

# Controlador neural para um motor CC: aspectos de projeto e simulação

Gustavo Arins

gustavoarins1612@gmail.com

Universidade Federal de Santa Catarina - Campus Joinville  
Brasil, Joinville, SC

Benjamin Grando Moreira

benjamin.grando@ufsc.br

Universidade Federal de Santa Catarina - Campus Joinville  
Brasil, Joinville, SC

## ABSTRACT

This work presents the development of a neural controller to control the speed of a CC motor, applying as a priority the response time, the overshoot percentage, and the required precision of the system. We developed a simulation of two alternatives for control: a PID controller and a controller using Artificial Neural Networks. To select the network architecture quantitative and qualitative analyzes were carried out to interpret the phenomena and the behavior of each proposed network architecture.

## KEYWORDS

Artificial Neural Networks, Control, PID Controller, Neural Controller

## 1 INTRODUÇÃO

Atualmente, os sistemas de controle automáticos são utilizados em diversos setores da indústria tais como: o controle de qualidade, linha de montagem, sistema de transportes, sistemas de eletrônica de potência, tecnologia espacial e robótica [4]. Com o avanço da tecnologia, expandiu-se o conhecimento do controle de processos de forma que a evolução foi inevitável, de técnicas simples como o clássico controlador Proporcional-Integral-Derivativo (PID), para outras mais refinadas como o Long Range Predictive Control (LRPC) [8].

Tecnologias vêm sendo desenvolvidas para aprimorar os métodos de controle já implementados na engenharia e obter desempenhos satisfatórios para aplicações que exigem precisão e tempo de resposta satisfatórios. Dentre esses trabalhos, [6] propõe estudo utilizando Redes Neurais Artificiais (RNA) e controle PID aplicados no controle de altitude de um quadricóptero, e [3] expõe o desenvolvimento de um controlador PID baseado em lógica Fuzzy para atuação em um pressurizador de um reator nuclear de água pressurizada.

As estratégias de controle com autoajuste de parâmetros são adequadas para sistemas com características que variam no tempo [10]. Sistemas de controle que utilizam *deep learning* são capazes de realizar autoajustes e, atualmente, tem sido aplicados no controle de processos não-lineares complexos [2], devido a capacidade de trabalho em relação à precisão e confiabilidade em sistemas não-lineares e invariantes no tempo.

Essa tecnologia de controle, utiliza os conceitos de Inteligência Artificial, com o emprego de RNA e surgem com a promessa de resolver problemas encontrados no controle de processos [7]. A característica não linear, a capacidade de empregar múltiplas variáveis de entrada e a habilidade de aprender por meio de dados de entrada e saída, torna atrativo o aproveitamento pelos profissionais

da área de controle, de modo que a rede aprenda o comportamento do sistema e seja capaz de reproduzi-lo de maneira precisa.

A escolha do método de controle baseado em RNA, denominado de Controlador Neural (CN), se mostra tolerante às mudanças dos parâmetros do sistema, tal como a variação do momento de inércia de um motor [10]. Essa técnica de controle, apresenta adequados tempos de resposta e acurácia em regime permanente, além disso, não há necessidade de realizar recalibrações na ocorrência de mudanças dos parâmetros do sistema, pois o próprio sistema já treinado, corrige os erros propagados.

Considerando a contextualização abordada, este trabalho utilizou a *deep learning* no controle da velocidade de motores de corrente contínua (CC), contando como prioridade o tempo de resposta, o percentual de sobressinal e a precisão requerida ao sistema. Dessa maneira, utilizando técnicas de aprendizado profundo de máquina, uma RNA foi modelada com o objetivo de mimetizar o comportamento de um controlador PID e, assim, controlar a velocidade do motor CC. Portanto, fez-se uma simulação dessas duas de alternativas para controle: o controlador PID e o CN. Realizou-se então, uma análise qualitativa, buscando a interpretação dos fenômenos e o comportamento de cada método empregado.

Este trabalho está organizado da seguinte maneira: a segunda seção apresenta elementos da fundamentação teórica, explicando as técnicas de controle utilizadas para a simulação desenvolvida; na terceira seção são apresentados os materiais e métodos aplicados para obtenção dos resultados; a simulação proposta é mostrada na seção 4; na seção 5 são apresentadas as arquiteturas de redes avaliadas; na seção 6 é apresentado como o resultado da rede neural artificial é aplicado na simulação; a seção 7 apresenta os resultados, com a definição de métricas de avaliação utilizadas; a conclusão do trabalho está disponível na seção 8.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este trabalho apresenta o desenvolvimento de uma simulação que utiliza dois tipos de controladores: um controlador PID e um Controlador Neural (controlador constituído de uma Rede Neural Artificial). Esses dois conceitos são sintetizados a seguir.

### 2.1 Controlador PID

O controlador PID ainda é o método de controle mais utilizado na indústria e a permanência desse controlador nos setores industriais, justifica-se pelo fácil manuseio nos ajustes de seus parâmetros, ou seja, o ganho proporcional ( $K_p$ ), o ganho integral ( $K_i$ ) e o ganho derivativo ( $K_d$ ). Na maioria das vezes, tais ajustes são feitos pelo método de tentativa e erro, não sendo necessário conhecimento técnico e teórico sobre a aplicação [9].

