



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

ELÁDIO GOUVEIA OLIVEIRA

**IDENTIFICAÇÃO E CONTROLE DIGITAL DE UMA PLANTA DIDÁTICA DE
MOVIMENTO ANGULAR PROPULSIONADA POR MOTOR E HÉLICE**

FORTALEZA

2020

ELÁDIO GOUVEIA OLIVEIRA

**IDENTIFICAÇÃO E CONTROLE DIGITAL DE UMA PLANTA DIDÁTICA DE
MOVIMENTO ANGULAR PROPULSIONADA POR MOTOR E HÉLICE**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Fabrício Gonzalez Nogueira.

Coorientador: MSc. Marcus Davi do Nascimento Forte

FORTALEZA

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- O46i Oliveira, Eládio Gouveia.
Identificação e controle digital de uma planta didática de movimento angular propulsionada por motor e hélice / Eládio Gouveia Oliveira. – 2020.
103 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2020.
Orientação: Prof. Dr. Fabrício Gonzalez Nogueira.
Coorientação: Prof. Me. Marcus Davi do Nascimento Forte.
1. Controle Discreto. 2. Identificação de Sistemas. 3. Método de Gauss-Newton. 4. Mínimos Quadrados Não Recursivo. 5. Sequência Binária Pseudoaleatória. I. Título.

CDD 621.3

ELÁDIO GOUVEIA OLIVEIRA

IDENTIFICAÇÃO E CONTROLE DIGITAL DE UMA PLANTA DIDÁTICA DE
MOVIMENTO ANGULAR PROPULSIONADA POR MOTOR E HÉLICE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica do
Centro de Tecnologia da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de Bacharel em Engenharia Elétrica.

Aprovada em: ___ / ___ / ____.

BANCA EXAMINADORA

Prof. Dr. Fabrício Gonzalez Nogueira
Universidade Federal do Ceará (UFC)

MSc. Marcus Davi do Nascimento Forte
Universidade Federal do Ceará (UFC)

Prof. Dr. Bismark Claire Torrico
Universidade Federal do Ceará (UFC)

Prof. Dr.-Ing. Diego de Sousa Madeira
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

Aos meus pais, Eliude e Inês, pelo amor, pela incansável dedicação, pelos incentivos, pela fibra moral; pela capacidade de enxergar na educação um meio de crescimento pessoal e desenvolvimento social, sem nunca medir esforços para nos proporcioná-la da melhor forma possível; por serem inspiração para que eu enfrente de cabeça erguida os desafios da vida sem jamais esquecer de onde viemos.

À minha irmã Elaine, além de toda família Gouveia e família Oliveira pelo carinho e apoio de toda uma vida.

À Raísa, por todo amor e carinho de tanto tempo; pelo companheirismo dentro e fora da faculdade, dando sempre todo apoio para que eu pudesse superar esse desafio; pela ajuda e força em tantas noites em claro enfrentando o cansaço em busca de conhecimento; por ser um mar de boas energias e bons sentimentos que me irradia sempre.

À família Bruno Norões pelo carinho, pela amizade e pelo acolhimento sempre que possível.

Aos meus amigos e amigas, irmãos de vida, aqui representados por George Fernandes, Fernando Sales, Ícaro Ronney, Matheus Narciso e Davi Amorim; pela longa amizade, pelo carinho e pelo companheirismo.

Aos meus amigos, empreendedores, e sócios visionários, Élio Júnior e Victor Levi; pelo companheirismo, por todas as ideias, por todos os *brainstorms*, pelo comprometimento em mudar o mundo através da educação.

A todos os meus colegas de curso que estiveram juntos nesta jornada, estudando, fazendo laboratórios, trabalhos, relatórios; que compartilharam desde momentos felizes a discussões políticas inflamadas: juntos crescemos.

A todos os meus professores pela árdua missão de vida, em especial, ao professor Dr. Fabrício Nogueira pela orientação ao longo deste trabalho.

“Eu sou eu e minha circunstância, e se não salvo a ela não salvo a mim” (Ortega y Gasset).

RESUMO

Veículos Aéreos Não Tripulados (VANTs) são uma inovação tecnológica cada vez mais presente no cotidiano da sociedade. Com as mais diversas possíveis aplicações, estes equipamentos despertam progressivamente o interesse da comunidade científica, servindo, no contexto deste trabalho, como fonte de inspiração.

A complexidade mecânica dos VANTs multirrotores foi, com objetivo didático, simplificada neste trabalho, ao ponto deste projeto contar com apenas um elemento propulsor (motor e hélice) guiado por um braço giratório, produzindo um movimento angular. A planta proposta para este trabalho foi construída juntamente com um sensor de ângulo baseado em um potenciômetro de escala linear.

Descreveu-se, neste trabalho, a revisão bibliográfica necessária para os procedimentos de identificação da planta e do projeto de um controlador digital.

Dois métodos de identificação foram propostos: Mínimos Quadrados Não Recursivo e Método de Gauss-Newton, tendo este último apresentado melhores resultados, sendo, portanto, a função de transferência resultante deste método escolhida para representar a planta.

Utilizou-se a alocação de polos como estratégia para o projeto do controlador digital, bem como utilizou-se a estrutura RST para representá-lo.

O primeiro controlador digital projetado não atendeu às especificações de desempenho propostas. Após ajustes, um novo controlador foi calculado e implementado, tendo resultados satisfatórios, dentro das especificações de desempenho propostas.

Palavras-chave: *Controle Discreto, Identificação de Sistemas, Transformada Z, Python, MATLAB, Controle de Sistemas Dinâmicos, Método de Gauss-Newton, Mínimos Quadrados Não Recursivo, Sequência Binária Pseudoaleatória.*

ABSTRACT

Unmanned Aerial Vehicles are a technological innovation increasingly present in society's daily lives. With diverse applications, these devices are increasingly attracting interest from the scientific community, serving, in the context of this work, as a source of inspiration.

The mechanical complexity of a multi-rotor UAV was simplified in this work for didactic purposes. This project has only one propelling element (composed by a motor and propeller) guided by a rotating arm, producing an angular movement. The plant proposed for this work was built as well as an angle sensor based on a linear scale potentiometer.

The necessary bibliographic review for the identification procedures of the plant and the design of a digital controller was described in this work.

Two identification methods have been proposed: Non-Recursive Least Squares and Gauss-Newton Method, with the last one showing better results, being, therefore, the transfer function resulting from this method chosen to represent the plant behavior.

Pole Placement was used as strategy for the design of the digital controller, as well as the RST structure to represent it.

The first designed digital controller did not meet the proposed performance specifications. After adjustments, a new controller was calculated and implemented, with satisfactory results, within the proposed performance specifications.

Keywords: *Digital Control, System Identification, Z-Transform, Python, MATLAB, Dynamic Systems Control, Gauss-Newton Method, Non-Recursive Least Squares, Pseudorandom Binary Sequence.*

LISTA DE FIGURAS

Figura 1 – Exemplo de VANT com 4 Rotores.....	15
Figura 2 – Metodologia de Identificação.....	19
Figura 3 – Sistema de Controle Digital	20
Figura 4 – ADCs, DACs e ZOHs	20
Figura 5 – Circuito RC	22
Figura 6 – Módulo de $ T(\omega) _{dB}$	23
Figura 7 – Diagrama de Blocos (Discretização com ZOH)	24
Figura 8 – Exemplo de Geração de SBPA.....	26
Figura 9 – Estrutura Canônica RST.....	34
Figura 10 – Polos Dominantes e Auxiliares (Plano Z)	37
Figura 11 – Controle Digital na Presença de Distúrbios e Ruído.....	39
Figura 12 – A planta e seus componentes.....	42
Figura 13 – Diagrama de Corpo Livre da Planta.....	43
Figura 14 – Placa de desenvolvimento Freescale FRDM-KL25Z	46
Figura 15 – Diagrama de blocos da interligação lógica do circuito de atuação	47
Figura 16 – Conversor de Nível de Tensão	48
Figura 17 – Diagrama de blocos da interligação lógica do circuito de atuação	49
Figura 18 – Parâmetros Elétricos de um Motor BLDC	50
Figura 19 – Sensor de ângulo	51
Figura 20 – Fixador do Potenciômetro	51
Figura 21 – Acoplador de Eixos	52
Figura 22 – MCP1541: Aplicação Típica	54
Figura 23 – Estrutura Completa do Sensor de Ângulo.....	54
Figura 24 – Diagrama de Comunicação para Identificação da Planta.....	57
Figura 25 – <i>Toolbox</i> System Identification do MATLAB (versão R2018b)	63
Figura 26 – Estimação da Complexidade do Modelo.....	64
Figura 27 – <i>Toolbox</i> System Identification (Gauss-Newton).....	68
Figura 28 – Processo de Estimação do Modelo ARX	69

LISTA DE GRÁFICOS

Gráfico 1 – Primeiro Ensaio Preliminar	52
Gráfico 2 – Segundo Ensaio Preliminar	55
Gráfico 3 – Ensaio de Resposta ao Degrau	58
Gráfico 4 – Atraso de Resposta (Com Ampliação).....	59
Gráfico 5 – Tempo de Subida (Com Ampliação)	60
Gráfico 6 – Módulo da Resposta em Frequência	60
Gráfico 7 – Ensaio de Identificação (0-60s).....	62
Gráfico 8 – Divisão dos dados Adquiridos.....	63
Gráfico 9 – Simulação do Modelo Identificado	65
Gráfico 10 – Simulação de Validação do Modelo Identificado.....	66
Gráfico 11 – Análise de Resíduos do Modelo Identificado.....	67
Gráfico 12 – Saída dos Modelos Medido e Simulado (Ampliada)	70
Gráfico 13 – Simulação de Validação do Modelo Identificado.....	70
Gráfico 14 – Análise dos Resíduos.....	71
Gráfico 15 – Resposta Desejada (Modelo de Sistema de Segunda Ordem).....	74
Gráfico 16 – Polos Dominantes.....	74
Gráfico 17 – Alocação de Polos	76
Gráfico 18 – Respostas ao Degrau	77
Gráfico 19 – Resposta em Frequência de $S_{yp}(z^{-1})$	79
Gráfico 20 – Resposta em Frequência de $S_{up}(z^{-1})$	80
Gráfico 21 – Ensaio do Sistema Controlado	81
Gráfico 22 – Alocação de Polos (2).....	82
Gráfico 23 – Respostas ao Degrau (2).....	83
Gráfico 24 – Resposta em Frequência de $S_{yp}(z^{-1})$ (2)	83
Gráfico 25 – Resposta em Frequência de $S_{up}(z^{-1})$ (2)	84
Gráfico 26 – Ensaio do Sistema Controlado (2).....	84
Gráfico 27 – Limite do Modelo Linearizado.....	85

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Introdução à Sistemas de Controle	14
1.2	Veículos Aéreos Não Tripulados.....	14
1.3	Motivação e Justificativa	15
1.4	Objetivos.....	16
1.4.1	<i>Objetivo geral</i>	16
1.4.2	<i>Objetivos específicos</i>	16
2	IDENTIFICAÇÃO DE SISTEMAS.....	17
2.1	O Modelo ARX.....	17
2.2	O Processo de Identificação.....	18
2.3	Aquisição de Dados	19
2.3.1	<i>Interfaceamento</i>	19
2.3.2	<i>Amostragem</i>	21
2.4	Filtragem dos Sinais Adquiridos.....	21
2.4.1	<i>Filtro Passa-Baixas de Primeira Ordem Analógico (RC)</i>	21
2.4.2	<i>Filtro Passa-Baixas de Primeira Ordem Digital</i>	23
2.5	Projeto do Sinal de Excitação	25
2.6	Estimação da Complexidade do Modelo	27
2.7	Cálculo dos Parâmetros do Modelo	27
2.7.1	<i>Condicionamento do Sinal</i>	27
2.7.2	<i>Regressão Linear (Mínimos Quadrados Não Recursivo)</i>	28
2.7.3	<i>Regressão Não Linear (Método de Gauss-Newton)</i>	29
2.8	Validação do Modelo	31
2.8.1	<i>Índices de Validação</i>	32

2.8.2	<i>Análises de Resíduos (Autocorrelação e Correlação Cruzada)</i>	32
3	CONTROLE DIGITAL	34
3.1	A Estrutura RST	34
3.2	Alocação de Polos	35
3.2.1	<i>Determinação dos Polos de Malha Fechada</i>	35
3.2.2	<i>Regulação</i>	38
3.2.3	<i>Definição dos filtros H_r e H_s</i>	39
3.2.4	<i>Solução da Equação Diofantina</i>	40
3.2.5	<i>Cálculo dos Polinômios R, S e T</i>	41
4	DESCRIÇÃO DA PLANTA	42
4.1	Análise Mecânica	43
4.2	Aquisição e Processamento de Dados	45
4.2.1	<i>Projeto do Circuito de Atuação</i>	46
4.2.2	<i>Projeto do Sensor de Ângulo</i>	50
5	IDENTIFICAÇÃO DA PLANTA	56
5.1	Projeto e Inclusão de um Filtro Digital	56
5.2	Metodologia Prática	56
5.3	Ensaio de Resposta ao Degrau	57
5.4	Projeto da Sequência Binária Pseudoaleatória	61
5.5	Identificação	62
5.5.1	<i>Ensaio de Identificação</i>	62
5.5.2	<i>Uso do toolbox System Identification do MATLAB</i>	63
5.5.3	<i>Identificação utilizando Mínimos Quadrados Não Recursivo</i>	64
5.5.4	<i>Identificação utilizando o Método de Gauss-Newton</i>	67
6	PROJETO E IMPLEMENTAÇÃO DO CONTROLADOR	73
6.1	Definição dos Índices de Desempenho Iniciais	73
6.2	Definições Acerca dos Polinômios H_s , H_r , S' , R' , P , A' e B'	74

6.3	Alocação de Polos.....	76
6.4	Cálculo de R, S e T	77
6.5	Funções de Sensibilidade Resultantes	79
6.6	Ensaio de Teste do Controlador	80
6.7	Ajuste dos Parâmetros do Controlador	81
6.8	Limite Prático do Modelo Identificado.....	85
7	CONCLUSÕES	86
	REFERÊNCIAS	88
	ANEXO A – PROGRAMA GERADOR DE SBPA (MBED).....	90
	ANEXO B – <i>SCRIPT</i> INTERFACE.PY (PYTHON).....	93
	ANEXO C – <i>SCRIPT</i> AQUISIÇÃO.M (MATLAB).....	94
	ANEXO D – <i>SCRIPT</i> MÍNIMOS QUADRADOS NÃO RECURSIVO (MATLAB).....	96
	ANEXO E – <i>SCRIPT</i> DE PROJETO DO CONTROLADOR (MATLAB)	99
	ANEXO F –PROGRAMA DE IMPLEMENTAÇÃO DO CONTROLADOR (MBED).....	102
	ANEXO G – <i>SCRIPT</i> INTERFACE2.PY (PYTHON).....	105

1 INTRODUÇÃO

1.1 Introdução a Sistemas de Controle

O projeto de sistemas de controle é um importante fator no desenvolvimento de novas tecnologias e suas aplicações. Controladores digitais ou analógicos estão, de forma ubíqua, presentes no nosso cotidiano: no ar-condicionado, nas geladeiras modernas, nas televisões e nos sistemas veiculares como injeção eletrônica, direção com assistência elétrica ou freio com sistema antibloqueio.

Segundo Nise (2016, p. 2), “um sistema de controle consiste em subsistemas e processos (ou plantas) construídos com o objetivo de se obter uma saída desejada com um desempenho desejado, dada uma entrada especificada”. A própria natureza, ao longo do tempo e por meio da evolução das espécies, desenvolveu sistemas bioquímicos de controle para variáveis importantes à sobrevivência, como a temperatura dos animais, o nível de açúcar no sangue, a produção de hormônios, entre outras (BRITES, 2013).

Neste ponto da nossa evolução, dada a evidente evolução tecnológica aplicada aos sistemas eletrônicos analógicos e digitais, criou-se um ambiente favorável para que sistemas controlados cada vez mais complexos surjam, tendo como base as teorias clássica e moderna de controle de sistemas dinâmicos.

1.2 Veículos Aéreos Não Tripulados

Os VANTS são exemplos de tecnologias que necessitam de um sistema complexo de controle e têm sido utilizados para as mais diversas aplicações: desde o uso bélico militar (SABBAGH, 2019), serviços de entrega de mercadorias (D’ONFRO, 2019), auxílio à indústria da construção civil (TKAC; MESAROS, 2019) ao uso mais comum como entretenimento doméstico e desportivo.

Com popularidade ascendente, estima-se que, em 2020, o crescimento de remessas de VANTs para aplicações empresariais em todo o mundo aumente 50% em relação à 2019, alcançando aproximadamente 526 mil unidades, e que até 2023 este número possa chegar a aproximadamente 1,3 milhão (GARTNER, 2019), conforme demonstrado na Tabela 1.

Tabela 1 – 5 Maiores Casos de Uso de Drones (2019 a 2023) em Milhares de Vendas

Caso de Uso	2019	2020	2021	2022	2023
Monitoramento de Construções	141,1	209,8	294,2	394,3	509,5
Monitoramento de Serviços Contra Incêndios	32,7	48,5	58,2	63,7	67,0
Investigações de Seguros	31,8	46,4	67,2	96,3	135,8
Aquisição de Evidência Policial	26,8	45,1	60,4	72,0	80,7
Transporte de Mercadorias	12,9	24,9	44,4	75,1	122,0
Outros	106,2	150,8	206,5	275,3	356,5
Total	351,5	525,6	730,9	976,7	1.271,6

Fonte: Gartner Inc. (2019)

Um exemplo de VANT de pequeno porte é ilustrado na Figura 1.

Figura 1 – Exemplo de VANT com 4 Rotores



Fonte: (FLYNT, Joseph, 2019).

É notório, portanto, o crescente interesse da sociedade neste tipo de tecnologia, em suas mais variadas áreas de aplicações, devido às possibilidades de aumento da produtividade em relação a serviços já existentes e à criação de novos serviços que venham a melhorar as condições e a qualidade de vida da comunidade em geral.

1.3 Motivação e Justificativa

Justifica a realização deste trabalho a importância de se compreender e poder controlar a dinâmica rotacional dos elementos motores comuns aos VANTs multirrotores como suporte prévio a um possível projeto e desenvolvimento deste tipo de veículo.

Motiva a realização deste trabalho a aplicação prática dos conhecimentos adquiridos nas disciplinas de Sistemas de Controle Dinâmicos, Teoria do Controle Discreto e

Métodos Numéricos, juntamente com o conhecimento diverso adquirido ao longo do curso de Engenharia Elétrica. Outrossim, a possível utilidade deste estudo no desenvolvimento de outros trabalhos e criação de novos produtos, também é fator de extrema motivação.

1.4 Objetivos

1.4.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver uma planta didática composta por um conjunto motor-hélice, capaz de produzir um movimento rotacional em um braço giratório, bem como projetar um sistema de controle que atue sobre a velocidade deste motor de forma a controlar permanentemente o ângulo de rotação do braço giratório em relação a seu eixo de referência, podendo, assim, estabilizar sua dinâmica transitória e eliminar possíveis erros de regime permanente.

1.4.2 Objetivos específicos

Entre os objetivos específicos deste trabalho estão:

- Projetar e construir um sensor de rotação baseado em um potenciômetro linear;
- Construir uma planta didática que reduza as complexidades multivariáveis de um VANT real, mas que seja capaz de representar a dinâmica rotacional de um conjunto atuador;
- Identificar a planta em questão a fim de obter uma função de transferência representativa do sistema e das características a ela associadas;
- Projetar um controlador digital capaz de estabilizar a dinâmica transitória do sistema e zerar o erro de regime permanente.

2 IDENTIFICAÇÃO DE SISTEMAS

Um sistema pode ser descrito como a combinação de componentes diversos que agem em conjunto para atingir determinado objetivo (OGATA, 2010, p. 3). Modelagem e identificação são processos e métodos de determinação de modelos matemáticos representativos das características essenciais de sistemas dinâmicos (COELHO A. e COELHO L., 2004, p. 10).

Visando à caracterização de um sistema particular, a modelagem físico-matemática pode ser utilizada a partir da aplicação teórica das leis da física (como conservação de massa, energia e momento) (COELHO A. e COELHO L., 2004, p. 10). A identificação do sistema, por sua vez, baseia-se nas medidas ou observações experimentais de um sistema (LJUNG; GLAD, 1994, p. 17).

Devido às complexidades inerentes à análise físico-matemática de todos os componentes presentes em determinado sistema, este método raramente pode ser utilizado na caracterização de sistemas reais, sendo o método de identificação, inerentemente experimental, o qual provê relações entre as entradas e as saídas de um sistema, o mais adequado para tal (LANDAU; ZITO, 2006, p. 201).

2.1 O Modelo ARX

Um modelo autorregressivo é caracterizado pela dependência linear da variável de saída em relação aos seus valores anteriores, conforme ilustrado na equação (1), onde $y(t)$ representa a saída atual do sistema, $\theta_{[0-n]}$ parâmetros escalares e $e(t)$ o erro de modelagem.

$$y(t) = \theta_0 + \theta_1 \cdot y(t - 1) + \theta_2 \cdot y(t - 2) + \dots + \theta_n \cdot y(t - n) + e(t) \quad (1)$$

Em meio a vários modelos possíveis para a representação de sistemas dinâmicos no processo de identificação, o modelo ARX (*AutoRegressive model with eXogenous input and output variables*, traduzido como Modelo Autorregressivo com Variáveis Exógenas de Entrada e Saída) constitui a maneira mais simples de caracterizar um sistema dinâmico guiado por uma entrada em presença de incertezas (DIVERSI; GUIDORZI; SOVERINI, 2009). Este modelo é definido segundo a equação (2).

$$A(q^{-1})y(t) = q^{-d}.B(q^{-1}).u(t) + e(t) \quad (2)$$

Sendo:

$$A(q^{-1}) = 1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_{nA} \cdot q^{-nA};$$

$$B(q^{-1}) = b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_{nB} \cdot q^{-nB};$$

d – O atraso de resposta do sistema (em número de amostras);

$u(t)$ – O sinal de entrada;

$e(t)$ – O erro de modelagem.

Convertendo-se a definição na forma de um operador de transferência pulsado, obtém-se a expressão exposta na equação (3).

$$y(t) = H(q^{-1})u(t) \quad (3)$$

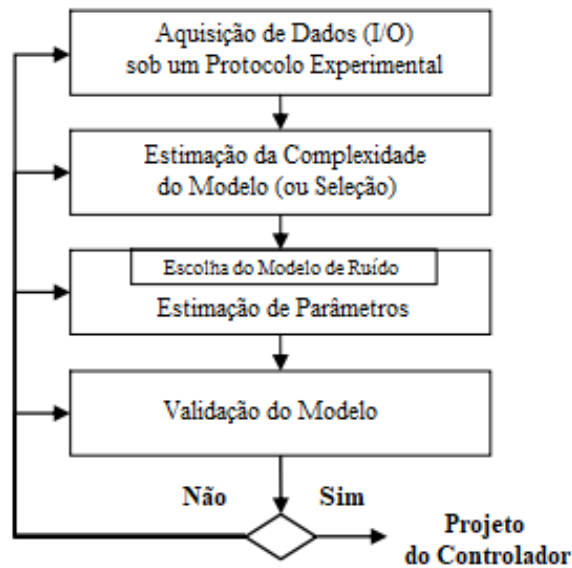
Logo, substituindo-se q^{-1} por z^{-1} , obtém-se em (4) a função de transferência pulsada do modelo (LANDAU; ZITO, p. 43).

$$H(z^{-1}) = \frac{z^{-d}B(z^{-1})}{A(z^{-1})} \quad (4)$$

2.2 O Processo de Identificação

A metodologia de identificação de um modelo paramétrico é descrita através do fluxograma contido na Figura 2.

Figura 2 – Metodologia de Identificação



Fonte: (LANDAU; ZITO, 2006, p. 204, tradução nossa)

Nas subseções posteriores, serão discutidas as etapas pertencentes ao processo de identificação.

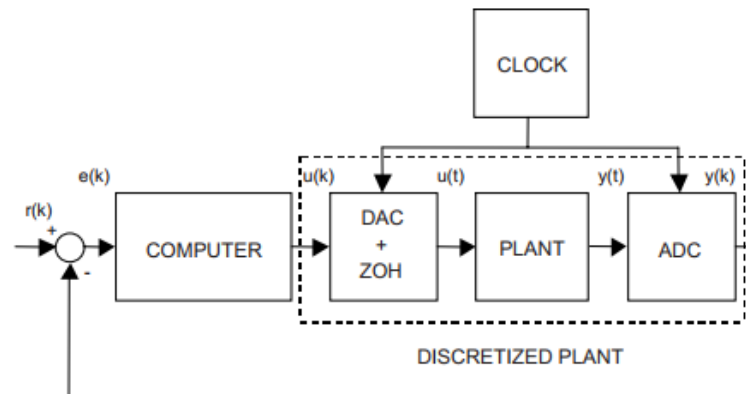
2.3 Aquisição de Dados

A aquisição de dados sobre um determinado sistema é o ponto inicial do processo de identificação.

2.3.1 Interfaceamento

O interfaceamento de um computador com o mundo físico e suas grandezas ocorre por meio de transdutores e atuadores conectados, respectivamente, a conversores analógicos-digitais (ADC) e conversores digitais-analógicos (DAC). Para que os valores discretos de saída se mantenham constantes a cada período de amostragem do sistema é estipulado um segurador de ordem zero (ZOH) juntamente com o conversor digital-analógico, conforme exibido na Figura 3.

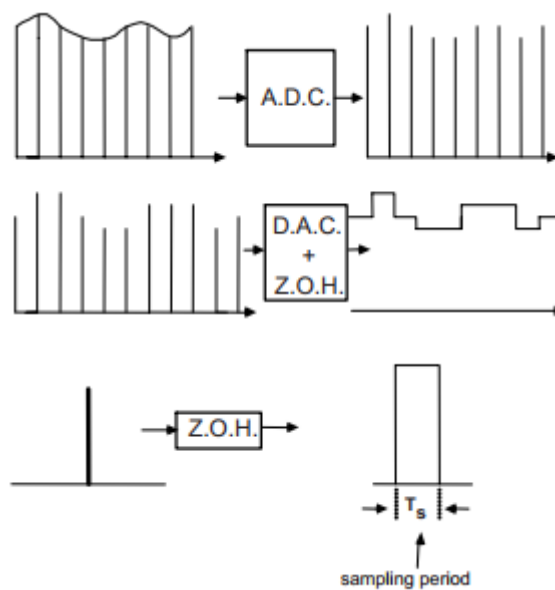
Figura 3 – Sistema de Controle Digital



Fonte: (LANDAU; ZITO, 2006, p. 26)

O funcionamento de ADCs, DACs e ZOHs é ilustrado pela figura Figura 4.

Figura 4 – ADCs, DACs e ZOHs



Fonte: (LANDAU; ZITO, 2006, p. 27)

Esse sistema em cascata DAC-Planta-ADC é interpretado pelo computador como o próprio sistema discreto a ser controlado por meio de sinais de entrada (lidos pelo ADC) e sinais de saída (gerados pelo DAC) (LANDAU; ZITO, 2006, p. 25) e dá ao computador a possibilidade de implementar estratégias e algoritmos de controle que não são possíveis de serem implementados por controladores analógicos (LANDAU; ZITO, 2006, p. 27).

