

**INSTITUTO FEDERAL
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Pelotas**

**Curso de Engenharia Elétrica
Monografia do Projeto de Fim de Curso**

**Desenvolvimento de uma Bancada Automatizada para a Identificação
de Parâmetros de Modelo Matemático de Motores de Corrente**

Contínua de Baixo Custo

Thales Gonçalves Ferreira

2023

Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense
Departamento de Ensino de Graduação e Pós-Graduação
Campus Pelotas
Curso de Engenharia Elétrica

Thales Gonçalves Ferreira

Orientador: Prof. Dr. Cláudio Luís d'Elia Machado

Coorientador: Prof. Me. Fabiano Sandrini Moraes

**Desenvolvimento de uma Bancada Automatizada para a Identificação de Parâmetros de
Modelo Matemático de Motores de Corrente Contínua de Baixo Custo**

Monografia do Projeto de Fim de Curso

Pelotas-RS

2023

Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense
Departamento de Ensino de Graduação e Pós-Graduação
Curso de Engenharia Elétrica

Monografia do Projeto de Fim de Curso

**Desenvolvimento de uma Bancada Automatizada para a Identificação de Parâmetros de
Modelo Matemático de Motores de Corrente Contínua de Baixo Custo**

Thales Gonçalves Ferreira

Relatório submetido como requisito parcial
para obtenção do grau de Engenheiro Eletricista

Prof. Dr. Cláudio Luís d'Elia Machado
Orientador

Prof. Me. Fabiano Sandrini Moraes
Coorientador

Banca examinadora

Prof. Dr. Cláudio Luís d'Elia Machado

Prof. Dr. Carlos Mendes Richter

Prof. Dr. Mauro André Barbosa Cunha

CESSÃO DE DIREITOS

AUTOR: Thales Gonçalves Ferreira

TÍTULO: Desenvolvimento de uma Bancada Automatizada para a Identificação de Parâmetros de Modelo Matemático de Motores de Corrente Contínua de Baixo Custo

GRAU: Engenheiro Eletricista ANO: 2023

É concedida ao Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense permissão para reproduzir cópias desta monografia de graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O(s) autor(es) reserva(m) outros direitos de publicação e nenhuma parte desta monografia de graduação pode ser reproduzida sem autorização por escrito do(s) autor(es).

Thales Gonçalves Ferreira
Av. 25 de Julho 327, casa 125
96065-620 – Pelotas – RS – Brasil

Agradecimentos

Gostaria de expressar minha profunda gratidão a todas as pessoas que contribuíram para a realização deste trabalho de conclusão de curso. Sem o apoio e incentivo de cada um de vocês, este projeto não seria possível. Portanto, dedico esta folha de agradecimentos às seguintes pessoas:

À minha esposa, Tamara, que esteve ao meu lado durante toda essa jornada acadêmica, oferecendo seu amor, paciência e compreensão. Seu apoio incondicional e motivação foram fundamentais para superar os desafios e alcançar este objetivo.

À minha família, pelo constante suporte e encorajamento ao longo dos anos. Vocês sempre acreditaram em mim.

Aos amigos Rafael Galli e Igor da Rocha Barros, por compartilharem seu conhecimento e experiência, bem como por fornecerem um ambiente de aprendizado inspirador. Suas orientações, feedbacks e colaboração foram essenciais para a realização deste trabalho.

Ao meu orientador, Cláudio Machado, pela sua orientação dedicada e valiosa durante todo o processo de elaboração deste trabalho. Sua expertise, sugestões e visão crítica foram fundamentais para o aprimoramento deste projeto.

Aos meus colegas e amigos de faculdade, pelos momentos compartilhados, apoio mútuo e troca de conhecimentos. Juntos, enfrentamos desafios e superamos obstáculos, tornando essa jornada acadêmica muito mais rica e gratificante.

Aos funcionários do curso de Engenharia Elétrica, pelos seus esforços em manter um ambiente propício ao aprendizado, fornecendo recursos e apoio administrativo necessário para o bom andamento do curso.

Aos professores do curso de Engenharia Elétrica, pela dedicação em transmitir conhecimentos, incentivar a pesquisa e aprimorar nossas habilidades técnicas. Suas aulas foram fundamentais para a formação acadêmica e profissional.

A todos que, direta ou indiretamente, contribuíram para o desenvolvimento deste trabalho, meu mais profundo agradecimento. Vocês fizeram parte dessa conquista e deixaram uma marca significativa em minha jornada acadêmica.

Resumo

Para projetar um controlador com base num modelo matemático não linear é fundamental possuir conhecimento dos valores dos parâmetros do modelo matemático do sistema que se pretende controlar. Este trabalho tem como objetivo desenvolver uma bancada automatizada para a identificação dos parâmetros de um modelo matemático não linear de um motorredutor de corrente contínua de baixo custo. Um motorredutor é um dispositivo que combina um motor elétrico e um sistema de redução de velocidade para fornecer torque e controlar a velocidade de rotação. A bancada experimental, composta por *hardware* e algoritmo de controle, permitirá obter os parâmetros necessários para projetos de controladores que tenham como base modelos matemáticos como o utilizado neste trabalho. Além disso, será implementado um controlador de velocidade em malha fechada para validar o modelo matemático obtido e apresentar os resultados de desempenho do controlador. O principal objetivo é oferecer uma metodologia eficiente e acessível para o desenvolvimento de projetos que utilizem motores de corrente contínua de baixo custo, ampliando assim suas possibilidades de aplicação em sistemas automatizados e robóticos.

Palavras-chaves: controlador não linear. identificação dos parâmetros. motorredutor de corrente contínua. modelo matemático não linear.

Abstract

Designing a controller based on a nonlinear mathematical model requires knowledge of the parameter values of the system under control. This work aims to develop an automated setup for parameter identification of a low-cost, nonlinear mathematical model of a DC gear motor. A gearmotor is a device that combines an electric motor and a speed reduction system to provide torque and control the rotation speed. The experimental setup, comprising hardware and control algorithms, will facilitate obtaining the necessary parameters for controller design based on mathematical models like the one used in this study. Additionally, a closed-loop speed controller will be implemented to validate the obtained mathematical model and present the controller's performance results. The primary objective is to provide an efficient and affordable methodology for the development of projects utilizing low-cost DC motors, thereby expanding their potential applications in automated systems and robotics.

Key-words: Controller design. Nonlinear mathematical model. Parameter identification. DC gear motor.

Lista de ilustrações

Figura 1 – Tabela característica do motor 0615N	14
Figura 2 – Tabela característica do motor de vidro elétrico	15
Figura 3 – Especificações do motor de kit de Arduino	15
Figura 4 – Modelo de motor CC	17
Figura 5 – Modelo de motor CC	18
Figura 6 – Representação da zona morta	21
Figura 7 – Ligação do motor para medição de parâmetros	21
Figura 8 – Circuito do primeiro experimento	26
Figura 9 – Exemplo de medição do primeiro experimento	26
Figura 10 – Tela apresentada pelo aplicativo <i>Magnetic Counter</i>	28
Figura 11 – Circuito com <i>encoder</i> E3856G5	30
Figura 12 – Circuito utilizado para medir o momento de inércia	31
Figura 13 – Resposta ao degrau de 2v para 3V do motor MA01 polarizado diretamente	34
Figura 14 – Resposta ao degrau de 2v para 3V do motor MA02 polarizado diretamente	34
Figura 15 – Visão geral do modelo 3D da bancada	35
Figura 16 – Vista frontal do modelo 3D da bancada	36
Figura 17 – Vista lateral do modelo 3D da bancada	36
Figura 18 – Visão geral da bancada	37
Figura 19 – Conexão entre o parafuso e o eixo do motor	37
Figura 20 – Módulo <i>encoder</i> com capacitor de filtro	38
Figura 21 – Circuito de controle de tensão	39
Figura 22 – Circuito de medições dos motores	40
Figura 23 – Curva de calibração do sensor de tensão	42
Figura 24 – Curva de calibração do sensor de corrente	42
Figura 25 – Fluxograma da função responsável por medir a zona morta	47
Figura 26 – Fluxograma da função responsável por medir a resistência da armadura	48
Figura 27 – Fluxograma da função responsável por medir o fator de proporcionalidade e o coeficiente de atrito viscoso	50
Figura 28 – Gráfico de $B(\omega)$ para o motor MA01	52
Figura 29 – Gráfico de $B(\omega)$ para o motor MA02	52
Figura 30 – Fluxograma da função responsável por medir a resposta ao degrau . . .	53
Figura 31 – Correntes medida pela bancada e simulada	55
Figura 32 – Velocidades medida pela bancada e simulada	55
Figura 33 – Valores medidos e simulados com os parâmetros encontrados pela bancada para o motor MA01	56
Figura 34 – Valores medidos e simulados com os parâmetros encontrados pela bancada para o motor MA02	57

Figura 35 – Diagrama controlador proporcional	59
Figura 36 – Resposta ao degrau da simulação do controlador proporcional	60
Figura 37 – Resposta ao degrau do controlador proporcional implementado no Arduino	60
Figura 38 – Resposta ao degrau da simulação da linearização por realimentação . .	63
Figura 39 – Resposta ao degrau da linearização por realimentação implementada no Arduino	64

Lista de tabelas

Tabela 1 – Dados do motor MA01 no primeiro experimento	27
Tabela 2 – Dados do motor MA02 no primeiro experimento	27
Tabela 3 – Dados do motor MA01 no segundo experimento com polarização direta	29
Tabela 4 – Dados do motor MA01 no segundo experimento com polarização reversa	29
Tabela 5 – Dados do motor MA02 no segundo experimento com polarização direta	29
Tabela 6 – Dados do motor MA02 no segundo experimento com polarização reversa	29
Tabela 7 – Dados do motor MA01 no terceiro experimento com polarização direta	31
Tabela 8 – Dados do motor MA01 no terceiro experimento com polarização reversa	32
Tabela 9 – Dados do motor MA02 no terceiro experimento com polarização direta	32
Tabela 10 – Dados do motor MA02 no terceiro experimento com polarização reversa	32
Tabela 11 – Parâmetros de referência dos motores MA01 e MA02	33
Tabela 12 – Tabela de calibração do sensor de tensão	41
Tabela 13 – Tabela de calibração do sensor de corrente	43
Tabela 14 – Tabela de funções da bancada	45
Tabela 15 – Parâmetros estimados dos motores MA01 e MA02	51
Tabela 16 – Parâmetros estimados dos motores MA01 e MA02	55

Lista de abreviaturas e siglas

CA	Corrente alternada
CC	Corrente contínua
DA	Digital para analógico
MA01	Motor 1 do modelo A
MA02	Motor 2 do modelo A
SQE	Soma quadrática do erro

Lista de símbolos

Alfabeto Grego

α	Polo em malha fechada com controlador não linear
ε	Erro no formato matricial
$\Theta(t)$	Deslocamento angular
θ	Parâmetros
$\hat{\theta}$	Parâmetros estimados
τ	Constante de tempo
$\varphi(k)$	Vetor de variáveis medidas
ϕ	Campo magnético
Φ	Variáveis medidas no formato matricial
$\omega(t)$	velocidade angular
$\tilde{\omega}$	Erro da velocidade
ω_d	Velocidade angular desejada
$\Omega(s)$	Velocidade angular no domínio frequência

Alfabeto Latino

a_n	Parâmetros que multiplicam a saída
$b(x)$	Função não linear dos estados
b_n	Parâmetros que multiplicam a entrada
B	Coeficiente de atrito viscoso
e	Erro
E_N	Representação do ruído no formato matricial
$f(x)$	Equação em função dos estados x
\dot{f}	Derivada da função
$G_p(t)$	Controlador proporcional
$i(t)$	Corrente do motor

J	Momento de inércia
k	Variável de amostras no tempo discreto
k_{dc}	Ganho do sistema em malha aberta
k_{dc1}	Ganho do amplificador não inversor
k_{dc2}	Ganho do amplificador subtrator
k_n	Constantes da lei de controle não linear
K_f	Constante de força contra-eletromotriz
K_t	Constante de torque do motor
K	Fator de proporcionalidade
$K_r(k)$	Matriz de ganho residual
K_p	Ganho proporcional
l_{zm}	Limite de zona morta
L	Indutância da armadura
n_a	Número de parâmetros a
n_b	Número de parâmetros b
$P(k)$	Matriz de pesos
$r(k)$	Representação do ruído
R_n	Resistores do circuito
R	Resistência da armadura
$R_{calculada}$	Resistência calculada
R_s	Resistor <i>shunt</i>
s	Variável de Laplace
ts	Tempo de amostragem
$T(t)$	Torque mecânico gerado pelo motor
$T_c(t)$	Torque mecânico gerado pela carga no rotor do motor
$u(t)$	Tensão aplicada nos terminais do motor
$u_{in}(t)$	Tensão aplicada nos terminais do motor considerando a zona morta

v	Lei de controle não linear
V	Tensão
$V_a(t)$	Tensão induzida na armadura
$V_{R_s}(t)$	Tensão do resistor <i>shunt</i>
V_m	Tensão do motor
V_{min}	Zona morta
$x^{(n)}$	Vetor de estados
$x_d^{(n)}$	Valor desejado para os estados
y	Saída
\hat{y}	Saída estimada
Y_N	Saída no formato matricial
Y_f	Valor final da saída
Y_i	Valor inicial da saída
$Y_{63\%}$	Valor correspondente a 63,2% do valor final da saída

Sumário

	Lista de ilustrações	4
1	INTRODUÇÃO	13
2	MODELAMENTO MATEMÁTICO DE UM MOTOR CC E IDENTIFICAÇÃO DE PARÂMETROS	17
2.1	Modelo matemático do motor de corrente contínua	17
2.1.1	Zona morta	20
2.2	Identificação dos parâmetros de um motor CC	20
2.2.1	Resistência da armadura R	22
2.2.2	Indutância da armadura L	23
2.2.3	Fator de proporcionalidade K	24
2.2.4	Coeficiente de atrito viscoso do motor B	24
2.2.5	Momento de inércia J	24
2.3	Obtenção de parâmetros dos motores CC utilizados	25
2.3.1	Experimento para identificação da resistência e da indutância da armadura	25
2.3.2	Experimento para identificação do coeficiente de atrito viscoso e do fator de proporcionalidade	28
2.3.3	Experimento para identificação do momento de inércia do rotor	30
2.3.4	Parâmetros obtidos	33
2.4	Simulando do modelo do motor cc no Scilab	33
3	BANCADA EXPERIMENTAL PARA MODELAMENTO E IDENTIFICAÇÃO DE PARÂMETROS DE MOTORES CC	35
3.1	Estrutura mecânica	35
3.2	Hardware utilizado	38
3.3	Operação da bancada	44
3.4	Identificação de parâmetros do modelo matemático utilizando a bancada	46
3.4.1	Identificação da zona morta	46
3.4.2	Identificação da resistência da armadura	48
3.4.3	Identificação do fator de proporcionalidade e do coeficiente de atrito viscoso	49
3.4.4	Identificação da indutância da armadura e do momento de inércia do rotor	52
3.5	Parâmetros obtidos pela bancada	54
4	VALIDAÇÃO DOS PARÂMETROS OBTIDOS PELA BANCADA . . .	58
4.1	Controlador proporcional	58

4.1.1	Implementando controlador proporcional	59
4.2	Linearização por realimentação	60
4.2.1	Implementando linearização por realimentação	61
5	CONCLUSÃO	65
	REFERÊNCIAS	66
	APÊNDICES	67
	APÊNDICE A – ESQUEMÁTICO DA PLACA DE ACIONAMENTO E MEDAÇÃO DO MOTOR	68
	APÊNDICE B – LAYOUT DA PLACA DE ACIONAMENTO E ME- DIÇÃO DO MOTOR	69
	APÊNDICE C – CÓDIGO UTILIZADO NO ARDUINO	70
	APÊNDICE D – CÓDIGO UTILIZADO NO SCILAB	79
	APÊNDICE E – ALGORITMOS PARA ESTIMAÇÃO DE PARÂME- TROS	94
E.0.1	Estimador dos mínimos quadrados	94
E.0.2	Estimador dos mínimos quadrados recursivo	95
E.0.3	Parâmetros estimados	96
	APÊNDICE F – DIRETÓRIO DO GITHUB	98

1 Introdução

Em um projeto de controlador não linear no qual o modelo matemático do motor é conhecido, é possível desenvolver estratégias de controle mais eficientes e precisas. A abordagem de controle não linear é utilizada para lidar com as não linearidades presentes nos sistemas dinâmicos. Ao considerar essas não linearidades, o sistema pode operar em uma faixa mais ampla. Em contraste, um controlador linear não seria capaz de lidar adequadamente com as não linearidades do sistema, resultando em um desempenho ruim ou instabilidade (SLOTINE; LI et al., 1991).

Num projeto de controlador baseado em modelo, é importante que os parâmetros do modelo matemático do sistema sejam conhecidos, uma vez que isso possibilita a simulação computacional do comportamento do sistema e serve como base para o projeto do controlador. Entre os parâmetros do modelo linear do motor de corrente contínua, observa-se a resistência da armadura, indutância da armadura, fator de proporcionalidade, coeficiente de atrito viscoso e o momento de inércia, os quais influenciam diretamente o comportamento dinâmico e em estado estacionário do motor.

Os motores de corrente contínua possuem ampla aplicação em sistemas automatizados e robóticos devido ao alto torque que é proporcionado e à facilidade de controle de acionamento. Além disso podemos, facilmente, encontrar no mercado os mais diversos tipos de circuitos de potência para acionamento de motores CC e também sensores para os mesmos. Para aplicações que exigem precisão em termos de posicionamento e velocidade, existem motores de corrente contínua de custo elevado, que são construídos de forma metódica e são acompanhados de informações detalhadas sobre suas características elétricas e mecânicas fornecidas pelos fabricantes. Um exemplo desses motores é o 0615N, fabricado pela Faulhaber. Ao analisar a ficha técnica do equipamento, é possível observar que uma ampla variedade de características do motor é fornecida pelo fabricante, conforme ilustrado na Figura 1.

Existem opções de motores CC mais acessíveis, como os motores elétricos responsáveis pelo acionamento dos vidros laterais dos carros. Esses motores são amplamente utilizados em diversos projetos devido à sua construção, ao custo baixo, por possuir um sistema de parafuso de rosca sem-fim e coroa com uma relação de transmissão que reduz a velocidade do motor em muitas vezes proporcionando torque elevado. No comércio, são chamados de "motores genéricos" de acionamento de vidros de veículos de forma a substituir uma grande quantidade de modelos de motores originais de diversas marcas de veículos, que os torna muito úteis. É possível encontrar algumas especificações técnicas para alguns fabricantes desses motores. No entanto, os dados disponíveis geralmente são incompletos e é difícil garantir a confiabilidade das informações obtidas, devido à dificuldade em identificar o fabricante específico do motor utilizado. A Figura 2 apresenta algumas características

Figura 1 – Tabela característica do motor 0615N

Series 0615 ... S		0615 N	1,5 S	003 S	4,5 S	
Values at 22°C and nominal voltage						
1 Nominal voltage	U_N	1,5	3	4,5	V	
2 Terminal resistance	R	3,9	16,2	37,7	Ω	
3 Efficiency, max.	η_{max}	52	50	48	%	
4 No-load speed	n_0	19 100	20 200	20 000	min^{-1}	
5 No-load current, typ. (with shaft ø 0,8 mm)	I_0	0,03	0,016	0,012	A	
6 Stall torque	M_H	0,24	0,22	0,21	mNm	
7 Friction torque	M_F	0,02	0,02	0,02	mNm	
8 Speed constant	k_n	13 840	7 346	4 872	min^{-1}/V	
9 Back-EMF constant	k_E	0,072	0,136	0,205	mV/min^{-1}	
10 Torque constant	K_M	0,69	1,3	1,96	mNm/A	
11 Current constant	k_I	1,449	0,769	0,51	A/mNm	
12 Slope of n-M curve	$\Delta n / \Delta M$	78 224	91 538	93 713	min^{-1}/mNm	
13 Rotor inductance	L	12	39	95	μH	
14 Mechanical time constant	τ_m	8	10	10	ms	
15 Rotor inertia	J	0,01	0,01	0,01	gcm^2	
16 Angular acceleration	α_{max}	244	221	213	$\cdot 10^3 rad/s^2$	
17 Thermal resistance	R_{th1} / R_{th2}	35 / 76			K/W	
18 Thermal time constant	τ_{w1} / τ_{w2}	2,6 / 110			s	
19 Operating temperature range:						
- motor		-30 ... +85 (optional version	-30 ... +125)		$^{\circ}C$	
- winding, max. permissible		+85 (optional version	+125)		$^{\circ}C$	
20 Shaft bearings		sintered bearings				
21 Shaft load max.:						
- with shaft diameter		0,8			mm	
- radial at 3 000 min ⁻¹ (1,5 mm from bearing)		0,5			N	
- axial at 3 000 min ⁻¹		0,1			N	
- axial at standstill		20			N	
22 Shaft play:						
- radial	\leq	0,03			mm	
- axial	\leq	0,15			mm	
23 Housing material		steel, black coated				
24 Mass		2			g	
25 Direction of rotation		clockwise, viewed from the front face				
26 Speed up to	n_{max}	24 000			min^{-1}	
27 Number of pole pairs		1				
28 Magnet material		NdFeB				

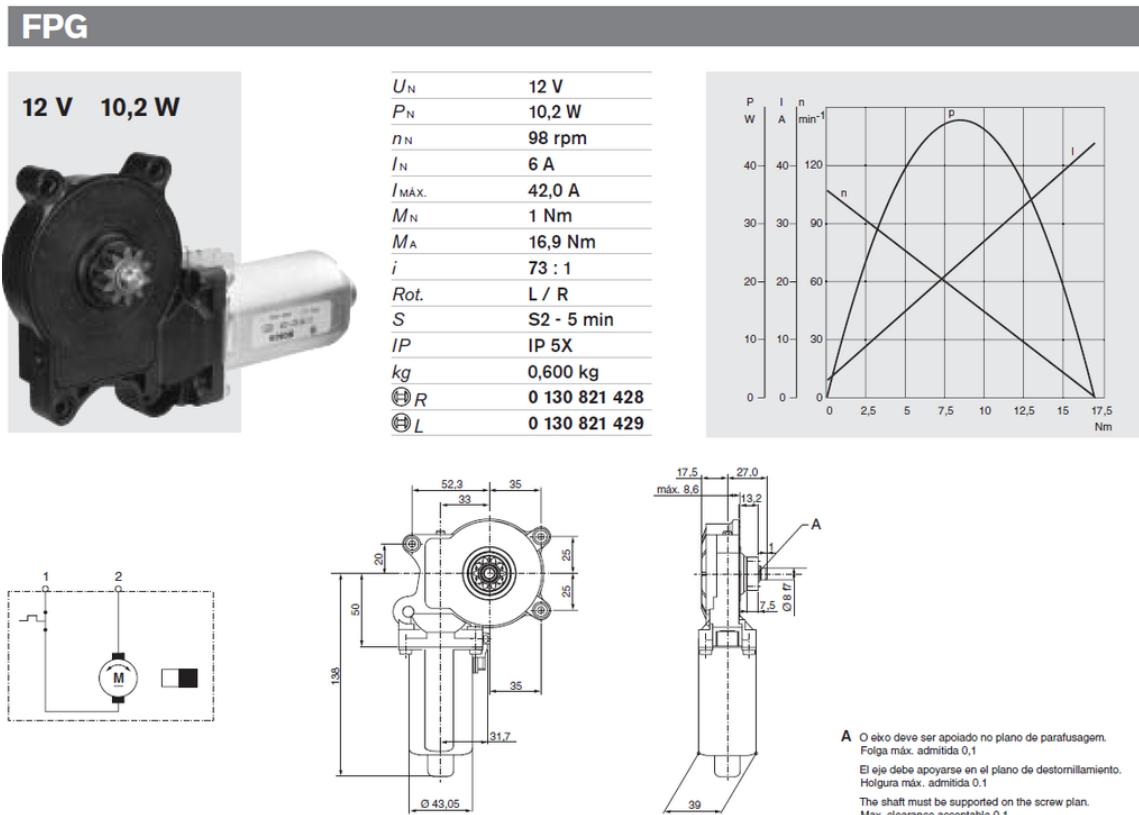
Fonte: (FAULHABER, 2023)

encontradas para um desses motores.

É igualmente importante considerar os motores de kit de Arduino. Essa opção é amplamente utilizada devido à sua acessibilidade, baixo custo e operação com tensões baixas (inferiores a 5V), além do baixo consumo de corrente, são muito utilizados em pequenos projetos de automação que utilizam o controlador Arduino. Isso possibilita a utilização de circuitos e componentes de potência mais simples e acessíveis. No entanto, é importante ressaltar que esses motores geralmente possuem uma construção menos precisa em comparação com opções mais avançadas. Encontrar informações detalhadas sobre esses motores pode ser desafiador e, muitas vezes, é difícil identificar o fabricante específico. A Figura 3 apresenta alguns detalhes disponíveis na loja MakerHero (anteriormente conhecida como FilipeFLop que também disponibilizava muitos artigos sobre projetos com Arduinos com linguagem acessível para iniciantes na área da automação).