Além disso, a performance do controlador PID depende das próprias características físicas e construtivas do sistema a ser controlado e, geralmente, esse processo é considerado como um sistema linear, entretanto, os sistemas reais apresentam características não lineares. Por essa razão, é possível que haja alteração das propriedades anteriormente calculadas ao longo do tempo de funcionamento do processo, resultando, na diferenciação de cálculos e ineficiência ao sistema [1].

Um exemplo desse fato ocorre, segundo [10], em uma válvula motorizada utilizada na aplicação de polímeros, na qual o atrito da válvula aumenta se não estiver operando, por um determinado período, devido ao depósito de partículas entre as áreas de assentamento, resultando na mudança do momento de inércia do rotor. Se nesta aplicação for utilizado um controlador PID, o sistema de controle se torna incapaz de fornecer a saída adequada, pois os parâmetros do controlador ajustados correspondem ao sistema sem qualquer atrito. Nesse caso, o controlador PID pode produzir *overshoots*, ou seja, um sinal que excede seu alvo, pois o desempenho do controlador se deteriora devido a alterações nos parâmetros do sistema.

## 2.2 Redes Neurais Artificiais

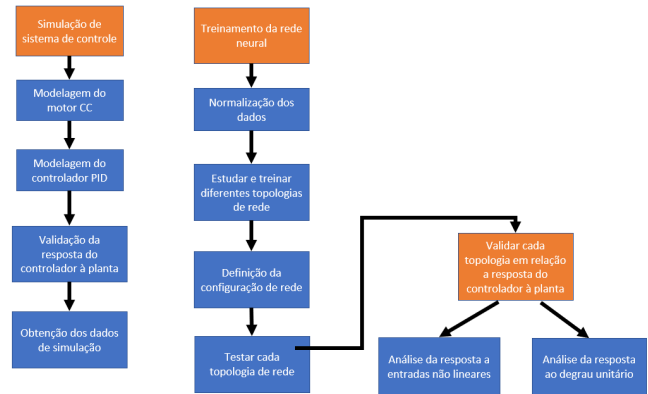
A capacidade de processamento de informações do cérebro humano tem motivado pesquisas que visam encontrar modelos que reproduzam suas características computacionais. Possibilitando, desta forma, que se realizem tarefas de maneira semelhante ao cérebro humano, garantindo maior controle e velocidade de processos. Seguindo esse raciocínio, as redes neurais artificiais (RNA) são modelos computacionais inspirados no sistema nervoso dos seres vivos. Possuem capacidade de aquisição de conhecimento, baseado em informações e são definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões [11].

A estrutura das redes de neurônios artificiais foi modelada em relação aos modelos conhecidos de sistemas nervosos biológicos e particularmente, do cérebro humano [11]. Assim sendo, o neurônio de uma Rede Neural Artificial (RNA) opera de maneira similar ao funcionamento de um neurônio biológico.

O neurônio artificial é constituído de sete elementos básicos [5]:

- (1) Sinais de entrada ( $x_1, x_2, \dots, x_n$ ): são os sinais advindos do meio externo e que representam os valores assumidos pelas variáveis de uma aplicação. São usualmente normalizados com o objetivo de incrementar a eficiência computacional dos algoritmos de aprendizagem;
- (2) Pesos sinápticos ( $w_1, w_2, \dots, w_n$ ): valores responsáveis por ponderar cada um dos sinais de entrada, permitindo quantificar as relevâncias de cada entrada em relação à funcionalidade do respectivo neurônio;
- (3) Somatório ( $\Sigma$ ): tem como função, agregar os sinais de entrada já ponderados para produzir um valor potencial de ativação;
- (4) Biais  $b$ : especifica qual será o nível apropriado para que o resultado obtido pelo somatório possa gerar um valor de disparo em direção a saída do neurônio;
- (5) Potencial de ativação  $v$ : resultado produzido pela diferença entre o somatório e o *bias*;

Figura 1: Etapas do desenvolvimento do projeto



- (6) Função de ativação  $\varphi$ : tem como função restringir a amplitude da saída de um neurônio. A função de ativação é também referida como função restritiva já que limita o intervalo permissível de amplitude do sinal de saída a um valor finito;
- (7) Sinal de saída  $y$ : valor produzido pelo neurônio em relação a um determinado conjunto de sinais de entrada.

Por meio das respectivas especificações de cada parâmetro, pode-se definir que uma rede neural recebe um vetor de entrada ( $x = x_1, x_2, \dots, x_n$ ) e realiza uma soma ponderada com pesos ( $w = w_1, w_2, \dots, w_n$ ) dessas entradas [5]. Ao final, ao resultado dessa soma é adicionado um *bias*, passando pela função de ativação  $\varphi$ , resultando por fim, em uma saída  $y$ .

Portanto, em termos matemáticos, um neurônio pode ser descrito pelas Equações 1 e 2:

$$u_k = \sum_{j=1}^n w_{kj}x_j \quad (1)$$

$$y_k = \varphi(u_k + b_k) \quad (2)$$

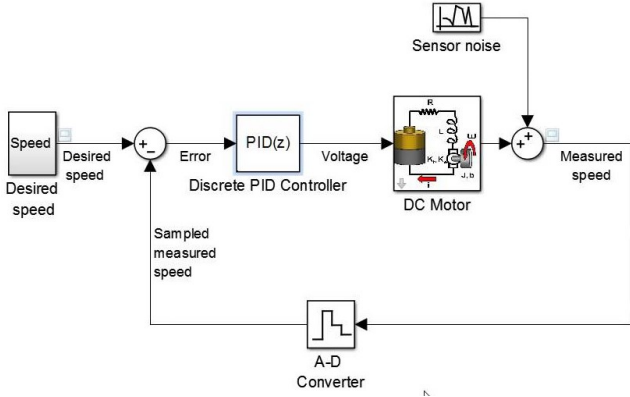
onde o vetor  $x$  ( $x_1, x_2, \dots, x_n$ ) são os sinais de entrada, o vetor  $w$  ( $w_1, w_2, \dots, w_n$ ) são os pesos sinápticos do neurônio  $k$  e o vetor  $y$  ( $y_1, y_2, \dots, y_n$ ). A saída do somatório ou combinador linear é determinado por  $u_k$ ,  $b_k$  representa o *bias* e  $\varphi$  é a função de ativação [5].

