

CONTROLE DE TRAJETÓRIA DE UM ROBÔ MÓVEL COM TRAÇÃO DIFERENCIAL USANDO LÓGICA NEBULOSA

Saulo Egberto Andreoti¹, Thainan Augustinho Menezes¹, Mário Eduardo Bordon²

^{1,2} DEPARTAMENTO DE ENGENHARIA ELÉTRICA – FACULDADE DE ENGENHARIA

UNIVERSIDADE ESTADUAL PAULISTA – UNESP – CAMPUS DE BAURU

Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bairro: Vargem Limpa

CEP: 17033-360 – Bauru – SP

Resumo: Este artigo apresenta o estudo comparativo de dois controladores de trajetória distintos para robôs móveis com tração diferencial, um com lógica de controle proporcional e outro com lógica de controle nebulosa. Os dois controladores consideram o modelo cinemático do veículo, que segue uma trajetória de referência composta por retas e arcos suaves. É utilizado um simulador concebido em ambiente Matlab®, desenvolvido para robôs muito maiores que o robô utilizado neste projeto (Robô FutFEB) e, para isso, foram necessárias diversas mudanças em seu código para adequar o programa ao robô de pequeno porte. Neste estudo, o controlador nebuloso apresenta resultados mais promissores quando comparado com o controlador proporcional.

Palavras Chaves: Controle Nebuloso, Controle de Trajetória, Controle de Velocidade, Robôs Móveis, Lógica Nebulosa.

Abstract: This paper presents the study of two different trajectory controllers, a proportional logic and a fuzzy logic, for differential-drive mobile robots. The controllers consider the cinematic model of the vehicle, which follows a reference trajectory, composed by straight lines segments and soft arcs. It is used a simulator designed in Matlab®, which was developed to larger robots than the one utilized in this project (FutFEB Robot), and for that was necessary several changes in the program code to adapt the program to the small robot. The fuzzy controller presents the most promising results compared to the proportional controller.

Keywords: Fuzzy Control, Trajectory Control, Speed Control, Mobile Robots, Fuzzy Logic.

1 INTRODUÇÃO

A robótica móvel é um tema bastante relevante na atualidade e para o desenvolvimento de pesquisas nesta área é necessário a integração de conhecimentos de diversas áreas, como a de computação e de engenharia, para a criação de sistemas de controle mais robustos, seguros, autônomos e inteligentes para robôs móveis.

A evolução dos robôs móveis tem recebido nos últimos anos um amplo destaque junto à mídia e à sociedade de um modo geral. As atenções na atualidade estão voltadas para os robôs móveis capazes de navegar no ambiente em que se encontram de forma autônoma (Jung, 2005).

Os robôs móveis autônomos usam diferentes configurações de dispositivos de hardware embarcados e as diferenças básicas estão nas funções e tarefas para as quais são projetados. Os principais dispositivos de hardware de um robô são os sensores e atuadores (Bekey, 2005), (Dudek, 2000). Uma descrição mais detalhada dos diferentes tipos de mecanismos de locomoção em robôs móveis pode ser encontrada em (Siegwart, 2004).

O projeto de um sistema robótico envolve a especificação e seleção de diferentes componentes, sensores e atuadores (cada um com suas especificidades) e a combinação destes em um sistema autônomo. Este sistema deve ser projetado de modo a ser dotado de dispositivos capazes de prover dados relevantes (obtidos através dos seus sensores), para que o sistema de controle robótico possa planejar e realizar o acionamento dos seus dispositivos atuadores de modo a executar a ação desejada (Wolf, 2009).

Os sensores individualmente fornecem apenas uma informação parcial, incompleta e sujeita a erros, sendo papel do sistema de controle realizar a fusão dos sensores e tratar estas informações de forma robusta. Os comandos de atuação também não são precisos, pois estão sujeitos a erros de posicionamento do robô e de acionamento dos motores, cabe ao sistema de controle prover formas de compensar e corrigir estes erros para que as tarefas sejam executadas de forma precisa, correta e segura (Thrun, 2005).

