



## **ROBÔS MÓVEIS X CONTROLE POR REDE NEURAL ARTIFICIAL USANDO FPGA MOTIVAM TRABALHOS DE PESQUISA NO CURSO DE ENGENHARIA ELÉTRICA.**

**Suely Cunha Amaro Mantovani** – e-mail: [suely@dee.feis.unesp.br](mailto:suely@dee.feis.unesp.br)  
**Hugo Bachi Artico** – e-mail: [articohugo@gmail.com](mailto:articohugo@gmail.com)

Universidade Estadual Paulista “Júlio de Mesquita Filho”.  
Faculdade de Engenharia Elétrica – FEIS –UNESP  
Endereço: Av. Brasil n.56, Centro.  
CEP – 15385000 –Ilha Solteira – São Paulo.

**Resumo:** *Visando preparar o aluno de graduação para as pesquisas em robótica e Inteligência Artificial foi proposto um trabalho utilizando uma Rede Neural Artificial para o controle de um robô móvel implementado em dispositivos lógicos programáveis-FPGA da família Cyclone II- placa DE2- ALTERA, usando uma linguagem de descrição de hardware, VHDL, de forma a controlar um protótipo desenvolvido em laboratório. A Rede implementada é a uma das mais básicas do tipo Perceptron Múltiplas Camadas, adequada para a aplicação em robótica, cujo treinamento é realizado no software MATLAB, obtendo os dados necessários para o processamento na fase de execução em hardware, sem prejuízo da sua eficiência, quando comparado ao mesmo sistema em software. O FPGA especificado possui elementos lógicos para sintetizar Redes Neurais em uma linguagem de descrição de hardware, facilitando a alteração do projeto, proporcionando a portabilidade entre dispositivos/software de desenvolvimento. Dos resultados observam-se o êxito do projeto aplicando a RNA PMC em FPGA no controle do robô em um percurso predefinido, e o crescimento técnico do aluno, demonstrando maior motivação para o desenvolvimento de trabalhos de Iniciação Científica.*

**Palavras-chave:** *Redes Neurais Artificiais, Robôs móveis, Perceptron Multicamadas, FPGAs.*

### **1. INTRODUÇÃO**

O curso de engenharia elétrica da Unesp de Ilha Solteira tem preparado seus alunos para a pesquisa nas áreas de robótica e automação de sistemas usando algoritmos inteligentes por meio de projetos de Iniciação Científica. Estes projetos são desenvolvidos por alunos que adquiriram os fundamentos teóricos necessários nas disciplinas de Cálculo e Física e estejam iniciando a fase profissional do curso. Desta

Organização



**UDESC**  
UNIVERSIDADE  
DO ESTADO DE  
SANTA CATARINA



Promoção





forma, apresenta-se neste artigo o desenvolvimento de um projeto de iniciação científica que proporcionou ao aluno o aprendizado da teoria e aplicação de Redes Neurais Artificiais na robótica, para controlar um protótipo de um robô móvel buscando sua autonomia, através de sensores ultrassônicos, dispositivos lógicos programáveis e uma linguagem de descrição de hardware.

A Rede Neural Artificial (RNA) é um modelo computacional baseado nas redes neurais biológicas, tem como características a possibilidade de aprendizado pela experiência, capacidade de generalização e tolerância às falhas, sendo comum sua simulação em um computador. Tem sido amplamente utilizada nas mais diversas áreas do conhecimento, tais como no reconhecimento/classificação de padrões, controle de processos (em robótica, por exemplo), controle inteligente, aproximador universal de funções, sistemas de previsão, otimização de sistemas, etc, visando a qualidade, eficiência e segurança dos sistemas. Devido principalmente, ao desenvolvimento dos dispositivos lógicos programáveis FPGAs (*Field Programmable Gate Arrays*) bem como, a sua capacidade de processamento, vários trabalhos têm sido propostos na literatura que abordam a implementação de RNAs em FPGA, em Azhar & Dimond (2002), Colman (2006), Lacerda & Chaves (2012), Camillo & Mantovani (2014), Pião, (2012) e Silva et al (2009).

