

# Mapeamento da cinemática inversa de manipuladores robóticos usando RNAs configuradas em paralelo aplicado a um manipulador de 5 GDL controlado pela placa Intel® Galileo Gen 2

Ricardo F. Nunes<sup>1</sup>[0000-0002-4565-6658] e Suely C. A. Mantovani<sup>2</sup>[0000-0002-6735-9427]

<sup>1</sup>Departamento de Engenharia Elétrica, Instituto Federal de Educação Ciência e Tecnologia de São Paulo- Presidente Epitácio, SP

<sup>2</sup>Departamento de Engenharia Elétrica, Universidade Estadual Paulista 'Julio de Mesquita Filho', Caixa Postal 15.385.000, Ilha Solteira, SP  
rnfunes@ifsp.edu.br, suely@dee.feis.unesp.br

**Abstract.** Neste trabalho apresenta-se uma abordagem para o mapeamento da cinemática inversa utilizando Redes Neurais Artificiais do tipo Perceptron Multicamadas configuradas em paralelo, tendo como caso de estudo o protótipo de um manipulador robótico de 5 graus de liberdade, composto por servomotores controlados pela plataforma de desenvolvimento Intel® Galileo Gen 2. As equações da cinemática inversa, normalmente apresentam múltiplas soluções, e para contornar esta particularidade propõe-se o uso de Redes Neurais Artificiais em razão da sua flexibilidade e capacidade de aprendizado que, por meio do treinamento, armazena o conhecimento da relação cinemática entre o sistema de coordenadas das juntas e a posição final da ferramenta do manipulador. A avaliação da eficiência do método proposto se deu por simulações, as quais demonstram pelos resultados obtidos, melhor desempenho quando comparadas a uma Rede Neural Artificial simples do tipo Perceptron Multicamadas.

**Keywords:** Manipuladores Robóticos, Cinemática Direta, Cinemática Inversa, Redes Neurais Artificiais.

## 1 Introdução

Os robôs manipuladores são considerados parte indispensável nas modernas fábricas, devido à capacidade de executar os mais variados tipos de tarefas de alto nível de complexidade e periculosidade, de forma eficiente e confiável, com relação custo/benefício incontestável. Visando o estudo e as diferentes aplicações práticas, nos projetos de robôs manipuladores duas linhas de pesquisas podem ser abordadas, a cinemática e a dinâmica. A cinemática trata do movimento sem considerar as forças que causam este movimento, o estudo das forças faz parte da dinâmica.

Na literatura a cinemática de robôs manipuladores é abordada por meio de dois modelos, o da cinemática direta e a inversa. A cinemática direta consiste em achar a orientação e a posição do efetuador terminal a partir do vetor de ângulos das articulações

e dos parâmetros geométricos do modelo. Denavit-Hartenberg [1] propuseram um método matricial para modelagem e solução sistemáticas de problemas que utilizam a técnica da cinemática direta.

A cinemática inversa é considerada uma técnica efetiva de controle de um braço robótico, consistindo em achar o vetor de ângulos das articulações, a partir da orientação e posição do efetuador terminal. Apresenta grandes desafios devidos a não linearidade das equações e múltiplas soluções para manipuladores com muitos graus de liberdade [2], [3].

Métodos tradicionais podem ser usados para calcular a cinemática inversa de manipuladores robóticos, tais como, os geométricos, numérico-iterativos e os algébricos, considerados trabalhosos se a estrutura do manipulador é muito complexa [4].

As redes neurais artificiais (RNAs) são uma das técnicas frequentemente propostas na literatura, para resolver o problema resultante da modelagem por meio da cinemática inversa [5],[6]. As RNAs são capazes de estabelecer as relações necessárias para controlar o manipulador robótico, por meio de treinamentos que permitem a sua flexibilidade e capacidade de aprendizagem. Alguns trabalhos combinam técnicas de RNAs, com outras técnicas, tais como, Algoritmo Genético, Lógica Fuzzy ou alguma outra heurística de otimização para acelerar a convergência da técnica que utiliza a cinemática inversa, de forma a proporcionar o aumento do desempenho do sistema [7],[8],[9].

