



INSTITUTO FEDERAL CATARINENSE
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO
CAMPUS SÃO BENTO DO SUL

Luan Diego Ramalho Arantes

**Implementação de controle de realimentação de estados ótimo com
observador de ordem mínima em um robô auto-equilibrado**

São Bento do Sul
2022

Luan Diego Ramalho Arantes

**Implementação de controle de realimentação de estados ótimo com
observador de ordem mínima em um robô auto-equilibrado**

Trabalho de Curso submetida ao programa de graduação em Engenharia de Controle e Automação do Instituto Federal Catarinense – Campus São Bento do Sul para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Jean Carlos Bortoli Dalcin, Me.

São Bento do Sul

2022

Luan Diego Ramalho Arantes

Implementação de controle de realimentação de estados ótimo com observador de ordem mínima em um robô auto-equilibrado

Este Trabalho de Conclusão de Curso, foi julgado adequado para a obtenção do título de Bacharel em Engenharia de Controle e Automação e aprovada em sua forma final pelo curso de Bacharelado em Engenharia de Controle e Automação do Instituto Federal Catarinense – *Campus São Bento do Sul.*

autenticação eletrônica na folha de assinaturas

Prof. Jean Carlos Bortoli Dalcin, Me.

Instituto Federal Catarinense

BANCA EXAMINADORA

autenticação eletrônica na folha de assinaturas

Prof. Marcelo De Lellis Costa de Oliveira, Dr.

Universidade Federal de Santa Catarina

autenticação eletrônica na folha de assinaturas

Prof. Thiago Javaroni Prati, Me.

Instituto Federal Catarinense

São Bento do Sul

2022



ATA Nº 4018 / 2022 - CGE/SBS (11.01.14.33)

Nº do Protocolo: 23821.002552/2022-16

São Bento Do Sul-SC, 12 de dezembro de 2022.

Aos 1º dias do mês de Dezembro de 2022, no Instituto Federal Catarinense ? Campus São Bento do Sul, no Auditório, ocorreu a apresentação da defesa do Trabalho de Curso (TC) do(a) aluno(a) **Luan Diego Ramalho Arantes** do Curso de Engenharia de Controle e Automação, como requisito parcial para a obtenção do título de **BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**, do Instituto Federal Catarinense ? Campus São Bento do Sul. O(A) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Conforme a avaliação do trabalho escrito e da apresentação, a Banca Examinadora considerou o trabalho **Aprovado** com média geral **9.2**.

Assinatura do Professor Dr. Marcelo de Lellis de Oliveira se encontra em documento em Anexo

(Assinado digitalmente em 12/12/2022 20:00)
JEAN CARLOS BORTOLI DALCIN
PROFESSOR ENS BASICO TECN TECNOLOGICO
CGE/SBS (11.01.14.33)
Matrícula: 2323474

(Assinado digitalmente em 19/12/2022 09:43)
THIAGO JAVARONI PRATI
COORDENADOR - TITULAR
CGE/LUZ (11.01.11.02)
Matrícula: 2251088

Visualize o documento original em <https://sig.ifc.edu.br/public/documentos/index.jsp>
informando seu número: **4018**, ano: **2022**, tipo: **ATA**, data de emissão: **12/12/2022** e o código
de verificação: **2834bafc0c**

Luan Diego Ramalho Arantes

**Implementação de controle de realimentação de estados ótimo com
observador de ordem mínima em um robô auto-equilibrado**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Engenharia de Controle e Automação e aprovado em sua forma final pelo Curso de Bacharelado em Engenharia de Controle e Automação do Instituto Federal Catarinense - *Campus São Bento do Sul.*

autenticação eletrônica na folha de assinaturas

Prof. Jean Carlos Bortoli Dalcin, Me. Instituto Federal Catarinense

BANCA EXAMINADORA

autenticação eletrônica na folha de assinaturas

Prof. Marcelo De Lellis Costa de Oliveira, Dr. Universidade Federal de Santa Catarina

autenticação eletrônica na folha de assinaturas

Prof. Thiago Javaroni Prati, Me. Instituto Federal Catarinense

São Bento do Sul, 20 de Dezembro de 2022.

AGRADECIMENTOS

Agradeço a todos, minha família, parentes, namorada e amigos que com seu suporte me fizeram chegar à conclusão do meu curso e começo de uma nova jornada.

Gostaria de agradecer em particular ao meu amigo Samuel pelo companheirismo e ajuda, sem seu auxílio com a impressão 3d esse trabalho não teria sido realizado.

Gostaria de agradecer ao meu amigo Vagner por ter compartilhado seus conhecimentos, os quais facilitaram o desenvolvimento deste trabalho.

Gostaria de agradecer aos meus amigos Leandro, Lucas e Elian por auxiliarem em algumas questões técnicas no desenvolvimento do projeto.

Um agradecimento especial para a instituição, seu integrantes e professores que foram responsáveis pelo meu desenvolvimento intelectual e garantiram o sucesso da minha jornada, em especial ao meu professor orientador por auxiliar no desenvolvimento deste projeto e no entendimento do conteúdo.

RESUMO

O controle de sistemas é uma ferramenta essencial para o desenvolvimento da sociedade e da tecnologia, ele nos permite controlar desde a temperatura de um quarto até o movimento complexo de um foguete. O estudo dos métodos de controle é muitas vezes atrelado a plantas de controle e a análise do seu desempenho. Um dos exemplos recorrentes na literatura é o estudo do robô auto-equilibrado, principalmente por ele se assemelhar a sistemas mais complexos, onde não linearidades são muito presentes. Este trabalho visa utilizar um robô auto-equilibrado como plataforma de estudo para implementar táticas de controle moderno. Aqui é abordado as etapas de construção, modelagem e identificação da planta de controle, bem como o desenvolvimento e implementação do controle em conjunto com um observador de ordem mínima para viabilizar o controle por realimentação de estados do sistema. O trabalho apresenta resultados satisfatórios que vão ao encontro dos objetivos estabelecidos, entretanto, o comportamento real difere do simulado e fica evidente algumas das limitações do uso do controle linear em sistemas mais complexos.

Palavras-chave: Controle moderno; Observador de ordem mínima;Robô auto-equilibrado.

ABSTRACT

The control of systems is an essential tool for the development of society and technology. It allows us to control from the temperature of a room to the complex movement of a rocket. The research of control methods is often bound to control plants and the analysis of their performance. One common example in the bibliography is the research of the self-balancing robot, because it resembles more complex systems, where non linearities are highly present. This work aims to use a self-balancing robot as a research platform to implement modern control tactics. In this paper, we will see the steps of construction, modeling and identification of the control plant, as well as the design and implementation of the control combined with a minimum order observer to provide the states that will be feedback in order to control the system. This work shows satisfactory results that prove the attainment of the established objectives; however, the behavior of the actual system differs from the simulation, and the limitations of the use of linear control in complex systems become evident.

