

IDENTIFICAÇÃO DA DINÂMICA DO OMNIDRIVE DE UM ROBÔ ATRAVÉS DA UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

Stéfano C. Santos*, Francisco L. Oliveira*, José M. A. Júnior*, José M. P. M. Júnior*

*Universidade Federal do Piauí
Departamento de Engenharia Elétrica
Teresina, PI, Brasil

Emails: stefano_sc_@hotmail.com, fco.leao@hotmail.com, jmjuni@ufpi.edu.br,
josemenezesjr@gmail.com

Abstract - In industrial robotics is necessary to have the monitoring and control of speed and distance, within pre-established limits for an application in which the robot is being used. These limits are important for applications requiring a high level of accuracy, so as to have an increased industrial productivity. This work expose the results of the identification of omnidrive dynamics of a mobile robot using Artificial Neural Networks (ANN). For this study were collected speed values of the servos, and then it was estimated the speed in the X, Y and omega directions, resulting in a satisfactory relationship between the speed of the servos and speed of the robot directions.

Keywords – omnidrive, robotics, artificial neural networks, systems identification, multilayer perceptron

Resumo – Na robótica industrial é necessário que se tenha o acompanhamento e o controle da velocidade e distância a ser percorrida pelo robô, dentro de limites pré-estabelecidos para a aplicação na qual o mesmo esteja sendo utilizado. Tais limites são importantes para aplicações que exigem um nível de exatidão elevada, de modo a se ter um aumento na produtividade industrial. Neste trabalho são expostos os resultados da identificação da dinâmica do omnidrive (controlador responsável por transmitir e interpretar dados entre o sistema operacional e as rodas omnidirecionais) de um robô móvel, utilizando Redes Neurais Artificiais (RNA). Para esse estudo, foram coletados valores de velocidades dos servomotores, para em seguida estimar as velocidades do robô nas direções X, Y e ômega, resultando numa relação satisfatória entre as velocidades dos servomotores e velocidade do robô nas direções citadas.

Palavras-chave – omnidrive, robótica, redes neurais artificiais, identificação de sistemas, perceptron de múltiplas camadas

1 Introdução

O estudo da robótica móvel é um tema bastante relevante e atual, onde esta área de estudos, pesquisas e desenvolvimento apresentou um grande salto em seu desenvolvimento nas últimas duas décadas. A aplicação prática de robôs móveis junto a diferentes atividades na sociedade vem demonstrando o quão promissor é o futuro desta área. Dentro deste contexto, constata-se a grande importância do desenvolvimento desta área de pesquisas (Wolf *et al*, 2009).

Experimentos em robótica móvel podem ser realizados de duas formas: diretos em um robô real ou em um robô simulado em ambiente virtual realista. A simulação de sistemas robóticos é essencial para agilizar o ciclo de desenvolvimento de sistemas de controle robóticos, eliminando desperdício de recursos, tanto financeiros como computacionais. Para que uma simulação seja útil, entretanto, ela deve capturar características importantes do mundo físico, onde o termo “importantes” tem relação ao problema em questão. No caso deste trabalho, é fundamental que existam restrições físicas no modelo e que

exista a possibilidade de trabalho em terrenos irregulares. (Pessin *et al*, 2007).

Nos estudos da robótica móvel, a criação de uma navegação autônoma é primordial. Para a operação em ambientes que se modificam dinamicamente, um robô móvel autônomo (RMA) deve ser capaz de adquirir e utilizar conhecimento sobre o ambiente. Sensores e atuadores normalmente estão sujeitos a erros e interferências. Dessa forma o sistema de controle de um robô deve sempre levar em conta a imprecisão dos equipamentos envolvidos. Um sistema robusto deve permitir que, mesmo com sensores e atuadores imprecisos, o RMA cumpra o seu objetivo (Pessin, 2013).

Devido à grande utilização da robótica nas mais diversas áreas de atuação, como por exemplo, industrial, acadêmica, doméstica, os robôs devem ter seu funcionamento otimizado. Uma forma de se obter essa otimização, é a utilização das RNAs em robótica. Como exemplo, por meio da utilização dessa técnica, resultados obtidos em simulações utilizadas para prever o comportamento de robôs se tornam mais satisfatórios, o que permite, por exemplo, que falhas de projeto ou oportunidades de otimização, possam vir a ser detectadas.

