

# Design and Implementation of Adaptive Neural PID for Non Linear Dynamics in Mobile Robots

F. G. Rossomando and C. M. Soria

**Abstract**— In this work, it will be reported original results concerning the application of PID Adaptive Neural controller in mobile robot in trajectory tracking control. In this control strategy the exact dynamical model of the robot will not need to be known and identified. To implement this strategy, two controllers are implemented separately: a kinematic controller and an adaptive neural PID controller. The uncertainty and dynamics variations in the robot dynamic are compensated by an adaptive neural PID controller. The resulting adaptive neural PID controller is efficient and robust in the sense that it succeeds to achieve a good tracking performance. The stability of the proposed technique (based on Lyapunov's theory) was demonstrated. Finally, experiments on a mobile robot have been developed to show the performance of the proposed technique, including the comparison with other controllers.

**Keywords**— MIMO system, Nonlinear control, adaptive control

## I. INTRODUÇÃO

O CONTROLE Proporcional-Integral-Derivativo (PID) é amplamente utilizado no controle de robôs industriais [1]. Na ausência de conhecimento do robô, um controlador PID pode ser o melhor controlador, porque não precisa do modelo e seus parâmetros podem ser ajustados em forma simples e separadamente [2], tendo em conta que a dinâmica do robô não varia com o tempo.

Geralmente, o controlador linear PID não é efetivo quando o sistema é muito complexo e se as incertezas podem introduzir não linearidades. [3], [4]. No campo de controle de robôs, o modelo de um robô não é simples de obter com precisão.

No controle do robô a influência dos parâmetros variantes no tempo não pode ser ignorada num modelo inexato. Alguns dos problemas dos algoritmos PID tradicionais que existem no controle de robôs são reflexados quando os seus parâmetros variam com o tempo enquanto os parâmetros do controlador PID não variam ao mesmo tempo ou são fixados num valor constante.

Portanto, os PID avançados têm sido desenvolvidos rapidamente no campo de controle de robôs, infundindo uma grande quantidade de algoritmos avançados com sintonia dos parâmetros do PID.

Sendo os controladores PID baseados sobre a tecnologia nebulosa [5], [6], [7]. Controladores PID baseados em redes neurais [8], [9]. Controladores PID baseados em algoritmos genéticos [10], [11], Controladores PID baseados em

algoritmos de colônias de formigas [12]. Controladores PID baseados em enxame de partículas [13], [14], [15], em algoritmos de recozimento simulado [16], etc.

Nos trabalhos de controladores PID baseados em redes neurais com aplicações em robótica móvel pode se descrever o trabalho de Li *et al.* [17], onde um algoritmo de controle híbrido foi proposto sob um controle PID dinâmico com modo de deslizamento e um controle cinemático *backstepping*. O ganho do controle de modo de deslizamento é ajustado pelo uso de uma rede RBF-NN com algoritmo de ajuste adaptável e o controlador produz dois torques baseados sob a dinâmica do modelo. Os resultados mostrados são baseados em praticas de simulação e o análise é feito em tempo contínuo.

Neste trabalho pretende mostrar o desenho e aplicação de um controlador PID neuronal adaptável sobre a dinâmica não linear de um robô móvel. A estrutura do robô é dividida em duas partes, uma com a parte cinemática, e a outra com a parte dinâmica do robô. Sobre a cinemática é aplicado uma técnica de controle de modelo inverso e não possui parâmetros variantes. Portanto não é preciso aplicar técnicas de controle adaptável. Sobre a parte dinâmica os parâmetros podem variar devido à fricção, perda de massa, escorregamento, etc., além de ser uma dinâmica não linear.

O controle PID neuronal adaptável, pretende compensar as variações dinâmicas e suas não linearidades, já que esta estudada como uma rede neuronal tipo recorrente.

A técnica de controle apresenta as seguintes vantagens:

- (1) Esta técnica de controle pode ser aplicada a um sistema MIMO não linear, como no caso da dinâmica de um robô móvel.
- (2) Não precisa se conhecer o modelo matemático exato do processo ou sistema (neste caso a dinâmica do robô).
- (3) O controlador PID neuronal MIMO seu desenho é simples e baseado na topologia de uma rede neural do tipo dinâmica que possui capacidade adaptável.
- (4) O controlador mostrado neste artigo está baseado nas ações de controle de velocidade linear e angular, e não no torque das rodas, como a maioria dos trabalhos nesta área.
- (5) O desenho, análise e implementação dos controladores foram feitos em tempo discreto para maior facilidade de implantação em computadores e equipes digitais.
- (6) É apresentada uma análise de estabilidade que demonstra a convergência do controlador proposto.

Este artigo é organizado como segue: Na Seção 2 apresenta uma revisão do sistema e mostra a representação

F. G. Rossomando, Universidad Nacional de San Juan (UNSJ-Conicet), Capital, San Juan, Argentina, San Juan, Capital, San Juan, Argentina, frosoma@inaut.unsj.edu.ar

C. M. Soria, Universidad Nacional de San Juan (UNSJ-Conicet), Capital, San Juan, Argentina, csoria@inaut.unsj.edu.ar

matemática do modelo do rebo uniclo. Os dois controladores cinemáticos e PID neuronal são estudados, respectivamente na Seção III e IV, e sua correspondente análise do erro. Na Seção V se demonstra a estabilidade do sistema proposto sob o robô móvel. Na Seção VI apresentam os resultados experimentais mostrando a eficiência dos controladores. E finalmente as conclusões são apresentadas na Seção VII.

