



GERAÇÃO DE TRAJETÓRIAS PLANAS PARA MANIPULADORES DO TIPO RR UTILIZANDO REDES NEURAIIS

Carlos André Dias Bezerra

Universidade Federal do Ceará, Departamento de Engenharia Mecânica
Cx. P. 12144 - 60.455-760 – Fortaleza, Ce, Brasil.

Douglas Eduardo Zampieri

André Mendeleck

Universidade Estadual de Campinas, Departamento de Mecânica Computacional
Cx. P. 6122 – 13083-970 – Campinas, SP, Brasil

Resumo. *O mapeamento de trajetórias representa um importante fator na precisão de um manipulador. Quanto menores os desvios de posicionamento e de orientação do manipulador, em relação a uma trajetória pré-estabelecida, maior será a precisão do movimento. Entretanto, alguns fatores interferem na execução de uma trajetória, confrontando a exatidão dos movimentos. Geralmente estes fatores são abordados de modo não sistêmicos, quer seja pela melhoria de modelos matemáticos, quer pela implementação de técnicas de controle mais elaboradas ou pela restrição de tarefas, por exemplo. Contudo, a utilização de redes neurais, atuando como um filtro capaz de auto-ajustar-se com os movimentos dos mecanismos em operação, pode ser uma maneira de simplificar estes problemas. Neste trabalho utiliza-se uma rede neural do tipo perceptron de múltiplas camadas para simular os movimentos de um robô tipo SCARA no plano bidimensional, atuando como um seguidor de trajetórias pré-estabelecidas. A área de trabalho e a posição angular dos segmentos são mapeados por redes neurais que serão responsáveis pela geração dos movimentos do manipulador. Apresentam-se algumas simulações dos movimentos, mostrando o comportamento da rede nas fronteiras da área de trabalho e em configurações onde os modelos geométricos tradicionais, geralmente, apresentam singularidades.*

Palavras-chave: *Geração de trajetórias, Redes neurais, Volume de trabalho.*

1. INTRODUÇÃO

No projeto de um manipulador vários aspectos técnicos são considerados, tais como o tipo da estrutura, acionamentos, juntas, tipo de controle, entre outros. Considerando-se o tipo da estrutura do manipulador (serial, paralela e híbrida), o número e tipo das juntas, e os comprimentos dos segmentos, o manipulador terá uma configuração peculiar ao desenvolver uma tarefa específica. Para realizar tal tarefa, o manipulador terá que mapear trajetórias, dentro da área de seu alcance, com precisão e em seguida executá-las eficientemente. Quanto

menores forem os desvios de posicionamento e de orientação do manipulador, em relação a uma trajetória pré-estabelecida, maior será a precisão do movimento. Entretanto, alguns fatores podem interferir na execução da trajetória, penalizando a exatidão dos movimentos. Por exemplo fenômenos de segunda ordem, que são desprezados e que influenciam de modo acentuado, tais como folgas nas engrenagens, flexibilidade dos elementos de transmissão, atritos nas juntas, entre outros. Geralmente estes problemas são abordados de modo não sistêmico, quer seja pela melhoria de modelos matemáticos, quer pela implementação de técnicas de controle mais elaboradas ou pela restrição de tarefas entre outros.

Considerando-se o tipo de controle, o elemento terminal pode ser controlado para acompanhar uma trajetória do tipo ponto-a-ponto, isto é, o manipulador armazena um conjunto de pontos e em seguida executa-os. O conjunto de todos os pontos alcançáveis pelo elemento terminal é denominado de volume de trabalho.

Neste trabalho será desenvolvido um algoritmo eficiente para a aproximação de trajetórias de um manipulador do tipo SCARA (*Selectively Compliant Arm for Robotic Assembly*). O algoritmo tem a finalidade de aproximar uma trajetória específica, dentro do volume de trabalho, partindo de uma posição inicial até uma posição final definida pelo usuário, utilizando-se redes neurais, que atuarão como um filtro capaz de auto-ajustar-se com os movimentos dos mecanismos em operação.

