

CONTROLE ADAPTATIVO AUTO-AJUSTÁVEL PARA CONTROLE DE SEGUIMENTO DE TRAJETÓRIAS DE UM ROBÔ MÓVEL

THIAGO D. LOBO, MARCUS D. DO N. FORTE, FABRÍCIO G. NOGUEIRA, BISMARCK C. TORRICO, THIAGO A. LIMA, JESSICA P. ARAÚJO.

Grupo de Pesquisa em Automação, Controle e Robótica, Departamento de Engenharia Elétrica, Universidade Federal do Ceará

E-mails: thiago.loboo@gmail.com, davi2812@dee.ufc.br, fnogueira@dee.ufc.br, bismark@dee.ufc.br, talves3456@gmail.com, jessicpa@hotmail.com.

Abstract— This work presents an application of an adaptive control strategy for reference trajectory tracking of a mobile robot with non-holonomic constraints. The controller consists of two cascade control loops, whose first loop is responsible for the wheeled motor speed control, and the second is responsible for a non-linear control action that allows reference trajectory tracking. In order to maintain performance specifications when the robot is subjected to internal disturbances or model uncertainty, an auto-tuning non-linear adaptive controller is proposed. It also allows the robot to adjust to different types of reference trajectory. Results of both computational simulations and experimental tests in a real mobile robot show the advantages and disadvantages of the proposed adaptive controller.

Keywords— Mobile robotics, adaptive control, non-linear control, trajectory tracking, motion control.

Resumo- Este trabalho apresenta a aplicação de uma estratégia de controle adaptativo para seguimento de trajetórias por robôs móveis com acionamento diferencial e restrições não-holonômicas. O sistema de controle avaliado é composto por duas malhas de controle em cascata, na qual a primeira é responsável pelo controle de velocidade das rodas de tração e a segunda é um controlador não linear que tem o objetivo de permitir ao robô o seguimento de trajetórias de referência. Para manter o desempenho do seguimento mesmo quando sujeito a incertezas e distúrbios internos, foi proposta a aplicação de um controlador não linear adaptativo auto-ajustável, o qual possibilita o robô se ajustar a diferentes tipos de trajetórias de referência. Resultados de simulações computacionais e testes experimentais em um robô móvel real mostram vantagens e desvantagens do controlador adaptativo proposto.

Palavras-chave: Robótica móvel, controle adaptativo, controle não-linear, seguimento de trajetórias, controle de movimento.

1 Introdução

O problema de seguimento de trajetórias por robôs móveis é um dos ramos que vem sendo abordado em trabalhos científicos nos últimos anos (Yang & Kim, 1999). Diferentes estratégias de controle já foram publicadas a fim de garantir a estabilidade e desempenho satisfatório do seguimento de trajetórias constantes e variantes no tempo. Dentre estas destacam-se as baseadas em controle backstepping.

Em (Fierro & Lewis, 1995), a estabilidade assintótica de um robô móvel usando backstepping é provada pelo Método Indireto de Lyapunov a fim de tornar possível a interação de um controlador cinemático e um controlador de torque. Em (Chen, 2011), é proposto um modelo de controlador adaptativo backstepping que combina, explicitamente, as restrições cinemáticas e a dinâmica móvel de um robô móvel para rastreamento de trajetórias.

Com a finalidade de superar incertezas na dinâmica e parâmetros de um robô móvel, é proposto em (Mohareri, Dhaouadi, & Rad, 2012) um controlador backstepping adaptativo para seguimento

de trajetórias, no qual a estratégia de adaptação é baseada no algoritmo do gradiente descendente. Para a atualização dos ganhos do controlador em tempo real, o cálculo da matriz jacobiana do método do gradiente descendente foi realizado de duas formas. A primeira estratégia foi o cálculo analítico utilizando os modelos cinemático e dinâmico do robô, na qual é necessário o conhecimento de parâmetros físicos do robô. A segunda proposta para o cálculo da jacobiana consiste de um estimador baseado em rede neurais. Embora seja demonstrado no artigo a metodologia matemática, são omitidos os resultados com a estratégia de adaptação utilizando os cálculos analíticos da jacobiana.

Diante da omissão do trabalho de (Mohareri, Dhaouadi, & Rad, 2012), uma das principais contribuições desse artigo é realizar a implementação e testes da estratégia de adaptação analítica para o sistema de controle de seguimentos de trajetórias em um robô não-holonômico móvel real sujeito a distúrbios e incertezas.

2 Modelagem matemática

A Figura 1 ilustra uma unidade diferencial do robô móvel utilizado no trabalho. Este possui duas rodas

com raio R , dispostas a uma distância L do centro do robô. O centro de massa da plataforma (x_c, y_c) foi projetado para estar localizado a uma distância a do centro do eixo das rodas motrizes, denominado ponto A . A ordenada (A, x_r, y_r) é o ponto de referência fixo para o robô móvel e a ordenada (o, x_I, y_I) é o ponto de referência inercial global.

A plataforma do robô tem uma massa total m e um momento de inércia I_c em torno do seu centro de massa. E a configuração pode ser descrita por cinco coordenadas generalizadas ou também chamadas de variáveis de estado do sistema:

$$q(t) = [x_c(t) \quad y_c(t) \quad \theta(t) \quad \varphi_r(t) \quad \varphi_l(t)]^T \quad (1)$$

onde (x_c, y_c) são as coordenadas do centro de massa no ponto de referência inercial, θ é o ângulo de posição do robô, e (φ_r, φ_l) são os ângulos das rodas motrizes direita e esquerda. Tais coordenadas são variáveis no tempo, contudo, para fins de simplificação, a notação de dependência do tempo não será explícita nestas variáveis.