2.3.2 Amostragem

Segundo Landau e Zito (2006, p. 31, tradução nossa), “a frequência de amostragem para controle digital de um sistema é escolhida de acordo com a largura de banda desejada para o sistema em malha-fechada”. Logo, a escolha da frequência de amostragem depende da performance do projeto do sistema controlado. Landau e Zito (2006, p. 31) estabelecem uma regra para a escolha desta frequência, conforme exibido na equação (6), sendo f_s a frequência de amostragem e f_B^{CL} a largura de banda do sistema desejado em malha-fechada.

$$f_s = 6 \text{ a } 25 f_B^{CL} \quad (5)$$

A mesma equação e os mesmos princípios são utilizados para a identificação de um sistema em malha aberta, sendo, entretanto, necessário estimar a largura de banda do sistema em malha aberta.

Segundo o teorema de Nyquist, para reconstruir um sinal contínuo a partir de um sinal amostrado, é necessário que a frequência de amostragem seja maior que o dobro da máxima componente de frequência do sinal amostrado, conforme a equação (6).

$$f_s > 2f_{max} \quad (6)$$

2.4 Filtragem dos Sinais Adquiridos

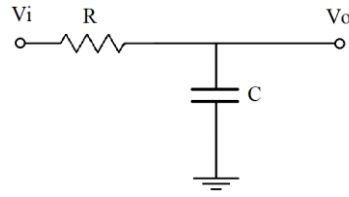
Com a finalidade de evitar distorções no espectro amostrado, é necessário introduzir filtros *anti-aliasing* de modo a atenuar as frequências superiores à metade da frequência amostrada e a assegurar a limitação da máxima componente de frequência do sistema, conforme entendimento proposto pelo teorema de Nyquist.

Existem vários tipos de filtros possíveis para implementação: analógicos passivos, analógicos ativos, digitais, de ordem simples ou superior. Nesta subseção, são descritos dois tipos de filtros passa-baixas a serem utilizados (cascateados ou não) como filtros *anti-aliasing*.

2.4.1 Filtro Passa-Baixas de Primeira Ordem Analógico (RC)

Seja o circuito RC ilustrado na Figura 5:

Figura 5 – Circuito RC



Fonte: Próprio Autor.

Tal circuito funciona como um divisor de tensão, no qual a tensão de saída, V_o , é dada, em função da tensão de entrada, V_i , segundo as equações (7) e (8):

$$V_o = \frac{Z_C}{Z_R + Z_C} V_i \quad (7)$$

$$V_o = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} V_i = \frac{1}{1 + j\omega RC} V_i = \left(\frac{1}{1 + (\omega RC)^2} - j \frac{\omega RC}{1 + (\omega RC)^2} \right) V_i \quad (8)$$

Sendo:

j – A unidade imaginária $j = \sqrt{-1}$;

ω – A frequência angular sinusoidal da tensão de entrada (em rad/s);

R – O valor de resistência do resistor (em Ω);

C – O valor de capacitância do capacitor (em F).

Define-se a função de frequência $T(\omega) = \frac{V_o}{V_i}$. Obtém-se o módulo de $T(\omega)$ segundo as equações (9) e (10):

$$|T(\omega)| = \sqrt{\left(\frac{1}{1 + (\omega RC)^2} \right)^2 + \left(\frac{\omega RC}{1 + (\omega RC)^2} \right)^2} = \frac{1}{\sqrt{1 + (\omega RC)^2}} \quad (9)$$

$$|T(\omega)|_{dB} = 20 \cdot \log \left(\frac{1}{\sqrt{1 + (\omega RC)^2}} \right) \quad (10)$$

Define-se também a frequência ω_c (frequência de corte) como a frequência que atenua a amplitude por um fator de $\frac{1}{\sqrt{2}}$ em relação à amplitude máxima ($\approx 0,71 \approx -3dB$).

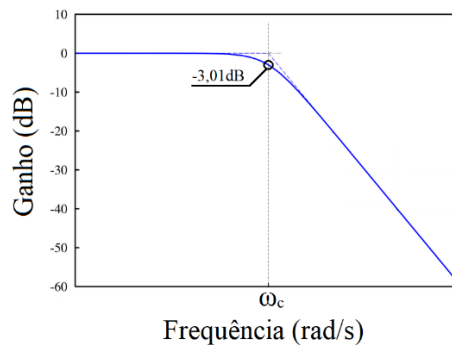
Calcula-se, portanto, ω_c a partir da equação (11):

$$|T(\omega_c)| = \frac{1}{\sqrt{2}} = \frac{1}{\sqrt{1 + (\omega_c RC)^2}} \quad (11)$$

$$\omega_c = \frac{1}{RC} \quad (12)$$

O comportamento gráfico de $|T(\omega)|_{dB}$ está ilustrado na Figura 6.

Figura 6 – Módulo de $|T(\omega)|_{dB}$



Fonte: Próprio Autor.

É possível constatar a tendência de atenuação da amplitude com o aumento do valor da frequência. Isto caracteriza este arranjo como um filtro passa-baixas. Deve-se especificar os valores de R e C de modo a projetar-se um filtro com frequência de corte desejada.

2.4.2 Filtro Passa-Baixas de Primeira Ordem Digital

É possível calcular uma função de transferência discreta de forma a refletir um comportamento semelhante a um filtro analógico passa-baixas de primeira ordem descrito na subseção 2.4.1.

Seja a relação entre a tensão de saída e entrada no circuito RC ($\frac{V_o}{V_i}$) dada na equação (8) na subseção anterior, definida, desta feita, pela equação (13):

$$\frac{V_o}{V_i} = \frac{1}{1 + j\omega RC} \quad (13)$$

Para frequências puramente sinusoidais, o operador de Laplace (s) pode ser descrito segundo a equação (14):

$$s = j\omega \quad (14)$$

Logo, define-se a função transferência $H(s)$, segundo a equação (15), evidenciando o circuito RC como um sistema contínuo de primeira ordem:

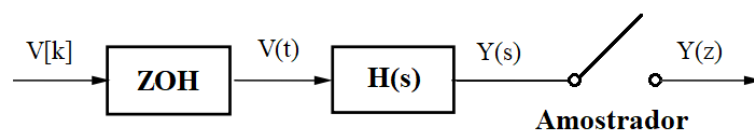
$$\frac{V_o(t)}{V_i(t)} = H(s) = \frac{1}{1 + (RC)s} \quad (15)$$

A relação exibida pela equação (12), aplicada na definição de $H(s)$ segundo a equação (15), origina a equação (16) a seguir, dando a forma geral de um filtro contínuo de primeira ordem com frequência de corte ω_c :

$$H(s) = \frac{1}{1 + \left(\frac{1}{\omega_c}\right)s} = \frac{\omega_c}{\omega_c + s} \quad (16)$$

Discretiza-se o filtro de primeira ordem contínuo, utilizando um ZOH como método de discretização (ilustrado pela Figura 7), de acordo com a equação (17).

Figura 7 – Diagrama de Blocos (Discretização com ZOH)



Fonte: Próprio Autor.

$$\frac{Y(z)}{V(z)} = \frac{z-1}{z} \cdot Z\left\{\frac{H(s)}{s}\right\} \quad (17)$$

Transforma-se o termo $\frac{H(s)}{s}$ em frações parciais, segundo a equação (18):

$$\frac{H(s)}{s} = G(s) = \frac{1}{s} \frac{\omega_c}{\omega_c + s} = \frac{1}{s} - \frac{1}{\omega_c + s} \quad (18)$$

Aplicando-se a transformada inversa de Laplace em $G(s)$, obtém-se a equação (19):

$$\mathcal{L}^{-1}\{G(s)\} = g(t) = 1 - e^{-\omega_c t} \quad (19)$$

Ao discretizar-se a função contínua $g(t)$ a partir da equação (19), sendo T_s o período de amostragem considerado, obtém-se:

$$g(kT_s) = 1 - e^{-\omega_c kT_s} \quad (20)$$

$$G(z) = \frac{z}{z-1} - \frac{z}{z-e^{-\omega_c T_s}} = \frac{z}{z-1} \frac{(1-e^{-\omega_c T_s})}{(z-e^{-\omega_c T_s})} \quad (21)$$

Substituindo-se (21) em (17) obtém-se, finalmente, a função de transferência discretizada de um filtro passa-baixas contínuo de primeira ordem, exibida pela equação (22):

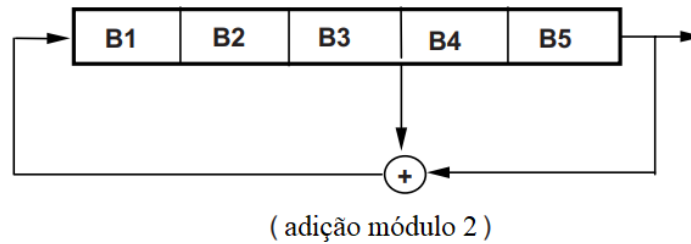
$$\frac{Y(z)}{V(z)} = \frac{z-1}{z} \cdot \frac{z}{z-1} \cdot \frac{(1-e^{-\omega_c T_s})}{(z-e^{-\omega_c T_s})} = \frac{1-e^{-\omega_c T_s}}{z-e^{-\omega_c T_s}} \quad (22)$$

2.5 Projeto do Sinal de Excitação

Para aumentar a qualidade do modelo a ser identificado é necessário que o sinal de entrada seja rico em frequências, de forma a fazer com que a saída do sistema revele suas características dinâmicas. Para tal, um possível tipo de sinal a ser aplicado é o proveniente de uma sequência binária pseudoaleatória (SBPA). Uma SBPA tem como características excitar os diferentes modos de um sistema, bem como, na identificação, prevenir colunas iguais na matriz de covariâncias (COELHO A. e COELHO L., 2004, p. 123) por meio da persistência de excitação do sistema, além de ser periódico com seu período dado pelo comprimento e correspondente duração de sua sequência numérica.

Uma SBPA é gerada mediante o deslocamento temporal de bits por meio de um registrador de deslocamento, sendo um novo bit criado a partir de operação definida entre dois outros bits do registrador a cada período de chaveamento da sequência, conforme ilustrados na Figura 8.

Figura 8 – Exemplo de Geração de SBPA



Fonte: (LANDAU; ZITO, 2006, p. 230, tradução nossa)

Uma sugestão de projeto para sinal do tipo SBPA é dada pelas equações e inequações (23), (24), (25) (RIVERA e FLORES, 2000, apud COELHO A. e COELHO L., 2004, p. 125) (BROSILOW e JOSEPH, 2002, apud COELHO A. e COELHO L., 2004, p. 125):

$$\underline{\omega} = \frac{1}{\beta_s \tau_R} \leq \omega \leq \frac{\alpha_S}{\tau_L} = \bar{\omega} \quad (23)$$

$$T_{sw} \leq \frac{2,8\tau_L}{\alpha_S} \quad (24)$$

$$N_s = 2^{n_r} - 1 \geq \frac{2\pi\beta_s\tau_R}{T_{sw}} \quad (25)$$

Sendo:

$\underline{\omega}$ – A frequência mínima de excitação do sinal de entrada (em rad/s);

$\bar{\omega}$ – A frequência máxima de excitação do sinal de entrada (em rad/s);

ω – A faixa de frequências a ser de fato excitada (em rad/s);

T_{sw} – O tempo de chaveamento (da sequência);

n_r – O tamanho (número de bits) do registrador da SBPA;

N_s – O comprimento da SBPA;

τ_L – A menor constante de tempo do sistema;

τ_R – A maior constante de tempo do sistema;

α_S – Um fator representativo do tempo de resposta em malha fechada como um múltiplo do tempo de resposta em malha aberta;

β_s – Um fator representativo do tempo de estabilização do processo;

Manipulando algebricamente as equações e inequações (23), (24), (25) e tomando-se f_1 e f_2 como as frequências (em Hz) nas fronteiras de ω , tem-se, em (26) e (27) que:

$$f_1 \geq \frac{1}{(2^{n_r} - 1)T_{sw}} \quad (26)$$

$$f_2 \leq \frac{0,44}{T_{sw}} \quad (27)$$

Sendo estimados, portanto, f_1 e f_2 do sistema em malha aberta, é possível projetar uma SBPA, calculando-se os parâmetros T_{sw} e n_r da sequência, assegurando-se que o produto $n_r T_{sw}$ seja maior ou igual ao tempo de subida do sistema, de forma que o modelo identificado capture as suas informações de ganho em regime permanente (LANDAU; ZITO, 2006, p. 279).

2.6 Estimação da Complexidade do Modelo

É necessário estimar a complexidade do modelo a ser identificado antes de proceder com o algoritmo de identificação. Isto significa estimar a ordem dos polinômios que compõem o numerador e o denominador do modelo matemático a ser adotado, além do valor do atraso de propagação e de tempo morto do sistema. É frequentemente adotada, para este fim, a estratégia de tentativa e erro, ainda que tenham sido desenvolvidas técnicas complexas para este fim (LANDAU; ZITO, 2006, p. 203), envolvendo, em tese, algoritmos iterativos (LANDAU; ZITO, 2006, p. 267).

2.7 Cálculo dos Parâmetros do Modelo

Uma vez definidos o sinal de entrada, a estrutura e a complexidade do modelo a ser adotado, é necessário, após a realização dos experimentos práticos e a coleta de dados, calcular os parâmetros do modelo matemático do sistema (método *offline*).

2.7.1 Condicionamento do Sinal

Uma vez que os modelos estruturais utilizados para a identificação do sistema são baseados em modelos dinâmicos, isto é, expressos em variações no entorno de um ponto de

operação, é necessário, antes da identificação, eliminar de ambos os sinais as componentes contínuas correspondentes ao ponto de operação.

Caso o ponto de operação seja estacionário, é preciso condicionar o conjunto de dados dos sinais de entrada e saída adquiridos, de forma a subtrair de cada dado amostrado o valor médio do conjunto de dados, criando assim um novo conjunto de dados com valor médio nulo (LANDAU; ZITO, 2006, p. 284).

2.7.2 Regressão Linear (Mínimos Quadrados Não Recursivo)

No século XVIII, Karl Friedrich Gauss desenvolveu o Princípio dos Mínimos Quadrados com o objetivo de calcular e prever órbitas de planetas e cometas a partir de observações realizadas (COELHO A. e COELHO L., 2004, p.100). Segundo Ljung e Sönderström (1983, *apud* COELHO A. e COELHO L., 2004, p.100),

O valor mais provável das grandezas desconhecidas é a que minimiza a soma dos quadrados da diferença entre os valores atualmente observados e os valores calculados multiplicados por números que medem o grau de precisão, onde quanto mais precisa a medida, maior a sua ponderação.

Seja um sistema genérico caracterizado pelo modelo exibido na equação (2). Ao desenvolvê-la na forma de equações de diferenças, obtém-se a equação (28):

$$y(t) = -a_1y(t-1) - a_2y(t-2) - \dots - a_{nA}y(t-nA) + b_1u(t-d-1) + b_2u(t-d-2) + \dots + b_{nB}u(t-d-nB) + e(t) \quad (28)$$

Seja $\varphi(t)$ o vetor de medidas, definido na equação (29), como:

$$\varphi^T(t) = [-y(t-1) - y(t-2) - \dots - y(t-nA) + u(t-d-1) + u(t-d-2) + \dots + u(t-d-nB)] \quad (29)$$

Seja, por sua vez, $\theta(t)$ o vetor de parâmetros, definido na equação (30), como:

$$\theta^T(t) = [a_1 \ a_2 \ \dots \ a_{nA} \ b_1 \ b_2 \ \dots \ b_{nB}] \quad (30)$$

É possível reescrever a equação (28) como um modelo de regressão linear da forma $y(t) = \varphi(t)^T\theta(t) + e(t)$, com N medições realizadas, obtendo-se, na equação (31):

$$\begin{bmatrix} y(0) \\ y(1) \\ \dots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} \phi^T(0) \\ \phi^T(1) \\ \dots \\ \phi^T(N-1) \end{bmatrix} \theta + \begin{bmatrix} e(0) \\ e(1) \\ \dots \\ e(N-1) \end{bmatrix} \quad (31)$$

Sendo a representação matricial da equação (31) dada por $Y = \phi\theta + E$, observa-se que, uma vez que a matriz ϕ é não-quadrada, é necessário estimar os valores de θ . Utiliza-se, portanto, o procedimento dos Mínimos Quadrados Não Recursivo.

Seja $\hat{\theta}$ a estimativa do vetor de parâmetros e \hat{Y} a melhor previsão de saída do sistema, tem-se segundo a equação (32), que:

$$\hat{Y} = \phi\hat{\theta} \quad (32)$$

Seja ε o erro de previsão. Este é definido segundo a equação (33):

$$\varepsilon = Y - \hat{Y} = Y - \phi\hat{\theta} \quad (33)$$

Define-se J como o estimador dos mínimos quadrados ponderados (estimador de Markov), o qual é obtido pelo critério de minimização do erro quadrático, conforme exibido na equação (34).

$$J = \|Y - \phi\hat{\theta}\|_{min}^2 \quad (34)$$

Com determinado desenvolvimento algébrico matricial, é possível provar que a estimativa do vetor de parâmetros se dá de acordo com a equação (35) (COELHO A. e COELHO L., 2004, p.104), tendo como condição necessária a não-singularidade da matriz $\phi^T\phi$ (condição de excitação) (COELHO A. e COELHO L., 2004, p.108).

$$\hat{\theta} = [\phi^T\phi]^{-1}\phi^TY \quad (35)$$

2.7.3 Regressão Não Linear (Método de Gauss-Newton)

Devido às possíveis não linearidades do sistema, pode valer à pena utilizar-se de um método de regressão que admita que a função de transferência a ser identificada seja não linear. Utilizando uma expansão via série de Taylor, sendo truncada depois da primeira derivada (representação linearizada da função original), o método de Gauss-Newton é um algoritmo

iterativo o qual possui o objetivo de minimizar a soma do quadrado dos resíduos entre os dados e as equações (funções de transferência) não lineares (CHAPRA e CANALE, 2011, p. 403).

Seja a equação (36) a expressão genérica da relação entre uma função não linear e um conjunto de dados amostrados, onde y_i é o valor medido; $f(x_i; a_0, a_1, a_2, \dots, a_m)$ é a função da variável independente, x_i , e função não linear dos parâmetros $a_0, a_1, a_2, \dots, a_m$; e e_i um erro aleatório de medição.

$$y_i = f(x_i; a_0, a_1, a_2, \dots, a_m) + e_i \quad (36)$$

Para um modelo com dois parâmetros, $f(x_i; a_0, a_1)$, doravante denominada apenas $f(x_i)$, tem-se, segundo a equação (37), a expansão de $f(x_i)$ via série de Taylor e truncamento após a primeira derivada:

$$f(x_i)_{j+1} = f(x_i)_j + \frac{\partial f(x_i)_j}{\partial a_0} \Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1} \Delta a_1 \quad (37)$$

Sendo j a aproximação inicial, $j+1$ a previsão, $\Delta a_0 = a_{0,j+1} - a_{0,j}$ e $\Delta a_1 = a_{1,j+1} - a_{1,j}$. Substituindo-se (37) em (36), obtém-se a equação (38):

$$y_i - f(x_i)_j = \frac{\partial f(x_i)_j}{\partial a_0} \Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1} \Delta a_1 + e_i \quad (38)$$

Representando-se a equação (38) em sua forma matricial, obtém-se a equação (39):

$$\{D\} = [Z_j]\{\Delta A\} + \{E\} \quad (39)$$

Onde $\{D\}$ é o vetor que contém os desvios entre os valores medidos e os valores da função; $[Z_j]$ é a matriz das derivadas parciais da função (matriz Jacobiana), calculadas na aproximação inicial j , sendo n o número de pontos experimentais disponíveis e $\partial f_i / \partial a_k$ a derivada parcial da função em relação ao k -ésimo parâmetro calculado no i -ésimo ponto dado; e $\{\Delta A\}$ é o vetor que contém a variação dos valores dos parâmetros adotados. $\{D\}$, $[Z_j]$ e $\{\Delta A\}$, respectivamente, são dadas segundo as equações (40), (41) e (42).

$$\{D\} = \begin{bmatrix} y_1 - f(x_1) \\ y_2 - f(x_2) \\ \dots \\ y_n - f(x_n) \end{bmatrix} \quad (40)$$

$$[Z_j] = \begin{bmatrix} \frac{\partial f_1}{\partial a_0} & \frac{\partial f_1}{\partial a_1} \\ \frac{\partial f_2}{\partial a_0} & \frac{\partial f_2}{\partial a_1} \\ \dots & \dots \\ \frac{\partial f_n}{\partial a_0} & \frac{\partial f_n}{\partial a_1} \end{bmatrix} \quad (41)$$

$$\{\Delta A\} = \begin{bmatrix} \Delta a_0 \\ \Delta a_1 \\ \dots \\ \Delta a_m \end{bmatrix} \quad (42)$$

De forma análoga à solução da regressão linear segundo a equação (35), obtém-se, utilizando a matriz inversa, a solução da equação (39), exibida na equação (43):

$$\Delta A = [Z_j^T Z_j]^{-1} Z_j^T D \quad (43)$$

O ajuste dos parâmetros é feito iterativamente, na forma das equações (44) e (45), até que atinja-se um critério de parada aceitável (convergência).

$$a_{0,j+1} = a_{0,j} + \Delta a_0 \quad (44)$$

$$a_{1,j+1} = a_{1,j} + \Delta a_1 \quad (45)$$

2.8 Validação do Modelo

Validar um modelo é investigar, dada sua intenção de uso, se este pode ser aceito ou não. Tal conceito está absolutamente relacionado à qualidade do modelo adquirido, refletida na estabilidade do modelo e na sua habilidade em reproduzir o comportamento do sistema (LJUNG; GLAD, 1994, p. 283).

2.8.1 Índices de Validação

Em meio à diversas técnicas de validação, é possível calcular alguns índices de desempenho do modelo identificado (BROSILOW; JOSEPH, 2002, *apud* COELHO A. e COELHO L., 2004, p.122). A seguir, são discutidos alguns índices de desempenho.

O somatório do erro quadrático (SEQ) e o desvio absoluto percentual médio (DAPM) são dados respectivamente nas equações (46) e (47):

$$SEQ = \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \quad (46)$$

$$DAPM = \frac{1}{N} \sum_{k=1}^N \frac{|y(k) - \hat{y}(k)|}{y(k)} [x \ 100\%] \quad (47)$$

O SEQ e o DAPM têm um significado evidente ao mensurar o erro entre os valores medidos e as saídas resultantes do modelo identificado.

O coeficiente de correlação múltipla (R^2), que provê um sentido mais prático ao SEQ, é dado na equação (48), sendo \bar{y} a média das N amostras da experimentação:

$$R^2 = 1 - \frac{\sum_{k=1}^N [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^N [y(k) - \bar{y}]^2} = 1 - \frac{SEQ}{\sum_{k=1}^N [y(k) - \bar{y}]^2} \quad (48)$$

Uma exata adequação entre os dados experimentais e o modelo é dada por um valor de $R^2 = 1$. R^2 entre 0,9 e 1 pode ser considerado suficiente para diversas aplicações (COELHO A. e COELHO L., 2004, p.123).

2.8.2 Análises de Resíduos (Autocorrelação e Correlação Cruzada)

Ainda com a intenção de caracterizar a qualidade e, portanto, a validade do sistema identificado, é necessário efetuar uma análise de resíduos.

Caso a estrutura da planta seja representativa da realidade e os graus dos polinômios A e B, bem como o valor do atraso de propagação e tempo morto da planta estiverem corretos, o erro de predição tenderá assintoticamente a um ruído branco (variável aleatória) (LANDAU; ZITO, 2006, p. 256), logo, seus valores de autocorrelação serão nulos para todos os valores de

atraso (deslizamento) maiores ou iguais a 1. Em situações reais, entretanto, mediante erros de estrutura, erros de ordem, não linearidades, entre outros fatores, Landau e Zito (2006, p. 257) sugerem um critério prático de validação definido na equação (50), sendo $RN(i)$ a função de autocorrelação normalizada, exposta na equação (49), N o número de amostras e $\varepsilon(t)$ a função erro centrada (erros de predição subtraídos do valor médio dos erros).

$$RN(i) = \frac{\sum_{t=1}^N \varepsilon(t)\varepsilon(t-i)}{\sum_{t=1}^N \varepsilon^2(t)} \quad (49)$$

$$|RN(i)| \leq \frac{1,8 \text{ a } 2,17}{\sqrt{N}} \quad (50)$$

É necessário notar, entretanto, que a equação (49) parte da premissa de um intervalo de confiança de 3% a 7% na hipótese de que a sequência $RN(i)$ possui uma distribuição assintoticamente gaussiana (normal), com média zero e variância igual a $1/\sqrt{N}$ (LANDAU; ZITO, 2006, p. 257).

A generalidade do modelo, por sua vez, pode ser analisada através da função correlação cruzada, $r_{u\xi}(k)$, entre o vetor de resíduos e o sinal de entrada, conforme exposto através da equação (51). Caso não haja correlação entre os sinais, isto é, $r_{u\xi}(k)$ seja nulo $\forall k$ inteiro $\in [0, N]$, é validada a generalidade do modelo, implicando que uma nova identificação com outro conjunto de dados de entrada resultará em um modelo semelhante.

$$r_{u\xi}(k) = \frac{\sum_{t=0}^N [\varepsilon(k)(y(t-k) - \bar{y})]}{\sqrt{\sum_{t=0}^N [\varepsilon(k)]^2} \sqrt{\sum_{t=0}^N [y(t-k) - \bar{y}]^2}} \quad (51)$$

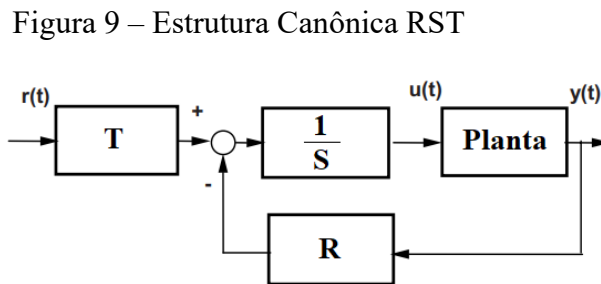
O critério de validação para este teste de generalidade é idêntico ao critério de validação para o teste de autocorrelação, definido segundo a equação (50) (LANDAU; ZITO, 2006, p. 266).

3 CONTROLE DIGITAL

Um sistema de controle digital possui diversas vantagens sobre um sistema de controle analógico. Além da flexibilidade trazida por uma implementação via software, abre-se um leque de possibilidades para utilização de outros algoritmos mais complexos e eficientes que um PID analógico (LANDAU; ZITO, 2006, p. 85).

3.1 A Estrutura RST

A estrutura canônica RST é utilizada para padronizar e facilitar a implementação de estruturas de controle digitais. Padronizando-se uma mesma estrutura lógica, há eventuais mudanças apenas na memória dos componentes do controlador (número e valor de coeficientes), os quais variarão dependendo da complexidade do sistema (LANDAU; ZITO, 2006, p. 86). Na Figura 9, é exibida a representação em diagrama de blocos de uma estrutura canônica do tipo RST.