Observa-se que os dados fornecidos na Figura 3 são bastante básicos e superficiais. No entanto, é importante destacar que esse tipo de motor é amplamente utilizado em aplicações que não exigem grande confiabilidade de operação e em pequenos projetos de sistemas automáticos. Para melhorar o desempenho e a aplicabilidade desse motor, é possível recorrer a análises computacionais e simulações para obter um melhor entendimento do seu comportamento. No entanto, para realizar essas análises, é necessário ter conhecimento dos parâmetros do modelo matemático específico do motor. Essa informação permitirá

Figura 2 – Tabela característica do motor de vidro elétrico



Fonte: (NOVA, 2012)

Figura 3 – Especificações do motor de kit de Arduino

Especificações:

- Tensão de Operação: 3-6VDC
- Torque: 0,35 Kgf/cm (3V) e 0,80 Kgf/cm (6V)
- Sistema magnético com anti-interferência
- Engrenagem com eixo duplo
- Redução: 1:48
- Corrente sem carga: $\leq 200\text{mA} @ 6\text{V}, \leq 150\text{mA} @ 3\text{V}$
- Velocidade sem carga: $200 \pm 10\%\text{RPM} @ 6\text{V}, 90 \pm 10\%\text{RPM} @ 3\text{V}$
- Dimensões: 70 x 37 x 23mm



Fonte: (MEKERHERO,)

uma abordagem mais precisa e eficiente ao realizar as simulações e análises necessárias. Diante do discutido, decidiu-se utilizar nesse trabalho os motores de kit de Arduino por serem mais baratos, facilmente adquiridos no mercado e mais fáceis de serem utilizados.

Para aplicações de alto desempenho e compromisso com pequenas tolerâncias a erros de posicionamento e velocidade, existem, por exemplo, os servoconversores WEG SCA 05 e SCA 06. Esses equipamentos são capazes de controlar motores WEG de ímã permanente, vêm equipados com drive de potência e sensores capazes de medir posição e velocidade. Esse servoconversor é capaz de autoajustar ganhos para controle em malha fechada, com

base nos parâmetros do motor configurados no dispositivo (WEG, 2006).

Considerando os fatos apresentados, a importância de conhecer os parâmetros do modelo matemático de um motor de corrente contínua para projetos de controladores não lineares e a dificuldade de obter esses valores para motores de baixo custo, decidiu-se desenvolver uma bancada com a qual se possa realizar essa identificação. Foi desenvolvida uma bancada automatizada composta por *hardware* e algoritmo de controle, que permitirá aos usuários obter de forma simples e rápida os parâmetros do modelo para motores de baixo custo, a fim de reproduzir seus comportamentos em simulação, além de permitir projetos de controladores baseados em modelos matemáticos. É desejado que o dispositivo seja facilmente replicável. Foi então desenvolvido utilizando componentes comuns e de fácil acesso. A validação do modelo matemático será apresentada por meio da implementação de um controlador de velocidade em malha fechada, e os resultados da simulação serão comparados com a prática.

Esta monografia está dividida em seis capítulos. No Capítulo 1, uma introdução é apresentada, fornecendo uma visão geral dos tópicos abordados neste trabalho.

No Capítulo 2, é apresentado o modelo matemático do motor de corrente contínua, juntamente com a explicação de como identificar seus parâmetros. Além disso, são fornecidos os parâmetros medidos dos motores utilizados neste estudo, bem como a comparação entre as respostas ao degrau, real e simulada.

No Capítulo 3, será apresentada uma descrição detalhada da bancada estimadora de parâmetros, englobando a estrutura mecânica, o *hardware* e o *firmware* do Arduino. Serão fornecidos todos os dados de calibração dos sensores, incluindo medições e erros obtidos para cada um deles.

No Capítulo 4, será apresentado o funcionamento do algoritmo responsável por operar a bancada, detalhando todo o processo realizado para obter os parâmetros de forma automatizada. Será dada uma descrição abrangente do funcionamento do algoritmo estimador de parâmetros utilizado pela bancada.

No Capítulo 5, serão apresentados os parâmetros obtidos automaticamente pela bancada, juntamente com a validação desses valores. Para isso, serão implementados dois controladores de velocidade: um proporcional e outro não linear. Neste capítulo, serão apresentados os resultados das simulações e das implementações práticas realizadas.

No Capítulo 6, a conclusão do trabalho será apresentada, discutindo-se os principais resultados obtidos e suas implicações. Será feita uma análise crítica dos objetivos alcançados e das limitações encontradas durante o desenvolvimento da pesquisa. Além disso, serão destacadas as principais contribuições do estudo para trabalhos futuros.

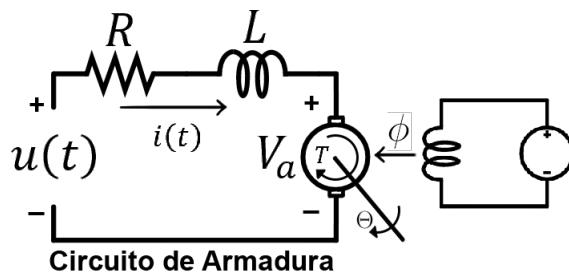
2 Modelamento matemático de um motor CC e identificação de parâmetros

Este projeto de fim de curso tem como objetivo o desenvolvimento de uma bancada para identificar automaticamente os parâmetros de modelo matemático de motores de corrente contínua de baixo custo, que são disponíveis no mercado e que normalmente não apresentam dados suficientes para que um projeto de controlador seja produzido com base em modelo matemático. Assim, apresenta-se nesse capítulo um modelo matemático de motor CC, a identificação dos parâmetros do modelo, e a estratégia experimental de identificação.

2.1 Modelo matemático do motor de corrente contínua

Para compreender o funcionamento do modelo matemático do motor de corrente contínua (CC), é necessário compreender que existem duas equações principais: uma para o subsistema elétrico e outra para o sub sistema mecânico. A equação do subsistema elétrico é obtida a partir da análise do modelo elétrico da armadura do motor, conforme mostrado na Figura 4. Ao aplicar a Lei de Kirchhoff das tensões no circuito da armadura (malha da esquerda), o modelo matemático do circuito é obtido como mostrado na equação (2.1)(OGATA, 2003).

Figura 4 – Modelo de motor CC



Fonte: (OGATA, 2003)

Considerando a Lei de Kirchhoff das tensões no circuito da armadura, tem-se a equação (2.1).

$$u(t) = V_R(t) + V_L(t) + V_a(t) \quad (2.1)$$

Substituindo $V_R(t) = Ri(t)$ e $V_L(t) = L \frac{di(t)}{dt}$ na equação (2.1), obtém-se

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + V_a(t) \quad (2.2)$$

Na equação (2.2), tem-se:

$u(t)$ é a tensão aplicada nos terminais do motor;

R é a resistência da armadura;

L é a indutância da armadura;

$i(t)$ é corrente consumida pelo motor;

$V_a(t)$ é a tensão induzida na armadura quando o rotor está girando. No caso de motores de imãs permanentes, o fluxo ϕ é mantido constante, a tensão $V_a(t)$ é diretamente proporcional à velocidade do motor (OGATA, 2003). Essa relação pode ser vista na equação (2.3), onde se pode ver $V_a(t)$ em função da velocidade angular.

$$V_a(t) = K_f \frac{d\Theta}{dt} = K_f \omega \quad (2.3)$$

Na equação (2.3), tem-se:

K_f é a constante de força contra-eletromotriz do motor;

$\Theta(t)$ é o deslocamento angular do motor;

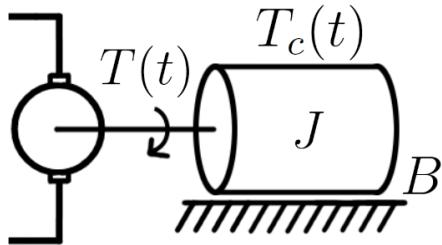
$\omega(t)$ é a velocidade angular do motor.

Substituindo a equação (2.3) na equação (2.2), obtém-se a equação (2.4), que descreve o comportamento elétrico do motor em termos da tensão de entrada do motor, corrente consumida e velocidade (OGATA, 2003).

$$\frac{di(t)}{dt} = \frac{1}{L}u(t) - \frac{R}{L}i(t) - \frac{K_f}{L}\omega(t) \quad (2.4)$$

A equação do subsistema mecânico é obtida a partir do torque exercido pelo motor, que é aplicado na inércia, no atrito viscoso e em uma carga $T_c(t)$, apresentado em função do deslocamento na equação (2.5)(OGATA, 2003). Na figura 5 pod-se visualizar um reprresentação do subsistema.

Figura 5 – Modelo de motor CC



Fonte: (OGATA, 2003)

$$T(t) = J \frac{d^2\Theta(t)}{dt^2} + B \frac{d\Theta(t)}{dt} + T_c(t) \quad (2.5)$$

Substituindo as derivadas da posição angular pela velocidade e sua derivada obtém-se a equação (2.6).

$$T(t) = J \frac{d\omega(t)}{dt} + B\omega(t) + T_c(t) \quad (2.6)$$

O torque gerado pelo motor $T(t)$ será diretamente proporcional a corrente na armadura, portanto, pode-se afirmar que:

$$T(t) = K_t i(t) \quad (2.7)$$

Com o motor a vazio, sem carga no rotor, deve-se considerar $T_c(t) = 0$. A equação do subsistema mecânico fica da seguinte forma:

$$\frac{d\omega(t)}{dt} = \frac{K_t}{J} i(t) - \frac{B}{J} \omega(t) \quad (2.8)$$

Na equação (2.8), tem-se:

J é o momento de inércia da armadura e da carga;

B é o coeficiente de atrito viscoso do motor;

$T(t)$ é o torque mecânico gerado pelo motor;

K_t é a constante de torque do motor.

No caso dos motores de corrente contínua, os valores das constantes de torque (K_t) e força contra-eletromotriz (K_f) são apresentados como bastante semelhantes. Por esse motivo, para simplificar a análise, eles serão considerados numericamente iguais (ROSÁRIO, 2005). Serão referidos apenas como K , que será chamado de fator de proporcionalidade, como nomeado por Geromel e Palhares (2004). Desse modo, as equações do modelo podem ser expressas como mostrado nas equações (2.9) e (2.10).

$$\frac{di(t)}{dt} = \frac{1}{L} u(t) - \frac{R}{L} i(t) - \frac{K}{L} \omega(t) \quad (2.9)$$

$$\frac{d\omega(t)}{dt} = \frac{K}{J} i(t) - \frac{B}{J} \omega(t) \quad (2.10)$$

Os parâmetros do modelo matemático linear do motor CC representado pelas equações (2.9) e (2.10), são as constantes J , B , K , L e R , que dependem das características construtivas do motor e terão valores diferentes quando os motores CC possuem características operacionais diferentes. A velocidade angular ω e a corrente i serão variáveis de estado do modelo, e poderão ser medidas durante o funcionamento do motor. Por fim, a tensão aplicada nos terminais do motor será referida como $u(t)$.

É importante destacar que nesta seção foi descrito o modelo linear do motor. No entanto, é necessário considerar que o motor apresenta características não lineares, como uma zona morta causada pelo atrito estático no eixo. Pode-se considerar o efeito do atrito nos mancais do motor e nos componentes de transmissão de movimento, se motorredutor,

causando um torque de oposição ao acionamento do motor quando uma tensão é aplicada em seus terminais. É necessário uma corrente de armadura mínima para que se produza um torque que compense o efeito do atrito. Tal comportamento pode ser interpretado como uma zona morta. Além disso, é importante levar em conta que o coeficiente de atrito viscoso (B) constante pode não representar o comportamento do atrito do motor, que é não linear. Pode-se então considerar que o coeficiente de atrito viscoso assume valores diferentes quando o motor opera em velocidades diferentes.

2.1.1 Zona morta

Em diversos modelos de motores, o seu rotor permanece imóvel até que a tensão aplicada em seus terminais seja superior a um determinado valor. Esse efeito é chamado de zona morta (SLOTINE; LI et al., 1991). Esse valor mínimo de tensão é chamado de limite de zona morta (l_{zm}). É possível modelar a zona morta de um motor utilizando a função descontínua mostrada na equação (2.11), onde $u(t)$ é a tensão aplicada na entrada do sistema e $u_{in}(t)$ é a tensão que será de fato aplicada no motor. Isso significa que a tensão no motor será zero se a tensão aplicada nos seus terminais for inferior a l_{zm} , por outro lado será igual ao valor aplicado na entrada se este for maior ou igual a l_{zm} .

$$u_{in}(t) = \begin{cases} 0, & |u(t)| < l_{zm} \\ u(t) - l_{zm}, & u(t) \geq l_{zm} \\ u(t) + l_{zm}, & u(t) \leq -l_{zm} \end{cases} \quad (2.11)$$

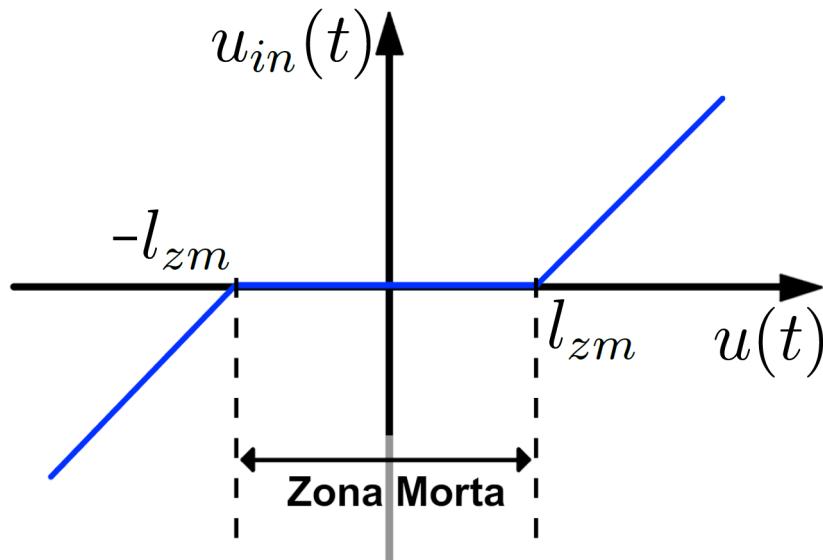
A zona morta do motor é representada pela Figura 6, onde a velocidade do rotor é relacionada à tensão aplicada na entrada do sistema. Conforme mencionado anteriormente, a ocorrência da zona morta é causada pelo atrito estático presente no eixo do rotor, assim como pela transmissão de movimento através de componentes mecânicos conectados ao rotor. É necessário que um torque mínimo seja fornecido pelo motor para que o rotor seja iniciado a partir do repouso (SLOTINE; LI et al., 1991).

O efeito mais comum decorrente da presença de uma zona morta em sistemas de controle é a redução da precisão na saída. Além disso, a zona morta pode ocasionar instabilidade devido à falta de resposta associada a ela. No entanto, em determinadas situações, é possível que a zona morta contribua para a diminuição das oscilações na saída ou até mesmo estabilize seu valor (SLOTINE; LI et al., 1991).

2.2 Identificação dos parâmetros de um motor CC

Para utilizar o modelo do motor representado pelas equações (2.9) e (2.10) em simulações computacionais ou projetos de controladores baseados em modelos matemáticos, é necessário identificar os seus parâmetros elétricos, como a resistência da armadura (R) e a

Figura 6 – Representação da zona morta



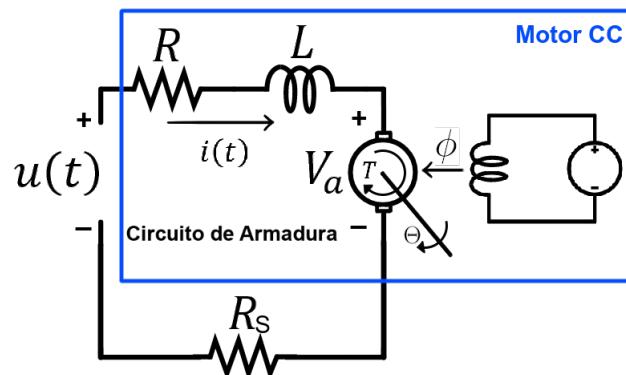
Fonte: Adaptado de Slotine, Li et al. (1991)

indutância da armadura (L), e os parâmetros mecânicos, como a inércia (J) e o coeficiente de atrito viscoso (B), e o fator de proporcionalidade (K).

A fim de realizar medições de corrente, é comum que um resistor *shunt* (R_s) seja utilizado em série com o motor. Dessa forma, a corrente que atravessa tanto o resistor quanto o motor, pode ser calculada medindo-se a tensão sobre o resistor *shunt*. No entanto, ao calcular a resistência da armadura, é importante que a resistência adicional introduzida pelo resistor *shunt* seja levada em consideração. Assim, a resistência calculada ($R_{calculada}$) é igual à soma de R_s e R , como mostrado na equação (2.12) (GEROMEL; PALHARES, 2004).

$$R_{calculada} = R_s + R \quad (2.12)$$

Figura 7 – Ligação do motor para medição de parâmetros



Fonte: O autor

Deve-se considerar a corrente do motor em duas situações: com o rotor travado e

girando. Isso é importante porque essa corrente passará completamente pelo resistor *shunt* (R_s). É importante lembrar que quanto menor for a resistência do resistor *shunt*, menos interferência ele terá na grandeza a ser medida. Portanto, é necessário conhecer os valores de corrente para escolher a potência adequada para o resistor *shunt*, garantindo que ele suporte a corrente máxima que será aplicada durante os experimentos.

Supondo que a corrente máxima a ser medida seja de 1A e o resistor escolhido seja de 1Ω , sua potência deve ser superior a 1 W. Neste trabalho, utilizou-se um resistor de filme metálico de 1Ω e 2 W. Esse valor foi escolhido por possuir potência mais do que suficiente para a realização do experimento, ter tolerância de 1% e estar disponível.

É importante reconhecer que, durante o processo de fabricação de componentes, não é possível garantir com 100% de certeza que todas as peças atendam às especificações. Podem ocorrer erros de produção que devem estar dentro das especificações de tolerância estabelecidas. A medição da resistência elétrica dos resistores utilizados é necessária para obter uma maior confiabilidade na medição indireta, como no caso da medição indireta da corrente elétrica que circula na armadura do motor.

É necessário que uma medição direta de resistência seja realizada. No entanto, um multímetro comum pode não ser capaz de medir seu valor com a exatidão necessária, uma vez que R_s é muito baixo, próximo de 1Ω . Para resolver esse problema, é necessário que uma tensão seja aplicada ao resistor, a corrente que passa por ele seja medida com um amperímetro e, em seguida, sua resistência seja calculada. Ao realizar a medição da resistência de R_s recomenda-se utilizar tensões baixas para evitar exceder sua potência máxima e causar danos ao componente (GEROMEL; PALHARES, 2004).

2.2.1 Resistência da armadura R

Com base na equação (2.13), observa-se que a resistência da armadura pode ser determinada quando o eixo do motor está travado ($\omega = 0$), a corrente elétrica entra em regime permanente ($\frac{di(t)}{dt} = 0$) e obtém-se a equação (2.14), onde $R_{calculada}$ é definido pela equação (2.12).

$$u(t) = R_{calculada}i(t) + L \left(\frac{di(t)}{dt} \right) + K\omega \quad (2.13)$$

Com base na equação (2.14), o valor de $R_{calculada}$ pode ser determinado quando o eixo do motor está travado, utilizando o mesmo procedimento para medir o valor de R_s .

$$R_{calculada} = \frac{u(\infty)}{i(\infty)} \quad (2.14)$$

A medição da tensão aplicada u pode ser realizada facilmente, enquanto a corrente i pode ser obtida através da medição da tensão sobre o resistor R_s (GEROMEL; PALHARES, 2004).

2.2.2 Indutância da armadura L

Para obter a indutância da armadura, deve-se analisar a resposta ao degrau da corrente na armadura. Para isso, aplica-se um degrau de tensão nos terminais do motor com seu eixo travado. O transitório da corrente na armadura pode ser detectado e medido usando um osciloscópio, medindo a tensão sobre o resistor *shunt* R_s . Com base na equação (2.13), com $\omega = 0$, obtém-se a equação (2.15), que representa o comportamento elétrico do circuito da armadura do motor.

$$u(t) = R_{calculada}i(t) + L\left(\frac{di(t)}{dt}\right) \quad (2.15)$$

Aplicando-se transformada de Laplace na equação (2.15), na qual substitui-se $R_{calculada}$ por $R + R_s$ e $i(t)$ por $\frac{V_{R_s}}{R_s}$. Obtendo-se a equação (2.16), na qual $U(s)$ representa a tensão aplicada no motor e V_{R_s} representa a tensão sobre o resistor:

$$U(s) = (R + R_s)\frac{V_{R_s}(s)}{R_s} + sL\frac{V_{R_s}(s)}{R_s} \quad (2.16)$$

Trata-se de um sistema de primeira ordem, definido pela equação (2.17), na qual o valor em regime permanente da tensão sobre R_s é representado por k_{dc} e a constante de tempo é representada por τ , a entrada é a tensão $U(s)$. A constante de tempo é o tempo que a saída, a tensão sobre R_s , leva para atingir 63,2% do valor final em estado estacionário (OGATA, 2003).

$$\frac{V_{R_s}(s)}{U(s)} = H(s) = \frac{k_{dc}}{\tau s + 1} \quad (2.17)$$

Para calcular a constante de tempo τ , é necessário isolar a expressão $\frac{V_{R_s}(s)}{U(s)}$ na equação (2.16) e deixa-lá no formato similar ao mostrado pela equação (2.17). Desta forma, é possível obter a função de transferência do circuito, apresentada na equação (2.18).

$$\frac{V_{R_s}(s)}{U(s)} = H(s) = \frac{\frac{R_s}{R+R_s}}{\frac{L}{R_s+R}s + 1} \quad (2.18)$$

Pode-se então afirmar que τ é igual a $\frac{L}{R_s+R}$, e a equação (2.19) pode ser utilizada para calcular o valor da indutância da armadura a partir da constante de tempo τ , dado que R e R_s são conhecidos. O valor da constante de tempo pode ser medido com o auxílio do osciloscópio.

$$L = \tau(R_s + R) \quad (2.19)$$

2.2.3 Fator de proporcionalidade K

Para obter o valor de K , uma tensão constante é aplicada ao motor e aguarda-se até que o motor alcance o regime permanente. É necessário que a velocidade ($\omega(\infty)$) e a corrente ($i(\infty)$) no motor em regime permanente sejam medidas, a fim de calcular o valor do fator de proporcionalidade utilizando a equação (2.20), na qual a tensão aplicada ao motor é representada por u (GEROMEL; PALHARES, 2004).

$$K = \frac{u(\infty) - R_{calculado} \cdot i(\infty)}{\omega(\infty)} \quad (2.20)$$

A equação (2.20) é deduzida através da equação (2.13), em que $u(t) = R_{calculada}i(t) + L\left(\frac{di(t)}{dt}\right) + K\omega$. No regime permanente, $\frac{di(t)}{dt} = 0$, sendo assim possível isolar a constante K .

2.2.4 Coeficiente de atrito viscoso do motor B

Assim como no fator de proporcionalidade, o coeficiente de atrito viscoso é calculado utilizando a velocidade e a corrente em regime permanente, por meio da equação (2.21), que é deduzida da equação (2.10), na qual $Ki(t) = J\frac{d\omega(t)}{dt} + B\omega(t)$. No regime permanente, quando $\frac{d\omega(t)}{dt} = 0$, é possível isolar B na equação.

$$B = K \frac{i(\infty)}{\omega(\infty)} \quad (2.21)$$

2.2.5 Momento de inércia J

Para medir o momento de inércia do rotor, o motor é alimentado, aguarda-se até que o regime permanente seja alcançado e, em seguida, a tensão de entrada é removida. Utilizando a mesma abordagem utilizada para medir a indutância, nesse caso a redução da velocidade é analisada para determinar a constante de tempo do sistema, considerando uma desaceleração.

Quando a alimentação é removida, o circuito da armadura estará aberto, portanto, a partir desse instante a corrente na armadura será zero ($i(t) = 0$). Aplicando-se transformada de Laplace na equação $Ki(t) = J\frac{d\omega(t)}{dt} + B\omega(t)$, a seguinte relação é utilizada: $\mathcal{L}\frac{dF(t)}{dt} = sF(s) - F(0)$. Dessa forma, a equação (2.22) é obtida.

$$0 = J(s\Omega(s) - \omega(0)) + B\Omega(s) \quad (2.22)$$

$$\Omega(s) = \frac{\omega(0)}{s + \frac{B}{J}} \quad (2.23)$$

Ao isolar $\omega(s)$ na equação (2.22), obtém-se a equação da velocidade do motor durante a desaceleração, mostrado na equação (2.23), em que $\omega(0)$ representa a velocidade do motor

em regime permanente antes da remoção da alimentação. A constante de tempo $\tau = J/B$, seria o intervalo de tempo necessário para que a velocidade angular atinja 63% do valor final. No entanto, considerando que o valor final é zero, é importante observar que τ corresponde ao tempo decorrido até que o sistema atinja 37% do valor inicial. O momento de inércia pode ser calculado utilizando a equação (2.24) (GEROMEL; PALHARES, 2004).

$$J = \tau B \quad (2.24)$$

2.3 Obtenção de parâmetros dos motores CC utilizados

Para verificar o erro dos parâmetros obtidos de forma automática (pela bancada), foi necessário que os valores dos parâmetros dos motores que seriam utilizados nos testes fossem conhecidos. Para isso, decidiu-se medir esses valores com instrumentos de laboratório que apresentam erros de medição pequenos e foram informados pelos fabricantes dos instrumentos. Dessa forma, com os valores obtidos, é possível avaliar os resultados da bancada. Para se obter um número maior de amostras de medição, foram obtidos os parâmetros de dois motores (MA01 e MA02), sendo que eles foram medidos utilizando as equações apresentadas na seção anterior.

