

UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

**Modelagem, Identificação e Compensação de Atrito via
Redes Neurais Artificiais de Motor de Corrente Contínua**

Leonardo Pires de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Emanuel da Silva Diaz Estrada
Co-orientador: Prof. Dr. Rodrigo Zelir Azzolin

Rio Grande, 2022

Ficha Catalográfica

S729m Souza, Leonardo Pires de.

Modelagem, identificação e compensação de atrito via redes neurais artificiais de motor de corrente contínua / Leonardo Pires de Souza. – 2022.

114 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2022.

Orientador: Dr. Emanuel da Silva Diaz Estrada.

Coorientador: Dr. Rodrigo Zelir Azzolin.

1. Zona morta 2. Modelos de atrito 3. Compensação 4. Redes neurais artificiais I. Estrada, Emanuel da Silva Diaz II. Azzolin, Rodrigo Zelir III. Título.

CDU 004.7

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

DISSERTAÇÃO DE MESTRADO

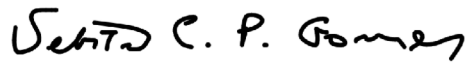
Modelagem, Identificação e Compensação de Atrito via Redes Neurais Artificiais de Motor de Corrente Contínua

Leonardo Pires de Souza

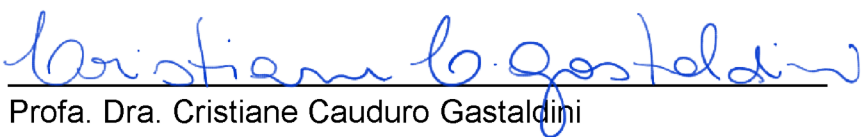
Banca examinadora:



Prof. Dr. Paulo Lilles Jorge Drews Junior



Prof. Dr. Sebastião Cícero Pinheiro Gomes



Profa. Dra. Cristiane Cauduro Gastaldini



Prof. Dr. Rodrigo Zelir Azzolin
Coorientador



Prof. Dr. Emanuel da Silva Diaz Estrada
Orientador

Dedico àqueles que mesmo em fortes crises financeiras, lutam pela educação. Dedico àqueles que mesmo diante de crises sanitárias, lutam pela educação. Dedico àqueles que mesmo sob crises políticas, lutam pela educação. Dedico àqueles que não sofrem de crises financeiras, sanitárias ou políticas mas que também lutam pela educação.

AGRADECIMENTOS

Eu Leonardo, inicio esta seção agradecendo a Deus por me fazer um cara de sorte.

Ao meu orientador, professor Rodrigo Zelir Azzolin pela dedicação e disponibilidade em ajudar neste trabalho, com covid ou sem covid, no frio de Rio Grande ou no calor de Apiacá. Ao professor Emanuel da Silva Diaz Estrada, por nos apoiar em momentos importantes.

Aos membros da banca examinadora, professor Paulo Lilles Jorge Drews Junior, professor Sebastião Cícero Pinheiro Gomes e professora Cristiane Cauduro Gastaldini, que nos ajudaram com dicas valiosas, acompanhadas de muita disponibilidade, boa vontade e paciência.

Agradeço profundamente aos meus primeiros "orientadores", o meu pai, Adilson Gonçalves de Souza, que com gestos me ensina a seguir o caminho do bem e manter o sorriso no rosto, e a minha mãe, Sileia Tebaldi Pires de Souza, que mesmo lá do céu, me traz muita paz e a certeza que tudo acaba dando certo.

Agradeço ao meu "grupo de pesquisa": Aline Gonçalves Pires de Souza, Laís Gonçalves Pires de Souza e Ivna Maria Bastos Vasconcelos. Um grupo muito alinhado e motivado, que me ajudam na resolução de problemas complexos do dia a dia, ainda que por vezes são as causadoras de problemas complexos do meio dia a dia.

A pessoas que fizeram a diferença nesse mundo: Dona Doca e Dona Lécia.

A FURG, uma universidade acolhedora, muito organizada e que nos proporciona uma educação de qualidade.

E aos meus muitos amigos: Patrick, Tiago, Talita, Stefane, Ronan, Lucas, Maycon, Saulo, Suellem, Suélen, Kaiki, Kauan, Geraldo, Sueli, Soninha, Vitória, Letícia, Roberto, Vitor Hugo, Otávio, Mariana, Giovanna, Ronaldo, Auxiliadora, Eder, Alex, Cacau, Keyllon, Keysiane, Zé, Neuza, Raquel, Celso, Betinha, Eduardo, Rafael, Rafaella, Kaka, Thiago, Carlos, Ana Claudia, Claudinha, Mauricio, Thalita, Juninho, Guilherme, Lidi-ane, Eloisa, Lolo, Lara, Laryssa, Claudio, Nete, Eumair, Iofronio, Rodrigo, Karol, Deisi, Tico, Rafaella, Braian, Haroldo, Rosamir, Cesar, Raissa, Samara, Marco, Karla, Bruno, Carolina, Jean, Joninha, Fun, Jhuan, Rodolfo, Moacyr, Murilo, Iago, João Vitor, Jeferson, Racy, Orlando, Renato, Elton, Lohcas, Lucas, Manoel Renan, Márcio, William, Alexandre, Rafael, Macabu, Evaldo, Thiago, Dimitrius, Yago, Troll, Neto, Hissa, Maravilha, João Victor, Breno, Daniel, Átila, Danilo, Ramom, Russel, Pedro, Maíra, Erika, Marioxis, Maria, Pâmela, Tobias, Karina, Jandira, Bernardo, Pauline, Suaidy, Carla. Uma lista tão grande que confirma que sou um cara de sorte.

*Quanto mais eu treino,
mais sorte tenho.*
— ARNOLD PALMER

RESUMO

SOUZA, Leonardo Pires de. **Modelagem, Identificação e Compensação de Atrito via Redes Neurais Artificiais de Motor de Corrente Contínua**. 2022. 114 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Os motores de corrente contínua (CC) são uma classe de motores comumente empregadas em sistemas de controle de torque e velocidade. Entretanto, o controle deste tipo de dispositivo em diversas aplicações é afetada por não linearidades do sistema. Uma dessas não linearidades, a zona morta, é recorrente em sistemas com motores de corrente contínua. A zona morta limita que o motor cumpra sua tarefa com precisão uma vez que desabilita a movimentação do motor para algumas entradas aplicadas. Visto os problemas que a zona morta pode causar em sistemas que requerem precisão de operação, alguns métodos de controle precisam ser implementados para compensá-la, além da necessidade de se representar bem o atrito não linear presente nestes sistemas. Como métodos convencionais de controle não vem se mostrando muito eficientes para esta tarefa, outras formas alternativas estão sendo empregadas e gerando bons resultados de compensação. A utilização de redes neurais para criação de neuro controladores mostra-se um tipo de estratégia de controle eficiente, devido principalmente a capacidade de aprendizado e a propriedade de aproximação universal destes controladores. Assim, este presente trabalho propõe o estudo, projeto e simulação de um sistema de compensação de atrito e zona morta baseado em redes neurais artificiais, além de avaliar os modelos de atrito de Luge e Gomes para a identificação do torque de atrito não linear de um atuador robótico do tipo moto-redutor *Harmonic-Drive*. Concluiu-se, através de resultados de posição e velocidade, que o mecanismo de compensação proposto foi capaz de melhorar o desempenho do atuador robótico.

Palavras-chave: Zona morta, modelos de atrito, compensação, redes neurais artificiais.

ABSTRACT

SOUZA, Leonardo Pires de. **Modeling, Identification and Friction Compensation by Artificial Neural Networks in Direct Current Motor Presenting Dead Zone.** 2022. 114 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Direct current motors (DC) are a class of motors that is commonly employed in torque and speed control systems. However, the control of these motors in real applications is affected by nonlinearities of the system. One of these nonlinearities, the dead zone, is recurrent in systems with direct current motors. The dead zone limits the motor to perform its task accurately as it disables motor movement for some applied inputs. Given the problems the dead zone can cause on systems that require operating accuracy, some control methods need to be implemented to compensate it, in addition to the need to well represent the non-linear friction present in these systems. As conventional control methods are not proving efficient for this task, some alternative methods are being employed and generating good compensation results. The use of artificial neural networks to development of neurocontroller has been shown to be efficient due to its learning ability and its universal approach property. Thus, this present work proposes the study, design and simulation of a friction and deadzone compensation system based on artificial neural networks, in addition to evaluating the Lugre and Gomes friction models for identification of the non-linear friction torque of a Harmonic-Drive gearmotor robotic actuator. It was concluded, through position and speed results, that the proposed compensation mechanism was able to improve the performance of the robotic actuator.

Keywords: dead zone, friction models, compensation, artificial neural network.

LISTA DE FIGURAS

Figura 1	Características de alguns tipos de não linearidade. Em (a) a característica da saturação, em (b) a característica da histerese, em (c) a característica liga-desliga e em (d) a característica da folga	21
Figura 2	Características da zona morta de motores	22
Figura 3	Erros de posição causados pela zona morta	23
Figura 4	Exemplos de motores elétricos	32
Figura 5	Universo tecnológico dos motores elétricos	33
Figura 6	Vista expandida de um motor CC	33
Figura 7	Modelagem do motor CC	34
Figura 8	Diagrama de blocos do motor CC	36
Figura 9	Atuador do tipo moto-redutor Harmonic-Drive	38
Figura 10	Estrutura interna do Harmonic-Drive	38
Figura 11	Funcionamento do atuador Harmonic-Drive	39
Figura 12	Sistemas de engrenagens e transmissão de movimento do Harmonic-Drive	40
Figura 13	Atuador com aproximação rígida e sem carga acoplada na saída . . .	40
Figura 14	Diagrama de blocos do atuador robótico	41
Figura 15	Bancada de testes	42
Figura 16	Curva característica da desaceleração	46
Figura 17	Validação dos parâmetros do atuador robótico para entradas em degrau. Simulação (linhas vermelhas) x Experimento (linhas azuis) . . .	47
Figura 18	Validação dos parâmetros do atuador robótico para entradas em sinais senoidais. Simulação (linhas vermelhas) x Experimento (linhas azuis)	47
Figura 19	Efeito de pré-deslizamento	50
Figura 20	Atraso de fricção	51
Figura 21	Força de ruptura	51
Figura 22	Comportamento elástico do modelo de Lugre	52
Figura 23	Curva estática do atrito em função da velocidade positiva do rotor . .	55
Figura 24	Curva estática do atrito em função da velocidade negativa do rotor . .	56
Figura 25	Curva estática do atrito em função da velocidade em ambos sentidos de rotação	56
Figura 26	Efeito <i>Stick-Slip</i>	57
Figura 27	Representação do atuador robótico com o modelo de atrito de Lugre .	61
Figura 28	Representação no simulink do modelo de atrito de Lugre	62

Figura 29	Simulação com modelo Lugre. Entrada de 20% do torque máximo. Velocidade Simulada em MA (linha vermelha) x Velocidade Experimental em MA (linha azul)	63
Figura 30	Simulação com modelo Lugre ampliado. Entrada de 20% do torque máximo. Velocidade Simulada em MA (linha vermelha) x Velocidade Experimental em MA (linha azul)	63
Figura 31	Simulação com modelo Lugre com foco as inversões de velocidade. Entrada de 20% do torque máximo. Velocidade Simulada em MA (linha vermelha) x Velocidade Experimental em MA (linha azul) . . .	64
Figura 32	Simulação com modelo Lugre com ampliação nas inversões de velocidade. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha vermelha) x Velocidade experimental em MA (linha azul)	64
Figura 33	Simulação com modelo Lugre. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha vermelha) x Velocidade experimental em MA (linha azul)	65
Figura 34	Simulação com modelo Lugre com foco as inversões de velocidade. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha vermelha) x Velocidade experimental em MA (linha azul) . . .	66
Figura 35	Representação do atuador robótico com o modelo de atrito de Gomes	67
Figura 36	Simulação com modelo Gomes. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)	68
Figura 37	Simulação com modelo Gomes ampliado. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)	68
Figura 38	Simulação com modelo Gomes com foco as inversões de velocidade. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)	69
Figura 39	Simulação com modelo Gomes com ampliação nas inversões de velocidade. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul) .	70
Figura 40	Simulação com modelo Gomes. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)	70
Figura 41	Simulação com modelo Gomes com foco as inversões de velocidade. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)	71
Figura 42	Representação de um neurônio artificial	73
Figura 43	Funções de ativação	74
Figura 44	Exemplo de rede alimentada adiante com camadas única	75
Figura 45	Exemplo de rede alimentada adiante com múltiplas camadas	76
Figura 46	Exemplo de rede recorrente	77
Figura 47	Aprendizado supervisionado	79
Figura 48	Aprendizado não-supervisionado	79
Figura 49	Modelo de um perceptron	80
Figura 50	Modelo perceptron de múltiplas camadas	82
Figura 51	Algoritmo <i>Backpropagation</i>	84

Figura 52	Mínimos locais	85
Figura 53	Arquitetura da RNA proposta	87
Figura 54	<i>Network/Data Manager</i>	87
Figura 55	Janela de criação e características da RNA criada	88
Figura 56	Visualização da RNA criada	89
Figura 57	Configuração e variáveis empregadas no treinamento do RNA	89
Figura 58	Padrões de treinamento - T_m	90
Figura 59	Padrões de treinamento - ω_{rs}	91
Figura 60	Padrões de treinamento - T_{at}	91
Figura 61	Processo de treinamento da RNA	92
Figura 62	Precisão da rede treinada	93
Figura 63	Torque de atrito simulado (linha azul) x torque de atrito estimado pela RNA (linha laranja)	94
Figura 64	Comparação da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta). Entrada 20% T_{max}	95
Figura 65	Comparação ampliada da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta). Entrada 20% T_{max}	95
Figura 66	Comparação da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta), com foco nas inversões de velocidade. Entrada 20% T_{max}	96
Figura 67	Comparação ampliada da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta), com foco nas inversões de velocidade. Entrada 20% T_{max}	96
Figura 68	Resultados de simulação com entrada degrau em velocidade e posição angular do sistema linear, não linear e não linear compensado	99
Figura 69	Resultados de simulação com entrada senoidal em velocidade e posição angular do sistema linear, não linear e não linear compensado	99

LISTA DE TABELAS

Tabela 1	Busca Geral - Área temática das publicações	28
Tabela 2	Busca Geral - Tipos de trabalhos	28
Tabela 3	Busca Geral - Trabalhos publicados por ano	28
Tabela 4	Busca por Redes Neurais - Área temática das publicações	29
Tabela 5	Busca por Redes Neurais - Tipos de trabalhos	30
Tabela 6	Busca por Redes Neurais - Trabalhos publicados por ano	30
Tabela 7	Valores Obtidos no Ensaio a Vazio para Identificação de b_1	44
Tabela 8	Parâmetros Identificados do Atuador Robótico	46
Tabela 9	Parâmetros Identificados pela Curva Estática para o Torque de Atrito do Modelo de Gomes	57
Tabela 10	Parâmetros utilizados para a simulação do modelo de atrito de Lugre .	61
Tabela 11	Parâmetros utilizados para a simulação do modelo de atrito de Gomes	67

LISTA DE ABREVIATURAS E SIGLAS

CA	Corrente Alternada
CC	Corrente Contínua
DC	<i>Direct Current</i>
<i>dof</i>	<i>Degree of Freedom</i>
FT	Função de Transferência
FURG	Universidade Federal do Rio Grande
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IET	<i>Institution of Engineering and Technology</i>
MA	Malha Aberta
Matlab	<i>Matrix Laboratory</i>
MF	Malha Fechada
MLP	<i>Multilayer Perceptron</i>
MSE	<i>Mean Square Error</i>
<i>nn toolbox</i>	<i>Neural Network Toolbox</i>
PI	<i>Proportional-Integrative</i>
PID	<i>Proportional-Integrative-Derivative</i>
PWM	<i>Pulse-Width-Modulation</i>
RNA	Redes Neurais Artificiais
SMC	<i>Sliding Mode Control</i>
ZM	Zona Morta

LISTA DE SÍMBOLOS

b	Coeficiente de Atrito Mecânico do Motor [$N.m.s/rad$]
b_1	Coeficiente de Atrito Mecânico do Atuador Robótico [$N.m.s/rad$]
c	Coeficiente de Atrito do Termo Quadrático [$N.m.s^2/rad^2$]
$C(s)$	Controlador
d	Número de Dentes do Moto-Redutor
e	Erro
f_c	Coeficiente de Atrito Seco [$N.m$]
f_{lim}	Maior Valor que Pode Assumir o Coeficiente de Atrito Viscoso Variável [$N.m.s/rad$]
f_s	Coeficiente de Atrito Estático [$N.m$]
f_v	Coeficiente de Atrito Viscoso [$N.m.s/rad$]
f_v^*	Coeficiente de Atrito Viscoso Variável [$N.m.s/rad$]
Fa	Torque de Atrito Não Linear Através do Modelo de Lugre [$N.m$]
$g(V_r)$	Efeito <i>Stribeck</i>
$G(s)$	Planta
$G_r(s)$	Planta Simplificada
i	Índice dos Parâmetros de Atrito do Modelo Gomes
i_a	Corrente da Armadura [A]
j	Índice do Neurônio na Camada de Saída
J	Inércia do Rotor [$Kg.m^2$]
J_1	Inércia do Atuador Robótico [$Kg.m^2$]
J_s	Inércia da Carga [$Kg.m^2$]
L_a	Indutância da Armadura [H]
M_p	Ultrapassagem Percentual
μ	Função de Ativação da RNA
n	Índice de Redução do Moto-Redutor

η	Taxa de Aprendizagem da RNA
net	Combinação Linear
P_{el}	Potência Elétrica [W]
P_{mec}	Potência Mecânica [W]
P_{Cu}	Perdas do Cobre [W]
ψ	Termo Momentum
R_a	Resistência da Armadura [Ω]
t_s	Tempo de Acomodação [s]
T_{at}	Torque de Atrito Não Linear [$N.m$]
T_{atRNA}	Torque de Atrito Não Linear Reconhecido pela RNA [$N.m$]
T_b	Torque do Atrito Viscoso Mecânico [$N.m$]
T_d	Torque de Distúrbio [$N.m$]
T_m	Torque Motor [$N.m$]
T_{max}	Torque Máximo do Motor [$N.m$]
τ	Tempo para o Sinal de Saída Atingir 63,2% da Amplitude em Regime Permanente [s]
τ_d	Constante de Tempo de Decaimento até 36,8% da Velocidade em Estado Estacionário [s]
τ_e	Constante de Tempo da Evolução da Corrente [s]
τ_0	Coeficiente de Rigidez
τ_1	Termo de Amortecimento no Movimento de Pré-Deslizamento
τ_2	Termo de Atrito Viscoso [$N.m.s/rad$]
$u(s)$	Sinal de Controle
V_a	Tensão de Entrada [V]
V_b	Tensão Induzida na Armadura [V]
V_r	Velocidade Relativa Entre Dois Corpos [rad/s]
V_s	Velocidade <i>Stribeck</i> [rad/s]
$x_1...x_D$	Entradas de uma RNA
y	Saída de um RNA
y_d	Saída Desejada de uma RNA
z	Deflexão Média das Hastes
k	Índice do Neurônio nas Camadas Ocultas
K	Constante de Elasticidade Interna
K_b	Constante de Força Contra-Eletromotriz [$V.s/rad$]

K_i	Ganho Integrativo
K_m	Constante de Torque [$N.m/A$]
K_p	Ganho Proporcional
K_0	Ganho do Sistema
$W_1...W_D$	Pesos Sinápticos
ω_n	Frequência Natural [rad/s]
$\omega_r, \dot{\theta}_r$	Velocidade Rotor [rad/s]
ω_{ref}	Velocidade de Referência [rad/s]
ω_{rs}	Velocidade Rotor Simulada [rad/s]
$\dot{\theta}_{lim}$	Velocidade Limite do Fenômeno <i>Stick-Slip</i> [rad/s]
$\dot{\theta}_{st}$	Menor Velocidade Possível em Malha Aberta [rad/s]
γ	Porcentagem da Menor Velocidade Possível em Malha Aberta
Δ	Incremento
Σ	Somador
ζ	Coefficiente de Amortecimento
δ	Parâmetro que Altera Passagem de Atrito Estático para Seco - Curva

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Objetivos	24
1.1.1	Objetivo Geral	24
1.1.2	Objetivos Específicos	24
1.2	Justificativa	25
1.3	Organização e Disposição do Trabalho	25
2	REVISÃO BIBLIOGRÁFICA	27
2.1	Compensação de Zona Morta em Motores CC	27
2.2	Redes Neurais Artificiais	29
3	MOTORES ELÉTRICOS	32
3.1	Motores de Corrente Contínua	33
3.1.1	Modelagem Matemática do Motor de Corrente Contínua	34
3.2	Atuadores Robóticos <i>Harmonic-Drive</i>	37
3.2.1	Modelagem do Atuador <i>Harmonic-Drive</i>	39
3.3	Identificação Paramétrica do Motor CC do Atuador <i>Harmonic-Drive</i>	41
3.3.1	Resistência da Armadura R_a	43
3.3.2	Indutância da Armadura L_a	43
3.3.3	Coeficiente de Atrito Mecânico b_1	43
3.3.4	Constante de Torque K_m	44
3.3.5	Constante de Força Contra-Eletromotriz K_b	45
3.3.6	Inércia do Rotor J_1	45
3.3.7	Validação dos Parâmetros Identificados	46
4	MODELOS DE ATRITO	49
4.1	Modelo de Atrito de Lugre	52
4.2	Modelo de Atrito de Gomes	53
5	SIMULAÇÕES E EXPERIMENTOS EM MALHA ABERTA COM OS MODELOS DE ATRITO	60
5.1	Simulações e Experimentos com o Modelo de Atrito de Lugre	61
5.1.1	Simulações em Baixa Velocidade - 20% T_{max}	62
5.1.2	Simulações em Média Velocidade - 33% T_{max}	62
5.2	Simulações e Experimentos com o Modelo de Atrito de Gomes	66
5.2.1	Simulações em Baixa Velocidade - 20% T_{max}	67
5.2.2	Simulações em Média Velocidade - 33% T_{max}	69

6	REDES NEURAIS ARTIFICIAIS	72
6.1	Neurônio Artificial	72
6.1.1	Função de ativação	73
6.2	Arquitetura das RNAs	75
6.2.1	Redes alimentadas adiante com camada única (<i>Feedfoward</i>)	75
6.2.2	Redes alimentadas adiante de múltiplas camadas (<i>Multilayer Feedfoward Networks</i>)	76
6.2.3	Redes Recorrentes (<i>Feedback</i>)	76
6.3	Processos de Aprendizagem	77
6.3.1	Aprendizado Supervisionado	78
6.3.2	Aprendizado Não-Supervisionado	78
6.4	Perceptron e Perceptron de Múltiplas Camadas	79
6.4.1	Algoritmo <i>Backpropagation</i>	82
7	SIMULAÇÕES E EXPERIMENTOS COM REDES NEURAIS ARTIFICIAIS	86
7.1	Criação da RNA	86
7.2	Treinamento da RNA	89
7.3	Validação da RNA Proposta	94
8	RESULTADOS DE COMPENSAÇÃO COM A RNA	98
9	CONCLUSÕES	100
	REFERÊNCIAS	102
ANEXO A	INFORMAÇÕES ADICIONAIS SOBRE REDES NEURAIS ARTIFICIAIS	107
A.1	Histórico das Redes Neurais Artificiais	107
A.2	Neurônio Biológico	108
ANEXO B	SINTONIA DO CONTROLADOR PI	110
ANEXO C	<i>DATASHEET</i> DO MOTOR MAXON RE 25 Ø25 MM, GRAPHITE BRUSHES, 20 WATT	114

1 INTRODUÇÃO

Desde os primórdios, a humanidade utiliza fontes motoras para obter energia e trabalho. Por muito tempo, a forma encontrada pelo homem para realizar trabalho e obter fontes motoras foi através da força proveniente da tração animal. Utilizar da contribuição de animais como os cavalos, o gado, cães e por grande parte das vezes do próprio homem, era a forma encontrada para atender a demanda. Posteriormente, com a descoberta de novas técnicas e ideias, passou-se a utilizar desde correntes de água até o vento e o vapor para obtenção de trabalho. Porém, com o caótico desenvolvimento das sociedades, houve uma necessidade imediata de encontrar formas motoras mais produtivas e eficientes. Tornou-se então um grande objetivo a busca por novas fontes motoras.

Essa grande busca culminou no desenvolvimento de um dispositivo que converte outras formas de energia em energia mecânica, de forma a impelir movimento a uma máquina, veículo ou carga. Este dispositivo foi nomeado motor. Para Hughes e Drury (2019), o motor consiste de nada mais que um arranjo de bobinas de cobre e aços laminados e é claramente um conversor inteligente de energia. Existem diversos tipos de motores, que são classificados de acordo com a forma de energia que eles utilizam para converter em energia mecânica. Pode-se citar motores que utilizam de energia térmica, como os motores de combustão, motores que convertem energia elétrica em energia mecânica, como os motores elétricos e dispositivos que utilizam de energia potencial ou cinética para gerar trabalho, como os motores a ar comprimido e os motores hidráulicos.

Dentre os tipos de motores citados anteriormente, os motores elétricos merecem destaque. Motores elétricos são parte da vida cotidiana e por vezes acabam passando despercebido (HUGHES; DRURY, 2019). Eles são responsáveis por grandes avanços industriais, principalmente por combinarem as vantagens da energia elétrica, como seu baixo custo, facilidade de transporte e ser uma fonte limpa, com uma forma construtiva simples, de baixo custo e grande versatilidade e aplicabilidade. Como observado por Mohan (2012), máquinas elétricas já existem por mais de um século e todos já se familiarizaram com sua função: movimentar mecanicamente alguma carga pela conversão de energia elétrica. Segundo o mesmo autor, em resposta a um comando elétrico de entrada, os motores elétricos eficientemente controlam a velocidade e/ou posição de uma carga mecânica.

A classe de motores elétricos é a mais utilizada nos dias atuais. Mohan (2012) lista dois fatores como os principais responsáveis pelo crescimento na utilização dos motores elétricos: revolucionários avanços na microeletrônica que resultam em circuitos mais poderosos e necessidades do mercado. Ainda no início dos anos 2000, este era um mercado que movimentava mais de 20 bilhões de dólares e este mercado está crescendo (THØGERSEN; BLAABJERG, 2004).

Historicamente, os motores de corrente contínua (Motores CC) têm sido os mais populares em aplicações de controle de posição e velocidade. Eles ganharam essa popularidade por seu baixo custo e facilidade de controle (MOHAN, 2012). Como definido por Fitzgerald et al. (2003), a versatilidade de máquinas de controle contínuo, combinado com sua relativa simplicidade, garantirão que elas continuem sendo utilizadas em uma grande variedade de aplicações. Existe um grande número de motores CC atualmente em uso e este número continuará crescendo (MOHAN, 2012).

Seguindo como base a definição de motores elétricos, um motor de corrente contínua, ou simplesmente um motor contínuo, é uma máquina elétrica que transforma a energia elétrica na forma de corrente contínua em energia mecânica para movimentação de cargas.

Ainda segundo Mohan (2012), a maior vantagem dos motores CC é a facilidade com que torque e velocidade podem ser controlados. Pela facilidade de controlar essas variáveis, máquinas de corrente contínua podem ser frequentemente usadas em aplicações que requerem uma ampla faixa de velocidades do motor ou controle preciso da saída do motor, tanto para operações dinâmicas quanto para estado estacionário (FITZGERALD et al., 2003).

Entretanto em implementações industriais, por exemplo, os motores de corrente contínua não apresentam apenas vantagens, principalmente devido a não linearidades dos sistemas. Sistemas físicos são inerentemente não lineares (SLOTINE; LI et al., 1991). Os sistemas não lineares são muito mais complexos em questão de comportamento. Segundo Jang, Chung e Jeon (2005), saturação, zona morta, folga (*backlash*), histerese e característica liga-desliga (*on-off*), são as não linearidades mais comuns em atuadores em sistemas práticos de controle. A Figura 1, retirada de Silva (2006), ilustra características de alguns destes tipos de não linearidades na qual em (a) é a característica da saturação, em (b) a histerese, em (c) a característica liga-desliga e em (d) a folga. É possível observar os diferentes tipos de comportamento da saída y do sistema para as entradas u fornecidas.

As não linearidades podem ser classificadas como intencionais ou inerentes. As não linearidades intencionais são aquelas introduzidas deliberadamente em um sistema para melhorar seu desempenho ou simplificar sua construção. Um exemplo simples de um sistema não linear intencional é um sistema operado convenientemente por relé. Não linearidades inerentes são aquelas que existem naturalmente nos sistemas. Usualmente estas não linearidades causam efeitos indesejáveis aos sistemas, devendo ser compensadas apropriadamente.

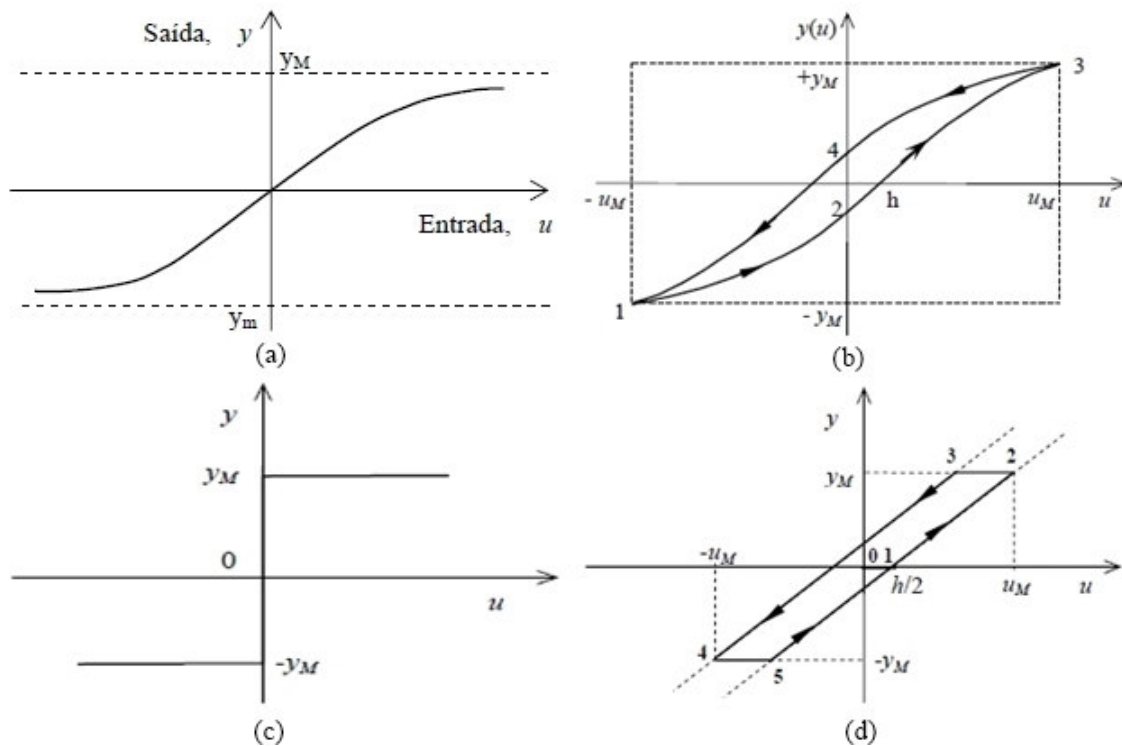


Figura 1: Características de alguns tipos de não linearidade. Em (a) a característica da saturação, em (b) a característica da histerese, em (c) a característica liga-desliga e em (d) a característica da folga

Fonte: (SILVA, 2006)

Em se tratando de sistemas compostos por motores de corrente contínua, a zona morta é uma não linearidade recorrente. De acordo com Jang, Chung e Jeon (2005), zona morta é uma não linearidade estática que descreve uma falta de sensibilidade de um sistema de trabalhar com pequenos sinais. Ela se configura como uma faixa operacional de valores de entrada que não produz resposta na dinâmica de saída do sistema (SALCEDO et al., 2010).

A zona morta é causada primariamente pela força de fricção de Coulomb do rotor do motor (KARA; EKER, 2004). Segundo Slotine, Li et al. (1991), a zona morta de um motor limita a habilidade deste motor de responder a pequenas entradas visto que o torque gerado por pequenas entradas será tipicamente menor que o torque requerido para superar a força de fricção de Coulomb. Como exemplificado por Slotine, Li et al. (1991), considerando o caso do motor CC, em um modelo ideal, assume-se que qualquer tensão aplicada as bobinas da armadura irá causar um movimento rotativo, onde pequenas tensões aplicadas causam pequenos movimentos.

Porém, na realidade, devido a força de fricção de Coulomb do rotor do motor, a rotação só irá ocorrer quando o torque gerado pelo motor for suficientemente grande para superar a força de fricção.

Para He, Wang e Hao (2015), a presença de zona morta fará com que o sistema perca

performance ou se torne instável. A zona morta evita que um sistema cumpra sua tarefa com precisão, uma vez que desabilita a rotação do motor sobre alguns torques aplicados, representando uma perda de informação quando o sinal fica dentro da banda morta, podendo causar ciclos limites, erro de rastreamento entre outros problemas.

A característica da zona morta é apresentada na Figura 2, retirada de Silva (2006), onde se observa que ela pode ser de dois tipos. Citada anteriormente, a banda morta representa os valores compreendidos entre $-Z_M$ e Z_M e para valores de entrada u , como tensão por exemplo, dentro da banda morta, os valores de saída $y(u)$, como torque, são zero.

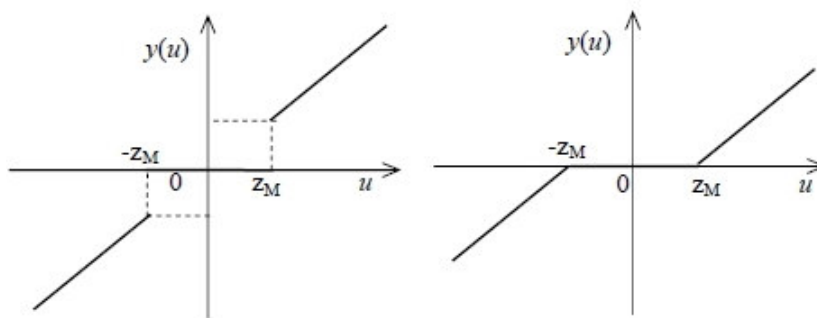


Figura 2: Características da zona morta de motores
Fonte: (SILVA, 2006)

Segundo Jang, Chung e Jeon (2005), embora existam algumas aplicações em malha aberta da qual a característica da zona morta é altamente desejada e por isso, aplicada de forma intencional, na maioria das aplicações em malha fechada, a zona morta causa efeitos indesejados na dinâmica de realimentação e no desempenho de controle do sistema. Afirmado por Borowski (2013), a zona morta do motor introduz erros de posição que podem afetar de forma severa o desempenho de uma aplicação com motores CC.

A Figura 3 demonstra a diferença de posição de um motor com e sem a não linearidade zona morta. É possível observar que a posição do rotor do motor com zona morta (linha tracejada) apresenta erros consideráveis se comparados a posição desejada do rotor no caso de um motor que não apresente zona morta (linha contínua).

Como observado por Borowski (2013), os erros gerados pela zona morta podem ser bem significativos e pode se tornar um grande problema em sistemas físicos que requerem alta precisão no controle de posição, por exemplo. A perfeita compensação da zona morta é difícil, mas seus efeitos de degradação do desempenho podem ser minimizados (VALDIERO; GUENTHER; NEGRI, 2005). Ainda segundo Borowski (2013), métodos de controle precisam ser implementados para compensar a zona morta.

Em virtude da dificuldade de controlar sistemas com não linearidades, imprecisões e incertezas, métodos convencionais de controle não são tão efetivos quanto necessário. Desta forma, alguns métodos de controle moderno estão sendo mais empregados nestas

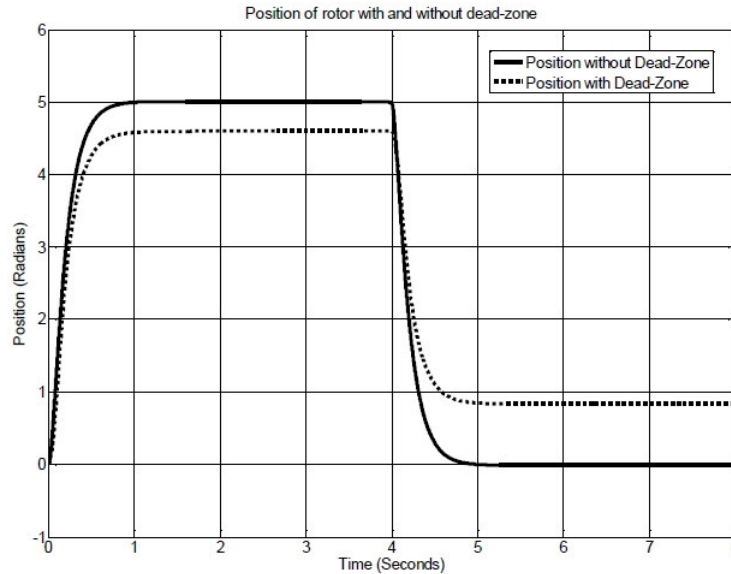


Figura 3: Erros de posição causados pela zona morta
Fonte: (BOROWSKI, 2013)

aplicações. De acordo com He, Wang e Hao (2015), controle por modos deslizantes (*Sliding Mode Control* - SMC), controle nebuloso (controle fuzzy), e outros métodos de controle inteligente, como as redes neurais artificiais (RNA), são usados para compensar os efeitos negativos da zona morta.