## 3 MATERIAIS E MÉTODOS

A metodologia deste trabalho utiliza dados provenientes de simulação e é uma pesquisa de teor exploratório. São gerados dados de velocidade de um motor CC, bem com sinais de tensão propagados por meio de um controlador PID modelado para o sistema. Com os dados obtidos, é possível treinar uma rede neural que gere um comportamento similar ao apresentado pelo controlador PID utilizado na simulação. Por fim, é realizada a comparação dos dois controladores via software de simulação. A Figura 1 ilustra as etapas de projeto.

A primeira etapa do projeto consiste na simulação do sistema de controle com o objetivo de obtenção dos dados para o treinamento

Figura 2: Sistema de controle do motor CC no Simulink



da RNA. Nesta etapa, é necessário modelar o motor CC para então definir o controlador PID que atenda aos critérios de controle para o sistema. Após a definição do controlador, é necessário validá-lo em relação aos requisitos de desempenho de controle e por fim, obter os dados de simulação.

Com os dados obtidos, inicia-se a etapa de treinamento da RNA, sendo nesta etapa realizada a normalização dos dados obtidos por meio da simulação. Para encontrar a configuração de rede, são testadas diferentes topologias com o objetivo de avaliar o comportamento de cada uma e assim definir as configurações de rede que apresentam melhores resultados em relação as etapas de treinamento, teste e validação da rede.

Por último, são avaliados cada modelo treinado na simulação do sistema de controle, com o objetivo de identificar a melhor configuração, bem como, analisar o comportamento de cada rede. Para isso, deve-se primeiramente validar a rede em função da resposta do sistema e então, analisar a resposta a entradas não lineares.

#### 4 SIMULAÇÃO DO SISTEMA

A primeira etapa do projeto consiste em simular o sistema de controle do motor CC para obtenção dos dados. Para isso, é necessário a utilização de software de simulação que seja possível modelar o motor CC, dimensionar o controlador PID e simular a resposta do sistema em função de entradas lineares e não lineares. A simulação do sistema foi realizada utilizando os softwares Matlab e Simulink.

A seguir são apresentados detalhamentos sobre a simulação e parâmetros do dimensionamento do controlador PID, bem como aspectos relacionados à obtenção de dados.

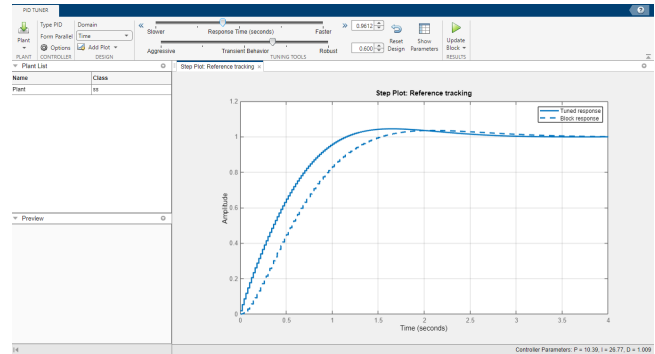
##### 4.1 Sistema de controle de motor CC

Para simular o controle de velocidade de um motor CC, utilizou-se a toolbox do Simulink denominada *PID Controller Design for a DC Motor* desenvolvida por [12], conforme demonstra a Figura 2. Com essa ferramenta, o usuário passa os parâmetros de um motor CC, como o coeficiente de amortecimento, momento de inércia, resistência e indutância. Cabe também ao usuário determinar os parâmetros  $K_p$ ,  $K_d$  e  $K_i$  controlador PID que são representados por blocos interligados.

Tabela 1: Parâmetros do motor CC simulado

J	Kt	b	L	R	Ke
0,01	0,01	0,1	0,5	1	0,01

Figura 3: Ferramenta de ajuste de parâmetros do controlador PID



O motor CC é modelado como um bloco onde um nível de tensão  $V$  (volts) e a velocidade angular  $\omega$  (rad/s) são, respectivamente, a entrada e a saída do sistema. As entradas são processadas em saídas utilizando as funções de transferência representadas nas Equações 3 e 4, denominadas de equações dinâmicas de um motor CC.

$$\frac{d\omega}{dt} = \frac{1}{J}(Kt \cdot i_a - b \cdot \omega) \quad (3)$$

$$\frac{di_a}{dt} = \frac{1}{L}(V - R \cdot i_a - Ke \cdot \omega) \quad (4)$$

onde  $J$  é o momento de inércia do rotor ( $\text{kg} \cdot \text{m}^2$ ),  $Kt$  é a constante de torque do motor ( $\text{N} \cdot \text{m} \cdot \text{rad/A}$ ),  $i_a$  é a corrente de armadura (A),  $b$  é o coeficiente de atrito viscoso do motor ( $\text{kg} \cdot \text{m}^2/\text{s}$ ),  $L$  é a indutância elétrica (H),  $R$  é a resistência elétrica ( $\Omega$ ) e  $V$  é a constante de força eletromotriz ( $\text{V} \cdot \text{s/rad}$ ) [2].