Fonte: Próprio Autor

Admite-se que a planta indicada no diagrama de blocos da Figura 9 representa a dinâmica em cascata DAC + ZOH + Sistema contínuo + ADC e é representada segundo um modelo ARX, exposto na equação (52).

$$H(z^{-1}) = \frac{z^{-d}B(z^{-1})}{A(z^{-1})} \quad (52)$$

Calcula-se a partir do desenvolvimento da equação (53), a expressão que relaciona o sinal de referência, $r(t)$, e o sinal de saída, $y(t)$, do sistema em malha fechada.

$$\frac{y(t)}{r(t)} = H_{CL}(z^{-1}) = \frac{\frac{T(z^{-1})z^{-d}B(z^{-1})}{S(z^{-1})A(z^{-1})}}{1 + \frac{R(z^{-1})z^{-d}B(z^{-1})}{S(z^{-1})A(z^{-1})}} \quad (53)$$

$$H_{CL}(z^{-1}) = \frac{\frac{T(z^{-1})z^{-d}B(z^{-1})}{S(z^{-1})A(z^{-1})}}{\frac{S(z^{-1})A(z^{-1}) + R(z^{-1})z^{-d}B(z^{-1})}{S(z^{-1})A(z^{-1})}} \quad (54)$$

$$H_{CL}(z^{-1}) = \frac{T(z^{-1})z^{-d}B(z^{-1})}{S(z^{-1})A(z^{-1}) + R(z^{-1})z^{-d}B(z^{-1})} \quad (55)$$

Seja $P(z^{-1})$ o polinômio representativo dos polos de malha fechada do sistema. Tem-se, segundo as equações (56) e (57), a seguinte identidade:

$$P(z^{-1}) = 1 + p_1 \cdot z^{-1} + p_2 \cdot z^{-2} + \dots + p_{nP} \cdot z^{-nP} \quad (56)$$

$$P(z^{-1}) = S(z^{-1})A(z^{-1}) + R(z^{-1})z^{-d}B(z^{-1}) \quad (57)$$

É possível, portanto, projetar $P(z^{-1})$ de forma que o sistema em malha fechada atinja determinada performance e calcular $R(z^{-1})$, $S(z^{-1})$ e $T(z^{-1})$ para permitir a implementação do controlador.

3.2 Alocação de Polos

A alocação de polos é uma estratégia que permite o projeto de controladores RST digitais para sistemas estáveis e instáveis sem restrição acerca dos graus dos polinômios $A(z^{-1})$ e $B(z^{-1})$, nem em relação ao tempo de atraso do sistema ou presença de zeros estáveis ou instáveis (LANDAU; ZITO, 2006, p. 105).

3.2.1 Determinação dos Polos de Malha Fechada

Para definir a performance do sistema em malha fechada, pode ser utilizado como base um sistema contínuo de segunda ordem discretizado, no qual seus polos serão utilizados como polos dominantes do sistema projetado.

Um sistema de segunda ordem em tempo contínuo pode ser expresso, em sua forma-padrão, segundo a equação (58) (OGATA, 2010, p. 151):

$$SYS(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (58)$$

Duas possíveis especificações de resposta transitória do sistema de segunda ordem exemplificado na equação (58), são dadas nas equações (59), (60) (OGATA, 2010, p. 156):

$$Te_{2\%} \approx \frac{4}{\xi\omega_n} \quad (59)$$

$$UP\% = 100 \cdot e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} \quad (60)$$

Sendo:

$SYS(s)$ – A função de transferência generalista de um sistema contínuo de segunda ordem com atraso;

$UP\%$ – A ultrapassagem percentual do sistema à entrada degrau;

$Te_{2\%}$ – O tempo de estabelecimento de regime permanente (com $\pm 2\%$ de margem) à entrada degrau;

ω_n – A frequência natural do sistema;

ξ – O coeficiente de amortecimento do sistema;

t – O tempo de atraso de resposta do sistema em relação à entrada.

Após definição das características de desempenho do sistema contínuo em questão, discretiza-se sua função de transferência por meio de um segurador de ordem-zero, obtendo-se a função discreta na forma da equação (61) (LANDAU; ZITO, 2006, p. 53).

$$H(z^{-1}) = \frac{z^{-d}(b_1 z^{-1} + b_2 z^{-2})}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (61)$$

Sendo d o atraso de resposta do sistema (em número de amostras), a_1 , a_2 , b_1 e b_2 os coeficientes resultantes da discretização e $z_{1,2}$ os polos do sistema discretizado, sendo estes exibidos na equação (62), com T_s representando o período de amostragem:

$$z_{1,2} = e^{-\xi\omega_n T_s \pm j\omega_n \sqrt{1-\xi^2} T_s} \quad (62)$$

De acordo com a equação diofantina, (57), o grau do polinômio $P(z^{-1})$ será função dos graus dos polinômios $A(z^{-1})$ e $B(z^{-1})$, bem como da estrutura a ser escolhida para $R(z^{-1})$ e $S(z^{-1})$. Logo, será necessário adicionar polos auxiliares aos polos dominantes calculados, tendo a forma do polinômio $P(z^{-1})$, portanto, definida na equação (63):

$$P(z^{-1}) = P_D(z^{-1})P_F(z^{-1}) \quad (63)$$

Sendo:

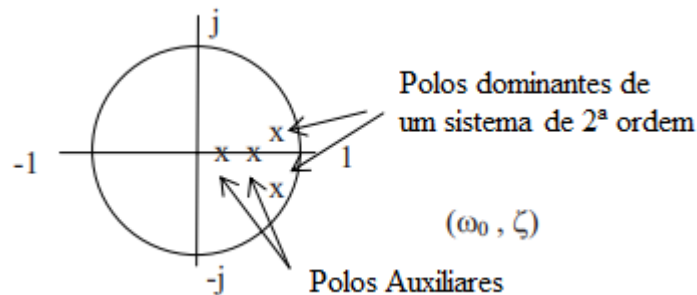
$P_D(z^{-1})$ – O polinômio contendo os polos dominantes;

$P_F(z^{-1})$ – O polinômio contendo os polos auxiliares.

Segundo Landau e Zito (2006, p. 108, tradução nossa), “como regra geral, os polos denominados ‘auxiliares’ são mais rápidos que os ‘dominantes’. Isto é expresso, para o caso de sistemas em tempo discreto, pela propriedade na qual os as raízes de $P_F(z^{-1})$ devem ter parte real menor que a parte real das raízes de $P_D(z^{-1})$ ”.

Na Figura 10, ilustra-se a escolha de polos auxiliares e dominantes num determinado sistema.

Figura 10 – Polos Dominantes e Auxiliares (Plano Z)



Fonte: (LANDAU; ZITO, 2006, p. 93, tradução nossa)

É ainda possível escolher os polos auxiliares a serem utilizados de modo a introduzir filtros em determinadas regiões do espectro, atenuando ruídos de medição, amenizar variações no sinal de controle e melhorando a robustez do sistema (LANDAU; ZITO, 2006, p. 108).

3.2.2 Regulação

De forma semelhante à definição proposta de $P(z^{-1})$ na equação (63), é interessante pré-especificar partes fixas dos polinômios $R(z^{-1})$ e $S(z^{-1})$ antes da realização do cálculo caracterizado pela equação diofantina, (57). Fatora-se, portanto, os polinômios $R(z^{-1})$ e $S(z^{-1})$ de acordo com as equações (64) e (65), definindo-se portanto os filtros $H_R(z^{-1})$ e $H_S(z^{-1})$, bem como os polinômios $R'(z^{-1})$ e $S'(z^{-1})$:

$$R(z^{-1}) = R'(z^{-1})H_R(z^{-1}) \quad (64)$$

$$S(z^{-1}) = S'(z^{-1})H_S(z^{-1}) \quad (65)$$

Os polinômios pré-especificados são os filtros $H_R(z^{-1})$ e $H_S(z^{-1})$. Os polinômios $R'(z^{-1})$ e $S'(z^{-1})$, por sua vez, serão calculados e terão estrutura conforme exibido nas equações (66) e (67):

$$R'(z^{-1}) = r'_0 + r'_1 \cdot z^{-1} + \dots + r'_{nR'} \cdot z^{-nR'} \quad (66)$$

$$S'(z^{-1}) = 1 + s'_1 \cdot z^{-1} + \dots + s'_{nS'} \cdot z^{-nS'} \quad (67)$$

A equação diofantina, (57), torna-se, portanto, a expressão exposta na equação (68).

$$P(z^{-1}) = A(z^{-1})H_S(z^{-1})S'(z^{-1}) + z^{-d}B(z^{-1})H_R(z^{-1})R'(z^{-1}) \quad (68)$$

Tendo-se $H_R(z^{-1})$ e $H_S(z^{-1})$ como polinômios fixos a serem escolhidos dada a necessidade do projeto, define-se $A'(z^{-1})$ e $B'(z^{-1})$ conforme as equações (69) e (70), com a restrição de que estes sejam coprimos (LANDAU; ZITO, 2006, p. 110).

$$A'(z^{-1}) = A(z^{-1})H_S(z^{-1}) \quad (69)$$

$$B'(z^{-1}) = z^{-d}B(z^{-1})H_R(z^{-1}) \quad (70)$$

Reescrevendo-se, portanto, a equação (68) na forma dada na equação (71):

$$P(z^{-1}) = A'(z^{-1})S'(z^{-1}) + B'(z^{-1})R'(z^{-1}) \quad (71)$$

As condições sobre as ordens dos polinômios $P(z^{-1})$, $R'(z^{-1})$ e $S'(z^{-1})$, as quais oferecem uma solução única de menor ordem são dadas nas expressões (72), (73) e (74) (LANDAU; ZITO, 2006, p. 110).:

$$n_p = \text{grau de } P(z^{-1}) \leq n_A + n_{H_S} + n_B + d - 1 \quad (72)$$

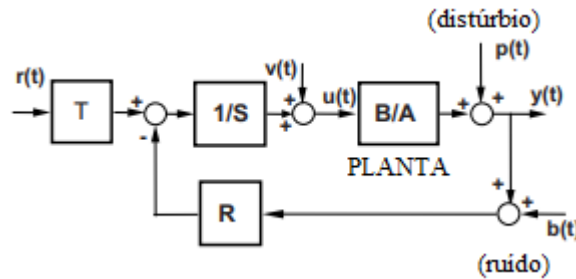
$$n_{S'} = \text{grau de } S'(z^{-1}) = n_B + n_{H_R} + d - 1 \quad (73)$$

$$n_{R'} = \text{grau de } R'(z^{-1}) = n_A + n_{H_S} - 1 \quad (74)$$

3.2.3 Definição dos filtros H_r e H_s

Como ferramenta para mensurar os impactos de distúrbios diversos característicos de um sistema real, considera-se o seguinte diagrama ainda baseado na estrutura canônica RST, exibido na Figura 11:

Figura 11 – Controle Digital na Presença de Distúrbios e Ruído



Fonte: (LANDAU; ZITO, 2006, p. 62, tradução nossa).

A função de transferência que caracteriza a relação entre o distúrbio $p(t)$ e a saída da planta, $y(t)$ (função de sensibilidade da saída), é calculada a partir da equação (75).

$$\frac{y(t)}{p(t)} = S_{yp}(z^{-1}) = \frac{1}{1 + \frac{-R(z^{-1})B(z^{-1})}{S(z^{-1})A(z^{-1})}} \quad (75)$$

$$S_{yp}(z^{-1}) = \frac{S(z^{-1})A(z^{-1})}{S(z^{-1})A(z^{-1}) + R(z^{-1})z^{-d}B(z^{-1})} \quad (76)$$

Logo, esta função permite a caracterização da performance do sistema do ponto de vista da rejeição de distúrbios. Caso seja necessário uma rejeição perfeita de distúrbios em regime permanente, é desejado que $S_{yp}(z^{-1})$ possua um termo $(1 - z^{-1})$ no numerador, o que leva a valores de ganho nulos para $z = 1$ (LANDAU; ZITO, 2006, p. 62), podendo a parte fixa do polinômio $S(z^{-1})$, $H_S(z^{-1})$, ser especificada para esta finalidade.

A função de transferência que caracteriza a relação entre o distúrbio $p(t)$ e a entrada da planta, $u(t)$ (função de sensibilidade da entrada), é calculada a partir da equação (77).

$$\frac{y(t)}{u(t)} = S_{up}(z^{-1}) = \frac{\frac{-R(z^{-1})}{S(z^{-1})}}{1 + \frac{R(z^{-1})B(z^{-1})}{S(z^{-1})A(z^{-1})}} \quad (77)$$

$$S_{up}(z^{-1}) = \frac{-A(z^{-1})R(z^{-1})}{S(z^{-1})A(z^{-1}) + R(z^{-1})z^{-d}B(z^{-1})} \quad (78)$$

A análise desta função permite avaliar a influência de distúrbios sobre a entrada da planta e especificar um fator do polinômio $R(z^{-1})$, neste caso, $H_R(z^{-1})$, se o controlador precisar reagir a distúrbios concentrados sobre determinada região de frequência (LANDAU; ZITO, 2006, p. 63).

3.2.4 Solução da Equação Diofantina

Os polinômios $R'(z^{-1})$ e $S'(z^{-1})$ advém da solução da equação diofantina expressa na equação (71), desenvolvida segundo a equação (79).

$$1 + p_1 \cdot z^{-1} + \dots + p_{nP} \cdot z^{-nP} = [1 + a'_1 \cdot z^{-1} + \dots + a'_{nA'} \cdot z^{-nA'}][1 + s'_1 \cdot z^{-1} + \dots + s'_{nS'} \cdot z^{-nS'}] + [b'_1 \cdot z^{-1} + \dots + b'_{nB'} \cdot z^{-nB'}][r'_0 + r'_1 \cdot z^{-1} + \dots + r'_{nR'} \cdot z^{-nR'}] \quad (79)$$

A solução da equação polinomial (79) caracteriza-se, após as devidas multiplicações no segundo membro da equação, em inspecionar e igualar os coeficientes de mesma ordem polinomial em ambos os lados da equação, gerando um sistema linear cuja forma matricial equivalente é dada na equação (80) (LANDAU; ZITO, 2006, p. 108).

$$Mx = p \quad (80)$$

Sendo x , p e M matrizes descritas segundo as equações (81), (82) e (83):

$$x = [1, s'_1, \dots, s'_{nS'}, r'_0, r'_1, \dots, r'_{nR'}] \quad (81)$$

$$p = [1, p_1, \dots, p_{nP}, 0, \dots, 0] \quad (82)$$

$$M = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\ a'_1 & 1 & & \cdot & b'_1 & & & \cdot \\ a'_2 & & & 0 & b'_2 & & & b'_1 \\ \cdot & & & 1 & \cdot & & & b'_2 \\ \cdot & & & a'_1 & \cdot & & & \cdot \\ a'_{nA'} & & & a'_2 & b'_{nB'} & & & \cdot \\ 0 & & & \cdot & 0 & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & a'_{nA'} & 0 & 0 & 0 & b'_{nB'} \end{bmatrix} \quad (83)$$

A solução do sistema linear via matriz inversa, pode ser obtida segundo a equação (84).

$$x = M^{-1}p \quad (84)$$

Calculando-se x , obtém-se, portanto, os coeficientes dos polinômios $R'(z^{-1})$ e $S'(z^{-1})$.

3.2.5 Cálculo dos Polinômios R , S e T

Uma vez calculados os polinômios $R'(z^{-1})$ e $S'(z^{-1})$, aplica-se suas definições segundo às equações (64) e (65), mediante os filtros $H_R(z^{-1})$ e $H_S(z^{-1})$ escolhidos, obtendo-se, portanto, os polinômios $R(z^{-1})$ e $S(z^{-1})$.

Seja a função de transferência em malha fechada do sistema dada pela equação (55). Com a finalidade de não introduzir zeros adicionais no sistema, projeta-se $T(z^{-1})$ a garantir um ganho unitário em regime permanente.

Uma vez garantido que $S(z^{-1})$ possui um termo $(1 - z^{-1})$ no numerador, garante-se também que $S(1) = 0$. Com o objetivo de preservar um ganho unitário em malha fechada, obtém-se, segundo o desenvolvimento da equação (85), a equação (86).

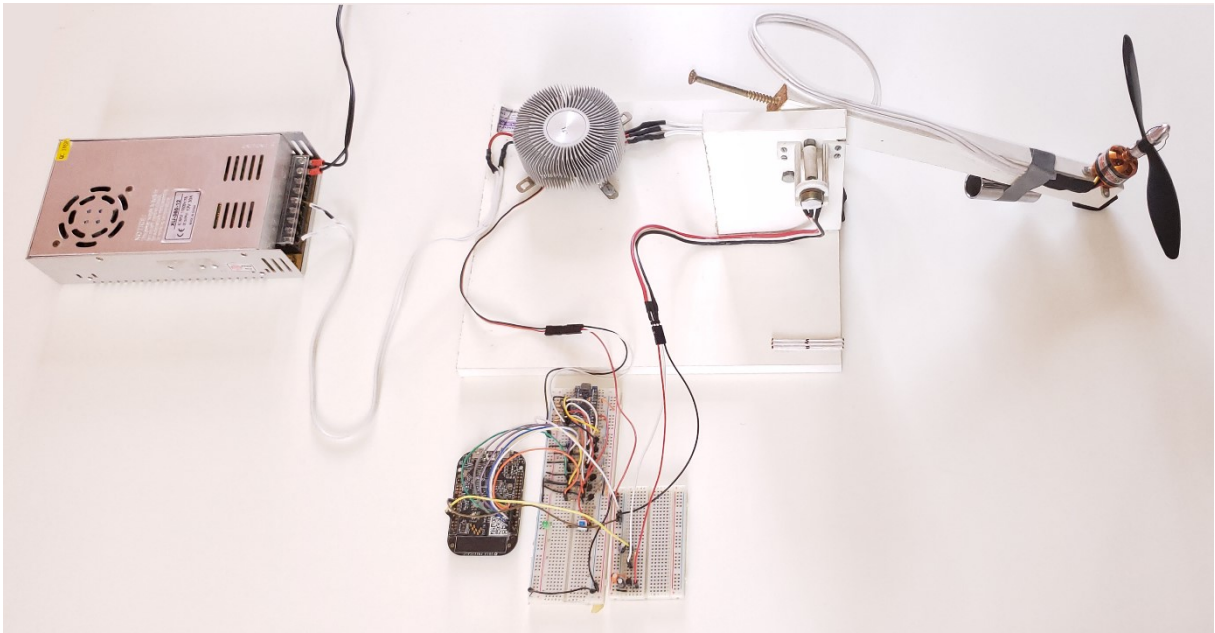
$$H_{CL}(1) = \frac{T(z^{-1})B(1)}{S(1)A(1) + R(1)B(1)} = 1 \quad (85)$$

$$T(z^{-1}) = R(1) \quad (86)$$

4 DESCRIÇÃO DA PLANTA

O sistema utilizado para a realização deste trabalho é descrito nesta seção e ilustrado na Figura 12.

Figura 12 – A planta e seus componentes



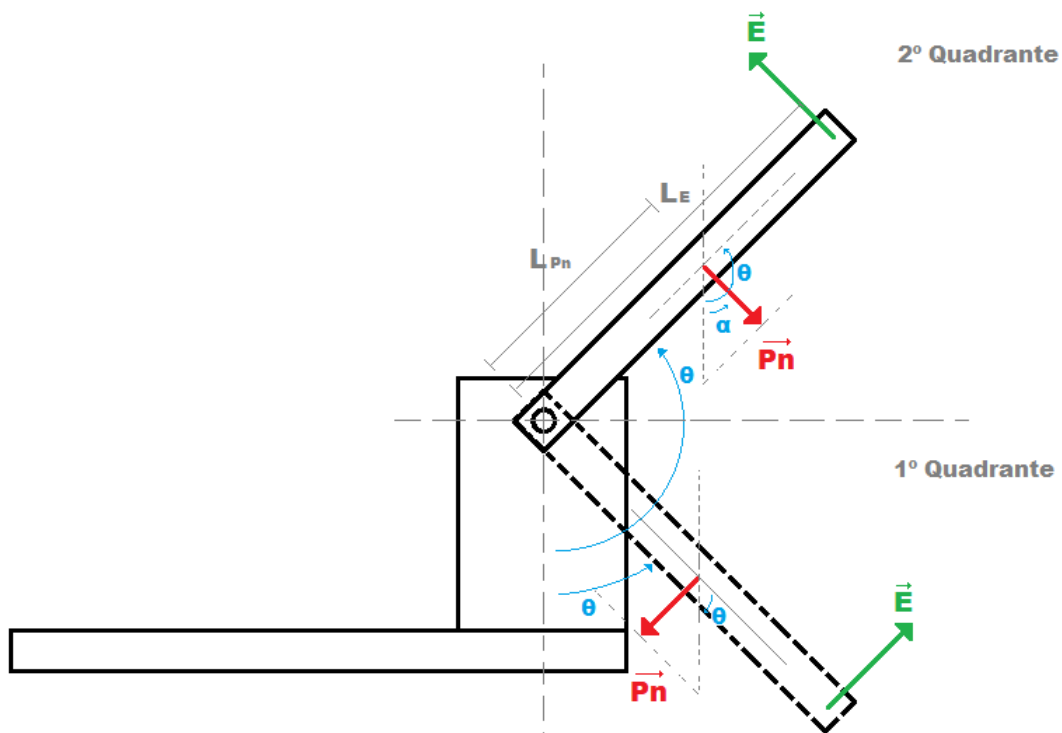
Fonte: Próprio Autor

Um motor elétrico do tipo BLDC (do inglês, *Brushless Direct Current*, traduzido como Sem Escovas de Corrente Contínua) instalado na ponta de um braço giratório, ao ser acionado, rotaciona a hélice a ele acoplada gerando um empuxo perpendicular ao braço móvel. Em contraposição, há, naturalmente, a componente normal do peso dos objetos que compõem o braço giratório, com direção perpendicular ao braço giratório. Quando o torque gerado pelo empuxo supera o torque gerado pelo peso normal do braço giratório, a este é imposto um movimento angular anti-horário até que seja atingido um novo ponto de equilíbrio em um ângulo superior. De forma análoga, um movimento angular horário é resultante, quando o valor do torque gerado pelo empuxo for inferior ao torque gerado pelo peso normal do braço giratório, até que seja atingido um novo ponto de equilíbrio em um ângulo inferior. Esta é uma análise que despreza possíveis perdas diversas no sistema.

4.1 Análise Mecânica

Conforme mostrado na Figura 13, a amplitude total de movimento do sistema foi dividida em dois quadrantes, sendo o primeiro caracterizado pelo ângulo de rotação do braço giratório menor ou igual a 90° em relação à base de apoio do sistema, e o segundo quadrante caracterizado pelo ângulo maior que 90° .

Figura 13 – Diagrama de Corpo Livre da Planta



Fonte: Próprio Autor

Ao aplicar-se a segunda lei de Newton em sua forma rotacional, supondo um movimento ascendente, obtém-se a equação (87).

$$\tau_{res}(t) = E(t) \cdot L_E - P_n(t) \cdot L_{Pn} = J \cdot \frac{d^2}{dt^2} [\theta(t)] \quad (87)$$

Sendo:

$\tau_{res}(t)$ – O valor instantâneo do torque resultante no eixo do braço giratório, em determinado instante t;

$E(t)$ – O valor instantâneo do empuxo gerado pelo motor através da hélice acoplada, em determinado instante t ;

L_E – O comprimento do braço de alavanca do empuxo em relação ao eixo do braço giratório;

$P_n(t)$ – A componente do peso total do braço giratório normal à sua superfície;

L_{P_n} – A distância entre o centro de gravidade do braço giratório (ponto de aplicação de $P_n(t)$) e seu eixo;

J – Momento de inércia total do braço giratório em relação ao seu eixo;

$\theta(t)$ – O valor instantâneo do ângulo de rotação do braço giratório, em determinado instante t .

Analisando-se $P_n(t)$ no primeiro quadrante, obtém-se a equação (88).

$$P_n(t) = m \cdot g \cdot \text{sen}[\theta(t)] \quad (88)$$

Em que:

m – O valor total da massa do braço giratório;

g – O valor da aceleração da gravidade.

Analisando-se, por sua vez, $P_n(t)$ no segundo quadrante, obtém-se a equação (89).

$$P_n(t) = m \cdot g \cdot \text{cos}[\alpha(t)] \quad (89)$$

Porém, visualiza-se trigonometricamente que:

$$\alpha(t) = \theta(t) - 90^\circ \quad (90)$$

Substituindo (90) em (89), tem-se:

$$P_n(t) = m \cdot g \cdot \text{cos}[\theta(t) - 90^\circ]$$

$$P_n(t) = m \cdot g \cdot \{ \text{cos}[\theta(t)] \cdot \text{cos}(90^\circ) + \text{sen}[\theta(t)] \cdot \text{sen}(90^\circ) \}$$

$$P_n(t) = m \cdot g \cdot \text{sen}[\theta(t)] \quad (91)$$

Logo, por meio de (88) e (91), conclui-se que $P_n(t)$ apresenta-se como uma mesma função ao longo de toda possível amplitude do movimento do braço giratório. Substituindo-se, portanto, (91) em (87), tem-se:

$$E(t) \cdot L_E - m \cdot g \cdot \text{sen}[\theta(t)] \cdot L_{Pn} = J \cdot \frac{d^2}{dt^2} [\theta(t)] \quad (92)$$

$$J \cdot \frac{d^2}{dt^2} [\theta(t)] = E(t) \cdot L_E - m \cdot g \cdot L_{Pn} \cdot \text{sen}[\theta(t)] \quad (93)$$

Considerando-se $A(t) \cdot \theta'' + B(t) \cdot \theta' + C(t) \cdot \theta = D(t)$ a forma geral de uma equação diferencial ordinária de segunda ordem, esta será linear se A, B, C e D forem funções apenas da variável independente t (SODRÉ, 2006).

Analogamente, uma vez que em (93) D é função de t e $\theta(t)$, a equação diferencial ordinária que representa a dinâmica do sistema será não-linear, considerando-se E(t) uma função de entrada do tipo degrau.

Vislumbra-se, portanto, a necessidade de, após definir-se um ponto de operação desejado da planta, linearizar seu comportamento em torno deste ponto de operação, visando à utilização de um controlador linear.

4.2 Aquisição e Processamento de Dados

De modo a permitir a aquisição e processamento dos valores de entrada e saída do sistema, foi utilizada a placa de desenvolvimento FRDM-KL25Z (ilustrada na Figura 14), da fabricante americana Freescale Semiconductors Ltd, hoje fundida à holandesa NXP Semiconductors N.V. Tal placa possui um microcontrolador com processador Arm Cortex-M0+, capaz de realizar quaisquer instruções com o desempenho necessário ao projeto.

Figura 14 – Placa de desenvolvimento
Freescale FRDM-KL25Z



Fonte: NXP Semiconductors N.V. (2020)

A FRDM-KL25Z possui, além de outros componentes, três interfaces que foram utilizadas:

- Uma interface de comunicação serial (via USB) para comunicação com os softwares desenvolvidos neste projeto;
- Um conversor analógico/digital de 16 bits para capturar o valor do ângulo do braço giratório;
- Pinos de entrada e saída digitais (GPIO) para comunicação com o circuito atuador.

É de grande valia, também, pontuar a utilização da plataforma MBED, a qual, dotada de um ambiente de desenvolvimento e compilador online, simplificou bastante o processo de programação da placa de desenvolvimento.

