

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

GUSTAVO ARINS

PROJETO DE UM CONTROLADOR NEURAL PARA OBTENÇÃO DO
COMPORTAMENTO DE UM PID

Joinville
2022

GUSTAVO ARINS

PROJETO DE UM CONTROLADOR NEURAL PARA OBTENÇÃO DO
COMPORTAMENTO DE UM PID

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Prof. Dr. Benjamin Grando Moreira.

Joinville
2022

AGRADECIMENTOS

A elaboração deste trabalho foi possível graças a força que Deus me deu, não somente na conclusão deste curso, mas durante toda a minha trajetória na universidade. Junto dEle, agradeço a Maria, por sempre estar me protegendo sob seu manto sagrado, curando as minhas inseguranças.

Aos meus pais, meus avós e meu irmão, que nunca mediram esforços para seguir este sonho junto comigo, sou muito grato por terem me ajudado durante toda a minha vida para eu chegar aonde estou.

Agradeço a minha namorada, que me acompanhou durante todo o desenvolvimento do trabalho, bem como ouviu todas as minhas angústias dessa fase e me incentivou a dar o meu melhor, batalhando ao meu lado para chegarmos aonde sonhamos juntos.

A todos os meus professores pela experiência e conhecimento repassados durante este período.

Por fim, agradeço ao meu orientador, por me guiar nesta fase repleta de pesquisas e estudos, me incentivando e auxiliando a fim de se fazer um bom trabalho.

"Não fui eu que ordenei você? Seja forte e corajoso! Não se apavore nem desanime, pois o Senhor, o seu Deus, estará com você por onde você andar. " (Josué 1:9) .

RESUMO

O objetivo desse trabalho é modelar uma rede neural artificial (RNA), denominada de controlador neural (CN), que controle a velocidade de um motor de corrente contínua (CC) e seja tolerante à variações do momento de inércia do rotor. Para a modelagem da rede, utilizou-se a linguagem de programação python e as frameworks de machine learning TensorFlow e Keras. O CN foi treinado com as entradas e saídas de um controlador PID sintonizado, ligado a um motor CC e simulados no Simulink. Para validar o modelo, realizou-se o treinamento de 18 redes neurais e buscou-se avaliar o resultado de cada uma em relação às métricas de treinamento e o desempenho na atuação do sistema. Este trabalho propõe um método de avaliação para a escolha do CN desejado para o sistema, comparando-se o desempenho do controlador PID com o CN, analisando os desempenhos e suas respostas transitórias e de regime permanente para uma entrada unitária e uma entrada complexa. Analisando os resultados de cada CN no sistema de controle, definiu-se uma arquitetura de rede como o controlador que conseguiu atingir os requisitos de desempenho para entradas, tanto unitárias quanto complexas, e o mesmo apresenta desempenhos similares ao controlador PID para entrada unitária e complexa. Além disso, avaliou-se o desempenho do CN em relação às variações de momento de inércia do motor, no qual, foi possível concluir que o sistema controlado pelo CN não apresenta mudanças significativas em seu comportamento, justificando a escolha como controlador.

Palavras-chave: Controlador neural. Controlador PID. Redes Neurais Artificiais.

ABSTRACT

The objective of this work is to model an artificial neural network (ANN), called a neural controller (NC), which controls the speed of a direct current (DC) motor and is tolerant to variations in the rotor's moment of inertia. For the modeling of the network, the python programming language and the TensorFlow and Keras machine learning frameworks were used. The NC was trained with the inputs and outputs of a PID controller connected to a DC motor simulated in Simulink. To validate the model, 18 neural networks were trained and an attempt was made to evaluate the result of each one in relation to the training metrics and the performance of the system. This work proposes an evaluation method for choosing the ideal NC for the system, comparing the performance of the PID controller with the NC, analyzing the performances and their transient and steady-state responses for a single input and a complex input. Analyzing the results of each NC in the control system, network 13 was defined as the controller that managed to reach the performance requirements for both unitary and complex inputs and it presents similar performances to the PID controller for unitary and complex input. In addition, the performance of the NC was evaluated in relation to the variations in the moment of inertia of the motor, in which it was possible to conclude that the system controlled by the NC does not present significant changes in its behavior, justifying the choice for the controller.

Keywords: Neural controller. PID controller. Artificial Neural Network.

LISTA DE FIGURAS

Figura 1 – Componentes de um neurônio biológico.	17
Figura 2 – Modelo matemático de um neurônio artificial.	17
Figura 3 – Gráfico da função limiar.	19
Figura 4 – Gráfico da função linear por partes.	20
Figura 5 – Gráfico da função sigmóide.	21
Figura 6 – Gráfico da função ReLU.	21
Figura 7 – Representação de sistema de controle simples.	22
Figura 8 – Representação de sistema de controle de malha aberta.	23
Figura 9 – Representação de sistema de controle de malha fechada.	23
Figura 10 – Diagrama de controle com os modelos matemáticos.	24
Figura 11 – Resposta de um sistema de segunda ordem e seus requisitos de controle.	25
Figura 12 – Controle PID de uma planta.	27
Figura 13 – Esquemático de etapas de projeto.	29
Figura 14 – Sistema de controle do motor CC no Simulink.	31
Figura 15 – Diagrama eletromecânico do motor CC no Simulink.	32
Figura 16 – Ferramenta de ajuste de parâmetros do controlador PID.	33
Figura 17 – Resposta do controlador ao degrau unitário.	34
Figura 18 – Obtenção dos dados de simulação do sistema de controle do motor CC.	35
Figura 19 – Conjunto de dados de entrada da rede neural.	36
Figura 20 – Conjunto de dados de saída da rede neural.	36
Figura 21 – Simulação comparativa de controladores PID e neural.	39
Figura 22 – Dados de teste simulados com as redes 14 e 16.	43
Figura 23 – Dados de teste simulados com as redes 14 e 16 (ampliado).	43
Figura 24 – Resultado do treinamento das redes 1 a 3.	44
Figura 25 – Resultado do treinamento das redes 1 a 3 (ampliado).	45
Figura 26 – Resultado do treinamento das redes 4 a 6.	46
Figura 27 – Resultado do treinamento das redes 4 a 6 (ampliado).	47
Figura 28 – Resultado do treinamento das redes 7 a 9.	48
Figura 29 – Resultado do treinamento das redes 7 a 9 (ampliado).	49
Figura 30 – Resultado do treinamento das redes 10 a 12.	50
Figura 31 – Resultado do treinamento das redes 10 a 12.	51
Figura 32 – Resultado do treinamento das redes 13 a 15.	52
Figura 33 – Resultado do treinamento das redes 13 a 15.	52

Figura 34 – Resultado do treinamento das redes 16 a 18.	53
Figura 35 – Resultado do treinamento das redes 16 a 18.	54
Figura 36 – Saída do controlador ao degrau unitário das redes 1 a 3.	55
Figura 37 – Resposta do sistema ao degrau unitário das redes 1 a 3.	55
Figura 38 – Saída do controlador ao degrau unitário das redes 4 a 6.	56
Figura 39 – Resposta do sistema ao degrau unitário das redes 4 a 6.	57
Figura 40 – Resposta do controlador ao degrau unitário das redes 7 a 9.	58
Figura 41 – Resposta do sistema ao degrau unitário das redes 7 a 9.	58
Figura 42 – Resposta do controlador ao degrau unitário das redes 10 a 12.	59
Figura 43 – Resposta do sistema ao degrau unitário das redes 10 a 12.	59
Figura 44 – Resposta do controlador ao degrau unitário das redes 13 a 15.	60
Figura 45 – Resposta do sistema ao degrau unitário das redes 13 a 15.	60
Figura 46 – Resposta do controlador ao degrau unitário das redes 16 a 18.	61
Figura 47 – Resposta do sistema ao degrau unitário das redes 16 a 18.	62
Figura 48 – Resposta do controlador a entradas aleatórias das redes 1 a 3.	64
Figura 49 – Resposta do controlador a entradas aleatórias das redes 4 a 6.	65
Figura 50 – Resposta do controlador a entradas aleatórias das redes 7 a 9.	66
Figura 51 – Resposta do controlador a entradas aleatórias das redes 10 a 12.	67
Figura 52 – Resposta do controlador a entradas aleatórias das redes 13 a 15.	68
Figura 53 – Resposta do controlador a entradas aleatórias das redes 16 a 18.	69
Figura 54 – Comparativo entre controladores em função do momento de inércia ($J < 0,01 \text{ kg s}^2$).	70
Figura 55 – Comparativo entre controladores em função do momento de inércia ($J > 0,01 \text{ kg s}^2$).	71

LISTA DE TABELAS

Tabela 1 – Parâmetros de controle PID e suas implicações (ANG et al., 2006)	28
Tabela 2 – Parâmetros de controle PID (AUTOR, 2022)	32
Tabela 3 – Parâmetros de controle PID	33
Tabela 4 – Requisitos de desempenho de controle obtidos	34
Tabela 5 – Parâmetros de controle PID	38
Tabela 6 – Métricas obtidas de cada rede	42
Tabela 7 – Parâmetros de rede do grupo 1 de modelos.	44
Tabela 8 – Parâmetros de rede do grupo 2 de modelos.	46
Tabela 9 – Parâmetros de rede do grupo 3 de modelos.	48
Tabela 10 – Parâmetros de rede do grupo 4 de modelos.	50
Tabela 11 – Parâmetros de rede do grupo 5 de modelos.	51
Tabela 12 – Parâmetros de rede do grupo 6 de modelos.	53
Tabela 13 – Métricas obtidas de cada rede	63

LISTA DE ABREVIATURAS E SIGLAS

CC	Corrente contínua
CN	Controlador neural
DNN	Deep neural networks
MSE	Mean square error
PID	Proporcional, integral e derivativo
RNA	Rede neural artificial
TCC	Trabalho de conclusão de curso

LISTA DE SÍMBOLOS

V	Volt
J	Momento de inércia
R	Resistência
L	Indutância
ω	Velocidade angular
K_t	Constante de torque do motor
K_e	Constante de tensão do terminal aberto
b	Coefficiente de atrito
K_p	Constante proporcional
K_i	Constante integral
K_d	Constante derivativa

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Redes neurais artificiais	16
2.1.1	Neurônios Biológicos	16
2.1.2	Neurônios artificiais	17
2.1.3	Função de ativação	19
2.1.3.1	Função de limiar	19
2.1.3.2	Função linear por partes	20
2.1.4	Função sigmóide	20
2.1.5	Função ReLU	21
2.2	Sistemas de controle	22
2.2.1	Sistemas de malha aberta	22
2.2.2	Sistemas de malha fechada	23
2.2.3	Projeto de sistemas de controle	24
2.2.4	Requisitos de controle	24
2.3	Controle PID	26
2.3.1	Análise dos parâmetros do controlador PID	27
3	METODOLOGIA	29
3.1	Simulação do sistema	30
3.1.1	Matlab/Simulink	30
3.1.2	Sistema de controle de motor CC	31
3.1.3	Dimensionamento do controlador PID	33
3.1.4	Obtenção de dados	34
3.1.5	Distribuição de dados	35
3.2	Treinamento da rede neural	37
3.2.1	Normalização dos dados	37
3.2.2	Configurações de rede	37
3.3	Simulação do controlador neural	38
3.3.1	Modelo de simulação	39
3.3.2	Resposta do sistema	40
4	ANÁLISE DE DADOS	41

4.1	Treinamento das redes neurais	41
4.1.1	Análise das métricas	41
4.1.2	Dados de teste	42
4.1.3	Resultados dos treinamentos	44
4.1.3.1	Primeiro grupo	44
4.1.3.2	Segundo grupo	46
4.1.3.3	Terceiro grupo	47
4.1.3.4	Quarto grupo	49
4.1.3.5	Quinto grupo	51
4.1.3.6	Sexto grupo	53
4.2	Obtenção de dados de simulação	54
4.2.1	Resposta ao degrau unitário	54
4.2.1.1	Primeiro grupo	55
4.2.1.2	Segundo grupo	56
4.2.1.3	Terceiro grupo	57
4.2.1.4	Quarto grupo	59
4.2.1.5	Quinto grupo	60
4.2.1.6	Sexto grupo	61
4.2.2	Requisitos de desempenho de controle	62
4.2.3	Resposta do sistema a entradas aleatórias	63
4.2.3.1	Primeiro grupo	64
4.2.3.2	Segundo grupo	64
4.2.3.3	Terceiro grupo	65
4.2.3.4	Quarto grupo	66
4.2.3.5	Quinto grupo	67
4.2.3.6	Sexto grupo	68
4.2.4	Escolha da configuração de rede	69
4.3	Alterações do momento de inércia	70
5	CONCLUSÕES	72
	REFERÊNCIAS	74

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 (FACCIN, 2004). 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) (RITA, 1995).

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, Magnusson (2018) propõe estudo utilizando redes neurais artificiais e controle PID aplicados no controle de atitude de um quadricóptero, e Brito 2015 expõe o desenvolvimento de um controlador PID baseado em lógica Fuzzy para atuação em um pressurizador de um reator nuclear de água pressurizada.

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 (SANTOS, 2009).

Além disso, o desempenho 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 (BOUDJEDIR et al., 2012).

Um exemplo desse fato ocorre, segundo Abraham e Shrivastava (2018), em uma válvula motorizada utilizada na aplicação de polímeros, 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 sobressinais, ou seja, um sinal que excede seu alvo, pois o desempenho do

controlador se deteriora devido a alterações nos parâmetros do sistema.

Diante desse problema, as estratégias de controle com autoajuste de parâmetros são adequadas para sistemas com características que variam no tempo (SHRIVASTAVA; ABRAHAM, 2018). Sistemas de controle que utilizam *deep learning* são capazes de realizar autoajustes e, atualmente, tem sido aplicados no controle de processos não-lineares complexos (CHEON et al., 2015), 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 redes neurais artificiais e surge com a promessa de resolver problemas encontrados no controle de processos (RIBEIRO, 2016). 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.

Redes neurais artificiais treinadas por retropropagação de erros têm o potencial de aprender modelos que se encontram sobre ou perto de uma variedade não linear (HINTON et al., 2012). Dessa forma, nos últimos anos, os avanços tanto em algoritmos de aprendizado, quanto em hardwares levaram a métodos eficientes para treinar as Deep Neural Networks (DNNs), que contêm muitas camadas de unidades ocultas não lineares e uma elevada camada de saída, conhecida como algoritmos de aprendizado profundo.

A escolha do método de controle baseado em redes neurais artificiais, denominado de controlador neural (CN), se mostra tolerante às mudanças dos parâmetros do sistema, tal como a variação do momento de inércia do rotor (SHRIVASTAVA; ABRAHAM, 2018). 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 repetir o comportamento de um controlador PID e, assim, simular o controle de velocidade de um motor CC, desconsiderando-se o uso de um conversor.

Portanto, fez-se uma simulação das duas topologias de controle: o controlador PID e o controlador neural. Realizou-se então, uma análise qualitativa, buscando a interpretação dos fenômenos e o comportamento de cada método empregado.

Este trabalho apresenta duas contribuições principais: a descrição da simulação dos dois processos de controle e uma proposta de modelo de avaliação para a escolha do CN. A primeira contribuição consiste na análise da resposta do CN em função da variação do momento de inércia do motor e se o mesmo apresenta tolerância ao aumento do momento de inércia. Já a segunda contribuição, consiste em um método de análise para a escolha do CN baseado na resposta do controlador a entradas unitárias e complexas, em função dos requisitos de desempenho de controle.

1.1 OBJETIVO

Buscando experimentar as possibilidades de controle com o uso de neurais artificiais, a pesquisa propõe os seguintes objetivos.

1.1.1 Objetivo Geral

Modelar e treinar uma rede neural artificial que seja capaz de simular o controle de velocidade de um motor CC de maneira robusta e que possua comportamento similar obtido com um controlador PID.

1.1.2 Objetivos Específicos

- Definir requisitos de controle para o motor CC obter o comportamento adequado;
- Desenvolver um controlador PID que atenda aos requisitos de controle definidos.
- Implementar o controlador neural, para garantir comportamento similar ao sistema controlado via controle PID;
- Comparar o comportamento e desempenho do controlador neural em relação ao controlador PID.

2 FUNDAMENTAÇÃO TEÓRICA

De modo a compreender os objetivos propostos pelo presente trabalho, é necessário apresentar conceitos a respeito de machine learning e os conceitos de desenvolvimento de controladores PID. Este capítulo apresenta explicações sobre as redes neurais artificiais e expõe concepções acerca de sistemas de controle e o projeto de controladores.

2.1 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.

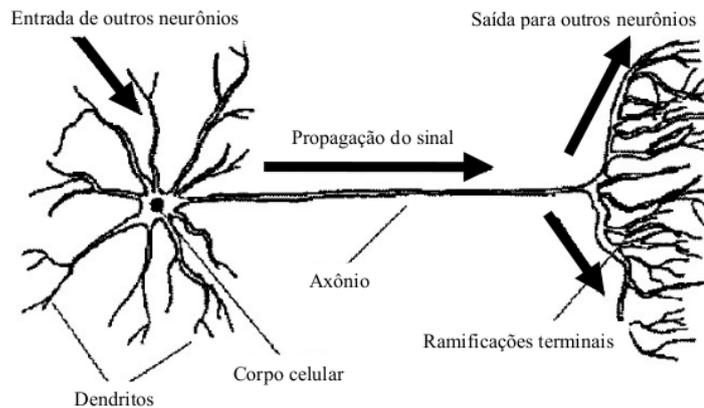
Seguindo esse raciocínio, as redes neurais artificiais (RNA) são modelos computacionais inspirados no sistema nervoso dos seres vivos. Possuem capacidade de aquisição de conhecimento, baseado em informações e são definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões (SILVA; SPATTI; FLAUZINO, 2010).

2.1.1 Neurônios Biológicos

O sistema nervoso humano é composto por neurônios, que são células que transmitem impulsos nervosos e capazes de se comunicar entre si por meio de sinapses. Desse modo, impulsos de entrada são recebidos e processados por um neurônio X, enviando, por meio de neurotransmissores, impulsos nervosos para outro neurônio Y, criando uma cadeia de informações em uma rede de neurônios, auxiliando na coordenação das atividades humanas (BRAGA; CARVALHO; LUDEMIR, 2000).

A Figura 1 apresenta um exemplo de neurônio biológico, onde os dendritos têm por função receber as informações, ou impulsos nervosos, oriundas de outros neurônios e conduzi-las até o corpo celular (COPPIN, 2010). Nessa região, a informação é processada, e novos impulsos são gerados. Esses impulsos são transmitidos a outros neurônios, passando através do axônio até os dendritos dos neurônios seguintes.

Figura 1 – Componentes de um neurônio biológico.



Fonte: Adaptado de Silva (1998, p. 9).

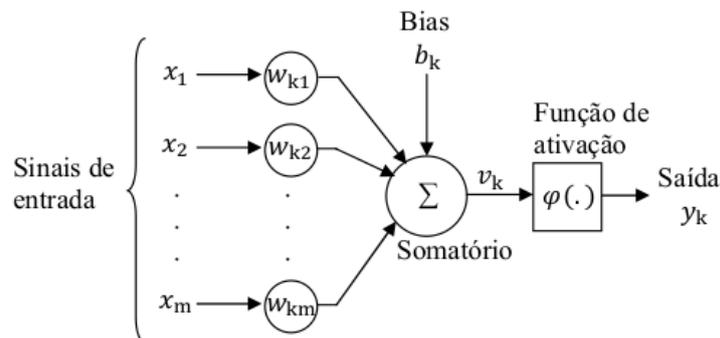
O ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro é chamado de sinapse. São pelas sinapses que os nodos se unem funcionalmente, formando redes neurais. As sinapses funcionam como válvulas, e são capazes de controlar a transmissão de impulsos (BRAGA; CARVALHO; LUDEMIR, 2000).

