não-holonômicos possuem como característica principal o fato de possuírem restrições que os impedem de alterar sua direção de deslocamento instantaneamente. Isso porque o espaço de velocidades em uma configuração qualquer tem dimensão menor do que o número de graus de liberdade do robô. O espaço de configuração do robô é dado pelas variáveis x e y que representam as coordenadas em que o robô se encontra em um determinado instante de tempo t e θ representa a orientação do robô. As variáveis ω_d e ω_e representam a velocidade angular das rodas direita e esquerda, respectivamente. O modelo em espaço de estados do robô diferencial é dado por¹:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

sendo v a velocidade linear do robô e ω , a velocidade angular. Elas dependem dos parâmetros L e r que representam a distância entre as rodas e o raio das rodas e são calculadas da seguinte forma:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2}(\omega_e + \omega_d) \\ \frac{r}{L}(\omega_d - \omega_e) \end{bmatrix}$$

3 Sobre a plataforma LEGO®

A plataforma LEGO Mindstorms® consiste nas peças para montagem do robô (rodas, eixos, etc.) e em um computador embarcado, onde é incluída a programação. A programação dos blocos foi realizada na linguagem NXC - *Not Exactly C* e os códigos foram compilados com o compilador NBC. Assim como em (Benitez et al., 2008), a plataforma experimental LEGO® foi escolhida por

¹Neste trabalho adotou-se o uso das estratégias de navegação no espaço de trabalho, e não como convencionalmente se faz no espaço de configurações. É uma escolha razoável, já que o espaço de trabalho, no caso de um robô diferencial não-holonômico, é um subconjunto do espaço de configurações.

ser flexível e permitir diversas montagens, além da sua fácil disponibilidade em instituições de ensino. Devido ao uso de funções matemáticas mais complexas, como a tangente hiperbólica e o arco tangente (para determinação da direção do gradiente), foi necessário o uso do *Enhanced NBC/NXC firmware*².

4 Arquitetura Proposta

4.1 Malhas de controle

As técnicas de planejamento de trajetória fornecem valores de referência para a velocidade angular (ω) e linear (v). Em cada ponto do espaço de trabalho do robô podem-se observar valores diferentes para v e ω . Esses valores de referência são transformados em valores de referência para a velocidade angular das rodas. Tal mapeamento é dado pelas equações abaixo:

$$\omega_d = \frac{2v + \omega L}{2r} \quad \text{e} \quad \omega_e = \frac{2v - \omega L}{2r}. \quad (2)$$

Corrigir a orientação do robô, ou seja, o ângulo θ , significa converter esse erro em uma velocidade angular que alinhe o robô à orientação de referência θ_{ref} . Na Figura 1 são exibidas as malhas de controle necessárias para a implementação do sistema de navegação do robô. Nela são destacadas duas malhas em cascata: a malha de controle de orientação, responsável pela geração de valores de referência para a segunda malha, que é responsável pelo controle das velocidades das rodas. Isso é importante para garantir que fatores como o peso do robô, carga da bateria, mudanças na superfície do espaço de trabalho, entre outros não influenciem significativamente a velocidade alcançada. Essa malha dispensa, por exemplo, o mapeamento entre valores PWM e a velocidade observada para o valor, o que estaria sujeito aos fatores mencionados. Existe, ainda, uma realimentação da posição para o bloco que representa as estratégias de navegação, para que θ_{ref} possa ser calculada naquele ponto do plano xy .

4.2 Implementação na plataforma LEGO Mindstorms® NXT

Para a criação de uma arquitetura portátil e flexível é necessário dividir em blocos as principais funções do sistema de controle e navegação, bem como estabelecer tarefas bem definidas para cada um deles. As estratégias de navegação, juntamente com o modelo matemático do robô diferencial fornecem valores de referência para velocidade das rodas. É importante também controlar a velocidade das rodas e garantir que elas cheguem de

²Informações técnicas a respeito dos aprimoramentos realizados nessa versão de *firmware* podem ser encontradas na página <http://bricxcc.sourceforge.net/firmware.html>