A metodologia consiste em treinar as redes neurais com os vetores de entrada gerados através das equações do modelo geométrico, gerar o volume de trabalho do manipulador, dividir o volume de trabalho em setores compostos de uma densidade de pontos (Kumar and Waldron, 1980, Sem and Mruthyunjaya, 1994, Fu, Gonzalez and Lee, 1987), e em seguida possibilitar ao manipulador acompanhar todas as possíveis trajetórias, definida por um ponto inicial e outro final, dentro do volume de trabalho. O volume de trabalho será definido no plano bidimensional XY, sendo composto por um conjunto de pontos $p(x,y)$.

Inicialmente será feita uma revisão da modelagem usualmente utilizada em manipuladores, para então detalhar-se o algoritmo utilizado na geração de pontos e no volume de trabalho. Finalmente serão apresentados os gráficos das simulações realizadas.

2. DEFINIÇÕES

Define-se situação como sendo o conjunto posição e orientação de um determinado segmento do manipulador. Nos manipuladores planos geralmente utilizam-se dois tipos de pares cinemáticos ou juntas: juntas prismáticas P e de rotação R, possuindo um grau de liberdade (gdl) cada, ou seja, cada junta será representada por apenas uma variável ou coordenada generalizada.

Neste trabalho foi utilizado um manipulador plano derivado da estrutura espacial RRP do manipulador SCARA (Figura 1), onde a junta prismática foi substituída por um ponto $p(x,y)$ localizado no centro geométrico da junta. Deve ser observado que o movimento de translação apenas varia a coordenada Z dos pontos do volume de trabalho. Não será analisada a orientação do elemento terminal.

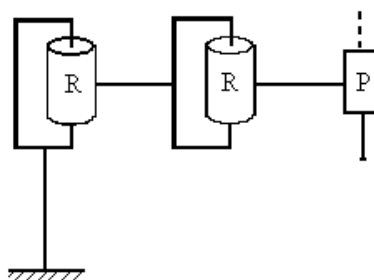


Figura 1- Representação do manipulador do tipo RRP - SCARA

O conjunto de todas as posições que um manipulador pode alcançar no espaço é definido como sendo o volume de trabalho ou espaço de trabalho. Em outras palavras, o volume de trabalho depende não só da geometria do manipulador como também das restrições das suas juntas. Podem existir, dentro do volume de trabalho, configurações singulares na qual o elemento terminal do manipulador não poderá mover-se em certas direções. Estas configurações singulares devem ser evitadas e podem ser detectadas através das expressões matemáticas da velocidade, especificamente, através da matriz Jacobiana. A matriz Jacobiana ou Jacobiano pode ser definida, de uma maneira bem simples, como uma função matricial das equações de velocidade, que relaciona matematicamente o espaço cartesiano das posições e orientações com o espaço das posições das juntas (Spong and Vidyasagar, 1989). O volume de trabalho de manipuladores seriais tem sido estudado usando métodos analíticos e numéricos por muitos autores, entre eles Selfridge (1983), Kumar e Waldron (1981), Mruthyunjaya e Sem (1994) e usando técnicas probabilísticas por Li, Tso e Zhang (1998), e por Rastegar e Fardanesh (1990), onde uma completa revisão bibliográfica por ser encontrada. Segundo Behera, Gopal, e Chaudhury (1996), as redes neurais oferecem uma possibilidade promissora para obter melhores soluções para os problemas de trajetórias de robôs, devido a sua excelente capacidade de aprender qualquer mapa complexo de exemplos de treinamento.

A modelagem geométrica estuda o modelo geométrico direto (MGD) e o modelo geométrico inverso (MGI) das estruturas cinemáticas. No modelo geométrico direto obtêm-se as coordenadas operacionais, que definem a situação do elemento terminal, em função da configuração do manipulador. No modelo geométrico inverso, obtêm-se as coordenadas generalizadas, que definem a configuração do manipulador, em função da posição e orientação do elemento terminal.

Neste trabalho serão utilizadas as equações do modelo geométrico direto para treinar a rede neural. O modelo geométrico inverso é obtido através de redes neurais do tipo perceptron de múltiplas camadas MLP (*Multi Layer Perceptron*) (Haykin, 1999), sem a necessidade de conhecimento das equações matemáticas. A rede neural mapeará a área de trabalho, determinará a posição angular dos segmentos e será a responsável pela geração dos movimentos do manipulador. O método é inerentemente numérico, sendo adequado para aplicações computacionais. O software utilizado para obtenção do simulação numérica e geração da rede neural foi o MATLAB®.