Muito vem sendo pesquisado em controles inteligentes usando redes neurais artificiais. As redes neurais artificiais foram desenvolvidas através da observação do funcionamento do cérebro humano. Para Haykin (1994), uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Já para Silva (2006), uma rede neural artificial (RNA) é uma estrutura matemática constituída por um número finito de unidades individualizadas, também designadas por neurônios, organizados em camadas.

Com a propriedade de aproximação universal e capacidade de aprendizado, as redes neurais artificiais vem provando serem ferramentas poderosas para o controle da dinâmica complexa de sistemas não lineares com parâmetros incertos (JANG; CHUNG; JEON, 2005).

Porém, para garantir uma boa resposta de redes neurais artificiais, é de extrema importância que esta tenha sido criada e treinada em sua melhor versão. Para o treinamento eficiente de uma RNA, seus padrões de treinamento devem ser coerentes. Desta forma, é necessária uma modelagem de atrito que seja fidedigna com as representações experimentais do sistema utilizado, como por exemplo os motores de corrente contínua, ainda que é sabido que baseado em sua natureza não linear, o atrito não é tão simples de modelar, sendo buscado uma representação aproximada do comportamento do sistema. Segundo

Olsson e Astrom (1996), um modelo de atrito é elemento essencial para um bom projeto de sistemas. Dentre as diversos estudos da modelagem do atrito, dois se mostram muito eficientes: o modelo de atrito de Lugre e o modelo de atrito de Gomes. Segundo Wit et al. (1995), seu modelo de Lugre proposto, captura a maior parte do comportamento de atrito observados experimentalmente, incluindo o efeito *Stribeck*, histerese, características de mola para atrito e variação da força de ruptura. Já segundo Gomes e Rosa (2003), o modelo de atrito de Gomes é mais realístico para atuadores robóticos do tipo moto redutor, sendo capaz de compensar o efeito *stick-slip*. Estes modelos representam boa parte dos fenômenos do atrito e por isso são muito difundidos nas aplicações cuja modelagem do atrito se torna necessária.

Assim, este presente trabalho propõe-se a desenvolver e aplicar uma rede neural artificial com o objetivo de reproduzir o torque de atrito não linear causado pela zona morta além de compensar os efeitos causados por este torque de atrito em um atuador robótico que possui um motor de corrente contínua. Além disto, modelar o torque de atrito gerado no sistema através de dois modelos de atrito: o modelo de Lugre e o modelo Gomes. Por fim realizar uma análise e descrição das redes neurais artificiais empregadas em métodos de controle moderno na compensação de zona morta em motores de corrente contínua.

1.1 Objetivos

1.1.1 Objetivo Geral

Este trabalho tem por objetivo o estudo, projeto e simulação de um sistema de compensação de zona morta baseado em redes neurais artificiais. São avaliados os modelos de atrito de Lugre e Gomes para a identificação do torque de atrito não linear de um atuador robótico do tipo moto-reductor *Harmonic-Drive*, parâmetro utilizado no treinamento das redes.

1.1.2 Objetivos Específicos

- Apresentar de forma didática e completa importantes questões sobre a modelagem de atrito e a criação e treinamento de redes neurais artificiais;
- Ampla revisão bibliográfica sobre zona morta e técnicas de controle para sua compensação;
- Estudo, modelagem, identificação de parâmetros e validação do modelo do atuador robótico utilizado no trabalho;
- Estudo, modelagem, identificação de parâmetros e validação do modelo de atrito proposto por Lugre;

- Estudo, modelagem, identificação de parâmetros e validação do modelo de atrito proposto por Gomes;
- Proposta de rede neural artificial capaz de identificar o torque de atrito não linear do sistema;
- Projeto de mecanismo de compensação de torque de atrito não linear e zona morta.

1.2 Justificativa

Visto os problemas recorrentes que a zona morta causa em sistemas de controle e a necessidade de compensá-los para aplicações que exijam precisão, o respectivo trabalho pretende apresentar uma forma de controle moderno que é capaz de atender, com eficiência, as demandas destes sistemas. Ainda, simular este método para comprovar a melhoria causada pela utilização desta estratégia de controle.

1.3 Organização e Disposição do Trabalho

Capítulo 1: Introdução. Neste primeiro capítulo são abordadas definições e questões relevantes sobre diferentes temas, sendo eles os motores elétricos de corrente contínua, as não linearidades de sistemas, zona morta em motores. Também são apresentadas as redes neurais artificiais, um método moderno de controle de sistemas não lineares, e os modelos de atrito. Expõem-se os objetivos, justificativas e a forma como o tema é abordado e organizado nos capítulos.

Capítulo 2: Revisão Bibliográfica. Neste capítulo são observados alguns materiais disponíveis no mercado com certa semelhança ao material proposto neste trabalho, apresentando suas características e diferenças.

Capítulo 3: Motores Elétricos. No capítulo, é apresentada toda a parte teórica envolvida nesta pesquisa no que se refere a motores elétricos de corrente contínua. É informada toda a modelagem dos motores CC, seu universo tecnológico e diagrama de blocos. Também é descrito o atuador robótico utilizado no trabalho, com sua modelagem, funcionamento e características. Por fim, é identificado os parâmetros do motor CC que compõe o atuador robótico.

Capítulo 4: Modelos de Atrito. Este capítulo se destina a apresentar os modelos de atrito de Lugre e Gomes, modelos que serão utilizados na identificação do torque de atrito não linear. Suas equações e definições são apresentadas, bem como um experimento para identificação dos parâmetros dos modelos.

Capítulo 5: Simulações e Experimentos em Malha Aberta com os Modelos de Atrito. Neste capítulo as simulações e experimentos realizados com ambos modelos de atrito são apresentados em busca de validar os modelos propostos, finalizando com uma comparação entre os resultados encontrados.

Capítulo 6: Redes Neurais Artificiais. É fornecido toda a parte teórica envolvida nesta pesquisa no que se refere a redes neurais artificiais. São apresentadas as formações dos neurônios artificiais, arquitetura de redes, processos de aprendizagem e algoritmos de correção de erros.

Capítulo 7: Simulações com as Redes Neurais Artificiais. Este capítulo se inicia com informações relevantes sobre a criação e o treinamento da RNA desenvolvida neste projeto, seguida de simulações realizadas com a intenção de confirmar a eficiência da RNA na reprodução e identificação do atrito não linear do sistema.

Capítulo 8: Resultados de Compensação com a RNA. São apresentados os conceitos da criação do mecanismo de compensação, com a apresentação de simulações de diversos resultados de compensação do atuador robótico através da RNA.

Capítulo 9: Conclusões. Aqui é feita uma observação sobre os resultados obtidos neste trabalho, além de sugerir trabalhos futuros.

Referências. Lista os trabalhos que serviram como base para o desenvolvimento desta pesquisa.

Anexo A: Informações Adicionais Sobre Redes Neurais Artificiais É exposto um breve histórico do surgimento das redes neurais artificiais e informações sobre o funcionamento dos neurônios biológicos, que inspiraram os neurônios artificiais.

Anexo B: Sintonia do Controlador PI. Apresentado neste anexo o projeto do controlador PI utilizado neste trabalho para geração de dados em malha fechada.

Anexo C: Datasheet do Motor Maxon RE 25 Ø25MM, Graphite Brushes, 20 Watt. Este anexo apresenta o *datasheet* do motor CC que compõe o atuador robótico empregado no trabalho.

2 REVISÃO BIBLIOGRÁFICA

Para uma sólida fundamentação da pesquisa proposta neste trabalho, foi realizada uma ampla revisão bibliográfica, que será apresentada neste capítulo e que pode ser complementada em Souza e Azzolin (2020). Este processo de revisão da bibliografia foi realizado em duas etapas principais: Busca sobre o tema geral, neste caso a compensação de zona morta em motores de corrente contínua e buscas específicas relacionadas a compensação de zona morta em motores CC com as redes neurais artificiais.

Assim, as informações deste capítulo serão apresentadas na ordem do tema mais geral para os mais específicos.

Em todas as etapas de pesquisa, foram feitas buscas em repositórios de dados eletrônicos, que se encontram disponíveis na internet. Os repositórios de dados consultados foram o SpringerLink, uma base de dados da editora Springer, que tem registrado mais de 10 milhões de documentos nos seus 23 anos de existência e o IEEE Xplore Digital Library, uma biblioteca digital produzida pelo IEEE (*Institute of Electrical and Electronics Engineers*) e o IET (*Institution of Engineering and Technology*), com mais de 5 milhões de documentos registrados.

A partir destas pesquisas, a distribuição destas publicações foi categorizada por quantidade de publicações por área de estudo, tipos de publicações e o ano em que foram publicadas. Tabelas para facilitar o entendimento serão expostas em cada seção seguinte. Os trabalhos mais relevantes foram destacados e melhor descritos.

2.1 Compensação de Zona Morta em Motores CC

Na etapa de pesquisa sobre o tema geral, foram feitas consultas aos repositórios através do uso de palavras chaves em inglês. Desta forma, no dia 21 de novembro de 2019, o tema "Compensação de zona morta em motores CC" foi introduzido na ferramenta de busca na forma de:

- Para o SpringerLink: "dc motor" AND "compensation" AND "dead zone";
- Para o IEEE Xplore: ((("All Metadata":dc motor) AND "All Metadata":compensation) AND "All Metadata":dead zone).

Retornaram 116 publicações, cujo trabalho mais antigo publicado foi a primeira versão do livro Shafer (2013). As categorizações dos resultados encontrados na busca são demonstradas nas tabelas 1, 2 e 3.

Tabela 1: Busca Geral - Área temática das publicações

Classificação	Área Temática	Quantidade
1	Engenharia	102
2	Ciência da Computação	9
3	Matemática	2
4	Física	2
5	Ciências Sociais	1

Fonte: Próprio Autor

Tabela 2: Busca Geral - Tipos de trabalhos

Classificação	Tipo	Quantidade
1	Artigo Científico	66
2	Capítulo de Livro	48
4	Trabalho de Referência	2

Fonte: Próprio Autor

Tabela 3: Busca Geral - Trabalhos publicados por ano

Ano de Publicação	Quantidade
Antes de 2000	14
Entre 2000 e 2010	23
Após 2011	79

Fonte: Próprio Autor

Dentre os trabalhos que merecem destaque pode-se citar Campos-Delgado et al. (2013). Neste trabalho foi proposta a criação, através do Matlab/Simulink, de um diagrama de blocos capaz de compensar a zona morta do motor CC, empregado para movimentar uma bomba usada no controle da pressão de perfusão renal de ratos.

Já nos trabalhos Takroui e Dhaouadi (2019a) e Takroui e Dhaouadi (2019b) os autores utilizam de um observador adaptativo para identificar e compensar a zona morta presente no sistema proposto.

Uma outra interessante estratégia de compensação de zona morta encontrada na literatura foi no trabalho Teng, Yang e Wang (2016), na qual se afirma que a zona morta pode também ser descrita com tempo morto e por esta razão elas podem ser consideradas como sistemas que sofrem de atraso de entrada. Assim, os autores resolveram aplicar a teoria de controle para sistemas com atraso de tempo para garantir a estabilidade assintótica do sistema e posteriormente a compensação da não linearidade.

Outro modelo de trabalho encontrado nas buscas, foram os trabalhos de visão geral sobre compensação de não linearidades. Neste tipo de trabalho é possível encontrar descrições sobre diferentes métodos de controle. Como exemplo, pode-se citar os trabalhos Armstrong-Hélouvry, Dupont e Wit (1994) e Bona e Indri (2005). Ambos trabalhos são focados em definições, fundamentações teóricas e as vantagens de aplicações dos diferentes métodos em diferentes sistemas. Porém, não desenvolvem os métodos nem os aplicam em simulações ou experimentações para gerar resultados.

Entretanto, o principal ponto que pode ser observado quando da realização desta pesquisa geral é a vasta aplicação de redes neurais artificiais como estratégias de controle para solucionar problemas do tipo identificação e compensação de zona morta. Esta constatação evidencia a eficiência das RNAs.

2.2 Redes Neurais Artificiais

Após a busca geral realizada, o início das buscas específicas foi através da adição das palavras chaves redes neurais (*neural network*). Assim, ainda no dia 21 de Novembro de 2019, foi adicionada na ferramenta de busca do SpringerLink a tag AND "neural network" enquanto que no IEEE Xplore na forma de AND "All Metadata":neural network. Assim, as tags utilizadas foram:

- Para o SpringerLink: "dc motor"AND "compensation"AND "dead zone"AND "neural network";
- Para o IEEE Xplore: (((("All Metadata":dc motor) AND "All Metadata":compensation) AND "All Metadata":dead zone) AND "All Metadata":neural network).

As buscas retornaram 45 trabalhos, cujo mais antigo é o artigo Lygouras (1999). A relação dos resultados encontrados por categorias são apresentados nas tabelas 4, 5 e 6.

Tabela 4: Busca por Redes Neurais - Área temática das publicações

Classificação	Área Temática	Quantidade
1	Engenharia	42
2	Ciência da Computação	3

Fonte: Próprio Autor

Após uma triagem dos trabalhos encontrados, foi possível observar que em alguns deles, a estratégia de controle de redes neurais artificiais não era desenvolvida, sendo simplesmente mencionada ao longo dos trabalhos ou sendo realizada uma revisão de conceitos e trabalhos.

Tabela 5: Busca por Redes Neurais - Tipos de trabalhos

Classificação	Tipo	Quantidade
1	Artigo Científico	29
2	Capítulo de Livro	16

Fonte: Próprio Autor

Tabela 6: Busca por Redes Neurais - Trabalhos publicados por ano

Ano de Publicação	Quantidade
Antes de 2000	1
Entre 2000 e 2010	11
Após 2011	33

Fonte: Próprio Autor

Tomando como base apenas os trabalhos que simularam ou obtiveram resultados experimentais na literatura, diferentes e interessantes estratégias da utilização de redes neurais são empregadas nos mais diversos trabalhos. Como por exemplo no artigo Jun e Gi (2000). Neste trabalho, é proposto um método de controle por rede neural para compensar a não linearidade, da qual o proposto neuro-controlador consiste de um controlador linear em paralelo com um controlador de rede neural (JUN; GI, 2000). A rede neural adaptará seus próprios pesos a partir da identificação do erro da planta.

Uma sequência de trabalhos chama a atenção. Inicialmente, no artigo Gervini e Gomes (2001) é apresentada uma estratégia de treinamento e proposta uma estrutura de uma rede neural para aprender o torque de atrito de um atuador robótico. Os autores criaram e treinaram uma rede neural do tipo *backpropagation* com dois neurônios de entrada para receber as variáveis torque do motor e velocidade do rotor e estimar uma saída referente a variável torque de atrito estimado, sendo este o valor responsável por compensar as perdas provenientes da zona morta. Simulações foram realizadas e ótimos resultados foram alcançados. Posteriormente, um segundo artigo Gervini, Gomes e Rosa (2003) utilizou desta mesma rede neural criada anteriormente, desta vez para realizar testes experimentais com um motor real isolado em uma bancada. O mecanismo de compensação proposto, que usa o torque de atrito estimado na saída da rede neural, se mostrou muito eficiente e reduziu significativamente a zona morta em torque no motor usado nos experimentos (GERVINI; GOMES; ROSA, 2003).

No artigo de Machado et al. (2007), emprega-se uma rede neural artificial para compensar, em torque, a zona morta e lógica fuzzy para fazer ajustes da rede neural. A saída da RNA, que obtém o torque estimado que compensa o torque perdido pela zona morta, é multiplicado por um ganho oriundo da lógica fuzzy, para que se compense as mudanças da zona morta provenientes do tempo e condições de operação.

Um caso semelhante é observado no artigo Hsu (2014), cuja proposta foi de desen-

volver um controle inteligente por modos deslizantes para compensar a zona morta. Este controlador inteligente, é composto de um controlador principal e um compensador robusto. Para criar um aproximador para o controlador principal, o autor utilizou um modelo de rede neural.

Por último, o trabalho Selmic e Lewis (2000) apresenta uma estratégia de utilização de redes neurais diferente das demais aqui descritas. Este trabalho cria um compensador que usa duas redes neurais, uma para estimar a zona morta desconhecida e outra para fornecer compensação adaptativa em caminho direto. Através de um algoritmo, os pesos de ambas redes neurais são ajustados e uma análise da estabilidade do sistema em malha fechada também é fornecido.

3 MOTORES ELÉTRICOS

O motor elétrico é uma máquina destinada a transformar energia elétrica em mecânica. Concebido no século XIX, este é o mais usado de todos os tipos de motores (hidráulico, pneumático, combustão interna, entre outros), pois combina as vantagens da utilização de energia elétrica - baixo custo, facilidade de transporte, limpeza e simplicidade de comando - com sua construção simples, grande versatilidade de adaptação às cargas dos mais diversos tipos e melhores rendimentos (WEG, 2005). Alguns motores elétricos são apresentados na Figura 4.

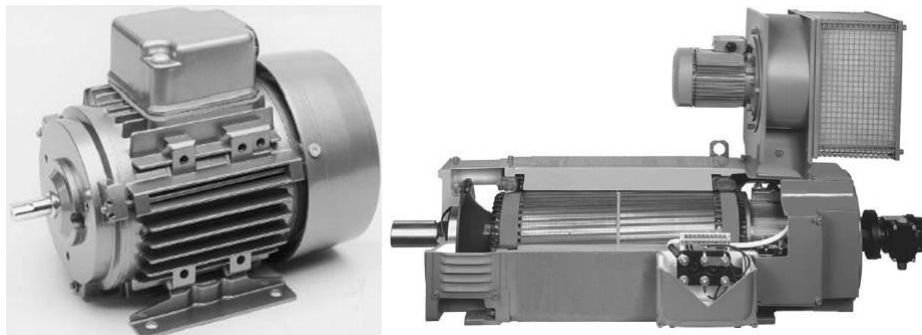


Figura 4: Exemplos de motores elétricos
Fonte: (HUGHES, 2019)

Desde sua criação, o motor elétrico passou por inúmeras alterações e, para que atendesse as necessidades que surgiram ao longo dos anos, foram desenvolvidos vários tipos de motores elétricos. Atualmente estes motores são divididos em três grupos: os de corrente alternada (CA), corrente contínua (CC) e universal. Essa classificação é fundamentada principalmente por características de alimentação destes motores. Cada grupo possui ainda ramificações internas com características que atendem as mais variadas exigências, sendo um mais vantajoso em detrimento de outro para determinadas aplicações. A Figura 5 apresenta o universo tecnológico dos motores elétricos existentes, os três tipos e suas diversas ramificações.

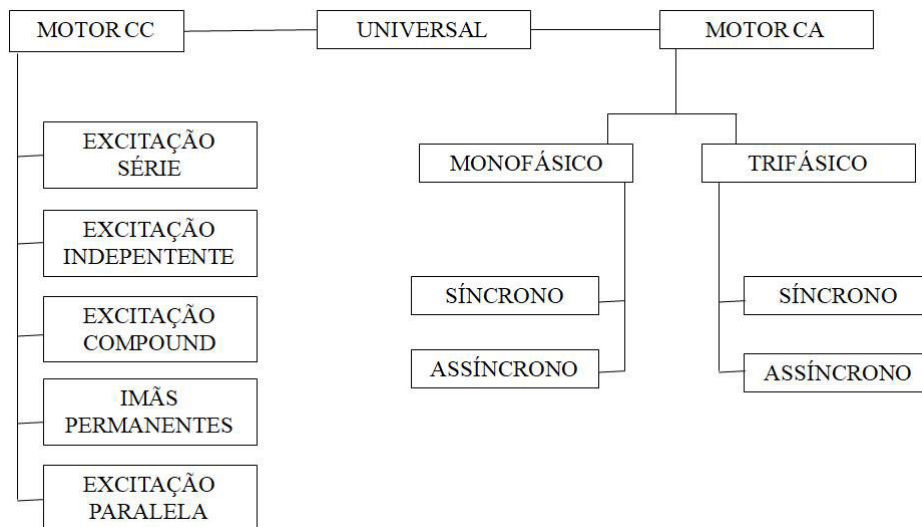


Figura 5: Universo tecnológico dos motores elétricos

Fonte: Próprio Autor

3.1 Motores de Corrente Contínua

Diferentemente dos motores CA na qual a velocidade é controlada pela frequência, forçando um controle mais complexo, com os motores CC a velocidade pode ser facilmente controlada através da variação da tensão aplicada ao motor. Por esse motivo, mesmo a alimentação das redes sendo feita por corrente alternada, os motores de corrente contínua ainda são muito empregados.

Esses motores poluem pouco o meio ambiente e sob o aspecto físico são normalmente pequenos. A Figura 6 ilustra um motor de corrente contínua e as partes que o compõe: um estator de ímã permanente (*stator magnets*), um rotor que carrega o enrolamento da armadura (*rotor winding*), um comutador (*commutator*) e as escovas (*brushes*) (MOHAN, 2012).

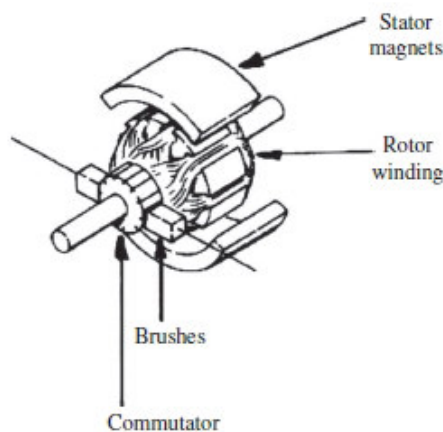


Figura 6: Vista expandida de um motor CC

Fonte: (MOHAN, 2012)

Eles são muito utilizados na indústria quando se faz necessário exercer o controle fino (efetivo) da velocidade. Como exemplo, cita-se a indústria de papel, máquinas têxteis, bobinadeiras e tornos (MAMEDE, 2010). Além destas aplicações, também são empregados em laminadoras de aço, tração ferroviária e uma ampla variedade de acionamentos industriais, robótica, impressoras e servos de precisão (HUGHES; DRURY, 2019), bem como na indústria de brinquedos.

Nesta presente dissertação, o motor CC utilizado será o que compõe um atuador robótico do tipo *harmonic-drive*. Atuadores elétricos do tipo moto-reductor *harmonic-drive* têm grande utilização na robótica, além de, comparado a moto-redutores convencionais, apresentarem excelentes características como alta relação de redução de velocidade, minimização de folga, alta precisão, pequeno porte, alta capacidade de torque, operação silenciosa e sem vibração e alta eficiência (SUNADA et al., 2007).

3.1.1 Modelagem Matemática do Motor de Corrente Contínua

Em sistemas físicos, quando se conhece a dinâmica do mesmo, é possível criar um modelo matemático que represente com precisão ou de forma bem semelhante aquele sistema. Este modelo matemático é definido como um conjunto de equações que representam a dinâmica do sistema. Um modelo matemático para um determinado sistema não é único e pode ser escolhida a forma que o sistema deve ser modelado (OGATA; SEVERO, 1998).

Para a modelagem do motor de corrente contínua foi utilizado um circuito equivalente, ilustrado na Figura 7 retirada de Dorf e Bishop (2011). Nesta figura observa-se a representação do motor, composta de dois segmentos, a parte mecânica e a parte elétrica.

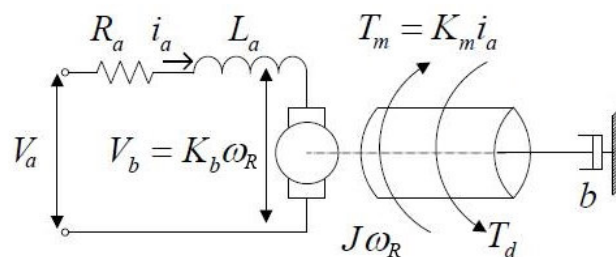


Figura 7: Modelagem do motor CC
Fonte: Adaptado de (DORF; BISHOP, 2011)

O sistema mecânico é composto pelas variáveis:

T_m - torque motor;

T_d - torque de distúrbio;

ω_R - velocidade angular do rotor;

J - inércia do rotor;

K_m - constante de torque;

b - coeficiente de atrito mecânico do motor.

Enquanto que o sistema elétrico é composto pelas variáveis:

V_a - tensão de entrada;

i_a - corrente da armadura;

R_a - resistência da armadura;

L_a - indutância da armadura e

K_b - constante de força contra-eletromotriz.

A partir das equações elétricas do circuito e das mecânicas do movimento, pode-se encontrar a dinâmica de um motor de corrente contínua. Aplicando a lei de Kirchhoff ao circuito, obtêm-se:

$$V_a(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + V_b(t), \quad (1)$$

na qual a relação entre o sistema elétrico e o mecânico é a força contra-eletromotriz, dada por:

$$V_b(t) = K_b \omega_R(t). \quad (2)$$

A partir da segunda Lei de Newton para sistemas rotacionais têm-se:

$$\sum T = J \dot{\omega}_R, \quad (3)$$

$$T_m - T_d - T_b = J \dot{\omega}_R. \quad (4)$$

O torque motor T_m é proporcional a corrente que sai da armadura i_a , logo

$$T_m = K_m i_a, \quad (5)$$

e o torque T_b é o torque originado do efeito do atrito viscoso mecânico do motor. Sua equação é dada por:

$$T_b = b \omega_R(t). \quad (6)$$

Substituindo a equação (6) na equação (4), obtêm-se a seguinte equação diferencial:

$$T_m - T_d - b \omega_R(t) = J \dot{\omega}_R(t), \quad (7)$$

logo,

$$T_m - T_d = J \dot{\omega}_R(t) + b \omega_R(t). \quad (8)$$

As equações (1), (2), (5) e (8) constituem um conjunto básico de equações que modelam o motor CC e é a partir delas que se obtêm os modelos sob forma de diagrama

de blocos e função de transferência (FT) para o motor. Realizando a transformada de Laplace em ambos os lados dessas equações, têm-se, no domínio da frequência,

$$V_a(s) = L_a s I_a(s) + R_a I_a(s) + K_b \omega_R(s), \quad (9)$$

$$T_m(s) = K_m I_a(s), \quad (10)$$

$$T_m(s) - T_d(s) = J s \omega_R(s) + b \omega_R(s). \quad (11)$$

A FT que associa a velocidade de saída do motor CC pela tensão de entrada pode ser obtida como segue. Combinando as equações (10) e (11), supondo a entrada $T_d(s) = 0$, têm-se:

$$I_a(s) = \frac{J s \omega_R(s) + b \omega_R(s)}{K_m}. \quad (12)$$

Substituindo a equação (12) em (9) é possível criar uma relação entre a tensão da armadura e a velocidade angular do rotor:

$$V_a K_m = J L_a s^2 \omega_R(s) + (L_a b + R_a J) s \omega_R(s) + (R_a b + K_b K_m) \omega_R(s). \quad (13)$$

Logo, a função de transferência do sistema é dada por:

$$\frac{\omega_R}{V_a} = \frac{K_m}{(J L_a) s^2 + (L_a b + R_a J) s + R_a b + K_b K_m}. \quad (14)$$

Já a representação em diagramas de blocos do motor CC, apresentada na Figura 8, pode ser obtida combinando os modelos elétrico (1), mecânico (8), com as equações (2) e (5).

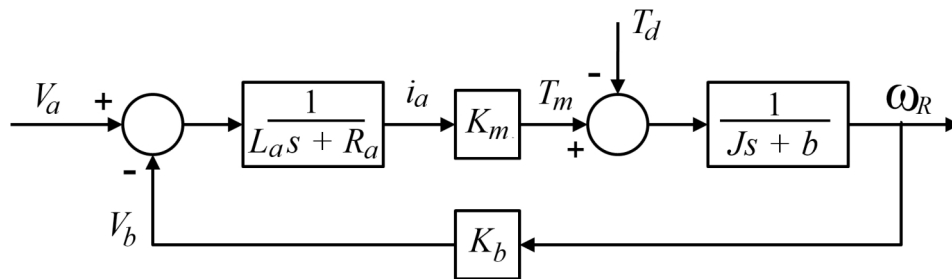


Figura 8: Diagrama de blocos do motor CC

Fonte: Próprio Autor

Os diagrama de blocos são uma forma bastante usual de representação de sistemas dinâmicos. É uma representação visual que facilita a compreensão das equações e suas associações dentro de um sistema. A simulação do modelo do motor CC por diagrama

de blocos permite a análise de diferentes variáveis de saída para diferentes variáveis de entrada.

3.2 Atuadores Robóticos *Harmonic-Drive*

Basicamente, os atuadores robóticos são motores elétricos (podendo ser também pneumáticos ou hidráulicos), e podem ser definidos como dispositivos primários responsáveis pelo movimento e articulação das partes móveis de um robô (GOORVER; WEISS, 1988).

O *harmonic-drive* é um dispositivo mecânico que possui engrenagens que garantem a redução do torque motor aplicado (GOMES et al., 2004). Este sistema de engrenagens oferece a possibilidade do trabalho em baixas velocidades de forma eficiente, ainda que, conceitualmente, considera-se o atrito como crítico. Este atuador apresenta um jogo angular praticamente nulo em razão da grande quantidade de dentes das engrenagens que se encontra em contato permanente dentro do moto-redutor (GOMES; ROSA, 2003).

A utilização de atuadores robóticos deste tipo fornece uma série de vantagens para os projetos, como por exemplo, apresentarem baixo peso, tamanho pequeno, jogo angular praticamente nulo, além da alta capacidade de transmissão de torque em razão do grande número de dentes das engrenagens em contato.

Além das vantagens já mencionadas, outra boa razão para se trabalhar com os moto-redutores *harmonic-drive* é sua vasta utilização no campo da robótica, cujas aplicações variam desde robôs humanóides, veículos aéreos, subaquáticos e terrestres, até em juntas rotacionais de manipuladores.

Porém, de acordo com Machado et al. (2007), a existência de uma zona morta em torque significativa pode degradar o desempenho de leis de controle projetadas, sendo que essa zona morta surge em razão dos atritos internos que têm comportamento fortemente não-linear. Assim, torques inferiores ao atrito estático não são capazes de movimentar cargas.

O atuador usado como base para simulações e experimentos nesta dissertação pode ser visualizado na Figura 9. Este é um modelo PSA-08-080 GH, feito pela empresa *Harmonic-Drive Technologies* e neste modelo têm-se um encoder incremental acoplado ao eixo do rotor, um motor CC e um redutor *harmonic-drive*. A partir de testes experimentais, observou-se a presença de uma zona morta em torque de aproximadamente 8% do torque máximo deste atuador, que de acordo com seu *datasheet*, é de $T_{max} = 2,104 Nm$ na saída do redutor.

A Figura 10 mostra as três estruturas internas que compõem o *harmonic-drive*: o gerador de onda (*wave generator*), uma engrenagem flexível (*flexspline*) e a engrenagem circular (*circular spline*). O gerador de onda, representado pela estrutura da cor amarela na Figura 10, é um rolamento geralmente preso no eixo de entrada, além de ficar encai-



Figura 9: Atuador do tipo moto-redutor Harmonic-Drive
Fonte: Próprio Autor

xado em uma peça elíptica, exercendo a função de conversor de torque. A engrenagem flexível, ilustrada em vermelho, é uma peça cilíndrica com dentes voltados para o lado de fora e com d dentes a menos que a engrenagem circular. Esta última, representada em azul, é um anel rígido com dentes internos, que permitem o encaixe na engrenagem flexível.



Figura 10: Estrutura interna do Harmonic-Drive
Fonte: Adaptado de (ZEILMANN et al., 2010)

A partir da junção destas três estruturas se dá o funcionamento do *harmonic-drive*. O gerador de ondas deforma a lâmina flexível em uma forma elíptica, o que faz com que os dentes desta estrutura sejam acoplados com os dentes da engrenagem circular. Esta engrenagem ou acoplamento entre as duas estruturas é responsável pela transmissão de movimento (ZEILMANN et al., 2010).

Sendo o gerador de onda girado no sentido horário e considerando a circular presa, a engrenagem flexível sofre deformação elástica mudando sua posição de acoplamento em relação à engrenagem circular. Afirmado por Sunada et al. (2007), no momento em que o gerador gira 180 graus no sentido horário, a engrenagem flexível move-se d dentes no sentido anti-horário. Ao final, quando o gerador gira uma volta completa (360 graus), a estrutura flexível move-se $2d$ dentes em relação à engrenagem circular no sentido anti-horário.

O funcionamento e movimentação das três estruturas internas do *harmonic-drive* é melhor compreendido com auxílio da Figura 11. Vê-se as situações onde o eixo está alinhado (0°), em 90° e após completar uma volta (360°). A cada giro no sentido horário

no *wave generator*, o *flexspline* move-se no sentido anti-horário dois dentes em relação ao *circular spline* (ROSSATO, 2009).

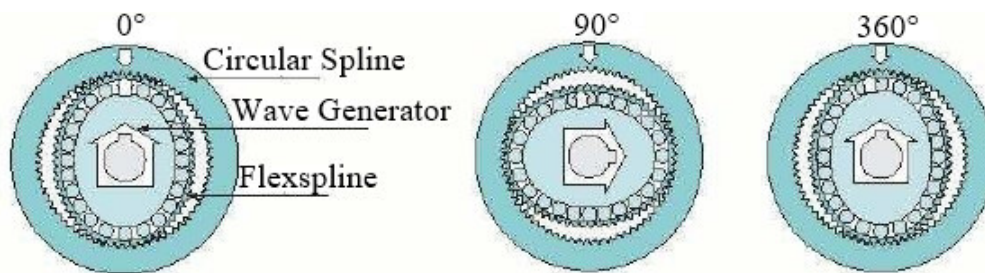


Figura 11: Funcionamento do atuador Harmonic-Drive
Fonte: Adaptado de (ROSSATO, 2009)

3.2.1 Modelagem do Atuador *Harmonic-Drive*

Como o atuador robótico do tipo moto-redutor *Harmonic-Drive* foi o dispositivo utilizado para simulações e experimentos neste trabalho, esta seção se destina a desenvolver a modelagem deste atuador robótico. Esta etapa é um desafio já que modelar a dinâmica de uma planta física nunca é feita com perfeição, principalmente devido às dificuldades impostas por incertezas como as não linearidade derivadas do atrito e a zona morta. Desta forma, é de suma importância compreender de fato as características e o funcionamento dos atuadores robóticos do tipo *harmonic-drive* para então realizar sua modelagem matemática.

Segundo Gomes e Rosa (2003) um motor do tipo moto-redutor pode ser visualizado como um elemento transmissor de movimento contendo uma elasticidade interna K e sua composição compreende um sistema de engrenagens e motor CC.

Neste sistema, representado na Figura 12, o torque motor T_m é aplicado ao rotor do motor com inércia do rotor J , produzindo uma velocidade rotor ω_R e fazendo a transmissão do torque aplicado até a carga acoplada ao eixo do motor. Porém, os sistemas reais são não lineares e nestes casos alguns atritos não lineares estão presentes na dinâmica do movimento, afetando a transmissão de forças. O torque aplicado não é integralmente transmitido a carga acoplada no eixo do motor, o que gera uma velocidade rotor após a caixa de engrenagens ω_r que é diferente da velocidade desejada quando da aplicação do torque.

Para se treinar uma rede neural artificial tornam-se necessários alguns padrões de treinamento e estes padrões podem ser gerados através da identificação experimental do torque de atrito não linear T_{at} . Entretanto, o modelo dinâmico do atuador com dois graus de liberdade não é conveniente para a identificação do modelo de atrito capaz de reproduzir com maior precisão o movimento. Visto que a constante elástica K para estes sistemas é normalmente alta (no caso do *harmonic-drive*, esta gera um modo de vibração de alta frequência, se comparada com a frequência de giro do moto-redutor (ZEILMANN

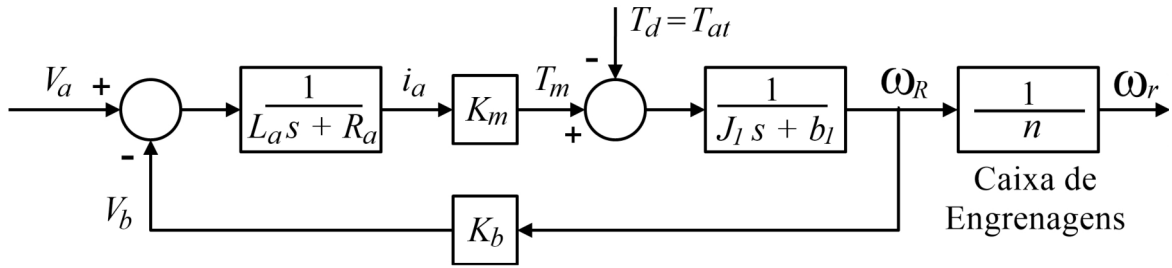


Figura 14: Diagrama de blocos do atuador robótico

Fonte: Próprio Autor

Nesse sistema é apresentada a parte conhecida do atrito mecânico ($b_1 \cdot \omega_r$). Já a variável T_d representa tanto o torque de distúrbio quanto a parte não linear do atrito mecânico (T_{at}). Segundo Zeilmann et al. (2010), considerando a não-linearidade do sistema, uma parcela de atrito entra na equação (15), como um torque de atrito (T_{at}). Se considerarmos que o motor não possui carga mecânica acoplada ao eixo, o torque de distúrbio passa a compreender apenas o torque de atrito não linear T_{at} . Assim, a equação (15) pode ser reescrita como:

$$T_m - T_{at} = J_1 \dot{\omega}_r(t) + b_1 \omega_r(t). \quad (16)$$

3.3 Identificação Paramétrica do Motor CC do Atuador *Harmonic-Drive*

Para realizar futuras simulações, é necessário obter os parâmetros do atuador robótico. O atuador utilizado como referência neste trabalho foi um atuador da Maxon, o mesmo que pode ser visto na Figura 9 do atuador robótico, cujo número de referência é 134971, o qual conta com um *encoder* incremental de 500 pulsos em sua estrutura.

