



USO DE REDES NEURAIS ARTIFICIAIS PARA MODELAGEM DO DESLOCAMENTO DE UM ROBÔ MÓVEL

Francisco Leão de Oliveira – fco.leao@hotmail.com
José Maria Pires de Menezes Júnior – josemenezesjr@ufpi.edu.br
José Medeiros de Araújo Júnior – jmjúnior@ufpi.edu.br
Stéfano Coelho Santos – stefano_sc_@hotmail.com
Universidade Federal do Piauí, Engenharia Elétrica
Campus Universitário Ministro Petrônio Portella – Bairro Ininga
CEP 64049-550 – Teresina – PI – Brasil

Resumo: *O robô é uma ferramenta de trabalho que nos permite novas formas de interação com o mundo, tendo um novo pensamento lógico, para que através de um meio de aprendizado contínuo sejamos capazes de colocar em prática conhecimentos adquiridos na graduação. Este trabalho é resultado de um projeto que tem como objetivo desenvolver um algoritmo de navegação robótica para deslocamento de um robô móvel. Como método foram aplicadas técnicas de navegação utilizando um robô didático da Festo (Robotino), sendo esse controlado pelo software MatLab. Foram feitas atividades explorando os sensores de infravermelho, sensores indutivos, a câmera e técnicas de navegações. Foi implementado um programa capaz de reconhecer um joystick e, através dele, fazer com que o Robotino se movimentasse em todas as direções (x, y), inclusive rodar em torno de seu próprio eixo (ômega). Por fim foi usado uma Rede Neural Artificial para modelar os dados de entrada e saída colhidos no robô.*

Palavras-chave: *Robotino, Rede Neural, Técnicas de Navegação, Engenharia.*

1 INTRODUÇÃO

Os robôs são dispositivos eletromecânicos ou biomecânicos capazes de realizar trabalho de maneira autônoma e pré-programada. Mas nem todos os sistemas automáticos são robôs, pois os sistemas automáticos de funções fixas não são assim considerados, pois não são reprogramáveis, a exemplo alguns brinquedos (ALMEIDA et al. 2013).

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 em nossa sociedade vem demonstrando o quão promissor é o futuro desta área. Por exemplo, seu uso em aplicações domésticas (e.g. aspiradores de pó e cortadores de grama robóticos), industriais (e.g. transporte automatizado e veículos de carga autônomos), urbanas (e.g. transporte público, cadeiras de rodas robotizadas), militares (e.g. sistemas de monitoramento aéreo remoto - VANTs, transporte de suprimentos e de armamento em zonas de guerra, sistemas táticos e de combate) e de segurança e defesa civil e militar (e.g. controle e patrulhamento de ambientes, resgate e exploração em ambientes hostis), demonstra a grande



gama de aplicações atuais dos robôs móveis e os interesses econômicos envolvidos em relação ao seu desenvolvimento e aplicação (WOLF et al., 2009).

O estudo de Robôs móveis é um tema motivador e interessante para a educação em todos os níveis de qualificação. Na Alemanha este sistema de aprendizagem é utilizado na educação geral, desde as escolas, cursos técnicos avançados e em universidades, para transmitir conhecimento em diversos campos da engenharia e tecnologia da informação (WEINERT & PENSKEY, 2012).

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 de aprendizagem de máquinas em robótica, como exemplo, a utilização de RNAs (Rede Neurais Artificiais).

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).

A utilização de robôs móveis para aplicação de RNAs tem se mostrado interessante. A união entre robôs e o ensino, sendo ela aplicada a alguma área de conhecimento, mostra-se propícia, pois o robô possui elementos tecnológicos e uma série de conceitos científicos, onde seus princípios básicos são abordados em grande parte das graduações de Engenharia. O ambiente de aprendizagem onde um professor ensina um aluno a aplicar conhecimentos de automação e controle em um dispositivo mecânico que podem ser controlados pelo computador é denominado de *Robótica Pedagógica* (SILVA, 2009).