De acordo com a figura 2, observa-se que os eixos cartesianos originais do manipulador SCARA foram modificados e a terceira junta, a junta prismática, foi suprimida visando simplificar as equações matemáticas do modelo geométrico direto. Tal simplificação é possível, pois o movimento de translação ao longo do eixo coordenado Z, não influencia as coordenadas X e Y do ponto $p(x,y)$. Portanto, as equações matemáticas para o modelo

geométrico direto do manipulador RR plano a ser analisado neste trabalho são (Spong and Vidyasagar, 1989):

$$\begin{aligned}x_2 &= a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \\y_2 &= a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2)\end{aligned}\quad (1)$$

O manipulador apresenta configuração singular em $\theta_2 = 0$ ou π .

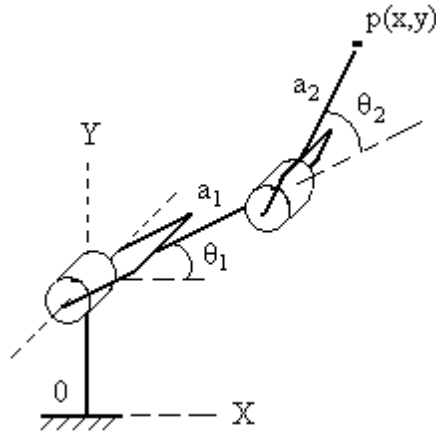


Figura 2- Mecanismo do tipo RR plano

Nos manipuladores industriais são usados motores de corrente contínua, juntamente com sensores de posição de juntas, ou motores do tipo passo para acionar as juntas de rotação. Ambos sistemas de acionamento são de natureza discretos, quer devido a resolução dos sensores ou do valor do passo. Isto quer dizer que na quase totalidade dos casos, a trajetória gerada pelo elemento terminal é discretizada. Neste trabalho, foi adotado um incremento para os ângulos de rotação das juntas R, visando discretizar a trajetória. Isto permite diminuir o número de pontos armazenados na memória do computador, minimizando, assim, o esforço computacional, sem contudo comprometer a precisão dos resultados. Estudos detalhados sobre a discretização de trajetórias contínuas para manipuladores podem ser encontrados em Lianos, Kiritsis e Astragathos (1991).

Para discretizar o volume de trabalho, foi adotado para cada um dos ângulos das juntas de rotação, θ_1 e θ_2 , um incremento específico para cada um deles. Adotar um incremento muito pequeno, embora perfeitamente possível, torna o armazenamento de pontos muito demorado, comprometendo por conseqüência o treinamento da rede neural. Outra questão é a visualização da trajetória realizada dentro do volume de trabalho. Para um número muito grande de pontos é praticamente impossível visualizá-la. Portanto, neste trabalho optou-se por gerar apenas um número limitado de pontos a serem alcançados.

3. ALGORITMO PARA GERAÇÃO DE TRAJETÓRIAS

De acordo com a Figura 3, o algoritmo para a geração de trajetórias do manipulador plano de 2 gdl é iniciado com a introdução dos pontos inicial e final da trajetória. Além destes é necessário fornecer o conjunto de pontos gerados a partir dos incrementos dos ângulos de cada junta de rotação R, que serão utilizados no treinamento das redes neurais. Estas redes, do tipo perceptron, determinarão as coordenadas generalizadas de cada uma das junta de rotação e as

coordenadas operacionais dos extremos de cada segmento do manipulador do tipo RR. Em um segundo passo estas mesmas redes permitirão obter a trajetória.

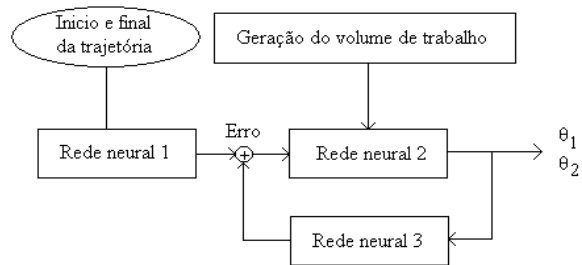


Figura 3 – Representação do algoritmo para geração da trajetória

O vetor de entrada para treinamento da rede 1 é formado pelo ponto inicial (X_i, Y_i) e final da trajetória (X_f, Y_f) , de acordo com a figura 4, onde b é o bias, $\pm 1/\Delta s$ são os pesos sinápticos e (X_1, Y_1) é a saída da rede. Os pontos inicial e final devem estar contidos no volume de trabalho.