Keywords: Modern Control; Minimum order observer; Self-balancing robot.

LISTA DE FIGURAS

Figura 1 – Motor de corrente contínua.	19
Figura 2 – Ponte H.	20
Figura 3 – MPU-6050.	21
Figura 4 – Raspberry Pi Pico.	22
Figura 5 – Placa de carregamento.	23
Figura 6 – <i>Stepdown</i>	23
Figura 7 – Diagrama da placa de conexão.	24
Figura 8 – Placa de conexão.	25
Figura 9 – Robô auto-equilibrado.	25
Figura 10 – Diagrama da interligação dos componentes do robô.	26
Figura 11 – Processo de medição do ângulo do motor.	27
Figura 12 – Processo de controle do robô.	28
Figura 13 – Ciclo de trabalho do robô.	29
Figura 14 – Diagrama de corpo livre.	31
Figura 15 – Diagrama de blocos para o motor em malha fechada.	37
Figura 16 – Resposta do motor a diferentes entradas do tipo degrau em malha fechada.	38
Figura 17 – Resposta do motor a diferentes entradas senoidais em malha fechada.	40
Figura 18 – Resposta do motor a diferentes entradas do tipo rampa em malha fechada.	41
Figura 19 – Resposta do motor a uma entrada de 5,99V em malha aberta.	42
Figura 20 – Resposta do sistema motor-rodas a diferentes entradas senoidais em malha fechada.	45
Figura 21 – Comportamento do ângulo θ quando I_c varia e $b_r=0,02 \text{ kg}\cdot\text{m}^2/\text{s}$	47
Figura 22 – Comportamento do ângulo θ conforme b_r varia e $I_c=0,06 \text{ kg}\cdot\text{m}^2$	49
Figura 23 – Comportamento do ângulo φ conforme I_c varia e $b_r=0,02 \text{ kg}\cdot\text{m}^2/\text{s}$	51
Figura 24 – Comportamento do ângulo φ conforme b_r varia e $I_c=0,06 \text{ kg}\cdot\text{m}^2$	53
Figura 25 – Resposta de θ com os parâmetros que mais se aproximam do real.	55
Figura 26 – Resposta de φ com os parâmetros que mais se aproximam do real.	56
Figura 27 – Resposta do sistema ao impulso.	60
Figura 28 – Algoritmo para a determinação do ganho \mathbf{K}	62
Figura 29 – Resposta do primeiro controle ao estado inicial $\mathbf{x}(0)$	64
Figura 30 – Resposta do segundo controle ao estado inicial $\mathbf{x}(0)$	66
Figura 31 – Simulação do controle com observador $\mathbf{x}(0)$	70
Figura 32 – Resultado prático.	72
Figura 33 – Resultado prático após o ajuste.	74

LISTA DE TABELAS

Tabela 1 – Características do Motor.	20
Tabela 2 – Variáveis extraídas da Tabela 1.	43
Tabela 3 – Resultado das medidas do robô.	44
Tabela 4 – Parâmetros do robô auto-equilibrado.	57

LISTA DE ABREVIATURAS E SIGLAS

DAC	<i>Digital-to-Analog Converter</i>
I2C	<i>Inter-Integrated Circuit</i>
LQR	<i>Linear-Quadratic Regulator</i>
PID	Controle Proporcional Integral Derivativo
PPR	Pulsos por revolução
PWM	<i>Pulse-Width Modulation</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	10
1.1.1	Objetivo geral	10
1.1.2	Objetivos específicos	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	ROBÔ AUTO-EQUILIBRADO	12
2.2	REPRESENTAÇÃO POR ESPAÇO DE ESTADOS E MECÂNICA LAGRANGIANA	13
2.3	DESENVOLVIMENTO DO CONTROLADOR POR REALIMENTAÇÃO DE ESTADOS	14
2.4	IMPLEMENTAÇÃO DE OBSERVADORES	17
3	ROBÔ AUTO-EQUILIBRADO	19
3.1	MOTORES	19
3.2	GIROSCÓPIO E ACELERÔMETRO	20
3.3	MICROCONTROLADOR	21
3.4	FONTE DE ALIMENTAÇÃO	22
3.5	ESTRUTURA	23
3.6	ALGORITMO	26
4	CARACTERIZAÇÃO DO SISTEMA	30
4.1	EQUAÇÕES DO SISTEMA	30
4.1.1	Equação do Robô	30
4.1.2	Equação do Motor	33
4.1.3	Espaço de Estados do Sistema	35
4.2	DETERMINAÇÃO DOS PARÂMETROS DO SISTEMA	36
4.2.1	Parâmetros do Motor	36
4.2.2	Parâmetros do Robô	44
5	CONTROLE	59
6	RESULTADOS E DISCUSSÃO	69
7	CONCLUSÃO	76
	REFERÊNCIAS	77
	APÊNDICE A – Descrição	82

1 INTRODUÇÃO

O controle realimentado é um conceito bastante antigo que é aplicado a uma infinidade de situações, por exemplo, um bípede consegue se manter de pé e andar sem cair graças ao controle realimentado. O mesmo princípio permite com que um foguete pouse sem intervenção humana (FRIEDLAND, 2012; FRANKLIN *et al.*, 2015). O controle de sistemas busca utilizar o princípio da realimentação para que um determinado processo ou atividade apresente resultados e desempenhos desejados. O estudo de sistema de controle evoluiu em conjunto com o avanço tecnológico, já que processos mais complexos requerem controles cada vez mais eficientes e precisos. O desafio de um engenheiro de controle de sistema vem sendo desenvolver métodos simples, mas confiáveis, que consigam ser aplicados ao controle de inúmeros processos (NISE, 2015; DORF; BISHOP, 2022).

Estudos envolvendo métodos de identificação de sistemas, modelagem e implementação de controladores, e táticas de processamento de sinais são os responsáveis pela evolução da área de controle e por dar início a novos conceitos que poderão ser utilizados para solucionar problemas maiores. Uma plataforma para o desenvolvimento desses estudos é a criação de plantas de controle, como é o exemplo do robô auto-equilibrado.

O caso de Grasser *et al.* (2002) foi um dos primeiros robôs auto-equilibrado a serem construídos com sucesso na literatura, entretanto não muito tempo depois diversos casos de novos estudos relacionados ao robô auto-equilibrado começaram a surgir, como, por exemplo, Ooi (2003), Kadir (2005) e Xiaogang *et al.* (2008). A escolha pelo robô auto-equilibrado é motivada por se tratar de um sistema naturalmente instável e não linear, os métodos utilizados para a estabilização e controle deste sistema podem ser de grande ajuda para sistemas mais complexos como o controle de satélites e foguetes.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Implementar um controle de realimentação de estados discreto em conjunto com um observador de ordem mínima para estabilizar um robô auto-equilibrado.

