

IMPLEMENTAÇÃO DE CONTROLADORES PI, FUZZY E NEURAL PARA CONTROLE DE UM MOTOR DE CORRENTE CONTÍNUA

THIAGO DE M. S. MOREIRA, ALEXANDRE S. BRANDÃO

Núcleo de Especialização em Robótica, Departamento de Engenharia Elétrica, Universidade Federal de Viçosa

Avenida Peter Henry Rolfs s/n, Campus Universitário, Viçosa, MG, Caixa Postal 36570-900

E-mails: thiago.moreira@ufv.br, alexandre.brandao@ufv.br

Resumo— Este trabalho apresenta o desenvolvimento de controladores para o controle de um motor de corrente contínua, uma vez que tais motores ainda são utilizados em muitos processos na indústria e em outras aplicações, como a robótica. Inicialmente, é proposto um modelo de primeira ordem com atraso de transporte para o motor, o qual é utilizado para sintonizar um controlador PI a partir do método da curva de reação de Ziegler-Nichols. Também são criadas uma lógica fuzzy e rede neural artificial. Por fim, os controladores são avaliados em malha fechada com o motor.

Palavras-chave— Controlador PI, lógica fuzzy, rede neural artificial, motor de corrente contínua.

1 Introdução

Em 2014 foi proposto pelos autores a construção, instrumentação e controle de um manipulador robótico planar de dois graus de liberdade. No caso, o foco do trabalho foi a construção do manipulador, e para cada um dos dois motores de corrente contínua (cc) foi feito o controle de posição a partir de um controlador proporcional. Entretanto, como expressado nas conclusões, seria interessante a otimização do controle de posição, assim como a implementação de um controle de velocidade (Moreira, 2014).

Desta forma, o presente trabalho tem como motivação o estudo e desenvolvimento de controladores PI, fuzzy e neural para posterior implementação no manipulador robótico, e também servir como referência para outros processos. A ideia é desenvolver um controle de velocidade para um motor cc genérico, mas que pode ser facilmente adaptado para qualquer motor cc específico.

A Figura 1 esquematiza a composição interna de um motor cc. Ao se aplicar uma corrente constante no circuito de campo, é gerado um campo magnético também constante. O circuito de armadura quando posicionado ortogonalmente a esse campo, é submetido a uma força magnética ao se aplicar uma tensão contínua em seus terminais. Assim, o torque resultante gira o rotor. Em função do enrolamento de armadura mover-se ortogonalmente ao campo magnético criado pelo circuito de campo, tem-se que a tensão em seus terminais é proporcional à velocidade angular com que gira (Nise, 2012).

Portanto, com o auxílio da Tabela 1, a partir das equações elétricas e mecânicas do motor cc pode-se demonstrar que a velocidade angular Ω se relaciona com a tensão de armadura V_a por:

$$G(S) = \frac{\Omega(S)}{V_a(S)} = \frac{K_m}{XS^2 + YS + Z} \quad (1)$$

Onde:

$$X = L_a J$$

$$Y = L_a b + R_a J$$

$$Z = R_a b + K_m K_b$$

Tabela 1. Identificação dos parâmetros do motor cc.

R_a	Resistência da armadura
L_a	Indutância da armadura
K_m	Constante de torque do motor
J	Momento de inércia da carga
b	Atrito viscoso da carga
K_b	Constante de força contra eletromotriz

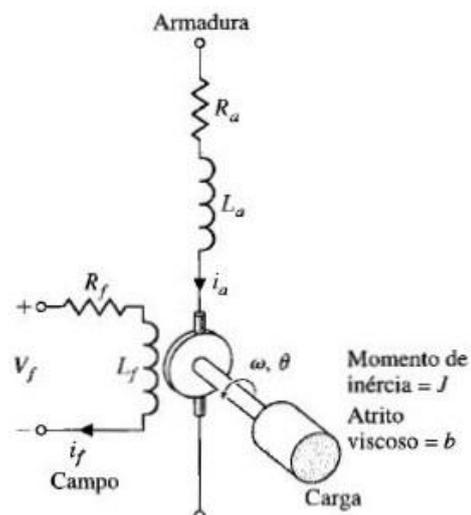


Figura 1. Esquemático do motor cc (Dorf e Bishop, 2001).

1.1 Identificação de Sistemas

Segundo Astrom e Hagglund (1995), a dinâmica de um processo pode ser determinada pela sua resposta a pulsos, degraus, rampas e outros sinais determinísticos.