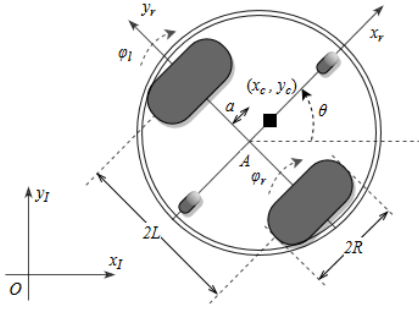


Figura 1. Modelo dos atuadores diferenciais do robô móvel e pontos de referência.

A velocidade de translação u e a velocidade angular ω da plataforma no ponto local podem ser calculadas em função das velocidades dos lados direito e esquerdo das rodas motrizes:

$$u = \frac{R}{2}(\dot{\varphi}_r + \dot{\varphi}_l), \quad \omega = \dot{\theta} = \frac{R}{2L}(\dot{\varphi}_r - \dot{\varphi}_l). \quad (2)$$

Para representar as condições práticas de não deslocamento lateral, tal restrição não-holonômica da unidade diferencial pode ser equacionada da seguinte forma:

$$\dot{y}_c \cos \theta - \dot{x}_c \sin \theta - a\dot{\theta} = 0, \quad (3)$$

onde \dot{y}_c e \dot{x}_c são os componentes de velocidade do centro de massa no ponto de referência inercial A , o que significa que a velocidade desse ponto é nula ao longo do eixo ortogonal local e será sempre na direção do eixo de simetria.

Outra restrição não-holonômica considerada é que não há derrapagem das rodas do robô, ou seja, tem-se um rolamento puro das rodas, sendo definido para a roda direita e esquerda respectivamente:

$$\dot{x}_c \cos \theta + \dot{y}_c \sin \theta + L\dot{\theta} = R\dot{\varphi}_r, \quad (4)$$

$$\dot{x}_c \cos \theta + \dot{y}_c \sin \theta - L\dot{\theta} = R\dot{\varphi}_l. \quad (5)$$

Essas três restrições não-holonômicas podem ser reunidas na seguinte forma matricial:

$$A(q)\dot{q} = 0, \quad (6)$$

onde,

$$A(q) = \begin{bmatrix} -\sin \theta & \cos \theta & -a & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 \\ \cos \theta & \sin \theta & -L & 0 & -R \end{bmatrix} \quad (7)$$

$$\dot{q} = [\dot{x}_c \quad \dot{y}_c \quad \dot{\theta} \quad \dot{\varphi}_r \quad \dot{\varphi}_l]^T. \quad (8)$$

A energia baseada na abordagem de *Lagrange* pode ser utilizada para encontrar o modelo dinâmico do robô móvel, sendo representado sob a forma geral (Yang e Kim, 1999):

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (9)$$

onde $M(q)$ é a matriz de inércia simétrica positiva, $V_m(q, \dot{q})$ é a matriz das pseudoforças centrípeta e de Coriolis, $F(\dot{q})$ é a matriz de atrito superficial do robô, $G(q)$ é o vetor gravitacional, τ_d é denotado distúrbios limitados desconhecidos incluindo dinâmicas não modeladas e não estruturadas, $B(q)$ é a matriz de transformação de entrada, τ é o vetor de entrada, $A^T(q)$ é a matriz associada às restrições e λ é o vetor de restrição das forças.

A locomoção plana do robô implica em eliminar os termos gravitacionais da equação dinâmica: $G(q)=0$. Além disso, assumindo que o atrito superficial do robô é desprezível no sistema, pode-se ter também definir $F(\dot{q})=0$.

Para fins de controle e simulação, o sistema acima pode ser transformado em uma representação mais adequada, onde o objetivo é eliminar o termo de restrição da equação.

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = S(q)v(t) = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \\ \frac{1}{R} & \frac{L}{R} \\ \frac{1}{R} & -\frac{L}{R} \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}, \quad (10)$$

onde $S(q)$ é a matriz cinemática direta. Uma vez que $S(q)$ é uma matriz de posto completo, formada por um conjunto de vetores suaves e linearmente independentes, pode-se provar a seguinte relação (FIERRO e LEWIS, 1995): $S^T(q)A^T(q)=0$.

A Equação anterior é útil para eliminar o termo de restrição a partir da equação dinâmica principal como mostrado no procedimento seguinte. Derivando a Equação (10), temos:

$$\ddot{q} = \dot{S}(q)v(t) + S(q)\dot{v}(t) \quad (11)$$

Substituindo a Eq. (11) em (9), resulta na seguinte equação:

$$M(q)[\dot{S}(q)v(t) + S(q)\dot{v}(t)] + \dots \quad (12)$$

$$\dots V_m(q, \dot{q})S(q)v(t) + \tau_d = B(q)\tau - A^T(q)\lambda$$

O passo seguinte para eliminar a matriz de restrição $A^T(q)\lambda$ é multiplicar a Eq. (12) por $S^T(q)$, como segue:

$$[S^T(q)M(q)\dot{S}(q) + S^T(q)V_m(q, \dot{q})S(q)]v(t) + \dots \quad (13)$$

$$[S^T(q)M(q)S(q)]\dot{v}(t) + S^T(q)\tau_d \dots$$

$$= S^T(q)B(q)\tau - S^T(q)A^T(q)\lambda$$

Como pode ser visto a partir da equação acima, o termo $S^T(q)A^T(q)$ que multiplica o coeficiente desconhecido de *Lagrange* λ será cancelado, resultando em:

$$[S^T(q)M(q)\dot{S}(q) + S^T(q)V_m(q, \dot{q})S(q)]v(t) + \dots \quad (14)$$

$$[S^T(q)M(q)S(q)]\dot{v}(t) + S^T(q)\tau_d \dots$$

$$= S^T(q)B(q)\tau$$

Pelas seguintes definições apropriadas, pode-se reescrever a Eq. (14) como se segue:

$$\bar{M}(q)\dot{v}(t) + \bar{V}_m(q, \dot{q})v(t) + \bar{\tau}_d = \bar{B}(q)\tau \quad (15)$$

