

Centro Federal de Educação Tecnológica de Minas Gerais
Campus Divinópolis
Graduação em Engenharia Mecatrônica

Matheus Delgado de Azevedo

Implementação de rede neural artificial para
compensação de não-linearidade em sistema mecânico

Divinópolis
2019

Matheus Delgado de Azevedo

Implementação de rede neural artificial para
compensação de não-linearidade em sistema mecânico

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.
Áreas de integração: computação, controle, mecânica.

Orientador: Prof. Dr. Valter Júnior de Souza Leite

Divinópolis
2019

Matheus Delgado de Azevedo

Implementação de rede neural artificial para
compensação de não-linearidade em sistema mecânico

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.
Áreas de integração: computação, controle, mecânica.

Comissão Avaliadora:

Prof. Dr. Valter Júnior de Souza Leite
Departamento de Engenharia
Mecatrônica / CEFET-MG *Campus V*

Prof. Dr. Cláudio Parreira Lopes
Departamento de Engenharia
Mecatrônica / CEFET-MG *Campus V*

Romulo José da Silva Júnior
Engenheiro mecatrônico
Cibercode Educação e Tecnologia LTDA

Divinópolis
2019

A todos que, encarando problemas, engrandecem a si mesmos com soluções impensadas.

Agradecimentos

Agradeço,

- aos meus pais, Afonso e Simone, por proverem meus acessos a educação para a conclusão desse curso, estando sempre presentes, atentos e motivantes.
- à minha companheira e amiga de longa data, Marina, pela paciência, carinho e suporte dados durante essa etapa, enfrentando juntos a árdua jornada da vida adulta.
- ao Prof. Dr. Valter Leite pelas excepcionais aulas e, em especial, orientações, apontando direções importantes para a conclusão deste e de outros trabalhos ao longo desse curso.
- ao Prof. Dr. Cláudio Lopes pelas dicas e ensinamentos, especialmente durante as etapas de projeto e fabricação da planta.
- aos meus amigos da Turma 8 por dividirem as dores e alegrias trazidas nesses anos de graduação.

"Sabei que o segredo das artes é corrigir a natureza."

Voltaire

Resumo

A modelagem de sistemas dinâmicos baseada em funções de transferência é comumente aplicada no controle de sistemas mecânicos. Tal representação demanda condições de linearidade e invariância no tempo para funcionar propriamente. Em alguns casos, o real comportamento do sistema pode não ser descrito satisfatoriamente por sua representação linear, implicando em inadequações de requisitos previstos no projeto do controlador. Como forma de identificar e mitigar os efeitos das não-linearidades presentes na dinâmica desses sistemas, métodos de regressão são normalmente empregados, e redes neurais artificiais (RNAs) são ferramentas para aproximação de funções e suas inversas. Neste trabalho de conclusão de curso, foi desenvolvido um sistema mecânico controlado por métodos da Teoria de Controle Clássico, e implementada uma RNA para compensação de sua não-linearidade. O comportamento do sistema foi descrito por um modelo de dinâmica linear acrescido de uma não-linearidade. Sobre o primeiro modelo, foi projetado um controlador baseado no método de alocação de polos no domínio da frequência complexa (domínio de Laplace), enquanto a função inversa da não-linearidade foi aproximada pela RNA. Em simulações, foi mostrado um aumento na eficiência do controlador linear com o emprego do compensador (rede neural).

Palavras-chave: Controle de sistemas não-lineares. Redes Neurais Artificiais. Controle Robusto.

Abstract

The transfer function based modeling of dynamical systems is commonly applied to the control of mechanical systems. Such representation demands linearity and time-invariance conditions to work properly. In some cases, the real behavior of the system cannot be satisfactorily described by its linear representation, implying in inadequacies of controller's project requirements. As a way to identify and mitigate the non-linearities effects presents in the dynamic of these systems, numerical regression methods are normally used, and artificial neural networks (ANNs) are tools to approximate functions and its inverses. In this research, was developed a mechanical system controlled by methods of Classical Control Theory, and implemented an ANN for its non-linearity compensation. The system behavior was described by a linear dynamic submodel in series with a non-linearity. With the first submodel, was projected a controller based in pole allocation in complex frequency domain (Laplace domain), and the inverse function of non-linearity was approximated by the ANN. In simulations, was shown an increase of efficiency in the linear controller by using the compensator (neural network).

Key-words: Nonlinear Control Systems, Artificial Neural Networks, Robustness.

Sumário

Lista de Figuras	xii
Lista de Tabelas	xiii
Lista de Códigos-fonte	xiv
Lista de Acrônimos e Notação	xv
1 Introdução	1
1.1 Definição do problema	4
1.2 Motivação	4
1.3 Objetivos do trabalho	5
1.3.1 Objetivos Gerais	5
1.3.2 Objetivos Específicos	5
1.4 Organização do texto	5
2 Fundamentos	7
2.1 Revisão Bibliográfica	7
2.2 Estado da Arte	9
2.3 Fundamentação Teórica	10
2.3.1 Linearidade de Sistemas	10
2.3.2 Não-linearidades	11
2.3.3 Motor de corrente contínua	14
2.3.4 Modelo de redução mecânica	15
2.3.5 Características de malha fechada	17
2.3.6 Redes neurais artificiais	18
2.3.7 Treinamento da rede neural	21
2.3.8 Projeto e discretização de controladores	24
2.3.9 Índices de desempenho	27
3 Desenvolvimento	28
3.1 Materiais e métodos	28
3.2 Modelagem do sistema	29
3.2.1 Escolha da não-linearidade	29
3.2.2 Modelo do sistema	29
3.2.3 Análise do modelo	30

3.3	Projeto e discretização do controlador	31
3.3.1	Projeto do controlador	31
3.3.2	Discretização do controlador	31
3.4	Projeto da planta	32
3.4.1	Formulação do mecanismo	32
3.4.2	Análise de restrições	34
3.4.3	Esboço do sistema	34
3.4.4	Escolha de materiais e processo de fabricação	35
3.4.5	Fabricação dos componentes	36
3.5	Projeto da rede neural artificial	37
3.5.1	Treinamento da rede	38
3.5.2	Redução de ordem da RNA	39
3.6	Desenvolvimento do sistema de controle	41
3.6.1	Sensores e atuadores	42
3.6.2	Programação do sistema	45
3.7	Simulações e testes	46
3.8	Custos de projeto	47
4	Resultados e Discussões	48
4.1	Modelagem do sistema	48
4.1.1	Modelo do motor	48
4.1.2	Validação do modelo	48
4.1.3	Modelo da não-linearidade	50
4.1.4	Análise do sistema	50
4.1.5	Características de desempenho	50
4.2	Projeto de compensadores	51
4.2.1	Projeto do controlador	51
4.2.2	Discretização do controlador	51
4.3	Implementação de rede neural	52
4.3.1	Sinal de entrada	52
4.3.2	Desenvolvimento da RNA	52
4.4	Construção da planta	55
4.4.1	Fabricação e montagem da planta	55
4.4.2	Análise de Restrições	55
4.5	Simulações e análises	56
4.5.1	Parâmetros	56
4.5.2	Simulações	56
4.5.3	Testes	59
5	Considerações Finais	62
5.1	Conclusões	62
5.2	Objetivos alcançados	63
5.3	Propostas de continuidade	63
A	Apêndice	65
	Referências	81

Lista de Figuras

1.1	Sistema em malha fechada. Fonte: elaborado pelo autor.	1
1.2	Folgas f e g em sistema de engrenagens (perfil simplificado)	2
1.3	dimensões de uma engrenagem	3
2.1	Modelos de Wiener e Hammerstein	8
2.2	Representação de zona morta	12
2.3	Diagrama de blocos da não-linearidade de <i>backlash</i>	12
2.4	Lugar das raízes do exemplo 2.7	13
2.5	Circuito elétrico do motor de corrente contínua	14
2.6	Torques em um motor de corrente contínua	15
2.7	Dimensões de um par de engrenagens para cálculo do <i>backlash</i>	17
2.8	Sistema $G(s)$ realimentado	18
2.9	Diagrama de um <i>perceptron</i>	19
2.10	Diagrama da RNA do exemplo 2.4	21
2.11	Função de custo $J(y, \hat{y})$	23
2.12	Gradiente $\nabla J(2)$	24
2.13	Diagrama de blocos do sistema	25
2.14	LGR do exemplo 2.7	26
3.1	Diagrama de blocos do sistema em malha aberta	30
3.2	Modelo 3D do sistema	32
3.3	Forças exercidas na porca anti- <i>backlash</i>	33
3.4	Vista explodida do sistema	35
3.5	Diagrama de blocos para treinamento da rede neural	38
3.6	Efeitos do <i>overfitting</i> em uma função	39
3.7	Diagrama de blocos do compensador	41
3.8	Dimensões do motor CC. Fonte: (AKIYAMA, 2019)	43
3.9	Dimensões do Encoder. Fonte: (ROBIZ, 2019)	44
3.10	Captura de tela da interface gráfica do sistema	46
4.1	Validação do modelo	49
4.2	Validação do modelo (excluindo integrador)	49
4.3	Resposta do modelo em realimentação para entrada degrau	50
4.4	Saída do sistema contendo e excluindo RNA (versão 1, simulação)	53
4.5	Saída do sistema contendo e excluindo RNA (versão 2, simulação)	53
4.6	Comparação de saídas entre compensadores.	54

4.7	Sistema físico de transmissão mecânica com <i>backlash</i> ajustável	55
4.8	Respostas do sistema em Malha Fechada sem compensador	57
4.9	Respostas do sistema em Malha Fechada com primeira versão do compensador	58
4.10	Respostas do sistema em Malha Fechada com segunda versão do compensador	59
4.11	Respostas do sistema em Malha Fechada sem compensador	60
4.12	Respostas do sistema em Malha Fechada com segunda versão do compensador	61

Lista de Tabelas

2.1	Relações de dimensões de uma engrenagem	16
3.1	Parâmetros da engrenagem	37
3.2	Motor CC AK360/78.8PL12S7000S. Fonte: (AKIYAMA, 2019)	42
3.3	Motor C2162-60006/DN505728. Fonte: (TREASE, 2019)	43
3.4	Encoder LPD3806-400BM-G5-24C. Fonte:(ROBIZ, 2019)	44
3.5	<i>Raspberry Pi 3 Model B</i> . Fonte: (RASPBerryPI, 2019)	44
3.6	Módulo Ponte-H L298N. Fonte: (HANDSON, 2019)	45
3.7	Fonte 400W-ATX-PFC-R	45
3.8	Tabela de custos (orçamento).	47
4.1	Tabela de índices de desempenho do controlador em malha fechada	51
4.2	Configurações de folgas estimadas na planta	56
4.3	Índices de desempenho (caso 1, simulação)	57
4.4	Índices de desempenho (caso 2, simulação)	57
4.5	Índices de desempenho (caso 3, simulação)	58
4.6	Índices de desempenho (caso 1, teste)	60
4.7	Índices de desempenho (caso 2, teste)	60
A.1	Relação de materiais e processos de fabricação	67

Lista de Códigos-Fonte

1	Código de execução do controlador	65
2	<i>Forward Propagation</i> da RNA	66

Lista de Acrônimos e Notação

ABNT	Associação Brasileira de Normas Técnicas
ABS	Acrilonitrila butadieno estireno, do inglês <i>Acrylonitrile Butadiene Styrene</i>
CAD	<i>Computer-aided design</i>
CC	Corrente Contínua
GPIO	Entrada-saída de propósito geral, do inglês, <i>General Purpose Input-Output</i>
GUI	Interface gráfica do usuário, em inglês, <i>graphical user interface</i>
IAE	<i>Integral absolute error</i>
ISE	<i>Integral square error</i>
ITAE	<i>Integral time absolute error</i>
LGR	Lugar Geométrico das Raízes
LSTM	Memória de curto e longo termo, do inglês <i>Long-Short Term Memory</i>
LTi	Linear e invariante no tempo, do inglês <i>Linear Time-invariant</i>
MA	Malha Aberta
MF	Malha Fechada
MIMO	múltiplas entradas, múltiplas saídas, do inglês <i>Multiple Inputs, Multiple Outputs</i>
NL	Não-linear ou Não-linearidade
PPR	<i>Pulses per Revolution</i> (Pulsos por Revolução)
PReLU	<i>Parametric Rectified Linear Unit</i>
RF	Resposta em Frequência
RNA	Rede Neural Artificial
RNN	Rede Neural Recorrente, do inglês <i>Recurrent Neural Network</i>
RPI	Raspberry Pi
SD	<i>Secure digital</i>

$x \rightarrow y$	Entrada x produz uma saída y
$x \approx y$	x é aproximadamente igual a y
$x \ll y$	x é muito menor que y
$A * b$	convolução da resposta impulsiva do sistema A com o sinal b
\mathbb{R}^n	conjunto dos vetores $n \times 1$ com elementos reais
h_a	altura de cabeça dos dentes de uma engrenagem [mm]
h_f	altura do corpo dos dentes de uma engrenagem [mm]
m	módulo de uma engrenagem
N	fator de redução em um par de engrenagens
p_c	passo circular [mm]
z	número de dentes em uma engrenagem.

Introdução

A análise de sistemas dinâmicos permitiu a melhor compreensão do comportamento da natureza para a humanidade. Diferentes leis físicas, descobertas por cientistas ao longo dos séculos, tornaram possível descrever, através de equações, a ação de fenômenos elétricos, mecânicos, térmicos, etc. As leis de Ohm e Kirchhoff, além das equações de Euler-Lagrange, por exemplo, apresentam modelos para o comportamento de sistemas elétricos e mecânicos (DORF; BISHOP, 2010). O entendimento da dinâmica de sistemas é também importante para a engenharia, uma vez que permite descrições do comportamento de elementos de projeto, aumentando o tempo de vida e desempenho requerido. Com os modelos, podem ser obtidos tanto a evolução da resposta do objeto estudado em função do tempo quanto suas condições estacionárias de funcionamento. Os regimes discutidos são denominados transitório e permanente, respectivamente, e são determinados através de equações diferenciais, que relacionam uma função com suas derivadas de diferentes ordens (OGATA, 2009).

Quando o sistema em malha aberta (configuração dita de componentes sem sensoriamento de suas propriedades) não atinge características de regime transitório e/ou permanente desejadas, são implementados controladores em malha fechada. Os compensadores possuem dinâmica projetada para adequar, através da diferença entre a condição desejada e o estado atual do sistema (obtidas através de sensores), em um sistema retroalimentado (DORF; BISHOP, 2010).

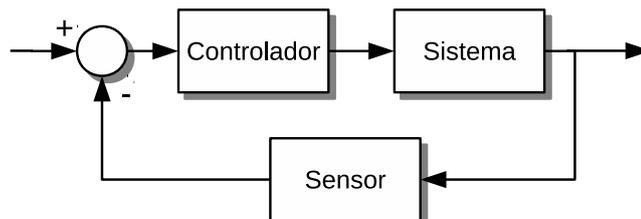


Figura 1.1: Sistema em malha fechada. Fonte: elaborado pelo autor.¹

¹Doravante, evitando repetições, subentende-se que recursos não referenciados são de própria autoria.

Os parâmetros esperados do projeto do compensador, para atingir as especificações requeridas, dependem da análise da dinâmica do sistema a ser estudado, o que requer a solução das equações diferenciais do modelo. Por vezes, encontrar a resposta das EDOs é somente analiticamente possível com o auxílio da Transformada de Laplace, que tem como requerimento a linearidade e invariância no tempo do sistema estudado. Contudo, a linearização da EDO sobre um ponto de operação, através da Série de Taylor, é capaz de gerar resultados suficientemente representativos em uma restrita faixa de operação (DORF; BISHOP, 2010). Desta forma, podemos projetar controladores para diferentes sistemas, como elétricos e mecânicos.

Sistemas mecânicos, também referidos como mecanismos, são definidos por NORTON (2008) como união de elementos organizados de forma a transmitir movimentos de uma maneira determinada. São usados sempre que necessária a transmissão de potência mecânica em variadas formas, e empregados em sistemas industriais, automobilísticos, robóticos, de transmissão de potência, içamento, etc. Cada elemento de máquina, componente fundamental de um mecanismo, é projetado de forma a suportar as solicitações mecânicas do ambiente inserido, e, por este motivo, a análise do comportamento da peça e da interação entre muitas é de grande importância. Como discutido, as ferramentas matemáticas (modelagem por equações diferenciais, Transformada de Laplace e Série de Taylor) fornecem representações linearizadas das características do sistema em torno de um ponto de operação. Entretanto, no estudo de mecanismos, é sabido que efeitos construtivos e de desgaste das peças provocam não-linearidades, influenciando nas respostas transitórias e permanentes do sistema estudado. Exemplos de não linearidades são folgas entre dentes de engrenagens (FLORES; GOMES, 2014), limitações dimensionais de peças e características de materiais em regime plástico.

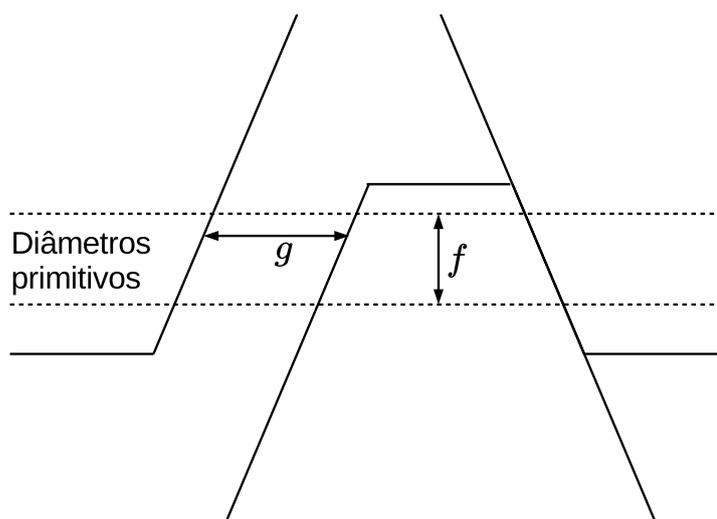


Figura 1.2: Folgas f e g em sistema de engrenagens (perfil simplificado)

Dos diferentes tipos de mecanismos contendo não-linearidades, destacam-se os sistemas de redução mecânica, sendo um dos principais elementos de máquinas utilizados em sistemas

automotivos, de transporte de cargas e robóticos, segundo NORTON (2008). Em pares, o engrenamento permite a transmissão de potência entre diferentes elementos, orientados distintamente no espaço. O projeto de uma engrenagem deve levar em conta o número de dentes z necessário para a transmissão, o fator de redução entre velocidade e torque de entrada e saída N o passo circular p_c e a altura dos dentes, como soma da sua cabeça h_a e seu corpo h_f (FLORES; GOMES, 2014). Todos os fatores são dependentes de seu módulo, denotado por m . O modelo mais simples de transmissão por engrenagens leva em conta apenas o fator de redução N , assumindo que a engrenagem tenha rigidez infinita. A Figura 1.3 a seguir ilustra uma engrenagem contendo os parâmetros apresentados

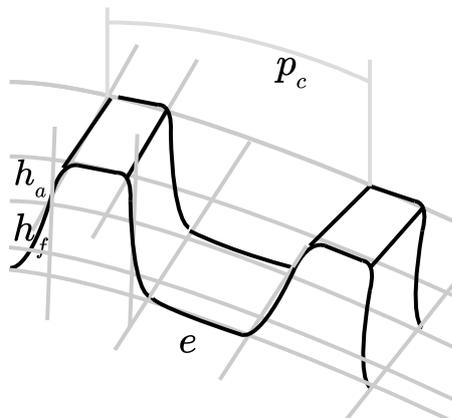


Figura 1.3: dimensões de uma engrenagem

A não-linearidade deste componente pode surgir através do desgaste do elemento, ampliando o efeito de folgas (mostradas na Figura 1.2 por f e g , também denominadas *backlashes*). Os efeitos citados, mesmo que inerentes e necessários para o funcionamento do sistema, são fontes de não-linearidades, prejudicando a compensação de sistemas. É importante ressaltar que todos os aspectos abordados nos exemplos anteriores são inevitáveis, e a pré-compensação não elimina o fenômeno, mas atenua os efeitos para o controlador linear projetado.

O problema da compensação de sistemas não lineares dependem de ferramentas diferentes das dispostas pelo projeto via função de transferência. Uma das estratégias de controle e pré-compensação baseia-se em algoritmos de regressão, como, por exemplo o método dos mínimos quadrados (DABIS; MOIR, 2007). Outras ferramentas computacionais de regressão são denominadas bio-inspiradas, citando como exemplo algoritmos genéticos e redes neurais artificiais. Esta última possui características de regressão de funções universais, a depender de sua função de ativação. Segundo (NAVARRO, 2012), em contraste com o método dos mínimos quadrados, as RNAs são capazes de prever o comportamento do sistema "com um passo a frente sem erros significativos". Outra limitação do método dos mínimos quadrados é de que a função a ser encontrada precisa ser linear (BERKLEY, 2019).

1.1 Definição do problema

Sabendo-se das características de regressão das RNAs, este trabalho pretende empregar uma rede neural para a compensação de uma não-linearidade mecânica, encontrada em um sistema mecânico a ser definido pelo autor. Deseja-se obter e comparar as respostas dos sistemas simulados e experimentais em malha fechada para um mesmo controlador linear visando atender características fixas de regimes transitório e permanente, em um primeiro caso ignorando as não-linearidades obtidas e, por último, empregando a RNA para a compensação.

O projeto do mecanismo, dimensionamento do motor, acompanhamento dos processos de fabricação e montagem das peças envolvem conhecimentos da área de mecânica, especificamente do eixo de Estruturas e Dinâmicas do curso. A implementação da rede neural e seleção de algoritmos de otimização para seu treinamento em sistemas embarcados, bem como a escolha de requisitos de *hardware* suficientes para a tarefa demandam conhecimentos em computação, com conteúdos mistos abordados nos eixos quatro, cinco e seis da grade curricular do curso. Para a análise do sistema, especificação de comportamento em malha fechada, projeto de controladores, discretização de sistemas e levantamento de parâmetros de desempenho é utilizado o oitavo eixo da grade curricular do curso, caracterizando a área de Controle.

1.2 Motivação

A união de algoritmos de otimização e o controle de sistemas já é utilizado em ambientes acadêmicos e industriais, tendo o método de regulação pelos Mínimos Quadrados alto reconhecimento. Em termos mercadológicos, o emprego do controle otimizado busca a melhor adequação de características transitórias e permanentes do sistema, buscando minimizar o valor do sinal de controle pelo tempo. Um grande volume de artigos relacionados desenvolvem a área, mas envolvem conceitos além dos propostos pelo ensino superior dos cursos de engenharia que contenham disciplinas relacionadas a controle de processos. Espera-se que este trabalho possa contribuir com o desenvolvimento de artigos futuros, trazendo técnicas conhecidas por estudantes de graduação.

O desenvolvimento e controle de sistemas físicos é uma ferramenta educacional aplicada em diversas universidades com cursos contendo disciplinas de controle de processos. É possível a utilização de *kits* didáticos, projetados de forma a minimizar as não-linearidades do sistema, facilitando o controle e o processo de aprendizado. Contudo, em outras instituições, é recomendado que o estudante construa seu próprio projeto. A baixa qualidade dos materiais do protótipo (vinda, normalmente, de limitações financeiras), indisponibilidade de sensores de precisão e inabilidade na construção manufaturada de projetos podem causar ao sistemas inúmeras não-linearidades, prejudicando o efetivo controle da planta. Pretende-se mostrar

com este trabalho que os efeitos não-lineares podem ser minimizados com o emprego de um compensador baseado em redes neurais artificiais.

1.3 Objetivos do trabalho

1.3.1 Objetivos Gerais

- Desenvolver uma rede neural para a pré-compensação de não linearidades de um sistema mecânico em malha fechada com controladores clássicos.
- Analisar, através de resultados obtidos por simulações e pela planta, a contribuição da RNA para o controle do sistema.

1.3.2 Objetivos Específicos

- Desenvolver, em ambiente simulado, um sistema mecânico, ressaltando uma de suas não-linearidades intrínsecas.
- Implementar e treinar uma rede neural para pré-compensação da não-linearidade escolhida.
- Projetar um controlador para o sistema mencionado, explorando a não-linearidade abordada.
- Reproduzir o sistema simulado com componentes físicos (construção da planta).
- Comparar desempenho da planta para o mesmo controlador com e sem o emprego da rede neural.

1.4 Organização do texto

Neste primeiro capítulo foi apresentado o problema a ser tratado, mostrando a relação entre sistemas mecânicos, controle de processos, não-linearidades, métodos de regressão e redes neurais artificiais. Foi trazida uma definição do problema a ser tratado, indicando também as áreas de integração do projeto. Por fim, foram levantadas as motivações para a realização do trabalho de conclusão de curso.

No capítulo 2 são apresentadas as fundamentações para a realização do trabalho. É apresentada uma revisão bibliográfica contendo os métodos utilizados ao longo dos anos para o controle de sistemas mecânicos, seguida do atual cenário de estratégias de tratamento dos efeitos não-lineares presentes nos sistemas. Por fim, são apresentados os conceitos teóricos fundamentais para o desenvolvimento deste trabalho de conclusão de curso.

O terceiro capítulo apresenta a metodologia empregada durante o desenvolvimento do trabalho. São descritas as etapas a serem tomadas para a construção do projeto, bem como apresentados os materiais e programas necessários para a tarefa. Também são descritos os custos para a continuidade do projeto.

Os resultados e discussões obtidos nesta etapa de elaboração do Trabalho de Conclusão de Curso são apresentados no capítulo 4. São descritas as atividades exercidas durante o semestre, bem como uma análise dos resultados obtidos.

Por fim, o quinto capítulo desse relatório apresenta um resumo dos objetivos alcançados durante o desenvolvimento do Trabalho de Conclusão de Curso, descreve planos e alternativas de elaboração de proposta de continuidades e apresenta considerações finais sobre o trabalho de conclusão de curso.

Fundamentos

Neste capítulo é descrito o histórico de estratégias utilizadas para o controle de processos mecânicos contendo não-linearidades intrínsecas, culminando na apresentações de abordagens atualmente utilizadas para o tratamento dos efeitos. Em seguida, são apresentados conceitos teóricos necessários para o desenvolvimento deste trabalho.

2.1 Revisão Bibliográfica

O controle de sistemas mecânicos é um processo amplamente aplicado em diferentes ambientes, como discutido no texto presente no primeiro capítulo deste trabalho. Nesse aspecto, a implementação de compensadores lineares baseados em função de transferência são normalmente tratados pelo projeto de controladores proporcionais, com termos integradores ou derivativos (nomeados, respectivamente, controladores P, PI e PID) (ASTROM; HAGGLUND, 1995), sendo ainda atualmente aplicados, por exemplo, em ambientes industriais. Devido ao longo período de estudos e testes, métodos práticos e teóricos para escolha dos coeficientes dos compensadores foram desenvolvidos (DWYER, 2006).

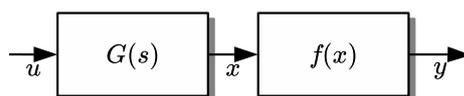
Um método de adequação de características de regimes transitórios (como máximo sobressinal e tempo de acomodação) e permanentes (erros de estado estacionário para entradas em degrau ou rampa) em sistemas em malha fechada analisados pela função de transferência em MA (malha aberta) é o de alocação de polos (OGATA, 2009). Nele, para um sistema linear modelado por $G(s)$, é estabelecido um número finito de polos e zeros de um controlador $C(s) = \frac{\prod(s+z_i)}{\prod(s+p_j)}$, com $i = 1, 2, \dots, n$ e $j = 1, 2, \dots, m$, $n \leq m$, de forma a adequar os critérios para a malha realimentada (DORF; BISHOP, 2010). Neste sentido, segundo (OGATA, 2009), uma das primeiras técnicas de projeto de compensadores desenvolvida é a baseada na resposta em frequência (RF), uma vez que apresenta robustez de projeto, devido às margens de fase e ganho asseguradas. Em outras palavras, se os erros de modelagem de $G(s)$ não são significativos, o controle mitiga seus efeitos no projeto de controladores via RF. Apesar de ser um dos primeiros desenvolvidos (com trabalhos datando da década de 1930), trabalhos rela-

tivamente recentes como o de (WANG, 2003) investigam melhoramentos da técnica. Outra abordagem consiste na análise μ , que utiliza procedimentos de otimização para o projeto de um controlador de ordem múltipla da planta, capaz de assegurar uma formatação específica (*shapping*) da resposta em frequência (MATHWORKS, 2019a).

O método de projeto de compensadores discutidos no parágrafo anterior se baseia na Transformada de Laplace para a modelagem do sistema, para a análise de desempenho em MA e para a alocação de polos e zeros de $C(s)$, o que, pela definição da ferramenta matemática, trabalha no domínio da frequência complexa $s = \sigma + j\omega$ (DORF; BISHOP, 2010). Técnicas como as referidas anteriormente são embasadas pela teoria de controle clássico. Em outra vertente, as equações diferenciais que regem o sistema estudado podem ser representadas por um espaço de estados, e a análise do mesmo pode ser feita, neste caso, no domínio do tempo. A teoria de controle moderno trata da análise de controlabilidade e observabilidade do sistema, além do estudo de técnicas de projeto de controladores baseado na realimentação de cada um de seus estados. Embora a área permita o projeto de controladores capazes de tratar não-linearidades, publicações tidas como referência para a graduação, por exemplo as de OGATA (2009) e DORF; BISHOP (2010), linearizam o sistema em torno de um ponto de operação e, usualmente, baseiam o projeto do controlador pelo método de alocação de polos.