Nesta seção, serão apresentados os experimentos que foram realizados para a obtenção desses parâmetros. Um total de três experimentos, baseados em Geromel e Palhares (2004), foram conduzidos: no primeiro experimento, buscou-se a obtenção da resistência R e da indutância L da armadura; o segundo experimento teve como objetivo determinar o fator de proporcionalidade K e o coeficiente de atrito viscoso B ; e, por fim, o terceiro experimento possibilitou o cálculo do momento de inércia J .

2.3.1 Experimento para identificação da resistência e da indutância da armadura

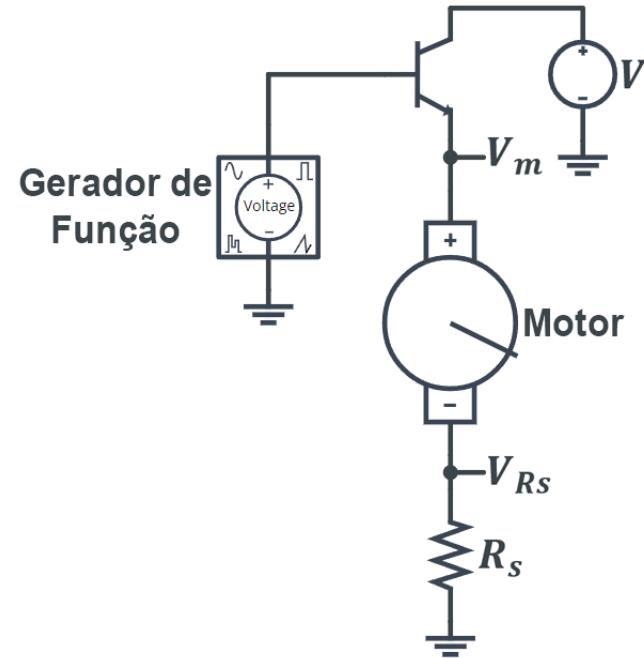
Para o primeiro experimento, o circuito apresentado na Figura 8 foi montado. Foi utilizado o gerador de função para aplicar um degrau unitário ao motor. As tensões sobre o motor e o resistor *shunt* foram medidas com o osciloscópio, sendo a segunda proporcional à corrente do motor. A fonte de tensão V, cujo valor não é relevante, pode ser de 5V, uma vez que a tensão sobre o motor será definida pelo gerador de função, pois o transistor está ligado como seguidor de emissor.

Para a realização dessas medições, é necessário que o motor esteja parado, conforme explicado nos itens 2.2.1 e 2.2.2, e para isso é preciso aplicar um degrau com amplitude baixa, portanto, a amplitude do sinal aplicado pelo gerador deve ser escolhida adequadamente. A fim de reduzir erros, as medições foram realizadas três vezes para os mesmos valores.

No osciloscópio, é preciso medir três valores: as tensões do motor e do resistor em regime permanente, juntamente com o tempo que a tensão do resistor leva para atingir

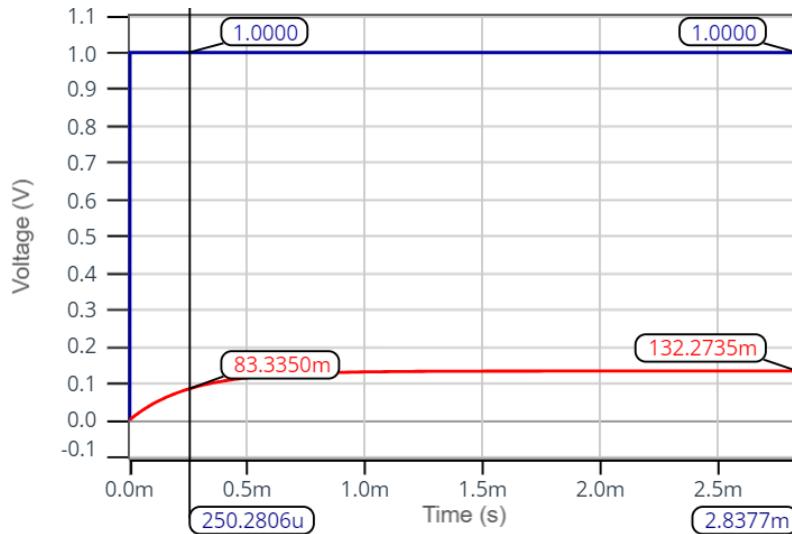
63,2% do valor de regime permanente, conforme apresentado no item 2.2.2. A Figura 9 ilustra as medições que devem ser realizadas no experimento, sendo que ela foi gerada utilizando um simulador.

Figura 8 – Circuito do primeiro experimento



Fonte: O autor

Figura 9 – Exemplo de medição do primeiro experimento



Fonte: O autor

O experimento foi realizado com os dois motores, denominados MA01 e MA02. As Tabelas 1 e 2 apresentam os dados obtidos para eles. Com base nos dados coletados, é possível calcular os valores da resistência e da indutância da armadura dos motores utilizando as equações (2.14), (2.12) e (2.19). Os valores obtidos para o motor MA01 são de $5,673\Omega$ para a resistência e $1,847mH$ para a indutância da armadura, enquanto para o

motor MA02 os valores são de $5,745\Omega$ para a resistência e $1,918mH$ para a indutância da armadura.

Tabela 1 – Dados do motor MA01 no primeiro experimento

$V_m(V)$	$V_{Rs}(V)$	$\tau(\mu s)$	$R_{motor}(\Omega)$	$L_{motor}(mH)$
0,775	0,11525	316,25	5,679	1,80
0,9	0,13425	317,5	5,659	1,80
1,105	0,1645	323,75	5,672	1,84
1,31	0,194	330	5,707	1,88
1,51	0,2255	340	5,651	1,92
Valor médio			5,673	1,847

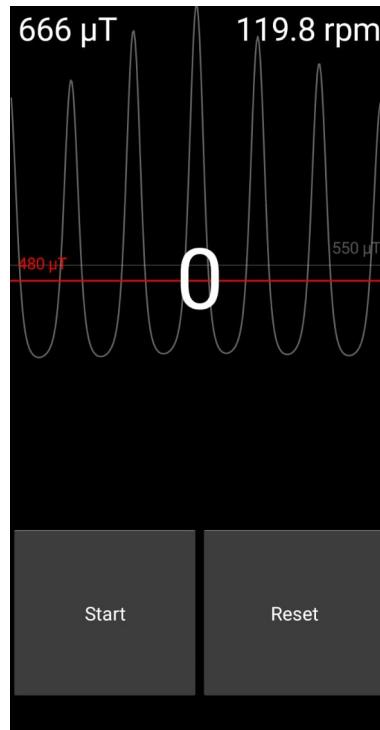
Tabela 2 – Dados do motor MA02 no primeiro experimento

$V_m(V)$	$V_{Rs}(V)$	$\tau(\mu s)$	$R_{motor}(\Omega)$	$L_{motor}(mH)$
0,78	0,113	315,0	5,842	1,840
0,91	0,134	320,0	5,745	1,838
1,13	0,167	340,0	5,711	1,942
1,34	0,197	342,5	5,748	1,969
1,53	0,228	352,5	5,680	2,002
Valor médio			5,745	1,918

2.3.2 Experimento para identificação do coeficiente de atrito viscoso e do fator de proporcionalidade

Para a realização do segundo experimento, é necessário medir a velocidade e a corrente do motor em regime permanente, tanto no sentido horário quanto no sentido anti-horário, denominados aqui como polaridade direta e reversa. Para medir a velocidade, um ímã foi acoplado à extremidade da roda ligada ao eixo de baixa rotação do motor, e utilizando o aplicativo *Magnetic Counter* para *Android*, a velocidade da roda em RPM foi medida com base nos pulsos gerados pelo ímã ao se aproximar e afastar do celular. Na Figura 10, pode-se observar a tela apresentada pelo aplicativo durante uma medição, onde a velocidade medida é exibida no canto superior direito. É importante ressaltar que a velocidade medida é a da roda, e para obter a velocidade do motor, deve-se considerar a caixa de redução, resultando em uma rotação do rotor 48 vezes maior que a da roda. Para medir a corrente, foi utilizado um multímetro conectado em série com o motor.

Figura 10 – Tela apresentada pelo aplicativo *Magnetic Counter*



Fonte: O autor

Para realizar o experimento variou-se os valores de tensão de 1,5V a 5V com incrementos de 0,5V, repetindo o processo três vezes. Os resultados apresentados nas Tabelas 3, 4, 5 e 6 são as médias dos valores obtidos por meio das medições realizadas. Nessas Tabelas, é possível observar a constante K e o coeficiente de atrito B para ambos os motores, considerando ambas as polaridades. Os cálculos foram realizados utilizando as equações (2.20) e (2.21).

Tabela 3 – Dados do motor MA01 no segundo experimento com polarização direta

$V_m(V)$	$I_m(A)$	$\omega(rad/s)$	$K(Vs)$	$B(Nms)$
1,5	0,062	206,64	5,558E-03	1,666E-06
2	0,065	291,84	5,583E-03	1,250E-06
2,5	0,069	382,56	5,506E-03	9,982E-07
3	0,072	462,48	5,600E-03	8,753E-07
3,5	0,076	548,40	5,599E-03	7,730E-07
4	0,079	637,92	5,565E-03	6,918E-07
4,5	0,083	730,92	5,516E-03	6,231E-07
5	0,086	817,44	5,518E-03	5,821E-07
Valor médio			5,556E-03	9,325E-07

Tabela 4 – Dados do motor MA01 no segundo experimento com polarização reversa

$V_m(V)$	$I_m(A)$	$\omega(rad/s)$	$K(Vs)$	$B(Nms)$
1,5	0,061	211,32	5,460E-03	1,577E-06
2	0,065	294,12	5,554E-03	1,220E-06
2,5	0,067	376,08	5,630E-03	1,010E-06
3	0,070	464,76	5,597E-03	8,466E-07
3,5	0,074	556,56	5,538E-03	7,325E-07
4	0,077	633,12	5,629E-03	6,835E-07
4,5	0,080	717,36	5,637E-03	6,322E-07
5	0,084	819,24	5,520E-03	5,678E-07
Valor médio			5,571E-03	9,086E-07

Tabela 5 – Dados do motor MA02 no segundo experimento com polarização direta

$V_m(V)$	$I_m(A)$	$\omega(rad/s)$	$K(Vs)$	$B(Nms)$
1,5	0,058	218,40	5,343E-03	1,418E-06
2	0,049	303,00	5,679E-03	9,112E-07
2,5	0,065	384,12	5,532E-03	9,401E-07
3	0,069	468,48	5,558E-03	8,180E-07
3,5	0,073	565,44	5,449E-03	7,031E-07
4	0,078	650,04	5,467E-03	6,533E-07
4,5	0,082	736,08	5,474E-03	6,092E-07
5	0,085	847,20	5,324E-03	5,356E-07
Valor médio			5,478E-03	8,236E-07

Tabela 6 – Dados do motor MA02 no segundo experimento com polarização reversa

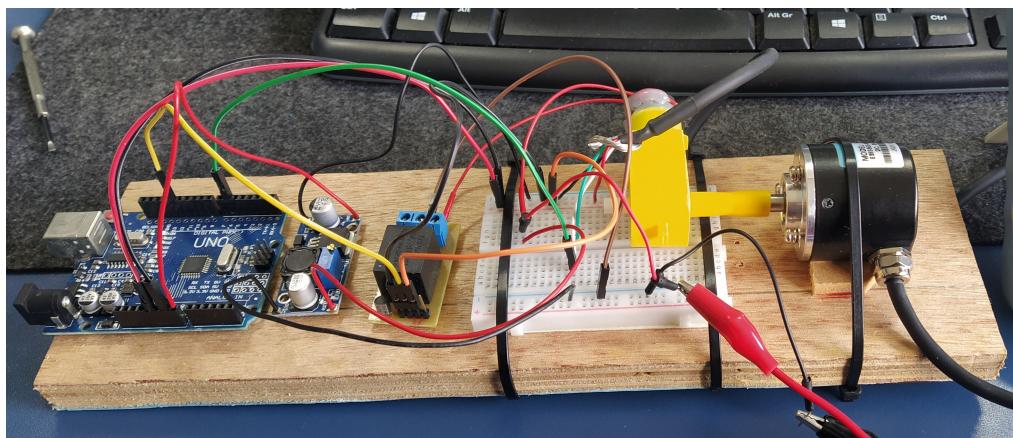
$V_m(V)$	$I_m(A)$	$\omega(rad/s)$	$K(Vs)$	$B(Nms)$
1,5	0,058	217,44	5,369E-03	1,433E-06
2	0,062	303,36	5,425E-03	1,105E-06
2,5	0,066	384,24	5,527E-03	9,447E-07
3	0,069	467,52	5,571E-03	8,224E-07
3,5	0,074	556,32	5,534E-03	7,317E-07
4	0,078	651,12	5,455E-03	6,550E-07
4,5	0,082	734,16	5,491E-03	6,117E-07
5	0,085	849,60	5,313E-03	5,306E-07
Valor médio			5,478E-03	8,236E-07

2.3.3 Experimento para identificação do momento de inércia do rotor

O terceiro experimento é considerado o mais difícil de ser realizado, uma vez que é necessário medir o tempo que leva para que o motor atinja 37% de sua velocidade de regime permanente após ser desligado (GEROMEL; PALHARES, 2004). Foi necessário empregar um Arduino e um *encoder* de 600 pulsos por volta, conforme demonstrado na Figura 11. Dessa forma, a medição da velocidade em baixas rotações, quando o motor está próximo de parar, pôde ser realizada. Para a execução do experimento, além do Arduino e do *encoder*, fez-se necessária a utilização de um módulo relé e um módulo *Buck*.

O módulo *Buck* é uma placa de circuito que possui um conversor CC-CC do tipo *Buck*, que é capaz de converter uma tensão de entrada mais alta em uma tensão de saída mais baixa. Ela também permite utilizar um *trimpot*, resistor variável, para fazer o ajuste da tensão, podendo operar como uma fonte de tensão ajustável.

Figura 11 – Circuito com *encoder* E3856G5

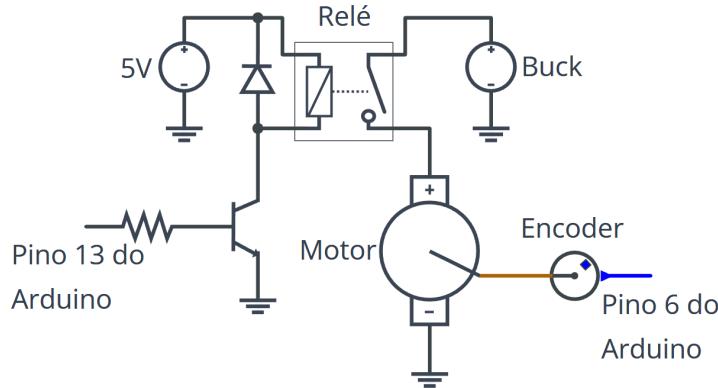


Fonte: O autor

O experimento foi conduzido da seguinte maneira: o motor foi alimentado pelo *Buck*, e o relé foi utilizado como uma chave para desligar o motor. Um relé foi escolhido devido à sua capacidade de desconectar completamente o motor do circuito, permitindo que ele gire livremente com seus terminais abertos. O *encoder* foi designado para medir a velocidade do motor ao longo de todo o experimento, permitindo que o tempo entre o desligamento do motor e a sua rotação atingir 37% do valor de regime permanente fosse medido pelo Arduino. O circuito utilizado no experimento é mostrado na Figura 12.

Antes das medições serem realizadas, a velocidade do motor em regime permanente é obtida pelo Arduino e armazenada em uma variável. Para a execução do experimento, o motor é ligado pelo Arduino por meio do pino 13 e aguarda-se que ele atinja o regime permanente. Em seguida, o motor é desligado e o tempo é contado. A contagem é encerrada quando a velocidade do motor atinge ou fica abaixo do valor limite, que corresponde a 37% do regime permanente, e esse tempo é registrado como a constante de tempo. A fim de obter resultados mais precisos, o programa repete esse procedimento sessenta vezes e calcula a média desses valores. Variando a tensão do motor de 1,5 a 5V, com incremento

Figura 12 – Circuito utilizado para medir o momento de inércia



Fonte: O autor

de 0,5V. Nas Tabelas 7, 8, 9 e 10, a seguir, os resultados obtidos para ambos os motores podem ser observados. A equação (2.24) foi utilizada para realizar os cálculos.

Tabela 7 – Dados do motor MA01 no terceiro experimento com polarização direta

$V_m(V)$	$\tau(ms)$	$B(Nms)$	$J(kgm^2)$
1,5	64,85	1,666E-06	1,081E-07
2	82,72	1,250E-06	1,034E-07
2,5	102,16	9,982E-07	1,020E-07
3	118,60	8,753E-07	1,038E-07
3,5	133,79	7,730E-07	1,034E-07
4	153,21	6,918E-07	1,060E-07
4,5	167,90	6,231E-07	1,046E-07
5	183,25	5,821E-07	1,067E-07
Valor médio			1,047E-07

Tabela 8 – Dados do motor MA01 no terceiro experimento com polarização reversa

$V_m(V)$	$\tau(ms)$	$B(Nms)$	$J(kgm^2)$
1,5	64,04	1,577E-06	1,010E-07
2	81,11	1,220E-06	9,892E-08
2,5	102,24	1,010E-06	1,032E-07
3	120,84	8,466E-07	1,023E-07
3,5	140,56	7,325E-07	1,030E-07
4	158,19	6,835E-07	1,081E-07
4,5	172,66	6,322E-07	1,092E-07
5	185,88	5,678E-07	1,055E-07
Valor médio		1,039E-07	

Tabela 9 – Dados do motor MA02 no terceiro experimento com polarização direta

$V_m(V)$	$\tau(ms)$	$B(Nms)$	$J(kgm^2)$
1,5	75,20	1,400E-06	1,053E-07
2	103,20	1,053E-06	1,087E-07
2,5	128,27	9,063E-07	1,162E-07
3	149,68	7,788E-07	1,166E-07
3,5	167,44	7,017E-07	1,175E-07
4	186,91	6,274E-07	1,173E-07
4,5	205,33	5,553E-07	1,140E-07
5	219,65	5,079E-07	1,116E-07
Valor médio		1,134E-07	

Tabela 10 – Dados do motor MA02 no terceiro experimento com polarização reversa

$V_m(V)$	$\tau(ms)$	$B(Nms)$	$J(kgm^2)$
1,5	80,28	1,418E-06	1,138E-07
2	103,52	9,112E-07	9,433E-08
2,5	127,44	9,401E-07	1,198E-07
3	148,18	8,180E-07	1,212E-07
3,5	164,44	7,031E-07	1,156E-07
4	180,52	6,533E-07	1,179E-07
4,5	203,31	6,092E-07	1,239E-07
5	222,62	5,356E-07	1,192E-07
Valor médio		1,157E-07	

2.3.4 Parâmetros obtidos

Após a realização dos três experimentos, os parâmetros dos dois motores foram obtidos e serão utilizados como referência para a validação de experimentos futuros. Os parâmetros obtidos são apresentados na Tabela 11. O próximo passo consiste em simular o comportamento dos motores utilizando o Scilab, a fim de validar os dados que serão medidos do motor em relação ao modelo de referência.

Tabela 11 – Parâmetros de referência dos motores MA01 e MA02

Parâmetro	Motor MA01		Motor MA02	
	Direta	Reversa	Direta	Reversa
$R(\Omega)$	5,673	5,673	5,745	5,745
$L(mH)$	1,847	1,847	1,918	1,918
$K(Vs)$	5,556E-03	5,571E-03	5,550E-03	5,478E-03
$B(Nms)$	9,325E-07	9,086E-07	8,163E-07	8,236E-07
$J(kgm^2)$	1,047E-07	1,039E-07	1,134E-07	1,157E-07

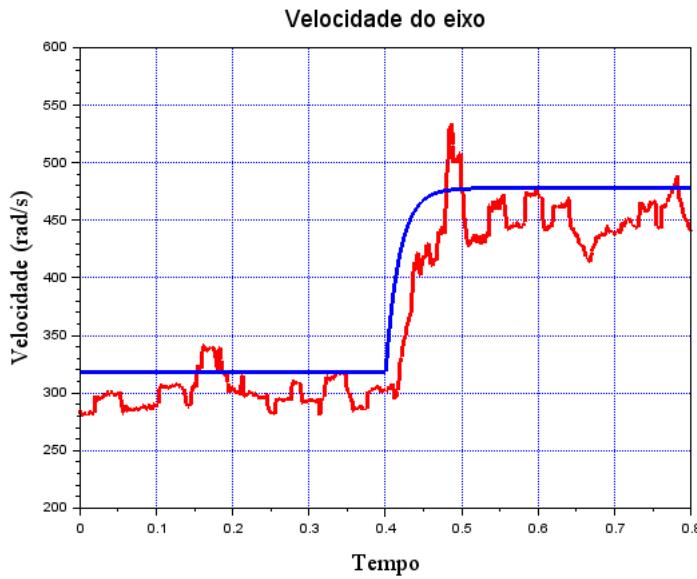
2.4 Simulando do modelo do motor cc no Scilab

Para realizar as simulações e outras atividades mais à frente descritas, utilizou-se o Scilab. Similar ao MATLAB, o Scilab é um programa utilizado para simulações numéricas de experimentos de diversos tipos, com a diferença que é gratuito. As simulações podem ser realizadas através de linhas de código, com implementação muito parecida com um programa em linguagem C, Python, entre outras. Porém, também podem ser realizadas através de diagramas de blocos, utilizando a ferramenta Xcos, similar ao simulink do MATLAB.

Utilizou-se o Scilab para simular o comportamento do motor em malha aberta, aplicando um degrau de tensão de 2V para 3V. Comparou-se os dados obtidos a partir da resposta ao degrau do motor com a simulação utilizando os parâmetros medidos. Comparando os dois resultados, pode-se verificar se os dados obtidos nos experimentos realizados para medir os parâmetros são coerentes com a realidade.

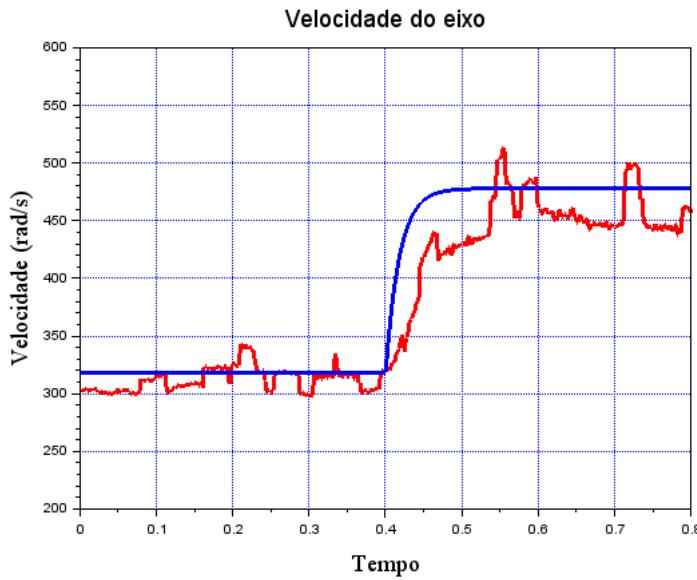
Para realizar esse experimento, utilizou-se um Arduino para aplicar um degrau de 2V para 3V no motor e enviar os dados para o Scilab. No computador, implementou-se o código para receber e organizar os dados enviados pelo microcontrolador e realizar a simulação do motor utilizando as equações (2.9) e (2.10), inseriu-se também a zona morta. Nas Figuras 13 e 14 pode-se observar os resultados, em azul a simulação e em vermelho a resposta real.

Figura 13 – Resposta ao degrau de 2v para 3V do motor MA01 polarizado diretamente



Fonte: O autor

Figura 14 – Resposta ao degrau de 2v para 3V do motor MA02 polarizado diretamente



Fonte: O autor

É perceptível que há um atraso de propagação, uma vez que a resposta do motor começa um pouco depois da simulação. Pode ser que isso seja causado pelo atraso na medição do sensor de velocidade, porém, como está em malha aberta, pode também ser causado pela zona morta. Também é notada uma pequena diferença no transitório, em que o modelo simulado é ligeiramente mais rápido que o real. Isso pode ser atribuído à omissão das não linearidades do motor no modelo e possíveis diferenças nos parâmetros com a realidade. No entanto, mesmo com essas pequenas diferenças, pode-se supor que o modelo é coerente com o comportamento do motor.

