



APLICAÇÃO DE REDES NEURAIS NA GERAÇÃO DE TRAJETÓRIAS PARA ROBÔS MÓVEIS

Bruno Garrido de Menezes *
Fabício Roulin Bittencourt °
Luis Enrique Zárate °
Marcelo Becker *

(*) IPUC – PUC Minas, Av. Dom José Gaspar, 500 - Belo Horizonte - MG – CEP 30535-610, Brasil, e-mail: marcelo.becker@pucminas.br

(°) Departamento de Ciências da Computação – PUC Minas, Av. Dom José Gaspar, 500 - Belo Horizonte - MG – CEP 30535-610, Brasil, e-mail: zarate@pucminas.br

Resumo. Neste trabalho é apresentada uma aplicação de Redes Neurais Artificiais para a geração de trajetórias de robôs móveis após memorização de um modelo de trajetória do ambiente. A rede neural permite prever a próxima localização do robô, através do par de coordenadas (x,y), a partir de posicionamentos anteriores. O resultado mostra o bom desempenho da rede para prever o novo par de coordenadas, em condições de erro no posicionamento do robô. Aborda-se também no artigo a possibilidade de introduzir memória neural no robô para deslocamentos em ambientes estáticos ou controlados, onde possa ser identificada previamente a trajetória ótima do robô móvel. Também é discutida a possibilidade de uso da estrutura apresentada em sistemas de controle preditivo. Ao final do artigo são apresentados resultados de simulação.

Palavras-chave: Robôs Móveis, Geração de Trajetórias, Redes Neurais, Controle Preditivo.

1. INTRODUÇÃO

Robôs móveis são veículos (sistemas programáveis) capazes de realizar múltiplas tarefas. Nos últimos anos têm sido aplicados nas mais diversas áreas, por exemplo: manuseio de materiais perigosos, combate a incêndios e calamidades, resgates, transporte de cargas em fábricas, exploração espacial, agricultura, etc. A tendência é que o uso de robôs móveis (autônomos ou não) expanda-se nos próximos anos em áreas como a mineração, transporte de cargas, agricultura e assistência a portadores de deficiências físicas (Becker, 2001).

Desde a década de 1960, quando o robô Hilare do LAAS (*Laboratoire d'Automatique et d'Analyse des Systemes* - França) causou furor por adotar a filosofia de dividir as tarefas de controle em blocos, um grande número de projetos surgiram, introduzindo diversos algoritmos, técnicas de processamento sensorial e de interpretação do ambiente, novas formas construtivas. Todo este desenvolvimento vem aumentando cada vez mais o número de aplicações de robôs móveis. Mas, qualquer que seja a sua aplicação, ele possui um sério problema de navegação que pode ser resumido pelas três seguintes questões: “Onde o robô está?”, “Onde ele está indo?” e “Como ele pode chegar lá?” (Leonard e Durrant-Whyte, 1992).

A primeira questão é um problema de localização. As demais são essencialmente de especificação do objetivo e de capacidade de planejar uma trajetória no ambiente para atingir este objetivo, evitando qualquer obstáculo que esteja em seu percurso. Baseado nestas três questões fundamentais da robótica móvel, têm-se as três áreas de pesquisa de navegação de robôs, em inglês: “localization”, “path planning” e “obstacle avoidance”. Para que um robô possa se localizar em um

ambiente, é desejável que ele faça ou possua um mapa deste ambiente. Logo, o mapeamento, em inglês, “*Map-Building*” é uma atividade precursora à localização.

O planejamento de trajetórias (“*Path Planning*”) é um tópico que pode ser subdividido em três grandes áreas de pesquisa: planejamento global de trajetórias (Brooks, 1986, entre outros), baseado em informações completas já disponíveis do ambiente; planejamento local, baseado em dados provenientes de sensores que monitoram o ambiente em tempo real (Becker, 2001, entre outros); e planejamento híbrido, onde uma trajetória ótima é preestabelecida e sensores embarcados no robô monitoram o ambiente (Carvalho *et al.*, 1999 entre outros). Métodos de planejamento globais de trajetórias podem resolver problemas de navegação em ambientes completamente conhecidos e estáticos, mas não são recomendados para ambientes desconhecidos, ou dinâmicos, ambientes difíceis de serem modelados que se alteram com o tempo, possuindo obstáculos desconhecidos que devem ser detectados para evitar colisões (Beom e Cho, 1995 e Fabro, *et.al.*, 2001). Nestes casos é necessário utilizar métodos que utilizam informações em tempo real do ambiente devem ser utilizados no navegador do veículo, ou seja, métodos de planejamento local e híbridos. O presente trabalho foca métodos de planejamento de trajetória híbridos.