4.2.1 Projeto do Circuito de Atuação

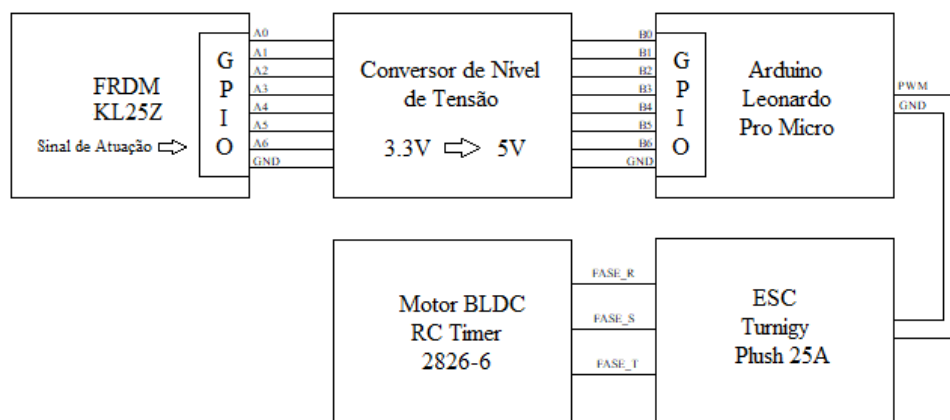
O circuito de atuação da planta possui quatro componentes:

- Um motor BLDC, de modelo 2826-6, da fabricante RC Timer, acoplado a uma hélice plástica de aproximadamente 20cm de diâmetro;
- Um ESC (do inglês, *Electronic Speed Controller*), traduzido como controlador eletrônico de velocidade, de modelo Plush 25A, do fabricante Turnigy, responsável por, baseado em um sinal de entrada, controlar a velocidade do motor BLDC;

- Um microcontrolador Arduino Leonardo Pro Micro para modular o sinal a ser enviado para o ESC e servir como interface (via GPIO) para a FRDM-KL25Z;
- Um arranjo de conversores de nível de tensão unidirecionais para realizar a comunicação entre a GPIO da FRDM-KL25Z (3.3V) e do Arduino (5V).

O diagrama de blocos ilustrado na Figura 15 exhibe a interligação elétrica entre os componentes do circuito de atuação.

Figura 15 – Diagrama de blocos da interligação lógica do circuito de atuação



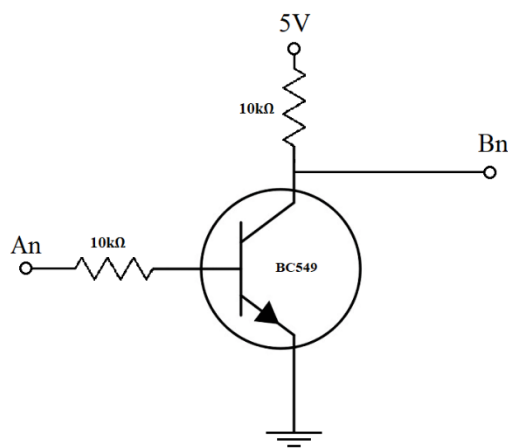
Fonte: Próprio Autor

De forma preliminar, foi verificado experimentalmente o intervalo de velocidade do motor que seria útil para este projeto, de forma a não superaquecer o motor ou colocar em perigo a integridade física do autor e do próprio projeto. Chegou-se à conclusão de que o intervalo de cerca de 20% a 30% de velocidade total disponível seria suficiente para a realização deste trabalho. Este ESC possibilita resolução de 0,1% de passo incremental na variação da velocidade em relação à sua velocidade máxima. São necessários, portanto, no mínimo, 100 passos incrementais a partir da velocidade mínima pretendida (20%) para que seja atingida a velocidade máxima pretendida (30%). Adotou-se, todavia, um intervalo de 127 passos incrementais, implementando-se a estratégia na qual a FRDM-KL25Z transmite por meio de sua interface GPIO (do inglês, General Purpose Input/Output, traduzido como Entrada/Saída de Propósito Geral) ao Arduino apenas o valor incremental a ser somado com o valor de velocidade mínima pré-estabelecida e configurada no Arduino. É necessário, portanto, a

utilização de no mínimo 7 bits de acionamento para transmitir o valor incremental de atuação, uma vez que $2^7 - 1 = 127$ (número de passos incrementais).

Devido à diferença de nível lógico de tensão entre a FRDM-KL25Z e o Arduino, foi necessário implementar entre eles 7 conversores de nível de tensão unidirecionais. Para cada conversor, foi escolhido o transistor NPN BC549 a ser utilizado como chave lógica. Com o intuito de saturá-lo, foram utilizados dois resistores de $10\text{k}\Omega$, polarizando-o na configuração emissor-comum. É mostrado, na Figura 16, o diagrama elétrico do conversor utilizado.

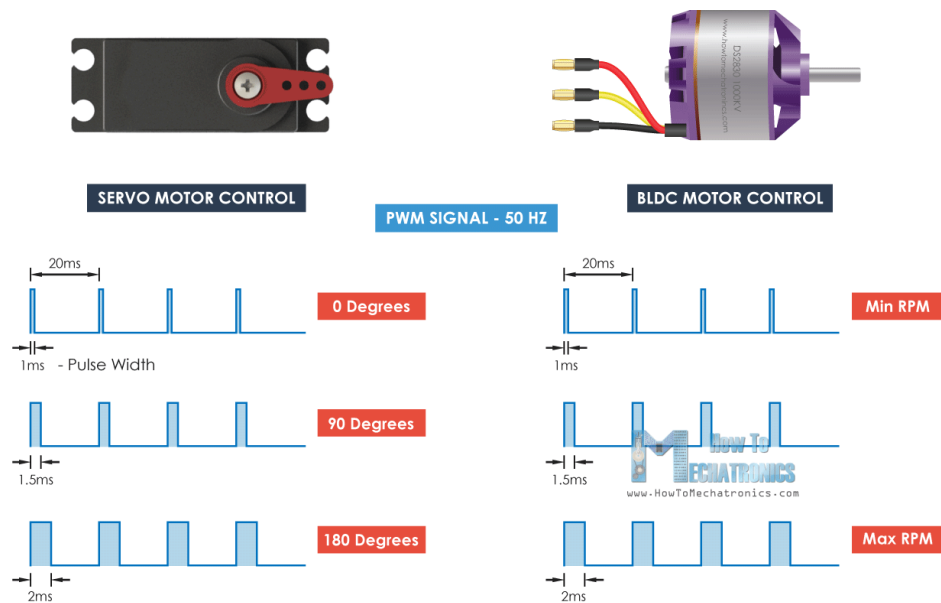
Figura 16 – Conversor de Nível de Tensão



Fonte: Próprio Autor

O Arduino tem como função principal receber os sinais digitais transmitidos pela FRDM-KL25Z, decodificá-los e gerar um sinal compreensível pelo ESC. Este tipo de ESC utiliza um protocolo muito comum em servo-motores: recebe um sinal PWM (do inglês, *Pulse Width Modulation*, traduzido como Modulação por Largura de Pulso), com período de 20ms e ciclo de trabalho entre 5% e 10% . Com 5% de ciclo de trabalho, o ESC imputa ao motor sua velocidade mínima (neste caso, zero). Com 10% , todavia, o ESC transfere ao motor a máxima velocidade de rotação possível. Este processo é ilustrado na Figura 17.

Figura 17 – Diagrama de blocos da interligação lógica do circuito de atuação

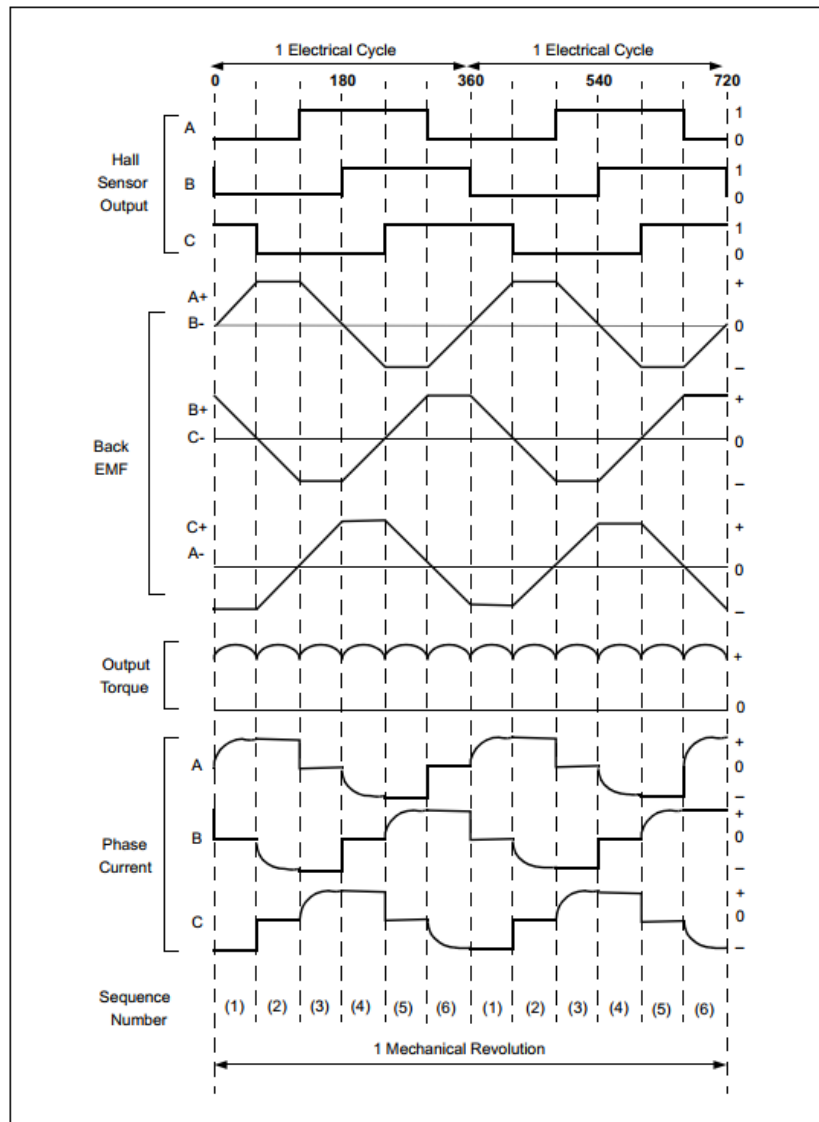


Fonte: DEJAN (2019)

É possível notar, entretanto, que este protocolo de comunicação limita a frequência de atuação do sistema a, no máximo, 50Hz, impactando diretamente na posterior escolha da frequência de amostragem.

O ESC atua sobre o motor BLDC, sintetizando três ondas com formas geralmente trapezoidais ou sinusoidais, defasadas de 120 graus, compondo um sistema trifásico. Na Figura 18, é possível visualizar as formas de onda relativas a um BLDC.

Figura 18 – Parâmetros Elétricos de um Motor BLDC

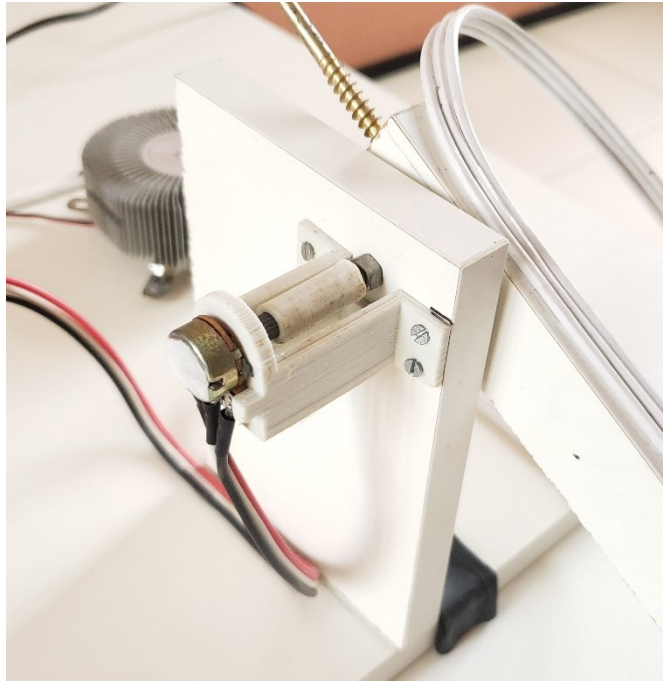


Fonte: Microchip Technology Inc (2003)

4.2.2 Projeto do Sensor de Ângulo

Para que o microcontrolador adotado neste projeto fosse capaz de adquirir um valor representativo do ângulo do braço giratório, foi projetado um sensor baseado em um potenciômetro linear de valor nominal 50k Ω , cujo resultado final é exibido na Figura 19.

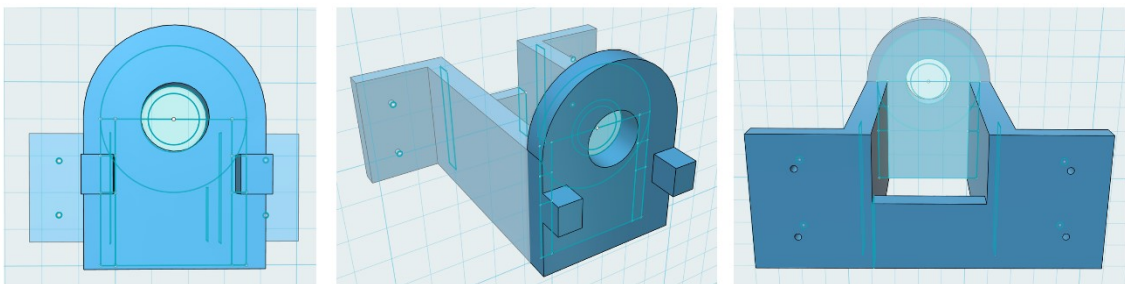
Figura 19 – Sensor de ângulo



Fonte: Próprio Autor

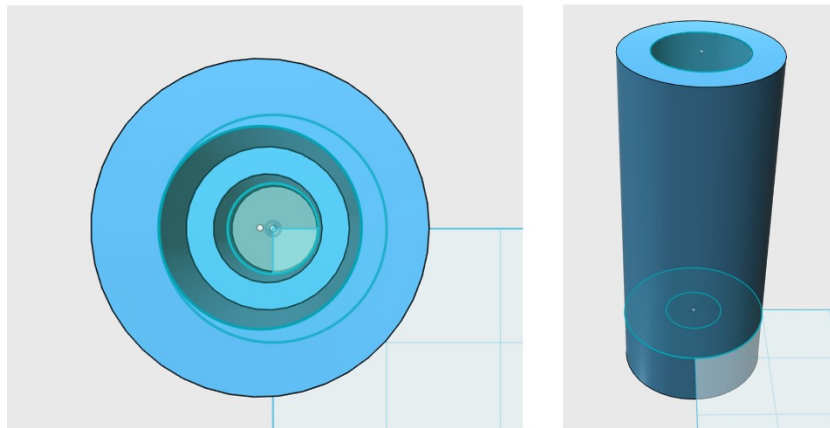
O potenciômetro foi acoplado ao eixo da planta por meio de duas peças plásticas impressas em uma impressora 3D. Os projetos em 3D das peças de acoplamento foram realizados utilizando o software 123D Design, da Autodesk, cujas modelagens são mostradas na Figura 20 e Figura 21.

Figura 20 – Fixador do Potenciômetro



Fonte: Próprio Autor

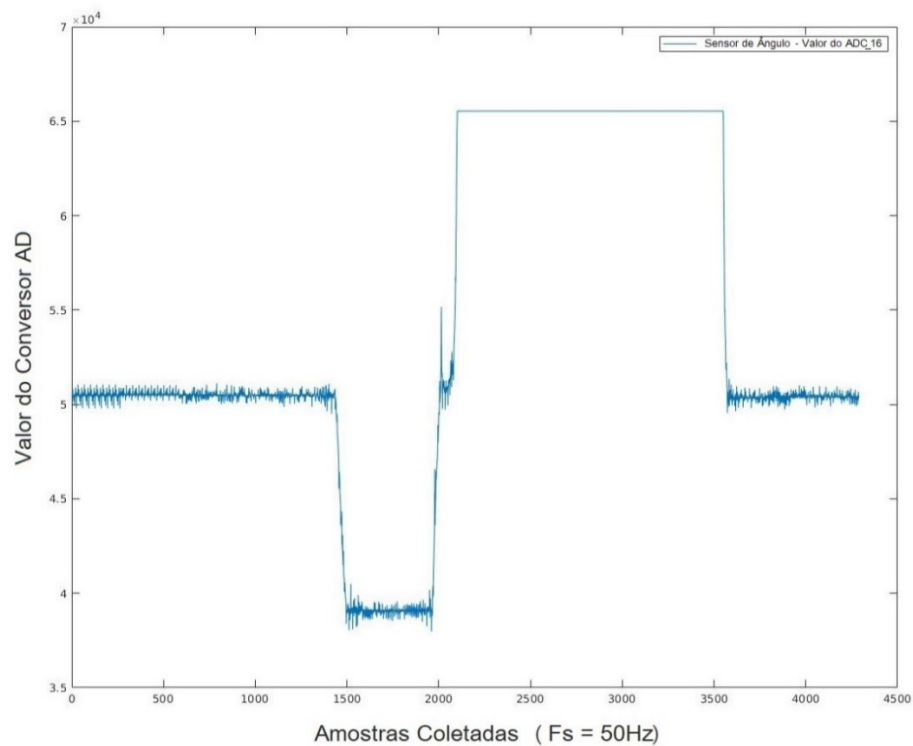
Figura 21 – Acoplador de Eixos



Fonte: Próprio Autor

Após análises preliminares do sinal do sensor adquirido pelo microcontrolador e enviado a um software de captura desenvolvido neste trabalho com o MATLAB, foi detectada a necessidade de condicionamento do sinal recebido devido ao grande ruído existente. O resultado do ensaio preliminar é exibido através do Gráfico 1.

Gráfico 1 – Primeiro Ensaio Preliminar



Fonte: Próprio Autor

Como tentativa de atenuação do ruído, duas soluções foram adotadas: o projeto e a implementação de um filtro *anti-aliasing* na saída do sensor juntamente com a utilização de um CI gerador de tensão de referência como alimentação do sensor.

Conforme pontuado na subseção anterior, a frequência máxima de amostragem do sistema está limitada a 50Hz. Portanto esta foi a frequência de amostragem utilizada *a priori* para balizar o projeto e implementação do filtro anti-aliasing.

De forma conservadora e prevendo um cascadeamento composto por um filtro analógico seguido por um filtro digital, foi estipulado, inicialmente, um filtro analógico RC passa-baixas com frequência de corte de 100Hz. Substituindo-se $\omega_c = 2\pi f_c$ na equação (12), tem-se:

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C} \quad (94)$$

Sendo:

f_c – A frequência de corte do filtro, em Hz;

R – A valor do resistor em série;

C – O valor do capacitor em paralelo.

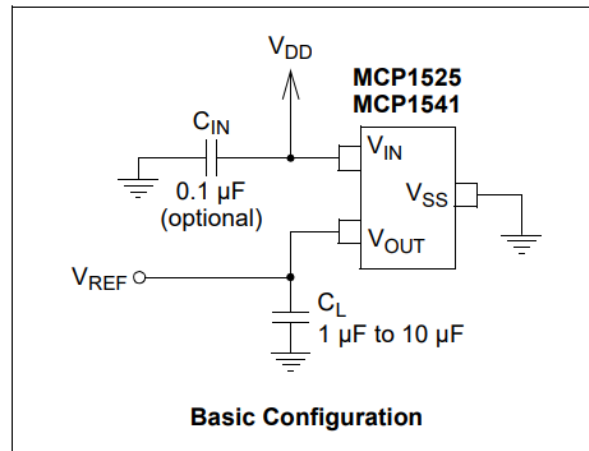
Para um valor de capacitância de 1 μ F e frequência de corte desejada de 100Hz, obtém-se, substituindo-se estes valores na equação (94), a equação (95):

$$R = \frac{1}{2 \cdot \pi \cdot 1 \cdot 10^{-6} \cdot 100} = 1,59k\Omega \quad (95)$$

Utilizou-se, portanto, um resistor de 1,5k Ω e um capacitor eletrolítico de 1 μ F.

Foi utilizado também o CI gerador de tensão de referência MCP1541, que assegura uma tensão fixa de 4,096V na alimentação do sensor. Segundo a folha de dados do MCP1541, seu circuito típico de aplicação tem a estrutura mostrada na Figura 22.

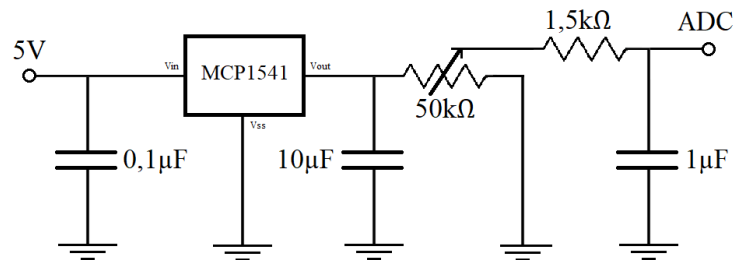
Figura 22 – MCP1541: Aplicação Típica



Fonte: Microchip Technology Inc. (2012)

Logo, a estrutura completa do sensor, incluindo o circuito de condicionamento de sinal é mostrada na Figura 23.

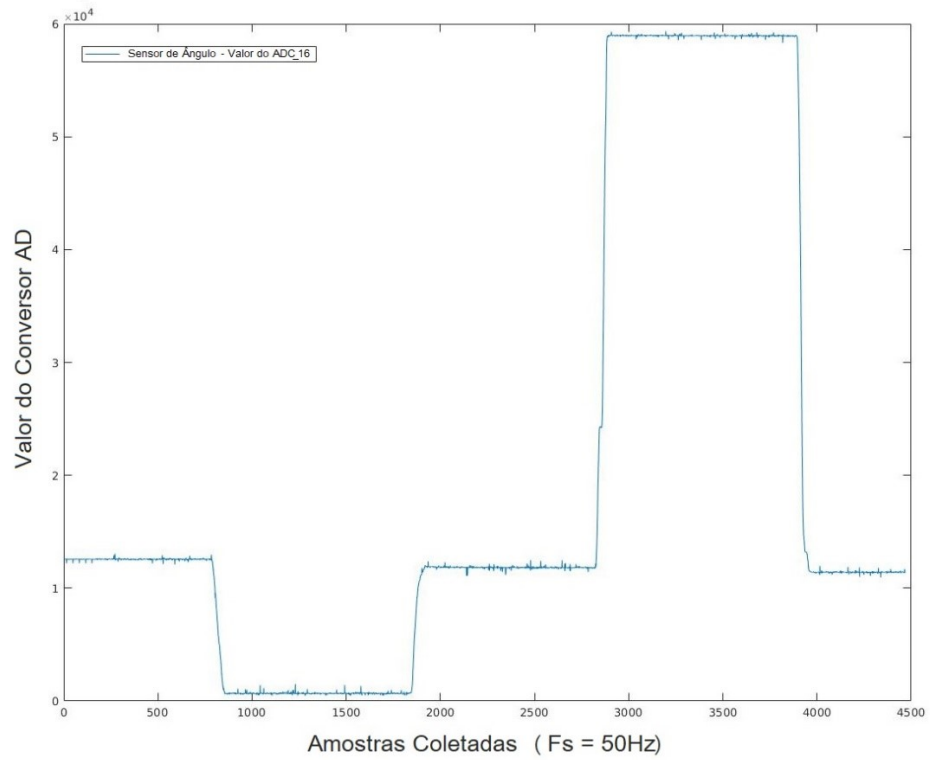
Figura 23 – Estrutura Completa do Sensor de Ângulo



Fonte: Próprio Autor

Uma segunda análise preliminar foi realizada, com o intuito de verificar a eficácia da solução projetada. Notou-se sensível melhora na qualidade do sinal, com evidente diminuição do ruído. É exibido, no Gráfico 2, o resultado do segundo ensaio preliminar.

Gráfico 2 – Segundo Ensaio Preliminar



Fonte: Próprio Autor

5 IDENTIFICAÇÃO DA PLANTA

Embora seja possível desenvolver expressões analíticas que relacionem os valores de entrada e saída para todos os componentes presentes no sistema, linearizando-as quando necessário, optou-se, neste trabalho, por identificar o sistema em questão representando-o por um modelo caixa-preta em razão da complexidade do processo real. Para tal, ensaios práticos são necessários de forma a levantar-se as variáveis necessárias para a caracterização da planta.

Como sinal de excitação do sistema, foi projetado e implementado um sinal de sequência binária pseudoaleatória de modo a prover uma gama definida de frequências como entrada do sistema e assim melhor caracterizá-lo.

5.1 Projeto e Inclusão de um Filtro Digital

Para complementar o filtro analógico descrito na subseção 4.2.2 e assim formar o filtro *anti-aliasing* necessário para este trabalho, projetou-se um filtro digital de primeira ordem com frequência de corte de 20Hz, assegurando a atenuação das frequências superiores à metade da frequência de amostragem inicialmente utilizada (50Hz), conforme teorema de Nyquist, dissertado na subseção 2.3.2.

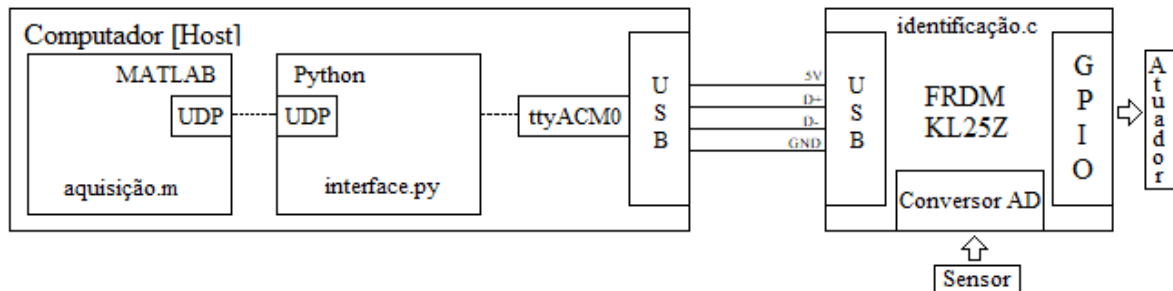
Sendo $\omega_c = 2\pi 20 = 125,66 \text{ rad/s}$, a frequência de corte do filtro digital, obtém-se, substituindo ω_c na equação (22), o filtro digital $H(z)$, segundo a equação (96):

$$H(z) = \frac{1 - e^{-\omega_c T_s}}{z - e^{-\omega_c T_s}} = \frac{0,9189}{z - 0,0810} \quad (96)$$

5.2 Metodologia Prática

Para realizar os experimentos de identificação, foram utilizados, além da FRDM-KL25Z como dispositivo de interface entre a planta e o computador, um arranjo de programas e *scripts* conforme exibidos na Figura 24.

Figura 24 – Diagrama de Comunicação para Identificação da Planta



Fonte: Próprio Autor

Foi instalado na FRDM-KL25Z um programa para identificação (disponível no ANEXO A), escrito na linguagem C com o auxílio da plataforma MBED, foi responsável por, além do processamento quanto à leitura do sensor e comandos de atuação programados, transmitir os dados para o computador utilizado na identificação do sistema a cada período de amostragem.