Onde,

$$\bar{M}(q) = S^T(q)M(q)S(q) \quad (16)$$

$$\bar{V}_m(q, \dot{q}) = S^T(q)M(q)\dot{S}(q) + S^T(q)V_m(q, \dot{q})S(q) \quad (17)$$

$$\bar{\tau}_d = S^T(q)\tau_d \quad (18)$$

$$\bar{B}(q) = S^T(q)B(q) \quad (19)$$

As equações acima representam as equações dinâmicas do robô considerando as restrições não-holonômicas e elas podem ser facilmente transformadas para a seguinte equação matricial simplificada:

$$\begin{bmatrix} m & 0 \\ 0 & m\alpha^2 + I_c \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -m\alpha\dot{\theta} \\ m\alpha\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} u \\ \theta \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (20)$$

Nota-se que o termo $\bar{\tau}_d$, que define os distúrbios limitados desconhecidos incluindo dinâmicas não modeladas e não estruturadas, não está incluído na Eq. (20) uma vez que esta representa a dinâmica modelada do sistema, contendo, portanto, apenas termos conhecidos.

3 Controlador adaptativo auto-ajustável

O controlador *backstepping* proposto neste trabalho é baseado no trabalho de (Mohareri, Dhaouadi, & Rad, 2012). Tal estrutura de controle é uma forma estável de controle de rastreamento para um robô móvel não-holonômico. A estrutura de controle é ilustrada na Figura 2.

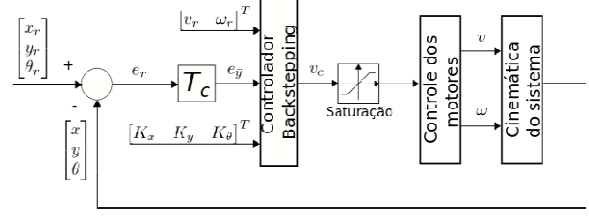


Figura 2. Diagrama de blocos do controlador não-linear.

3.1 Trajetória de Referência

A trajetória de referência é definida através do vetor $q_r = [x_r \ y_r \ \theta_r]$, relativos a um eixo de coordenadas global, calculado a partir de especificações de velocidade linear (v_r) e angular (ω_r) desejadas para cada trajetória.

$$\begin{cases} x_r = \int v_r \cos \theta_r dt \\ y_r = \int v_r \sin \theta_r dt \\ \theta_r = \int \omega_r dt \end{cases} \quad (21)$$

3.2 Odometria

O algoritmo de odometria realiza a estimação do vetor postura do robô $q = [x \ y \ \theta]$. A postura é calculada através da integração numérica das velocidades linear (v) e angular (ω) do robô, obtidas a partir da leitura de *encoders* nos eixos das duas rodas ($\dot{\phi}_d$ e $\dot{\phi}_e$). Estes valores são utilizados como realimentação de postura para os controladores *backstepping*.

$$\begin{cases} x_c = \int v \cos \theta dt = \int \left(\frac{R}{2} (\dot{\phi}_d + \dot{\phi}_e) \right) \cos \theta dt \\ y_c = \int v \sin \theta dt = \int \left(\frac{R}{2} (\dot{\phi}_d + \dot{\phi}_e) \right) \sin \theta dt \\ \theta_c = \int \omega dt = \int \left(\frac{R}{2L} (\dot{\phi}_d - \dot{\phi}_e) \right) dt \end{cases} \quad (22)$$

3.3 Transformação de frame e cálculo do erro

Considerando o referencial do robô móvel representado na Figura 1, o erro é assim calculado:

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = T_e e_r \quad (23)$$

onde e_r é a diferença entre a trajetória de referência $q_r = [x_r \ y_r \ \theta_r]$ e a postura do robô estimada pelo algoritmo de odometria $q = [x \ y \ \theta]$, e a matriz T_e é a matriz de rotação.

3.4 Controlador não-linear

A lei de controle que calcula o vetor de velocidades linear e angular de referência do robô é (Kanayama, 1990):

$$v_c = \begin{bmatrix} v_r \cos(e_\theta) + K_x e_x \\ \omega_r + K_y v_r e_y + K_\theta v_r \sin(e_\theta) \end{bmatrix} \quad (24)$$

onde $v_c = f(e_p, v_r, K)$ e $K = (K_x, K_y, K_\theta)$.

Como pode ser visto nas equações acima, o controlador tem três ganhos os quais são definidos como valores constantes positivos. Os termos $v_r \cos(e_\theta)$ e ω_r são as ações *feedforward* do controlador, enquanto os demais termos compõem a ação *feedback*.

Dentre as características deste controlador não-linear destaca-se a simplicidade de implementação, uma vez que se baseia somente no modelo cinemático. Contudo, é necessário cuidados no ajuste dos ganhos para cada trajetória de referência (KANAYAMA, 1990).