Neste trabalho propõe-se a aplicação de RNAs do tipo Multilayer Perceptron (MLP) configuradas em paralelo, para o mapeamento da cinemática inversa de robôs com mais alto grau de liberdade. A técnica é aplicada considerando-se trajetórias e pontos distribuídos em todo o volume de trabalho, adotando-se como caso de estudo, um protótipo de um manipulador robótico didático com 5 graus de liberdade (GDL) construído em laboratório, usando a placa de desenvolvimento Intel® Galileo Gen 2. As RNAs configuradas em paralelo são comparadas a uma RNA.

Neste artigo organiza-se o texto de forma que na seção 2, apresenta-se a cinemática dos manipuladores robóticos. Na seção 3 descreve-se a estratégia das Redes Neurais Artificiais e os padrões de treinamento utilizados. Os resultados são apresentados na seção 4, seguidos das conclusões.

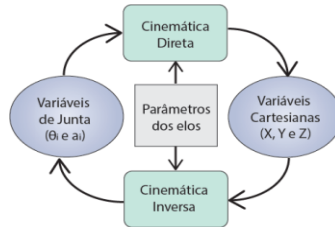
## 2 Cinemática de Manipuladores Robóticos

Um manipulador robótico consiste de uma sequência de elos interconectados por meio de juntas (articulações), composto por um pulso e um efetuador, ao qual podem ser acopladas ferramentas para o robô realizar tarefas preestabelecidas.

Cada junta do manipulador contribui, normalmente, com 1 GDL. O movimento em cada grau de liberdade se dá por atuadores ativados por um sinal de um controlador que é integrado com o modelo cinemático do manipulador. Os modelos cinemáticos representam matematicamente o movimento espacial realizado pelo robô[10].

A resolução de problemas que tratam da cinemática direta de manipuladores é considerada simples, se comparada ao da forma inversa [3]. Na cinemática inversa os ângulos entre as juntas podem ser determinados pelos parâmetros dos elos e a posição do

efetuador no volume de trabalho. Na Figura 1 apresenta-se um diagrama que ilustra as relações entre as cinemáticas direta e inversa.

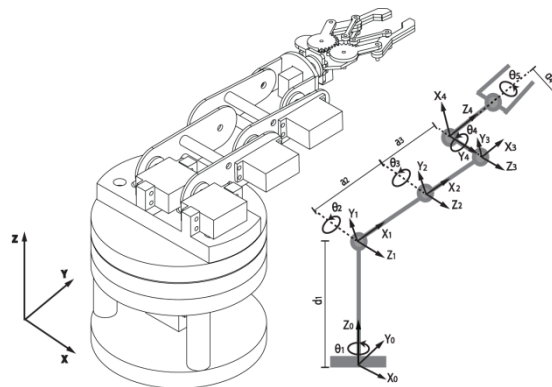


**Fig. 1.** Representação das relações entre as cinemáticas do robô.

Desta forma, na cinemática inversa, considerando um ponto  $P(P_x, P_y, P_z)$  e um manipulador com  $n$  GDL, deve-se determinar os ângulos das juntas, dados por  $\theta_i = f(P_x, P_y, P_z)$ , para  $i = 1, \dots, n$ . Neste trabalho propõe-se a solução do problema da cinemática inversa de um manipulador robótico utilizando RNAs e tendo como base de conhecimento as relações oriundas da cinemática direta por meio da sistemática proposta por Denavit-Hartenberg (D-H), independente do número de articulações e dos graus de liberdade.

## 2.1 Modelo cinemático direto

A sistemática de D-H para a solução da cinemática direta é recomendada para manipuladores com mais de 3 GDL, e é utilizada como um padrão de representação de manipuladores robóticos. Para a aplicação desta técnica deve-se determinar o sistema de referências para a estrutura de elos e juntas de um manipulador de 5GDL, conforme Figura 2, obtendo-se o modelo cinemático direto na representação de D-H, com parâmetros físicos e os limites das juntas da Tabela 1.