Para que o computador adquira os dados provenientes da FRDM-KL25Z, foi escrito um *script* em Python, chamado *interface.py* (disponível no ANEXO B), com o objetivo de prover acesso à interface serial de escrita e leitura de dados, via USB (*Universal Serial Bus*). De posse dos dados transmitidos pela FRDM-KL25Z, o *script* *interface.py* foi responsável por encaminhar as informações recebidas para o *script* *aquisição.m*, escrito e executado no MATLAB, por meio do protocolo UDP. O *script* *aquisição.m* (disponível no ANEXO C), por sua vez, foi responsável por, além de guardar os dados recebidos na memória do computador visando à análise posterior, plotar um gráfico instantâneo representativo dos dados recebidos.

5.3 Ensaio de Resposta ao Degrau

Inicialmente, visando à identificação do sistema em questão, foi realizado um ensaio de resposta ao degrau.

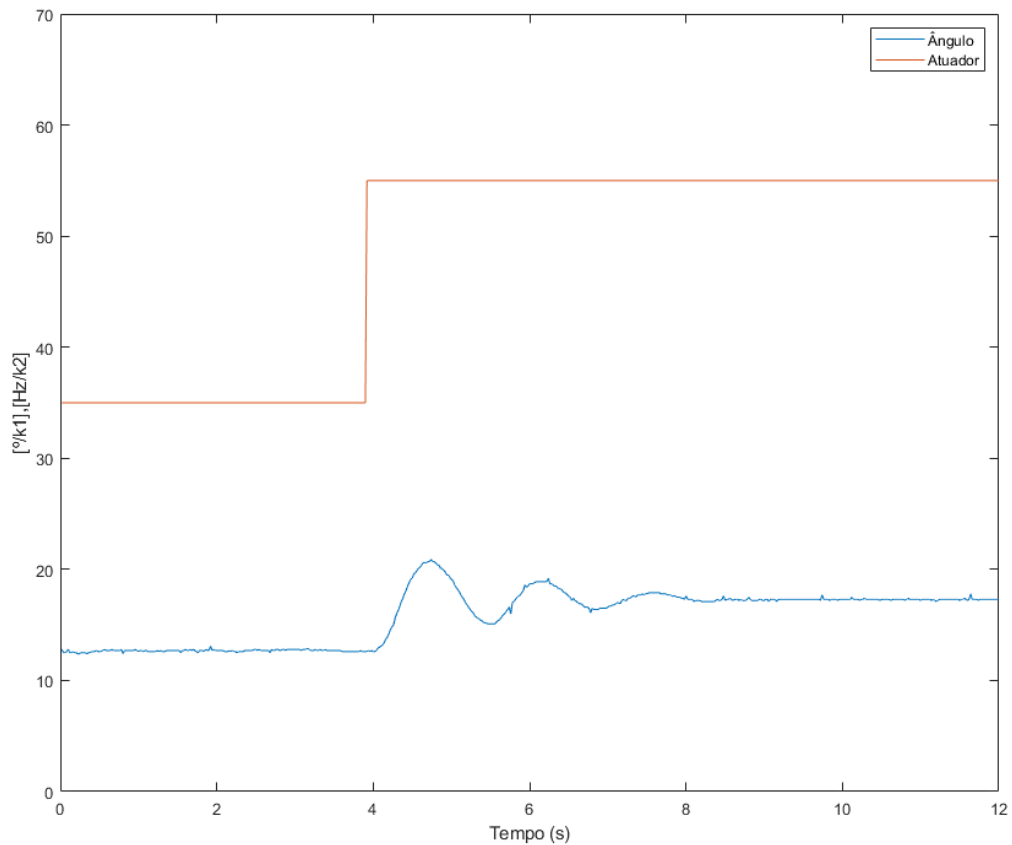
Devido à grande diferença de escala entre os valores lidos pelo sensor (0 a 65535) e os valores possíveis para atuação (0 a 127), foi necessária a adequação de escalas para melhor visualização dos resultados. Este pré-processamento foi realizado na própria FRDM-KL25Z e foi escolhido um intervalo de variação máxima de 0 a 100 para ambos, conforme demonstrado nas equações (97) e (98).

$$\text{Valor de atuação convertido} = \frac{100 \cdot \text{Valor_Atuado}_{[de\ 0\ a\ 127]}}{127} \quad (97)$$

$$\text{Valor do ângulo convertido} = \frac{100 \cdot \text{Valor_Lido_Pelo_Sensor}_{[de\ 0\ a\ 65535]}}{65535} \quad (98)$$

O resultado do ensaio de resposta ao degrau é exibido no Gráfico 3. Como não houve calibração da frequência angular do motor e da abertura angular do braço giratório com relação às unidades usuais utilizadas, assume-se k_1 e k_2 como constantes de conversão a serem determinadas posteriormente, sem prejuízo para análise pretendida neste trabalho.

Gráfico 3 – Ensaio de Resposta ao Degrau



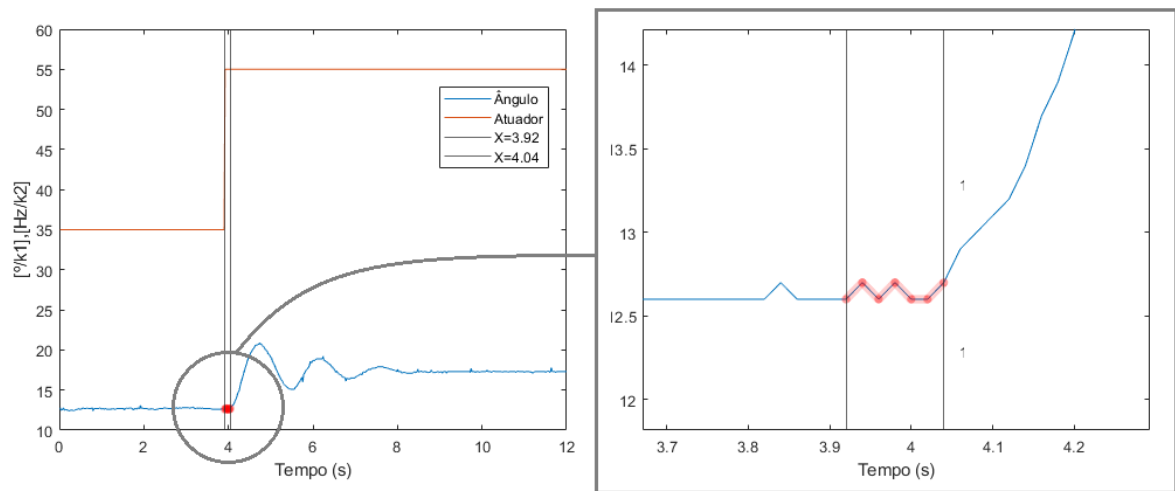
Fonte: Próprio Autor

É possível perceber, de antemão, o comportamento oscilatório sub-criticamente amortecido da resposta transitória do sistema em malha aberta e do presente erro de regime permanente caracterizado pela diferença entre o valor de referência (valor de atuação) e o valor em regime permanente do sistema. Nota-se que, neste ponto de operação, dados os valores do degrau de entrada e das perturbações sofridas naturalmente, é possível afirmar que o sistema é

estável. Estas perturbações (que se dão como pequenas oscilações percebidas em regime permanente) já eram esperadas devido à natureza aerodinâmica do sistema.

Foi detectado um atraso de 6 amostras (totalizando 0.12s) na resposta do sistema quando houve a mudança no sinal de referência. Este efeito pode ser melhor visualizado no Gráfico 4.

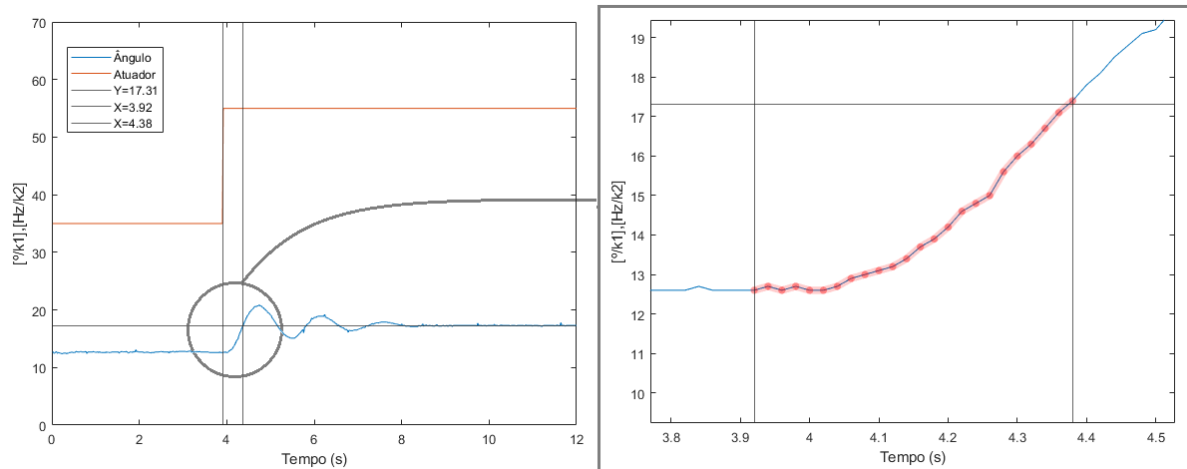
Gráfico 4 – Atraso de Resposta (Com Ampliação)



Fonte: Próprio Autor

O tempo de subida neste projeto foi considerado como o tempo necessário para o sistema atingir 100% do valor de regime permanente. Logo, como é possível visualizar no Gráfico 5, o tempo de subida foi de 0.12s (23 amostras).

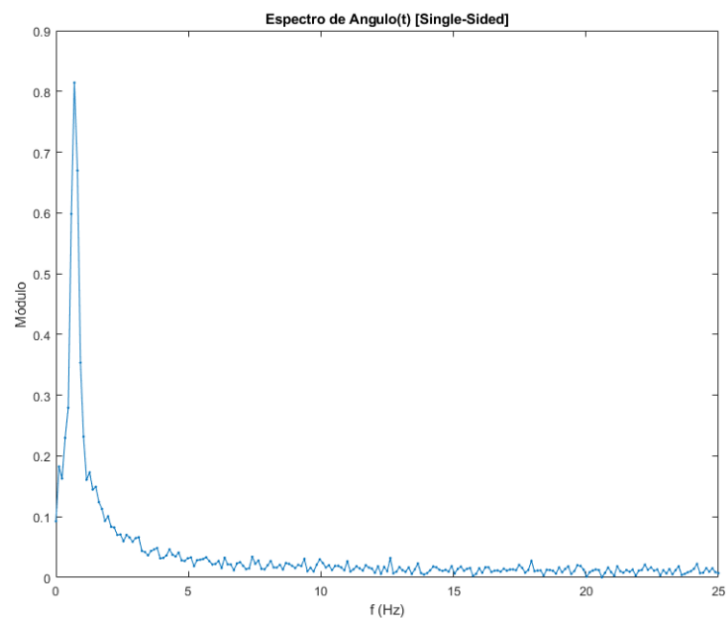
Gráfico 5 – Tempo de Subida (Com Ampliação)



Fonte: Próprio Autor

O módulo da resposta em frequência do sistema foi obtido mediante a transformada rápida de Fourier, utilizando-se o MATLAB, e é exibido no Gráfico 6.

Gráfico 6 – Módulo da Resposta em Frequência



Fonte: Próprio Autor

Conforme exposto na subseção 2.3.2, estima-se (com grande margem para mais), de acordo com o Gráfico 6, uma largura de banda de malha aberta de 3Hz. Utilizando-se a

equação (5) como referência, percebe-se que o sistema com frequência de amostragem de 50Hz tende a estar sobreamostrado. Isto pode ser positivo, caso queira-se implementar técnicas de *oversampling* (LANDAU; ZITO, 2006, p.282). Porém, por uma questão de simplicidade, optou-se por utilizar a frequência de amostragem igual à frequência do *loop* de controle mesmo com a possibilidade de sobreamostragem.

5.4 Projeto da Sequência Binária Pseudoaleatória

Para o sinal de excitação, composto por uma sequência binária pseudoaleatória, optou-se pelas seguintes especificações iniciais:

- f_1 (valor da menor componente de frequência do sinal) igual a 0,05Hz;
- f_2 (valor da maior componente de frequência do sinal) igual a 3Hz.

Logo, substituindo as especificações desejadas nas equações (23), (24) e (25), obtém-se:

$$3 \leq \frac{0,44}{T_b} \quad (99)$$

$$T_b \leq 0,146 \quad (100)$$

$$0,05 \geq \frac{1}{(2^N - 1) \cdot T_b} \quad (101)$$

$$N \geq 7,11 \quad (102)$$

Sendo:

T_b – O período de transição de células do sinal SBPA;

N – O número de células da sequência do sinal SBPA;

Repetindo-se os cálculos realizado acima, escolhendo-se $N = 7$ e $T_b = 0,14$ obtém-se uma faixa de frequências de excitação do sistema entre 0,056 Hz e 3,14 Hz, adotada para a continuidade deste trabalho.

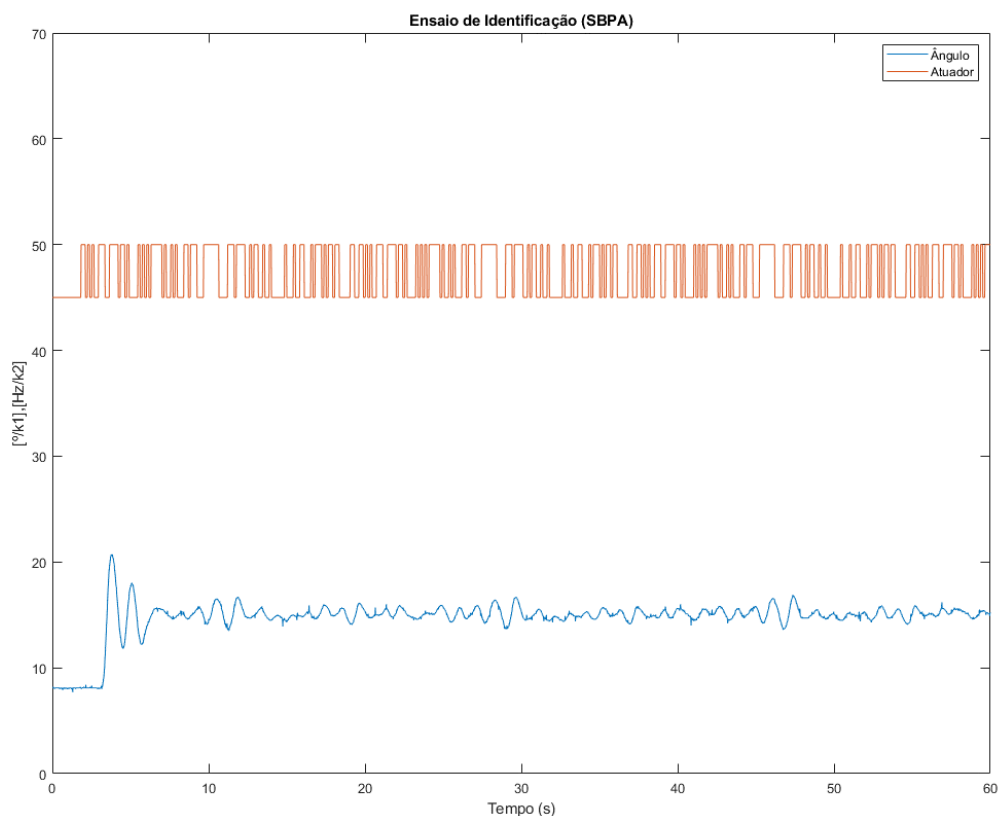
Uma vez que $N \cdot T_b = 0,98s$ é maior que $0,12s$ (tempo de subida do sistema), será possível adquirir informações sobre o valor de regime permanente do sistema.

5.5 Identificação

5.5.1 Ensaio de Identificação

Utilizando como entrada do sistema o sinal SBPA projetado na subseção 5.4, foi realizado o ensaio de identificação do sistema. Parte dos valores de ângulo e atuação coletados são exibidos no Gráfico 7.

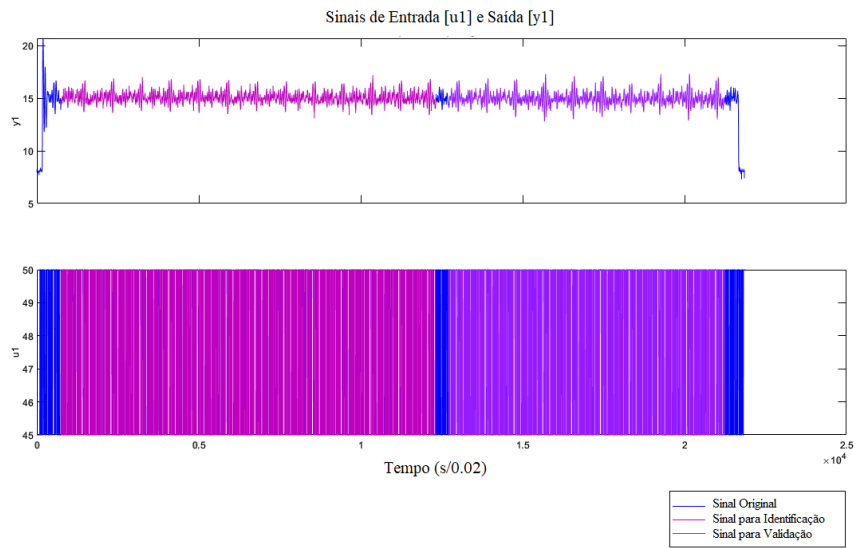
Gráfico 7 – Ensaio de Identificação (0-60s)



Fonte: Próprio Autor

De posse da resposta do sistema, esta foi dividida em dois conjuntos de dados com o objetivo de utilizar o primeiro para a identificação do sistema e o segundo para a validação do sistema identificado. A divisão dos dados é exibida no Gráfico 8.

Gráfico 8 – Divisão dos dados Adquiridos

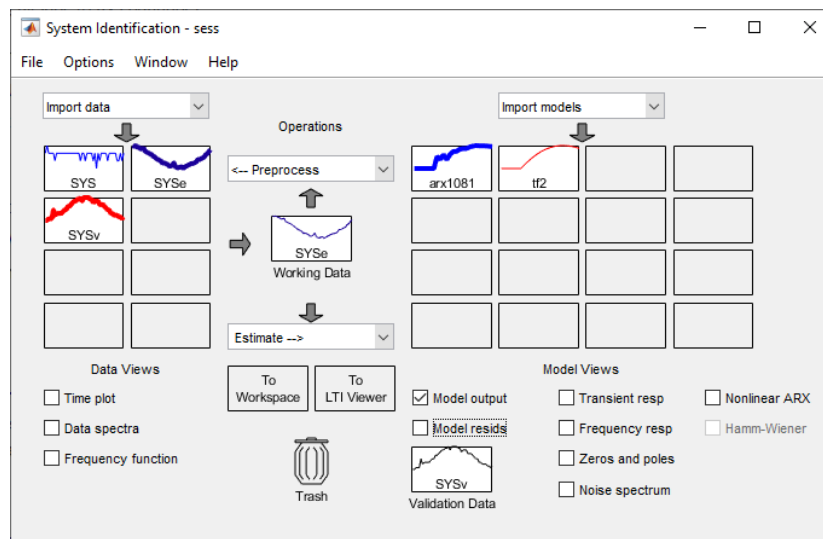


Fonte: Próprio Autor

5.5.2 Uso do toolbox System Identification do MATLAB

O MATLAB, é uma plataforma científica que, dentre outras funções, traz um *toolbox* dedicado à identificação de sistemas, chamado System Identification. Sua interface principal é mostrada na Figura 25, a seguir.

Figura 25 – Toolbox System Identification do MATLAB (versão R2018B)



Fonte: Próprio Autor

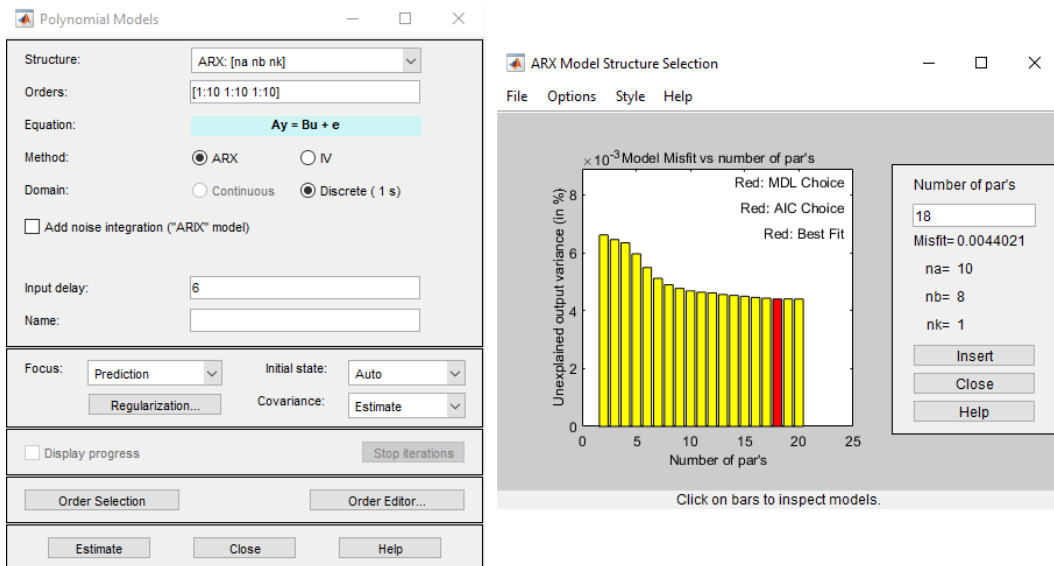
Esta ferramenta apresenta vários algoritmos e métodos úteis de identificação e análise de sistemas dinâmicos, sendo bastante utilizada neste trabalho.

5.5.3 Identificação utilizando Mínimos Quadrados Não Recursivo

Procedeu-se com a identificação do sistema utilizando-se, primeiramente, o método dos Mínimos Quadrados Não Recursivo, descrito na subsecção 2.7.2 deste trabalho.

Para definir a complexidade do modelo ARX a ser identificado, utilizou-se a ferramenta de seleção de ordem, relativa à estimação de modelos polinomiais do *toolbox* System Identification, do MATLAB, cuja saída é mostrada pela Figura 26, a seguir.

Figura 26 – Estimação da Complexidade do Modelo



Fonte: Próprio Autor

A forma geral do modelo ARX indicado é, portanto, representada pela equação (103).

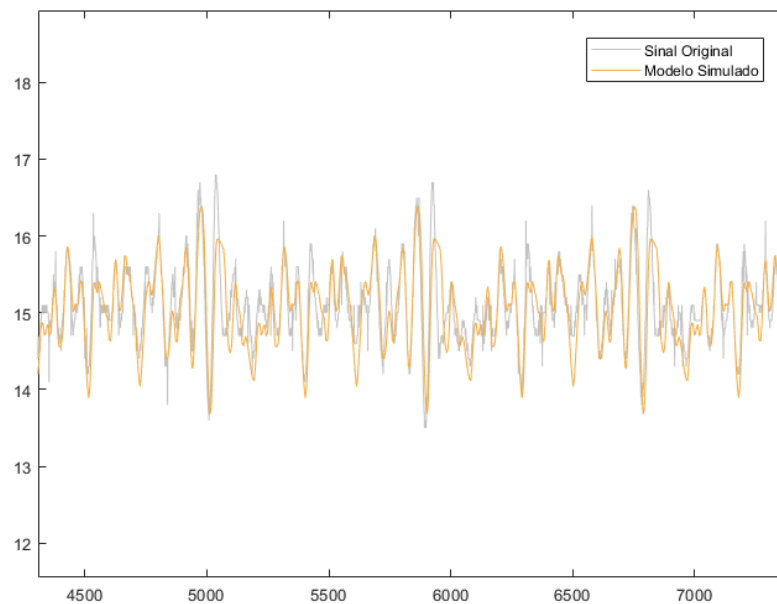
$$G(z^{-1}) = z^{-6} \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_8 \cdot z^{-8}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_9 \cdot z^{-9} + a_{10} \cdot z^{-10}} \quad (103)$$

Escreveu-se um *script* utilizando o MATLAB, para a identificação via Mínimos Quadrados Não Recursivo (disponível no ANEXO D), materializando-se a teoria descrita na subsecção 2.7.2. Este *script* encontra-se no anexo C deste trabalho.

A função de transferência do modelo resultante é exibida pela equação (104), com sua simulação ilustrada pelo Gráfico 9.

$$G(z^{-1}) = z^{-6} (0,01214 \cdot z^{-1} - 0,0005711 \cdot z^{-2} + 0,001859 \cdot z^{-3} + 0,0006854 \cdot z^{-4} + 0,0007773 \cdot z^{-5} + 0,002067 \cdot z^{-6} - 0,0002968 \cdot z^{-7} + 0,007799 \cdot z^{-8}) / (1 - 0,4415 \cdot z^{-1} + 0,2834 \cdot z^{-2} - 0,2059 \cdot z^{-3} - 0,1712 \cdot z^{-4} - 0,08112 \cdot z^{-5} - 0,05179 \cdot z^{-6} + 0,03618 \cdot z^{-7} + 0,06876 \cdot z^{-8} + 0,07416 \cdot z^{-9} + 0,133 \cdot z^{-10}) \quad (104)$$

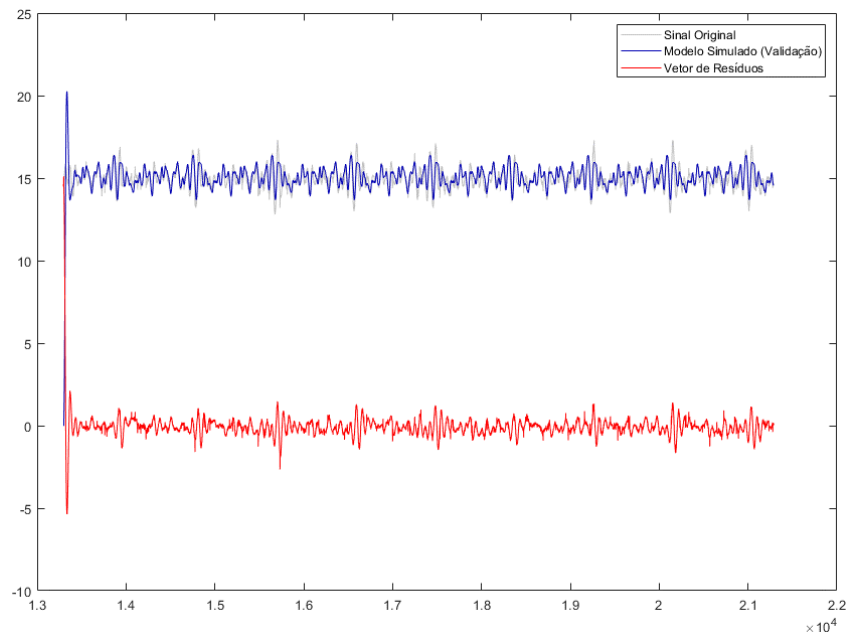
Gráfico 9 – Simulação do Modelo Identificado



Fonte: Próprio Autor

Utilizando-se o conjunto de dados de validação com a finalidade de validar o modelo, são obtidos os gráficos resultantes da simulação de validação, com inclusão do gráfico de resíduos, ilustrados pelo Gráfico 10.