3.5 Controle dos motores

O controlador PID mostrado na Figura 3-2 é utilizado para fazer a velocidade do robô seguir a velocidade de referência da entrada. Em Mohareri (2012) é proposto um controlador PD para o controle de velocidade das duas rodas na forma:

$$\tau = K_p e_c + K_D \frac{de_c}{dt} \quad (25)$$

onde $e_c = [e_l \quad e_r]$ é um vetor com os erros de velocidade angulares das rodas da esquerda e direita, respectivamente, e K_p é definido como:

$$K_p = \begin{bmatrix} K_{Pr} & 0 \\ 0 & K_{Pl} \end{bmatrix}, \quad (26)$$

4 Estratégia de adaptação online dos ganhos

A estrutura de controle proposta adapta o ganho do controlador cinemático base para minimizar a função custo:

$$F = \frac{1}{2} \sum \gamma_x e_x^2 + \gamma_y e_y^2 + \gamma_\theta e_\theta^2 \quad (27)$$

onde as constantes γ_x, γ_y e γ_θ são ponderações para os respectivos erros de posição e_x, e_y e e_θ .

Definindo-se $P(\alpha) = P(K_x, K_y, K_\theta)$ como uma solução (ótimo) da função F , o gradiente $\nabla(F) = 0$ é chamado de ponto estacionário da função. Considerando-se o tempo discreto com índice k sendo a iteração atual, o gradiente de F é calculado por:

$$\nabla(F) = \frac{\partial F}{\partial \alpha} = \alpha_{k+1} - \alpha_k \quad (28)$$

onde α_{k+1} é o vetor solução da próxima iteração e α_k é solução no instante atual k . Portanto os valores de $\alpha = K_x, K_y, K_\theta$ são atualizados de acordo com:

$$\alpha_{k+1} = \alpha_k + \nabla(F) \quad (29)$$

Porém, por definição, o gradiente aponta para o máximo da função F . Como o objetivo é a minimização dos erros, é necessário a convergência no sentido contrário do gradiente:

$$-\nabla(F) = -\frac{\partial F}{\partial \alpha} = \alpha_{k+1} - \alpha_k \quad (30)$$

Então neste caso os valores de $\alpha = K_x, K_y, K_\theta$ são atualizados de acordo com:

$$\alpha_{k+1} = \alpha_k - \nabla(F) \quad (31)$$

A fim de se ajustar o passo do algoritmo do gradiente descendente, é utilizado um termo adicional η ponderando o gradiente:

$$\alpha_{k+1} = \alpha_k - \eta \nabla(F) \quad (32)$$

Substituindo-se os devidos ganhos $\alpha = K_x, K_y, K_\theta$ na função F , os ganhos do controlador são atualizados pelas seguintes equações ($n=k$):

$$K_x(n) = K_x(n-1) - \eta_x \frac{\partial F}{\partial K_x}, \quad (33)$$

$$K_y(n) = K_y(n-1) - \eta_y \frac{\partial F}{\partial K_y}, \quad (34)$$

$$K_\theta(n) = K_\theta(n-1) - \eta_\theta \frac{\partial F}{\partial K_\theta}, \quad (35)$$

onde η_x, η_y e η_θ são os passos do algoritmo de gradiente descendente.

Os ganhos do controlador de base cinemática são considerados parte da função custo e são otimizados e atualizados de acordo com o método de gradiente descendente. Para o cálculo dos termos $\partial F / \partial K_x, \partial F / \partial K_y$ e $\partial F / \partial K_\theta$, é necessário calcular a derivada parcial da função custo em relação vetor de parâmetros:

$$\frac{\partial F}{\partial \alpha} = \gamma_x e_x \frac{\partial e_x}{\partial \alpha} + \gamma_y e_y \frac{\partial e_y}{\partial \alpha} + \gamma_\theta e_\theta \frac{\partial e_\theta}{\partial \alpha} = e_p^T \left(\Gamma \frac{\partial e_p}{\partial \alpha} \right) \quad (36)$$

onde:

$$\Gamma = \begin{bmatrix} \gamma_x & 0 & 0 \\ 0 & \gamma_y & 0 \\ 0 & 0 & \gamma_\theta \end{bmatrix} \quad (37)$$

Substituindo a Eq. (23) na Eq. (36), temos:

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = T_e(q_r - q) \quad (38)$$

$$\frac{\partial F}{\partial \alpha} = e_p^T \Gamma \frac{\partial(T_e(q_r - q))}{\partial \alpha} = -e_p^T \Gamma T_e \frac{\partial q}{\partial \alpha}.$$

Usando a regra da cadeia, a derivada $(\partial q / \partial \alpha)$ pode ser escrita como:

$$\frac{\partial q}{\partial \alpha} = \frac{\partial q}{\partial v_c} \frac{\partial v_c}{\partial \alpha}. \quad (39)$$

A primeira derivada parcial da equação acima, $(\partial q / \partial v_c)$, é definida como a matriz Jacobiana com relação às entradas de velocidade do sistema:

$$\frac{\partial q}{\partial v_c} = J = \begin{bmatrix} \frac{\partial x}{\partial v_c} & \frac{\partial q}{\partial \omega_c} \\ \frac{\partial y}{\partial v_c} & \frac{\partial y}{\partial \omega_c} \\ \frac{\partial \theta}{\partial v_c} & \frac{\partial \theta}{\partial \omega_c} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \\ J_{31} & J_{32} \end{bmatrix}. \quad (40)$$

Enquanto que a segunda derivada parcial da Eq. (39), $(\partial v_c / \partial \alpha)$, é calculada como:

$$\frac{\partial v_c}{\partial \alpha} = \begin{bmatrix} \frac{\partial v_c}{\partial K_x} & \frac{\partial v_c}{\partial K_y} & \frac{\partial v_c}{\partial K_\theta} \\ \frac{\partial \omega_c}{\partial K_x} & \frac{\partial \omega_c}{\partial K_y} & \frac{\partial \omega_c}{\partial K_\theta} \end{bmatrix} = \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin(e_\theta) \end{bmatrix}. \quad (41)$$