Contudo, conforme discutido nos tópicos introdutórios deste trabalho, sistemas mecânicos são fontes de não-linearidades intrínsecas, não tratáveis por controladores baseados pelos métodos exemplificados nos parágrafos anteriores. Desta forma, considerando as NLs (Não linearidades), modelos não-lineares como os de Wiener e Hammerstein são usados (WILLS *et al.*, 2013). Nesses paradigmas, a NL é tratada como uma característica estática f , enquanto a dinâmica do sistema pode ser separada como parte linear G . Os modelos se diferem na ordenação das características lineares e não-lineares: no modelo de Wiener, a dinâmica do sistema é tratada posteriormente à NL. No modelo de Hammerstein, a não linearidade precede a dinâmica. Um terceiro modelo combina os anteriores (MATHWORKS, 2019b). Desta forma, as não-linearidades são aproximadas, por exemplo, como polinômios de grau arbitrário.

Modelo de Wiener:



Modelo de Hammerstein:

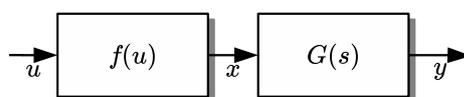


Figura 2.1: Modelos de Wiener e Hammerstein

Para encontrar os polinômios das NLs, as ferramentas de regressão são comumente utilizadas, sendo o método dos mínimos quadrados amplamente utilizado (NAVARRO, 2012). Entretanto, por se tratar de um método de regressão linear, possui limitações próprias na aproximação de funções. Segundo HAYKIN (2005), o método linear de otimização dos mínimos quadrados é usado para o ajuste de pesos de um neurônio de uma rede neural (como visto a seguir na seção de fundamentação teórica), enquanto a RNA é composta de diversos neurônios, oferecendo combinações não permitidas pelo método dos mínimos quadrados. Assim, as redes neurais podem ser empregadas, como visto em (SELMIC; LEWIS, 2000), (GAO; SELMIC, 2005), (NA *et al.*, 2014) e (CHEN; GE; HOW, 2010). Devido a sua capacidade de aproximativa generalizada, provada pelo teorema da aproximação universal de CYBENKO (1989), diversos problemas de não-linearidades localmente descontínuas em sistemas dinâmicos puderam ser tratadas. A exemplo, os artigos de LAM; PUTMAN; LORENZ (1995), SELMIC; LEWIS (2002), SELMIC; LEWIS (2000) e GAO; SELMIC (2005) tratam de modelos de histerese, atrito, zona morta e combinações de efeitos, respectivamente.

2.2 Estado da Arte

Neste tópico, são discutidas algumas abordagens recentes que tratam da compensação de sistemas não-lineares, utilizando redes neurais artificiais para a compensação de não-linearidades em um sistema mecânico. Como descrito a seguir, as RNAs são empregadas para efetivo controle da planta.

Partindo das propriedades de aprendizado e aproximação de funções em redes neurais, a ferramenta é usada para o fim de pré-compensação de funções descontínuas, não-simétricas e, de forma geral, de difícil modelagem. Como exemplos, as publicações de NA *et al.* (2014), HUANG; TAN (2012), HE *et al.* (2015) e HE; DONG; SUN (2015) se enquadram como empregadores de RNAs (Redes Neurais Artificiais). Nos dois primeiros casos, um controlador adaptativo é aplicado em um servomecanismo, tendo o atrito entre peças considerado, e compensando o efeito com uma RNA. Segundo NA *et al.* (2014), apenas uma RNA com pesos escalares foi capaz de aproximar não-linearidades desconhecidas e diminuir os custos computacionais. (HE; DONG; SUN, 2015) descreve a junção de um controlador adaptativo para um manipulador robótico de n links, com restrições de saturação na entrada. As redes neurais artificiais foram usadas para aproximar as não-linearidades, com o intuito de reduzir os efeitos de saturação. HE *et al.* (2015) tratam, com o emprego de uma rede neural artificial, os efeitos causados por uma zona morta presente em um atuador robótico de n links, ao passo que pré-compensa restrições de saída do manipulador. Segundo os autores, as restrições não foram desrespeitadas, ainda mantendo a estabilidade global do sistema.

O emprego de redes neurais em sistemas não-lineares também é usado não apenas para

pré-compensação, mas para o controle em malha fechada. Neste aspecto, diversos trabalhos podem ser encontrados. Destacam-se os trabalhos de CHEN; GE; HOW (2010) e HE; CHEN; YIN (2016). O primeiro artigo trata-se de um controle adaptativo por redes neurais em sistemas MIMO (*Multiple Inputs, Multiple Outputs*) não-lineares com incertezas de entrada e o segundo foca no controle adaptativo por RNAs em um sistema robótico de n links com restrições de entrada e saída. No primeiro artigo, é mostrada a estabilidade do sistema de controle, apontando que a rede contribui significativamente para a compensação do sistema. O último trabalho conclui que, dadas as restrições de entrada e saída do sistema, a rede neural funciona como controlador adaptativo, desempenhando melhora na compensação do sistema.

A técnica de mitigação de efeitos de não-linearidades ou controle completo de sistemas da classe é também abordada nas técnicas de controle robusto, nas quais as não-linearidades são tratadas como imprecisões e perturbações que devem ser tratadas pelo controlador, com publicações exemplificadas por (LI; JIA, 2016), (LI; CHENG; YAO, 2019) e (YAO; JIAO; MA, 2014).

2.3 Fundamentação Teórica

Esta seção descreve os principais embasamentos teóricos das ferramentas empregadas para o desenvolvimento deste trabalho de conclusão de curso. Orientado pela apresentação do problema principal tratado neste trabalho, as definições, propriedades e condições matemáticas dos principais elementos usados no desenvolvimento deste trabalho são apresentados.

2.3.1 Linearidade de Sistemas

Segundo LATHI (2007), um sistema é dito linear se, para quaisquer sinais de entrada $x_1(t)$ e $x_2(t)$ com suas respectivas saídas $y_1(t)$ e $y_2(t)$, e quaisquer constantes reais k_1 e k_2 :

$$k_1x_1(t) + k_2x_2(t) \rightarrow k_1y_1(t) + k_2y_2(t) \quad (2.1)$$

Em que o operador $a \rightarrow b$ é lido como "a produz a saída b". Para ilustrar a propriedade discutida considere o sistema modelado pelo exemplo a seguir:

Exemplo 2.1 *Um sistema descrito pela equação diferencial $\tau\ddot{y}(t) + \dot{y}(t) = Kx(t)$ é linear.*

Solução: *Se $y = k_1y_1 + k_2y_2$, então $\dot{y} = k_1\dot{y}_1 + k_2\dot{y}_2$ e $\ddot{y} = k_1\ddot{y}_1 + k_2\ddot{y}_2$. Assim, $\tau\ddot{y} + \dot{y} = \tau k_1\ddot{y}_1 + k_1\dot{y}_1 + \tau k_2\ddot{y}_2 + k_2\dot{y}_2 = Kx$. Rearranjando os termos, e retomando a equação inicial, $Kx = k_1(\tau\ddot{y}_1 + \dot{y}_1) + k_2(\tau\ddot{y}_2 + \dot{y}_2)$. Substituindo os termos e dividindo os lados por K , temos $x = k_1x_1 + k_2x_2$, demonstrando a linearidade da equação diferencial.*

Se os parâmetros da equação diferencial do sistema não variam com o tempo, então uma entrada deslocada de T segundos, denotada por $x(t + T)$ produz uma saída $y(t + T)$,

também atrasada de T segundos. Essa propriedade define um sistema invariante no tempo. (LATHI, 2007) Acerca de um sistema linear e invariante no tempo, como o do exemplo anterior, podemos encontrar a relação entre saída e entrada do sistema, desconsiderando suas condições iniciais, através de sua função de transferência, pela aplicação da Transformada de Laplace. No exemplo anterior, podemos encontrar a relação de saída e entrada no domínio da frequência complexa como $\frac{Y(s)}{X(s)} = \frac{K}{s(\tau s+1)}$. A aplicação da transformada é de fundamental importância para a análise de estabilidade de sistemas e projeto de controladores pelo modo clássico, discutido neste trabalho.

2.3.2 Não-linearidades

Embora sistemas lineares se mostrem presentes no mundo real (a exemplo de circuitos elétricos de elementos passivos como resistores, capacitores e indutores), efeitos intrínsecos da construção ou comportamento de diferentes elementos podem desrespeitar a condição proposta pela Equação (2.1). Como anteriormente citado, sistemas mecânicos são fontes de diversas não-linearidades. A exemplo, é mencionado o *backlash*, ou folga entre os dentes de uma engrenagem em um sistema de redução mecânica.

A folga entre os dentes interrompe o engrenamento, fazendo com que o movimento do pinhão não seja transmitido à coroa de forma imediata. Entretanto, a depender do perfil dos dentes dos elementos, uma vez que o contato ocorre, a movimentação existe até que haja uma reversão no sentido de giro (NORTON, 2008). Se o *backlash* for suficientemente grande, de forma a descumprir a continuidade do engrenamento, variações de posição com reversão de movimento no pinhão podem acarretar em anulação da respectiva rotação da coroa, fazendo com que o sistema apresente uma não-linearidade do tipo zona morta, sendo equacionado por (2.2):

$$D_z(u) = \begin{cases} m_1(u + \alpha), & u < -\alpha \\ m_2(u - \alpha), & u > \alpha \\ 0, & |u| \leq \alpha \end{cases} \quad (2.2)$$

Em que $\alpha > 0$ é a largura da zona morta, dependente do *backlash* presente na engrenagem e (m_1, m_2) são constantes reais positivas. O modelo é mencionado em publicações como a de BRUMS *et al.* (2006). Uma representação gráfica da função $D_z(u)$ pode ser vista na Figura 2.2 a seguir.

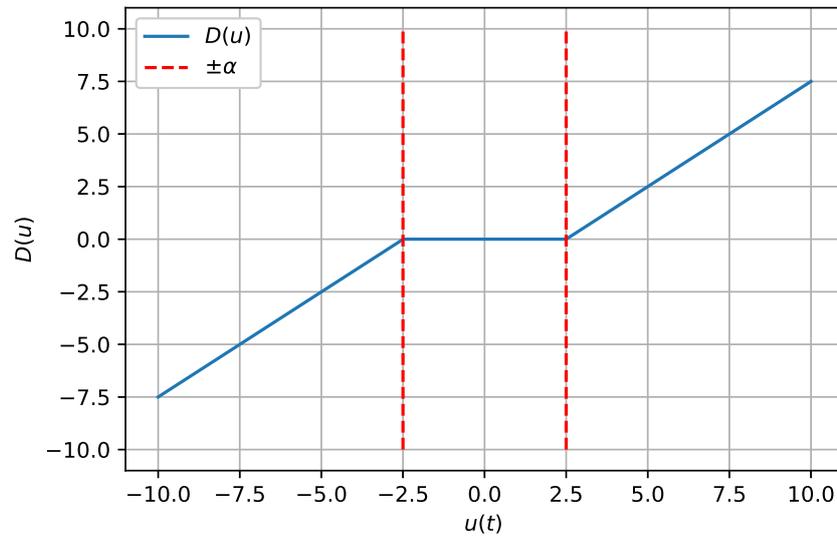
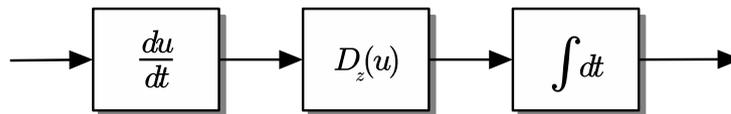


Figura 2.2: Representação de zona morta

Para a estimação da posição de um sistema mecânico contendo a não-linearidade, o *backlash* $B(u)$ pode ser obtido derivando a saída linear do atuador, aplicando este sinal em $D_z(u)$ e integrando para encontrar a posição real, como ilustrado pela Figura 2.3 (BRUMS *et al.*, 2006)

Figura 2.3: Diagrama de blocos da não-linearidade de *backlash*

A descontinuidade na derivada de $D_z(u)$ faz intuitiva a noção de que um sistema contendo folgas é não-linear, o que pode ser ilustrado pelo seguinte exemplo:

Exemplo 2.2 Seja o sistema representado por um backlash dado por

$$y = B(u) = \int D_z(\dot{u}) dt$$

No primeiro caso, investigamos $u = k_1 u_1 + k_2 u_2$. Tendo $\dot{u} = k_1 \dot{u}_1 + k_2 \dot{u}_2$ e usando a Equação (2.2), então

$$D_z(\dot{u}) = \begin{cases} m_1(k_1 \dot{u}_1 + k_2 \dot{u}_2 + \alpha), & u < -\alpha \\ m_2(k_1 \dot{u}_1 + k_2 \dot{u}_2 - \alpha), & u > \alpha \\ 0 & |u| \leq \alpha \end{cases}$$

e assim,

$$B(u) = \begin{cases} m_1(k_1u_1 + k_2u_2 + \alpha t), & u < -\alpha \\ m_2(k_1u_1 + k_2u_2 - \alpha t), & u > \alpha \\ c & |u| \leq \alpha \end{cases}$$

para $c \in \mathbb{R}$. Considerando $y = k_1y_1 + k_2y_2$, temos:

$$k_1y_1 + k_2y_2 = k_1 \begin{cases} m_1(u_1 + \alpha t), & u_1 < -\alpha \\ m_2(u_1 - \alpha t), & u_1 > \alpha \\ c_1 & |u_1| \leq \alpha \end{cases} + k_2 \begin{cases} m_1(u_2 + \alpha t), & u_2 < -\alpha \\ m_2(u_2 - \alpha t), & u_2 > \alpha \\ c_2 & |u_2| \leq \alpha \end{cases}$$

$$k_1y_1 + k_2y_2 = \begin{cases} m_1(k_1u_1 + k_2u_2) + m_1(k_1 + k_2)\alpha t, & u < -\alpha \\ m_2(k_1u_1 + k_2u_2) - m_2(k_1 + k_2)\alpha t, & u > \alpha \\ c_1 + c_2 & |u| \leq \alpha \end{cases}$$

Assumindo, por contradição, que o sistema é linear, então $k_1u_1 + k_2u_2 \rightarrow k_1y_1 + k_2y_2$, o que não se sustenta, uma vez que, fora do intervalo $|u| \leq \alpha$, $k_1 + k_2$ é necessariamente unitário. Os intervalos da zona morta também são alterados, uma vez que $|k_1u_1| + |k_2u_2| \geq |k_1u_1 + k_2u_2|$.

O problema também pode ser interpretado através do lugar geométrico das raízes, considerando que $B(u)$ fornece ganhos variáveis. Em malha fechada, com a mudança, a localização dos polos do sistema se altera com o tempo, causando inadequações aos critérios de projeto estabelecidos pelo projeto de controladores por função de transferência usando o método de alocação. O efeito ocorre uma vez que o projeto de controladores clássicos assume que os autovalores do sistema não variam com o tempo.

Exemplo 2.3 Um sistema de função de transferência $G(s) = \frac{K}{s(\tau s + 1)}$ como a do exemplo 2.1, com $K = 0.9$ e $\tau = 0.1$ realimentado produz um LGR conforme a Figura 2.4.

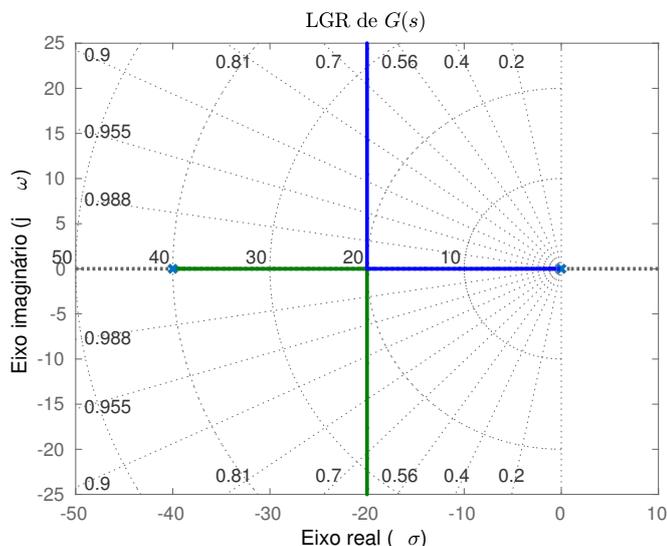


Figura 2.4: Lugar das raízes do exemplo 2.7

Quando aplicado um ganho variável K , os polos em malha fechada se deslocam seguindo a curva mostrada na figura anterior, alterando os tempos de acomodação e oscilação, definidos pelos valores real (eixo horizontal) e imaginário (vertical) dos polos, respectivamente.

2.3.3 Motor de corrente contínua

O comportamento dinâmico de um motor de corrente contínua pode ser determinado através da análise do sistema em suas características elétricas e mecânicas. Primeiramente, podemos considerar o circuito da Figura 2.5.

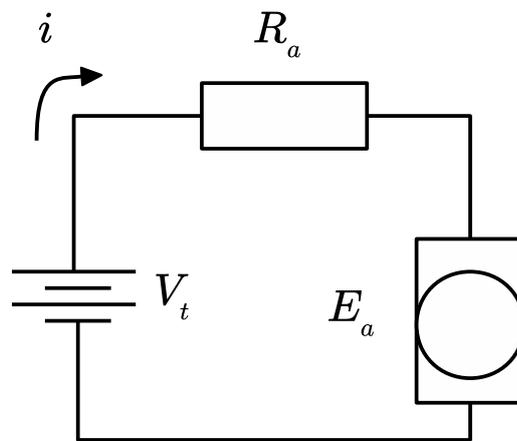


Figura 2.5: Circuito elétrico do motor de corrente contínua

Pelas leis de Kirchoff, a somatória de tensões na malha é nula, o que implica na dissipação de parte da potência pela resistência de armadura. O restante da tensão é convertida em velocidade angular $\dot{\theta}_m$ do motor, sendo proporcional à tensão eletromotriz (UMANS (2014)). A equação (2.3), que relaciona a velocidade do motor com a tensão nos terminais é dada por:

$$\begin{aligned} V_t &= R_a i + K_m \dot{\theta}_m, \\ \tau_m &= K_m i. \end{aligned} \quad (2.3)$$

Em relação ao componente mecânico, a segunda lei de Newton para rotações indica, pela Figura 2.6 que o torque provido pela fonte é convertido em aceleração angular $\ddot{\theta}_m$ dependente do momento de inércia J do rotor e em velocidade angular $\dot{\theta}_m$ proporcional ao coeficiente de atrito b . A equação (2.4) abaixo descreve essa relação:

$$\tau_m = J \ddot{\theta}_m + b \dot{\theta}_m. \quad (2.4)$$

Ignorando as condições iniciais, pelas Equações (2.3) e (2.4), temos as relações no domínio

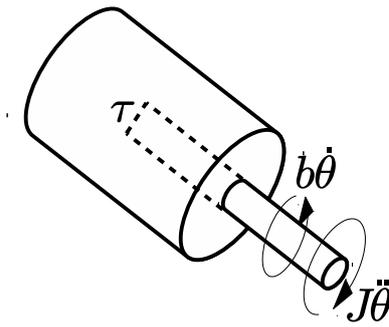


Figura 2.6: Torques em um motor de corrente contínua

da frequência:

$$\frac{I}{V} = R_a \quad \frac{\tau_m}{i} = K_m \quad \frac{\theta_m}{\tau_m} = \frac{1}{s(Js+b)} \quad (2.5)$$

Combinadas, é possível determinar a função de transferência de um motor de corrente contínua com entrada em tensão e saída em posição angular descrita pela equação (2.6) seguinte:

$$\frac{\theta_m}{V} = \frac{K_m/R_a}{s(Js + b + K_m/R_a)} \quad (2.6)$$

Além de determinar a função de transferência do atuador, o modelo do atuador nos fornece quais os parâmetros relevantes para a identificação do elemento. Assim, pelas Equações (2.3), (2.4) e (2.6), Os parâmetros a serem identificados por ensaios dispostos por UMANS (2014) são: Resistência de armadura (R_a), Constante elétrica do motor (K_m), Momento de inércia do rotor (J) e coeficiente de atrito (b).

Para a determinação do primeiro componente (R_a), com o motor estacionado, será medida a resistividade entre os terminais do atuador. Com os dados do *datasheet* referido em AKIYAMA (2019), são fornecidos os valores de tensão, corrente e rotação para um ensaio a vazio, sendo possível estimar o valor da constante K_m . Por fim, os valores de J e b podem ser determinados medindo a resposta temporal do sistema em malha aberta para uma entrada degrau. Os valores não são medidos de forma explícita, mas podem ser obtidos, por uma aproximação de um modelo de primeira ordem com integrador, os valores de ganho K e constante de tempo τ do motor.

2.3.4 Modelo de redução mecânica

Segundo NORTON (2008), reduções são mecanismos para transmissão de potências entre uma engrenagem motora (pinhão, denotada por índice 1) e movida (coroa, descrita por índice 2), pelo contato entre os elementos. No evento, as velocidades lineares v_1 e v_2 se equivalem.

Como as velocidades angulares ω_1 e ω_2 segue, por $v = \frac{d}{2}\omega$, a relação vista na equação (2.7):

$$N = \frac{d_1}{d_2} = \frac{z_1}{z_2} = \frac{\omega_2}{\omega_1} \quad (2.7)$$

em que N é o fator de redução do mecanismo, z_1, z_2 são os números de dentes do pinhão e da coroa, respectivamente, d_1, d_2 denotam os diâmetros dos mesmos elementos e ω_1, ω_2 suas respectivas velocidades angulares. A equação (2.7) mostra que a razão entre as velocidades angulares dos elementos, e conseqüentemente, as posições angulares, é determinada pelos diâmetros primitivos das engrenagens. O parâmetro pode ser determinado por (2.8):

$$d = mz \quad (2.8)$$

em que m é o módulo da engrenagem e z é o número de dentes. Pela equação (2.8), é possível perceber que $N = \frac{z_1}{z_2}$. FLORES; GOMES (2014) e SHIGLEY; MISHCKE (1989) determinam em suas publicações uma série de passos para o dimensionamento de uma engrenagem. As informações constam na Tabela 2.1.

Tabela 2.1: Relações de dimensões de uma engrenagem

Parâmetro	Equação
Diâmetro primitivo	$d = mz$
Altura da cabeça	$h_a = m$
Altura do corpo	$h_f = 1,25m$
Diâmetro de coroa	$d_a = m(z + 2)$
Diâmetro de raiz	$d_f = m(z - 2,5)$
Distância entre eixos	$a = m \frac{z_1 + z_2}{2}$

O modelo de um sistema de redução mecânica pode ser entendido de duas formas: tendo o comportamento ideal, conforme visto em (NORTON, 2008) e (FLORES; GOMES, 2014), e verificando detalhadamente os efeitos das dimensões da engrenagem. Para este tópico a segunda abordagem será tomada, utilizando a não-linearidade exemplificada pelo diagrama Figura 2.3.

Considerando o projeto de engrenagens proposto por SHIGLEY; MISHCKE (1989) e FLORES; GOMES (2014), as dimensões de espessura e espaçamento entre os dentes de uma engrenagem devem ter uma diferença denominada *backlash*, essencial para o engrenamento. A Figura 2.7 mostra, através da representação do contato entre duas engrenagens, dimensões relevantes para o cálculo do *backlash* (perfil dos dentes simplificados).

Segundo INDUSTRY (2019) e SHIGLEY; MISHCKE (1989), para um par de engrenagens com módulo m , número de dentes z e ângulo de pressão β e diâmetros primitivos afastados

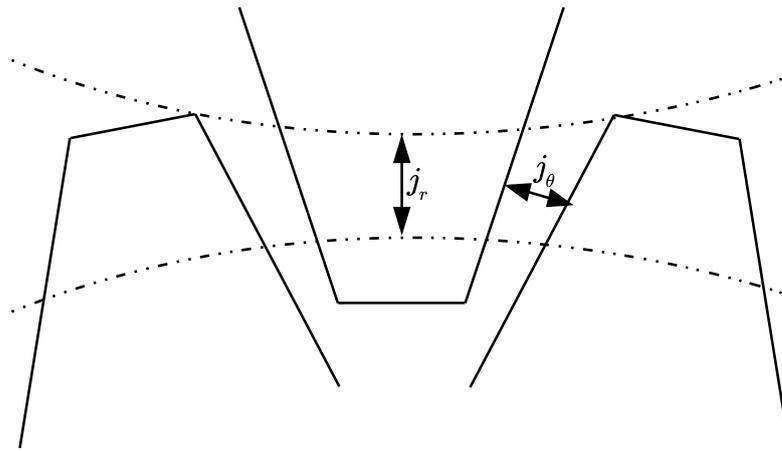


Figura 2.7: Dimensões de um par de engrenagens para cálculo do *backlash*

de uma distância j_r , denominada *backlash* radial, o ângulo j_θ máximo descrito pelo pinhão sem que a coroa seja movida é dado pela relação (2.9):

$$j_\theta = 360^\circ \frac{2}{\pi m z} \operatorname{tg}(\beta) j_r. \quad (2.9)$$

O valor de j_θ pode ser empregado para determinar a banda α da zona morta, como visto em (BRUMS *et al.*, 2006). Assumindo que a banda é simétrica, a Equação (2.2) tem como zona morta o intervalo $[-\alpha, \alpha]$. A função de *backlash* $B(u)$ é determinada por

$$B(u) = \int D_z(\dot{u}) dt, \quad (2.10)$$

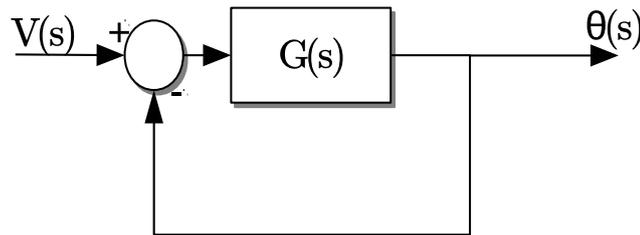
para $m_1 = m_2 = 1$, como ilustrado pelo Exemplo 2.2. O modelo depende do parâmetro α , que depende do *backlash* j_θ , também dependente de diversas variáveis. Assim, precisar analiticamente o valor da banda é uma tarefa árdua, necessitando de recursos computacionais para sua determinação de forma rápida.

2.3.5 Características de malha fechada

Considerando um sistema como a do Exemplo 2.1, com função de transferência de ganho K , único integrador e polo em $s = -a$, com $a = 1/\tau$, como mostra a Equação (2.11):

$$G(s) = \frac{K}{s(s+a)}, \quad (2.11)$$

Podemos determinar seu comportamento em termos de regime transitório e estacionário quando realimentado conforme a Figura 2.8 abaixo:

Figura 2.8: Sistema $G(s)$ realimentado

Para a Figura 2.8, a função de transferência em malha fechada é determinada pela equação 2.12:

$$G_f(s) = \frac{Ka}{s^2 + as + Ka} \quad (2.12)$$

sendo um sistema de segunda ordem, com $t_s = \frac{4}{\zeta\omega_n} = 8/a$ (apenas se $\zeta < 1$), $\omega_n = \sqrt{aK}$ e $\zeta = \frac{\sqrt{a}}{2\sqrt{K}}$. O erro de estado estacionário pode ser calculado pelo teorema do valor final, resultando em $e_{ss} = 0$, pelo efeito do integrador presente em $G(s)$. A depender dos valores de a e K , o sistema pode ter $\zeta \geq 1$, não apresentando sobressinal e aumentando o tempo de acomodação t_s .

A análise das Equações (2.6) e (2.12) são de fundamental importância para o projeto de controladores, uma vez que fornecem informações de sobressinal e tempo de acomodação em malha aberta e em realimentação negativa, respectivamente. A escolha de polos e zeros do controlador, segundo OGATA (2009), depende da escolha das características de regime transitório e permanente do sistema.

2.3.6 Redes neurais artificiais

Segundo GOODFELLOW; BENGIO; COURVILLE (2016), uma rede neural artificial é um modelo para aprendizado de máquina com o objetivo de aproximação de funções. Dado um conjunto de entrada $I \subseteq \mathbb{R}^i$ e de saída $O \subseteq \mathbb{R}^o$ mapeados por uma função F , uma rede neural é um modelo $F^*(x, \theta)$, em que $x \in I$ representa uma determinada entrada de i elementos, o é o número de saídas da rede e θ são os parâmetros internos da rede, aprendidos por diferentes métodos de otimização. Através de uma RNA são possíveis resolver diferentes classes de problemas, sendo mais utilizadas para:

- Classificação, se O tem valores discretos
- Regressão, se I e O são definidos um mapeamento contínuo.

CYBENKO (1989), em seu artigo, define o Teorema da Aproximação Universal, que apresenta a capacidade representativa de um mapeamento entre espaços do \mathbb{R}^N através de uma combinação linear de funções sigmoidais σ_k . Matematicamente, uma saída $y = F^*(x) \in O$ pode ser representada pela equação (2.13):

$$F^*(x) = \sum_k a_k \sigma_k (w_k^T x + \theta_k) \quad (2.13)$$

Os parâmetros a, θ, w são internos da rede e precisam ser aprendidos. Uma representação simplificada para o cálculo da Equação (2.13) é mostrada pela Figura 2.9. Ao modelo é dado o nome de *perceptron*, e seu funcionamento é inspirado no de um neurônio comum. No sistema biológico, os dendritos são carregados com diferentes concentrações de elementos químicos, em contato com diferentes neurônios. A corrente elétrica provida pelo neurônio anterior causa uma reação nos componentes presentes nos dendritos, sendo somada e propagada pelo axônio até suas extremidades, com diferentes elementos químicos, limitando a corrente em cada terminal (CHOLLET, 2018).