Um FPGA é um dispositivo projetado para ser configurado pelo usuário após a sua fabricação; por ser programável vem sendo usado em toda e qualquer área, principalmente na educação, pela capacidade de reconfiguração de sua arquitetura ou funções. São constituídos basicamente por *arrays* de blocos lógicos, pinos de I/O, blocos de memórias RAMs e canais de roteamento. Podem ser programados por diagramas esquemáticos ou linguagem de descrição de hardware - HDL. Os FPGAs representam uma importante plataforma de *hardware* para a implementação de uma Rede Neural Artificial, com a possibilidade de modificar dinamicamente os algoritmos.

Nesta trabalho foi programada uma RNA do tipo Perceptron Múltiplas Camadas (PMC) em um FPGA, Kit da Altera-DE2 disponível em laboratório, visando sua aplicação no controle de um protótipo de um robô móvel, com base nas medidas de distâncias dadas por quatro sensores ultrassônicos. e dois motores CC (Corrente Contínua), de forma a fornecer a sua autonomia e mobilidade, respectivamente e com razoável desempenho. A implementação em FPGA torna o sistema semelhante ao comportamento dos neurônios biológicos, porque facilita a alteração do projeto, proporcionando a portabilidade entre dispositivos/*softwares* de desenvolvimento e a execução multitarefa em paralelo, ao nível de *hardware*.

## 2. FUNDAMENTOS TEÓRICOS

As RNAs surgiram na década de 1940 quando o Warren McCulloch, psiquiatra e neuroanatomista do Instituto Tecnológico de Massachusetts e o matemático Walter Pitts da Universidade de Illinois, estudaram o comportamento entre células nervosas vivas e implementaram um modelo eletrônico simulando o comportamento destas células. Este modelo era composto de amplificadores e resistores, e tinha como função o somatório de vários sinais de entradas formando apenas uma saída (HAYKIN, 2001).

Na RNA os neurônios são interligados por sinapse, recebem uma informação, processam e respondem com um resultado, esta informação é processada paralelamente e de forma distribuída. Tem a capacidade de aprender e efetuar operações lógicas. Uma

Organização



**UDESC**  
UNIVERSIDADE  
DO ESTADO DE  
SANTA CATARINA



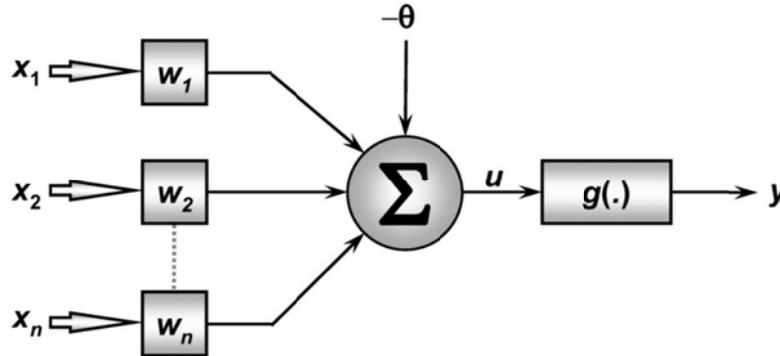
Promoção





das mais simples Redes Neurais é a Perceptron que é constituída apenas por um neurônio artificial com uma camada de entrada e uma camada com apenas uma saída simples e binária, conforme mostrado na Figura 1.

Figura 1 - Exemplo genérico da Rede Perceptron.