A RNA é constituída de unidades de processamentos 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 se dá 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 (HAYKIN, 2001).

Simplificadamente, as redes neurais podem ser classificadas em supervisionadas e não supervisionadas; e para cada um dos dois tipos existem vários modelos disponíveis (BRAGA, 2007). A supervisão neste caso é atribuída ao fato da rede treinar segundo alguma disposição preliminar que a supervisiona a um resultado; a não supervisão se dá quando os pesos se ajustam por conta própria, sem a necessidade dessa disposição.

Entre as redes supervisionadas, ressalta-se o *perceptron* multicamada (do inglês, *Multilayer Perceptron*, MLP), treinada com algoritmo de retropropagação de erro (*error back-propagation*), que se baseia em aprendizagem por correção de erro por propagação e retropropagação. Durante a aprendizagem, um conjunto de formação de padrões de entrada é apresentado a MLP, que calcula o padrão de saída; caso haja diferença entre a saída da rede e a desejada, os pesos são ajustados para reduzir este erro (Negnevitsky, 2005).

Em um *perceptron* de camada única, há apenas um peso para cada entrada e somente uma saída. No entanto, na MLP, existe uma camada de entrada, uma ou mais ocultas (ou intermediárias, ou escondidas) e uma de saída, com vários pesos em cada uma delas. A capacidade da MLP de aprender tarefas complexas se dá exatamente por conta dos neurônios da camada escondida, extraíndo progressivamente a cada época as características mais significativas dos padrões de entrada. Comumente, para aproximação de função, uma tarefa não tão complexa para uma ferramenta computacional poderosa como uma rede neural, utiliza-se apenas uma camada intermediária (RODRIGUES, et al., 2016).



O presente estudo tem como objetivo apresentar resultados da aplicação de uma rede neural artificial MLP aplicada em técnicas de navegação robótica para modelagem do deslocamento de um robô móvel (Robotino).

2 ROBOTINO

O Robotino é um robô didático projetado pela Festo para o aprendizado industrial. Utilizando uma tecnologia de movimento omnidirecional e com um corpo em forma de disco com sensores ultrassônicos em seu perímetro, ele é o robô ideal para cumprir tarefas e participar de competições. Sua programação é baseada no Robotino View, forma gráfica de programação em blocos, que permite que usuários de vários níveis de conhecimento possam programa-lo. Ainda assim, ele pode ser programado em outras linguagens, como C++, Java ou Matlab (FESTO, 2010).

O Robotino é um robô circular com três rodas de movimento independente e omnidirecional. Ele é equipado com nove sensores de distância e um sensor de colisão em sua circunferência, permitindo um bom controle de seus movimentos, além de uma câmera pela qual o robô consegue visualizar o ambiente em sua volta (TREVISAN, 2013). O Robotino possui um simulador, Robotino Sim Demo, onde é possível simular sua movimentação omnidirecional juntamente com suas funções e sensores. Através da Figura 1 a seguir é possível observar o Robotino.

Figura 1. - Robotino



3 METODOLOGIA

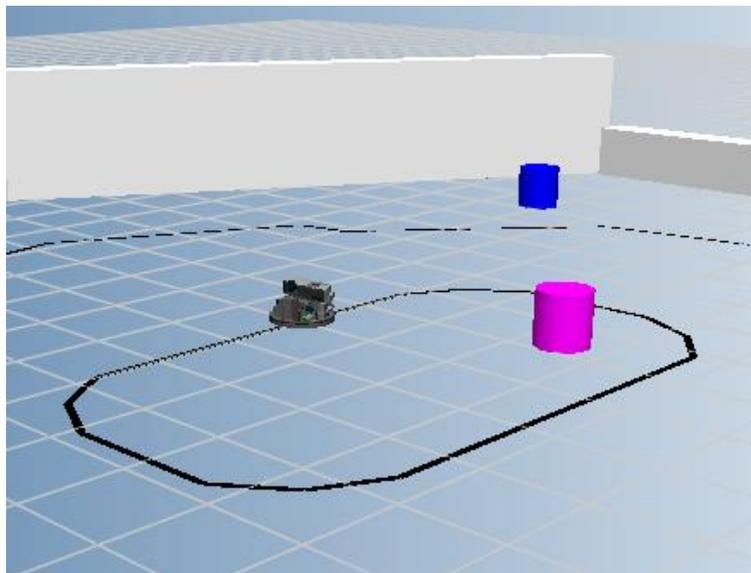
O Robotino é programado em várias linguagens de programação, inicialmente foi estudado suas funções e sensores, utilizando seu simulador Robotino Sim Demo e a interface própria da Festo de programação, Robotino View. Foram realizadas atividades explorando os sensores de infravermelho, sensores indutivos, a câmera e técnicas de navegações.