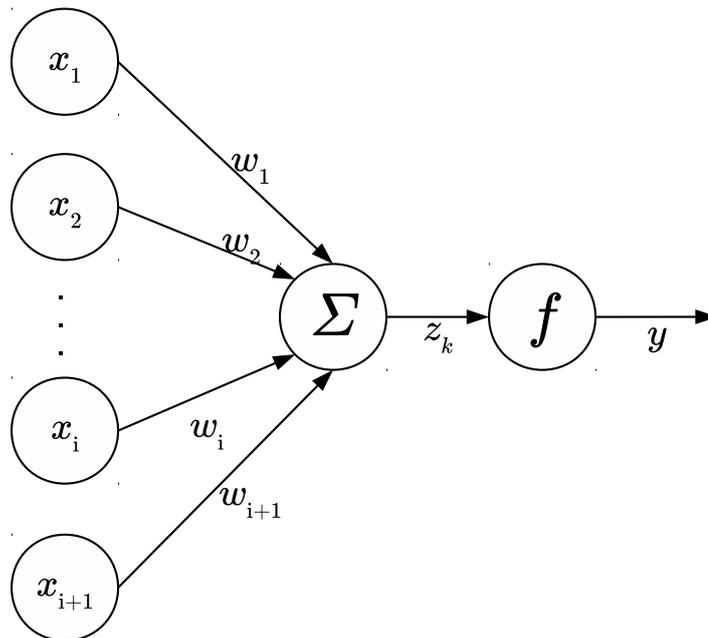


Figura 2.9: Diagrama de um *perceptron*

De forma semelhante ao neurônio biológico, o vetor de entrada $x \in I$ contendo i características é aumentado, produzindo uma entrada $X = [x, 1]$ e multiplicado por uma matriz de pesos W_k de dimensões $(i + 1, n_k)$. O resultado do produto $z_k = X_k W_k$, de dimensão $(1, n_k)$ é levado à uma função de ativação f_k (aplicada em cada elemento do vetor z_k), servindo como entrada para a camada seguinte. Podemos descrever a entrada de uma próxima camada de

uma rede neural (e, em sua última camada L , saída), pelo conjunto de equações (2.14):

$$z_k = X_k W_k \quad (2.14a)$$

$$x_{k+1} = f_k(z_k) \quad (2.14b)$$

$$y = x_L \quad (2.14c)$$

para $k = 0, \dots, L - 1$. Comparando as equações (2.13) e (2.14), percebe-se a ausência do termo θ_k na segunda. Isto se deve pela forma aumentada da entrada, que permite agrupar θ_k na matriz de pesos W_k . Um exemplo de cálculo exercido por uma rede neural é visto a seguir:

Exemplo 2.4 *A saída de uma rede de duas camadas, com três características de entrada e duas de saída, tendo $W_0 = \begin{bmatrix} 1 & 1 \\ 3 & 4 \\ 0 & 1 \\ 0.2 & 0.1 \end{bmatrix}$, $W_1 = \begin{bmatrix} 0.1 & 0.2 \\ 1.3 & 0 \\ 1 & 1 \end{bmatrix}$ e $f_0(z) = f_1(z) = e^{-z}$, a uma entrada $x = [0.2 \ 0.4 \ 1]$ equivale a $y = [0.2799 \ 0.2457]$.*

Solução: *Dados $i = 3$, $o = 2$ e $L = 2$, a matriz aumentada X_0 de entrada será:*

$$X_0 = [x \ 1] = [0.2 \ 0.4 \ 1 \ 1].$$

Assim,

$$z_0 = X_0 W_0 = [0.2 \ 0.4 \ 1 \ 1] \begin{bmatrix} 1 & 1 \\ 3 & 4 \\ 0 & 1 \\ 0.2 & 0.1 \end{bmatrix} = [1.6 \ 2.9].$$

Logo,

$$x_1 = f_0(z_0) = \begin{bmatrix} e^{-1.6} \\ e^{-2.9} \end{bmatrix}^T = \begin{bmatrix} 0.2019 \\ 0.0550 \end{bmatrix}^T.$$

Aumentando novamente a matriz de entrada para a camada $L = 1$,

$$X_1 = [x_1 \ 1] = [0.2019 \ 0.0550 \ 1].$$

Continuando,

$$z_1 = X_1 W_1 = \begin{bmatrix} 1.2734 \\ 1.4038 \end{bmatrix}^T$$

e

$$x_2 = f_1(z_1) = [0.2799 \ 0.2457]$$

. Como $L = 2$, $y = x_2$.

O Exemplo 2.4 descreve o processo de *forward propagation* (propagação para frente) de uma rede neural de $L = 2$ camadas. Denotando $x_l^{(k)}$ como a k -ésima característica da entrada da camada de número l , o diagrama da Figura 2.10 mostra a topologia da rede, bem como indica os cálculos realizados.

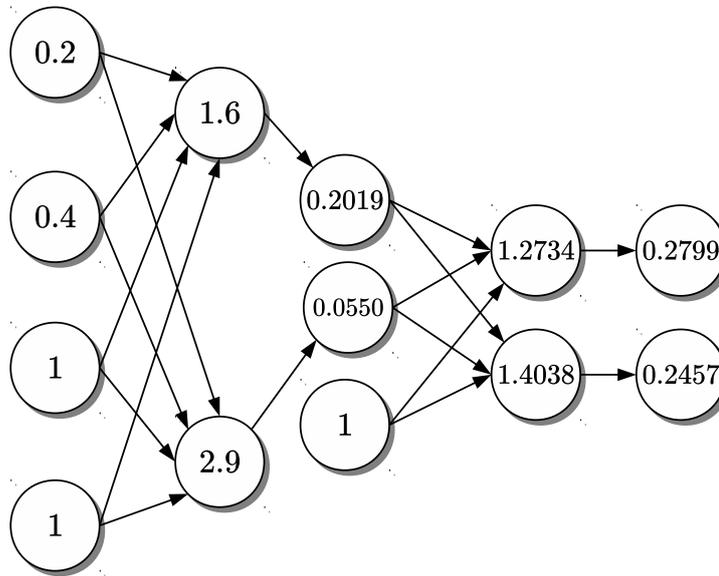


Figura 2.10: Diagrama da RNA do exemplo 2.4

Cada aresta ligada a um elemento de entrada corresponde a um elemento $w_l^{(k)}$ da matriz de pesos da respectiva camada (W_l). Analisando a Figura 2.10, é possível perceber que a primeira coluna de dados compõe o vetor de entrada aumentado X_0 . As arestas que ligam cada elemento $X_0^{(k)}$ representam a multiplicação entre cada elemento da matriz W_0 , gerando elementos das colunas seguintes, z_0 e $f_0(z_0) = x_1$, respectivamente. A matriz aumentada X_1 é vista na terceira coluna, sendo multiplicada por W_1 para a valoração de z_1 (coluna 4). Finalmente, como $L = 2$, a última coluna representa a saída da RNA y . Com o exemplo, é possível perceber que uma rede neural é uma função de entrada $x \in I$ e saída $y \in O$ dependente dos pesos W_k e das funções de ativação $f_k(z_k)$, para $k = 1, \dots, L - 1$.

2.3.7 Treinamento da rede neural

Como visto anteriormente, uma rede neural de L camadas é uma aproximação F^* de um mapeamento $F^* : I \rightarrow O$, dependente de matrizes de pesos W_k , com $k = 0, \dots, L - 1$. Para que a rede produza saídas \hat{y} satisfatoriamente próximas das esperadas (y), o conjunto de pesos internos deve ser ajustado por algum método de otimização, objetivando $\hat{y} \approx y$. Segundo GOODFELLOW; BENGIO; COURVILLE (2016), para problemas de classificação e regressão, diferentes algoritmos de otimização para treinamento dos pesos podem ser empregados. Contudo, em comum ter como objetivo a minimização de uma função de custo $J(y, \hat{y})$ (HAYKIN,

2005), também definida pelo problema a ser tratado pela rede. O exemplo a seguir demonstra uma possível função de custo para otimização

Exemplo 2.5 A função $J(y, \hat{y}) = \ln \cosh(y - \hat{y})$ tem (i) mínimo global em $y = \hat{y}$, (ii) comportamento modular ($|\xi|$) quando $\hat{y} \ll y$ e (iii) forma parabólica com concavidade positiva quando $\hat{y} \approx y$.

Para a propriedade (i), considere $\xi = y - \hat{y}$. A derivada de J , $\frac{dJ}{d\xi}$ é dada por

$$\frac{dJ}{d\xi} = \frac{d}{d\xi} \ln \cosh(\xi) = \tanh(\xi).$$

Para encontrar pontos críticos de uma função, iguala-se a derivada a zero. Portanto,

$$\frac{dJ}{d\xi} = 0 \Rightarrow \tanh(\xi) = 0$$

. Usando a identidade $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, temos um ponto crítico para $e^\xi = e^{-\xi}$. Aplicando \ln em ambos os lados da igualdade, $\xi = -\xi \Rightarrow \xi = 0$. Como $\ln \cosh 0 = 0$, o ponto crítico da função ocorre em $\hat{y} = y$. A determinação do ponto como mínimo decorre da propriedade (iii).

(ii): Considerando o caso $\hat{y} \ll y$, adotamos $\xi = y - \hat{y}$ e fazemos $\xi \rightarrow \infty$. Determinamos que J tem crescimento assintótico $O(J) = |\xi|$ se:

$$\lim_{\xi \rightarrow \infty} \frac{J(\xi)}{k|\xi|} = c$$

para $c, k \in \mathbb{R}$. A simples aplicação de $\xi \rightarrow \infty$ mostra que o limite tende a $\frac{\infty}{\infty}$, sendo possível a aplicação da regra de L'Hôpital. Logo,

$$\lim_{\xi \rightarrow \infty} \frac{J(\xi)}{k|\xi|} = \frac{1}{k} \lim_{\xi \rightarrow \infty} \frac{e^\xi - e^{-\xi}}{e^\xi + e^{-\xi}} = \lim_{\xi \rightarrow \infty} \frac{J(\xi)}{k|\xi|} = \frac{1}{k} \lim_{\xi \rightarrow \infty} \frac{e^{2\xi} - 1}{e^{2\xi} + 1}$$

Fazendo a substituição $u = e^{2\xi}$:

$$\lim_{\xi \rightarrow \infty} \frac{J(\xi)}{k|\xi|} = \frac{1}{k} \lim_{u \rightarrow \infty} \frac{u - 1}{u + 1} = \frac{1}{k}$$

Tomando $c = 1/k$, é visto que $O(J) = |\xi|$ quando $\hat{y} \ll y$. Quando $\xi \rightarrow 0$ (iii), podemos representar J como um polinômio de MacLaurin com grau $n = 2$. Assim,

$$J(\xi) \approx \frac{\ln \cosh(0)}{0!} (\xi - 0)^0 + \frac{\tanh(0)}{1!} (\xi - 0) + \frac{\text{sech}^2(0)}{2!} (\xi - 0)^2$$

Como o primeiro e segundo termo da linha anterior são nulos, $J(\xi) \approx \frac{\xi^2}{2}$, quando $\xi \rightarrow 0$, ou $\hat{y} \approx y$. A função $\frac{\xi^2}{2}$ é uma parábola de concavidade $a = 1/2 > 0$, determinando que o ponto crítico da origem é um ponto de mínimo. A Figura 2.11 mostra uma representação de $J(\xi)$.

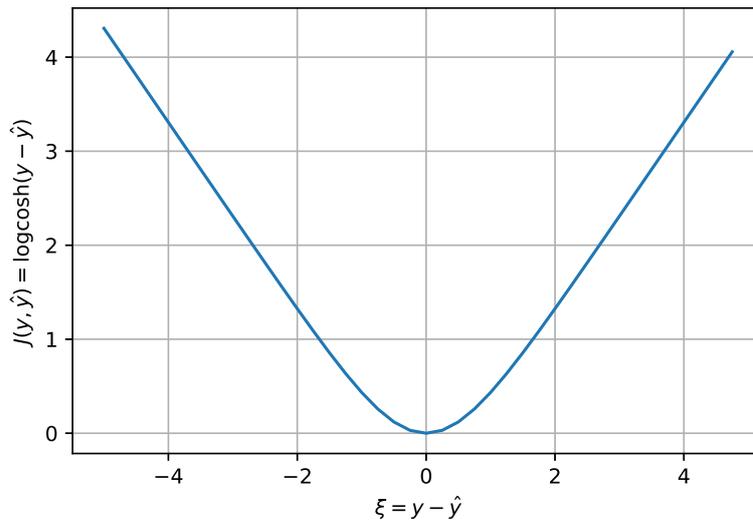


Figura 2.11: Função de custo $J(y, \hat{y})$

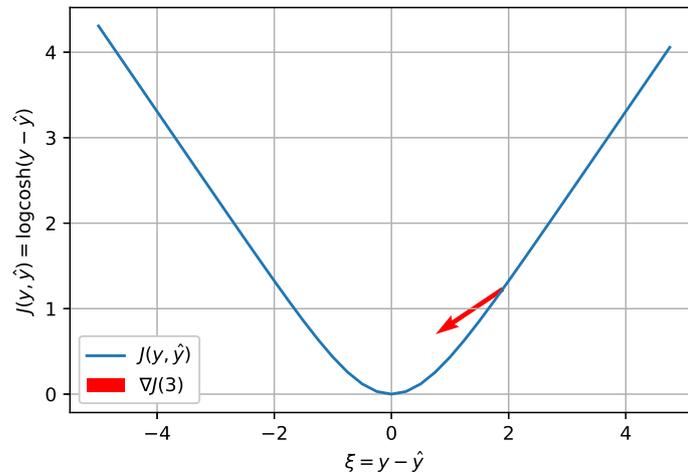
Com a função de custo J , é possível determinar qual a contribuição de cada peso W_l para o erro do sistema. Segundo HAYKIN (2005) e GOODFELLOW; BENGIO; COURVILLE (2016), o cálculo da contribuição δW_l , dada pela derivada $\frac{\partial J}{\partial W_l}$, é encontrada reescrevendo o termo \hat{y} da função de custo a partir dos pesos W_l . O cálculo desta etapa é referido como *backpropagation* ou retropropagação, em tradução livre. Pela regra da cadeia, a contribuição δW_l de uma matriz de pesos W_l é dada pela Equação (2.15):

$$\delta W_l = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_L} \frac{\partial z_L}{\partial z_{L-1}} \dots \frac{\partial z_l}{\partial W_l} \quad (2.15)$$

O cálculo da contribuição δW_l é de grande importância para o aprendizado da rede, uma vez que determina a variação em cada elemento $w_l^{(k)}$ de W_l , servindo como parâmetro fundamental do algoritmo de otimização para treinamento da RNA.

Segundo GOODFELLOW; BENGIO; COURVILLE (2016), HAYKIN (2005) e CHOLLET (2018), o algoritmo do Gradiente Descendente se caracteriza como um método comum para aprendizado dos pesos de uma rede neural. A derivada (ou vetor gradiente, quando $i < 1$ e $o < 1$, representando as quantidades de entradas e saídas da RNA, respectivamente) da função de custo J , representada por ∇J , aponta para a direção de um mínimo de J .

Exemplo 2.6 Para o exemplo 2.5, considere $\xi = 2$. o vetor $\nabla J(2)$ é mostrado na Figura 2.12 a seguir:

Figura 2.12: Gradiente $\nabla J(2)$

Considerada a contribuição calculada pelo *backpropagation*, o método do gradiente descendente define, em cada passo t , a atualização do peso W_l no instante t , denotado por $W_l^{(t)}$, pela equação:

$$W_l^{(t+1)} = W_l^{(t)} - a\delta W_l \quad (2.16)$$

em que $0 < a \leq 1$ é uma constante de aprendizado. A escolha de a é arbitrária e depende de cada treino. Valores altos da constante podem fazer com que a otimização não convirja e valores baixos causam o aumento do tempo de treinamento (HAYKIN, 2005), (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.8 Projeto e discretização de controladores

Embora a implementação de redes neurais artificiais aplicando o *forward propagation* pela equação (2.14), e seu respectivo treinamento pela regra do gradiente, mostrado pela Equação (2.16) sejam capazes de criar um modelo F^* que aproxime a função inversa do *backlash*, modelado pela Equação (2.10), a dinâmica do sistema é determinada pelo ramo direto do sistema em malha fechada, como representado pelo diagrama de blocos da Figura 2.13.

Como a rede neural é projetada para anular o efeito de $B(u)$, a dinâmica do sistema é determinada pelo produto entre a dinâmica do atuador e um controlador $C(s)$. Portanto, as adequações de características de regime transitório e permanente dependem de uma eficiente escolha dos polos e zeros de $C(s)$. OGATA (2009) e DORF; BISHOP (2010) oferecem alternativas de projetos de controladores baseados em função de transferência pelo método de alocação de polos. Em sua essência, são determinados, pelos comportamentos esperados em malha fechada (sobressinal, tempo de acomodação, erro de regime permanente, etc.),

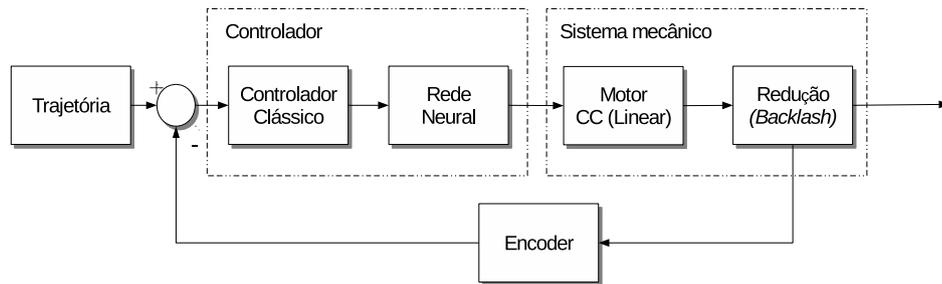


Figura 2.13: Diagrama de blocos do sistema

quais os polos desejados. Em seguida, são aplicados métodos para a obtenção da função de transferência de um controlador $C(s)$ responsável por "deslocar" os polos de $G(s)$ para os esperados.

OGATA (2009) mostra diversos procedimentos adotados para o projeto de um controlador clássico, baseado em alocação de polos. Nesses processos, é obtida uma função de transferência contendo polos p e zeros z , além de um ganho estático K_c , sendo todos os parâmetros dependentes do projeto.

Quando não há a necessidade de um controlador de ordem elevada como o da Equação (2.18), o projeto de controladores considera o caso das características de regime transitório serem satisfatórias, sendo necessária apenas a correção em regime permanente. Para este controlador, é dado o nome de compensador de atraso, com equação disposta por (2.17):

$$C(s) = K_c \frac{s + z}{s + p}, \quad (2.17)$$

com $z > p$. No caso da inadequação das características de regime transitório, é empregado um controlador de avanço, de equação semelhante à (2.17), mas considerando $z < p$. Quando combinados, é criado um controlador denominado de atraso-avanço, como mostra a Equação 2.18 a seguir:

$$C(s) = K_c \frac{(s + z_1)(s + z_2)}{(s + p_1)(s + p_2)} \quad (2.18)$$

A determinação de K_c , p e z (ou p_1, p_2 e z_1, z_2 no caso da equação (2.18)) depende das etapas de projeto a serem tomadas. OGATA (2009) mostra diferentes processos a serem tomados para a determinação de $C(s)$. Destacam-se o projeto pelo lugar geométrico das raízes (LGR) e o projeto por resposta em frequência (RF), exemplificados a seguir:

Exemplo 2.7 Para um sistema do exemplo 2.1 com $K = 0.9$ e $\tau = 0.025 \Rightarrow a = 40$, deseja-se obter 20% de Overshoot e tempo de acomodação de $T_s = 0.1$ segundos em malha fechada. Projete um controlador baseado pelo LGR

Solução: Verificando que as características em MF do sistema realimentado (dado pela equação (2.12)) tem $\zeta > 1$ e não apresentam comportamentos desejados em regime transitório, será necessário projetar um controlador de avanço, para que os polos em MF atinjam $s_d = -40 \pm 78.1j$. Definindo o zero em $z = -\zeta\omega_n = -40$, e verificando que a contribuição dos ângulos dos polos e dos zeros de $C(s)$ e $G(s)$ devem ser múltiplos de 180° , o que define o polo do controlador em $p = -80.00$ e $K_c = 200$. Pela Equação (2.17), temos o controlador:

$$C(s) = 200 \frac{s + 40}{s + 80},$$

que é capaz de acomodar o sistema em $T_s = 0.0979$ segundos com overshoot de 19%, conforme requerido pelo projeto. O LGR do sistema pode ser visto pela Figura 2.14 a seguir:

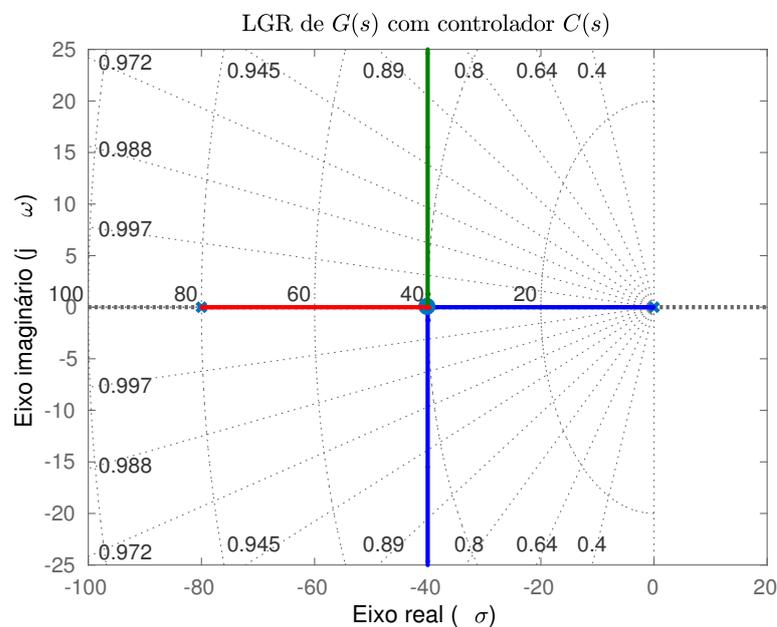


Figura 2.14: LGR do exemplo 2.7

Exemplo 2.8 Visando obter um sistema em malha fechada com $T_s = 0.2$ segundos e sobressinal máximo de 20%, projete um controlador baseado no método da resposta em frequência.

Solução: Convertendo os requisitos em termos de margem de fase, temos $MF = 48.15^\circ$. Adotando um ângulo de segurança de $\phi = 5^\circ$ e seguindo os procedimentos disponíveis em (OGATA (2009)), é obtido o controlador

$$C(s) = \frac{1530s + 1}{25.57s + 1}$$

Que tem $T_s \approx 0.18$ e sobressinal de 20%, conforme requerido pelo exemplo.

Embora os controladores obtidos pelos exemplos 2.7 e 2.8 sejam distintos, ambos são capazes de atender aos requisitos de malha fechada. É possível projetar compensadores baseando-se

em erros de regime permanente. Entretanto, como a equação (2.6) mostra a presença de um integrador no modelo do motor de corrente contínua (CC), o erro de estado estacionário se torna nulo (DORF; BISHOP (2010)).

Uma vez que a rede neural e o controlador será implementado em um sistema digital, será necessário a discretização do controlador $C(s)$ em um domínio z , obtido pela amostragem regular de $C(s)$ em intervalos T de amostragem. PHILIPS; NAGLE (1995) abordam diferentes critérios para a adoção do período de amostragem T , além da transformação de sistemas do domínio contínuo para o discretizado, pela Transformada Z.

2.3.9 Índices de desempenho

A análise de contribuição da RNA pode ser calculada através da análise de índices de desempenho. Se $y(t)$ é a resposta ao degrau de um sistema realimentado de controlador C com função de transferência semelhante à obtida no exemplo 2.1 (Equação 2.6), $y_B(t)$ é a resposta do sistema com *backlash*, dada por $y_B(t) = B(G_f * u(t))$ e $y_w(t)$ é a resposta ao sinal $w(t) = F^*(u(t))$, ou $y_w(t) = B(G_f * w(t))$, então podemos definir uma função de erro $E(y, y_k)$ sobre um intervalo de tempo de t_0 segundos pela Equação (2.19):

$$E(y, y_k) = \int_0^{t_0} |y(\tau) - y_k(\tau)| d\tau \quad (2.19)$$

O tempo t_0 corresponde ao tempo de simulação e/ou de teste na planta. Podemos definir a eficiência η da rede neural pela diferença entre o erro do sinal pré-compensado $y_w(t)$ e o obtido sem pré-compensação ($y_B(t)$), como na equação 2.20:

$$\eta = 1 - \frac{E(y, y_w)}{E(y, y_B)} \quad (2.20)$$

A partir da equação (2.20), é possível determinar que $\eta \approx 0$ aponta pequena contribuição da rede neural para a pré-compensação do sistema, $\eta \approx 1$ mostra que a rede foi capaz de mitigar eficientemente os efeitos da não-linearidade e, se $\eta < 0$, a rede teve efeito contrário ao esperado, prejudicando o desempenho do sistema em malha fechada.

Desenvolvimento

Este capítulo descreve as principais etapas elaboradas durante este trabalho de conclusão de curso. A primeira seção descreve os principais materiais e tecnologias para a elaboração do projeto, como *softwares* e componentes. Em seguida, são descritas as principais etapas de projeto e execução de atividades propostas. Serão detalhados os principais fatores para a escolha de parâmetros do projeto e quais os procedimentos adotados. Informações adicionais como desenhos técnicos e códigos-fonte estão disponíveis no Apêndice do texto.

3.1 Materiais e métodos

Parte do desenvolvimento do trabalho de conclusão de curso se deu pela utilização de diferentes materiais e tecnologias para projeto, análise e implementação do sistema e dispositivo, sendo usados *softwares* de *design*, programas de análise e simulação e IDEs. Além disso, o desenvolvimento sistema físico contou com sensores, atuadores e peças fabricadas por diferentes métodos, discutidos nas seções seguintes.

Para o desenvolvimento dos componentes físicos da planta, foi utilizado o *software FreeCAD* na versão 0.17. O programa é gratuito, de fácil acesso e usabilidade, apresentando diferentes funções de CAD para desenho de peças. Para simular a dinâmica do sistema em malha aberta e fechada, além de auxiliar nos processos de projeto e discretização do controlador, foi usado o *software MATLAB 2016b*, com o auxílio do pacote *Simulink*, em sua versão estudantil. O pacote *Simulink* apresenta uma alternativa simples para simulação de sistemas dinâmicos.

A programação da rede neural e de funcionalidades do sistema foi realizada com o auxílio da IDE Spyder, na versão 3.2.6., utilizando os pacotes *TensorFlow 1.17*, *Keras 2.2.4*, *SciPy 1.2.1*, *NumPy 1.17.3*, *Matplotlib 3.0.3*, *Control 0.8.1*, *GPIOZero 1.5.1* e *PyQT 5.0*.

Os componentes físicos da planta podem ser divididos em quatro grupos: atuadores, sensores, peças fabricadas e peças não fabricadas. No primeiro, foram utilizados os motores de corrente contínua AK360/78.8PL12S7000S e C2162-60006 / DN505728. Para o sensor usado,

foi escolhido o *encoder* LPD3806-400BM-G5-24C. As peças da primeira categoria citada foram fabricadas em ABS ou alumínio ABNT 6061. As demais peças são constituídas de materiais diversos (aços, madeira, etc.). As escolhas de cada componente serão detalhadas em seções seguintes desse capítulo.

3.2 Modelagem do sistema

O seguinte texto descreve as etapas de modelagem do projeto, detalhando os processos de levantamento dos modelos lineares e não lineares da planta, além da análise e simulações de regime transitório e permanente possível pelos modelos obtidos. O texto será dividido na escolha da não-linearidade estudada, levantamento de modelos, análise de regimes transitórios e permanentes e simulações em malha aberta e fechada.

3.2.1 Escolha da não-linearidade

Como discutido no capítulo de Introdução do trabalho, sistemas mecânicos são fontes de diversas não-linearidades. Os efeitos, causados por erros de projeto e/ou fabricação, desgaste ao longo do tempo ou intrínsecos de sua função, podem causar inadequações no desempenho de seu sistema, mesmo que controlados por compensadores projetados com base na teoria de controle clássico. A primeira etapa do trabalho, portanto, consistiu na escolha de estudo de uma não-linearidade encontrada em um sistema mecânico para compensação via RNA.

Entre os diversos possíveis sistemas e suas não-linearidades intrínsecas, foi escolhido o efeito causado pela folga (ou *backlash*) entre um par de engrenagens de dentes retos, durante o controle de posição angular de um conjunto coroa-pinhão. Nesse sistema, o pinhão, acoplado ao atuador, move uma engrenagem acoplada a um sensor *encoder*, a fim de manter uma posição angular (*setpoint*) regulada pelo usuário. O sistema possui um mecanismo secundário responsável por fornecer um afastamento programado entre as engrenagens, configurando diferentes valores de *backlash* para análise.

3.2.2 Modelo do sistema

A modelagem do sistema pôde então ser dividida em duas partes. A primeira consiste no modelo da dinâmica linear do sistema, sendo representada por funções de transferência. Já o segundo modelo contém a não-linearidade estudada. O modelo típico para motores CC é descrito pela Equação (2.11). Já o efeito da não-linearidade do *backlash* em um par de engrenagens pode ser descrito pelo modelo da Equação (2.10).

Considerando que o sistema consiste no controle de posição de um mecanismo coroa-pinhão, sua dinâmica foi unicamente expressa pela dinâmica do atuador. Assim, a parte linear do modelo pode ser descrita pela função de transferência do motor de corrente contínua

AK360/78.8PL12S7000S. O levantamento do modelo do motor ocorreu com a aplicação das equações (2.3), (2.4) e (2.6). O valor da resistência de armadura R_a foi medida entre os terminais com um multímetro e os dados restantes foram extraídos de informações fornecidas pelo fabricante (AKIYAMA, 2019). Como forma de validar o modelo, o sistema foi excitado com uma entrada em degrau e avaliada sua velocidade. Dessa forma, o efeito do integrador é desprezado e pode ser visto o efeito do segundo polo.