Na criação de sistemas robóticos, é necessário em grande parte das aplicações o uso de sistemas de simulação computacional, para sejam feitos os testes e as correções no projeto antes da aplicação real. Isto gera maior segurança e acaba diminuindo até mesmo os gastos do projeto, pois com a análise prévia em ambientes simulados pode-se observar elementos que não se adequariam de forma concreta ao projeto antes mesmo de se fazer testes reais. Para se criar um modelo para simulação é necessário criar um modelo simplificado da realidade usando um ambiente computacional. Na área de robótica isto significa modelar: sensores, atuadores e comportamento físico do robô (Osório, 2006)

Nos dias atuais é quase impossível imaginar a vida humana sem as facilidades que a robótica e a automação têm trazido ao longo das últimas décadas. As necessidades atuais se voltam a equipamentos que possam trabalhar de forma mais autônoma e inteligente, como é o caso dos robôs móveis autônomos.

No cerne destas tecnologias está o estudo e desenvolvimento de controladores para estas aplicações, de modo que os robôs autônomos trabalhem de forma mais segura e que os erros apresentados no controle de trajetória destes equipamentos sejam minimizados ao máximo, gerando assim mais eficiência e confiabilidade.

Este artigo está organizado da seguinte forma: na Seção 2 mostra-se o modelo do Robô FutFEB e na Seção 3 encontra-se a caracterização dos controladores. Na Seção 4 é discutido o trabalho proposto para este artigo e na Seção 5 mostram-se os materiais e métodos utilizados neste trabalho. Já na Seção 6, tem-se os resultados coletados e a discussão dos mesmos, e por último, na Seção 7 apresentam-se as conclusões do trabalho.

2 ROBÔ MÓVEL DE TRACÇÃO DIFERENCIAL

Utiliza-se neste trabalho um robô móvel seguidor de linha de tração diferencial. A locomoção do robô é feita por duas rodas acopladas a motores de corrente contínua por intermédio de caixas de redução e rodas livres para assegurar a estabilidade do robô. Na Figura 1 observa-se a configuração do robô móvel.

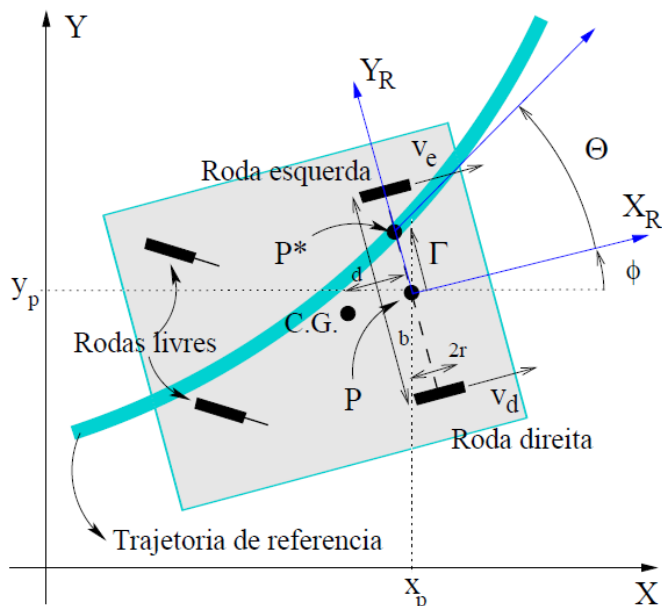


Figura 1 - Robô Móvel com Sistema de Tração Diferencial, sendo Γ o Deslocamento Ortogonal e Θ o Desvio Angular (Borges, 2003)

O sistema de coordenadas no qual a tarefa de navegação é definida é descrito por $C: X \times Y$, e o sistema de coordenadas do robô por $C_R: X_R \times Y_R$. A origem de C_R está sobre o ponto de referência P , que se localiza no centro das rodas de tração, com o eixo Y_R coincidindo com os eixos das rodas.

Os raios das rodas de tração direita e esquerda são definidos respectivamente por r_d e r_e , a distância entre as rodas é descrita por b , e d é a distância entre o ponto P e o centro de gravidade do robô $C.G.$

Na configuração cartesiana, a posição do robô é descrita pelo par ordenado (x_p, y_p) , e o ângulo ϕ entre os eixos X_R e X descreve sua orientação. O erro de trajetória é dado pelo deslocamento ortogonal Γ e o desvio angular Θ .

Define-se $P^* = (x^*, y^*)$ como um ponto de interseção entre a trajetória e a reta imaginária coincidente com o eixo das rodas. O deslocamento ortogonal Γ representa a distância $P^* - P$, sendo essa diferença negativa, caso P^* esteja à direita de P .

O Robô FutFEB, usado neste artigo, é formado por placas de acrílico mantidas distantes por espaçadores e parafusos que definem as bordas laterais e garantem ao robô o formato de um cubo, com dimensões não superiores a $7,5 \times 7,5 \times 7,5$ cm.