## II. MODELO DO ROBÔ

### 2.1 Descrição do sistema robótico móvel

Nesta seção se revisa o modelo do robô móvel do tipo uniclo como é mostrado na Figura 1, com os parâmetros e as variáveis de interesse.

Aqui,  $x_1$  e  $x_2$  são as velocidades linear e angular desenvolvidas pelo robô,  $G$  é o centro de massa do robô,  $c$  é a posição da roda livre,  $h$  é o ponto de interesse com as coordenadas  $r_x, r_y$ , no plano XY,  $\psi$  é a orientação do robô,  $a$  é a distância entre o ponto de interesse e o ponto central do eixo virtual vinculado às rodas de tração.

A representação matemática do modelo completo discretizado [18], é dada pelo modelo:

Modelo Cinemático Discreto

$$\begin{pmatrix} r_x(k+1) \\ r_y(k+1) \\ \psi(k+1) \end{pmatrix} = T_0 \begin{pmatrix} \cos\psi(k) & -a\sin\psi(k) \\ \sin\psi(k) & a\cos\psi(k) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} r_x(k) \\ r_y(k) \\ \psi(k) \end{pmatrix} + \begin{pmatrix} \delta_{rx} \\ \delta_{ry} \\ 0 \end{pmatrix} \quad (1)$$

Modelo de dinâmica discreta.

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} = \begin{pmatrix} \frac{\Omega_3}{\Omega_1} x_2^2(k) + \frac{\Omega_4}{\Omega_1} x_1(k) \\ -\frac{\Omega_5}{\Omega_2} x_1(k) x_2(k) + \frac{\Omega_6}{\Omega_2} x_2(k) \end{pmatrix} + \dots + \begin{pmatrix} 1/\Omega_1 & 0 \\ 0 & 1/\Omega_2 \end{pmatrix} \begin{pmatrix} u_{1ref}(k) \\ u_{2ref}(k) \end{pmatrix} + \begin{pmatrix} \delta_v \\ \delta_w \end{pmatrix} \quad (2)$$

Os elementos do vetor de incertezas  $\delta_{rx}$  e  $\delta_{ry}$  são funções da velocidade de deslizamento e da orientação do robô,  $\delta_1$  e  $\delta_2$  são funções dos parâmetros físicos como a massa, a inércia, diâmetros das rodas e parâmetros dos servos, forças aplicadas sobre as rodas, e outros fatores são considerados como perturbações.

O modelo do robô apresentado em (1) y (2) divide se numa parte cinemática e numa parte dinâmica respectivamente. Pelo tanto, aplicam-se dois controladores, um deles baseado na linearização por retroalimentação para a cinemática e o PID-NN para a dinâmica do robô.

Onde os parâmetros discretizados são definidos como:

$$\begin{aligned} \frac{\Omega_3}{\Omega_1} &= \left( \frac{\vartheta_1}{\vartheta^2} T_0 \right); \quad \frac{\Omega_4}{\Omega_1} = \left( 1 - \frac{\vartheta_2}{\vartheta^2} T_0 \right); \quad \frac{1}{\Omega_1} = \left( \frac{1}{\vartheta^2} T_0 \right) \\ \frac{\Omega_5}{\Omega_2} &= \left( -\frac{\vartheta_3}{\vartheta^2} T_0 \right); \quad \frac{\Omega_6}{\Omega_2} = \left( 1 - \frac{\vartheta_4}{\vartheta^2} T_0 \right); \quad \frac{1}{\Omega_2} = \left( \frac{1}{\vartheta^2} T_0 \right) \end{aligned} \quad (3)$$

Onde  $T_0$  é tempo de amostragem e  $k$  é o tempo discreto.

Os parâmetros identificados para o robô móvel Pioneer DX2 são:  $\vartheta_1=0,3037$ ;  $\vartheta_2=0,2768$ ;  $\vartheta_3=-0,0004018$ ;  $\vartheta_4=0,9835$ ;  $\vartheta_5=-0,003818$ ;  $\vartheta_6=1,0725$ ;  $T_0=0,1$ seg.

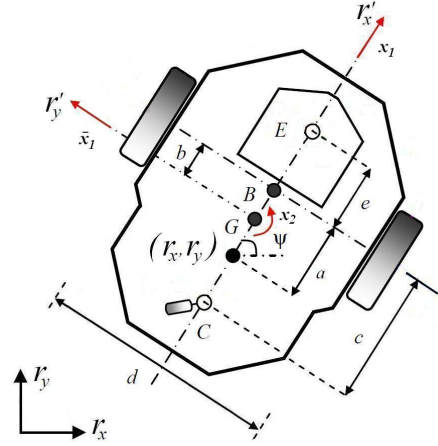


Figura 1. Parâmetros do robô móvel uniclo.