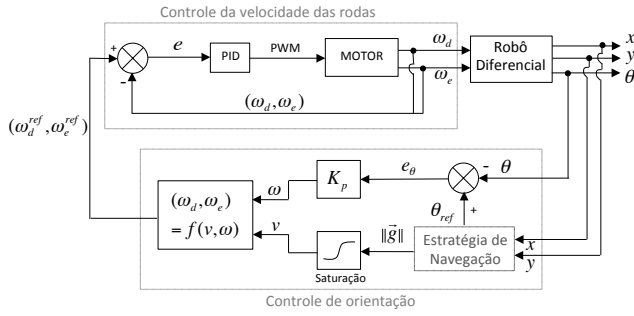


Figura 1: Diagrama de blocos do sistema de controle e navegação.

maneira rápida e se mantenham, com o mínimo de oscilações na velocidade de referência. Controlar a velocidade das rodas implica a necessidade de se medi-las. Essa medição também é necessária para se estimar a posição do robô, uma vez que, no presente trabalho, a localização é realizada por meio de odometria.

Para que cada um dos módulos pudesse ser implementado de forma independente na plataforma LEGO®, foi explorado o recurso denominado *task* presente na linguagem NXC (Hansen, 2007). Uma *task* ou tarefa permite que determinados trechos de programa possam ser executados de maneira concorrente. Essa simultaneidade tem grande utilidade, uma vez que o robô se move ao mesmo tempo em que o sensor mede a velocidade, a velocidade nas rodas é controlada, a posição estimada e a orientação de referência é calculada a partir das coordenadas e da velocidade do robô. Esse recurso, se usado da maneira correta, pode melhorar o desempenho do robô no que diz respeito principalmente ao tempo para se chegar ao alvo e agilidade para desviar de obstáculos.

5 Estratégias de Planejamento

O planejamento de trajetória tem um importante papel no desempenho de um robô real. Isso se torna ainda mais evidente ao se tratar robôs com restrições não-holonômicas, como é o caso tratado neste trabalho. Trajetórias menos suaves podem levar a um esforço de controle maior ou, ainda, podem não ser suficientes para evitar colisões com os obstáculos. Trajetórias muito conservadoras podem fazer com que o espaço a ser percorrido seja muito maior até o alvo, do que trajetórias menos conservadoras. Torná-las menos conservadoras pode, em contrapartida, fazer com que os robôs se aproximem demais dos obstáculos, o que pode levar a colisões. Um fator importante que deve ser considerado ao implementar as estratégias de planejamento em robôs reais é a saturação dos motores: a trajetória planejada deve fornecer valores de referência que respeitem as limitações do robô.

5.1 Campos Potenciais

Os campos potenciais se baseiam na existência de um potencial atrativo $U_{atrativo}$ e um repulsivo, $U_{repulsivo}$. O potencial atrativo deve obedecer o critério de ser monotonicamente crescente à medida em que a distância em relação ao alvo aumenta (Choset et al., 2005). Outras características desejáveis para a função que caracteriza o potencial atrativo são a continuidade e diferenciabilidade, além de um gradiente cujo módulo decresça à medida em que o alvo se torna mais próximo. Uma maneira simples de se determinar uma função de potencial para a ação de ir ao alvo (potencial atrativo) é considerando-se uma função que cresça quadraticamente com a distância até o alvo.

O direcionamento do robô será dado, então, pelo caminho gerado pela direção oposta ao gradiente da função. Um problema que pode decorrer do emprego dessa técnica é a existência de um mínimo local, o que pode levar o robô a não se mover ou fazer com que haja um comportamento cíclico.

5.2 Funções de Navegação

As Funções de Navegação, diferentemente dos campos potenciais, codificam as ações de ir até o alvo e desviar de obstáculos em uma só função. Além de fornecerem uma função cujo campo abrange as ações esperadas para o robô, as Funções de Navegação também solucionam a questão do mínimo local. Essas funções, descritas inicialmente por (Rimon and Koditschek, 1992) possuem somente um mínimo global e são funções da distância em relação ao obstáculos. Podem, inclusive, ser aplicadas a espaços n dimensionais (Choset et al., 2005). Essas funções podem também ser aplicadas a espaços *star-shaped* e não-euclidianos. Entretanto, o escopo deste trabalho se limitou a tratar espaços circulares e espaços bi-dimensionais. Neste caso, os obstáculos também são modelados como círculos. As informações do ambiente em que o robô se encontra são codificadas da seguinte maneira:

$$\mathcal{QO}_i = \{q | \beta_i(q) \leq 0\}, \quad (3)$$

sendo $\mathcal{Q}$ a função referente ao obstáculo $\mathcal{O}_i$. Assim, β_i é negativo dentro dos obstáculos, zero nos limites dos obstáculos e positivo fora deles.