1.1.2 Objetivos específicos

- Projetar e construir um robô auto-equilibrado.
- Modelar matematicamente o sistema e identificar os parâmetros que melhor descrevem o comportamento do sistema.
- Desenvolver e simular um controle, baseado em um observador de estados, que estabilize o sistema.
- Implementar na prática o controle criado e analisar os resultados encontrados.

ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em sete capítulos. Após este capítulo introdutório o próximos discorrem sobre o desenvolvimento. Fundamentação teórica trata dos principais conceitos utilizados ao longo do trabalho, robô auto-equilibrado aborda os componentes e o funcionamento da planta de controle desenvolvida, caracterização do sistema demonstra como foi encontrado um modelo para o sistema a ser controlado, controle apresenta o desenvolvimento do controlador e do observador e por fim em resultados e discussão são expostos os resultados atingidos e as principais considerações. O último capítulo trata da conclusão do trabalho, apresentando também sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar uma revisão teórica dos principais conceitos utilizados para o desenvolvimento deste trabalho, que inclui: funcionamento e conceitos importantes relacionados ao robô auto-equilibrado, representações de equações diferenciais através de espaço de estados, noções básicas de mecânica lagrangiana, desenvolvimento do controle por realimentação de estados e criação de observadores de estado.

2.1 ROBÔ AUTO-EQUILIBRADO

Um robô auto-equilibrado é de forma geral constituído por duas rodas fixadas a uma estrutura vertical. O funcionamento se assemelha ao controle vertical de um humano, onde o robô utiliza o movimento das rodas para se manter no ponto de equilíbrio (ADEEL *et al.*, 2013). A dinâmica do robô auto-equilibrado é semelhante a do sistema do pêndulo invertido montado em um carrinho apresentado por Ogata *et al.* (2010), Friedland (2012) e Franklin *et al.* (2015), com a diferença que o movimento da estrutura vertical, o pêndulo invertido, influencia as duas rodas, que agem como se fossem o carrinho do exemplo.

Na maioria dos casos o movimento do robô é feito através de motores de corrente contínua, entretanto, existem alguns casos em que é utilizado um motor de passo para executar essa função (KUNG, 2019). Os sensores mais comumente utilizados nos robôs desenvolvidos são *encoders*, giroscópios e acelerômetros (GRASSER *et al.*, 2002; XU; LI, M.; PAN, 2011; ADEEL *et al.*, 2013; JUANG; LUM, 2013), entretanto em alguns casos são utilizados sensores ultrassônicos ou câmeras para fazer além do controle de estabilidade o controle de trajetória do robô (RUAN; LI, W., 2014; HSU *et al.*, 2018).

O modelo matemático do robô auto-equilibrado é obtido na literatura através das Equações de Euler-Lagrange, das Leis de Newton, ou ainda pelo método de Kane. Embora o resultado encontrado por estes métodos sejam não lineares, muitas vezes ocorre a linearização dos resultados para a simplificação do desenvolvimento do controle(OOI, 2003; CHAN; STOL; HALKYARD, 2013; FRANKOVSKY *et al.*, 2017; ROMLAY *et al.*, 2019).

Por se tratar de uma planta de controle extremamente difundida, existem diversos tipos de controles que já foram testados e discutidos, podemos citar controles mais simples como PID, que acabam apresentando um resultado insatisfatório, com alto sobressinal e oscilação elevada (ADEEL *et al.*, 2013; WARDOYO *et al.*, 2015; NIKITA; PRAJWAL, 2021). Também é comum encontrar modelos linearizados com controle por alocação de polos ou por regulador linear quadrático, que acabam atingindo um resultado melhor em relação ao esperado (GRASSER *et al.*, 2002; RUAN; ZHAO, 2008; XU; LI, M.; PAN, 2011; JUANG; LUM, 2013; WENXIA; WEI, 2017; KIM *et al.*, 2015). Por fim, temos controles mais complexos baseados em lógica difusa, controladores de modo deslizante

e adaptativos que geram resultados bem mais promissores e que permitem o controle do robô em situações adversas (RUAN; CAI, 2009; RUAN; CHEN, 2010; WU; ZHANG, 2011; CHAN; STOL; HALKYARD, 2013; WASIF *et al.*, 2013; QINGCHENG; JIAN, 2014; WARDOYO *et al.*, 2015; ROMLAY *et al.*, 2019).

2.2 REPRESENTAÇÃO POR ESPAÇO DE ESTADOS E MECÂNICA LAGRANGIANA

O primeiro passo no desenvolvimento do controle de um sistema é determinar um modelo matemático que descreve o comportamento dinâmico da planta. Uma das formas de descrever a planta no controle clássico é através da relação entre as entradas e saídas no domínio da frequência, cuja representação é conhecida como função transferência. Já no controle moderno o comportamento da planta é descrito por um conjunto de equações diferenciais de primeira ordem conhecidas como equações do espaço de estados. Esta seção busca apresentar tal representação bem como demonstrar como ela pode ser adquirida através da mecânica lagrangiana (FRIEDLAND, 2012).

A forma padrão mais utilizada na literatura para representar o espaço de estados de um sistema linear tem o seguinte formato (OGATA *et al.*, 2010; FRANKLIN *et al.*, 2015; DORF; BISHOP, 2022):

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{1}$$

Onde \mathbf{x} , \mathbf{u} e \mathbf{y} são os vetores que representam respectivamente as variáveis de estados, as entradas e as saídas do sistema respectivamente. Enquanto os termos \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} são denominados: matriz do sistema, matriz das entradas, matriz das saídas e termo de transmissão direta (FRANKLIN *et al.*, 2015).

Em muitos casos a entrada do sistema não exerce uma influência direta nas saídas, portanto

$$\mathbf{D} = 0,\tag{2}$$

resultando em:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx}\end{aligned}\tag{3}$$

As matrizes \mathbf{A} , \mathbf{B} e \mathbf{C} são extraídas da análise das equações diferenciais que descrevem o comportamento do sistema no tempo. Uma das formas de se encontrar estas equações é através da mecânica lagrangiana, que será abordada a seguir (FRIEDLAND, 2012).

Enquanto na mecânica clássica o estudo da dinâmica de um corpo é feito através da análise das forças e torques, processo que muitas vezes demanda análises de diagramas e contribuições vetoriais, a mecânica lagrangiana tem sua base no princípio de mínima

ação e nos permite determinar as equações que descrevem o comportamento dinâmico do sistema através das energias cinéticas e potenciais presentes nele (GOLDSTEIN; POOLE; SAFKO, 2014; NOLTE, 2019).