Considerando a resposta ao degrau, uma aproximação para o processo é possível de ser obtida por meio do modelo de três parâmetros, como expresso na Equação 2. Ele é caracterizado pelo ganho estático K, a constante de tempo T e o tempo morto (ou atraso de transporte) L.

$$G(S) = \frac{K}{TS + 1} e^{-LS} \quad (2)$$

A Figura 2 exemplifica a determinação gráfica do modelo de três parâmetros para uma entrada degrau unitário. Observa-se que a constante de tempo pode ser calculada como B-A ou C-A, originando duas variações para o modelo. Normalmente, a definição do parâmetro T pelo ponto B origina uma melhor aproximação (Astrom e Hagglund, 1995).

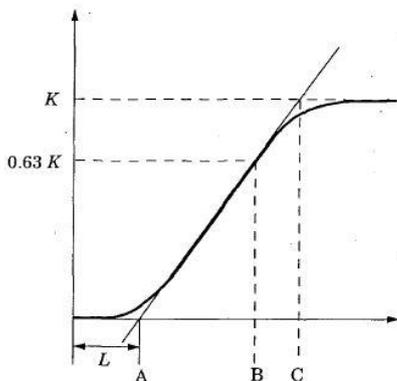


Figura 2. Determinação gráfica do modelo de três parâmetros (Astrom e Hagglund, 1995).

Assim, através do modelo de três parâmetros representado pela Equação 2, o método da curva de reação de Ziegler e Nichols (1942) propõe uma forma direta para o cálculo dos parâmetros de controladores P, PI e PID, como sintetizado na Tabela 2.

Tabela 2. Determinação dos parâmetros de controladores P, PI e PID pelo método da curva de reação de Ziegler-Nichols.

Controlador	Kp	Ti	Td
P	T/L	∞	0
PI	0,9T/L	L/0,3	0
PID	1,2T/L	2L	0,5L

1.2 Lógica Fuzzy

A lógica fuzzy foi inicialmente proposta por Zadeh (1965), com a intenção de eliminar a necessidade de descrever os fenômenos através de números

exatos, passando a descrevê-los por meio de termos qualitativos, assim como o ser humano faz na tomada de decisões.

Desta forma, é possível estabelecer um formalismo matemático para palavras como “frio”, “morno” e “quente”, de modo que os sistemas computacionais também possam compreender esses termos vagos. Isso é feito incluindo os valores entre 0 (falso) e 1 (verdadeiro) em sua lógica, distribuindo o conceito falso/verdadeiro ao longo do intervalo.

Considere, como exemplo, a variável linguística “velocidade” em um intervalo de 0 a 100 rad/s, denominado universo de discurso. Pode-se associar a essa variável conjuntos fuzzy (ou valores linguísticos), como “baixa”, “média” e “alta”, distribuídos por funções de pertinência, que assumem valores reais entre 0 e 1. De acordo com Nguyen et al. (2003), as funções de pertinência mais comumente utilizadas na teoria de controle são a trapezoidal, triangular, gaussiana, sigmoideal e função-S. A Figura 3 mostra essas definições para uma função de pertinência trapezoidal.

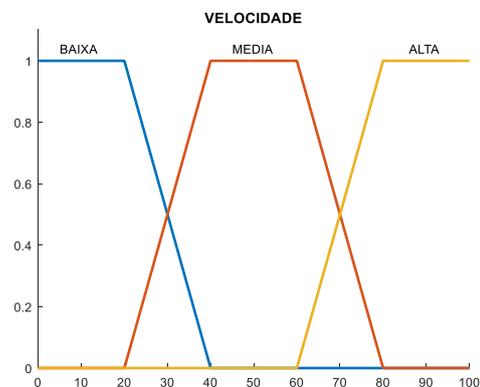


Figura 3. Exemplo de variável linguística.

Logo, um sistema de inferência fuzzy possibilita que computadores compreendam o pensamento humano a partir de um conjunto de regras escritas em linguagem natural. O mecanismo de inferência fuzzy mais conhecido é o sistema de inferência Mamdani, proposto por Mamdani e Assilian (1975) para o controle de uma caldeira de vapor. Ele supõe que as variáveis linguísticas de entrada e saída sejam definidas por conjuntos fuzzy.

Os passos necessários para a aplicação do sistema de inferência Mamdani, como representados na Figura 4, são:

1. Determinar as variáveis linguísticas e conjuntos fuzzy.
2. Especificar o grau de pertinência a algum conjunto fuzzy para cada valor numérico de entrada. Este processo é denominado fuzzificação.