Na Figura 1 tem-se as entradas, vetor  $X$ , os pesos, vetor  $W$  e a saída da rede sinal,  $y$ . Tem-se ainda o valor escalar  $\theta$  ( $W_0 \times X_0$ ), ou potencial de ativação, e a saída  $y$  é dada pela aplicação de uma função de ativação,  $g(\cdot)$  em  $u$  (escalar). A camada de entrada,  $X$  é composta por um vetor de pesos sinápticos e uma matriz com os sinais que a rede deve processar. Com o treinamento da Rede, os valores dos pesos são ajustados com métodos iterativos até que a camada de saída seja igual à desejada. Considerando um número significativo de amostras e que a camada de saída da rede é igual ao esperado, utiliza-se a rede para obter uma saída para qualquer valor de entrada. A camada de saída realiza a soma de todos os sinais de entrada já ponderados (multiplicados pelos pesos) e aplica uma função de ativação a esta soma de sinais. No caso da Rede Perceptron, somente podem ser aplicadas duas funções de ativação - a função degrau e a função sinal, que classificam os resultados para apenas dois valores distintos, sendo a função degrau para 1 ou 0, e a função sinal para 1 ou -1.

Definida a arquitetura das RNAs existem duas etapas necessárias para o seu funcionamento que são o treinamento (alteração dos pesos sinápticos) e a execução da rede visando observar o seu aprendizado. O treinamento da Perceptron é realizado somente de maneira supervisionada (requer uma matriz de dados com amostras de entradas e suas respectivas saídas desejadas).

Antes do treinamento, com os pesos iniciais e as entradas de sinais armazenados em vetores e/ou matrizes, é possível calcular as saídas desta rede para estes parâmetros. Como a rede ainda não foi treinada, sua saída provavelmente não vai obter as saídas desejadas, as quais estão armazenadas em outro vetor.

A Rede Perceptron simples é utilizada para classificar amostras que possuem características linearmente separáveis tendo uma saída binária. Em aplicações como a proposta neste trabalho que é requerido controlar dois motores de um robô, de maneira independente, a Perceptron Múltiplas Camadas (PMC) é a recomendada.

A PMC apresenta múltiplas saídas e não binária, sendo aplicada em robótica e constituída de no mínimo 3 camadas, sendo elas, a camada de entrada, a camada

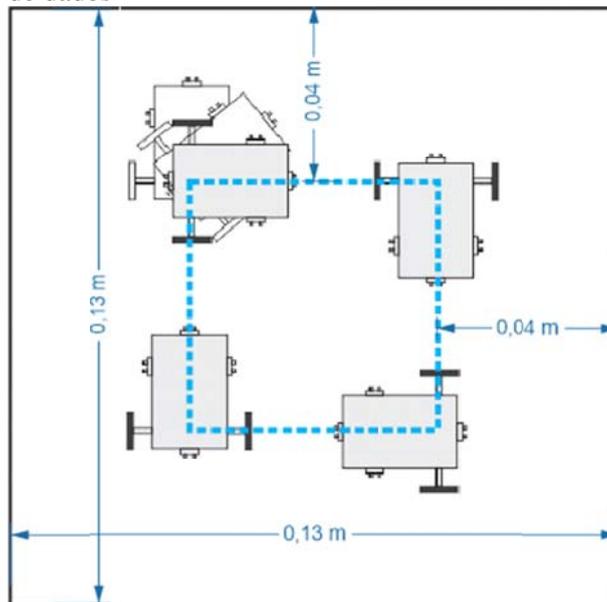


escondida e a camada de saída. Sua construção é semelhante à união de várias Redes Perceptron simples, interligadas de maneira série e paralela. Como a rede é formada de vários neurônios, a camada de saída não precisa ser singular, ou seja, pode ter mais de uma saída (HAYKIN, 2001).

### 3. PMC IMPLEMENTADA PARA O CONTROLE DO ROBÔ MÓVEL

Para implementar uma PMC aplicada ao controle de um robô móvel é necessário montar um banco de dados que modela o percurso utilizado no treinamento da rede, contendo as informações das entradas da rede e por conseguinte os dados dos sensores – distâncias a serem utilizadas no seu treinamento. Banco de dados muito extensos exigem RNAs maiores, possuindo muitas camadas neurais ou camadas com grande número de neurônios. Por outro lado, o banco de dados precisa conter elementos suficientes para que a RNA consiga aprender as informações deste banco, porém a plataforma de desenvolvimento que será executada a PMC, embora ofereça bons recursos de *hardware* e *software*, está limitada aos recursos do processador e memórias. A trajetória que o robô deve realizar, mostrada na Figura 2, forma o banco de dados que contém as distâncias de cada sensor e o sentido de rotação para cada motor.