2.1.2 Neurônios artificiais

A estrutura das redes de neurônios artificiais foi modelada em relação aos modelos conhecidos de sistemas nervosos biológicos e particularmente, do cérebro humano (SILVA; SPATTI; FLAUZINO, 2010). Assim sendo, o neurônio de uma rede neural artificial (RNA) opera de maneira similar ao funcionamento de um neurônio biológico.

O primeiro modelo matemático do neurônio foi proposto por McCulloch e Pitts em 1943, sendo o mesmo ainda utilizado como representação de um neurônio artificial (ROSA, 2011). A Figura 2 mostra o modelo matemático proposto.

Figura 2 – Modelo matemático de um neurônio artificial.



Fonte: Adaptado de Haykin (2001, p. 36).

Portanto, com base nesse modelo, verifica-se que o neurônio artificial é constituído por sete elementos básicos (HAYKIN, 2001):

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-se quantificar as relevâncias de cada entrada em relação à funcionalidade do respectivo neurônio;
3. Somatório (Σ): Tem como função, agregar os sinais de entrada já ponderados para produzir um valor potencial de ativação;
4. Bias 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*;
6. Função de ativação φ : 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 e diante do modelo roposto por McCulloch e Pitts, 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 (HAYKIN, 2001). Ao final, ao resultado dessa soma é adicionado um *bias*, passando pela função de ativação φ , 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 φ é a função de ativação (HAYKIN, 2001).

2.1.3 Função de ativação

A função de ativação é aquela que processa o sinal gerado pela combinação linear das entradas e dos pesos das sinapses, para gerar o sinal de saída do neurônio e tem por finalidade limitar a saída do neurônio dentro de um intervalo de valores (SILVA; SPATTI; FLAUZINO, 2010). Esse intervalo depende de qual tipo de função de ativação é utilizada, sendo comumente utilizadas, quatro tipos de funções:

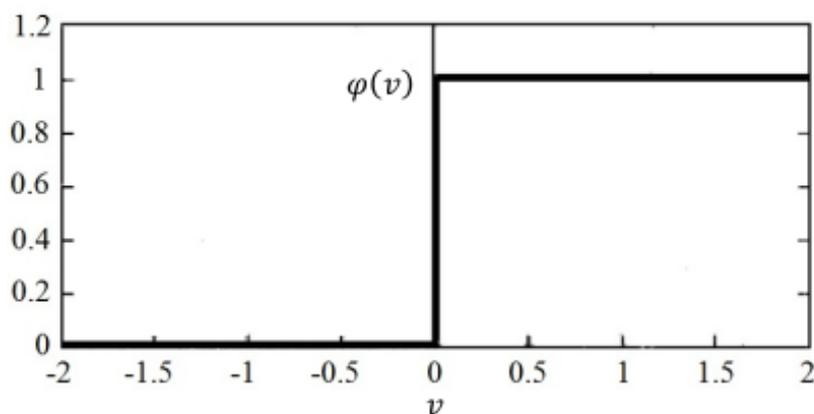
1. Função de limiar;
2. Função linear por partes;
3. Função sigmóide;
4. Função ReLU.

2.1.3.1 Função de limiar

Para esse tipo de função de ativação, a saída do neurônio é igual a zero, quando seu valor for negativo e 1, quando seu valor for positivo. A função de limiar ou Heavside, é dada pela Equação 3 e representada pela Figura 3.

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (3)$$

Figura 3 – Gráfico da função limiar.



Fonte: Adaptado de Haykin (2001, p. 39).

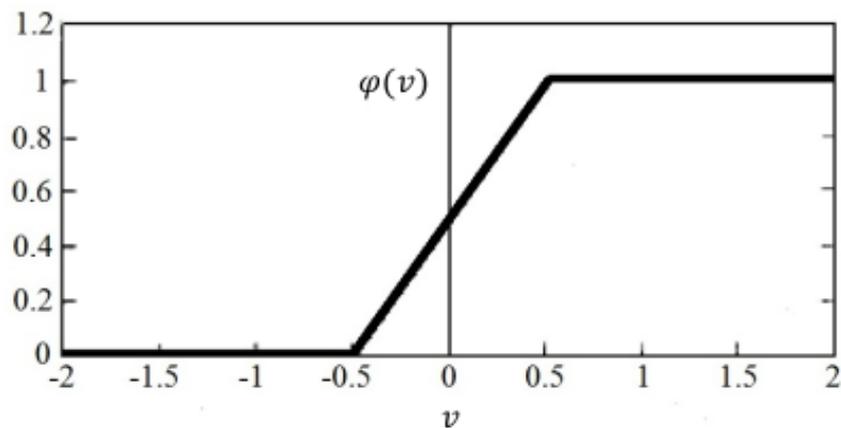
A função de ativação, representada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido ou potencial de ativação v (HAYKIN, 2001), que representa a diferença entre o resultado do somatório e o *bias*.

2.1.3.2 Função linear por partes

Para a função linear por partes, dada pela Equação 4, assume-se que o fator de amplificação dentro da região linear de operação é unitário. De acordo com a Figura 4, a função linear pode ser restringida para produzir valores constantes em um intervalo de $[0,1]$, obtendo a característica de uma função do tipo rampa.

$$\varphi(v) = \begin{cases} 1, & v \geq \frac{1}{2} \\ v & -\frac{1}{2} < v < \frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (4)$$

Figura 4 – Gráfico da função linear por partes.



Fonte: Adaptado de Haykin (2001, p. 39).

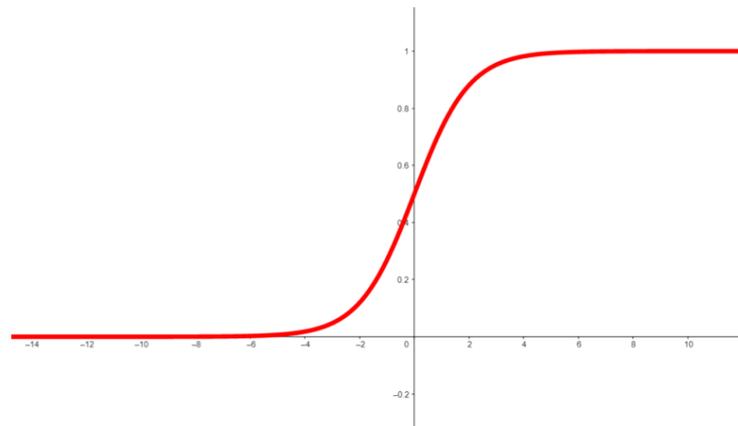
Dessa forma, a função linear por partes apresenta valores dos potenciais de ativação quando estes estão definidos no intervalo de $[-\frac{1}{2}, \frac{1}{2}]$, caso contrário, restringe-se aos valores limites.

2.1.4 Função sigmóide

A função sigmóide simboliza a função de ativação mais utilizada presente na construção de das redes neurais artificiais. É definida como uma função estritamente crescente que exhibe equilíbrio entre comportamento linear e não-linear. A Figura 5 indica o comportamento da função sigmóide e é representada pela Equação 5.

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (5)$$

Figura 5 – Gráfico da função sigmóide.



Fonte: Adaptado de Silva (2010, p. 39).

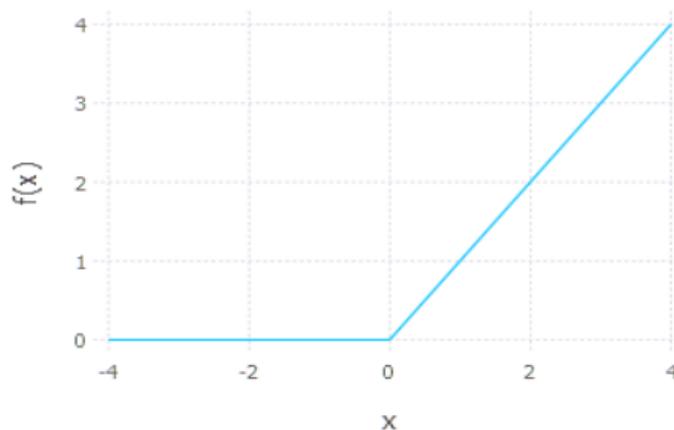
Onde a é o parâmetro de inclinação da função sigmóide, onde ao variar este parâmetro, é possível variar a inclinação da função de ativação.

2.1.5 Função ReLU

A função ReLU é inspirada nos neurônios do cérebro que retornam a um valor positivo ou a 0. Atualmente, a ReLU é a função de ativação mais usada no mundo (RIZZO; CANATO, 2020). A representação gráfica para a função ReLU é esboçada através da Figura 6 e seu comportamento matemático indicado pela Equação 6.

$$\varphi(v) = \max(0, v) \quad (6)$$

Figura 6 – Gráfico da função ReLU.



Fonte: Adaptado de Rizzo (2020).

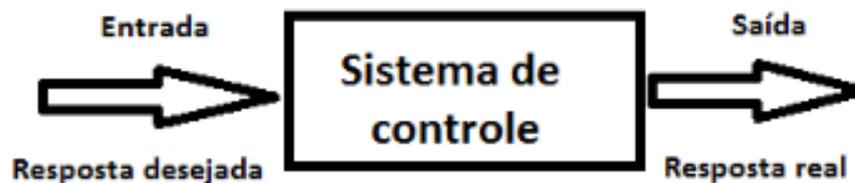
A função de ativação ReLU não ativa todos os neurônios ao mesmo tempo, ou seja, se o valor de entrada for negativo, atribui-se o valor zero à saída e o neurônio não

é ativado (SILVA; CASSIA; CARVALHO, 2021).

2.2 SISTEMAS DE CONTROLE

Um sistema de controle consiste em subsistemas e processos, denominados plantas, construídos com o objetivo de se obter uma saída desejada com desempenho desejado, para uma entrada específica fornecida (NISE, 2011). Logo, o sistema de controle deve controlar a saída da planta por meio de modelação matemática, com o objetivo de atingir a entrada especificada, a Figura 7 mostra de forma simplificada um sistema de controle com entradas e saídas.

Figura 7 – Representação de sistema de controle simples.



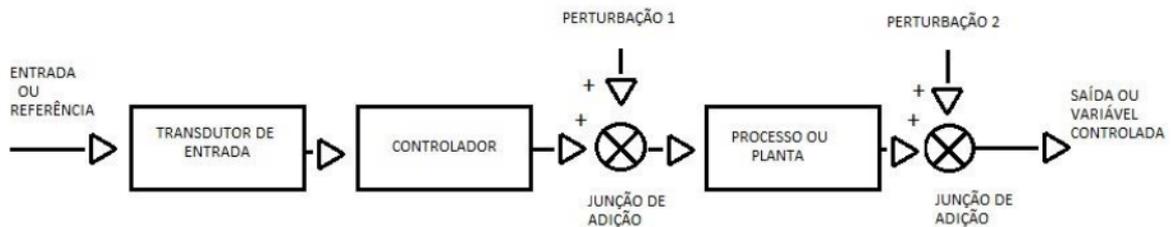
Fonte: Autor.

De maneira geral, o controle de sistemas busca otimizar um processo de modo que uma saída específica seja alcançada a partir de uma entrada de referência fornecida. Dessa forma, os sistemas de controle podem se dividir em dois tipos: sistemas de malha aberta e sistemas de malha fechada.

2.2.1 Sistemas de malha aberta

Um sistema de malha aberta pode ser entendido como um sistema em que o controle independe do valor de saída. Os sistemas de malha aberta não são capazes de corrigir perturbações e são comandados simplesmente pela entrada. A Figura 8 ilustra um sistema de malha aberta.

Figura 8 – Representação de sistema de controle de malha aberta.



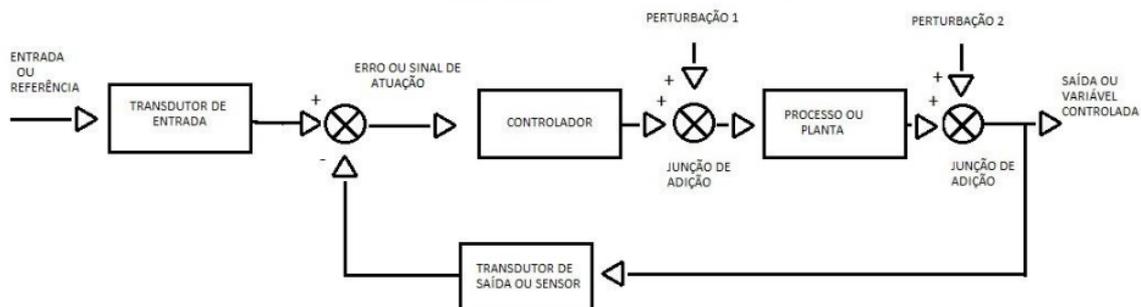
Fonte: Adaptado de Nise (2011, p. 7).

Em um sistema de Malha Aberta genérico, o transdutor converte o sinal de entrada, o controlador aciona o processo ou planta e, na saída, tem-se o sinal controlado. Observa-se por meio da Figura 8 que as perturbações são somadas ao processo, e não há compensação para tais perturbações (NISE, 2011).

2.2.2 Sistemas de malha fechada

Em um sistema de malha fechada, o controle depende do valor de saída, ou seja, esse tipo de sistema tem uma realimentação para compensar possíveis perturbações. O controlador é modelado matematicamente e por meio da realimentação na saída, é possível corrigir a saída do sistema para convergir aos valores de entrada desejados. A Figura 9 mostra o diagrama de tal sistema.

Figura 9 – Representação de sistema de controle de malha fechada.



Fonte: Adaptado de Nise (2011, p. 7).

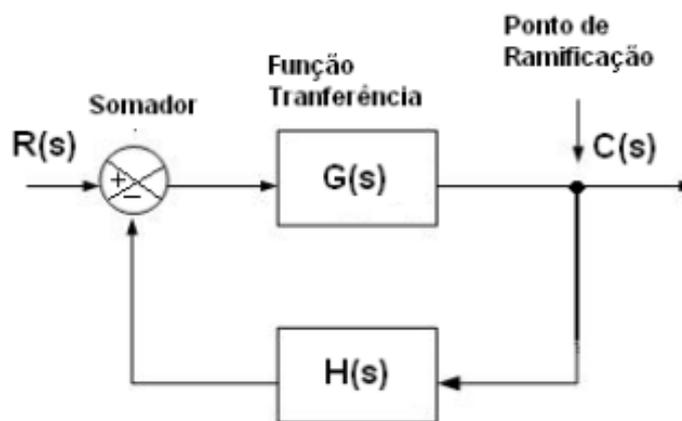
Onde o transdutor converte o sinal de entrada para a grandeza do sistema, o controlador aciona a planta ou processo e, diferentemente do sistema de malha aberta, o sinal retorna ao outro transdutor, podendo ser um sensor que efetua a leitura do valor dessa variável. Se necessário, as devidas correções são executadas, e esse sinal ou variável retorna ao controlador para compensar o erro gerado na saída (NISE, 2011).

2.2.3 Projeto de sistemas de controle

Os sistemas de controle reais são geralmente não lineares e para resolver esse problema, busca-se a aproximação do sistema por modelos matemáticos lineares (OGATA, 2013). Dessa forma, é necessário conhecer o comportamento do sistema a ser controlado para poder dimensionar o controlador ideal.

A Figura 10 mostra o diagrama de sistema de controle de malha fechada que relaciona, de forma simples, as entradas e saídas do sistema e sua função de transferência.

Figura 10 – Diagrama de controle com os modelos matemáticos.



Fonte: Adaptado de Ogata (2013, p. 16).

Onde $R(s)$, $G(s)$, $C(s)$ e $H(s)$ representam a entrada do sistema, a função de transferência da planta a ser controlada, a saída gerada e o modelo matemático do sensor de leitura, respectivamente. Sendo assim, busca-se inicialmente obter a função de transferência do sistema, que relaciona a saída com a entrada do sistema.

A Equação 7 demonstra a função de transferência para o sistema de controle com realimentação.

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (7)$$

Com essa equação, é possível modelar matematicamente um controlador que atenda requisitos de resposta do sistema ao degrau unitário.

2.2.4 Requisitos de controle

Considerando que um sistema de segunda ordem estabelece grande parte dos sistemas lineares encontrados atualmente, os requisitos de controle são estabelecidos por meio da resposta ao degrau unitário de um sistema de segunda ordem

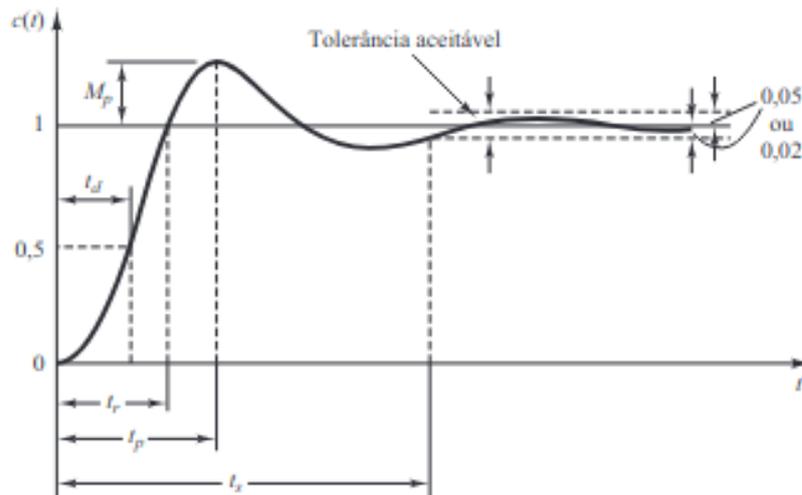
subamortecido. Esse sistema apresenta uma resposta oscilatória, contudo, convergente ao valor determinado.

A Equação 8 apresenta a função de transferência de um sistema de segunda ordem.

$$G(s) = \frac{K\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8)$$

Onde ζ , K e ω_n representam o coeficiente de amortecimento, o valor em estado estacionário e a frequência natural, respectivamente. A Figura 11 representa a resposta de um sistema subamortecido ao degrau unitário, bem como os requisitos de controle gerados pela resposta do sistema ao degrau unitário.

Figura 11 – Resposta de um sistema de segunda ordem e seus requisitos de controle.



Fonte: Adaptado de Ogata (2013, p. 155).

Observa-se que a resposta do sistema de segunda ordem ao degrau unitário apresenta parâmetros que informam o comportamento do sistema. A lista a seguir descreve cada parâmetro:

- Tempo de atraso (T_d): Define-se como o tempo requerido para que a resposta alcance metade de seu valor final pela primeira vez;
- Tempo de subida (T_r): Tempo requerido para que a 0% a 100% do valor final;
- Tempo de pico (T_p): Tempo para que a resposta ao degrau unitário atinja o primeiro pico de sobressinal;
- Máximo de sobressinal (M_p): Valor máximo de pico da resposta. A porcentagem máxima de sobressinal (PSS) é definida por:

$$PSS = \frac{c(tp) - c(\infty)}{c(\infty)} \quad (9)$$

O valor máximo de sobressinal indica diretamente a estabilidade relativa do sistema;

- Tempo de acomodação (T_s): Tempo necessário para que a curva de resposta alcance valores em uma faixa (geralmente de 2% ou 5%) em torno do valor final, aí permanecendo indefinidamente;
- Erro Estacionário (E_s): É a diferença percentual entre o estado desejado para o sistema e o estado atual efetivo do sistema em regime permanente.