Dessa forma, a capacidade de prever o comportamento de um robô antes de seu desenvolvimento formal, é de suma importância para o sucesso de um projeto (Camargo *et al*, 2014).

No presente trabalho, a rede neural foi utilizada na identificação da dinâmica do *Robotino*, que é um robô móvel didático fabricado e desenvolvido exclusivamente para o treinamento no campo de automação e tecnologia. Este sistema de robô possui utilização em diversas linhas de desenvolvimento, pois além de sua capacidade física para implementação de aplicações diversas, permite desenvolver uma navegação autônoma através da criação de algoritmos de controle por meio de programação (Weber, 2007).

Desta forma, este trabalho pretende investigar a eficiência das redes neurais artificiais no processo de identificação da dinâmica de um robô móvel, tendo em vista a sua simplicidade de implementação e a possibilidade de otimização do sistema de controle do robô.

2 Robotino

A seguir são apresentadas as principais características do *Robotino*, em termos dos componentes estruturais de hardware e de software.

2.1 Estrutura mecânica (hardware):

Através da figura 1 a seguir é possível visualizar o sistema de robô móvel (*Robotino*) utilizado e algumas de suas características.



Figura 1. *Robotino*

- Rodas omnidirecionais;
- 3 motores DC com redutores;
- Sensores diversos;
- Câmera com conexão USB;
- Ponto de acesso a rede wireless LAN;

- Display para exibição de informações necessárias;

2.2 Estrutura de interface (software):

Juntamente com o *Robotino*, são disponibilizados um software de simulação e um software de programação, por blocos funcionais, para realizar as simulações do robô. Estes softwares são desenvolvidos pela própria empresa de fabricação do dispositivo móvel (*Festo*), portanto apresenta dinâmica bastante fiel ao comportamento real do robô. Para a simulação utiliza-se o software *Robotino® SIM Demo* e, para a programação, o software *Robotino® View*.

Os ambientes de simulação e programação estão exibidos nas figuras 2 e 3 respectivamente a seguir.

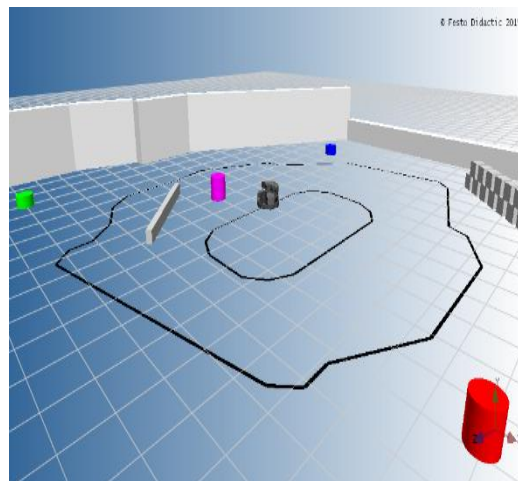


Figura 2. *Robotino® SIM Demo*

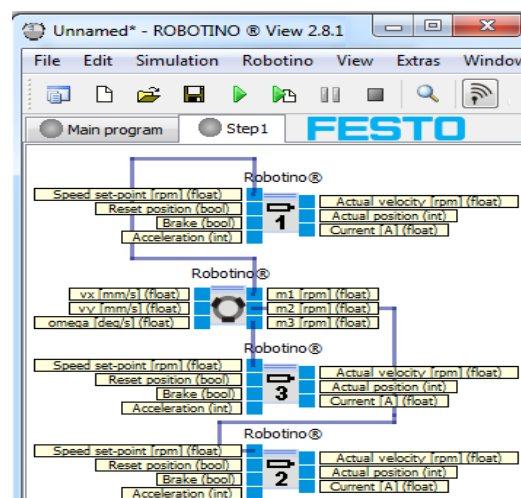


Figura 3. *Robotino® View*

Além da programação por blocos, os comandos do robô podem ser realizados em outras linguagens de programação, entre elas estão o C, C++, Java, .Net, Matlab, LabVIEW e Microsoft Robotics Developer Studio.

3 Redes Neurais Artificiais

O desenvolvimento do estudo em RNAs tem se consolidado desde que o cérebro humano foi reconhecido como um processador de informações complexo, não linear e paralelo. Ou seja, é um computador (sistema de processamento de informações) que difere totalmente de um computador digital convencional (Haykin, 2001).

Através de sua organização estrutural por meio de neurônios, o cérebro tem a capacidade de realizar certos processamentos (reconhecimento de padrões, controle motor, entre outros) muito mais rapidamente que o mais eficiente computador digital hoje existente (Haykin, 2001).