A estrutura de acrílico também confere ao robô a sustentação de todos os componentes que nele estão presentes, como a placa do circuito eletrônico, os motores, as baterias e as rodas livres (*ball casters*).

O movimento do robô é realizado por dois micromotores da *Pololu Robotics & Electronics* com caixas de redução de 50:1 e uma tensão de alimentação de 6 V.

O Robô FutFEB possui uma massa total de 262 gramas, sendo a massa do corpo de robô calculada (sem o sistema de tração) igual a $m_c = 243g$, pois cada sistema de tração tem massa igual a $m_w = 9,5g$. O raio das rodas e a distância entre as rodas são: $r = r_d = r_e = 1,7$ cm e $b = 7,5$ cm. O ponto P coincide com o $C.G.$, deste modo, tem-se $d = 0$ cm. A Figura 2 mostra uma foto do Robô FutFEB.

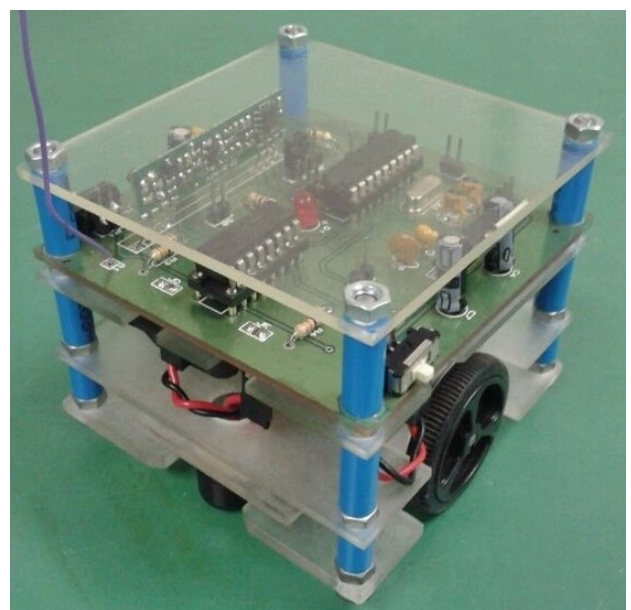


Figura 2 - Robô FutFEB com Tração Diferencial

3 CONTROLADORES

Os robôs móveis atualmente tem grande importância no âmbito tecnológico e se estende até os âmbitos econômicos e sociais ao executar tarefas sem a necessidade de intervenção humana.

O sistema de controle dos robôs móveis é formado pela união de um hardware e um software, que integrados, conferem ao sistema um comportamento adequado.

Além dos sensores e atuadores, que são utilizados na criação desses robôs, é necessário um sistema de controle que utiliza todos os recursos disponíveis desses robôs para que ele execute a atividade proposta. Neste contexto, destaca-se o controle de trajetória, que consiste em descobrir a melhor forma de levar um robô de um lugar ao outro.

O projeto de sistemas de controle de trajetória de robôs móveis pode ser abordado de duas formas, uma considerando apenas o modelo cinemático do robô e outra levando em consideração o modelo cinemático e dinâmico do robô.

Este artigo apresenta um controlador de trajetória, baseado no modelo cinemático de robôs móveis com tração diferencial sujeito a perturbações.

3.1 Arquitetura do Controlador

Um dos importantes atributos do controlador de trajetória é manter o veículo com velocidade constante (velocidade de cruzeiro) ou produzir variações de velocidade suaves (Borges, 2000). Para obter uma velocidade constante nas curvas da trajetória, as velocidades angulares $\omega_d(t)$ e $\omega_e(t)$ das rodas direita e esquerda, devem ser definidas da seguinte forma:

$$\omega_d(t) = \frac{v_p}{r} + \Delta\omega(\Gamma(t), \theta(t)) \quad (1)$$

$$\omega_e(t) = \frac{v_p}{r} - \Delta\omega(\Gamma(t), \theta(t)) \quad (2)$$

A velocidade do robô é descrita como:

$$v_p = \frac{\omega_d(t) + \omega_e(t)}{2} r \quad (3)$$

Considerando uma velocidade de cruzeiro constante, define-se a grandeza $v(t) = v_p$. O termo diferencial $\Delta\omega(\Gamma(t), \theta(t))$ representa a trajetória curvilínea do robô com um raio de curvatura instantâneo $R_c(t)$ definido por:

$$R_c(t) = \frac{v_p}{\Delta\omega(\Gamma(t), \theta(t))} \cdot \frac{b}{2r} \quad (4)$$

Desta forma, o termo $\Delta\omega(\Gamma(t), \theta(t))$ pode ser considerado como uma variável de controle e pode ser determinado pelas variáveis $\Gamma(t)$ e $\theta(t)$ que descrevem o desvio do robô em relação à trajetória de referência.