Todos os obstáculos são incluídos na composição da ação repulsiva. Esses obstáculos, bem como os limites do espaço são reunidos em um produto, como é possível observar na Equação 4:

$$\beta(q) = \prod_{i=0}^n \beta_i(q) \quad (4)$$

É necessário considerar o raio e centro do espaço para assegurar que o robô não se afaste demais,

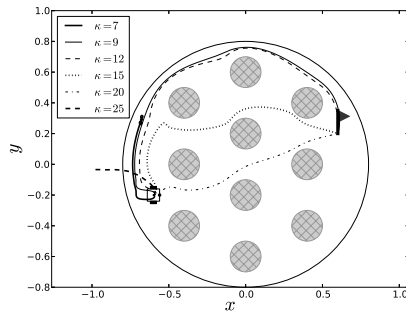


Figura 2: Simulação numérica da trajetória do robô modelado como unicycle cinemático para diferentes valores de κ .

ao se desviar de um obstáculo. O centro e o raio do espaço são dados por q_0 e r_0 .

As FNs possuem o parâmetro κ , que é responsável pela suavidade da função ao redor dos obstáculos. Quanto maior o valor do parâmetro κ , mais abrupta será a variação da curva em torno de um obstáculo e, conseqüentemente, o robô chegará mais próximo deles. A influência do parâmetro κ na trajetória pode ser visualizada na Figura 2.

Sendo $d(q, q_{alvo})$ a distância entre a coordenada q e o alvo, a FN é dada por:

$$\rho(q) = \frac{d^2(q, q_{alvo})}{[(d(q, q_{alvo}))^{2\kappa} + \beta(q)]^{\frac{1}{\kappa}}} \quad (5)$$

5.3 Wave-front Planner

O algoritmo Wave-front Planner, inicialmente descrito em (Barraquand et al., 1991), tem por objetivo mapear o ambiente real em que o robô se encontra em um espaço discretizado. O conceito central dessa discretização são os chamados *grids*. Inicialmente cada uma das células desse *grid* é rotulada em 0, para o caso de não conterem partes de obstáculos (ou seja, regiões livres) e 1, quando são partes de obstáculos.

Após o mapeamento entre plano e *grid* é realizada então a propagação para que as demais células sejam classificadas. O valor contido em cada célula (após a aplicação do algoritmo) é considerado o valor do gradiente dentro de uma determinada região delimitada pela célula em questão. Considerando-se um *grid* bidimensional, a propagação pode ser feita considerando-se quatro ou oito células vizinhas por meio de uma busca em largura. Na Figura 3 são exibidas duas iterações do processo de propagação considerando-se quatro elementos adjacentes.

Regra 1 Os vizinhos de uma célula cujo gradiente seja n e que possuam valor 0, ou seja, não foram classificadas, receberão o valor $n + 1$.

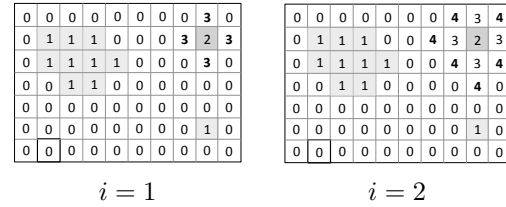


Figura 3: Duas iterações da propagação dos valores utilizando-se uma vizinhança de quatro elementos.