O valor de cada parâmetro de dimensionamento do motor CC são apresentados na Tabela 1.

##### 4.2 Dimensionamento do controlador PID

Utilizando uma ferramenta do Simulink chamada PID Tuner App, foram ajustados os parâmetros do controlador PID para controlar o motor CC através das funções de transferência representadas pelas Equações 3 e 4 e da passagem de requisitos de controle realimentado de velocidade e precisão desejados para a resposta. A Figura 3 demonstra a interface da ferramenta com o objetivo de apresentar a resposta em função do comportamento transitente e tempo de reposta. Buscou-se um meio termo entre velocidade de resposta e robustez, ajustando os parâmetros do controlador PID para os valores apresentados na Tabela 2.

A resposta do controlador atende aos requisitos de desempenho de controle, conforme demonstra a Tabela 3. As colunas  $T_s$ , PSS e  $V_f$  representam o tempo de acomodação, percentual de sobressinal

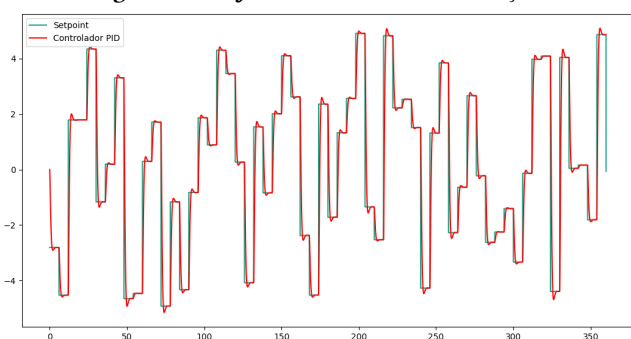
Tabela 2: Parâmetros de controle PID

$K_p$	$K_i$	$K_d$	N
5,61	17,25	0,00	0,00

Tabela 3: Requisitos de desempenho da resposta do sistema ao degrau unitário

Ts	Ts obtido	PSS	PSS obtido	Vf	Vf obtido
< 5%	3,38%	< 3s	2,75s	1 rad/s	1 rad/s

Figura 4: Conjunto de dados da simulação



e o valor final, respectivamente. Pode-se notar que o desempenho do controlador PID atende aos requisitos de controle estipulados.

### 4.3 Obtenção de dados

A obtenção de dados é realizada por meio da simulação do sistema de controle do motor CC, obtendo os valores de entradas e saídas para realizar o treinamento da RNA. Esse conjunto de dados, consiste nos valores de saída da planta, ou seja a velocidade do motor, os valores de velocidade desejados, bem como, os sinais de tensão obtidos pela saída do controlador PID.

São obtidos os valores de simulação necessários para o treinamento do controlador neural: a velocidade desejada, a velocidade do motor CC e saída o controlador PID. Os valores de velocidade do motor são representados em rad/s, contudo, os valores de saída do controlador representam a tensão necessária para alimentar o motor em volts. O tempo de amostragem do sistema consiste em 0,02 segundos.

Foram simuladas diversas situações de uso do sistema: aumentos e diminuição de velocidades mais severos, bem como mais suaves. Dessa forma, para garantir essas situações, busca-se tratar as velocidades desejadas de maneira estocástica, de forma que as velocidades sejam alteradas ciclicamente. A Figura 4 demonstra o conjunto de dados de entrada da rede neural utilizado neste estudo.

Tabela 4: Arquiteturas de RNA propostas

Rede	Neurônios	Funções de ativação
1	[100, 100, 100, 100, 100]	[S, R, R, R, R]
2	[100, 100, 100, 100, 100]	[R, S, S, S, S]
3	[10, 20, 50, 100, 200]	[S, R, R, R, R]
4	[10, 20, 50, 100, 200]	[R, S, S, S, S]
5	[200, 100, 50, 20, 10]	[S, R, R, R, R]
6	[200, 100, 50, 20, 10]	[R, S, S, S, S]
7	[100, 100, 100]	[S, S, R]
8	[100, 100, 100]	[R, R, S]
9	[20, 50, 100]	[S, S, R]
10	[20, 50, 100]	[R, R, S]
11	[100, 50, 20]	[S, S, R]
12	[100, 50, 20]	[R, R, S]
13	[100, 100, 100]	[S, S, S]
14	[20, 50, 100]	[S, S, S]
15	[100, 50, 20]	[S, S, S]
16	[100, 100, 100]	[R, R, R]
17	[20, 50, 100]	[R, R, R]
18	[100, 50, 20]	[R, R, R]

## 5 TREINAMENTO DA REDE NEURAL ARTIFICIAL

Com o conjunto de parâmetros de controle definidos e os dados de entrada e saída da rede neural já obtidos, é possível então realizar a modelagem da RNA. Foram avaliadas diversas configurações de redes com o objetivo de encontrar uma topologia com melhor resultado para o controle.

### 5.1 Configurações de rede

São destacadas 18 configurações de rede distintas com o objetivo de avaliar o resultado obtido em cada uma, considerando as etapas de treinamento, testes e validação do modelo. Todas as redes são redes Perceptron Multi-Camadas (MLP - do inglês Multi Layer Perceptron) e, como função de ativação, foram considerados o uso das funções ReLU e Sigmoide. Além disso, todas as redes utilizaram o otimizador Adam.