A organização hierárquica das operações de controle pode ser conferida na Figura 3. Existem três níveis de controle: o nível de controle de velocidade, o nível de controle de trajetória e o nível de navegação. No nível de navegação, a velocidade de cruzeiro do robô é definida. No nível de controle de trajetória, as velocidades das rodas direita e esquerda são definidas para manter o veículo na trajetória de referência. Neste trabalho, os controladores de velocidade são baseados na estratégia de controle MRAC (*Model Reference Adaptive Control*).

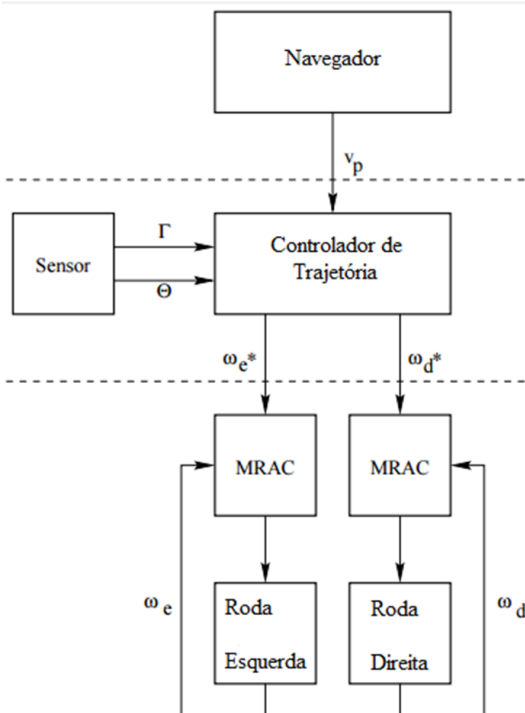


Figura 3 - Organização Hierárquica das Operações de Controle (Borges, 2000)

Neste artigo são abordados dois tipos de controladores:

- Controlador Proporcional;
- Controlador Nebuloso.

3.2 Controlador Proporcional

Esse tipo de controlador cria variáveis de controle que tem uma relação linear com as variáveis de estado do robô. Essa variável de controle é responsável pela ação dos atuadores, que nesse caso são os motores que criam a velocidade adequada para cada roda para que seja feita a correção na trajetória.

Neste tópico há a descrição do controlador proporcional para o modelo do robô descrito na Seção 2 e apresentado no artigo intitulado como “Controladores Cinemáticos de Trajetória para Robôs Móveis com Tração Diferencial” (Borges, 2003). Neste artigo foi proposto um controlador com uma estratégia que permite manter a velocidade do veículo constante ($v(t) = v_p$). Para garantir a velocidade de cruzeiro constante, tem-se:

$$v_d(t) = v_p + r\Delta\omega(t) \quad (5)$$

$$v_e(t) = v_p - r\Delta\omega(t) \quad (6)$$

Onde:

$v_d(t)$ = velocidade da roda direita

$v_e(t)$ = velocidade da roda esquerda

$\Delta\omega(t)$ = variável de controle proporcional definida por:

$$\Delta\omega(t) = K_r\Gamma(t) + K_\theta\theta(t) \quad (7)$$

Onde K_r e K_θ são constantes positivas e esses ganhos do controlador são determinados analiticamente. Essas constantes são definidas pela Equação 8 e Equação 9, onde Γ_c representa um erro constante no segmento de arcos.

$$K_\theta = \sqrt{\frac{2v_p K_r b}{r}} \quad (8)$$

$$K_r = \frac{2v_p b}{r((R + 2\Gamma_c)^2 - R)^2} \quad (9)$$

3.3 Controlador Nebuloso

Devido ao aumento da complexidade dos sistemas robóticos, a possibilidade de descrever estes sistemas através de equações matemáticas se torna muito complexa, ou até mesmo inviável.

Como alternativa a este problema, pode-se utilizar mecanismos de inferência baseados em regras lógicas de controle, advindas de termos linguísticos associados às variáveis de entrada. Para isso as variáveis de entrada são desmembradas em conjuntos discretos de termos linguísticos, devidamente ponderados, estabelecendo uma resposta às situações de entrada do sistema.