A rede 1 é formada por 2 neurônios, um para cada coordenada cartesiana, ambos com função de ativação do tipo linear. Define-se X_2 e Y_2 como os vetores que contém os pontos que formam o volume de trabalho do manipulador, ou seja, as coordenadas cartesianas que representam a posição do elemento terminal. A saída da rede 1 é a entrada da rede 2, e juntamente com a saída da rede 3, formará o erro. Na seção 4 será detalhado o funcionamento das redes.

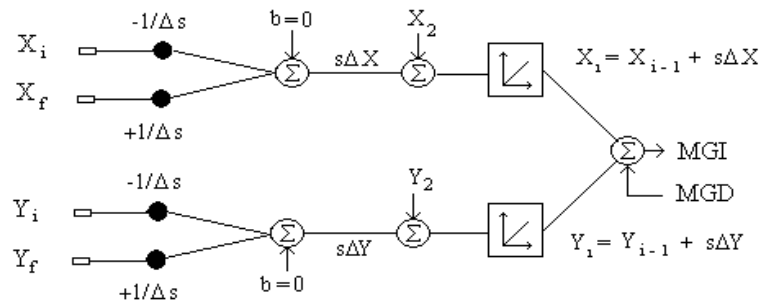


Figura 4 – Rede neural 1

Para o treinamento da rede neural 2, que representa o MGI, foi utilizado o vetor de exemplos $P[X_2, Y_2]$ que contém os pares ordenados, $p(x_2, y_2)$, gerados a partir da Eq. (1), com incrementos de $\Delta\theta_1$ radianos para θ_1 e de $\Delta\theta_2$ radianos para θ_2 . A saída desejada corresponde ao vetor $Ti[\theta_1, \theta_2]$, que é o vetor alvo a ser alcançado no treinamento. A dimensão do vetor X_2 , que é igual a dimensão do vetor Y_2 , é função dos incrementos de $\Delta\theta_1$ e de $\Delta\theta_2$.

Para o treinamento da rede neural 3, que representa o MGD, foi utilizado o vetor $Pd[\theta_1, \theta_2]$, gerado a partir de incrementos discretos, $\Delta\theta_1$ e $\Delta\theta_2$, como vetor de exemplos. A saída desejada corresponde ao vetor alvo $Td[X_2, Y_2]$, gerado a partir da Eq. (1).

Na figura 5 são esquematizadas as redes 2 e 3, do tipo MLP. Ambas são compostas por duas camadas intermediárias, a primeira com 5 neurônios e a segunda com 10 neurônios e pela camada de saída. O treinamento das redes neurais 2 e 3 foi realizado através do método de retroalimentação do erro, ou *Backpropagation* (Haykin, 1999). A função de ativação

adotada nas 2 camadas intermediárias é do tipo tangente sigmoidal, e a função de ativação da camada de saída é do tipo linear. As camadas intermediárias poderiam ter mais neurônios, mas a rede 2-5-10-2 é relativamente rápida para treinar e sua resposta é precisa com acentuada convergência. Deve ser observado que o vetor de entrada tem dimensão $N \times 2$, onde N é o número de linhas, sendo função dos incrementos dos ângulos das juntas.