Métodos híbridos referem-se à combinação de um modelo de trajetória preestabelecida com navegação por sensores. Nesta abordagem, a partir do planejamento da trajetória, que pode estar completo ou incompleto, os sensores informam sobre a presença de obstáculos que podem estar na trajetória planejada. Um exemplo da aplicação destes métodos pode ser encontrada na utilização de robôs móveis com sensores embarcados em ambientes monitorados por camera de video. Desta forma, através do processamento das informações dos dados da camera, é possível gerar um mapa e uma trajetória ótima preestabelecida. Os sensores embarcados, por sua vez, geram as informações necessárias para o procedimento de auto-localização e desvio de obstáculos nesse ambiente (Carvalho *et al.*, 1999).

As dificuldades existentes na procura de soluções mais adequadas para a navegação de robôs têm colocado o foco da atenção em técnicas de Inteligência Artificial (IA). Dentro da IA, Redes Neurais Artificiais (RNA) têm recebido uma grande atenção para solucionar problemas não lineares por aprendizado. Como as RNAs são baseadas no neurônio biológico os quais são capazes de associar e memorizar dados e informação, neste trabalho é apresentada uma estrutura de memorização neural capaz de gerar trajetórias previamente planejadas que pode ser combinada em futuros trabalhos com sensores para evitar colisões com obstáculos.

A literatura mostra que robôs móveis apresentam dificuldades ao se depararem com configurações do tipo “U”. Na presença de tais configurações, as trajetória geradas podem corresponder a ciclos infinitos (Melo e Figueiredo, 2001). A configuração proposta mostra o bom desempenho da estrutura na presença de desvios no posicionamento do robô e de trajetórias planejadas complexas. A estratégia consiste em gerar o novo par de coordenadas (x,y) a partir de posicionamentos anteriores conhecidos do robô. Esta estratégia pode ser usada em sistemas de controle preditivo.

2. ESTRUTURA NEURAL PARA MEMORIZAÇÃO DE TRAJETÓRIAS

2.1. Modelo de Trajetória para a Representação Neural

O ambiente de simulação de trajetórias considerado neste trabalho, é baseado no 1º quadrante do plano de coordenadas cartesiano (x,y) . Neste plano, uma trajetória que leva ao robô do seu ponto inicial até seu ponto final pode ser definida genericamente como:

$$c(X,Y) = \{ (x,y) / x \in \mathfrak{R}^+ \geq 0 \text{ e } y \in \mathfrak{R}^+ \geq 0 \} \quad (1)$$

Conhecidos *à priori* todos os pontos (x,y) , da Eq. (1), o objetivo é treinar uma RNA capaz de memorizar tal trajetória. Quando a rede estiver treinada e posta em operação o objetivo é reconstruir a trajetória a partir de posições anteriores do robô. Para isto podem ser definida duas estratégias:

$$\text{Estrat\u00e9gia 1: } \left\{ \begin{array}{l} \hat{y}(t+k) = f(y(t), y(t-1), \dots, y(t-h)) \\ \hat{x}(t+k) = f(x(t), x(t-1), \dots, x(t-h)) \end{array} \right. \quad (2)$$

$$(3)$$

$$\text{Estrat\u00e9gia 2: } (\hat{y}(t+k), \hat{x}(t+k)) = f(y(t), y(t-1), \dots, y(t-h), x(t), x(t-1), \dots, x(t-h)) \quad (4)$$

onde: \hat{y} : \u00e9 a coordenada estimada para y ; \hat{x} : \u00e9 a coordenada estimada para x ; k : \u00e9 o horizonte de estimat\u00e3o, considerado neste trabalho como $k = 1$; e h : \u00e9 a ordem do estimador, considerada neste trabalho como $h = 10$.

As Eqs. (2) e (3) correspondem a duas estruturas independentes de estimat\u00e3o para x e y respectivamente, j\u00e1 a Eq. (4) corresponde a uma estrutura conjunta para estimar de x e y . Observe que dependendo do valor de k ($k = 1, 2, \dots$), as estruturas podem ter um comportamento preditivo capaz de ser inserida num sistema de controle preditivo para trajet\u00f3rias. Tema a ser abordado em trabalhos futuros.

2.2. Redes Neurais Artificiais e a Estrutura de Memoriza\u00e7\u00e3o

Nos \u00faltimos anos RNAs t\u00eam sido propostas como poderosas ferramentas computacionais devido aos baixos tempos de processamento que podem ser alcan\u00e7ados quando a rede j\u00e1 treinada est\u00e1 em opera\u00e7\u00e3o.