O processo se baseia na representação do sistema através de um número de coordenadas generalizadas q igual ao grau de liberdade do sistema. As coordenadas definidas são utilizadas para descrever as parcelas de energia cinética T e energia potencial U e determinar o lagrangiano L (FRIEDLAND, 2012; GOLDSTEIN; POOLE; SAFKO, 2014; NOLTE, 2019):

$$L = T - U \quad (4)$$

As equações diferenciais do sistema são obtidas pela solução das equações de Euler-Lagrange:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (5)$$

Q_i representa as forças (ou torques, no caso de movimento rotacional) generalizadas que afetam as coordenadas q_i mas são externas ao sistema e não podem ser obtidas pela análise da energia potencial (NOLTE, 2019).

A função de dissipação de Rayleigh nos permite adicionar à equação de Euler-Lagrange as perdas causadas por atrito (GOLDSTEIN; POOLE; SAFKO, 2014), de modo que:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i - \frac{\partial \mathcal{F}}{\partial q_i} \quad (6)$$

Com:

$$\mathcal{F} = \frac{1}{2} \sum_i (k_x v_i x^2 + k_y v_i y^2 + k_z v_i z^2) \quad (7)$$

O resultado das equações de Euler-Lagrange serão equações diferenciais que descrevem a dinâmica do sistema em função das coordenadas q_i e das forças Q_i (FRIEDLAND, 2012). Estas equações podem ser convertidas para o formato de espaço de estados, linearizados e escritas no formato da Equação (3).

2.3 DESENVOLVIMENTO DO CONTROLADOR POR REALIMENTAÇÃO DE ESTADOS

Após o terem sido determinadas as equações da dinâmica do sistema no espaço de estados, a próxima etapa é desenvolver o controlador do sistema. Esta seção visa apresentar as etapas e conceitos básicos no desenvolvimento de um controlador por realimentação de estados.

Algumas técnicas de controle requerem que o sistema seja completamente controlável e/ou completamente observável, portanto esta acaba sendo uma das primeiras características que devem ser verificadas no desenvolvimento do controlador.

As ideias de controlabilidade e observabilidade foram introduzidas por Kalman nos anos 50, que dita o seguinte: um sistema é controlável se uma sequência de entradas for capaz de transferir o sistema de um estado inicial para qualquer outro estado (final) em um período de tempo finito; por outro lado, um sistema é dito observável se o seu estado inicial puder ser calculado através de uma série de medidas feitas em um tempo finito (FRIEDLAND, 2012; PHILLIPS; NAGLE; CHAKRABORTTY, 2015; DORF; BISHOP, 2022).

Como apresentado por Dorf e Bishop (2022), a controlabilidade e observabilidade de um sistema podem ser verificadas pela análise das matrizes de controlabilidade \mathbf{P}_c e de observabilidade \mathbf{P}_o :

$$\mathbf{P}_c = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix} \quad (8)$$

$$\mathbf{P}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix} \quad (9)$$

Caso o posto da matriz \mathbf{P}_c seja igual ao número de linhas dela mesma, então o sistema é completamente controlável, e se o posto da matriz \mathbf{P}_o for igual ao número de colunas dela mesma, o sistema é dito completamente observável(DORF; BISHOP, 2022).

O controle de realimentação de estados surge do princípio de que se um sistema é completamente controlável, então existe um vetor de ganho \mathbf{K} que ao ser usado na realimentação das variáveis de estado consegue realocar os polos do sistema para qualquer valor. Ao considerar a referência como zero, a ação de controle é calculada como (OGATA *et al.*, 2010; PHILLIPS; NAGLE; CHAKRABORTTY, 2015; DORF; BISHOP, 2022):

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (10)$$

Existem duas formas bastante utilizadas para determinar o ganho \mathbf{K} : alocação de polos e regulador linear quadrático. Quando a localização dos polos desejados é conhecida ou facilmente determinada pelos parâmetros de projeto convém utilizar a alocação de polos. Entretanto, quando esse não é o caso, o ganho do controlador pode ser obtido através do regulador linear quadrático(LQR) pela escolha dos parâmetros \mathbf{Q} e \mathbf{R} e da minimização do índice de desempenho (FRIEDLAND, 2012; PHILLIPS; NAGLE; CHAKRABORTTY, 2015).

No primeiro método o ganho \mathbf{K} é determinado pela comparação entre a equação característica do sistema em malha fechada com a equação característica que contém os polos desejados s_i (OGATA *et al.*, 2010):

$$|s\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}| = (s - s_1)(s - s_2) \cdots (s - s_n) \quad (11)$$

A solução da Equação (11) nos retorna a matriz de ganho K que transfere os polos do sistema para a posição desejada.

O segundo método é dito ótimo, já que ele busca chegar ao melhor controle encontrando um ganho \mathbf{K} que minimize o índice de desempenho J a partir dos parâmetros \mathbf{Q} e \mathbf{R} , fatores relacionados ao comportamento das variáveis de estados e das entradas respectivamente. O índice de desempenho do regulador linear quadrático é (OGATA *et al.*, 2010; FRANKLIN *et al.*, 2015; CASTRUCCI; BITTAR; SALES, 2018):

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (12)$$

Quando trabalhamos com realimentação de estados, o índice de desempenho pode ser minimizado através da solução da equação de Riccati (OGATA *et al.*, 2010):

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0 \quad (13)$$

Onde o ganho \mathbf{K} que minimiza o índice J é (OGATA *et al.*, 2010):

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (14)$$

\mathbf{Q} e \mathbf{R} são parâmetros de projeto que determinam a rapidez de convergência das variáveis de estado e energia gasta durante o controle, respectivamente. A determinação da matriz \mathbf{P} através da Equação (13) permite com que o controlador seja determinado pela Equação (14), em muitos casos métodos computacionais são utilizados para se chegar a essa solução (PHILLIPS; NAGLE; CHAKRABORTTY, 2015; CASTRUCCI; BITTAR; SALES, 2018).

Quando se trata de sistemas amostrados ou digitais, os métodos de alocação de polos e de regulação linear quadrática são análogos em comparação ao utilizado em sistemas contínuos, as mudanças a serem consideradas são as seguintes (PHILLIPS; NAGLE; CHAKRABORTTY, 2015; CASTRUCCI; BITTAR; SALES, 2018; AGUIRRE, 2019):

A Equação (11) é escrita em relação aos polos discretos z_i :

$$|z\mathbf{I} - \mathbf{A}_d + \mathbf{B}_d \mathbf{K}| = (z - z_1)(z - z_2) \cdots (z - z_n) \quad (15)$$