Portanto, a derivada desejada da Eq.(39), pode ser encontrada substituindo as Eqs. (40) e (41) na Eq.(39):

$$\frac{\partial q}{\partial \alpha} = J \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin(e_\theta) \end{bmatrix} \quad (42)$$

Sendo assim, a derivada da função custo com relação aos ganhos do controlador da Eq. **Erro! Fonte de referência não encontrada.**36) fica:

$$\frac{\partial F}{\partial \alpha} = -e_p^T \Gamma T_e J \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin(e_\theta) \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial K_x} & \frac{\partial J}{\partial K_y} & \frac{\partial J}{\partial K_\theta} \end{bmatrix} \quad (43)$$

A principal parte na derivação da Eq. (43) é o procedimento de calcular a matriz Jacobiana do sistema. A matriz Jacobiana pode ser estimada usando as equações exatas do sistema (Mohareri, Dhaouadi e Rad, 2012).

4.1 Cálculo online da jacobiana

A matriz Jacobiana $J = (\partial q / \partial v_c)$ pode ser calculada usando as equações dinâmica e cinemática do robô. Decompondo-a, temos a seguinte forma:

$$J = \frac{\partial q}{\partial v_c} = \frac{\partial q}{\partial v} \frac{\partial v}{\partial \tau} \frac{\partial \tau}{\partial e_c} \frac{\partial e_c}{\partial v_c}. \quad (44)$$

Para calcular a derivada parcial do primeiro termo do produto mostrado na equação acima, usamos a Eq. (10) da cinemática do robô como segue:

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = Sv. \quad (45)$$

Integrando a equação acima ao longo do tempo, temos:

$$q = \int \dot{q} dt = \int Sv dt \quad (46)$$

Portanto, a derivada parcial do primeiro termo do produto é:

$$\frac{\partial q}{\partial v} = \int S dt = \int \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} dt. \quad (47)$$

A derivada parcial do segundo termo do produto da Eq. (44), pode ser encontrada a partir da Eq. (15) da dinâmica do sistema, onde simplificando-a para a forma compacta, temos que:

$$\bar{M}(q) \dot{v}(t) + \bar{V}_m(q, \dot{q}) v(t) = \bar{\tau}. \quad (48)$$

onde $\bar{\tau} = \bar{B}(q) \tau - \bar{\tau}_d$ é o torque equivalente total.

Isolando a derivada do vetor velocidade na Eq.(48), temos:

$$\dot{v}(t) = -\bar{M}(q)^{-1} \bar{V}_m(q, \dot{q}) v(t) + \bar{M}(q)^{-1} \bar{\tau}. \quad (49)$$

Portanto, para encontrar o vetor velocidade, integra-se a Eq.(49):

$$v(t) = \int_0^t -\bar{M}(q)^{-1} \bar{V}_m(q, \dot{q}) v(t) dt + \int_0^t \bar{M}(q)^{-1} \bar{\tau} dt \quad (50)$$

A Eq. (50) mostra que apenas o seu segundo termo depende do vetor torque. Como resultado disso, a matriz Jacobiana desejada é:

$$\frac{\partial v}{\partial \tau} = \begin{bmatrix} \frac{\partial u}{\partial \bar{\tau}_1} & \frac{\partial u}{\partial \bar{\tau}_2} \\ \frac{\partial \omega}{\partial \bar{\tau}_1} & \frac{\partial \omega}{\partial \bar{\tau}_2} \end{bmatrix} = \begin{bmatrix} \int_0^t \bar{M}(q)_{11}^{-1} dt & \int_0^t \bar{M}(q)_{21}^{-1} dt \\ \int_0^t \bar{M}(q)_{12}^{-1} dt & \int_0^t \bar{M}(q)_{22}^{-1} dt \end{bmatrix} \quad (51)$$

$$= \begin{bmatrix} \int_0^t \frac{1}{m} dt & 0 \\ 0 & \int_0^t \frac{1}{I_c + ma^2} dt \end{bmatrix} = \begin{bmatrix} \frac{1}{m} t & 0 \\ 0 & \frac{1}{I_c + ma^2} t \end{bmatrix}.$$

A partir da Eq.(25), o torque de entrada gerado pode ser aproximado por:

$$\tau = K_p e_c \quad (52)$$

A derivada parcial do controlador PD é desconsiderada na equação acima por ser muito pequena ou desprezível. Assim, a derivada parcial do terceiro termo do produto da Eq. (48) é dada por:

$$\frac{\partial \tau}{\partial e_c} = K_p = \begin{bmatrix} K_p & 0 \\ 0 & K_p \end{bmatrix} \quad (53)$$

O próximo termo do produto da Eq. (44), a quarta derivada parcial, fica na forma:

$$\frac{\partial e_c}{\partial v_c} = \frac{\partial(v_c - v_a)}{\partial v_c} = 1 \quad (54)$$

onde v_c é o vetor de velocidades linear e angular de referência v_a é o vetor de velocidades linear e angular medidas do robô.