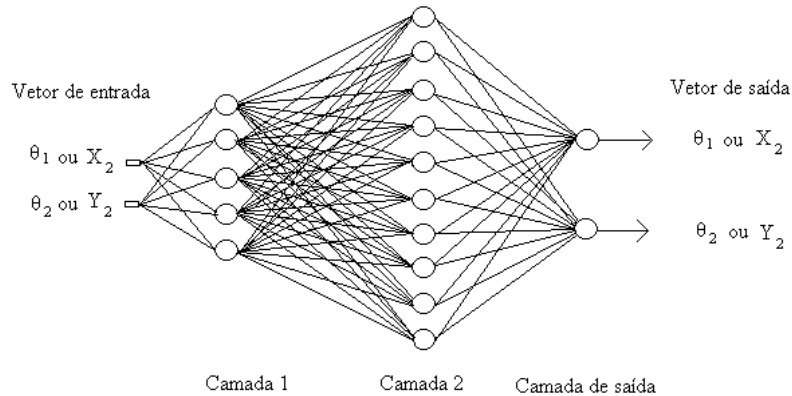


Figura 5 – Rede neural do tipo MLP.

O treinamento das redes foi realizado em código MATLAB[®]. O critério de parada do treinamento foi um erro quadrático médio menor do que 10^{-4} . Optou-se por não incluir neste trabalho os programas em código MATLAB[®], mas somente os gráficos com os resultados.

4. SIMULAÇÃO

O plano bidimensional XY foi dividido em setores, de tal modo que o volume de trabalho (figura 6) fosse distribuído entre eles. Cada setor corresponde a um incremento de $\Delta\theta_1$ e o número de pontos por setor é função do incremento $\Delta\theta_2$. Quanto menor for o incremento na primeira junta de rotação, maior será o número de setores e, quanto menor for o incremento na segunda junta de rotação, maior será o número de pontos $p(x_2, y_2)$ por setor. Ao dividir-se o volume de trabalho em setores com um número inteiro associado a cada um deles, nota-se que alguns destes setores terão uma concentração de pontos maior em relação aos demais, devido a superposição dos setores. Quanto maior a densidade de pontos em um setor, maior será a precisão ao alcançar este ponto, conforme observado por Ebert-Uphoff e Chirikjian (1995).

Para a geração do volume de trabalho, optou-se por analisá-lo somente no primeiro quadrante, por simplicidade de visualização e compreensão do trabalho.

Para realizar a trajetória desejada dentro do volume de trabalho, no caso uma reta com início e extremidade respectivamente em (X_i, Y_i) e (X_f, Y_f) , o algoritmo gera o volume de trabalho e treina as redes 2 e 3 para determinar a qual setor que qualquer um dos pontos pertence, mesmo que ele não tenha sido apresentado à rede. Caso o ponto não pertença ao volume de trabalho, a rede retorna o valor zero para o respectivo setor e o ponto não é armazenado.

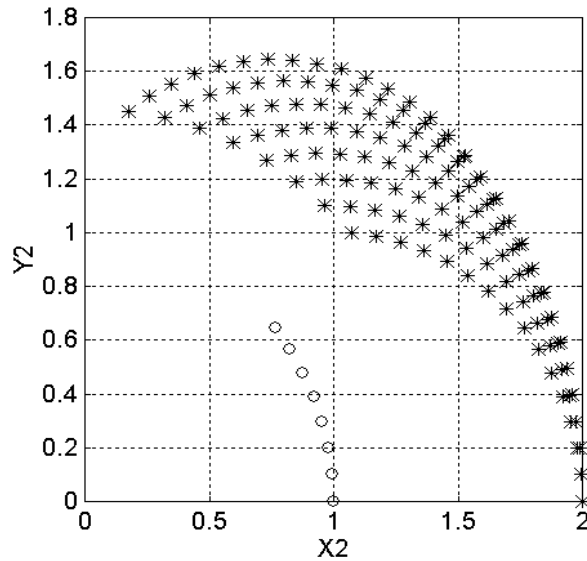


Figura 6 – Volume de trabalho do manipulador RR plano

A rede 1 tem por função determinar o conjunto de pontos (X_1, Y_1) que definem exatamente uma reta com início e final da trajetória em (X_i, Y_i) e (X_f, Y_f) respectivamente. Este conjunto de pontos é formado através de incrementos $s\Delta x$ e $s\Delta y$ somados ao ponto inicial (X_i, Y_i) até atingir o ponto final (X_f, Y_f) . Somente são armazenados os pontos (X_1, Y_1) dentro do volume de trabalho. Em seguida todos os pontos determinados pelas redes 2 e 3, (X_2, Y_2) , são apresentados a rede 1. Os pontos (X_2, Y_2) que apresentarem o menor erro em relação aos pontos (X_1, Y_1) são selecionados como pontos da trajetória do manipulador. Em seguida os pontos (X_2, Y_2) selecionados, são ligados entre si dentro do volume de trabalho, gerando uma aproximação da trajetória exata. O vetor de saída do modelo geométrico inverso $T_i [\theta_{r1}, \theta_{r2}]$ também é obtido através da rede neural e armazenado.