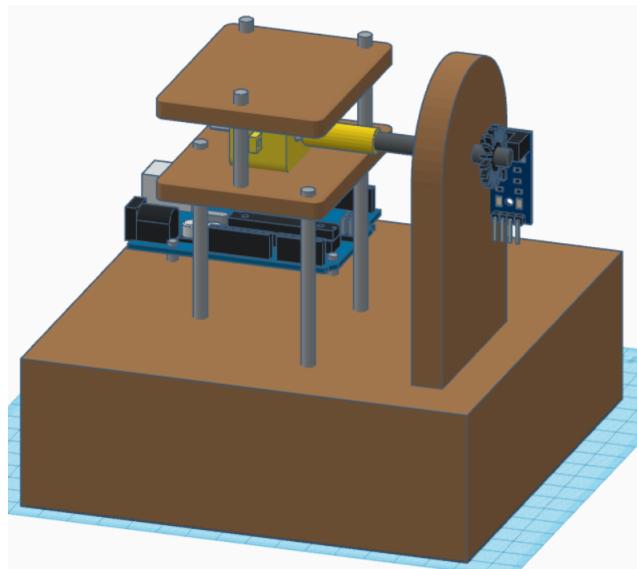
3 Bancada experimental para modelamento e identificação de parâmetros de motores CC

Neste capítulo, serão apresentados os detalhes da estrutura mecânica, *hardware* e *firmware* da bancada. O foco principal será dado à placa de acionamento e medição do motor, assim como ao algoritmo desenvolvido para o Arduino.

3.1 Estrutura mecânica

Sua construção foi projetada de forma a acomodar o motor utilizado nos testes da maneira mais conveniente possível, tornando fácil a instalação e remoção do motor no equipamento. No entanto, com pequenas modificações, ela pode acomodar outros motores de tamanho e formato semelhantes. E mostrada na Figura 15 uma visão geral do modelo 3D da bancada.

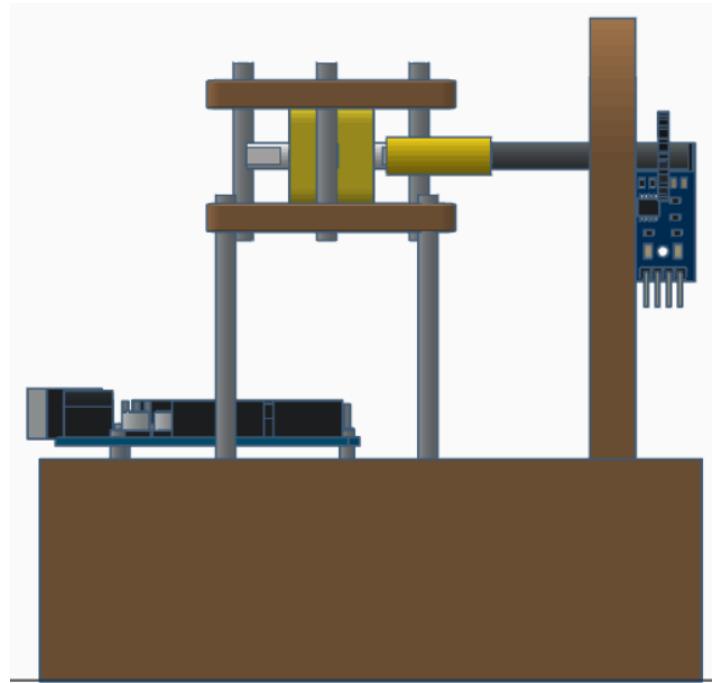
Figura 15 – Visão geral do modelo 3D da bancada



Fonte: O autor

O eixo do motor é conectado a um parafuso por meio de uma mangueira, como mostrado na cor amarela. O parafuso está fixo em um mancal e possui na outra extremidade um disco do *encoder*, com 18 dentes. Essa configuração permite que o disco esteja sempre fixo e alinhado com o módulo do sensor, independentemente do motor utilizado. Dessa forma, é possível substituir facilmente o motor sem alterar a posição do sensor. Pode-se ter uma melhor visualização desse arranjo na Figura 16.

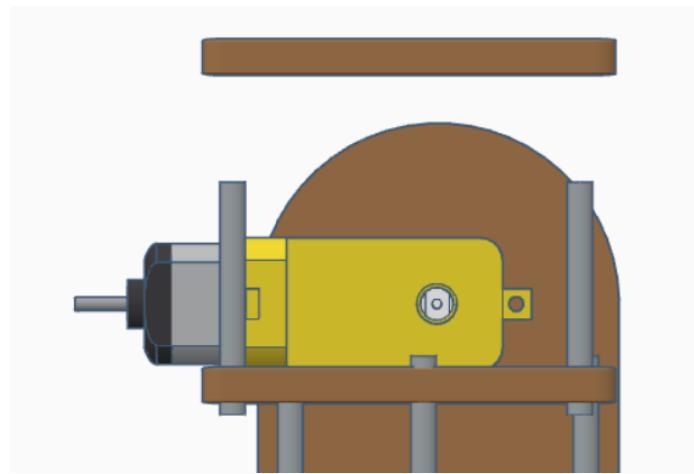
Figura 16 – Vista frontal do modelo 3D da bancada



Fonte: O autor

A base onde o motor está fixado pode ter sua altura ajustada por meio de parafusos, de modo a alinhar o eixo do motor com o parafuso. Além disso, há um suporte na parte superior que mantém o motor fixo na base. Na Figura 17, é possível observar como a parte superior do suporte pode ser removida, permitindo a remoção fácil do motor.

Figura 17 – Vista lateral do modelo 3D da bancada

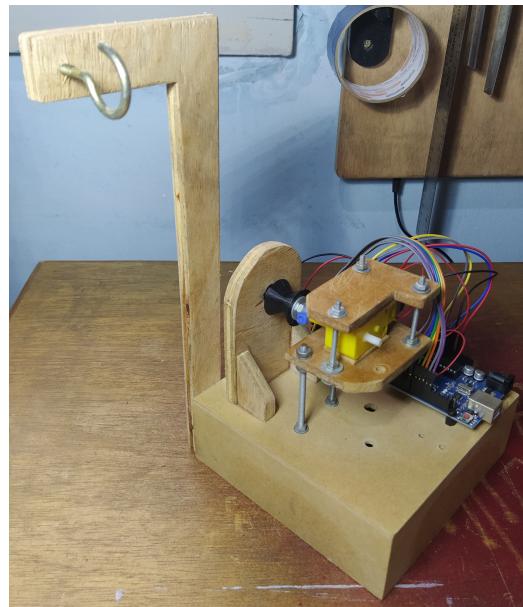


Fonte: O autor

O projeto apresentado nas Figuras 15, 16 e 17 serviu como referência para a construção da bancada. No entanto, durante o processo de fabricação decidiu-se adicionar uma polia e uma haste, permitindo implementar um sistema de controle de posição para regular a altura de um objeto suspenso no gancho, sendo puxado por uma corda fixada na polia.

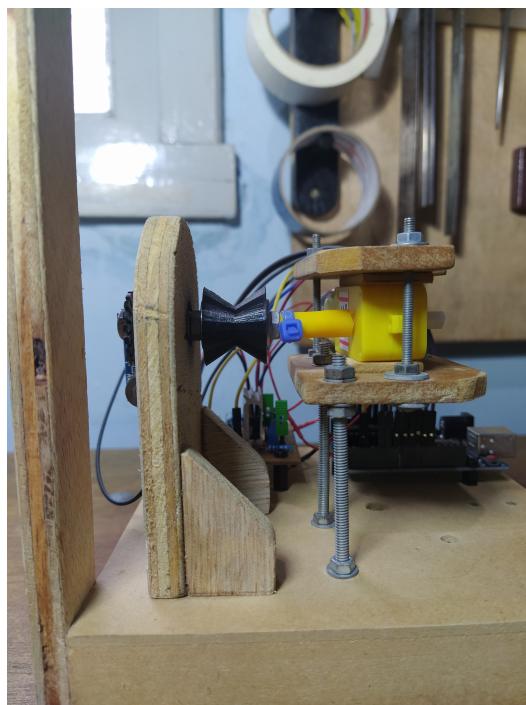
No entanto, esse controlador acabou não sendo implementado no trabalho. Os resultados podem ser visualizados nas Figuras 18 e 19.

Figura 18 – Visão geral da bancada



Fonte: O autor

Figura 19 – Conexão entre o parafuso e o eixo do motor



Fonte: O autor

3.2 Hardware utilizado

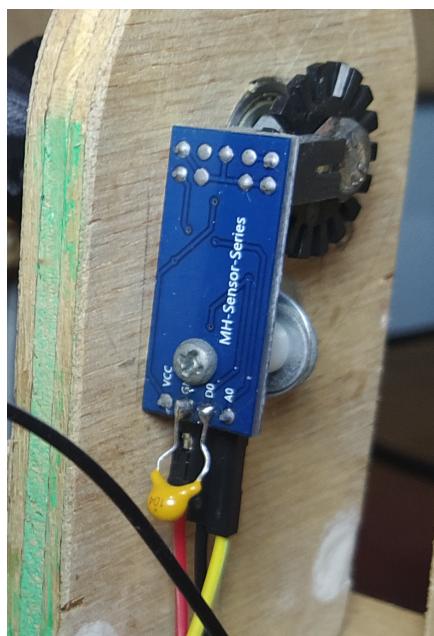
Com a estrutura da bancada conhecida, é possível ter uma compreensão mais detalhada de seu *hardware*. Ele é composto principalmente por três placas: Arduino Uno, módulo *encoder* e placa de acionamento e medição do motor, sendo esta última desenvolvida especificamente para este trabalho.

Optou-se por utilizar um Arduino Uno, pois esse é o modelo mais comum no mercado. Ele terá a função de calcular a velocidade do motor a partir do *encoder*, controlar sua tensão e ler os sensores da placa de acionamento e medição. Também será responsável por realizar os experimentos e envio de dados para o programa estimador de parâmetros que estará rodando no computador, descritos em detalhes na secção 3.3.

O Arduino Uno possui conversores analógicos digitais de 10 bits, o que é suficiente para a implementação dos sensores utilizados. Possui interrupção externa, que foi utilizada para realizar as leituras de velocidade do *encoder*. Conta também com uma grande variedade de funções, tais como "millis" e "micros", que facilitam muito na temporização. Para a comunicação com o computador ele permite utilizar a velocidade 20000000 de bits por segundo, o que permite realizar as leituras mais rapidamente. Entretanto, ele possui algumas limitações quanto ao tempo de amostragem. Implementou-se o programa mais simples possível, porém, o melhor tempo de amostragem atingido foi de 1ms.

O módulo *encoder* foi utilizado para medir a velocidade do motor. No entanto, durante os testes, foi percebida a necessidade de realizar uma pequena alteração nele. Foi necessário adicionar um capacitor de $100nF$ em sua saída para eliminar ruídos que estavam interferindo na medição, conforme mostrado na Figura 20.

Figura 20 – Módulo *encoder* com capacitor de filtro



Fonte: O autor

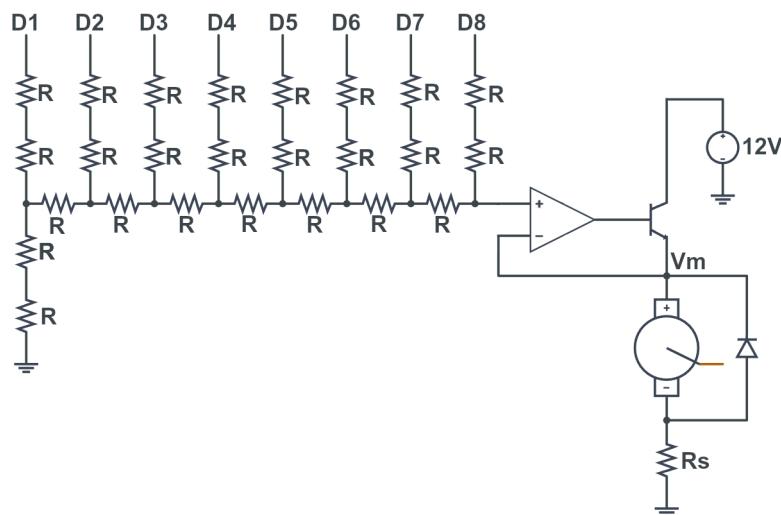
A saída do *encoder* será uma onda quadrada, com frequência proporcional à velocidade angular o motor. Portanto, para calcular a velocidade do motor basta medir o período do sinal fornecido pelo sensor e calcular a frequência angular (CARVALHO et al., 2010). Para facilitar a medição utilizou-se a interrupção externa do Arduino, porém, esse assunto será abordado com mais detalhes na seção 3.3.

A placa de acionamento e medição do motor, é responsável por controlar e medir a tensão aplicada no dispositivo e também obter o valor da sua corrente. Para definir a tensão aplicada no motor utilizou-se um conversor digital analógico R2R em conjunto com um amplificador operacional e um transistor ligado na configuração seguidor de emissor (HOROWITZ; HILL, 2017), como mostrado na Figura 21.

Um circuito R2R é um conversor utilizado para transformar um sinal digital em um sinal analógico. Ele emprega uma rede de resistores configurados de forma específica, como pode ser visto na Figura 21, para converter informações binárias em tensões analógicas correspondentes. Quando um sinal digital é aplicado ao circuito, a corrente percorre os resistores conectados, gerando quedas de tensão proporcionais. Essas quedas de tensão são somadas para produzir uma tensão analógica que representa o valor binário do sinal de entrada.

O circuito R2R é utilizado para definir a tensão aplicada na entrada não inversora do amplificador operacional. Nessa configuração do circuito seguidor de emissor, o amplificador operacional garante que a tensão no ponto V_m será a mesma da entrada não inversora. Isso ocorre pois o amplificador operacional faz a realimentação através da sua entrada inversora, fornecendo na sua saída a tensão necessária para que a tensão nas suas entradas seja a mesma. Portanto, a tensão no ponto V_m será praticamente a mesma definida pelo circuito R2R (HOROWITZ; HILL, 2017).

Figura 21 – Circuito de controle de tensão



Fonte: O autor

O resistor R_s , que está em série com o motor, é utilizado para medir a corrente que flui

por ele (COELHO; COELHO, 2004). O circuito de medição, que pode ser visto na Figura 22, é composto pelo amplificador AO2, configurado como não inversor, que amplifica a tensão do resistor para melhorar a resolução da medição e facilitar a medição de valores menores (AGUIRRE, 2013). Tomando como referência os valores obtidos nas Tabelas 1 e 2 decidiu-se que o ganho (k_{dc1}) do amplificador seria 11, portanto, o valor dos resistores será de $100k\Omega$ para R_1 e $10k\Omega$ para R_2 . Para realizar o cálculo dos resistores utilizou-se a equação (3.1), atribuindo o valor de R_2 e calculando R_1 .

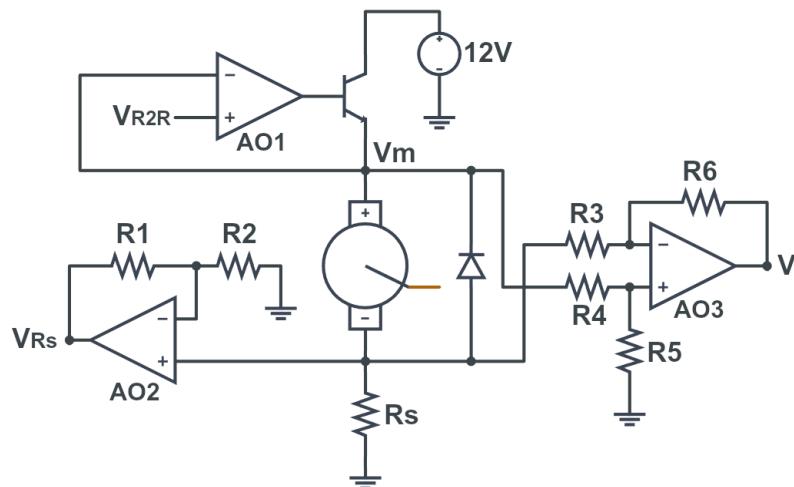
$$k_{dc1} = 1 + \frac{R_1}{R_2} \quad (3.1)$$

O amplificador AO3, configurado como amplificador subtrator, é utilizado para isolar a tensão sobre o motor das demais, permitindo que ela seja medida facilmente. Isso é possível pois ele amplifica a diferença entre a tensão nas entradas do amplificador diferencial (HOROWITZ; HILL, 2017). A equação (3.2) pode ser utilizada para calcular o valor dos resistores, levando em conta que $R_6 = R_5$ e $R_3 = R_4$. Entretanto, não foi necessário utilizá-la, pois como não deseja-se amplificar essa tensão o ganho (k_{dc2}) utilizado para esse circuito foi 1. Neste caso $R_3 = R_4 = R_5 = R_6$, utilizou-se resistores de $10k\Omega$.

$$k_{dc2} = \frac{R_6}{R_3} (V_m - V_{Rs}) \quad (3.2)$$

Resumidamente, a saída V_{Rs} é usada para medir a tensão sobre o resistor R_s , que é proporcional à corrente que passa por ele, enquanto a saída V fornece a tensão sobre o motor. Com essa placa, é possível alimentar o motor com tensões de 0 a 5V, além de medir sua tensão e corrente. Dessa forma, o sistema pode realizar todas as medições necessárias.

Figura 22 – Circuito de medições dos motores



Fonte: O autor

Foi necessário realizar a calibração dos sensores de tensão e corrente devido a possíveis diferenças geradas pelas tolerâncias dos componentes e outros fatores, mesmo que os valores

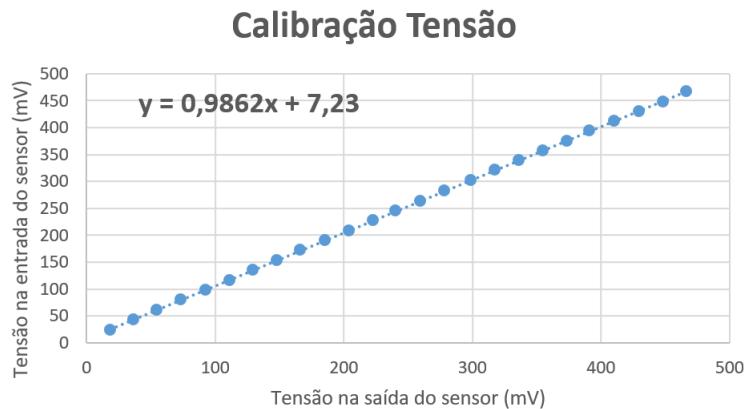
de ganho e outros parâmetros possam ser calculados. Para realizar os experimentos, aplicou-se estímulos conhecidos na entrada dos sensores e registrou-se suas saídas (AGUIRRE, 2013). Com base em uma Tabela contendo diversas combinações de valores de entrada e saída, foi possível aproximar a função de transferência que relaciona os dois.

No caso da calibração do medidor de tensão, um resistor de 20Ω foi colocado no lugar onde o motor seria conectado. Em seguida, utilizando o controle de tensão do circuito, diferentes valores de tensão foram aplicados sobre o resistor, e as tensões sobre o resistor e na saída do sensor foram medidas. Os resultados podem ser encontrados na Tabela 12. A coluna medição representa a tensão aplicada na entrada do sensor, as colunas 1, 2 e 3 representam as tensões medidas na saída do sensor, e a coluna seguinte representa a média das três medições anteriores. A coluna calculado apresenta o valor obtido para a tensão de entrada a partir da tensão média, utilizando a função de transferência obtida por meio do gráfico gerado a partir da tabela, o qual pode ser visualizado na Figura 23. A última coluna representa o erro relativo obtido pelo sensor.

Tabela 12 – Tabela de calibração do sensor de tensão

Medição $V_i(mV)$	1 $V_o(mV)$	2 $V_o(mV)$	3 $V_o(mV)$	Média $V_o(mV)$	Calculado $V_i(mV)$	Erro Relativo
23,90	18,20	18,20	18,18	18,19	24,18	1,19%
42,30	36,40	36,40	36,30	36,37	42,17	0,31%
60,80	54,60	54,60	54,60	54,60	60,21	0,97%
79,70	73,50	73,60	73,50	73,53	78,94	0,95%
98,20	92,40	92,40	92,40	92,40	97,61	0,60%
116,60	111,10	111,15	111,10	111,12	116,13	0,40%
135,00	129,60	129,63	129,60	129,61	134,43	0,42%
153,90	148,10	148,10	148,10	148,10	152,73	0,76%
172,10	166,40	166,39	166,40	166,40	170,83	0,74%
190,30	185,40	185,45	185,40	185,42	189,65	0,34%
208,80	204,30	204,33	204,40	204,34	208,38	0,20%
227,50	223,00	223,00	222,90	222,97	226,81	0,30%
245,60	240,00	240,50	240,50	240,33	243,99	0,65%
263,90	259,30	259,35	259,30	259,32	262,78	0,43%
282,20	278,50	278,50	278,50	278,50	281,76	0,16%
302,40	298,50	298,50	298,55	298,52	301,56	0,28%
320,70	317,50	317,50	317,50	317,50	320,35	0,11%
338,80	336,50	336,45	336,50	336,48	339,13	0,10%
357,00	355,00	355,00	354,90	354,97	357,42	0,12%
375,60	373,50	373,40	373,60	373,50	375,76	0,04%
393,80	391,40	391,30	391,30	391,33	393,41	0,10%
412,00	410,50	410,50	410,50	410,50	412,37	0,09%
430,10	429,70	429,70	429,75	429,72	431,39	0,30%
448,50	448,70	448,60	448,60	448,63	450,11	0,36%
466,70	466,50	466,50	466,50	466,50	467,78	0,23%

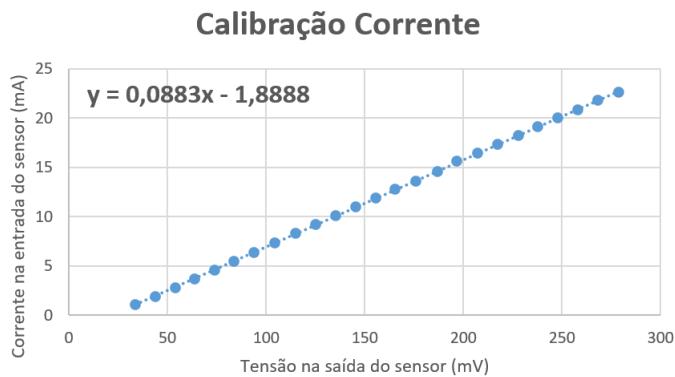
Figura 23 – Curva de calibração do sensor de tensão



Fonte: O autor

Para realizar a calibração do sensor de corrente, foi realizado um experimento semelhante ao anterior. Neste caso, a corrente que passa pelo resistor foi considerada como a variável de entrada do sensor, e foi medida utilizando um multímetro. Assim como na Tabela 12, na Tabela 13 a coluna medição apresenta a corrente aplicada na entrada do sensor, enquanto as três colunas seguintes representam as tensões obtidas na saída em cada experimento. A quinta coluna apresenta a média das três medições, e na coluna seguinte é apresentado o valor da tensão aplicada na entrada com base na média da saída, utilizando a função de transferência obtida. Essa função de transferência pode ser visualizada no gráfico mostrado na Figura 24. Por fim, é apresentado o erro relativo percentual obtido pelo sensor.

Figura 24 – Curva de calibração do sensor de corrente



Fonte: O autor

Tabela 13 – Tabela de calibração do sensor de corrente

Medição $I(mA)$	1 $V_o(mA)$	2 $V_o(mA)$	3 $V_o(mA)$	Média $V_o(mA)$	Calculado $I(mA)$	Erro Abs
1,10	33,70	33,60	33,50	33,60	1,05	4,43%
1,90	44,00	43,70	43,70	43,70	1,94	2,32%
2,80	54,00	53,70	53,90	53,90	2,85	1,63%
3,70	64,40	63,80	63,90	63,90	3,73	0,80%
4,60	74,00	73,90	73,80	73,90	4,61	0,30%
5,50	84,00	83,50	83,79	83,79	5,49	0,22%
6,40	94,50	93,80	93,80	93,80	6,37	0,42%
7,30	105,10	104,20	104,20	104,20	7,29	0,11%
8,30	115,60	114,40	114,80	114,80	8,23	0,85%
9,20	125,90	124,50	125,15	125,15	9,14	0,61%
10,10	136,20	135,00	135,25	135,25	10,04	0,62%
11,00	146,00	145,40	145,61	145,61	10,95	0,43%
11,90	156,20	155,55	155,70	155,70	11,84	0,46%
12,80	166,30	165,40	165,50	165,50	12,71	0,69%
13,60	176,50	175,80	175,80	175,80	13,62	0,16%
14,60	188,00	186,50	186,70	186,70	14,59	0,10%
15,60	198,40	196,60	196,66	196,66	15,47	0,86%
16,46	209,00	206,90	207,21	207,21	16,40	0,37%
17,30	218,00	217,10	217,30	217,30	17,29	0,06%
18,20	229,20	227,50	227,90	227,90	18,23	0,15%
19,10	239,30	237,40	237,60	237,60	19,08	0,08%
20,00	249,30	247,60	247,80	247,80	19,99	0,07%
20,80	259,50	257,80	258,00	258,00	20,89	0,42%
21,80	270,00	268,30	268,30	268,30	21,80	0,01%
22,60	280,00	278,40	278,70	278,70	22,72	0,52%

Durante os testes, foi constatado que os sensores apresentaram resultados satisfatórios e condizentes com o erro obtido na calibração, mesmo para valores fora da faixa que foi calibrada. Dessa forma, pode-se afirmar que os sensores atendem às necessidades. O esquemático completo da placa e o seu layout podem ser vistos nos apêndices A e B.

3.3 Operação da bancada

Nesta seção, serão apresentados os principais pontos da operação da bancada e da identificação de parâmetros. No funcionamento da bancada, o Arduino desempenha quatro tarefas principais. Essas tarefas incluem a medição da zona morta, da resistência da armadura, seu fator de proporcionalidade e coeficiente de atrito viscoso, além das medições de velocidade, tensão e corrente durante a resposta ao degrau do motor. A seguir, serão descritas detalhadamente as etapas para a realização de cada uma dessas tarefas.