Portanto, combinando as equações anteriores, Eqs. (47) (51) (53) e (54), a matriz Jacobiana desejada pode ser encontrada da seguinte forma:

$$J = \int \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} dt \int \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & \frac{1}{I_c + ma^2} \end{bmatrix} dt \begin{bmatrix} K_{p_r} & 0 \\ 0 & K_{p_l} \end{bmatrix} \cdot 1 \quad (55)$$

$$J = \int S dt \frac{\partial v}{\partial \tau} \begin{bmatrix} K_{p_r} & 0 \\ 0 & K_{p_l} \end{bmatrix}$$

3.4 Discretização para implementação no robô real

A estratégia de adaptação de ganhos está definida no domínio do tempo contínuo. Contudo, para a implementação no robô real foi necessária a discretização dos blocos dinâmicos. Para isto foi utilizado o Método de Euler, $s = (1 - q^{-1})/T_s$, para $T_s = 100 \text{ ms}$.

Considerando-se o índice de tempo discreto k , a matriz $\frac{\partial q}{\partial v}$ discretizada fica na forma:

$$S(k) = \begin{bmatrix} S_{11}(k) & S_{12}(k) \\ S_{21}(k) & S_{22}(k) \\ S_{31}(k) & S_{32}(k) \end{bmatrix} \quad (56)$$

onde,

$$\begin{cases} S_{11}(k) = S_{11}(k-1) + T_s \cos \theta(k) \\ S_{12}(k) = S_{12}(k-1) + T_s a \sin \theta(k) \\ S_{21}(k) = S_{21}(k-1) + T_s \sin \theta(k) \\ S_{22}(k) = S_{22}(k-1) + T_s a \cos \theta(k) \\ S_{31}(k) = 0 \\ S_{32}(k) = kT_s \end{cases} \quad (57)$$

Outro termo dinâmico que precisa ser discretizado é o cálculo de $\frac{\partial v}{\partial \tau}$:

$$\frac{\partial v}{\partial \tau} = \begin{bmatrix} (1/m)kT_s & 0 \\ 0 & kT_s / (I_c + ma^2) \end{bmatrix} \quad (58)$$

A estimativa de postura do robô via odometria discretizada resultante é dada por:

$$\begin{cases} \theta(k) = \theta(k-1) + t_s \omega(k) \\ x(k) = x(k-1) + t_s v(k) \cos \theta(k) \\ y(k) = y(k-1) + t_s v(k) \sin \theta(k) \end{cases} \quad (59)$$

As equações discretizadas foram utilizadas tanto nos modelos de simulação computacional quanto na implementação prática no robô real.

5 Resultados de Simulação

Para as simulações foram adotados os seguintes parâmetros físicos para o robô móvel: Massa (m) = 20kg; raio das rodas (r) = 7cm; largura do eixo (L) = 20cm; distância entre o centro do eixo e o centro de massa (a) = 1cm; momento de inércia (I_c) = $m \cdot L^2 = 0,2 \text{ kg/m}^2$. O valor dos ganhos do controlador proporcional de velocidade da rodas esquerda e direita são dados por $K_{p_r} = K_{p_l} = 10$. Com os parâmetros do robô definidos, determinou-se que o tempo de amostragem de 100ms.

O ajuste dos ganhos do controlador backstepping de ganho fixo serão utilizados como condição inicial dos ganhos do controlador backstepping adaptativo. Após algumas simulações, o melhor resultado encontrado foi com os seguintes valores de ganho:

$$K_{x_0} = 10, \quad K_{y_0} = 70 \quad \text{e} \quad K_{\theta_0} = 20 \quad (60)$$

Com esse ajuste para os controladores backstepping clássico e para o adaptativo.

Também foi realizada a comparação com uma estratégia de controle não-linear bastante referenciada na literatura (Klančar, 2005). Tal estratégia possui estrutura similar à estudada neste artigo. Os seguintes valores de parâmetros foram utilizados:

$$\xi = 0,6, \quad \omega_n = 2 \quad \text{e} \quad g = 30. \quad (61)$$

O desempenho dos controladores foi avaliado para uma trajetória de referência na forma de quadrado (Figura 3). Verifica-se que o controlador adaptativo inicia o rastreamento de trajetória com desempenho semelhante ao do controlador *backstepping* clássico. Isso ocorre devido a suas condições iniciais de ganhos serem as mesmas. O controlador de Klančar por sua vez conseguiu rastrear a trajetória de referência mais rapidamente que os controladores *backstepping*. Contudo, a cada variação da trajetória de referência é notável o reajuste do controlador adaptativo, melhorando a resposta a cada curva.

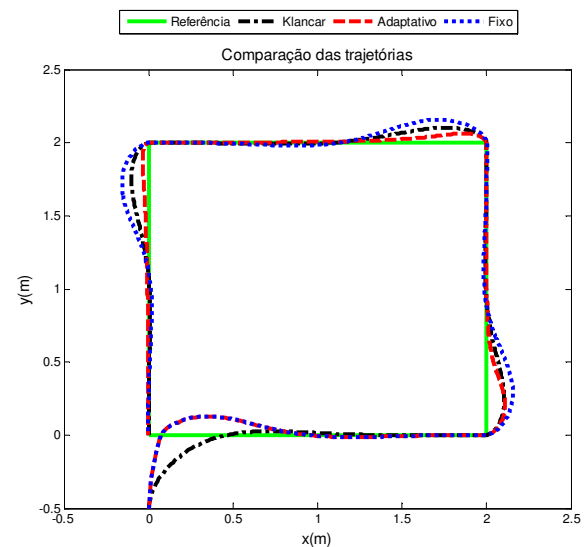


Figura 3. Rastreamento para a trajetória de um quadrado de lado dois (2m).