## III. CONTROLADOR CINEMÁTICO.

O desenho do controlador cinemático é baseado no modelo cinemático do robô. O controlador cinemático proposto é:

$$\begin{pmatrix} x_{1ref}(k) \\ x_{2ref}(k) \end{pmatrix} = \begin{pmatrix} \cos\psi(k) & \sin\psi(k) \\ -\frac{1}{a}\sin\psi(k) & \frac{1}{a}\cos\psi(k) \end{pmatrix} \dots \begin{pmatrix} r_{xref}(k+1) + l_x \tanh(k_x \tilde{r}_x(k)/l_x) \\ r_{yref}(k+1) + l_y \tanh(k_y \tilde{r}_y(k)/l_y) \end{pmatrix} - \begin{pmatrix} r_x(k) \\ r_y(k) \end{pmatrix} \quad (4)$$

Trocando (7) na parte superior de (4) baixo a consideração de um seguimento perfeito de velocidade, a equação de malha fechada é,

$$\begin{pmatrix} \tilde{r}_x(k+1) \\ \tilde{r}_y(k+1) \end{pmatrix} - \begin{pmatrix} l_x & 0 \\ 0 & l_y \end{pmatrix} \begin{pmatrix} \tanh(k_x \tilde{r}_x(k)/l_x) \\ \tanh(k_y \tilde{r}_y(k)/l_y) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (5)$$

Definindo o vetor de erro de saída

$\tilde{\mathbf{h}}(k) = [\tilde{r}_x(k) \ \tilde{r}_y(k)]^T = [r_{xs}(k) \ r_{ys}(k)]^T - [r_x(k) \ r_y(k)]^T$ , eq (8) pode ser escrito como:

$$\tilde{\mathbf{h}}(k+1) = \begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x} \tilde{r}_x(k)\right) & l_y \tanh\left(\frac{k_y}{l_y} \tilde{r}_y(k)\right) \end{bmatrix}^T \quad (6)$$

O qual implica que  $\tilde{\mathbf{h}}(k) \rightarrow 0$  quando  $k \rightarrow \infty$ . A consideração de seguimento perfeito de velocidade pode ser relaxada quando é analisada a estabilidade do sistema de controle completo.

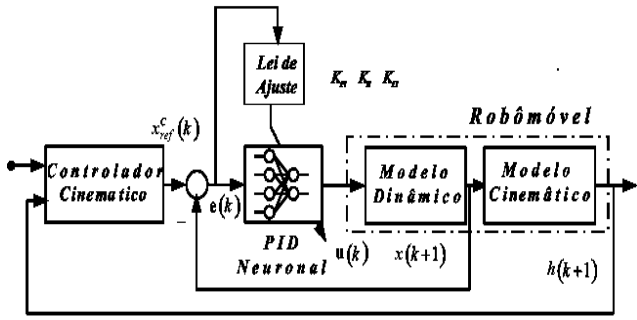


Figura 2. Estrutura de Controle do robô móvel.

#### IV. CONTROLADOR PID-NEURONAL DINÂMICO

Nesta secção é apresentada a principal contribuição do trabalho: um controlador PID não linear com ações de controle acoutadas que permitem uma estabilidade assintótica global, provendo uma solução ao controle global de set-point de dinâmicas não lineares, muitas das quais usam o PID clássico, sendo exemplificada na dinâmica de um robô móvel.

O controlador PID-Neuronal dinâmico deve receber as referências de velocidade linear e angular, as quais são geradas pelo controlador cinemático, e são geradas outras duas velocidades de saída (linear e angular) que são enviadas aos servos do robô, como são mostradas na Figura 2. Definindo o vetor de erro de velocidades de saída como:

$$\mathbf{e}(k) = (e_1(k), e_2(k))^T \quad (7)$$

Onde:

$$e_1(k) = x_{1ref}^c(k) - x_1(k) \quad e_2(k) = x_{2ref}^c(k) - x_2(k)$$

O modelo do controlador PID estático discreto é apresentado como:

$$o_i(k) = K_{P_i}e_i(k) + K_{I_i}\sum_{j=1}^k e_i(j)T_0 + \frac{K_{D_i}}{T_0}(e_i(k) - e_i(k-1)) \quad (8)$$

Onde o subíndice  $i$  faz referência a velocidade linear ( $i=1$ ) ou angular ( $i=2$ ) do robô móvel.

Considerando da ação de controle para o instante de tempo  $(k-1)$  é:

$$o_i(k-1) = K_{P_i}e_i(k-1) + K_{I_i}\sum_{j=1}^{k-1} e_i(j)T_0 + \dots + \frac{K_{D_i}}{T_0}(e_i(k-1) - e_i(k-2)) \quad (9)$$

Fazendo a diferença entre  $o(k)$  e  $o(k-1)$  das equações (11) e (12) respetivamente, o sinal  $o(k)$  pode ser descrito como:

$$o_i(k) = o_i(k-1) + K_{P_i}[e_i(k) - e_i(k-1)] + \dots + K_{I_i}e_iT_0 + \frac{K_{D_i}}{T_0}(e_i(k) - 2e_i(k-1) + e_i(k-2)) \quad (10)$$

Destá forma o controlador PID pode ser obtido em função do instante  $o(k-1)$ . Mas o objetivo deste trabalho é desenvolver uma rede neural baseada no auto-ajuste de um sistema de controle PID, onde as saídas de velocidade (linear e angular) possam seguir as referências geradas pelo controlador cinemático. Onde os parâmetros do PID ( $K_{D_i}$ ,  $K_{I_i}$ ,  $K_{P_i}$ ) podem se sintonizar de forma adaptável e ótima para

minimizar o erro de velocidade com respeito às perturbações externas.