Para cada estrat\u00e9gia abordada no item anterior uma representa\u00e7\u00e3o neural ser\u00e1 aplicada. Para a *estrat\u00e9gia 1*, a estrutura de estimat\u00e3o \u00e9 formada por duas redes neurais, Eqs. (5) e (6), com 10 entradas e 1 sa\u00edda cada uma.

$$(y(t), y(t-1), \dots, y(t-10)) \xrightarrow{\text{Rede Neural}} (\hat{y}(t+1)) \quad (5)$$

$$(x(t), x(t-1), \dots, x(t-10)) \xrightarrow{\text{Rede Neural}} (\hat{x}(t+1)) \quad (6)$$

Para a *estrat\u00e9gia 2*, a estrutura de estimat\u00e3o \u00e9 formada por uma \u00fanica rede Eq. (7), com 20 entradas e 2 sa\u00eddas.

$$(y(t), y(t-1), \dots, y(t-10), x(t), x(t-1), \dots, x(t-10)) \xrightarrow{\text{Rede Neural}} (\hat{y}(t+1), \hat{x}(t+1)) \quad (7)$$

As RNAs utilizadas neste trabalho correspondem a uma rede multicamadas, a qual se aproxima das teorias conexionistas que procuram descrever o funcionamento do c\u00e9rebro. O tipo de aprendizado desta rede \u00e9 conhecido como *aprendizado supervisionado*, baseada no sistema *retropropaga\u00e7\u00e3o do erro* (*“error backpropagation”*). As RNAs consideradas neste trabalho, s\u00e3o redes de duas camadas, onde o n\u00famero de neur\u00f4nios da camada escondida foi escolhido como sendo igual a $(2N+1)$ neur\u00f4nios, onde N \u00e9 o n\u00famero de entradas da rede. O n\u00famero de neur\u00f4nios na camada de sa\u00edda, s\u00e3o iguais a 1 e a 2, para a estrat\u00e9gia 1 e 2, respectivamente. A fun\u00e7\u00e3o n\u00e3o linear sigm\u00f3ide, foi escolhida neste trabalho como a fun\u00e7\u00e3o de transfer\u00eancia do ax\u00f4nio, por ser a mais consistente com a biof\u00edsica do neur\u00f4nio biol\u00f3gico (Verschure, 1993).

Uma RNA com aprendizado do tipo *retropropaga\u00e7\u00e3o do erro* utiliza um padr\u00e3o de dados na entrada da rede e uma sa\u00edda desejada. Para cada conjunto padr\u00e3o os pesos da rede s\u00e3o ajustados para minimizar a diferen\u00e7a entre as sa\u00eddas da rede e as desejadas. O erro \u00e9 minimizado utilizando a t\u00e9cnica do gradiente com um fator de converg\u00eancia chamado *taxa de aprendizagem*.

Para alimentar o processo de treinamento das RNAs foi gerada uma trajetória para o robô, com mudanças drásticas, a ser apresentada na próxima seção. Geralmente o maior esforço para o treinamento de uma rede neural encontra-se na coleta de dados e no pré-processamento deles. O pré-processamento consiste na normalização dos dados da entrada e da saída. Para o tipo de rede considerada o único requisito é que os valores das entradas e das saídas se encontrem no intervalo de 0 a 1. Os seguintes procedimentos foram adotados para normalizar os dados das entradas associados às suas respectivas saídas, antes de usá-los no treinamento da rede neural:

a) Como os valores de 0 e 1 são valores infinitos para a função sigmóide, é recomendável diminuir este intervalo para valores de 0,2 e 0,8 respectivamente, com o objetivo de facilitar a convergência durante o treinamento da rede.

b) Os dados foram normalizados e desnormalizados através das seguintes expressões:

$$f^a(L_o) = L_n = (L_o - L_{mín}) / (L_{máx} - L_{mín}) \quad (8-a)$$

$$f^b(L_n) = L_o = L_n * L_{máx} + (1 - L_n) * L_{mín} \quad (8-b)$$

onde : L_n é o valor normalizado; L_o o valor a normalizar; $L_{mín}$ e $L_{máx}$ são valores mínimos e máximos dentre os valores das variáveis.

c) Pelo exposto no item (a) faze-se necessário a mudança de escala dos dados, para valores que proporcionem quando normalizados, valores limites entre 0,2 e 0,8. As fórmulas para calcular $L_{mín}$ e $L_{máx}$ são as seguintes:

$$L_{mín} = (4 \cdot \text{Limite}_{\text{Inf.}} - \text{Limite}_{\text{Sup}}) / 3 \quad (9-a)$$

$$L_{máx} = (\text{Limite}_{\text{Inf.}} - 0,8 \cdot L_{mín}) / 0,2 \quad (9-b)$$

As Eqs. (9-a) e (9-b) são obtidas da substituição na Eq. (8-a) de $L_n = 0,2$ e $L_o = \text{Limite}_{\text{Inf}}$ e de $L_n = 0,8$ e $L_o = \text{Limite}_{\text{Sup}}$. Onde, $\text{Limite}_{\text{Inf}}$ e $\text{Limite}_{\text{Sup}}$ são os valores mínimos e máximos, respectivamente, dos conjuntos de dados originais.