Dessa forma, durante a modelação do controlador para o sistema a ser controlado, deve-se buscar atender aos requisitos de controle citados.

2.3 CONTROLE PID

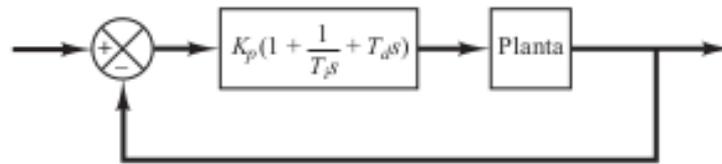
O controlador Proporcional Integral Derivativo (PID) consiste em um método utilizado para controlar o comportamento de sistemas e processos em geral. A utilidade dos controles PID está na sua aplicabilidade geral à maioria dos sistemas de controle. Em particular, quando o modelo matemático da planta não é conhecido e, portanto, métodos de projeto analítico não podem ser utilizados, esses tipos de controladores se tornam os mais úteis (OGATA, 2013).

A resposta de um controlador PID padrão é estabelecido por três parâmetros: o ganho proporcional (K_p), o ganho integral (K_i) e o ganho derivativo (K_d). A Equação 10 consiste na função de transferência no domínio do tempo que relaciona os parâmetros do controlador PID e a resposta do sistema.

$$c(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (10)$$

Onde $e(t)$ é o sinal de erro gerado pela diferença entre a saída da planta e o valor desejado e $c(t)$ é a resposta do controlador. Dessa forma, deve-se buscar encontrar os parâmetros K_p , K_i e K_d , em função dos requisitos de controle desejados, para utilizá-lo no controle da planta. A Figura 12 mostra o diagrama de um controlador PID.

Figura 12 – Controle PID de uma planta.



Fonte: Adaptado de Ogata (2013, p. 522).

Alterando os parâmetros K_p , K_i e K_d , é possível determinar o comportamento das respostas transitória e estacionária do controlador PID (NISE, 2011). O conjunto de 3 parâmetros definem a velocidade de resposta transitória e sua acurácia em regime permanente, dessa forma, torna-se inviável definir os valores exatos de parâmetros que alcance exatamente todos os requisitos desejados de velocidade e acurácia para o processo, pois quando se privilegia um desses fatores, o outro acaba por se deteriorar.

O comportamento que o parâmetro K_p fornece, define-se por uma ação de controle geral proporcional ao sinal de erro. Já K_i está relacionado ao erro em regime permanente e K_d interfere diretamente na velocidade de resposta transitória do controlador (ANG et al., 2006).

2.3.1 Análise dos parâmetros do controlador PID

Os parâmetros de controle do controlador PID são ajustados de modo que o sistema seja estável e atenda, em diferentes níveis, um ou mais dos seguintes objetivos, a depender das necessidades específicas da aplicação (ANG et al., 2006):

- Estabilidade;
- Rápida resposta em sistema transitório;
- Desempenho em regime permanente;
- Maior robustez contra incertezas de modelagem;
- Atenuação de ruídos.

Evidencia-se que muitos desses objetivos são disjuntos, ou seja, a priorização de um objetivo leva a degeneração dos outros. A Tabela 1 ilustra algumas das relações existentes entre os parâmetros da resposta e o aumento do valor dos parâmetros de controle, onde T_d , T_r , M_p , T_s e E_s , representam o tempo de atraso, tempo de subida, máximo de sobressinal, tempo de acomodação e erro em estado estacionário, respectivamente.

Tabela 1 – Parâmetros de controle PID e suas implicações (ANG et al., 2006)

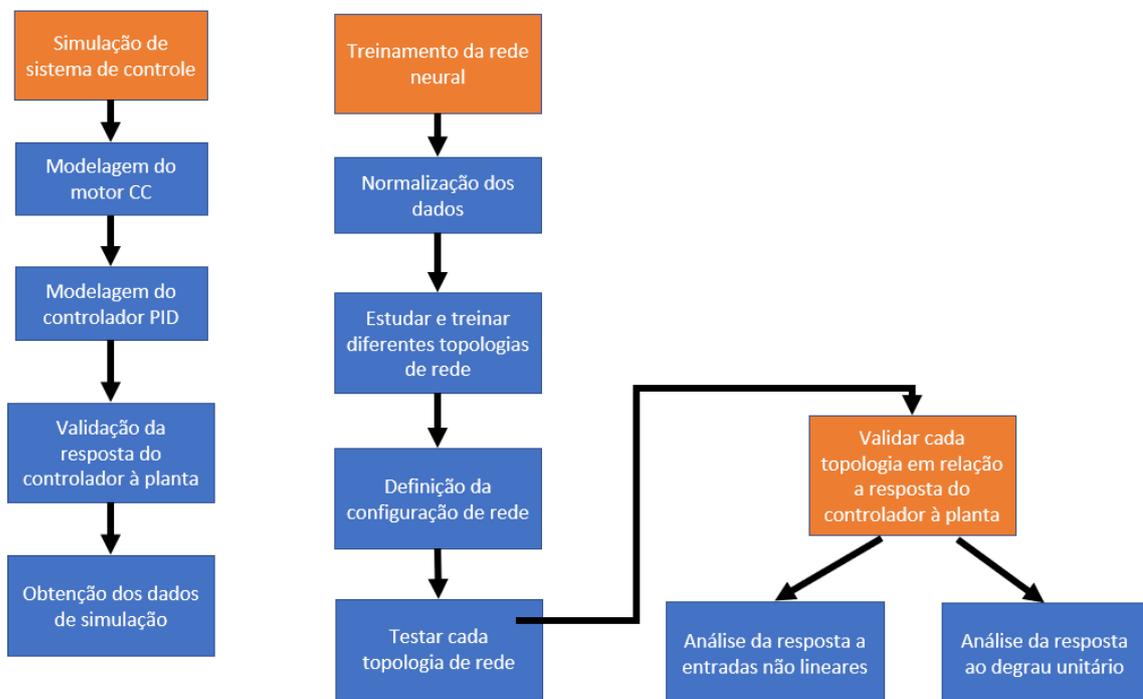
Parâmetro	Td	Tr	Mp	Ts	Es	Estabilidade
↑ Kp	↓	↓	↑	↑	↓	↓
↑ Ki	↓	↓	↑	↑	↓	↓
↑ Kd	↓	↓	↓	↓	↓ ou ↑	↑

Conhecendo as implicações de cada parâmetro de controle, pode-se então modelar o controlador PID que garanta os requisitos de desempenho em regime transitório e permanente. Sabendo que há ilimitados valores possíveis para os parâmetros de controle, deve-se avaliar se a resposta gerada pela escolha desse controlador, garante o comportamento desejado.

3 METODOLOGIA

A metodologia utilizada neste trabalho utiliza dados provenientes de simulação e a pesquisa é 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 Artificial (RNA) 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 13 mostra as etapas de projeto.

Figura 13 – Esquemático de etapas de projeto.



Fonte: Autor

De forma macro, a primeira etapa do projeto consiste na simulação do sistema de controle com o objetivo de obtenção dos dados para o treinamento da rede neural. 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 rede neural, onde nesta etapa deve-se realizar a normalização dos dados obtidos por meio da simulação.

Para encontrar a configuração de rede ideal, deve-se testar 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, deve-se avaliar 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 ao degrau unitário e então, analisar a resposta a entradas não lineares.

3.1 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.

3.1.1 Matlab/Simulink

O Matlab (MATrix LABoratory) é um software interativo, que integra a capacidade de fazer cálculos, visualização gráfica e programação em um ambiente de utilização simples, em que problemas e soluções são expressos em uma linguagem matemática familiar (CARVALHO, 2015). Os usos típicos para o Matlab incluem:

- Cálculos matemáticos;
- Desenvolvimento de algoritmos;
- Modelagem, simulação e confecção de protótipos;
- Gráficos científicos;
- Análise, exploração e visualização de dados;
- Desenvolvimento de aplicações, incluindo a elaboração de interfaces gráficas com o usuário.

O Matlab também possui bibliotecas auxiliares que otimizam o tempo gasto para realizar tarefas, uma vez que, o usuário pode utilizar as funções já definidas, poupando o tempo de criá-las (BECKER et al., 2010). Além disso, o Matlab possui diversos módulos que agrupam funções para diferentes aplicações.

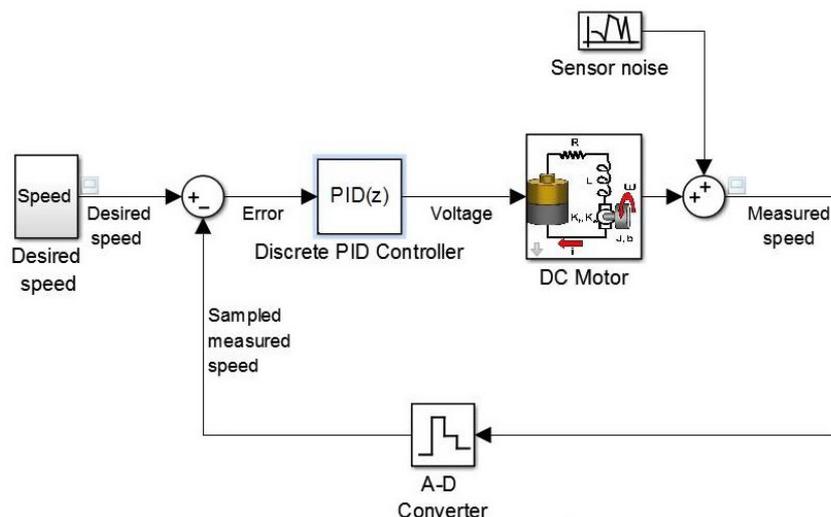
O Simulink é um pacote adicional do MATLAB capaz de modelar, simular, e analisar sistemas dinâmicos de múltiplos domínios, além disso, suporta sistemas lineares e não lineares modelados em tempo contínuo e discreto. Com esse pacote,

é possível utilizar uma ferramenta gráfica de diagramação de blocos e um conjunto personalizável de bibliotecas de blocos para montar os sistemas a serem simulados.

3.1.2 Sistema de controle de motor CC

Para simular o controle de velocidade de um motor CC de excitação independente, utilizou-se a toolbox do Simulink denominada *PID Controller Design for a DC Motor* desenvolvida por (TUREVSKIY, A., 2019), conforme demonstra a Figura 14. Com essa ferramenta, o usuário informa os parâmetros de um motor CC, como o coeficiente de amortecimento, momento de inércia, resistência e indutância. O usuário também precisa determinar os parâmetros K_p , K_d e K_i controlador PID que são representados por blocos interligados.

Figura 14 – Sistema de controle do motor CC no Simulink.



Fonte: Autor

O sistema é simulado e seu comportamento determinado, permitindo a análise do sistema para diversas situações e parametrizações possíveis e, dessa forma, estabelecer os impactos de cada mudança de parâmetro no funcionamento do sistema. Essas informações são utilizadas para parametrizar o controlador PID com o intuito de que o comportamento do motor CC satisfaça as condições de funcionamento determinadas no projeto do sistema, buscando baixos percentuais de sobressinal, acuracidade na resposta em regime permanente e menores tempos de acomodação para o controle do sistema.

Nessa toolbox, o motor CC é modelado como um bloco onde um nível de tensão V (volts) e a velocidade angular ω (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 11 e 12, denominadas de equações dinâmicas

de um motor CC.

$$\frac{d\omega}{dt} = \frac{1}{J}(K_t \cdot i_a - b \cdot \omega) \quad (11)$$

$$\frac{di_a}{dt} = \frac{1}{L}(V - R \cdot i_a - K_e \cdot \omega) \quad (12)$$

Onde J é o momento de inércia do rotor ($\text{kg}\cdot\text{m}^2$), K_t é a constante de torque do motor ($\text{N}\cdot\text{m}\cdot\text{rad}/\text{A}$), 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 (Ω) e V é a constante de força eletromotriz ($\text{V}\cdot\text{s}/\text{rad}$) (CHEON et al., 2015).

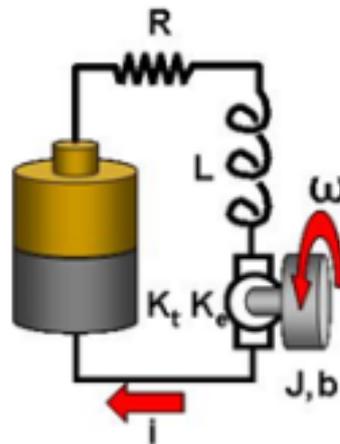
O valor de cada parâmetro de dimensionamento do motor CC são apresentados na Tabela 2.

Tabela 2 – Parâmetros de controle PID (AUTOR, 2022)

J	K_t	b	L	R	K_e
0,01	0,01	0,1	0,5	1	0,01

Dessa forma, com os parâmetros do motor CC definidos, pode-se visualizar por meio da Figura 15 o diagrama eletromecânico do motor CC e seus parâmetros.

Figura 15 – Diagrama eletromecânico do motor CC no Simulink.



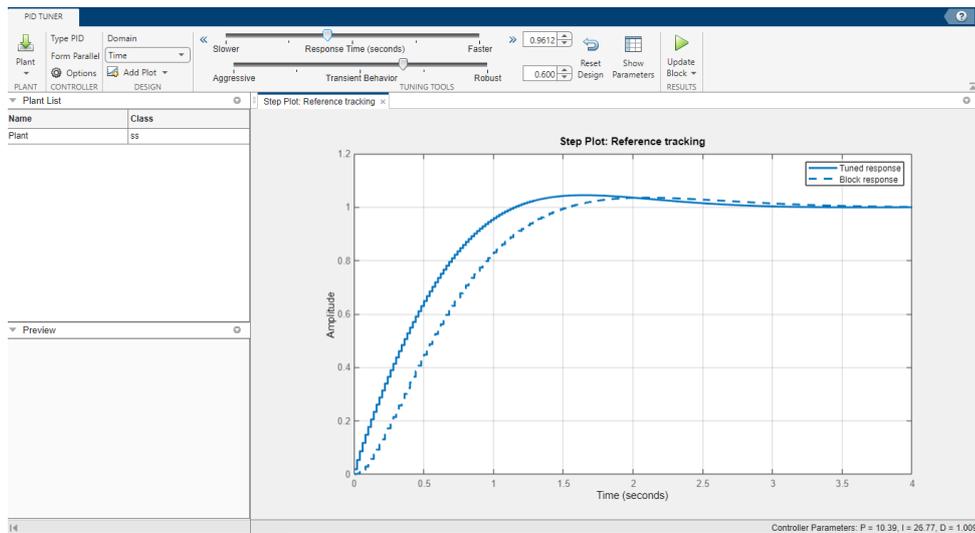
Fonte: Autor.

Com a modelagem do motor pronta, deve-se montar e parametrizar o controlador PID que satisfaça as necessidades de tempo de resposta e acurácia determinadas para o processo.

3.1.3 Dimensionamento do controlador PID

Utilizando uma ferramenta do Simulink chamada PID Tuner App, foi possível ajustar os parâmetros do controlador PID para controlar o motor CC através das funções de transferência representadas pelas Equações 11 e 12 e da passagem de requisitos de controle realimentado de velocidade e precisão desejados para a resposta. A Figura 16 demonstra a interface da ferramenta com o objetivo de apresentar a resposta em função do comportamento transiente e tempo de reposta.

Figura 16 – Ferramenta de ajuste de parâmetros do controlador PID.



Fonte: Autor.

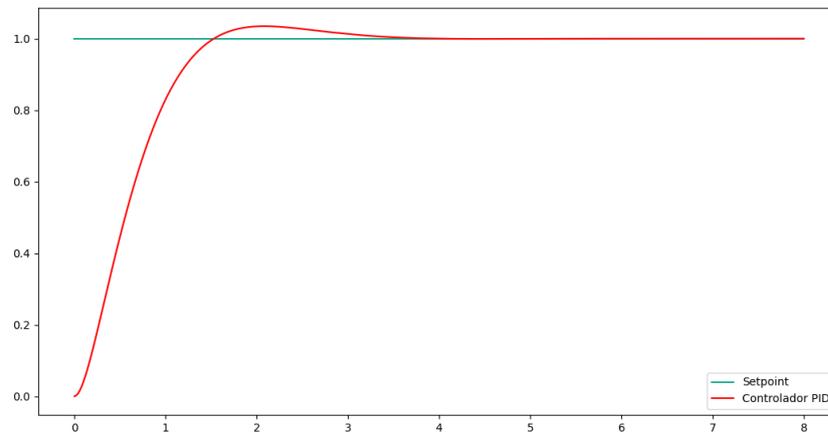
Se o sistema for ajustado para uma velocidade de resposta mais rápida, a saída do controlador permanece com oscilação por um período transitório. Porém, se o sistema for ajustado para uma maior precisão em regime permanente, a resposta ficou mais lenta. Portanto, buscou-se um meio termo entre velocidade de resposta e robustez, ajustando os parâmetros do controlador PID para os valores apresentados na Tabela 3.

Tabela 3 – Parâmetros de controle PID

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

Onde K_p , K_i e K_d representam os ganhos proporcional, integral e derivativo do controlador PID e N representa o coeficiente de filtro no diferenciador. Com esses parâmetros é possível obter a resposta do sistema ao degrau unitário. A Figura 17 mostra a resposta do sistema em comparação ao valor desejado: 1 rad/s.

Figura 17 – Resposta do controlador ao degrau unitário.



Fonte: Autor

Portanto, a resposta do controlador atende aos requisitos de desempenho de controle, conforme demonstra a Tabela 4.

Tabela 4 – Requisitos de desempenho de controle obtidos

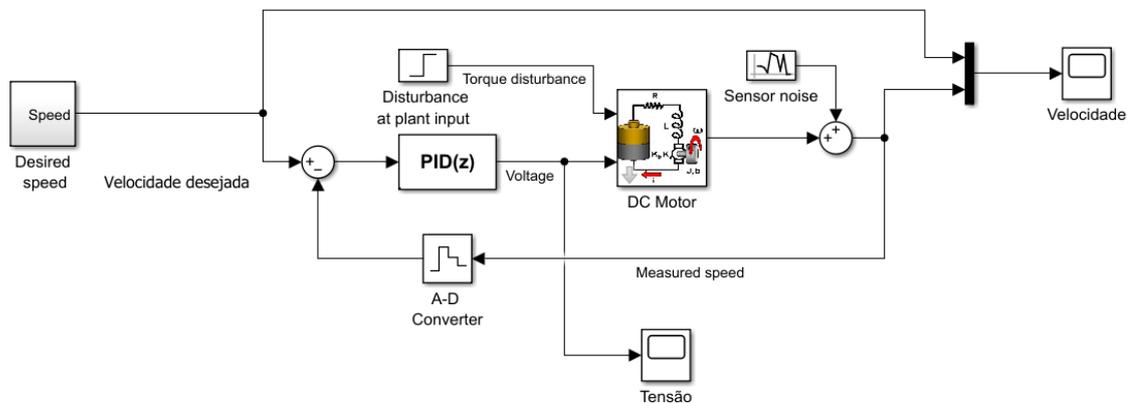
PSS desejado	PSS obtido	Ta desejado	Ta obtido	Vf desejado	Vf obtido
< 5%	3,38%	< 3s	2,75s	1 (\pm 0,05) rad/s	1 rad/s

Onde T_s , PSS e Vf representam o tempo de acomodação, percentual de sobressinal e o valor final, respectivamente. Portanto, pode-se notar que o desempenho do controlador PID atende aos requisitos de controle estipulados.

3.1.4 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. A Figura 18 mostra os pontos de dados coletados do sistema.

Figura 18 – Obtenção dos dados de simulação do sistema de controle do motor CC.



