

APRENDIZAGEM BASEADA EM PROJETOS: UM CONTROLADOR NEBULOSO DESENVOLVIDO NA PLATAFORMA ARDUÍNO

ERICK ALESSANDRO GONÇALVES DOS SANTOS^{1*},
FELIPE CASSEMIRO ULRICHSEN², RICARDO MELOS³, NIVAL NUNES DE ALMEIDA⁴

¹Engenheiro Eletricista, UERJ, Rio de Janeiro-RJ. Fone: (21) 983-278-233, erick.alessandro@gmail.com

² Engenheiro Eletricista, UERJ, Rio de Janeiro-RJ. Fone: (21) 988-340-316, felipeulrichsen@uerj.br

³ Engenheiro Eletricista, UERJ, Rio de Janeiro-RJ. Fone: (51) 8331-3042, eng.ricardomelos@gmail.com

⁴ Dr. Prof. Eng. Elétrica, UERJ, Rio de Janeiro-RJ. Fone: (21) 2334-2165, nivalnunes@yahoo.com.br

Apresentado no
Congresso Técnico Científico da Engenharia e da Agronomia – CONTECC' 2015
15 a 18 de setembro de 2015 - Fortaleza-CE, Brasil

RESUMO: Este trabalho apresenta uma experiência de aprendizagem ativa a partir do desenvolvimento de um controlador nebuloso, que foi inicialmente simulado no ambiente MATLAB e, posteriormente, processado na plataforma Arduino. Dois projetos foram desenvolvidos por alunos sem conhecimento de lógica fuzzy e puderam aperfeiçoar seus estudos na área de controle e automação industrial de maneira complementar ao estudado em outras disciplinas. Os resultados alcançados legitimaram o aprendizado por estes estudantes nos projetos desenvolvidos, especialmente motivando-os a carreira de engenharia, onde puderam realizar atividades bem próximas daquelas encontradas na área de atuação profissional.

PALAVRAS-CHAVE: Aprendizagem Ativa, Controle Nebuloso, Controlador PID, Arduino, MatLab.

PROJECT-BASED LEARNING: A FUZZY CONTROLLER DEVELOPED IN THE ARDUINO PLATFORM

ABSTRACT: This paper presents an active learning experience from the development of a fuzzy controller, which was initially simulated in MATLAB and subsequently processed in the Arduino platform. Two projects were developed by students without knowledge of fuzzy logic and could improve his studies in the control area and industrial automation complementary way to study in other disciplines. The results achieved legitimized the learned by these students in developing projects, especially motivating them to engineering career, where they could perform well next activities of those found in the professional area.

KEYWORDS: Active Learning, Fuzzy, PID, Arduino, MatLab.

INTRODUÇÃO

A disciplina de Controle de Processos por Computador (CPC) é ministrada no 10º período do curso de Engenharia Elétrica - ênfase em Sistemas e Computação da Universidade do Estado do Rio de Janeiro (UERJ). Nesta etapa o estudante já cursou as disciplinas de Análise de Sistemas Físicos (ASF) e Controle e Servomecanismos (CeS), disciplinas da Teoria de Controle. O conhecimento é adquirido, normalmente, em aulas teóricas com simulações computacionais e experimentos laboratoriais clássicos, ou seja, os estudantes comprovam o conteúdo apresentado. No entanto, verifica-se que os estudantes normalmente compreendem os modelos e as teorias de controle, mas ainda estão distantes das concepções práticas usuais em engenharia. Para diminuir esta lacuna, é possível fazer uso das modernas técnicas de *software* e *hardware* a fim de promover aplicações em atividades similares às encontradas na realidade profissional de forma que os mesmos aprendam e participem ativamente de um projeto.

Neste sentido, a cadeira de CPC, como faz parte de seu programa o estudo de instrumentação e controle e automação industrial, tem apresentado como proposta de aprendizado um projeto de um controle de um sistema/processo. Para este propósito, estão envolvidos os conteúdos tradicionais das disciplinas de Teoria de Controle, Eletrônica Analógica e Digital, Instrumentação e Controle, Linguagens de Programação e

Princípios de Comunicações. Além disso, para controlar o sistemas/processo industrial, é necessário utilizar dispositivos tais como: sensores, atuadores, sistemas de controle, sistemas de supervisão e aquisição de dados, e fazer uso dos conhecimentos da engenharia elétrica e de computação.