3. MODELO DO AMBIENTE E SIMULAÇÃO

Para este trabalho foi criado um mapa do ambiente como sendo uma matriz de 800 x 800 posições, representando o mundo simulado de 0,640 m². A Eq. (10) mostra a trajetória gerada para navegação do robô neste ambiente:

$$\text{Trajetória}(x, y) = \begin{cases} A_o + B_o \sin\left(\frac{2P}{T_1} x\right) & , \quad 0 \leq x < 200 \\ A_o \leq y \leq A_o + B_o & , \quad x = 200 \\ (A_o + B_o) + C_o \sin\left(\frac{2P}{T_2} x\right) & , \quad 200 < x \leq 400 \\ A_o + B_o & , \quad 400 < x \leq 500 \\ A_o + B_o \leq y \leq A_o & , \quad x = 500 \\ A_o & , \quad 500 < x \leq 600 \\ A_o + D_o(x - 600) & , \quad 600 < x \leq 700 \\ E_o + F_o(x - 700)^2 & , \quad 700 < x \leq 800 \end{cases} \quad (10)$$

onde: $A_o = 400$, $B_o = 200$, $C_o = 100$, $D_o = 1$, $E_o = 300$, $F_o = 0,01$.

3.1. Obtenção dos Conjuntos de Treinamento

Os conjuntos de treinamento foram obtidos a partir da Eq. (10), gerando um total de 1200 valores para x e y . Para definir os conjuntos de treinamento é necessário considerar seqüências de valores para x e y como mostrado nas Eqs. (11) e (12).

$$\begin{bmatrix} y(t) & y(t+1) & y(t+2) & \cdots & y(t+9) \\ y(t+1) & y(t+2) & y(t+3) & \cdots & y(t+10) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y(t+1189) & y(t+1190) & y(t+1191) & \cdots & y(t+1198) \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} x(t) & x(t+1) & x(t+2) & \cdots & x(t+9) \\ x(t+1) & x(t+2) & x(t+3) & \cdots & x(t+10) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x(t+1189) & x(t+1190) & x(t+1191) & \cdots & x(t+1198) \end{bmatrix} \quad (12)$$

Para a *estratégia 1*, as saídas de ambas as redes são dadas pelas Eqs. (13) e (14):

$$[y(t+10) \quad y(t+11) \quad \cdots \quad y(t+1199)]^T \quad (13)$$

$$[x(t+10) \quad x(t+11) \quad \cdots \quad x(t+1199)]^T \quad (14)$$

Para a *estratégia 2*, as Eqs. (13) e (14) são combinadas para configurar as duas saídas da rede.

3.2. Treinamento da Rede Neural

Para ambas estratégias as RNAs consideradas correspondem a redes multicamadas tipo *perceptron* com 21 neurônios na camada escondida. Observa-se na Figura 1 a estrutura da RNA para a *estratégia 2*.

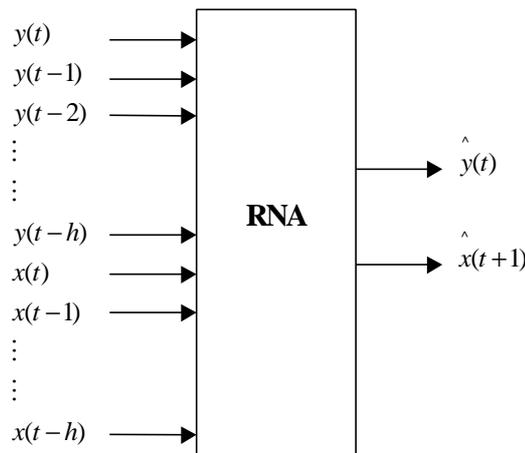


Figura 1. Estrutura da RNA para a *estratégia 1*.

Após a normalização dos conjuntos de dados e o treinamento das redes, foram obtidos os resultados mostrados na Tab. 1.

Tabela 1. Resultados obtidos para as *estratégias 1 e 2*.

<i>Estratégia</i>	Coordenada	N_d	N_{iter}	ϵ_m	ϵ_g
1	x	1.200	5.000	0,003500	0,0015900
	y	1.200	17.000	0,007900	0,0079600
2	x	1.200	200.000	0,000164	0,0011522
	y	1.200	200.000	0,004800	0,0011522

onde: N_d : é o número de dados utilizados no treinamento da RNA; N_{iter} : é o número de iterações necessárias para o treinamento da RNA; ϵ_m : é o erro médio ($\epsilon_{mx} = (\sum(x_{real} - x_{RNA}))/N_d$) e ($\epsilon_{my} = (\sum(y_{real} - y_{RNA}))/N_d$); e ϵ_g : é o erro global durante o processo de treinamento.

3.3 Resultados de Simulação

A Figura 2 mostra a geração da trajetória a partir da memorização das redes neurais treinadas para o caso da *estratégia 1*. Observa-se na Fig. (2-b) que até a coordenada $x < 200$ as redes geram praticamente as trajetórias sem apresentar grandes desvios. Para $x = 200$, Fig. (2-c), existe um prévio ajuste até a rede corrigir a trajetória. Neste caso, ambas coordenadas x e y estão sendo estimadas separadamente.