Fonte: Autor

Portanto, é possível notar, por meio do sistema apresentado na Figura 18, que deseja-se obter os valores de simulação necessários para o treinamento do controlador neural: a velocidade desejada, a velocidade do motor CC e saída do 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.

3.1.5 Distribuição de dados

Como forma de obter o maior número possível de dados para que a rede neural possa efetuar o treinamento de maneira adequada, procura-se dados que simulem diversas situações: 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.

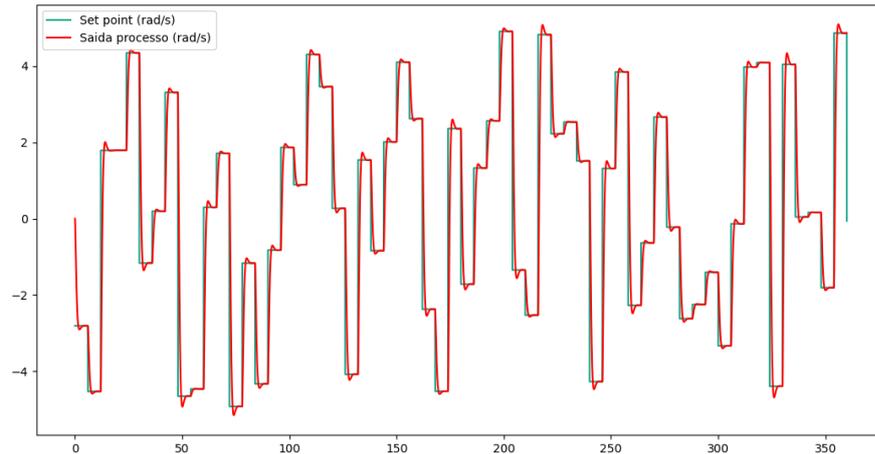
Portanto, de acordo com o tempo de acomodação da resposta transitória do sistema em relação ao controlador PID, é possível definir um tempo de ciclo de mudanças de velocidades desejadas que forneça um intervalo ideal e que garanta a amostragem da convergência de velocidade do sistema. O tempo de acomodação para a entrada degrau unitário equivale a 2,75 segundos, dessa forma, deve-se definir um tempo no mínimo duas vezes maior que o obtido. Define-se então, 6 segundos como o tempo ideal de mudança de velocidade desejada.

O conjunto de dados de entrada para o treinamento da RNA consiste na velocidade desejada e na velocidade obtida por meio do controle PID. A velocidade desejada varia estocasticamente de -5 a 5 rad/s, garantindo então a obtenção de dados

que variam em torno dessa faixa de velocidade.

A Figura 19 mostra o conjunto de dados de entrada da RNA utilizado neste estudo. Tal conjunto contém o valor da velocidade desejada juntamente com o valor de velocidade obtido no instante de tempo.

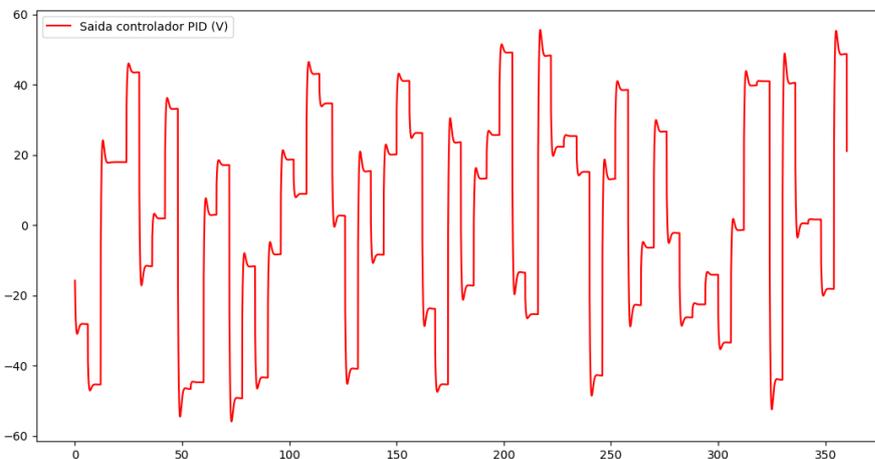
Figura 19 – Conjunto de dados de entrada da rede neural.



Fonte: Autor

Representando os dados de saída do treinamento da RNA, os sinais de saída do controlador PID são mostrados na Figura 20. Esse conjunto de dados representa o valor de tensão em que o controlador PID deve obter para atingir o comportamento de velocidade desejado e por sua vez, a RNA deve buscar obter comportamento similar.

Figura 20 – Conjunto de dados de saída da rede neural.



Fonte: Autor

É possível notar que o conjunto de dados escolhido, abrange valores de velocidade desejada com variações tanto severas quanto suaves. Contudo, observa-se que esse conjunto de dados possui maior número de dados com variações severas do que suaves, sendo que tal característica pode influenciar nos resultados do controlador neural.

3.2 TREINAMENTO DA REDE NEURAL

Com o conjunto de parâmetros de controle definidos e os dados de entrada e saída da RNA já obtidos, é possível então realizar a modelagem dessa rede. Contudo, por meio de testes preliminares, percebeu-se que ao variar a configuração da RNA, a resposta da mesma varia de maneira significativa.

Portanto, julga-se necessário avaliar uma série de configurações de redes com o objetivo de encontrar a topologia que apresentou melhor resultado nas etapas de treinamento, testes e validação.

3.2.1 Normalização dos dados

Para obter boas métricas de treinamento de dados, primeiramente deve-se buscar a normalização dos dados, com o objetivo de reduzi-los a uma escala menor e assim, aumentar a eficiência no treinamento das redes. Portanto, após a normalização dos dados, pode-se encontrar de forma mais justa a configuração de rede ideal para o controlador neural.

3.2.2 Configurações de rede

Utilizando o conjunto de dados já obtidos, buscou-se treinar 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. Portanto, a Tabela 5 mostra as configurações utilizadas neste estudo.

Tabela 5 – Parâmetros de controle PID

Rede	Nº de camadas	Neurônios	Funções de ativação	Otimizador
1	5	[100, 100, 100, 100, 100]	[S, R, R, R, R]	Adam
2	5	[100, 100, 100, 100, 100]	[R, S, S, S, S]	Adam
3	5	[10, 20, 50, 100, 200]	[S, R, R, R, R]	Adam
4	5	[10, 20, 50, 100, 200]	[R, S, S, S, S]	Adam
5	5	[200, 100, 50, 20, 10]	[S, R, R, R, R]	Adam
6	5	[200, 100, 50, 20, 10]	[R, S, S, S, S]	Adam
7	3	[100, 100, 100]	[S, S, R]	Adam
8	3	[100, 100, 100]	[R, R, S]	Adam
9	3	[20, 50, 100]	[S, S, R]	Adam
10	3	[20, 50, 100]	[R, R, S]	Adam
11	3	[100, 50, 20]	[S, S, R]	Adam
12	3	[100, 50, 20]	[R, R, S]	Adam
13	3	[100, 100, 100]	[S, S, S]	Adam
14	3	[20, 50, 100]	[S, S, S]	Adam
15	3	[100, 50, 20]	[S, S, S]	Adam
16	3	[100, 100, 100]	[R, R, R]	Adam
17	3	[20, 50, 100]	[R, R, R]	Adam
18	3	[100, 50, 20]	[R, R, R]	Adam

Onde S e R, representam as funções de ativação sigmóide e ReLU, respectivamente. Além disso, a porcentagem de separação de dados de teste ocorreu de maneira diferente em cada grupo de configurações de rede:

- Rede 1 a 6: 0% de dados separados para teste
- Rede 7 a 12: 5% de dados separados para teste
- Rede 13 a 18: 15% de dados separados para teste

Definiu-se grupos com diferentes percentuais de dados separados para teste para avaliá-los de maneira isolada e analisar se é possível que esta variação reproduz interferência nos resultados.

3.3 SIMULAÇÃO DO CONTROLADOR NEURAL

Como forma de entender o comportamento de cada configuração de rede neural treinada, deve-se realizar a simulação do sistema de controle do motor CC. Para tal atividade, utilizou-se o software de simulação para obter os resultados e em seguida, fez-se a comparação de cada modelo de controlador neural para obter a melhor topologia de controlador dos modelos de rede selecionados.

3.3.2 Resposta do sistema

Com o objetivo de avaliar cada modelo treinado em relação aos comportamentos da velocidade do motor CC e do controlador PID, deve-se inicialmente analisar o desempenho dos controladores em relação a entrada degrau unitário. Dessa forma, buscou-se aplicar uma entrada unitária ao sistema, ou seja, 1 rad/s, para avaliar o o comportamento de cada modelo de controlador neural em relação aos requisitos de controle definidos: percentual de sobressinal, tempo de acomodação e acurácia em regime permanente.

Por fim, após encontrar o comportamento de cada modelo, investigou-se o desempenho de cada modelo em relação a resposta do sistema a entradas com características não determinísticas. Portanto, por meio dessa análise, avaliou-se a rede neural que apresentou melhor comportamento e fez-se a comparação com o desempenho do sistema controlado via PID.

4 ANÁLISE DE DADOS

Nesta seção os resultados obtidos pelo treinamento das redes definidas são analisados por meio do software de simulação Matlab e comparados com cada configuração de rede, juntamente com a aplicação do controlador PID. Este capítulo expressa, primeiramente, os resultados obtidos no treinamento das redes neurais, evidenciando as métricas adotadas para a análise de cada configuração de rede. Então, a resposta de cada modelo de controlador em relação a resposta do sistema ao degrau unitário é analisada, e por fim, os resultados obtidos por meio de simulação de entradas aleatórias.

4.1 TREINAMENTO DAS REDES NEURAIS

Por meio da definição das configurações de rede a serem estudadas, pode-se avaliar o comportamento de cada modelo treinado. Dessa forma, obtém-se nesta etapa os resultados do treinamento de cada configuração de rede.

4.1.1 Análise das métricas

Após realizar o treinamento de cada rede, deve-se analisar os resultados relacionados às métricas de cada rede. A Tabela 6 mostra os resultados de cada topologia de rede obtidos em relação ao erro quadrático médio e ao erro médio absoluto.

Tabela 6 – 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

Onde MSE e MAE representam o erro quadrático médio e o erro médio absoluto, respectivamente.

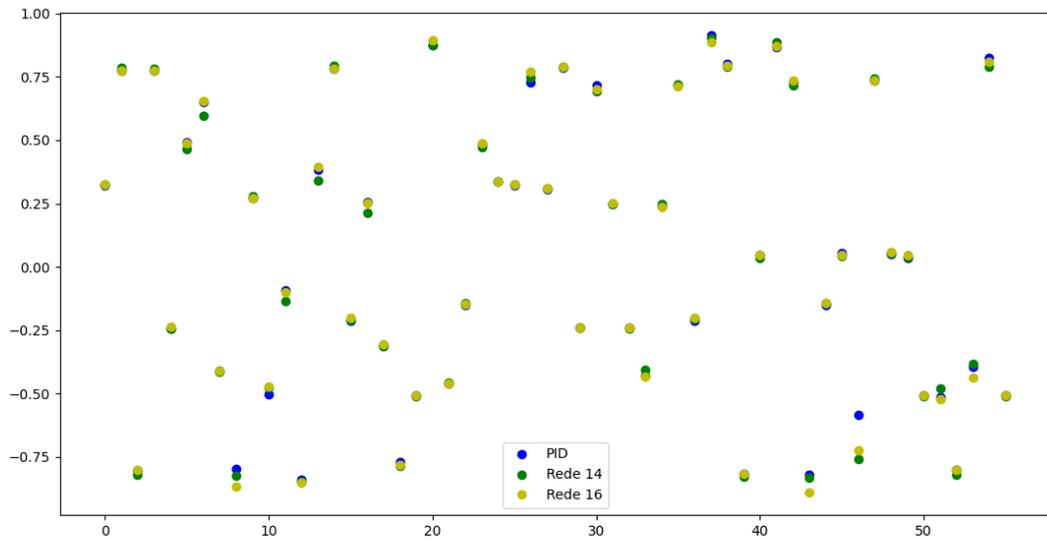
As redes que possuem domínio de função de ativação ReLU, apresentam métricas menores em relação às redes com predominância de funções sigmóide. Portanto, pode-se concluir que as funções de ativação ReLU, diminuem de maneira acentuada tanto o erro médio quadrático e o erro médio absoluto.

4.1.2 Dados de teste

Com o objetivo de avaliar a influência das métricas em relação aos dados de teste, pode-se comparar as redes que apresentaram maiores diferenças de erro quadrático médio e erro médio absoluto entre si. Portanto, vê-se a necessidade de avaliar as redes 14 e 16 que apresentaram erro quadrático médio equivalente a 0,0011 e 0,0004, respectivamente em relação aos dados de teste, com o intuito de aferir se essa diferença de métricas se destingue de maneira relevante aos dados de teste.

A Figura 22 mostra um gráfico comparativo entre os dados de teste das redes 14 e 16, juntamente com os dados da saída PID, que representam os valores alvo.

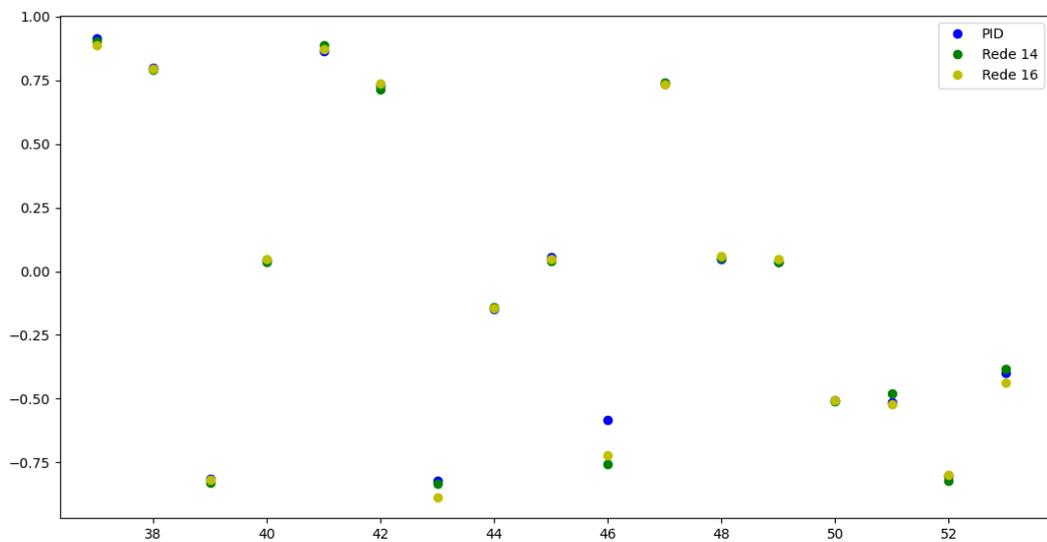
Figura 22 – Dados de teste simulados com as redes 14 e 16.



Fonte: Autor

Portanto, pode-se notar que as saídas geradas pelas redes 14 e 16, apresentam valores similares entre si e em relação aos valores gerados pelo controlador PID, cujo qual, são os dados que se deseja atingir. Contudo, em alguns pontos, nota-se a diferença desses valores em relação aos valores gerados pelo controlador PID, conforme a Figura 23, que mostra a ampliação de uma região dos dados.

Figura 23 – Dados de teste simulados com as redes 14 e 16 (ampliado).



Fonte: Autor

Analisando o gráfico da Figura 23 observa-se que em alguns pontos os valores gerados pela rede 14 possuem maiores proximidades em relação aos valores do controlador PID do que os valores gerados pela rede 16, cujo qual, apresentam erros menores que a rede 14. Portanto, pode-se concluir que as métricas geradas pelos treinamentos dessas duas redes, não apresentam diferença elevada entre si.

4.1.3 Resultados dos treinamentos

Nesta etapa, foram obtidas as respostas dos treinamentos de cada configuração de rede e comparadas umas com as outras em relação com o comportamento do controlador PID. Dessa forma, divide-se as análises em três grupos, sendo que em cada grupo, contém-se 6 configurações de rede.

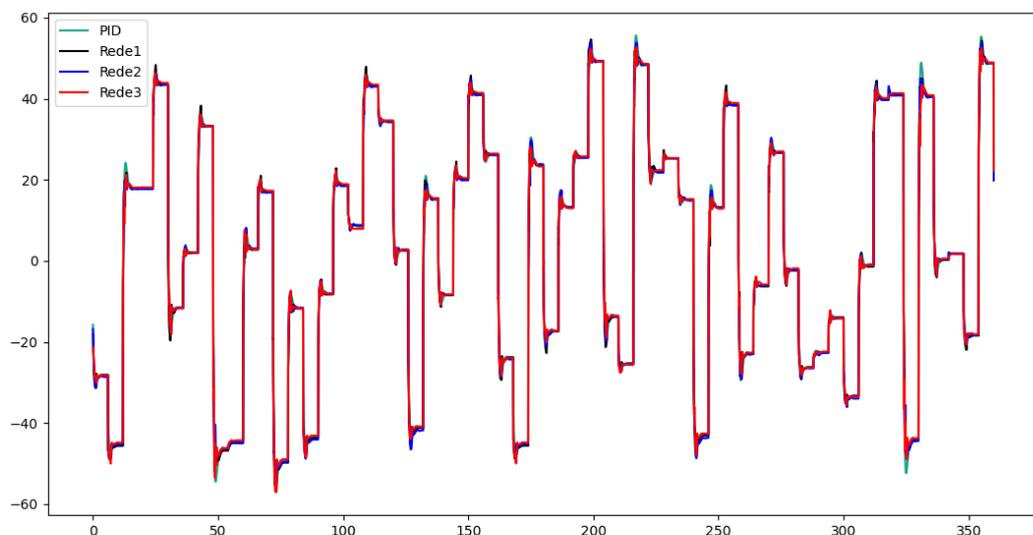
4.1.3.1 Primeiro grupo

O resultado obtido por meio do treinamento das redes 1, 2 e 3, é mostrado na Figura 24. De forma a relembrar as funções de ativação utilizadas, a Tabela 7 informa as configurações de cada topologia de rede.

Tabela 7 – Parâmetros de rede do grupo 1 de modelos.

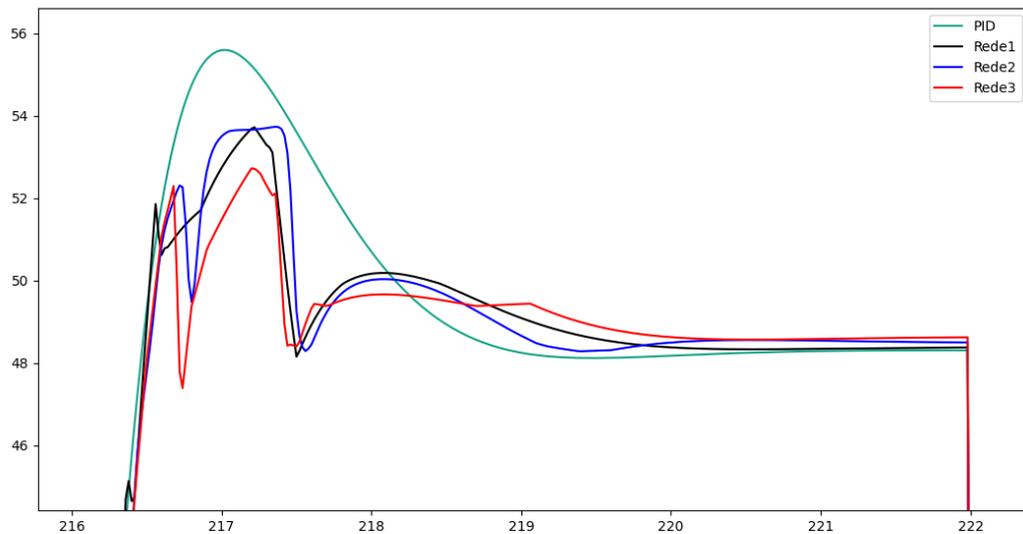
Rede	Neurônios	Funções de ativação
1	[100, 100, 100, 100, 100]	[S, R, R, R, R]
2	[100, 100, 100, 100, 100]	[R, S, S, S, S]
3	[10, 20, 50, 100, 200]	[S, R, R, R, R]

Figura 24 – Resultado do treinamento das redes 1 a 3.