O índice de desempenho, a equação de Riccati e o ganho do controlador para o regulador linear quadrático em sistemas discretos tornam-se (CASTRUCCI; BITTAR; SALES, 2018):

$$J = \frac{1}{2} \sum_{k=0}^{\infty} (\mathbf{x}^\top(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^\top(k) \mathbf{R} \mathbf{u}(k)) \quad (16)$$

$$\begin{aligned} \mathbf{P} &= \mathbf{A}_d^\top \left(\mathbf{P} - \mathbf{P} \mathbf{B}_d \mathbf{W}^{-1} \mathbf{B}_d^\top \mathbf{P} \right) \mathbf{A}_d \\ \mathbf{W} &= \mathbf{R} + \mathbf{B}_d^\top \mathbf{P} \mathbf{B}_d \end{aligned} \quad (17)$$

$$\mathbf{K} = \mathbf{W}^{-1} \mathbf{B}_d^\top \mathbf{P} \mathbf{A}_d \quad (18)$$

2.4 IMPLEMENTAÇÃO DE OBSERVADORES

O processo de desenvolvimento de um controlador de realimentação de estados se baseia na hipótese de que todos as variáveis de estados são medidas, entretanto, quando esse não é o caso, o princípio da separação permite com que um observador de estados seja desenvolvido em paralelo para estimar as variáveis que não podem ser medidas diretamente (FRANKLIN *et al.*, 2015). Esta seção visa apresentar o processo de desenvolvimento do observador de estados e como ele pode ser implementado em controladores digitais.

Por definição podemos ter um observador de ordem plena, onde todas as variáveis de estado do sistema são estimadas, e um observador de ordem mínima, onde as variáveis medidas são confiáveis o suficiente para que apenas as variáveis restantes precisem ser estimadas (OGATA *et al.*, 2010). Como será visto a seguir, a cálculo dos dois diferentes observador é feito de maneira bem similar.

A equação que representa o observador de ordem plena é semelhante à representação no espaço de estados do sistema (Equação (3)), entretanto surge um ganho \mathbf{L} , cujo objetivo é garantir que os estados estimados $\hat{\mathbf{x}}$ converjam para os reais de forma rápida, através do controle do erro entre as saídas estimadas $\hat{\mathbf{y}}$ e as saídas reais \mathbf{y} (OGATA *et al.*, 2010). A dinâmica do observador é descrita pela seguinte equação:

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{A} \hat{\mathbf{x}} + \mathbf{B} \mathbf{u} \\ \hat{\mathbf{y}} &= \mathbf{C} \hat{\mathbf{x}} \end{aligned} \quad (19)$$

O erro \mathbf{e} entre as variáveis estimadas e as variáveis reais pode ser encontrado pela subtração da Equação (3) com a Equação (19):

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{LC})\mathbf{e} \quad (20)$$

Com base na Equação (20) fica claro que o processo de determinação do ganho \mathbf{L} é o mesmo do ganho de realimentação de estados \mathbf{K} , onde agora o objetivo é alocar os polos da função característica $|s\mathbf{I} - \mathbf{A} + \mathbf{LC}|$, ou $|z\mathbf{I} - \mathbf{A}_d + \mathbf{LC}_d|$ para um observador discreto, de forma com que o erro se estabilize em zero rápido o suficiente para o controle ser realizado corretamente. Via de regra a dinâmica do observador deve ser de duas a quatro

vezes mais rápida que a dinâmica do controle por realimentação de estados (CASTRUCCI; BITTAR; SALES, 2018).

Em casos em que as saídas podem ser medidas com precisão e a presença de ruídos é pequeno, o uso de um observador de ordem plena se torna redundante e podemos desenvolver um observador de ordem mínima, que garante resultados similares, mas de forma mais simples (OGATA *et al.*, 2010). Para construir o observador de ordem mínima, primeiro o espaço de estados do sistema deve ser escrito no seguinte formato:

$$\begin{aligned} \begin{bmatrix} \dot{x}_a \\ \dot{x}_b \end{bmatrix} &= \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} B_a \\ B_b \end{bmatrix} u \\ \mathbf{y} &= [\mathbf{I} \quad \mathbf{0}] \begin{bmatrix} x_a \\ x_b \end{bmatrix} \end{aligned} \quad (21)$$

Onde \mathbf{x}_a e \mathbf{x}_b são as variáveis que podem ser medidas e as variáveis que devem ser estimadas pelo observador respectivamente. Como apresentado por Ogata *et al.* (2010), existe uma conversão direta entre o observador de ordem mínima e o observador de ordem plena, de modo que o erro do observador de ordem mínima é dado por:

$$\dot{\mathbf{e}} = (\mathbf{A}_{bb} - \mathbf{L}\mathbf{A}_{ab})\mathbf{e} \quad (22)$$

Os observadores de ordem plena e ordem mínima podem ser implementados em controladores digitais através de suas respectivas equações a diferenças:

$$\hat{\mathbf{x}}(k+1) = (\mathbf{A}_d - \mathbf{L}\mathbf{C}_d)\hat{\mathbf{x}}(k) + \mathbf{B}_d\mathbf{u}(k) + \mathbf{L}\mathbf{y}(k) \quad (23)$$

$$\hat{\mathbf{x}}_b(k+1) = (\mathbf{A}_{bb} - \mathbf{L}\mathbf{A}_{ab})\hat{\mathbf{x}}(k) + (\mathbf{B}_b - \mathbf{L}\mathbf{B}_a)\mathbf{u}(k) + \mathbf{L}\mathbf{y}(k+1) + (\mathbf{A}_{ba} - \mathbf{L}\mathbf{A}_{aa})\mathbf{y}(k) \quad (24)$$

Onde a Equação (23) é referente ao observador de ordem plena e a Equação (24) é referente ao observador de ordem mínima.

3 ROBÔ AUTO-EQUILIBRADO

Com o intuito de implementar técnicas de controle moderno em um sistema real e obter resultados práticos, um robô auto-equilibrado foi desenvolvido, o que nada mais é que uma planta naturalmente instável e não linear. Este capítulo procura apresentar os principais componentes que constituem o robô, apontando o funcionamento e o motivo por trás de cada componente e como eles interagem entre si.

Os atuadores escolhidos para movimentar as rodas do robô foram motores de corrente contínua, enquanto os sensores escolhidos foram *encoders* presos aos motores, giroscópios e acelerômetros, os quais são a combinação de componentes mais comumente encontrada na literatura. O robô necessita de um microcontrolador que consiga adquirir os dados dos sensores, calcular a ação de controle e controlar os motores. Também é preciso uma fonte de energia que consiga alimentar todos os componentes do projeto.

3.1 MOTORES

Foram escolhidos dois motores de corrente contínua com caixa de redução e *encoder* incremental acoplado. A redução garante que o motor produzirá torque o suficiente para movimentar o corpo do robô, e os componentes estão acoplados ao motor de tal forma que se torna um conjunto bem compacto, facilitando a montagem e garantindo um bom aproveitamento de espaço. O motor pode ser visto na Figura 1, enquanto as principais características do motor são apresentadas na Tabela 1.

Figura 1 – Motor de corrente contínua.



Fonte: Baú da Eletrônica (2022).

O *encoder* incremental do motor envia sinais pulsados através dos canais A e B, que

Tabela 1 – Características do Motor.