Para representar uma rede neuronal com o controlador PID é preciso ter uma função de ativação na saída da rede, é aplicada uma função  $\tanh$  que é uma saturação continua apresentada na eq. (15), a saída do controlador deve ser menor que ação de controle máxima permitida em concordância com a consideração 2, da secção anterior:

$$\delta_i = u_{iMax} \quad (11)$$

Além disso, é aplicada uma saturação em cada ação do controlador PID, para garantir sua estabilidade, como será demonstrada na Secção V.

**Definição 1:** a função de ativação  $f(.)$  é definida como:

$$f(o_i) = \delta_i \tanh\left(\frac{o_i}{\delta_i}\right) \quad (12)$$

O controlador definido na equação (13) fica desta forma:

$$o_i(k) = f(o_i(k-1)) + \dots + K_{P_i}f(e_i(k) - e_i(k-1)) + K_{I_i}f(e_iT_0) + \dots + K_{D_i}f\left(\frac{e_i(k) - 2e_i(k-1) + e_i(k-2)}{T_0}\right) \quad (13)$$

A Figura 3 mostra a arquitetura da rede PID neuronal.

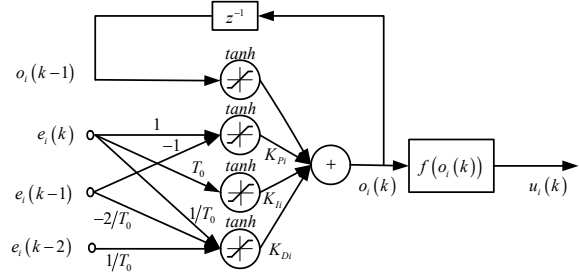


Figura 3. arquitetura da rede PID neuronal.

Para garantir a convergência e o desempenho da função de otimização deve ser definido um índice que seja função dos erros de seguimento de velocidades. Definido como:

$$E(k) = \frac{1}{2}(\mathbf{e}^T(k)\mathbf{e}(k)) = \frac{1}{2}\sum_{i=1}^2 e_i^2(k) \quad (14)$$

Os parâmetros  $K_{D_i}$ ,  $K_{I_i}$ ,  $K_{P_i}$  para o controlador PID neuronal podem ser ajustados usando o método do gradiente descendente, sendo as regras de atualização dos parâmetros do controlador são:

$$\mathbf{K}_{\theta_i}(k+1) = \mathbf{K}_{\theta_i}(k) + \Delta\mathbf{K}_{\theta_i}(k) = \begin{pmatrix} K_{P_i}(k+1) \\ K_{I_i}(k+1) \\ K_{D_i}(k+1) \end{pmatrix} = \begin{pmatrix} K_{P_i}(k) \\ K_{I_i}(k) \\ K_{D_i}(k) \end{pmatrix} - \begin{pmatrix} \eta_{P_i}(\partial E(k)/\partial K_{P_i}) \\ \eta_{I_i}(\partial E(k)/\partial K_{I_i}) \\ \eta_{D_i}(\partial E(k)/\partial K_{D_i}) \end{pmatrix} \quad (15)$$

Onde os valores de  $\eta_{P_i, I_i, D_i}$  são os fatores de aprendizagem para o controlador PID neuronal. Da equação (15) usando a regra da cadeia são obtidas as seguintes equações:

$$\begin{aligned}\frac{\partial E(k)}{\partial K_{P_i}} &= \frac{\partial E(k)}{\partial x_i} \frac{\partial x_i}{\partial u_i(k)} \frac{\partial u_i(k)}{\partial o_i(k)} \frac{\partial o_i(k)}{\partial K_{P_i}} \\ \frac{\partial E(k)}{\partial K_{I_i}} &= \frac{\partial E(k)}{\partial x_i} \frac{\partial x_i}{\partial u_i(k)} \frac{\partial u_i(k)}{\partial o_i(k)} \frac{\partial o_i(k)}{\partial K_{I_i}} \\ \frac{\partial E(k)}{\partial K_{D_i}} &= \frac{\partial E(k)}{\partial x_i} \frac{\partial x_i}{\partial u_i(k)} \frac{\partial u_i(k)}{\partial o_i(k)} \frac{\partial o_i(k)}{\partial K_{D_i}}\end{aligned}\quad (16)$$

Os valores das derivadas parciais da regra da cadeia são as seguintes:

$$\frac{\partial E(k)}{\partial x_i(k)} = -e_i(k) \quad (17)$$

No caso de  $\partial x_i/\partial u_i$  a dinâmica é desconhecida e deverá ser aproximada pela relação  $sign(\Delta x_i/\Delta u_i)$  que é conhecida como método indireto de aproximação.

$$\frac{\partial \hat{x}_i}{\partial u_i(k)} = sign\left(\frac{\Delta x_i}{\Delta u_i(k)}\right) \quad (18)$$

A relação  $\partial u_i/\partial o_i$  é a derivada da função de ativação  $tanh$ , esta derivada só tem validade na região linear da função, onde a ação de controle não produz níveis que podem fazer dano ou ruptura à dinâmica do robô, sendo a derivada:

$$\frac{\partial u_i(k)}{\partial o_i(k)} = \frac{\partial f(o_i(k))}{\partial o_i(k)} = \left[1 - f^2(o_i(k))\right] \quad (19)$$