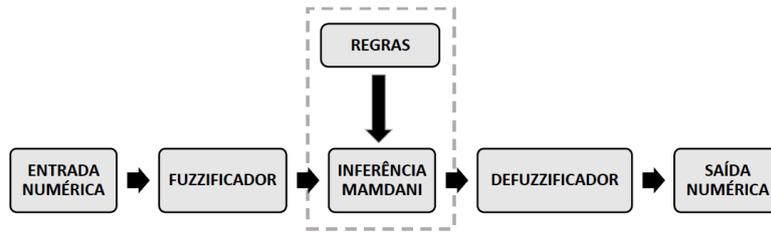


Figura 4. Aplicação do sistema de inferência Mamdani.

3. Determinar um conjunto de regras do tipo “se-então” para relacionar os conjuntos fuzzy de entrada e saída. Exemplo: “se x é A, então y é B”.
4. Processar os resultados de cada regra em um conjunto fuzzy de saída agregado.
5. Analisar a área do conjunto fuzzy de saída agregado, e entregar um valor numérico de saída. Este processo é denominado defuzzificação.

Segundo Nguyen et al. (2003), o método de defuzzificação mais amplamente utilizado é o centroide. Entretanto, existem diversos outros métodos, como o bisetor, menor do máximo, média do máximo e maior do máximo.

2 Metodologia

É importante mencionar que todas as simulações são desenvolvidas nos ambientes MATLAB e Simulink.

São considerados valores didáticos para os parâmetros do motor cc, como indicados na Tabela 3. Entretanto, a mesma análise pode ser feita para qualquer modelo específico de motor, uma vez que seus parâmetros sejam conhecidos.

Tabela 3. Parâmetros do motor cc.

Ra	2 Ω
La	0,1 H
Km	0,2 N.m/A
J	0,01 kg/m ²
b	0,01 N.m/rad/s
Kb	0,2 V.s/rad

Através da Equação 1 e rearranjando os termos, calcula-se a sua função de transferência:

$$G(S) = \frac{200}{S^2 + 21S + 60} \quad (3)$$

É possível determinar um modelo de primeira ordem com atraso de transporte que se aproxime da Equação 3, mas que ainda descreva bem o processo. Desta forma, são propostos os dois modelos com três

parâmetros citados na Seção 1.1, sendo ilustrados nas Figuras 5 e 6, e comparados com o processo real na Figura 7.

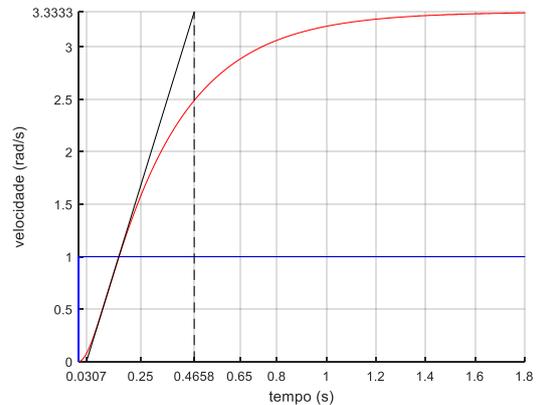


Figura 5. Modelo C-A.

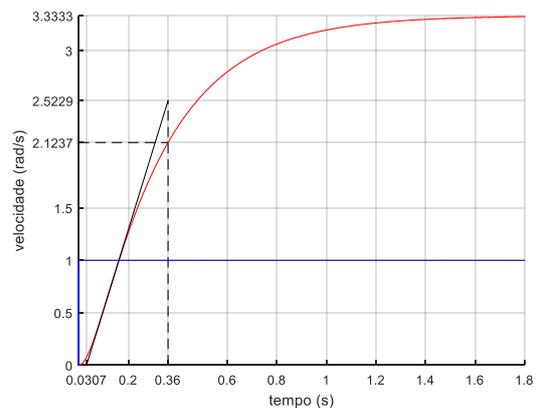


Figura 6. Modelo B-A.