Os parâmetros do modelo linear do motor, como resistência da armadura, indutância da armadura, fator de proporcionalidade, coeficiente de atrito viscoso e momento de inércia, podem ser identificados pela bancada desenvolvida. Além disso, ela também é capaz de encontrar algumas características não lineares, como a zona morta e o mapa de atrito (variação do coeficiente de atrito em função da velocidade).

O projeto requer a realização de cálculos fundamentais, que são executados no computador. Para isso, foi utilizado um algoritmo desenvolvido no Scilab. A comunicação com o Arduino é feita por meio da comunicação serial, utilizando a porta serial virtual criada pelo Arduino quando conectado ao computador. Assim, o programa solicita que a bancada realize as medições e processa os dados recebidos. Os experimentos são realizados na seguinte ordem: a determinação da zona morta do motor, em seguida a resistência da armadura, seguida pelo fator de proporcionalidade, mapeamento do atrito em função da velocidade e, por fim, utiliza-se os dados da resposta ao degrau para estimar a indutância e o momento de inércia.

O processo de encontrar os parâmetros do motor é dividido em duas etapas principais. A primeira etapa consiste na medição dos valores que podem ser obtidos em regime estacionário, como a zona morta, resistência da armadura, fator de proporcionalidade e coeficiente de atrito viscoso. A segunda etapa envolve a estimativa dos parâmetros que são encontrados durante o transitório, sendo eles a indutância e o momento de inércia. A seguir, serão descritas as etapas necessárias para a realização dessas medições.

Foi desenvolvido um protocolo de comunicação de forma que o Arduino recebe um caractere correspondente a cada função a ser realizada e, se necessário, retorna os dados via serial no formato: "dado1, dado2, dado3,..., dadoN;".

Na Tabela 14, pode-se visualizar o caractere atribuído a cada função da bancada. Assim, quando o caractere "a" é recebido, a bancada desliga o motor. O caractere "b" é responsável pela medição da zona morta e envio dos dados via serial, e assim por diante para as demais funções. Grande parte dessas funções já foi explicada nos itens anteriores, mas algumas ainda não foram abordadas.

A função de resetar o Arduino, como o próprio nome sugere, realiza um reset no programa, evitando possíveis falhas que foram identificadas durante os testes práticos. Já a função de monitor contínuo liga o motor, realiza a leitura da tensão, corrente e velocidade,

Tabela 14 – Tabela de funções da bancada

Caractere	Função
a	Desliga o motor
b	Medir zona morta
c	Medir resistência da armadura
d	Medir o fator de proporcionalidade e o coeficiente de atrito viscoso
e	Medir resposta ao degrau
f	Aciona motor com 3V
g	Resetar o arduino
z	Monitor contínuo

e envia os dados pela serial continuamente.

Para receber os dados da comunicação serial, foi utilizada a função *serialEvent*. Que é uma função especial que é chamada pelo *loop* principal do programa sempre que há dados disponíveis para leitura na porta serial. Dessa forma, sempre que um dado é recebido, a execução do programa é interrompida e o valor é armazenado em uma variável de seleção. Para processar esse dado e selecionar a função a ser executada pelo Arduino, foi utilizado um *switch case*, no qual a variável de seleção é comparada com os caracteres listados na Tabela 14.

Para enviar um valor para o conversor DA, foi implementado um código simples. Conforme ilustrado na Figura 21, o circuito utiliza 8 bits, correspondentes a 8 pinos do Arduino. A função desenvolvida recebe o valor da tensão e verifica se ele está dentro do intervalo de 0 a 5. Caso contrário, o valor é limitado a esse intervalo. Em seguida, cada bit do número é enviado para um dos pinos do microcontrolador, que estão conectados ao conversor.

Para ler o *encoder*, foi utilizada uma combinação de interrupção externa e a função *micros*. A interrupção foi configurada para ser acionada tanto na borda de subida quanto na borda de descida, a fim de obter um maior número de pontos e medir com maior precisão a resposta ao degrau. *Micros* é uma função baseada no temporizador interno do Arduino que retorna o tempo em microsssegundos decorridos desde o início da execução do programa. Ela foi utilizada para medir o tempo entre cada evento de interrupção. Esse valor foi armazenado em uma variável global, que será utilizada sempre que for necessário calcular a velocidade do motor.

Existem outras funções relevantes para calcular a tensão, corrente e velocidade do motor. No entanto, apesar de serem importantes para o funcionamento do dispositivo, são códigos simples que aplicam apenas as equações matemáticas necessárias para obter os valores. Portanto, não é necessário descrevê-las neste trabalho. O código completo pode ser visto no apêndice C.

3.4 Identificação de parâmetros do modelo matemático utilizando a bancada

Para a medição da zona morta, todos os cálculos são realizados pela bancada. O programa precisa apenas enviar o caractere "b" e aguardar a resposta. Ao receber o dado, que está em formato de string, ele é convertido para inteiro. Esse processo de solicitar, receber os dados e convertê-los para inteiro é o mesmo para o restante do código.

No caso da medição da resistência da armadura, o processo inicial é o mesmo. Envia-se o caractere "c" e aguarda-se a resposta. No entanto, dessa vez a bancada enviará vários valores de tensão e corrente do motor. Esses dados são recebidos, tratados e armazenados em uma matriz de medições. Em seguida, o programa calcula a resistência para cada par de valores utilizando a Lei de Ohm. O valor final da resistência da armadura é a média dos valores obtidos.

Para medir o fator de proporcionalidade e o coeficiente de atrito viscoso, envia-se o caractere "d". A bancada retornará os valores de tensão, corrente e velocidade do motor em diferentes tensões de acionamento. Os dados são processados da mesma forma que os anteriores. Utilizando a equação (2.20), calcula-se os valores de K para cada linha da matriz, e o valor final da constante é a média de todos eles. Para calcular o fator de proporcionalidade, utiliza-se a equação (2.21).

Além disso, com as medições realizadas, também é possível mapear o atrito em função da velocidade. Tendo os valores da constante de atrito em diferentes velocidades, pode-se utilizá-los para obter uma equação para calcular o seu valor em função da velocidade do motor.

Portanto, essas medições permitem encontrar três parâmetros do modelo linear do motor, além de identificar duas não linearidades: a zona morta e a variação do atrito em função da velocidade do motor.

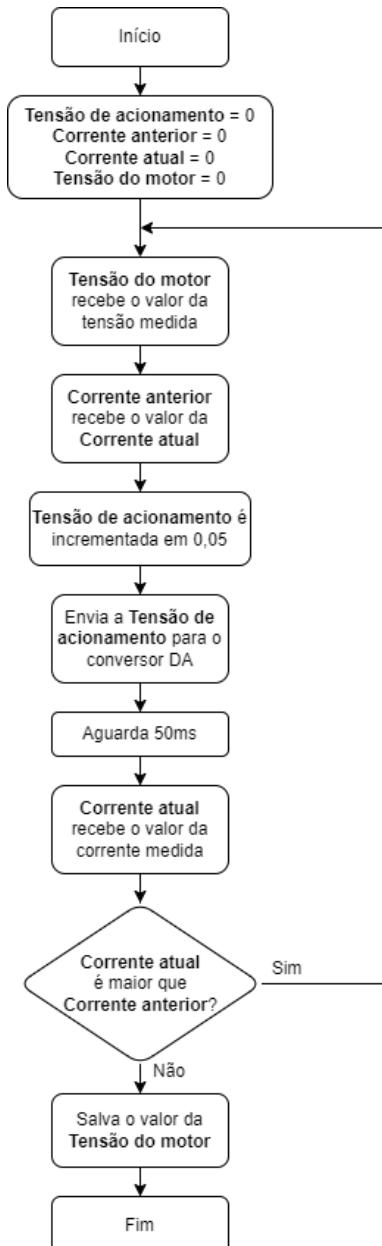
3.4.1 Identificação da zona morta

Para medir a zona morta, inicialmente é aplicado 0V no motor. Em seguida, inicia-se um laço que será repetido enquanto a corrente atual no motor for maior que a corrente anterior. Isso ocorre porque, com o motor parado, a corrente irá aumentar proporcionalmente à tensão, mas quando o rotor começa a se mover, a corrente diminui.

Dentro do laço, o programa realiza a medição da tensão sobre o motor e armazena o valor em uma variável. Em seguida, incrementa-se a tensão aplicada no motor em 0,05V, aguarda-se 50ms e se realiza a medição da corrente atual no motor. No início de cada iteração do laço, a corrente anterior é atualizada para ser utilizada na condição de saída do laço.

Dessa forma, o programa aumenta gradualmente a tensão sobre o motor e, quando

Figura 25 – Fluxograma da função responsável por medir a zona morta



Fonte: O autor

detecta o movimento, interrompe o aumento. O último valor medido será a zona morta. O fluxograma que demonstra o funcionamento dessa função pode ser visualizado na Figura 25. É importante destacar que essa função é executada dez vezes e o valor final é calculado como a média dessas leituras.

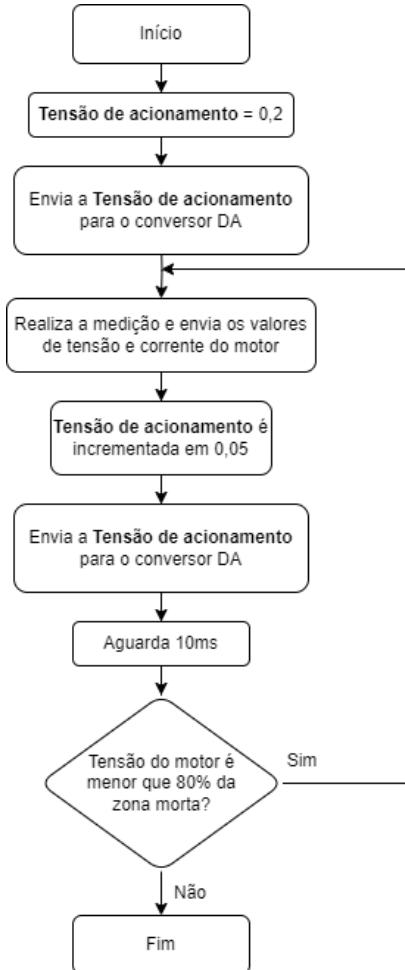
Para a zona morta a bancada identificou para o motor MA01, 1,58V em polarização direta e 1,65V em polarização reversa, enquanto para o motor MA02 foram, 1,31V em polarização direta e 1,21V em polarização reversa. Nos experimentos realizados com instrumentos de laboratório os valores obtidos para o motor MA01, 1,2V em ambas as polaridades, enquanto para o motor MA02 foram, 1V em polarização direta e 1,1V em polarização reversa. A diferença entre os valores obtidos deve-se principalmente à presença

do resistor *shunt*. Quando realizou-se a medição em laboratório o motor foi ligado diretamente a uma fonte de tensão, por outro lado na bancada existe o resistor *shunt* ligado em série com ele. Além disso, na bancada o motor está conectado ao eixo do encoder, aumentando o atrito presente no eixo.

3.4.2 Identificação da resistência da armadura

Para medir a resistência da armadura, é necessário que o motor esteja parado. O processo é semelhante ao utilizado para medir a zona morta descrito anteriormente. No entanto, para melhorar os resultados, os cálculos não são realizados no Arduino. Em vez disso, os dados são enviados pela porta serial para serem processados no computador.

Figura 26 – Fluxograma da função responsável por medir a resistência da armadura



Fonte: O autor

Inicialmente, é aplicada uma tensão de 0,2V no motor. Em seguida, o programa entra em um laço que é executado enquanto a tensão sobre o motor for menor ou igual a 80% da zona morta. Isso garante que todas as medições sejam realizadas com o motor parado e com diferentes valores de tensão.

Dentro do laço, o programa envia pela porta serial a tensão e a corrente do motor. Em seguida, incrementa-se a tensão em 0,05V e se aguarda 10ms. Dessa forma, o *software*

que está sendo executado no computador pode calcular a resistência da armadura com diferentes valores de tensão, sendo o resultado final a média entre os cálculos realizados. O fluxograma que demonstra o funcionamento dessa função pode ser visto na Figura 26.

A bancada identificou que a resistência do motor MA01, $6,580\Omega$ em polarização direta é $6,3\Omega$ em polarização reversa, enquanto para o motor MA02 foram, $6,328\Omega$ em polarização direta e $6,42\Omega$ em polarização reversa. Nos experimentos realizados com instrumentos de laboratório os valores obtidos para o motor MA01, $5,673\Omega$ em ambas as polaridades, enquanto para o motor MA02 foi, $5,745\Omega$ em ambas as polaridades. Pode-se perceber uma diferença entre os valores, isso ocorre devido as limitações do circuito de medição, resolução de medição e cálculos do Arduino.

3.4.3 Identificação do fator de proporcionalidade e do coeficiente de atrito viscoso

Esse processo deve ser realizado com o motor em movimento. Portanto, primeiramente é aplicado nele um valor correspondente a 150% da zona morta do mesmo. Em seguida, o programa entra em um laço que é executado enquanto a tensão sobre o motor for menor que a tensão máxima definida no programa.

Dentro do laço, os valores de tensão, corrente e velocidade do motor são medidos e enviados pela serial. Em seguida, a tensão enviada para o motor é incrementada em $0,1V$ e é aguardado um segundo. Para obter melhores resultados, os cálculos são realizados pelo programa em execução no computador.

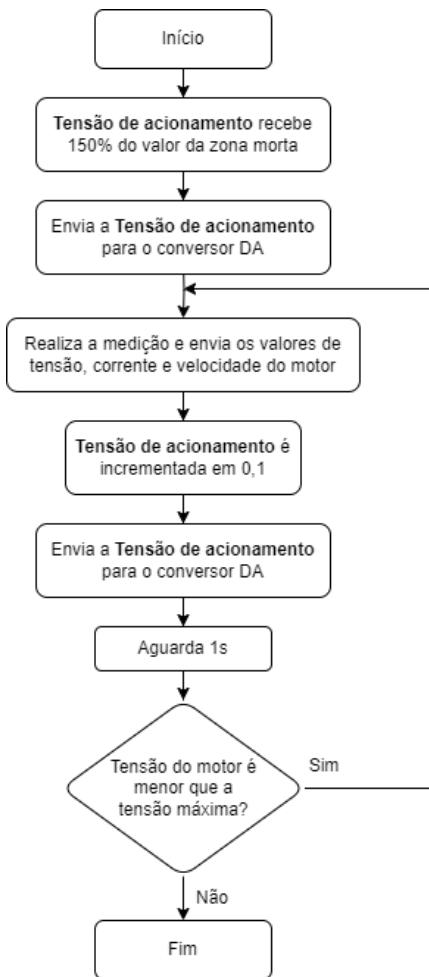
Dessa forma, o experimento é realizado em diferentes condições de funcionamento do motor, permitindo o cálculo do coeficiente de atrito em diversas velocidades. O fluxograma que demonstra o funcionamento completo dessa função pode ser visualizado na Figura 27.

O valor do fator de proporcionalidade encontrado pela bancada para o motor MA01 foi, $5,744 \cdot 10^{-3}Vs$ em polarização direta e $5,787 \cdot 10^{-3}Vs$ em polarização reversa, enquanto para o motor MA02 foram, $5,529 \cdot 10^{-3}Vs$ em polarização direta e $5,667 \cdot 10^{-3}Vs$ em polarização reversa. Nos experimentos realizados com instrumentos de laboratório os valores obtidos para o motor MA01 foi, $5,556 \cdot 10^{-3}Vs$ em polarização direta e $5,571 \cdot 10^{-3}Vs$ em polarização reversa, enquanto para o motor MA02 foram, $5,55 \cdot 10^{-3}Vs$ em polarização direta e $5,478 \cdot 10^{-3}Vs$ em polarização reversa.

Neste caso a diferença entre os valores não está relacionada somente às imprecisões dos circuitos e componentes utilizados para as medições desse experimento, pois a resistência da armadura é utilizada no cálculo. Portanto, a diferença encontrada nela afeta o valor do fator de proporcionalidade.

Para o coeficiente de atrito viscoso, os valores obtidos pela bancada podem ser vistos na Tabela 15, onde apresenta-se o valor do coeficiente para diferentes velocidades do motor. É difícil comparar esses valores com os apresentados pelas Tabelas 3, 4, 5 e 6, pois as

Figura 27 – Fluxograma da função responsável por medir o fator de proporcionalidade e o coeficiente de atrito viscoso



Fonte: O autor

velocidades não são exatamente as mesmas. Porém, pode-se ver que os valores obtidos pela bancada são similares aos medidos em laboratório. Da mesma forma que para o fator de proporcionalidade, as divergências entre os valores são geradas pelas limitações dos componentes e também as diferenças obtidas nos parâmetros anteriores.

Tabela 15 – Parâmetros estimados dos motores MA01 e MA02

Motor MA01				Motor MA02			
Direta		Reversa		Direta		Reversa	
$\omega(\text{rad/s})$	$B(\text{Nms})$	$\omega(\text{rad/s})$	$B(\text{Nms})$	$\omega(\text{rad/s})$	$B(\text{Nms})$	$\omega(\text{rad/s})$	$B(\text{Nms})$
339,36	1,3E-06	346,08	1,3E-06	378,24	9,0E-07	362,40	1,0E-06
353,76	1,3E-06	359,04	1,2E-06	395,04	9,0E-07	372,96	1,0E-06
376,32	1,2E-06	379,68	1,2E-06	410,88	8,0E-07	390,24	9,0E-07
396,00	1,1E-06	396,00	1,1E-06	430,56	8,0E-07	408,48	9,0E-07
410,40	1,1E-06	413,76	1,1E-06	445,44	8,0E-07	423,84	9,0E-07
431,04	1,0E-06	429,12	1,1E-06	465,12	8,0E-07	442,08	8,0E-07
441,12	1,0E-06	444,48	1,0E-06	482,40	8,0E-07	458,40	8,0E-07
461,76	1,0E-06	459,36	1,0E-06	502,56	7,0E-07	473,76	8,0E-07
478,08	1,0E-06	478,56	1,0E-06	517,44	7,0E-07	495,36	7,0E-07
501,60	9,0E-07	499,68	9,0E-07	534,72	7,0E-07	510,72	7,0E-07
516,48	9,0E-07	513,60	9,0E-07	555,36	7,0E-07	527,52	7,0E-07
532,80	9,0E-07	532,32	9,0E-07	571,20	7,0E-07	548,64	7,0E-07
552,00	8,0E-07	550,56	9,0E-07	588,96	6,0E-07	569,76	7,0E-07
573,60	8,0E-07	564,48	9,0E-07	606,72	6,0E-07	580,80	7,0E-07
587,52	8,0E-07	583,68	8,0E-07	625,44	6,0E-07	601,92	7,0E-07
605,28	8,0E-07	598,56	8,0E-07	640,80	6,0E-07	618,24	6,0E-07
619,68	8,0E-07	616,80	8,0E-07	662,88	6,0E-07	637,92	6,0E-07
640,32	8,0E-07	631,20	8,0E-07	677,76	6,0E-07	653,76	6,0E-07
654,72	7,0E-07	650,88	8,0E-07	700,32	6,0E-07	674,40	6,0E-07
675,36	7,0E-07	669,12	8,0E-07	710,40	6,0E-07	690,72	6,0E-07
696,48	7,0E-07	682,08	7,0E-07	734,88	5,0E-07	712,80	6,0E-07
707,52	7,0E-07	701,76	7,0E-07	745,44	5,0E-07	733,92	6,0E-07

Com o auxílio do *software* Excel, utilizando os valores apresentados na Tabela 15 para gerar as equações (3.3), (3.4), (3.5) e (3.6). Com essas equações é possível calcular o valor do coeficiente de atrito em função da velocidade. Desta forma pode-se inserir esta não linearidade no comportamento do motor nas simulações.

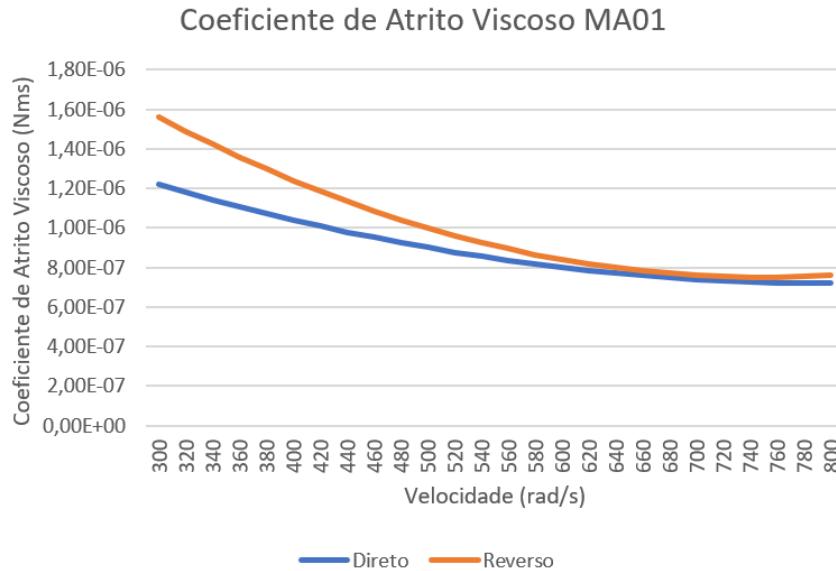
$$B(\omega) = 2 \cdot 10^{-12} \omega^2 - 3,2 \cdot 10^{-9} \omega + 2 \cdot 10^{-6} \quad (3.3)$$

$$B(\omega) = 4 \cdot 10^{-12} \omega^2 - 6 \cdot 10^{-9} \omega + 3 \cdot 10^{-6} \quad (3.4)$$

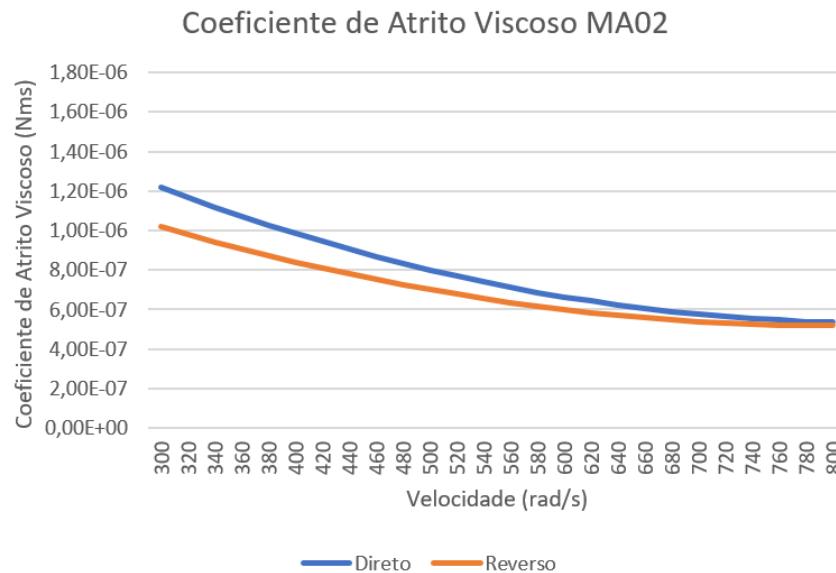
$$B(\omega) = 2,4 \cdot 10^{-12} \omega^2 - 4 \cdot 10^{-9} \omega + 2,2 \cdot 10^{-6} \quad (3.5)$$

$$B(\omega) = 2 \cdot 10^{-12} \omega^2 - 3,2 \cdot 10^{-9} \omega + 1,8 \cdot 10^{-6} \quad (3.6)$$

Nas figuras 28 e 29 pode-se visualizar o comportamento do coeficiente de atrito viscoso ($B(\omega)$), para os dois motores utilizando as equações equações (3.3), (3.4), (3.5) e (3.6).

Figura 28 – Gráfico de $B(\omega)$ para o motor MA01

Fonte: O autor

Figura 29 – Gráfico de $B(\omega)$ para o motor MA02

Fonte: O autor

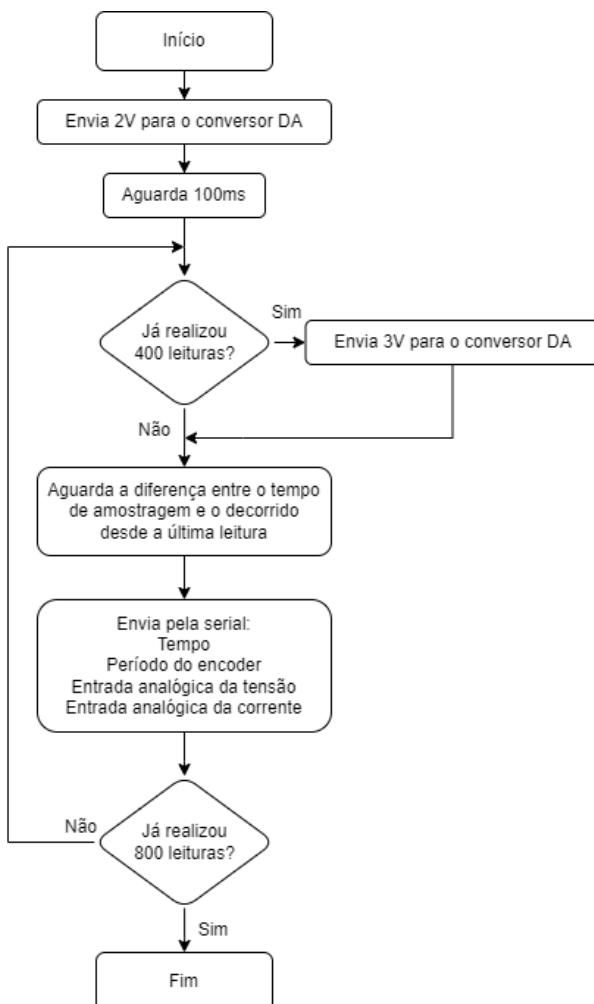
3.4.4 Identificação da indutância da armadura e do momento de inércia do rotor

Para realizar essa medição, o motor é mantido em regime permanente alimentado com 2V, e em seguida, a tensão é aumentada para 3V, aplicando um degrau unitário no motor. Durante todo o processo, são enviados pela serial o tempo, o período do *encoder* e as leituras das entradas analógicas dos sensores de tensão e corrente. É importante destacar que, nesse passo, o programa da bancada não realiza conversões dos valores lidos (saída

do conversor AD) para os valores reais, a fim de tornar o sistema mais rápido e permitir utilizar 1ms como tempo de amostragem. As conversões serão feitas posteriormente pelo programa no computador.