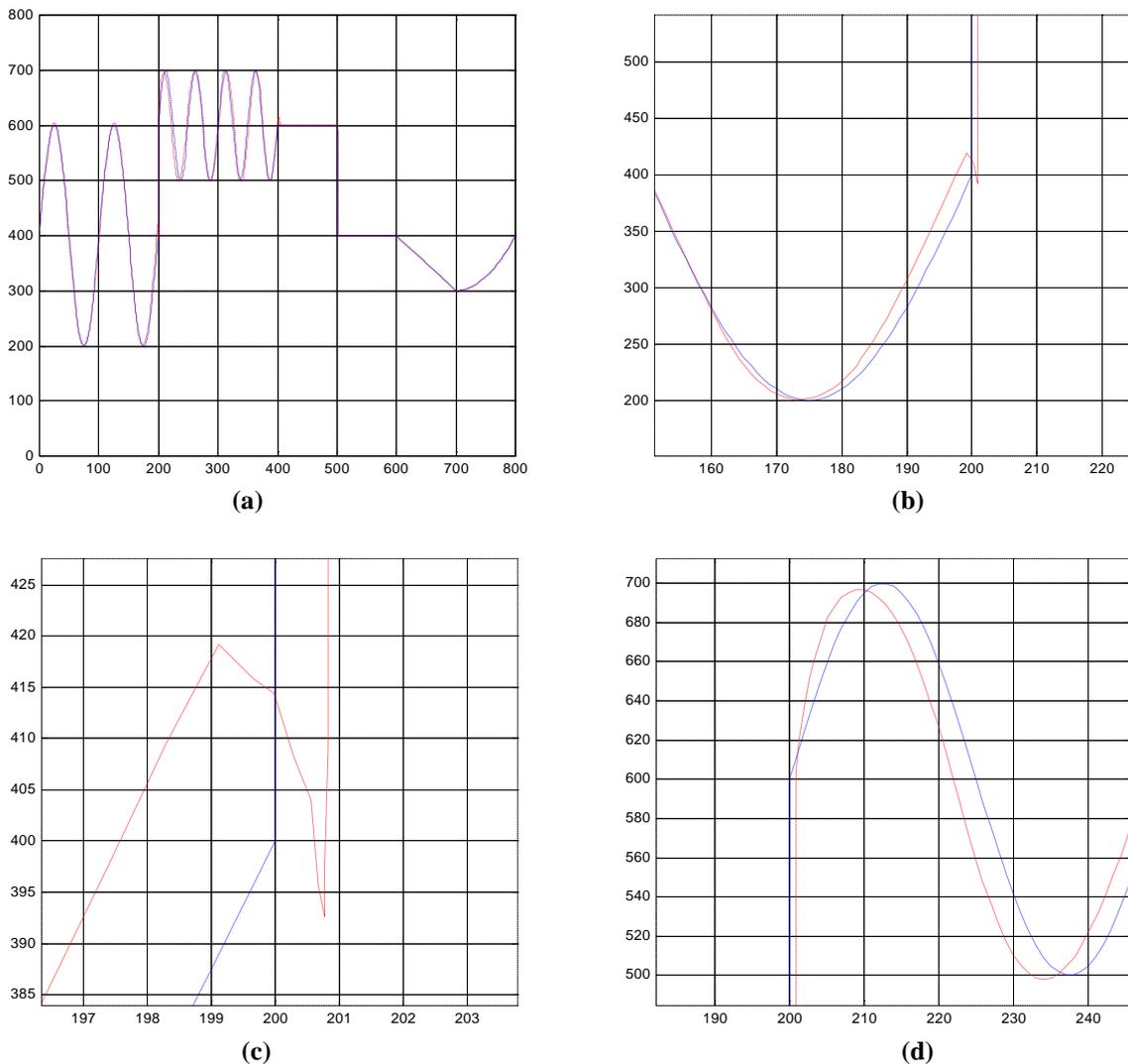


Figura 2. Geração de trajetórias para o caso da *estratégia 1*.

