

# Perceptron Linear Aplicado na Modelagem de Circuitos Elétricos

Manuela Bonatto Xavier da Silveira<sup>1</sup>, Sibilla Batista da Luz França<sup>1</sup>, Eduardo Gonçalves de Lima<sup>1</sup>  
<sup>1</sup>GICS, UFPR, Curitiba, Brasil  
manuelaxavier@ufpr.br

**Resumo** — O perceptron é uma unidade básica de rede neural que pode ser utilizado para modelagem matemática de circuitos elétricos. O objetivo do trabalho é realizar a modelagem em linguagem de descrição de hardware de um circuito elétrico com memória, usando um perceptron, para implementação em FPGA. O perceptron utilizado foi de regressão linear com uma função de ativação que combina sinais aplicados em sua entrada, no caso, tensões e correntes atuais e passadas. O sistema estudado foi um circuito com um resistor, um capacitor e duas fontes de tensões independentes em série, o qual foi analisado no software QUCS. No presente trabalho, o foco foi o desenvolvimento do perceptron utilizando o Método dos Mínimos Quadrados em MATLAB. Como resultado, foram obtidos os coeficientes que ajustam o melhor perceptron de duas entradas e os valores normalizados dos dados encontrados. No momento, os estudos concentram-se no desenvolvimento do código para implementação em FPGA.

## I. INTRODUÇÃO

A modelagem matemática de sistemas físicos por meio de redes neurais tem ganhado destaque nas últimas décadas [1], especialmente no campo da engenharia elétrica, onde a precisão na simulação de circuitos é essencial. Dentre os elementos dentro de redes neurais, se destaca o perceptron, exemplificado na Figura 1, que permite uma compreensão clara de como funciona uma rede neural em termos matemáticos.

Esse modelo basicamente analisa um conjunto de dados de entrada e utilizando uma função de ativação gera uma saída, posteriormente, usada para tomada de decisões com relação ao sistema estudado. Neste contexto, a utilização de perceptrons na modelagem de circuitos elétricos em hardware programável, como FPGAs (Field-Programmable Gate Arrays) [2], [3], oferece uma abordagem promissora para o desenvolvimento de sistemas digitais de alto desempenho.

O foco deste trabalho foi o desenvolvimento do perceptron (com a função de ativação linear), utilizando softwares matemáticos e de simulação de circuitos, para implementação, posteriormente, na FPGA. O Método dos Mínimos Quadrados [4] foi o método utilizado para

encontrar os coeficientes do melhor perceptron, que pode ser expresso por uma equação do estilo:

$$Y = C_1 \cdot Vs_1 + C_2 \cdot Vs_2 + C_3,$$

onde os valores de  $C_1$ ,  $C_2$  e  $C_3$  são os valores calculados,  $Vs_1$  e  $Vs_2$ , são as entradas no momento atual e passado e  $Y$  a saída do sistema ao longo do tempo.

O circuito elétrico em estudo é composto por elementos como resistores e capacitores, além de fontes de tensão independentes, cujas interações dinâmicas ao longo do tempo são simuladas e modeladas adaptadas a lógica da equação do perceptron. O uso do perceptron aplicado ao contexto de circuitos elétricos é interessante pela sua capacidade de armazenar e prever dados relacionados aos componentes ao longo do tempo. Assim, a análise se torna mais completa, com a possibilidade de garantir o melhor funcionamento e proteção dos elementos do circuito.

A abordagem adotada envolve a coleta de dados do circuito por simulação transiente, de forma tabelada e senoidal e a posterior aplicação no código descrito do modelo perceptron em questão. Este estudo busca contribuir para o avanço no campo da modelagem de circuitos com memória, além de abrir possibilidades para o uso de redes neurais em projetos de sistemas digitais reconfiguráveis.

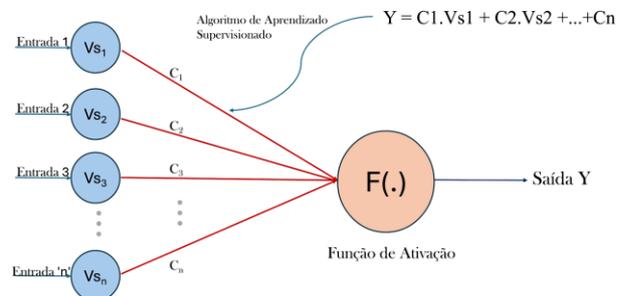


Fig 1. Modelo de funcionamento do perceptron

## II. DESENVOLVIMENTO

### A. Codificação em MATLAB

Utilizando o MATLAB (Matrix Laboratory), software de alta performance para cálculo numérico e manipulação de matrizes, foi codificado o Método dos Mínimos Quadrados, de forma a facilitar todo o processo; sendo necessário modificar apenas os valores das matrizes que representam os sinais de entrada na aplicação prática.