**Fig. 2.** Sistemas de coordenadas do manipulador construído.

**Table 1.** Parâmetros de D-H do protótipo e limites das juntas

Junta	$\theta_i$ [°]	$d_i$ [mm]	$\alpha_i$ [°]	$a_i$ [mm]	$\theta_{\min}$ [°]	$\theta_{\max}$ [°]
1	$\theta_1$	160	90	0	0	180
2	$\theta_2$	0	0	80	0	90
3	$\theta_3$	0	0	70	0	45
4	$\theta_4$	-41,75	90	0	-90	0
5	0	156	0	0	0	0

Analisando-se o sistema de referência, obtêm-se para cada junta  $i$ , quatro parâmetros,  $\theta_i$  - ângulo da junta;  $d_i$ - distância da junta;  $a_i$ - distância do elo; e,  $\alpha_i$ - ângulo de torção do elo. Estes parâmetros são utilizados nas matrizes de transformações homogêneas que relacionam o sistema de referência da junta atual com o sistema de referência da junta anterior, representada pela Equação (1).

$$A_{i-1,i} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \cdot \sin\theta_i & \sin\alpha_i \cdot \sin\theta_i & a_i \cdot \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cdot \cos\theta_i & -\sin\alpha_i \cdot \cos\theta_i & a_i \cdot \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

A matriz de transformação total do robô se dá pela associação de cada referencial da junta anterior ( $i - 1$ ), com a junta atual,  $i$ , ou seja, da relação da base até a primeira junta, da primeira à segunda, e assim por diante, até o referencial do sistema de coordenadas do terminal efetuator [1].

### 3 Estratégia adotada para as RNAs

As RNAs são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Têm como vantagem a habilidade de representarem relações lineares e não lineares, aprenderem a partir destas relações e serem modeladas pelos dados que chegam até elas.

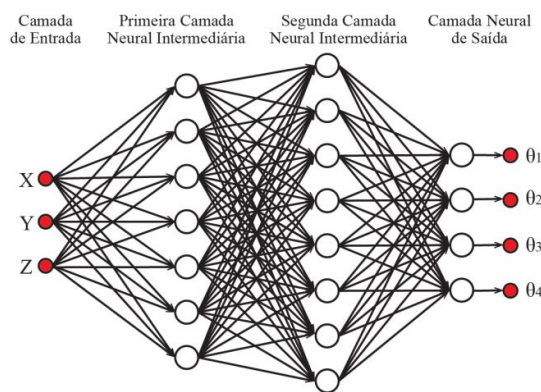
Nas RNAs os neurônios artificiais são organizados de forma paralela. Assemelham-se ao cérebro humano, cujo conhecimento adquirido por meio de um processo de aprendizagem é armazenado nas interconexões internas conhecidas como pesos sinápticos. No processo de treinamento a RNA é apresentada a diversos padrões de entrada e seus respectivos padrões de saída desejados, de forma a ajustar os parâmetros obtendo-se a mínima diferença entre a saída da RNA e a saída desejada. Após esta fase, espera-se que, caso seja apresentado um padrão de entrada não pertencente ao conjunto de treinamento, a RNA seja capaz de inferir um provável padrão de saída, demonstrando habilidade de generalização das informações.

A existência de propriedades como, a habilidade de aprendizado e generalização de padrões, estimulam aplicações de RNAs em reconhecimento de padrões, controle de processos, na robótica, entre outros [11].

A RNA utilizada neste trabalho é a Perceptron Multicamadas (MLP), uma das mais comuns, que utiliza treinamento supervisionado. A topologia da RNA escolhida é apresentada na Figura 3, composta por 3 neurônios na camada de entrada, referentes às

variáveis do sistema de coordenadas retangulares  $x$ ,  $y$  e  $z$ ; 7 e 8 neurônios na primeira e segunda camadas intermediárias, respectivamente, e 4 neurônios na camada de saída referentes aos ângulos das juntas.

A MLP aprende usando um algoritmo de Retropropagação do Erro ou *Error Back-propagation* (BP), em que a entrada de dados é repetidamente apresentada à rede neural, e em cada apresentação a saída da RNA é comparada à saída desejada, e um erro é computado. Este erro é realimentado e é utilizado para ajustar os pesos sinápticos, tal que o erro decresça a cada iteração e o modelo da rede convirja de forma a produzir a saída desejada, chamado processo de treinamento.