Paralelamente a programação utilizando o Robotino View foi estudado uma forma de utilizar a linguagem Matlab para programar o Robotino. Para isso fez necessário o reconhecimento e comunicação do Matlab com o Robotino, através de APIs e toolbox específicas. Após esta etapa de comunicação foi desenvolvido um programa capaz de reconhecer um joystick, e através dele, fazer com que o Robotino se movimentasse em todas as direções (x, y), inclusive rodar em torno de seu próprio eixo (ω).



Em primeiro momento o programa foi implementado e testado utilizando o simulador Robotino SIM, onde foi possível analisar cada simulação desenvolvida durante os testes realizados. Na Figura 2 pode-se observar o ambiente de simulação desses testes.

Figura 2. Interface do Simulador Robotino.



Durante as simulações, desenvolveu-se os seguintes exercícios com o robô:

- Seguir uma linha com curvas;
- Deslocamento especificado em diversas direções;
- Detecção de cores;
- Posicionamento através de sensores.

Com as simulações foram verificadas diversas possibilidades de movimentação com o Robotino, permitindo compreensão da capacidade de seu sistema. Isso foi possível devido à análise dos dados durante cada exercício desenvolvido, como posição, velocidade e direcionamento do robô.

Para a aquisição de dados, foram armazenadas as informações dos sensores infravermelhos e de posição do robô. Em seguida, os dados foram tratados, fazendo-se um pré-processamento e caracterizando uma série temporal em questão. Em seguida foram estudadas técnicas de modelagem, partindo de técnicas mais simples e clássicas até técnicas mais complexas, utilizando redes neurais artificiais, para reconhecimento e navegação do robô. Desta forma, este projeto fez uso do problema de imitação de comportamento por meio de aprendizagem de máquinas, não só revisitando uma estratégia clássica de navegação - o método de seguir paredes (wall-following) - mas também realizando um estudo de caso como a dimensão temporal influencia o desempenho dos classificadores nesta tarefa.

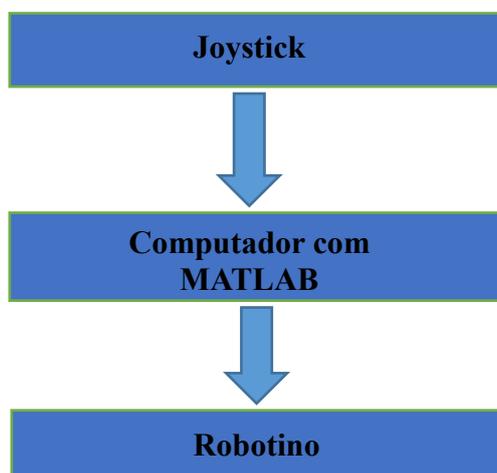
4 RESULTADOS E DISCUSSÕES

Todo sistema montado está representado na Figura 3. O joystick é conectado ao computador através da porta Usb e o MatLab, através de uma função própria, reconhece o



joystick atribuindo valores aos seus respectivos botões. Dessa forma, foi implementado um código que utiliza a entrada (botões do joystick) para incrementar velocidades e direções no Robotino, que são as saídas do sistema a ser modelado.