Esta capacidade de alta performance e processamento dá-se por meio de uma grande e complexa estrutura de neurônios. Isso permite que o cérebro tenha habilidade de desenvolver sua própria linha de aprendizado por meio do conhecimento empírico adquirido (Ferneda, 2006).

A rede neural artificial é constituída de unidades de processamento mais simples distribuídas paralelamente. Ou seja, o processamento de informações de uma rede neural se assemelha ao processamento do cérebro, já que o conhecimento adquirido pela rede também dá-se a partir de um processo de aprendizagem por meio de seu ambiente de estudo. Além disso, assim como no cérebro, o armazenamento de conhecimento na rede neural é obtido por meio de conexões entre as unidades de processamento, através de forças de conexão conhecidas como pesos sinápticos. Este procedimento é conhecido como algoritmo de aprendizagem (Bertachi *et al.*, 2013).

Existem várias arquiteturas de redes neurais, tais como, as redes de funções de base radial, redes recorrentes, múltiplas camadas, etc. Neste trabalho será abordada a rede do tipo *Multi-Layer-Perceptron* (MLP), conhecida também como rede neural de múltiplas camadas. Este tipo de arquitetura consiste em uma RNA organizada em camadas, onde só há conexão entre neurônios de camadas vizinhas. São nessas conexões, ou pesos sinápticos, que reside a capacidade de armazenamento de informações de uma RNA (Linhares *et al.*, 2007).

Uma rede neural possui pelo menos uma camada de entrada e uma de saída. Entretanto, para que a rede possa aprender

problemas com um mínimo de complexidade, deve existir ao menos uma camada intermediária, denominada camada oculta. Cybenko (1989), demonstrou que qualquer função contínua pode ser aproximada por uma RNA com apenas uma camada oculta formada por neurônios com funções de ativação sigmoidais e camada de saída formada por neurônios lineares.

Para efetuar o treinamento de uma MLP, uma das alternativas é utilizar o algoritmo da retropropagação do erro (*backpropagation*). Esta é a mais usual técnica de aprendizado supervisionado. Neste tipo de aprendizado a rede adquire informações a partir de um conjunto de pares de entrada-saída fornecidos à rede através da figura de um supervisor (Martiniano *et al.*, 2016).

3.1 Modelo de identificação neural

Um dos procedimentos realizados para identificar a dinâmica presente em ambientes de simulação ou sistemas reais é a escolha de um modelo ou estrutura de identificação a ser utilizado. De uma forma geral, seleciona-se uma família de estruturas de modelagem apropriada para descrever a dinâmica do sistema, e em seguida é feita a seleção de um subconjunto desta família de estruturas de modelagem.

Neste trabalho, utilizou-se a família das redes neurais de múltiplas camadas. A estrutura de identificação utilizada para estimar as velocidades do *Robotino* foi o modelo NNFIR (*neural network finite impulse response*). Este tipo de estrutura é BIBO (*Bounded Input, Bounded Output* – entrada limitada, saída limitada) estável por não possuir realimentação de saída estimada. Esta é uma característica importante para a análise de estabilidade de sistemas não-lineares, devido a estes sistemas terem um comportamento mais complexo do que os sistemas lineares (Araújo Júnior, 2014).

A saída estimada pela rede, $\bar{y}$, é calculada pela expressão da equação 1 a seguir:

$$\bar{y}(k+d) = g[\varphi(k,d,n).W] \quad (1)$$

onde W é a matriz de pesos. O termo φ representa o vetor regressor, que é definido pelos parâmetros k (instante), d (atraso de transporte do sistema) e n (ordem de saída do sistema).

Para o modelo não-linear NNFIR, o vetor regressor φ é calculado pela equação 2 abaixo.

$$\varphi(k,d,n) = [u(k), u(k-1), \dots, u(k-n), -1] \quad (2)$$

Na equação 2 anterior, o termo u representa a entrada do sistema.

A seguir, na figura 4, é possível ver o esquema correspondente às equações descritas anteriormente do modelo NNFIR.