A Tabela 4 apresenta as configurações utilizadas neste estudo, sendo na coluna "Funções de ativação" indicado com a letra R a função ReLU e a função Sigmoide indicada com a letra S. Utilizando a rede 9 como exemplo, ela possui 3 camadas: a primeira camada constituída de 20 neurônios utilizando função de ativação sigmoide; a segunda camada possui 50 neurônios, também com a função de ativação sigmoide; e a terceira camada possui 100 neurônios com o uso da função de ativação ReLU.

O número de neurônios e camadas escondidas foi definido inicialmente de forma experimental, objetivando alcançar um desempenho considerado suficiente. Depois foram propostas quantidades de neurônios derivadas da rede obtida experimentalmente, dando origem as redes indicadas na Tabela 4.

Os dados obtidos pela simulação foram exportados para treinamento de cada RNA. Para treinamento e teste offline é utilizada a plataforma Tensorflow<sup>1</sup>.

### 6 SIMULAÇÃO DO CONTROLADOR NEURAL

Como forma de entender o comportamento de cada configuração de RNA treinada, é realizada a simulação do sistema de controle do motor CC. Utilizou-se o Matlab com a mesma toolbox do controlador PID, contudo, adicionou-se uma ferramenta responsável por prever as respostas de controle por meio de RNA treinada.

O Simulink possui a função denominada *predict*, que fornece a previsão de respostas para os dados na entrada usando a rede treinada especificada. A Figura 5 apresenta o sistema comparativo entre os dois controladores: PID e neural.

Na parte superior do sistema<sup>2</sup> mantém-se o sistema de controle PID utilizado para a obtenção de dados. Na parte inferior, o bloco de função *predict*, é responsável por prever as saídas por meio das entradas: velocidade desejada e velocidade do motor CC.

Como na etapa de treinamento da RNA utilizou-se dados normalizados, na etapa de simulação deve-se normalizar os dados de entrada da RNA, com o objetivo de atingir os mesmos efeitos. Contudo, a saída gerada pelo controlador neural é normalizada, portanto, deve-se realizar o processo inverso: a conversão dos dados normalizados em dados reais de controle.

### 7 ANÁLISE DAS ARQUITETURAS PARA O CONTROLADOR NEURAL

Após realizar o treinamento de cada rede, deve-se analisar os resultados relacionados às métricas de cada rede. A Tabela 5 apresenta os resultados de cada topologia de rede em relação ao Erro Quadrático Médio (MSE) e ao Erro Médio Absoluto (MAE).

As redes que possuem domínio de função de ativação ReLU, apresentam valores de métricas menores em relação às redes com predominância de funções sigmoid. Portanto, pode-se concluir que as funções de ativação ReLU, diminuem de maneira acentuada tanto o MSE e o MAE.

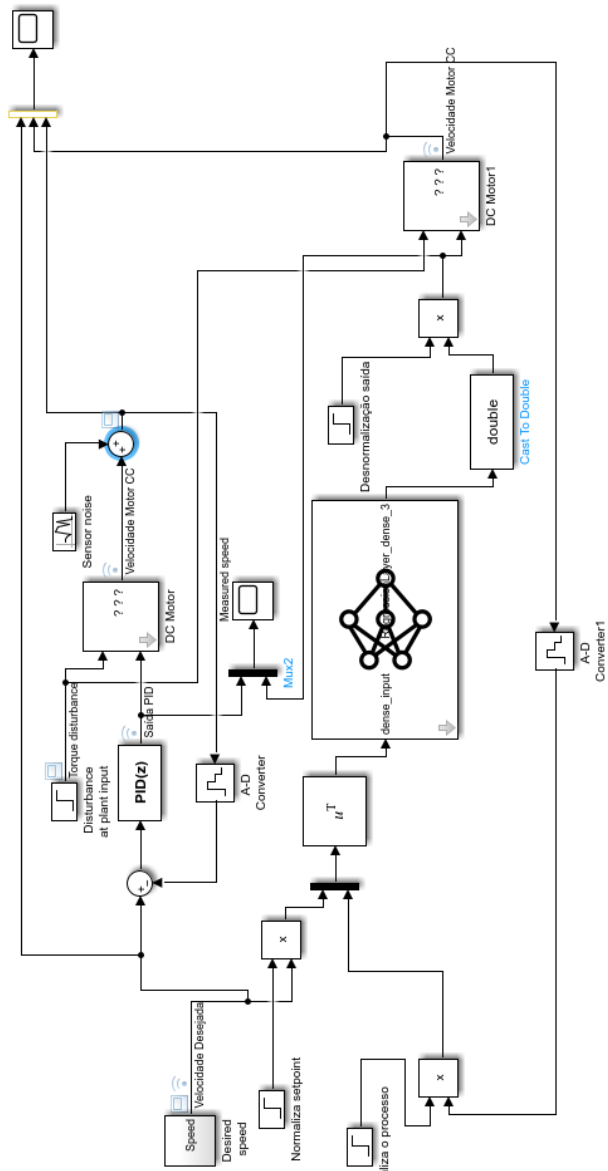
A escolha de um controlador neural não está restrita ao melhor desempenho das métricas MSE e MAE, devendo também ser observado o comportamento no controle do sistema. Para apresentar de forma breve, são trazidas as análises de 6 das arquiteturas propostas.