Atentando para os esforços de controle mostrados na Figura 4, observa-se que os controladores conseguem fazer com que as velocidade linear e angular convirjam para a referência ($0,5\text{m/s}$ e 0 rad/s , respectivamente) em um período próximo de 10s . Percebe-se, ainda, que o sinal de controle de velocidade linear e angular enviado pelo controlador *backstepping* adaptativo oscila menos a medida que o robô percorre a trajetória.

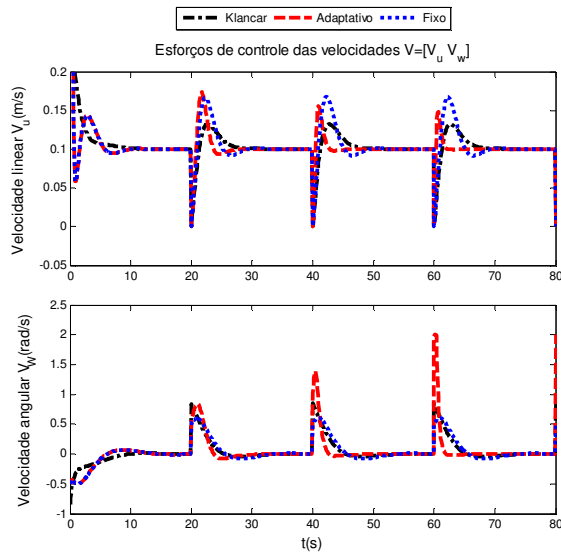


Figura 4. Esforço de controle das velocidades na simulação.

Ao verificar o gráfico dos erros mostrado na Figura 5, observa-se que os controladores *backstepping*, tanto o fixo como o adaptativo, corrigem os erros de forma mais rápida que o controlador de Klancar. Percebe-se, ainda, que o controlador adaptativo corrige os erros das coordenadas de trajetória de forma mais rápida a medida que o robô se move ao longo da trajetória.

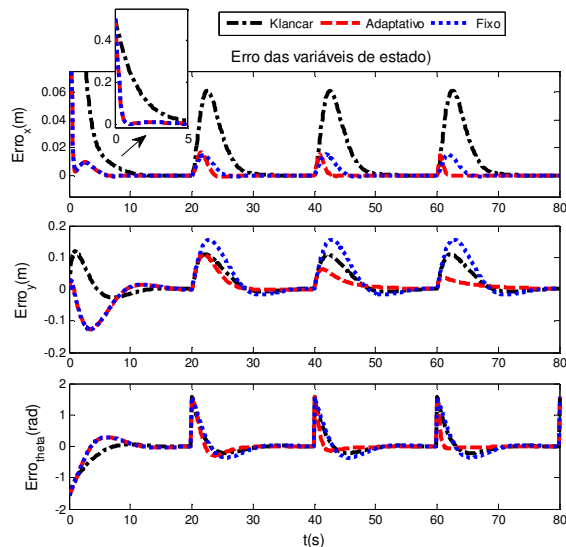


Figura 5. Erros de trajetória das variáveis de estado x , y e θ para a trajetória de um quadrado.

Na Figura 6 verifica-se que a capacidade de ajuste *online* dos ganhos do controlador adaptativo a cada curva da trajetória quadrada. Dessa forma, o desempenho de rastreamento é mais preciso à medida que o robô se move ao longo da trajetória.

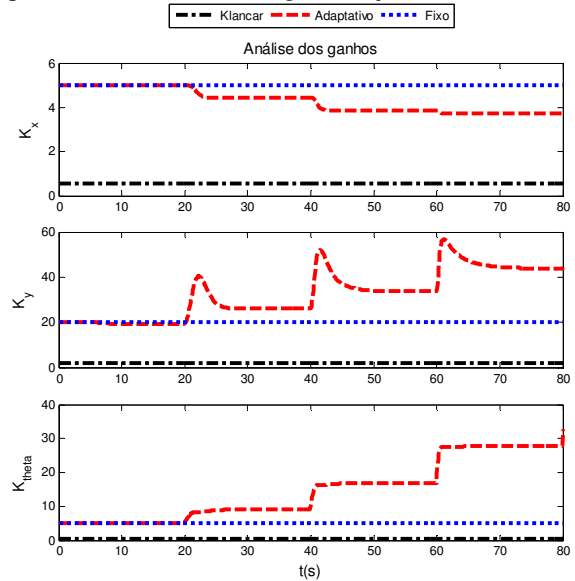


Figura 6. Ganhos dos controladores na simulação.

6 Resultados Experimentais

Os ensaios experimentais foram realizados em um robô móvel com duas rodas desenvolvido no departamento de engenharia elétrica da Universidade Federal do Ceará (UFC) (Figura 7).

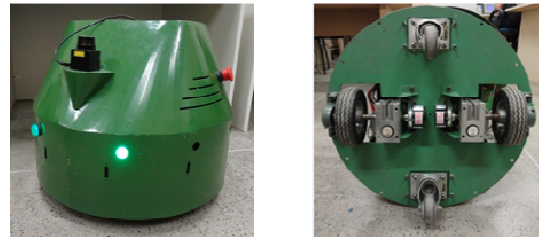


Figura 7. Robô móvel com duas rodas desenvolvido na UFC.

Considerou-se que o robô estará em uma condição inicial $(X_0, Y_0) = (0, 0, -0,5)$, posicionado a um ângulo $\theta = 90$ graus em relação ao eixo de referência.

A Figura 8 apresenta o seguimento da trajetória de referência. Percebe-se que para o ajuste adotado o controlador Klancar resultou em uma curva com sobre sinal reduzido quando comparado com os demais. Além disso, devido à adaptação dos ganhos, percebe-se que o controlador adaptativo se aproxima da resposta do controlador Klancar a medida que a trajetória é percorrida.