Analisando o funcionamento do mecansimo, foi possível perceber que as dinâmicas linear e não-linear do sistema podem ser descritas em um diagrama de blocos como conectadas em série, contendo o modelo linear no início e o efeito do *backlash* em seguida. Isso ocorre uma vez que o motor CC, de dinâmica linear, move o pinhão. Esse elemento, por sua vez, é afastado da coroa-encoder, sendo a causa da não-linearidade do sistema. A Figura 3.1 mostra a disposição dos modelos em um diagrama de blocos:

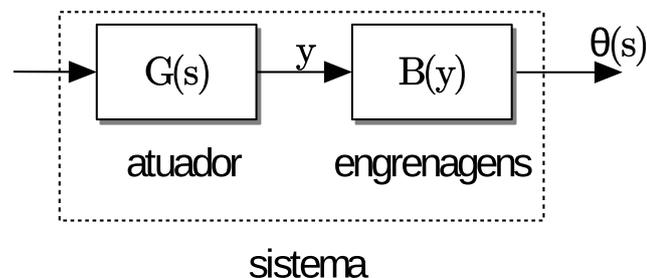


Figura 3.1: Diagrama de blocos do sistema em malha aberta

3.2.3 Análise do modelo

Tendo o modelo linear do sistema, foi possível prever o comportamento do mesmo em regime transitório e permanente, em malha aberta e malha fechada. As respostas do sistema a entradas degrau foram comparadas com o comportamento esperado, discutido no capítulo de fundamentação teórica do texto. O efeito da NL também foi estudado em malha aberta, para um sinal de entrada com pequenas variações em amplitude.

Tendo as informações de características em malha aberta, foi possível também determinar os critérios de desempenho em malha fechada. Nessa etapa, foram definidos parâmetros para o comportamento do sistema, quando realimentado e com sinal modulado or um controlador. O controlador em questão foi projetado a partir dos critérios de desempenho exigidos em malha fechada.

3.3 Projeto e discretização do controlador

Para atender os critérios de desempenho exigidos do sistema em malha fechada, foi necessário o projeto e implementação de um controlador. As características desejadas podem ser determinada pela alocação de polos e zeros, e os métodos de projeto se baseiam nos métodos discutidos no capítulo de Fundamentação Teórica, resultando em um controlador de função de transferência de forma mostrada na Equação (2.17).

3.3.1 Projeto do controlador

De forma a cumprir os critérios de desempenho em malha fechada, avaliado pelo modelo baseado em função de transferência do sistema, foi projetado um controlador linear. Essa etapa foi iniciada através da escolha de um dos possíveis métodos de síntese de controladores, discutidos no capítulo de fundamentação teórica. Da análise de seu desempenho desejado em malha fechada, foi possível definir qual o tipo do controlador (atraso, avanço ou atraso-avanço).

Foi escolhido o método do Lugar Geométrico das Raízes, como uma alternativa simples de projeto, sendo possível visualizar, no plano complexo, o cancelamento de polos do modelo da planta com zeros do controlador. O cancelamento causa alterações no regime transitório do sistema, como requerido pelo projeto. Assim, o controlador tem zeros mais próximos da origem do plano complexo que seu polo, se caracterizando como controlador de avanço. Ainda pelo LGR, foi possível determinar o ganho do controlador necessário para atender melhor as características de desempenho em malha fechada do sistema.

3.3.2 Discretização do controlador

O projeto do controlador discreto teve início com a escolha do período de amostragem. A escolha ocorreu de forma a posicionar no plano de frequência complexa discreta (plano Z), os polos discretos do modelo da planta em $p_G \approx 0.6$, como recomendado em (PHILIPS; NAGLE, 1995). Conhecido o valor de T_s adequado ao critério, foi possível obter o modelo discreto da dinâmica da planta. Da mesma forma, o controlador foi discretizado, obtendo-se seu modelo discreto e, por conseguinte, a equação de diferenças que o rege.

Uma proximidade entre o valor do zero do controlador e do polo discretizado do modelo em (2.11) implica na anulação dos efeitos da dinâmica do sistema em malha fechada, como previsto no projeto do controlador em tempo contínuo. Tendo a função de transferência de $C(z)$, foi possível levantar a sua equação de diferença, para determinar u_k , em função dos sinais de controle anteriores e do erro e_k , implementada em código para controle do sistema físico.

3.4 Projeto da planta

O seguinte texto detalha o processo de projeto do mecanismo empregado para controle de posição. Esse texto descreve os processos de concepção do mecanismo, análise de restrições de valores máximos e mínimos de parâmetros do sistema, esboço do sistema, escolha dos materiais e processos de fabricação, relatório da fabricação das peças e montagem da planta. A Figura 3.2 uma representação do sistema.

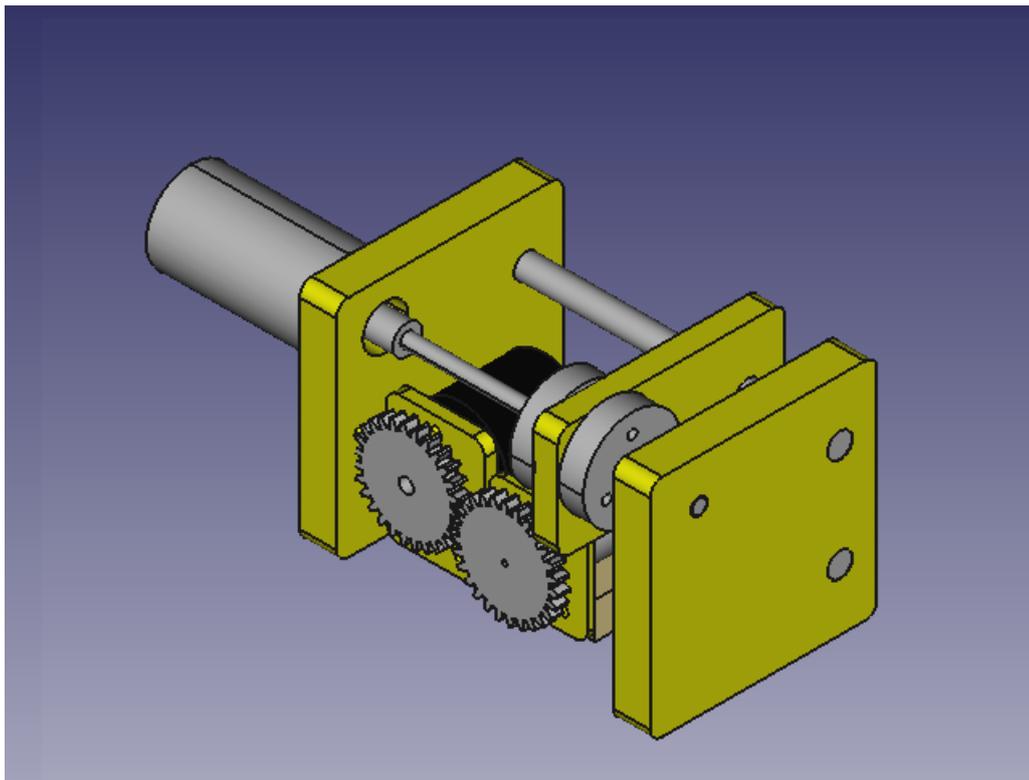


Figura 3.2: Modelo 3D do sistema

3.4.1 Formulação do mecanismo

Das diversas folgas em pares de engrenagens, foi escolhida a responsável pelo afastamento de seus centros. Na situação ideal, as circunferências primitivas dos elementos são tangentes e, segundo (NORTON, 2008), a lei do engrenamento permite que a transmissão ocorra continuamente. Isso acontece porque um par de dentes entra em contato e se distanciam no início e fim (respectivamente) de um segmento de reta, nomeado linha de ação. Ao contrário, quando existe afastamento dos diâmetros primitivos, a linha de ação é encurtada, fazendo com que o contato do primeiro par de engrenagens ocorra com atraso. Se mantida a velocidade, entretanto, o engrenamento ocorre de forma contínua, pela lei do engrenamento. O arco percorrido pelo primeiro dente do pinhão sem que haja contato é denotado por j_{θ} e pode ser relacionado com o afastamento dos diâmetros primitivos j_r pela equação (2.9).

Para controlar o afastamento j_r e, conseqüentemente, a folga j_θ , uma das engrenagens foi acoplada sobre um carro (contendo uma porca) e um parafuso, posicionado perpendicularmente à eixos guias, paralelos entre si e ao parafuso. As guias restringem a movimentação do carro apenas da direção de seus eixos, fornecendo um grau de liberdade. O parafuso, constituído por uma barra roscada M6 de passo $p = 1mm$ conforme (ABNT, 2004), realiza, quando rotacionado em seu próprio eixo, contato com a rosca interna da porca de mesmo passo. Acoplando a porca no carro travado pelos eixos-guia, o único movimento possível acontece no sentido das guias e parafuso. O sistema pode ser visto pela Figura 3.3.

Como forma de estimar o valor de j_θ , buscando eliminar as fontes contribuintes de erros de medição, foi desenvolvida também uma porca *anti-backlash*. Considerando que o sistema discutido se comporta como um par cremalheira-engrenagem (SHIGLEY; MISHCKE, 1989), deve-se considerar as folgas desse par, e como essa pode influenciar em erros de medição de j_θ . Para solucionar o problema, duas porcas (consultar apêndice) contendo um furo central roscado com mesmo passo do parafuso e furos auxiliares foram construídas. As peças são unidas por parafusos acoplados em seus furos auxiliares. Molas helicoidais de compressão também são inseridas na peça, com a finalidade de afastar as duas porcas.

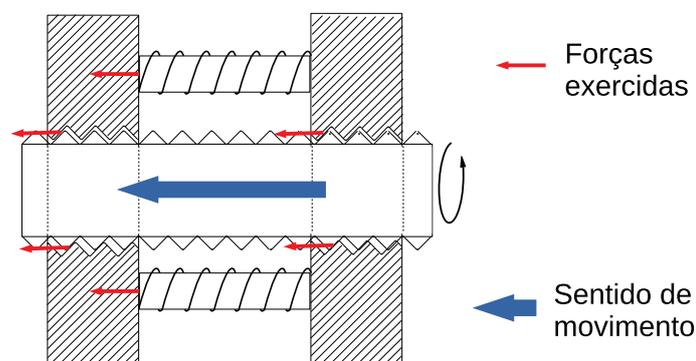


Figura 3.3: Forças exercidas na porca *anti-backlash*

A Figura 3.3 mostra as principais tensões exercidas pelo conjunto. É possível perceber que, pela reação de compressão das molas, as porcas se afastam até obterem contato com os flancos das roscas interna e externa. O efeito evita folgas entre o parafuso e a porca, diminuindo erros de estimativa de j_θ .

Visando reduzir erros de estimativa de j_θ , o giro do parafuso e conseqüente deslocamento do carro sobre os eixos-guia aconteceram pela atuação de um motor de corrente contínua. Energizando o motor com tensão e tempo de ativação fixos, é causado um mesmo deslocamento θ_p por pulso. Pela equação (3.2), um mesmo valor de Δj_θ é aplicado por acionamento. O uso do motor CC se faz necessário para adquirir precisão no acréscimo da folga, uma vez que os valores são relativamente baixos e implicam diretamente em parâmetros da não-linearidade

do sistema, como visto nas equações (2.9) e (2.10).

3.4.2 Análise de restrições

A seguinte subseção detalha o processo seguido para obtenção dos valores mínimos e máximos de folgas entre os pares de dentes de engrenagens, configurado pelo mecanismo, além do valor estimado aumentado por revolução do parafuso. Sabendo que o mecanismo de parafuso e porca realiza um afastamento j_r entre os diâmetros primitivos a partir de um deslocamento angular θ_p , podemos relacionar os dois valores pela Equação (3.1):

$$j_r = \frac{\theta_p}{2\pi} p \quad (3.1)$$

Reavaliando a Equação (2.9), temos a relação entre j_r e j_θ . Por (2.9) e (3.1), temos a Equação (3.2):

$$j_\theta = \frac{2 \tan(\beta)}{\pi m z} \theta_p. \quad (3.2)$$

Adotando $\theta_p = 4\pi$ (duas voltas) e sabendo que os módulos das engrenagens, seus números de dentes e ângulos de pressão são constantes e relatados na Tabela 2.1, é possível relacionar a folga ganha Δj_θ por cada deslocamento (pulso) realizado pelo parafuso.

Na situação de contato extremo, os dentes estão no limiar de afastamento, fazendo o maior ângulo permitido para o engrenamento. O afastamento ocorre até o ponto crítico em que as engrenagens deixam de se tocar, que acontece quando a distância entre os círculos extremos excede o diâmetro de coroa da engrenagem. O *backlash* máximo foi denotado por $j_{\theta M}$. Quando $j_r = 0$, foi preciso considerar também que existe um *backlash* mínimo, denotado por $j_{\theta 0}$.

3.4.3 Esboço do sistema

Após diferentes revisões no projeto do mecanismo, foi adotado o seguinte sistema, mostrado em vista explodida na Figura 3.4. O detalhamento de cada peça pode ser visto através de sua respectiva representação em desenho técnico, constando no capítulo de Apêndice desse trabalho. Após a fabricação das peças, a montagem do sistema físico ocorreu de acordo com a numeração dos componentes mostrados na Figura 3.4.

o processo de usinagem convencional. Para a engrenagem, foram realizadas as etapas de torneamento e fresamento. Como ferramenta de redução dos erros de estimativa do *backlash* no par de engrenagens, a porca anti-*backlash* demanda maior tolerância dimensional. Por esse motivo, foram empregados os processos de torneamento, furação e roscamento nas peças. Um acoplador usado entre motor secundário e a barra foi também construído pelos processos de torneamento, furação e roscamento.

O carro, deslocado pelo giro do parafuso com a porca anti-*backlash*, e as estruturas de sustentação dos eixos-guia e barra roscada, foram construídos pelo processo de impressão 3D. A escolha do processo ocorreu pelo baixo custo de fabricação e pela desnecessidade de tolerância dimensional em seus componentes, em contraste com as peças até então discutidas. O terceiro grupo de peças, constituídos de bases e suportes de madeira, eixos-guia e barra roscada, foram adquiridos sem processos críticos de fabricação, exceto pela furação das chapas de madeira, para aparafusamento dos componentes.

Acerca dos materiais utilizados em cada peça, foi considerada a disponibilidade prévia dos materiais, visando a redução de custos de fabricação da planta. A Tabela A.1, disponível no Apêndice, mostra a relação de peças destacadas pela Figura 3.4, seus materiais e processos de fabricação.

3.4.5 Fabricação dos componentes

Engrenagens

O efeito do *backlash* medido no projeto deve acontecer devido ao afastamento entre os diâmetros primitivos das engrenagens. Como discutido anteriormente, o afastamento deve possuir valores mínimos e máximos. Para medir os efeitos da folga entre um par de engrenagens, foram projetadas duas peças idênticas de mesmo material.

O projeto das engrenagens para transmissão teve como critérios limitantes o diâmetro mínimo e máximo das engrenagens, além da limitação fornecida pelo maquinário disponível. Considerando um material bruto com diâmetro máximo de $d_{max} = 2'' = 50,8mm$ e um sensor *encoder*, limitante mínimo, com diâmetro de $d_{min} = 38mm$, foi obtida uma primeira faixa limitante do projeto. As ferramentas disponíveis permitiam a construção de engrenagens com módulo de $m = 1,5mm$ e uma relação de redução da máquina de $i = 40$. Pela tabela, 2.1 foram obtidos os seguintes parâmetros para a escolha do número de engrenagens:

$$d_{min} \leq m(z + 1) \leq d_{max}$$

$$24 \leq z \leq 33$$

A escolha de redução unitária se deu pela facilidade da fabricação dos componentes e controle do sistema, uma vez que o valor do fator de redução não possui influência direta no

sistema de controle, podendo ser modelada por um ganho estático de valor $-N$. Com os valores de m e z , pelas dimensões da Tabela 2.1, temos:

Tabela 3.1: Parâmetros da engrenagem

Parâmetro	Símbolo	Valor	Unidade
Módulo	m	1,5	mm
Número de dentes	z	27	
Diâmetro primitivo	d_p	40,5	mm
Diâmetro máximo	d_{max}	42	mm
Largura	b	5	mm

Para o acoplamento da engrenagem aos sensores e atuadores, foram usados parafusos M4 perpendiculares ao eixo de rotação, com uma extensão do eixo da engrenagem. O desenho dos componentes pode ser visto em seus respectivos desenhos técnicos (consultar Apêndice).

Demais componentes

Verificando que j_θ possui pequenas variações por pulso do motor secundário, sua estimativa se torna imprecisa devido a fontes de erro. Então, foi necessário desenvolver mecanismos para minimizar os efeitos de folgas externas, não provenientes do afastamento das engrenagens. Além do par de eixos-guia, foi desenvolvida uma porca anti-*backlash*. Um segundo acoplador, para unir o motor secundário à barra roscada M6, foi também desenvolvido, com sistema de acoplamento similar às engrenagens. Os desenhos técnicos da porca anti-*backlash* e do acoplador podem ser vistos na seção de Apêndice.

3.5 Projeto da rede neural artificial

O seguinte texto detalha os procedimentos tomados para o projeto e implementação da rede neural artificial usada para compensar os efeitos não-lineares causados pelo *backlash* em um par de engrenagens. Nesse texto, são abordados os critérios para a escolha de número de camadas, dimensões das matrizes de peso e funções de ativação e os valores de entrada e saída esperada para aproximação do modelo.

O projeto e implementação da rede neural artificial se deu em duas etapas. Na primeira, foi verificado se uma rede F^* é capaz de compensar os efeitos de uma característica não linear B modelada pela equação (2.10) em um sistema com dinâmica G descrita pela equação (2.11), estabilizando o sistema em malha fechada. Considerando apenas o critério de estabilidade, o número de camadas e dimensões das matrizes de pesos não foram restritivas nessa primeira etapa. A função de ativação foi escolhida conforme (SELMIC; LEWIS, 2000). Em uma etapa posterior, foram consideradas as limitações computacionais do *hardware* do sistema de controle. Tendo em vista a velocidade de processamento e memória principal do *Raspberry Pi*

3, como visto na Tabela 3.5, foi levantada a necessidade da redução da ordem das matrizes e/ou redução do número de camadas da RNA. O cálculo de matrizes de ordem elevada pode configurar-se como alto custo computacional, elevando o tempo de processamento e causando atrasos no sistema, a depender da dinâmica do sistema. (PHILIPS; NAGLE, 1995). Para que a ordem então pudesse ser reduzida, as escolhas de funções de ativação, ordem das matrizes e quantidades de camadas foram reprojatadas.

3.5.1 Treinamento da rede

Em ambas as etapas, o treinamento da rede neural aconteceu em malha aberta. Considerando que a principal função da RNA é de mitigar os efeitos não-lineares do *backlash* em um par de engrenagens, e que o modelo de B mostrado pela equação (2.10) não causa instabilidade em malha aberta, então F^* pode ser treinada em malha aberta. A Figura 3.5 mostra o diagrama de blocos para o treinamento:

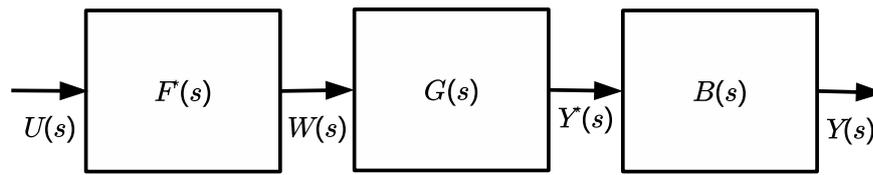


Figura 3.5: Diagrama de blocos para treinamento da rede neural

Tendo $y^*(t)$ como a representação no tempo da saída linear (sem a interferência do *backlash*) e $y(t) = B(y^*(t))$ a saída real do sistema, podemos calcular a função de custo $J(y, \hat{y}) = J(y^*, y)$, tendo a saída esperada como y^* e a real, produzida pelo sinal $u(t)$, como y . Dessa forma, definimos as entradas e saídas da RNA, bem como suas dimensões, como $i = o = 1$.

Contudo, pela definição trazida por CYBENKO (1989) da Equação (2.13), uma rede neural com apenas funções de ativação sigmoidais não é capaz de simular respostas dinâmicas, podendo aproximar modelos estáticos. Assim, a eficácia da RNA, medida pela Equação (2.20) é diminuída pela dinâmica de $G(s)$ não modelada. Apesar do fato, o sinal $w = F^*(u)$ causa alterações na resposta de $G(s)$, se a rede for corretamente treinada (ou seja, se $J(y, \hat{y}) \rightarrow 0$), aumentando o valor de η da Equação (2.20).

Acerca do sinal de entrada para treinamento, deve ser considerado que o processo de treino de uma rede neural convencional não tem memória de suas entradas anteriores (GOODFELLOW; BENGIO; COURVILLE, 2016). Dessa forma, o sinal de entrada $u(t)$ do sistema deve ser escolhido de forma a excitar a não-linearidade da planta em diferentes intensidades, proporcionando a maior quantidade de saídas esperadas possível para treino.

3.5.2 Redução de ordem da RNA

Conforme discutido em parágrafos anteriores, a complexidade computacional do cálculo de *forward propagation* da rede neural pode causar atrasos de tempo no sistema em malha fechada, podendo causar instabilidades. Outra motivação para a simplificação da estrutura do compensador consiste no possível *overfitting* da rede. Em um modelo de RNA de matrizes de ordem desnecessariamente elevadas, o processo de treino pode ajustar o conjunto de pesos para não apenas aproximar o sinal de forma a capturar a tendência do sinal, mas pode interpretar erros de medição como respostas legítimas do sistema. A Figura 3.6 representa erros de *overfitting*.

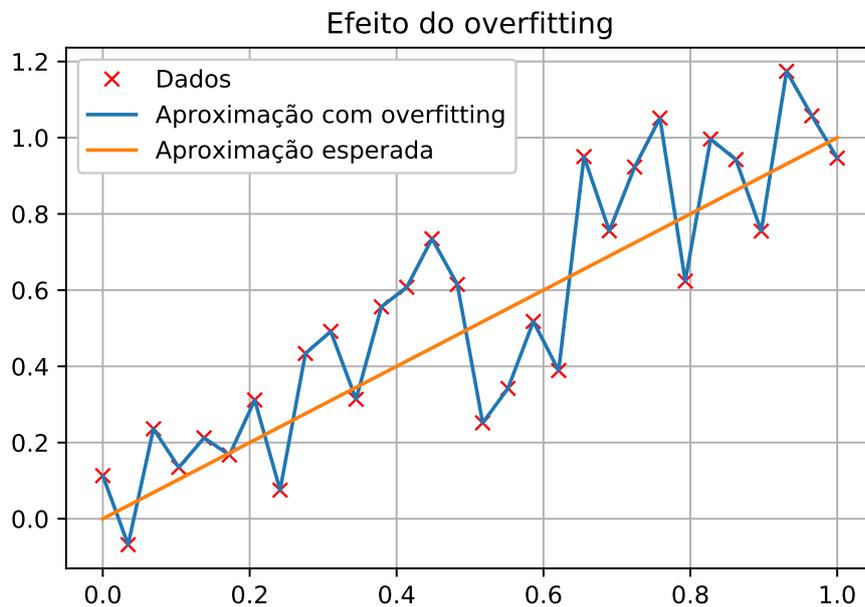


Figura 3.6: Efeitos do *overfitting* em uma função

Fez-se necessária, portanto, a redução da ordem da RNA. A simplificação da estrutura do compensador pode ser feita reduzindo a dimensão das matrizes dos conjuntos de pesos, diminuindo a quantidade de camadas da rede e alterando as funções de ativação em cada camada.

Definindo funções de ativação

Uma forma de reduzir a complexidade da rede neural parte da investigação de funções de ativação aplicáveis. Segundo (SELMIC; LEWIS, 2000), a inversão de uma não linearidade do tipo zona morta, descrita na equação (2.2), pode ser aproximada por uma rede neural contendo funções de ativação do tipo PReLU, exemplificadas na Equação (3.3):

$$f_k(u) = \begin{cases} u & u > 0 \\ \beta u & u \leq 0 \end{cases} \quad (3.3)$$

Em que $\beta \leq 1$ é um parâmetro treinado pela rede, e de dimensão da entrada u da função de ativação. Entretanto, para a aproximação de uma não linearidade descrita pelo modelo da Equação (2.10) requer uma investigação minuciosa do efeito.

Considerando a principal função do compensador como mitigar os efeitos não-lineares do *backlash* e analisando a equação do modelo da Equação (2.10), foi possível relacionar que, em tempo discreto, as operações de diferenciação e integração podem ser descritas pelas aproximações mostradas no conjunto de equações (3.4) (PHILIPS; NAGLE, 1995):

$$\int u(t)dt \approx u_{k-1} + T_s u_k \quad (3.4a)$$

$$\frac{du}{dt} \approx \frac{u_k - u_{k-1}}{T_s} \quad (3.4b)$$

Tendo em vista o modelo da Equação (2.2) e as aproximações do conjunto de equações (3.4), $B(u)$ pode ser descrito como mostra a Equação (3.5):

$$B(u_k) = D_z(u_k - u_{k-1}) + u_{k-1} \quad (3.5)$$

É desejado encontrar a função inversa B^{-1} do modelo. Usando a Equação (3.5), é possível obter uma aproximação para B^{-1} , como descrito na Equação (3.6):

$$B^{-1}(u_k) = D_z^{-1}(u_k - u_{k-1}) + u_{k-1} \quad (3.6)$$

Contudo, a função D_z , mostrada na Equação (2.2) não é invertível. Além disso, não é uma função contínua ou possui gradiente, como requerido pelo método de *backpropagation*. O gradiente da função de ativação é usado para atualização dos valores de peso, como visto pelas Equações (2.15) e (2.16). Portanto, uma aproximação de D_z^{-1} , condizente com os critérios estabelecidos pelo método de *backpropagation*, é necessária. É possível então definir uma aproximação D^* da função inversa da função de zona morta e também definir seu gradiente, denotado por ∇D^* , como mostrado na Equação (3.7):

$$D^*(u) = \begin{cases} u - \alpha & u < -\epsilon \\ u(1 + \frac{\alpha}{\epsilon}) & |u| \leq \epsilon \\ u + \alpha & u > \epsilon \end{cases} \quad (3.7a)$$

$$\nabla D^*(u) = \begin{cases} 1 & u < -\epsilon \\ (1 + \frac{\alpha}{\epsilon}) & |u| \leq \epsilon \\ 1 & u > \epsilon \end{cases} \quad (3.7b)$$

Em que α é o fator de *backlash*, visto na Equação (2.2) e $0 < \epsilon < \alpha$, foi um valor arbitrado como $\epsilon = \alpha/5$. É possível perceber que D^* é contínua e invertível. Definido o gradiente ∇D^* , a função se tornou própria para a aplicação do algoritmo de *backpropagation* do compensador.

Redução de camadas e dimensões de pesos

Analisando o modelo da Equação (3.6), é possível perceber que os parâmetros necessários dependem de valores anteriores do sinal de controle u , entrada da rede. Como a RNA não possui realimentações ou estados de memória, como no caso das redes LSTM (*Long-short term memory*) (GOODFELLOW; BENGIO; COURVILLE, 2016), (CHOLLET, 2018) e a dimensão de entrada da RNA precisa tem ordem $i = 1$, como mencionado em parágrafos anteriores, u_{k-1} precisa ser estimado a partir da entrada u_k da rede. Considerando que $u(t)$ é contínuo, como esperado pela saída de um controlador LTI (*Linear, time-invariant*) (DORF; BISHOP, 2010), usando a equação 3.4b, é possível estimar u_{k-1} como um produto da entrada u_k por uma matriz de pesos $W_0 \in \mathbb{R}^{(1,2)}$. Sabendo da dimensão do primeiro peso, sabemos que a resposta da primeira camada, usando as etapas descrita em (2.14), que a saída z_1 tem ordem $(1, 2)$. Como a saída tem ordem $(1, 1)$, uma vez que representa o sinal condicionado, é necessária mais uma camada com peso $W_1 \in \mathbb{R}^{(2,1)}$.

A RNA, portanto, deve possuir duas camadas, com pesos de dimensões iguais as de W_0 e W_1 , respectivamente. A função de ativação da primeira camada é linear, para estimação de u_{k-1} . O resultado é levado à função de ativação D^* , e regulado pela última camada da rede, através de outra função de ativação linear. O compensador pode ser descrito como no diagrama de blocos da Figura 3.7:

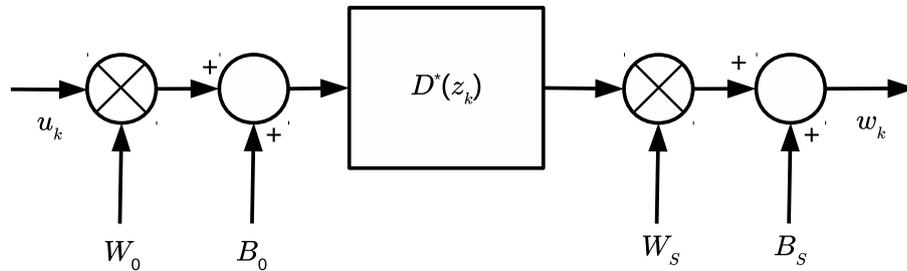


Figura 3.7: Diagrama de blocos do compensador

A comparação entre as duas versões do compensador ocorreu através do resíduo das RNAs para um sinal de entrada v de validação, com elementos distribuídos aleatoriamente. Segundo (SOUZA, 2018), cálculo dos resíduos é dado pela Equação (3.8):

$$Res \{F^*\} = -\nabla F^* = -\frac{\partial}{\partial u} F^* \quad (3.8)$$

3.6 Desenvolvimento do sistema de controle

Paralelamente ao processo de fabricação das peças, foi desenvolvido o sistema de controle para o *Raspberry Pi*. Na implementação do sistema, foram usados os módulos *gpiozero* para o

acionamento dos pinos de GPIO do RPI, PyQT para a geração de código da interface gráfica do usuário (GUI), além dos módulos *control* e *matplotlib* para controle e disposição de gráficos.