As ações de controle decorrentes de um conjunto termos linguísticos associados às variáveis de entrada são obtidas através do uso de conceitos da lógica nebulosa.

Nesta subseção, apresenta-se a descrição de um controlador nebuloso desenvolvido por (Bordon, 2011), que utiliza o modelo de robô descrito na Seção 2. O controlador nebuloso deste artigo pode ser construído com microcontroladores e conversores A/D e D/A de 8 bits.

Primeiro é necessário fazer a *fuzzificação* das variáveis de entrada (analogicas) em variáveis digitais equivalentes, que no formato de um único byte, varia de $[-128:128]$. A partir desses valores, é possível determinar os termos linguísticos e os graus de pertinência dos conjuntos de entrada. Os valores de Θ e Γ são, respectivamente, $\pm 90^\circ$ e ± 10 cm. Na Tabela 1 é apresentado o processo de *fuzzificação* das variáveis de entrada com peso w_Θ e w_Γ variando de $[-2, 2]$ e índices de pertinência μ_Θ e μ_Γ variando de $[0, 1]$ que está descrito na Figura 4.

Tabela 1 - Digitalização das Variáveis de Entrada (Θ e Γ)

Entradas		Digitalização Θ_D, Γ_D	Termo Linguístico	Peso w_Θ, w_Γ
$\Theta [^\circ]$	$\Gamma [cm]$			
$[-90:-45]$	$[-10:-5]$	$[-128:-64]$	NG	-2
$[-45:0]$	$[-10:0]$	$[-128:0]$	NP	-1
$[-45:45]$	$[-5:5]$	$[-64:64]$	ZE	0
$[0:45]$	$[0:10]$	$[0:128]$	PP	1
$[45:90]$	$[5:10]$	$[64:128]$	PG	2

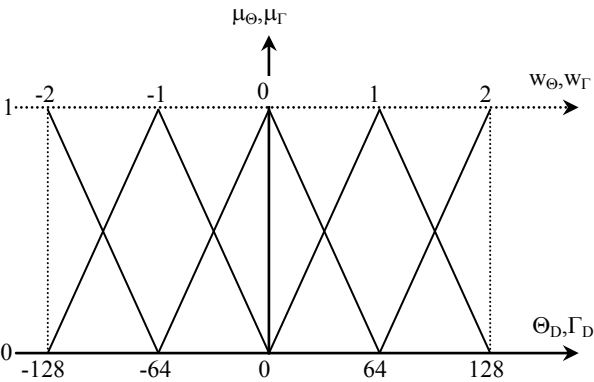


Figura 4 - Funções de Pertinência de Entrada

Depois da *fuzzificação* das variáveis é feita a *defuzzificação* que gera a variável de controle $\Delta\omega$. O valor máximo é de $\Delta\omega$ é de $\pm 10 \cdot \pi \cdot \text{rad/s}$. A Tabela 2 apresenta a variável de saída $\Delta\omega_D$ do controlador que é composta por 8 bits.

Tabela 2 - Digitalização da Variável de Saída $\Delta\omega$.

Saída $\Delta\omega [\pi \cdot \text{rad/s}]$	Digitalização $\Delta\omega_D$	Termo Linguístico	Peso $w_{\Delta\omega}$
$[-10:-7,5]$	$[-128:-96]$	NGG	-4
$[-10:-5]$	$[-128:-64]$	NG	-3
$[-7,5:-2,5]$	$[-96:-32]$	NM	-2
$[-5:0]$	$[-64:0]$	NP	-1
$[-2,5:2,5]$	$[-32:32]$	ZE	0
$[0:5]$	$[0:64]$	PP	1
$[2,5:7,5]$	$[32:96]$	PM	2
$[5:10]$	$[64:128]$	PG	3
$[7,5:10]$	$[96:128]$	PGG	4

Tabela 3 - Regras de controle de Trajetória

$\Delta\omega$		Γ	-50	-25	0	25	50
		w_Γ	-2	-1	0	1	2
Θ	w_Θ	$w_{\Delta\omega}$	NG	NP	ZE	PP	PG
-90	-2	NG	-4	-3	-2	-1	0
-45	-1	NP	-3	-2	-1	0	1
0	0	ZE	-2	-1	0	1	2
45	1	PP	-1	0	1	2	3
90	2	PG	0	1	2	3	4

A Tabela 3, mostra as 25 regras do controlador nebuloso e empregando cálculos matemáticos simples é possível calcular para cada combinação de entrada as ações de controle $\Delta\omega$.