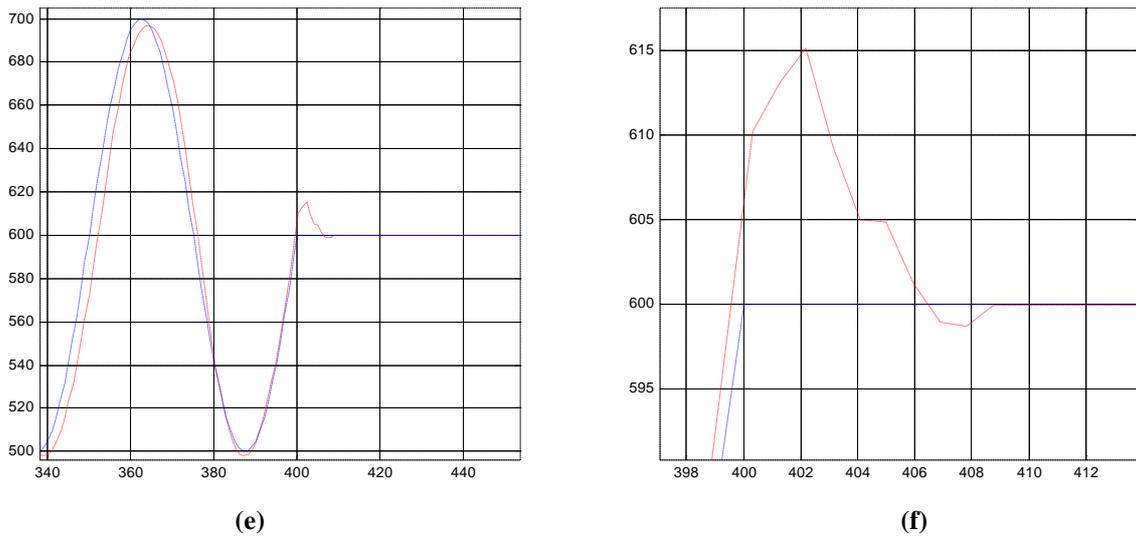


Figura 2. Geração de trajetórias para o caso da *estratégia 1* (continuação).

Para $x = 400$, observa-se um erro de 3% na mudança da trajetória na coordenada y (Fig. 2-f), sendo que as redes neurais conseguem se adaptar e apresentar um erro praticamente nulo na coordenada $x = 409$.

A Figura 3 mostra a geração da trajetória a partir da memorização da rede neural treinada para o caso da *estratégia 2*. Observe que em torno da coordenada x , próximo de $x = 400$ (Fig. 3-b) a RNA consegue gerar a trajetória, sem apresentar grandes desvios. Para $x = 400$ (Fig. 3-b), existe um erro significativo de 20 unidades na coordenada y , embora a trajetória seja alcançada em $x = 420$ (Fig. 3-b). Já no trecho de $450 < x < 700$ (Fig. 3-c) a RNA gera a trajetória memorizada com erro da ordem de $x = 3$ unidades em $y = 595$ (Fig. 3-d).

Por outro lado, a movimentação de robôs móveis está sujeita a erros de posicionamento devido principalmente a superfícies não regulares, desgaste não uniforme das rodas, escorregamentos do robô, etc. Isto pode alterar drasticamente a trajetória planejada evitando que o robô alcance seu alvo preestabelecido. De forma a verificar o desempenho da RNA na presença de erros de posicionamento nos deslocamentos anteriores, foi introduzido ruído no par de coordenadas (x,y) que a alimentam.

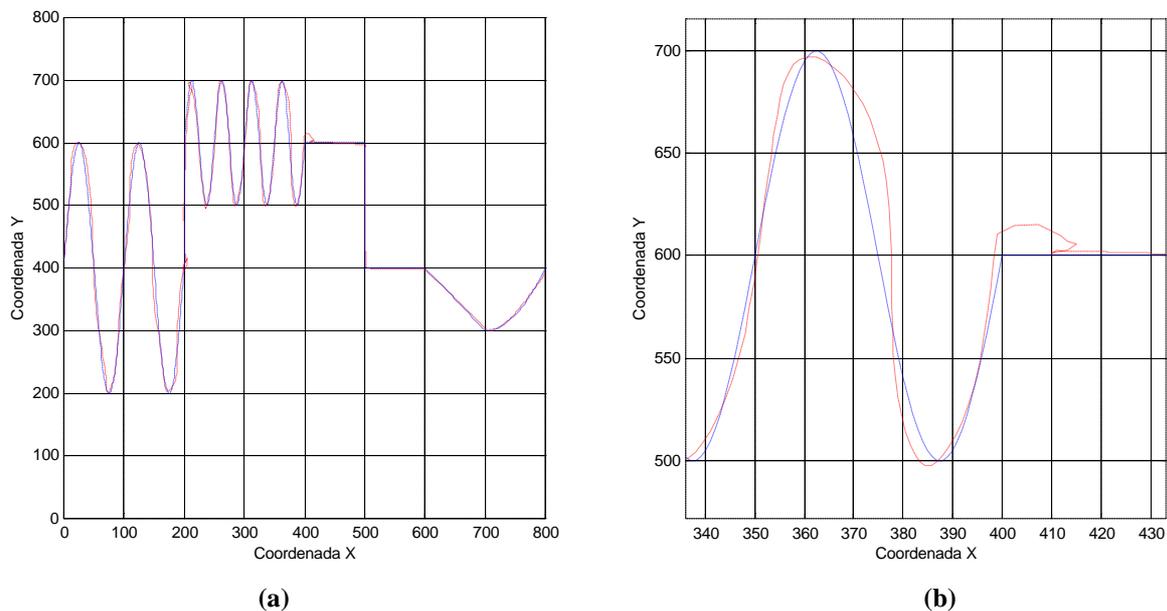
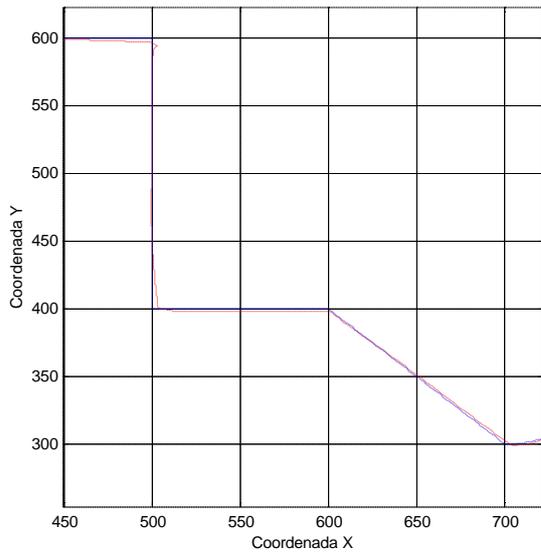
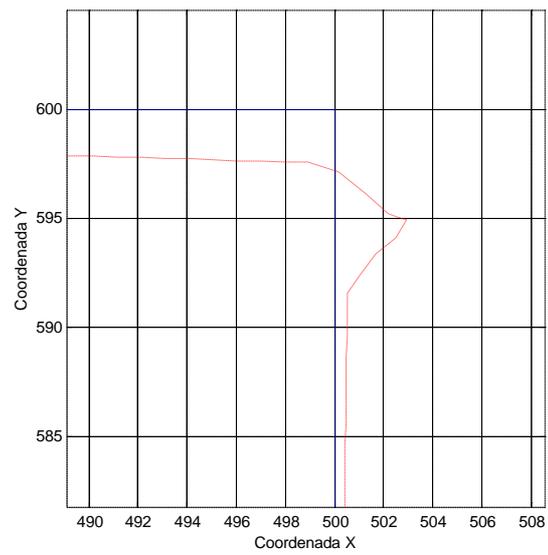