O algoritmo *Wave-Front* possui como característica o fato de não possuir pontos de mínimo local. Além disso, segue o menor trajeto entre a coordenada inicial e o alvo. Entretanto, isso ocorre sob o custo de o robô se deslocar a uma distância consideravelmente curta dos obstáculos (Choset et al., 2005). Um outro ponto que merece destaque é a complexidade computacional associada ao tamanho do ambiente em questão. Relacionado a esse aspecto pode-se também mencionar o fato de que o planejamento da trajetória é realizado por meio de uma busca em todo o espaço, o que representa um custo considerável, principalmente se o ambiente e a quantidade de células do espaço discretizado for muito grande.

6 Resultados Experimentais

O planejamento de trajetória foi realizado considerando-se dois tipos de cenário: (1) um paralelepípedo entre o robô e o alvo. A orientação inicial do robô corresponde à inclinação da reta que contém a posição inicial do robô e o alvo, ou seja, $\frac{\pi}{4}$. (2) seis obstáculos distribuídos pelo cenário e relativamente próximos uns dos outros. Além disso, a disposição dos obstáculos apresenta bifurcações, de forma a permitir diferentes trajetos até o alvo.

Para comparar os resultados quanto à trajetória realizada, pode-se recorrer à Figura 4. Nela podem-se visualizar imagens estroboscópicas geradas a partir da sobreposição de quadros que representam o estado do robô a cada 2s. Assim, também é possível observar a velocidade desenvolvida ao longo da trajetória. Além disso, a curva que sobrepõe a imagem representa a posição estimada pela odometria. Na Tabela 1 são apresentados valores numéricos referentes ao tempo total (t_{total}) para chegar até o alvo, a distância total percorrida (D_{total}), o erro quadrático médio ($MSE(\theta)$) de orientação e a distância entre a posição final do robô.

No primeiro cenário, observa-se uma trajetória mais agressiva para os CPs. Isso ocorre devido ao fato de que, dentro dos limites do círculo passa a atuar somente o potencial repulsivo. Além disso, nos instantes que antecedem a chegada aos limites do círculo, a velocidade está próxima ao seu

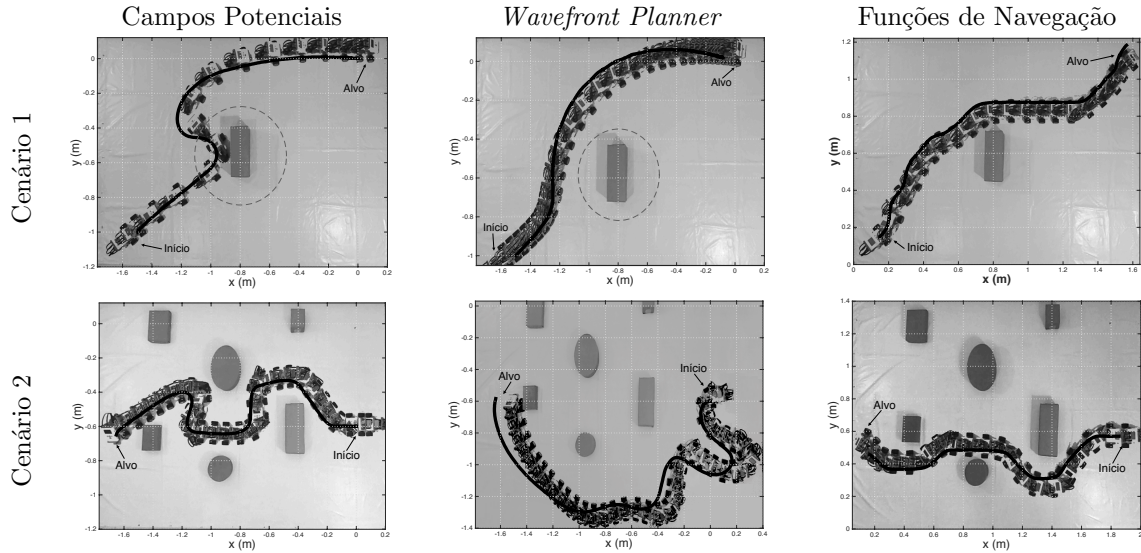


Figura 4: Resultados experimentais das t cnicas de planejamento em diferentes tipos de cen rio.

limite m ximo ($0,1m/s$), devido   dist ncia em rela  o ao alvo. Essas duas condi  es s o respons veis pelo desvio realizado de maneira acentuada e agressiva, no que diz respeito   aproxima  o em rela  o ao obst culo. Isso   refletido nos valores num ricos: com os CPs, apesar de o tempo necess rio para concluir a tarefa ter sido o menor dentre as tr s estrat gias, o MSE foi o maior, assim como a dist ncia total, dado o esfor o necess rio para desviar de maneira brusca dos obst culos.