Agora considerando as derivadas parciais com respeito aos ganhos do controlador as quais são:

$$\begin{cases} \frac{\partial o_i(k)}{\partial K_{P_i}} = f\left(e_i(k) - e_i(k-1)\right) \\ \frac{\partial o_i(k)}{\partial K_{I_i}} = f\left(e_i(k) T_0\right) \\ \frac{\partial o_i(k)}{\partial K_{D_i}} = f\left(\frac{e_i(k) - 2e_i(k-1) + e_i(k-2)}{T_0}\right) \end{cases} \quad (20)$$

E as expressões obtidas da eq. (19) ficam como:

$$\begin{aligned}K_{P_i}(k+1) &= K_{P_i}(k) - \eta_{P_i} e_i(k) \left[ sign\left(\frac{\Delta x_i}{\Delta u_i(k)}\right) \dots \right. \\ &\quad \left. f'(o_i(k)) \cdot \left[ f\left(e_i(k) - e_i(k-1)\right) \right] \right] \\ K_{I_i}(k+1) &= K_{I_i}(k) - \eta_{I_i} e_i(k) \left[ sign\left(\frac{\Delta x_i}{\Delta u_i(k)}\right) \dots \right. \\ &\quad \left. f'(o_i(k)) \cdot \left[ f\left(e_i(k) T_0\right) \right] \right] \\ K_{D_i}(k+1) &= K_{D_i}(k) - \eta_{D_i} e_i(k) \left[ sign\left(\frac{\Delta x_i}{\Delta u_i(k)}\right) \dots \right. \\ &\quad \left. f'(o_i(k)) \cdot \left[ f\left(\frac{e_i(k) - 2e_i(k-1) + e_i(k-2)}{T_0}\right) \right] \right]\end{aligned}\quad (21)$$

A análise desta técnica de controle é mostrada na seção seguinte.

## V. ANALISE DE ESTABILIDADE DO CONTROLADOR.

Definindo uma função definida positiva discreta,

$$V(k) = \frac{1}{2} \sum_{i=1}^2 e_i^2(k) \quad (22)$$

Calculando a diferença discreta da função de Lyapunov,

$$\begin{aligned}\Delta V(k) &= V(k+1) - V(k) = \frac{1}{2} \sum_{i=1}^2 [e_i^2(k+1) - e_i^2(k)] \\ &= \Delta V_1(k) + \Delta V_2(k) = \dots \\ &= \frac{1}{2} [e_1^2(k+1) - e_1^2(k)] + \frac{1}{2} [e_2^2(k+1) - e_2^2(k)]\end{aligned}\quad (23)$$

O valor de  $e(k+1)$  pode ser calculado pela seguinte equação:

$$e_i(k+1) = e_i(k) + \Delta e_i(k) \quad (24)$$

Onde a diferença do erro  $\Delta e(k)$  no processo de aprendizagem pode ser expressa pela equação:

$$\Delta e_i(k) = \frac{\partial e_i(k)}{\partial x_i(k)} \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \Delta \mathbf{K}_{\theta_i}^T \quad (25)$$

Considerando a eq.(25), sendo  $\Delta \mathbf{K}_{\theta_i} = [\Delta K_{P_i}, \Delta K_{D_i}, \Delta K_{I_i}]$  a matriz de parâmetros da rede PID neuronal para a variável de saída  $x_i$  (linear ou angular). Fazendo a análise para uma variável de saída e substituindo (25) em (24) é obtida:

$$\begin{aligned}\Delta V_i(k) &= e_i(k) \Delta e_i(k) + \frac{1}{2} \Delta e_i^2(k) = \dots \\ &= e_i(k) \frac{\partial e_i(k)}{\partial x_i(k)} \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \Delta \mathbf{K}_{\theta_i}^T + \frac{1}{2} \left( \frac{\partial e_i(k)}{\partial x_i(k)} \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \Delta \mathbf{K}_{\theta_i}^T \right)^2\end{aligned}\quad (26)$$

Substituindo o valor de  $\Delta \mathbf{K}_{\theta_i}$  na eq. (26):

$$\begin{aligned}\Delta V_i(k) &= e_i(k) \frac{\partial e_i(k)}{\partial x_i(k)} \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \eta_{\theta_i} e_i(k) \left( \frac{\partial e_i(k)}{\partial \mathbf{K}_{\theta_i}} \right)^T + \dots \\ &\quad + \frac{1}{2} \left( \frac{\partial e_i(k)}{\partial x_i(k)} \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \eta_{\theta_i} e_i(k) \left( \frac{\partial e_i(k)}{\partial \mathbf{K}_{\theta_i}} \right)^T \right)^2\end{aligned}\quad (27)$$

Reorganizando a equação (27) obtém-se:

$$\begin{aligned}\Delta V_i(k) &= -\eta_{\theta_i} e_i^2(k) \left( \frac{\partial e_i(k)}{\partial i(k)} \right)^2 \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \left( \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \right)^T + \dots \\ &\quad + \frac{1}{2} \eta_{\theta_i}^2 e_i^2(k) \left( \frac{\partial e_i(k)}{\partial x_i(k)} \right)^4 \left( \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \left( \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \right)^T \right)^4\end{aligned}\quad (28)$$