Tratando-se de um dispositivo embarcado, o controle de pontos de operação e acionamento de dispositivos secundários via GPIO no *Raspberry Pi* torna-se facilitado com a implementação de uma interface gráfica de usuário. O sistema desenvolvido possui ferramentas para ativação do motor secundário, girando a barra roscada no passo desejado, acrescentando ou removendo um *backlash* estimado de j_θ por passo, Δj_θ . É possível também alterar o ponto de operação (*setpoint*) r para valores no intervalo $[0, 2\pi]$, correspondendo às posições angulares do sistema de transmissão. Também foram programados módulos para controle discreto do sistema, instrumentação do sensor e acionamento do atuador principal.

3.6.1 Sensores e atuadores

Atuadores

O atuador do sistema mecânico estudado neste trabalho de conclusão de curso é um motor de corrente contínua. A partir deste, o pinhão é movido, girando a coroa, afastada por uma distância configurável. Tendo em vista que não serão necessárias altas cargas sobre o motor par de engrenagens, os critérios de seleção do atuador envolvendo torques foram desprezados. Assim, as restrições para dimensionamento do motor CC foram de valores máximos e mínimos para a fonte de alimentação da planta, disponibilidade comercial e custo. Dentro dos critérios, foi utilizado o motor CC AK360/78.8PL12S7000S, de dados mostrados na Tabela 3.2 e desenho técnico (fornecido pelo fabricante), a seguir:

Tabela 3.2: Motor CC AK360/78.8PL12S7000S. Fonte: (AKIYAMA, 2019)

Tensão		Sem Carga		Partida
Faixa	Nominal	Rotação	Corrente	Torque
6V - 24V	12V	7000 RPM	140 mA	407 gf.cm
Máximo Rendimento				
Rotação	Torque	Corrente	Potência	Rendimento
5700 RPM	78,8 gf.cm	590 mA	4,58 W	63%

Para a realização do afastamento programado j_r que causa o *backlash*, foi utilizado um motor de corrente contínua C2162-60006/DN505728. A escolha do dispositivo se deu por sua fácil disponibilidade, sendo reaproveitado de projetos anteriores desenvolvidos pelo autor. Considerada a funcionalidade a ser desempenhada pelo atuador (giro da barra para deslocamento do carro), foi necessário o emprego de um motor com maior disponibilidade de torque do que a possível pelo motor principal (AK360/78.8PL12S7000S). As informações técnicas do dispositivos podem ser conferidas pela Tabela 3.3:

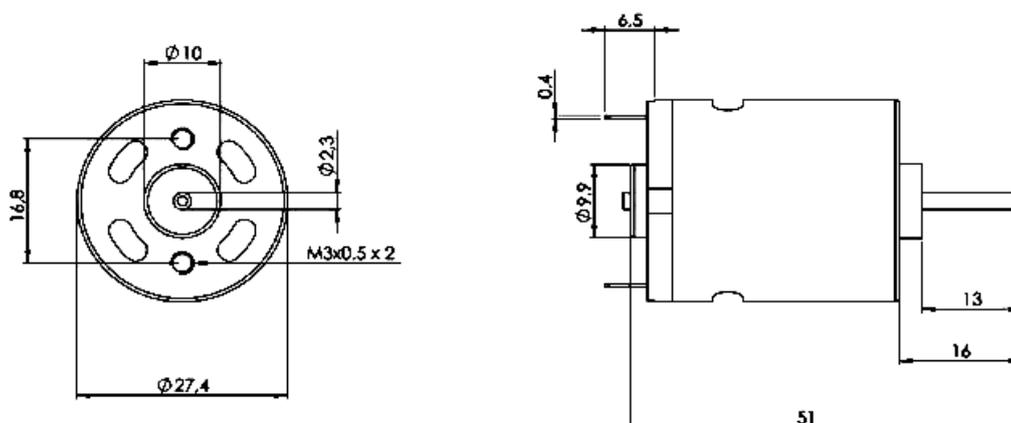


Figura 3.8: Dimensões do motor CC. Fonte: (AKIYAMA, 2019)

Tabela 3.3: Motor C2162-60006/DN505728. Fonte: (TREASE, 2019)

Referência			Unidade	Referência		Unidade
Tensão				Rotação máxima		
Nominal	Mínima	Máxima		4550		<i>RPM</i>
19	2	24	<i>V</i>	476		<i>rad/s</i>
Corrente				Dimensões		
Mínima	Máxima			Motor		
0.15	2.5		<i>A</i>	Diâmetro	Comprimento	
Resistência				37	64	<i>mm</i>
8.2			Ω	Eixo		
Torque máximo				Diâmetro	Comprimento	
28.7			<i>N.cm</i>	3.1	85	<i>mm</i>
40.68			<i>in.oz</i>			

Sensor

A planta também utiliza com um *encoder* rotacional, acoplado na coroa, para a medição da posição do elemento, monitorando a saída do sistema. O sensor utilizado foi um *encoder* LPD3806-400BM-G5-24C, de 400 PPR. A escolha deste sensor se dá pela precisão, disponibilidade comercial e baixo custo. Os detalhes do motor se encontram na Tabela 3.4 a seguir.

Sistema embarcado

Os desenvolvimentos do controlador, projetado conforme etapas discutidas futuramente, além da rede neural para compensação, devem ser executados em um sistema embarcado com entradas e saídas digitais de fácil acesso. Nesse caso, o computador escolhido foi um *Raspberry Pi 3 Model B*, de características listadas pela Tabela 3.5.

A escolha do microcomputador se deve pela alta capacidade de processamento, necessária

Tabela 3.4: Encoder LPD3806-400BM-G5-24C. Fonte:(ROBIZ, 2019)

	Mín	Máx	Unidade
Rotação		5000	RPM
Resposta em Frequência	0	20	KHz
Faixa	5	24	V
Resolução	400		PPR
Diâmetro do eixo	6.0		mm
Comprimento do eixo	12.9		mm

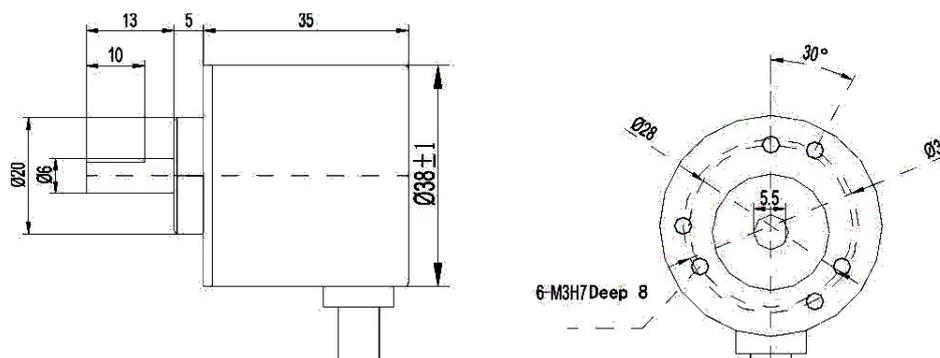


Figura 3.9: Dimensões do Encoder. Fonte: (ROBIZ, 2019)

Tabela 3.5: *Raspberry Pi 3 Model B*. Fonte: (RASPBERYPi, 2019)

Processador			
Número de núcleos	4	Frequência	1.2GHz
Memória Principal			
RAM	1 GB		
Portas			
USB	2	GPIO	40

para os cálculos requeridos pela rede neural, sendo uma alternativa de baixo custo. A disponibilidade dos pinos GPIO (*General Purpose Input-Output*, ou "entradas e saídas de propósito geral", em tradução livre) também configura um critério de escolha do *Raspberry Pi* (RPI).

Módulo Ponte-H

Para o controle dos motores principal e secundário, foi utilizado o módulo Ponte-H L298N. O componente foi adotado devido a sua facilidade de configuração e uso, sendo um componente reaproveitado de projetos anteriores e não apresentando, portanto, custos adicionais ao projeto. A Tabela 3.6 mostra as principais características do componente L298, núcleo do módulo L298N.

Tabela 3.6: Módulo Ponte-H L298N. Fonte: (HANDSON, 2019)

Características Elétricas			Unidade
Tensão (motores)	Mín.	Máx.	
	5	35	V
Corrente (motores)			
	0	2	A
Tensão (lógico)	Baixo	Alto	
	$-0.3 \leq V_{in} \leq 1.5$	$2.3 \leq V_{in} \leq 5$	V

Fonte de potência

Para alimentar o microcomputador, os atuadores e o sensor *encoder*, foi usada uma fonte CC 400W-ATX-PFC-R. O componente foi reaproveitado de projetos anteriores, não apresentando custos adicionais para o trabalho. A Tabela 3.7 mostra as principais características da fonte.

Tabela 3.7: Fonte 400W-ATX-PFC-R

Tensão	Corrente (Mín)	Corrente (Máx)	Regulação	Tensão Ripple
5V	3A	35A	$\pm 5\%$	50mV
12V	2A	30A	$\pm 7.5\%$	120mV
-5V	0A	0.5A	$\pm 5\%$	150mV
-12V	0A	0.8A	$\pm 5\%$	150mV
+3.3V	1A	28A	$\pm 5\%$	50mV
+5VSB	0.1A	2A	$\pm 5\%$	50mV

3.6.2 Programação do sistema

Módulos empregados

A programação do controlador e da rede neural foi feita na linguagem *Python*, na versão 3.6.7. A linguagem possui bibliotecas (como os módulos *TensorFlow v1.17*, *Keras v2.2.4*, *SciPy v1.2.1* e *NumPy v1.17.3*) que facilitam o desenvolvimento de soluções de aprendizado de máquina, realizando cálculos de forma eficiente e com código limpo. Para a elaboração dos códigos de *forward propagation* e treinamento da rede, foram utilizados os módulos *Keras*, *SciPy* e *NumPy*. A geração de gráficos (resultados) foi feita com o auxílio do módulo *Matplotlib v3.0.3* e o módulo *Control v0.8.1* foi empregado para simulações da dinâmica da planta, para treinamento da RNA. O desenvolvimento da interface gráfica do sistema ocorreu com o uso da biblioteca *PyQT v5.0*

Interface gráfica do usuário

A interface gráfica do usuário (ou GUI), permite exercer as funcionalidades do sistema de forma simplificada e intuitiva. Através dela, podemos selecionar o valor do ponto de operação

(ou *setpoint*); comandar o acionamento do motor secundário, aumentando ou diminuindo a folga entre o par de engrenagens; acionar a atuação do controlador e RNA na malha fechada e visualizar, em forma de gráficos, a resposta do sistema em tempo real. O *design* da interface gráfica foi feito com o módulo PyQT. A interface gráfica resultante pode ser vista na Figura 3.10.

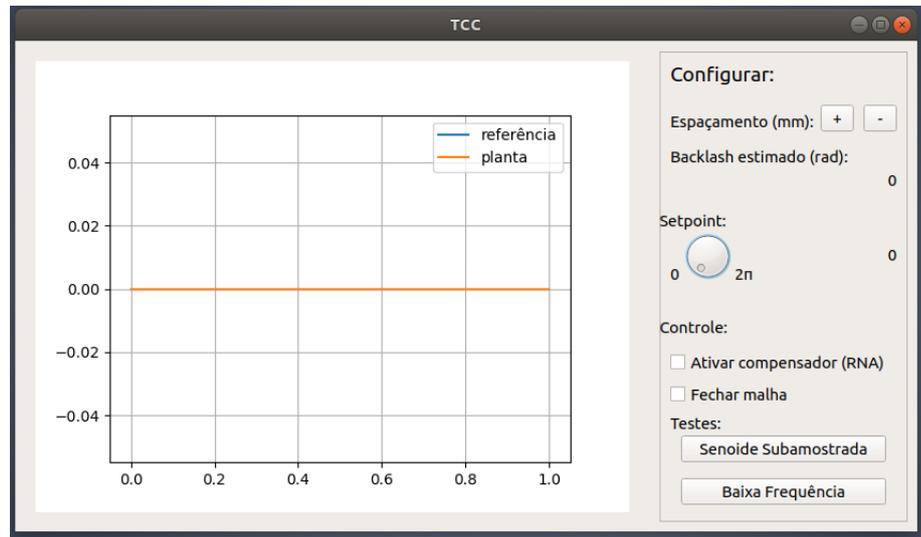


Figura 3.10: Captura de tela da interface gráfica do sistema

Funcionalidades

Para o controle manual do *setpoint* do sistema, um botão radial de cem divisões foi empregado, contendo valores no intervalo $[0, 2\pi]$. Dessa forma, é possível alterar o ponto de operação do sistema para valores de 0 até 2π avançando com um passo de $\frac{2\pi}{100}$. Para o acionamento automático, de sinais de validação de eficácia do compensador em MF, são usados botões nomeados "senoide subamostrada" e "baixa frequência", como pode ser visto na Figura 3.10. O acionamento do motor secundário foi feito através de dois botões que permitem uma rotação pré-definida, possibilitando as configurações discutidas anteriormente, vistas na Tabela 4.2. O fechamento da malha com a atuação do controlador (discutido na seção a seguir) e do compensador ocorreu com a marcação de *checkboxes*. Ainda na tela principal do programa desenvolvido, é mostrado um gráfico, atualizado com um atraso de T_s segundos, contendo o sinal de referência (*setpoint*) e da saída da planta.

3.7 Simulações e testes

Com todos os materiais definidos, ocorreram as simulações e testes do sistema. Uma vez já definidos os modelos da planta $G(s)$, o compensador $C(s)$, a topologia de controle,

desenvolvida e treinada a rede neural, a simulação aconteceu sem a efetiva construção da planta.

Foram comparados, para um mesmo valor de parâmetros do efeito da NL, as respostas temporais do sistema realimentado contendo e excluindo a pré-compensação pela rede neural. Os testes foram, em um primeiro momento, feitos com um sinal de entrada contendo diversas entradas em degrau, visando ressaltar o comportamento dinâmico do sistema. Em seguida, foram feitos testes com sinais de referências em baixas frequências, visando ressaltar erros de regime estacionário, diminuindo os efeitos dinâmicos do modelo.

Depois de testadas, as malhas foram comparadas com um sistema de controle ideal. Desta forma, foi possível perceber o efeito da NL na resposta do sistema, bem como a atuação da rede neural. Para realizar uma medida quantitativa, os erros relativos ao sistema ideal foram comparados usando a Equação (2.19) e o ganho η da compensação pela rede neural foi dado pela Equação (2.20).

3.8 Custos de projeto

A Tabela 3.8 a seguir mostra os custos estimados para a elaboração deste trabalho de conclusão de curso. Os parâmetros para seleção e estimativa de preços de cada componente da tabela se encontra justificada em sua respectiva apresentação no capítulo de metodologia deste texto.

Tabela 3.8: Tabela de custos (orçamento).

Produto	Custo (R\$)
Computador	
<i>Raspberry Pi 3 Model B</i>	196,99
Cartão de memória SD	38,90
Encoder	
LPD3806-400BM-G5-24C	134,90
Motor CC	
AK360-78.8PL12S7000S	15,90
Materiais	
Barra de alumínio 6061 $\phi 2'' \times 150$	20,00
Filamento ABS ou PLA	40,00
Total	446,69

Resultados e Discussões

4.1 Modelagem do sistema

4.1.1 Modelo do motor

Seguindo as etapas dispostas no capítulo de Desenvolvimento e selecionando o ponto de operação nominal, com as equações (2.3), (2.4) e (2.6), foi levantado o modelo da Equação (4.1):

$$G(s) = \frac{21.71}{s^2 + 7.495s} \quad (4.1)$$

4.1.2 Validação do modelo

A validação do modelo foi feita em posição e velocidade. A primeira contém o modelo da forma (4.1) e a segunda exclui o integrador. Com a validação do modelo em velocidade, é possível levantar características de comportamento do sistema, como o efeito do polo não dominante. A validação do modelo de posição pode ser vista na Figura 4.1, e a validação do modelo em velocidade (excluindo o integrador) pode ser vista na Figura 4.2.

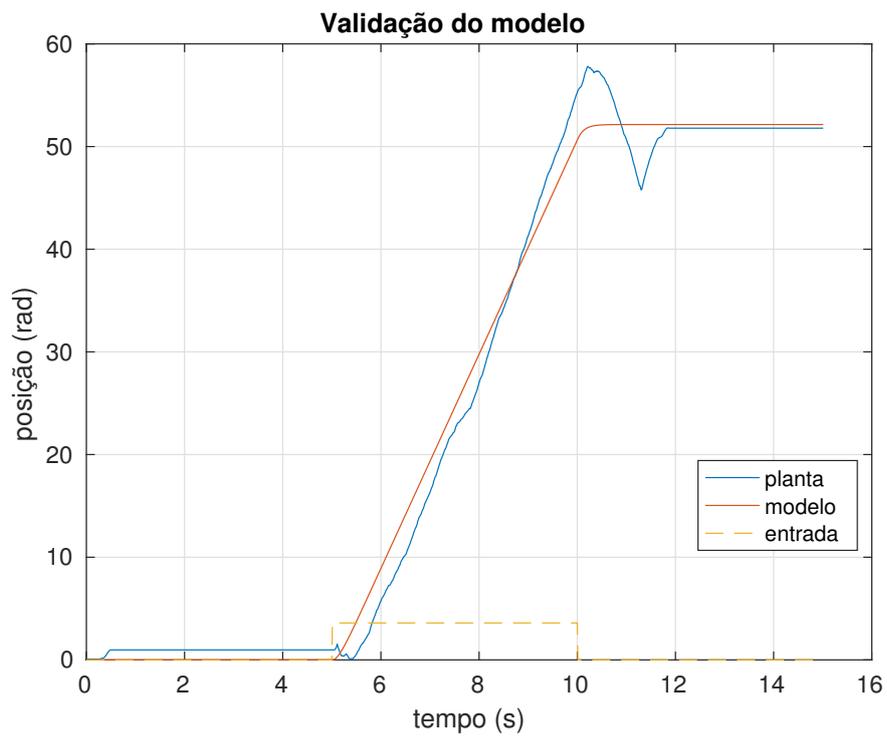


Figura 4.1: Validação do modelo

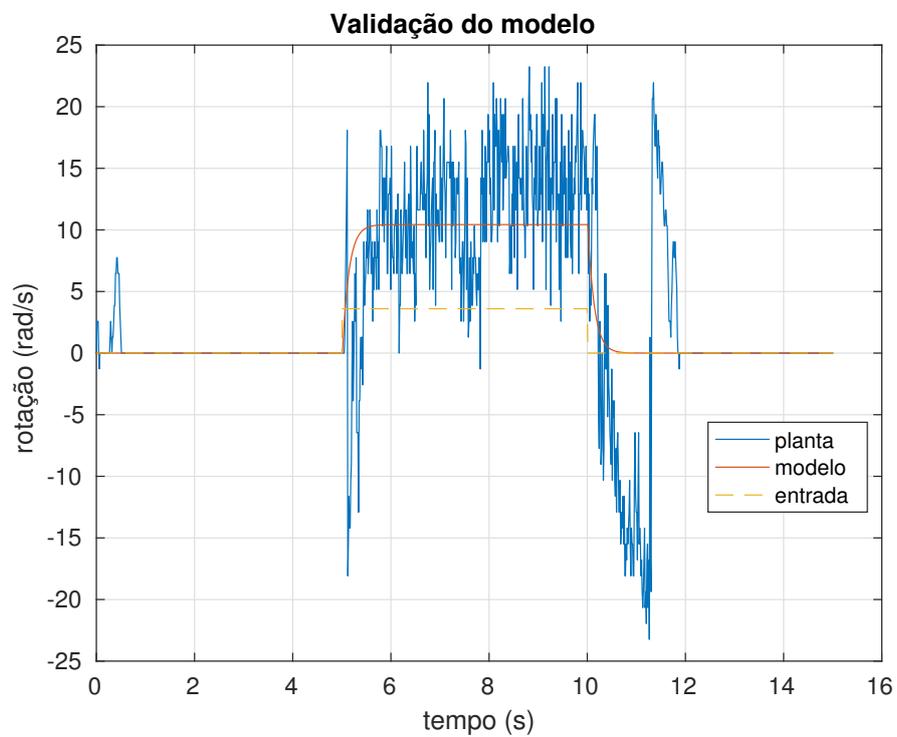


Figura 4.2: Validação do modelo (excluindo integrador)

4.1.3 Modelo da não-linearidade

Seguindo o modelo da Equação (2.10), e estimando a banda α da função representada na Equação (2.2) pela Equação (2.9), o modelo da NL em função do parâmetro j_r é dado pela Equação (4.2):

$$B(x) = \int D_z(\dot{x}, j_r) dt \quad (4.2a)$$

$$D_z(x, j_r) = \begin{cases} x + \frac{4 \tan \beta}{mz} j_r & x < -\frac{4 \tan \beta}{mz} j_r \\ x - \frac{4 \tan \beta}{mz} j_r & x > \frac{4 \tan \beta}{mz} j_r \\ 0 & |x| \leq \frac{4 \tan \beta}{mz} j_r \end{cases} \quad (4.2b)$$

4.1.4 Análise do sistema

Acerca do modelo descrito na Equação (4.1), foi possível perceber que, devido ao integrador presente no sistema, é garantido erro de estado estacionário nulo para entradas em degrau, com o sistema em malha fechada. Sobre o modelo de velocidade (excluindo a ação do integrador), foi possível perceber que o tempo de acomodação do sistema tem valor de $t_s = 0.53$ segundos. Ainda em MF, é possível perceber que o sistema se acomoda em 0.822 segundos apresentando sobressinal de 7%, como mostra a Figura 4.3.

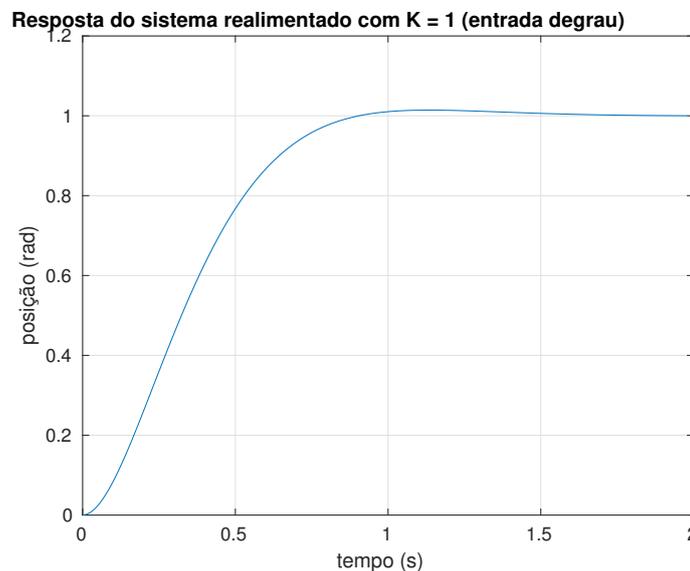


Figura 4.3: Resposta do modelo em realimentação para entrada degrau

4.1.5 Características de desempenho

Acerca dos modelos em malha aberta e análise dos regimes transitórios e permanentes do sistema, foram levantadas as seguintes características de desempenho desejadas para o sistema em malha fechada com a adoção de um controlador:

- Tempo de acomodação de $t_s = 0.53$ segundos;
- Erro de estado estacionário nulo para entradas em degrau;
- Sobressinal máximo de 20%.
- Menor tempo de subida possível.

4.2 Projeto de compensadores

4.2.1 Projeto do controlador

Para o modelo da Equação (4.1), foi implementado um controlador desejando obter as características de desempenho discutidas anteriormente. Foi obtido o controlador:

$$C(s) = 12.15 \frac{s + 7.49}{s + 14.98} \quad (4.3)$$

Para avaliar o controlador, foi medido, para uma entrada $u(t) = e^{-0.1t} \cos(0.3t^2)$, no intervalo $t \in [0, 10]$ segundos, os valores de IAE, ITAE e ISE da resposta em malha fechada, contendo e excluindo a não-linearidade simulada $B(u)$ com $\alpha = 0.048$. Também é mostrada a eficiência η da Equação (2.20) comparando o sistema em relação a não-linearidade.

Tabela 4.1: Tabela de índices de desempenho do controlador em malha fechada

$\alpha = 0.048$	Contendo NL	Excluindo NL
IAE	236.1014	222.4738
ISE	99.7544	89.4892
ITAE	1438.9	1365.9
η	0.0577	1.000

A Tabela 4.1 mostra os índices de desempenho do controlador. O valor de $\eta = 5.77\%$ indica que, para o caso crítico de *backlash*, o controle em malha fechada não atinge os critérios desejados, embora o controlador tenha sido corretamente projetado.

4.2.2 Discretização do controlador

Escolha do período de amostragem

O período de amostragem foi obtido através do modelo contínuo da planta, mostrado na Equação (4.1) Adotando $T_s = 0.067$ segundos, Os polos do modelo discreto da planta estão localizados em $z_0 = 1$ e $z_1 = 0.6052$.

Discretização do controlador

Discretizando o controlador da equação (4.3), foi obtido

$$C(z) = 12.15 \frac{z - 0.6833}{z - 0.3665} \quad (4.4)$$

Sabendo que $C(z)$ representa a relação entre o sinal de controle u_k e o erro e_k no domínio da frequência discreta, foi obtida a equação de diferenças, descrita na Equação (4.5):

$$u_k = 0.3665u_{k-1} + 12.15e_k - 8.3012e_{k-1} \quad (4.5)$$

A implementação do controlador pode ser vista no Código-fonte A (consultar Apêndice).

4.3 Implementação de rede neural

4.3.1 Sinal de entrada

O sinal de entrada escolhido para treinamento da RNA foi o disposto na Equação (4.6):

$$u(t) = e^{-0.1t} \cos(0.3t^2) \quad (4.6)$$

Analisando o sinal, é possível notar que a amplitude do sinal diminui ao longo do tempo t , ao passo em que a frequência de oscilação do termo $\cos(0.3t^2)$ aumenta linearmente. A derivada do sinal $\dot{u}(t)$, portanto, também aumenta, enquanto a amplitude do sinal diminui. Dessa forma, o efeito do *backlash* pode ser melhor estudado, apresentando maiores quantidades de respostas esperadas. De acordo com os métodos de Resposta em Frequência, a resposta $y(t)$ de um sistema a uma entrada $u(t)$ altera apenas as características de fase e ganho da entrada, não alterando a frequência do sinal.

4.3.2 Desenvolvimento da RNA

Primeira versão

Para a elaboração da primeira versão do compensador, foi implementada uma rede neural de $L = 4$ camadas, com pesos de dimensões $W_0 \in \mathbb{R}^{(1 \times 50)}$, $W_1, W_2 \in \mathbb{R}^{(50 \times 50)}$, e $W_3 \in \mathbb{R}^{(50 \times 1)}$, tendo $i = o = 1$. Nas camadas 0 até 3, as funções de ativação f_k foram do tipo PReLU (*parametric rectified linear unit*, do inglês "unidade linear retificada parametrizada"), mostrada na Equação (3.3). Na última camada, foi usada uma multiplicação por um vetor escalar, também aprendido pela rede.

O algoritmo de treinamento foi o do Método do Gradiente Descendente, e a função de custo a ser minimizada foi $J(y, \hat{y}) = \ln \cosh(y - \hat{y})$, com propriedades demonstradas pelo exemplo 2.5. A Figura 4.4 mostra a saída do sistema contendo e excluindo o compensador para uma simulação da planta em malha aberta, com um sinal aleatório de entrada. Para a primeira versão, a média dos resíduos da RNA foi de $Res\{F_{v1}\} = 0.85434747$.

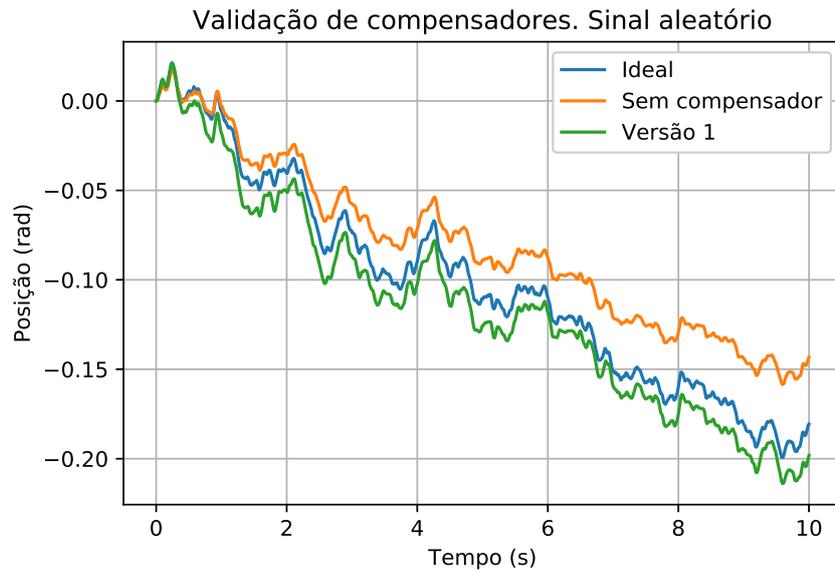


Figura 4.4: Saída do sistema contendo e excluindo RNA (versão 1, simulação)

Segunda versão

Na segunda versão, a média dos resíduos da RNA calculada foi de $Res\{F_{v2}\} = 0.7764067$. Na segunda versão do compensador, os algoritmos de treinamento e função de custo foram mantidos. A Figura 4.5 mostra a saída do sistema contendo e excluindo o compensador para uma simulação da planta em malha aberta, contendo a RNA na segunda versão.