Observa-se de maneira macro, a saída gerada pelas redes possui comportamentos similares. Contudo, deve-se observar de forma ampliada o comportamento específico de cada rede, dessa forma, a Figura 25 mostra o desempenho das redes de maneira detalhada.

Figura 25 – Resultado do treinamento das redes 1 a 3 (ampliado).



Fonte: Autor

Nota-se que o comportamento obtido pela rede 1 apresenta oscilações pouco agressivas e em certos pontos indicam suavidade, assim como, identifica-se que a rede em questão gera valor máximo relativamente próximo do esperado, bem como o valor final apresenta similaridade com o valor obtido pelo controlador PID. O comportamento oscilatório com característica mais suave, é obtido possivelmente por causa da primeira camada oculta conter a função de ativação sigmóide e a precisão no valor final, bem como alguns pontos de oscilações mais agressivas são causadas pela predominância da função ReLU.

A rede 2 também apresenta comportamentos oscilatórios com tendência oscilatória, porém em alguns pontos mais agressivos, além disso, possui acurácia em valor final. Percebe-se que pela primeira camada possuir função de ativação ReLU, a mesma busca atingir de maneira agressiva o valor desejado, contudo, as camadas ocultas seguintes possuem função sigmóide, o que torna o comportamento mais suave.

A rede 3 possui comportamento oscilatório e agressivo e por conter neurônios crescentes e predominância na função ReLU, torna o desempenho dessa configuração inviável, por apresentar performance com altas oscilações. Portanto, pode-se concluir

por meio desse grupo, que a predominância das funções de ativação ReLU, tornam a configuração da rede mais oscilatória e sem suavidade.

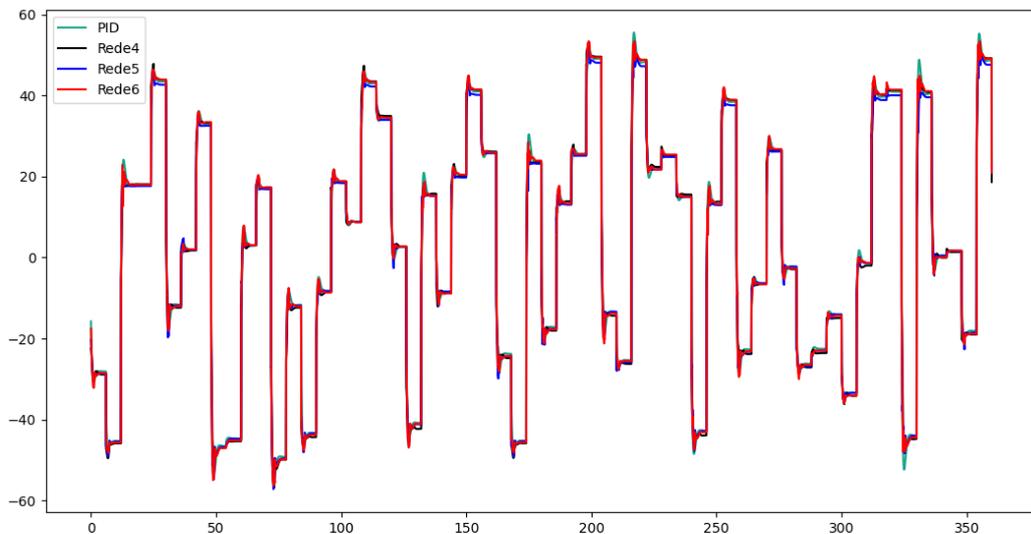
4.1.3.2 Segundo grupo

O resultado obtido por meio do treinamento das redes 4, 5 e 6, é mostrado na Figura 26. De forma a relembrar as funções de ativação utilizadas, a Tabela 8 apresenta as funções utilizadas nesse grupo de modelos.

Tabela 8 – Parâmetros de rede do grupo 2 de modelos.

Rede	Neurônios	Funções de ativação
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]

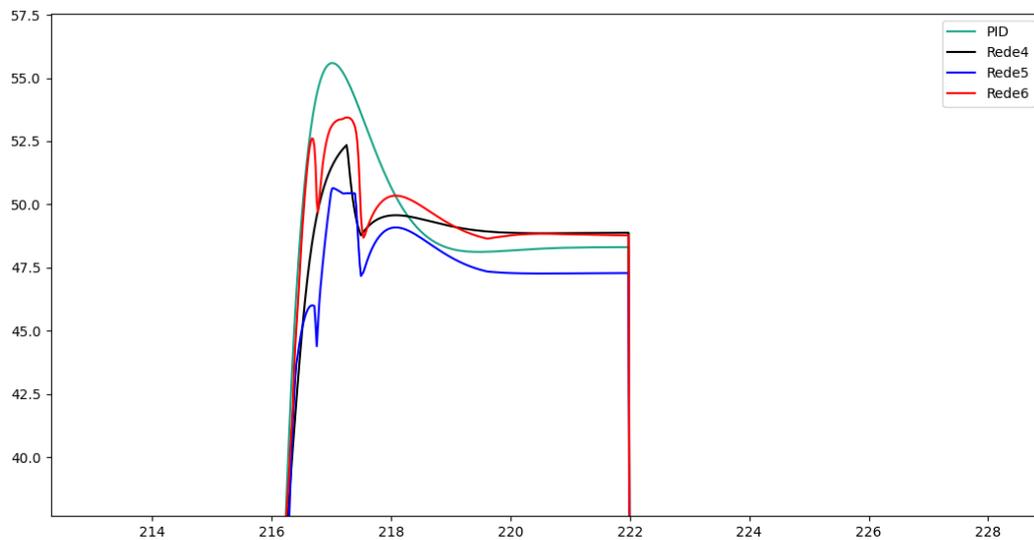
Figura 26 – Resultado do treinamento das redes 4 a 6.



Fonte: Autor

Observa-se que de maneira macro, a saída gerada pelas redes possui comportamentos similares. Contudo, é necessário observar de forma ampliada o comportamento das saídas, dessa forma, a Figura 27 mostra o desempenho das redes de maneira detalhada.

Figura 27 – Resultado do treinamento das redes 4 a 6 (ampliado).



Fonte: Autor

Portanto, nota-se que o comportamento gerado pela rede 4 apresenta valor final similar ao desejado, além disso, não possui oscilações e com suavidade nos sinais, tal comportamento suave pode ser justificado pela predominância da função de ativação sigmóide e a aproximação dos valor final, justifica-se devido a função ReLU na primeira camada oculta.

Já a rede 5, não apresenta precisão em relação ao valor final e contém variações ao longo de seu desempenho, além disso, quando o comportamento do controlador PID apresenta o seu valor máximo, o valor de pico da rede 5 possui diferença de aproximadamente 5V, o que torna essa configuração inviável. Pode-se notar que pela primeira camada oculta conter a função de ativação sigmóide, o desempenho dessa configuração gera imprecisão no valor final e a predominância da função de ativação ReLU, torna o comportamento da saída oscilatório.

A rede 6 apresenta característica oscilatória, contudo, as oscilações manifestam-se de maneira mais suave, além disso, para essa configuração, o valor final possui precisão em relação a resposta do controlador PID. A precisão pode ser justificada devido a primeira camada oculta conter a função de ativação ReLU e as oscilações suaves, ocorrem devido a predominância das funções sigmóide.

4.1.3.3 Terceiro grupo

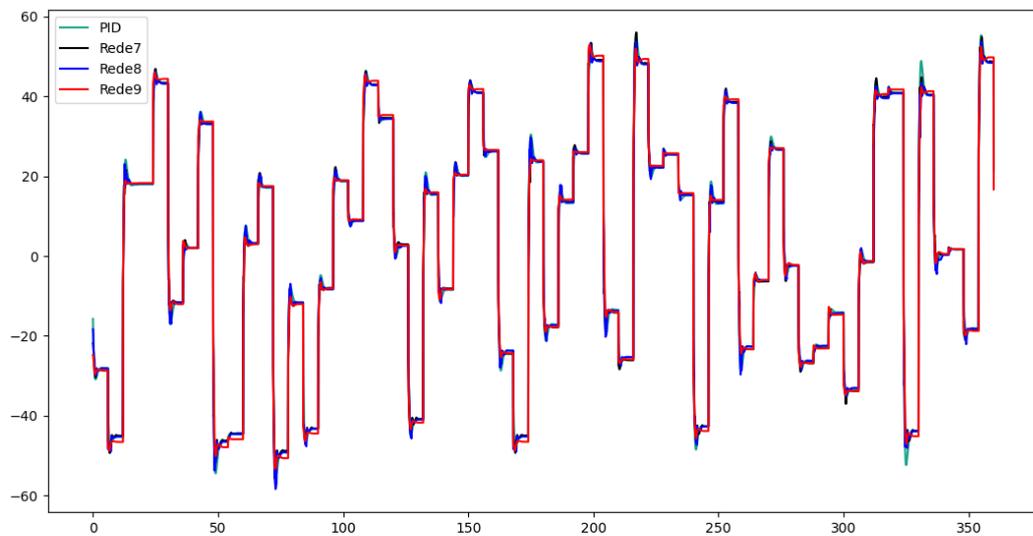
O resultado obtido por meio do treinamento das redes 7, 8 e 9, é mostrado na Figura 28. De forma a relembrar as funções de ativação utilizadas, a Tabela 9 apresenta

as funções utilizadas nesse grupo de modelos.

Tabela 9 – Parâmetros de rede do grupo 3 de modelos.

Rede	Neurônios	Funções de ativação
7	[100, 100, 100]	[S, S, R]
8	[100, 100, 100]	[R, R, S]
9	[20, 50, 100]	[S, S, R]

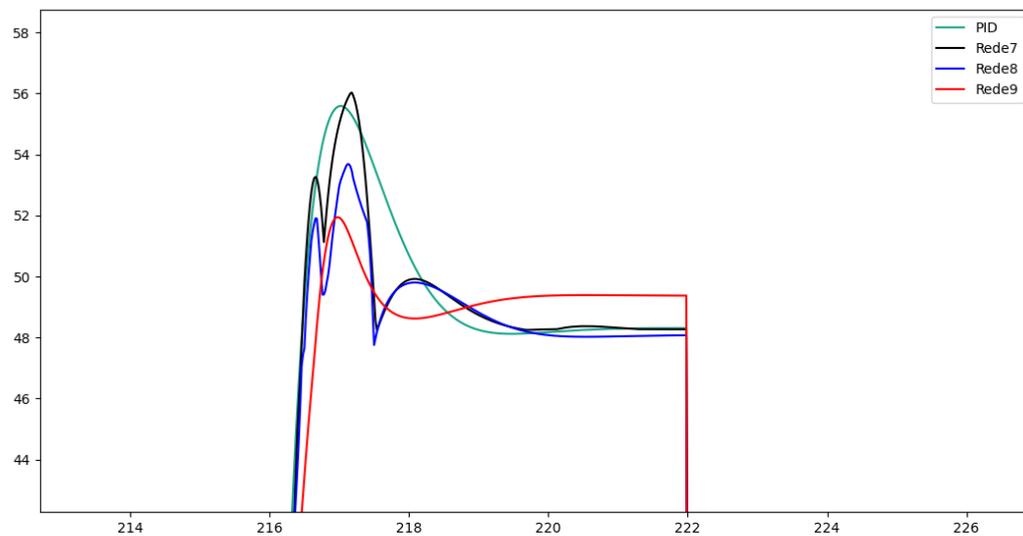
Figura 28 – Resultado do treinamento das redes 7 a 9.



Fonte: Autor

Observa-se que a saída gerada pelas redes possui comportamentos similares ao apresentado pelo controlador PID. Contudo, deve-se observar de forma ampliada o comportamento das saídas, dessa forma, a Figura 29 mostra o desempenho das redes de maneira detalhada.

Figura 29 – Resultado do treinamento das redes 7 a 9 (ampliado).



Fonte: Autor

Observa-se que o comportamento da rede 7 apresenta oscilações, contudo, com desempenho mais suave, além disso, possui acurácia tanto no valor máximo quanto no valor final em relação aos mesmos instantes de tempo obtidos pelo controlador PID. Pode-se concluir, analisando esse ponto específico, que a rede 7 apresenta comportamento adequado, isso pode ser gerado por essa configuração de rede apresentar menor número de camadas, bem como maior número de funções de ativação sigmóide, concedendo apenas a uma função ReLU, na última camada, o cargo de atingir a exatidão no valor desejado.

A rede 8 mostra um comportamento oscilatório similar ao apresentado pela rede 7, contudo não apresenta a mesma precisão nos valores finais e máximo, conforme gerado pela rede 7. A rede 9, não apresenta oscilações e demonstrou copiar o comportamento do controlador PID, contudo, sem precisão nos valores final e máximo, tal comportamento pode-se justificar pela predominância da função sigmoid.

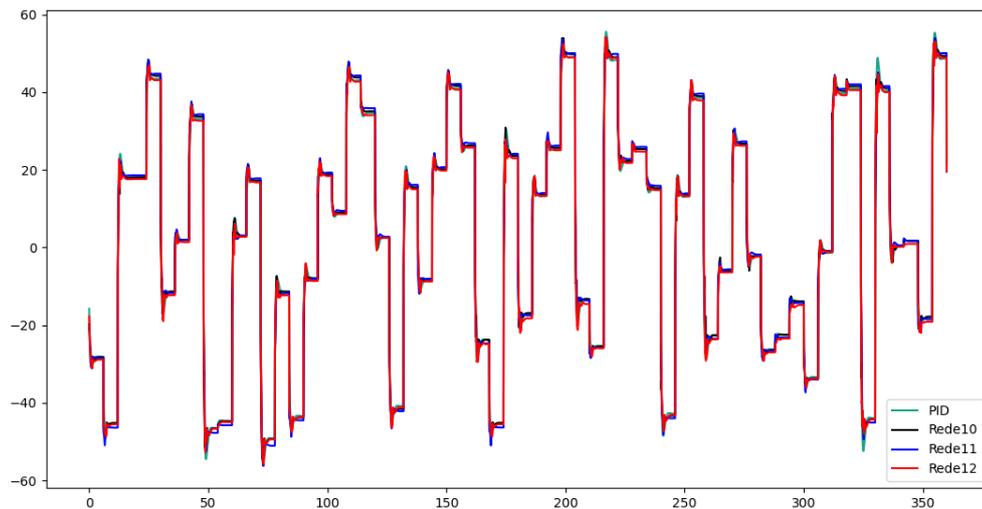
4.1.3.4 Quarto grupo

O resultado obtido por meio do treinamento das redes 10, 11 e 12, é demonstrado na Figura 30. De forma a relembrar as funções de ativação utilizadas, a Tabela 10 apresenta as funções utilizadas nesse grupo de modelos.

Tabela 10 – Parâmetros de rede do grupo 4 de modelos.

Rede	Neurônios	Funções de ativação
10	[20, 50, 100]	[R, R, S]
11	[100, 50, 20]	[S, S, R]
12	[100, 50, 20]	[R, R, S]

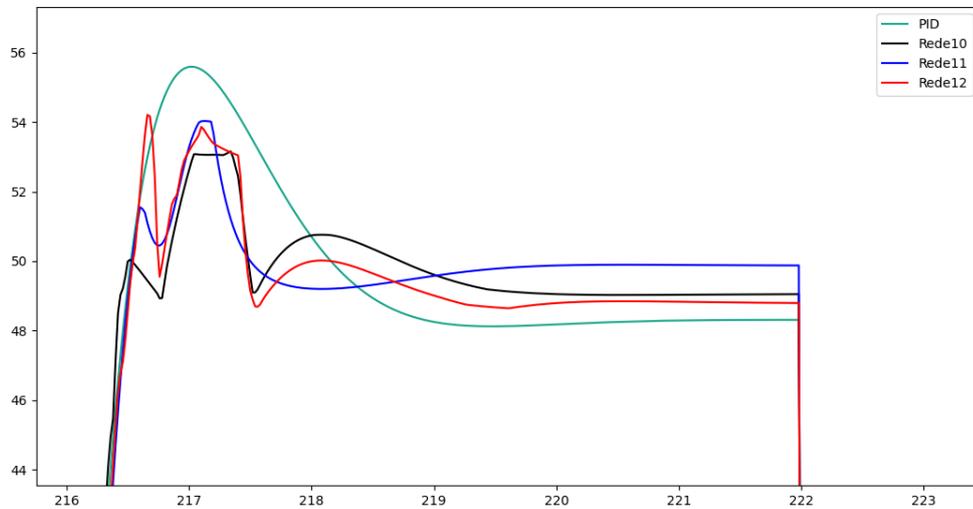
Figura 30 – Resultado do treinamento das redes 10 a 12.



Fonte: Autor

A Figura 31 demonstra o desempenho das redes de maneira detalhada.

Figura 31 – Resultado do treinamento das redes 10 a 12.



Fonte: Autor

Nota-se que as redes 10, 11 e 12 apresentam comportamentos com altas oscilações, evidenciando o desempenho da rede 12 em que possui mudanças de valores elevadas. Além disso, observa-se que em todas as configurações de rede desse grupo, não há precisão tanto no valor final, como nos valores máximos.

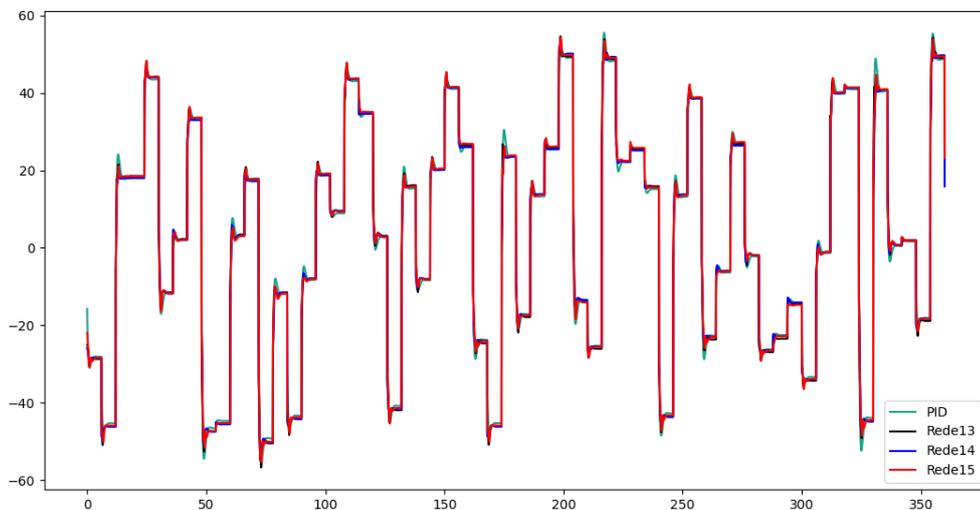
4.1.3.5 Quinto grupo

O resultado obtido por meio do treinamento das redes 13, 14 e 15, é demonstrado na Figura 32. De forma a relembrar as funções de ativação utilizadas, a Tabela 11 apresenta as funções utilizadas nesse grupo de modelos.