Da equação (28) para que  $\Delta V_i(k)$  seja menor que zero.

$$\begin{aligned}\Delta V_i(k) &= \left[ -1 + \frac{1}{2} \eta_{\theta_i} \left( \frac{\partial e_i(k)}{\partial x_i(k)} \right)^2 \left\| \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \right\|^2 \right] \dots \\ &\quad \dots \eta_{\theta_i} e_i^2(k) \left( \frac{\partial e_i(k)}{\partial x_i(k)} \right)^2 \left\| \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \right\|^2 < 0\end{aligned}\quad (29)$$

Da eq. (29) deve-se cumprir que:

$$\frac{1}{2} \eta_{\theta_i} \left\| \frac{\partial e_i(k)}{\partial x_i(k)} \right\|^2 \left\| \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \right\|^2 < 1 \quad (30)$$

Onde os valores de  $\eta_{P_i, I_i, D_i} < 1$  e

$$\left( \frac{\partial e_i(k)}{\partial x_i(k)} \right)^2 = 1 \quad \left\| \frac{\partial x_i(k)}{\partial \mathbf{K}_{\theta_i}} \right\|^2 = \left| \frac{\partial x_i(k)}{\partial u_i} \right|^2 \left| \frac{\partial u_i(k)}{\partial o_i(k)} \right|^2 \left\| \frac{\partial o_i(k)}{\partial \mathbf{K}_{\theta_i}} \right\|^2 \quad (31)$$

Sendo

$$\begin{aligned}\left| \frac{\partial x_i(k)}{\partial u_i} \right|^2 &= \left| sign\left(\frac{\Delta x_i}{\Delta u_i(k)}\right) \right|^2 = 1 \quad \left| \frac{\partial u_i(k)}{\partial o_i(k)} \right|^2 = \left| f'(o_i(k)) \right|^2 \leq 1 \\ \left\| \frac{\partial o_i(k)}{\partial \mathbf{K}_{\theta_i}} \right\|^2 &= \left( f\left(e_i(k) - e_i(k-1)\right) \right)^2 + \left( f\left(e_i(k) T_0\right) \right)^2 + \dots \\ &\quad + \left( f\left(\frac{e_i(k) - 2e_i(k-1) + e_i(k-2)}{T_0}\right) \right)^2 \leq 3|\delta_i|^2\end{aligned}\quad (32)$$

De acordo com (30) a (32)

$$\eta_{\theta_i} 3|\delta_{iMax}|^2 < 1 \quad \text{and} \quad \frac{1}{3|\delta_{iMax}|^2} > \eta_{\theta_i} > 0 \quad (33)$$

O que constitui um resultado prático que permite afirmar que o erro de controle de posição seja finalmente limitado em

termos do erro de aproximação do PID neuronal.



Figura 4. Robô móvel Pioneer 2DX.

## VI. RESULTADOS EXPERIMENTAIS E DISCUÇÃO

O robô móvel Pioneer 2-DX possui a bordo um PC Pentium III (Figura 4), os algoritmos de controle proposto aplicaram-se no robô que admite velocidades lineares e angulares, como os sinais de entrada de referência.

No experimento o robô deve seguir uma trajetória, definida pela equação (34), a técnica proposta será comparada com um controle PID estático (sem ajuste de ganhos). A Fig. 5 mostra as ações de controle sob o robô. Na Fig. 6 mostra as referências  $r_x$  e  $r_y$  e a trajetória seguida pelo robô móvel.

A Fig. 7 mostra a trajetória seguida pelo robô no plano XY no início da trajetória o erro é maior pela falta de ajuste dos ganhos do controlador PID neuronal, pode se ver que o erro do controle PID estático (sem ajuste de ganhos) é maior no percorrido da trajetória. A Fig. 8 a evolução do erro quadrático meio no percorrido da trajetória, entre o controle PID neuronal e o controle PID estático. A evolução do ajuste dos ganhos do controlador PID é mostrada na Fig. 9. Os resultados obtidos demonstram a viabilidade da aplicação do algoritmo PID neuronal em processos não lineares.

$$\begin{pmatrix} r_{xref} & r_{yref} \end{pmatrix}^T = \begin{pmatrix} 1.5 \sin\left(\frac{\pi k T_0}{40}\right) & 0.4 \sin\left(\frac{\pi k T_0}{20}\right) \end{pmatrix}^T \quad (34)$$

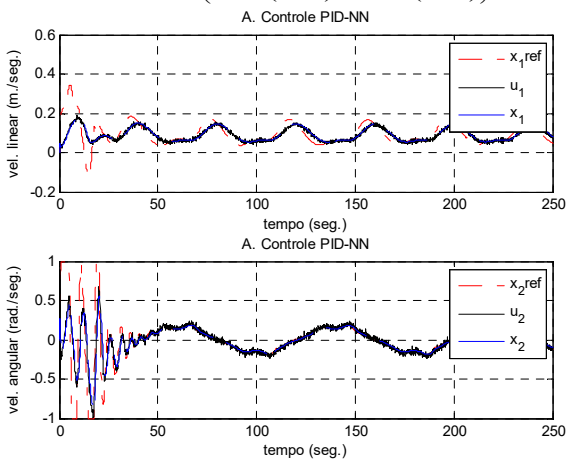


Figura 5. Referências e ações de controle.