Para realizar o planejamento via WF empregou-se uma vizinhan a de oito elementos visando a uma trajet ria mais suave. Al m disso, v foi fixada em $0,05\text{ m/s}$. Apesar disso, o MSE foi mais alto, se comparado com as FNs. A trajet ria descrita via FN, por sua vez, foi a que apresentou maior suavidade, o que se reflete em seu MSE, que foi consideravelmente menor que o das outras duas estrat gias. Al m disso a velocidade desenvolvida foi mais baixa, o que favoreceu esse resultado. Apesar de apresentar o melhor resultado no que diz respeito   adequa  o do rob  n o-holon mico   trajet ria planejada e da dist ncia total percorrida, o tempo necess rio para chegar ao alvo foi o maior entre as tr s estrat gias.

Devido   natureza do cen rio 2, a velocidade linear v foi limitada em $0,05\text{ m/s}$ para todas as estrat gias, para evitar colis es. Al m disso, para os CPs, dentro dos limites dos c rculos que envolvem os obst culos, foi adicionada/ajustada uma componente respons vel pela a  o de ir ao alvo, de forma que $U_{resultante} = U_{repulsiva} + 0,5U_{atr tiva}$. O MSE que, no cen rio 1 era o maior dentre as tr s estrat gias, passou a ser o menor.

A FN, em contrapartida, apresentou o pior resultado para o cen rio 2 em todas as m tricas, para $\kappa = 5$. Esse resultado est  relacionado ao par metro empregado para κ . A Figura 2 representa a modelagem do cen rio 2 e foi empregada

para realizar o planejamento da trajet ria na plataforma LEGO . Na figura   poss vel observar a trajet ria gerada para diferentes valores de κ . Valores menores de κ fazem com que a trajet ria se afaste consideravelmente dos obst culos, procurando sempre os limites do cen rio. Isso ocorre porque o termo da equa  o (5) respons vel por codificar a a  o de ir ao alvo, ou seja $d(q, q_{alvo})^{2\kappa}$ se torne menor, se comparado ao produto $\beta(q)$, que codifica a a  o de desviar dos obst culos. Dessa forma, para o cen rio 2 o ideal seria o uso de valores maiores para κ .

Nesse ponto foi observada uma limita  o da implementa  o na plataforma/linguagem escolhida. Usar valores muito elevados para κ exige maior precis o, uma vez que $d(q, q_{alvo})^{2\kappa} \gg \beta(q)$ (ver equa  o (5)). Ao calcular a κ - sima raiz da soma desses termos, h  uma limita  o na representa  o num rica. Isso porque n meros em ponto flutuante s o representados, no LEGO NXT, da forma normalizada segundo o padr o IEEE-754 (Hansen, 2007). Assim, a representa  o desses n meros possui uma precis o m xima de 7 casas decimais, o que n o   suficiente para representar os n meros gerados pelo planejamento da trajet ria. Dessa forma, foi necess rio o uso de $\kappa = 5$. Como   poss vel observar na Figura 2, existe um curto movimento oscilat rio no in cio da trajet ria. No experimento com o LEGO  esse comportamento foi amplificado, aumentando-se o MSE, o tempo para realiza  o de trajet ria, bem como a dist ncia total percorrida para tal valor de κ . Uma outra quest o notada foi o erro de odometria, que se acumulou principalmente em curvas muito acentuadas, provavelmente devido   imprecis o ao medir a velocidade das rodas. O erro na localiza  o fez com que o rob  parasse 10cm distante do alvo.

Em rela  o ao tempo e   dist ncia percorrida, a t cnica WF foi a que apresentou melhor resul-

tado. Apesar do bom resultado, se comparado  s demais t cnicas a trajet ria planejada exige que o rob  se aproxime demais de alguns obst culos. Para contornar essa quest o, pode-se estender os obst culos ao se gerar o *grid*.