Normalmente, uma forma simples e confiável de obter tais parâmetros, representados no diagrama de blocos da Figura 14, é através do *datasheet* fornecido pelo próprio fabricante. Porém, após buscas pelo *datasheet* do motor supracitado, observou-se que este é muito antigo e descontinuado, tendo as buscas no site do fabricante redirecionado a um outro modelo de motor equivalente, o *RE 25 Ø25 mm, Graphite Brushes, 20 Watt*.

Por questões de confiabilidade e garantia de que os parâmetros utilizados sejam os mais precisos possíveis, decidiu-se realizar testes em laboratório para identificar experimentalmente os parâmetros do atuador robótico. Um bom conhecimento dos parâmetros do modelo é fator determinante para resultados eficientes e para realização da validação do modelo. Assim, foram realizados diferentes ensaios para identificação dos parâmetros elétricos e mecânicos, utilizando a bancada de testes representada na Figura 15. Esta bancada é composta dos seguintes elementos:

- Computador;

- Microcontrolador Arduino Due;
- Atuador Robótico PSA-08-080 GH;
- Protoboard;
- Ponte H Modelo L298N;
- Resistores;
- Fios *Jumpers* de Conexão;
- Multímetro;
- Fonte CC Controlada.

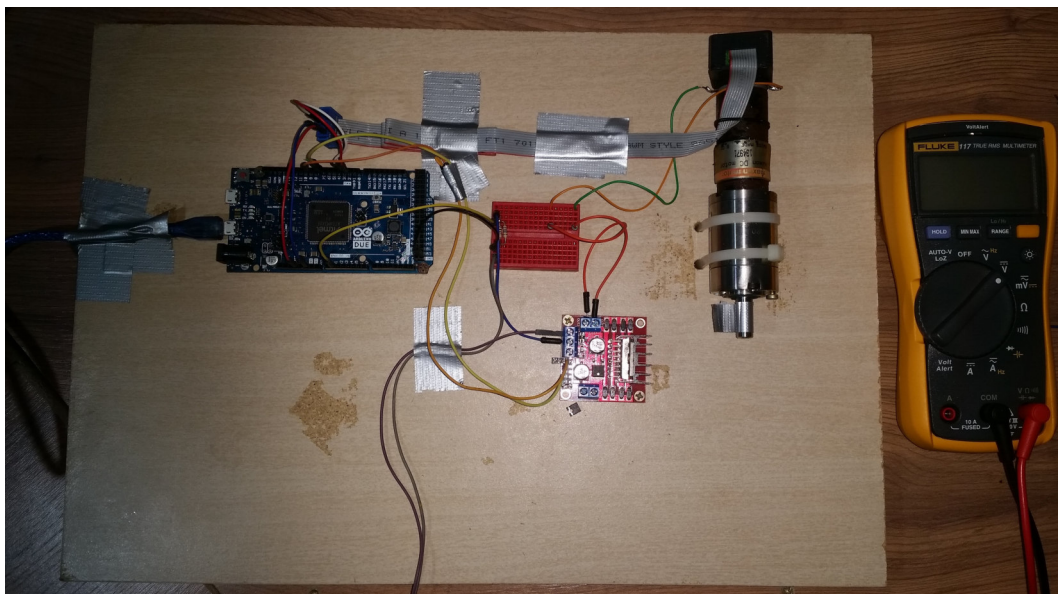


Figura 15: Bancada de testes
Fonte: Próprio Autor

Os parâmetros do motor CC foram obtidos pelo ensaio a vazio e pelo ensaio de rotor bloqueado. A partir destes ensaios foi adquirido os valores de velocidade angular do eixo do motor ω_r , corrente de armadura i_a e tensão da armadura V_a . Vale ressaltar que durante todos esses ensaios, a caixa de engrenagens do conjunto do atuador robótico foi mantida conectada ao motor CC, o que significa dizer que os parâmetros identificados, como por exemplo inércia do rotor J_1 e coeficiente de atrito mecânico b_1 , correspondem ao sistema motor CC e caixa de engrenagens.

3.3.1 Resistência da Armadura R_a

Para o cálculo da resistência da armadura R_a foi realizado o teste de rotor bloqueado do atuador robótico. Este teste consiste em bloquear o rotor do atuador e posteriormente aplicar uma tensão reduzida aos terminais do motor capaz de induzir uma corrente de armadura i_a equivalente a corrente nominal do motor. É possível concluir com auxílio da Figura 7 que, estando o rotor bloqueado, a velocidade $\omega_r = 0$. Pela equação (2), têm-se $V_b = 0$. Nesta condição a tensão V_a é aplicada somente ao resistor R_a . Com esta configuração, o valor da resistência da armadura pode ser calculado pela lei de Ohm a partir dos valores obtidos no ensaio. Tendo encontrado $i_a = 1,2A$ e $V_a = 2,86V$, e sendo a equação de R_a expressa na seguinte forma:

$$R_a = \frac{V_a}{i_a} \quad (17)$$

$$R_a = 2,383\Omega \quad (18)$$

3.3.2 Indutância da Armadura L_a

Para obtenção da indutância da armadura, novamente é realizado o teste de rotor bloqueado do atuador robótico. Foi aplicado um degrau de tensão nos terminais do motor a fim de monitorar a estabilização da corrente de armadura i_a . Neste momento, é possível obter a constante de tempo da evolução da corrente τ_e , que representa o tempo em que a corrente alcança 63,2% do valor em estado estacionário. Comparando o sistema com a forma padrão de sistemas de primeira ordem, observa-se que $\tau_e = L_a/R_a$, o que permite dizer que a indutância da armadura é dada por:

$$L_a = \tau_e R_a \quad (19)$$

Após o ensaio realizado, foi identificado o valor de $\tau_e = 0,001s$. Seguindo a equação (19), e utilizando a resistência da armadura previamente calculada - equação (18) - a indutância da armadura é:

$$L_a = 0,002383H \quad (20)$$

3.3.3 Coeficiente de Atrito Mecânico b_1

Para a obtenção do coeficiente de atrito mecânico b_1 , foi medida a potência elétrica P_{el} absorvida pelo motor após o ensaio a vazio, que representa o acionamento de motor CC e caixa de engrenagens sem cargas acopladas. Desprezando as perdas magnéticas do sistema, pode-se afirmar que a potência elétrica P_{el} representa o somatório das perdas do cobre P_{Cu} com a potência mecânica P_{mec} . Logo:

$$P_{el} = P_{Cu} + P_{mec} \quad (21)$$

sendo que $P_{el} = V_a i_a$ e $P_{Cu} = R_a i_a^2$.

Visto que neste ensaio o motor está girando sem cargas acopladas e a uma velocidade constante, pode-se dizer que a potência mecânica é equivalente as perdas por atrito nos mancais do motor:

$$P_{mec} = T_b \omega_r \quad (22)$$

Substituindo os termos apresentados na equação (22) em (21), o coeficiente de atrito mecânico é dado por:

$$b_1 = \frac{V_a i_a - R_a i_a^2}{\omega_r^2} \quad (23)$$

Desta forma, foi realizada uma bateria de testes e análises a fim de medir encontrar a velocidade rotor ω_r e a corrente da armadura i_a para diferentes valores de tensão da armadura V_a aplicados. Alguns exemplos destes valores obtidos são representados na Tabela 7. A partir destes valores e da equação (23), foi possível encontrar um valor médio para o coeficiente de atrito, sendo este:

$$b_1 = 9e^{-6} Nms/rad \quad (24)$$

Tabela 7: Valores Obtidos no Ensaio a Vazio para Identificação de b_1

	$R_a[\Omega]$	$V_a[V]$	$i_a[A]$	$\omega_r[rad/s]$
Teste 1	2.3833	9	0,3	525
Teste 2	2.3833	2	0,073	121

Fonte: Próprio Autor

3.3.4 Constante de Torque K_m

Novamente o ensaio do motor a vazio será usado para calcular a constante de torque K_m . Como citado anteriormente, a partir deste ensaio são medidas a tensão, corrente e velocidade. Mantendo a velocidade rotor em um valor constante, pode-se assumir que o torque T_m é igual ao torque do atrito nos mancais do rotor T_b . Assim, conclui se que:

$$K_m i_a = b_1 \omega_r \quad (25)$$

chegando assim na seguinte equação:

$$K_m = \frac{b_1 \omega_r}{i_a} \quad (26)$$

Desta forma, o valor da constante de torque é:

$$K_m = 0,01575 Nm/A \quad (27)$$

3.3.5 Constante de Força Contra-Eletromotriz K_b

Como a potência mecânica convertida no eixo do motor é igual a potência elétrica relacionada a força contra-eletromotriz V_b , pode-se dizer que:

$$P_{mec} = T_m \omega_r = V_b i_a \quad (28)$$

Substituindo as equações (5) e (6) na equação (28), têm-se:

$$(K_m i_a) \omega_r = (K_b \omega_r) i_a \quad (29)$$

Assim, tanto constante de torque quanto a constante de força contra-eletromotriz apresentam o mesmo valor:

$$K_b = K_m = 0,01575 Vs/rad \quad (30)$$

3.3.6 Inércia do Rotor J_1

Por fim, ainda pelo ensaio do motor a vazio, é obtido o parâmetro da inércia do rotor. Neste teste é utilizado o método do retardamento, que consiste em acelerar e estabilizar a velocidade do rotor, e logo em seguida desligar sua alimentação para que a resposta em velocidade chegue a zero. A partir de então, analisar a curva característica da desaceleração (Figura 16) a fim de observar a queda da velocidade no tempo. Busca-se obter a constante de tempo τ_d , que representa o tempo da velocidade decair até 36,8% da velocidade em estado estacionário. No teste realizado, a velocidade inicia o decaimento no tempo aproximado de 23 segundos. Se o sistema for comparado a uma FT padrão de um sistema de primeira ordem, pode-se dizer que a contante de tempo $\tau_d = J_1/b_1$, e a inércia do rotor é dada por:

$$J_1 = \tau_d b_1 \quad (31)$$

Como $\tau_d = 0,09s$, o valor identificado para inércia do rotor J_1 é:

$$J_1 = 8,1e^{-7} Kg.m^2 \quad (32)$$

A Tabela 8 traz uma visão geral dos parâmetros identificados nesta seção a partir dos diversos ensaios. Para efeitos de comparação, foi apresentado no anexo C deste trabalho o *datasheet* do motor equivalente ao utilizado. Observa-se coerência nos parâmetros identificados e os fornecidos pelo fabricante. Reforça-se que na identificação realizada neste trabalho, os dados encontrados são equivalentes ao atuador robótico, não apenas ao motor CC. Isto pode explicar algumas diferenças nos valores dos parâmetros se comparados.

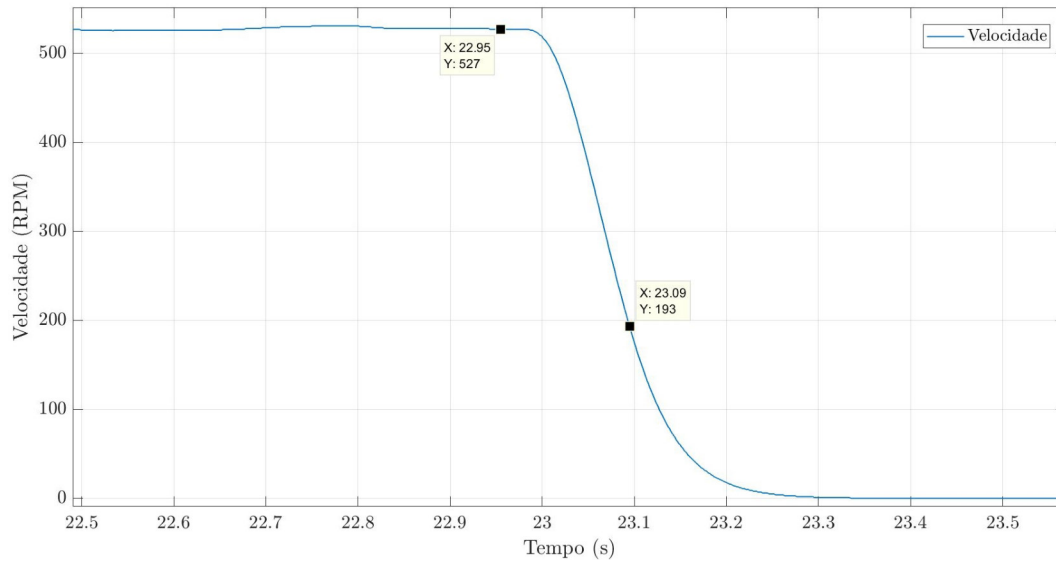


Figura 16: Curva característica da desaceleração

Fonte: Próprio Autor

Tabela 8: Parâmetros Identificados do Atuador Robótico

Parâmetro	Descrição	Valor
R_a	Resistência da Armadura	$2,38333\Omega$
L_a	Indutância da Armadura	$0,00238H$
b_1	Coefficiente de Atrito Mecânico do Atuador	$9e^{-6}N.m.s/rad$
K_m	Constante de Torque	$0,01575N.m/A$
K_b	Constante de Força Contra-Eletromotriz	$0,01575V.s/rad$
J_1	Inércia do Atuador Robótico	$8,1e^{-7}Kg.m^2$

Fonte: Próprio Autor

3.3.7 Validação dos Parâmetros Identificados

Para validar os parâmetros identificados, foram realizadas comparações entre os resultados de simulações e os resultados dos experimentos, em malha aberta (MA), para sinais em degrau e senoidais. As comparações podem ser visualizadas nas figuras a seguir. No primeiro teste, Figura 17, foram aplicadas entradas em degrau nos valores de 9, 8, 6, 4 e 2V de tensão V_a em ambos sentidos de rotação, e os resultados foram coerentes entre as velocidades e correntes experimentais (linhas azuis) e as velocidades e correntes simuladas (linhas vermelhas).

Enquanto que no segundo teste, Figura 18, foi aplicada uma entrada em sinal senoidal de amplitude 5V de tensão V_a e frequência de 0,1Hz, e os resultados novamente se mostraram semelhantes entre as velocidades e correntes experimentais (linhas azuis) e as velocidades e correntes simuladas (linhas vermelhas).

Estes testes validam o sistema simulado, que se comportou de forma coerente se comparado aos experimentos realizados com o atuador robótico considerado neste trabalho. Pode-se destacar o efeito da zona morta observado entre as curvas de velocidade do eixo

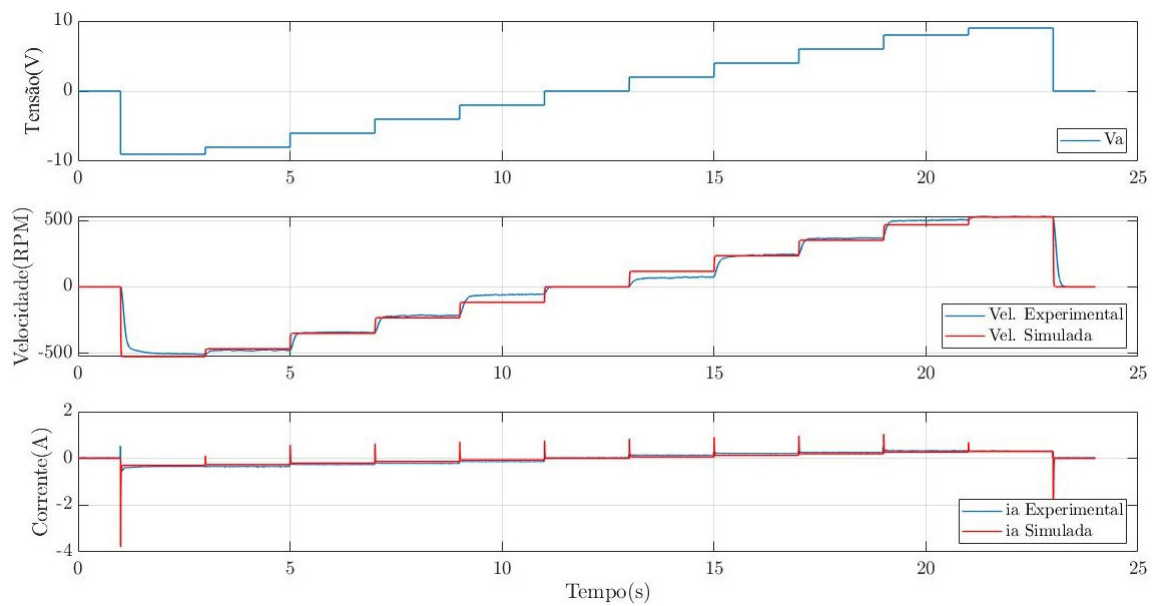


Figura 17: Validação dos parâmetros do atuador robótico para entradas em degrau. Simulação (linhas vermelhas) x Experimento (linhas azuis)

Fonte: Próprio Autor

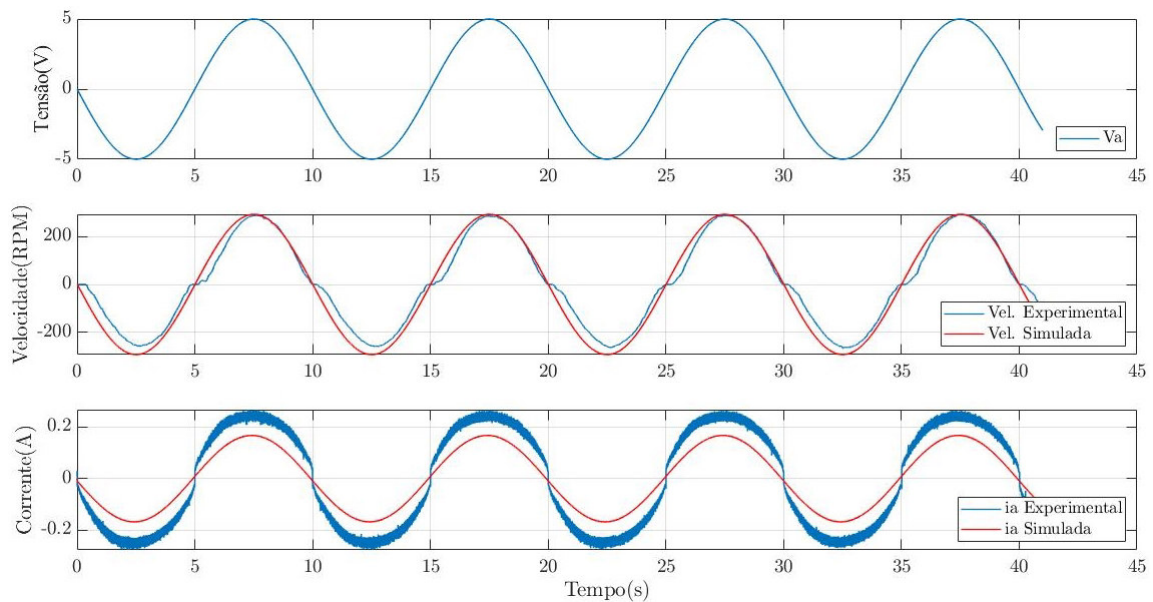


Figura 18: Validação dos parâmetros do atuador robótico para entradas em sinais senoidais. Simulação (linhas vermelhas) x Experimento (linhas azuis)

Fonte: Próprio Autor

experimental e simulada, no cruzamento por zero, na curva de velocidade da Figura 17, já que esse efeito não foi ainda inserido na simulação. Este comportamento será modelado nos capítulos que se seguem. Adicionalmente, observa-se pequena defasagem da velocidade simulada com relação à velocidade experimental, na curva de velocidade da Figura 18, possivelmente causada pelo filtro utilizado para cálculo da velocidade. A curva da corrente na Figura 18 mostra um ruído na corrente medida, o qual é resultado do chaveamento do conversor do tipo Buck utilizado nos testes de bancada. Uma pequena diferença de amplitude também é observada e pode ser resultado de imprecisão dos parâmetros obtidos ou também, efeitos do conversor, o qual não foi modelado. Por fim, ressalta-se que a variável de saída do modelo simulado apresenta boa correlação com o modelo experimental como já observado na curva de velocidade da Figura 18 e pode-se afirmar que os parâmetros obtidos representam o sistema experimental satisfatoriamente.

4 MODELOS DE ATRITO

A modelagem do atrito é extremamente importante para a dinâmica e o controle de sistemas mecânicos. Como afirmado em Fronza (2015), tendo o atrito uma forte influência no desempenho e comportamento de sistemas, representações e previsões de fenômenos são importantes em simulações e análises, além de poderem ser utilizados em estratégias de compensação.

Modelar corretamente o atrito pode ser visto como o grande desafio para se realizar o controle preciso de atuadores robóticos. Em um sistema robótico, o atrito é um dos grandes problemas a ser estudado e entendido, especialmente devido a sua complexidade e ao crescente aumento da procura por processos que exigem alta precisão de posicionamento e manipulação (ZEILMANN et al., 2010). Em relação a robôs, o atrito pode se apresentar em diversas partes que o compõem, como os rotores, as transmissões, os mancais de sustentação e os contatos das articulações.

Para realizar simulações que representem adequadamente o sistema analisado é necessário dispor-se de um bom modelo para o atrito. Isto permite que elas possam ser utilizadas em várias tarefas, tais como determinar os ganhos para os controladores, estudar o seu desempenho dinâmico, analisar a estabilidade e desenvolver métodos de compensação (PERONDI, 2002).

Devido a natureza não linear do atrito, desenvolver um modelo de atrito baseando-se nos primeiros princípios da mecânica clássica é simplesmente impossível. Logo, ao criar um modelo de atrito, já se busca uma forma aproximada da representação, na busca de assegurar uma representação confiável, ainda que não perfeita, para ser utilizada em aplicações de controle.

De uma forma geral, quando se trabalha com sistemas de controle de movimento, é necessário dar muita atenção aos efeitos oriundos do atrito. Esta é a razão pela qual o atrito é um fenômeno que tem sido bastante estudado nos últimos anos (MACHADO et al., 2007). O comportamento do atrito foi amplamente debatido durante o século XX com destaque aos trabalhos Armstrong-Helouvry (1993), Armstrong-Helouvry e Dupont (1993), Armstrong-Hélouvry, Dupont e Wit (1994), Wit et al. (1995) e Besançon-Voda e Besançon (1999), que realizavam experimentos em condições ideais, utilizando su-

perfícies limpas e estáveis, além de uso de velocidades constantes. Em adição os trabalhos que estudam os efeitos do atrito a partir de testes com atuadores do tipo *harmonic-drive* como Taghirad e Belanger (1996), Dhaouadi (2003), Gervini, Gomes e Rosa (2003) e Gomes e Rosa (2003).

Atualmente, muitas pesquisas estão sendo desenvolvidas com base no estudo do atrito dinâmico, já que muitos fenômenos não lineares dificultam a modelagem do atrito enquanto que muitos processos estão exigindo alto grau de precisão destes modelos, o que acarreta aumento da importância de uma representação do atrito de elevada precisão.

Alguns fenômenos do atrito não aparecem para experimentos realizados com velocidade constante. Abaixo mostram-se algumas características presentes no atrito (OLSSON; ASTROM, 1996).

- Pré-deslizamento (*pre-sliding*);
- Efeito de *stribeck*;
- Atraso de fricção (*frictional lag*);
- Força de ruptura (*break-away*);
- Efeito prende-desliza (*stick-slip*).

Pré-deslizamento: ao iniciar o movimento, observam-se deslocamentos muito pequenos antes do deslizamento entre as superfícies, chamados de micro-deformações. Verifica-se também uma relação de dependência na forma de histerese da força de atrito em relação ao deslocamento. A Figura 19, retirada de Fronza (2015), demonstra o comportamento de pré-deslizamento, sendo as linhas pontilhadas os deslocamentos permanentes, resultantes da interrupção de forças aplicada .

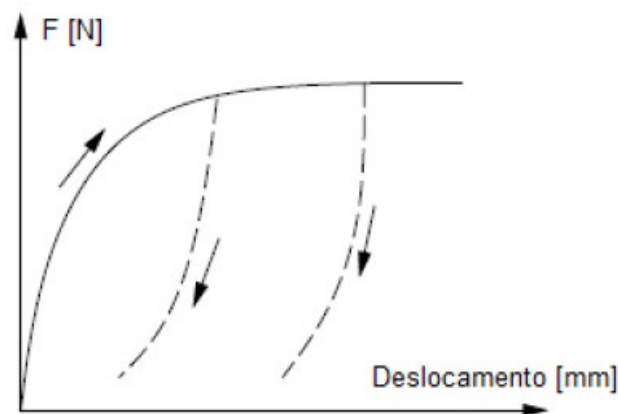


Figura 19: Efeito de pré-deslizamento
Fonte: (FRONZA, 2015)

Efeito de *stribeck*: na presença de superfícies lubrificadas, no instante em que o corpo rompe o atrito estático até chegar ao atrito viscoso, acontece uma rápida redução da força de atrito, que é conhecida como curva de *stribeck* (MIOTTO, 2009).

Atraso de fricção: descrito na Figura 20, este fenômeno demonstra o comportamento do atrito quando a velocidade é variada em uma única direção. Percebe-se uma relação de histerese quando ocorre variação de velocidade, devido à existência de um atraso entre a variação da velocidade e a variação da força de atrito. O ciclo de histerese pode aumentar dependendo da viscosidade e da frequência da variação da velocidade (ZEILMANN et al., 2010).

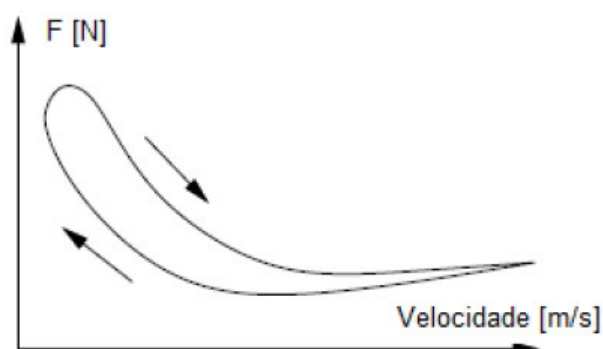


Figura 20: Atraso de fricção
Fonte: (FRONZA, 2015)

Força de ruptura: o atrito estático se define como o atrito antes do movimento entre duas superfícies que se encontram em contato. A força de ruptura, ou força de quebra, é a força necessária para vencer o atrito estático e iniciar o movimento (CASANOVA et al., 2007). A Figura 21 ilustra o fenômeno da força de ruptura.

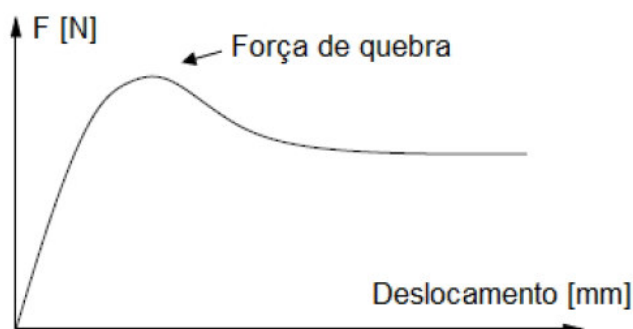


Figura 21: Força de ruptura
Fonte: (FRONZA, 2015)

Efeito *Stick-Slip*: é o comportamento que corresponde à transição do regime de pré-deslizamento para o regime de deslizamento e vice-versa. Estas transições ocorrem em duas situações: Nas reversões de velocidade, isto é, quando a velocidade aproxima-se de

zero e a direção do movimento muda e, com o aumento da força de atrito até alcançar a força de ruptura (ZEILMANN et al., 2010).

Visto esses fenômenos relacionados ao atrito, pode-se concluir que um bom modelo de atrito é aquele que reproduz a maior parte destes fenômenos que existem na realidade física.

Com o objetivo principal de estudar os efeitos causados pelo atrito e buscar alternativas de compensá-lo, surgiram ao longo dos anos diversas técnicas para reproduzir um comportamento preciso do fenômeno (GOMES, 2011). Desta forma, este trabalho apresentará dois diferentes modelos de atrito: o modelo de Lugre e o modelo de Gomes.

4.1 Modelo de Atrito de Lugre

Um modelo dinâmico que descreve os efeitos do atrito de forma mais precisa, principalmente para baixas velocidades e em reversão de velocidades, foi proposto por Wit et al. (1995). A este modelo foi dado o nome de Lugre devido ao fato de ser originário de pesquisas realizadas no Instituto de Tecnologia de Lund, na Suécia, e no Laboratório de Controle Automático de Grenoble, na França.

O modelo de Lugre tem como principal característica a consideração do movimento em regime de pré-deslizamento, que não é considerado nos modelos clássicos para o atrito, geralmente baseados em mapas estáticos que descrevem a relação entre a força de atrito e a velocidade (PERONDI, 2002). Isto significa que além de reproduzir a maioria dos comportamentos não-lineares observados, como o efeito de *strobeck* e o atrito viscoso, a histerese, força de ruptura, este modelo vai além e observa a existência do contato através de hastes elásticas e o fenômeno de pré-deslizamento, já descritos anteriormente no modelo de Dahl.

O fenômeno elástico relata que, sob o ponto de vista microscópico, dois corpos rígidos fazem contato por meio de hastes elásticas (Figura 22). Tais hastes são opostas umas às outras e, quando ocorre movimento entre as partes, exercem forças de atrito proporcionais às suas tensões, que tendem a deformar as hastes elásticas na região de contato. Caso esta deformação seja suficientemente grande, os corpos começam a deslizar.

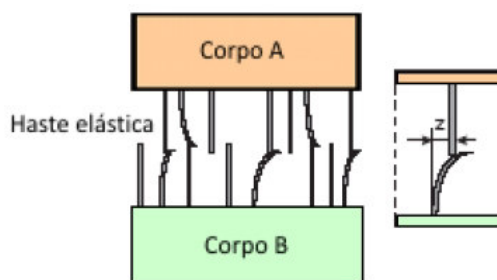


Figura 22: Comportamento elástico do modelo de Lugre
Fonte: (GOMES, 2011)

Uma função $g(V_r)$ é incluída no sistema para representar o fenômeno do efeito *stirbeck*, onde V_r é a velocidade relativa entre dois corpos e δ o parâmetro que altera a forma da passagem do atrito estático para o seco (ZEILMANN et al., 2010). A variável f_c é o coeficiente do atrito seco, f_s é o coeficiente do atrito estático e V_s é a velocidade *stirbeck*. A função é dada pela equação (33):

$$g(V_r) = f_c + (f_s - f_c)e^{\left(\frac{|V_r|}{V_s}\right)^\delta} \quad (33)$$

O modelo considera que o atrito está relacionado também com a posição em razão do contato entre as hastes, e não somente com a velocidade. Quando a tensão com que uma haste contribui excede um determinado valor, há uma ruptura entre a união das hastes. Sendo assim, a equação do modelo de Lugre possui uma variável z que modela a deflexão média das hastes (OLSSON; ASTROM, 1996).

A partir da inclusão da variável z , da constante elástica que relaciona atrito com deslocamento antes do deslizamento τ_0 , bem como da função *stirbeck* $g(V_r)$, a equação de deflexão média das hastes para o modelo de Lugre fica representada como na equação (34):

$$\frac{dz}{dt} = V_r - \frac{\tau_0 |V_r|}{g(V_r)} z \quad (34)$$

Por fim, acrescentam-se os parâmetros τ_1 , que representa um termo de amortecimento no movimento de pré-deslizamento e τ_2 , termo de atrito viscoso, na equação do modelo. Portanto, a força de atrito é proporcional à deformação média e depende também da taxa de variação da deformação média. Além disso, o efeito do atrito viscoso é inserido a partir de uma função dependente da velocidade (ZEILMANN et al., 2010).

Logo, considerando todos os parâmetros supracitados, o modelo de atrito de Lugre fica definido como na equação (35), sendo F_a o torque de atrito não linear calculado pelo modelo de Lugre:

$$F_a = \tau_0 z + \tau_1 \frac{dz}{dt} + \tau_2 V_r \quad (35)$$

Em síntese, o modelo de Lugre é o modelo de Dahl modificado, no qual acresceram-se o efeito de *stirbeck* e o atrito viscoso. É importante ressaltar que há uma dificuldade na identificação dos parâmetros τ_0 , τ_1 e δ , principalmente, sendo que a maioria dos artigos que utilizam este modelo não deixa claro como é que estes parâmetros foram identificados (GOMES, 2011).

4.2 Modelo de Atrito de Gomes

O modelo de Gomes, que foi inicialmente proposto em Gomes (1995), se diferencia dos modelos clássicos por sua forma de representação do atrito. O modelo representa o

atrito como um efeito pertencente à dinâmica própria do sistema, ou seja, o atrito não é representado nas equações diferenciais como um torque externo, mas como parte integrante da dinâmica própria do sistema (ZEILMANN et al., 2010).

Uma das principais contribuições do modelo de Gomes é a representação do fenômeno *stick-slip*, um comportamento não-linear comum em atuadores robóticos operando em baixas velocidades (GOMES; ROSA, 2003). Neste fenômeno a velocidade rotor vai a zero quando o torque resultante da articulação é inferior ao atrito estático, permanecendo assim até o momento em que o torque sai da zona morta, onde a velocidade atinge um limite, a partir do qual ocorrerá o deslizamento.

Para encontrar os parâmetros dos modelos de atrito, é necessário a obtenção da curva característica estática do atrito. Para isso, realizou-se experimentos na qual foram aplicados vários torques constantes em um atuador *harmonic-drive*, como o representado anteriormente. Os valores de torque variaram na faixa de 8% a 20% do torque máximo, nos dois sentidos (positivo e negativo), com o objetivo de medir a velocidade após o regime estacionário e, assim, obter a curva estática que relaciona velocidade com torque aplicado.

Para escolher os valores de torque aplicados é necessário se atentar a limitações físicas do atuador, que após determinado valor de torque, apresentava uma limitação da resposta em velocidade.

A partir dos resultados experimentais, realizados em malha aberta, interpolam-se os pontos nos dois sentidos e têm-se dois polinômios considerando que o índice i pode ser para velocidade positiva (p) ou negativa (n). Essa é outra característica do modelo de Gomes: representar o atrito em dois sentidos diferentes de rotação do atuador, horário ou positivo e anti-horário ou negativo (GOMES; ROSA; ALBERTINI, 2006).

Assim, a equação (36), representa o torque de atrito, para cada sentido de rotação:

$$T_{at} = f_{ci} + f_{vi}\dot{\theta} + c_i\dot{\theta}^2 \quad (36)$$

Onde:

f_{ci} é o torque de atrito seco (velocidade positiva ou negativa);

f_{vi} é o coeficiente de atrito viscoso (velocidade positiva ou negativa);

c_i é o coeficiente de atrito do termo quadrático (velocidade positiva ou negativa).

Realizando a análise da curva estática de atrito, considera-se o torque de atrito igual ao torque motor aplicado para cada velocidade estacionária. Isso se deve ao fato de que não existem outros torques externos além do motor. As figuras 23, 24 e 25, demonstram as curvas estáticas que relacionam o torque motor e a correspondente velocidade em estado estacionário.

Vale ressaltar que neste experimento foram aplicados quatro torques motores diferentes, para cada sentido de rotação, objetivando a obtenção da curva estática. Porém, ideal-

mente, um maior número de torques aplicados no experimento garantirá uma identificação paramétrica mais confiável, principalmente dos coeficientes de atrito do termo quadrático c_i . No caso deste experimento, com poucos pontos formando a curva característica estática, não foi possível obter os valores de c_p e c_n .

Outro ponto importante de ser mencionado é a necessidade de realizar o “aquecimento” do motor, que consiste em aplicar entradas como de torque senoidal por um tempo antes de realizar os experimentos. O atuador “aquecido” produz menos atrito e resultados mais condizentes com a realidade que com o motor “frio” (repouso).