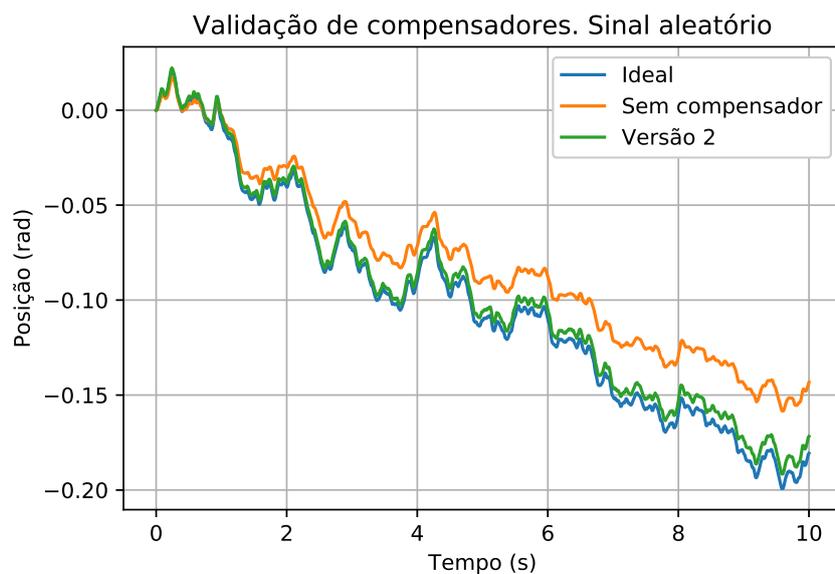


Figura 4.5: Saída do sistema contendo e excluindo RNA (versão 2, simulação)

Comparação de compensadores

Para comparar a compensação entre as RNAs com um sinal de controle fixo, foi usado o sinal da equação (4.6). A saída pode ser vista na Figura 4.6.

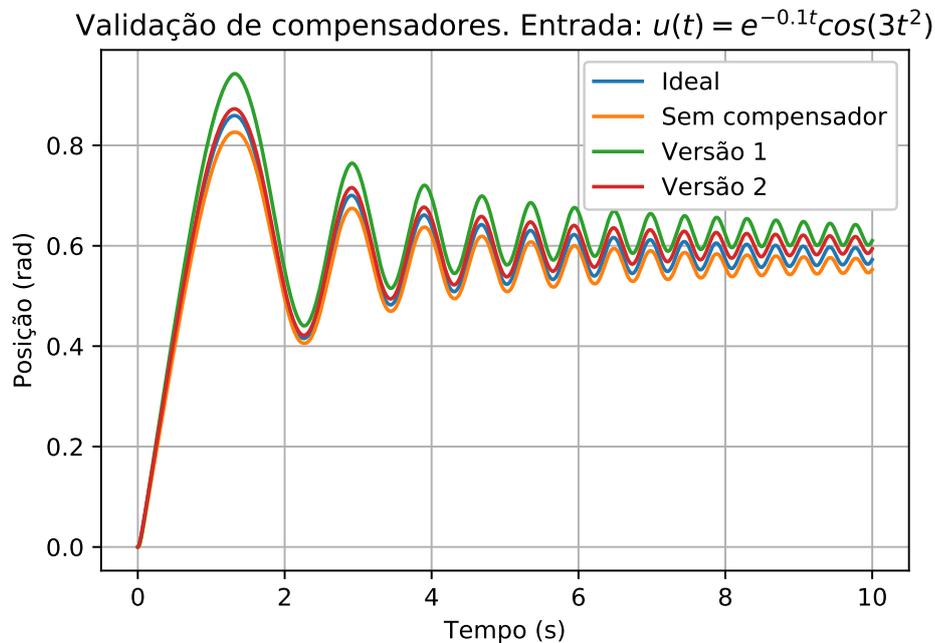


Figura 4.6: Comparação de saídas entre compensadores.

É possível notar que o sinal do compensador da segunda versão (em vermelho) se aproxima da saída esperada (ideal, em azul) com maior qualidade, quando comparada com a primeira versão (verde). Numericamente, o resultado pode ser interpretado por $Res\{F_{v2}\} < Res\{F_{v1}\}$. Os códigos-fonte dos algoritmos de *forward propagation* podem ser vistos no capítulo de Apêndice.

4.4 Construção da planta

4.4.1 Fabricação e montagem da planta

Após os processos de formulação do mecanismo, projeto e fabricação das peças, foi construída a versão física do sistema, representada na Figura 4.7. Para acionamento dos motores, uma ponte-H L298N foi utilizada, em conjunto com o computador Raspberry Pi 3.

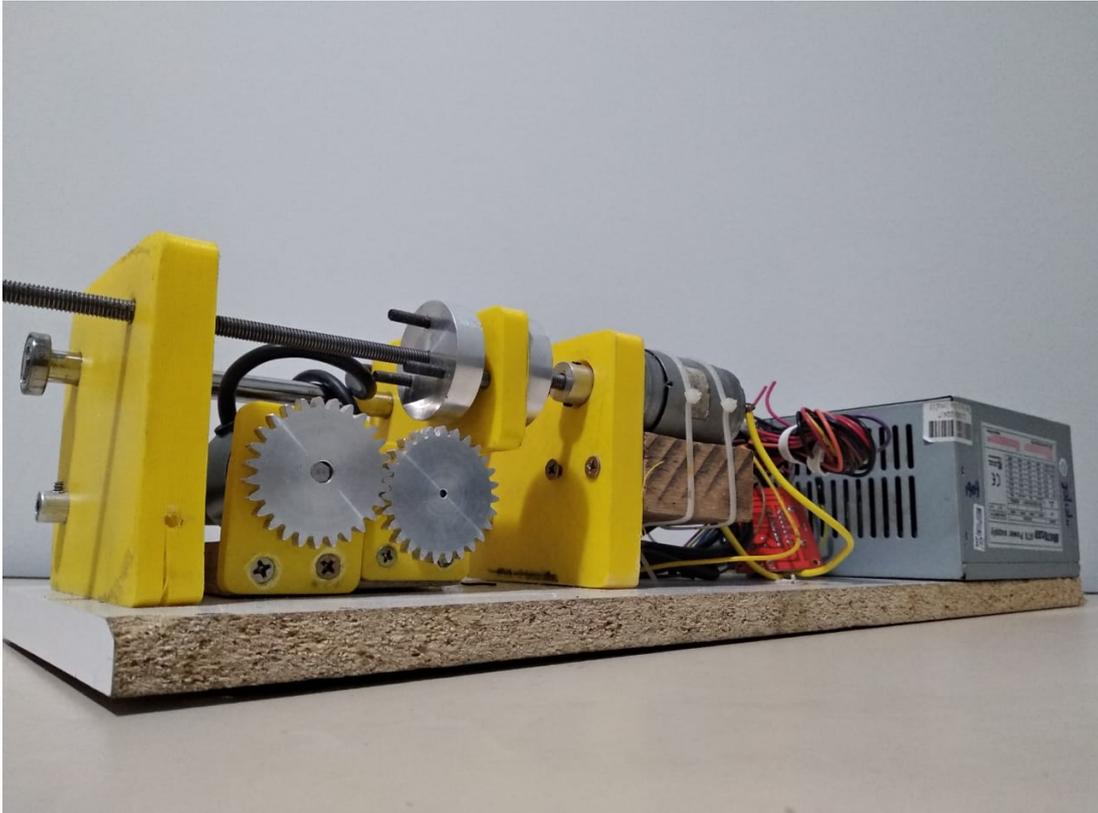


Figura 4.7: Sistema físico de transmissão mecânica com *backlash* ajustável

4.4.2 Análise de Restrições

Através da análise das Equações (3.1) e (3.2), juntamente com os valores da Tabela 3.1, foi obtido o valor da folga ganha por deslocamento realizado pelo parafuso do mecanismo de afastamento, mostrado na Equação (4.7):

$$\Delta j_{\theta} = 1,126 \cdot 10^{-2} \text{ rad/pulso.} \quad (4.7)$$

Medindo experimentalmente $j_{\theta 0}$ após a montagem do mecanismo e pelos parâmetros da engrenagens, mostrados na Tabela 2.1, foram obtidos os valores limites de folgas entre o par de engrenagens, representados na inequação 4.8:

$$1,438 \cdot 10^{-2} \leq j_{\theta} \leq 5,392 \cdot 10^{-2} \quad (4.8)$$

Dividindo $j_{\theta M} - j_{\theta 0}$ por Δj_{θ} , temos aproximadamente quatro pulsos possíveis para levar o sistema, o que significa quatro diferentes configurações de *backlash*, detalhados na Tabela 4.2.

Tabela 4.2: Configurações de folgas estimadas na planta

Configuração	j_{θ} (10^{-2} rad)	Informação
1	1,438	Backlash mínimo permitido
2	2,564	Configuração 2
3	3,689	Configuração 3
4	4,816	Backlash máximo permitido

4.5 Simulações e análises

Para o modelo do sistema $G(s)$ da Equação (4.1) realimentado com o controlador $C(s)$ da equação (4.3) em série, foram feitas simulações aplicando dois sinais de entrada no sistema contendo e excluindo a NL representada na Equação (4.2).

4.5.1 Parâmetros

Em cada caso de simulação, foram estabelecidos parâmetros de entrada do sistema. Foram aplicados dois sinais de entrada, denotados $u(t)$ e $v(t)$, respectivamente, descrito nas equações (4.9a) e (4.9b). O fator de *backlash* α foi arbitrado em 0.022 radianos, e o tempo de simulação foi de $t_F = 10$ segundos.

$$u(t) = \frac{\alpha}{2} \sin_{1,2}(0.25\pi t) \quad (4.9a)$$

$$v(t) = \alpha \sin(0.1\pi t), \quad (4.9b)$$

em que $\sin_k(x)$ é a função seno subamostrada com período de amostragem de k segundos. Dessa forma, o sinal $u(t)$ se comporta como uma sequência de entradas em degrau, servindo como parâmetro de análise do regime transitório do sistema. Já $v(t)$ representa uma entrada de baixa frequência.

4.5.2 Simulações

Caso 1 - Malha Fechada sem compensador

Na primeira simulação, foram aplicados os sinais $u(t)$ e $v(t)$ nos modelos da planta e do controlador, contendo e excluindo a NL da Equação (4.2). A partir do caso ideal (sem o efeito simulado do *backlash*), foram levantados os índices de desempenho IAE, ISE e ITAE. As respostas do sistema e os índices de desempenho podem ser vistos, respectivamente, na Figura 4.8 e Tabela 4.3.

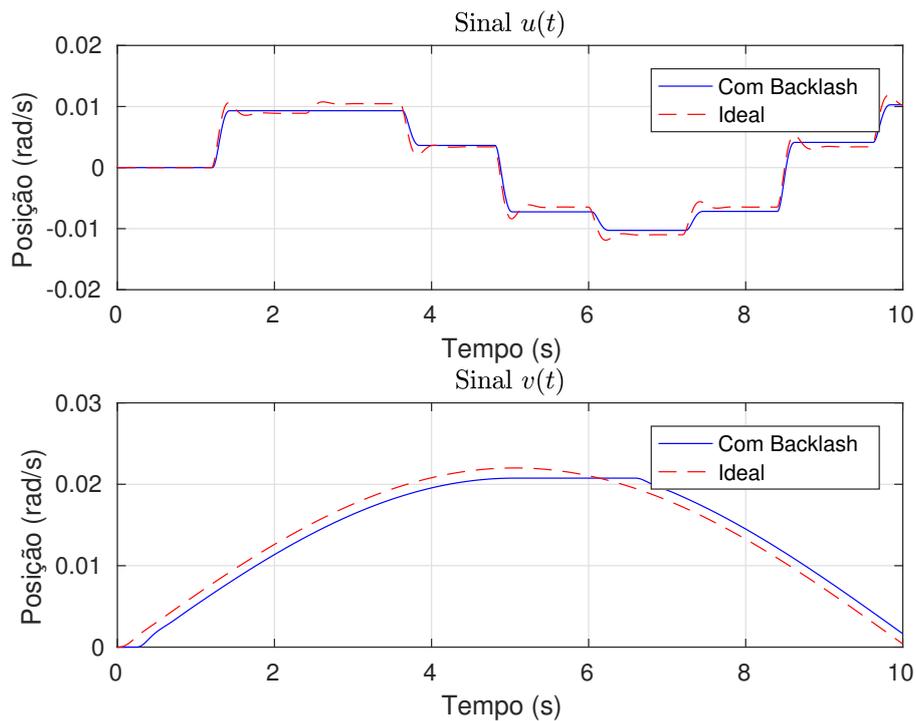


Figura 4.8: Respostas do sistema em Malha Fechada sem compensador

Tabela 4.3: Índices de desempenho (caso 1, simulação)

	IAE	ISE	ITAE
$u(t)$	65.1988	4.5627	393.1892
$v(t)$	70.6597	5.6040	475.5493

Foi possível notar, pelos valores de IAE, ISE e ITAE, que o efeito da não-linearidade provoca alterações na resposta esperada do sistema. O primeiro conjunto de dados foi usado como referência para avaliar o desempenho dos compensadores.

Caso 2 - Malha Fechada com primeira versão do compensador

Adicionando a primeira versão do compensador, foram simuladas as saídas do sistema contendo a não-linearidade da Equação (4.2) para os sinais $u(t)$ e $v(t)$. Além dos índices de desempenho anteriores (IAE, ISE e ITAE), foi medido o índice η da Equação (2.20) para comparar o ganho obtido com a implementação do compensador. A Figura 4.9 mostra os sinais de saída do sistema e a Tabela 4.4 apresenta os índices de desempenho obtidos.

Tabela 4.4: Índices de desempenho (caso 2, simulação)

	IAE	ISE	ITAE	η
$u(t)$	46.2776	3.7358	268.3382	0.2902
$v(t)$	13.5726	0.1937	71.4699	0.8079

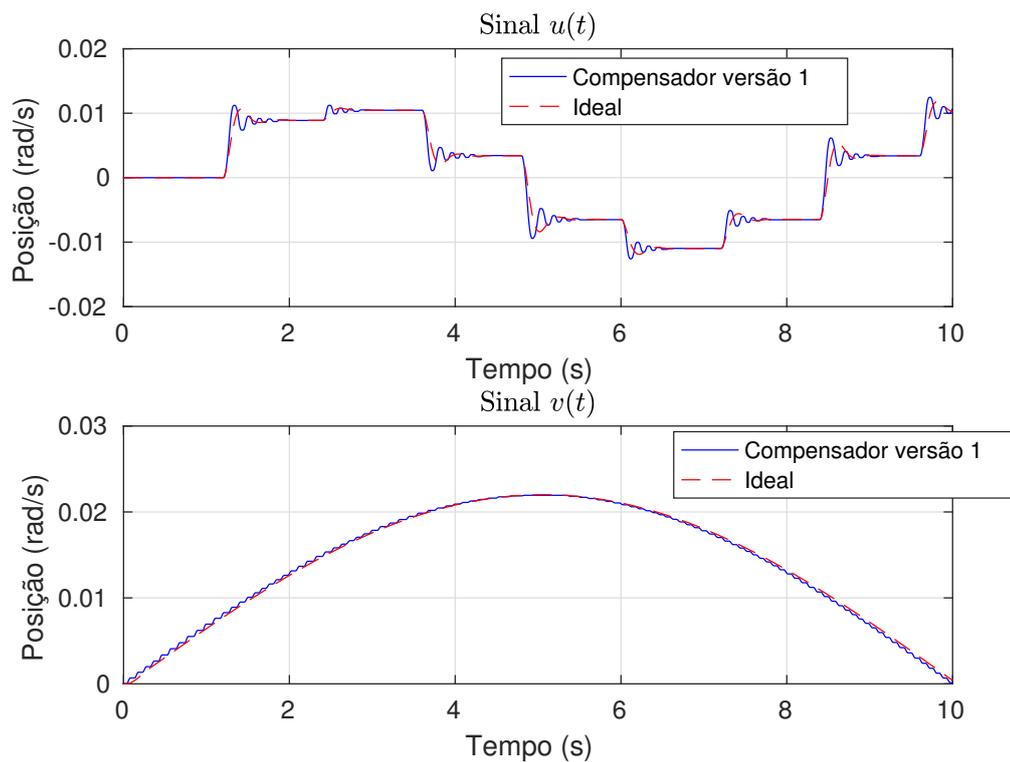


Figura 4.9: Respostas do sistema em Malha Fechada com primeira versão do compensador

Com a adição do compensador, os índices de desempenho relativos ao erro (IAE, ISE e ITAE) foram diminuídos, comprovando o aumento da eficiência do controle em malha fechada. O ganho obtido, em relação ao sistema sem compensador foi de $\eta = 29.02\%$ para o sinal $u(t)$ e $\eta = 80,79\%$ para o sinal $v(t)$.

Caso 3 - Malha Fechada com segunda versão do compensador

Para a segunda versão do compensador, os testes com os sinais $u(t)$ e $v(t)$, bem como o valor de α e tempo t_F de simulação foram repetidos. A resposta do sistema às entradas e os índices de desempenho calculados podem ser vistos na Figura 4.10 e Tabela 4.5, respectivamente.

Tabela 4.5: Índices de desempenho (caso 3, simulação)

	IAE	ISE	ITAE	η
$u(t)$	53.2828	2.7656	267.3895	0.1828
$v(t)$	24.6889	0.7542	151.1802	0.6506

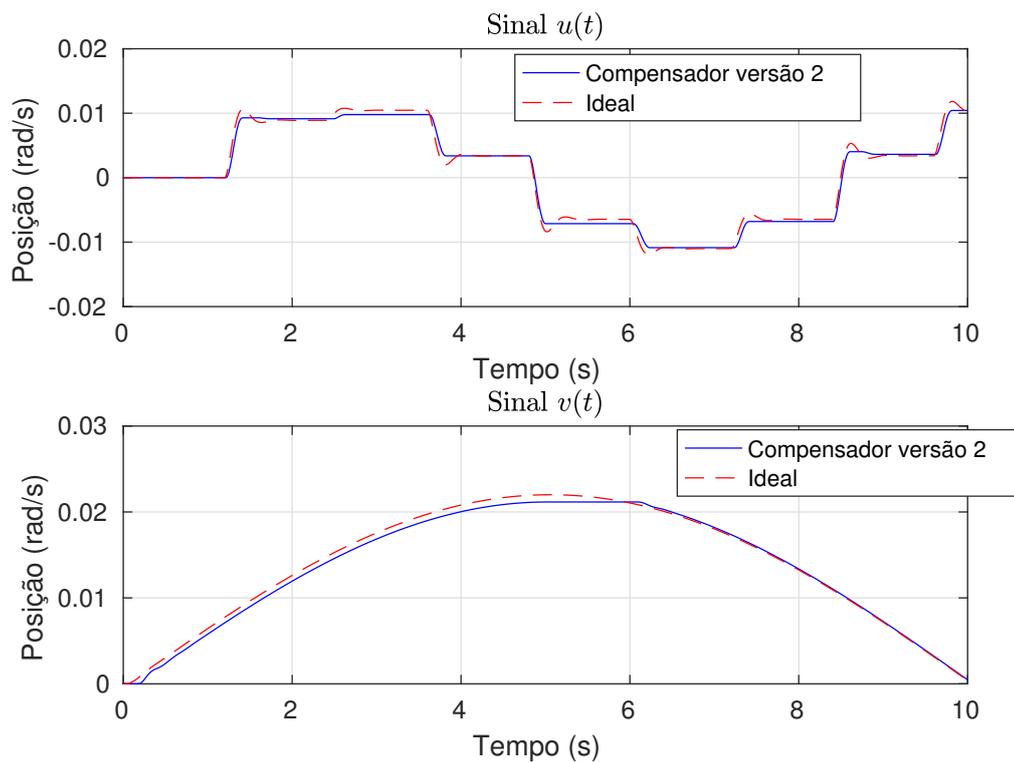


Figura 4.10: Respostas do sistema em Malha Fechada com segunda versão do compensador

Acerca das informações contidas na Tabela 4.5, o ganho η calculado pela Equação 2.20 obtido foi de $\eta = 18.28\%$ para o sinal $u(t)$ e de $\eta = 65,06\%$ para o sinal $v(t)$. Comparando o parâmetro η com a primeira versão da RNA, é possível notar uma queda de desempenho de 10,74% para o sinal $u(t)$ e 15.73% para o sinal $v(t)$. Contudo, o tempo de computações durante o período de simulação foi drasticamente reduzido para a segunda versão, comparada com o primeiro compensador desenvolvido.

4.5.3 Testes

Para os testes, o segundo caso (malha fechada com primeira versão do compensador) não foi implementado na planta, devido ao elevado tempo de computação necessário para cálculo de *forward propagation* da RNA.

Caso 1 - Malha fechada sem compensador

No primeiro teste com a planta, foram aplicados os sinais $u(t)$ e $v(t)$. A partir do caso ideal, realizado com o modelo da planta mostrada na Equação (4.1), foram levantados os índices de desempenho IAE, ISE e ITAE. As respostas do sistema e os índices de desempenho podem ser vistos, respectivamente, na Figura 4.11 e Tabela 4.6.

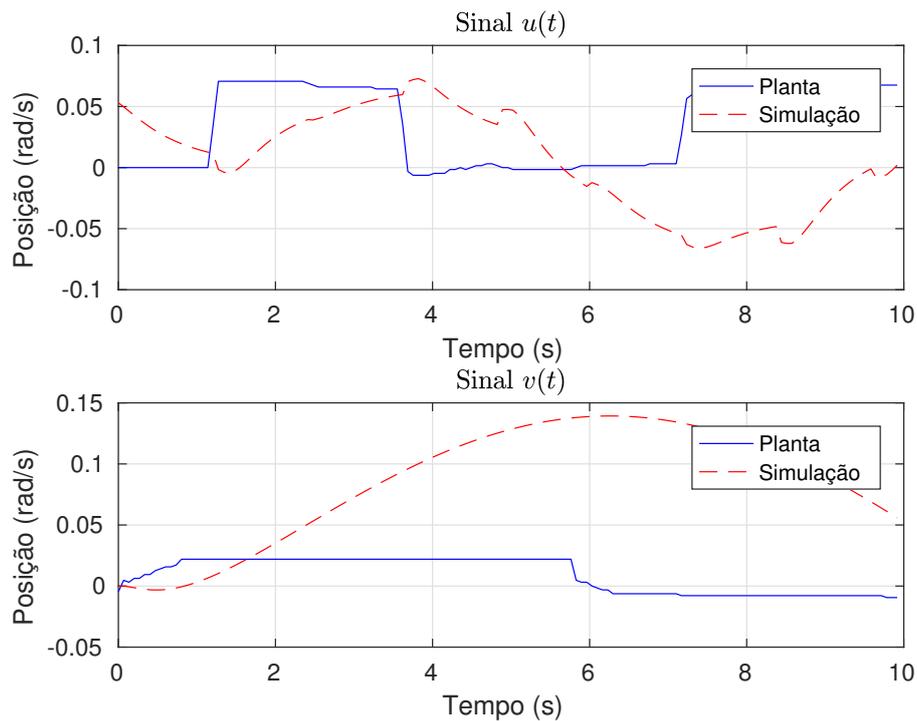


Figura 4.11: Respostas do sistema em Malha Fechada sem compensador

Tabela 4.6: Índices de desempenho (caso 1, teste)

	IAE	ISE	ITAE
$u(t)$	76.0142	50.4699	401.0391
$v(t)$	84.9378	60.7014	423.6172

Foi possível notar, pelos valores de IAE, ISE e ITAE, comparados com a Tabela 4.3, que o efeito da não-linearidade causa alterações ao modelo, como visto nos resultados em simulação.

Caso 2 - Malha fechada com compensador

Aplicando a segunda versão do compensador, foram levantados índices de desempenho do controle da planta e verificada sua resposta temporal. A Tabela 4.7 e a Figura 4.12 mostram os índices de desempenho e a resposta da planta.

Tabela 4.7: Índices de desempenho (caso 2, teste)

	IAE	ISE	ITAE	η
$u(t)$	75.0978	49.3741	410.3932	0.0121
$v(t)$	81.1956	55.9421	393.5092	0.0441

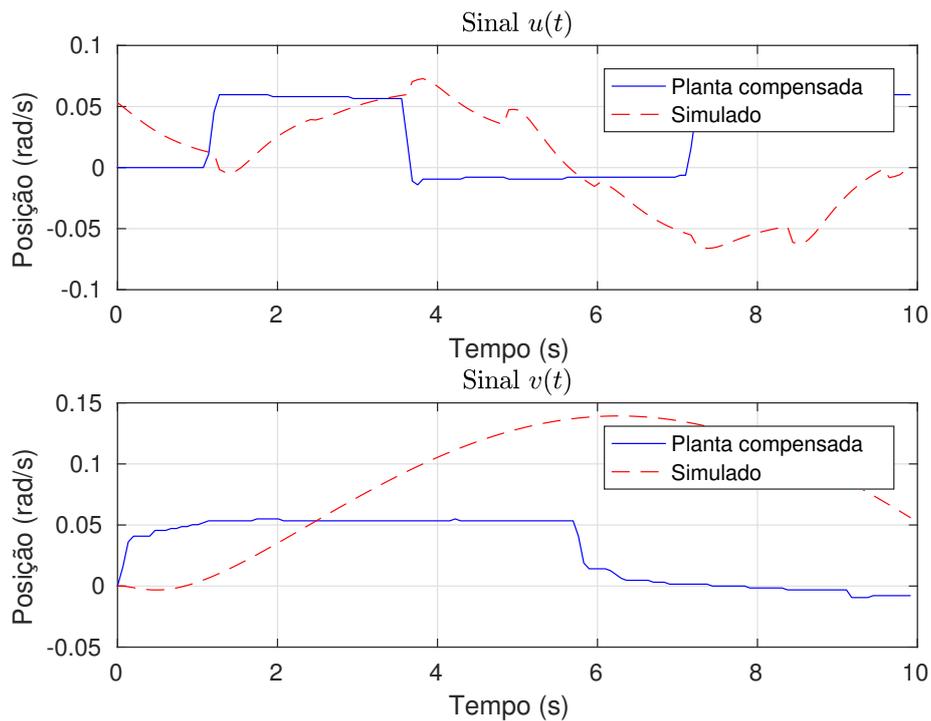


Figura 4.12: Respostas do sistema em Malha Fechada com segunda versão do compensador

Os resultados da Tabela 4.7 mostram aumento de eficiência da RNA de $\eta = 1.21\%$ para o primeiro sinal ($u(t)$) e $\eta = 4.41\%$ para o sinal $v(t)$. Devido a discrepância entre o ganho obtido na compensação da RNA em simulação e nos testes realizados, faz-se necessário levantar hipóteses que justifiquem o efeito.

Embora os valores sejam menores do que os encontrados em simulação, o efeito pode ser explicado por outros tipos de não-linearidades, como atrito (vindo entre as engrenagens, entre engrenagem e eixo do sensor/atuador, do eixo do sensor/atuador, etc.); zona morta do sensor e atuador; tensões de *ripple* da fonte ATX; entre outros. A contribuição de NLS não modeladas, quando acumuladas, desencadeiam efeitos prejudiciais ao controle do sistema.

A amplitude do sinal de controle também pode ser considerada um fator crítico para a redução da eficiência da RNA, uma vez que apresenta valores no intervalo $|u| \leq \alpha$ da Equação (2.2). A baixa amplitude no sinal foi arbitrada como forma de avaliar, em plena atuação, o comportamento da não-linearidade para o controle do sistema.

Imprecisões no modelo da dinâmica linear e da não-linearidade podem contribuir com a discrepância notada. Uma vez que o treinamento da rede neural foi executado em malha aberta com os modelos simulados, a mesma se torna dependente de seus parâmetros, diminuindo sua robustez e obtendo maiores propensões para erros no controle do sistema.

Considerações Finais

O seguinte capítulo descreve as considerações finais realizadas sobre o desenvolvimento deste trabalho de conclusão de curso. São apresentadas as principais conclusões do trabalho; objetivos alcançados e propostas de continuidade do projeto.

5.1 Conclusões

A compensação de sistemas mecânicos se apresenta como um desafio de considerável dificuldade. As não-linearidades presentes em sistemas podem causar inadequações aos critérios de projeto dispostos pelo processo do projeto do controlador. De forma a corrigir os efeitos das não-linearidades, este trabalho propôs o uso de uma rede neural artificial para compensação de uma não-linearidade presente em um sistema mecânico. Assumindo que o efeito possa ser modelado através de um conjunto de funções, foi usada a propriedade de aproximação universal da RNA para encontrar sua função inversa, alterando o sinal de controle provido pelo compensador. Dessa forma, a excitação da não-linearidade é abrandada pela modulação do sinal de controle pela inversa.

O trabalho de conclusão de curso apresentou uma solução de compensação do efeito do *backlash* presente em um sistema de transmissão mecânica. Para a realização da tarefa, foram necessários o levantamento de modelos lineares e não-lineares do sistema; projeto e fabricação de componentes mecânicos (par de engrenagens, porcas anti-*backlash*, suportes, etc.); projetos e implementações de redes neurais artificiais e controladores digitais baseados em funções de transferência em tempo discreto e o levantamento de índices de desempenho capazes de medir o aumento da eficiência do controlador apresentado pela adição do compensador.

Este projeto foi concebido inicialmente percebendo a dificuldade apresentada por estudantes de graduação em Engenharia Mecatrônica no controle de sistemas mecânicos, devido à construção de plantas com peças de baixa qualidade, projetadas, fabricadas e montadas com imperícia. Contudo, o problema pode ocorrer, com menor efeito, em sistemas industriais, robóticos, automobilísticos, etc. O projeto de pré-compensadores baseado em redes neurais

artificiais apresenta uma alternativa relativamente simples, quando equiparada ao Controle Ótimo e Controle Robusto, para mitigação dos efeitos não lineares.

A adição do compensador obtido pelo emprego de redes neurais artificiais se mostrou eficiente para o melhoramento do controle do sistema em malha fechada. Em simulações, foram obtidos índices que comprovam melhoras de até 80,79% do desempenho do controlador, aproximando a resposta da dinâmica ideal. EM testes com a planta, a contribuição efetiva máxima da RNA foi de 4,41%.