**Fig. 3.** Topologia da RNA proposta.

O algoritmo de treinamento é, portanto, constituído de duas etapas: a propagação e a retropropagação. Na etapa de propagação, através de um vetor  $j$  de entrada, obtêm-se os respectivos valores de saída,  $o_j$ , que são comparados aos valores desejados,  $d_j$ , para gerar o erro,  $e_j$ . Este erro é retropropagado para ajustar os valores dos pesos sinápticos, num processo de otimização baseado no gradiente descendente.

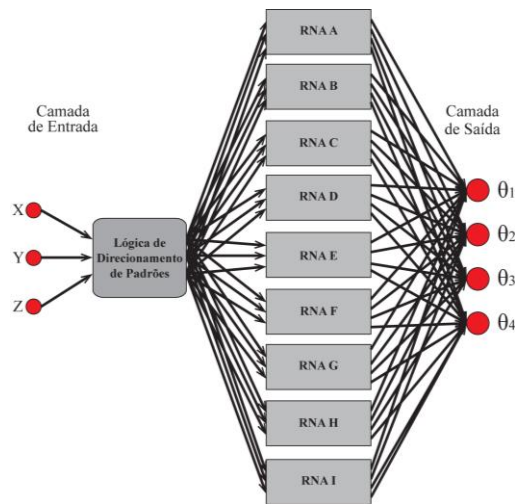
Neste trabalho foram propostas e implementadas RNAs configuradas em paralelo para resolver a cinemática inversa de robôs manipuladores, todas com a mesma topologia da RNA anteriormente mencionada (Figura 3), modelo adaptado de [7].

### 3.1 RNAs configuradas em paralelo

Na solução do problema da cinemática inversa, 9 RNAs são configuradas em paralelo, nomeadas de “A” a “I”, ativadas de acordo com o valor da coordenada  $z$  do plano cartesiano representado na Tabela 2. Uma RNA específica é ativada em relação à coordenada  $z$  da entrada pela lógica de direcionamento de padrões (algoritmo) de treinamento, Figura 4, proporcionando a obtenção de RNAs especializadas em uma determinada faixa do volume de trabalho do manipulador, ao invés de uma única rede.

**Table 2.-** Faixa de operação para cada RNA

Rede Neural Artificial	Faixa de operação(mm)
RNA A	$Z \leq 100$
RNA B	$100 < Z \leq 150$
RNA C	$150 < Z \leq 200$
RNA D	$200 < Z \leq 250$
RNA E	$250 < Z \leq 300$
RNA F	$300 < Z \leq 350$
RNA G	$350 < Z \leq 390$
RNA H	$390 < Z \leq 430$
RNA I	$Z > 430$

**Fig. 4.** Arquitetura das RNAs configuradas em paralelo

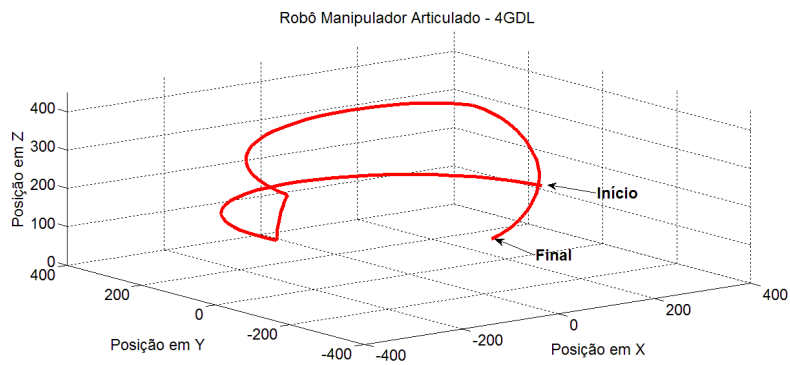
### 3.2 Padrões de treinamento gerados

Os padrões de treinamento são gerados pela cinemática direta via sistemática de D-H, com os parâmetros obtidos por meio do sistema de coordenadas do manipulador construído. O volume de trabalho é limitado aos valores de ângulos máximos e mínimos de cada junta (Tabela 1), restrições devidas aos limites físicos do protótipo e para reduzir as suas redundâncias, embora as equações da cinemática direta sejam válidas para qualquer valor de ângulo.