Figura 2– Trajetória utilizada para compor o banco de dados



O robô tem comportamento diferenciado para cada posição, tais como, seguir em frente, virar a direita, virar à esquerda ou voltar em ré e realizará esta trajetória com os motores em velocidade constante, utilizando as saídas da PMC para alterar o sentido de giros de cada motor (rodando os motores em sentidos contrários), para curvas ou seguir em linha reta com ambos os motores girando no mesmo sentido.

Tem-se na Figura 3 a estrutura da PMC implementada para o controle do robô estruturada em 3 camadas: a camada de entrada, uma camada escondida e a camada de saída. O processamento dos sinais é realizado pelas camadas escondida e de saída. Esta

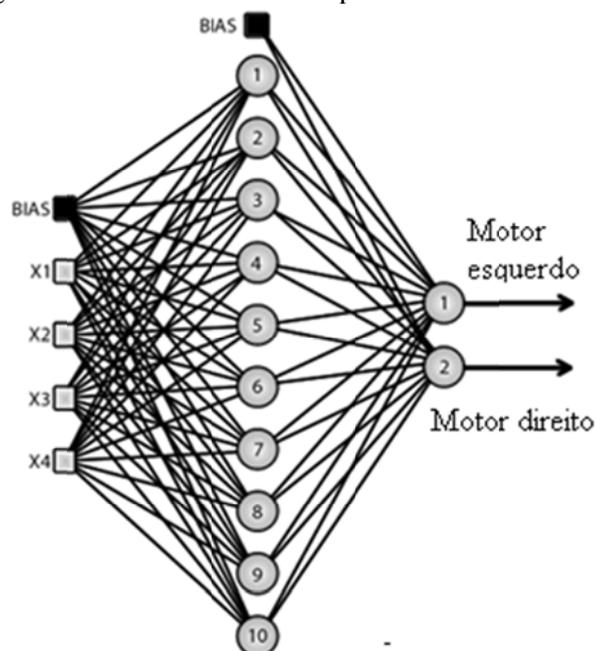


configuração é limitada, principalmente pela capacidade de processamento da plataforma de desenvolvimento.

A **camada de entrada** recebe o número de entradas do sistema (4 entradas), e 4 neurônios na camada de entrada. A **camada escondida** é a única manipulável em número de neurônios, sendo possível adicionar novas camadas escondidas, se necessário. A **camada de saída** deve ter o número de neurônios conforme o número de saídas desejadas, neste caso deve-se controlar os dois motores do protótipo, portanto têm-se dois neurônios. Todas as camadas possuem função de ativação Bi-Hiperbólica e limiar de disparos. As funções de ativação não precisam ser as mesmas para todas as camadas, elas têm como principal premissa ser diferenciáveis para a realização da fase de treinamento.

No treinamento da PMC a rede deve aprender a produzir a saída desejada, cujo algoritmo é composto de duas etapas, a *Backpropagation* e o *Forward* (processo de execução da Rede Neural).

Figura 3– Estrutura da PMC implementada



### 3.1 Resultado do Treinamento da PMC

O treinamento da PMC foi realizado com uma normalização dos sinais de entrada obtendo-se valores menores que 1. O erro médio quadrático obtido no treinamento foi 0,044%, resultando em valores de saída muito próximos aos valores desejados. O erro encontrado por amostra foi de aproximadamente 0,3. O tempo de treinamento foi obtido por épocas (número de vezes necessário para apresentar todas as amostras do conjunto de treinamento visando o ajuste dos pesos). Durante o treinamento foram necessárias 251.763 épocas, sendo que cada época com 612 correções por amostras. Considerando um erro abaixo de 1%, a PMC em questão conseguiria convergir com cerca de 25 mil épocas.



### 3.2 Implementação da PMC em VHDL

O processo *Forward* é implementado em VHDL no ambiente de desenvolvimento de projeto Quartus II e na placa DE2 contendo um FPGA da família Cyclone II, EP2C35F672C6 (ALTERA, 2016).