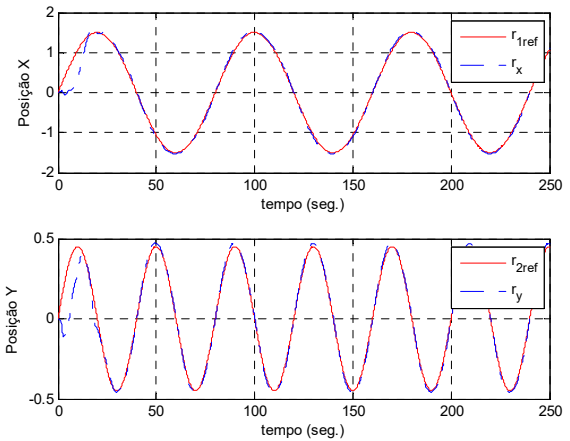


Figura 6. Sinais de referência de posição e saída de posição do robô móvel.

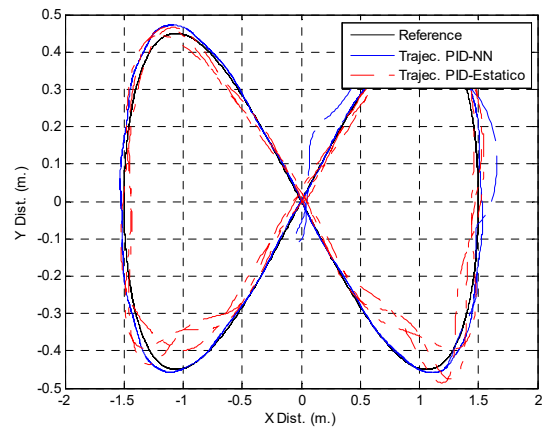


Figura 7. Trajetória seguida pelo robô móvel com controlador PID-NN (azul) e controlador PID estático (vermelho).

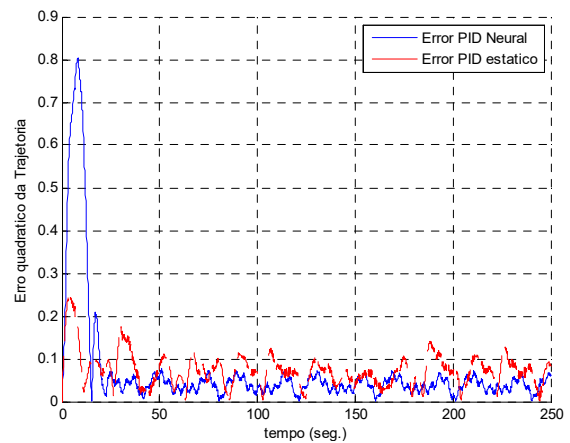


Figura 8. Erro quadrático instantâneo da posição do robô entre o controlador PID - NN (azul) e o controlador PID estático (vermelho).

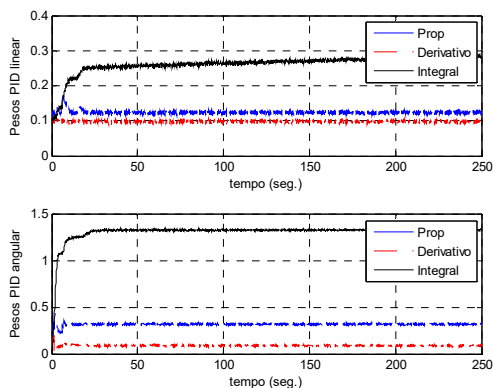


Figura 9. Evolução temporal dos pesos da rede PID neural durante o experimento.

O controlador PID neuronal adaptável demonstra ser robusto com respeito aos erros de modelagem, além muito eficaz na rejeição de perturbações, e não produz nenhum erro constante causada por qualquer dos parâmetros das incertezas ou perturbações externas. Por outro lado o controlador PID estático é vulnerável as mudanças na dinâmica e as incertezas do modelo, já que o controle PID estático esta desenhado para um modelo linearizado da dinâmica do robô. A lei de controle desenvolvida neste trabalho para um sistema não linear não precisa do modelo dinâmico do robô. Sendo estas incertezas e não linearidades do modelo e as variações na dinâmica do robô demonstram a robustez do controlador PID neuronal adaptável. O tema da estabilidade global do sistema de laço fechado foi demonstrado analiticamente a través da teoria de estabilidade de Lyapunov (Secção V). Esta proposta de controle inteligente pode-se considerar como uma solução geral para o controle de sistemas não lineares e em particular para o caso de sistemas robóticos, quando a dinâmica é variável o possuem incertezas no modelo.

## VII. CONCLUSÕES

Neste trabalho, propõe-se um controlador PID neuronal adaptável para seguimento de trajetória de robô móvel do tipo unicycle, se demostrou que os erros de controle de seguimento estão delimitados, e os limites são calculados em função do erro de aproximação PID neuronal.