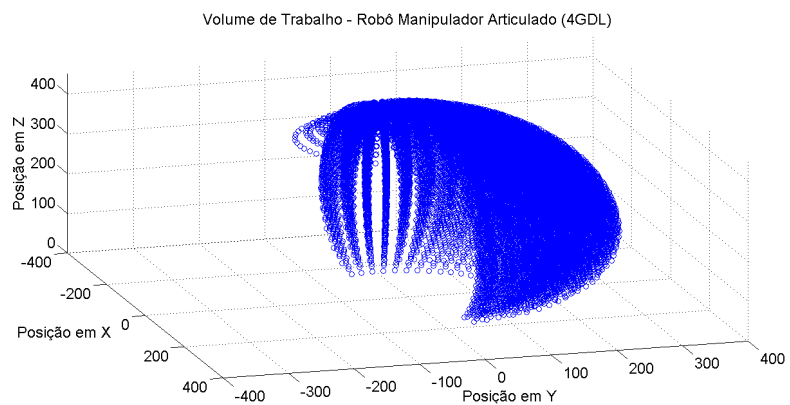
O protótipo sob estudo possui 5 GDL, mas consideram-se somente 4 para a geração dos padrões de treinamento, pois ao eliminar um grau de liberdade, o número de padrões reduz exponencialmente quando se deseja mapear todo o espaço de trabalho do protótipo, para o programa usando a cinemática direta.

Os padrões de treinamento para a aprendizagem das RNAs são obtidos a partir de duas abordagens distintas. A primeira por meio da geração de uma trajetória constituída

de 120 pontos, conforme a Figura 5, percorrendo 3 quadrantes do plano cartesiano. A outra abordagem utiliza um conjunto constituído de 22.797 pontos distribuídos em todo o volume de trabalho delimitado, Figura 6.



**Fig. 5.** Trajetória gerada por 120 pontos

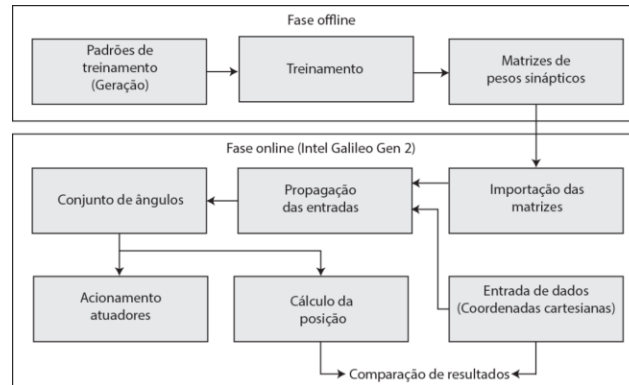


**Fig. 6.** Pontos distribuídos no espaço de trabalho

## 4 Resultados

### 4.1 Fase de treinamento e simulações (*offline*)

O projeto foi desenvolvido em duas fases, uma relacionada à geração dos padrões de treinamento e o treinamento que ocorre de forma *offline* visando obter o banco de conhecimentos das matrizes de pesos sinápticos. Na outra fase, a de validação online aplicada ao protótipo, o banco de conhecimentos das matrizes de pesos sinápticos, gerados na fase *offline* são importados para a plataforma de desenvolvimento microprocessada utilizada. Na Figura 7 tem-se o diagrama de blocos para as duas fases.



**Fig. 7.** - Diagrama de blocos funcional das fases *offline* e *online* do projeto.

Essa estratégia de desenvolvimento é adotada devido a exigência de recursos e tempos computacionais elevados para o treinamento das RNAs, inviáveis para o processamento *online* em plataformas de desenvolvimento com recursos limitados de armazenamento e processamento. Nas etapas da fase *offline* utiliza-se um notebook, processador Intel Core i7-4510U, 2,6GHz, RAM de 8GB, e o software MATLAB.

O conjunto de RNAs configuradas em paralelo é submetido ao aprendizado pelos dois padrões de treinamento gerados. Os resultados destas simulações são comparados com os resultados obtidos das simulações para os mesmos padrões, usando somente uma RNA.

## 4.2 Simulações para as trajetórias

Para simulações da trajetória supracitada, ambos os processos de treinamento, RNA simples e as RNAs em paralelo foram interrompidas à época de número 20.000. O algoritmo de treinamento, para a RNA simples, obteve o melhor desempenho com um erro quadrático médio de, aproximadamente,  $9 \cdot 10^{-5}$ . Para o algoritmo composto por RNAs em paralelo, o erro quadrático médio variou entre  $4 \cdot 10^{-7}$  na RNA de melhor desempenho, e  $2 \cdot 10^{-4}$  na de pior desempenho.