A linguagem VHDL possui bibliotecas para aritmética com valores inteiros, reais, e binários, nesta implementação optou-se pela base binária, pois proporciona uma melhor visualização e o uso dos recursos da placa.

As conversões para binário foram utilizadas com diferentes dimensões de palavras (bits utilizados para a representação de um número) e com a representação de complemento de 2 (representação de números com sinal), com a dimensão da palavra variável, para isso devem ser usadas as bibliotecas com vetores binários com sinal.

### 3.3 Processo *Forward* da PMC em VHDL

A implementação do processo de *Forward* da PMC programado em VHDL é apresentado no Fluxograma da Figura 4. As camadas neurais, com exceção da camada de entrada, são divididas em duas etapas. A primeira etapa de uma camada neural é a somatória dos sinais de suas entradas ponderadas pelos pesos sinápticos. Esta etapa é realizada salvando os pesos sinápticos (contendo o valor de  $W_0$  para o *bias* obtidos na fase do treinamento) em uma memória ROM e criando acumuladores para cada neurônio da camada. Os acumuladores vão receber o valor do somatório entre a multiplicação dos sinais de entrada e a *bias* com os pesos salvos em constantes.

Em seguida aplica-se a função de ativação aos valores obtido do somatório. A função de ativação para o processo de treinamento da rede é a Bi-hiperbólica com um coeficiente de inclinação de 0,35. Considerando que se trata de uma função não linear e não polinomial, sua implementação em VHDL é complexa, por isso, utilizando processamento paralelo (todos os neurônios realizam cálculos ao mesmo tempo) foram utilizadas linearizações por partes.

## 4. RESULTADOS E DISCUSSÕES

O protótipo montado para o desenvolvimento do projeto é apresentado na Figura 5. Seu chassi foi dimensionado para dois motores de 8,5 rpm (sem carga), modelo P\_VDC fabricado pela MICRORED (2016), e para o tamanho e peso da plataforma de desenvolvimento. No protótipo são posicionados os quatro sensores ultrassônicos, US-100, que medem as distâncias entre o robô e os obstáculos, permitindo o desvio e a sua navegação em um ambiente.

Para a determinação da distância optou-se pelos sensores ultrassônicos devido a medirem distâncias na faixa de 3 cm - 350 cm com boa precisão, possibilitando o controle por uma Rede Neural PMC. O modelo utilizado é o US-100 do fabricante E-Gizmo (2016), constituído por transdutores eletroestáticos, um emissor e um receptor com frequência de ressonância de 40 kHz. Apresenta compensação de temperatura para fazer a correção da medida da velocidade do som, excitação/saída do tipo TTL para tensões de alimentação com tolerância de 3,3 V – 5 V DC.

Organização



Promoção





Figura 4- Fluxograma da execução na plataforma DE2 em VHDL.