Figura 3. Sistema de funcionamento.



Esta etapa obteve desempenho de acordo com o esperado, com Robotino respondendo de forma precisa a todos os comandos do controle. A Figura 4 mostra um modelo do controle utilizado. As teclas do joystick possuem as seguintes funções incrementadas ao robô:

- Teclas direcionais horizontais: movimento no eixo X, movimento para esquerda e direita.
- Teclas direcionais verticais: movimento no eixo Y, movimento para frente e para trás.
- Tecla R1: rotaciona o robô no sentido horário.
- Tecla L1: rotaciona o robô no sentido anti-horário.
- Tecla 3: aumenta a velocidade.
- Teclas 2: diminui a velocidade.
- Tecla 1: aciona os freios.

Figura 4. Joystick.



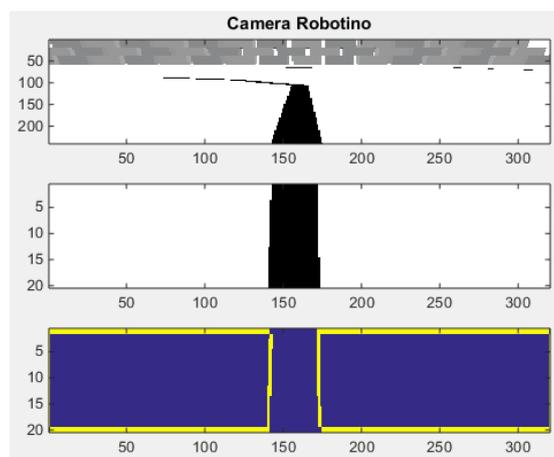
Com essas configurações foi possível movimentar o Robotino de diversas formas, entre elas fazendo com que o Robotino seguisse uma linha com curva. Para isso foi necessário um aperfeiçoamento da prática de controle do robô através sucessivas simulações em laboratório.



Após o controle estar funcionando, foi implementado um código para verificar a imagem em tempo real, enquanto o robô estivesse em movimento. Na Figura 5 é mostrado 3 imagens: a primeira imagem refere-se ao captado pela câmera, a segunda trata-se de um corte de 20 pixels no sentido vertical da primeira imagem e a terceira traz uma imagem binarizada (imagem que possui apenas dois níveis de cor) da segunda imagem.

Com o controle e a imagem em tempo real funcionando tornou-se possível que o usuário controlasse o robô pela tela do computador, possibilitando visualização precisa do movimento do robô ao seguir a linha, inclusive nos momentos de curva, onde a linha sai do centro da imagem. A medida que a linha se afasta do centro da imagem o erro tende a aumentar, pois o erro é calculado na diferença entre o pixel central e o pixel da linha. Dessa forma o principal objetivo aqui é manter a linha observada na Figura 5 no centro da imagem, preservando, assim, o erro o menor possível.

Figura 5. Imagem da câmera do Robotino.



Para fazer com que o robô seguisse a linha foi utilizado dois movimentos, o primeiro a uma velocidade constante na direção da linha, ou seja o robô andaria para sua frente. Já o segundo movimento seria o rotacional em torno de seu eixo, onde o robô faria a rotação para fazer uma curva, sendo que a velocidade de movimentação dele para frente continuaria constante.

A medida que o robô se afastava do centro da imagem, da linha central, principalmente nos locais de curva, é possível corrigir o posicionamento do robô utilizando o controle para fazer modificações na rotação de eixo do robô fazendo assim com que o robô gire fazendo a curva, colocando a linha no centro de sua imagem captada pela câmera. A velocidade angular (V_w) usada para girar o robô é em função do erro, quanto maior o erro maior será a velocidade de giro, tanto no sentido horário como no anti-horário.

Foi também implementado um código em Matlab capaz de coletar os valores dos erros da velocidade em Y e da velocidade em ômega. Em seguida esses valores foram armazenados numa matriz de dados e salvos em um arquivo de texto. Através desses dados foi possível implementar uma rede neural para treinar o robô afim de percorrer a pista com menor erro possível.