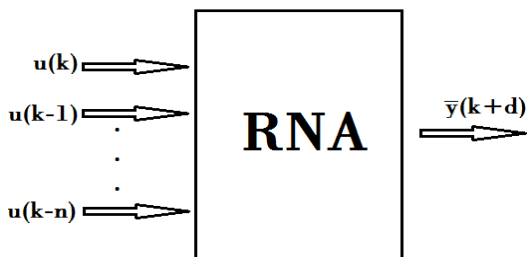


Figura 4. Estrutura NNFIR

Como é possível ver, nesta figura, as entradas da rede consistem num vetor de regressores da variável $u(k)$.

4 Procedimentos de Treinamento e Validação

Um sistema básico de identificação, independentemente da estrutura de modelagem utilizada, é formado por quatro etapas: coleta de dados, seleção da estrutura de modelagem, estimativa do modelo dinâmico e validação do modelo.

A primeira etapa do processo tem como objetivo coletar do sistema a ser identificado conjuntos de dados que descrevam o seu comportamento dentro de sua amplitude de operação. A ideia consiste em variar os sinais de entrada do sistema e verificar a influência desta variação nos sinais de saída do mesmo. A seguir, no esquema da figura 5, é possível visualizar este processo.

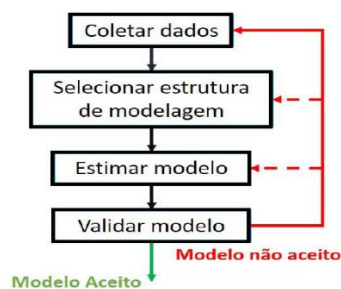


Figura 5. Sistema básico de identificação

Para a obtenção das amostras de dados a serem utilizadas nas etapas de treinamento e validação do modelo obtido, utilizou-se a interface de simulação do *Robotino* juntamente com a linguagem de programação do software MatLab.

Através da conexão do MatLab com o simulador, foi executado um código seguidor de linha, ou seja, através do ambiente de simulação fez-se com que o *Robotino* virtual seguisse uma linha não retilínea com o objetivo de variar as velocidades de cada servo do robô, sendo estes, identificados pelas siglas M1, M2, M3. Com a movimentação criada pelos 3 servos, foi feita a coleta das velocidades, através do MatLab, nas direções x , y e ω . A figura 6 especifica essas direções e a posição dos servomotores.

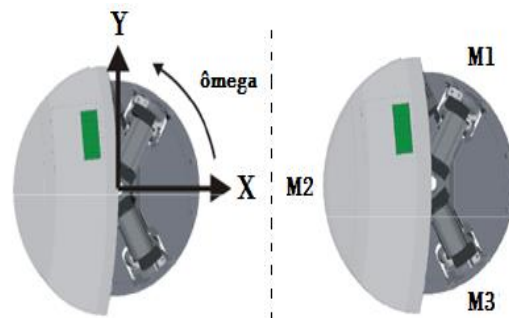


Figura 6. Direções de movimentação do robô

Anteriormente foi discutida a questão da escolha de uma estrutura de modelagem. Devido ao comportamento não-linear dos dados coletados, optou-se por utilizar uma rede neural do tipo MLP, onde esta é formada pela camada de entrada, camada oculta com neurônios utilizando funções de ativação sigmoidais e camada de saída.

Na etapa de validação do modelo dinâmico, a partir dos dados coletados e da estrutura de modelagem selecionada, é possível comprovar o sistema em análise. Isto consiste em efetuar o treinamento de uma MLP.

Neste trabalho, para o treinamento da rede, foi utilizado o algoritmo *backpropagation* (Haykin, 2001). Nesta etapa, o grupo de dados coletados foi dividido de maneira que 80% da coleta fosse destinada ao treinamento e 20% destinada à sua validação. Os dados de entrada e de saída do sistema foram normalizados para valores entre 0 e 1 (Vieira *et al*, 2005), com o intuito de facilitar a realização do treinamento e adequar os dados à função de ativação.

Após o modelo dinâmico do sistema ser aferido é necessário averiguar se este atende aos requisitos exigidos pelo sistema. A etapa de validação do modelo consiste em testar os 20% dos dados coletados durante a simulação, destinados à esta etapa, como informado anteriormente. Caso o valor estimado pela MLP ainda não seja aceitável, deve-se retornar para alguma das etapas de treinamento com o intuito de otimizar o modelo de estimação da rede.

5 Resultados

Os resultados foram obtidos utilizando-se uma MLP formada por três camadas. O objetivo é encontrar uma configuração (número de neurônios, ajuste dos pesos sinápticos e número de épocas) que seja capaz de identificar a dinâmica do *omnidrive* do *Robotino* a partir da variação das velocidades de seus servos.