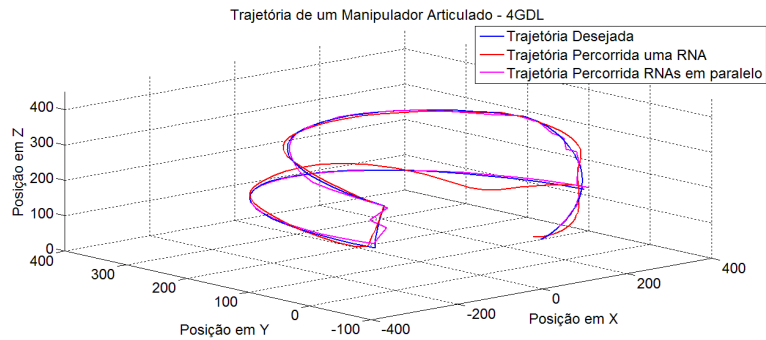
O desempenho dos algoritmos é comparado na Tabela 3, pelas médias aritméticas dos erros para cada junta (em graus), após a validação dos resultados, ou seja, a soma dos erros de cada padrão dividida pelo número de padrões apresentados. No caso do algoritmo das RNAs configuradas em paralelo soma-se e divide-se a média aritmética de cada RNA, pelo número de RNAs utilizadas, alcançando a média das médias aritméticas.

**Table 3.** Desempenho dos algoritmos RNA simples e RNA em paralelo para uma Trajetória

	J <sub>1</sub>	J <sub>2</sub>	J <sub>3</sub>	J <sub>4</sub>
Média aritmética dos erros (uma RNA)	1,92°	1,43°	1,27°	1,64°
Média das médias aritméticas dos erros (RNAs em paralelo)	0,82°	0,61°	0,52°	0,20°



Graficamente é possível comparar o desempenho dos dois algoritmos pela análise da Figura 8.



**Fig. 8.** Desempenho dos algoritmos RNA simples e RNAs em paralelo, para uma trajetória.

### 4.3 Simulação para pontos distribuídos em todo o volume de trabalho

O treinamento de uma RNA simples considerando o conjunto de pontos distribuídos em todo o volume de trabalho foi interrompido na época de número 10.000, apresentando um melhor desempenho na época de número 94 com um erro quadrático médio de  $25 \cdot 10^{-4}$ . Observa-se que a RNA atingiu o melhor desempenho prematuramente e, posteriormente, demonstrou um comportamento de saturação de conhecimento.

Para RNAs configuradas em paralelo o treinamento também foi interrompido na época de número 10.000, onde a RNA que obteve o melhor resultado, RNA A, apresentou um erro quadrático médio de  $5 \cdot 10^{-4}$ , na época de número 9762 e a RNA que obteve o pior desempenho, RNA H, apresentou um erro de  $49 \cdot 10^{-4}$  na época de número 414. Na Tabela 4 têm-se os valores das médias aritméticas dos erros para cada junta (em graus).

**Table 4.-** Desempenho dos algoritmos RNA simples e em paralelo, para pontos distribuídos no volume de trabalho.

	$J_1$	$J_2$	$J_3$	$J_4$
Média aritmética dos erros (uma RNA)	7,84°	8,74°	12,17°	10,33°
Média das médias aritméticas dos erros (RNAs em paralelo)	1,55°	7,86°	10,74°	5,78°

Analisando-se a média aritmética dos erros das juntas,  $J_i, i = 1, \dots, 4$ , para as duas simulações pode-se concluir que o algoritmo que utiliza a configuração de RNAs em paralelo obteve melhor desempenho no aprendizado da relação cinemática do manipulador em estudo.