Gráfico 10 – Simulação de Validação do Modelo Identificado



Fonte: Próprio Autor

A partir da simulação de validação do modelo, são calculados os índices indicados na subseção 2.8 deste trabalho e exibidos na Tabela 2, a seguir.

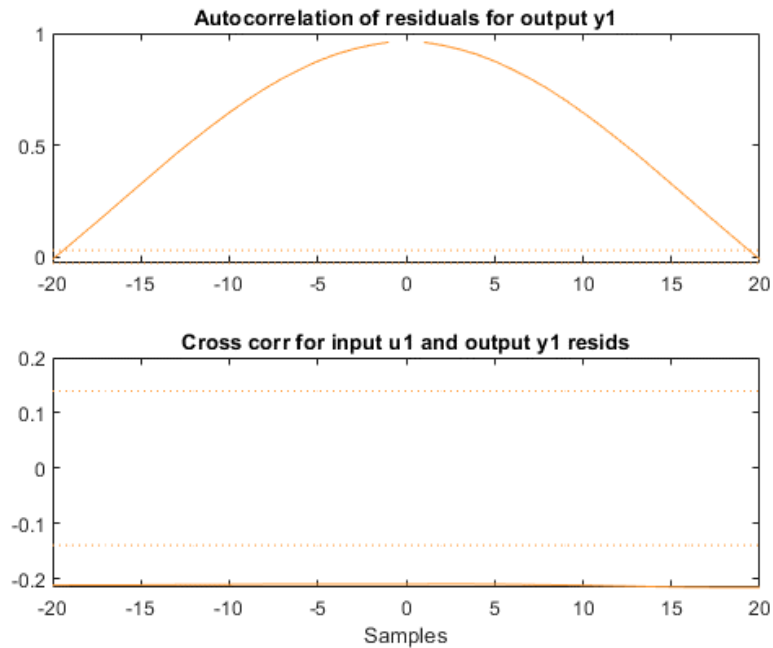
Tabela 2 – Índices de Validação do Modelo 1

Índice	Valor
Somatório do erro quadrático (SEQ)	1251,31
Desvio absoluto percentual médio (DAPM)	2,11%
Coefficiente de correlação múltipla (R^2)	0,55

Fonte: Próprio Autor.

A análises de autocorrelação e correlação cruzada dos resíduos do modelo, descritas na subseção 2.8 deste trabalho, foram calculadas utilizando o *toolbox* System Identification do MATLAB e são exibidas no Gráfico 11, a seguir.

Gráfico 11 – Análise de Resíduos do Modelo Identificado



Fonte: Próprio Autor

Com 7226 amostras consideradas para esta validação, tem-se um limite para os valores de autocorrelação e correlação cruzada, aplicando-se a relação contida na equação (50), utilizando-se um nível de confiança de 3%, tal que:

$$|RN(i)| \leq \frac{2,17}{\sqrt{7226}} = 0,026 \quad (105)$$

Analisando-se a validade do modelo, constata-se que há uma certa capacidade do modelo em caracterizar a dinâmica do sistema, embora, deficientemente. É possível verificar graficamente que ainda há bastante informação contida nos resíduos. Tal fato é corroborado pelas análises de correlação e autocorrelação dos resíduos, ambas acima do limite sugerido pela teoria.

5.5.4 Identificação utilizando o Método de Gauss-Newton

Um novo processo de identificação do sistema foi realizado utilizando-se, para tal, o Método de Gauss-Newton, descrito na subseção 2.7.3 deste trabalho.

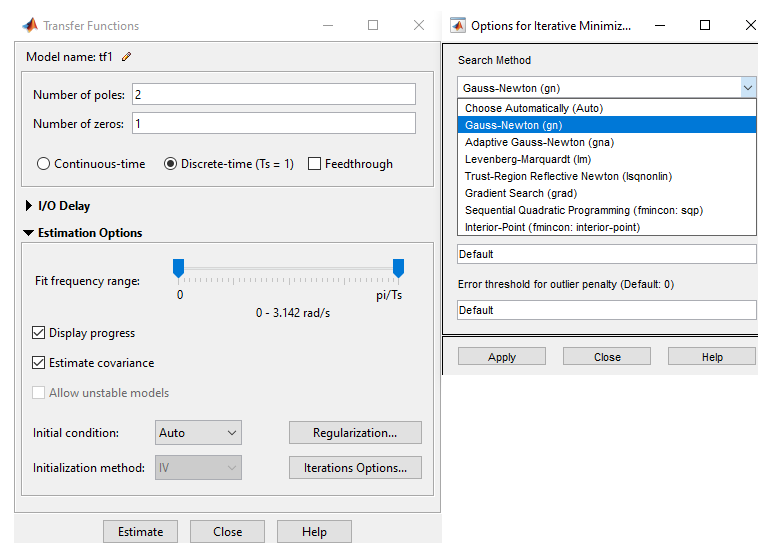
Como seleção inicial da complexidade do sistema, foi escolhido um modelo de sistema com polinômio B de primeiro grau e polinômio A de segundo grau, bem como um

atraso de resposta já conhecido de 6 amostras. A forma geral do modelo ARX escolhido é representada pela equação (58).

$$G(z^{-1}) = z^{-6} \frac{b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \quad (106)$$

Não foi necessário escrever um algoritmo para a utilização deste método, uma vez que o *toolbox* System Identification do MATLAB já contém esta funcionalidade, conforme demonstrado através da Figura 27.

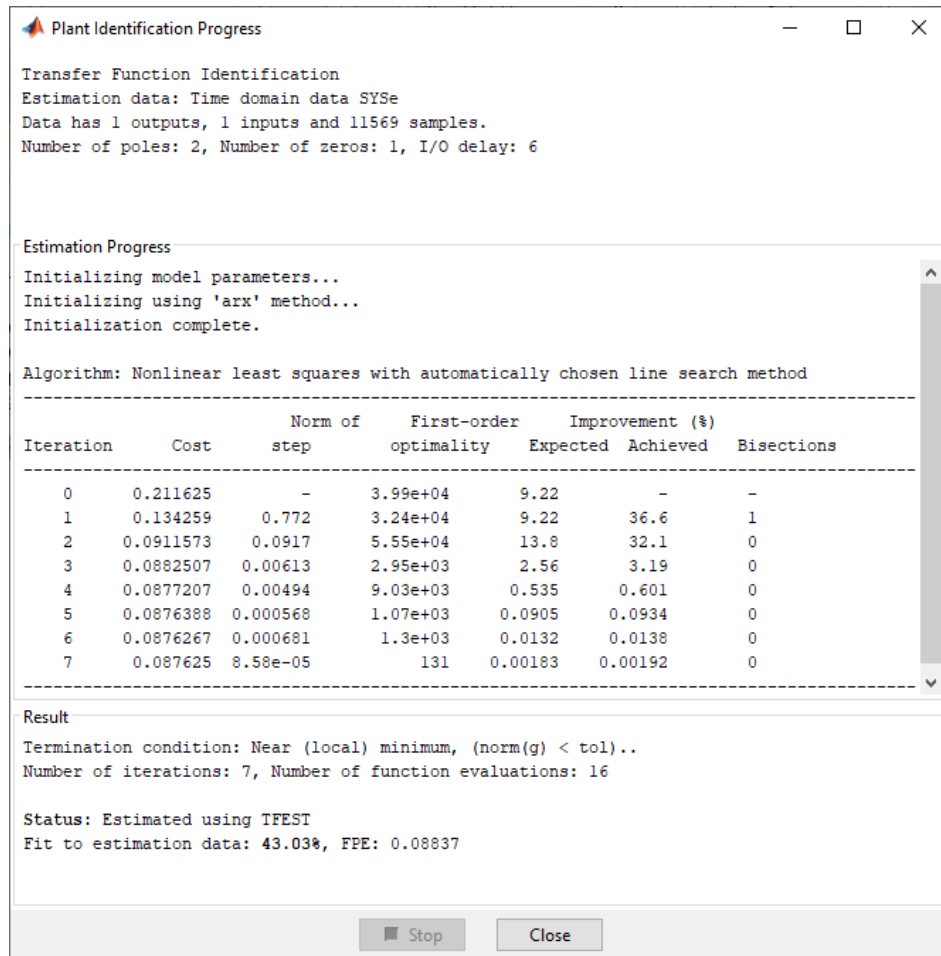
Figura 27 – *Toolbox* System Identification (Gauss-Newton)



Fonte: Próprio Autor

O relatório produzido durante o processo de estimação do modelo é exibido na Figura 28.

Figura 28 – Processo de Estimação do Modelo ARX



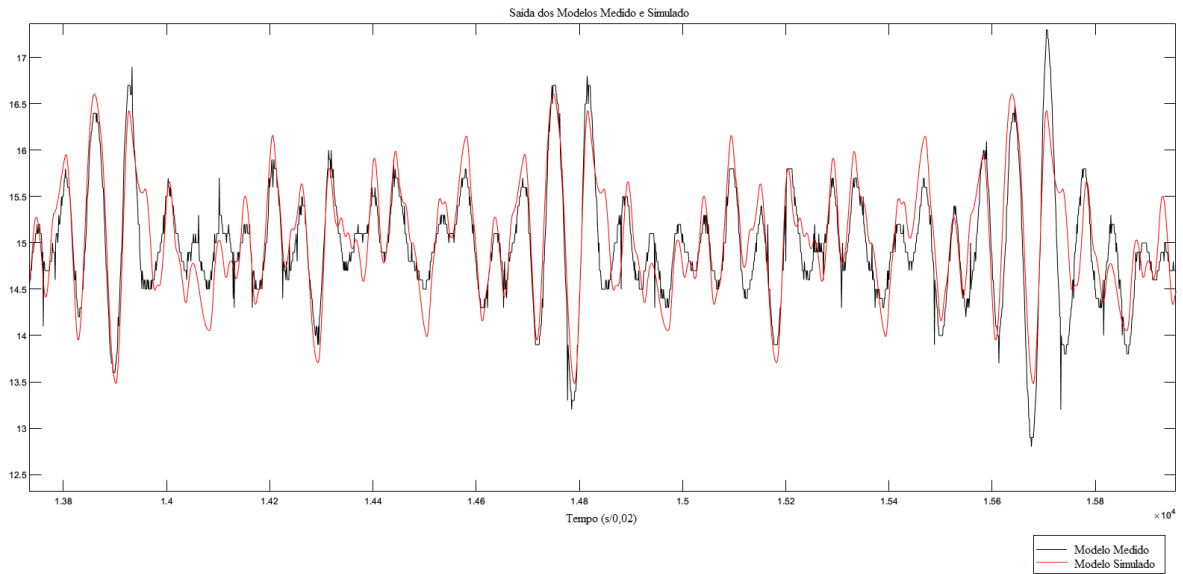
Fonte: Próprio Autor

O modelo resultante, exibido na equação (107), teve, segundo o relatório de estimação, 43,03% de adequação ao conjunto de dados de validação.

$$G(z^{-1}) = z^{-6} \frac{0,00321 \cdot z^{-1}}{1 - 1,9492 \cdot z^{-1} + 0,9586 \cdot z^{-2}} \quad (107)$$

É ilustrada pelo Gráfico 12, a saída do sistema simulado em comparação com a saída medida do sistema no ensaio de identificação.

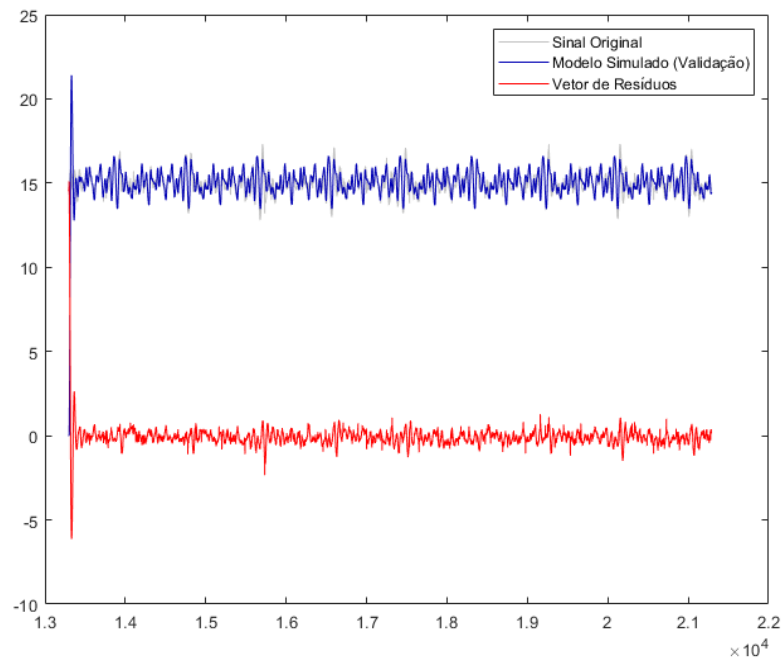
Gráfico 12 – Saída dos Modelos Medido e Simulado (Ampliada)



Fonte: Próprio Autor

Utilizando-se o conjunto de dados de validação com a finalidade de validar o modelo, são obtidos os gráficos resultantes da simulação de validação, com inclusão do gráfico de resíduos, ilustrados pelo Gráfico 13.

Gráfico 13 – Simulação de Validação do Modelo Identificado



Fonte: Próprio Autor

A partir da simulação de validação do modelo, são calculados os índices indicados na subseção 2.8 deste trabalho e exibidos na Tabela 3, a seguir.

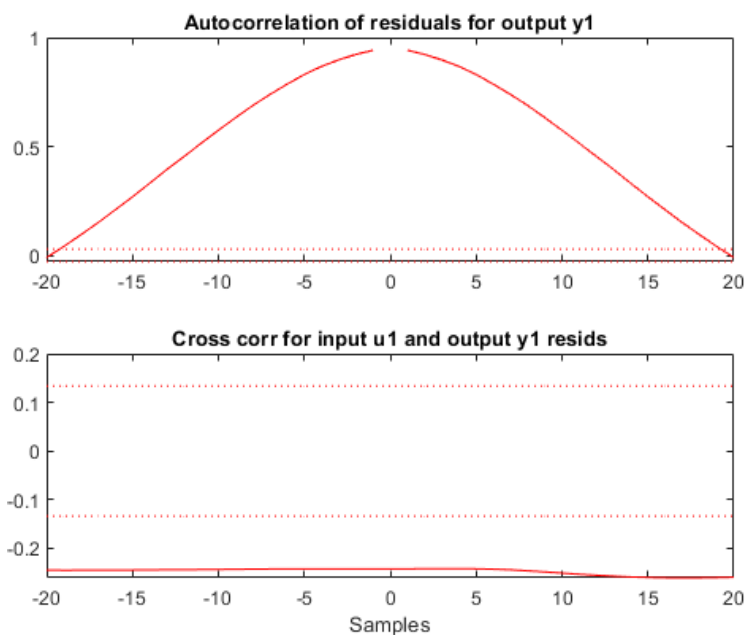
Tabela 3 – Índices de Validação do Modelo 2

Índice	Valor
Somatório do erro quadrático (SEQ)	896,29
Desvio absoluto percentual médio (DAPM)	1,85%
Coefficiente de correlação múltipla (R^2)	0,67

Fonte: Próprio Autor.

A análises de autocorrelação e correlação cruzada dos resíduos do modelo, descritas na subseção 2.8 deste trabalho, foram calculadas utilizando o *toolbox* System Identification do MATLAB e são exibidas no Gráfico 14, a seguir.

Gráfico 14 – Análise dos Resíduos



Fonte: Próprio Autor

Assim como no processo anterior, com 7226 amostras consideradas para esta validação, tem-se os mesmos limites para os valores de autocorrelação e correlação cruzada do modelo anterior, aplicando-se a relação contida na equação (50), utilizando-se um nível de confiança de 3%, tal que:

$$|RN(i)| \leq \frac{2,17}{\sqrt{7226}} = 0,026 \quad (108)$$

Analisando-se a validade deste modelo, por sua vez, constata-se que há uma melhora em seus índices de validação, quando comparado com o modelo identificado utilizando-se Mínimos Quadrados Não Recursivo. É possível verificar graficamente, também, que há uma capacidade melhor de representar a dinâmica transitória do sistema em comparação ao primeiro modelo, mesmo que ainda haja certa informação contida nos resíduos. Embora este modelo seja melhor que o primeiro, as análises de correlação e autocorrelação dos resíduos continuam acima do limite sugerido pela teoria.

Mesmo que este modelo não seja uma representação com alto nível de exatidão em relação ao sistema físico original, este foi escolhido para representá-lo devido à sua menor ordem, logo, maior simplicidade, e melhores índices de validação.

6 PROJETO E IMPLEMENTAÇÃO DO CONTROLADOR

Neste trabalho, foi projetado e implementado um controlador digital com a estrutura RST, conforme descrito na subseção 3.1 deste trabalho. Este controlador será ajustado utilizando como estratégia a alocação de polos.

6.1 Definição dos Índices de Desempenho Iniciais

Os índices de desempenho de referência para o sistema controlado consistiram em:

- Tempo de estabelecimento (com 2% de margem) menor ou igual a 3 segundos (sem considerar o tempo morto do sistema);
- Ultrapassagem percentual da resposta transitória menor ou igual a 5%;
- Erro de regime permanente nulo.

Para adequar o sistema controlado às especificações de projeto, utilizou-se como base um sistema contínuo de segunda ordem com atraso, derivado da equação (58), definido pela equação (109), a seguir.

$$SYS(s) = e^{-t.s} \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (109)$$

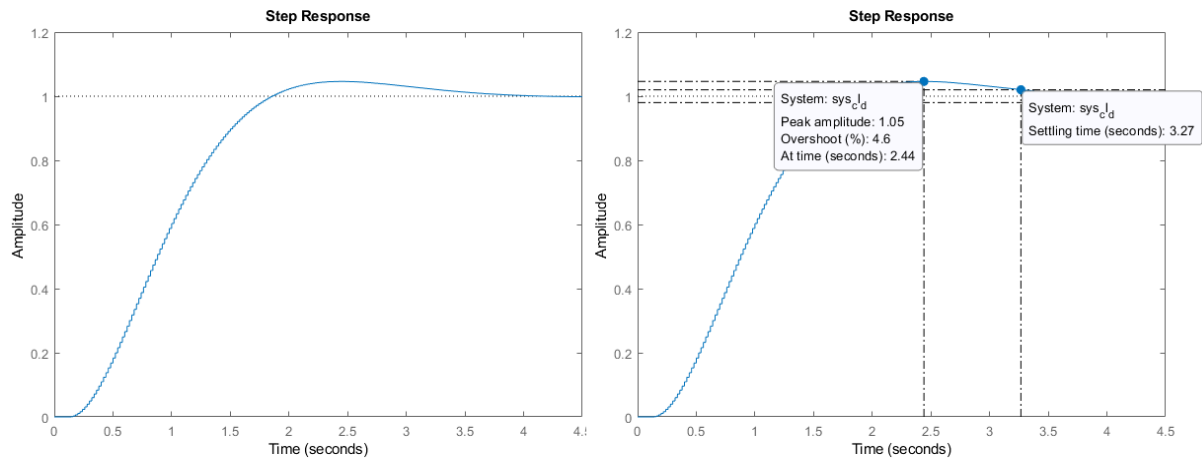
As especificações de desempenho de ultrapassagem percentual e tempo de estabelecimento, definidas pelas equações (59) e (60), são calculadas por substituição direta das especificações acima descritas, resultando, após devidos arredondamentos, nas equações (110) e (111), respectivamente.

$$\xi = 0,7 \quad (110)$$

$$\omega_n = 1,9 \text{ rad/s} \quad (111)$$

Após a discretização do sistema representado pela equação (109), obteve-se a resposta pretendida inicialmente, conforme exibido no Gráfico 15.

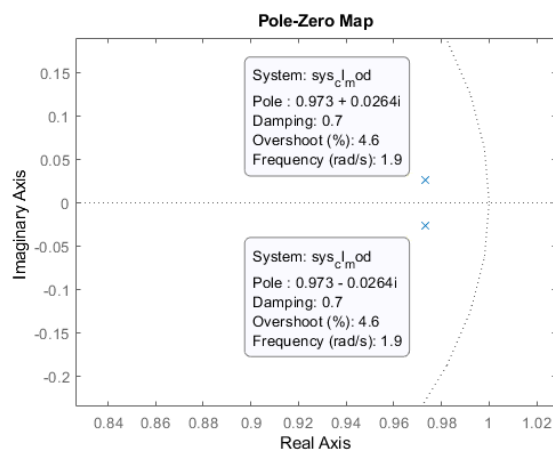
Gráfico 15 – Resposta Desejada (Modelo de Sistema de Segunda Ordem)



Fonte: Próprio Autor

Os polos presentes no sistema discreto de segunda ordem foram utilizados como polos dominantes, juntamente com a posterior adição de polos auxiliares, para, assim, comporem a resposta em malha fechada do sistema controlado. Os polos do sistema de segunda ordem após sua discretização são exibidos Gráfico 16.

Gráfico 16 – Polos Dominantes



Fonte: Próprio Autor

6.2 Definições Acerca dos Polinômios H_s , H_r , S' , R' , P , A' e B'

Com o objetivo de zerar o erro de regime permanente do sistema, usa-se $H_S(z^{-1})$, neste caso, como um filtro integrador, conforme a equação (112).

$$H_S(z^{-1}) = 1 - z^{-1} \quad (112)$$

Para este projeto, *a priori*, não foi vislumbrada a necessidade de um filtro $H_R(z^{-1})$, portanto, conforme a equação (113):

$$H_R(z^{-1}) = 1 \quad (113)$$

Sejam N_A , N_B , $N_{S'}$, $N_{R'}$ e d , respectivamente, os graus dos polinômios $A(z^{-1})$, $B(z^{-1})$, do polinômio $S'(z^{-1})$, do polinômio $R'(z^{-1})$ e o atraso de resposta. Para que haja uma solução única e mínima para a equação característica do sistema projetado, os graus dos polinômios S' e R' foram calculados conforme as equações (73) e (74), resultando nas equações (114) e (115), a seguir.

$$N_{S'} = 6 \quad (114)$$

$$N_{R'} = 2 \quad (115)$$

Conforme definido na subseção 3.2.1, $P(z^{-1})$ é o polinômio característico da resposta em malha fechada do sistema projetado. A ordem de $P(z^{-1})$ é definida pela equação (72), resultando, via substituição direta, na equação (116), a seguir.

$$N_p = 9 \quad (116)$$

O polinômio $A'(z^{-1})$ é definido como a convolução entre o polinômio $A(z^{-1})$ proveniente da planta e o polinômio $H_S(z^{-1})$, conforme a equação (69). Logo, substituindo-se (112) em (69), origina-se a equação (117), a seguir.

$$A'(z^{-1}) = 1 - 2,9492 \cdot z^{-1} + 2,9078 \cdot z^{-2} - 0,9585 \cdot z^{-3} \quad (117)$$

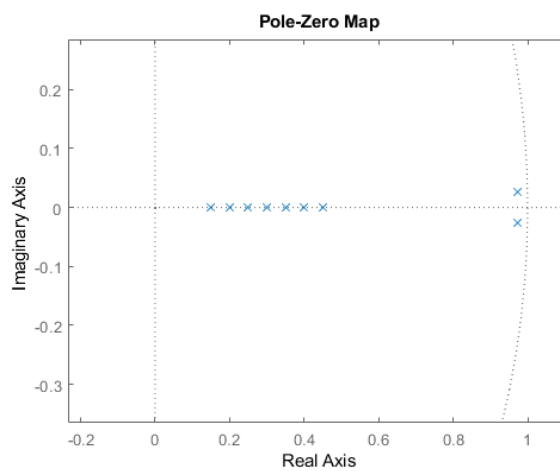
O polinômio $B'(z^{-1})$, por sua vez, é definido, segundo a equação (70), como a convolução entre o polinômio $B(z^{-1})$, proveniente da planta, o monômio z^{-6} , representativo do atraso do sistema, e o polinômio $H_r(z^{-1})$, neste caso, unitário. Logo, origina-se a equação (118), a seguir.

$$B'(z^{-1}) = 0.0032 \cdot z^{-7} \quad (118)$$

6.3 Alocação de Polos

Ao analisar-se o grau do polinômio $P(z^{-1})$, por meio da equação (116), conclui-se que além dos dois polos dominantes projetados na subseção 6.1 será necessário alocar mais 7 polos auxiliares para compor a resposta desejada do sistema controlado. Estes foram alocados através do método de tentativa e erro, sobre o eixo real em 0,15, 0,20, 0,25, 0,30, 0,35, 0,40 e 0,45, respectivamente, sendo, portanto, polos mais rápidos que os polos dominantes do sistema. A alocação de todos os polos é exibida no Gráfico 17.

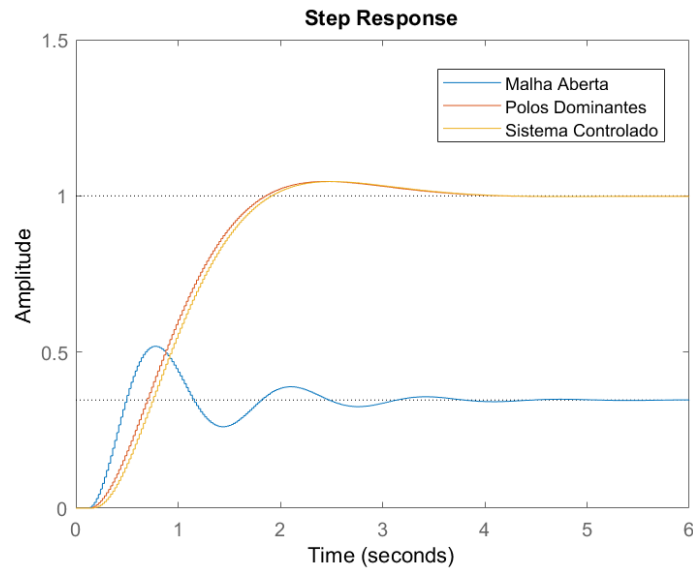
Gráfico 17 – Alocação de Polos



Fonte: Próprio Autor

No Gráfico 18 são exibidas: a resposta projetada via sistema de segunda ordem, a resposta modificada do futuro sistema em malha fechada com a inserção de polos auxiliares e a resposta do sistema identificado em malha aberta, para efeito de comparação.

Gráfico 18 – Respostas ao Degrau



Fonte: Próprio Autor

Uma vez verificado que as especificações desejadas de performance do sistema em malha fechada foram qualitativamente mantidas com a inserção dos polos auxiliares, deu-se prosseguimento ao projeto.

6.4 Cálculo de R, S e T

Mediante cálculos e definições realizados na subseção 6.2, sendo $N_{A'} = 3$ e $N_{B'} = 7$, define-se a matriz quadrada M, de tamanho 10 ($N_{A'} + N_{B'}$), através da equação (119).