Para inicializar o código, foi aplicado 'clear', 'close all' e 'clc' para limpar todas as variáveis e o histórico para evitar mistura de informações. Em seguida, foram definidos dois vetores linha (X e Y) de tamanho 1x26 com os valores hipotéticos que corresponderão as tensões de fonte e capacitor do circuito. Esses vetores serão as entradas do perceptron, e utilizados da maneira transposta como vetores coluna 26x1 (.')

Monta-se uma matriz de regressão (XX) de tamanho 25x3, tendo como primeira coluna o vetor X (vetor de entradas), percorrido do segundo valor ao último. Essa coluna será multiplicada pelo primeiro coeficiente (C1) do perceptron. A segunda coluna, a ser multiplicada pelo segundo coeficiente (C2), é formada, também, pelo vetor X, porém percorrido do primeiro valor ao penúltimo. A terceira e última coluna da matriz é formada de um vetor de 1's, com tamanho 25x1 (ones(25,1)) que irá ser usada para multiplicar o terceiro coeficiente do perceptron (C3).

Para definir a matriz de coeficientes, pelo Método dos Mínimos Quadrados, faz-se uma divisão entre a matriz de regressão montada e o vetor de entradas Y. Assim, encontra-se o valor dos três coeficientes. Para a determinação dos valores de saída do perceptron utilizando o MSE, utiliza-se XX multiplicada pelo vetor dos coeficientes, o que equivale à substituição no modelo do perceptron.

Por fim, foi adicionada ao código, a plotagem do gráfico correspondente ao melhor perceptron em relação às entradas e o cálculo dos coeficientes.

### B. Aquisição de Dados

Para coleta de dados experimentais, foi feita a simulação de um circuito elétrico no software QUCS. O circuito simulado, mostrado na Figura 2, contém um resistor ( $R1 = 50 \Omega$ ), um capacitor ( $C = 1/100\pi \text{ F}$ ;  $V_c(0) = 10 \text{ V}$ ) e duas fontes de tensão independentes: uma AC ( $V1 = 12\sin(2\pi 10^9 t)$ ; frequência de 1 GHz e amplitude de 12 V) e uma DC ( $V2 = 10 \text{ V}$ ).

Utilizando a simulação transiente (tipo de simulação que estuda o comportamento de um sistema ao longo do tempo, levando em consideração as variações que ocorrem) foi possível analisar as tensões de entrada e sobre o capacitor ao longo do tempo em forma tabelada. A simulação foi realizada no intervalo de 9 a 10 ns (com  $\Delta t =$

40 ps) e obtendo 26 amostras de tensões (com precisão de 5 casas decimais).

A coleta dos dados das tensões em transiente é essencial para a aplicação no perceptron, pois é a partir da simulação que a análise com memória se torna possível.

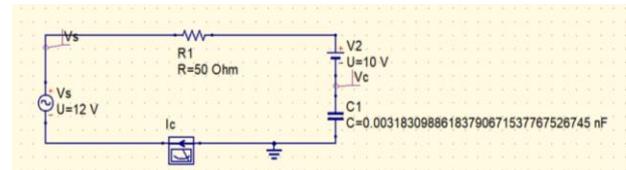


Figura 2: Circuito simulado no QUCS

### C. Aplicação dos dados no Método dos Quadrados

Colocando os dados obtidos (com 5 casas decimais) no código do MATLAB, feito anteriormente, obtemos os coeficientes do melhor perceptron, utilizando o método da Subseção II.A, onde Y são as tensões sobre o capacitor ( $V_c$ ) e X, as tensões na fonte ao longo do tempo ( $V_s$ ), com a matriz XX associando os valores de tensões passadas e atuais da fonte senoidal. Lembrando que é ignorada a primeira linha de valores de tensão na primeira coluna, por não termos o valor da tensão anterior da fonte a ser comparado. A plotagem do gráfico das tensões do capacitor por tensões de saída foi feita logo em seguida, dentro do próprio MATLAB.