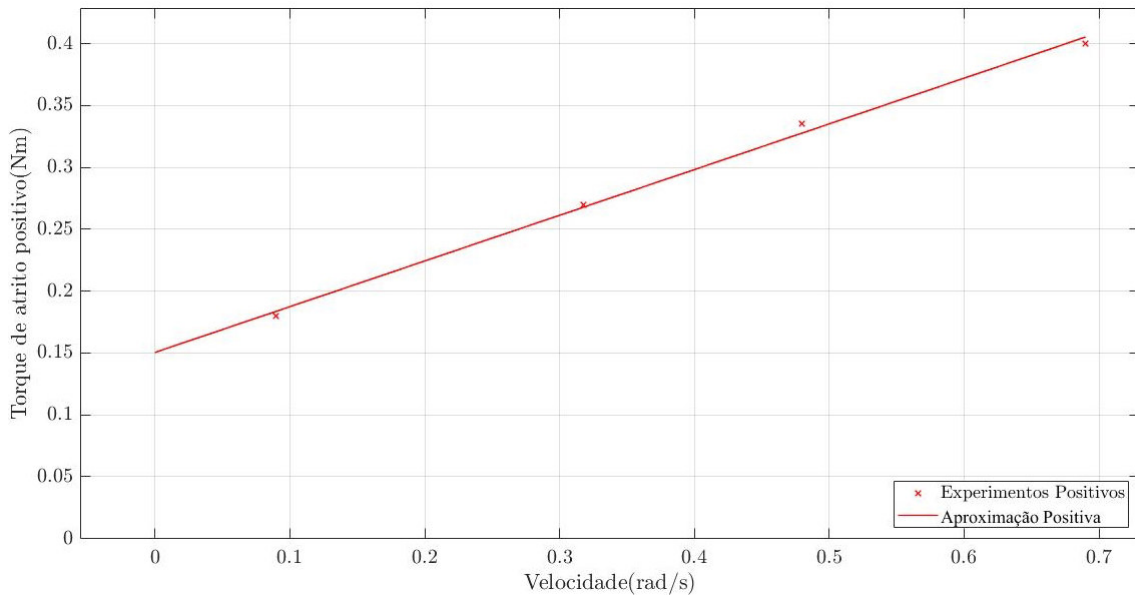


Figura 23: Curva estática do atrito em função da velocidade positiva do rotor

Fonte: Próprio Autor

Os parâmetros para o torque de atrito do modelo de Gomes foram identificados experimentalmente a partir da curva estática do atrito e podem ser visualizados na Tabela 9. Para identificação dos parâmetros utilizou-se no Matlab a função *polyfit*, que é uma função que realiza o ajuste dos coeficientes de um polinômio, tendo este qualquer grau, e que sua saída fornece um vetor com coeficientes. Para a identificação dos parâmetros positivos como f_{cp} , f_{vp} e c_p , foi realizada a análise via *polyfit* da curva estática do atrito da Figura 23, enquanto que para parâmetros negativos como f_{cn} , f_{vn} e c_n , a curva analisada via *polyfit* foi a da Figura 24.

O torque de atrito, sob a forma de atrito viscoso variável, é escrita como um efeito dinâmico interno, onde f_v^* é o coeficiente de atrito viscoso variável e $\dot{\theta}_r$ é a velocidade rotor. Sua representação é dada por:

$$T_{at} = f_v^* \dot{\theta}_r \quad (37)$$

Para formular o algoritmo que determina o coeficiente de atrito viscoso variável, segundo Gomes, Rosa e Albertini (2006), existe uma velocidade limite $\dot{\theta}_{lim}$, a partir da

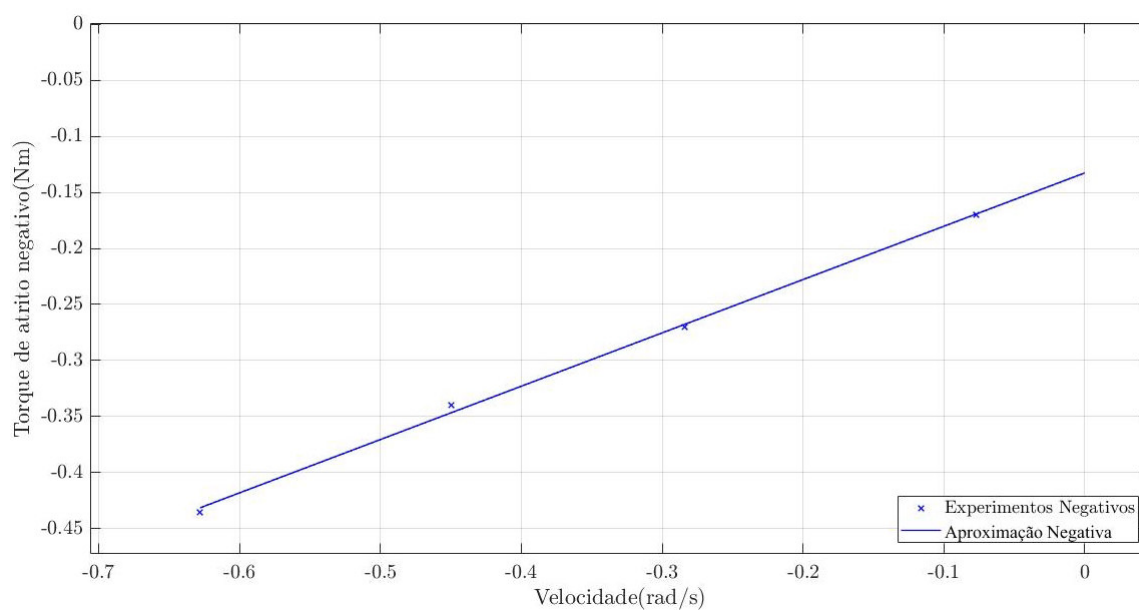


Figura 24: Curva estática do atrito em função da velocidade negativa do rotor
Fonte: Próprio Autor

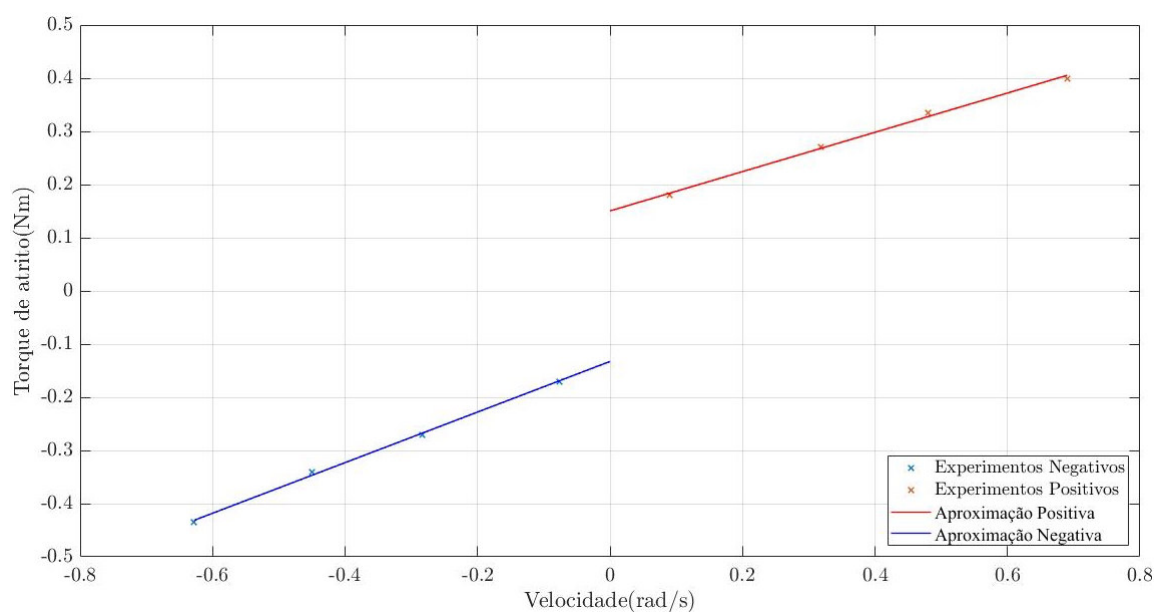


Figura 25: Curva estática do atrito em função da velocidade em ambos sentidos de rotação
Fonte: Próprio Autor

Tabela 9: Parâmetros Identificados pela Curva Estática para o Torque de Atrito do Modelo de Gomes

Parâmetro	Descrição	Valor
f_{cp}	Atrito Seco Positivo	$0,1505 \text{ Nm}$
f_{cn}	Atrito Seco Negativo	$-0,1326 \text{ Nm}$
f_{vp}	Atrito Viscoso Positivo	$0,3694 \text{ Nm.s/rad}$
f_{vn}	Atrito Viscoso Negativo	$0,4756 \text{ Nm.s/rad}$
c_p	Coefficiente de Atrito Positivo	$0 \text{ Nm.s}^2/\text{rad}^2$
c_n	Coefficiente de Atrito Negativo	$0 \text{ Nm.s}^2/\text{rad}^2$

Fonte: Próprio Autor

qual, o fenômeno *stick-slip* deixa de ocorrer no momento em que se atinge esta velocidade. *Stick* representa o “movimento preso” e ocorre quando o torque resultante é menor do que o atrito seco e quando a velocidade rotor $\dot{\theta}_r$ é menor do que a velocidade limite $\dot{\theta}_{lim}$. A Figura 26 ilustra os efeitos *stick* (nas trajetórias C e D) e *slip* (trajetória B).

De acordo com Zeilmann et al. (2010), o efeito *stick* é quando a velocidade vai a zero e só sai do repouso a partir do momento em que o torque passa a ser maior do que o atrito seco por aproximação, ou estático. Ainda afirmado por Zeilmann et al. (2010), neste instante, o efeito *stirbeck* faz com que a trajetória *slip*, ou seja, de deslize, seja menos amortecida do que a trajetória *stick*.

É importante observar que existem transições verticais da trajetória B para C ou D, quando um modo *slip* se transforma em *stick*, e transições de C ou D para B, quando a transformação é contrária (GOMES, 2011). Essas transições verticais podem acontecer até que se alcance uma velocidade nula.

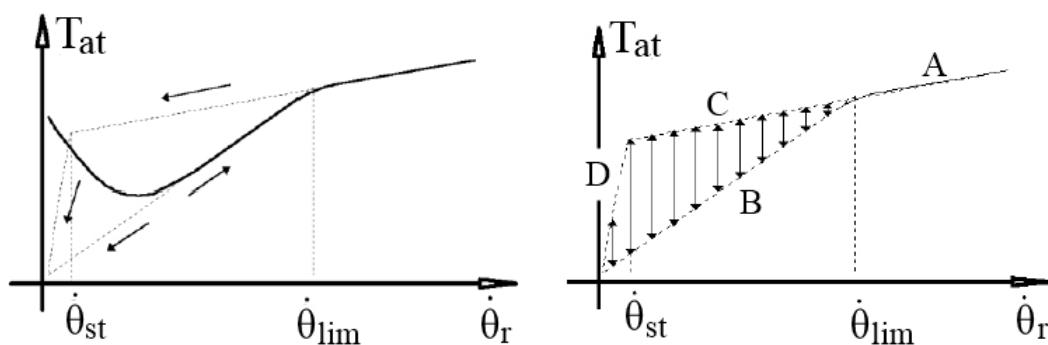


Figura 26: Efeito *Stick-Slip*

Fonte: Adaptado de (ZEILMANN et al., 2010)

Proposto por Gomes, Rosa e Albertini (2006), o algoritmo usado para determinar o coeficiente de atrito viscoso variável f_v^* , representando o atrito, bem como o efeito *stick-slip* em regiões de baixa velocidade, é dado por:

Uma nova variável f_{lim} é encontra no algoritmo, e ela representa o maior valor que pode assumir o coeficiente de atrito viscoso variável f_v^* para que exista o efeito *stick-slip*.

Algorithm 1: Determinando o coeficiente de atrito viscoso variável

```

if  $|\dot{\theta}_r| \geq \dot{\theta}_{lim}$  then
   $f_v^* = \frac{f_{ci} + c_i \dot{\theta}_r^2}{|\dot{\theta}_r|} + f_{vi}$ ; {fora da região de stick-slip (A)}
else
  if  $|T_m| > f_s$  then
     $f_v^* = \frac{f_{ci} + f_{vi} \dot{\theta}_{lim} \text{sign}(\dot{\theta}_r) + c_i \dot{\theta}_{lim}^2}{\dot{\theta}_{lim} \text{sign}(\dot{\theta}_r)}$ ; {trajetória slip (B)}
  end
else
  if  $\dot{\theta}_r = 0$  then
     $\dot{\theta}_r = eps$ ;  $f_v^* = \frac{f_{ci} + c_i \dot{\theta}_r^2}{|\dot{\theta}_r|} + f_{vi}$ ; {trajetória stick (C)}
  end
  if  $f_v^* > f_{lim}$  then
     $f_v^* = f_{lim}$ ; {trajetória stick final (D)}
  end
end
end
end
  
```

Ela pode ser calculada através da equação (38). Outras variáveis encontradas no algoritmo são a velocidade limite - menor velocidade estacionária possível em malha aberta - $\dot{\theta}_{lim}$ e o atrito estático f_s .

$$f_{lim} = \left| \frac{f_{ci} + c_i \dot{\theta}_{st}^2}{\dot{\theta}_{st}} \right| + f_{vi} \quad (38)$$

Sobre a equação (38), $\dot{\theta}_{st}$ é a menor velocidade fisicamente possível. Abaixo deste valor existem apenas deslocamentos micro-elásticos entre os materiais em contato, o que configura a fase de pré-deslizamento (CASANOVA et al., 2007). Esta variável é representada por 5% da menor velocidade estacionária possível em malha aberta e por isso pode ser calculada ao multiplicar a velocidade limite $\dot{\theta}_{lim}$ por um parâmetro γ que é uma porcentagem da menor velocidade possível em malha aberta ($\gamma = 0,05$). Assim, sua forma é dada por:

$$\dot{\theta}_{st} = \gamma \dot{\theta}_{lim} \quad (39)$$

Conforme a equação do atuador com aproximação rígida, a dinâmica pode ser posta em forma matricial (GOMES; ROSA, 2003). O modelo de Gomes fica então representado:

$$\begin{pmatrix} \dot{\theta}_r \\ \ddot{\theta}_r \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-f_v^*(\dot{\theta}_r, T_m)}{J_1} \end{bmatrix} \begin{pmatrix} \theta_r \\ \dot{\theta}_r \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{J_1} \end{pmatrix} T_m \quad (40)$$

Neste modelo, o atrito não é mais motor, como é o caso dos diversos outros modelos, mas pertence à dinâmica própria do sistema (GOMES; ROSA, 2003). Observa-se também que apenas com experimentos em malha aberta é possível obter tanto a curva estática do atrito quanto identificar os parâmetros do modelo.

5 SIMULAÇÕES E EXPERIMENTOS EM MALHA ABERTA COM OS MODELOS DE ATRITO

Neste capítulo serão apresentadas as simulações e experimentos realizados em malha aberta para cada modelo de atrito estudado no capítulo anterior (modelo de Lugre e modelo de Gomes). Ressalta-se que todas as simulações realizadas bem como todas as etapas deste projeto, foi utilizado o *Matrix Laboratory* (Matlab) e Simulink, que é um software de programação e computação numérica desenvolvido pela MathWorks. Segundo a MathWorks (2020), o Matlab possui mais de quatro milhões de usuários por todo o mundo, que utilizam o produto para as mais diversas aplicações como *Deep Learning*, robótica e sistemas de controle.

A intenção da comparação gráfica entre simulações e experimentos é a validação dos modelos de atrito propostos, bem como verificar a eficiência de ambos para a modelagem de atrito do atuador robótico empregado.

Como importante critério a ser seguido, todas as simulações foram realizadas sob as mesmas condições do experimento, o que significa que as mesmas entradas em tensão experimental aplicadas foram reproduzidas como entradas nas simulações.

Por meio do microcontrolador Arduino, foi aplicado um sinal senoidal PWM (*Pulse-Width-Modulation*) ao motor do atuador robótico. O sinal PWM é então convertido em um sinal de tensão V_a , que gera um respectivo sinal de corrente i_a , produzindo um torque motor no atuador T_m . O torque máximo T_{max} gerado por este atuador é de $2,104Nm$.

Para cada um dos modelos de atritos estudados, foram realizadas duas simulações que utilizaram a mesma frequência de $0,1Hz$ do sinal senoidal de entrada porém com diferentes amplitudes, de 20% e 33% do torque máximo T_{max} . Essas porcentagens representam os dados experimentais obtidos para o sistema utilizado neste trabalho. Com finalidade de classificação, as simulações com 20% do T_{max} foram consideradas como simulações em baixa velocidade enquanto que as simulações com 33% do T_{max} foram consideradas como simulações em média velocidade. Simulações com altas velocidades não foram realizadas, uma vez que o efeito da zona morta é menos significativo quando as velocidades do motor são maiores.

5.1 Simulações e Experimentos com o Modelo de Atrito de Lugre

Para realização das simulações do modelo de atrito de Lugre, é preciso identificar os parâmetros necessários para reproduzir o modelo. Alguns destes parâmetros foram identificados a partir da análise da curva estática do motor do atuador, Figura 25, como por exemplo o atrito seco f_c , o atrito viscoso f_v e a velocidade stribeck V_s , enquanto que outros parâmetros foram considerados através do método de tentativa e erro, como por exemplo o coeficiente de rigidez τ_0 , o termo de amortecimento τ_1 e a curva δ . Estes parâmetros estão representados na Tabela 10.

Tabela 10: Parâmetros utilizados para a simulação do modelo de atrito de Lugre

Parâmetro	Descrição	Valor
f_c	Atrito Seco	0,1415 Nm
f_s	Atrito Estático	0,1316 Nm
V_s	Velocidade Stribeck	0,08 rad/s
τ_0	Coeficiente de Rigidez	3000
τ_1	Termo de Amortecimento	0,01
$\tau_2 = f_v$	Coeficiente de Atrito Viscoso	0,4225 Nm.s/rad
-	Taxa de atrito	0,93
δ	Curva	-2

Fonte: Próprio Autor

Ressalta-se que $f_c = (f_{cp} - f_{cn})/2$, que $f_s = (\text{taxa de atrito}) \cdot f_c$, e $f_v = (f_{vp} + f_{vn})/2$.

Os sistemas do atuador robótico e modelo de atrito de Lugre foram simulados no Simulink e podem ser visualizado nas Figuras 27 e 28. Vê-se na Figura 27 a representação do atuador robótico (blocos em azul), além do modelo de atrito de Lugre (bloco branco). As equações que compõem o bloco do modelo de atrito de Lugre são representadas na Figura 28, sendo que as diferentes cores dos blocos representam diferentes equações. Os blocos em verde claro representam a equação (33), em amarelo a equação (34) e em verde escuro representam a equação (35). A resposta de saída do modelo de Lugre será o torque de atrito não linear calculado por este modelo F_a , que retorna ao sistema afetando o torque de entrada no sistema mecânico.

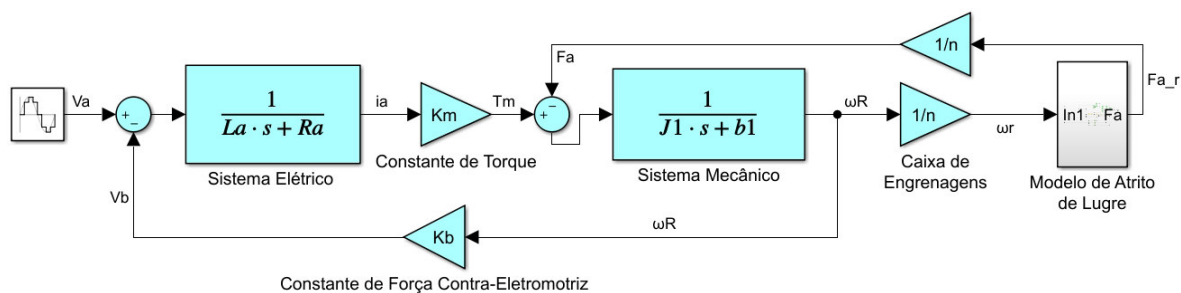


Figura 27: Representação do atuador robótico com o modelo de atrito de Lugre

Fonte: Próprio Autor

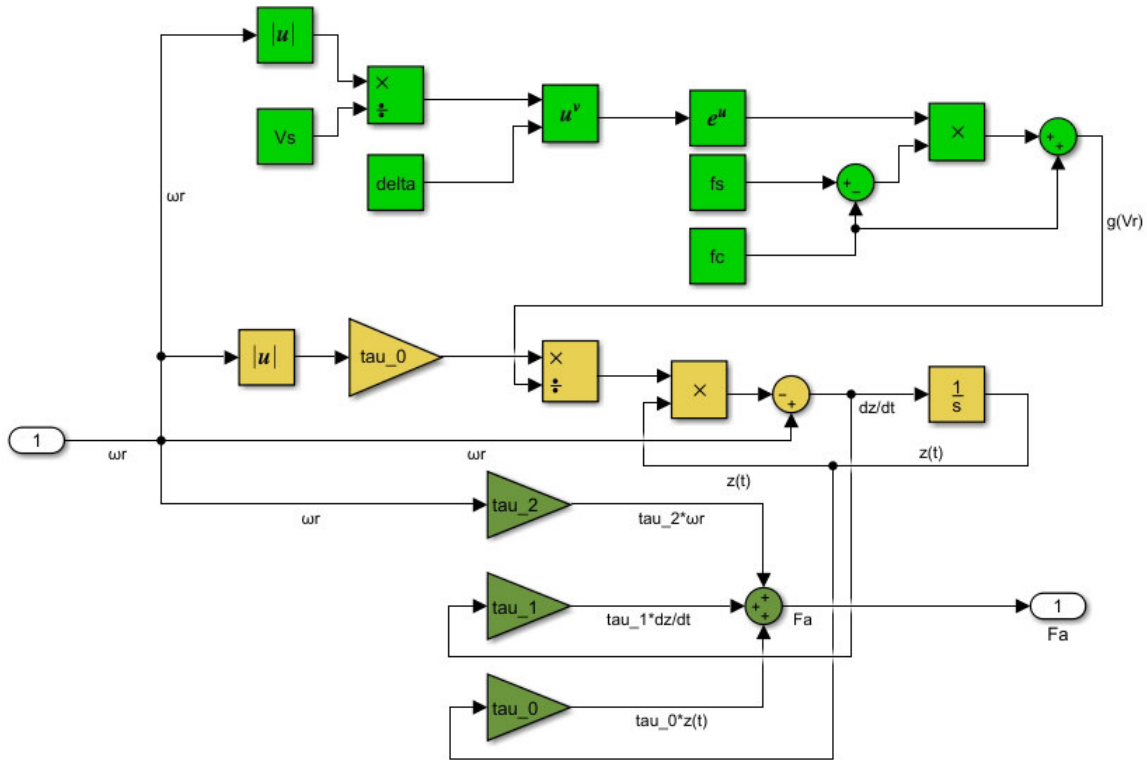


Figura 28: Representação no simulink do modelo de atrito de Lugre

Fonte: Próprio Autor

5.1.1 Simulações em Baixa Velocidade - 20% T_{max}

Na primeira simulação aplicou-se um sinal senoidal com uma frequência de $0,1\text{Hz}$ e amplitude de aproximadamente 20% do torque máximo do atuador, que é de $2,104\text{Nm}$. As Figuras 29, 30, 31 e 32 apresentam o resultado desta simulação com o modelo de atrito de Lugre.

As Figuras 29 e 30 demonstram que a velocidade simulada em malha aberta com o modelo de Lugre representa de forma muito eficiente a modelagem de atrito do atuador robótico, já que as curvas estão bem próximas das curvas resultantes dos experimentos em malha aberta, ainda que nos picos positivos de velocidade exista uma pequena diferença, sendo a velocidade simulada menor que a experimental.

Enquanto que nas Figuras 31 e 32 o foco é demonstrar a simulação em relação a inversão da velocidade - quando esta passa por zero. A melhor estimação de parâmetros, como os realizados por tentativa e erro, poderiam melhorar ainda mais a representação da curva próxima a inversão de velocidade.

5.1.2 Simulações em Média Velocidade - 33% T_{max}

Na segunda simulação aplicou-se um sinal senoidal com uma frequência de $0,1\text{Hz}$ e amplitude de aproximadamente 33% do torque máximo do atuador, que é de $2,104\text{Nm}$. Destaca-se que o sistema apresentou pequenas interferências mecânicas do motor CC nos

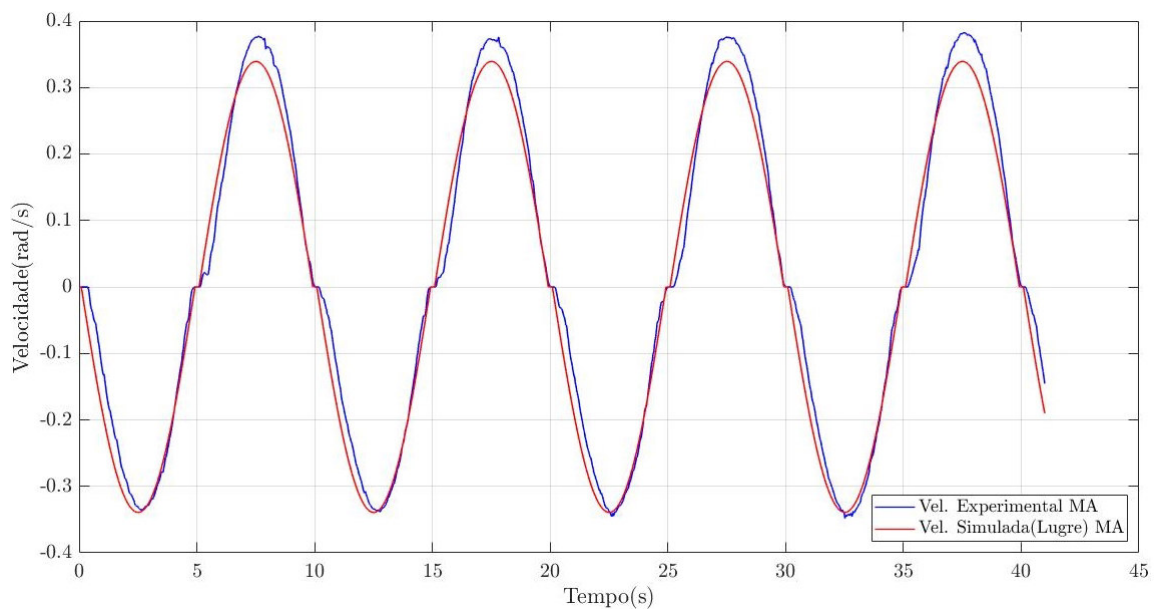


Figura 29: Simulação com modelo Lugre. Entrada de 20% do torque máximo. Velocidade Simulada em MA (linha vermelha) x Velocidade Experimental em MA (linha azul)

Fonte: Próprio Autor

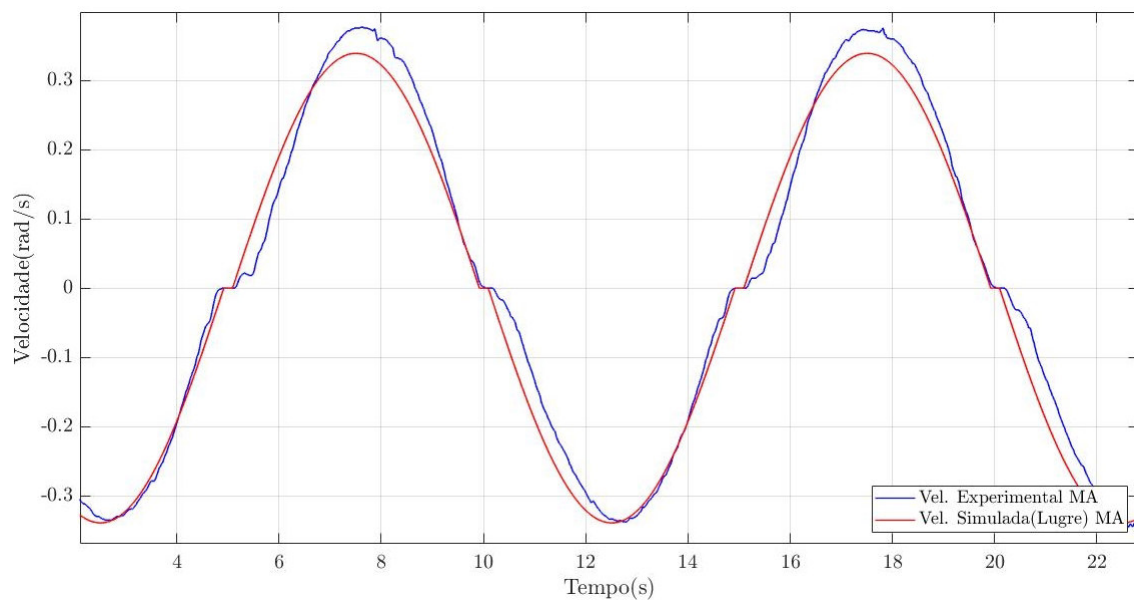


Figura 30: Simulação com modelo Lugre ampliado. Entrada de 20% do torque máximo. Velocidade Simulada em MA (linha vermelha) x Velocidade Experimental em MA (linha azul)

Fonte: Próprio Autor

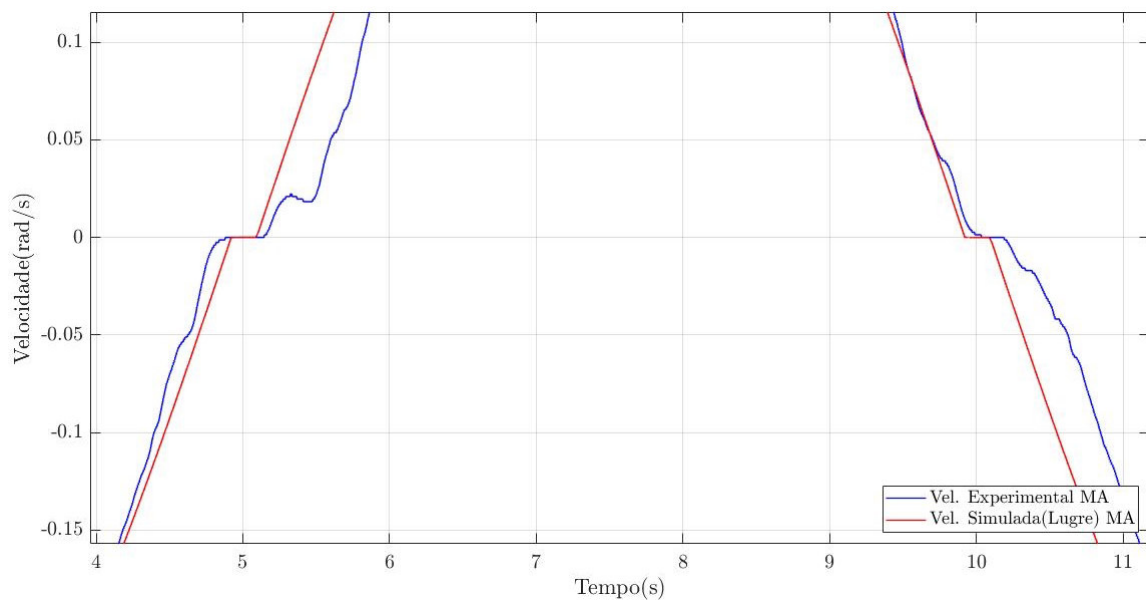


Figura 31: Simulação com modelo Lugre com foco as inversões de velocidade. Entrada de 20% do torque máximo. Velocidade Simulada em MA (linha vermelha) x Velocidade Experimental em MA (linha azul)

Fonte: Próprio Autor

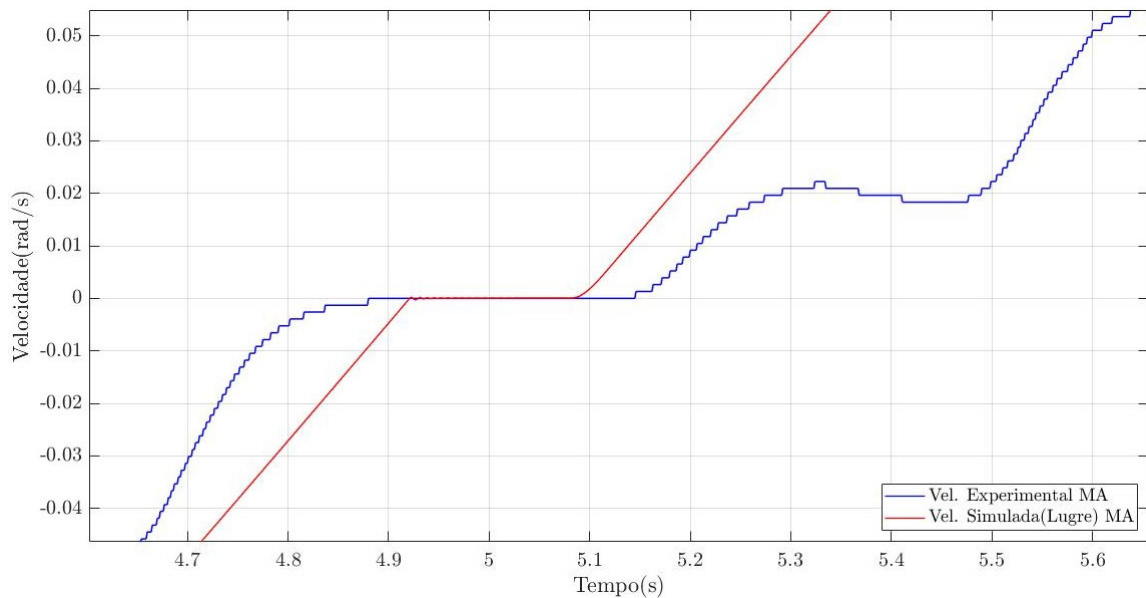


Figura 32: Simulação com modelo Lugre com ampliação nas inversões de velocidade. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha vermelha) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

dados representados próximos de 5, 10, 15 e 20 segundos, na velocidade experimental. Esses picos negativos foram observados tanto pelo sensor de velocidade quanto pelo sensor de corrente. As Figuras 33 e 34 apresentam o resultado desta simulação.

Os resultados simulados apresentados na Figura 33 são similares aos resultados em baixa velocidade (Figuras 29 e 30). Nesta simulação, não são verificados erros significativos nos picos positivos de velocidade, porém os erros são mais perceptíveis nos picos negativos se comparados a simulação anterior.

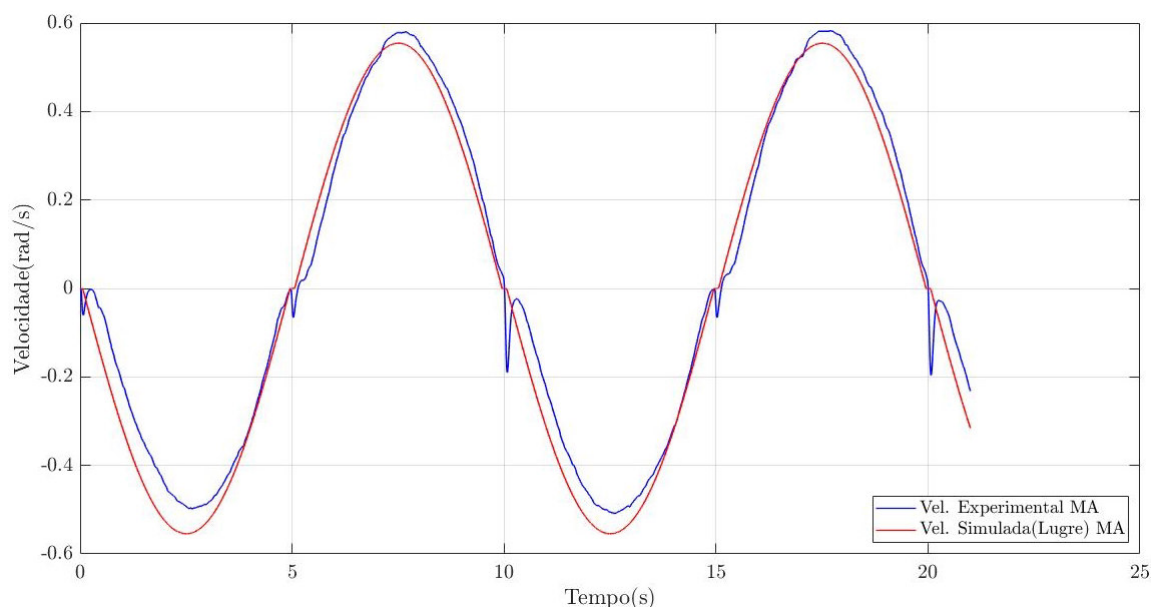


Figura 33: Simulação com modelo Lugre. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha vermelha) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

Na Figura 34, focada na inversão de velocidade, os resultados simulados representam de forma eficiente os experimentos, mesmo sendo observado que nestes experimentos picos estranhos próximos da velocidade zero foram registrados por erro de leitura do microcontrolador Arduino.