O Arduino realiza 800 amostras no total para o experimento, divididas igualmente entre a alimentação do motor em 2V e 3V. Para isso, utiliza-se um contador que, ao atingir a metade, altera o valor da tensão enviada para o conversor DA. Durante o envio dos dados, o programa também possui um mecanismo de atraso para manter o período de amostragem fixo, parando a execução do programa pelo tempo necessário para executar a taxa de amostragem correta. O fluxograma apresentado na Figura 30 demonstra o funcionamento completo dessa função.

Figura 30 – Fluxograma da função responsável por medir a resposta ao degrau



Fonte: O autor

Os dados são recebidos pelo computador, onde são utilizados em um algoritmo estimador de parâmetros que pode ser visto com mais detalhes no apêndice E. Após passar os dados pelo algoritmo estimador dos mínimos quadrados recursivo, obtém-se o valor de cada parâmetro. No entanto, na execução do estimador a resistência da armadura, o fator de proporcionalidade e o coeficiente de atrito viscoso são substituídas pelos valores obtidos nas

medições anteriores. Dessa forma, é forçado o estimador a encontrar valores coerentes para o momento de inércia e a indutância. Portanto, ao aplicar os novos dados no estimador, parte dos parâmetros já se aproxima de seu valor real.

O valor da indutância da armadura encontrada pela bancada para o motor MA01 foi, $0,859mH$ em polarização direta e $0,88mH$ em polarização reversa, enquanto para o motor MA02 foram, $1,23mH$ em polarização direta e $0,881mH$ em polarização reversa. Nos experimento realizados com instrumentos de laboratório os valores obtidos para o motor MA01 foi, $1,847mH$ em ambas as polaridades, enquanto para o motor MA02 foi, $1,918mH$ em ambas as polaridades.

Para o momento de inércia os valores encontrados pela bancada para o motor MA01 foi, $2.10^{-7}kgm^2$ em polarização direta e $3.10^{-7}kgm^2$ em polarização reversa, enquanto para o motor MA02 foram, $2.10^{-7}kgm^2$ em ambas as polaridades. Nos experimento realizados com instrumentos de laboratório os valores obtidos para o motor MA01 foi, $1,047.10^{-7}kgm^2$ em polarização direta e $1,039.10^{-7}kgm^2$ em polarização reversa, enquanto para o motor MA02 foram, $1,134.10^{-7}kgm^2$ em polarização direta e $1,157.10^{-7}kgm^2$ em polarização reversa.

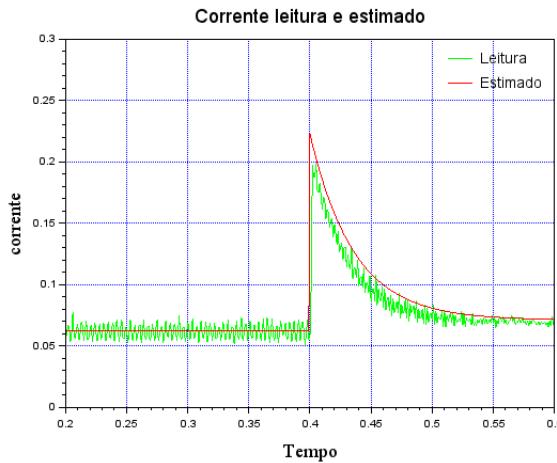
Esses são os parâmetros mais difíceis de obter, pois seu valor é obtido analisando o comportamento do motor durante o transitório. Por esse motivo eles são obtidos através do algoritmo estimador de parâmetros. As diferenças entre os valores obtidos pela bancada e os medidos em laboratório podem ser causados diretamente pelas limitações dos componentes utilizados nas medições. Enquanto em laboratório utilizou-se o osciloscópio para analisar o transitório da corrente, a bancada mediu a corrente utilizando um resistor *shunt*. O mesmo ocorre com a velocidade do motor, em laboratório utilizou-se um *encoder* de 600 pulsos por volta, o da bancada possui 18 pulsos por volta. Porém, mesmo com as limitações o estimador de parâmetros consegue encontrar valores satisfatórios para esses parâmetros.

Após a estimação dos parâmetros ser executada, o programa será encerrado e apresentará os valores obtidos na tela do Scilab. Além disso, serão gerados gráficos para comparar o resultado da simulação do modelo matemático utilizando os parâmetros obtidos com a velocidade medida pela bancada. Nas Figuras 31 e 32, é possível observar em verde o valor medido pela bancada e em vermelho o valor obtido por simulação utilizando os parâmetros obtidos pela bancada.

3.5 Parâmetros obtidos pela bancada

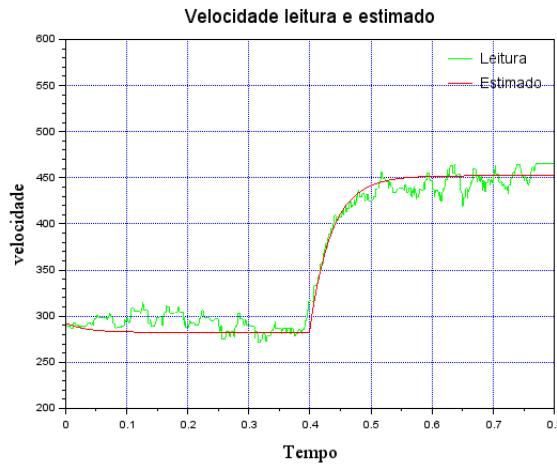
O estimador foi aplicado nos dois motores, tanto na polarização direta quanto na reversa. A Tabela 16 apresenta os parâmetros encontrados pela bancada, as equações (3.7), (3.8), (3.9) e (3.10) representam a variação do coeficiente de atrito viscoso em função da velocidade. Esses são os valores que foram utilizados para os parâmetros em todos os outros experimentos deste trabalho.

Figura 31 – Correntes medida pela bancada e simulada



Fonte: O autor

Figura 32 – Velocidades medida pela bancada e simulada



Fonte: O autor

Tabela 16 – Parâmetros estimados dos motores MA01 e MA02

Parâmetro	Motor MA01		Motor MA02	
	Direta	Reversa	Direta	Reversa
$R(\Omega)$	6,580	6,300	6,328	6,420
$L(mH)$	0,859	0,880	1,230	0,881
$K(Vs)$	$5,744E - 3$	$5,787E - 3$	$5,529E - 3$	$5,667E - 3$
$B(Nms)$	Equação (3.7)	Equação (3.8)	Equação (3.9)	Equação (3.10)
$J(kgm^2)$	$2,0E - 7$	$3,0E - 7$	$2,0E - 7$	$2,0E - 7$
$V_{min}(V)$	1,58	1,65	1,31	1,21

$$B(\omega) = 2 \cdot 10^{-12} \omega^2 - 3,2 \cdot 10^{-9} \omega + 2 \cdot 10^{-6} \quad (3.7)$$

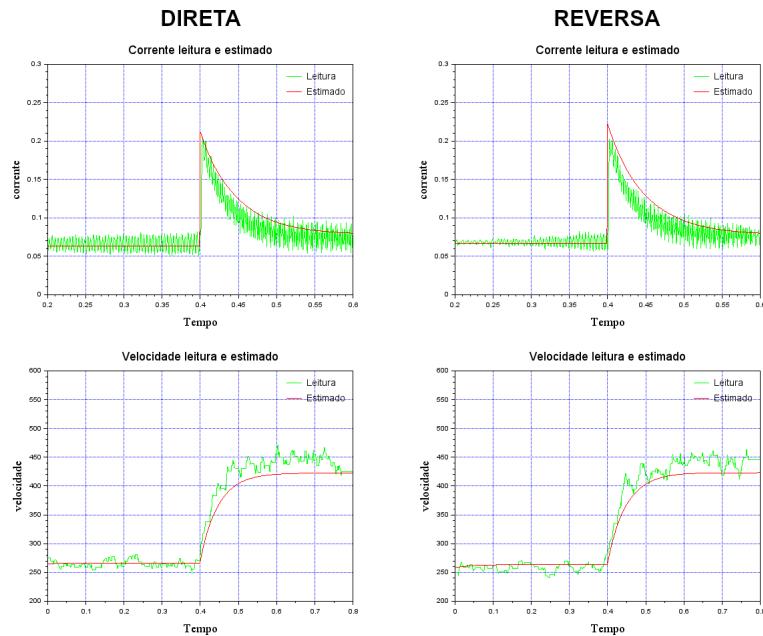
$$B(\omega) = 4 \cdot 10^{-12} \omega^2 - 6 \cdot 10^{-9} \omega + 3 \cdot 10^{-6} \quad (3.8)$$

$$B(\omega) = 2,4 \cdot 10^{-12} \omega^2 - 4 \cdot 10^{-9} \omega + 2,2 \cdot 10^{-6} \quad (3.9)$$

$$B(\omega) = 2 \cdot 10^{-12} \omega^2 - 3,2 \cdot 10^{-9} \omega + 1,8 \cdot 10^{-6} \quad (3.10)$$

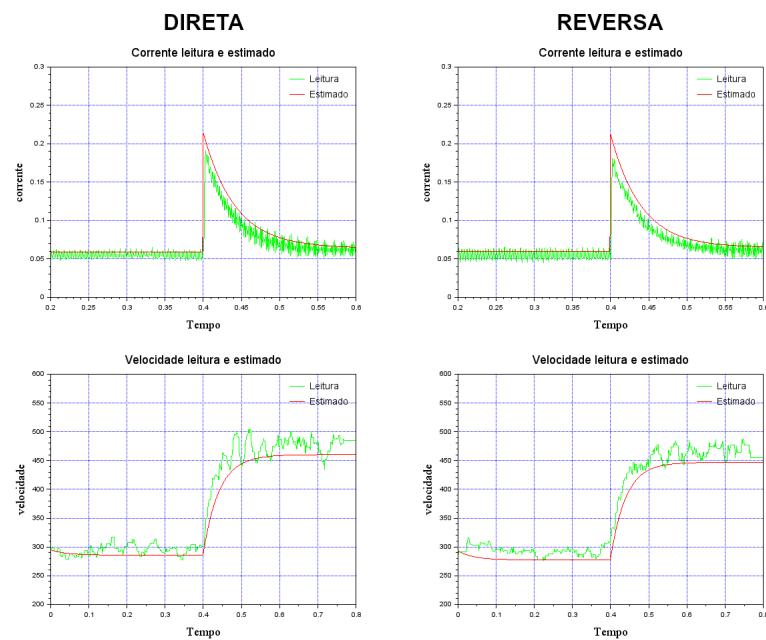
Nas Figuras 33 e 34, é possível visualizar os resultados gerados pela bancada. Os gráficos exibem a corrente e a velocidade medidas pelo dispositivo, bem como os resultados obtidos na simulação utilizando os parâmetros encontrados. Pode-se observar que a simulação apresenta um comportamento similar ao que foi medido.

Figura 33 – Valores medidos e simulados com os parâmetros encontrados pela bancada para o motor MA01



Fonte: O autor

Figura 34 – Valores medidos e simulados com os parâmetros encontrados pela bancada para o motor MA02



Fonte: O autor

4 Validação dos parâmetros obtidos pela bancada

Para validar os parâmetros obtidos pela bancada, foram desenvolvidos dois controladores de velocidade para o motor MA01, sendo polarizado diretamente. Um controlador proporcional, simples, com o ganho escolhido por tentativa e erro, foi desenvolvido. O outro controlador, que utiliza linearização por realimentação para cancelar as não linearidades do motor, foi desenvolvido com base nos parâmetros obtidos pela bancada.

Ao analisar as simulações do motor em malha aberta, constatou-se que ele possui um polo próximo a -30. Portanto, foi decidido que o sistema realimentado deveria possuir um polo em -100. Essa escolha foi feita levando em consideração a facilidade de alcançar esse valor utilizando o controlador proporcional implementado com um Arduino Uno, com um tempo de amostragem de 0,5 ms. Nos testes, uma entrada de 8 rad/s foi aplicada ao sistema e, após a estabilização, foi aplicado um degrau unitário, elevando a entrada para 9 rad/s.

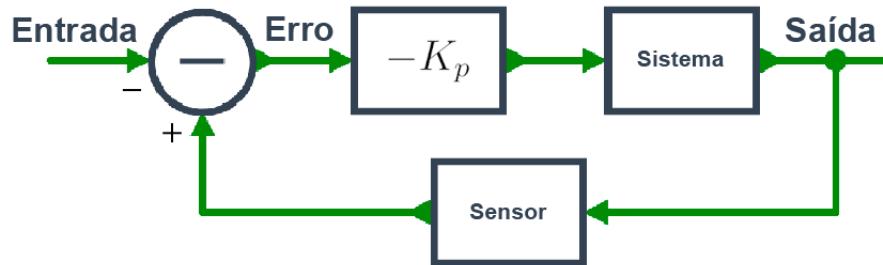
Foram realizadas simulações utilizando o SciLab, bem como experimentos práticos para ambos os controladores. Os resultados obtidos e as discussões sobre eles serão apresentados neste capítulo. As equações (2.9) e (2.10) foram discretizadas para realizar as simulações, levando em consideração os parâmetros encontrados pela bancada, apresentados na Tabela 16, incluindo as não linearidades. Dessa forma, espera-se obter melhores resultados. Além disso, o período de amostragem do controlador implementado discretamente em um Arduino foi inserido, ou seja, o controlador só atua no sistema a cada 0,5 ms.

Para a implementação prática, foram utilizados o Arduino Uno, a estrutura e o circuito da bancada. No entanto, foi necessário alterar o *encoder*, pois o modelo utilizado anteriormente era muito lento para medir a velocidade do motor, o que resultava em comportamento oscilatório do sistema realimentado. Isso acontecia devido à baixa velocidade de rotação do motor. Portanto, foi utilizado o mesmo modelo mostrado na Figura 11. Para ambos os controladores todas as leis de controle foram implementadas no Arduino Uno.

4.1 Controlador proporcional

O controlador proporcional atua de maneira proporcional ao erro entre a entrada (valor desejado) e a saída (valor real), na Figura 35 apresenta-se uma representação em blocos do controlador. A entrada do sistema será o erro $\tilde{\omega} = \omega - \omega_d$ (sendo ω_d a velocidade desejada do motor), multiplicado por uma constante K_p , que será chamada de ganho proporcional, como mostrado na equação (4.1). Quanto maior for o ganho, maior será a ação de controle exercida por esse controlador, aumentando a velocidade de resposta do sistema. Entretanto, a saída irá apresentar erro de estado estacionário, sendo esta a principal desvantagem desse controlador (SHIRAHIGE, 2007).

Figura 35 – Diagrama controlador proporcional



Fonte: O autor

$$u(t) = -K_p \tilde{\omega} \quad (4.1)$$

4.1.1 Implementando controlador proporcional

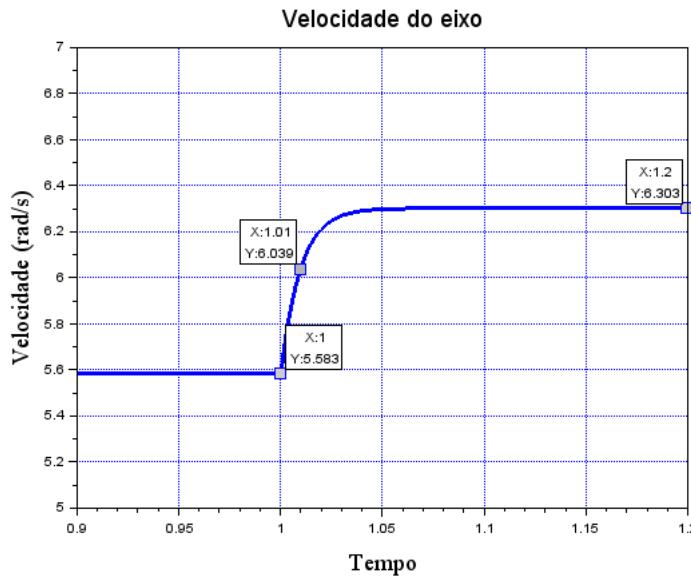
Primeiramente, o controlador foi implementado na simulação, e o ganho proporcional foi incrementado até que o polo desejado fosse alcançado no sistema realimentado. O valor obtido para o ganho foi de 0,8. Na Figura 36, a simulação desse controlador pode ser visualizada.

O polo do sistema foi medido a partir dos valores da saída em regime permanente, antes e depois da aplicação do degrau, utilizando a equação (4.2), conforme apresentado em Coelho e Coelho (2004). Em outras palavras, quando a saída atinge 5,039, ela representa 63,2% do valor em regime permanente. Esse ponto foi alcançado em 1,01 s na simulação, considerando que o degrau foi aplicado em 1 s. Assim, a constante de tempo do sistema é de 10 ms, resultando em um polo em -100.

$$Y_{63\%} = Y_i + 0,632(Y_f - Y_i) \quad (4.2)$$

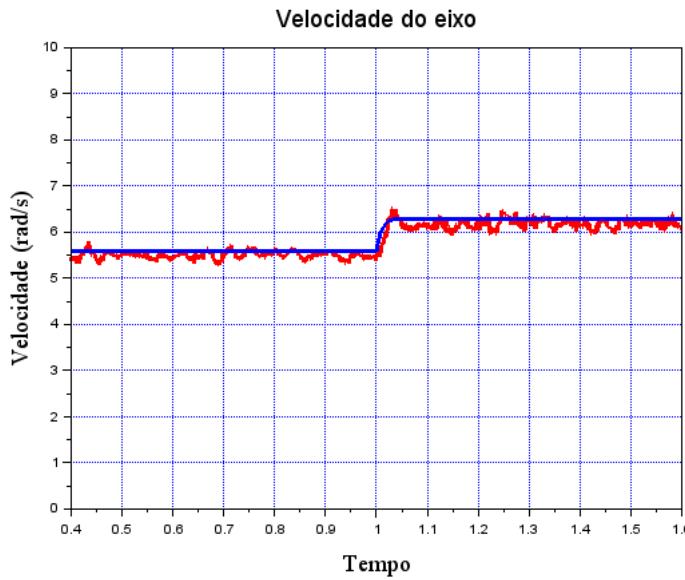
Ao implementar o controlador no Arduino com o mesmo ganho, foi obtido o resultado mostrado na Figura 37. Pode-se perceber que o resultado obtido foi coerente com a simulação, apresentando algumas oscilações na medição. Essas oscilações são causadas por diversos fatores, entre eles o principal pode ser o acoplamento mecânico entre o *encoder* e o eixo do motor. Com o motor rodando a vazio, qualquer desalinhamento entre os eixos interfere no movimento do rotor. Lembrando que a entrada do sistema era 8 rad/s e depois 9 rad/s.

Figura 36 – Resposta ao degrau da simulação do controlador proporcional



Fonte: O autor

Figura 37 – Resposta ao degrau do controlador proporcional implementado no Arduino



Fonte: O autor

4.2 Linearização por realimentação

A linearização por realimentação é uma estratégia de controle que consiste em alterar algebraicamente o comportamento dinâmico de um sistema não linear, de modo que ele passe a apresentar um comportamento linear. Desta forma é possível aplicar estratégias de controle linear em sistemas não lineares. Para utilizar a linearização por realimentação a representação de estados, como na equação (4.3), deve estar na forma canônica controlável, onde u é a entrada do sistema, $x^{(n)}$ é um vetor de estados que representa a saída, $f(x)$ e

$b(x)$ são funções não lineares dos estados (SLOTINE; LI et al., 1991).

$$x^{(n)} = f(x) + b(x)u \quad (4.3)$$

Portanto, o sistema não linear deve estar representado em espaço de estados como mostrado na equação (4.4).

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \vdots \\ x_n \\ f(x) + b(x)u \end{bmatrix} \quad (4.4)$$

Com o sistema expresso na forma canônica controlável, utilizando-se a entrada de controle como $u = \frac{1}{b}[v - f]$ e assumindo que $b \neq 0$, as não linearidades f e b serão canceladas. Assim o sistema terá uma representação simples, definida pela equação (4.5). Dessa forma é possível controlar o comportamento do sistema utilizando a lei de controle no formato mostrado na equação (4.6). Os valores de k_n devem ser escolhidos de modo que as raízes do polinômio resultante estejam todas no lado esquerdo semi-plano complexo, fazendo a dinâmica do sistema ser exponencialmente estável (SLOTINE; LI et al., 1991).

$$x^{(n)} = v \quad (4.5)$$

$$v = -k_0x - k_1\dot{x} - \dots - k_{n-1}x^{(n-1)} \quad (4.6)$$

Para uma entrada do tipo degrau unitário no sistema, é possível utilizar a seguinte lei de controle: $v = -\alpha e(t)$, onde $e(t)$ representa o erro definido por $e(t) = x - x_d$. Dessa forma, a equação do sistema em malha fechada será $x^{(n)} = -\alpha e(t)$, sendo α o polo do sistema realimentado (SLOTINE; LI et al., 1991).

Para que a saída do sistema seja capaz de seguir uma trajetória, é necessário utilizar a seguinte lei de controle: $v = x_d^{(n)} - k_0e - k_1\dot{e} - \dots - k_{n-1}e^{n-1}$. Aqui, a derivada primeira do erro é definida como $\dot{e}(t) = \dot{x} - \dot{x}_d$, e as demais derivadas são representadas por $e^{n-1}(t) = x^{n-1} - x_d^{n-1}$ (SLOTINE; LI et al., 1991).

Na implementação do controlador, optou-se por utilizar a lei $v = -\alpha e(t)$, pois o objetivo era analisar a resposta ao degrau do sistema.