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a'_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a'_2 & a'_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a'_3 & a'_2 & a'_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a'_3 & a'_2 & a'_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a'_3 & a'_2 & a'_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a'_3 & a'_2 & a'_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a'_3 & a'_2 & a'_1 & b'_7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a'_3 & a'_2 & 0 & b'_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a'_3 & 0 & 0 & b'_7 \end{bmatrix} \quad (119)$$

Substituindo-se os coeficientes dos polinômios $A'(z^{-1})$ e $B'(z^{-1})$ na matriz M, é obtida sua expressão, exposta pela equação (120), a seguir.

$$M = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -2,9492 & 1 & \dots & 0 & 0 & 0 & 0 \\ 2,9078 & -2,9492 & \dots & 0 & 0 & 0 & 0 \\ -0,9585 & 2,9078 & \dots & 0 & 0 & 0 & 0 \\ 0 & -0,9585 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 0 & \dots & -2,9492 & 0,0032 & 0 & 0 \\ 0 & 0 & \dots & 2,9078 & 0 & 0,0032 & 0 \\ 0 & 0 & \dots & -0,9585 & 0 & 0 & 0,0032 \end{bmatrix} \quad (120)$$

Define-se a matriz-coluna de incógnitas, X, composta pelos coeficientes dos polinômios $R'(z^{-1})$ e $S'(z^{-1})$, através da equação (121).

$$X^T = [s'_0 \quad s'_1 \quad s'_2 \quad s'_3 \quad s'_4 \quad s'_5 \quad s'_6 \quad r'_0 \quad r'_1 \quad r'_2] \quad (121)$$

Define-se também a matriz-coluna de resultados, P, composta pelos coeficientes do polinômio $P(z^{-1})$, através da equação (122).

$$P^T = [1 \quad p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5 \quad p_6 \quad p_7 \quad p_8 \quad p_9] \quad (122)$$

Substituindo-se os coeficientes do polinômio $P(z^{-1})$ na matriz-coluna P, é obtida sua expressão, conforme exibido pela equação (123).

$$P^T = [1 \quad -4,0467 \quad 6,8914 \quad -6,4949 \quad 3,7470 \quad -1,3793 \quad 0,3244 \\ -0,0471 \quad 0,0038 \quad -0,0001] \quad (123)$$

Para obter-se os valores da matriz-coluna de incógnitas e, finalmente, os parâmetros do controlador, foi necessário resolver a equação matricial representada pela equação (124).

$$MX = P \quad (124)$$

$$M^{-1}MX = M^{-1}P \quad (125)$$

$$X = M^{-1}P \quad (126)$$

Obteve-se, então, o vetor-solução X, da equação matricial (124), exibida pela equação (127), a seguir.

$$X^T = [1 \quad -1,0975 \quad 0,7467 \quad -0,1426 \quad 0,1044 \quad 0,0593 \quad 0,0594 \quad 17,0851 \quad -34,6683 \quad 17,6166] \quad (127)$$

Calcula-se, finalmente, $S(z^{-1})$ e $R(z^{-1})$ mediante as definições contidas nas equações (64) e (65), originado portanto, as equações (128) e (129), respectivamente.

$$S(z^{-1}) = 1 - 2,0978.z^{-1} + 1,8442.z^{-2} - 0,8895.z^{-3} + 0,2471.z^{-4} - 0,0450.z^{-5} - 0,0001.z^{-6} - 0,0592.z^{-7} \quad (128)$$

$$R(z^{-1}) = 17,0851 - 34,6682.z^{-1} + 17,6167.z^{-2} \quad (129)$$

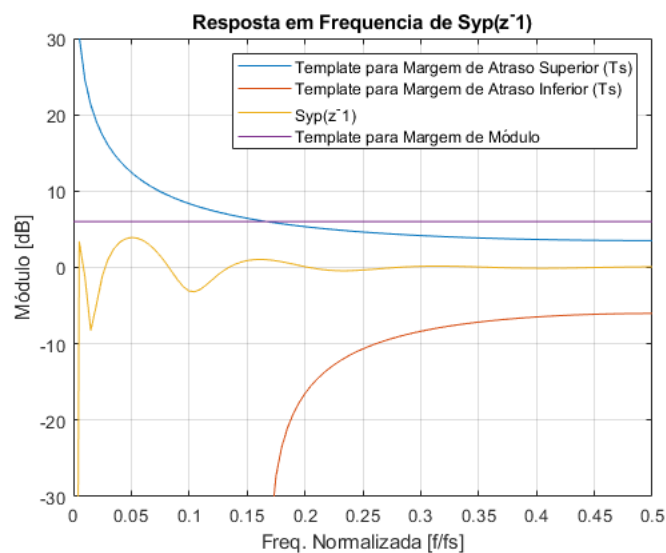
$T(z^{-1})$, por sua vez, é calculado de acordo com a definição pela equação (86), originando, portanto, a equação (130).

$$T(z^{-1}) = 0.0335 \quad (130)$$

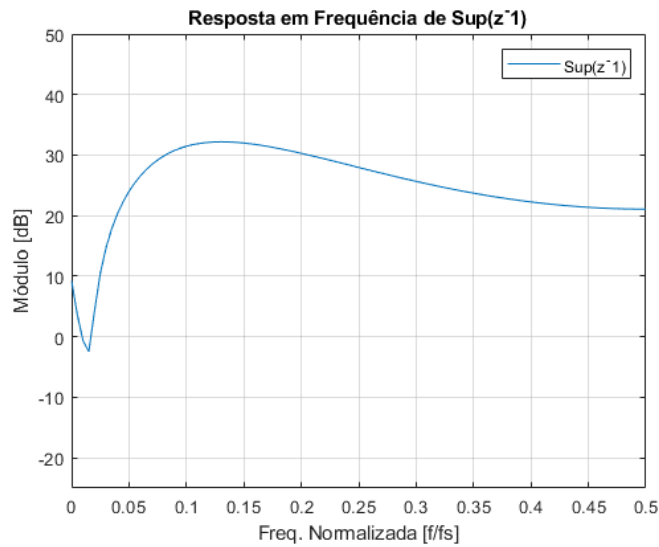
6.5 Funções de Sensibilidade Resultantes

São exibidas por meio do Gráfico 19 e do Gráfico 20, respectivamente, as funções de sensibilidade $S_{yp}(z^{-1})$ e $S_{up}(z^{-1})$, definidas na subseção 3.2.3 deste trabalho.

Gráfico 19 – Resposta em Frequência de $S_{yp}(z^{-1})$



Fonte: Próprio Autor

Gráfico 20 – Resposta em Frequência de $S_{up}(z^{-1})$ 

Fonte: Próprio Autor

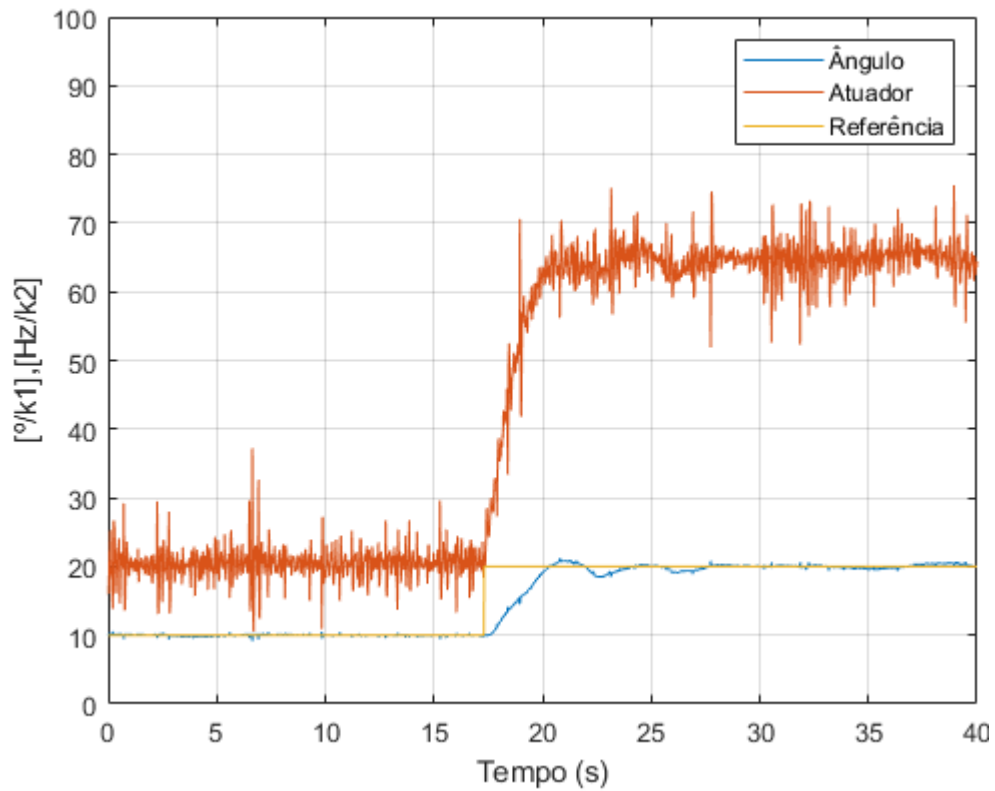
6.6 Ensaio de Teste do Controlador

A metodologia prática deste ensaio é semelhante à apresentada na subseção 5.1, com a diferença do script `interface.py` ter sido substituído pelo script `interface2.py` (disponível no ANEXO G) e o programa de identificação instalado na FRDM-KL25Z ter sido substituído por um programa com a implementação das funções de controle (disponível no anexo F).

O script `interface2.py` além da responsabilidade de ler os dados recebidos da FRDM-KL25Z via USB, também teve a responsabilidade de enviar dados para a FRDM-KL25Z. Neste caso, os dados enviados foram inseridos pelo usuário e consistiam nos valores de referência utilizados pelo algoritmo de controle.

É ilustrado pelo Gráfico 21 o resultado do ensaio de teste do controlador utilizado, em torno do ponto quiescente identificado.

Gráfico 21 – Ensaio do Sistema Controlado



Fonte: Próprio Autor

É possível perceber, de antemão, a tentativa bem sucedida do controlador de manter o erro de regime permanente do sistema nulo (na prática, mínimo).

No degrau ascendente, observa-se que houve uma ultrapassagem percentual de 4,5%, 0,5% menor que o especificado na subseção 6.1, enquanto o tempo de estabelecimento do degrau ascendente foi de 4,3s, 43,3% maior do que o especificado para este projeto.

6.7 Ajuste dos Parâmetros do Controlador

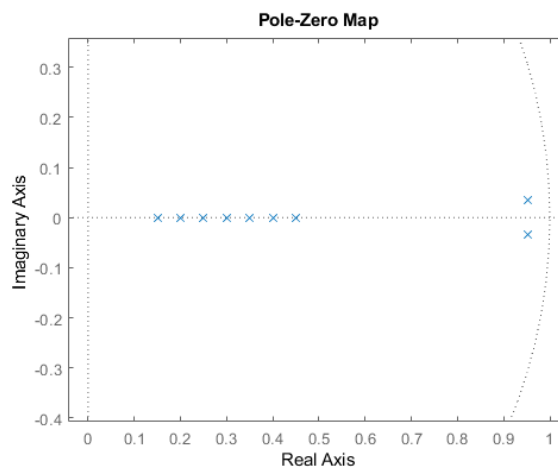
Com o objetivo de tornar a resposta transitória mais rápida, obedecendo-se, portanto, as especificações iniciais de projeto, alterou-se os parâmetros do controlador projetado. Os novos parâmetros de coeficiente de amortecimento e frequência natural do sistema são mostrados através das equações (131) e (132), a seguir.

$$\xi = 0,8 \quad (131)$$

$$\omega_n = 3 \text{ rad/s} \quad (132)$$

Os polos dominantes, resultantes da discretização de um sistema de segunda ordem (com atraso) com os parâmetros reespecificados, juntamente com os polos auxiliares utilizados (os mesmos da situação anterior), são ilustrados pelo Gráfico 22, a seguir.

Gráfico 22 – Alocação de Polos (2)



Fonte: Próprio Autor

De forma idêntica à subseção 6.4, calculou-se os parâmetros do novo controlador, os quais são exibidos pelas equações (133), (134) e (135).

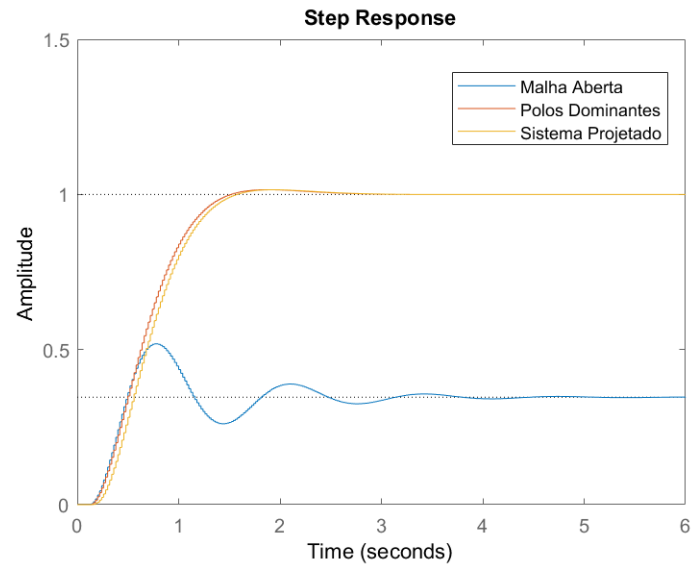
$$S(z^{-1}) = 1 - 2,0557 \cdot z^{-1} + 1,7982 \cdot z^{-2} - 0,8582 \cdot z^{-3} + 0,2409 \cdot z^{-4} - 0,0408 \cdot z^{-5} + 0,0021 \cdot z^{-6} - 0,0864 \cdot z^{-7} \quad (133)$$

$$R(z^{-1}) = 26,2398 - 51,8877 \cdot z^{-1} + 25,7295 \cdot z^{-2} \quad (134)$$

$$T(z^{-1}) = 0.0817 \quad (135)$$

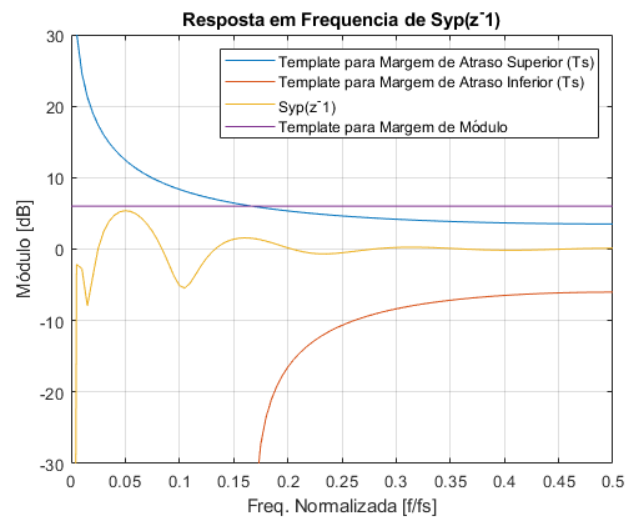
A nova resposta projetada via sistema de segunda ordem, a resposta modificada do novo sistema em malha fechada e a resposta do sistema identificado em malha aberta são exibidas pelo Gráfico 23, a seguir.

Gráfico 23 – Respostas ao Degrau (2)

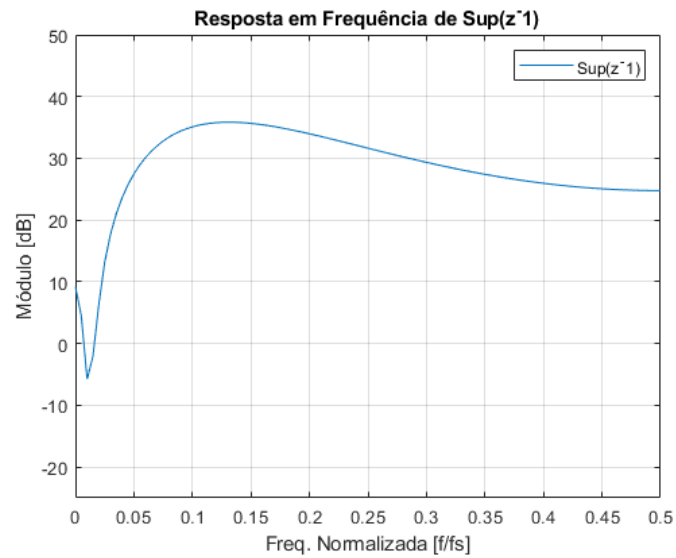


Fonte: Próprio Autor

As novas funções de sensibilidade, $S_{yp}(z^{-1})$ e $S_{up}(z^{-1})$, são exibidas através do Gráfico 24 e do Gráfico 25, respectivamente.

Gráfico 24 – Resposta em Frequência de $S_{yp}(z^{-1})$ (2)

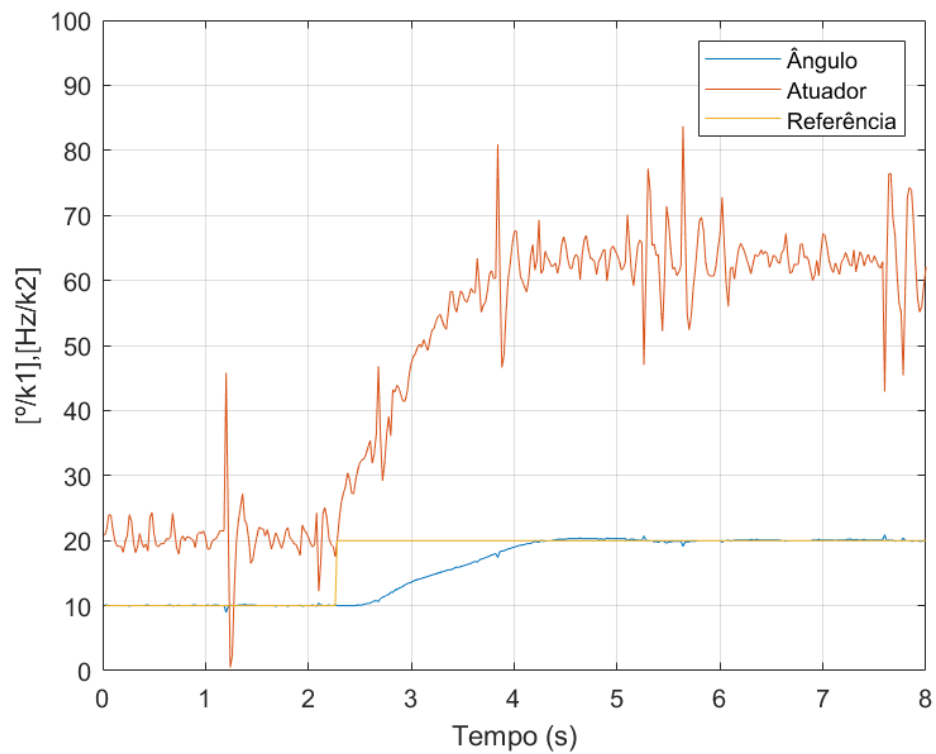
Fonte: Próprio Autor

Gráfico 25 – Resposta em Frequência de $S_{up}(z^{-1})$ (2)

Fonte: Próprio Autor

Após as devidas modificações de projeto, um novo ensaio de testes foi realizado e o resultado é exibido no Gráfico 26, a seguir.

Gráfico 26 – Ensaio do Sistema Controlado (2)



Fonte: Próprio Autor

Com este novo controlador, também é possível perceber, de antemão, a tentativa bem sucedida do controlador de manter o erro de regime permanente do sistema nulo (na prática, mínimo).

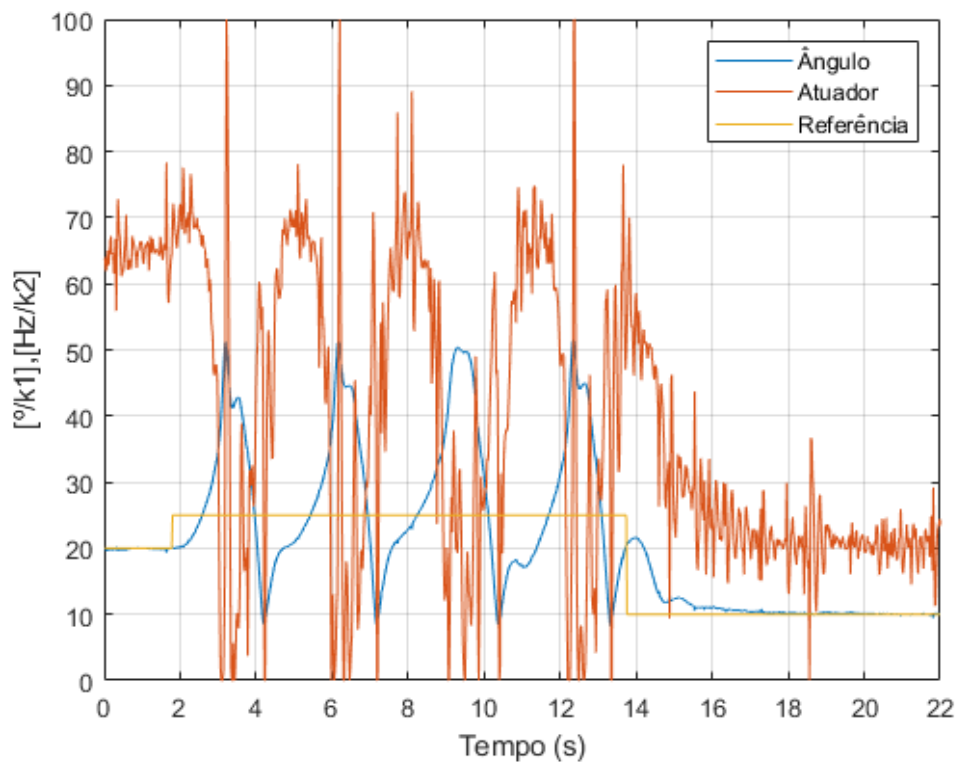
No degrau ascendente, observa-se que houve uma ultrapassagem percentual de 2%, 3% menor que o especificado na subseção 6.1, enquanto o tempo de estabelecimento do degrau ascendente foi de 1,9s, 36,7% menor que o especificado, satisfazendo, portanto, com bastante margem, as especificações para este projeto.

6.8 Limite Prático do Modelo Identificado

Dadas as condições de identificação em torno de 15 unidades de ângulo ($^{\circ}/k1$), era certo que havia um limite operacional (um ângulo máximo) para o qual o modelo perderia sua capacidade de representar a realidade, levando o sistema à instabilidade.

Testes experimentais mostraram que, para este caso, o sistema entra em colapso a partir de 25 unidades de ângulo ($^{\circ}/k1$), conforme ilustrado pelo Gráfico 27, a seguir.

Gráfico 27 – Limite do Modelo Linearizado



Fonte: Próprio Autor

7 CONCLUSÕES

O interesse de pesquisa que motivou este trabalho partiu de um tema tão instigante quanto complexo, cujo interesse é crescente na sociedade, devido à grande perspectiva de impacto e mudança significativa na forma como nós realizamos certas atividades, dada a introdução cotidiana crescente de veículos aéreos não tripulados em nossas vidas.

A revisão bibliográfica realizada nas seções 2 e 3 permitiu a aplicação bem sucedida da teoria de identificação de sistemas discretos e controle digital aplicada a um caso prático, inerentemente exposto às mais diferentes complicações experimentais do mundo real, como ruídos e distúrbios aerodinâmicos.

A divisão de um problema complexo em problemas menores e mais simples é solução adequada ao ambiente acadêmico, sendo posta em prática por este trabalho. O projeto da planta foi bem sucedido em criar um ambiente de testes de um sistema de dinâmica rotacional com propulsão aerodinâmica, livrando-se de possíveis complicações relacionadas a sistemas reais e multivariáveis.

A modelagem e impressão 3D tornaram possível a fabricação de peças essenciais para compor o projeto.

O uso da placa de desenvolvimento FRDM-KL25Z em conjunto com a plataforma de desenvolvimento MBED proporcionou bastante facilidade ao projeto, tanto na produção de software quanto na comunicação com o computador por meio da interface serial.

O projeto do sensor foi bem sucedido, possibilitando a medição da variável controlada (ângulo do braço giratório) em meio à um ambiente ruidoso. Considera-se, entretanto, a utilização de um tipo de sensor baseado em um *encoder* digital para melhorar significativamente a relação sinal-ruído do sistema.

Também foi bem sucedido o esquema de atuação do conjunto motor-hélice. Embora tenha-se adotado uma solução complexa, esta mostrou-se eficaz.

O MATLAB, com sua gama de funções e ferramentas, possibilitou a aquisição de dados para identificação do sistema, assim como agilidade no projeto do controlador e uma série de testes para validação e geração dos gráficos presentes neste trabalho.

O uso do Python em combinação com o MATLAB proporcionou um método indireto da aplicação da técnica computacional de multitarefas, uma vez que um *script* em Python foi responsável por coletar e transmitir os dados via interface serial e um *script* no MATLAB foi responsável por exibí-los em tempo real (além de armazená-los). Coube então, ao sistema operacional, gerenciar estas tarefas em paralelo.

Uma vez que não foi possível, durante as pesquisas para realização deste trabalho, assegurar que o algoritmo utilizado pelo MATLAB para identificação de funções polinomiais no *toolbox* System Identification fosse realmente o Mínimos Quadrados Não Recursivo (apresentado na revisão bibliográfica deste trabalho), foi necessário (e possível, graças à fundamentação teórica), criar este algoritmo, a fim de garantir a coesão do trabalho.

A identificação via Método de Gauss-Newton apresentou melhores resultados que a identificação via Mínimos Quadrados Não Recursivo. Corrobora este fato tanto as análises quantitativas (índices de validação e análise de resíduos) quanto análises qualitativas por meio de gráficos das simulações de validação, realizadas na subseções 5.5.3 e 5.5.4 deste trabalho.

Apesar dos resultados obtidos em relação à análise de resíduos (correlação cruzada e autocorrelação) e índices de validação do sistema escolhido, ficarem abaixo do recomendado pela teoria, foi possível, ainda assim, obter bons resultados na prática, o que afirma que o modelo tem boa verossimilhança com o sistema real em torno do ponto quiescente.

O projeto proposto de um controlador com estrutura RST via alocação de polos que realizasse as especificações descritas na subseção 6.1 foi bem sucedido após o reajuste dos parâmetros de projeto, mediante análise prática dos ensaios de teste do controlador.

Notou-se, entretanto, a alta sensibilidade da saída do controlador em relação a variações da saída da planta em função do ruído de medição (função de sensibilidade da entrada, $S_{up}(z^{-1})$). A melhora desta característica poderia se dar devido ao aprimoramento do sensor ou por meio da adição de polos ao filtro $H_r(z^{-1})$, de modo a atenuar a resposta do controlador em altas frequências.

Conclui-se, mediante o exposto, que o objetivo geral, juntamente com os objetivos específicos deste trabalho foram alcançados de forma satisfatória, envolvendo para tal, a fusão de conhecimentos diversos adquiridos ao longo do curso de Engenharia Elétrica, bem como nas pesquisas direcionadas à produção deste trabalho e por ele referenciadas.