Figura 3. Geração de trajetórias para o caso da *estratégia 2*.

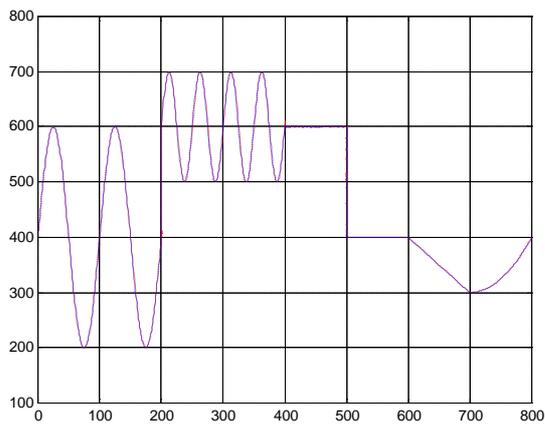


(c)

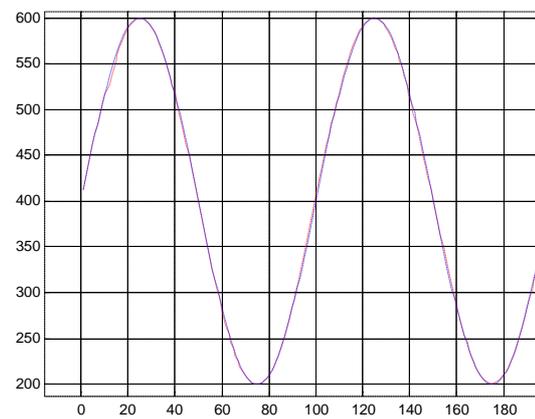


(d)

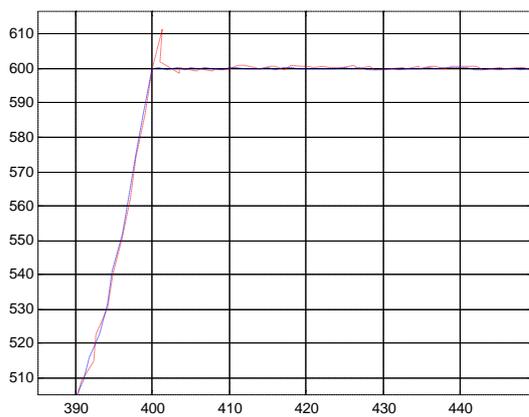
Figura 3. Geração de trajetórias para o caso da *estratégia 2* (continuação).



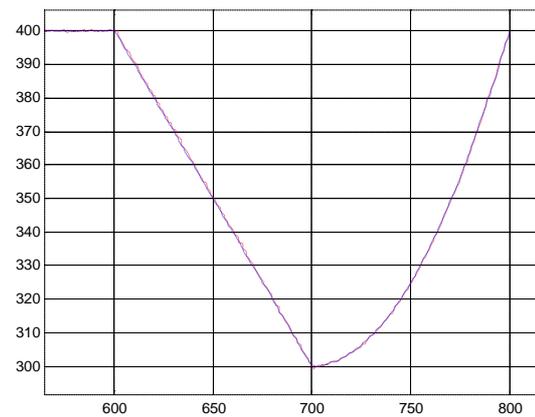
(a)



(b)



(c)



(d)

Figura 4. Geração de trajetórias na presença de ruído para o caso da *estratégia 2*.