## III. RESULTADOS

Como resultado da simulação feita na Subseção II.B, foi obtido um gráfico da relação entre a tensão da fonte senoidal e a tensão sobre o capacitor; também foi obtida uma tabela com a relação de valores dessas tensões. A figura 3 mostra esses dados. Aplicando os dados obtidos pela simulação no código do MATLAB apresentado nas Subseções II.A e II.C, foi possível gerar o gráfico (Figura 3), e identificar os 3 coeficientes do melhor perceptron. Ao analisar o gráfico obtido é possível verificar que a modelagem se mostrou extremamente precisa, pela sobreposição quase perfeita entre os pontos medidos e calculados do perceptron.

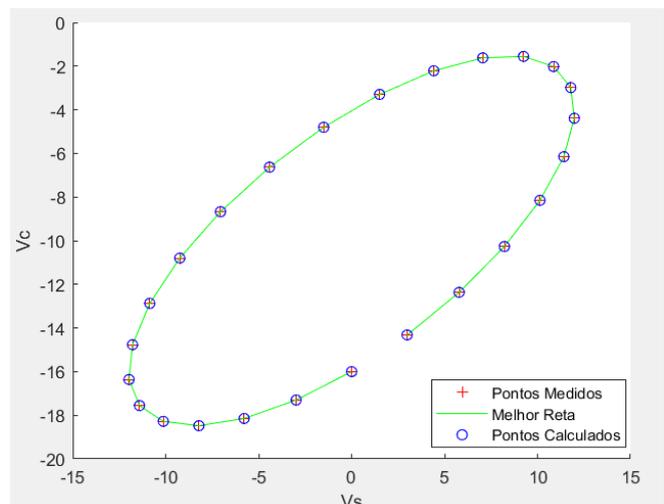


Figura 3: Gráfico do melhor perceptron (Pontos calculados sobrepondo os pontos medidos)

Após o processo, obtivemos os seguintes valores para os coeficientes:  $C1 = 1.4475$ ,  $C2 = 2.0106$ ,  $C3 = -10.0002.0$

Em seguida, foi feita a normalização dos valores obtidos dos coeficientes e das tensões de entrada e capacitor para minimizar a quantidade de recursos lógicos necessários para implementação do circuito em FPGA.

Analisando os valores, temos que o menor é -18,4802, o maior valor 11,9763 e os valores com maior quantidade de decimais têm 5 casas decimais.

Para normalizar os valores, nesse caso multiplicamos os valores por 10.000, assim os que contém 5 casas decimais se tornaram inteiros. Pensando em sistema base, usamos  $2^{14}$ , valor mais próximo de 10.000, obtendo resolução de  $1/2^{14}$ .

#### IV. CONCLUSÕES

O desenvolvimento deste projeto permitiu a implementação de um perceptron linear aplicado à modelagem de um circuito elétrico com memória. Através da simulação do circuito no software QUCS e da aplicação do Método dos Mínimos Quadrados no MATLAB para determinar os coeficientes do perceptron, foi possível reproduzir o comportamento dinâmico do circuito, demonstrando a eficácia dessa abordagem.

Com a aplicação dos dados no código, foi obtida a equação que melhor representa o perceptron, cumprindo a primeira parte do objetivo deste trabalho. Os próximos passos envolvem a descrição do código em hardware, que será realizada utilizando a ferramenta ISE Design Suite.

O código VHDL a ser desenvolvido incluirá as operações lógicas de multiplicação e soma dos valores obtidos pelo perceptron, conforme detalhado na introdução. A implementação será feita na FPGA Xilinx Virtex 5 LX50T, englobando não apenas a programação lógica, mas também a validação e otimização do design para garantir que o perceptron opere de forma eficiente, assim como foi verificado matematicamente durante o estudo.

#### AGRADECIMENTOS

Agradecemos Luis Miliavacca por ajudar com o uso das funcionalidades do MATLAB. Os autores agradecem o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) no âmbito do Programa PIBITI CNPq UFPR.

#### REFERÊNCIAS

- [1] HAYKIN, S. Neural Networks and Learning Machines, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2009.
- [2] PEDRONI, Volnei A. Circuit design with VHDL, 3rd ed. MIT press, 2020.
- [3] GASINO, V.M.R. Descrição em linguagem de hardware de um modelo linear sem memória, Curitiba, 2023.
- [4] D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, 6th ed. Hoboken, NJ, USA: Wiley, 2014.