Tabela 11 – Parâmetros de rede do grupo 5 de modelos.

Rede	Neurônios	Funções de ativação
13	[100, 100, 100]	[S, S, S]
14	[20, 50, 100]	[S, S, S]
15	[100, 50, 20]	[S, S, S]

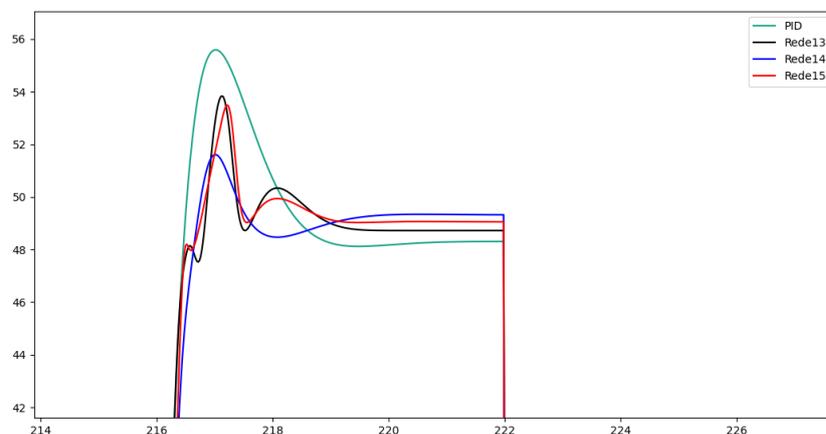
Figura 32 – Resultado do treinamento das redes 13 a 15.



Fonte: Autor

A Figura 33 demonstra o desempenho das redes de maneira detalhada.

Figura 33 – Resultado do treinamento das redes 13 a 15.



Fonte: Autor

É possível notar que as redes 13 e 15 apresentam comportamentos similares, não havendo diferença significativa na distribuição de neurônios. Como as configurações de rede contêm em todas as camadas neurais, a função de ativação sigmoide, nota-se que a resposta gerada por essas redes apresentam características de oscilação suave, não sobrepondo os valores desejados, contudo, não apresentam precisão nos valores final e máximo em relação ao comportamento do controlador PID.

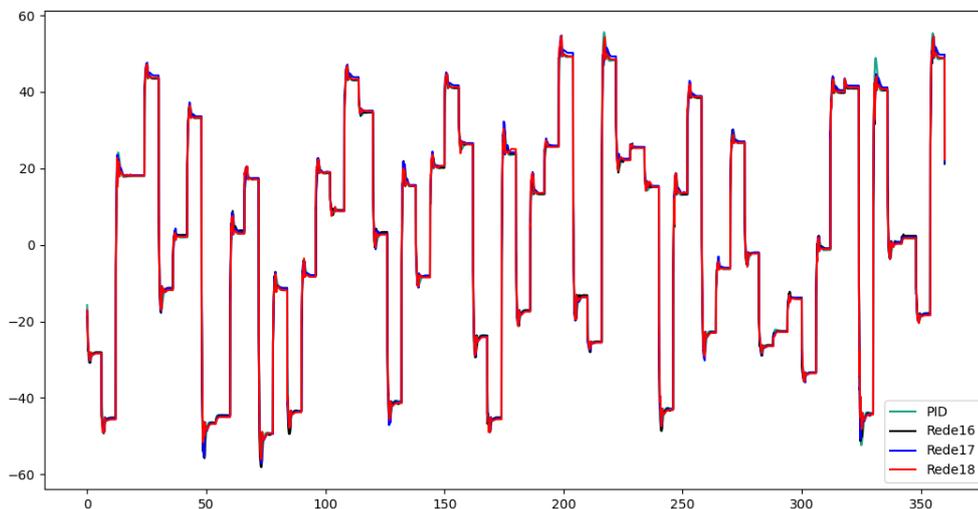
4.1.3.6 Sexto grupo

O resultado obtido por meio do treinamento das redes 16, 17 e 18, é demonstrado na Figura 34. De forma a relembrar as funções de ativação utilizadas, a Tabela 12 apresenta as funções utilizadas nesse grupo de modelos.

Tabela 12 – Parâmetros de rede do grupo 6 de modelos.

Rede	Neurônios	Funções de ativação
16	[100, 100, 100]	[R, R, R]
17	[20, 50, 100]	[R, R, R]
18	[100, 50, 20]	[R, R, R]

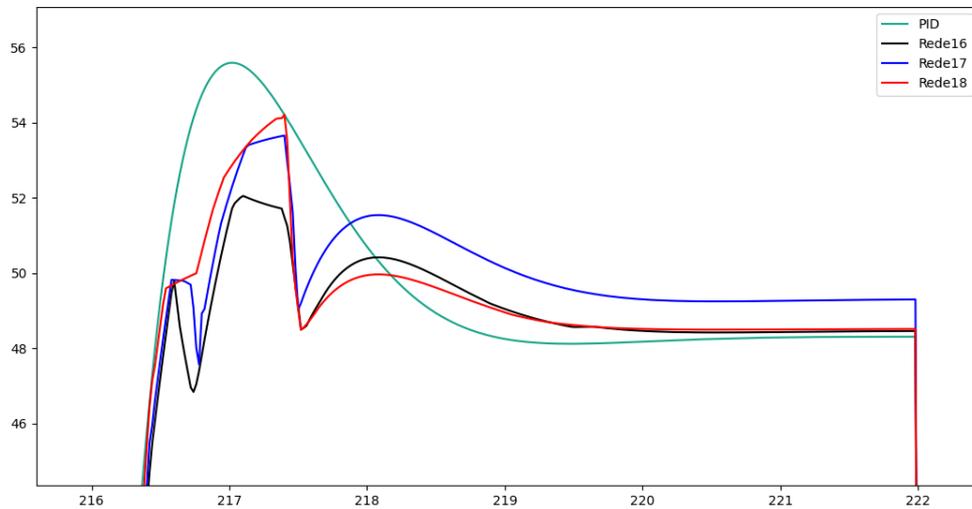
Figura 34 – Resultado do treinamento das redes 16 a 18.



Fonte: Autor

A Figura 35 demonstra o desempenho das redes de maneira detalhada.

Figura 35 – Resultado do treinamento das redes 16 a 18.



Fonte: Autor

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 não ocorre acurácia em vaalor final. Contudo, a rede 18 gera oscilações mais suaves e, juntamente com a rede 16, apresentam resposta mais próxima em regime permanente do valor desejado.

Portanto, pode-se concluir por meio desse grupo que utilizou somente a função ReLU em suas camadas neurais, que o comportamento apresentado da função ReLU apresenta caráter mais agressivo ao buscar atingir os valores desejados. Pode-se notar que em certo tempo sem oscilação dos valores do controlador PID, as redes com funções ReLU tendem a atingir com maior proximidade em arelação ao valor final desejado.

4.2 OBTENÇÃO DE DADOS DE SIMULAÇÃO

Nesta etapa, obteve-se o modelo treinado de cada configuração de rede com o objetivo de utiliza-lo em simulação no software Matlab. Dessa forma, tem-se como intuito, analisar incialmente a resposta do controlador e do sistema a entrada degrau unitário e por fim, avaliar a resposta a entradas aleatórias.

4.2.1 Resposta ao degrau unitário

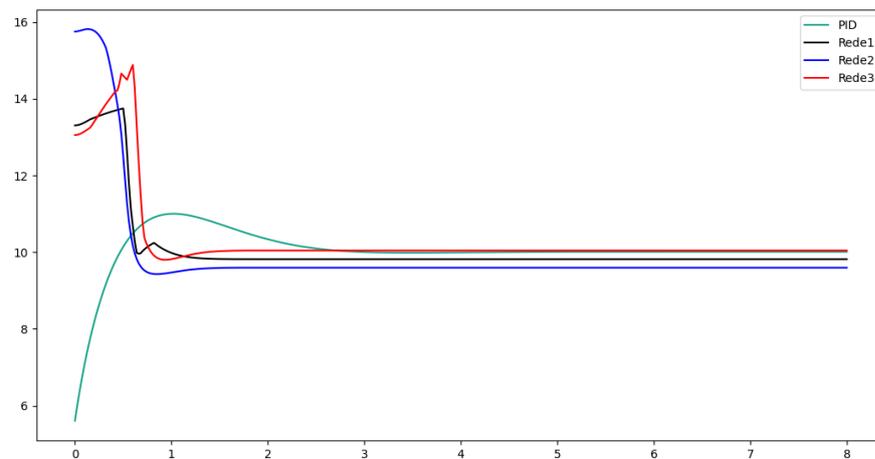
Neste estágio de simulação, pretende-se observar a resposta do sistema ao degrau unitário em relação ao desempenho do controlador neural e da resposta

da velocidade do motor CC. Portanto, observou-se o comportamento de cada rede dividindo-as em 6 grupos de 3 configurações de rede cada.

4.2.1.1 Primeiro grupo

A resposta do controlador obtido por meio do modelos das redes 1, 2 e 3 gerados, é demonstrado na Figura 36.

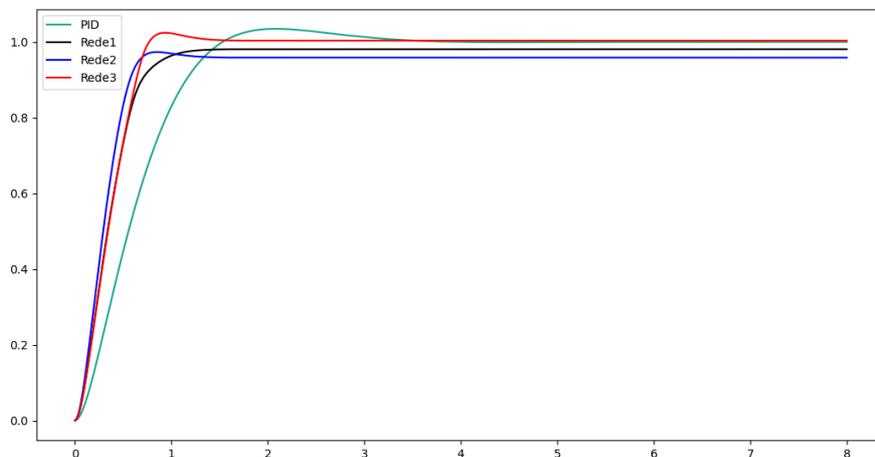
Figura 36 – Saída do controlador ao degrau unitário das redes 1 a 3.



Fonte: Autor

A Figura 37 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária.

Figura 37 – Resposta do sistema ao degrau unitário das redes 1 a 3.



Fonte: Autor

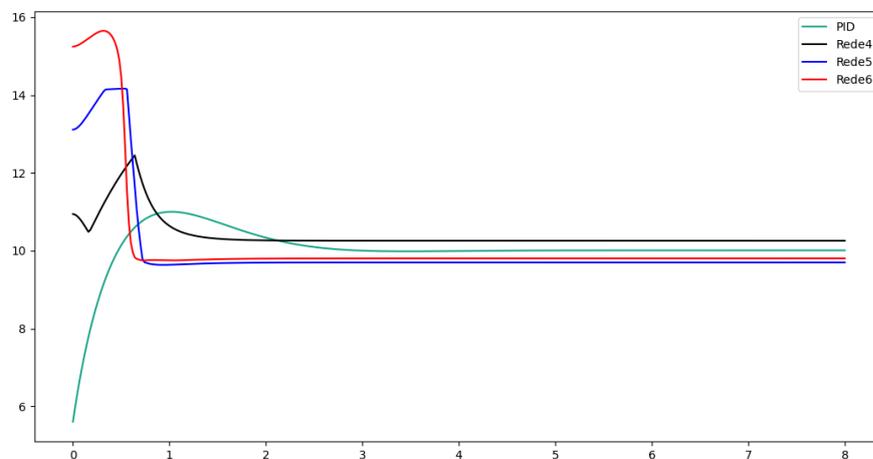
Percebe-se que as saídas geradas pelo controlador neural geram respostas mais rápidas em relação ao comportamento do controlador PID. Além disso, nota-se que a resposta do sistema ao degrau unitário para a rede 3 apresentou menor erro em regime permanente em relação ao valor desejado, tal efeito foi gerado devido ao sinal de controle dessa rede possuir valores similares aos obtidos pelo controlador PID.

As redes 1 e 2 apresentaram menor acurácia no valor final, devido também ao sinal de controle para ambas as redes não apresentarem comportamentos similares em regime permanente. Portanto, nota-se que para obter a velocidade final mais próxima da desejada, o sinal de controle deve ser próximo do controlador PID e quanto mais alta a tensão, mais rápido o sistema tende a convergir para o valor final, conforme observado pelo desempenho da rede 3.

4.2.1.2 Segundo grupo

A resposta do controlador obtido por meio do modelos das redes 4, 5 e 6 gerados, é demonstrado na Figura 38.

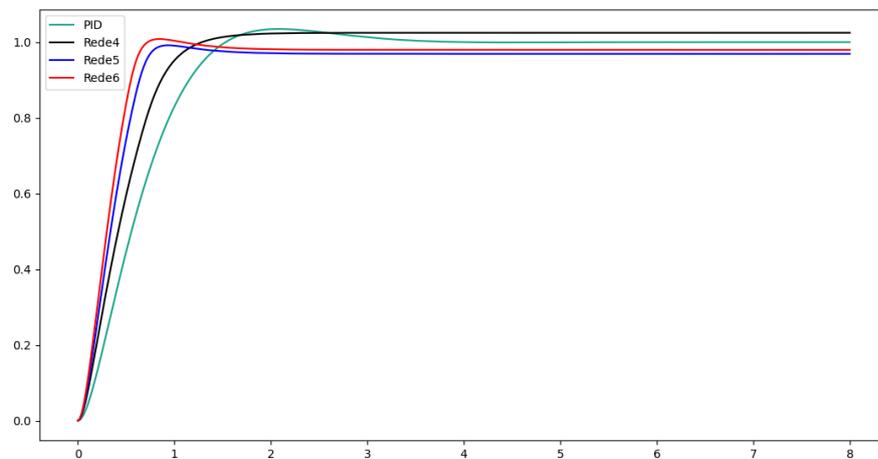
Figura 38 – Saída do controlador ao degrau unitário das redes 4 a 6.



Fonte: Autor

A Figura 39 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária.

Figura 39 – Resposta do sistema ao degrau unitário das redes 4 a 6.



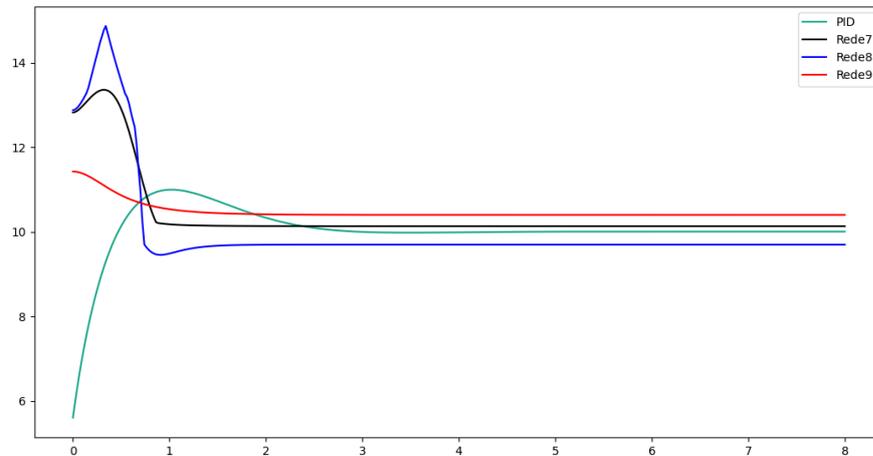
Fonte: Autor

Nota-se que o comportamento da rede 4 apresenta resposta mais lenta em relação ao desempenho das redes 5 e 6, contudo, indica maior acurácia no valor final. Como a rede 5 e 6, apresentam valores de tensões iniciais maiores, a resposta desses sistemas indicam maiores velocidades na obtenção do valor final, contudo, somente a rede 4 apresenta precisão no valor final, devido ao sinal de controle em regime permanente possuir valores próximos ao controlador PID.

4.2.1.3 Terceiro grupo

A resposta do controlador obtido por meio do modelos das redes 7, 8 e 9 gerados, é demonstrado na Figura 40.

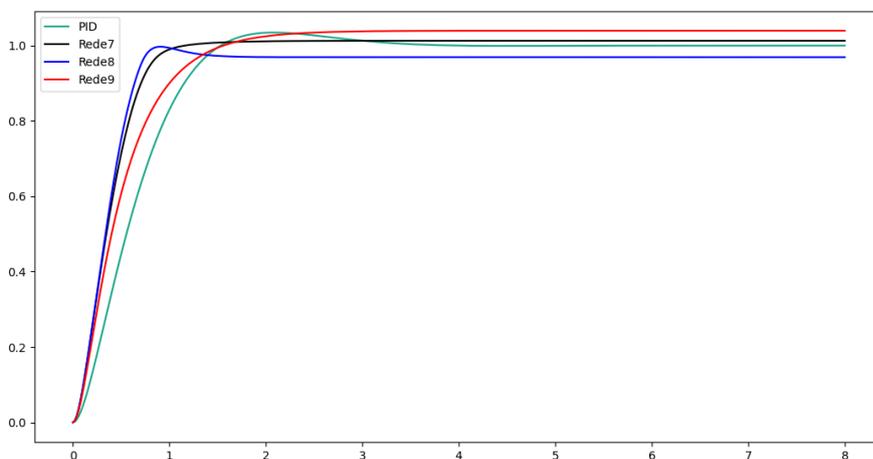
Figura 40 – Resposta do controlador ao degrau unitário das redes 7 a 9.



Fonte: Autor

A Figura 41 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária.

Figura 41 – Resposta do sistema ao degrau unitário das redes 7 a 9.



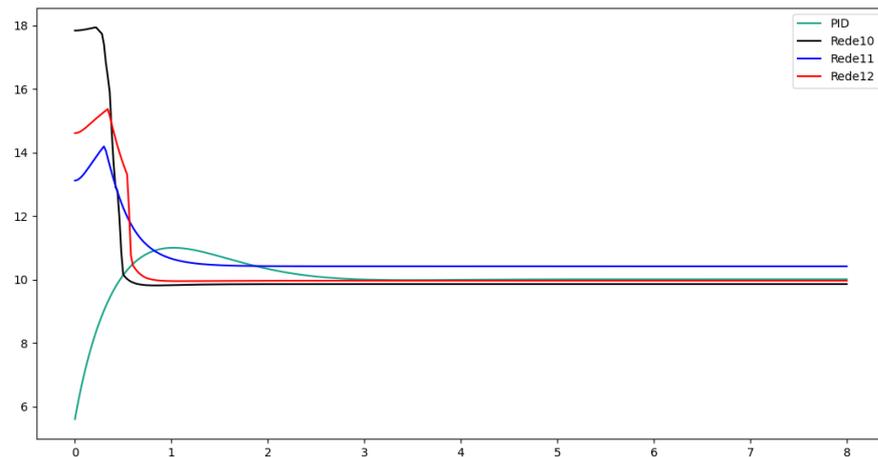
Fonte: Autor

A rede 7 apresenta precisão no valor final devido aos valores em regime permanente do controlador neural dessa configuração possuírem similaridade aos valores gerados pelo controlador PID, o tempo de obtenção do valor final também possui alta velocidade de resposta. Contudo, as redes 8 e 9 apresentam valores em estado estacionário diferentes dos obtidos pelo controlador PID, além disso, a rede 9 apresenta lentidão para atingir o valor final.