O modelo de atrito de Lugre considera diversos efeitos do atrito, como por exemplo o efeito *Stribeck* e o fenômeno de pré-deslizamento e contato entre hastes elásticas e por isso garante resultados simulados próximos aos resultados experimentais, se tornando um modelo muito empregado e que fornece ótima contribuição em estudos envolvendo a modelagem de atrito. No entanto, o modelo desconsidera a aplicação do torque motor nos dois sentidos de rotação (GOMES, 2011). Por esta razão, na simulação observa-se que, apesar de um melhor desempenho no sentido negativo, não há garantia de bom desempenho no sentido positivo da velocidade, e vice-versa.

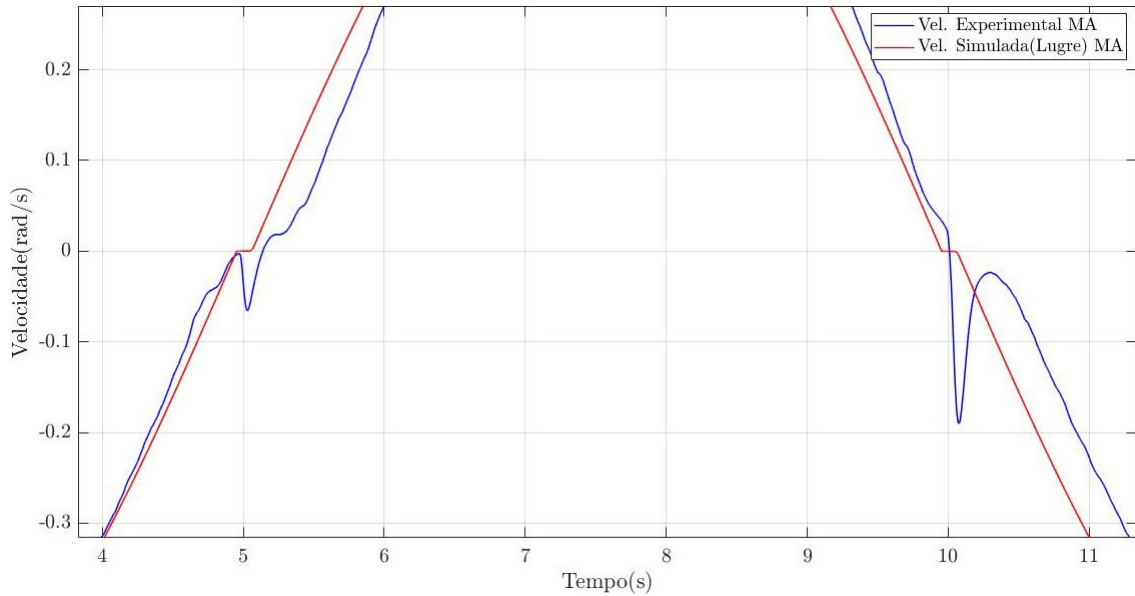


Figura 34: Simulação com modelo Luge com foco as inversões de velocidade. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha vermelha) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

5.2 Simulações e Experimentos com o Modelo de Atrito de Gomes

Novamente se torna necessária a identificação de parâmetros do modelo de atrito de Gomes para realizar simulações com o mesmo. Diferentemente do que ocorre com o modelo de Luge, todos os parâmetros do modelo de atrito de Gomes são de fácil identificação, necessitando apenas da análise da curva estática do atrito do motor, representados nas Figuras 23, 24 e 25. Neste modelo, não existe a necessidade de definir parâmetros através do método de tentativa e erro. Além disso, o modelo de Gomes considera o atrito nos dois sentidos de rotação, o sentido horário (positivo) e o sentido anti-horário (negativo). Assim, os parâmetros utilizados nas simulações estão representados na Tabela 11.

Os sistemas do atuador robótico e modelo de atrito de Gomes foram simulados no Matlab através da representação do Algoritmo 1 e da forma matricial do atuador robótico - equação (40) - em linhas de código. A representação gráfica deste sistema pode ser visualizada na Figura 35 na qual o atuador robótico e o algoritmo utilizado no cálculo do coeficiente de atrito viscoso variável f_v^* , do modelo de atrito de Gomes, foram representados pelo bloco branco.

A entrada utilizada na simulações do modelo de atrito de Gomes é o torque motor T_m , obtido através da corrente de armadura experimental i_a multiplicada pela constante de torque K_m . Além disso utiliza-se a velocidade rotor simulada ω_{rs} obtida na saída do modelo da equação (40). É importante destacar que um requisito para a utilização do modelo de atrito de Gomes é que a entrada de torque motor T_m deve ser obtida experimen-

Tabela 11: Parâmetros utilizados para a simulação do modelo de atrito de Gomes

Parâmetro	Descrição	Valor
f_{cp}	Atrito Seco Positivo	$0,1505 \text{ Nm}$
f_{cn}	Atrito Seco Negativo	$-0,1326 \text{ Nm}$
f_{vp}	Atrito Viscoso Positivo	$0,3694 \text{ Nm.s/rad}$
f_{vn}	Atrito Viscoso Negativo	$0,4756 \text{ Nm.s/rad}$
c_p	Coefficiente de Atrito Positivo	$0 \text{ Nm.s}^2/\text{rad}^2$
c_n	Coefficiente de Atrito Negativo	$0 \text{ Nm.s}^2/\text{rad}^2$
-	Taxa de Atrito	$0,93$
$\dot{\theta}_{lim}$	Vel. Limite	$0,08 \text{ rad/s}$
$\dot{\theta}_{st}$	Menor Vel. Fisicamente	$0,004 \text{ rad/s}$
γ	Gamma	$0,05$

Fonte: Próprio Autor

talmente, para a mesma tensão de acionamento V_a . Nesse caso, conforme supracitado, T_m é obtido através da medição da corrente de armadura experimental i_a e do resultado da equação (5).

A resposta de saída do modelo de Gomes será o coeficiente de atrito viscoso variável f_v^* que, multiplicado pela velocidade rotor, retorna ao sistema em forma de torque de atrito não linear T_{at} , afetando o torque de entrada no sistema mecânico.

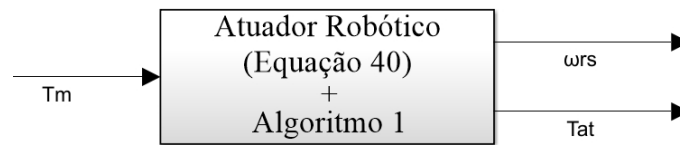


Figura 35: Representação do atuador robótico com o modelo de atrito de Gomes

Fonte: Próprio Autor

5.2.1 Simulações em Baixa Velocidade - 20% T_{max}

O Algoritmo 1 e o modelo apresentado pela equação (40) foram simulados, com os parâmetros da Tabela 9, seguindo o seguinte procedimento: na primeira simulação aplicou-se um sinal senoidal com uma frequência de $0,1 \text{ Hz}$ e amplitude de aproximadamente 20% do torque máximo do atuador, que é de $2,104 \text{ Nm}$. Nas Figuras 36, 37, 38 e 39 são apresentados os resultados da simulação com o modelo de atrito de Gomes.

As Figuras 36 e 37 demonstram a eficiência do modelo de Gomes na simulação da velocidade em comparação a velocidade adquirida experimentalmente. É possível notar que a velocidade simulada em malha aberta por meio do modelo de atrito de Gomes (linha preta) é muito próxima da velocidade experimental em malha aberta (linha azul) tanto no sentido positivo quanto no negativo da velocidade.

Já nas Figuras 38 e 39, uma ampliação próxima de zero, ou seja, nas inversões de

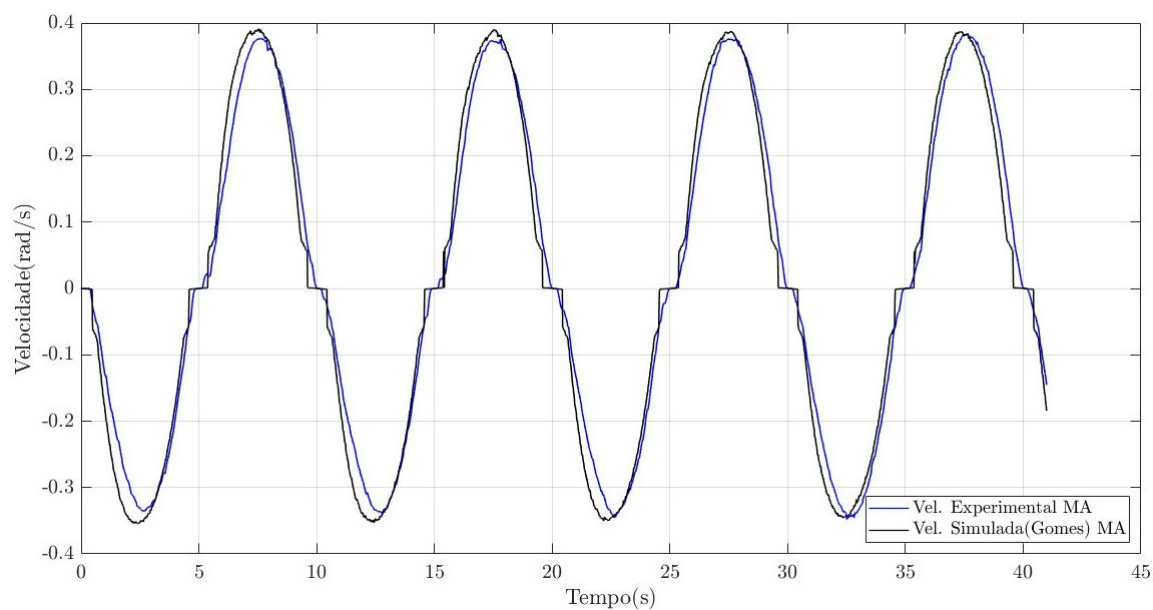


Figura 36: Simulação com modelo Gomes. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

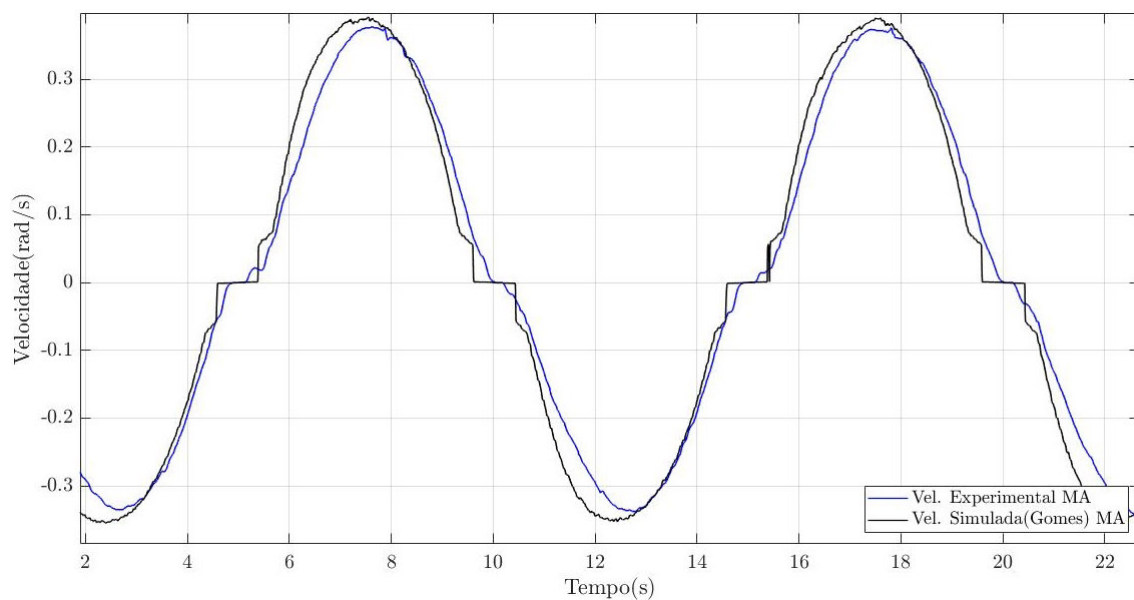


Figura 37: Simulação com modelo Gomes ampliado. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

velocidade, é demonstrada para análise do efeito *stick-slip*. É possível notar a percepção do modelo a inversão de velocidade e a forma com que esta foi reproduzida na simulação.

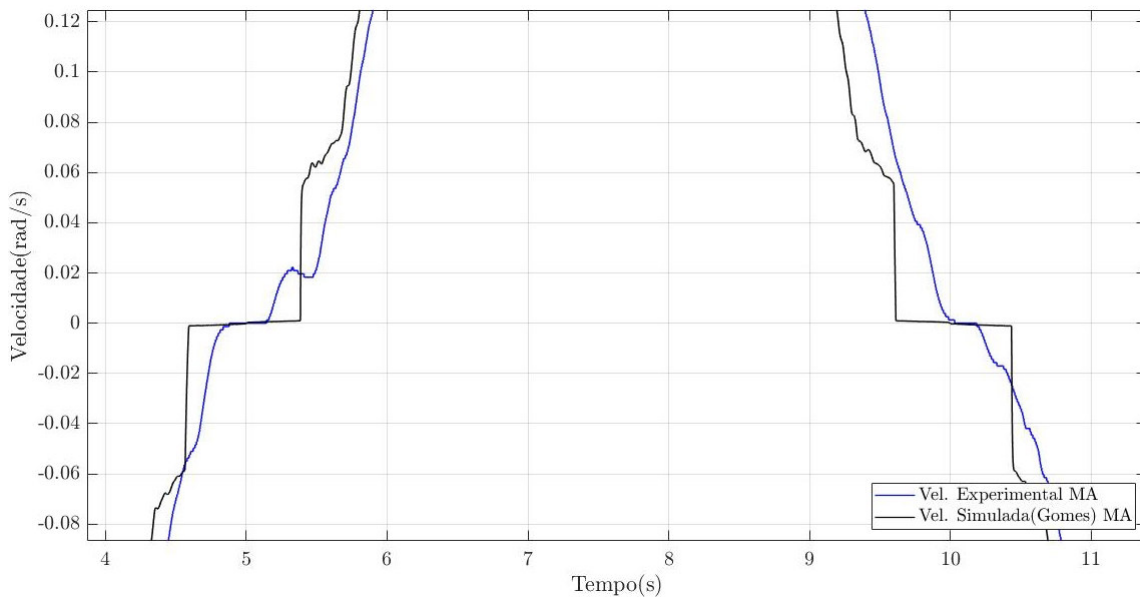


Figura 38: Simulação com modelo Gomes com foco as inversões de velocidade. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

5.2.2 Simulações em Média Velocidade - 33% T_{max}

Na segunda simulação aplicou-se um sinal senoidal com uma frequência de $0,1Hz$ e amplitude de aproximadamente 33% do torque máximo do atuador, que é de $2,104Nm$. As Figuras 40 e 41 apresentam o resultado desta simulação. Destaca-se que o sistema apresentou pequenas interferências mecânicas do motor CC nos dados representados próximos de 5, 10, 15 e 20 segundos, na velocidade experimental. Esses picos negativos foram observados tanto pelo sensor de velocidade quanto pelo sensor de corrente.

Os resultados simulados e apresentados na Figura 40 são coerentes quando comparados a velocidade experimental.

Na Figura 41, focada na inversão de velocidade, os resultados simulados são eficientes na comparação com os resultados experimentais. Com 33% do torque máximo, observou-se um comportamento estranho (picos negativos na corrente) na passagem pelo zero. Estes são os erros de leitura anteriormente mencionados. Como o modelo de atrito de Gomes recebe diretamente esse sinal de corrente i_a para gerar o torque motor T_m (multiplicando pela constante de torque K_m), a simulação reproduziu esse comportamento estranho.

Ambos modelos representaram de forma coerente o atrito no atuador robótico utilizado no estudo, gerando bons resultados em comparação aos experimentos.

Em baixas velocidades, no sentido negativo de rotação, ambos os modelos obtiveram

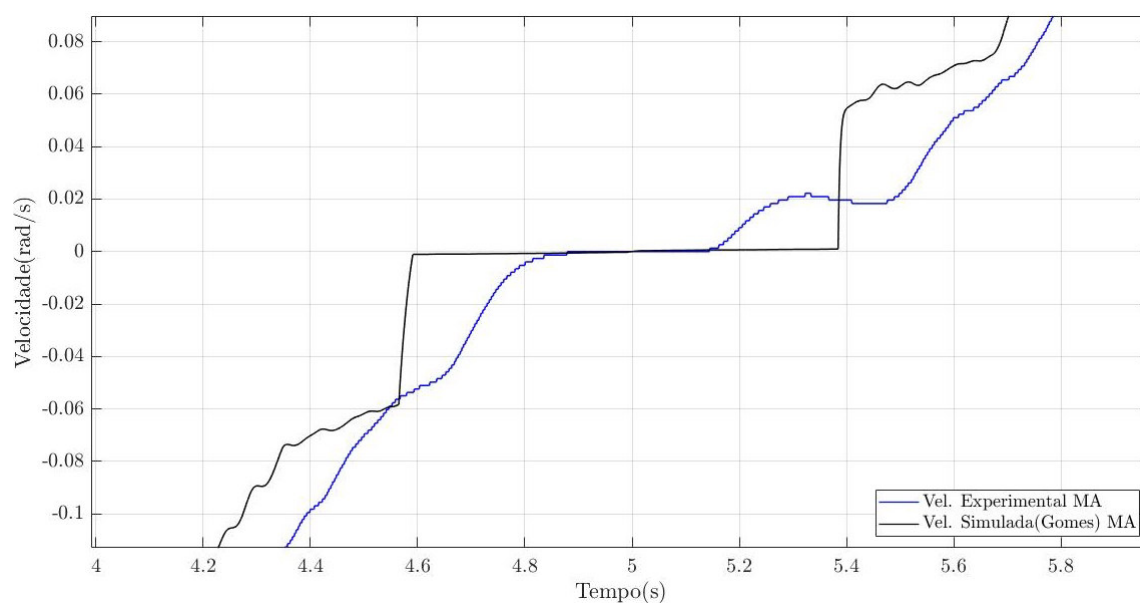


Figura 39: Simulação com modelo Gomes com ampliação nas inversões de velocidade. Entrada de 20% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

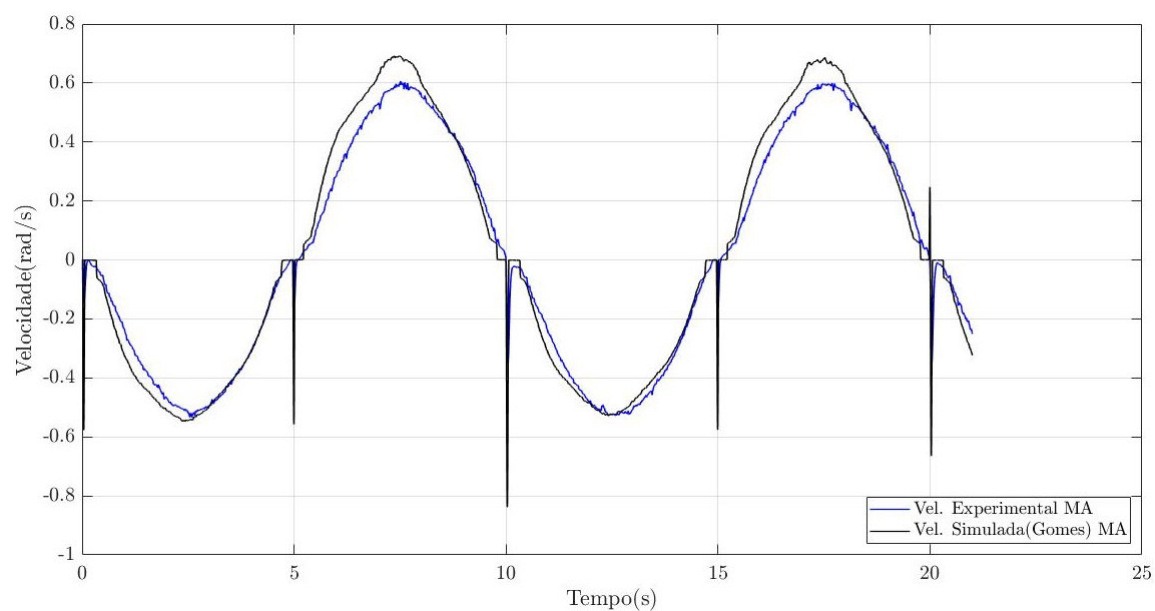


Figura 40: Simulação com modelo Gomes. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

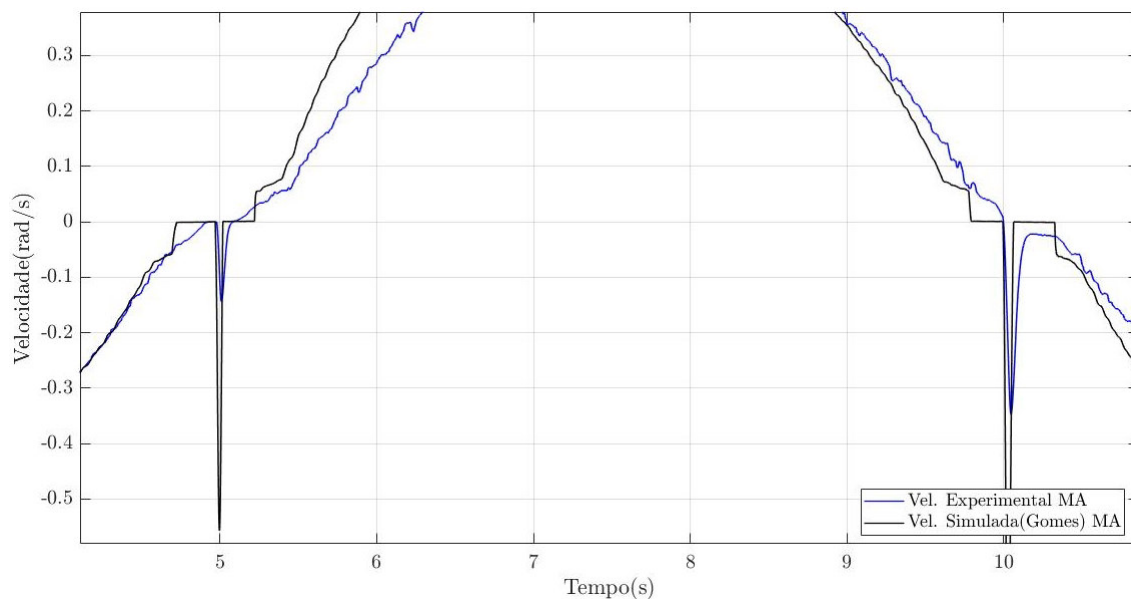


Figura 41: Simulação com modelo Gomes com foco as inversões de velocidade. Entrada de 33% do torque máximo. Velocidade simulada em MA (linha preta) x Velocidade experimental em MA (linha azul)

Fonte: Próprio Autor

resultados semelhantes, porém no sentido positivo, o modelo de Lugre se comporta com uma velocidade menor em comparação ao pico da velocidade experimental, enquanto que o modelo de Gomes se comporta com velocidade semelhante ao pico de velocidade experimental. Já em médias velocidades, foi observado um maior erro na representação da velocidade negativa no modelo de Lugre, erro que não foi observado no modelo Gomes. Em se tratando de velocidades positivas, os modelos apresentaram resultados favoráveis. Finalmente, quando consideradas as inversões de velocidade, o modelo de Gomes realiza uma representação mais eficaz do comportamento experimental do sistema, ainda que o modelo de Lugre também modele de forma coerente.

O modelo de atrito de Gomes representa grande parte das características que compõem o atrito, com destaque a representação do efeito *stick-slip* e da consideração de diferentes parâmetros do atrito em diferentes sentidos de rotação. Além disso, é de fácil identificação paramétrica - fator que facilita sua implementação.

A partir destas características, somadas a boa representação do atrito em malha aberta em comparação aos experimentos realizados, o modelo de Gomes foi escolhido para os próximos passos desta dissertação.

6 REDES NEURAIS ARTIFICIAIS

O cérebro humano é um computador (sistema de processamento de informação) altamente complexo, não linear e paralelo, que tem capacidade de organizar seus constituintes estruturais (neurônios) de forma a realizar certos processamentos muito mais rapidamente que o mais rápido computador digital hoje existente (HAYKIN, 2001).

O funcionamento do cérebro humano inspirou o surgimento das redes neurais artificiais (RNA), que são sistemas paralelos distribuídos compostos por unidades de processamento simples, também chamadas de nodos ou neurônios, dispostos em uma ou mais camadas e interligados por inúmeras conexões, que realizam cálculos de determinadas funções matemáticas (BRAGA; FERREIRA; LUDERMIR, 2007). A rede extrai seu poder computacional da sua distribuição de estrutura maciçamente paralela e da sua habilidade de aprender/generalizar, tornando possível a resolução de problemas complexos nas mais diferentes áreas do conhecimento (HAYKIN, 2001).

Assim, as RNAs, também chamadas de neurocomputadores, processadores paralelamente distribuídos e redes conexionistas, surgiram como uma alternativa à abordagem simbólica, que se baseava na lógica (KRATZER, 1991). A solução de problemas através de RNAs é bastante atrativa, já que a forma como estes são representados internamente pela rede e o paralelismo natural inerente à arquitetura das RNAs cria a possibilidade de um desempenho superior ao dos modelos convencionais (BRAGA; FERREIRA; LUDERMIR, 2007).

6.1 Neurônio Artificial

O neurônio é uma unidade fundamental de processamento de informação de uma RNA (WU; MCLARTY, 2012). O neurônio artificial, nada mais é que uma representação aproximada de um neurônio biológico. A Figura 42 abaixo, retirada de Grassi (2004), ilustra um modelo representativo de um neurônio artificial proposto por McCulloch e Pitts.

A representação matemática dos neurônios artificiais resulta em um modelo com terminais de entrada ($x_1, x_2, \dots, x_D$), equivalentes aos dendritos dos neurônios biológicos, e terminais de saída (y), equivalentes aos axônios. Adicionalmente, assim como os

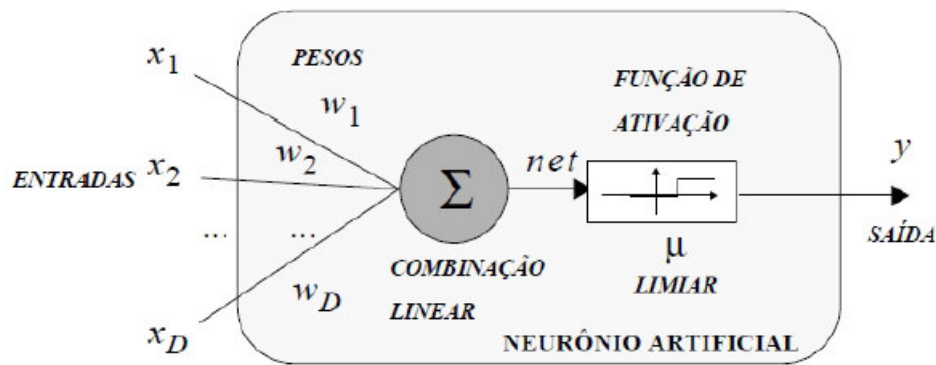


Figura 42: Representação de um neurônio artificial
Fonte: (GRASSI, 2004)

neurônios reais, possuem uma forma de processamento interno que gera um sinal de saída em função do sinal de entrada.

Entretanto, enquanto a saída de um neurônio biológico está em constante alteração no tempo, a de um neurônio artificial muda somente em intervalos discretos no tempo, isto é, quando os dados de entrada mudam (BATTISTELLA, 2004).

Para o processamento interno dos sinais de entrada, são identificados outros três elementos básicos de uma rede neural: os pesos sinápticos, o somador e a função de ativação.

- Os pesos sinápticos ou elos de conexão ($W_1, W_2, \dots, W_D$) são pesos que serão atribuídos a entrada associada. Eles emulam o comportamento das sinapses no neurônio biológico e seus valores podem ser positivos ou negativos.
- O somador (Σ) tem a função de somar os sinais de entrada, que foram ponderadas pelas sinapses do neurônio. Estas operações constituem um combinador linear.
- A função de ativação (μ), também conhecida como função restritiva, restringe a amplitude e o intervalo permissível da amplitude do sinal de saída. Normalmente os valores da amplitude de saída de um neurônio se compreendem em intervalos como $[0,1]$ e $[-1,1]$.

As entradas do neurônio ($x_1, x_2, \dots, x_D$), que podem ser estímulos do sistema ou saídas de outros neurônios, são multiplicadas pelos pesos sinápticos correspondentes ($W_1, W_2, \dots, W_D$) gerando entradas ponderadas. Estas entradas ponderadas serão somadas e o valor de saída da combinação linear (net) será comparada a um valor limite para ativação do neurônio. Caso a ativação aconteça, após a restrição da função de ativação (μ), obtém-se o valor da saída (y).

6.1.1 Função de ativação

Em uma RNA, cada neurônio tem associado uma função de ativação, responsável pela intensidade do sinal a ser transmitido pelas conexões para os neurônios das camadas

adjacentes (VALENÇA; LUDERMIR, 1999).

Os gráficos ilustrados na Figura 43, retirados de Beale, Hagan e Demuth (2012), são exemplos de funções de ativação. Dentre as funções de ativação, as funções não lineares são as mais aplicadas, como por exemplo a função degrau ou limiar (*hardlim*), a função linear (*purelin*), a tangente hiperbólica sigmoidal (*tansig*) e a função sigmóide (*logsig*).

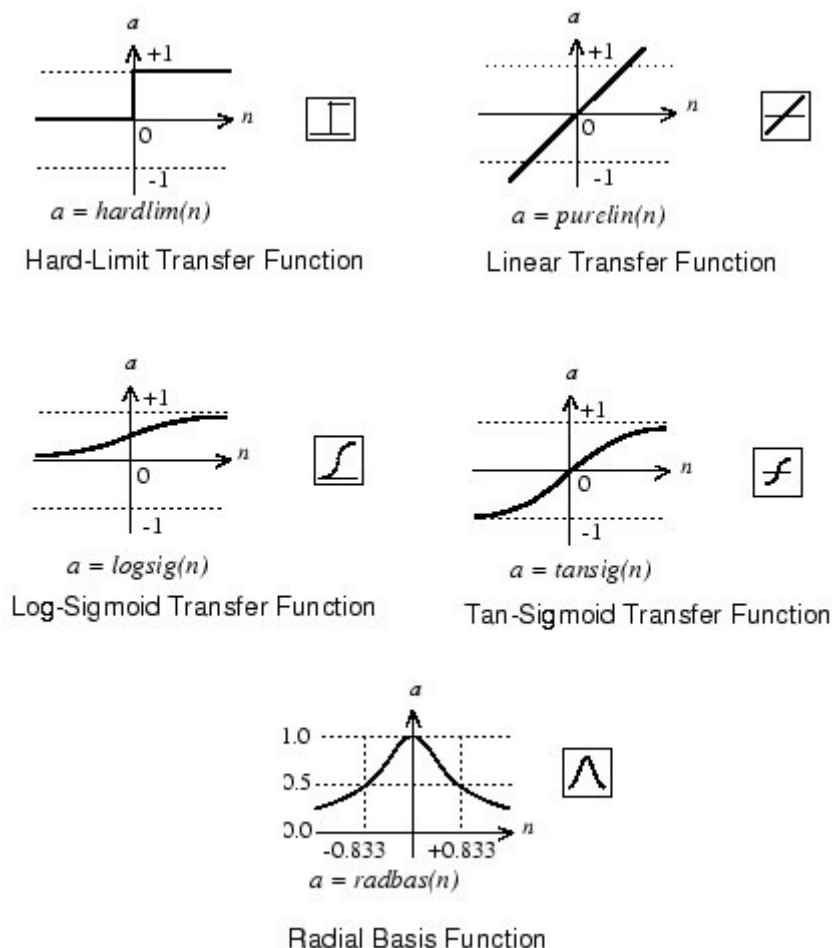


Figura 43: Funções de ativação
Fonte: (BEALE; HAGAN; DEMUTH, 2012)

A função degrau gera resposta 1 se o valor de entrada na função de ativação da rede for maior ou igual a 0 (valores não negativos). Caso contrário, gera resposta igual a 0. De acordo com Beale, Hagan e Demuth (2012), a função degrau cria na rede a habilidade de classificar vetores de entradas pela divisão do espaço amostral em duas regiões.

A função linear não limitará a saída da rede, sendo utilizada para armazenamento de entrada e saída de dados. Esta forma de função de ativação pode ser vista como uma aproximação de um amplificador não-linear (HAYKIN, 2001).

Segundo Braga, Ferreira e Ludermir (2007) a função sigmoide é a função de ativação mais utilizada e são associadas ao uso de perceptron multicamadas. Esta função pode ser de dois tipos: a sigmoidal logística (*logsig*) e a tangente hiperbólica sigmoidal (*tansig*).

O que as diferencia são seus intervalos de saída. Enquanto nas funções *logsig* o intervalo de saída é de 0 a 1, nas funções *tansig* este intervalo se compreende entre -1 e 1.

6.2 Arquitetura das RNAs

A arquitetura de uma rede neural, também denominada topologia da rede, refere-se basicamente a forma com que os neurônios estão dispostos e organizados dentro de uma rede.

A definição da arquitetura de uma RNA é um parâmetro importante na sua concepção, uma vez que ela restringe o tipo de problema que pode ser tratado pela rede (BRAGA; FERREIRA; LUDERMIR, 2007). Para definir a arquitetura, alguns parâmetros são levados em consideração. São eles: o número de camadas da rede, o número de nodos em cada camada, o tipo de conexão entre nodos e a topologia da rede.

Assim, as redes neurais são divididas em três diferentes classes de arquitetura: As redes alimentadas adiante com camada única (*feedforward*), as redes alimentadas adiante de múltiplas camadas (*multilayer feedforward networks*) e as redes recorrentes (*feedback*).

6.2.1 Redes alimentadas adiante com camada única (*Feedforward*)

Este tipo de rede é a forma mais simples de redes em camadas uma vez que possui apenas uma camada de neurônios (nós computacionais), a camada de saída. A camada de entrada não é considerada já que ela não realiza qualquer tipo de computação. Os valores correspondentes as entradas são projetados sobre a camada de saída, mas não vice-versa, por isso a rede se configura alimentada adiante ou acíclica. A Figura 44 representa uma rede alimentada adiante com camada única.

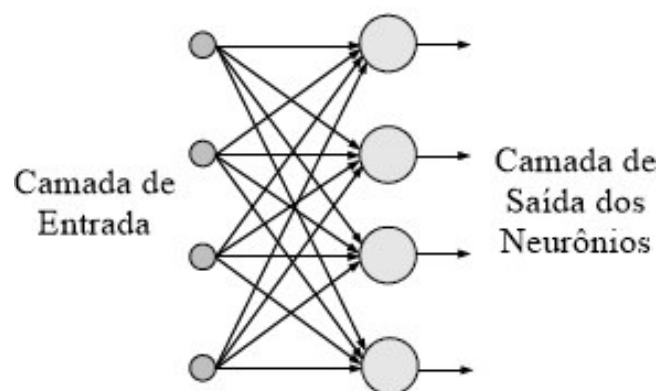


Figura 44: Exemplo de rede alimentada adiante com camadas única
Fonte: Adaptado de (HAYKIN, 2001)

6.2.2 Redes alimentadas adiante de múltiplas camadas (*Multilayer Feedforward Networks*)

Este tipo de rede, também chamadas de MLP (*Multilayer Perceptron*) se distingue das redes *feedforward* de camada única pelo fato de possuírem camadas ocultas entre as camadas de entrada e saída. Os nós computacionais destas camadas ocultas são conhecidos como neurônios ocultos. Desta forma, os neurônios ocultos (intermediários) tem a função de intervir de maneira útil entre a entrada e saída da rede a fim de tornar a rede capaz de extrair estatísticas de elevada ordem. Pode haver uma ou mais camadas intermediárias.

Os nós fonte (entrada) da rede fornecem os valores que constituem os sinais de entrada da camada oculta. Os sinais de saída de uma camada oculta se tornarão os valores de entrada de uma próxima camada oculta ou propriamente da camada de saída.

Como os valores de entrada de uma camada de neurônios serão apenas os sinais de saída da camada precedente, este tipo de rede se configura como alimentada adiante. A Figura 45 representa uma rede alimentada adiante com múltiplas camadas.

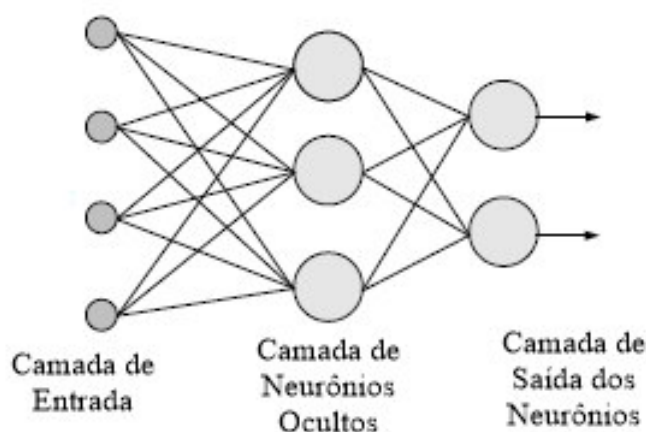


Figura 45: Exemplo de rede alimentada adiante com múltiplas camadas
Fonte: Adaptado de (HAYKIN, 2001)

As redes *feedforward* de múltiplas camadas são as redes mais difundidas na literatura.