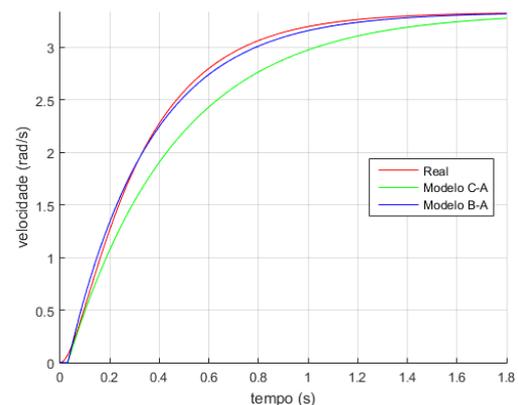


Figura 7. Comparação dos modelos obtidos com o processo real.

Pode-se observar que o Modelo B-A obtido apresenta um melhor resultado, exibindo um erro relativamente pequeno durante todo o percurso da curva em relação ao modelo de segunda ordem indicado na Equação 3.

Logo, a função de transferência a ser considerada nas simulações é:

$$G(s) = \frac{3,3333}{0,3293s + 1} e^{-0,0307s} \quad (4)$$

Uma vez definida a função de transferência simplificada, através da Tabela 2 determina-se uma aproximação inicial para os parâmetros do controlador PI a partir do método da curva de reação de Ziegler-Nichols. Após a sintonia fina tem-se os parâmetros indicados na Tabela 4. Por sua vez, a Figura 8 mostra o bloco desenvolvido do controlador PI. As entradas 1 e 2 são, respectivamente, o setpoint e a variável controlada.

Tabela 4. Parâmetros do controlador PI.

Kp	1,20
Ti	0,33

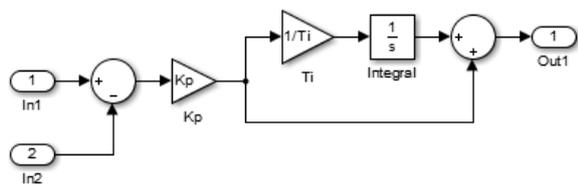


Figura 8. Bloco do controlador PI.

Já a lógica fuzzy é desenvolvida com o auxílio da ferramenta “Fuzzy Logic Toolbox” do MATLAB. Ela é proposta a partir do sistema de inferência Mamdani e defuzzificação por centroide. A Figura 9 ilustra as variáveis linguísticas, com seus respectivos conjuntos fuzzy, funções de pertinência e universo de discurso. As regras são indicadas na Tabela 5.

Assim, o bloco desenvolvido do controlador fuzzy é mostrado na Figura 10. É possível observar as duas variáveis de entrada no controlador, as quais fornecem as informações do erro da variável controlada em relação ao setpoint desejado, bem como a derivada do erro, indicando se o mesmo está aumentando, diminuindo ou constante com o tempo. A ação integral, presente na saída do controlador, remove o erro em regime permanente.

Tabela 5. Regras da lógica fuzzy.

erro / derro	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

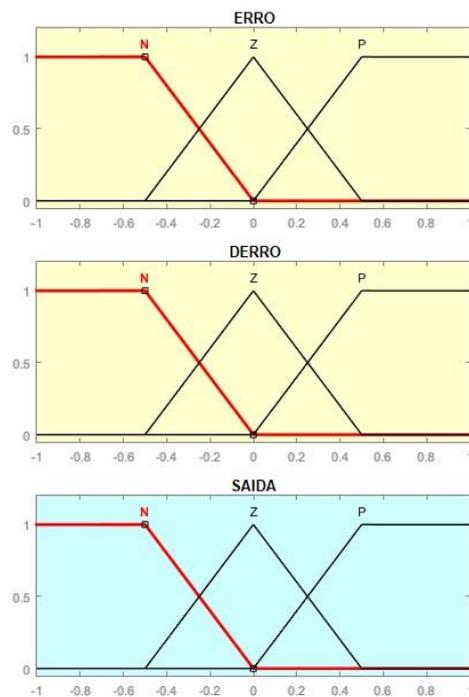


Figura 9. Variáveis linguísticas de entrada “erro” e “derro”, e variável linguística de saída “saida” utilizadas na lógica fuzzy.

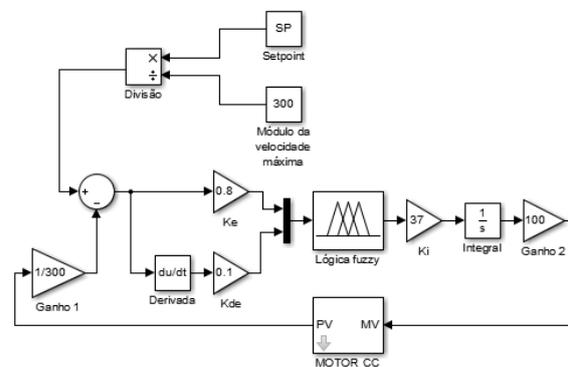


Figura 10. Bloco do controlador fuzzy.