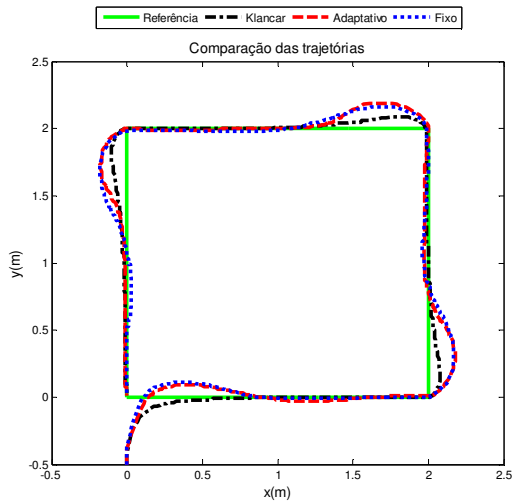


Figura 8. Seguimento do robô real para uma trajetória na forma de um quadrado.

A Figura 9 apresenta os respectivos valores de esforço de controle. Naturalmente o controlador Klancar apresenta um esforço de controle superior quando comparado com os demais controladores.

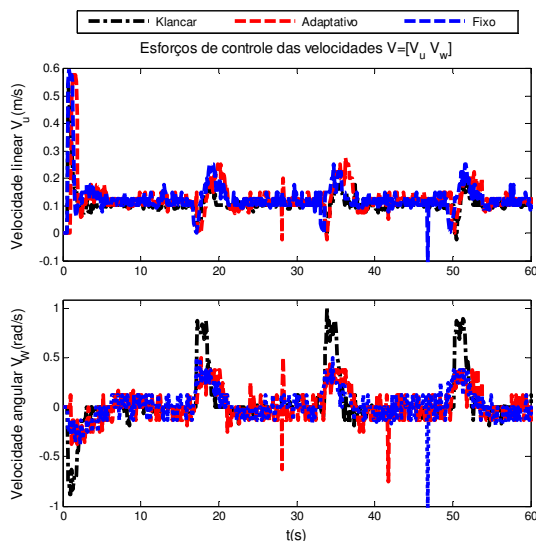


Figura 9. Esforço de controle nos resultados experimentais.

A Figura 10 apresenta os respectivos valores de erros e os valores dos ganhos dos controladores são apresentados na Figura 11. Percebe-se claramente que a redução dos erros do controlador adaptativo ao longo da trajetória é devido à adaptação dos ganhos.

Agradecimentos

Os autores do trabalho agradecem o CNPQ, a CAPES e a FUNCAP, através do suporte de bolsas de estudo, auxílio PROAP e projeto 88887.114030/2015-01.

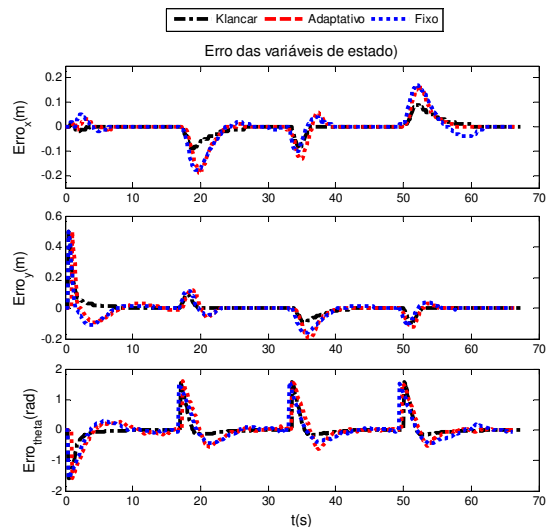


Figura 10. Erros de trajetória nos resultados experimentais.

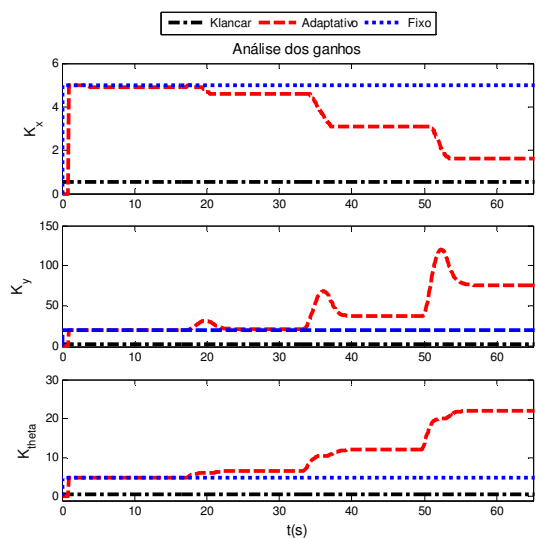


Figura 11. Ganhos dos controladores nos ensaios experimentais.

Referências Bibliográficas

- Fierro, R.; Lewis, F. L. Control of a Nonholonomic Mobile Robot: Backstepping Kinematic Into Dynamics. Proceedings of the 34th Conference on Decision & Control, New Orleans, LA, n. IEEE, December 1995.
- Kanayama, Y. A Stable Tracking Control Method for an Autonomous Mobile Robot. Proceedings of the IEEE Conference on Robotics Automation. Santa Bárbara CA, p. 384 - 389. 1990.
- Klančar, G., Matko, D. and Blažič, S., 2005, June. Mobile robot control on a reference path. In Intelligent Control, 2005. Proceedings of the IEEE International Symposium on Control and Automation (pp. 1343-1348). IEEE.
- MOHARERI, O.; DHAOUADI, R.; RAD, A. B. Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks. pp. 54-66, 2012.