O algoritmo para gerar as trajetórias, descrito acima, pode ser representado através da seguinte seqüência de comandos:

```

Para:  $\theta_1 = 0 : \text{Pi}/4$ 
       $\theta_2 = 0 : \text{Pi}/2$ 
      calcule  $X_2$  e  $Y_2$  usando a Eq. 1
      treine a rede 2 e obtenha os pesos e bias
      treine a rede 3 e obtenha os pesos e bias
      setor=setor+1
Novo  $\theta_1$  e  $\theta_2$ 
Solicite os pontos  $(X_i, Y_i)$  e  $(X_f, Y_f)$ 
  Calcule  $X_1 = X_{i-1} + s\Delta x$  e  $Y_1 = Y_{i-1} + s\Delta y$  no intervalo  $(X_i, Y_i)$  a  $(X_f, Y_f)$ 
  Teste cada ponto de  $(X_1, Y_1)$  a qual setor pertence utilizando as redes 2 e 3
  Teste se  $(X_1, Y_1) - (X_2, Y_2) < \text{erro}$ ,
    Sim : plote  $(X_2, Y_2)$ 
Não: próximo ponto  $(X_{i-1}, Y_{i-1})$ 
Plote o volume de trabalho e ligue os pontos  $(X_2, Y_2)$  próximos a reta
  
```

Figura 7 – Algoritmo para gerar trajetórias

5. RESULTADOS

São apresentados dois casos de trajetória a serem seguidas. Na figura 8a e 8b, a trajetória está contida totalmente no volume de trabalho. Na figura 8c e 8d, a trajetória está contida parcialmente no volume de trabalho. Em ambos os casos a trajetória exata é representada por uma linha reta. O ângulo da primeira junta de rotação foi variado de $0 < \theta_1 < \pi/4$ e o ângulo da Segunda junta foi variado de $0 < \theta_2 < \pi/2$. Foram adotados os seguintes valores para as constantes que representam as dimensões do braço e antebraço, $a_1=1$ e $a_2=1$, de acordo com a figura 2.