*	M�trica	CP	FN	WF
1	t_{total}	39,3s	74,4s	46,9s
	D_{total}	2,4m	2,0m	2,2m
	$MSE(\theta)$	498,2	1,8	391,5
	$D(q_f, q_{alvo})$	0,08m	0,02m	0,05m
2	t_{total}	59,8s	76,2s	42,3s
	D_{total}	2,4m	2,9m	2,1m
	$MSE(\theta)$	218,8	467,2	458,1
	$D(q_f, q_{alvo})$	0m	0,2m	0,06m

Tabela 1: Resultados num ricos dos experimentos.

7 Conclus es

Este trabalho tratou da implementa  o de um sistema de navega  o e controle em um rob  de tra  o diferencial para tr s estrat gias largamente utilizadas no planejamento de trajet rias na plataforma LEGO  NXT. Al m disso, as estrat gias foram comparadas segundo um conjunto de m tricas. As compara  es mostraram que o ajuste de par metros, se realizado de maneira incorreta, pode levar o rob  a realizar uma trajet ria pouco suave, exigindo consider vel esfor o de controle da orienta  o. Al m disso, em um ambiente com muitos obst culos, o ajuste das FN pode esbarrar na quest o da precis o, o que se mostrou uma limita  o da plataforma empregada. A implementa  o da estrat gia WF Planner exige que a velocidade do rob  seja adequadamente dimensionada em fun  o do tamanho dos obst culos, visto que a trajet ria planejada faz com que o rob  se aproxime demais dos obst culos.

Refer ncias

- Alves, P., Costelha, H. and Neves, C. (2013). Localization and navigation of a mobile robot in an office-like environment, *Autonomous Robot Systems (Robotica)*, 2013 13th International Conference on, pp. 1–6.
- Arantes, J. M. O., J nior, M. A. A. and Nepomuceno, E. G. (2010). Navega  o e mapeamento para um rob  aut nomo: Estrat gia baseada em aut matos celulares e no algoritmo wavefront, *XVIII Congresso Brasileiro de Aut mática*, pp. 4365–4372.
- Barraquand, J., Langlois, B. and Latombe, J.-C. (1991). Numerical potential field techniques for robot path planning, *Advanced Robotics*, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference 2: 1012–1017.
- Benitez, A., Moreno, C. J. and Vallejo, D. (2008). Localization control for lego robot's navigation, *Electronics, Communications and Computers*, 2008. CONIELECOMP 2008, 18th International Conference on, pp. 187–192.
- Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E. and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA.
- Colledanchise, M., Dimarogonas, D. V. and  gren, P. (2013). Obstacle avoidance in formation using navigation-like functions and constraint based programming, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5234–5239.
- G mez, E. J., Santa, F. M. and Sarmiento, F. H. M. (2013). A comparative study of geometric path planning methods for a mobile robot: Potential field and voronoi diagrams, *Engineering Mechanics and Automation (CIIMA)*, 2013 II International Congress of, pp. 1–6.
- Hansen, J. (2007). *Not eXactly C (NXC) Programmer's Guide*.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal of Robotics Research* 5(1): 90–98.
- Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation, *Robotics and Automation*, 1991. Proceedings., 1991 IEEE International Conference on, pp. 1398–1404 vol.2.
- Park, J. J. and Kuipers, B. (2011). A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment, *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on, pp. 4896–4902.
- Poomarin, W., Chanchaen, R. and Sangveraphunsiri, V. (2016). Automatic docking with obstacle avoidance of a differential wheel mobile robot, *International Journal of Mechanical Engineering and Robotics Research*, Vol. 5, pp. 11–16.
- Rimon, E. and Koditschek, D. (1992). Exact robot navigation using artificial potential fields, Vol. 8, pp. 501–518.
- Urukubo, T. (2015). Feedback stabilization of a nonholonomic system with potential fields: application to a two-wheeled mobile robot among obstacles, *Nonlinear Dynamics* 81(3): 1475–1487.