4.2.1 Implementando linearização por realimentação

Através da equação (2.10), a dinâmica da velocidade do motor pode ser expressa, conforme demonstrado na equação (4.7).

$$\frac{d\omega(t)}{dt} = \frac{K}{J}i(t) - \frac{B}{J}\omega(t) \quad (4.7)$$

Sabendo que a corrente pode ser descrita pela equação (2.9), conforme apresentado na equação (4.8).

$$i(t) = \frac{1}{R}u(t) - \frac{L}{R}\frac{di(t)}{dt} - \frac{K}{R}\omega(t) \quad (4.8)$$

Ao substituir $i(t)$ na equação (4.7), a dinâmica do motor pode ser escrita como descrito pela equação (4.9), onde o comportamento do atrito B é descrito pela equação (3.7).

$$\frac{d\omega(t)}{dt} = \frac{K}{JR}u(t) - \frac{KL}{JR}\frac{di(t)}{dt} - \frac{K^2}{JR}\omega(t) - \frac{B}{J}\omega(t) \quad (4.9)$$

Se a entrada de controle $u(t)$ for definida como:

$$u(t) = \frac{JR}{K} \left(\frac{KL}{JR}\frac{di(t)}{dt} + \frac{K^2}{JR}\omega(t) + \frac{B}{J}\omega(t) + v \right) \quad (4.10)$$

Os termos da equação (4.9) são cancelados quando aplica-se a equação (4.10) na entrada de controle, resultando na dinâmica do sistema descrita pela equação (4.11). Ao implementar o controlador, o termo da derivada da corrente ($\frac{di(t)}{dt}$) na equação (4.10) não foi considerado na lei de controle. Isso se deve ao fato de que essa dinâmica é muito rápida e não foi possível realizar esse cálculo no Arduino devido a restrições como o baixo período de amostragem, limitações nos sensores de corrente e a presença de ruídos nas medições.

$$\frac{d\omega(t)}{dt} = v \quad (4.11)$$

Ao escolher v como $v = -\alpha\tilde{\omega}$, onde $\tilde{\omega}$ é o erro definido por $\tilde{\omega} = \omega - \omega_d$ (sendo ω_d a velocidade desejada do motor), e α é um número estritamente positivo, a dinâmica do sistema em malha fechada é descrita pela equação (4.12).

$$\frac{d\omega(t)}{dt} + \alpha\tilde{\omega} = 0 \quad (4.12)$$

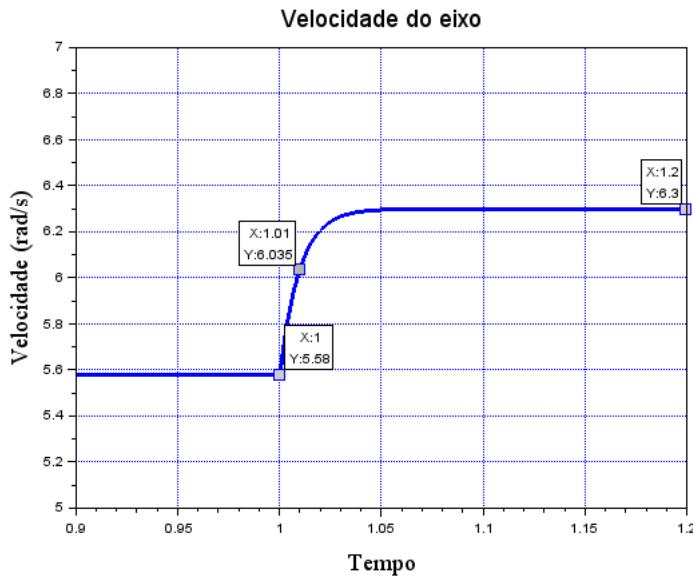
O valor de α pode ser calculado para definir o polo do sistema em malha fechada. Substituindo $\tilde{\omega}$ na equação (4.11) e aplicando a transformada de Laplace, obtém-se o seguinte resultado:

$$s\omega(s) + \alpha(\omega(s) - \omega_d) = 0 \quad (4.13)$$

Escrevendo a equação (4.13) como $\frac{\omega(s)}{\omega_d}$, obtém-se a equação (4.14):

$$\frac{\omega(s)}{\omega_d} = \frac{\alpha}{s + \alpha} \quad (4.14)$$

Figura 38 – Resposta ao degrau da simulação da linearização por realimentação



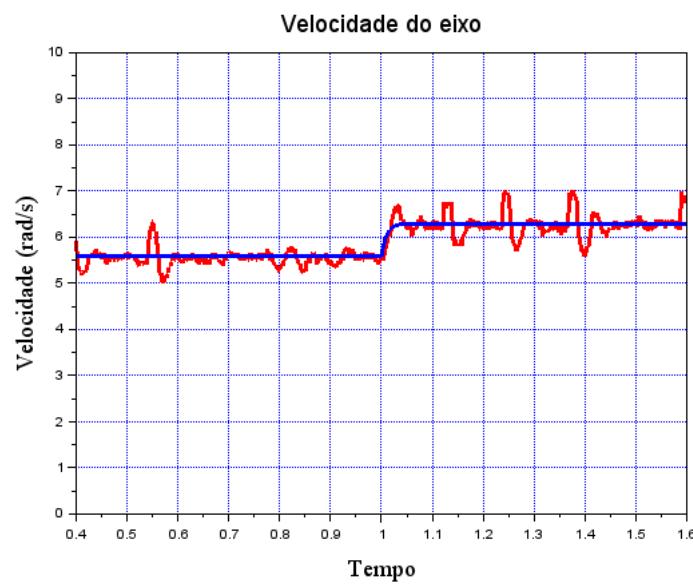
Fonte: O autor

Para alcançar um polo em -100, é necessário estabelecer o valor de α como 100. A entrada do sistema (ω_d) variou de 5,58 para 6,3, a fim de obter resultados semelhantes aos do controlador proporcional para os valores de saída correspondentes. A Figura 38 mostra o resultado da simulação desse controlador, onde o polo desejado foi alcançado.

Ao implementar o controlador no Arduino, pode-se observar na Figura 39 algumas oscilações em torno da velocidade desejada. Elas são causadas pelos mesmos efeitos comentados para o controlador proporcional. Porém, neste deve-se considerar também o efeito do controlador, que devido à limitação do período de amostragem do Arduino pode intensificar as oscilações na velocidade do motor. Mesmo assim, percebe-se que a saída do sistema permanece próxima aos valores desejados, apesar das oscilações na medição. Os resultados são consistentes com as simulações e os valores desejados para o sistema.

Os programas do Arduino e as simulações utilizadas nesse capítulo estão disponíveis em um diretório do GitHub que pode ser acessado utilizando o link presente no apêndice F.

Figura 39 – Resposta ao degrau da linearização por realimentação implementada no Arduino



Fonte: O autor

5 Conclusão

Na presente pesquisa, foi exposto o funcionamento e desenvolvimento de uma bancada automatizada para a identificação dos parâmetros do modelo matemático de motores de corrente contínua de baixo custo. Foram apresentados os detalhes do *hardware* e do *software* do dispositivo, incluindo os dados de calibração dos sensores e os fluxogramas que descrevem o funcionamento dos algoritmos. Dois controladores de velocidade foram projetados para validar os dados: um controlador proporcional e outro não linear que utilizou a linearização por realimentação.

Inicialmente, os parâmetros do modelo matemático do motor foram obtidos manualmente por meio de três experimentos. Com a implementação da bancada automatizada, tornou-se possível obter alguns valores semelhantes aos medidos manualmente, outros nem tanto. Além disso, realizou-se uma comparação entre as simulações e a resposta real do motor, revelando a similaridade dos comportamentos de ambos. Portanto, pode-se afirmar que os parâmetros obtidos pelo dispositivo são coerentes com a realidade.

Por fim, os resultados da implementação dos controladores projetados puderam ser visualizados. Especialmente no caso do controlador não linear, em que o modelo matemático não linear do motor foi utilizado para compensar as não linearidades do dispositivo, foi possível impor comportamento desejado em malha fechada, porém, com oscilações. Ao comparar as simulações com os resultados medidos na prática, pode-se constatar que os parâmetros obtidos pela bancada automática são considerados satisfatórios para a utilização na implementação de simulações e no projeto de controladores para motores CC de baixo custo.

Sugere-se os seguintes temas para trabalhos futuros:

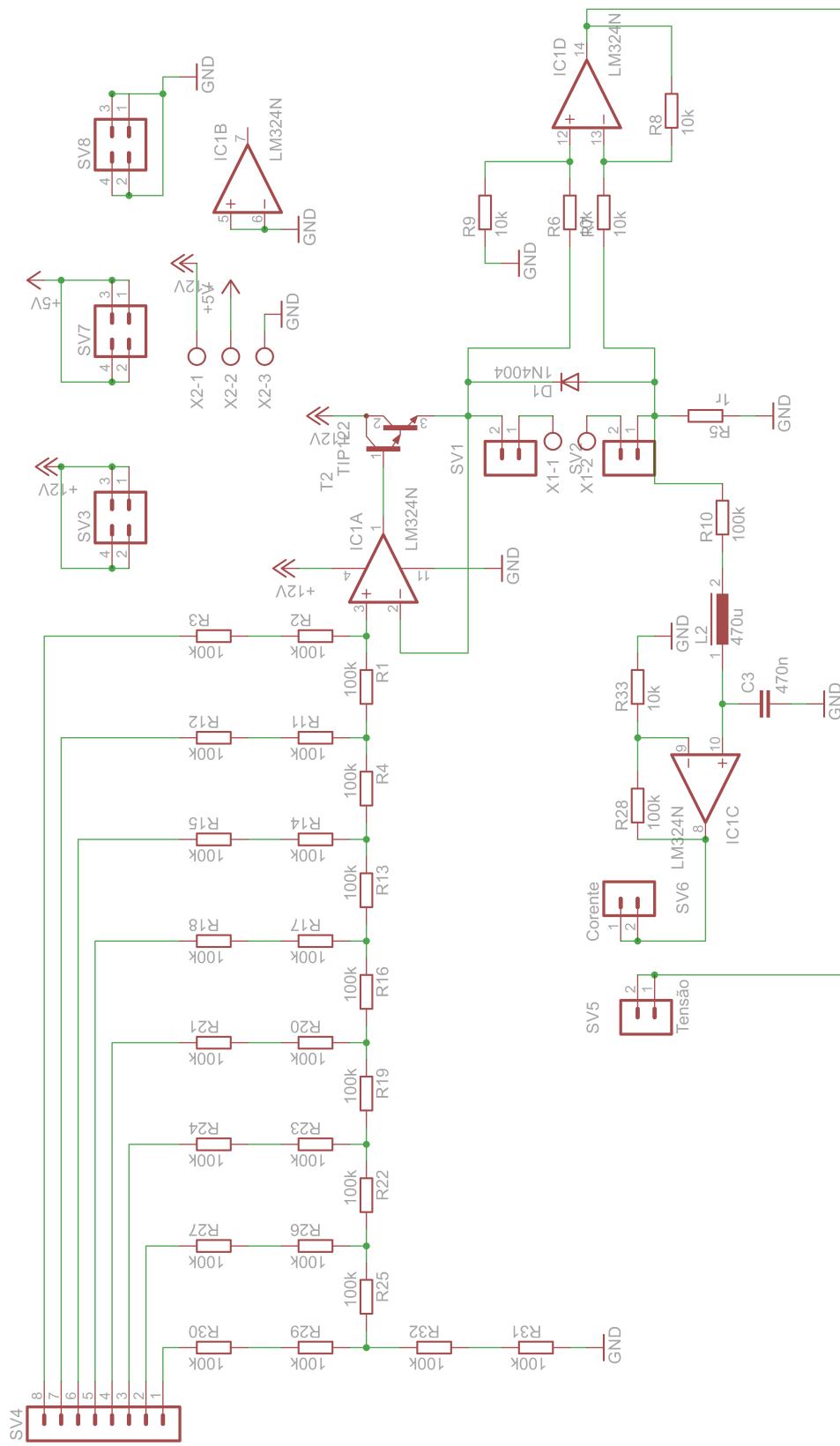
- Uma interface de usuário intuitiva e amigável pode ser projetada e desenvolvida para a bancada automatizada, facilitando a configuração dos experimentos e a visualização dos resultados;
- A expansão da bancada automatizada pode ser explorada, possibilitando o trabalho com outros tipos de motores, como motores de corrente alternada (CA) ou motores de passo;
- Uma análise comparativa entre diferentes técnicas de controle pode ser realizada, utilizando-se os parâmetros identificados pela bancada automatizada, a fim de avaliar o desempenho de cada técnica em termos de eficiência e estabilidade;
- Desenvolver uma versão da bancada automatizada com a incorporação de sensores aprimorados, visando a obtenção de medições mais precisas e sensíveis dos parâmetros do motor de corrente contínua.

Referências

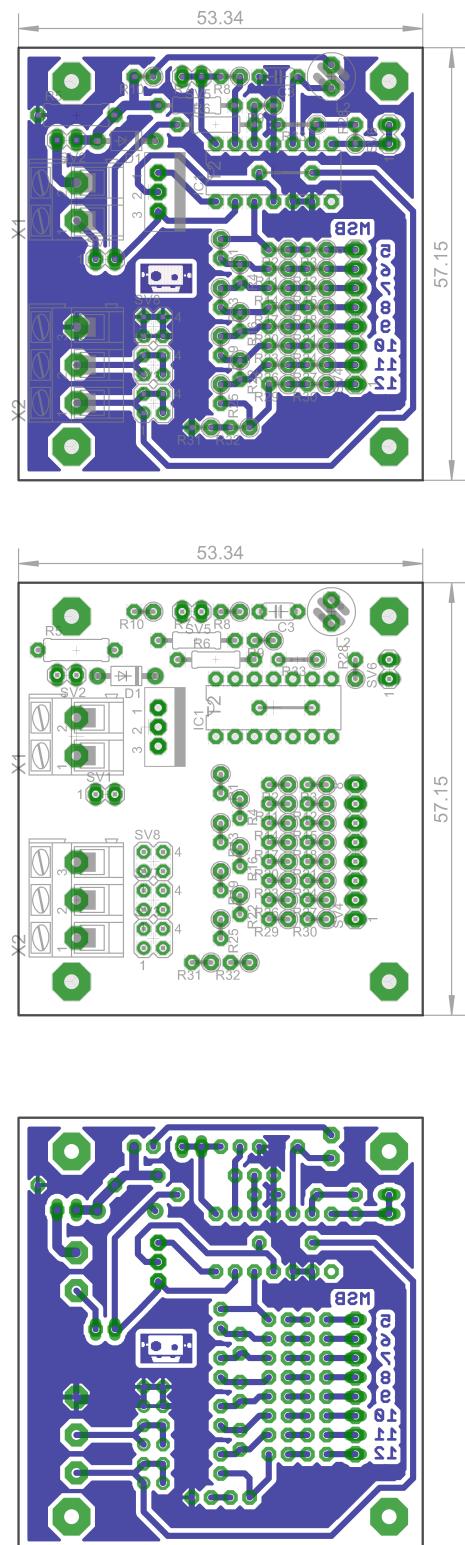
- AGUIRRE, L. A. *Encyclopédia de automática (vol. 3): controle e automação*. [S.l.]: Editora Blucher, 2007. v. 3. Citado 3 vezes nas páginas 94, 95, and 96.
- AGUIRRE, L. A. Fundamentos de instrumentação. *Luis Antonio Aguirre*, 2013. Citado 2 vezes nas páginas 40 and 41.
- CARVALHO, E. A. et al. Medição de velocidade angular com alta resolução usando encoders de baixa resolução e pll. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 21, p. 616–625, 2010. Citado na página 39.
- COELHO, A. A. R.; COELHO, L. dos S. *Identificação de sistemas dinâmicos lineares*. [S.l.: s.n.], 2004. Citado 5 vezes nas páginas 40, 59, 95, 96, and 97.
- FAULHABER. *Datasheet: Motor 0615N*. 2023. <https://www.faulhaber.com/fileadmin/Import/Media/EN_0615_S_FMM.pdf> Último acesso: 09.05.2023. Citado na página 14.
- GEROMEL, J. C.; PALHARES, Á. G. Análise linear de sistemas dinâmicos: Teoria. *Ensaios Práticos e Exercícios*, Edgar Blücher, 2004. Citado 6 vezes nas páginas 19, 21, 22, 24, 25, and 30.
- HOROWITZ, P.; HILL, W. *A arte da eletrônica: circuitos eletrônicos e microeletrônica*. [S.l.]: Porto Alegre: Bookman, 2017. Citado 2 vezes nas páginas 39 and 40.
- MEKERHERO. <<https://www.makerhero.com/produto/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo/>> Último acesso: 09.05.2023. Citado na página 15.
- NOVA, J. *Motor dc de vidro elétrico mabuchi*. 2012. <<https://labdegaragem.com/forum/topics/motor-dc-de-vidro-elettrico-mabuchi>> Último acesso: 09.05.2023. Citado na página 15.
- OGATA, K. *Engenharia de Controle Moderno. 4.ed.* São Paulo: Pearson Prentice Hall. [S.l.]: Inc, 2003. Citado 3 vezes nas páginas 17, 18, and 23.
- ROSÁRIO, J. M. *Princípios de mecatrônica*. [S.l.]: Pearson Educación, 2005. Citado na página 19.
- SHIRAHIGE, A. B. *Metodologia para aferição da ação de controle proporcional integral derivativa implementada em controladores industriais*. Tese (Doutorado) — Universidade de São Paulo, 2007. Citado na página 58.
- SLOTINE, J.-J. E.; LI, W. et al. *Applied nonlinear control*. [S.l.]: Prentice Hall Englewood Cliffs, NJ, 1991. v. 199. Citado 4 vezes nas páginas 13, 20, 21, and 61.
- WALPOLE, R. E. *Probabilidade & Estatística para engenharia e ciências*. [S.l.]: Pearson Prentice Hall, 2009. Citado na página 94.
- WEG. *Manual do servoredutor*. 2006. <<https://versatronicsul.com.br/wp-content/uploads/2019/02/WEG-SCA-05.pdf>> Último acesso: 20.06.2023. Citado na página 16.

Apêndices

APÊNDICE A – Esquemático da placa de acionamento e medição do motor



APÊNDICE B – Layout da placa de acionamento e medição do motor



APÊNDICE C – Código utilizado no Arduino

```

//Thales Gonçalves Ferreira
//Engenharia Elétrica – IFSUL
//Bancada Automatizada para a Identificação de Parâmetros – PFC

/* Legenda dos comandos
a – Desliga o motor (em mili)
b – Medir zona morta (em mili)
c – Medir resistência (em ohm)
d – Medir fator de proporcionalidade e atrito
e – Medir transitórios
f – Aciona motor com v_low
g – resetar
z – Monitor em tempo real
*/
#define ts 1000 //1000us

#define v_low    2
#define v_high   3

#define v_max   4500

#define da8     5
#define da7     6
#define da6     7
#define da5     8
#define da4     9
#define da3    10
#define da2    11
#define da1    12

#define sensor_i 0
#define sensor_v 1
#define encoder 2

#define med    10

```

```
#define leituras 800

float v_min_motor = 1000; //mV
float r_motor = 0;
float k_motor = 0;

int contador_tempo = 0;
float tempo_acumulado = 0;

long int tempo = 99999999, tempo_anterior = 0;
char selecao = 'a';
int flag = 0;
bool flag_girando = 0;

double tempo_leitura = 0, tempo_leitura_ant = 0;
char c;
int contador = 0;
int cont = 0;
int x = 0;
bool flag_change = 1;

#define config_int_default attachInterrupt(digitalPinToInterrupt
(encoder), int_encoder, CHANGE);

void (*funcReset)() = 0;

void setup() {
    Serial.begin(2000000);

    //Definir pinos como entradas e saídas
    pinMode(da8, OUTPUT);
    pinMode(da7, OUTPUT);
    pinMode(da6, OUTPUT);
    pinMode(da5, OUTPUT);
    pinMode(da4, OUTPUT);
    pinMode(da3, OUTPUT);
    pinMode(da2, OUTPUT);
    pinMode(da1, OUTPUT);

    pinMode(13, OUTPUT);
```

```
pinMode(sensor_i, INPUT);
pinMode(sensor_v, INPUT);

//configura interrupção externa
pinMode(encoder, INPUT_PULLUP);
config_int_default
selecao_tensao(0);
flag = 0;

}

void loop() {
    switch (selecao) {
        case 'a':
            selecao_tensao(0);
            break;
        case 'b':

            v_min_motor = tensao_minima();
            Serial.print(v_min_motor);
            delay(200);
            Serial.print('!');
            selecao = 1;
            break;
        case 'c':
            calculo_resistencia();
            selecao = 1;
            break;
        case 'd':
            calculo_k();
            selecao = 1;
            break;
        case 'e':

            selecao_tensao(v_low);
            delay(100);
            tempo_leitura = 0;
```

```
for (contador = 0; contador < leituras; contador++) {  
  
    if (contador == leituras / 2) {  
        selecao_tensao(v_high);  
    }  
  
    if (contador > 0) {  
        tempo_leitura = micros() - tempo_leitura_ant;  
        delayMicroseconds(ts - tempo_leitura);  
    }  
    tempo_leitura_ant = micros();  
  
    Serial.print(String(ts) + ',' + String(tempo) + ',' +  
        String(analogRead(sensor_v)) + ',' + String(  
            analogRead(sensor_i)) + ';' );  
  
}  
selecao = 6;  
selecao_tensao(v_low);  
delay(100);  
Serial.print('!');  
break;  
case 'f':  
    selecao_tensao(v_low);  
    break;  
case 'g':  
  
    funcReset();  
    break;  
case 'z':  
    float v_pot = (analogRead(2) * 5.0) / 1023.0;  
    selecao_tensao(v_pot);  
    Serial.println("V:" + String(leitura_tensao(2000)) + "I:  
    " + String(leitura_corrente(2000)) + "w:" + String  
        (leitura_velocidade(2000)));  
    break;  
}  
}
```

```
#define media_v 10
float tensao_minima() {
    float v_motor_media = 0;
    for (int x = 0; x < media_v; x++) {
        bool flag_led = 0;
        float v_out = 0, v_motor = 0;
        float corrente = 0, corrente_anterior = 0;
        flag_girando = 0;
        selecao_tensao(0);
        delay(50);
        do {
            v_motor = leitura_tensao(100);
            corrente_anterior = corrente;
            flag_led = !flag_led;
            digitalWrite(13, flag_led);
            v_out = v_out + 0.05;
            selecao_tensao(v_out);
            delay(50);
            corrente = leitura_corrente(500);

        } while (corrente > corrente_anterior);
        v_motor_media = v_motor_media + v_motor;
        selecao_tensao(0);
    }
    v_motor_media = v_motor_media / media_v;
    return v_motor_media;
}

#define media_r 5
float calculo_resistencia() {
    float resistencia_media = 0;
    for (int x = 0; x < media_r; x++) {
        bool flag_led = 0;
        float v_out = 0.2;
        selecao_tensao(v_out);
        do {
            Serial.print(String(leitura_tensao(2000)) + ',', + String(
                leitura_corrente(2000)) + ';' );
            flag_led = !flag_led;
        }
```

```
    digitalWrite(13, flag_led);
    v_out = v_out + 0.05;
    selecao_tensao(v_out);
    delay(10);
} while (leitura_tensao(100) <= ((v_min_motor) * 0.8));
selecao_tensao(0);
}

delay(200);
Serial.print('!');
}

void calculo_k() {

    float k_media = 0;
    bool flag_led = 0;
    float v_out = 1.5 * (v_min_motor / 1000.0);
    int contador = 0;
    selecao_tensao(v_out);
    float i = 0;
    float w = 0;
    float vm = 0;

    do {
        v_out = v_out + 0.1;
        flag_led = !flag_led;
        digitalWrite(13, flag_led);
        selecao_tensao(v_out);
        delay(2000);

        w = leitura_velocidade(100);
        i = leitura_corrente(2000);
        vm = leitura_tensao(2000);

        Serial.print(String(w) + ',', + String(i) + ',', + String(vm)
            + ',');

    } while (vm <= v_max);
    selecao_tensao(0);
    delay(200);
    Serial.print('!');
}
```

```
}

float leitura_tensao(float media) {
    int x1;
    float leitura = 0;
    for (x1 = 0; x1 < media; x1++) {
        leitura = leitura + analogRead(sensor_v);
    }
    leitura = leitura / media;
    leitura = (leitura * 5.0) / 1.0230;
    return leitura;
}

float leitura_corrente(float media) {
    int x1;
    float leitura = 0;
    for (x1 = 0; x1 < media; x1++) {
        leitura = leitura + analogRead(sensor_i);
    }
    leitura = leitura / media;
    leitura = (leitura * 5.0) / 1.0230;
    leitura = 0.0884 * leitura;
    return leitura;
}

float leitura_velocidade(int media) {
    float w = 0;
    double tempo_calculo_velocidade = 0;
    detachInterrupt(digitalPinToInterrupt(encoder));
    flag_change = 0;
    contador_tempo = 0;
    tempo_acumulado = 0;

    tempo = 0;
    tempo_anterior = micros();

    while (digitalRead(encoder) == 0);
```

```
for (contador_tempo = 0; contador_tempo < media;
     contador_tempo++) {
    while (digitalRead(encoder) == 1);
    tempo = tempo + (micros() - tempo_anterior) ;
    tempo_anterior = micros();
    while (digitalRead(encoder) == 0);
}
tempo_calculo_velocidade = tempo / media;
w = w + (349065.85) / (tempo_calculo_velocidade); //roda
config_int_default
flag_change = 1;
return w;

}

void int_encoder()
{
    flag_girando = 1;
    double _tempo = tempo;
    tempo = micros() - tempo_anterior ;
    tempo_anterior = micros();
    contador_tempo++;
    tempo_acumulado = tempo_acumulado + tempo;
}

void serialEvent() {
    while (Serial.available()) {
        selecao = Serial.read();
    }
}

void selecao_tensao(float V) {

    if (V > 5) {
        V = 5;
    }
    if (V < 0) {
        V = 0;
    }
    V = V * 255.0 / 5.0;
    digitalWrite(da1, ((int)V & 0b00000001));
}
```

```
digitalWrite(da2, ((int)V & 0b00000010) >> 1);  
digitalWrite(da3, ((int)V & 0b00000100) >> 2);  
digitalWrite(da4, ((int)V & 0b00001000) >> 3);  
digitalWrite(da5, ((int)V & 0b00010000) >> 4);  
digitalWrite(da6, ((int)V & 0b00100000) >> 5);  
digitalWrite(da7, ((int)V & 0b01000000) >> 6);  
digitalWrite(da8, ((int)V & 0b10000000) >> 7);  
}
```

APÊNDICE D – Código utilizado no SciLab

```

//Thales Gonçalves Ferreira
//Engenharia Elétrica – IFSUL
//Bancada Automatizada para a Identificação de Parâmetros – PFC

if( exists( "h" )==0)then
h=0;
end
closeserial(h)

clc();
clear();
close(winsid());

deslocamento=30;

leituras = 800;//número de amostrar do experimento

rodada_max=1;

media=10;
testes=15;
index_testes=1;
lambda=1
teta1=[1/(1D-6);1/(1D-6);1D-6];
teta2=[1D-6;1D-6];

com = 7;//computador
h = openserial(com, "2000000,n,8,1");//abri a porta serial

function []= nom_graf(titulo ,label_x ,label_y ,limite )
a=gca();//obtendo as configurações dos eixos
a.parent.background=8; // background = branco
a.grid = [2 ,2];//o valor [5 ,5] dá a cor azul do grid
a.title.text=titulo
a.title.font_size = 4;
a.x_label.text=label_x ;
a.x_label.font_style = 2; //tipo de fonte – 1=normal; 3=italico