Observa-se na Fig. (4) a trajetória gerada na presença de ruído através da RNA para o caso da *estratégia 2*. O ruído introduzido foi aleatório e variou na faixa de $\pm 0,05\%$. Deve-se observar que esta porcentagem corresponde a $0,05\%$ da unidade de referência para cada par de coordenadas

(x,y)). Por exemplo, para a coordenada $x = 700$, 0,05% corresponde a 0,35 da unidade de referência. Se a unidade for 1 m, o erro considerado é de 35 cm. Mesmo sendo esse erro significativo, ainda assim a RNA consegue reconstruir a trajetória memorizada.

Para a faixa de $0 < x < 180$ (Fig. 4-b) a trajetória é gerada praticamente sem erro algum. Para $x = 400$ (Fig. 4-c), houve um erro na coordenada y de 11 unidades voltando a trajetória após 1 (uma) unidade na coordenada x . Da faixa de $600 < x < 800$ (Fig. 4-d) a trajetória é gerada sem perturbação do ruído.

4. CONCLUSÕES E PERSPECTIVAS FUTURAS

Neste trabalho, a possibilidade do uso de redes neurais para geração de trajetórias para robôs móveis foi apresentada. O gerador de trajetória é baseado na memorização neural de uma trajetória planejada previamente que permite levar ao robô móvel de sua posição inicial até seu alvo (posição final). O gerador neural utiliza dados de posicionamentos anteriores trabalhando como um preditor do próximo par de coordenadas (x,y) . Observou-se que a estrutura proposta pode ser facilmente adaptada para prever a próxima coordenada num horizonte estendido de k unidades (ver Eqs. (2), (3) e (4)). Este resultado é promissor para que a mesma topologia de rede seja utilizada como um controlador preditivo, seja em aplicações de processos industriais ou em robótica móvel. Na presença de ruído observou-se o bom desempenho do gerador neural que atuou como um filtro, atenuando a influência da perturbação.

5. AGRADECIMENTOS

Os autores agradecem ao programa PIBIC/CNPq através do projeto FIP 2001/35-P pelo apoio financeiro.

6. REFERÊNCIAS

- Becker, M., 2001, "Obstacle avoidance procedures for mobile robots", XVI Brazilian Congress of Mechanical Engineering, Vol. 15, pp. 126-135.
- Beom, H.R. e Cho, H.S., 1995, "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning", IEEE Trans. on Syst., Man, and Cyber., Vol. 25, n. 3, pp. 464-477.
- Brooks, R.A., 1986, "A robust layered control system for a mobile robot", IEEE Journal of Robotics Automation, 2(1), pp.14-23.
- Carvalho, J.R.H., *et al.*, 1999, "A methodology for aerial imagery-based ground robot navigation", Proc. of XV International Conference on CAD / CAM Robotics & Factories of the Future.
- Fabro, J.A., Lopes, H.S. e Arruda, L.V.R., 2001, "Navegação de robôs autônomos utilizando redes neurais, com planejamento de trajeto por algoritmos genéticos baseados em um mapa fuzzy", V Brazilian Conference on Neural Networks, Rio de Janeiro, RJ, Brazil, pp. 247-252.
- Leonard, J.J. e Durrant-Whyte, H.F., 1992, "Directed sonar sensing for mobile robot navigation", Kluwer Academic Publishers.
- Melo, W.S. e Figueiredo, M., 2001, "Memória de curta duração em sistema autônomo aplicado à navegação de robôs", V Brazilian Conference on Neural Networks, Rio de Janeiro, RJ, Brazil, pp. 493-498.
- Verschure, P.F.M., 1993, "Formal minds and biological brains", IEEE Expert, pp. 66-75.

THE USE OF NEURAL NETWORKS TO GENERATE PATHS FOR MOBILE ROBOTS

Bruno Garrido de Menezes *

Fabrício Roulin Bittencourt °

Luis Enrique Zárate °

Marcelo Becker *

(*) IPUC – PUC Minas, Av. Dom José Gaspar, 500 - Belo Horizonte - MG – CEP 30535-610, Brazil, e-mail: marcelo.becker@pucminas.br

(°) Departamento de Ciências da Computação – PUC Minas, Av. Dom José Gaspar, 500 - Belo Horizonte - MG – CEP 30535-610, Brazil, e-mail: zarate@pucminas.br

***Abstract.** In this work an application of Neural Networks for the generation of trajectories for autonomous robots is presented. The neural network allows to estimate the next location of the robot, through the coordinate (x,y), from previous positioning. The result shows the good performance of the net to estimate the new coordinate, in error conditions in the robot's positioning. It is also discussed the possibility to introduce neural memory in the robot for displacements in static or controlled environment, where the optimal robot trajectory can be previously identified. At the end of the paper simulation results are presented.*

***Keywords:** Autonomous Robots, Generation of Trajectories, Neural Networks.*