MATERIAL E MÉTODOS

Para a construção de sistemas de controle são ainda utilizados dispositivos como micro-controladores, controladores lógicos programáveis (CLPs) e sistemas microprocessados, dentre outros. Estes dispositivos possuem em comum, entradas e saídas (portas), que servem para realizar a comunicação com os dispositivos periféricos (sensores e atuadores). Estas portas também podem ser destinadas a realizar a comunicação com outros sistemas, a fim de fornecer dados de leitura dos sensores ou até receber instruções externas para os atuadores.

Um exemplo de um dispositivo bem útil, de baixo custo de aquisição e de boa flexibilidade para aplicações, é o *Arduino*, que vem a ser uma plataforma de prototipagem eletrônica de *hardware* livre, projetada com um microcontrolador *Atmel AVR* e suporte de entrada/saída embutido, com uma linguagem de programação padrão, a qual tem origem em *Wiring*, e é essencialmente *C/C++*. Para projeto apresentado neste trabalho, foi utilizado o *ArduinoAT Mega 2560*, onde na Figura 1 apresenta-se um exemplo de seu formato e suas características.

Figura 1 – Placa de prototipagem Arduino AT Mega 2560



Características Básicas	ATmega2560
Tamanho:	5,3cm x 10,2cm x 1,0cm
Tensão de operação:	5V
Tensão de entrada (recomendada):	7-12V
Tensão de entrada (limites):	6-20V
Pinos de entrada/saída (I/O) digitais:	54 (dos quais 14 podem ser saídas PWM)
Pinos de entrada analógicas:	16
Corrente DC por pino I/O:	40mA
Corrente DC para pino de 3,3V:	50mA
Memória Flash:	256KB (dos quais, 8KB são usados pelo <i>bootloader</i>)
SRAM:	8KB
EEPROM:	4KB
Velocidade de Clock:	16MHz

Para um sistema de controle automático, é necessário comparar uma variável do processo, o que se deseja controlar (valor real da grandeza de saída do processo), e com outra variável (uma grandeza de referência ou valor desejado para variável de saída), cujo cálculo da diferença entre as duas determina-se o **erro** (desvio). A partir deste **erro**, produz-se um sinal de controle que busca reduzir o mesmo a um valor nulo (ou a um valor pequeno), isto é, tal que **variável de saída** seja próxima, ou igual, ao **valor desejado**. Assim sendo, os controladores industriais podem ser qualificados de acordo com a sua ação de controle, podendo ter as seguintes classificações:

1. **Controlador liga-desliga**, é aquele cuja saída do controlador a ação muda de *ligada* para *desligada*, ou vice-versa, à medida que o sinal do erro passa pelo ponto de ajuste. Ele possui uma oscilação contínua da variável controlada em torno do **setpoint** (valor desejado ou valor referência). Um bom exemplo é o controle de temperatura de uma geladeira ou um ar condicionado;

2. **Controlador Proporcional**: ele atua de modo que ação de controle seja proporcional ao erro, isto é, à diferença entre o valor desejado e o valor atual da variável controlada;

3. **Controlador Integral**: a principal função da ação integral é fazer com que o processo siga com erro nulo. Contudo, a ação integral aplicada isoladamente tende a piorar a estabilidade relativa do sistema, por isto é geralmente utilizada em conjunto com a ação proporcional;

4. **Controlador Proporcional e Integral**: é a combinação do controle que atua proporcionalmente ao erro, sendo este erro posteriormente anulado pela ação integral. Tem como característica uma demora na correção do valor da variável controlada;

5. **Controlador Proporcional e Derivativo**: é a combinação do controle que atua proporcionalmente ao erro com velocidade antecipativa. Tem como característica a sobre-elevação (*overshooting*) sobre o valor desejado, que pode até ocasionar danos ao sistema controlado se for muito elevada;

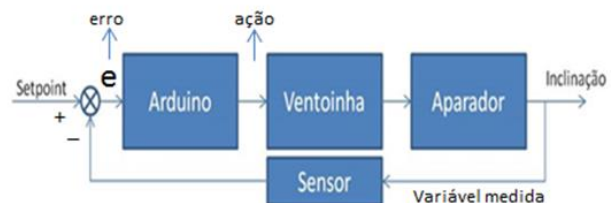
6. **Controlador Proporcional, Integral e Derivativo**: é uma forma de controle, que junta as ações: derivativa, integral e proporcional, fazendo assim com que o sinal de erro seja minimizado pela

ação proporcional, tenda a zero pela ação integral e que a correção seja obtido com uma velocidade antecipativa pela ação derivativa.