A Figura 6 apresenta o desempenho das redes 13, 14 e 15 associado com uma definição de *setpoint*.

É possível notar que as redes 13 e 15 apresentam comportamentos similares, não representando diferença significativa em relação ao quantidade de neurônios utilizada em cada camada. A função de ativação utilizadas nessas 3 propostas de rede é a sigmoid, sendo que as respostas geradas por essas redes apresentam características de oscilação suave. Contudo, não apresentam precisão nos valores final e máximo em relação ao comportamento do controlador PID.

A Figura 7 apresenta o desempenho das redes 16, 17 e 18 no mesmo *setpoint* da análise anterior. Nota-se que a rede 16 apresenta comportamento com alta oscilação e com eficácia na obtenção do valor final. Já a rede 17 apresenta alta oscilação, contudo, nessa rede

Figura 5: Simulação para comparar os controladores PID e neural (imagem rotacionada)



não ocorre acurácia no valor final. A rede 18, gera oscilações mais suaves e apresenta resposta mais precisa em valor final.

As redes 16, 17 e 18 utilizaram somente a função de ativação ReLU e o comportamento obtido apresenta caráter mais agressivo ao buscar atingir os valores desejados. Pode-se notar um certo tempo sem oscilação dos valores do controlador PID. Em relação as redes 13, 14, 15 (que utilizam apenas função de ativação sigmoid), as redes com função ReLU tenderam a atingir maior precisão o valor

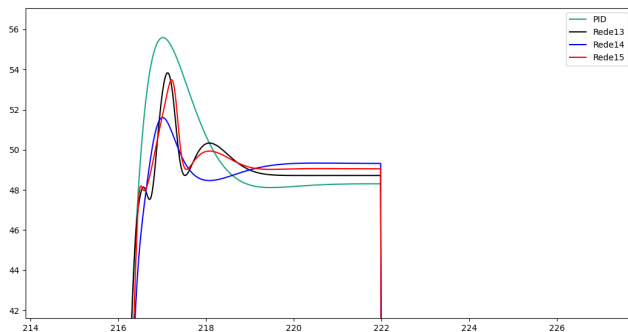
<sup>1</sup><https://www.tensorflow.org>

<sup>2</sup>Uma vez que figura foi rotacionada 90 graus, a parte superior está localizada à esquerda na figura do artigo.

Tabela 5: Métricas obtidas de cada rede

Rede	MSE	MAE
1	0,0006	0,0132
2	0,0003	0,0102
3	0,0005	0,0123
4	0,0009	0,0177
5	0,0007	0,0146
6	0,0004	0,0112
7	0,0007	0,0154
8	0,0004	0,0115
9	0,0010	0,0193
10	0,0006	0,0138
11	0,0009	0,0177
12	0,0005	0,0117
13	0,0009	0,0177
14	0,0011	0,0193
15	0,0009	0,0165
16	0,0004	0,0108
17	0,0004	0,0108
18	0,0005	0,0117

Figura 6: Análise do desempenho das redes 13, 14 e 15



final desejado e por isso tiveram melhor desempenho nas métricas de avaliação.

É importante destacar que a análise do comportamento é uma análise subjetiva e associada com o objetivo do controle, devendo o projetista determinar os comportamentos aceitáveis ao sistema, ou seja, são específicas de cada tipo de problema e equipamentos sendo controlados. De maneira complementar para avaliar o desempenho do sistema, outras métricas podem ser utilizadas, como as relacionadas com os requisitos de desempenho de controle.

### 7.1 Requisitos de desempenho de controle

Para analisar quantitativamente cada configuração de rede, deve-se analisar os requisitos de controle desejados. A Tabela 6 demonstra os resultados obtidos.

Figura 7: Resultado do treinamento das redes 16, 17 e 18

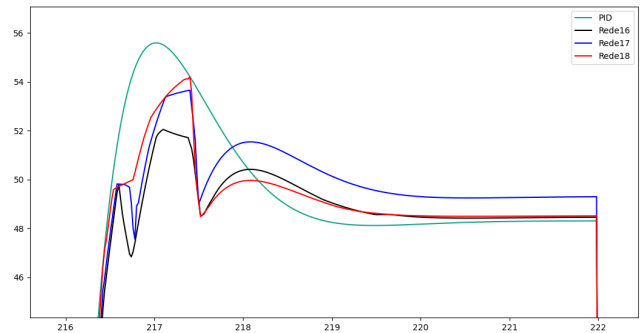


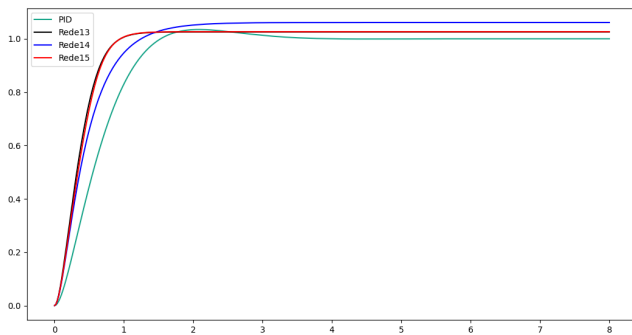
Tabela 6: Métricas qualitativas obtidas de cada rede

Rede	PSS	Ta	Vf
1	0,1%	0,96s	0,980V
2	1%	1,03s	0,960V
3	2%	1,55s	1,003V
4	-	1,67s	1,024V
5	2,2%	1,31s	0,97V
6	2,97%	1,34s	0,98V
7	-	1,10s	1,01V
8	2,7%	1,25s	0,97V
9	-	1,91s	1,04V
10	1,6%	1,2s	0,98V
11	-	1,3s	1,04V
12	0,3%	1,5s	1,04V
13	-	1,2s	0,995V
14	-	1,62s	1,06V
15	-	1,58s	1,02V
16	-	1,30s	1,01V
17	3,5%	1,15s	1,01V
18	5%	1,35s	0,93V

Observa-se que somente as respostas obtidas pelas redes 14 e 18 apresentaram valores distintos dos requisitos de controle esperados. Tais resultados se deram pela imprecisão na resposta em regime permanente.