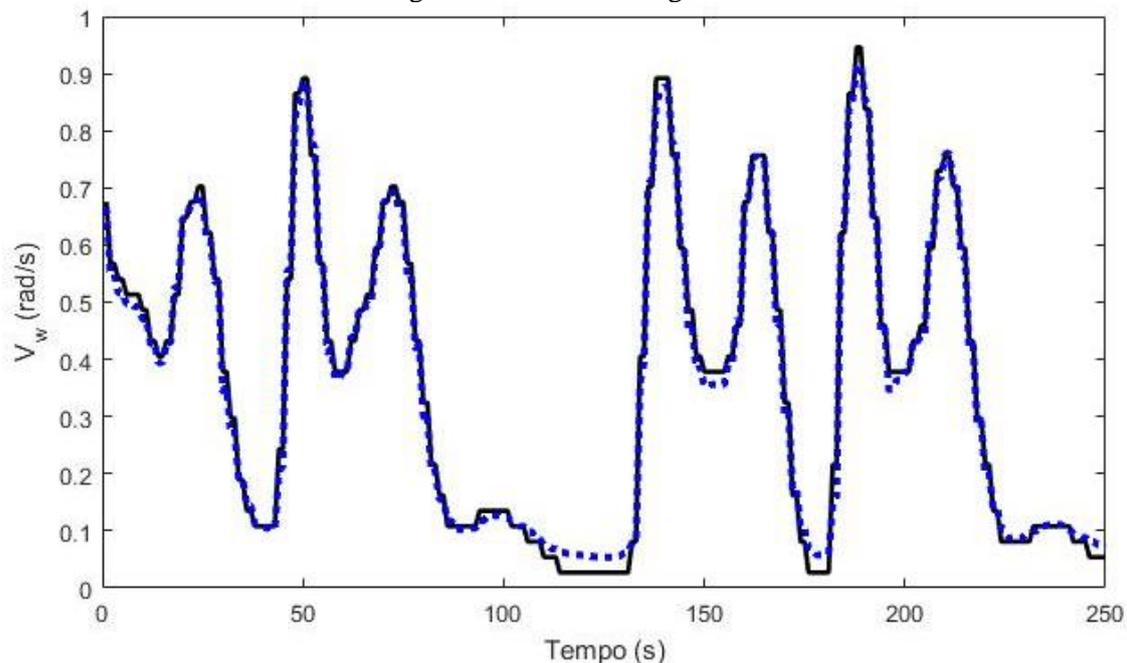
A rede neural aqui utilizada tem como objetivo modelar o sistema de correção de trajetória implementado pelo joystick e com intervenção humana. Dessa forma, após a rede



MLP ter sido treinada e testada, ela teria por meta fazer correções automáticas de trajetória de acordo com o erro detectado entre o centro do robô e a linha a ser seguida.

Para cumprir tal atividade, foi utilizando uma RNA do tipo MLP, com dez neurônios da camada escondida, três entradas, para fazer a modelagem do sistema de entrada e saída da movimentação do robô. Como entrada da RNA foi utilizado o vetor de regressão do erro (entre o centro do robô e a linha a ser seguida) atual e mais dois consecutivos erros anteriores. Como saída da rede, foi analisado a correção de trajetória, utilizando dessa forma a velocidade ômega (V_w). A Figura 6 mostra os resultados após a RNA ter sido treinada. Os valores para saída da velocidade ômega reais são visualizados em preto contínuo e os valores gerados pela rede em azul pontilhado.

Figura 6. Velocidade angular.



Como é possível observar, a estimativa para a velocidade 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 de rede em identificar a dinâmica do robô.

5 CONCLUSÃO

O Robotino é rico em aplicações voltadas para o ensino de automação e controle. Portanto, com a ajuda do MatLab é possível fazer projetos voltados à Robótica. Utilizar um joystick é uma forma interessante de controlar o Robotino, coletando informações de diversos tipos de sensores e usando os dados para implementar técnicas capazes de realizar atividades de forma autônoma.