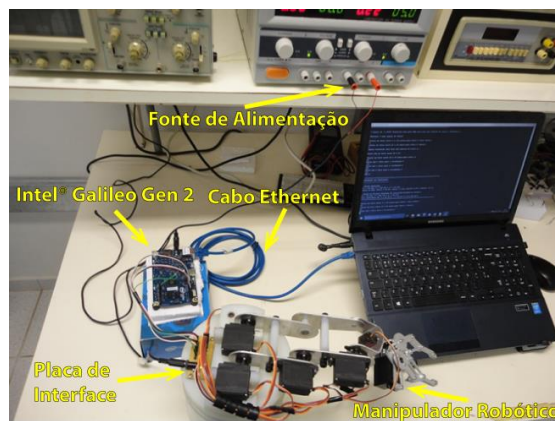
#### 4.4 O protótipo e a Fase de operação (*online*)

O manipulador robótico em estudo, da Fig. 9, é do tipo articulado (RRR) de 5 GDL, sendo 3 GDL necessários para o movimento do “braço” e dois para o “pulso”. Os movimentos do protótipo são proporcionados pelo acionamento de sete servomotores, sendo dois para o movimento do braço (MG996R - TowerPRO), três entre base e antebraço (MG995 - Futaba), um para o punho (MG90S - TowerPRO) e um para a garra (S3003-TowerPRO). Foi confeccionado em alumínio e nylon nos laboratórios do DEE-FEIS-UNESP e do IFSP, Campus Presidente Epitácio-SP-Brasil.



**Fig. 9.** Protótipo do manipulador robótico.

A plataforma Intel® Galileo Gen 2 [12] considerada uma Single Board Controller (SBCt) é utilizada para a realização da etapa de operação (*online*) do algoritmo desenvolvido, pois possui recursos de hardware/software suficientes para esta aplicação, tais como, número de saídas PWM, capacidade de processamento e de memória (Processador Intel® Quark™ SoC X1000, 16K cache e 400 MHz, RAM de 256 MB, DDR3) e a possibilidade de interfaceamento e programação em linguagens de alto nível. Um programa em linguagem Python é implementado na SBCt, para atender ao usuário. Apresentam-se na Figura 10 os componentes usados na montagem para os testes no protótipo.



**Fig. 10.** Montagem do protótipo com o Intel Galileo Gen 2

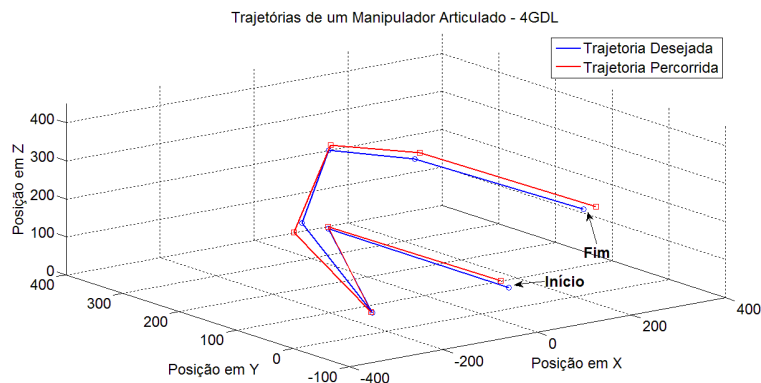
Nesta figura, a plataforma de controle Intel® Galileo Gen 2 é ligada a um notebook via um cabo ethernet para o acesso, usando o protocolo de rede SSH (Secure Shell) e por intermédio do software PuTTY.

O notebook é utilizado para a visualização, entrada de dados e comandos por teclado para o sistema, pois, na fase online, o manipulador robótico é acionado com base nos parâmetros desejados pelo usuário. Uma interface entre o manipulador e a plataforma de controle interliga os terminais para acionamento e fornece a alimentação para os servomotores dada por uma fonte externa de 5 Vcc.

Os terminais dos servomotores conectados via entradas GPIO (General Purpose Input/Output) da plataforma de controle são acionados por sinais PWM (*Pulse Width Modulation*) de 50Hz, gerados por um programa residente na placa Galileo Gen 2.

Os testes são iniciados pela importação das matrizes dos valores dos pesos sinápticos que representam o conhecimento das nove RNAs em paralelo, para o cartão de memória microSD (*Secure Digital Card*) do Galileo Gen 2.