Portanto, percebeu-se por meio desses testes que os controladores neurais apresentam altas velocidades de resposta para atingir o valor final de velocidade. Além disso, observou-se que os controladores que conseguiram obter precisão nos valores finais obtiveram melhores resultados de resposta, bem como, os controladores que possuem sinais de tensão elevados inicialmente, apresentaram respostas mais rápidas.

**Figura 8: Resposta do sistema ao degrau unitário das redes 13, 14 e 15**



## 7.2 Escolha da configuração de rede

As avaliações anteriores foram realizadas offline e a partir dos dados extraídos da simulação com o controlador PID. Os dados foram utilizados para treinamento e avaliação da rede. Para escolha da rede é proposta uma avaliação adicional aplicando a RNA como controlador neural para o controle da planta com um novo conjunto de *setpoints*.

Para a escolha de configuração de rede para o controlador neural foram definidos dois critérios de simulação: (1) a resposta do sistema ao degrau unitário; e (2) entradas de maneira aleatória. A configuração de rede que garantir comportamento aceito para ambas as simulações, deve ser escolhido para tal aplicação.

A configuração de rede que apresentou comportamentos aceitos tanto na simulação com entrada em degrau unitário como com entradas aleatórias é a rede 13, que atendeu de maneira satisfatória os dois casos. Em relação a etapa de simulação ao degrau unitário, a mesma possuiu valores de tempo de acomodação e acurácia no valor final coerentes, além disso, não conteve valor de percentual de sobressinal.

A Figura 8 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária. As redes 13 e 15 apresentam comportamentos similares entre si, bem como precisão média em torno do valor final, contudo, possuem alta velocidade de resposta para obter o valor final. A rede 12 por sua vez, possui velocidade de resposta relativamente maior que a resposta do controlador PID, porém, essa rede apresenta baixa precisão na resposta em regime permanente.

Em relação a etapa de simulação a entradas aleatórias, a rede 13, atingiu valores coerentes e precisos, evidenciando as 3 últimas entradas, cujo qual, diversas configurações de rede possuíram dificuldade em apresentar acurácia nesses pontos.

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 13, 14 e 15 gerados, é apresentado na Figura 9. Esse grupo de redes apresenta configurações de rede sem variação de funções de ativação, possuindo apenas a função sigmoid em cada camada. As redes 13 e 14 demonstraram resultados mais próximos do esperado durante todo o ciclo, inclusive nas 3 últimas entradas, as duas topologias de rede apresentam precisão significativa. Contudo,

a rede 15, não apresenta acuracidade em regime permanente em determinados pontos desse ciclo em questão.

## 8 CONCLUSÃO

Este trabalho apresentou uma simulação para avaliar o desempenho de propostas de arquiteturas de RNAs para um controlador neural. O Controlador Neural (CN) proposto objetiva mimetizar o comportamento do sistema alcançado com o controlador PID. Alternativamente, poderia ser proposto um CN que aprende apenas baseado na observação do sistema.

É importante destacar que a RNA buscou apenas mimetizar o comportamento controlado pelo PID. Sendo assim, não era objetivo do CN ser superior ao controlador PID. Mesmo assim, é esperado que o CN seja mais adaptável a mudanças sutis do funcionamento da planta. Por exemplo, as interferências do ambiente de trabalho em aplicações reais, resultam em alterações no momento de inércia do motor. Tal fato faz com que o controlador PID modelado em função dos parâmetros iniciais, não apresente o mesmo comportamento, gerando maiores oscilações na resposta do sistema. O aumento do momento de inércia gera maiores oscilações ao sistema controlado por PID do que naquele controlado pelo CN.