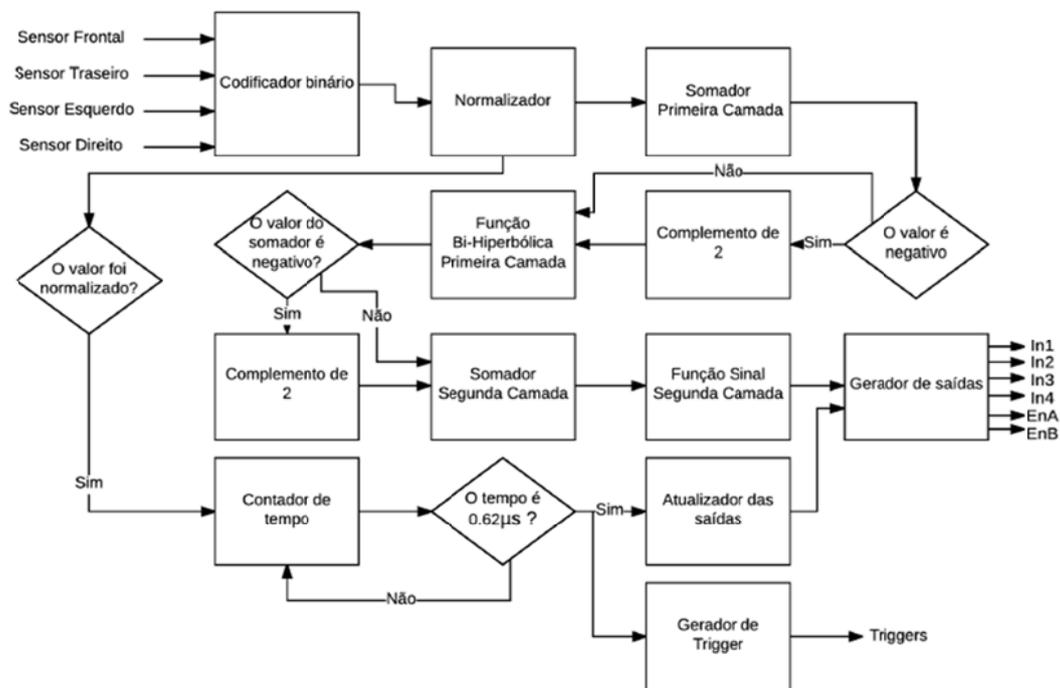
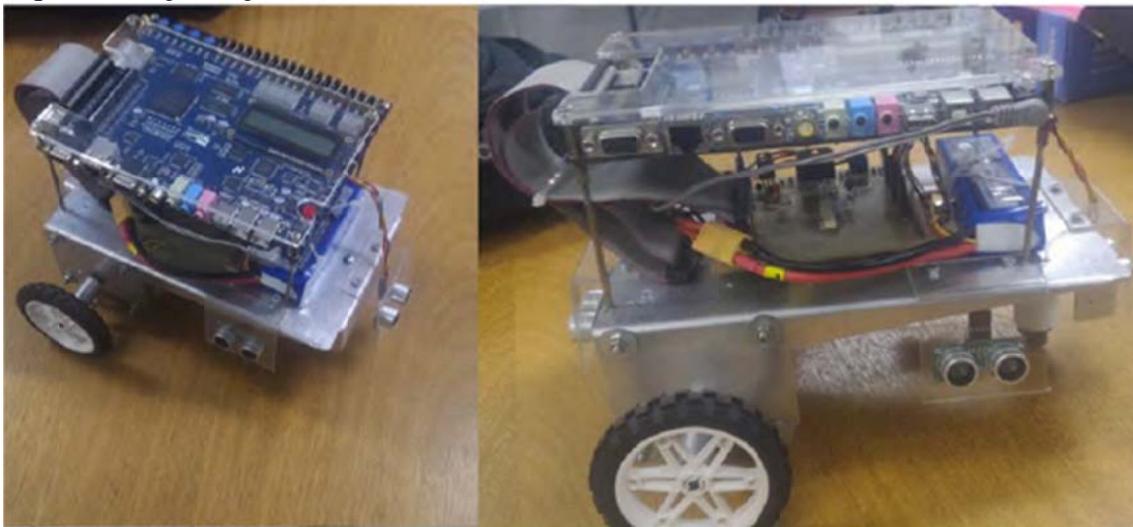


Figura 5 – O protótipo.



Neste sensor o emissor e o receptor são ligados a um circuito digital capaz de ler o tempo de atraso entre a emissão e a recepção da onda. Com este tempo de atraso e considerando a velocidade da onda ultrassônica no ambiente, determina-se a distância do objeto utilizando a cinemática básica, em que a velocidade é a variação da distância em relação ao tempo.