A Equação 10 apresenta o operador de agregação $W_{\Delta\omega}$ e a Equação 11 o $\Delta\omega_D$.

$$W_{\Delta\omega} = W_{\Delta\omega(w_{\Theta a}, w_{\Gamma a})} \cdot (\mu_{\Theta a} + \mu_{\Gamma a}) + W_{\Delta\omega(w_{\Theta b}, w_{\Gamma a})} \cdot (\mu_{\Theta b}) + W_{\Delta\omega(w_{\Theta a}, w_{\Gamma b})} \cdot (\mu_{\Gamma b}) - W_{\Delta\omega(w_{\Theta a}, w_{\Gamma a})} \tag{10}$$

$$\Delta\omega_D = W_{\Delta\omega} \cdot 32 \tag{11}$$

4 O TRABALHO PROPOSTO

Este trabalho tem como objetivo o estudo comparativo de um controlador proporcional (Borges, 2003) e um controlador nebuloso (Bordon, 2011) para o controle de trajetória de um robô móvel com tração diferencial. Para isto, utilizou-se um simulador concebido em ambiente MATLAB® (Borges, 2003). Este estudo envolve um robô móvel com tração diferencial de pequenas dimensões utilizado pela equipe FutFEB de futebol de robôs, descrito na Seção 2.

Para que pudesse ser feita a simulação do controlador nebuloso para o Robô FutFEB foi necessário adaptar o código fonte do simulador para atender as especificações do robô móvel usado neste projeto, além de ter sido feita uma mudança na trajetória de referência para adequar as dimensões da mesma às novas dimensões do robô.

Foi feita a simulação do Robô FutFEB percorrendo a trajetória de referencia. Para análise do simulador foram feitos ensaios com os controladores proporcional e nebuloso e os resultados foram comparados entre si.

5 MATERIAIS E MÉTODOS

O simulador concebido em ambiente MATLAB® utilizado neste trabalho (Borges, 2003) é um sistema de simulação que engloba a especificação mecânica do robô, os sensores de localização do mesmo e o ambiente no qual o robô deve se movimentar de forma autônoma.

A resistência e a indutância no rotor são respectivamente, R_m e L_m , a constante que multiplica a velocidade angular do eixo do motor para obter a força eletromotriz de rotação é K_m , ρ é o fator de redução do sistema de engrenagens e $K_{\tau m}$ é a constante que multiplica o torque do motor para obter a corrente do circuito do rotor. Nas simulações, para o Robô FutFEB, são utilizados os seguintes parâmetros: $R_m = 1,0$, $L_m = 50 \times 10^{-6}$, $K_m = 1,0$, $\rho = 50$, $K_{\tau m} = 1,0$.

A base de tempo usada nas simulações é $T = 0,1 \text{ ms}$ e o tempo máximo de simulação é $T_{MAX} = 50 \text{ s}$, no sistema original a base de tempo para simulação era de 1 ms. O robô móvel é colocado na seguinte posição no instante inicial: $\Gamma_0 = 2,5 \text{ cm}$ e $\Theta_0 = \frac{\pi}{2} \text{ rad}$. A velocidade de cruzeiro usada é $v_p = 0,1 \text{ m/s}$.

Alguns pontos de controle sobre a trajetória de referência são colocados sobre a trajetória (A, B, C, D e E), que é composta por retas e arcos. Os arcos da trajetória possuem raio de 25 cm.

Foram efetuadas simulações para o Robô FutFEB utilizando os controladores proporcional e nebuloso com as especificações apresentadas acima.

6 RESULTADOS E DISCUSSÃO

Na Figura 5 pode-se observar a interface do simulador de trajetória, sendo em vermelho a trajetória de referência, e em azul a trajetória traçada pelo robô utilizando o controlador. Já na Figura 6, tem-se a interface das trajetórias com o robô recebendo o controlador nebuloso.