Para fins de um aprendizado baseado em problemas, foi demandado um controlador PID clássico, para controlar um processo de livre escolha pelos estudantes. Deste modo, estes poderiam ficar motivados a propor um problema a ser controlado que garantisse um aprendizado a partir do desenvolvimento de um projeto. Assim, seria possível fazer uso dos conhecimentos teóricos básicos, realizar as simulações e os testes laboratoriais, agregando-se demandas de projetos para confirmar os resultados alcançados, onde se desenvolveria um protótipo bem próximo de uma aplicação real.

Neste caso, primeiramente, foi desenvolvido um controlador que atuou a sobre a inclinação de uma aleta (ou aparador) através de um fluxo de ar de uma ventoinha (Figura 2). O controlador aciona a ventoinha que desloca o aparador. Um sensor de proximidade mede a posição atual da aleta (variável medida), o controlador avaliava o erro encontrado (a diferença entre o *setpoint* e a posição da aleta) e ajustava o fluxo de ar (ação) de modo posicioná-la no ponto desejado, de maneira estável, de modo rápido e com a menor oscilação possível.

Figura 2: O controlador PID para regular a Aleta e o respectivo em diagrama de blocos do sistema



Analisando os resultados e a modelagem do controlador simulado no programa *MATLAB*, foi construído o processo: aleta e sua estrutura, e posteriormente foi realizada a programação do *Arduino* para fins de se implementar um controlador PID. No ajuste dos ganhos, foi utilizado o método *Ziegler-Nichols*. Este método visa determinar cada ganho do controlador (constantes proporcional, integral e derivativa) através da observação dos períodos da dinâmica do sistema, uma vez que não havia uma formulação matemática que representasse o processo (função de transferência). Vale registrar que exceto a placa *Arduino*, o material utilizado no projeto foi adquirido de sucata que foi reciclado, podendo-se visualizar o experimento (processo e algoritmo de controle) no sítio:

<https://www.youtube.com/watch?v=G9gA6ZX67Ck>

A partir desse projeto com o controlador PID, foi usada a mesma metodologia para o desenvolvimento de um controlador *Fuzzy*. Neste caso, foi construído um controlador que atuava sobre a distância que um barco ficaria de um aparador, que pode também ser visualizado no sítio:

<https://www.youtube.com/watch?v=g5anFsp0WHc>

No que concerne aos controladores nebulosos (difusos) pode-se afirmar que eles são baseados na lógica *fuzzy* e são caracterizados por não serem lineares. A lógica *fuzzy* tem como objetivo estabelecer os fundamentos para efetuar um raciocínio aproximado a partir de proposições imprecisas que são descritas numa língua natural e ainda pode ser compreendida como uma extensão da lógica proposicional, que apenas possui dois valores: verdadeiro e falso.

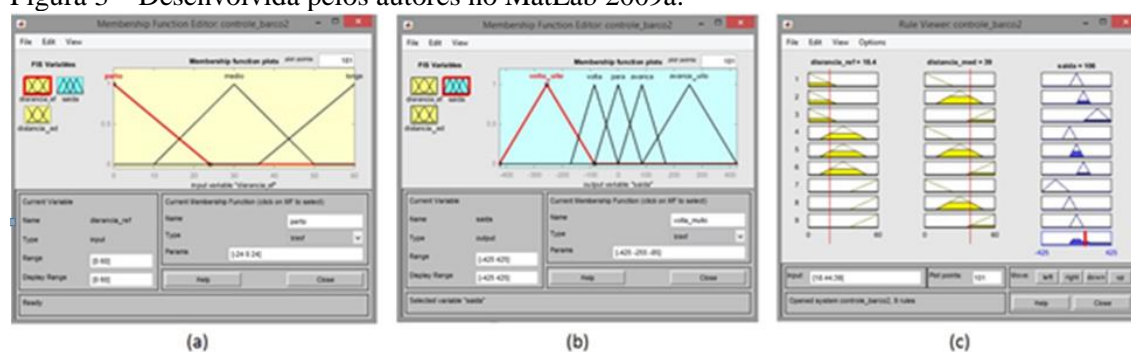
Assim como um controlador convencional, o nebuloso transforma variáveis de entradas em variáveis de saída, que atuam num processo ou num sistema. Além disso, os valores de entrada são os valores reais medidos na forma de sinais e o processamento computacional num controlador deste tipo é feito em três etapas:

1. **Fuzzificação:** é modelagem matemática da informação através de variáveis de entrada por meio de conjuntos nebulosos. Nesta etapa se faz uso das funções de pertinência, que são modelos matemáticos para termos linguísticos, podendo ser funções com os seguintes formatos: triangular, função trapezoidal ou função gaussiana.
2. **Inferência:** é onde se definem quais são os conectivos lógicos usados para estabelecer a relação *fuzzy* que modela a base de regras. A base de regras é um dos componentes mais importantes de um sistema, pois as regras são expressas como implicações lógicas do modo *se...então*;
3. **Defuzzificação:** que traduz o estado da variável de saída *fuzzy* para um valor numérico (escalar).

É importante ressaltar que existem diferentes métodos de inferência *fuzzy* e com diferentes propriedades. Neste projeto foi utilizado o **método de Mamdani** vislumbrando sua implementação no *Arduino* e por estar disponível no *Fuzzy Logic toolbox* do *MATLAB* 2009, na Figura 3 apresentamos a simulação de nosso controle. Esse método tem como base a regra de composição de inferência Máx-Mín, onde ocorrerá a *defuzzificação* dos valores de entrada *fuzzy* (Figura 3a) através da função de saída (Figura 3b) utilizando o método do centro de área (Figura 3c). A saída deste processo pode ser a união lógica de duas ou mais funções de pertinência definidas no discurso da variável de saída (Figura 3c).

Para programação do *Arduino* foram usados dois vetores que armazenaram os resultados das funções de pertinência da distância definida e distância medida. Em seguida foi feito o *and* das regras e por fim armazenado o resultado em uma matriz três por três. Logo depois temos as regras de inferência realizando a *defuzzificação* que é feita pelo centroide da área. Neste ponto temos uma aproximação, os trapézios foram aproximados para triângulos para diminuir a complexidade dos cálculos por se tratar de um sistema de tempo real.

Figura 3 – Desenvolvida pelos autores no MatLab 2009a.



Legenda:

a) Curvas de pertinência da variável de entrada

b) Curvas de pertinência da variável saída

c) Aplicação do método mamdani para cálculo da saída

CONCLUSÕES

Os resultados alcançados legitimaram o aprendizado por estes estudantes nos projetos desenvolvidos. Embora os projetos fossem simples a primeira vista, foram encontradas várias dificuldades que aos poucos foram superadas. Essas questões puderam ser contornadas nas simulações computacionais, mas em projetos reais os estudantes verificaram que não eram tão simples assim. Contudo, conseguiram dar andamento ao projeto de forma profissional, devido, especialmente, ao fato de estarem motivados a resolver problemas que levam em conta aspectos dos encontrados na carreira de engenharia, podendo assim, realizar atividades bem próximas daquelas encontradas na área de atuação profissional.

REFERÊNCIAS

- Ogata, Katsuhiko. Engenharia de Controle Moderno. 3ª ed. Rio de Janeiro: LTC, 2000.
- Ross, T.J. *Fuzzy Logic with Engineering Applications*. 2ª ed., London, 2004.
- Silveira, M. A.; Parise, J. A. R.; Campos, R. C.; Carmo, L. C. S.; Almeida, N. N. Project-Based Learning (PBL) Experiences in Brazil. In: Xiangyun Du; Erik de Graaff; Anette Kolmos. (Eds.) Research on PBL Practice in Engineering Education. Rotterdam: Sense Publishers, 2009, Chapter 13, p. 155-168.

AGRADECIMENTOS

Os autores agradecem ao Professor Orlando Bernardo Filho pelas sugestões e ajuda no desenvolvimento do trabalho e ao graduando em engenharia Helder Nery Ferreira por sua contribuição no desenvolvimento do projeto de controle da aleta.