4.2.1.4 Quarto grupo

A resposta do controlador obtido por meio do modelos das redes 10, 11 e 12 gerados, é demonstrado na Figura 42.

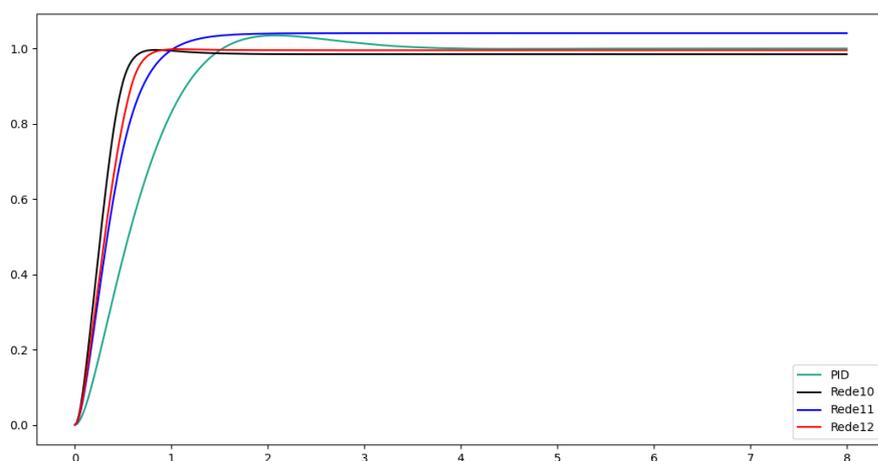
Figura 42 – Resposta do controlador ao degrau unitário das redes 10 a 12.



Fonte: Autor

A Figura 43 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária.

Figura 43 – Resposta do sistema ao degrau unitário das redes 10 a 12.



Fonte: Autor

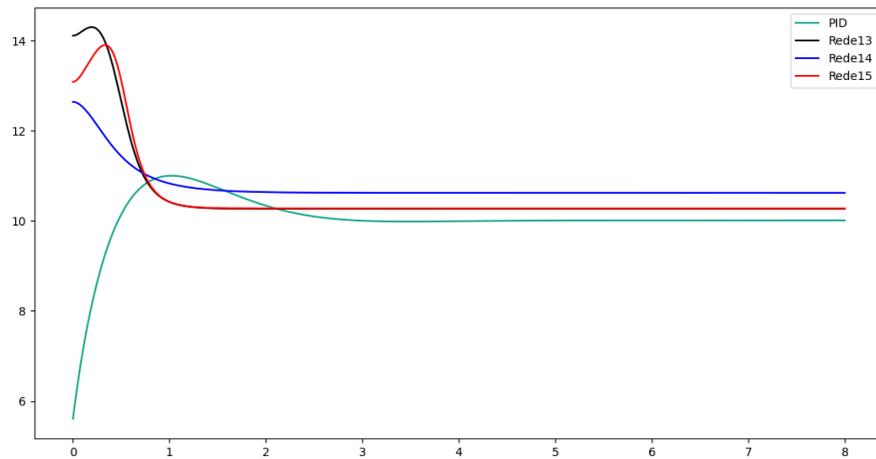
As redes 10 e 12 apresentam elevadas velocidades para atingir o valor em estado estacionário, bem como alta precisão em valor final ao se comparar com os

valores obtidos pelo controlador PID. Contudo, a resposta obtida pela rede 11 apresenta baixa acuracidade, bem como resposta mais lenta que as demais redes desse grupo.

4.2.1.5 Quinto grupo

A resposta do controlador obtido por meio do modelos das redes 13, 14 e 15 gerados, é demonstrado na Figura 44.

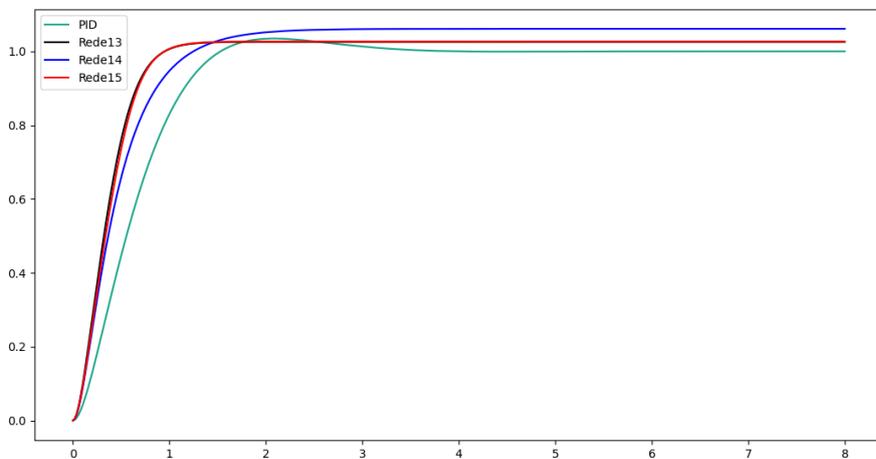
Figura 44 – Resposta do controlador ao degrau unitário das redes 13 a 15.



Fonte: Autor

A Figura 45 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária.

Figura 45 – Resposta do sistema ao degrau unitário das redes 13 a 15.



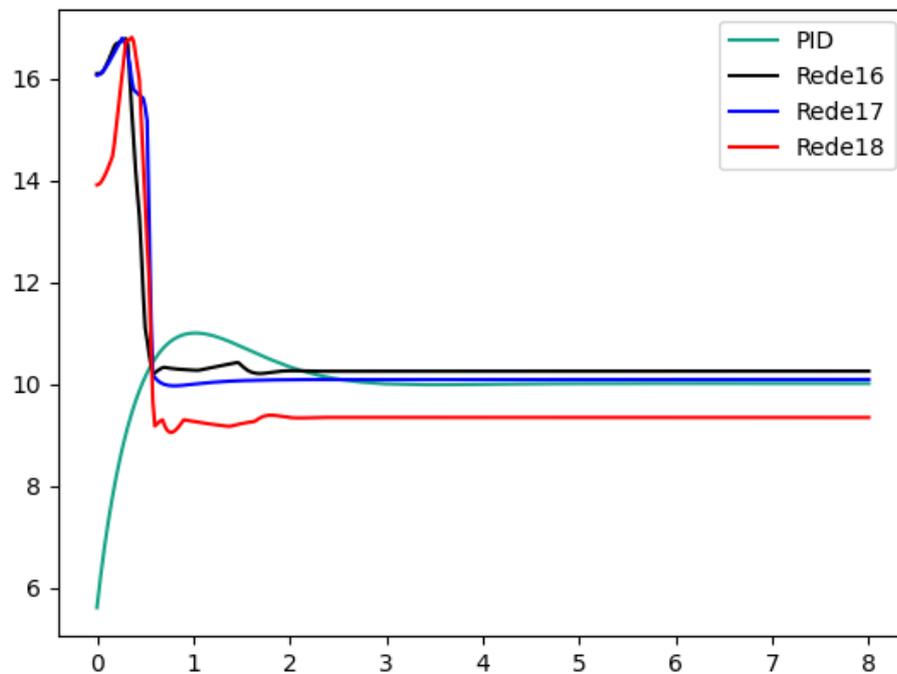
Fonte: Autor

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.

4.2.1.6 Sexto grupo

A resposta do controlador obtido por meio do modelos das redes 16, 17 e 18 gerados, é demonstrado na Figura 46.

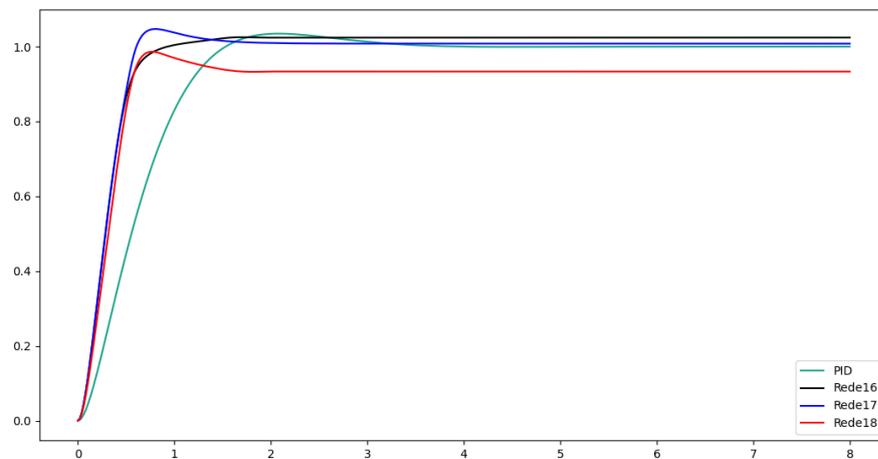
Figura 46 – Resposta do controlador ao degrau unitário das redes 16 a 18.



Fonte: Autor

A Figura 47 demonstra o desempenho das redes em relação a resposta do sistema a entrada unitária.

Figura 47 – Resposta do sistema ao degrau unitário das redes 16 a 18.



Fonte: Autor

Nota-se que a rede 18 não apresenta proximidade aos valores desejados em regime permanente, porém com alta velocidade de resposta. As redes 16 e 17, demonstram alta velocidade de resposta e valores similares ao apresentado pelo controlador PID em regime permanente.

4.2.2 Requisitos de desempenho de controle

Para analisar quantitativamente cada configuração de rede, deve-se analisar se cada topologia atingiu o requisito de controle desejado. A Tabela 13 mostra os resultados obtidos.

Tabela 13 – Métricas 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

Algumas redes não apresentaram resposta que com sobressinal, devido a não possuírem comportamento de um sistema de segunda ordem subamortecida. Observa-se que somente as respostas obtidas pelas redes 14 e 18 apresentaram valores distintos dos requisitos de controle esperados. Tais resultados foram obtido devido imprecisão na resposta em regime permanente.

Além disso, observou-se que para todas as configurações de rede, o sinal do controlador inicia em valores altos de tensão e então o mesmo diminui. Tal efeito se deve ao fato do conjunto de dados de treinamento possuir, para a entrada unitária, oscilação de decaimento da tensão.

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.

4.2.3 Resposta do sistema a entradas aleatórias

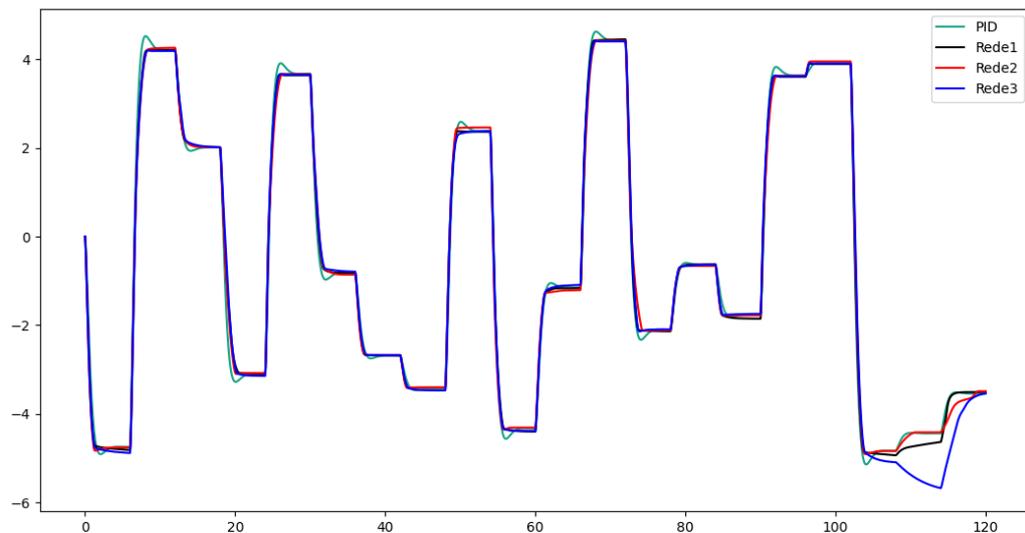
Nesta etapa testou-se as 18 configurações de rede aplicando entradas aleatórias durante 120 segundos, gerando 20 diferentes entradas durante todo o ciclo. Dessa forma, divide-se as análises em 6 grupos de 3 configurações cada, com o

objetivo de avaliar cada topologia.

4.2.3.1 Primeiro grupo

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 1, 2 e 3 gerados, é demonstrado na Figura 48.

Figura 48 – Resposta do controlador a entradas aleatórias das redes 1 a 3.



Fonte: Autor

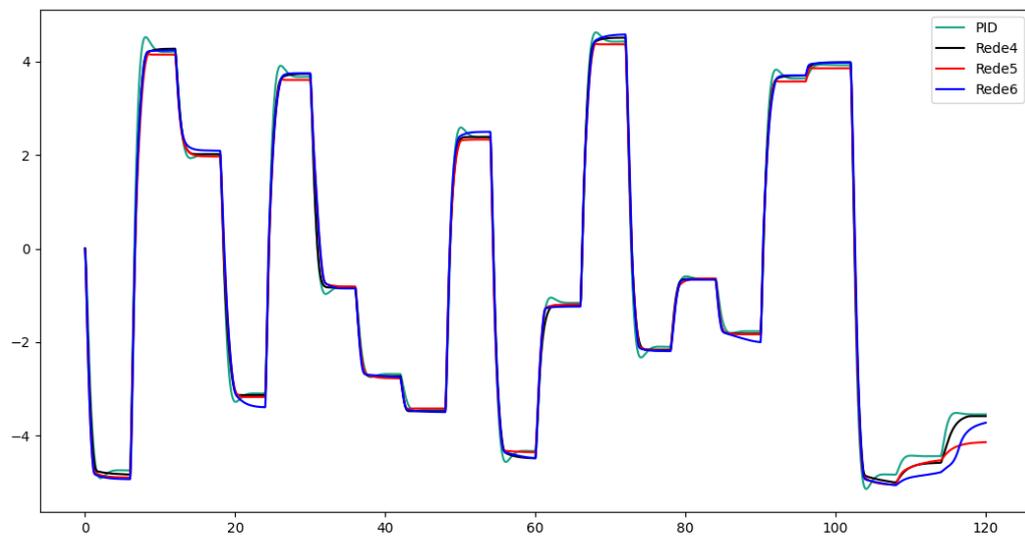
Em basicamente todas as partes do tempo as configurações desse grupo apresentam similaridade ao comportamento apresentado pelo controlador PID. A rede 1 demonstra valores relativamente próximos aos obtidos pelo controlador PID, contudo, na penúltima variação de entrada apresenta valor mais distante do desejado.

As redes 2 e 3, apresentam em determinados pontos, imprecisão no valor final em relação ao valor desejado. A rede 2, contém lentidão na obtenção do valor final para a última entrada, porém, a rede 3 demonstra imprecisão nas 3 últimas variações de entrada.

4.2.3.2 Segundo grupo

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 4, 5 e 6 gerados, é demonstrado na Figura 49.

Figura 49 – Resposta do controlador a entradas aleatórias das redes 4 a 6.



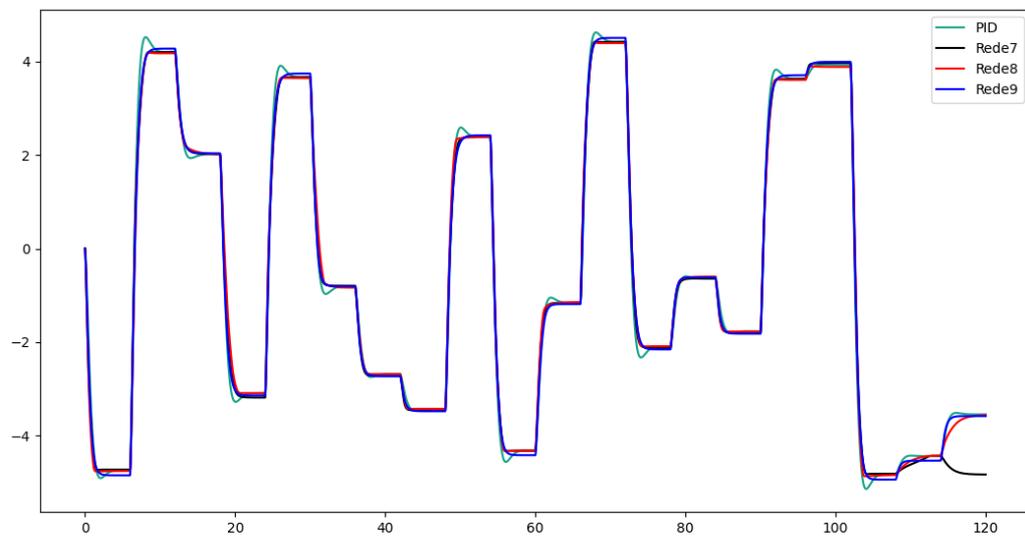
Fonte: Autor

A rede 6 apresenta variação elevada em diversos pontos na resposta do sistema em relação aos valores desejados. Em alguns pontos, as redes 4 e 5 demonstram imprecisão, contudo, menor que a métrica obtida pela rede 6. Todas as configurações de rede desse grupo demonstram imprecisão nos valores das 3 últimas entradas.

4.2.3.3 Terceiro grupo

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 7, 8 e 9 gerados, é demonstrado na Figura 50.

Figura 50 – Resposta do controlador a entradas aleatórias das redes 7 a 9.



Fonte: Autor

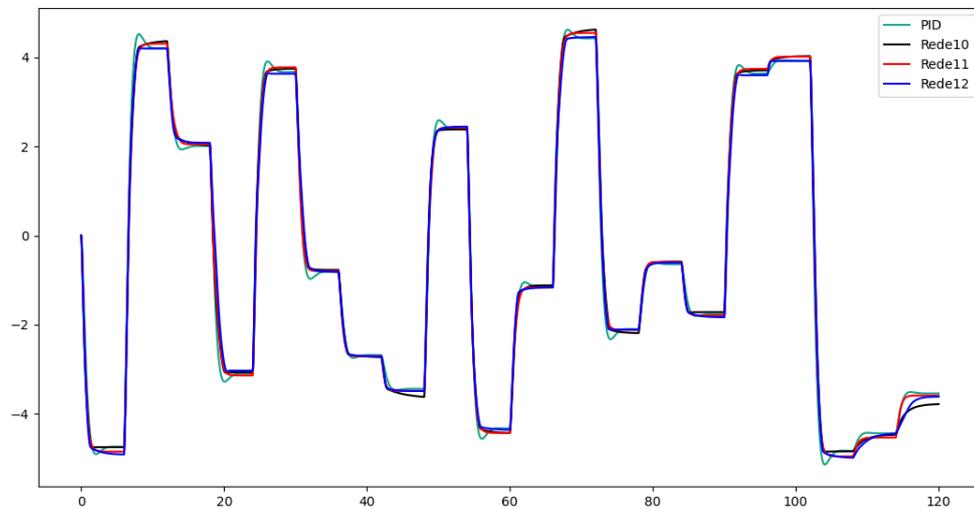
Em determinadas partes de todo o ciclo, a rede 9 apresenta imprecisão nos valores de velocidade em estado estacionário em relação aos valores gerados pelo controlador PID. O comportamento dessa topologia apresenta melhor resposta em relação as demais redes desse grupo em relação às 3 últimas entradas.

As redes 7 e 8 apresentam maior precisão em relação aos valores em regime permanente desejados. Além disso as redes 7 e 8 não demonstram bons comportamentos nas últimas duas entradas.

4.2.3.4 Quarto grupo

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 10, 11 e 12 gerados, é demonstrado na Figura 51.

Figura 51 – Resposta do controlador a entradas aleatórias das redes 10 a 12.