Ainda na Figura 10 vê-se que são empregados alguns ganhos. Eles são necessários para adequar o controlador ao processo. Como ilustrado na Figura 9, o universo de discurso das entradas e saída é $U = [-1,1]$, mas para o processo em questão é considerado um range da velocidade (setpoint) entre -300 e 300 rad/s. Desta forma, os ganhos são utilizados para garantir que valores entre -1 e 1 cheguem ao controlador, e em contrapartida os níveis desejados pelo setpoint possam ser atingidos pelo processo. Como pode ser visto, a variável controlada sempre é dividida por 300 antes de ser enviada ao bloco somador. De forma similar, o setpoint sempre é dividido por 300, garantindo um valor dentro do universo de discurso estipulado na lógica fuzzy.

Por fim, para o treinamento da rede neural é necessário que existam informações prévias acerca de como o controlador deve se comportar para controlar o processo.

Pode-se pensar que o controlador PI é mais amplamente empregado na indústria pela sua grande eficiência e facilidade de implementação. Portanto, é proposto neste trabalho que o mesmo seja utilizado como referência, fornecendo os dados para a criação da rede neural.

Logo, são feitos ensaios com o controlador PI submetido a entradas aleatórias, mas em três ranges distintos, para a coleta de amostras do erro, variável controlada e sinal de controle. O erro e a variável controlada devem ser as entradas da rede neural, enquanto o sinal de controle, a saída. O processo de amostragem está resumido na Tabela 6, ao passo que a Figura 11 exemplifica um dos ensaios.

Tabela 6. Resumo do processo de amostragem do controle PI.

Ensaio	Range (rad/s)	Nº de amostras
1	-1 a 1	20003
2	-5 a 5	20003
3	-300 a 300	20003
Total		60009

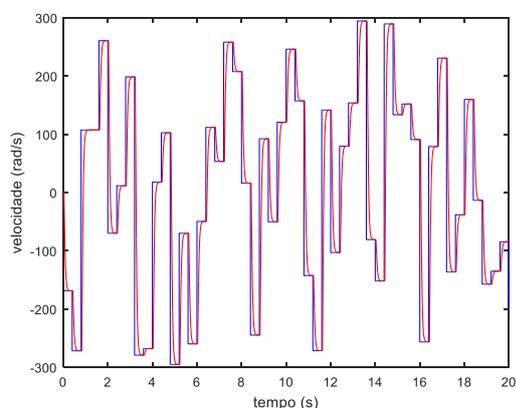


Figura 11. Ensaio 3 para amostragem do controle PI.

Uma vez de posse das amostras, é possível realizar o treinamento da RNA. Para isso, é empregada a ferramenta “Neural Fitting app” do MATLAB. A Tabela 7 sintetiza o treinamento efetuado. Já a Figura 12 apresenta o bloco desenvolvido do controlador neural. De forma análoga ao controle PI, as entradas 1 e 2 são, respectivamente, o setpoint e a variável controlada.

Tabela 7. Resumo do processo de treinamento da RNA.

	Nº de neurônios	Função de ativação
Camada oculta	8	Sigmoide logarítmica
Camada de saída	1	Linear
Algoritmo de treinamento	Levenberg-Marquardt	

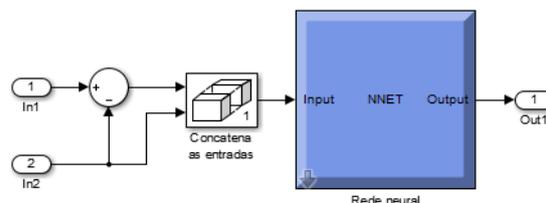


Figura 12. Bloco do controlador neural.

3 Resultados

Para o teste dos controladores elaborados são propostas duas entradas distintas, a degrau unitário e uma entrada aleatória com range de 0 a 300 rad/s. Isso pode ser visto nas Figuras 13 e 14.

O objetivo definido para o trabalho é a implementação de controladores (sintonia do controlador PI, e desenvolvimento da lógica fuzzy e rede neural) para o controle do motor cc especificado. Não são estabelecidos critérios minuciosos para a comparação da performance de cada controlador. De uma forma geral busca-se que o processo em malha fechada não apresente sobressinal e atinja o regime permanente no menor tempo possível. Neste caso, nota-se um tempo de subida maior com o controle fuzzy, mas sem sobressinal em todos.