```

```

a.x_label.font_size = 4;
a.y_label.text=label_y;
a.y_label.font_style = 2; //tipo de fonte - 1=normal; 3=itálico
a.y_label.font_size = 4;
a.data_bounds = limite;//limite = [xmin , ymin ; xmax , ymax]
e=gce();
e.children(1).thickness=3;
endfunction

function y=vel(t)
pulsos=36;//número de pulsos gerados pelo encoder
// no meu caso n = 2*número de dentes
mult=48;//Valor da redução
y = (mult*2*%pi*1000000)./(t*pulsos);
endfunction

function y=corrente_cal(ad)
v=(4.888)*ad//Converte o valor do AD em tensão

// essa equação deve ser obtida calibrando o sensor
y = (0.0884*v - 1.919)./1000;//calcula a corrente
endfunction

function y=tensao_cal(ad)
v=(0.004888)*ad//Converte o valor do AD em tensão

y = v;//calcula a tensão
endfunction

function dxdt=f_estimado(_t,x)
x1=x(1,:); //Corrente
x2=x(2,:); //Velocidade

if _t<t(leituras/2-1) then
u=VL
else
u=VH
end

```

```
end

_b=interpln ([ medicoes (: ,1) ' ;_b_medido_todos ' ] ,x2) ;

_A=[-_R/_L _-K/_L; _K/_j _-b/_j ];
_B=[1/_L; 0];

dxdt=_A*[x1;x2]+_B*u; // Equação de estado

endfunction

//////////Começa a comunicação os o arduino

readserial(h);
sleep(2000)
n = 999;
_n = 0;

while n > 0
_n = ( serialstatus(h)(1))
sleep(100)
n= ( serialstatus(h)(1)) - _n//2401
end
readserial(h);
//////////Começa as leituras

rodada=1

pasta= "Resultado\com"+string(media)+"\medias"+string(clock()
(3))+"\\"+string(clock()(2))+"\\"+string(clock()(1));
mkdir(pasta);

while (index_testes<size(testes)(2)+1)

mkdir(pasta+"\figuras_"+string(testes(index_testes)));
mkdir(pasta+"\resultados");

while (rodada<=rodada_max)
```

```
hora_inicio=clock();  
  
_R=0;  
_K=0;  
_L=0;  
_j=0;  
_b=0;  
  
clc();  
disp("Indice de testes:" + string(index_testes) + " / " + string(size(testes)(2)));  
disp("Rodada:" + string(rodada) + " / " + string(rodada_max));  
  
p1=eye(3,3)*500  
p2=eye(2,2)*500  
  
  
disp("Medindo a zona morta")  
recebe=""  
leitura=""  
writeserial(h, 'b');  
  
while (serialstatus(h)(1)) < 1  
  
end  
  
while recebe ~= "!"  
recebe=readserial(h,1);  
  
if (recebe ~= "!")  
leitura=leitura+recebe;  
end  
end  
  
V_minimo=strtod(leitura);  
disp("zona morta" + string(V_minimo))  
disp("Medindo R")  
  
recebe=""  
leitura=""
```

```
writeserial(h, 'c');

while ( serialstatus(h)(1) ) < 1

end

while recebe ~= "!"
recebe=readserial(h,1);

if( recebe ~= "!" )
leitura=leitura+recebe;
end
end

leitura=(strsplit(leitura , ";" ));

clear medicoes;

for i1=1:size(leitura)(1)-1
medicoes(i1 , :)=strtod(strsplit(leitura(i1) , ","));
end

_R_medido_todos = medicoes(:,1) ./ medicoes(:,2);

_R_medido = (sum(_R_medido_todos)/(size(_R_medido_todos)(1)))-1;

disp("Resistência "+string(_R_medido))
disp("Medindo K e B")

recebe=""
leitura=""
writeserial(h, 'd');

while ( serialstatus(h)(1) ) < 1

end

while recebe ~= "!"
recebe=readserial(h,1);
```

```
if(recebe ~= "!")
leitura=leitura+recebe;
end
end

leitura=(strsplit(leitura ,";"));

clear medicoes;

for i1=1:size(leitura)(1)-1
medicoes(i1 ,:)=strtod(strsplit(leitura(i1 )," ,"));
end

medicoes (: ,1)=48.*medicoes (: ,1);

_K_medido_todos = ((medicoes (: ,3) ./1000)-(_R_medido+1).*((medicoes (: ,2) ./1000)))./(medicoes (: ,1));

_K_medido = sum(_K_medido_todos)/(size(_K_medido_todos)(1));

_b_medido_todos = _K_medido_todos .* ((medicoes (: ,2) ./1000)./medicoes (: ,1));

velocidade_e_b =[medicoes (: ,1) _b_medido_todos];

_b_medido=sum(_b_medido_todos)/size(_b_medido_todos)(1);

disp("Fator de proporcionalidade "+string(_K_medido))

teta1=[_R_medido/(1D-6);_K_medido/(1D-6);1D-6];
teta2=[1D-6;1D-6];

writeserial(h, 'f');//aciona motor com V low

sleep(1000);

for ind=1:testes(index_testes)
writeserial(h, 'g');//resetar
closeserial(h)
```

```
sleep(5000);

h = openserial(com, "2000000,n,8,1"); //abri a porta serial
readserial(h);
sleep(2000)
n = 999;
_n = 0;

while n > 0
    _n = (serialstatus(h)(1))
    sleep(100)
    n= (serialstatus(h)(1)) - _n//2401
    end
    readserial(h);
    sleep(1000);
    readserial(h);

corrente=zeros(leituras,1)
velocidade=zeros(leituras,1)
tensao=zeros(leituras,1)
tempo=zeros(leituras,1)
for imed=1:media
clc();
disp("Indice de testes : " + string(index_testes) + " / " + string(size(testes)(2)));
disp("Rodada : " + string(rodada) + " / " + string(rodada_max));
disp("Teste : " + string(ind) + " / " + string(testes(index_testes)));
disp("Média : " + string(imed) + " / " + string(media));
disp("R = " + string(_R));
disp("L = " + string(_L));
disp("K = " + string(_K));
disp("j = " + string(_j));
disp("b = " + string(_b));
recebe= ""
leitura= ""

while (serialstatus(h)(1)) > 0 //limpa buffer
readserial(h);
sleep(100)
```

```
end

writeserial(h, 'e'); //aquisição dos transitórios

while (serialstatus(h)(1)) < 1000

end

while serialstatus(h)(1)>0
recebe=recebe+string(readserial(h));
sleep(100);
end

//recebe=readserial(h);

disp("Dados\u2022 recebidos")

leitura=strsplit(recebe, ";");

if(size(leitura)(1) >= 800) then

//aux=zeros(size(leitura)(1)-1,4)

for i=1:leituras
if(size(strtod(strsplit(leitura(i,1), " , " )))(1)==4) then
aux(i,:)=strtod(strsplit(leitura(i,1), " , " ));
else
aux(i,:)=aux(i-1,:);
end
end

_tempo=aux(:,1)./1000000;
_velocidade=vel(aux(:,2));
_tensao=tensao_cal(aux(:,3));
_corrente=corrente_cal(aux(:,4));
aux_backup=aux;
for i=1:size(tempo)(1)
tempo(i)=tempo(i)+_tempo(i)
end
```

```
for i=2:leituras
if(_velocidade(i)>700) then
    _velocidade(i)=_velocidade(i-1);
end
end

for i=1:size(velocidade)(1)
    velocidade(i)=velocidade(i)+_velocidade(i)
end

for i=1:size(tensao)(1)
    tensao(i)=tensao(i)+_tensao(i)
end

for i=1:size(corrente)(1)
    corrente(i)=corrente(i)+_corrente(i)
end

else
    i_med=i_med-1;
end
writeserial(h, 'f');//aciona motor com V low
end

tempo=tempo./ media
velocidade=velocidade./ media
tensao=tensao./ media
corrente=corrente./ media

tempo_csv(:,ind)=tempo;
velocidade_csv(:,ind)=velocidade ;
corrente_csv(:,ind)=corrente ;
tensao_csv(:,ind)=tensao ;

//////////////////Estimando os parâmetros

ordem=0

x1=corrente;
```

```
x2=velocidade ;  
  
for i=1:size(velocidade)(1)-deslocamento  
x2(i)=velocidade(i+deslocamento);  
end  
for i=(size(velocidade)(1)-deslocamento+1):size(velocidade)(1)  
x2(i)=x2(i-1);  
end  
  
U=tensao;  
  
clear _x1;  
clear _x2;  
  
//x1p=zeros(size(x1)(1),1);  
  
x1p=diff(x1)  
for i=2:size(x1p)(1)  
x1p(i)=x1p(i)/tempo(i+ordem/2)  
end  
//x2p=zeros(size(x2)(1),1);  
x2p=diff(x2)  
for i=2:size(x2p)(1)  
x2p(i)=x2p(i)/tempo(i+ordem/2)  
end  
  
y1=x1p;  
y2=x2p;  
  
for i=ordem+1:leituras-ordem-1  
  
fi1=[-x1(i); -x2(i); U(i)];  
fi2=[x1(i); -x2(i)];  
  
k1=(p1*fi1)*(inv(lambda+fi1'*p1*fi1));  
k2=(p2*fi2)*(inv(lambda+fi2'*p2*fi2));  
  
teta1=teta1+k1*(y1(i)-fi1'*teta1);  
teta2=teta2+k2*(y2(i)-fi2'*teta2);
```

```
p1=((eye(size(p1)(1),size(p1)(2))-k1*f1')*p1)/lambda;
p2=((eye(size(p2)(1),size(p2)(2))-k2*f2')*p2)/lambda;

end

_L=1/(teta1(3));

_K=_K_medido;
_R=_R_medido;

_j=_K/(teta2(1));

_b=_b_medido;

if (_L<0)then
disp("Deu um negativo aqui em L")
_L=-_L;
end

if (_j<0)then
disp("Deu um negativo aqui em j")
_j=-_j;
end

if (_b<0)then
disp("Deu um negativo aqui em b")
_b=-_b;
end

teta1=[_R/_L;_K/_L;1/_L];
teta2=[_K/_j;_b/_j];

parametros_csv(1,ind)=string(_R);
parametros_csv(2,ind)=string(_L);
parametros_csv(3,ind)=string(_K);
parametros_csv(4,ind)=string(_j);
parametros_csv(5,ind)=string(_b);

end
```

```
writeserial(h, 'a');//para o motor  
/////////////////comparando modelo com obtido  
  
VL=sum(U(1:380))/380;  
VH=sum(U(600:800))/200;  
  
x01=sum(x1(1:380))/380;  
x02=sum(x2(1:380))/380;  
  
_b=sum(_b_medido_todos)/size(_b_medido_todos)(1)  
  
_A=[-_R/_L -_K/_L; _K/_j -_b/_j];  
_B=[1/_L; 0];  
  
clc();  
  
disp("Zona morta "+string(V_minimo))  
  
disp("R=" + string(_R))  
disp("L=" + string(_L))  
disp("K=" + string(_K))  
disp("B=" + string(_b))  
disp("J=" + string(_j))  
  
disp("Wuuuuuuuuuuuuuu")  
  
disp(velocidade_e_b);  
  
parametros_encontrador(1)="Parametros"  
parametros_encontrador(2)="Tensao_minima: , " + string(V_minimo);  
parametros_encontrador(3)="Resistencia: , " + string(_R);  
parametros_encontrador(4)="Indutancia: , " + string(_L);  
parametros_encontrador(5)="Constante_K: , " + string(_K);  
parametros_encontrador(6)="Media_de_b: , " + string(_b);  
parametros_encontrador(7)="Momento_de_inercia: , " + string(_j);  
parametros_encontrador(8)="V, I, w, b";  
  
for i=1:length(velocidade_e_b(:,1))  
    parametros_encontrador(8+i)=string(medicoes(i,3)/1000)+", "
```

```

    string( medicoes(i,2)/1000)+" , "+string( velocidade_e_b(i,1))+" ,
    "+string( velocidade_e_b(i,2));
end

for(i_tempo=1:leituras)
t(i_tempo)=sum(tempo(1:i_tempo));
end

if( real(spec(_A)(1))<0 && real(spec(_A)(2))<0) then
x = ode("stiff", [x01;x02], 0, t, f_estimado);
_x1=x(1,:)';
_x2=x(2,:)';
close(winsid());;

figure(90)
plot(t,x2, "g-");
//nom_graf("Leitura e modelo estimado velocidade", "Tempo ", "
velocidade");
plot(t,_x2, "r-");
legends(["Leitura"; "Estimado"], [[3],[5]], with_box=%f, opt=1,
font_size=3);
nom_graf("Velocidade_leitura_e_estimado", "Tempo ", "velocidade"
,[0,200;0.8,600]);

figure(91)
plot(t,corrente, "g-");
//nom_graf("Leitura e modelo estimado corrente", "Tempo ", "
corrente");
plot(t,_x1, "r-");
legends(["Leitura"; "Estimado"], [[3],[5]], with_box=%f, opt=1,
font_size=3);
nom_graf("Corrente_leitura_e_estimado", "Tempo ", "corrente"
,[0.2,0;0.6,0.3]);

xs2png(90,pasta+"\figuras_"+string(testes(index_testes))+"\\
velocidade_leitura_e_estimado"+string(rodada)+".png");
xs2png(91,pasta+"\figuras_"+string(testes(index_testes))+"\\
corrente_leitura_e_estimado"+string(rodada)+".png");

```

```
resultado(1+(index_testes-1)*10,1+(rodada-1)*3)="Rodada";
resultado(2+(index_testes-1)*10,1+(rodada-1)*3)="Erro\u2022Corrente";
resultado(3+(index_testes-1)*10,1+(rodada-1)*3)="Erro\u2022velocidade
";
resultado(1+(index_testes-1)*10,3+(rodada-1)*3)="Duracao\u2022(min)";
resultado(2+(index_testes-1)*10,3+(rodada-1)*3)=string((clock()-
hora_inicio)(4)*60+(clock()-hora_inicio)(5));

resultado(1+(index_testes-1)*10,2+(rodada-1)*3)=string(rodada);

resultado(4+(index_testes-1)*10,2+(rodada-1)*3)="Estimados"
resultado(4+(index_testes-1)*10,3+(rodada-1)*3)="Desejados"
resultado(4+(index_testes-1)*10,4+(rodada-1)*3)="Erro"

resultado(5+(index_testes-1)*10,1)="L"
resultado(6+(index_testes-1)*10,1)="R"
resultado(7+(index_testes-1)*10,1)="K"
resultado(8+(index_testes-1)*10,1)="j"
resultado(9+(index_testes-1)*10,1)="b"

resultado(5+(index_testes-1)*10,2+(rodada-1)*3)=string(_L);
resultado(6+(index_testes-1)*10,2+(rodada-1)*3)=string(_R);
resultado(7+(index_testes-1)*10,2+(rodada-1)*3)=string(_K);
resultado(8+(index_testes-1)*10,2+(rodada-1)*3)=string(_j);
resultado(9+(index_testes-1)*10,2+(rodada-1)*3)=string(_b);

rodada=rodada+1;
end

end
rodada=1;
index_testes=index_testes+1;

csvWrite(resultado, pasta+"\resultados\"+ "Resultado até teste "+
string(testes(index_testes-1))+".csv")

end
closeserial(h); //Fecha\u2022a\u2022porta\u2022serial

csvWrite(resultado, \pasta+\resultados\+"Resultado"+string(date
```

```
( )+"_" +string( clock () (4) )+"_" +string( clock () (5) )+"_" +string( rodada)+".csv")  
  
csvWrite( parametros_encontrador , pasta+\parâmetros.csv )  
  
disp( "Tempo de trabalho ")  
  
disp( clock ()-hora_inicio )
```

APÊNDICE E – Algoritmos para estimação de parâmetros

Para desenvolver o estimador de parâmetros foi necessário entender os conceitos matemático por trás dos algoritmos para estimação de parâmetros. Nesta seção serão apresentados esses conceitos utilizados no desenvolvimento do algoritmo utilizado na bancada.

A estimação de parâmetros consiste em, utilizar os valores das variáveis observadas de um sistema para encontrar um modelo paramétrico que seja capaz de descrever o seu comportamento dinâmico. Essa análise pode ser feita a partir da equação (E.1), onde $y(k)$ é a saída, $\varphi(k)$ o vetor de variáveis medidas, θ os parâmetros e $r(k)$ representa o ruído (AGUIRRE, 2007).

$$y(k) = \varphi^T(k)\theta + r(k) \quad (\text{E.1})$$

Deseja-se determinar os parâmetros θ de modo que consiga-se minimizar a soma dos quadrados dos erros entre a saída, $y(k)$, e a sua estimativa, $\varphi^T(k)\theta$, para as amostras obtidas. Usando como base a equação (E.2), essa minimização para estimar os parâmetros é chamada de método os mínimos quadrados (WALPOLE, 2009).

$$SQE = \frac{1}{2} \sum_j |e(j)|^2 = \frac{1}{2} \sum_j |y(j) - \hat{y}(j)|^2 = \frac{1}{2} \sum_j |y(j) - \varphi^T(j)\hat{\theta}|^2 \quad (\text{E.2})$$

E.0.1 Estimador dos mínimos quadrados

Tomando-se como base o modelo de regressão linear descrito pela equação (E.1), onde φ e θ são vetores coluna e podem ser escritos como demonstrado nas equações (E.3) e (E.4), considerando que existem n_a e n_b variáveis conhecidas para o conjunto de dados (AGUIRRE, 2007):

$$\varphi^T(k) = [-y(k-1) \dots -y(k-n_a) \ u(k-1) \dots u(k-n_b)] \quad (\text{E.3})$$

$$\theta^T = [a_1 \dots a_{n_a} \ b_1 \dots b_{n_b}] \quad (\text{E.4})$$

A equação (E.1) pode ser escrita na forma vetorial como mostrado na equação (E.5). Onde o vetor de observação Y_N , a matriz regressão Φ_N e o vetor de erro E_N são definidos como mostrado na equação (E.6). E θ é o vetor com os parâmetros (AGUIRRE, 2007).

$$Y_N = \Phi_N\theta + E_N \quad (\text{E.5})$$

$$Y_N = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}; \Phi_N = \begin{bmatrix} \Phi^T(1) \\ \Phi^T(2) \\ \vdots \\ \Phi^T(N) \end{bmatrix}; E_N = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(N) \end{bmatrix} \quad (\text{E.6})$$

Reescrevendo a equação (E.2) utilizando a notação de vetorial obtém-se a equação (E.7). Onde $\hat{\theta}$ representa os valores estimados dos parâmetros θ e o erro entre os valores observados Y_N e os estimados será $\varepsilon(k) = Y_N - \Phi_N \hat{\theta}$ (AGUIRRE, 2007).

$$SQE = \frac{1}{2} \sum_j \varepsilon^2(j) = \frac{1}{2} \sum_j (Y_N - \Phi_N \hat{\theta})^T (Y_N - \Phi_N \hat{\theta}) \quad (\text{E.7})$$

Deseja-se encontrar os valores de $\hat{\theta}$. Para isso primeiro é necessário derivar a equação (E.7) em relação a $\hat{\theta}$ e igualando-a a zero, o resultado pode ser visto na equação (E.8). Sendo necessário somente isolar $\hat{\theta}$ na equação (E.8), como mostrado na equação (E.9) (AGUIRRE, 2007).

$$\frac{dSQE}{d\hat{\theta}} = -Y_N^T \Phi_N + \hat{\theta}^T (\Phi_N^T \Phi_N) = 0 \quad (\text{E.8})$$

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} Y_N^T \Phi_N \quad (\text{E.9})$$

E.0.2 Estimador dos mínimos quadrados recursivo

Algoritmos estimadores de parâmetros podem ser utilizados em diversas aplicações, entre elas pode-se ressaltar supervisão, rastreamento de parâmetros em controle adaptativo, filtros, predição de valores, processamento de sinais e outros (COELHO; COELHO, 2004). Em aplicações onde deseja-se realizar o processamento em tempo real é necessário utilizar algoritmos computacionais recursivos. Desta forma pode-se modificar o estimador dos mínimos quadrados para que ele utilize as estimativas anteriores, $k - 1$, para realizar o cálculo do instante k . Reescrevendo a equação (E.9) na forma recursiva obtém-se as equações (E.10) e (E.11) (AGUIRRE, 2007).

$$P^{-1}(k) = \Phi_N^T \Phi_N = \sum_{j=1}^k \varphi(j) \varphi^T(j) = P^{-1}(k-1) + \varphi(k) \varphi^T(k) \quad (\text{E.10})$$

$$\hat{\theta}(k) = P(k) \left\{ \sum_{j=1}^k \varphi(j) y(j) \right\} = P(k) \left\{ \sum_{j=1}^{k-1} \varphi(j) y(j) + \varphi(k) y(k) \right\} \quad (\text{E.11})$$

Para remover o somatório da equação (E.11), deve-se substitui-lo como mostrado nas equações (E.12) e (E.13).

$$\hat{\theta}(k-1) = P(k-1) \left\{ \sum_{j=1}^{k-1} \varphi(j)y(j) \right\} \quad (\text{E.12})$$

$$\sum_{j=1}^{k-1} \varphi(j)y(j) = P^{-1}(k-1) \hat{\theta}(k-1) \quad (\text{E.13})$$

Isolando $P^{-1}(k-1)$ na equação (E.10) e substituindo na equação (E.13), obtém-se a equação (E.14).

$$\sum_{j=1}^{k-1} \varphi(j)y(j) = (P^{-1}(k) - \varphi(k)\varphi^T(k)) \hat{\theta}(k-1) = P^{-1}(k)\hat{\theta}(k-1) - \varphi(k)\varphi^T(k)\hat{\theta}(k-1) \quad (\text{E.14})$$

Substituindo a equação (E.14) na equação (E.11), obtém-se o valor estimado para θ no instante k , mostrado na equação (E.15), onde $K_r(k) = P(k)\varphi(k)$ é o ganho que é multiplicado pelo erro residual (AGUIRRE, 2007). O valor de $P(k)$ é calculado utilizando o lema de inversão de matrizes (COELHO; COELHO, 2004). As equações (E.16) e (E.17) são utilizadas para calcular, respectivamente, os valores de $K_r(k)$ e $P(k)$.

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\varphi(k) \{y(k) - \varphi^T(k)\hat{\theta}(k-1)\} = \hat{\theta}(k-1) + K_r(k) \{y(k) - \varphi^T(k)\hat{\theta}(k-1)\} \quad (\text{E.15})$$

$$K_r(k) = P(k-1) \varphi(k) \left\{ 1 + \varphi^T(k)P(k-1)\varphi(k) \right\}^{-1} \quad (\text{E.16})$$

$$P(k) = P(k-1) - K_r(k)\varphi^T(k)P(k-1) \quad (\text{E.17})$$

E.0.3 Parâmetros estimados

Antes de começar a descrição do funcionamento do estimador, deve-se compreender o conceito por trás de duas variáveis. Em primeiro lugar, o número de médias define a quantidade de leituras da resposta ao degrau que serão realizadas para tirar a média e, em seguida, passar os valores pelo estimador. A variável "testes" armazena a quantidade de vezes que os dados serão processados pelo estimador. Portanto, pode-se considerar que o algoritmo estimador é executado a quantidade de vezes definida em "testes" e, em cada execução, aplica-se o número de graus definido na variável "médias".

O algoritmo é iniciado enviando o caractere "g" para a bancada, comando responsável por realizar o reset no Arduino. Após a reestabelecimento da comunicação com a bancada devido ao reset, as leituras são iniciadas para calcular a média. Para isso, o programa envia o caractere "e" e aguarda o recebimento dos dados de uma resposta ao degrau, incluindo

tempo, velocidade, tensão e corrente. Os dados são processados e convertidos de string para inteiro. Após os cálculos necessários, os valores são armazenados nas respectivas variáveis. Esse passo é repetido, conforme mencionado anteriormente.

Após a conclusão das leituras, o programa calcula a média dos valores. Em seguida, esses valores são organizados para aplicação no estimador. Além disso, é necessário calcular as derivadas da velocidade e da corrente. Para isso, utiliza-se a técnica de derivação discreta da diferença finita, que consiste em calcular a diferença entre dois valores consecutivos e dividir esse valor pelo tempo entre as duas leituras, conforme descrito na equação (E.18) (COELHO; COELHO, 2004).

$$\dot{f}(k) = \frac{f(k) - f(k-1)}{ts} \quad (\text{E.18})$$

As equações do modelo linear do motor devem ser colocadas no formato correto para utilização no estimador recursivo. Portanto, utilizando a equação (E.5) como referência, pode-se escrever as duas equações do modelo linear, (2.10) e (2.9), como mostrado nas equações (E.19), (E.20) e (E.21). Desta forma os valores medidos podem ser aplicados perfeitamente no algoritmo estimador de parâmetros.

$$Y_1 = \frac{di(t)}{dt}; Y_2 = \frac{d\omega(t)}{dt} \quad (\text{E.19})$$

$$\Phi_1^T = \begin{bmatrix} -i(t) & -\omega(t) & u(t) \end{bmatrix}; \quad \Phi_2^T = \begin{bmatrix} i(t) & -\omega(t) \end{bmatrix} \quad (\text{E.20})$$

$$\theta_1^T = \begin{bmatrix} \frac{R}{L} & \frac{K}{L} & \frac{1}{L} \end{bmatrix}; \quad \theta_2^T = \begin{bmatrix} \frac{K}{J} & \frac{B}{J} \end{bmatrix} \quad (\text{E.21})$$

APÊNDICE F – Diretório do GitHub

Utilize o QR Code ou o link a seguir para acessar o diretório do GitHub com todos os arquivos do projeto.

<https://github.com/thalestgf/PFC.git>