6.2.3 Redes Recorrentes (*Feedback*)

As redes recorrentes se distinguem das redes alimentadas adiante por possuírem pelo menos um laço de realimentação. Este laço pode ser de auto-realimentação ou de um neurônio realimentando um neurônio da própria camada ou de uma camada de neurônios precedente. As redes *feedback* podem ou não possuir camadas ocultas de neurônios. Exemplo de rede recorrente é encontrado na Figura 46.

Uma rede neural pode ser completamente conectada, onde todos os neurônios de uma camada estão conectados com todos os neurônios da camada seguinte ou fracamente conectada quando não são estabelecidas todas as conexões possíveis (HAYKIN, 2001).

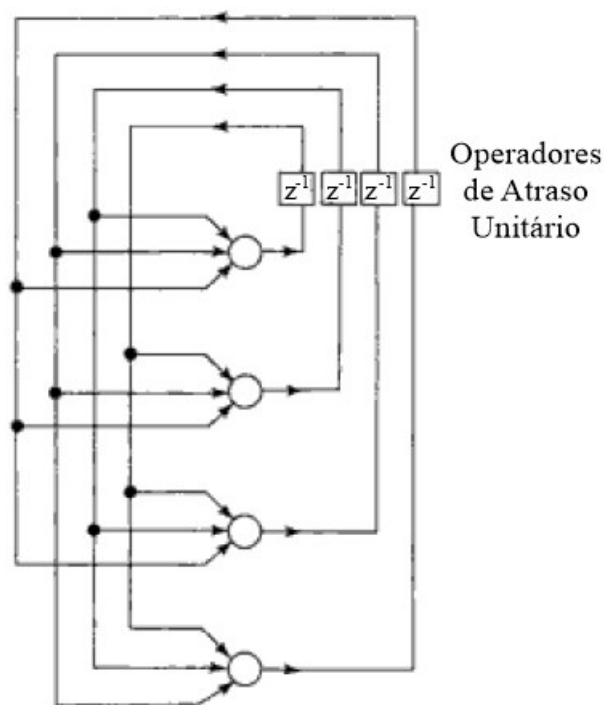


Figura 46: Exemplo de rede recorrente
 Fonte: Adaptado de (HAYKIN, 2001)

Na maioria dos estudos, a complexidade do problema não é conhecida, fazendo com que se defina uma quantidade mínima de neurônios, acrescentando ou diminuindo gradativamente até se alcançar o objetivo, que é a minimização do erro médio quadrático do conjunto de treinamento (BRAGA; FERREIRA; LUDERMIR, 2007).

6.3 Processos de Aprendizagem

A propriedade que é de importância primordial para uma rede neural é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho através de aprendizagem (HAYKIN, 2001).

A utilização de uma RNA para solucionar problemas passa por uma fase inicial de aprendizagem. De acordo com Mendel (1970), pode-se definir aprendizagem como um processo pelo qual os parâmetros de uma rede neural são ajustados através de uma forma continuada de estímulo pelo ambiente no qual a rede está operando, sendo o tipo específico de aprendizagem realizada definido pela maneira particular como ocorrem os ajustes realizados nos parâmetros.

Uma rede neural tem a capacidade de aprender sobre seu ambiente através de um processo iterativo de ajustes dos pesos sinápticos e níveis de bias, tornando-se cada vez mais instruída sobre seu ambiente conforme mais iterações do processo de aprendizagem ocorrem. A fase de aprendizado é de extrema importância, devendo ser feita de forma rigorosa e verdadeira para que se evite criar modelos inautênticos de rede.

Ao conjunto de regras pré-estabelecidas para solucionar problemas de aprendizagem dá-se o nome de algoritmo de aprendizagem. Existem diversos tipos de algoritmos de aprendizagem, que oferecem vantagens e desvantagens específicas para determinados tipos de aplicação. O que diferencia os algoritmos de aprendizagem, em geral, é a forma pela qual se ajusta um peso sináptico de um neurônio.

Dentre os diversos métodos desenvolvidos de treinamento de uma rede neural, destacam-se dois paradigmas: aprendizado supervisionado e aprendizado não-supervisionado.

6.3.1 Aprendizado Supervisionado

É o modelo de aprendizado mais comum para treinamento de RNAs. Conhecido como aprendizado supervisionado por contar com a presença de um “supervisor” (professor) que fornece os valores de entrada e saída desejadas, além de indicar bons ou maus comportamentos da rede, direcionando o processo de treinamento.

Basicamente neste tipo de aprendizado o “supervisor” fornece a rede neural um vetor de treinamento com valores de entrada e saída. A rede então terá sua saída calculada comparada a saída desejada (erro). O objetivo deste aprendizado é o de ajustar os parâmetros da rede (pesos sinápticos) a fim de minimizar os erros entre a saída desejada e a saída calculada pela rede neural. Essa iteratividade do treino deverá ser repetida com todo o conjunto de treinamento, até que a taxa de acerto esteja dentro de uma faixa considerada satisfatória. É devido a esse processo que uma RNA é capaz de aprender e generalizar (SCHMITT, 2009).

A soma dos erros quadráticos de todas as saídas é normalmente utilizada como medida de desempenho da rede e também como função de custo a ser minimizada pelo algoritmo de treinamento (BRAGA; FERREIRA; LUDERMIR, 2007). Quando a condição desejada é alcançada, torna-se dispensável a função do “supervisor”, deixando com que a rede neural lide por conta própria com o ambiente.

O aprendizado supervisionado, ilustrado na Figura 47, pode ser implementado *off-line* e *on-line*.

Quando o treinamento é *off-line*, uma vez que se obtém uma solução para a rede, os dados se mantêm fixos. Para adicionar novos dados, um novo treinamento deve ser realizado. Quanto ao treinamento *on-line*, os dados da rede mudam continuamente mantendo a rede em processo constante de adaptação.

6.3.2 Aprendizado Não-Supervisionado

No método de aprendizado não-supervisionado, ou auto-organização, não existe a figura do “supervisor” (professor) acompanhando o processo. Neste tipo de aprendizado, somente valores de entrada são fornecidos a rede. Segundo Jang, Sun e Mizutani (1997), um sistema de aprendizado não-supervisionado extrai recursos ou regularidades

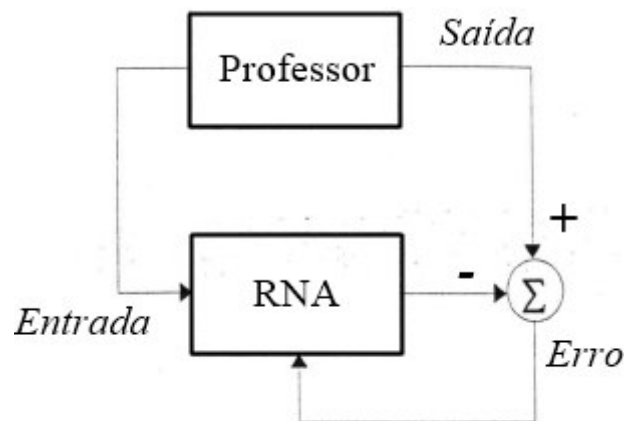


Figura 47: Aprendizado supervisionado
 Fonte: Adaptado de (BRAGA; FERREIRA; LUDERMIR, 2007)

em padrões presentes, sem receber quaisquer informações do ambiente externo.

Uma vez que a rede tenha se ajustado as regularidades estatísticas dos dados de entrada, ela desenvolve a habilidade de formar representações internas para codificar as características da entrada e, desse modo, de criar automaticamente novas classes (BECKER, 1991). A rede se torna uma espécie de classificadora, sendo capaz de identificar e categorizar os dados de entrada redundantes.

Por este motivo, como afirmado por Braga, Ferreira e Ludermir (2007), este tipo de aprendizado só se torna possível quando existe redundância nos dados de entrada, uma vez que sem redundância seria impossível encontrar padrões ou características nos dados de entrada. A Figura 48 representa o aprendizado não-supervisionado.

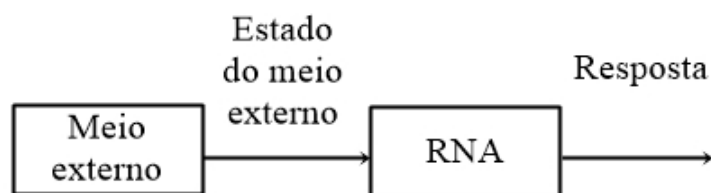


Figura 48: Aprendizado não-supervisionado
 Fonte: Adaptado de (BRAGA; FERREIRA; LUDERMIR, 2007)

6.4 Perceptron e Perceptron de Múltiplas Camadas

O perceptron representa uma das primeiras tentativas de desenvolver um sistema inteligente de auto-aprendizado utilizando componentes simples (JANG; SUN; MIZUTANI, 1997). O modelo perceptron surge em 1958, quando em seus trabalhos, Frank Rosenblatt introduz o conceito de aprendizado em RNAs. Segundo Jang, Sun e Mizutani (1997), Rosenblatt projeta o perceptron com o objetivo de explicar e modelar as habilidades de reconhecimento de padrões de sistemas visuais biológicos.

O modelo perceptron é o modelo mais simples de RNA. Ele se baseava no neurônio artificial criado por McCulloch e Pitts em 1948, com pesos ajustáveis e bias, e com a adição de uma regra de aprendizado. Sua arquitetura (Figura 49) era composta de unidades de entrada, também chamadas de retina, um nível intermediário formado por unidades de associação e uma camada de saída de onde se obtém as respostas. Embora esta topologia original possua três níveis, ela é conhecida como perceptron de uma única camada, já que somente o nível de saída (unidades de resposta) possui propriedades adaptativas (BRAGA; FERREIRA; LUDERMIR, 2007).

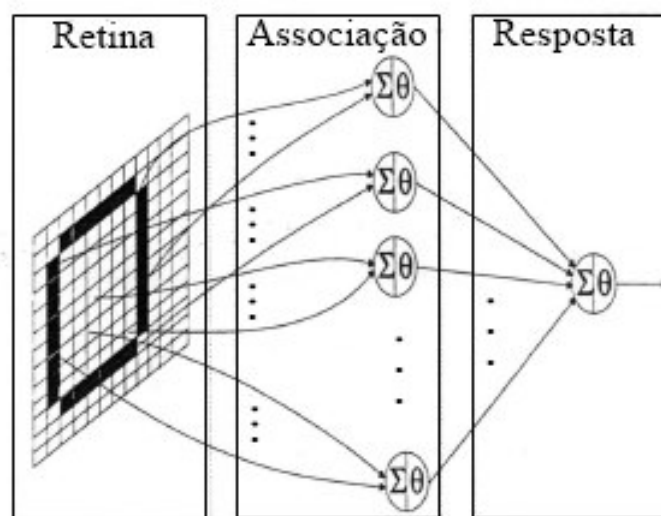


Figura 49: Modelo de um perceptron

Fonte: Adaptado de (BRAGA; FERREIRA; LUDERMIR, 2007)

Posteriormente, Rosenblatt demonstrou, através do teorema de convergência do perceptron, que qualquer neurônio baseado no modelo de McCulloch e Pitts, e treinado com o algoritmo de aprendizado do perceptron, irá sempre convergir, levando em conta que o problema seja linearmente separável.

O algoritmo de aprendizado do perceptron tem como função permitir a adaptação dos pesos. Assim, durante o processo de aprendizado, deseja-se obter um valor de incremento (ΔW) para ser aplicado aos pesos (W) a fim de gerar um valor atualizado dos pesos ($W(t + 1)$) mais próximo do valor desejado. Como mencionado por Braga, Ferreira e Ludermir (2007), os algoritmos de aprendizado em RNAs visam ao desenvolvimento de técnicas para a obtenção do valor de ΔW mais apropriado para a obtenção da solução do problema em questão.

A regra de atualização dos pesos de um nodo de um perceptron simples é descrita na equação (41):

$$W(t + 1) = W(t) + \eta ex, \quad (41)$$

Onde:

η é uma constante chamada de taxa de aprendizagem e mede a rapidez com que o vetor de pesos será atualizado;

e é o erro devido a saída atual;

x é o vetor de entrada no neurônio.

O algoritmo de treinamento do perceptron simples é descrito da seguinte forma por Braga, Ferreira e Ludermir (2007):

1. Inicializar η e o vetor de pesos W ;
2. Repete para cada neurônio;
3. Caso exista erro ($e \neq 0$), atualizar o vetor de pesos para cada um dos neurônios da rede segundo a regra descrita na Equação (41);
4. Repetir até e tender a 0 para todos os elementos do conjunto de treinamento em todos os neurônios da rede.

Apesar de toda a euforia causada pelo perceptron, este não teve tanto sucesso principalmente após os trabalhos de Minsky e Papert em 1969, que provaram a incapacidade do modelo de resolver problemas não linearmente separáveis. Esta constatação levou a um grande desinteresse pela área durante a década de 70. Os estudos com RNAs só voltaram a ganhar importância após as descrições da rede de John Hopfield em 1982 e do algoritmo *backpropagation* em 1986.

Com os trabalhos da década de 80 foi observado que, com a utilização de rede com múltiplas camadas, muitas limitações do perceptron deixam de existir. A esta extensão natural do perceptron, dá-se o nome Perceptron de Múltiplas Camadas (*Multilayer Perceptron* - MLP).

A estrutura deste tipo de modelo consiste em uma camada de entrada (nós fonte), uma ou mais camadas intermediárias, constituídas de neurônios que serão as unidades computacionais da rede e uma camada de saída, responsável pela resposta da rede. Além de contar uma ou mais camadas ocultas, o modelo de cada neurônio da rede MLP deve incluir uma função de ativação não-linear, como por exemplo a função sigmoideal (BRAGA; FERREIRA; LUDERMIR, 2007).

O perceptron multicamadas, representado na Figura 50, apresenta um poder computacional maior do que aquele apresentado pelas redes sem camadas intermediárias, podendo implementar qualquer função seja ela linearmente separável ou não (SILVA et al., 2004).

A função das redes MLP é de transformar de forma sucessiva os valores de entrada de dados, através das camadas intermediárias, para fornecer a camada de saída uma representação tratável. Neste modo, o processamento feito por cada neurônio de uma determinada camada é alcançado pela combinação de processamentos realizados pelos neurônios da camada anterior, conectados a ele. O fluxo de dados da rede progride sempre adiante, de camada em camada e quanto mais próximas da camada de saída, mais complexas são as funções implementadas.

Afirmado por Braga, Ferreira e Ludermir (2007), as unidades intermediárias de uma

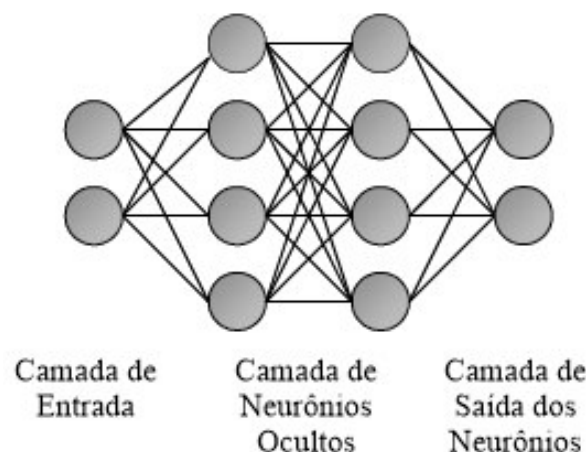


Figura 50: Modelo perceptron de múltiplas camadas
 Fonte: Adaptado de (HAYKIN, 2001)

rede MLP funcionam como detectores de características, que geram uma codificação interna dos padrões de entrada para serem utilizadas na definição da saída da rede. Dado um número suficientemente grande de unidades intermediárias, é possível formar representações internas para qualquer conjunto de padrões de entrada. Porém não é recomendado a utilização de um grande número de camadas intermediárias, uma vez que os erros propagados de uma camada intermediária a camada intermediária anterior, durante um treinamento, são na verdade estimativas de erros. Sendo assim, com grande número de camadas intermediárias, pode-se propagar por muitas vezes estimativas de estimativas de erro, o que tornará a rede menos útil e precisa.

De acordo com Cybenko (1989), uma rede com uma camada intermediária pode implementar qualquer função contínua. Redes com duas camadas intermediárias podem implementar qualquer função, seja ela linearmente separável ou não, com a sua precisão intimamente ligada ao número de neurônios utilizados nas camadas ocultas (CYBENKO, 1988).

Assim como no modelo perceptron de camada única, não existe uma fórmula para determinar quantos neurônios devem formar as camadas intermediárias do perceptron de múltiplas camadas, sendo este número normalmente definido de forma empírica.

Segundo Haykin (2001), os MLPs têm sido aplicados com sucesso para resolver diversos problemas difíceis, através de seu treinamento de forma supervisionada com um algoritmo muito popular, conhecido como algoritmo de retropropagação (*backpropagation*), que é baseado na regra de aprendizagem por correção de erro.

6.4.1 Algoritmo *Backpropagation*

O algoritmo de retropropagação dos erros, também conhecido como *backpropagation*, é o algoritmo mais usado para o treinamento de redes MLP. Desenvolvido em 1986 por Rumelhart, Hinton e Williams, o algoritmo *backpropagation* foi um dos principais

responsáveis pelo ressurgimento do interesse em RNAs.

Segundo Braga, Ferreira e Ludermir (2007), o *backpropagation* é um algoritmo supervisionado que utiliza pares (entrada, saída desejada) para, por meio de um mecanismo de correção de erros, ajustar os pesos da rede. Este algoritmo busca procurar iterativamente a mínima diferença entre as saídas desejadas e as saídas obtidas pela rede neural, com o mínimo de erro. Dessa forma, ajustando os pesos sinápticos através da retropropagação do erro encontrado em cada iteração (RUMELHART; HINTON; WILLIAMS, 1986).

Segundo Grassi (2004), o *backpropagation* é uma metodologia simples de implementar, e sua capacidade de generalização é atraente para um grande espectro de aplicações em problemas de reconhecimento e classificação de padrões. Este algoritmo possui um bom desempenho quando não se dispõe de hardware potente.

O treinamento deste algoritmo ocorre em duas fases, como ilustrado na Figura 51 retirada de Matsunaga (2012), que percorrem sentidos diferentes dentro da rede neural. Estas fases são conhecidas como fase *forward* e fase *backward*:

- Na fase *forward*, ou fase de propagação adiante, os dados seguem no sentido da camada de entrada para a camada de saída. Os valores de entrada são fornecidos a rede que calcula os respectivos valores de saída e os compara com os valores de saída desejados, gerando um erro. Nesta fase, os pesos sinápticos da rede são todos fixos.
- Na fase *backward*, ou fase de retropropagação, os dados seguem no sentido da camada de saída para a camada de entrada. Utilizando-se do erro da camada de saída gerado na fase *forward*, se atualizam os pesos das conexões através de gradiente descendente. Desta forma, nesta segunda fase, os pesos são ajustáveis.

Em suma, o treinamento se inicia com um padrão de entrada sendo fornecido, a camada de entrada da rede, que propaga esses valores a cada uma das outras camadas, até que seja gerada uma saída (y) na camada de saída. Esta saída calculada é comparada a saída desejada (y_d) gerando um erro (e). O sinal do erro é então retropropagado da camada de saída para as camadas intermediárias anteriores que receberão uma porção do sinal do erro total, proporcional a sua contribuição para a formação da saída. Com base neste sinal de erro recebido, os pesos são atualizados a fim de fazer a rede alcançar o valor de saída desejado (y_d).

O ajuste dos pesos sinápticos neste algoritmo é baseado na regra delta, formulada por Widrow e Hoff, utilizada em treinamento de redes Adaline. A generalização deste algoritmo para redes de múltiplas camadas é conhecida como regra delta generalizada ou *backpropagation* (BRAGA; FERREIRA; LUDERMIR, 2007) e consiste na aplicação do gradiente descendente (VALENÇA; LUDERMIR, 1999).

A Equação (42) representa a fórmula utilizada para calcular o erro médio quadrático (*Mean Square Error* - MSE) e o ajuste de pesos nas Equações (43), (44) e (45).

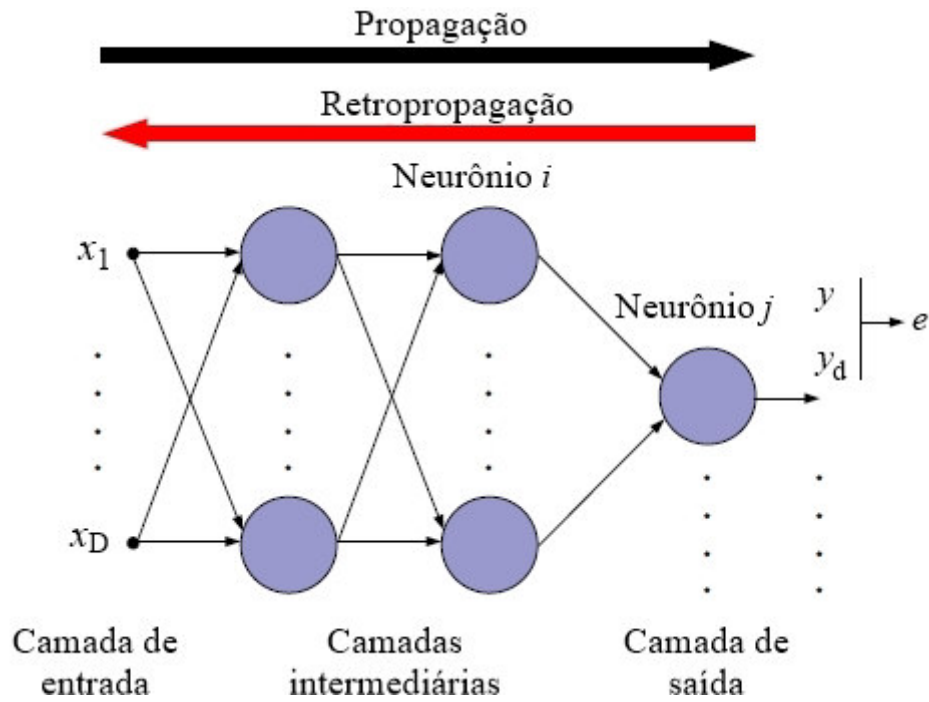


Figura 51: Algoritmo *Backpropagation*
 Fonte: Adaptado de (MATSUNAGA, 2012)

$$MSE = \frac{1}{2}(y_d - y)^2 \quad (42)$$

$$\Delta W_{ij} = \eta e_{nj} \chi_{ni} \quad (43)$$

Para neurônios na camada de saída:

$$e_{nj} = f'_j(net_{nj})(y_{dnj} - y_{nj}) \quad (44)$$

Para neurônios nas camadas ocultas:

$$e_{nj} = f'_j(net_{nj}) \sum_k e_{nk} W_{jk} \quad (45)$$

Onde:

y_d é o valor de saída desejado da RNA;

y é o valor de saída obtido;

n é o índice da n -ésima entrada da rede;

i é o índice do neurônio na camada de entrada;

j é o índice do neurônio na camada de saída;

k é o índice do neurônio nas camadas ocultas.

Para realizar os cálculos necessários para minimização do erro é importante considerar alguns parâmetros como, por exemplo, o número de iterações por padrão de entrada para

o erro alcançar um valor mínimo no treinamento, a ocorrência de um mínimo local ou global e a capacidade da rede de escapar dos mínimos locais (HAYKIN, 2001).

Mínimos locais são pontos na superfície de erro que, embora apresentem estabilidade, não são a saída ótima. A Figura 52 representa os mínimos locais.

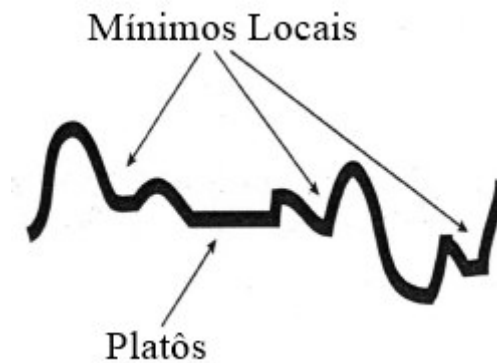


Figura 52: Mínimos locais

Fonte: Adaptado de (BRAGA; FERREIRA; LUDERMIR, 2007)

Dentre as diversas técnicas empregadas para evitar o problema dos mínimos locais além de acelerar o processo de treinamento, se destaca a adição de um termo *momentum*, representado pela Equação (46).

$$\psi = \alpha (W_{ji}(t) - W_{ji}(t - 1)) \quad (46)$$

A fórmula de ajuste de pesos com o termo *momentum* se torna:

$$W_{ji}(t + 1) = W_{ji}(t) + \eta e_j(t) \chi_i(t) + \alpha (W_{ji}(t) - W_{ji}(t - 1)) \quad (47)$$

A inclusão do termo *momentum* na fórmula de ajuste dos pesos aumenta a velocidade de aprendizado (aceleração), reduzindo o perigo de instabilidade (BRAGA; FERREIRA; LUDERMIR, 2007).

7 SIMULAÇÕES E EXPERIMENTOS COM REDES NEURAIS ARTIFICIAIS

Neste capítulo serão apresentadas considerações importantes na criação, treinamento e validação da rede neural artificial empregada no trabalho. Como em todos os capítulos anteriores, o processo de criação e treinamento da RNA foi realizada pelo software Matlab, com uso específico da *toolbox Neural Network Toolbox (nntool)*, interface gráfica que facilita o desenvolvimento de redes neurais artificiais permitindo de forma simplificada criar, treinar, exportar e importar dados de redes neurais. O processo de validação da RNA também foi realizado através do Matlab.

7.1 Criação da RNA

Para criar a rede neural artificial que será capaz de identificar o torque de atrito não linear, a fim de posteriormente compensá-lo no sistema, é necessário decidir sua arquitetura. Baseado na premissa de que redes neurais com uma camada intermediária já são capazes de solucionar problemas complexos, foi utilizado o modelo de redes alimentadas adiante de múltiplas camadas (*multilayer feedforward networks*), ou perceptron de múltiplas camadas, composto de:

- Dois neurônios na camada de entrada fornecendo a rede neural artificial os valores de torque motor T_m e velocidade do rotor ω_r ;
- Quatro neurônios na camada oculta;
- Um neurônio na camada de saída que corresponde ao valor do torque de atrito não linear T_{at} .

Esta arquitetura de rede, representada na Figura 53, foi proposta inicialmente por Gervini e Gomes (2001).

A partir do comando *nntool* na janela de comando do Matlab, abre-se uma janela *Network/Data Manager* como a da Figura 54. É nesta interface que grande parte do processo de criação, importação e exportação de variáveis ocorrerá.

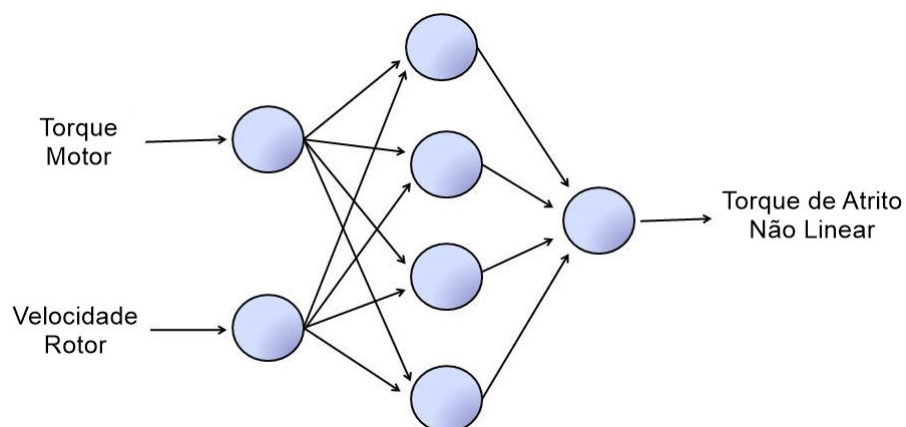


Figura 53: Arquitetura da RNA proposta
 Fonte: Adaptado de (GERVINI; GOMES, 2001)

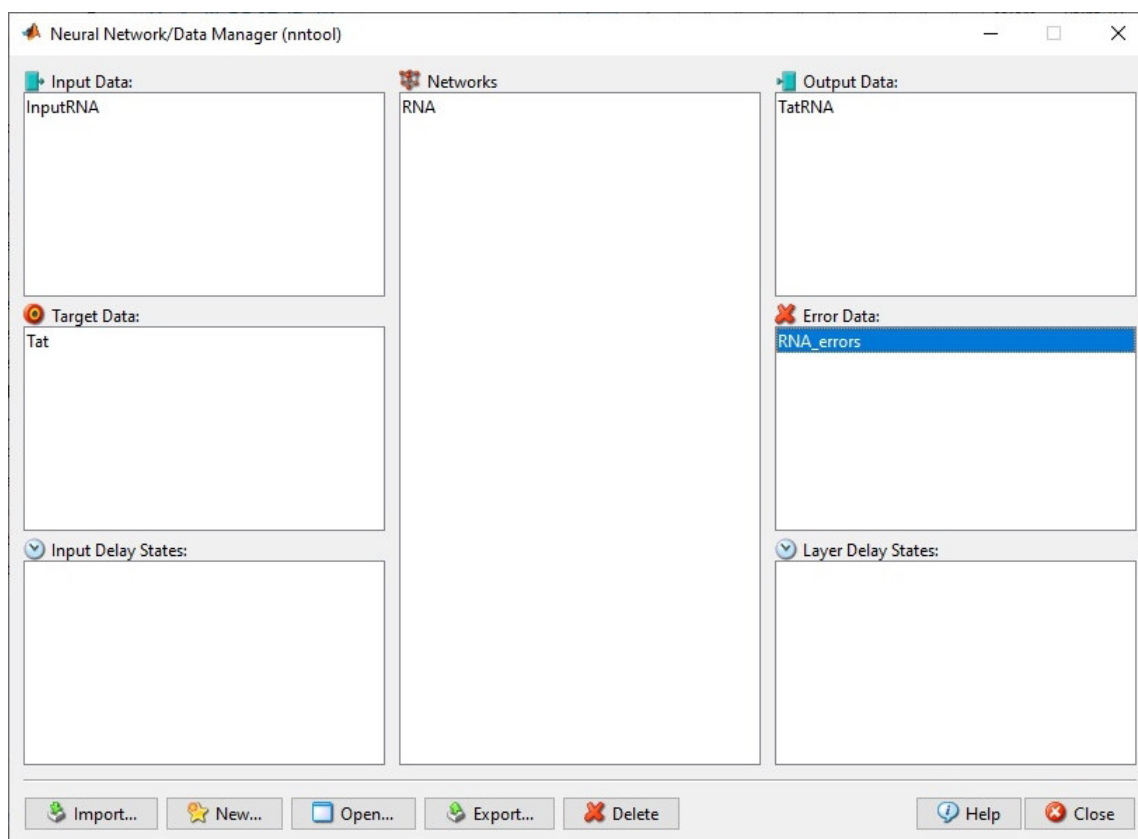


Figura 54: *Network/Data Manager*
 Fonte: Próprio Autor

Foi definido que o processo de aprendizagem empregado nesta rede será o aprendizado supervisionado (melhor descrito no capítulo 6), na qual é preciso informar dados de entrada (*Input data*) e saída desejada (*Target data*) da rede neural, para que esta autorregule seus pesos no processo de treinamento. Assim, foi fornecida a matriz "InputRNA" que contém dados de torque motor T_m e velocidade do rotor ω_r como entrada, e um vetor de dados T_{at} como *Target Data*, contendo os valores de torque de atrito não linear simulados através do modelo de atrito de Gomes, para as respectivas entradas fornecidas.

Como função de ativação dos neurônios foi utilizada a tangente hiperbólica (*tansig*) e o treinamento da rede foi através do algoritmo *backpropagation* com adição do termo *momentum* (*traingdm*). As figuras 55 e 56 demonstram respectivamente, a janela de criação da rede com os parâmetros supracitados, bem como uma visualização da rede criada.

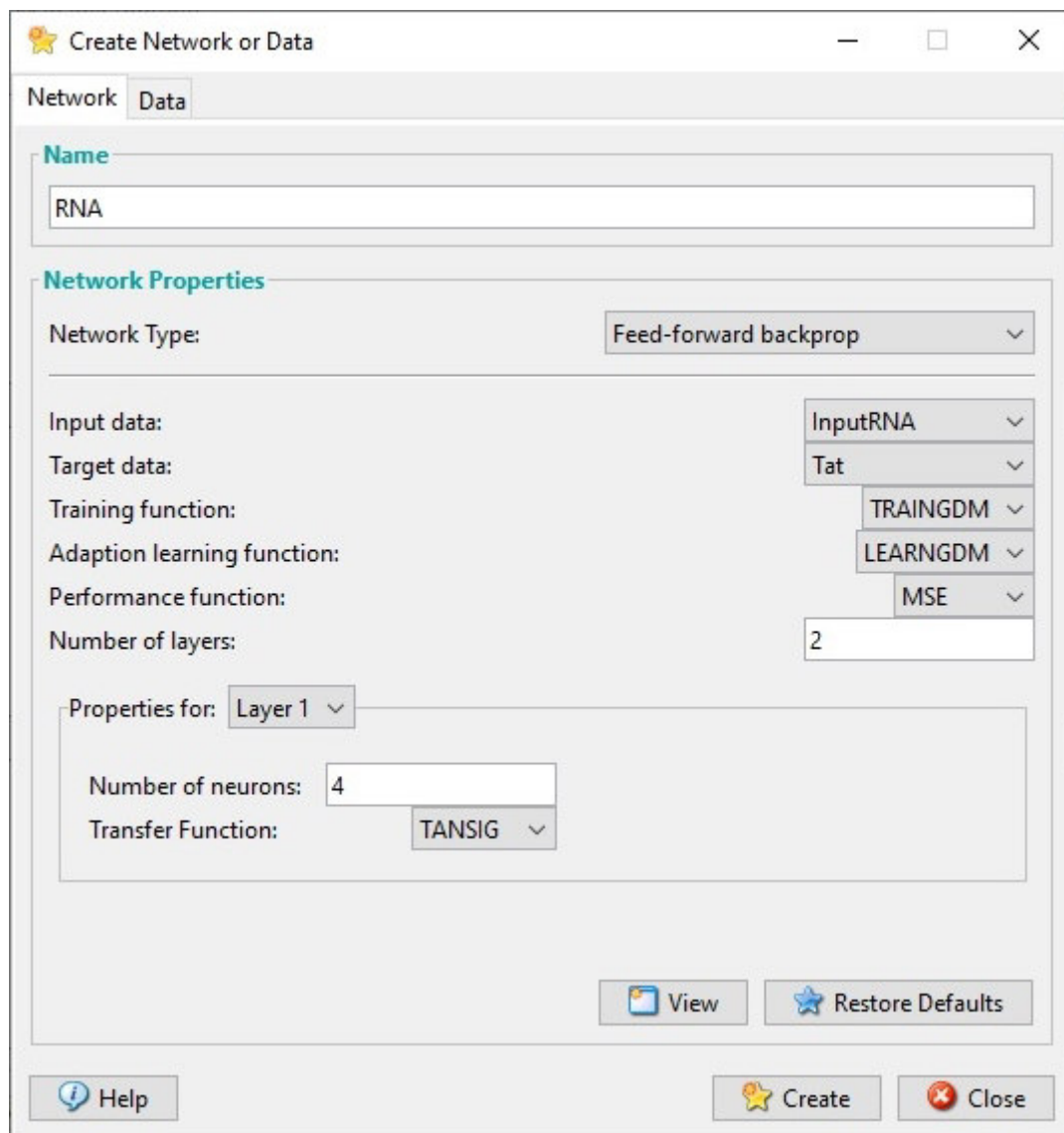


Figura 55: Janela de criação e características da RNA criada

Fonte: Próprio Autor

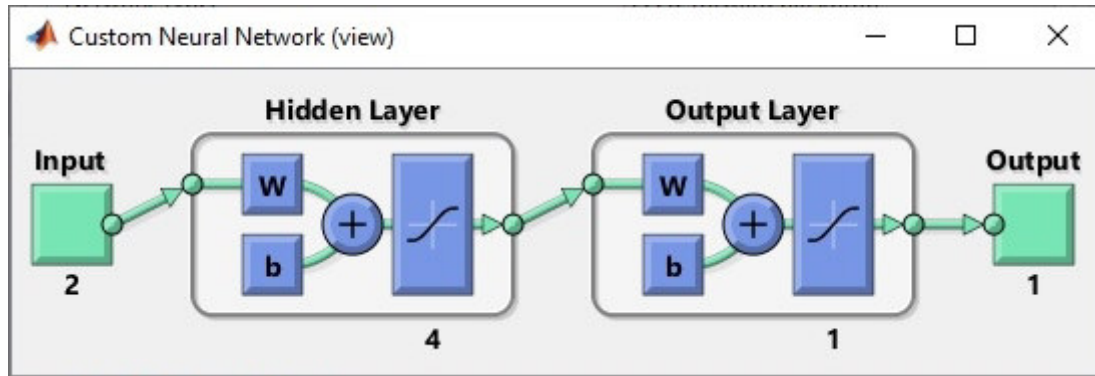


Figura 56: Visualização da RNA criada
Fonte: Próprio Autor

7.2 Treinamento da RNA

Após a criação, foi dado início ao processo de treinamento da RNA pela geração de padrões de treinamento, que são os dados que serão fornecidos como entrada e saída da rede neural. É importante salientar que, para o treinamento da RNA, são utilizados resultados experimentais e dados de simulação. O modelo do atuador robótico - equação (40) - é alimentado a partir do T_m (obtido da bancada experimental via i_a). A velocidade rotor simulada (ω_{rs}) na saída do modelo - equação (40) - e o T_m supracitado são as entradas do treinamento da RNA. Juntamente a estas variáveis também é fornecido o torque de atrito não linear T_{at} à rede neural como uma variável de saída, uma vez que o tipo de treinamento empregado é o de aprendizado supervisionado. A Figura 57 representa graficamente a forma de treinamento empregada nesta RNA bem como as variáveis fornecidas neste processo.