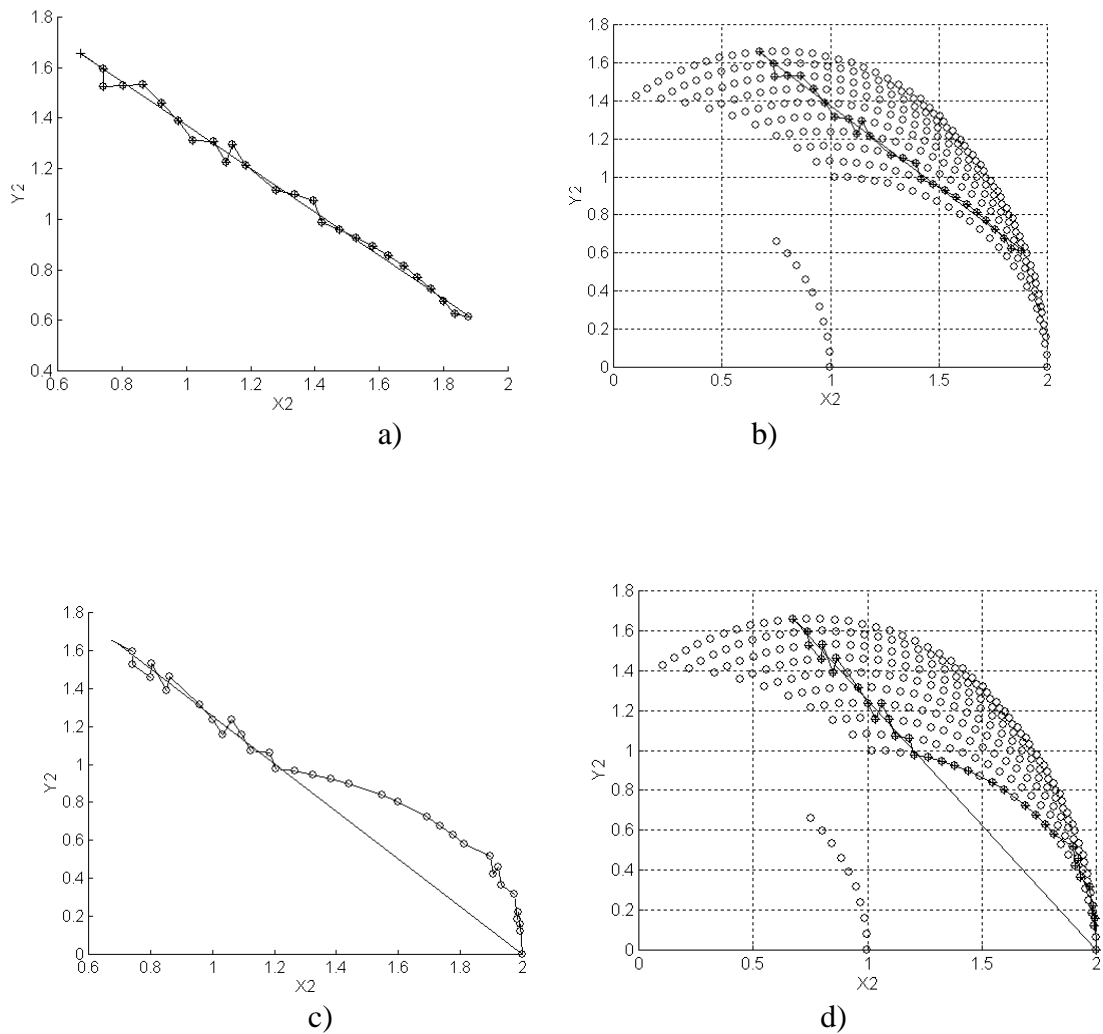


Figura 8 – Trajetórias exata e calculada no volume de trabalho

As redes 2 e 3 também foram usadas para gerarem o volume de trabalho via rede neurais, conforme pode ser visto na figura 9. É necessário esclarecer-se que os dados para o treinamento das redes 2 e 3 poderiam ser obtidos a partir de medições no manipulador real, e portanto, não seria necessário o uso das equações do modelo geométrico direto e inverso para o treinamento das redes.

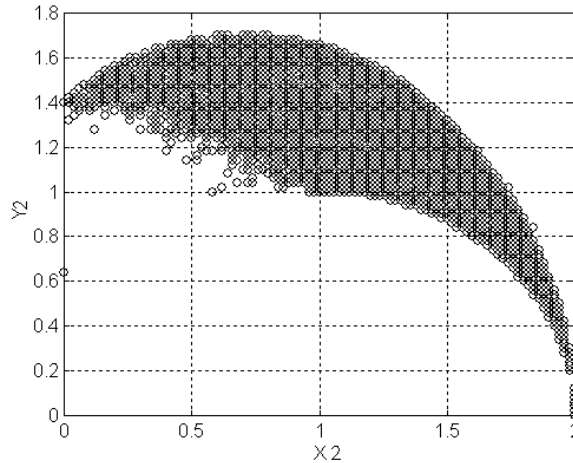


Figura 9 – Volume de trabalho gerado por rede neural

6. CONCLUSÃO

A metodologia apresentada permite determinar a trajetória entre dois pontos quaisquer dentro do volume de trabalho de um manipulador plano do tipo RR, sem o conhecimento das equações cinemáticas do manipulador.

A determinação da trajetória é obtida através do uso de redes neurais do tipo perceptron, que são treinadas com os pontos do volume de trabalho. Eles podem ser obtidos através do modelo geométrico direto do manipulador RR. No caso de um modelo experimental, eles podem ser determinados pelas leituras das variáveis de cada junta de rotação. Após o treinamento das redes, o algoritmo é capaz de traçar retas entre dois pontos quaisquer dentro do volume de trabalho.