A Rede Neural MLP implementada apresenta resposta satisfatória para a modelagem da função real da velocidade angular do Robotino. Entre as quantidades de neurônios implementados observou-se uma maior precisão na saída com o uso de poucos neurônios na camada escondida (entre sete e doze), necessitando de menos épocas para que o treinamento da rede culmine em uma resposta satisfatória quando comparada a função original.



Agradecimentos

Os autores agradecem a Universidade Federal do Piauí (UFPI) e a Coordenação do curso de Bacharelado em Engenharia Elétrica.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, A. R.; CARRARA, A. R. S.; YUAN, W. S. As Vantagens de Robôs de Combate na Área de Pesquisa. COBENGE – Congresso Brasileiro de Educação em Engenharia, Gramado, 2013.

BRAGA, A. P.; FERREIRA, A. C. P. L.; LUDERMIR, T. B. Redes neurais artificiais: teoria e aplicações. LTC Editora, 2007.

CAMARGO, J. T. F.; VERASZTO, E. V.; BARRETO, G. 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, 2014.

FESTO. *Robotino Manual*. Festo Didactic GmbH & Co. KG, 73770 Denkendorf, Germany, 2010.

HAYKIN, Simon. Redes neurais: princípios e prática. trad. Paulo Martins Engel. - 2.ed. - Porto Alegre: Bookman, 2001.

NEGNEVITSKY, M. Artificial intelligence: a guide to intelligent systems. 2. Ed. King's Lynn: Pearson, 2007. pp. 175-185.

RODRIGUES, F. I. B.; CARVALHO, R. R. M.; GONÇALVES, G. P.; BARBOSA, K. B. P. Estudo de Algoritmo MLP como Aproximado de Função. CONTECC - Congresso Técnico Científico da Engenharia e da Agronomia. Foz do Iguaçu – PR, 2016

SILVA, Alzira Ferreira da; UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE, Programa de Pós-Graduação em Ciência e Engenharia Elétrica. RoboEduc: uma metodologia de aprendizado com robótica educacional, 2009. 127p. Tese (Doutorado).

TREVISAN, V. Robotino. Disponível em: < <http://www.roboliv.re/conteudo/robotino>>
Acesso em: 03 jun. 2017.

WEINERT, H.; PENSKY, D. Mobile Robotics in Education – South African and International Competitions. ROBOMECH – Robotics and Mechatronics Conference Of South Africa, Gauteng, South Africa, 2012.

WOLF, D. F. ; OSÓRIO, F. S. ; SIMÕES, E. ; TRINDADE JR., O.. Intelligent Robotics: From Simulation to Real World Applications. Congresso da SBC – Sociedade Brasileira de Computação, São Paulo, 2009.

USE OF ARTIFICIAL NEURAL NETWORKS FOR MODELING THE DISPLACEMENT OF A MOBILE ROBOT

Abstract: *A robot is a tool that allows us to interact with the world in new ways, by having a new logical thinking so that by a continuous learning method we are able to put in practice the knowledge acquired in the undergraduate courses. This work is the result of a project that aims to the development of a navigation algorithm for a mobile robot. Navigation methods were applied to a robot from Festo (Robotino), which is controlled by a MatLab software.*

Organização



Promoção



Joinville/SC – 26 a 29 de Setembro de 2017
UDESC/UNISOCIESC
“Inovação no Ensino/Aprendizagem em
Engenharia”



COBENGE 2017
XLV CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA

Some activities were designed exploring the infrared and inductive sensors and the camera. It was implemented a program capable of recognizing a joystick and, through it, make the Robotino move in all directions (x , y), even rotate around its own (ω) axis. Finally, an Artificial Neural Network was used to model the input and output data collected from the robot.

Key-words: Robotino, Neural Network, Navigation Techniques, Engineering.

Organização



Promoção