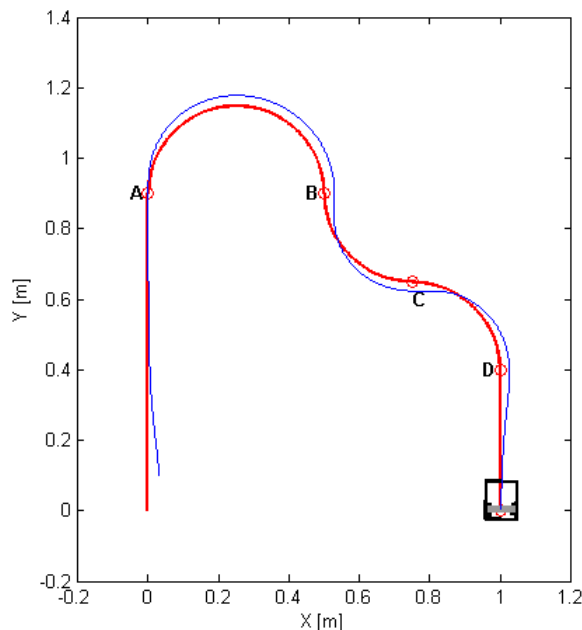


Figura 5 - Trajetória do Robô com Controlador Proporcional

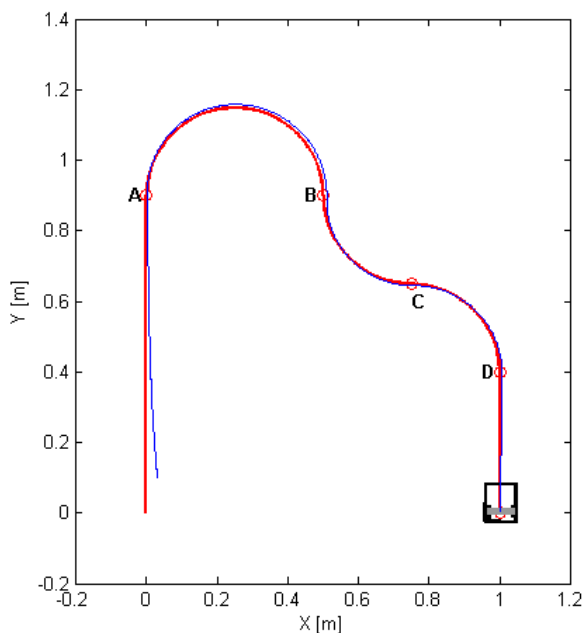


Figura 6 - Trajetória do Robô Móvel com Controlador Nebuloso

A resposta do controlador proporcional, com o deslocamento ortogonal e desvio angular, medidos ao longo da trajetória percorrida pelo robô pode ser vista na Figura 7 (a) e Figura 7 (b) respectivamente. O mesmo pode ser observado na Figura 8 para o controlador nebuloso.

Como esperado, o controlador nebuloso apresenta melhores resultados quando comparado com o controlador proporcional, e isto pode ser comprovado pelo fato da trajetória do robô (em azul) estar bem mais próxima da trajetória de referência (em vermelho) na simulação do controlador nebuloso (Figuras 5), do que na simulação do controlador proporcional (Figura 6).

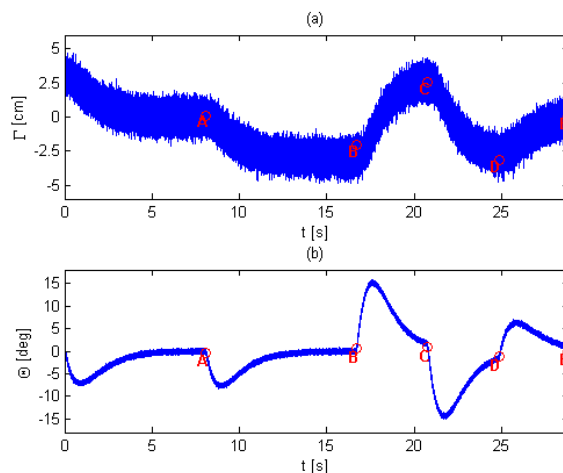


Figura 7 - Resposta do Controlador Proporcional. (a) Deslocamento Ortogonal (b) Desvio Angular.

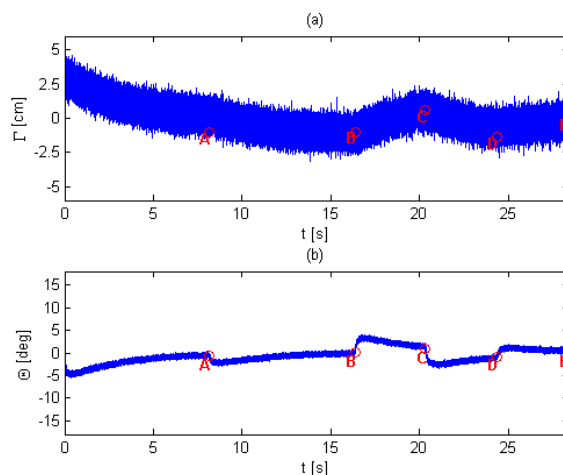


Figura 8 - Resposta do Controlador Nebuloso. (a) Deslocamento Ortogonal (b) Desvio Angular.