Para verificar a metodologia foram simuladas duas trajetórias distintas: uma completamente dentro do volume de trabalho e outra, parcialmente fora do mesmo. Em ambos os casos a rede foi capaz de identificar a trajetória exata e realizar a tarefa pela aproximação ótima.

Assim, é possível utilizar-se a mesma metodologia para a geração de trajetórias mais complexas, tais como parábolas e circunferências.

Outro aspecto a ser destacado é a capacidade da mesma rede gerar também o volume de trabalho.

A utilização do software MATLAB[®] permitiu uma fácil e rápida implementação computacional do algoritmo, sem a necessidade de uma infra-estrutura de grande porte.

Agradecimentos

A Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES pela bolsa de estudos.

REFERÊNCIAS

Behera, L., Gopal, M. and Chaudhury, S., 1996. On Adaptive Trajectory Tracking of a Robot Manipulator Using Inversion of its Neural Emulator. IEEE Transaction on Neural Networks, november, vol 7 no 6, pp 1401-1414.

- Chirikjian, G.S., 1994. A binary paradigm for robotic manipulators. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, May.
- Ebert-Uphoff, I. and Chirikjian, G.S., 1995. Generation of Binary Manipulator Workspaces and Work Envelopes, Proceedings of the IASTED Int. Conf. On Rob. And Manuf., Cancun, Mexico.
- Ebert-Uphoff, I. and Chirikjian, G.S., 1995. Efficient workspace generation for binary manipulators with many actuators. Journal of Robotic Systems, June.
- Fu, K.S., Gonzalez, R.C. and Lee, C.S.G., 1987. Robotics: Control, Sensing, Vision and Intelligence. Mcgraw Hill, Singapore.
- Haykin, S., 1999. Neural Networks a comprehensive foundation, 2ed, Prentice Hall, New Jersey.
- Kumar, A and Waldron, K.J., 1980. Numerical plotting of surfaces of positioning accuracy of manipulators. Mech. Mach. Theory, 16(4):361 -- 368.
- Li Q, Tso, S.K. and Zhang W.J., 1998. Trajectory Tracking Control of Robot Manipulator Using Neural Network Torque Compensator, Proceedings of the Institution of Mechanical Engineers, Journal of Syst. and Ctrl Engineering, part I, vol 212, no 15, 1998.
- Lianos, T., Kiritsis, D. and Aspragathos, N., 1991. A Direct Algorithm for Manipulators Continuous Path Control, Robotics and Computer-Integrated Manufacturing, Vol.8, No 2.
- Rastegar, J. and Deravi, P., 1987. The effect of joint motion constraints of the workspace and number of configurations of manipulators. Mech. Mach. Theory, 22(5):401 -- 409.
- Rastegar, J. and Fardanesh, B., 1990. Manipulator workspace analysis using the Monte Carlo method, Mech. Mach. Theory, 25(2):233 - 239.
- Selfridge, R.G., 1983. The reachable workarea of a manipulator. Mech. Mach. Theory, 18(2):131 -- 137.
- Sen, D. and Mruthyunjaya, T.S., 1994. A discrete state perspective of manipulator workspaces. Mech. Mach. Theory, 29(4):591-605.
- Spong, M.W., and Vidyasagar, M., 1989. Robot dynamics and control, John Wiley & Sons. New York.

GENERATION OF PLANE TRAJECTORY FOR MANIPULATOR RR TYPE USING NEURAL NETWORKS

Abstract. *The trajectory planning is an important factor in a manipulator precision. So smaller deviation of position and orientation of manipulator, related with a known path, than bigger will be the movement precision. However, some factors have an influence in the execution of the path. Generally these factors are treated either by non-systemic modes, or by the mathematical model improvement or by the implementation of elaborated control techniques, or by task constraints. Thus, the use of neural networks, acting as a filter that is able to self tuning with the operating mechanisms movements, can be a way to simplify these problems. In this paper a neural networks type multi lawyer perceptron has been used to simulate the robot SCARA type movements in a bidimensional plane, acting as follower of known paths. The workspace and the angular position of the links are mapped by neural networks that will responsible for the manipulator movement generation. It presents some simulations of movements that shows the net performance in the boundary of workspace area and in some configurations that the traditional geometric models usually presents singularity.*

Key-words: *Trajectory Generation, Neural Networks, Workspace.*