A medida de distância é obtida pela excitação do emissor com um trem de pulsos

Organização



Promoção

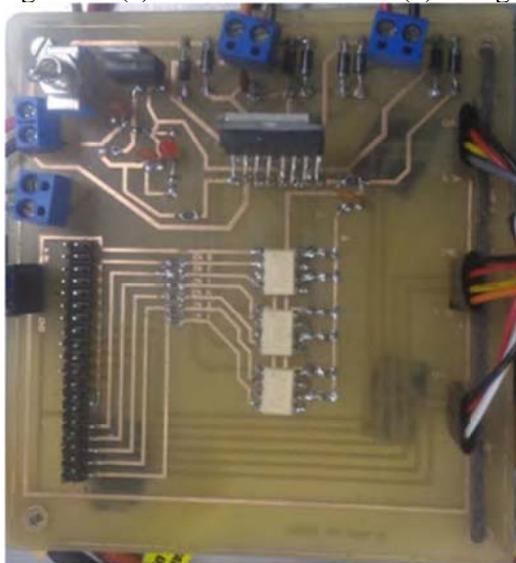




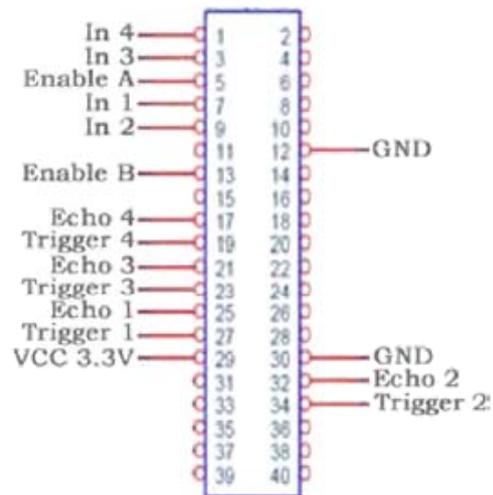
de duração de 5 a 10  $\mu$ s, geralmente espaçado por 1ms. O sensor recebe um sinal que é o *echo* do sinal emitido e transforma esse sinal em digital, gerando um pulso de duração determinado pela largura do pulso do *echo* recebido.

Uma interface entre a placa DE2 e o robô (driver dos motores) foi desenvolvida sendo apresentada na Figura 6 (a). Nesta são utilizados dispositivos SMD (resistores e opto acopladores) e um *flat cable* (padrão HD de 40 pinos), com a pinagem mostrada na Figura 6 (b), que faz a conexão entre as duas placas, interface e a DE2, por meio do seu barramento de expansão.

Figura 6– (a) Driver dos motores. (b) Pinagem de conexão.



(a)



(b)

Usando o percurso treinado para a Rede PMC foram realizados testes com robô com movimentos nos sentidos horário e anti-horário. Alguns resultados são mostrados na sequência de fotos da Figura 7.

Figura 7– Robô em movimento no sentido horário.



Organização

Promoção



## 5. CONSIDERAÇÕES FINAIS

Neste trabalho foram usadas RNAs para controlar um protótipo de um robô móvel usando como plataforma de desenvolvimento, a placa DE2 da ALTERA.

Para a realização do protótipo foram usados quatro sensores de distância do tipo ultrassônico e dois motores CC. Os sensores ultrassônicos fornecem as entradas de dados para rede, de forma que o robô execute o seu percurso de maneira autônoma, com erros menores que 1cm. Para o acionamento por PWM dos motores CC, uma placa (*driver*) foi confeccionada.

As RNAs PMC convergem para os sistemas desejados, demonstrando ser um bom método a ser utilizado quando não existe conhecimento sobre o sistema, a não ser por suas entradas e saídas, apresentando um erro quadrático médio abaixo de 1%, com resultados, em geral, muito próximos aos desejados. A dificuldade encontrada na implementação das RNAs consiste na elaboração de um banco de dados que descreva o comportamento do sistema visando o treinamento. O tempo de treinamento para a PMC apresentou uma média de 15h, variando conforme o número de amostras treinadas em computador pessoal.

Outro processo em VHDL que gera certa dificuldade é a implementação da função de ativação, neste caso foram utilizadas linearizações por partes. Este processo necessita de muitas unidades comparadoras, restringindo as implementações de camadas escondidas maiores que 10 neurônios, sem diminuir a precisão da função. RNAs muito grandes gastariam muitas unidades lógicas para utilizar um bloco de função por neurônio, sendo mais viável implementar por *lookup tables*, sem paralelismo.