O mesmo pode ser concluído na análise e comparação das Figuras 7 e 8. Na Figura 7, pode-se verificar que deslocamento ortogonal máximo fica em torno de 4.5 cm, enquanto que, na Figura 8, tem-se o máximo em torno de 2.5 cm ao longo da trajetória. O mesmo pode ser observado no desvio angular, onde o controlador nebuloso apresenta valor máximo próximo de 5 graus, enquanto o controlador proporcional apresenta um valor próximo de 15 graus. É visível que o desvio angular e deslocamento ortogonal estiveram mais próximos do desejável (mais próximo de zero), na resposta do controlador nebuloso que na resposta do controlador proporcional em todos os pontos da trajetória.

7 CONCLUSÕES

Este artigo apresenta a abordagem de um simulador com controladores baseados no modelo cinemático de robôs móveis com tração diferencial. Originalmente, o simulador havia sido projetado para robôs de grande porte, e o que se almejava neste trabalho era a adaptação deste sistema para robôs de pequenas proporções. É possível observar que tanto o simulador como os controladores de trajetória se comportaram da forma esperada para o Robô FutFEB. Na análise comparativa dos resultados, pode-se observar que o controlador nebuloso é mais eficiente que o controlador proporcional. O passo de simulação foi diminuído na proporção de dez vezes em relação ao passo de simulação original para operação da interface e simulação dos controladores. Isto resultou em um maior número de iterações, requerendo um poder computacional maior para as simulações.

REFERÊNCIAS BIBLIOGRÁFICAS

- Bekey, G. A., (2005). “Autonomous Robots: From Biological Inspiration to Implementation and Control”. *The MIT Press*: Cambridge, London, pp. 563.
- Bordon, M. E.; Franchin, M. N. and Castanho, J. E. C., (2011). Sistema Nebuloso de Arquitetura Simplificada para Controle de Trajetória e Velocidade de Robôs Móveis. *Anais do X SBAI - Simpósio Brasileiro de Automação Inteligente*, São João del Rei, pp. 671-676.
- Borges, G. A.; Lima, A. M. N. and Deep, G. S., (2000). “Design of an Output Feedback Trajectory Controller for an Automated Guided Vehicle”. *Anais do XIII Congresso Brasileiro de Automática (CBA)*, Sociedade Brasileira de Automática, Florianópolis, pp. 2366-2371.
- Borges, G. A.; Lima, A. M. N. and Deep, G. S., (2003). “Controladores Cinemáticos de Trajetória para Robôs Móveis com Tração Diferencial”. *Anais do VI SBAI - Simpósio Brasileiro de Automação Inteligente*, Bauru, pp. 787-792.
- Dudek, G. and Jenkin, M., (2000). “Computational Principles of Mobile Robotics”. *The MIT Press*: Cambridge, London, pp. 280.
- Jung, C. R.; Osório, F. S.; Kelber, C. and Heinen, F., (2005). Computação Embarcada: Projeto e Implementação de Veículos Autônomos Inteligentes. *XXIV Jornada de Atualização em Informática*, São Leopoldo, RS, Vol. 1, pp. 1358-1406.
- Osório, F. S.; Musse, S. R.; Vieira, R.; Heinen, M. R. and Paiva, D. C., (2006). “Increasing Reality in Virtual Reality Applications through Physical and Behavioral Simulation. *Proceedings of the Virtual Concept International Conference*, Berlin, Springer Verlag, v. 1, pp. 1-45.
- Siegwart, R. and Illah, R. N., (2004). “Introduction to Autonomous Mobile Robots”. *A Bradford Book, The MIT Press*: Cambridge, London, pp. 317.
- Thrun, S.; Burgard, W. and Fox, D., (2004). “Probabilistic Robotics”. *The MIT Press*: Cambridge, London, pp. 667.
- Wolf, D. F.; Simões, E. V.; Osório, F. S. and Trindade Jr., O., (2009). Robótica Inteligente: Da Simulação às Aplicações no Mundo Real. *XXVIII Jornada de Atualização em Informática*, pp 1-51.