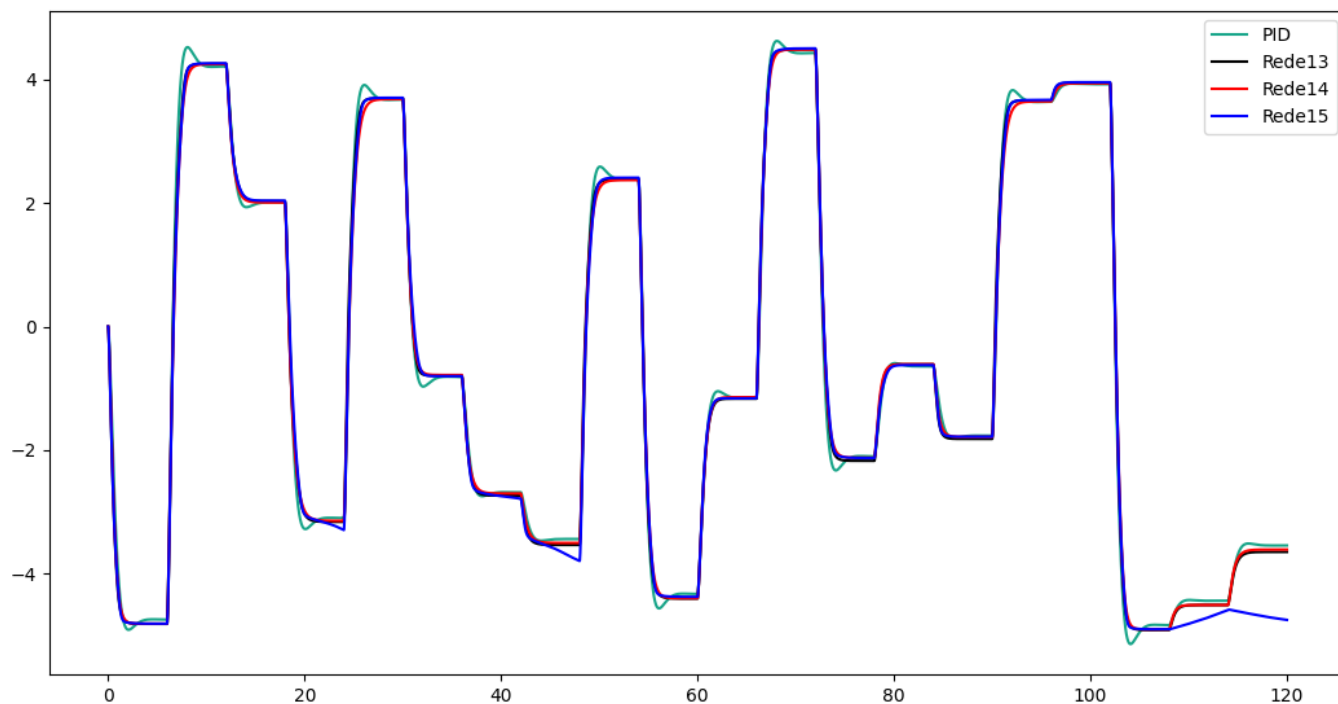
A configuração de rede que apresentou comportamentos aceitos tanto na simulação com entrada em degrau unitário como com entradas aleatórias é a rede 13, cujo qual atendeu de maneira satisfatória os dois casos. Em relação a etapa de simulação ao degrau unitário, a mesma apresenta valores de tempo de acomodação e acurácia no valor final coerentes, além disso, não conteve valor de percentual de sobressinal. Em relação a etapa de simulação a entradas aleatórias, a rede 13, atingiu valores coerentes e precisos, evidenciando as 3 últimas entradas, cujo qual, diversas configurações de rede possuíram dificuldade em apresentar acurácia nesses pontos.

Por fim, este trabalho apresentou métodos de avaliação para escolha da RNA para o CN. Destaca-se que a rede que apresentou melhor desempenho para o sistema (rede 13), não foi a que apresentou melhores métricas MSE e MAE (opções normalmente embasadoras na escolha de uma RNA). Sendo assim, é importante utilizar, para a definição da RNA utilizada no controlador neural, quais aspectos do comportamento são aceitáveis para o sistema e utilizar também parâmetros de avaliação da qualidade do controle para embasar a escolha.

## REFERENCES

- [1] H. BOUDJEDIR, O. BOUHALI, F. YACEF, and N. RIZOUG. 2012. Adaptive neural network for a quadrotor unmanned aerial vehicle. *International Journal in Foundations of Computer Science & Technology* 2, 4 (jul. 2012), 1151–1164.
- [2] K. CHEON, J. KIM, M. Hadamdache, and D. LEE. 2015. On replacing pid controller with deep learning Controller for DC Motor System. *Journal of Automation and Control Engineering* 3, 6 (dez. 2015), 452–456.
- [3] P. S. T. DE BRITO. 2015. *Desenvolvimento de um controlador pid-fuzzy para o controle de nível de água de um pressurizador de um reator nuclear*. 2015. Mestrado em Ciências. Departamento de Engenharia Nuclear, Universidade Federal de Pernambuco, Recife.
- [4] F. FACCIN. 2004. *Abordagem inovadora no projeto de controladores PID*. 2004. Mestrado em Engenharia. Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [5] S. HAYKIN. 2001. *Redes Neurais: princípios e prática*. Bookman, Porto Alegre.
- [6] M. MAGNUSSON. 2018. Comparativo entre controlador PID e redes neurais artificiais no controle de atitude de um quadricóptero. 2018.
- [7] A. E. RIBEIRO. 2016. Controle de temperatura com utilização de controlador neural. 2016.

Figura 9: Resposta do controlador a entradas aleatórias das redes 13, 14 e 15



[8] J. D. RITA. 1995. *Controle de processos usando redes neurais artificiais: uma aplicação experimental*. 1995. Mestrado em Engenharia Química. Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

[9] S. H. SANTOS. 2009. Ajuste de controle pid utilizando algoritmo genético. 2009.

[10] S. SHRIVASTAVA and A. J. ABRAHAM. 2018. Dc motor speed control using machine learning Algorithm. *International Journal in Foundations of Computer*

*Science & Technology* 7, 4 (abr. 2018), 33–37.

[11] N. I. Silva, H. D. Spatti, and R. A. Flauzino. 2010. *Redes neurais artificiais para engenharia e ciências aplicadas*. Artliber, São Paulo.

[12] TUREVSKIY, A. 2019. *PID controller Design for a DC motor*. TUREVSKIY, A. <https://www.mathworks.com/matlabcentral/fileexchange/26275-pid-controller-design-for-a-dc-motor>