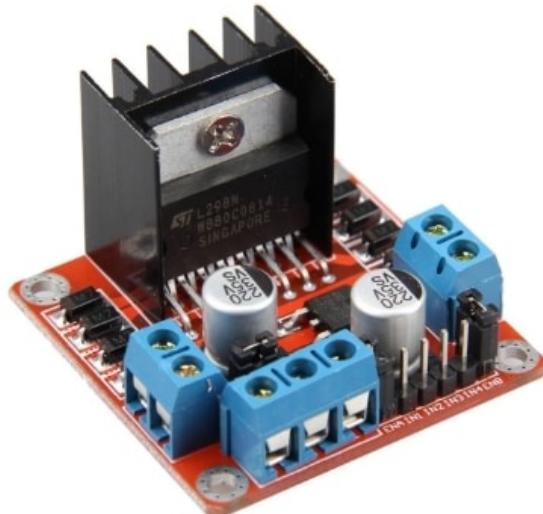
Tensão Nominal	6V
Motor em Vazio	210 rpm/0,13 A
Eficiência Máxima	2,0 kgf·cm/170 rpm/2,0 W/0,60 A
Potência Máxima	5,2 kgf·cm/110 rpm/3,1 W/1,10 A
Torque de Parada	10 kgf·cm/3,20 A
Caixa de Redução	34:1
Resolução do <i>Encoder</i>	374 PPR

Fonte: Adaptado de Baú da Eletrônica (2022).

permitem determinar a mudança no ângulo do motor. O ângulo do motor é determinado com base na mudança medida através do *encoder*. Esta característica faz com que a leitura do *encoder* e o cálculo do ângulo seja feito constantemente (DELTA COMPUTER SYSTEMS, 2022; DYNAPAR, 2022).

O controle da velocidade dos motores é feito com o auxílio do módulo de ponte H baseado no chip L298n, que permite o controle do sentido e da velocidade de rotação dos dois motores através de sinais PWM. Foi optado pelo uso do módulo pronto já que ele simplifica a montagem do robô e garante um funcionamento confiável (FILIPEFLOP, 2022b). O modulo é apresentado na Figura 2.

Figura 2 – Ponte H.



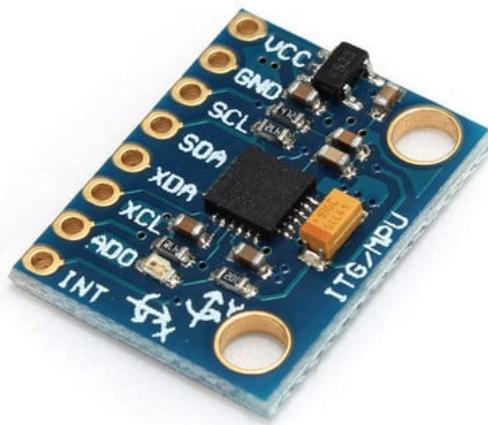
Fonte: FILIPEFLOP (2022b).

3.2 GIROSCÓPIO E ACELERÔMETRO

Visando monitorar o ângulo do robô foi implementado o MPU-6050, um sensor que une um giroscópio e um acelerômetro de 3 eixos com um microprocessador. O principal

motivo por trás da escolha do sensor foi a alta quantidade de vezes que ele aparece na literatura sendo utilizado no robô auto-equilibrado. O sensor apresenta bons resultados e é confiável para a aplicação proposta neste trabalho. O dispositivo é responsável por coletar os sinais analógicos de velocidade angular e aceleração dos sensores, converter através de um DAC de 16 bits para digital e enviar os dados por I2C (INVENSENSE INC., 2013). O sensor pode ser visto na Figura 3.

Figura 3 – MPU-6050.



Fonte: FILIPEFLOP (2022a).

Tanto o acelerômetro quanto o giroscópio podem ser utilizados para se chegar ao ângulo do robô. O acelerômetro consegue determinar o ângulo através do monitoramento da direção da aceleração da gravidade, entretanto a medição está exposta a erros causados por acelerações geradas pelo movimento do carinho. Por outro lado, o giroscópio gera uma velocidade angular que pode ser integrada no tempo para se chegar ao ângulo, porém este método acaba acumulando erros e o valor desvia-se do real com o passar do tempo.

Ao serem utilizados separadamente, o acelerômetro tem uma boa resposta para longos períodos de tempo, enquanto o giroscópio entrega uma resposta melhor para curtos períodos de tempo. Ao fundirmos ambos os sensores conseguimos atingir um resultado melhor, unindo as vantagens de cada sensor, o que pode ser feito através da implementação de um filtro de Kalman nas medidas entregues pelo MPU-6050. (JUAN RINCÓN PASAYE; ALBERTO BONALES VALENCIA; JIMÉNEZ PÉREZ, 2013; PALANISAMY, 2016).

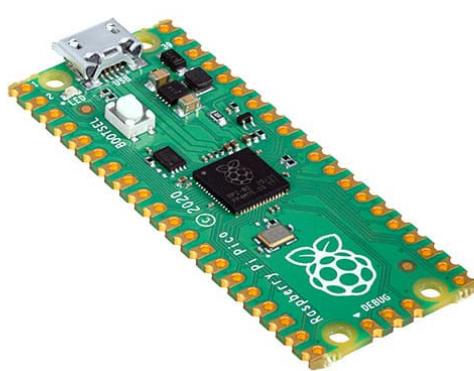
3.3 MICROCONTROLADOR

Neste projeto o microcontrolador deve exercer as funções de processar os sinais obtidos pelos sensores, realizar os cálculos necessários para o controle e enviar a ação

de controle para a ponte H. O microcontrolador também é responsável por armazenar os dados dos testes de campo e transmiti-los por comunicação serial para serem feitos análises dos resultados práticos.

Pensando nisso foi escolhido a placa baseada no microcontrolador RP2040, o Raspberry Pi Pico. As suas capacidades de processamento e de armazenamento atendem as necessidades do projeto, e a placa se encontrava disponível para uso. A placa possui as seguintes características físicas: Memória Flash de 2MB, Memória RAM de 256kB e dois núcleos ARM Cortex M0+ com um *clock* de até 133MHz. Dentre as diversas funções que a placa possui as mais relevantes para esse projeto são a comunicação I2C, os canais PWM de 16 bits e os temporizadores internos (RASPBERRY PI LTD, 2022b). A placa pode ser vista na Figura 4.

Figura 4 – Raspberry Pi Pico.



Fonte: FILIPEFLOP (2022c).