Uma grande importância do controlador PID neuronal adaptável proposto é sua robustez frente às incertezas do modelo e grande redução do erro de controle maior que o controlador PID estático convencional, sua simples aplicação aos problemas de engenharia. Os resultados experimentais mostram um bom desempenho do controlador PID proposto que podem ser aplicados a qualquer processo com dinâmicas não lineares, entre eles as dinâmicas dos robôs móveis e industriais.

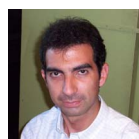
## REFERÊNCIAS

[1] Y. Jin, Decentralized Adaptive Fuzzy control of Robot Manipulador, IEEE Transactions on Systems, Man and Cybernetics., Part B, 28(1):47-57,1998.  
 [2] Spong, M. W., & Vidyasagar, M. (2008). *Robot dynamics and control*. John Wiley & Sons..

[3] Li, S.-D., Zhu, J.: Optimization of MEO Regional Communication Satellite Constellation with Genetic Algorithm. *Journal of System Simulation*, 17(6), 1366–1470 (2005).  
 [4] Jin, J., Su, Y.: A improved adaptive genetic algorithm. *Computer Engineering and Application*, 29(3), 64–70 (2005).  
 [5] Precup, R.-E., Preitl, S., Faur, G.: PI predictive fuzzy controllers for electrical drive speed control: methods and software for stable development. *Computers in Industry* 52, 253–270 (2003).  
 [6] Ding, Y., Ying, H., Shao, S.: Typical Takagi-Sugeno PI and PD fuzzy controllers: analytical structures and stability analysis. *Information Sciences* 151, 245–262 (2003).  
 [7] Ahn, K.K., Truong, D.Q.: Online tuning fuzzy PID controller using robust extended Kalman filter. *Journal of Process Control* 19, 1011–1023 (2009).  
 [8] Lu W., Yang J.H., Liu X. D., The PID Controller Based on the Artificial Neural Network and the Differential Evolution Algorithm, *Journal of Computers*, Vol 7, No 10 (2012), 2368-2375, Oct 2012  
 [9] Mahmud, K., "Neural network based PID control analysis," *Global High Tech Congress on Electronics (GHTCE)*, 2013 IEEE , vol., no., pp.141,145, 17-19, Nov. 2013.  
 [10] Gan, S.-C., Yang, P.-X.: PID self-tuning based on fuzzy genetic algorithm. *Journal of North China Electric Power University* 32(5), 43–46 (2005).  
 [11] Mahony, T.O., Downing, C.J., Fatla, K.: Genetic Algorithm for PID Parameter Optimization: Minimizing Error Criteria. In: *Process Control and Instrumentation*, July 26-28, pp. 148–153. University of Strathclyde (2000).  
 [12] He, G., Tan, G.: An Optimal Nonlinear PID Controller Based on Ant Algorithm. *Programmable Controller & Factory Automation*, 99–105 (2007).  
 [13] Kennedy, J., Eberhart, R.C.: *Swarm Intelligence*. Morgan Kaufmann Publishers (2001)  
 [14] Gaing, Z.L.: A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR system. *IEEE Trans. on Energy Conversion* 19(2), 384–391  
 [15] Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: *Proc. IEEE Int. Conf., Evolution Computer*, Anchorage, AK, pp. 69–73 (1998)  
 [16] Rutenbar, R.A.: Simulated Annealing Algorithms: An overview. *IEEE Circuits and Devices Magazine* 5(1), 19–26 (1989)  
 [17] Li, Y., Wang, Z., and Zhu, L. (2010) Adaptive Neural Network PID Sliding Mode Dynamic Control of Nonholonomic Mobile Robot, *IEEE International Conference on Information and Automation (ICIA)*, pp. 753–757.  
 [18] Rossomando, F.G., Soria, C., Carelli, R.: 'Autonomous mobile robot navigation using RBF neural compensator', *Control Eng. Pract.*, Elsevier, 2010, 19, (3), pp. 215–222



**Francisco Guido Rossomando**, é desde março de 2008, funcionário no Governo na Província de San Juan, atualmente é Sub-secretário de Promoção da Atividade Científica, San Juan, Argentina. Recebeu o título de Doutor em Engenharia Elétrica menção Automação pela Universidade Federal de Espirito Santo (UFES), Brasil, em 2006; Mestre em Engenharia em Controle de Processos pela Universidade Nacional de San Juan (UNSJ), San Juan, Argentina, em 2002; e graduado em Engenharia Eletrônica pela Universidade Nacional de San Juan (UNSJ), San Juan, Argentina, em 1998. Também é pesquisador adjunto do CONICET, Argentina. Atualmente suas pesquisas se concentram na área de controle adaptável e redes neurais artificiais, aplicadas aos processos industriais e robótica.



**Carlos Miguel Soria**, nasceu na Província de Tucumán, Argentina. No ano 1996 recebeu o título de Engenheiro Eletricista da Faculdade de Ciências Exatas da Universidade Nacional de Tucumán (UNT) no Estado do mesmo nome, Argentina. No ano 2000 recebeu o título de Magister em Sistemas de Controle na Faculdade de Engenharia da Universidade Nacional de San Juan (UNSJ). Recebeu o título de Doutor em Sistemas de Controle pela Universidade Nacional de San Juan (UNSJ), Argentina, em 2005. Foi bolsista de pós-doutorado e atualmente é pesquisador adjunto no CONICET (Consejo Nacional de Investigaciones Científicas Tecnológicas).