Para realizar um teste de posicionamento via cinemática inversa simulando uma trajetória no protótipo, executa-se o algoritmo da RNA MLP configurada em paralelo. São escolhidos 7 pontos (x, y, z), como pontos desejados dentro do volume de trabalho treinado, visando traçar a trajetória e verificar a precisão obtida pelo algoritmo proposto. A resposta do algoritmo gerou uma segunda trajetória por onde o manipulador supostamente posicionou a garra. Com esses dados, os desejados e os percorridos, foram plotadas as trajetórias ilustradas na Figura 11. Demonstra-se a partir daí que a RNA em paralelo foi treinada de forma satisfatória, pois apresenta uma resposta próxima da trajetória desejada, podendo ser usada para resolver a cinemática inversa de um manipulador robótico.



**Fig. 11.-** Resposta do algoritmo na fase *online*.

## 5 Conclusão

Neste trabalho apresenta-se uma alternativa para resolver o problema da cinemática inversa utilizando RNAs configuradas em paralelo, aplicadas a um protótipo de um manipulador robótico de 5GDL controlado pela plataforma Intel® Galileo Gen 2. O

conjunto de nove RNAs configuradas em paralelo foi comparado a uma única RNA, ambas compostas por RNAs MLP de mesmas características.

Cada RNA configurada em paralelo atua em uma região do volume de trabalho do protótipo, em relação à coordenada  $z$  do plano cartesiano. Para cada padrão de treinamento apresentado, ativa-se apenas uma RNA por vez, ajustando um número relativamente baixo de sinapses, todavia, todo o conhecimento, é armazenado de forma organizada em todas as RNAs, o que possibilita em maior utilização de sinapses, sem perder a eficiência no treinamento.

Analisando os resultados da comparação da RNA em paralelo com uma simples, conclui-se que a técnica proposta apresenta melhor desempenho no mapeamento da cinemática inversa, embora os parâmetros das RNAs e a sua convergência, possam ser melhorados usando algoritmos híbridos - algoritmo genético ou lógica fuzzy.

## Referências Bibliográficas

- [1] S. Niku, *Introduction to Robotics*. New Jersey: John Wiley & Sons, 2010.
- [2] A. T. Hasan, a. M. S. Hamouda, N. Ismail, and H. M. a a Al-Assadi, “An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator,” *Adv. Eng. Softw.*, vol. 37, no. 7, pp. 432–438, 2006.
- [3] J. J. Craig, *Robótica*, 3rd ed. São Paulo: Editora Pearson, 2012.
- [4] S. Alavandar and M. J. Nigam, “Neuro-Fuzzy based Approach for Inverse Kinematics Solution of Industrial Robot Manipulators,” *Int. J. Comput. Commun. Control*, vol. III, no. 3, pp. 224–234, 2008.
- [5] V. M. H. Dávila and J. S. B. Read, “Cinemática inversa de un manipulador robótico con redes neuronales,” *Encuentro Investig. en Ing. Eléctrica*, p. 5, 2004.
- [6] D. R. Raj, I. J. Raglend, and M. Anand, “Inverse kinematics solution of a five joint robot using feed forward and Radial Basis Function Neural Network,” *Comput. Power, Energy Inf. Commuincation*, pp. 117–122, 2015.
- [7] B. Daya, S. Khawandi, and M. Akoum, “Applying Neural Network Architecture for Inverse Kinematics Problem in Robotics,” *J. Softw. Eng. Appl.*, vol. 3, no. 3, pp. 230–239, 2010.
- [8] Y. Feng, W. Yao-nan, and Y. Yi-min, “Inverse Kinematics Solution for Robot Manipulator based on Neural Network under Joint Subspace,” *Int. J. Comput. Commun. Control*, vol. 7, no. 3, pp. 459–472, 2012.
- [9] P. Jha, B. B. Biswal, and O. P. S. Sahu, “Inverse Kinematic Solution of Robot Manipulator Using Hybrid Neural Network,” *Int. J. Mater. Sci. Eng.*, 2014.
- [10] F. E. Fernandes Junior, “Estudo e implementação de redes neurais e algoritmos genéticos para resolução de cinemática inversa de um manipulador robótico com 5 graus de liberdade,” Universidade Estadual de Campinas, Campinas, 2014.
- [11] S. S. HAYKIN, *Redes Neurais. Princípios e prática*, 2nd ed. Porto Alegre: BOOKMAN COMPANHIA ED, 2001.