3.4 FONTE DE ALIMENTAÇÃO

Para alimentar todos os componentes eletrônicos do robô foi desenvolvido uma placa de circuito impresso que une um caixa para três baterias de Li-ion, uma placa de proteção para carga de baterias BMS 3S 20A e um conector *jack*. A placa criada garante o carregamento e descarregamento das baterias de forma segura. As três baterias de Li-ion em série fornecem uma tensão de 12.6V em média, entretanto os componentes utilizados necessitam de uma tensão menor. Para reduzir a tensão para a faixa desejada e também garantir a estabilidade na alimentação do sistema foi utilizado um módulo *stepdown* ajustável. A placa criada e o modulo mencionado podem ser vistos respectivamente na Figura 5 e na Figura 6.

Figura 5 – Placa de carregamento.



Fonte: O Autor.

Figura 6 – *Stepdown*.

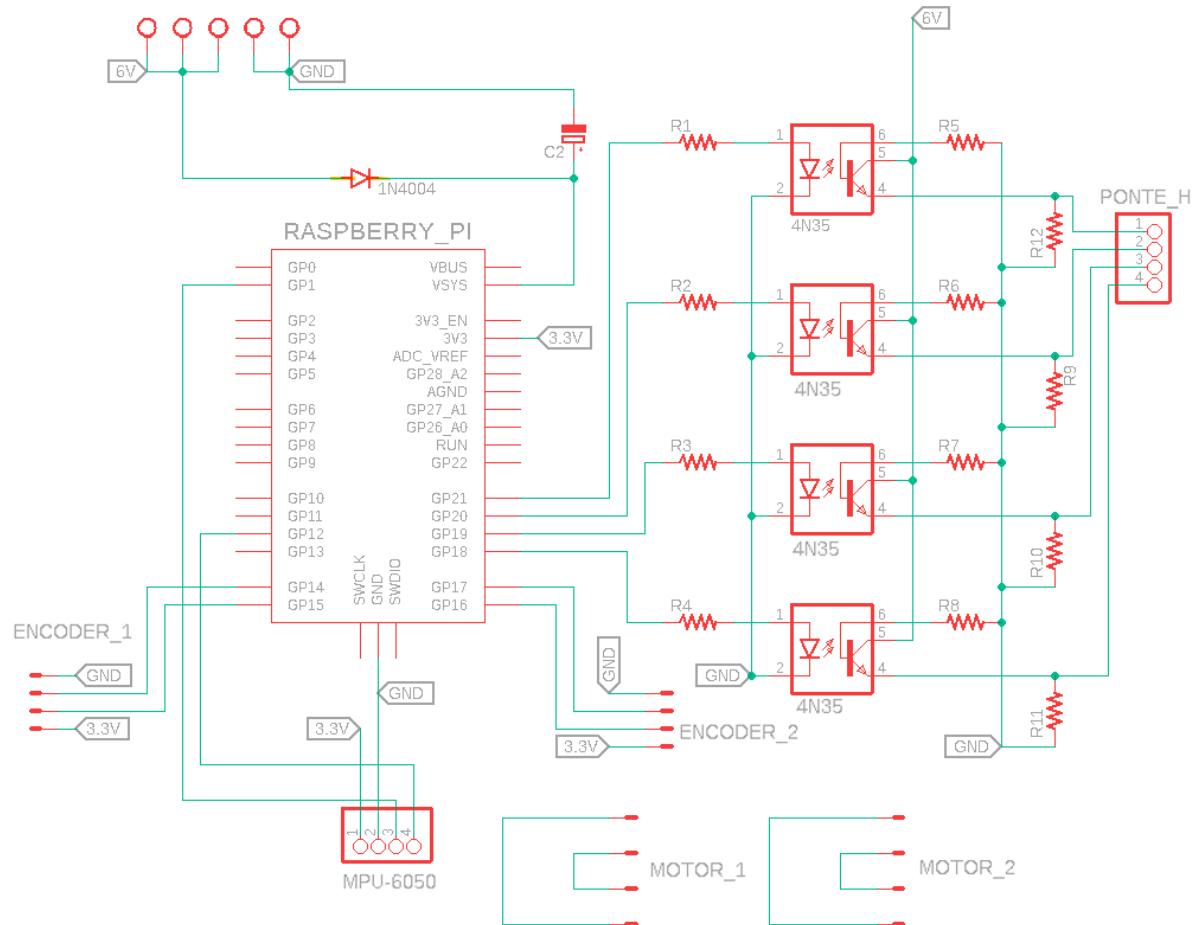
Fonte: Mercado Livre (2022).

3.5 ESTRUTURA

Para facilitar a conexão entre os componentes do robô foi criado uma placa com conectores, bornes e pinos. Na placa criada também está presente um circuito com optoacopladores para isolar as saídas do Raspberry Pi Pico, um diodo para proteger contra inversão na alimentação e um capacitor para filtrar possíveis ruídos presentes na alimen-

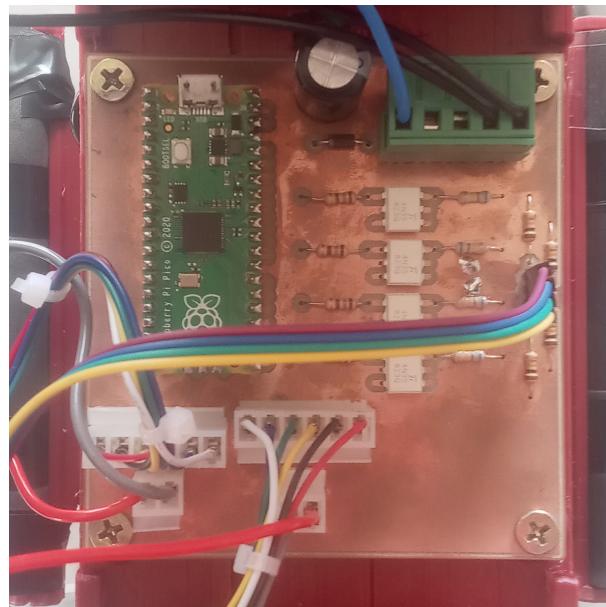
tação do microcontrolador. O diagrama elétrico da placa e a mesma podem ser vistos na Figura 7 e Figura 8 respectivamente.

Figura 7 – Diagrama da placa de conexão.



Fonte: O Autor.

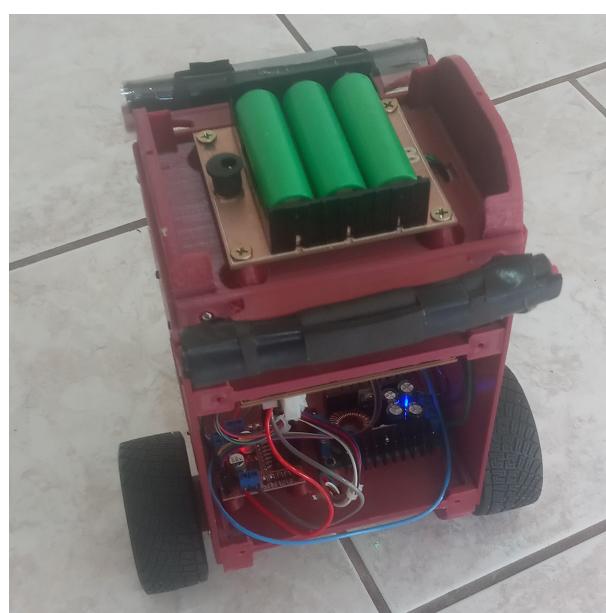
Figura 8 – Placa de conexão.