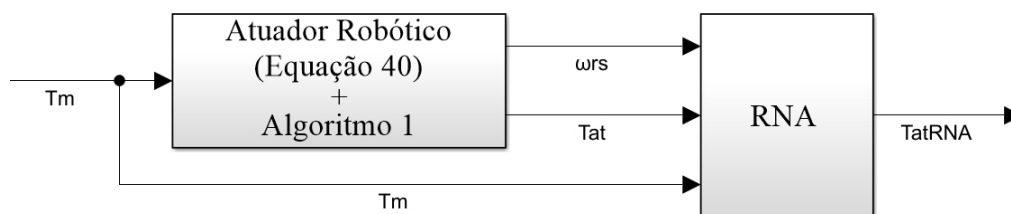


Figura 57: Configuração e variáveis empregadas no treinamento do RNA
Fonte: Próprio Autor

Uma série de simulações foram realizadas aplicando diferentes sinais de tensão de entrada no sistema para obtenção dos dados referentes a entrada da rede (torque motor T_m e velocidade rotor simulada ω_{rs}), bem como dos dados de saída da rede (torque de atrito não linear T_{at}). Em resumo, foram aplicados ao sistema diferentes tipos de sinais de entrada, todos dados experimentais, sendo eles:

- Quatro segmentos de senoide em malha aberta;

- Dez segmentos de degrau em malha aberta;

As Figuras 58, 59 e 60 representam, respectivamente, os valores de entrada de torque motor T_m e velocidade rotor simulada ω_{rs} e o valor de saída de torque de atrito não linear T_{at} , utilizados como padrões de treinamento para a rede neural.

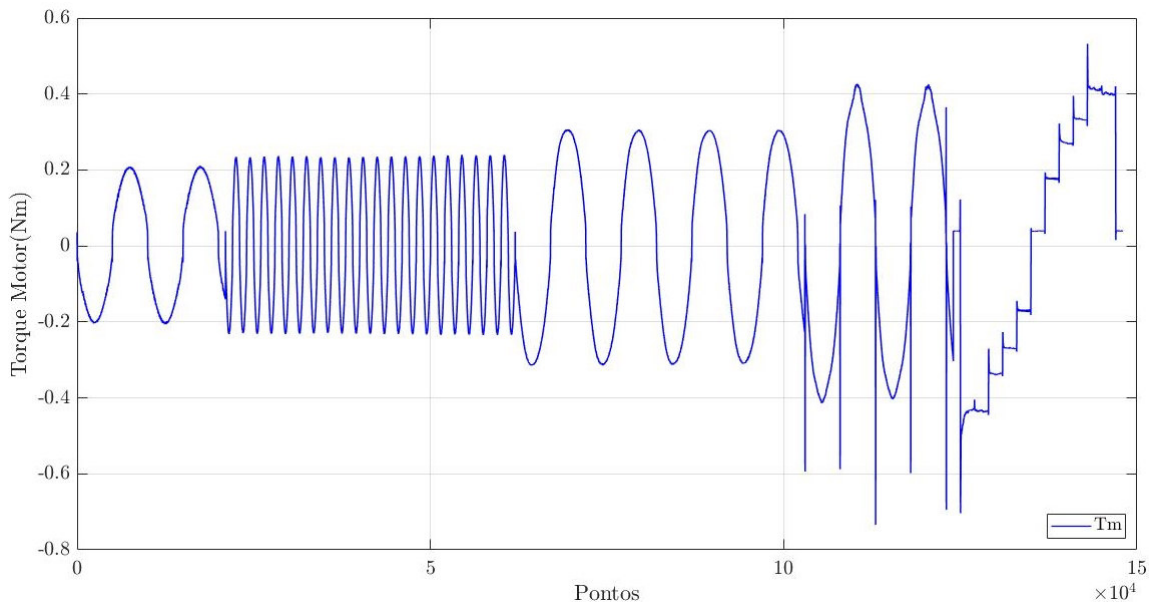


Figura 58: Padrões de treinamento - T_m

Fonte: Próprio Autor

A utilização de simulações do sistema atuador robótico e modelo de atrito em malha aberta e malha fechada é importante para que nesta etapa obtenha-se uma grande variedade de torques motores, velocidades rotores e atritos não lineares, já que a diversificação dos padrões de treinamento de uma rede neural é importante para sua generalização.

Definidos os padrões de treinamento da rede, foi iniciado o processo de treinamento da mesma, como pode ser visualizado na Figura 61.

Superada a dificuldade de tempo requerido para se treinar uma rede de alta taxa de confiabilidade, sua eficiência foi comprovada através da utilização da técnica de regressão, que é um método estatístico para descrever a relação entre variáveis de saída, ou resposta, a uma ou mais variáveis de entrada, ou variáveis de predição. Pode-se observar na Figura 62 que os resultados de regressão, tanto para dados utilizados no treinamento, validação e teste da rede apresentam eficiência maior que 99,9%. Vale ressaltar que os dados foram divididos de forma aleatória, resultando 70% como dados de treinamento, 15% como dados de validação e 15% como dados de teste. Os dados destinados a treinamento são utilizados apenas na fase de treinamento da RNA, para ajuste de pesos. Os dados destinados a validação são utilizados para medir a generalização da rede, enquanto que os dados destinados a teste são usados para medir o desempenho da rede, sem afetar a etapa de treinamentos.

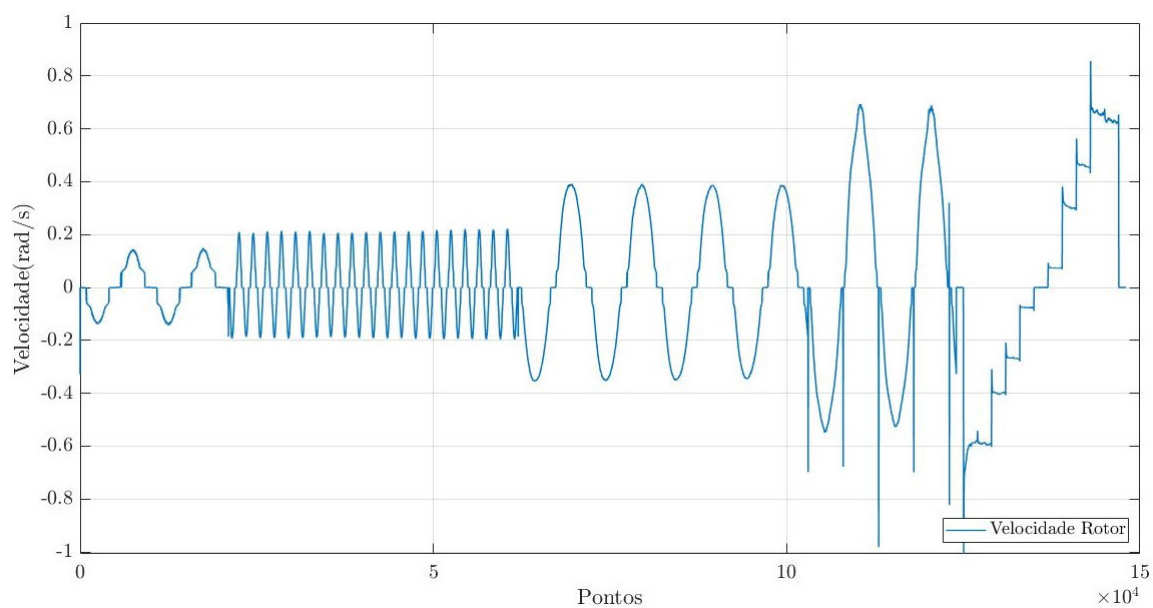


Figura 59: Padrões de treinamento - ω_{rs}
 Fonte: Próprio Autor

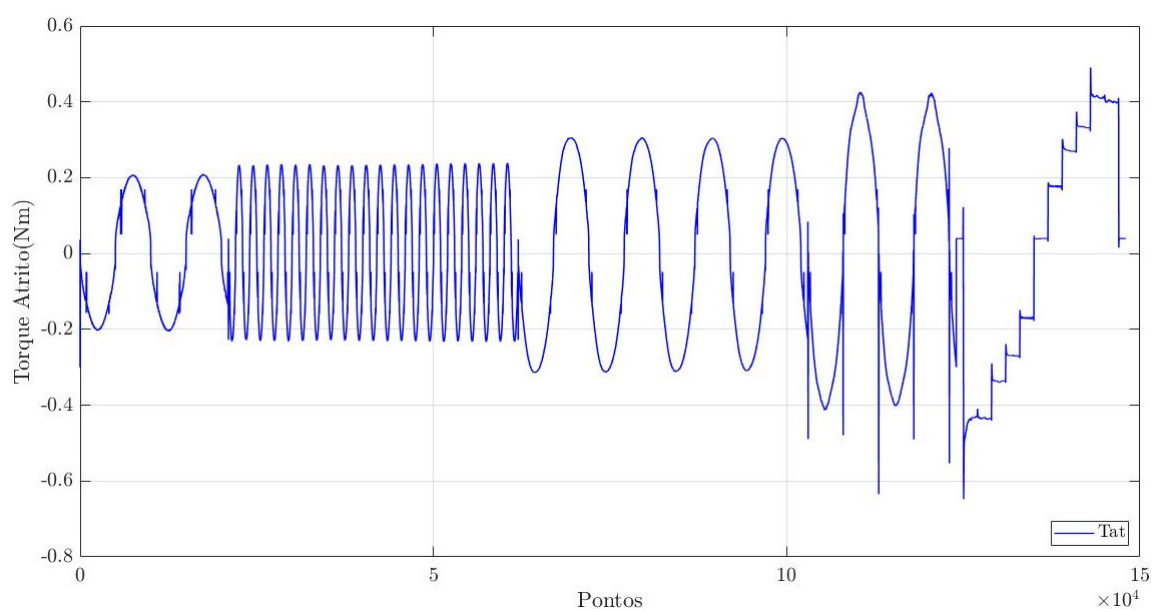


Figura 60: Padrões de treinamento - T_{at}
 Fonte: Próprio Autor

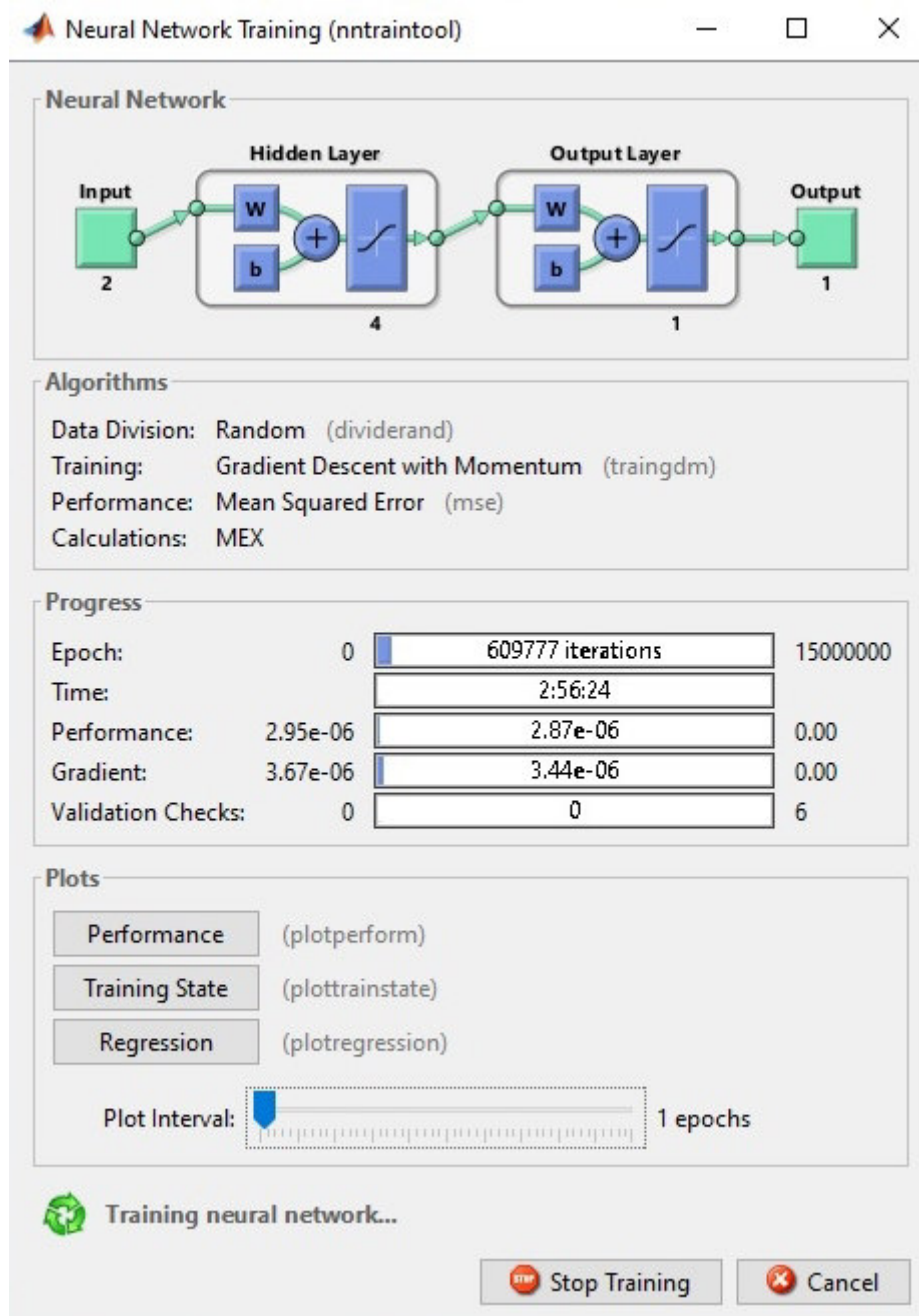


Figura 61: Processo de treinamento da RNA
Fonte: Próprio Autor

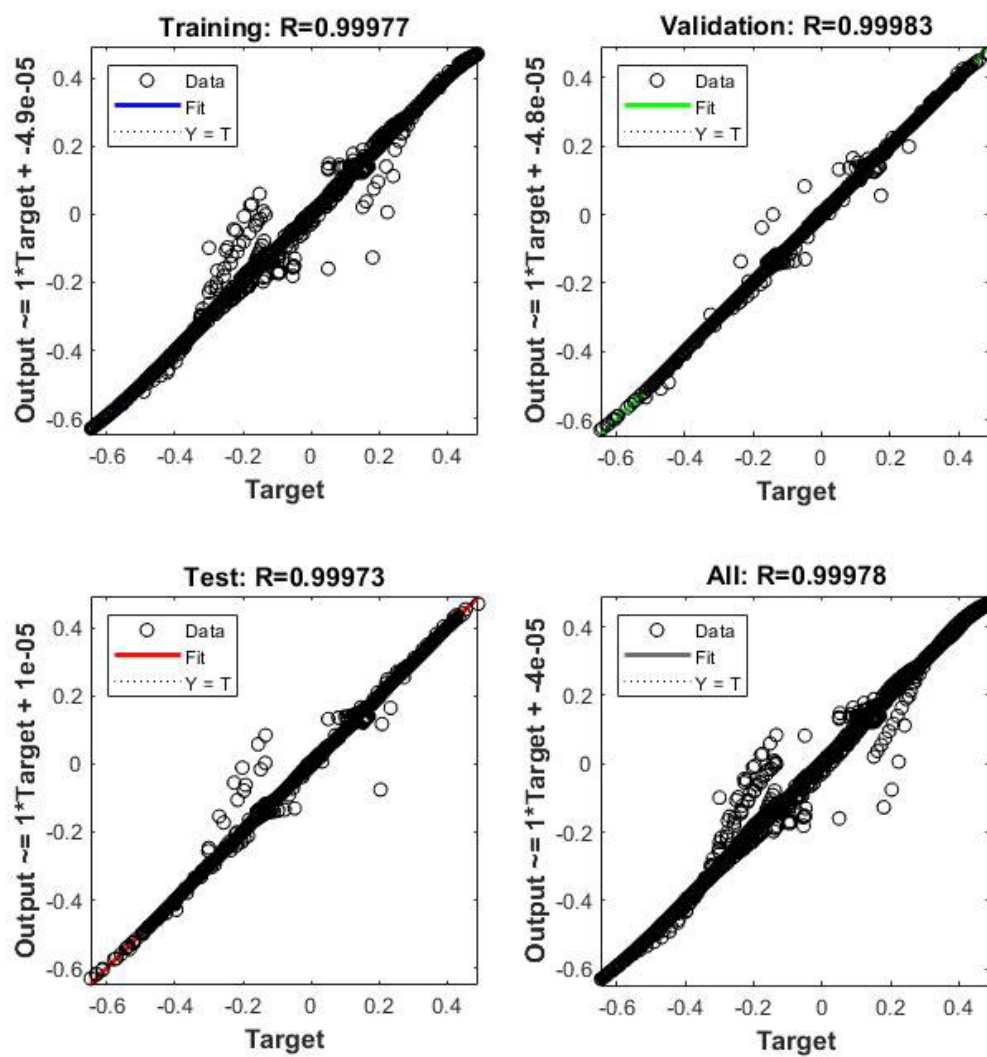


Figura 62: Precisão da rede treinada

Fonte: Próprio Autor

Por fim, foi realizada uma comparação (Figura 63) entre o torque de atrito não linear simulado com o modelo de atrito de Gomes (T_{at} , em azul), usado como um padrão de treinamento da rede neural, e o torque de atrito não linear que foi reconhecido pela rede neural artificial proposta ($T_{at}RNA$, em laranja). Os resultados se mostraram precisos, principalmente levando em consideração a grande variedade de dados de entrada do sistema.

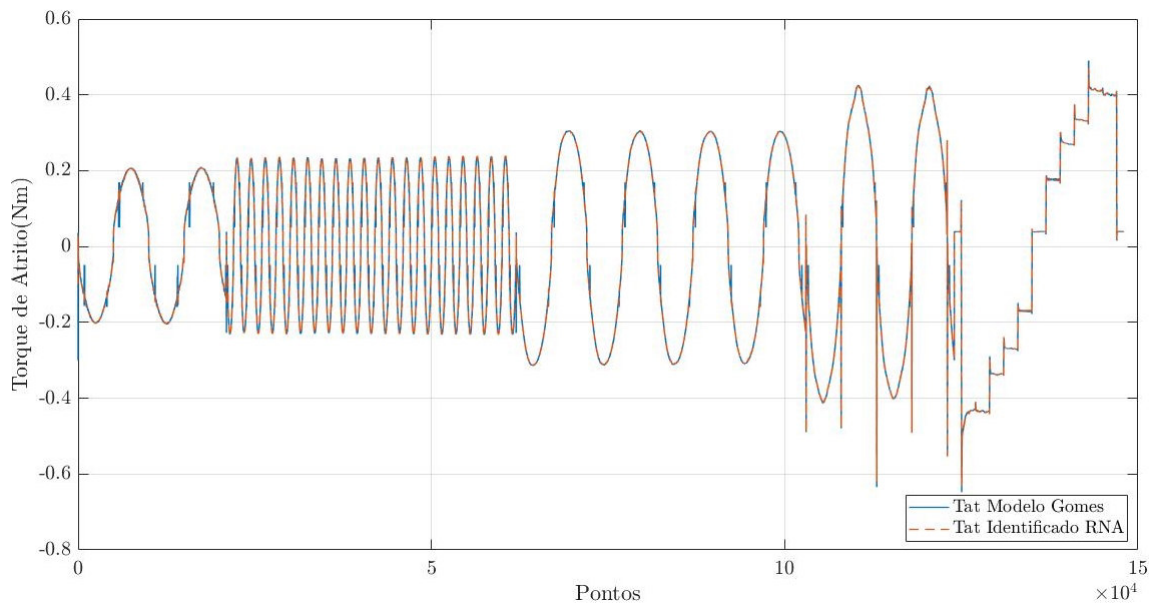


Figura 63: Torque de atrito simulado (linha azul) x torque de atrito estimado pela RNA (linha laranja)

Fonte: Próprio Autor

7.3 Validação da RNA Proposta

Dado como encerrado o processo de treinamento, a rede neural artificial proposta foi considerada pronta para ser usada uma vez que seus pesos já estão ajustados. Assim, nova simulação foi realizada para validação da RNA. Nesta simulação, foram fornecidos a rede apenas os valores de entrada (torque motor T_m e velocidade rotor ω_r) que seguirão direto o passo *forward*, gerando na saída da RNA o respectivo valor de torque de atrito não linear T_{at} . Uma vez identificado o torque de atrito, este é incorporado ao sistema com o motor CC, gerando uma resposta em velocidade simulada que será comparada com a velocidade experimental. Ressalta-se que nesta simulação a RNA substitui o modelo de atrito de Gomes na identificação do torque de atrito.

Foi utilizada uma entrada em sinal senoidal com frequência de 0,1Hz e amplitude de 20% do torque máximo T_{max} que é 2,104Nm. Os resultados podem ser visualizados nas Figuras 64, 65, 66 e 67.

É possível concluir, com a análise das figuras, que a representação da velocidade simulada com a rede neural (linha preta) é fidedigna com a velocidade experimental (li-

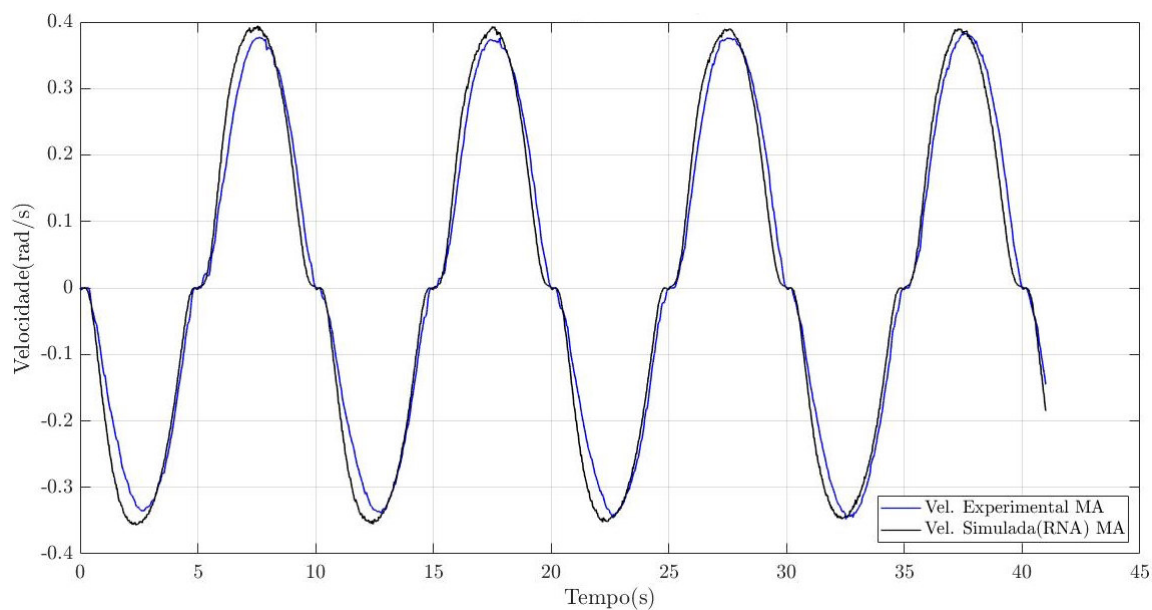


Figura 64: Comparação da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta). Entrada 20% T_{max}

Fonte: Próprio Autor

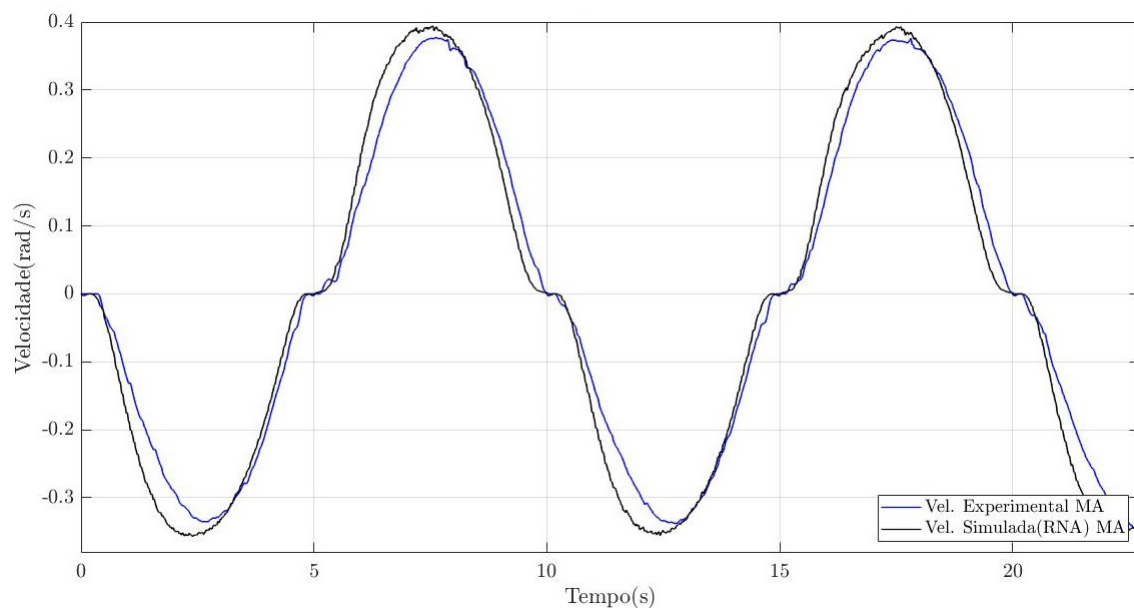


Figura 65: Comparação ampliada da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta). Entrada 20% T_{max}

Fonte: Próprio Autor

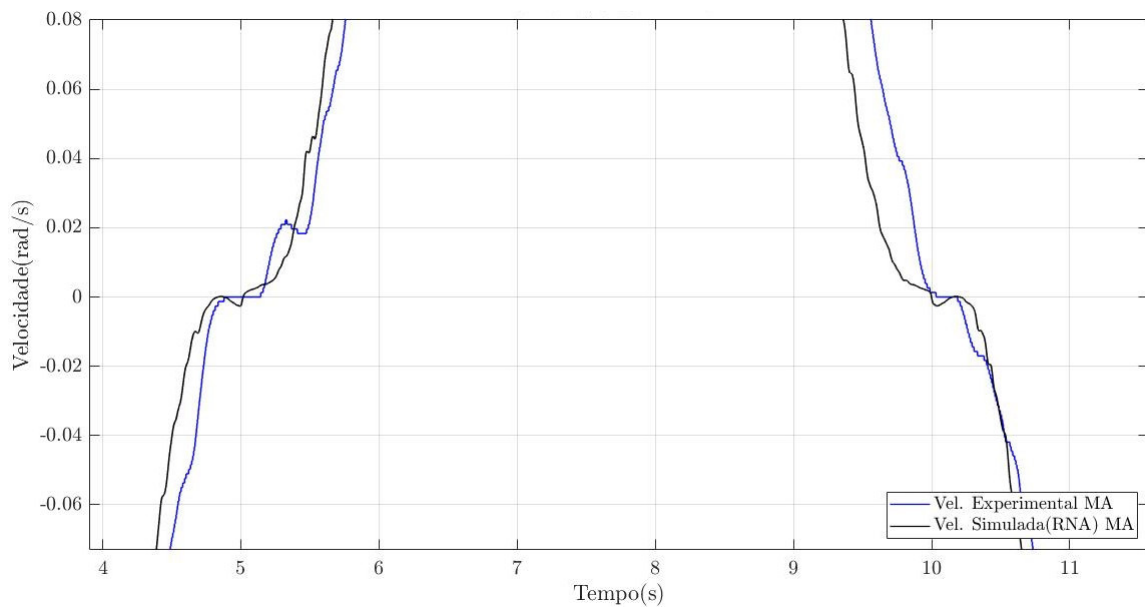


Figura 66: Comparação da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta), com foco nas inversões de velocidade. Entrada 20% T_{max}

Fonte: Próprio Autor

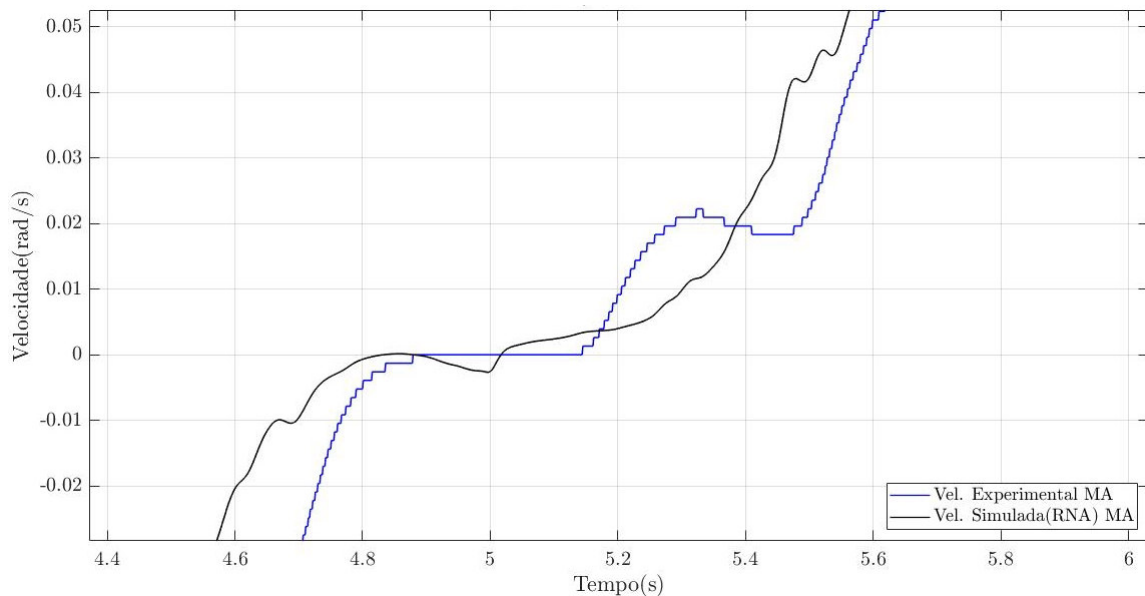


Figura 67: Comparação ampliada da velocidade experimental em MA (linha azul) x Velocidade simulada com a RNA em MA (linha preta), com foco nas inversões de velocidade. Entrada 20% T_{max}

Fonte: Próprio Autor

nha azul) obtida via modelo de atrito de Gomes. Nas regiões de inversão de velocidade (próximo de zero) a curva da velocidade simulada reproduz bem o efeito *stick-slip*. O número de dados fornecidos como padrões de treinamento para a RNA e sua arquitetura se mostraram adequadas, garantindo um bom treinamento de rede. Assim, a rede neural artificial criada se comportou eficientemente na identificação do torque de atrito não linear de atuadores rebóticos do tipo *Harmonic-Drive*.

8 RESULTADOS DE COMPENSAÇÃO COM A RNA

Para realizar a compensação do sistema, um mecanismo de compensação de atrito simples foi criado, tendo como função básica a rejeição direta do atrito não linear identificado pelo modelo de Gomes, com o atrito não linear estimado pela RNA. As simulações foram realizadas com um passo de discretização de 0,001 segundos como em todo o trabalho. A equação (48) demonstra a forma assumida pelo mecanismo de compensação:

$$T_m(t+1) = T_m(t) - T_{at}(t) + T_{at}RNA(t). \quad (48)$$

O torque motor T_m no instante $t+1$ será o próprio torque motor no instante t acrescido do torque de atrito não linear estimado na saída da RNA - $T_{at}RNA(t)$ - menos o torque de atrito não linear identificado pelo modelo Gomes $T_{at}(t)$.

Concluído o mecanismo de compensação de atrito, uma sequência de simulações foram realizadas com o intuito de demonstrar a eficiência do mecanismo de compensação tanto em respostas de velocidade quanto de posição para o atuador robótico. Para obtenção dos resultados apresentados na Figura 68, foram aplicados diferentes degraus de tensão na entrada do sistema. Estes resultados foram gerados em malha aberta tanto para velocidade quanto para posição angular. Esta figura confronta os resultados de três simulações: considerando o sistema linear, o caso do sistema não linear sem compensação e o caso do sistema não linear com a estratégia de compensação do atrito não linear. Fica evidente a melhora nos resultados gerados pela estratégia de compensação, mesmo que para valores de velocidade próximos de zero, o sistema ainda se mostra com certas deficiências que são responsáveis por pequenos erros em posição.

Já para obter os resultados apresentados na Figura 69, foram aplicados diferentes sinais senoidais de tensão na entrada do sistema. Estes resultados foram gerados em malha aberta tanto para velocidade quanto para posição angular. Novamente a figura confronta os resultados de três simulações: considerando o sistema linear, o caso do sistema não linear sem compensação e o caso do sistema não linear com a estratégia de compensação do atrito não linear. Os resultados obtidos a partir da estratégia de compensação fornecem respostas melhores tanto em posição angular quanto em velocidade. Vale destacar que o primeiro sinal senoidal, no intervalo de 0 a 10 segundos, foi de amplitude tão baixa que

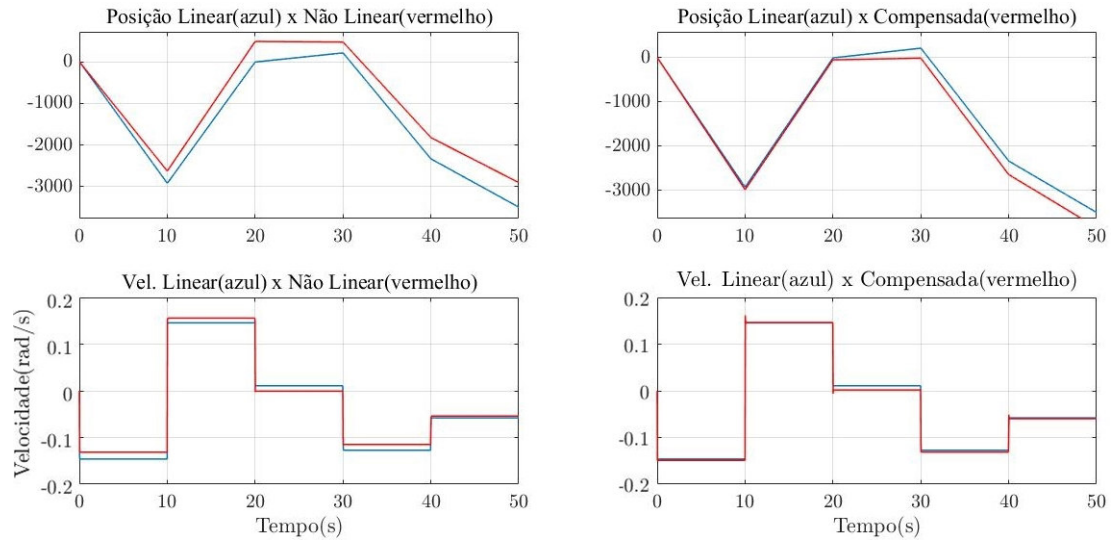


Figura 68: Resultados de simulação com entrada degrau em velocidade e posição angular do sistema linear, não linear e não linear compensado

Fonte: Próprio Autor

não se tornou visível na figura. Nesta senoide o erro de posição também ocorreu, pelas mesmas razões explicadas anteriormente.