Nos testes práticos com o protótipo do robô, resultados satisfatórios foram obtidos, mostrando o seguimento de um percurso em sentido horário e sem colisão com as paredes, mantendo-se próximo à trajetória esperada.

Com este projeto foi observado um grande interesse do aluno pelo desenvolvimento do trabalho e a sua motivação para buscar novos desafios.

### *Agradecimentos*

À FAPESP pelo apoio financeiro - Processo: 2015/16956-1

## REFERÊNCIAS BIBLIOGRÁFICAS

ALTERA (Org.). Development and Education Board - DE2. Disponível em: [ftp://ftp.altera.com/up/pub/Webdocs/DE2\\_UserManual.pdf](ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf). Acesso em: 4 mar. 2016.

AZHAR, M. A. H. B.; DIMOND, K. R. Design of an FPGA Based Adaptive Neural Controller For Intelligent Robot Navigation. In: Proceedings of the Euromicro Symposium on Digital System Design, 2002.

CAMILO, J. R.; MANTOVANI, S. C. A. Controle de robôs móveis usando lógica fuzzy e dispositivos lógicos programáveis. Trabalho de Conclusão de Curso. FEISUNESP. 2014.

COLMAN, D. L. Projeto de Redes Neurais Artificiais Aplicadas em Problemas de Automação Industrial. 2006.

Organização



**UDESC**  
UNIVERSIDADE  
DO ESTADO DE  
SANTA CATARINA



Promoção





E-GIZMO. Ultrasonic Sonar Module. Disponível em: <http://www.egizmo.com/KIT/images/ultrasonicsonar/ultrasonic%20sonar%20module%201r0.pdf>. Acesso em: 04 mar 2016.

HAYKIN, S. S. Redes Neurais. Princípios e prática. 2. ed. Porto Alegre: BOOKMAN COMPANHIA ED, 2001.

LACERDA, W.S.; CHAVES, R.M. Implementação em Hardware da função de ativação do neurônio artificial utilizando instrução customizada para o processador NIOS II. ENIA, 2012.

MICRORED. Microredutor Modelo P-VDC. Disponível em: <http://www.microred.com.br/page16.html>. Acesso em: 8 out. 2016.

PIÃO, S. S.; CODÁ, L. M. R. Implementação de rede Neural Artificial em FPGA utilizando VHDL. Trabalho de Conclusão de Curso. EESC-USP, 2012.

SILVA, C. A. DE A.; DORIA NETO, A. D.; DE OLIVEIRA, J. A. N.; DE MELOR, J. D. Implementação de uma rede neural artificial em FPGA: Aplicação da MLP como arquitetura modular, 2009.

## **MOBILE ROBOT X ARTIFICIAL NEURAL NETWORK CONTROL USING FPGA MOTIVATES RESEARCH WORK IN THE UNDERGRADUATE COURSE IN ELECTRICAL ENGINEERING.**

*Abstract: Aiming to prepare the undergraduate student for research in robotics and Artificial Intelligence, a work was proposed using an Artificial Neural Network for the control of a mobile robot implemented in programmable logic devices- FPGA of the Cyclone family II- board DE2- ALTERA, using a description language of Hardware, VHDL, in order to control a robot prototype developed in the laboratory. The implemented Network is one of the most basic multi layers perceptron (MLP) type, suitable for the application in robotics, whose training is performed in the MATLAB software, obtaining the necessary data on in the execution phase in hardware, without prejudice to its efficiency, when compared to the same system in software. The specified FPGA has logical elements to synthesize Neural Networks in a hardware description language, facilitating the design change, providing portability between development devices / software. From the results, we can observe the success of the project, applying the neural network PML in FPGA in the control of the robot in a defined path, and the technical growth of the student, demonstrating greater motivation for the development of Scientific Initiation work.*

**Key-words:** Artificial Neural Network, Mobile Robot, Multilayer Perceptron, FPGAs,

Organização



**UDESC**  
UNIVERSIDADE  
DO ESTADO DE  
SANTA CATARINA



Promoção