A movimentação do robô foi simulada computacionalmente no software de simulação *Robotino® SIM Demo*. O treinamento e validação da rede neural estudada foram realizados através da implementação de uma MLP no ambiente de programação MatLab.

Foram testadas diversas configurações de rede, variando-se o número de neurônios da camada oculta, a magnitude dos grupos de dados para o treinamento e validação do modelo encontrado. O número de neurônios da camada oculta foi variado de 1 até 10 neurônios. O número de épocas testado variou de 100 até 10000 épocas. A rede apresentou também um coeficiente de aprendizagem de 0,2.

Os melhores resultados foram obtidos a partir do grupo de dados de 1300 amostras, com 1000 épocas, apresentando uma estrutura 3:4:3 (3 neurônios de entrada, 4 neurônios na camada oculta, 3 neurônios de saída). O erro médio quadrático (EMQ) de treinamento correspondente às saídas da rede foi de 0,01237, sendo este o menor valor encontrado após a realização de vários treinamentos. A curva de convergência do EMQ está apresentada na figura 7.

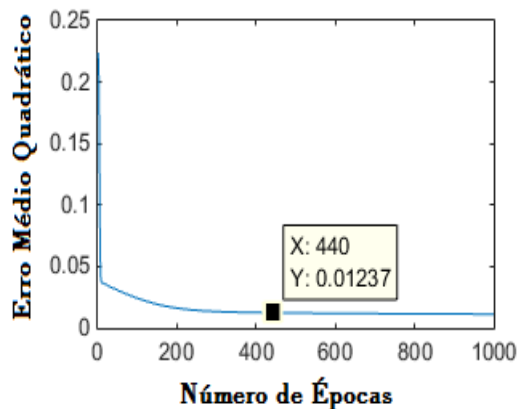


Figura 7. Curva de convergência do erro quadrático médio

Na validação do modelo dinâmico do *omnidrive*, foram consideradas as velocidades nas direções x e y em conjunto com o deslocamento angular. Os gráficos de validação nas direções x, y e ômega do modelo dinâmico construído através de uma MLP estão representados respectivamente nas figuras 8, 9 e 10 a seguir.

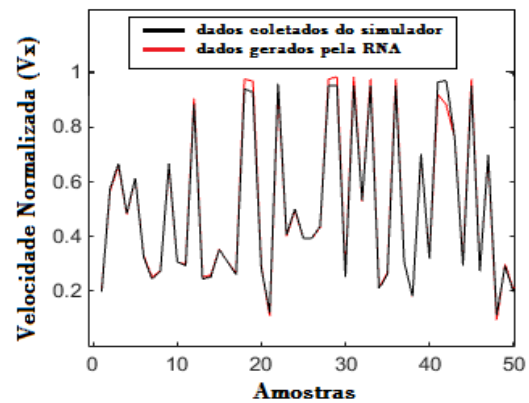


Figura 8. Validação na direção X

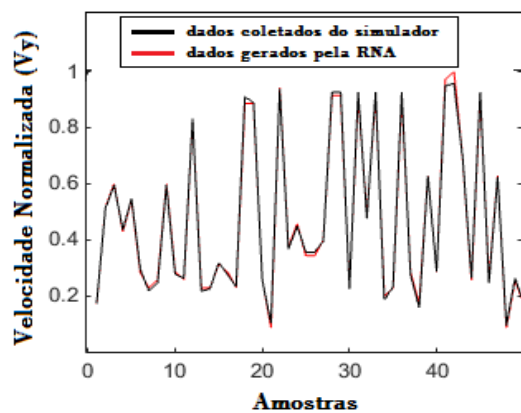


Figura 9. Validação na direção Y

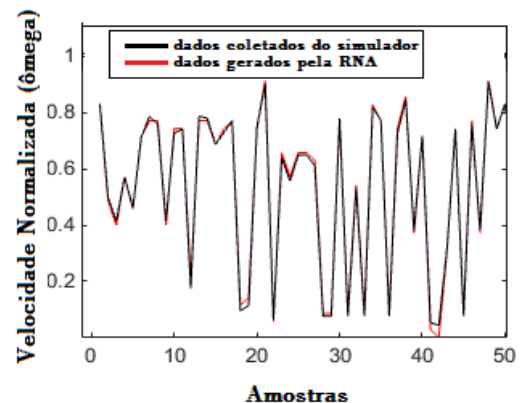


Figura 10. Validação na direção ômega

Como é possível observar, a estimativa das velocidades nas direções x, y e ômega realizada pela RNA foi muito próxima da desejada, apresentando valores estimados satisfatórios para o objetivo do trabalho. Mesmo nas áreas extremas, a saída da rede ficou suficientemente próxima da saída desejada, confirmando a capacidade da rede em identificar a dinâmica do robô.