5.2 Objetivos alcançados

Dos objetivos específicos listados pelo autor (conforme seção de introdução), até a data de publicação deste relatório, foram alcançados:

- Desenvolver, em ambiente simulado, um sistema mecânico, ressaltando uma de suas não-linearidades intrínsecas.
- Implementar e treinar uma rede neural para pré-compensação da não-linearidade escolhida.
- Projetar um controlador para o sistema mencionado, explorando a não-linearidade abordada.
- Reproduzir o sistema simulado com componentes físicos (construção da planta).
- Comparar desempenho da planta para o mesmo controlador com e sem o emprego da rede neural.

5.3 Propostas de continuidade

Com a finalidade de investigar a compensação de efeitos não-lineares em sistemas mecânicos e avaliar o desempenho do controlador com a adição do compensador, são apresentadas propostas de continuidade. As seguintes atividades podem ser executadas para contribuir com o aumento da eficiência do controle em malha fechada de sistemas não-lineares.

Para capturar a dinâmica da planta com menores erros de modelagem, funções de transferências de maiores ordens, contendo zeros ou atrasos podem ser levantadas. Um modelo mais complexo e preciso permite análises de regime transitório e permanente com maior acurácia. Em outra possível abordagem, a modelagem por espaço de estados pode ser feita. Nesse modelo, as técnicas de treinamento da rede neural podem ser investigadas em relação ao tempo de treinamento e aumento da eficiência da compensação ao controle em malha fechada.

O levantamento de modelos mais precisos, considerando, por exemplo, o efeito dinâmico das engrenagens, permite projetos de controladores de diferentes ordens e novas topologias de controle. Nesse sentido, métodos para projetos de controladores não abordados nesse trabalho podem ser feitos, como o Método de Ackerman para realimentação de estados. Caso o modelo levantado contenha atrasos ou efeitos de integradores, diferentes topologias de controle, como um Preditor de Smith ou técnicas de anti *wind-up* podem ser usadas.

Outras versões de redes neurais com características de aproximações dinâmicas, como as redes LSTM ou RNNs (*Recursive Neural Networks*) podem ser aplicadas para melhor aproximação dos efeitos não-lineares do sistema. Dessa forma, o compensador pode ser treinado para aproximar não somente as características estáticas da não-linearidade, como proposto no Teorema da Aproximação Universal, mas também saídas que dependam de estados anteriores. Unindo essa abordagem com a modelagem por espaço de estados, o compensador pode ser treinando não apenas para mitigar o efeito da não-linearidade do sistema, mas também prover coeficientes para a realimentação dos estados da planta, podendo agir mutuamente como controlador e compensador.

Apêndice

O seguinte capítulo apresenta dados referenciados em capítulos anteriores do texto, organizados em tabelas, figuras e códigos-fonte.

```
class Controle:

    def __init__ (self, GPIOAtuador, GPIOSensor, vSat = 12.):
        # Controle dos componentes via GPIO:
        self.Atuador = Output (GPIOAtuador)
        self.Sensor = Input (GPIOSensor)
        # Sinais de controle anteriores:
        self.u_k1 = 0.; self.e_k1 = 0.;
        # Referencia a ser seguida:
        self.referencia = 0.

    def aplicar (self):
        # Verifica se há dado a ser processado:
        if self.dadosAProcessar == True:
            # Calcula erro e_k:
            e_k = self.referencia - self.Sensor.valor;
            # Lei de controle:
            u_k = (0.3665 * self.u_k1) + (12.15 * e_k)
            u_k = u_k - (8.3012 * self.e_k1)
            # Sobrescreve valores:
            self.u_k1 = u_k
            self.e_k1 = e_k
            # Saturação:
            if abs (u_k) >= vSat:
                u_k = vSat * u_k / abs (u_k)
            # Aplica no atuador:
            self.Atuador.aplicar (u_k)
            # Apaga dados a processar:
            self.dadosAProcessar = False
```

Código-fonte 1: Código de execução do controlador

```

class RNA:

    def __init__(self, layers):
        # Cria lista de pesos e funções de ativação:
        self.W = []; self.B = [];
        self.f = []; self.b = [];

        # Adiciona camadas de ativação:
        for k, L_k in enumerate (layers):
            self.W[k] = L_k['weight']
            self.B[k] = L_k['bias']
            self.f[k] = L_k['func']
            self.p[k] = L_k['param'] # beta (v.1), alpha (v.2)
        # Salva número de camadas:
        self.numLayers = len (layers)

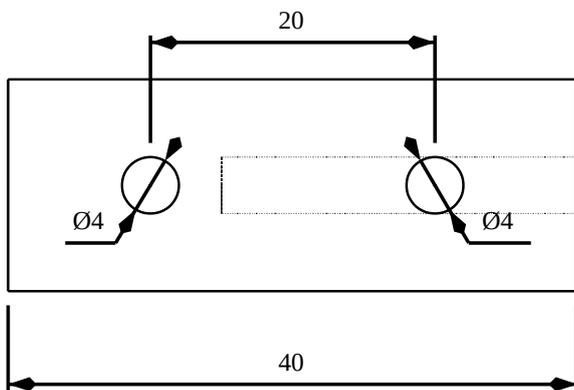
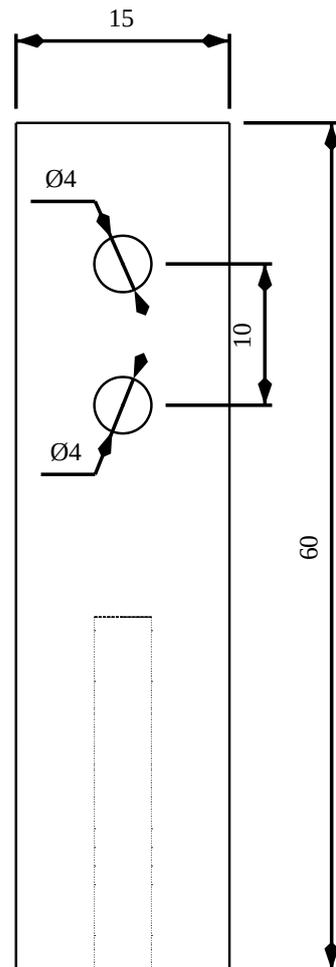
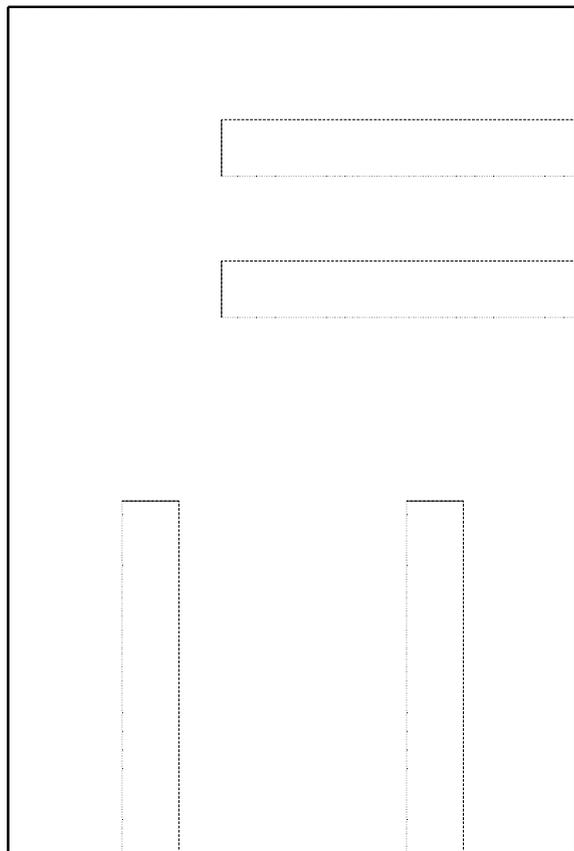
    def forwardPropagation (self, u_k):
        # Iteração 0: copia sinal de entrada:
        z_k = u_k
        # Iterações 1...N: multiplica pelos pesos
        # e aplica na função de ativação:
        for k in range(self.numLayers):
            z_k = np.dot (z_k, self.W[k]) + self.B[k]
            z_k = self.f[k] (z_k, self.p[k])
        # Retorna valor final:
        return z_k

```

Código-fonte 2: *Forward Propagation* da RNA

Tabela A.1: Relação de materiais e processos de fabricação

Nº	Nome	Processos	Materiais			
			ABS	Alumínio 6061	Madeira	Outros
2	Base para motor A	Furação			x	
3	Suporte frontal para motor A	Impressão 3D	x			
9	Carro do motor A	Impressão 3D	x			
4	Engrenagem Pinhão	Torneamento Fresamento Furação Roscamento		x		
10	Porca anti- <i>backlash</i> não rosqueada nos furos auxiliares	Torneamento Furação Roscamento		x		
11	Parafusos MA M4 para furo auxiliar					x
12	Molas de compressão					x
13	Porca anti- <i>backlash</i> rosqueada nos furos auxiliares	Torneamento Furação Roscamento		x		
14	Barra roscada MA M6					x
16	Acoplador para motor B	Torneamento Furação Roscamento		x		
18	Suporte lateral com furo para motor B	Impressão 3D	x			
17	Base para motor B	Furação			x	
19	Suporte lateral com furo para barra M6	Impressão 3D	x			
6	Base para encoder	Furação			x	
7	Suporte frontal para encoder	Impressão 3D	x			
8	Engrenagem coroa	Torneamento Fresamento Furação Roscamento		x		
20	Eixos-guia					x



Autor:
Matheus D. Azevedo

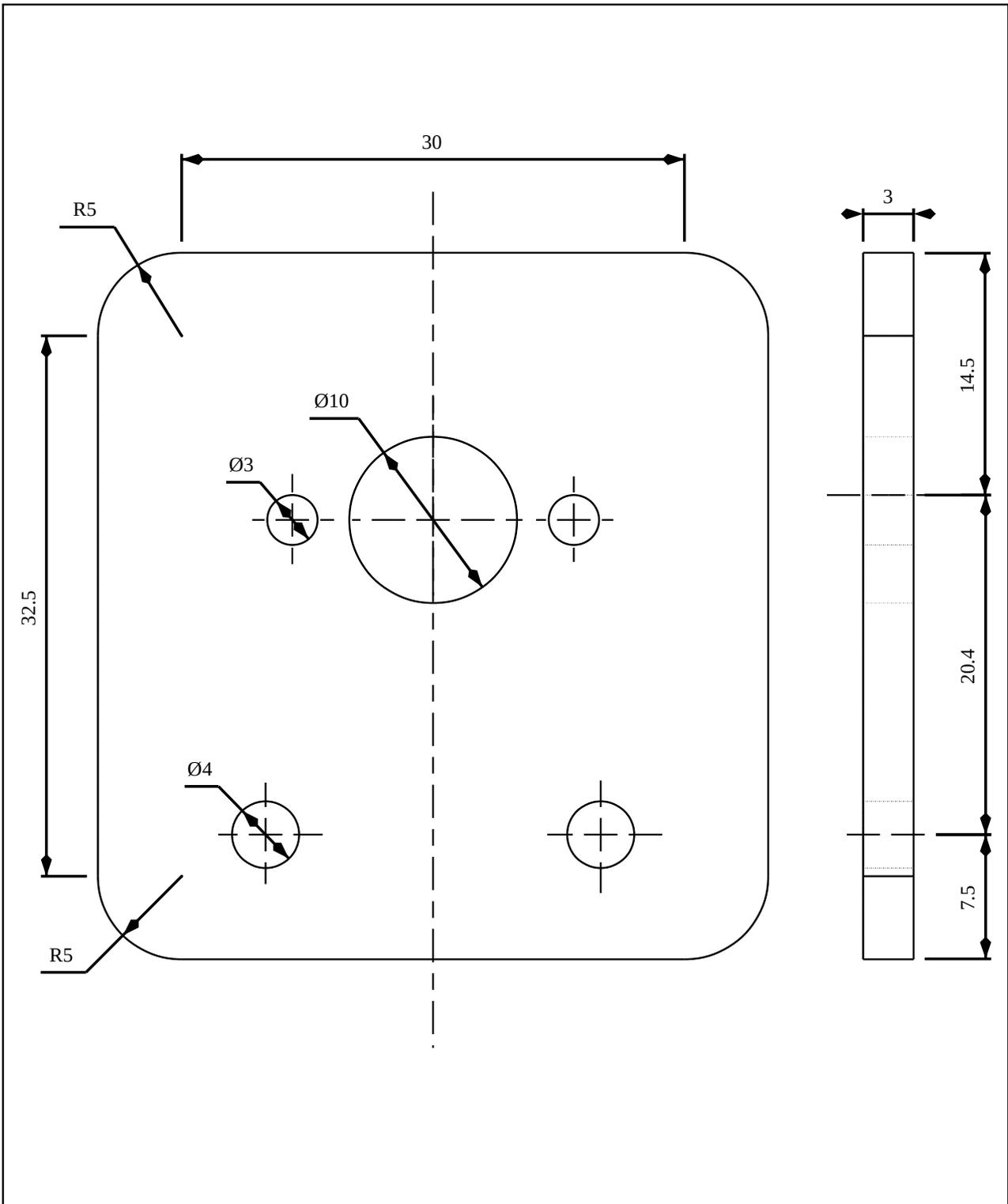
Título:
Base para Motor A

Descrição:
Base para motor principal
acoplada ao carro
(peça 9) e suporte frontal para motor A
(peça 3).

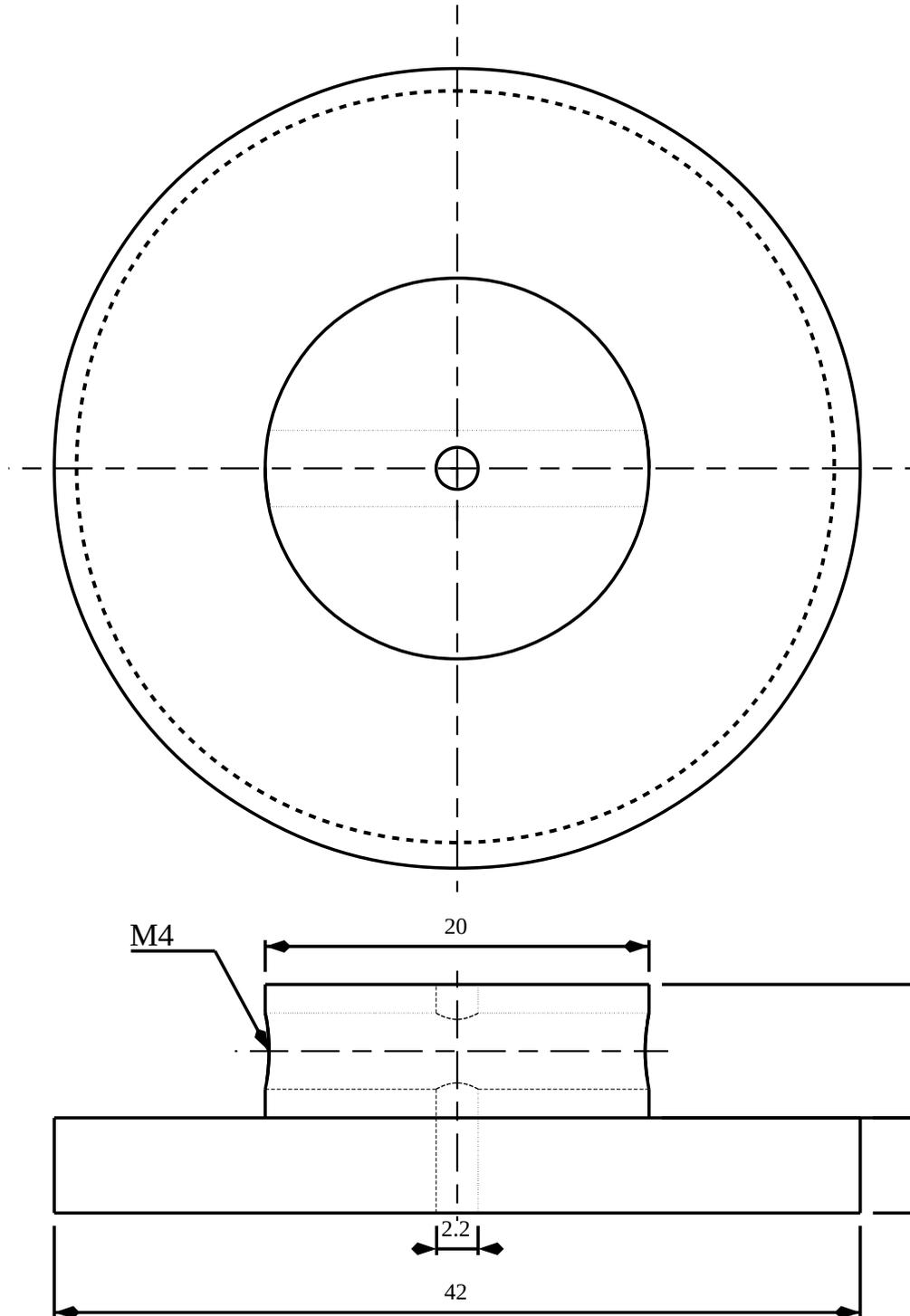
Página: 1 / 13	Tipo de folha: A4	Escala: 2:1
-------------------	----------------------	----------------

Peça: 2
Desenho: 1

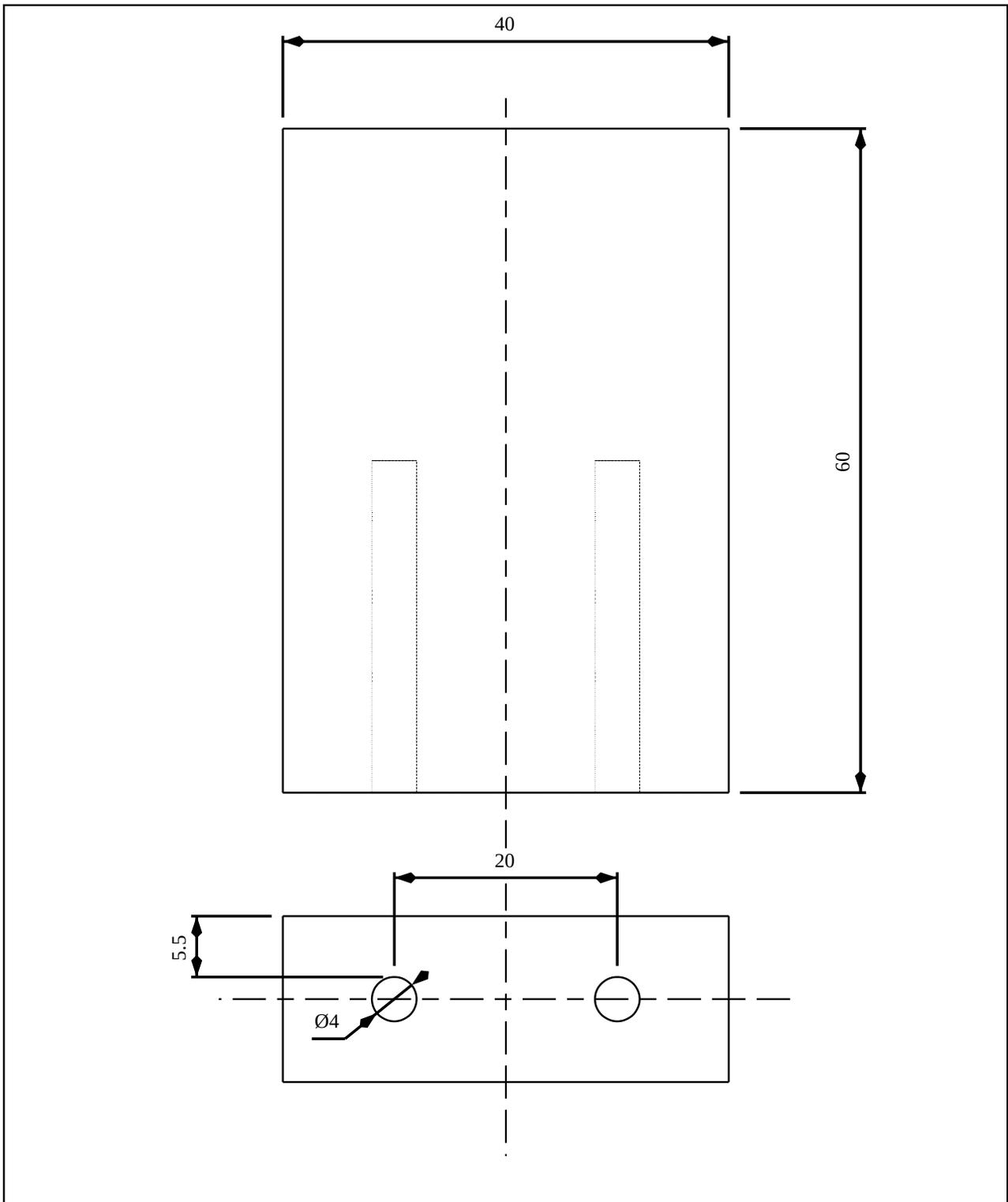
Data: 14/09/2019	Revisão: REV A
---------------------	-------------------



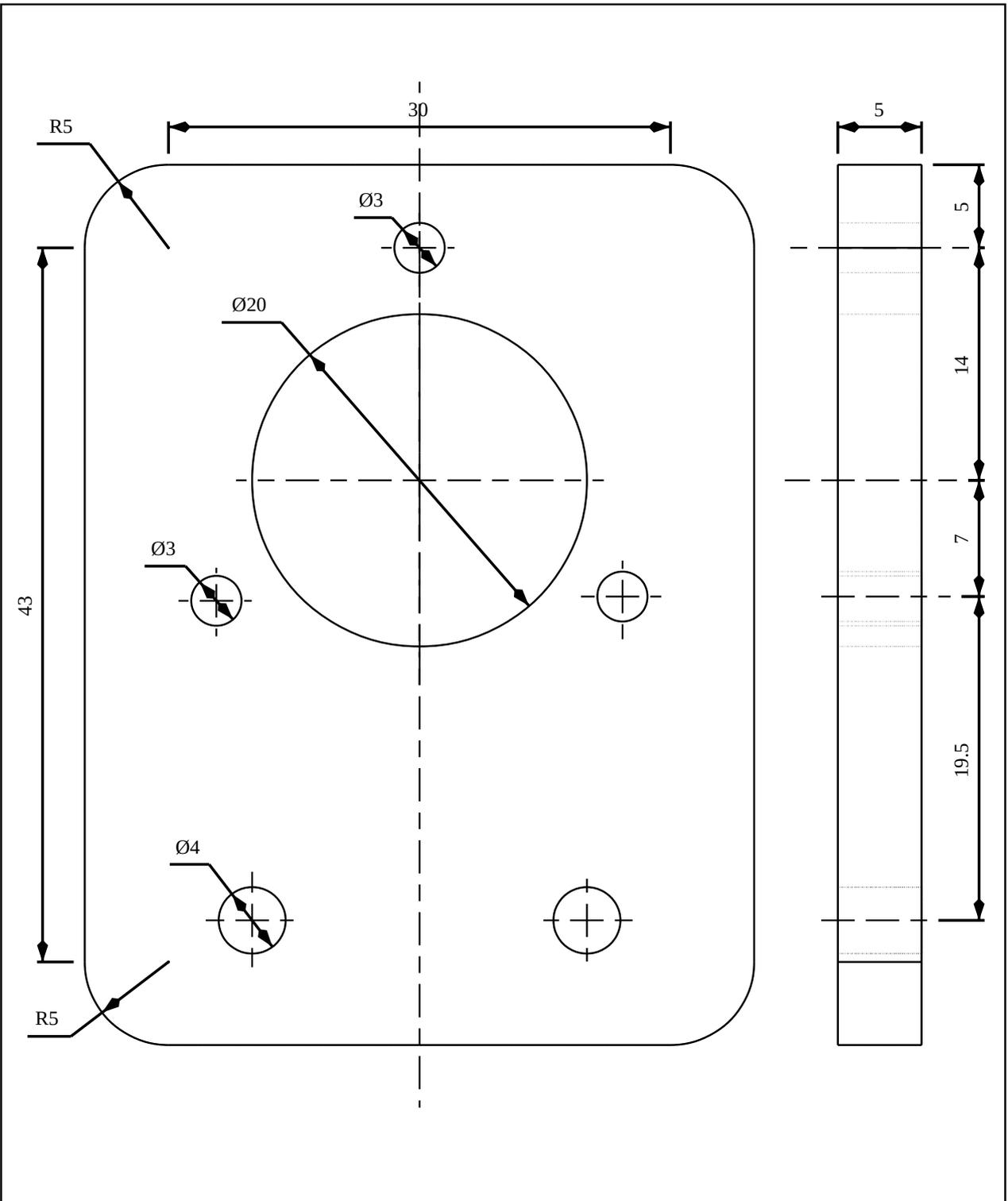
Autor: Matheus D. Azevedo	Título: Suporte para Motor A		
Descrição: Suporte frontal para motor principal acoplada à base para motor A (peça 3).	Página: 2 / 13	Tipo de folha: A4	Escala: 3:1
	Peça: 3		
	Desenho: 2		
Data: 14/09/2019		Revisão: REV A	



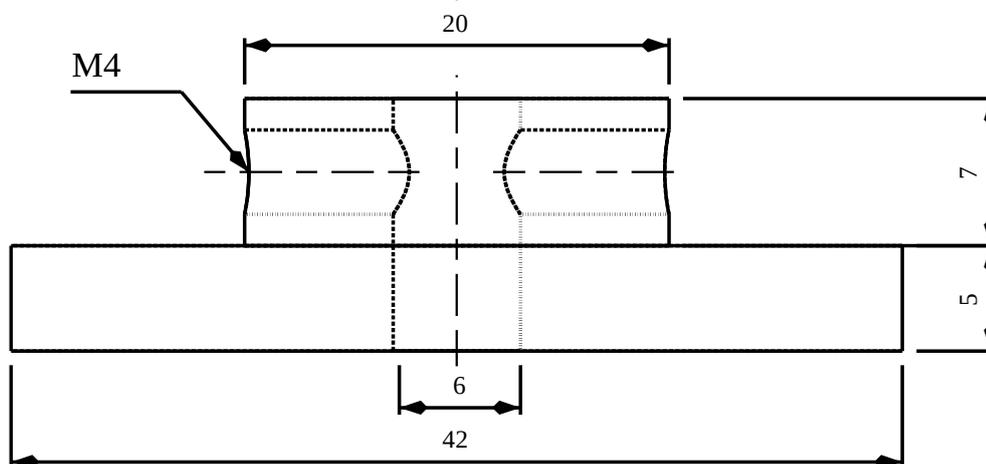
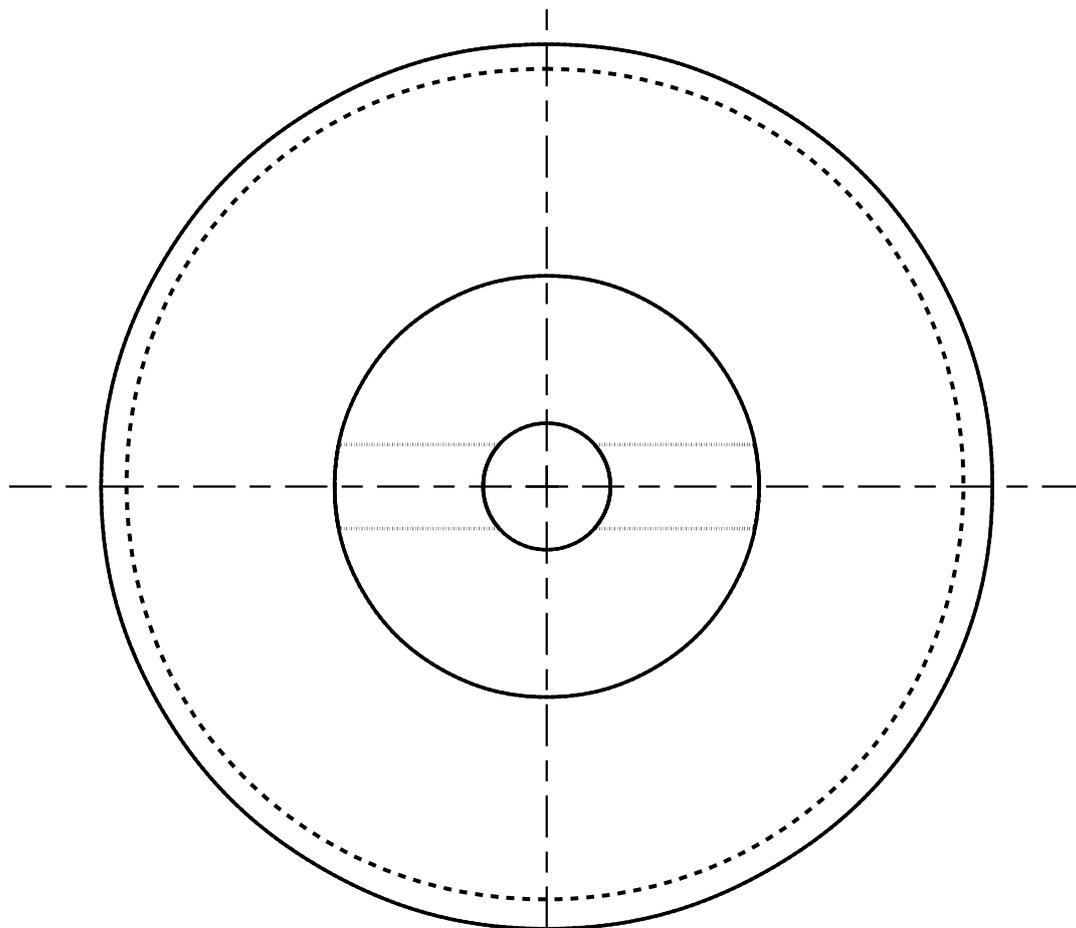
Autor: Matheus D. Azevedo	Título: Engrenagem 2 - Pinhão
Descrição: Engrenagem de 27 dentes e módulo 1.5 mm acoplada no motor principal usado no trabalho de conclusão de curso	Página: A4 Tipo de folha: 3 / 13 Escala: 3:1
	Peça: 4 Desenho: 3
	Data: 14/09/2019 Revisão: REV A