Fonte: Autor

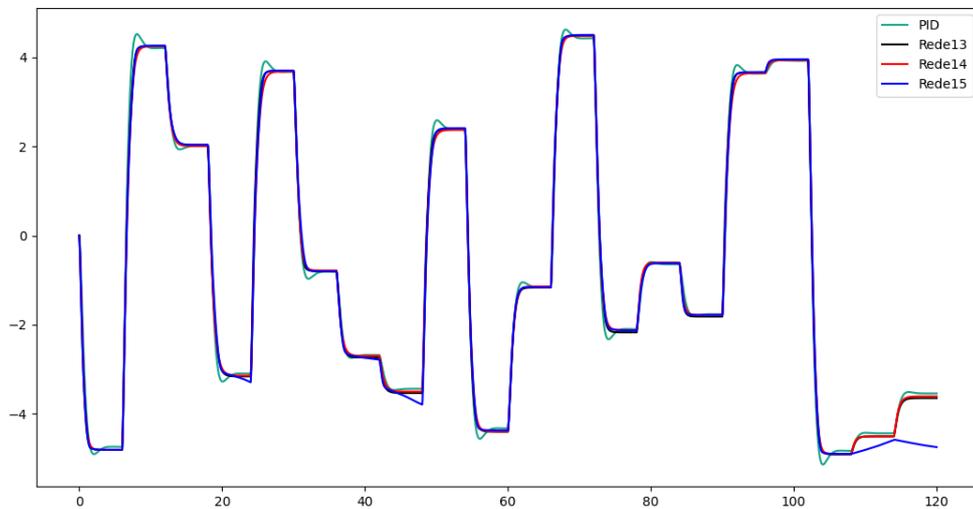
As redes 10 e 11 apresentam comportamentos similares em boa parte do ciclo, gerando imprecisão em relação aos valores desejados. Contudo, a rede 11 apresenta comportamento similar ao apresentado pelo controlador PID nas três últimas entradas.

A rede 12 demonstra maior precisão em relação aos valores desejados, contudo, é possível notar que na primeira e antepenúltima entrada, a resposta demonstra menos precisão. O restante do ciclo, os valores obtidos por essa configuração demonstram acurácia.

4.2.3.5 Quinto grupo

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 13, 14 e 15 gerados, é demonstrado na Figura 52.

Figura 52 – Resposta do controlador a entradas aleatórias das redes 13 a 15.



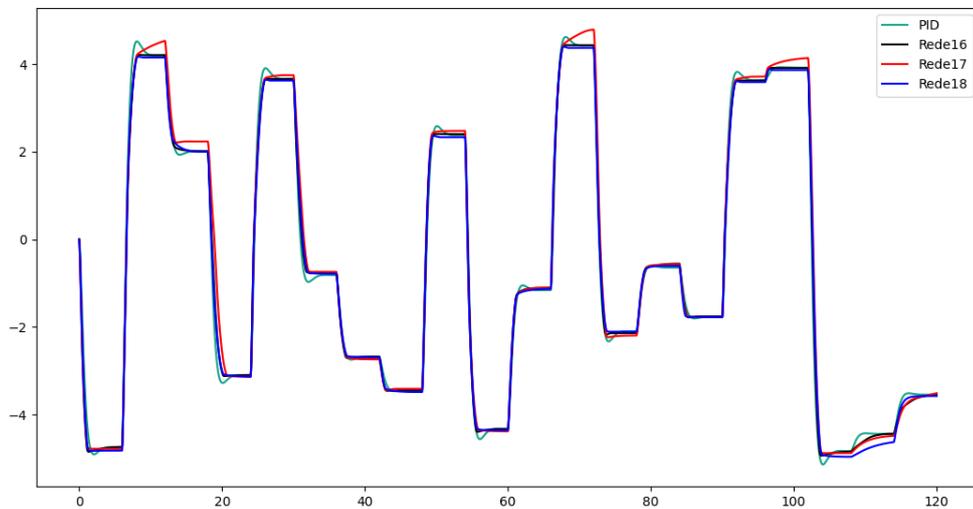
Fonte: Autor

Esse grupo de redes apresenta configurações de rede sem variação de funções de ativação, possuindo apenas a função sigmóide 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.

4.2.3.6 Sexto grupo

A resposta do controlador a entradas aleatórias, obtido por meio do modelos das redes 16, 17 e 18 gerados, é demonstrado na Figura 53.

Figura 53 – Resposta do controlador a entradas aleatórias das redes 16 a 18.



Fonte: Autor

Nota-se de maneira evidente que a rede 17 não apresenta precisão nos valores em relação ao controlador PID em grande parte do ciclo. As redes 16 e 18 apresentam maior precisão, contudo a rede 16 demonstra a maior precisão em relação às demais configurações do grupo.

Além disso, é possível notar que nas três últimas entradas, as redes 16 e 18 conseguem atingir o valor esperado, porém de maneira lenta em comparação com a resposta do controlador PID.

4.2.4 Escolha da configuração de rede

Para a escolha de configuração de rede que o controlador neural pode possuir, deve-se levar em conta as duas simulações feitas: a resposta do sistema ao degrau unitário, bem como, às entradas de maneira aleatória. A configuração de rede que garantir comportamento aceito para ambas as simulações, deve ser escolhido para tal aplicação.

Portanto, 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 possuiu valores de tempo de acomodação e acurácia no valor final coerentes, além disso, não conteve valor de percentual de sobressinal.

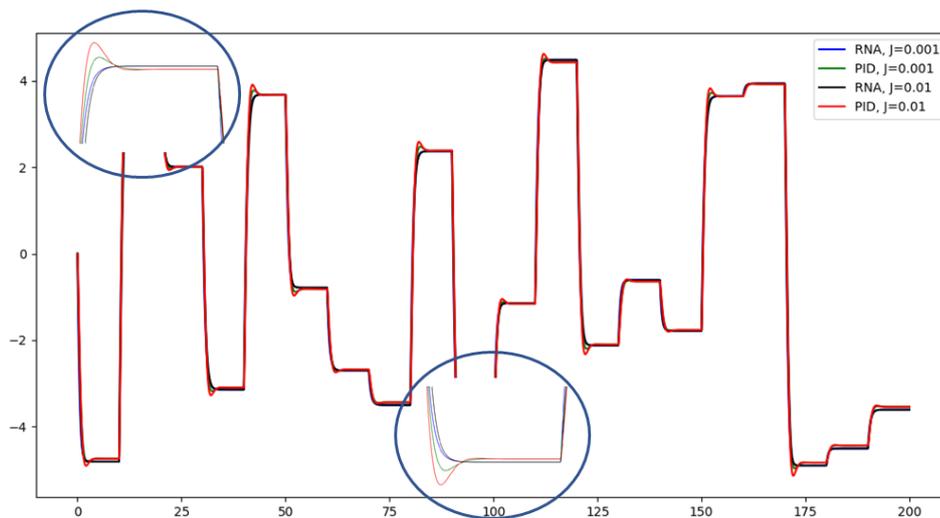
Em relação a etapa de simulação a entradas aleatórias, a rede 13, atingiu valores coerentes e precisos, evidenciando as 3 últimas entradas, cujo qual, diversas

configurações de rede possuíam dificuldade em apresentar acurácia nesses pontos.

4.3 ALTERAÇÕES DO MOMENTO DE INÉRCIA

Segundo Abraham e Shrivastava (2018), as interferências do ambiente de trabalho em aplicações reais, resultam em alterações no momento de inércia do rotor. 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 a qualquer entrada. A Figura 54 demonstra o comportamento do controlador PID, comparado com o controlador neural para momentos de inércia menores que $0,01 \text{ kgm}^2$.

Figura 54 – Comparativo entre controladores em função do momento de inércia ($J < 0,01 \text{ kg s}^2$).

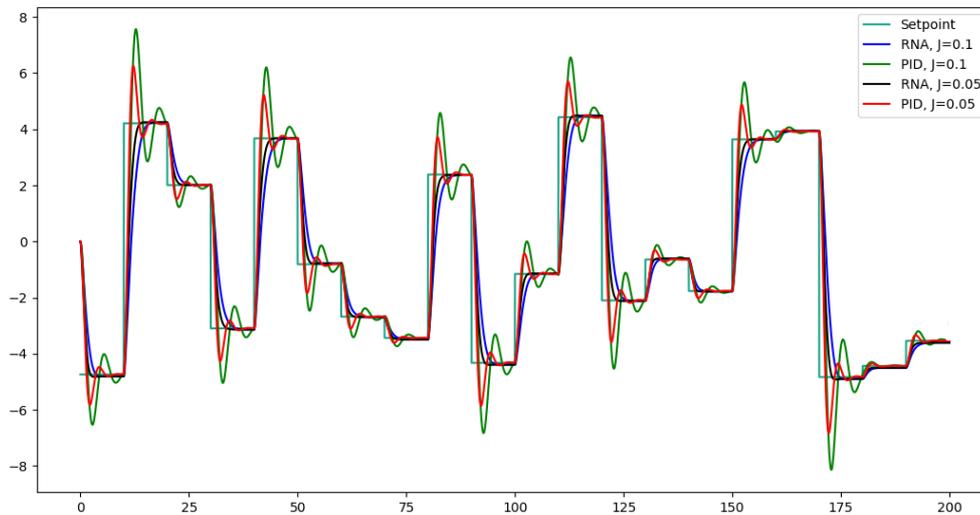


Fonte: Autor

Percebe-se que não há diferença significativa entre os dois valores de momento de inércia. Contudo, observa-se que ao diminuir o valor de J , a velocidade de resposta dos dois controladores aumenta.

Contudo, realizou-se outra simulação dos dois controladores, variando o momento de inércia com valores maiores. Dessa forma, buscou-se analisar o comportamento dos dois controladores em função de momentos de inércia maiores que $0,01 \text{ kgm}^2$. A Figura 55 demonstra a resposta dos dois sistemas para essas variações.

Figura 55 – Comparativo entre controladores em função do momento de inércia ($J > 0,01 \text{ kg}\cdot\text{s}^2$).



Fonte: Autor

É possível observar que o aumento do momento de inércia gera maiores oscilações ao sistema controlado por PID, já para o controlador neural, não houve mudança no comportamento da resposta em relação a mudança de parâmetro, logo, o controlador neural conseguiu exercer a convergência ao valor final em todos os valores de momento de inércia. Tal fato se justifica pelo controlador neural possuir o comportamento do controlador PID, não dependendo da resposta do sistema em função do modelo matemático em tempo real.

Além disso, percebeu-se que com o aumento do valor de momento de inércia do motor CC, a resposta do sistema se apresentou mais lenta, tanto para o controlador PID, como para o controlador por redes neurais. Logo, a oscilação gerada no sistema controlado via PID, se justifica devido a lentidão da convergência para o valor final. Portanto, pode-se concluir que o desempenho do controlador neural se apresentou mais eficiente com a mudança de valores de momento de inércia, pois o mesmo não apresentou diferenças de comportamento com a variação dos parâmetros, além disso, apresentou-se resposta mais rápida em relação ao controlador PID.

5 CONCLUSÕES

O controlador PID ainda é o método de controle mais utilizado na indústria e a permanência desse controlador nos setores industriais, justifica-se pela facilidade de implementá-lo em ambientes fabris. O desempenho desse tipo de controlador é dependente das características físicas e construtivas do sistema a ser controlado, dessa forma, se houver diferenciação entre variáveis do processo, o controlador perde sua eficiência.

Este trabalho tem como objetivo projetar um controlador neural capaz de obter o mesmo comportamento de um PID e tolerante às variações do processo. Dessa forma, realizou-se a simulação de um sistema de controle de um motor CC com o objetivo de avaliar o desempenho de cada controlador e suas respostas à variação do momento de inércia do rotor, verificando se o CN possui independência às físicas do sistema.

Buscando projetar um CN, primeiramente realizou-se a modelação de um controlador PID baseado no sistema de controle do motor CC e nos requisitos de controle desejados. Dessa maneira, com o PID construído, executou-se a simulação do sistema, variando as referências de velocidade desejadas, com o objetivo de obter as entradas para treinamento da RNA, caracterizados pelo setpoint e velocidade obtida pelo sistema e, as saídas representadas pela tensão estabelecida pelo PID.

Com os dados gerados pela simulação obtidos e separados, foi possível realizar o treinamento das redes para análise da melhor configuração de rede, baseando-se no erro médio quadrático e no comportamento gerado em relação a resposta desejada: o controlador PID. Dessa forma, realizaram-se testes durante o treinamento das redes, buscando variar o número de camadas, a quantidade de nós e função de ativação dos neurônios.

Observou-se que a variação das funções de ativação impactava maior diferença nos resultados, onde as funções testadas foram: softmax, tangente hiperbólica, ReLU e sigmóide. Sendo assim, observou-se que as funções de ativação sigmóide e ReLU obtiveram convergência nos resultados com 1000 épocas de treinamento e as demais não apresentaram resultados que convergiam para métricas de erro médio quadrático inferiores a 0,01, bem como, apresentaram alta diferença de comportamento em relação aos dados de teste.

Desse modo, buscou-se avaliar 18 configurações de rede, variando as funções de ativação sigmoide e ReLU e o número de camadas e quantidades de nós em cada camada. Percebeu-se com o treinamento de cada topologia de rede, que as redes que possuíam predominância da função ReLU geravam melhores métricas de treinamento

em relação as redes treinadas com a função sigmoide. Contudo, as redes configuradas com a função sigmoide apresentaram comportamentos com variações mais suaves, diferentemente das RNA's com utilização da função ReLU.

Sendo assim, este trabalho contribui com a proposta de um modelo de avaliação para a escolha do CN ideal para o sistema, com o objetivo de encontrar a configuração que atenda os requisitos desejados, baseando-se na resposta do sistema ao degrau unitário e a entradas aleatórias. Portanto, os 18 modelos de RNA foram utilizados como o controlador do sistema para avaliar o desempenho de cada CN e então, definir a configuração que atenda aos requisitos desejados.

Inicialmente, realizou-se a simulação e análise da resposta de cada CN à entradas unitárias e em seguida, aplicou-se a simulação do sistema a entradas aleatórias. As redes 13 e 14 apresentaram melhores comportamentos durante toda a simulação do sistema a entradas aleatórias, contudo, somente a rede 13 apresentou respostas que atenderam aos requisitos de desempenho de controle a entradas unitárias.

Com a definição da rede 13 como a melhor topologia para o CN empregado a esse sistema, esse trabalho se propôs a avaliar se o desempenho do sistema controlado por uma RNA, tem a capacidade de ser independente às variações dos parâmetros do motor CC. Dessa forma, realizou-se a simulação do sistema variando o momento de inércia do motor, observando que com o aumento dessa variável, o sistema controlado via PID apresentou elevadas oscilações em sua resposta a entradas aleatórias. Já o sistema controlado por CN, não apresentou resposta oscilatória, demonstrando desempenho similar em todos os casos.

Considerando que a configuração de rede escolhida apresentou comportamentos que atendem tanto aos requisitos de controle a entradas unitárias, quanto desempenhos similares ao controlador PID para entradas aleatórias, conclui-se que o projeto de um controlador baseado em uma RNA, demonstra ser possível de ser aplicado. Além disso, a respeito da resposta à variação do momento de inércia do sistema controlado por um CN, conclui-se que o modelo de controlador responde de maneira mais rápida e menos oscilatória que um controlador PID.

Desse modo, uma proposta de futuro trabalho seria projetar o controlador neural em uma aplicação real, buscando avaliar o seu desempenho em aplicação prática e comparando novamente com o sistema controlado por PID. Além disso, realizar o projeto de um CN para um sistema de múltiplas entradas, cujo qual o controlador PID não apresenta desempenho ótimo na prática.

REFERÊNCIAS

- ANG, K. H. et al. Pid control system analysis and design. **IEEE Control Systems**, v. 26, n. 1, p. 32–41, nov. 2006.
- BECKER, A. J. et al.
- Noções básicas em programação matlab.** Notas de aula — Departamento de Matemática, Universidade Federal de Santa Maria, Santa Maria, 2010.
- BOUDJEDIR, H. et al. Adaptive neural network for a quadrotor unmanned aerial vehicle. **International Journal in Foundations of Computer Science Technology**, v. 2, n. 4, p. 1151–1164, jul. 2012.
- BRAGA, P. A.; CARVALHO, F. L. P. A.; LUDEMIR, B. T. **Redes neurais artificiais: teoria e aplicações.** Rio de Janeiro: LTC, 2000.
- DE BRITO, P. S. T. **Desenvolvimento de um controlador pid-fuzzy para o controle de nível de água de um pressurizador de um reator nuclear.** 2015. Dissertação (Mestrado em Ciências) — Departamento de Engenharia Nuclear, Universidade Federal de Pernambuco, Recife, 2015.
- CARVALHO, S. A. **Estudo da utilização do Matlab/Simulink aplicado ao controle de sistemas a eventos discretos.** 2015. Trabalho de conclusão de curso (Graduação em Engenharia de controle e automação) — Escola de engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.
- CHEON, K. et al. On replacing pid controller with deep learning controller for dc motor system. **Journal of Automation and Control Engineering**, v. 3, n. 6, p. 452–456, dez. 2015.
- COPPIN, B. **Inteligência artificial.** Rio de Janeiro: LTC, 2010.
- FACCIN, F. **Abordagem inovadora no projeto de controladores PID.** 2004. Dissertação (Mestrado em Engenharia) — Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.
- HAYKIN, S. **Redes Neurais: princípios e prática.** Porto Alegre: Bookman, 2001.
- HINTON, G. et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. **IEEE Signal Processing Magazine**, v. 19, n. 6, p. 82–97, nov. 2012.
- MAGNUSSON, M. **Comparativo entre controlador PID e redes neurais artificiais no controle de atitude de um quadricóptero.** 2018. Trabalho de conclusão de curso (Graduação em Engenharia de Computação) — Departamento acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Pato Branco, 2018.
- NISE, S. N. **Engenharia de sistemas de controle.** Rio de Janeiro: LTC, 2011.
- OGATA, K. **Engenharia de controle moderno.** São Paulo: Pearson, 2013.

RIBEIRO, A. E. **Controle de temperatura com utilização de controlador neural**. 2016. Trabalho de conclusão de curso (Graduação em Engenharia de Automação Industrial) — Centro Federal de Educação Tecnológica de Minas Gerais, Araxá, 2016.

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

RIZZO, V. I.; CANATO, C. L. R. Inteligência artificial: funções de ativação. **Revista Prospectus**, v. 2, n. 2, p. 51–65, fev. 2020.

ROSA, G. L. J. **Fundamentos da inteligência artificial**. Rio de Janeiro: LTC, 2011.

SANTOS, S. H. **Ajuste de controle pid utilizando algoritmo genético**. 2009. Trabalho de conclusão de curso (Graduação em Engenharia de Computação) — Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2009.

SHRIVASTAVA, S.; ABRAHAM, A. J. Dc motor speed control using machine learning algorithm. **International Journal in Foundations of Computer Science Technology**, v. 7, n. 4, p. 33–37, abr. 2018.

SILVA, N. I.; SPATTI, H. D.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. Scotts Valley: Artibeler, 2010.

SILVA, T. A. M.; CASSIA, G. S.; CARVALHO, J. L. A. Utilização de características radiômicas para classificar gliomas de baixo grau com relação ao status da codeleção cromossômica 1p/19q. **XXXIX SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS**, v. 1, p. 23–28, set. 2021.

TUREVSKIY, A. **PID controller Design for a DC motor**.

Disponível em: <https://www.mathworks.com/matlabcentral/fileexchange/26275-pid-controller-design-for-a-dc-motor>. Acesso em: 06 out. 2022.