6 Conclusão

Este trabalho teve como objetivo, investigar a eficiência da utilização de uma rede neural artificial na identificação da dinâmica de um robô omnidirecional, de modo que a partir desse modelo identificado, seja possível estimar as velocidades do *omnidrive* do *Robotino*, um robô móvel de caráter didático fabricado e desenvolvido exclusivamente para o treinamento no campo de automação e tecnologia. Isto foi realizado tomando como base os valores das velocidades coletadas, por meio de simulação, bem como a relação entre as velocidades dos servos e as velocidades de direção do robô. Os resultados obtidos mostram que a aplicação de redes neurais para a configuração de um *omnidrive* é bastante satisfatória.

Desta forma, as estimativas obtidas podem ser utilizadas em conjunto com estratégias de controle, permitindo a construção de um sistema de navegação eficaz, bem como a possibilidade de diminuição do número de sensores presentes no robô.

Em um próximo trabalho, visando aplicações reais, análises estatísticas preliminares e questões de condicionamento de dados devem ser levadas em consideração.

Agradecimentos

Os autores agradecem à CAPES pelo apoio financeiro e, também, ao departamento de engenharia elétrica da Universidade Federal do Piauí (UFPI) pela estrutura e equipamentos oferecidos para o desenvolvimento do trabalho

Referências

- Haykin, S. (2001). *Redes neurais: princípios e prática*. trad. Paulo Martins Engel. - 2.ed. - Porto Alegre: Bookman.
- Pessin, G. (2013). *Estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs*. USP - São Carlos.
- Camargo, J. T. F., Veraszto, E. V., Barreto, G. (2014). *Simulação do modelo cinemático inverso de um robô através do uso de redes neurais artificiais: um complemento ao ensino de robótica*. COBENGE - Congresso Brasileiro de Educação em Engenharia, São Paulo.
- Wolf, D. F., Osório, F. S., Simões, E., Trindade Jr., O. (2009). *Intelligent Robotics: From Simulation to Real World Applications*. XXIX Congresso da SBC - Sociedade Brasileira de Computação, Rio Grande do Sul - RS.
- Pessin, G., Osório, F., Musse, S., Nonnemacher, V., Ferreira, S. S. (2007). *Desenvolvimento de um ambiente para simulação de robótica móvel em áreas florestais*. VI Simpósio de Informática da Região Centro do Rio Grande do Sul.
- Weber, R. (2007). *Robotino Manual*. Festo Didactic - Denkendorf, Germany.
- Martiniano, A., Ferreira R. P., Ferreira A., Sassi, R. J. (2016). *Utilizando uma rede neural artificial para aproximação da função de evolução do sistema de Lorentz*. Revista Produção e Desenvolvimento, Rio de Janeiro - RJ.
- Cybenko, G. (1989). *Approximation by superpositions of a sigmoidal function, mathematics of control, signals, and systems* 2(4): 303-314.
- Vieira, W. G., Sodré, C. H., Barcellos, K. B. and Dantas, L. C. (2005). *Process identification with artificial neural network applied to experimental data from a continuous distillation column*, VII Congresso Brasileiro de Redes Neurais, Natal-RN.
- Linhares, L. L. S., Araújo Júnior, J. M., Araújo, F. M. U. (2007). *Redes neurais artificiais para identificação da fração molar de pentano na composição do GLP*. VIII Simpósio Brasileiro de Automação Inteligente, Santa Catarina.
- Ferneda E. (2006). *Redes neurais e sua aplicação em sistemas de recuperação de informação*. Ciência da Informação - SciELO.
- Bertachi, A. H., Silva, L. R. B., Sumar, R. R., Angelico, B. A., Goedel, A. (2013). *Controle de um processo multivariável em uma planta didática industrial utilizando redes neurais*. XI Simpósio Brasileiro de Automação Inteligente, Ceará.
- Araújo Júnior, J. M. (2014). *Identificação não linear usando uma rede fuzzy wavelet neural network modificada*. Rio Grande do Norte - RN.