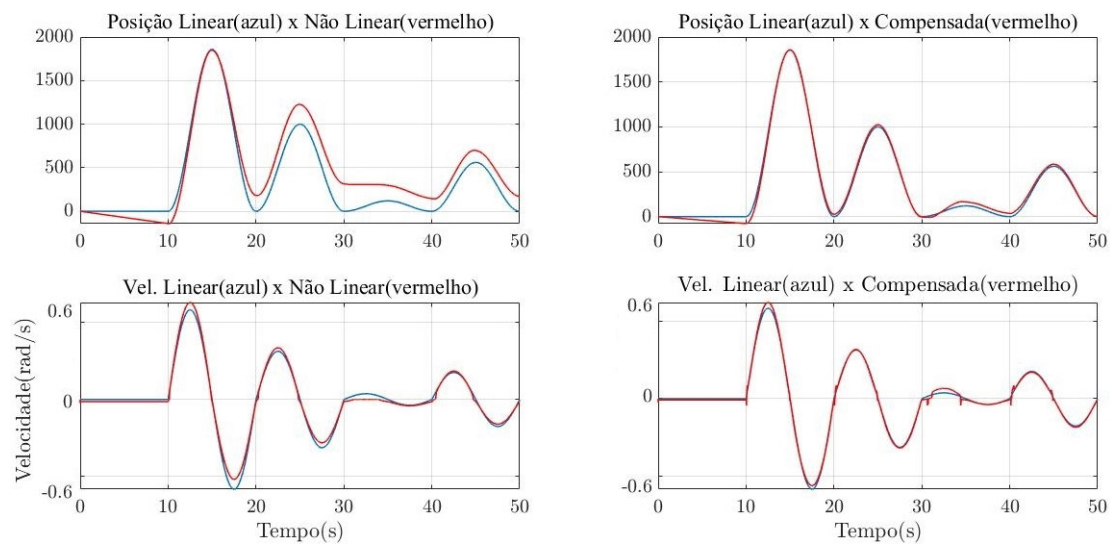


Figura 69: Resultados de simulação com entrada senoidal em velocidade e posição angular do sistema linear, não linear e não linear compensado

Fonte: Próprio Autor

9 CONCLUSÕES

Esta dissertação trata do projeto e simulação de um mecanismo de compensação de zona morta e torque de atrito não linear, uma não linearidade recorrente em motores de corrente contínua, baseado em uma rede neural artificial - RNA. Para os experimentos e simulações foi utilizado um atuador robótico do tipo moto-redutor *Harmonic-Drive*, descrevendo toda sua modelagem.

Para fundamentar o projeto proposto, realizou-se uma grande revisão bibliográfica, dividida em duas etapas principais. Em um primeiro momento, uma pesquisa mais ampla ocorreu, a fim de demonstrar a problemática envolvida e alguns métodos mais aplicados para sua resolução. A partir da segunda etapa de pesquisa, iniciou-se uma busca mais específica, focando em publicações que relacionassem a compensação da zona morta em motores de corrente contínua com redes neurais artificiais.

Em seguida foi realizada a modelagem do sistema a ser utilizado, o qual compreende um motor CC acoplado ao *Harmonic-Drive* e *encoder* incremental. Os parâmetros elétricos e mecânicos do sistema a serem utilizados em simulação foram obtidos a partir de testes de bancada. Em seguida foi realizada a validação destes parâmetros através de comparação dos resultados de simulação com resultados experimentais, para mesmos sinais de entrada.

O modelo de atrito de Gomes foi preferido ao modelo de atrito Luge para gerar os padrões de treinamento da RNA empregada, visto que este apresentou características de grande parte dos fenômenos que compõem o atrito, como o efeito *stick-slip*, de considerar diferentes parâmetros do atrito em diferentes sentidos de rotação, ser de fácil identificação paramétrica. Foi realizada a identificação dos parâmetros de atrito e posteriormente simulações com dois torques diferentes para cada modelo, com a finalidade de avaliar as respectivas características apresentadas e a validação com testes experimentais. O modelo de Gomes apresentou resultados coerentes quando comparados aos resultados experimentais.

Com base em exemplos de trabalhos e fundamentos teóricos adquiridos das pesquisas de literatura, foi dado início as etapas de desenvolvimento e aplicação de um compensador neural de atrito. Seus resultados iniciais para identificação de torque de atrito não linear,

causados por efeito da zona morta se mostraram bem animadores, superando a marca de 99,9% de precisão em relação ao torque identificado através do modelo de atrito de Gomes. Simulações foram realizadas com finalidade de validação da rede neural treinada, obtendo-se bons resultados, ainda que não foi possível representar de forma tão satisfatória o efeito *stick*, que acontece nas inversões de velocidade. Uma melhor modelagem de atrito, seguido de mais padrões de treinamento da rede neural compreendida na região de *stick* podem melhorar a representação deste efeito.

Por fim, foi simulado o compensador de atrito não linear ao modelo do motor-reductor, que correspondeu com grande parte das expectativas iniciais, realizando uma boa compensação, acarretando na melhora de desempenho, confirmadas através de resultados em malha aberta em posição e velocidade do sistema. Estes resultados confirmam que as redes neurais artificiais são ferramentas poderosas na compensação de zona morta e atrito não linear em motores elétricos de correntes contínua.

Acredita-se que esta dissertação apresentou de forma didática informações importantes sobre a modelagem de atrito, e a utilização de redes neurais artificiais na compensação de atrito em atuadores robóticos, como foi objetivado inicialmente.

Como passos futuros, testar a RNA proposta em experimentos com robôs, como por exemplo um manipulador robótico ou os robôs da equipe de futebol de robôs FURGbOL, da Universidade Federal do Rio Grande - FURG, a fim de comprovar também a eficiência deste método de compensação em sistemas físicos reais. Outra proposta seria aplicar um diferente método de compensação, como por exemplo o controle por modos deslizantes, para realizar uma comparação da eficiência das redes neurais artificiais com alguma outras estratégias de compensação.

REFERÊNCIAS

- ARMSTRONG-HELOUVRY, B. Stick slip and control in low-speed motion. *IEEE Transactions on Automatic Control*, IEEE, v. 38, n. 10, p. 1483–1496, 1993.
- ARMSTRONG-HELOUVRY, B.; DUPONT, P. Friction modeling for control. In: IEEE. *1993 American Control Conference*. [S.l.], 1993. p. 1905–1909.
- ARMSTRONG-HÉLOUVRY, B.; DUPONT, P.; WIT, C. C. D. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, Elsevier, v. 30, n. 7, p. 1083–1138, 1994.
- AZEVEDO, F. M. D.; BRASIL, L. M.; OLIVEIRA, R. C. L. de. *Redes neurais com aplicações em controle e em sistemas especialistas*. [S.l.]: Visual Books, 2000.
- BATTISTELLA, E. Extração de regras de redes neurais artificiais aplicadas ao problema da determinação da estrutura secundária de proteínas. Universidade do Vale do Rio do Sinos, 2004.
- BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. Neural network toolbox™ user's guide. In: CITESEER. *R2012a, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098*, www.mathworks.com. [S.l.], 2012.
- BECKER, S. Unsupervised learning procedures for neural networks. *International Journal of Neural Systems*, World Scientific, v. 2, n. 01n02, p. 17–33, 1991.
- BESANÇON-VODA, A.; BESANÇON, G. Analysis of a two-relay system configuration with application to coulomb friction identification. *Automatica*, Elsevier, v. 35, n. 8, p. 1391–1399, 1999.
- BONA, B.; INDRI, M. Friction compensation in robotics: an overview. In: IEEE. *Proceedings of the 44th IEEE Conference on Decision and Control*. [S.l.], 2005. p. 4360–4367.
- BOROWSKI, M. F. Analysis and control of a dc electric drive exhibiting dead zone characteristics. *University of New Hampshire, Department of Electrical and computer Engineering*, 2013.
- BRAGA, A. de P.; FERREIRA, A. C. P. de L.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: LTC Editora Rio de Janeiro, Brazil:, 2007.
- CAMPOS-DELGADO, D. U. et al. Closed-loop control of renal perfusion pressure in physiological experiments. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 60, n. 7, p. 1776–1784, 2013.

CASANOVA, C. C. et al. Modelagem e compensação de atrito em robôs manipuladores com flexibilidades nas juntas: identificação, análise e resultados experimentais. Florianópolis, SC, 2007.

CYBENKO, G. Continuous valued neural networks with two hidden layer are sufficient. medford: Tufts university. 1988.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989.

DHAOUADI, R. Nonlinear friction compensation in harmonic drives with hysteresis. In: IEEE. *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*. [S.l.], 2003. v. 1, p. 278–283.

DORF, R. C.; BISHOP, R. H. *Modern control systems*. [S.l.]: Pearson, 2011.

FITZGERALD, A. E. et al. *Electric machinery*. [S.l.]: McGraw-Hill New York, 2003. v. 5.

FRONZA, G. R. Desenvolvimento de um protótipo de equipamento de avaliação macroscópica de modelos de atrito. 2015.

GARTNER, L.; HIATT, J. *Tratado de Histologia em cores. Ed.* [S.l.]: Guanabara-Koogan. Rio de Janeiro, 1999.

GERVINI, V. I.; GOMES, S. C. Identificação e compensação de atritos não lineares de atuadores robóticos via redes neurais artificiais. In: *V Congresso Brasileiro de Redes Neurais*. [S.l.: s.n.], 2001. p. 475–480.

GERVINI, V. I.; GOMES, S. C. P.; ROSA, V. D. A new robotic drive joint friction compensation mechanism using neural networks. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, SciELO Brasil, v. 25, n. 2, p. 129–139, 2003.

GOMES, S. C. P. Modelagem de atritos internos às articulações de robôs manipuladores. *XIII COBEM, CD-ROM, Belo Horizonte*, 1995.

GOMES, S. C. P. et al. Validação experimental de modelos dinâmicos de atuadores e estruturas flexíveis. In: *XV Congresso Brasileiro de Automática*. [S.l.: s.n.], 2004.

GOMES, S. C. P.; ROSA, V. S. D. A new approach to compensate friction in robotic actuators. In: IEEE. *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. [S.l.], 2003. v. 1, p. 622–627.

GOMES, S. C. P.; ROSA, V. S. da; ALBERTINI, B. C. Active control to flexible manipulators. *IEEE/ASME Transactions on Mechatronics*, IEEE, v. 11, n. 1, p. 75–83, 2006.

GOMES, S. d. S. *Modelagem e controle de atuadores robóticos e veículos subaquáticos não tripulados*. Dissertação (Mestrado), 2011.

GOORVER, M.; WEISS, M. et alli, *Robótica Tecnologia e Programação*. [S.l.]: McGraw-Hill, São Paulo, 1988.

GRASSI, L. H. M. Identificação e controle de sistemas dinâmicos utilizando redes wavelets. Universidade Estadual Paulista (UNESP), 2004.

- HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR, 1994.
- HAYKIN, S. *Redes Neurais Artificiais: Princípios e Práticas*. 2ª edição. [S.l.]: Editora Bookman, Porto Alegre, 900p, 2001.
- HE, Y.; WANG, J.; HAO, R. Adaptive robust dead-zone compensation control of electro-hydraulic servo systems with load disturbance rejection. *Journal of Systems Science and Complexity*, Springer, v. 28, n. 2, p. 341–359, 2015.
- HSU, C.-F. Intelligent total sliding-mode control with dead-zone parameter modification for a dc motor driver. *IET Control Theory & Applications*, IET, v. 8, n. 11, p. 916–926, 2014.
- HUGHES, A.; DRURY, B. *Electric motors and drives: fundamentals, types and applications*. [S.l.]: Newnes, 2019.
- JANG, J. O.; CHUNG, H. T.; JEON, G. J. Saturation and deadzone compensation of systems using neural network and fuzzy logic. In: IEEE. *Proceedings of the 2005, American Control Conference, 2005*. [S.l.], 2005. p. 1715–1720.
- JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]. *IEEE Transactions on automatic control*, IEEE, v. 42, n. 10, p. 1482–1484, 1997.
- JUN, O.; GI, J. A parallel neuro-controller for dc motors containing nonlinear friction. *Neurocomputing*, v. 30, n. 3, p. 233–248, 2000.
- KARA, T.; EKER, I. Nonlinear closed-loop direct identification of a dc motor with load for low speed two-directional operation. *Electrical Engineering*, Springer, v. 86, n. 2, p. 87–96, 2004.
- KRATZER, K. P. *Neuronale Netze: Grundlagen und Anwendungen*. [S.l.]: Hanser, 1991.
- LYGOURAS, J. N. Dc thruster controller implementation with integral anti-wind up compensator for underwater rov. *Journal of Intelligent and Robotic Systems*, Springer, v. 25, n. 1, p. 79–94, 1999.
- MACHADO, C. C. et al. Adaptive neuro-fuzzy friction compensation mechanism to robotic actuators. In: IEEE. *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*. [S.l.], 2007. p. 581–586.
- MAMEDE, J. *Instalações elétricas industriais*. Rio de Janeiro: Editora LTC, 2010.
- MATHWORKS. *Company Overview*, p. 1–2, Feb 2020.
- MATSUNAGA, V. Y. Curso de redes neurais utilizando o matlab. *Belém do Pará*, 2012.
- MENDEL, J. M. Adaptive learning and pattern recognition systems. *Theory and applications*, Academic Press, 1970.
- MIOTTO, F. E. M. *Modelagem Matemática da Dinâmica do Atrito e sua Aplicação no controle ótimo de um atuador hidráulico*. Tese (Doutorado) — Dissertação de mestrado em modelagem matemática. Ijuí: UNIJUÍ, 2009, 110p, 2009.

- MOHAN, N. *Electric machines and drives: a first course*. [S.l.]: Wiley,, 2012.
- OGATA, K. *Modern control engineering*. [S.l.]: Prentice hall, 2010.
- OGATA, K.; SEVERO, B. *Engenharia de controle moderno*. [S.l.]: Prentice Hall do Brasil, 1998.
- OLSSON, H.; ASTROM, K. J. Observer-based friction compensation. In: IEEE. *Proceedings of 35th IEEE Conference on Decision and Control*. [S.l.], 1996. v. 4, p. 4345–4350.
- PERONDI, E. A. Controle não-linear em cascata de um servoposicionador pneumático com compensação do atrito. 2002.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- ROSSATO, D. B. *Desenvolvimento de um sistema aberto para ensino de robôs manipuladores*. Tese (Doutorado) — Universidade de São Paulo, 2009.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.
- SALCEDO, C. M. et al. Modelo de controlador com compensação de zona morta para uma base móvel de robôs. *Universidade Federal da Bahia: Dissertação de Mestrado*, v. 3, 2010.
- SCHMITT, P. Redes neurais artificiais aplicadas na classificação litológica das formações palermo e rio bonito na jazida do leão-rs, com base em perfis geofísicos. Universidade do Vale do Rio dos Sinos, 2009.
- SELMIC, R.; LEWIS, F. Deadzone compensation in systems using neural motion control networks. *IEEE Trans. Autom. Control*, v. 45, n. 4, p. 602–613, 2000.
- SHAFER, W. H. *Masters Theses in the Pure and Applied Sciences: Accepted by Colleges and Universities of the United States and Canada. Volume 24*. [S.l.]: Springer Science & Business Media, 2013. v. 24.
- SHEPHERD, G. M. *The synaptic organization of the brain*. [S.l.]: Oxford university press, 2004.
- SILVA, A. N. R. d. et al. *SIG: uma plataforma para introdução de técnicas emergentes no planejamento urbano, regional e de transportes*. [S.l.]: Ed. Autores, 2004.
- SILVA, G. V. M. da. Controle não linear. *Escola superior de tecnologia de Setúbal*, 2006.
- SLOTINE, J.-J. E.; LI, W. et al. *Applied nonlinear control*. [S.l.]: Prentice hall Englewood Cliffs, NJ, 1991. v. 199.
- SOUZA, L. P. de; AZZOLIN, R. Z. Dead zone compensation in direct current motors: A review. *Anais da Sociedade Brasileira de Automática*, v. 2, n. 1, 2020.

SUNADA, M. M. et al. Acionamento de um posicionador linear de ultraprecisão empregando uma redução harmonic drive com controle de velocidade. Florianópolis, SC, 2007.

TAFNER, M. A.; XEREZ, M. de; FILHO, I. W. R. *Redes neurais artificiais: introdução e princípios de neurocomputação*. [S.l.]: Eko, 1995.

TAGHIRAD, H.; BELANGER, P. An experimental study on modelling and identification of harmonic drive systems. In: IEEE. *Proceedings of 35th IEEE Conference on Decision and Control*. [S.l.], 1996. v. 4, p. 4725–4730.

TAKROURI, M.; DHAOUADI, R. Adaptive identification and compensation of nonlinear friction in a voice-coil linear servomotor. In: IEEE. *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*. [S.l.], 2019. p. 455–460.

TAKROURI, M.; DHAOUADI, R. Adaptive identification and compensation of nonlinear friction in a voice-coil linear servomotor. In: IEEE. *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*. [S.l.], 2019. p. 455–460.

TENG, L.; YANG, H.; WANG, Y. Model reference tracking control of linear motor with dead-zone via switched systems subjected to time-varying delay. In: IEEE. *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. [S.l.], 2016. p. 589–594.

THØGERSEN, P.; BLAABJERG, F. Adjustable speed drives in the next decade: Future steps in industry and academia. *Electric Power Components and Systems*, Taylor & Francis, v. 32, n. 1, p. 13–31, 2004.

VALDIERO, A. C.; GUENTHER, R.; NEGRI, V. J. D. New methodology for identification of the dead zone in proportional directional hydraulic valves. In: *18th International Congress of Mechanical Engineering, Ouro Preto, Brazil, Nov.* [S.l.: s.n.], 2005. p. 6–11.

VALENÇA, M.; LUDERMIR, T. Introdução às redes neurais. *Galvão, CO; Valença, MJS (Org.). Sistemas Inteligentes; aplicações a recursos hídricos e ciências ambientais*, p. 19–59, 1999.

WEG, C. WEG, 2005. Disponível em: http://catalogo.weg.com.br/tec/_cat/tech/_motor/_sel/_web.asp.

WIT, C. C. D. et al. A new model for control of systems with friction. *IEEE Transactions on automatic control*, IEEE, v. 40, n. 3, p. 419–425, 1995.

WU, C. H.; MCLARTY, J. W. *Neural networks and genome informatics*. [S.l.]: Elsevier, 2012.

ZEILMANN, A. P. et al. Teoría e experimentação no estudo sobre atritos em atuadores robóticos. *Mecânica Computacional*, v. 29, n. 26, p. 2689–2708, 2010.

ANEXO A INFORMAÇÕES ADICIONAIS SOBRE REDES NEURAIS ARTIFICIAIS

A.1 Histórico das Redes Neurais Artificiais

O trabalho pioneiro de Warren McCulloch e Walter Pitts em 1943 foi o primeiro modelo artificial de um neurônio biológico. Em seu artigo intitulado “*A Logical Calculus of the Ideas Immanent in Nervous Activity*”, os autores descrevem um cálculo lógico de redes neurais que unificou o estudo de lógica matemática com os estudos da neurofisiologia. Segundo Haykin (2001), com um número suficiente de unidades simples e com conexões sinápticas ajustadas apropriadamente e operando de forma síncrona, McCulloch e Pitts mostraram que uma rede assim constituída realizaria a computação de qualquer função computável. Como mencionado por Braga, Ferreira e Ludermir (2007), o trabalho de McCulloch e Pitts se concentra muito mais em descrever um modelo artificial de um neurônio e apresentar suas capacidades computacionais do que em apresentar técnicas de aprendizado.

O pioneirismo em relação ao estudo das técnicas de aprendizado foi apresentado no livro “*The Organization of Behavior*”, do psicólogo Donald Hebb. O autor descreveu de forma matemática que a variação dos pesos de entrada nos nodos é responsável pela capacidade de aprendizado de uma rede neural artificial. A regra de Hebb, ainda hoje é utilizada em algoritmos de aprendizado. Mais tarde, Widrow e Hoff sugeriram uma regra de aprendizado conhecida como regra delta, ainda hoje bastante utilizada (AZEVEDO; BRASIL; OLIVEIRA, 2000).

Em 1958, Frank Rosenblatt desenvolveu uma nova topologia de RNA e demonstrou que se fossem acrescidas sinápsas ajustáveis, as RNAs poderiam ser treinadas para classificação de certos tipos de padrões. A rede utilizada era chamada de perceptron (ROSENBLATT, 1958). O perceptron simples descrito por Rosenblatt possui três camadas: a primeira recebe as entradas do exterior e possui conexões fixas (retina); a segunda recebe impulsos da primeira através de conexões cuja eficiência de transmissão (peso) é ajustável e, por sua vez, envia saídas para a terceira camada (resposta) (BRAGA; FERREIRA; LUDERMIR, 2007).

Na década de 70, os estudos com redes neurais foram adormecidos, principalmente por conta do livro publicado por Minsky e Papert em 1969. Neste livro (*Perceptrons: an introduction to computational geometry*) os autores teceram duras críticas ao perceptron, devido sua incapacidade de solucionar problemas que não eram linearmente separáveis. Destacaram também que o algoritmo de aprendizado que garantia a convergência para modelos com uma única camada não garantia o mesmo para redes perceptron com múltiplas camadas.

As pesquisas com as redes neurais só voltaram a recuperar sua credibilidade com o trabalho do físico e biólogo John Hopfield, em 1982, que contestou, com sucesso, as teses matemáticas de Minsky e Papert, e assim, restaurando o interesse pelas pesquisas na área (TAFNER; XEREZ; FILHO, 1995). Já em meados de 1986, foi desenvolvido por Rumelhart, Hilton e Williams (*Learning representations by back-propagating errors*) o algoritmo de treinamento de retropropagação (*backpropagation*). Este algoritmo reforçou que a visão de Minsky e Papert sobre o perceptron eram bastante pessimistas e que as RNAs de múltiplas camadas são capazes de resolver “problemas difíceis de aprender” (BRAGA; FERREIRA; LUDERMIR, 2007). Este trabalho exerceu forte influência na utilização do algoritmo *backpropagation*, tornando-o o mais popular para o treinamento de perceptrons de múltiplas camadas.

A.2 Neurônio Biológico

O entendimento do funcionamento do cérebro humano se tornou mais fácil após o trabalho pioneiro de Ramón y Cajál (1911) intitulado “*Histologie du système nerveux de l’homme et des vertébrés*”, que introduziu a ideia dos neurônios como constituintes estruturais do cérebro. O neurônio é uma célula com núcleo e corpo (soma), da qual reações químicas e elétricas realizam o processamento e a transmissão de informações através de impulsos nervosos. Estima-se que haja aproximadamente 10 bilhões de neurônios no córtex humano e 60 trilhões de sinapses ou conexões (SHEPHERD, 2004).

Os neurônios biológicos são divididos em três seções: o corpo celular, os dendritos e o axônio. Cada uma destas seções possui funções específicas que se complementam. Os dendritos são ramificações arbóreas que têm a função de receber as informações (impulsos nervosos) de outro neurônio e conduzi-las até o corpo celular. Uma vez a informação no corpo celular, ela será processada e novos impulsos serão gerados. Estes novos impulsos, que serão transmitidos a outros neurônios, passarão pelo axônio, um tubo longo e fino, que quase tocam os dendritos do próximo neurônio.

Ao ponto de conexão entre um terminal axonal de um neurônio e o dendrito de outro neurônio se dá o nome de sinapse. Segundo Gartner e Hiatt (1999), as sinapses são os locais onde os impulsos nervosos são transmitidos de uma célula pré-sináptica (o neurônio) para uma célula pós-sináptica (outro neurônio). Assim, as sinapses permitem que os

neurônios se comuniquem uns com os outros e com células efetoras. É 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 – isto é, o fluxo de informação – entre os nodos na rede neural (BRAGA; FERREIRA; LUDERMIR, 2007).

O comportamento individual dos neurônios biológicos, representado na Figura 70, é bem compreendido do ponto de vista funcional, e é nesse comportamento conhecido que se baseiam as redes neurais artificiais.

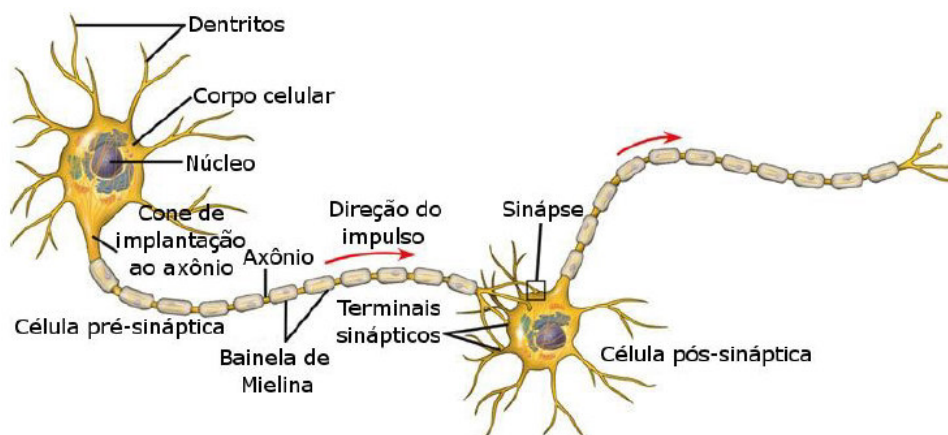


Figura 70: Representação de um neurônio biológico
Fonte: (SESI Educa, 2014)

ANEXO B SINTONIA DO CONTROLADOR PI

Uma vez que para um bom treinamento da rede neural é importante gerar diversos padrões de treinamento, incluindo dados em malha fechada, torna-se necessário o desenvolvimento e aplicação de um controlador. No caso deste trabalho, um controlador do tipo proporcional-integrativo (controlador PI) foi utilizado. Como definido por Ogata (2010), o controle PI utiliza duas das três contribuições do Controle PID (*Proportional-Integrative-Derivative*) e, para o controle PI, tem-se a garantia de erro nulo em regime permanente, devido a ação integral e uma rápida convergência no rastreamento de referência (DORF; BISHOP, 2011). Este tipo de controlador é muito difundido na literatura, possui baixo custo computacional e apresenta simplicidade de implementação. Sendo assim, foi criado no Simulink um controlador PI, que é ilustrado na Figura 71.

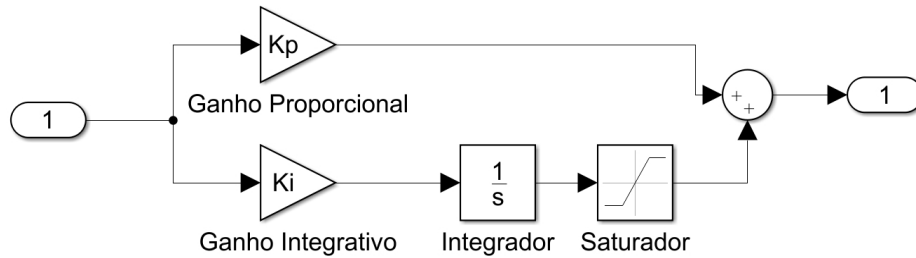


Figura 71: Representação do Controlador PI

Fonte: Próprio Autor

A sintonia do controle PI utilizada neste trabalho é fundamentada sobre o comportamento dinâmico desejado. Deseja-se que a resposta deste sistema de controle apresente como critérios de desempenho, ultrapassagem percentual (M_p) de no máximo 1% e um tempo de acomodação (t_s) de 0,03 segundos (critério de 2%).

Atribuindo o valor de ultrapassagem percentual em (49), encontra-se o coeficiente de amortecimento $\zeta = 0,83$ e, a partir de (50), aplicando o valor do tempo de acomodação t_s e o ζ , obtém-se a frequência natural $\omega_n = 160,64 \text{ rad/s}$.

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \rightarrow \zeta = \sqrt{\frac{\ln^2(M_p)}{\pi^2 + \ln^2(M_p)}} \quad (49)$$

$$t_s = \frac{4}{\omega_n \zeta} \quad (50)$$

A estratégia de controle implementada pode ser observada no diagrama de bloco apresentado na Figura 72, na qual seu funcionamento se baseia em receber uma velocidade de referência $\omega_{ref}(s)$, que é comparada a uma velocidade $\omega(s)$ medida pelo sensor, gerando um erro $e(s)$. Este erro será a entrada do controlador $C(s)$, que gera em sua saída um sinal de controle $u(s)$. Este sinal de controle será aplicado a planta $G(s)$, no caso deste trabalho, o atuador robótico do tipo moto-redutor com um motor CC.

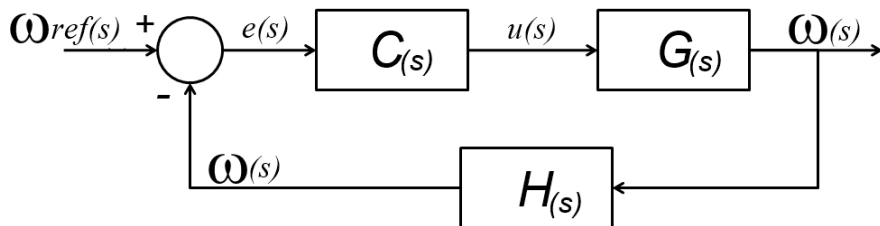


Figura 72: Malha de controle de velocidade com controlador PI
Fonte: Próprio Autor

A função de transferência deste sistema em malha fechada para o controle de velocidade se configura como

$$\frac{\omega(s)}{\omega_{ref}(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)H(s)}, \quad (51)$$

sendo,

$$H(s) = 1, \quad (52)$$

e $C(s)$, por se tratar de um controlador do tipo PI (que possui ganho proporcional K_p e ganho integrativo K_i)

$$C(s) = \frac{K_p s + K_i}{s}. \quad (53)$$

A função de transferência da planta $G(s)$, que representa um motor CC tendo como entrada a tensão de entrada V_a e como saída a velocidade $\omega(s)$, é encontrada após substituição dos valores da Tabela 8 em (14). Logo,

$$G(s) = \frac{\omega(s)}{V_a} = \frac{8,17e^6}{s^2 + 1012,5s + 1,398e^5}. \quad (54)$$

A fim de diminuir a complexidade da análise do sistema, bem como do projeto do controlador, foi realizada a redução da ordem da função de transferência $G(s)$, através do método do lugar das raízes, que permite a análise da localização dos polos de uma função de transferência no plano- s , a qual afeta diretamente a resposta transitória do sistema. Através da equação (55), pode-se observar que o sistema contém dois polos em malha

aberta e nenhum zero. Os polos são -847,55 e -164,93. Assim, o polo -164,93, mais próximo do eixo imaginário, representa o polo dominante, enquanto que o polo mais distante de zero é relacionado as dinâmicas elétricas do motor, exercendo menor influência no sistema.

$$\frac{\omega(s)}{V_a} = \frac{8,17e^6}{(s + 847,55)(s + 164,93)} \quad (55)$$

Segundo Ogata (2010), se a magnitude da parte real de um polo for de 5 a 10 vezes maior que a magnitude da parte real de um polo ou de um par de polos complexos dominantes, então este polo pode ser simplificado por ser insignificante na resposta transitória. Como o segundo polo é aproximadamente 5 vezes maior que o primeiro, este se torna simplificável. Eliminando este polo e mantendo o ganho do sistema, a nova FT simplificada $G_r(s)$ é apresentada em (56).

$$G_r(s) = \frac{8,17e^6}{847,55(s + 164,93)} \rightarrow \frac{58,44}{0,0061s + 1} \quad (56)$$

Com essa configuração, o sistema também pode ser representado por:

$$G_r(s) = \frac{K_0}{s\tau + 1} \quad (57)$$

na qual o tempo em que o sinal de saída demora para atingir 63,2% da amplitude do sinal em regime permanente $\tau = 0,0061$ s e o ganho do sistema $K_0 = 58,44$ (OGATA, 2010).

Substituindo as equações (52), (53) e (56) na equação (51), têm-se

$$\frac{\omega(s)}{\omega_{ref}(s)} = \frac{\frac{K_p s + K_i}{s} \cdot \frac{K_0}{s\tau + 1}}{1 + \frac{K_p s + K_i}{s} \cdot \frac{K_0}{s\tau + 1} \cdot 1}, \quad (58)$$

$$\frac{\omega(s)}{\omega_{ref}(s)} = \frac{(K_p s + K_i)K_0}{s^2\tau + s + (K_p s + K_i)K_0}, \quad (59)$$

$$\frac{\omega(s)}{\omega_{ref}(s)} = \frac{(K_0 K_p)s + K_i K_0}{s^2\tau + (K_0 K_p + 1)s + K_i K_0}, \quad (60)$$

resultando em:

$$\frac{\omega(s)}{\omega_{ref}(s)} = \frac{\frac{(K_0 K_p)s}{\tau} + \frac{K_i K_0}{\tau}}{s^2 + \frac{(K_0 K_p + 1)s}{\tau} + \frac{K_i K_0}{\tau}}. \quad (61)$$

De acordo com Ogata (2010), o modelo de referência para um sistema de segunda ordem é dado por:

$$\frac{y}{u} = \frac{2\zeta\omega_n + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \quad (62)$$

Realizando a equivalência dos polinômios característicos das equações (61) e (62), é

possível obter os ganhos do controlador, sendo o ganho proporcional K_p :

$$K_p = \frac{2\zeta\omega_n\tau - 1}{K_0}, \quad (63)$$

$$K_p = \frac{2 \cdot 0,83 \cdot 160,64 \cdot 0,0061 - 1}{58,44} = 0,011 \quad (64)$$

E o ganho integrativo K_i :

$$K_i = \frac{\omega_n^2\tau}{K_0}, \quad (65)$$

$$K_i = \frac{160,64^2 \cdot 0,0061}{58,44} = 2,69 \quad (66)$$

Uma simulação do sistema de controle PI com os ganhos calculados é mostrada na Figura 73. Como pode ser observado, tanto a ultrapassagem percentual quanto o tempo de acomodação apresentados são próximos das especificações do projeto.

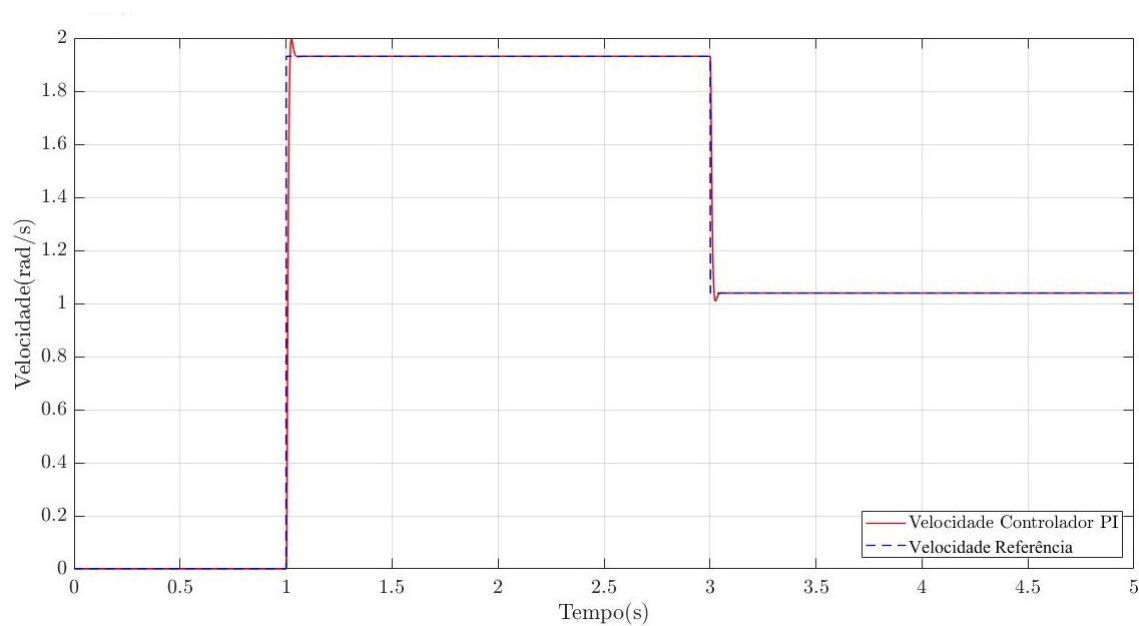


Figura 73: Simulação do controlador PI para entrada em degrau
Fonte: Próprio Autor

ANEXO C *DATASHEET* DO MOTOR MAXON RE 25 Ø25 MM, GRAPHITE BRUSHES, 20 WATT

VALUES AT NOMINAL VOLTAGE	
Nominal voltage	24 V
No load speed	9560 rpm
No load current	36.9 mA
Nominal speed	8330 rpm
Nominal torque (max. continuous torque)	26.3 mNm
Nominal current (max. continuous current)	1.16 A
Stall torque	243 mNm
Stall current	10.4 A
Max. efficiency	85 %
CHARACTERISTICS	
Terminal resistance	2.32 Ω
Terminal inductance	0.238 mH
Torque constant	23.4 mNm/A
Speed constant	408 rpm/V
Speed / torque gradient	40.3 rpm/mNm
Mechanical time constant	4.55 ms
Rotor inertia	10.8 gcm ²
THERMAL DATA	
Thermal resistance housing-ambient	14 K/W
Thermal resistance winding-housing	3.1 K/W
Thermal time constant winding	12.5 s
Thermal time constant motor	612 s
Ambient temperature	-30...+100 °C
Max. winding temperature	+125 °C
MECHANICAL DATA	
Bearing type	ball bearings
Max. speed	14000 rpm
Axial play	0.05 - 0.15 mm
Radial play	0.025 mm
Max. axial load (dynamic)	3.2 N
Max. force for press fits (static)	64 N
(static, shaft supported)	800 N
Max. radial load	16 N, 5 mm from flange

Figura 74: *Datasheet* Motor Maxon
Fonte: (MAXON, 2021)