REFERÊNCIAS

- NISE, N. S. **Engenharia de Sistemas de Controle**. 6. ed. Rio de Janeiro: LTC, 2016.
- LANDAU, I. D.; ZITO, G. **Digital Control Systems: Design, Identification and Implementation**. Berlim: Springer, 2006.
- COELHO, A. A. R.; COELHO, L. S. **Identificação de Sistemas Dinâmicos Lineares**. 1. ed. Florianópolis: Editora da Universidade Federal de Santa Catarina, 2004.
- OGATA, K. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Pearson Prentice Hall, 2010.
- CHAPRA, S. C.; CANALE, R. P. **Métodos Numéricos para Engenharia**. 5. ed. Porto Alegre: AMGH, 2011.
- LJUNG, L.; GLAD, T. **Modeling of Dynamic Systems**. Englewood Cliffs: P T R Prentice Hall, 1994.
- BRITES, A. D. **Sistema endócrino - Regulação e controle das funções do corpo**. 2013. Disponível em: <https://educacao.uol.com.br/disciplinas/biologia/sistema-endocrino-regulacao-e-controle-das-funcoes-do-corpo.htm>. Acesso em: 2 fev. 2020.
- SABBAGH, D. Killer drones: how many are there and who do they target? **The Guardian**, Manchester, 18 nov. 2019. Disponível em: <https://www.theguardian.com/news/2019/nov/18/killer-drones-how-many-uav-predator-reaper>. Acesso em: 2 fev. 2020.
- D'ONFRO, J. Amazon's New Delivery Drone Will Start Shipping Packages 'In A Matter Of Months'. **Forbes**, 5 jun. 2019. Disponível em: <https://www.forbes.com/sites/jilliandonfro/2019/06/05/amazon-new-delivery-drone-remars-warehouse-robots-alexa-prediction/#3375446d145f>. Acesso em: 2 fev. 2020.
- TKAC, M., MESAROS, P. Utilizing drone technology in the civil engineering. Selected Scientific Papers - **Journal of Civil Engineering**. 2019.

GARTNER Inc. **Gartner Forecasts Global IoT Enterprise Drone Shipments to Grow 50% in 2020**. 4 dez. 2019. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2019-12-04-gartner-forecasts-global-iot-enterprise-drone-shipmen>. Acesso em: 2 fev. 2020.

FLYNT, J. **12 Best Professional Camera Drones of 2019**. 4 abr. 2019. Disponível em: <https://3dinsider.com/professional-camera-drones/>. Acesso em: 2 fev. 2020.

SODRÉ, U. **Equações Diferenciais Ordinárias de Segunda Ordem**. 17 nov. 2006. Disponível em: <http://www.uel.br/projetos/matessencial/superior/edo/edo2ord.htm>. Acesso em: 5 mar. 2020.

NXP SEMICONDUCTORS N.V. **FRDM-KL25Z: Freedom Development Platform for Kinetis® KL14, KL15, KL24, KL25 MCUs**. Disponível em: <https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z>. Acesso em: 14 abr. 2020.

DEJAN. **Arduino Brushless Motor Control Tutorial | ESC | BLDC**. 2019. Disponível em: <https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/>. Acesso em: 14 abr. 2020.

MICROCHIP TECHNOLOGY INC. **MCP1525/41 2.5V and 4.096V Voltage References**. 2012.

YEDAMALE P. **Brushless DC (BLDC) Motor Fundamentals**. Microchip Technology Inc., 2003.

DIVERSI, R.; GUIDORZI, R.; SOVERINI, U. Identification of ARX and ARARX Models in the Presence of Input and Output Noises. **European Journal of Control**. Dez. 2010.

ANEXO A – PROGRAMA GERADOR DE SBPA (MBED)

```

1 //Identificação do Sistema
2 //Programa Gerador de SBPA
3 //7 Celulas (Tb = 7Ts = 0.14s)
4
5 #include "mbed.h"
6 Ticker tick;
7
8 AnalogIn input(PTB1);
9 AnalogIn input_ref(PTC1);
10 DigitalOut LED_R(LED_RED);
11 DigitalOut LED_G(LED_GREEN);
12 DigitalOut LED_B(LED_BLUE);
13 Serial pc(USBTX, USBRX,19200);
14
15 DigitalOut o0(PTC10);
16 DigitalOut o1(PTC6);
17 DigitalOut o2(PTC5);
18 DigitalOut o3(PTC4);
19 DigitalOut o4(PTC3);
20 DigitalOut o5(PTC0);
21 DigitalOut o6(PTC7);
22
23 int msg_count = 0;
24
25 //====Manipulação de Referencia =====
26 //Numero de samples - delay de inicializacao
27 int n_init_delay = 500; //5s
28
29 //Incremento do global_drive (tamanho do degrau)
30 const float inc_global_drive = 5;
31 //Condicao inicial:
32 const float default_drive = 45;
33 float global_drive = default_drive;
34
35 //Controle da Referencia
36 //PRBS (7 CELULAS)
37 int new_value_prbs;
38 int cel_prbs[7] = {1,0,1,0,1,1,0};
39 void prbs_change(){
40     new_value_prbs = cel_prbs[3] ^ cel_prbs[6];
41     cel_prbs[6] = cel_prbs[5];
42     cel_prbs[5] = cel_prbs[4];
43     cel_prbs[4] = cel_prbs[3];
44     cel_prbs[3] = cel_prbs[2];
45     cel_prbs[2] = cel_prbs[1];
46     cel_prbs[1] = cel_prbs[0];
47     cel_prbs[0] = new_value_prbs;
48 }
49
50 int cont_prbs = 0;
51 float degrau_control(){
52     //PRBS sampling control
53     cont_prbs = cont_prbs + 1;
54     if (cont_prbs==7){
55         cont_prbs = 0;
56         prbs_change();

```

```

56         //Muda o GlobalDrive
57         if (cel_prbs[6]==1){
58             global_drive = default_drive+inc_global_drive;
59             LED_B = 1;
60             LED_G = 0;
61         }
62         else{
63             global_drive = default_drive;
64             LED_B = 0;
65             LED_G = 1;
66         }
67     }
68     return global_drive;
69 }
70
71 //=====
72
73 //Funcao de leitura do sensor
74 float pot_read;
75 float get_pot(){
76     pot_read = input.read_ul6();
77     return ((100*pot_read)/65535);
78 }
79
80 //Funcao de atuacao do motor
81 unsigned int i_drive = 0;
82 void atuar(float f_drive){ //f_drive = [0,100]
83     //Conversao
84     i_drive = (int)((127*f_drive)/100);
85     //Saturacao fisica
86     if (i_drive>127){
87         i_drive = 127;
88     }
89     o6 = !((i_drive & 64) >> 6);
90     o5 = !((i_drive & 32) >> 5);
91     o4 = !((i_drive & 16) >> 4);
92     o3 = !((i_drive & 8) >> 3);
93     o2 = !((i_drive & 4) >> 2);
94     o1 = !((i_drive & 2) >> 1);
95     o0 = !(i_drive & 1);
96 }
97
98 // Main Loop & Sample Interrupt
99 //-----
100
101 //Parametros do filtro
102 double sig_y[3] = {0,0,0};
103 double sig_y_raw[3] = {0,0,0};
104 double FB[2] = {0,0.918997407842057};
105 double FA[2] = {1,-0.081002592157943};
106 void sample_interrupt(){
107
108     //Filtro de Entrada e Atualizacao de Y[0]
109     sig_y_raw[0] = get_pot();
110     sig_y[0] = FB[1]*sig_y_raw[1];
111     sig_y[0] -= FA[1]*sig_y[1];
112
113     //Atualizacao do vetor sinal y e y_raw

```

```
114     sig_y[2] = sig_y[1];
115     sig_y[1] = sig_y[0];
116     sig_y_raw[2] = sig_y_raw[1];
117     sig_y_raw[1] = sig_y_raw[0];
118
119     //Atuação
120     if (n_init_delay!=0){
121         n_init_delay = n_init_delay - 1;
122     }else{
123         LED_R = 1;
124         atuar(degrau_control());
125     }
126
127     //Transmite os dados
128     msg_count = msg_count + 1;
129     pc.printf("%d,%.1f,%.1f,0\n",msg_count,sig_y[0],global_drive);
130 }
131
132 int main(){
133     //Acende o LED Vermelho
134     LED_R = 0;
135     LED_B = 1;
136     LED_G = 1;
137     //Configura o timer para 20ms (0.02s)
138     tick.attach(&sample_interrupt, 0.02f);
139     //Loop Principal
140     while (true){
141     }
142 }
143
```

ANEXO B – SCRIPT INTERFACE.PY (PYTHON)

```
#Interface.py
#Leitura e encaminhamento de dados
import serial
import socket
import time

#String -> Matlab
line_read = ''

#Sockets
udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serial_comm = serial.Serial('/dev/ttyACM0', 19200, timeout=1)

#Loop principal
while 1:
    line_read = serial_comm.readline()
    udp_socket.sendto(line_read, ("127.0.0.1", 5001))
    #print(line_read)
```

ANEXO C – SCRIPT AQUISIÇÃO.M (MATLAB)

```

%Programa de Aquisição e Plotagem de Dados
%Identificação e Controle do Sistema

clear all
clc

MSG = [];
REF = [];
POT = [];
DRIVE = [];

%Prepara os gráficos
SENS_GRAPH=plot(NaN,NaN,'r');
hold on
grid on
DRIVE_GRAPH=plot(NaN,NaN,'color',[143/256, 213/256, 227/256]);
REF_GRAPH=plot(NaN,NaN,'g');

%ylim([0 100]);
interv = 500;
axis([0,interv,0,100]);
SENS_VEC = zeros(1, interv);
DRIVE_VEC = zeros(1,interv);
REF_VEC = zeros(1,interv);

%UDP Receiver
udpr = dsp.UDPReceiver('LocalIPPort',5001);
setup(udpr);

ictl = 1;
valid_counter = 0;
while true;
    dataReceived = udpr();
    unicodestr = native2unicode(dataReceived, 'UTF-8');
    REC = split(unicodestr,',');
    %Mensagem Válida
    if length(REC)==4
        valid_counter = valid_counter + 1;

        p4 = split(REC{4},newline);
        REC{4} = p4{1};

        %REC % (MSG -> POT -> DRIVE)

    %Reseta o vetor de resultados
    if ictl == (interv + 1)
        ictl = 1;
        SENS_VEC = zeros(1, interv);
        REF_VEC = zeros(1,interv);
        DRIVE_VEC = zeros(1,interv);
    else

```

```
        ictl = ictl + 1;
    end

    %Atualiza os vetores de dados
    MSG(valid_counter) = str2double(REC{1});
    POT(valid_counter) = str2double(REC{2});
    DRIVE(valid_counter) = str2double(REC{3});
    REF(valid_counter) = str2double(REC{4});

    %Atualiza os vetores visuais
    SENS_VEC(ictl) = str2double(REC{2});
    DRIVE_VEC(ictl) = str2double(REC{3});
    REF_VEC(ictl) = str2double(REC{4});

    %Plota os gráficos
    set(REF_GRAPH, 'XData', 1:interv, 'YData', REF_VEC(1:interv));
    set(SENS_GRAPH, 'XData', 1:interv, 'YData', SENS_VEC(1:interv));
    set(DRIVE_GRAPH, 'XData', 1:interv, 'YData', DRIVE_VEC(1:interv));
    drawnow;
end
end
```

ANEXO D – SCRIPT MÍNIMOS QUADRADOS NÃO RECURSIVO (MATLAB)

```

%Identificação de Sistemas
%Mínimos Quadrados Não-Recursivo

clc
clear all
close all

%Carrega os dados
load('dados_para_ident.mat');
%Vetor de saída: POT
%Vetor de entrada: DRIVE

%Range para identificacao
ident_x_i = 533;
ident_x_f = 13477;
%Range para validacao
valid_x_i = 13294;
valid_x_f = 21290;
valid_offset = 770;

%PLOT - Sinal Inicial
figure;
plot(POT(ident_x_i:ident_x_f));
hold on;
plot(DRIVE(ident_x_i:ident_x_f));

%Estrutura do Sistema
ts = 0.02;
na = 10;
nb = 8;
d = 6;

if ((nb+d)>na)
    back_offset = nb+d;
else
    back_offset = na;
end

%Loop do Sistema Matricial
Y = [];
FI = [];

for i=(ident_x_i+back_offset):(ident_x_f)
    fi_t = [];
    %-y(t-[1..na])
    for j=1:na
        fi_t = [fi_t -1*POT(i-j)];
    end
    %Y(t)
    Y = [Y;POT(i)];
    %+u(t-[1+d]..(d+nb))

```



```

    for j=(1+d):(d+nb)
        fi_t = [fi_t DRIVE(i-j)];
    end
    FI = [FI;fi_t];
end
TETA = ((inv((FI')*FI))*(FI'))*Y;

%Loop de Formacao do Sistema Discreto Identificado
V_DEN = [1];
for i=1:na
    V_DEN = [V_DEN TETA(i)];
end
V_NUM = zeros(1,d+1);
for i=(na+1):length(TETA)
    V_NUM = [V_NUM TETA(i)];
end
SYS = filt(V_NUM,V_DEN,1)

%Simulacao
%Sistema Sem condicionamento
[y_sim,t_sim,x_sim] =
lsim(SYS,DRIVE(ident_x_i:ident_x_f),[ident_x_i:ident_x_f]);

figure
plot([ident_x_i:ident_x_f],POT(ident_x_i:ident_x_f),'Color',[0.76,0.76,0.76
]);
hold;
plot([ident_x_i:ident_x_f],y_sim,'Color',[245,173,66]/255);
legend('Sinal Original','Modelo Simulado');
%=====
%Validacao:
[y_valid_sim,t_valid_sim,x_valid_sim] =
lsim(SYS,DRIVE(valid_x_i:valid_x_f),[valid_x_i:valid_x_f]);

%Residuos
res_y = zeros(1,length(y_valid_sim));
res_y_abs_perc = zeros(1,length(y_valid_sim));
res_y_med = zeros(1,length(y_valid_sim));

%Médias
media_y = mean(y_valid_sim(1+valid_offset:length(y_valid_sim)));
media_res_y = mean(res_y(1+valid_offset:length(res_y)));

for i=1:length(y_valid_sim)
    res_y(i) = POT(i+valid_x_i-1)-y_valid_sim(i);
    res_y_abs_perc(i) = abs(res_y(i))/POT(i+valid_x_i-1);
    res_y_med(i) = POT(i+valid_x_i-1)-media_y;
end

%Plot da Validação
figure
plot([valid_x_i:valid_x_f],POT(valid_x_i:valid_x_f),'Color',[0.76,0.76,0.76
]);
hold;

```

```
plot([valid_x_i:valid_x_f],y_valid_sim,'Color',[0,0,180]/255);
plot([valid_x_i:valid_x_f],res_y,'red');
legend('Sinal Original','Modelo Simulado (Validação)','Vetor de Resíduos');

%Calculos (Levando em conta um certo offset inicial na validação)
SEQ = 0;
DAPM = 0;
R2 = 0;
for i=(1+valid_offset):length(res_y)
    SEQ = SEQ + res_y(i)^2;
    R2 = R2 + res_y_med(i)^2;
    DAPM = DAPM + res_y_abs_perc(i);
end
DAPM = DAPM*100/(valid_x_f-valid_x_i-valid_offset+1);
R2 = 1 - SEQ/R2;

fprintf('SEQ = %d\n',SEQ);
fprintf('DAPM = %.2f%%\n',DAPM);
fprintf('R² = %.2f\n',R2);
```

ANEXO E – SCRIPT DE PROJETO DO CONTROLADOR (MATLAB)

```

%Projeto de Controlador por Alocação de Polos

clear all
close all
clc
format long

ts = 0.02;
z = tf('z',ts);

d = 6;
B_L = [0 0 0 0 0 0 0 0.003216254754234];
A = [1 -1.949263923601094 0.958552095844954];
A_L = conv(A,[1 -1]);

Hol = filt(B_L,A,ts);

%Especificacoes de malha fechada de um sistema proposto =====
w0 = 3;%rad/s
csi = 0.8;

% w0 = 0.5;%rad/s
% csi = 0.7;

s = tf('s');
sys_cl = (w0^2)*exp(-s*(d*ts))/(s^2 + 2*csi*w0*s + w0^2);
%Discretizacao
sys_cl_d = c2d(sys_cl,ts);

pol_dom = pole(sys_cl_d);

%Adicao de polos auxiliares (7) =====
pol_aux = 0.15 + [0;0.05;0.1;0.15;0.2;0.25;0.3];

P = poly([pol_dom;pol_aux]);

sys_cl_mod = filt(sys_cl_d.Numerator,poly([pol_dom;pol_aux]),ts);

%Plot de verificação =====
% figure
% step(Hol);
% hold on;
% step(sys_cl);
% step(sys_cl_d);
% K = step(sys_cl_mod);
% sys_cl_mod_corr = sys_cl_mod/K(length(K));
% step(sys_cl_mod_corr);
% legend
%
% figure
% pzmap(sys_cl_mod)

```

```

% return
=====

%Construcao da matriz M (10x10)
M = [];
%Parte A'n:
nblinha = 7;
for i=1:nblinha
    lin = [];
    lin = [zeros(1,i-1) A_L zeros(1,nblinha-i)];
    M = [M lin'];
end
%Parte B'n:
nalinha = 3;
for i=1:nalinha
    lin = [];
    lin = [zeros(1,i-1) B_L zeros(1,nalinha-i)];
    M = [M lin'];
end

%Solucao =====
X = inv(M)*P';
S_L = [X(1:7)];
S_L = S_L';
Hs = [1 -1];
R = [X(8:10)];
R = R';
T = sum(R); %T = R(1)
S = conv(Hs,S_L);

%Diagrama de Blocos =====
tf_S = filt(1,S,ts);
tf_R = filt(R,1,ts);
tf_T = filt(T,1,ts);
tf_planta = filt(B_L,A,ts);

SYS = tf_S*tf_planta;
SYS = feedback(SYS,tf_R);
SYS = tf_T*SYS;

%Plot do Sistema Controlado =====
figure
step(Hol);
hold
legend
step(sys_cl_d);
step(SYS);
figure
pzmap(SYS);

%Robustez =====
Ts=ts;

%Syp

```

```

Syp = tf([conv(A,S)], [conv(A,S) + conv(B_L,R)], Ts);
figure
w1=[0:0.01*pi:pi]/Ts;
[MSyp, PSyp]=bode(Syp,w1);
MSypdB=20*log10(MSyp);
for Ib=1:length(MSypdB)
    MSypdB(1,Ib)=MSypdB(1,1,Ib);
end
MOD1=(sqrt(2-2*cos(w1*Ts))+1)./(sqrt(2-2*cos(w1*Ts)));
MODdB1=20*log10(MOD1);
MOD2=(sqrt(2-2*cos(w1*Ts))-1)./(sqrt(2-2*cos(w1*Ts)));
MODdB2=real(20*log10(MOD2));
for Ic=1:(0.16667*(2*pi/Ts))/(0.01*pi/Ts)
    MODdB2(Ic)=-94.8284;
end
DELTA_M=0.5;
MODdB3=-20*log10(DELTA_M)*ones(1,length(w1));
plot(w1/(2*pi/Ts),MODdB1,w1/(2*pi/Ts),MODdB2,w1/(2*pi/Ts),MSypdB,w1/(2*pi/Ts),MODdB3);grid
title('Resposta em Frequencia de Syp(z^-1)');
axis([0 0.5 -30 30]);
legend('Template para Margem de Atraso Superior (Ts)', 'Template para Margem de Atraso Inferior (Ts)', 'Syp(z^-1)', 'Template para Margem de Módulo');
xlabel('Freq. Normalizada [f/fs]');
ylabel('Módulo [dB]');

%Sup
Sup = tf([conv(A,R)], [conv(A,S) + conv(B_L,R)], Ts);
figure
w1=[0:0.01*pi:pi]/Ts;%w1=Wnorm/Ts
[MSup, PSup]=bode(Sup,w1);
MSupdB=20*log10(MSup);
for Ib=1:length(MSupdB)
    MSupdB(1,Ib)=MSupdB(1,1,Ib);
end
MOD1=(sqrt(2-2*cos(w1*Ts))+1)./(sqrt(2-2*cos(w1*Ts)));
MODdB1=20*log10(MOD1);
MOD2=(sqrt(2-2*cos(w1*Ts))-1)./(sqrt(2-2*cos(w1*Ts)));
MODdB2=real(20*log10(MOD2));
for Ic=1:(0.16667*(2*pi/Ts))/(0.01*pi/Ts)
    MODdB2(Ic)=-94.8284;
end
DELTA_M=0.5;
MODdB3=-20*log10(DELTA_M)*ones(1,length(w1));

plot(w1/(2*pi/Ts),MSupdB);grid
title('Resposta em Frequência de Sup(z^-1)');
axis([0 0.5 -25 50]);
xlabel('Freq. Normalizada [f/fs]');
ylabel('Módulo [dB]');
legend('Sup(z^-1)');

```

ANEXO F –PROGRAMA DE IMPLEMENTAÇÃO DO CONTROLADOR (MBED)

```

1 //Implementação do Controlador (2ª versão)
2
3 #include "mbed.h"
4 Ticker tick;
5
6 AnalogIn input(PTB1);
7 AnalogIn input_ref(PTC1);
8 DigitalOut LED_R(LED_RED);
9 DigitalOut LED_G(LED_GREEN);
10 DigitalOut LED_B(LED_BLUE);
11 Serial pc(USBTX, USBRX,19200);
12
13 DigitalOut o0(PTC10);
14 DigitalOut o1(PTC6);
15 DigitalOut o2(PTC5);
16 DigitalOut o3(PTC4);
17 DigitalOut o4(PTC3);
18 DigitalOut o5(PTC0);
19 DigitalOut o6(PTC7);
20
21 float global_pot = 0;
22 int msg_count = 0;
23 uint8_t NUM_REC = 0;
24
25 //Numero de samples - delay de inicializacao
26 int n_init_delay = 500; //5s
27
28 //=====
29
30 //Funcao de leitura do sensor
31 float pot_read;
32 float get_pot(){
33     pot_read = input.read_ul6();
34     global_pot = ((100*pot_read)/65535);
35     return global_pot;
36 }
37
38 //Funcao de atuação do motor
39 unsigned int i_drive = 0;
40 void atuar(float f_drive){ //f_drive = [0,100]
41     //Conversao
42     i_drive = (int)((127*f_drive)/100);
43     //Saturacao fisica
44     if (i_drive>127){
45         i_drive = 127;
46     }
47     o6 = !((i_drive & 64) >> 6);
48     o5 = !((i_drive & 32) >> 5);
49     o4 = !((i_drive & 16) >> 4);
50     o3 = !((i_drive & 8) >> 3);
51     o2 = !((i_drive & 4) >> 2);
52     o1 = !((i_drive & 2) >> 1);
53     o0 = !(i_drive & 1);
54 }
55 //Parametros do filtro
56 double sig_y_raw[3] = {0,0,0};

```

```

56 double FB[2] = {0,0.918997407842057};
57 double FA[2] = {1,-0.081002592157943};
58
59 //Parametros da Planta
60 double B[8] = {0,0,0,0,0,0,0,0.003315741293108};
61 double A[3] = {1,-1.933337821723211,0.943800700396816};
62
63 //Parametros do Controlador
64 double T = 0.081702984523165;
65 double R[3] = {26.239891337684075,-51.887786156240097,25.729597803079187};
66 double S[8] = {1,-2.055768522568330,1.798244640862376,-
67 0.858294991266574,0.240993826389628,-0.040830317638961,0.002120895299646,-
68 0.086465531077784};
69
70 double sig_x = 0;
71 double sig_w = 0;
72 double sig_y[3] = {0,0,0};
73 double sig_z[8] = {0,0,0,0,0,0,0,0};
74 double sig_r = 15;
75
76 //Função de Controle =====
77 const int up_sat = 100; //maxima velocidade [0,100]
78
79 float drive_control(){
80     //Filtro de Entrada e Atualizacao de Y[0]
81     sig_y_raw[0] = get_pot();
82     sig_y[0] = FB[1]*sig_y_raw[1];
83     sig_y[0] -= FA[1]*sig_y[1];
84
85     //[Bloco R] Calculo do sinal w
86     sig_w = R[0]*sig_y[0];
87     sig_w += R[1]*sig_y[1];
88     sig_w += R[2]*sig_y[2];
89
90     //Calculo do sinal x
91     sig_x = T*sig_r;
92
93     //[Bloco 1/S] Calculo do sinal z atual
94     sig_z[0] = (sig_x - sig_w);
95     sig_z[0] -= S[1]*sig_z[1];
96     sig_z[0] -= S[2]*sig_z[2];
97     sig_z[0] -= S[3]*sig_z[3];
98     sig_z[0] -= S[4]*sig_z[4];
99     sig_z[0] -= S[5]*sig_z[5];
100    sig_z[0] -= S[6]*sig_z[6];
101    sig_z[0] -= S[7]*sig_z[7];
102
103    //Saturacao
104    if (sig_z[0]>up_sat){
105        sig_z[0] = up_sat;
106    }
107    else if (sig_z[0]<0){
108        sig_z[0] = 0;
109    }
110
111    //Atualizacao do vetor sinal z
112    sig_z[7] = sig_z[6];
113    sig_z[6] = sig_z[5];

```

```

114     sig_z[5] = sig_z[4];
115     sig_z[4] = sig_z[3];
116     sig_z[3] = sig_z[2];
117     sig_z[2] = sig_z[1];
118     sig_z[1] = sig_z[0];
119
120     //Atualizacao do vetor sinal y e y_raw
121     sig_y[2] = sig_y[1];
122     sig_y[1] = sig_y[0];
123     sig_y_raw[2] = sig_y_raw[1];
124     sig_y_raw[1] = sig_y_raw[0];
125
126     return sig_z[0];
127 }
128
129 // Main Loop & Sample Interrupt
130 //-----
131 void sample_interrupt(){
132     //Atualiza referencia
133     if(pc.readable()){
134         NUM_REC = (uint8_t)pc.getc();
135         //Check
136         if (NUM_REC<=100){
137             sig_r = NUM_REC;
138         }
139     }
140     //Atualizacao
141     if (n_init_delay!=0){
142         n_init_delay = n_init_delay - 1;
143         sig_y[0] = get_pot();
144         sig_y[2] = sig_y[1];
145         sig_y[1] = sig_y[0];
146     }else{
147         LED_R = 1;
148         atuar(drive_control());
149     }
150
151     //Transmite os dados
152     msg_count = msg_count + 1;
153     pc.printf("%d,%.1f,%.1f,%.1f%\n",msg_count,sig_y[0],sig_z[0],sig_r);
154 }
155
156 int main(){
157     //Acende o LED Vermelho
158     LED_R = 0;
159     LED_B = 1;
160     LED_G = 1;
161     //Configura o timer para 20ms (0.02s)
162     tick.attach(&sample_interrupt, 0.02f);
163     //Loop Principal
164     while (true){
165

```


ANEXO G – SCRIPT INTERFACE2.PY (PYTHON)

```

#Interface2.py
#Leitura, escrita e encaminhamento de dados
import serial
import socket
import time
import _thread

#Valor de ajuste do setpoint
ref_value = 0

#String -> Matlab
line_read = ''

#Sockets
udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serial_comm = serial.Serial('/dev/ttyACM0', 19200, timeout=1)

#Thread de escrita
def serial_handler():
    print('[Lendo serial em /dev/ttyACM0]')
    while 1:
        try:
            ref = int(input('Digite o valor da referência:'))
        except ValueError:
            print('Erro: valor de referência inválido.')
        else:
            if (ref<0 or ref>100):
                print('Erro: valor de referência fora do limite.')
            else:
                ref_value = ref
                serial_comm.write(bytes([ref_value]))
                print('Novo valor de referência: '+str(ref_value))

    try:
        _thread.start_new_thread(serial_handler, ())
    except:
        print("Erro: não foi possível criar a thread.")
        exit()

#Loop principal
while 1:
    line_read = serial_comm.readline()
    udp_socket.sendto(line_read, ("127.0.0.1", 5001))

```