Autor: Matheus D. Azevedo	Título: Base para Encoder		
Descrição: Base para sensor encoder acoplada ao suporte frontal para encoder (peça 7).	Página: 4 / 13	Tipo de folha: A4	Escala: 2:1
	Peça: 6		
	Desenho: 4		
Data: 14/09/2019		Revisão: REV A	



Autor: Matheus D. Azevedo		Título: Suporte para Encoder		
Descrição: Suporte frontal para sensor encoder acoplada à base para encoder (peça 6).	Página:	5 / 13	Tipo de folha:	A4
	Peça:	7		
	Desenho:	5		
	Data:	21/09/2019	Revisão:	REV A

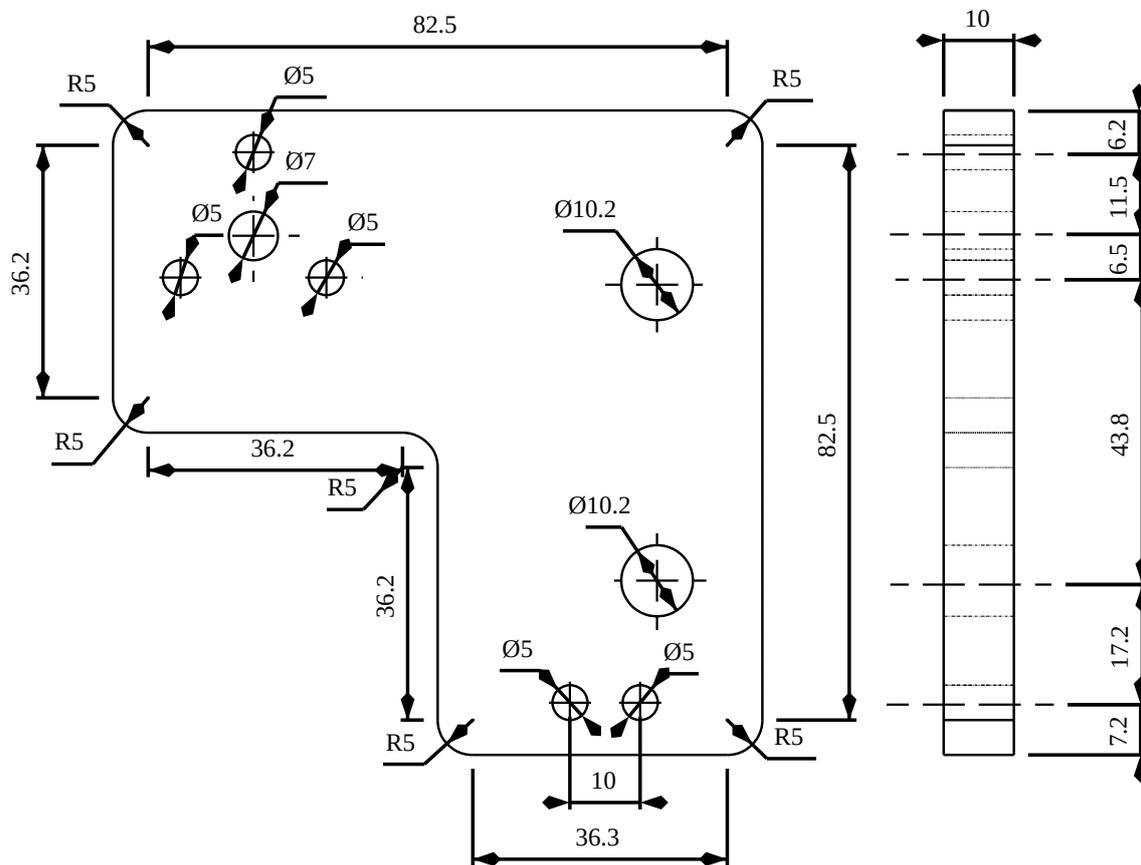


Autor:
Matheus D. Azevedo

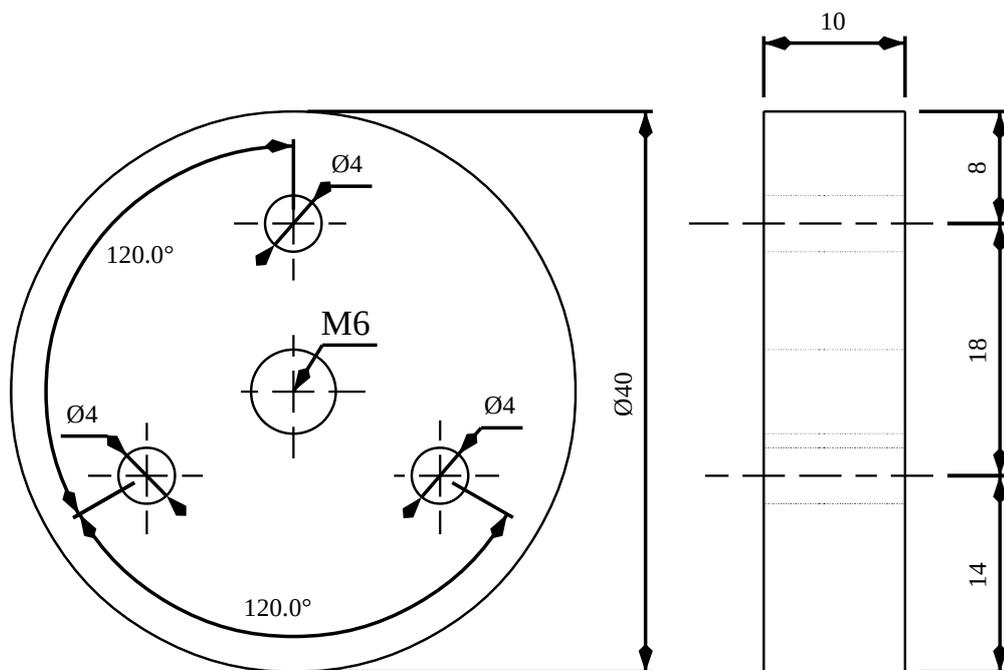
Título:
Engrenagem 2 - Coroa

Descrição:
 Engrenagem de 27 dentes e módulo 1.5 acoplada no encoder usada no trabalho de conclusão de curso

Página: A4	Tipo de folha: 6 / 13	Escala: 2:1
Peça: 8		
Desenho: 6		
Data: 21/09/2019	Revisão: REV A	



Autor: Matheus D. Azevedo	Título: Carro de movimentação		
Descrição: Peça de movimentação do motor principal (motor acoplado na porca anti-backlash (peças 10 até 13) e nos eixos-guia (peça 20) e base para motor A (peça 2)	Página: 7 / 13	Tipo de folha: A4	Escala: 1:1
	Peça: 9		
	Desenho: 7		
Data: 21/09/2019		Revisão: REV A	



Autor:
Matheus D. Azevedo

Título:
Porca 2 - Não rosqueada

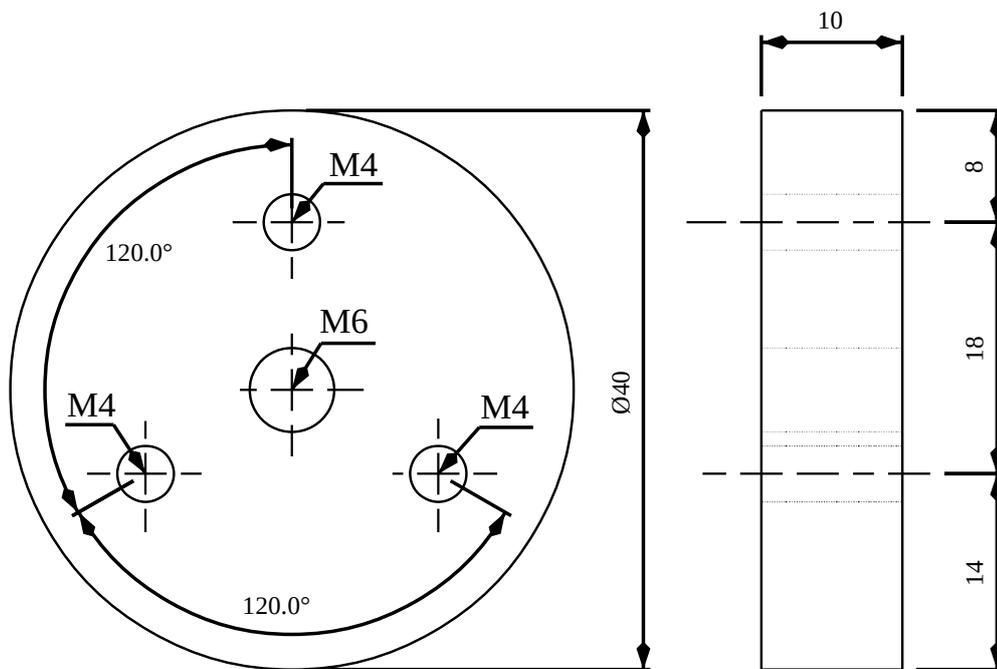
Descrição:
**Porca anti-backlash com furos auxiliares
 não rosqueados
 usada no trabalho de
 conclusão de curso**

Página: A4	Tipo de folha: 8 / 15	Escala: 3:1
----------------------	---------------------------------	-----------------------

Peça: 10

Desenho: 8

Data: 28/09/2019	Revisão: REV A
----------------------------	--------------------------

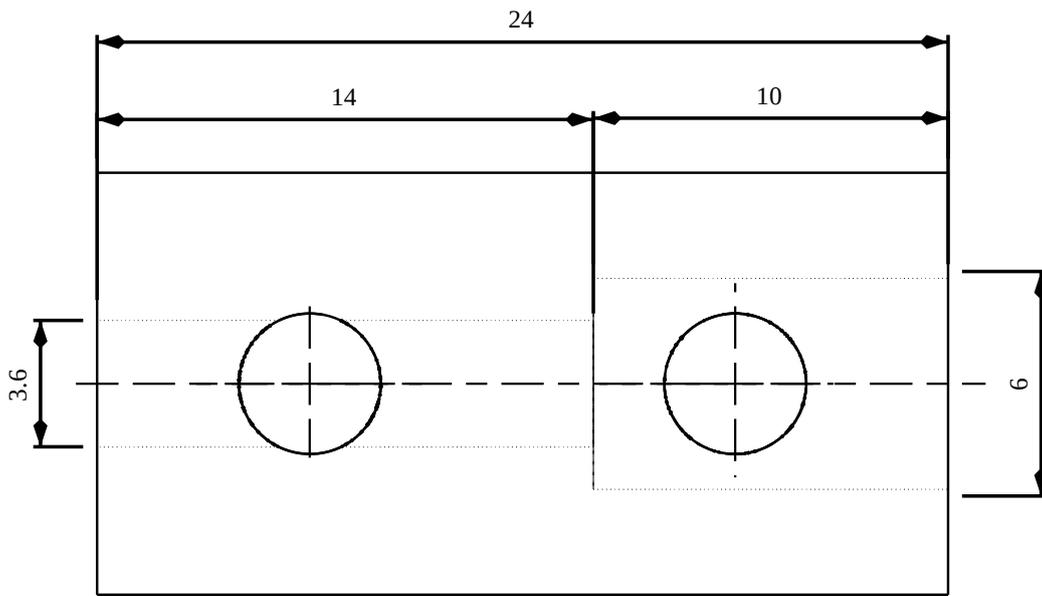


Autor:
Matheus D. Azevedo

Título:
Porca 1 - Rosqueada

Descrição:
**Porca anti-backlash com furos auxiliares
 rosqueados
 usada no trabalho de
 conclusão de curso**

Página: A4	Tipo de folha: 9 / 13	Escala: 3:1
Peça: 13		
Desenho: 9		
Data: 28/09/2019		Revisão: REV A

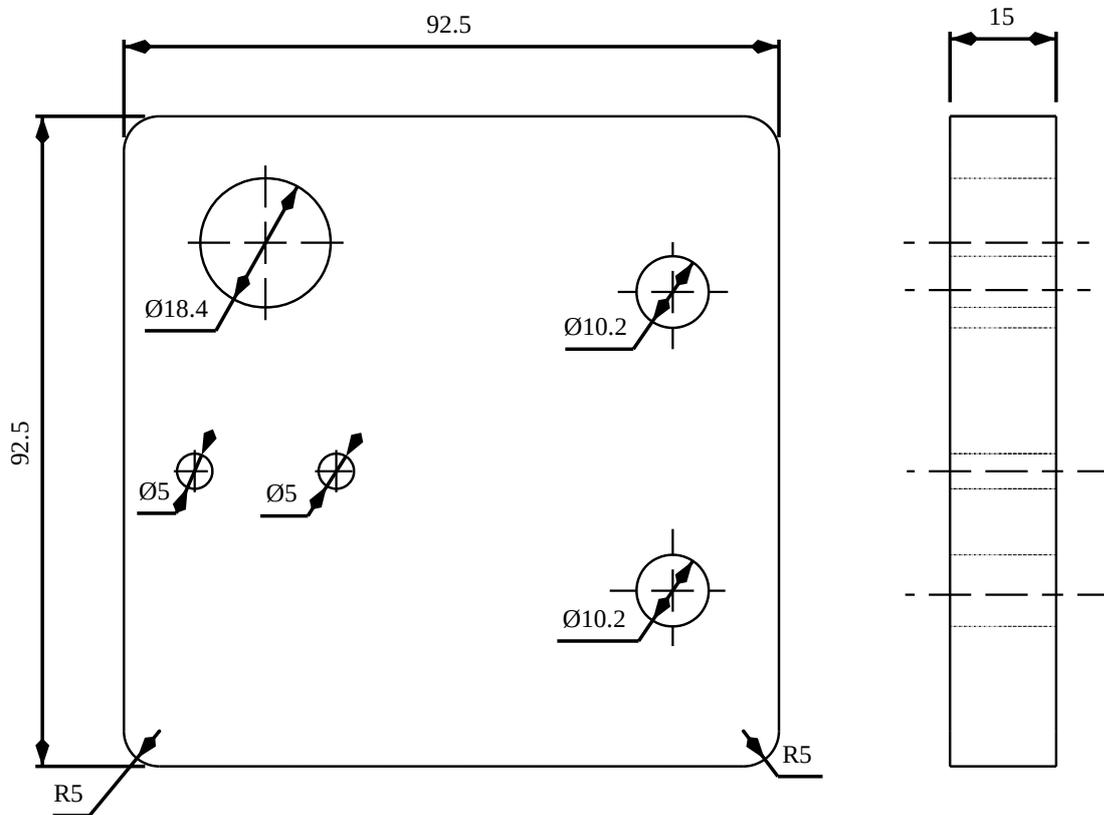


Autor:
Matheus D. Azevedo

Título:
Acoplador

Descrição:
Acoplador para motor secundário

Página: A4	Tipo de folha: 10 / 13	Escala: 5:1
Peça: 16		
Desenho: 10		
Data: 28/09/2019		Revisão: REV A



Autor:
Matheus D. Azevedo

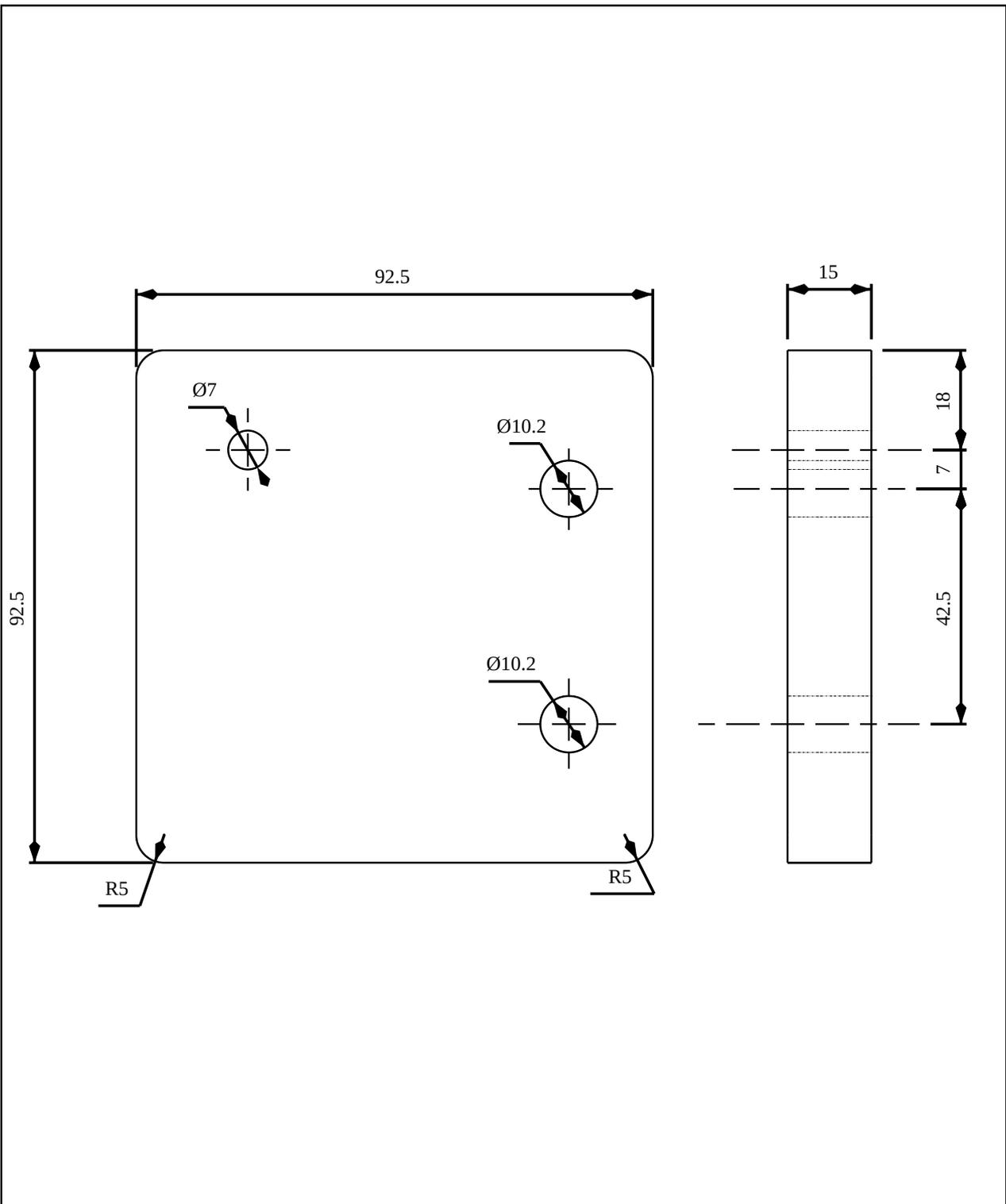
Título:
Suporte lateral com furo para motor B

Descrição:
Suporte lateral com furo passante para motor para inserção do acoplador (Peça 16).

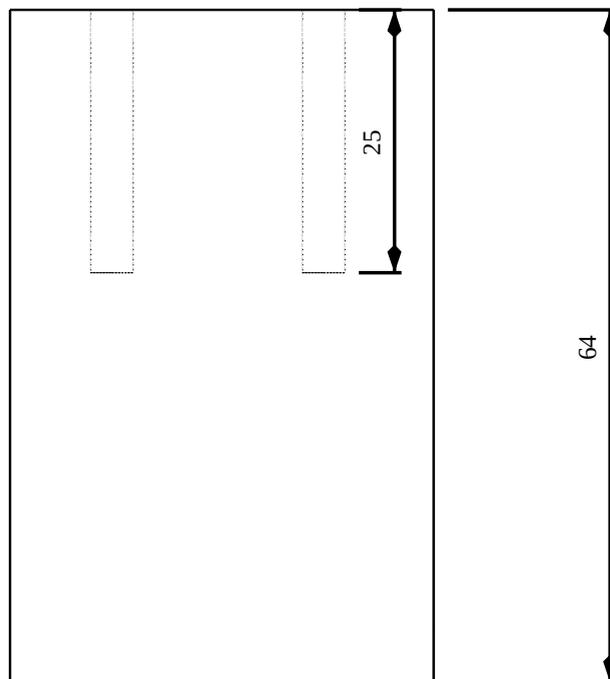
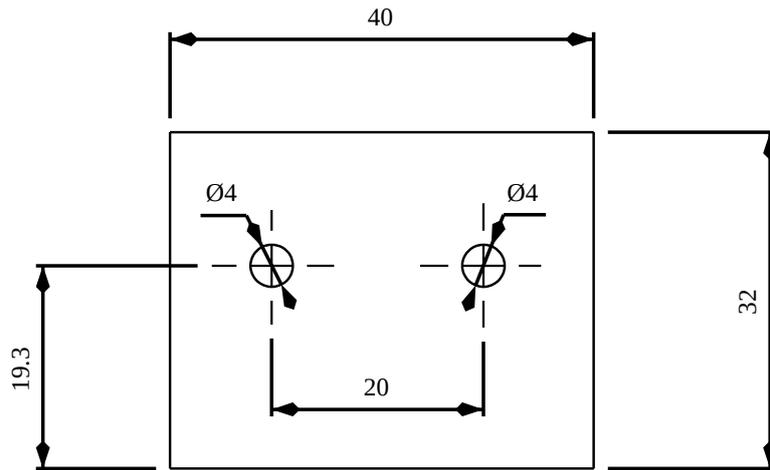
Página: 11 / 13	Tipo de folha: A4	Escala: 1:1
---------------------------	-----------------------------	-----------------------

Peça: 18
Desenho: 11

Data: 29/09/2019	Revisão: REV A
----------------------------	--------------------------



Autor: Matheus D. Azevedo		Título: Suporte lateral com furo para barra M6		
Descrição: Suporte lateral com furo passante para acoplamento da barra roscada MA M6	Página:	Tipo de folha:	Escala:	
	12 / 13	A4	1:1	
	Peça:	19		
	Desenho:	12		
Data:	Revisão:			
29/09/2019	REV A			



Autor:
Matheus D. Azevedo

Título:
Base para motor B

Informações complementares:

Base para acoplamento do motor B
ligada
ao suporte lateral (Peça 19).

Página: 13 / 13	Tipo de folha: A4	Escala: 1.5:1
--------------------	----------------------	------------------

Peca:	17
-------	----

Desenho:	13
----------	----

Data: 29/09/2019	Revisao: REV A
---------------------	-------------------

Referências

- ABNT. *NBR ISO 7240 – Roscas métricas ISO de uso geral dimensões básicas*. 2004.
- AKIYAMA, M. *Folha de dados: série ak360*. Disponível em <<https://bit.ly/2wA62hm>>. Acesso em 5 de dezembro de 2019.
- ASTROM, K. J.; HAGGLUND, T. *PID Controllers: theory, design and tuning*. Research Park 27709, NC: Instrument Society of America, 1995.
- BERKLEY. *Regression Analysis*. acesso em 5 de dezembro de 2019.
- BRUMS, R. M. R. *et al. Motion Control Systems with backlash*. 2006. Dissertação (Mestrado em Ciência da Computação) — .
- CHEN, M.; GE, S. S.; HOW, B. V. E. Robust Adaptive Neural Network Control for a Class of Uncertain MIMO Nonlinear Systems With Input Nonlinearities. *EEE Transactions on Neural Networks*, [S.l.], n.5, p.796 – 812, 2010.
- CHOLLET, F. *Deep Learning with Python*. [S.l.]: Manning Publications Co, 2018.
- CYBENKO, G. Approximation by Superpositions of a Sigmoidal Function. *Math Control Signals Systems*, [S.l.], n.2, p.303–314, 1989.
- DABIS, H. S.; MOIR, T. J. Least Mean Squares as a Control System. *International Journal of Control*, [S.l.], n.54, p.321–335, 2007.
- DORF, R. C.; BISHOP, R. H. *Modern Control Systems*. Upper Saddle River, NJ: Pearson, 2010.
- DWYER, A. O. *Handbook of PI and PID Controller Tuning Rules*. 57 Shelton Street, London: Imperial College Press, 2006.
- FLORES, P.; GOMES, J. *Cinemática e Dinâmica de Engrenagens. Teoria e Exercícios de Aplicação*. Braga, Portugal: McGraw Hill, 2014.

- GAO, W.; SELMIC, R. S. Saturation and Deadzone Compensation of Systems using Neural Networks and Fuzzy Logic. *American Control Conference*, [S.l.], p.1715–1720, 2005.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016.
- HANDSON, T. *L298N Dual H-Bridge Motor Driver*. Disponível em <<https://bit.ly/2JEh7VR>>. Acesso em 5 de dezembro de 2019.
- HAYKIN, S. *Neural Networks: a comprehensive foundation*. [S.l.]: Pearson Prentice Hall, 2005.
- HE, W.; CHEN, Y.; YIN, Z. Adaptive Neural Network Control of an Uncertain Robot With Full-State Constraints. *IEEE Transactions on Neural Networks*, [S.l.], n.46, p.620 – 629, 2016.
- HE, W.; DONG, Y.; SUN, C. Adaptive Neural Impedance Control of a Robotic Manipulator With Input Saturation. *IEEE Transactions on systems, man, and cybernetics: systems*, [S.l.], p.334 – 344, 2015.
- HE, W. *et al.* Neural Network Control of a Robotic Manipulator With Input Deadzone and Output Constraint. *IEEE Transactions on systems, man, and cybernetics: systems*, [S.l.], p.1–13, 2015.
- HUANG, S.; TAN, K. K. Intelligent Friction Modeling and Compensation Using Neural Network Approximations. *IEEE Transactions on Industrial Electronics*, [S.l.], p.3342–3349, 2012.
- INDUSTRY, K. G. *Gear Backlash - Technical Reference*. Disponível em <<https://bit.ly/30CoL9S>>. Acesso em 5 de dezembro de 2019.
- LAM, D. R. S. S.-L.; PUTMAN, J. A.; LORENZ, R. D. Neural Network Compensation of Gear Backlash Hysteresis in Position-Controlled Mechanisms. *IEEE Transactions on Industry Applications*, [S.l.], n.6, p.1475–1483, 1995.
- LATHI, B. P. *Sinais e Sistemas Lineares*. New York, NY: Bookman, 2007.
- LI, C.; CHENG, Z.; YAO, B. Identification and adaptive robust precision motion control of systems with nonlinear friction. *Nonlinear Dynamics*, [S.l.], 2019.
- LI, M.; JIA, Y. Precompensation decoupling control with High performance for 4WS velocity-varying vehicles. *International Journal of systems and science*, [S.l.], 2016.
- MATHWORKS. *Real μ Analysis*. Disponível em <<https://bit.ly/2Mmgmob>>. Acesso em 5 de dezembro de 2019.

- MATHWORKS. *What are Hammerstein-Wiener models?* Disponível em <<https://bit.ly/2wA62hm>>. Acesso em 5 de dezembro de 2019.
- NA, J. *et al.* Adaptive Prescribed Performance Motion Control of Servo Mechanisms with Friction Compensation. *IEEE Transactions on Industrial Electronics*, [S.l.], p.486–494, 2014.
- NAVARRO, C. M. *Comparação entre o método dos Mínimos Quadrados e Redes Neurais Artificiais para a estimação de parâmetros de um motor de corrente contínua*. 2012. Dissertação (Mestrado em Ciência da Computação) — UENF-RJ.
- NORTON, R. L. *Cinemática e Dinâmica dos mecanismos (com unidades do SI*. [S.l.]: McGraw Hill, 2008.
- OGATA, K. *Modern Control Engineering*. New York, NY: Pearson, 2009.
- PHILIPS, C. L.; NAGLE, H. T. *Digital Control Systems Analysis and Design*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- RASPBERRYPI. *Raspberry Pi 3 Model B*. Disponível em <<https://bit.ly/2W0BVz9>>. Acesso em 5 de dezembro de 2019.
- ROBIZ, E. R. *LPD3806-400BM-G5-24C AB Two Phase 5-24V 400 Pulses Incremental Optical Rotary Encoder*. Disponível em <<https://bit.ly/2NhK6A0>>. Acesso em 5 de dezembro de 2019.
- SELMIC, R. S.; LEWIS, F. L. Deadzone Compensation on Motion Control Systems using Neural Networks. *IEEE Transactions on Automatic Control*, [S.l.], n.4, p.602–613, 2000.
- SELMIC, R. S.; LEWIS, F. L. Neural-network approximation of piecewise continuous functions: application to friction compensation. *IEEE Transactions on Neural Networks*, [S.l.], n.13, p.745 – 751, 2002.
- SHIGLEY, J. E.; MISHCKE, C. R. *Standard Handbook of Machine Design*. New York, NY: McGraw Hill, 1989.
- SOUZA, M. J. F. *Otimização – notas de aula*. Disponível em <<https://bit.ly/3323ozL>>. Acesso em 5 de dezembro de 2019.
- TREASE, B. P. *Chaokey Mechanical Assemblies*. Disponível em <<https://bit.ly/32fesII>>. Acesso em 5 de dezembro de 2019.
- UMANS, S. D. *Máquinas Elétricas de Fitzgerald e Kingsley*. 7.ed. New York, NY: McGraw Hill, 2014.

- WANG, W. The Exact and Unique Solution for Phase-Lead and Phase-Lag Compensation. *IEE Transactions on Education*, [S.l.], 2003.
- WILLS, A. *et al.* Identification of HammersteinWiener models. *Automatica*, [S.l.], n.49, p.70–81, 2013.
- YAO, J.; JIAO, Z.; MA, D. Adaptive robust control of dc motors with extended state observer. *IEEE Transactions on Industrial Electronics*, [S.l.], 2014.