Como esperado, o controle neural mostra uma resposta praticamente idêntica ao PI, uma vez que o seu treinamento se baseia no comportamento do processo com o controlador PI. É importante destacar que a rede neural possui duas entradas, pois o resultado não é satisfatório quando somente o erro é fornecido à rede.

Desta forma, a elaboração do controlador neural está condicionada ao conhecimento prévio do sistema, e sua performance está diretamente relacionada com a quantidade de dados que se tem do processo. Como mostrado na Tabela 6, são utilizadas 60009 amostras para o treinamento da rede. Através das Figuras 13 e 14 vê-se que o controlador neural apresenta um bom desempenho para pequenos e grandes degraus, dentro do range de velocidade estipulado para o processo.

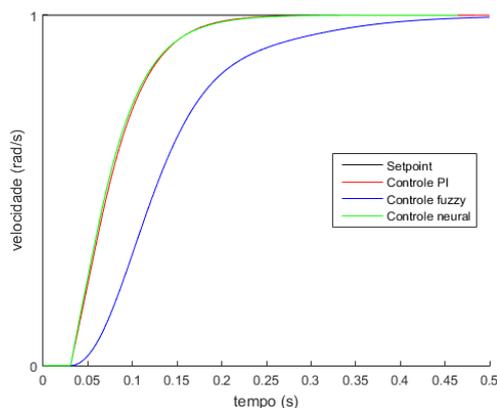


Figura 13. Respostas dos controladores a um degrau unitário.

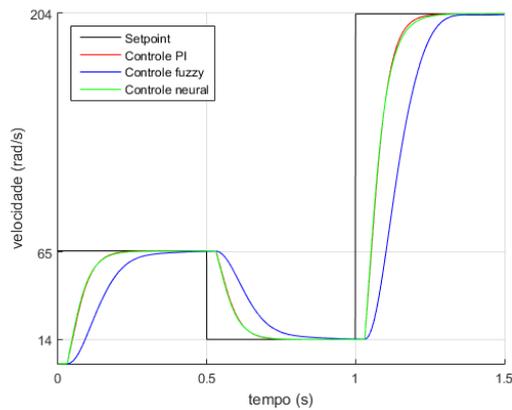


Figura 14. Respostas dos controladores a uma entrada aleatória com range de 0 a 300 rad/s.

4 Conclusão

Como visto, foram desenvolvidos e testados controladores PI, fuzzy e neural para o controle de um motor de corrente contínua.

Dentro da proposta do sistema em malha fechada apresentar uma resposta rápida e sem sobressinal, o controle fuzzy exibiu um tempo maior para entrar em regime permanente, mas sem sobressinal, bem como os outros dois controles.

De uma forma geral, caso se conheça o processo e o modelo que o descreva, então é interessante que se utilize tal modelo para calcular os parâmetros de um controlador PI, que nesta situação é uma solução robusta e relativamente simples. Entretanto, caso não se conheça o processo ou ele seja complicado demais para descrevê-lo com um modelo matemático, então o controlador fuzzy pode ser útil, uma vez que seu funcionamento se assemelha à forma como o ser humano raciocina.

Referências Bibliográficas

- Astrom, K. and Haggglund, T. (1995). PID Controllers: Theory, Design, and Tuning. 2nd ed. Research Triangle Park, NC: Instrument Society of America.
- Dorf, R. and Bishop, R. (2001). Sistemas de Controle Modernos. 8. ed. Rio de Janeiro: LTC.
- Mamdani, E. and Assilian S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1-13.
- Moreira, T. Construção, Instrumentação e Controle de um Manipulador Robótico Planar de Dois Graus de Liberdade. Universidade Federal de Viçosa, 2014.
- Nguyen, H., Prasad, N., Walker, C. and Walker, E. (2003). A First Course in Fuzzy and Neural Control. Boca Raton, Florida: Chapman & Hall/CRC.

Nise, N. (2012). Engenharia de Sistemas de Controle. 6. ed. Rio de Janeiro: LTC.

Zadeh, L. (1965). Fuzzy Sets. Information and Control, vol. 8, no. 3, pp. 338-353.

Ziegler, J. and Nichols, N. (1942). Optimum Settings for Automatic Controllers. Trans. ASME, vol. 64, pp. 759-768.