Fonte: O Autor.

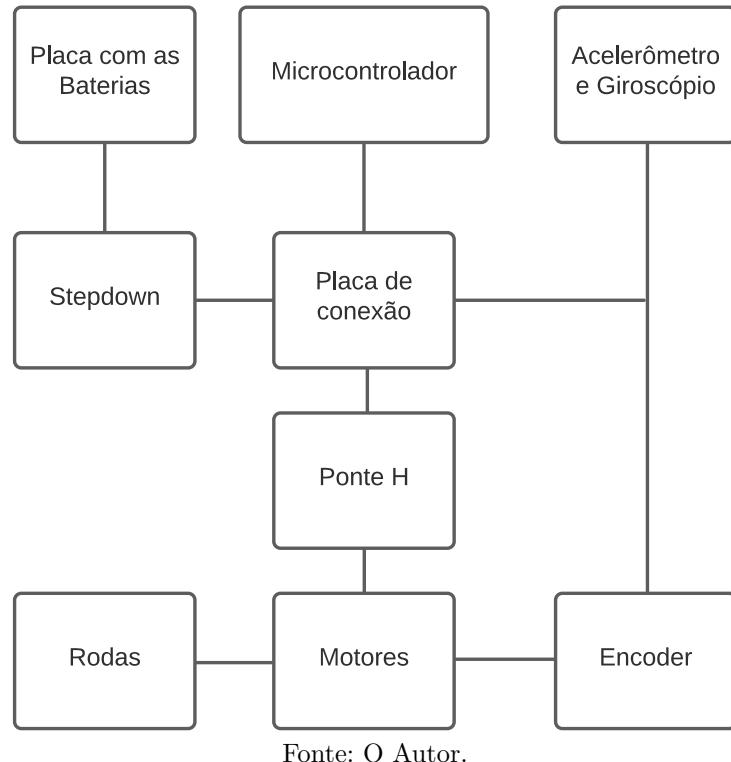
Baseado em outros projetos presentes na literatura, foi desenhado e impresso em 3d peças para o suporte de todos os componentes, de forma com que eles possam ser parafusados ou encaixados na estrutura com facilidade. O robô foi projetado para ter um centro de massa distante do chão, visando aumentar a instabilidade do sistema e dificultar o desenvolvimento do controle. O robô construído e como os seus componentes se interligam podem ser vistos na Figura 9 e Figura 10 respectivamente.

Figura 9 – Robô auto-equilibrado.



Fonte: O Autor.

Figura 10 – Diagrama da interligação dos componentes do robô.



Fonte: O Autor.

3.6 ALGORITMO

Existem diversas formas de programar a placa Raspberry Pi Pico, entretanto para este projeto foi escolhido a programação através do pacote de desenvolvimento do fabricante o qual permite que a programação seja feita em linguagem C. O pacote de desenvolvimento facilita a programação através de inúmeras bibliotecas desenvolvidas especificamente para trabalhar com o RP2040, ao mesmo tempo que garante ao programador um alto controle dos parâmetros e funcionamentos do microcontrolador, permitindo a criação de algoritmos mais específicos e eficientes (RASPBERRY PI LTD, 2022a).

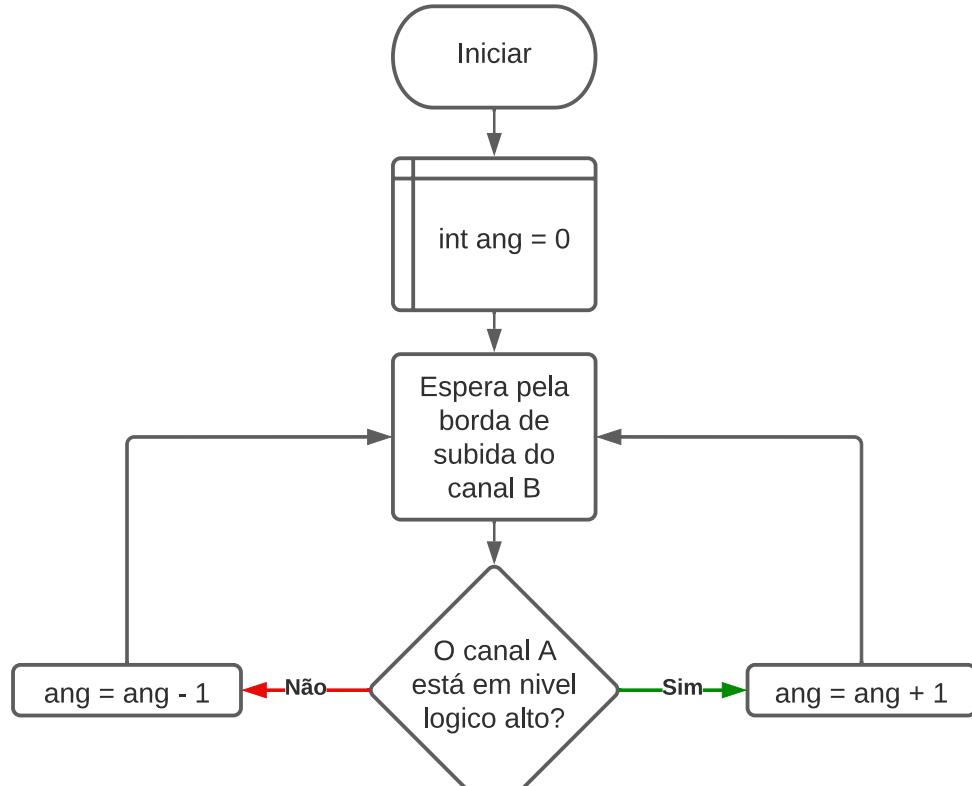
As principais funções do programa criado para este projeto são:

- Monitorar os valores enviados pelo *encoder* a fim de se ter o ângulo atual do motor.
- Inicializar e adquirir os dados do sensor MPU-6050 por I2C e determinar o ângulo do robô através do filtro de Kalman.
- Calcular a ação de controle com base nos estados medidos e enviar o resultado para a ponte H através do PWM.

Como o monitoramento do ângulo do motor requer que o microcontrolador esteja constantemente contando a quantidade de pulsos enviados pelo *encoder*, um dos núcleos foi separado exclusivamente para esta função, garantindo assim a precisão na medida do ângulo atual do motor. O funcionamento da medição do ângulo do motor é demonstrado

pelo fluxograma da Figura 11.

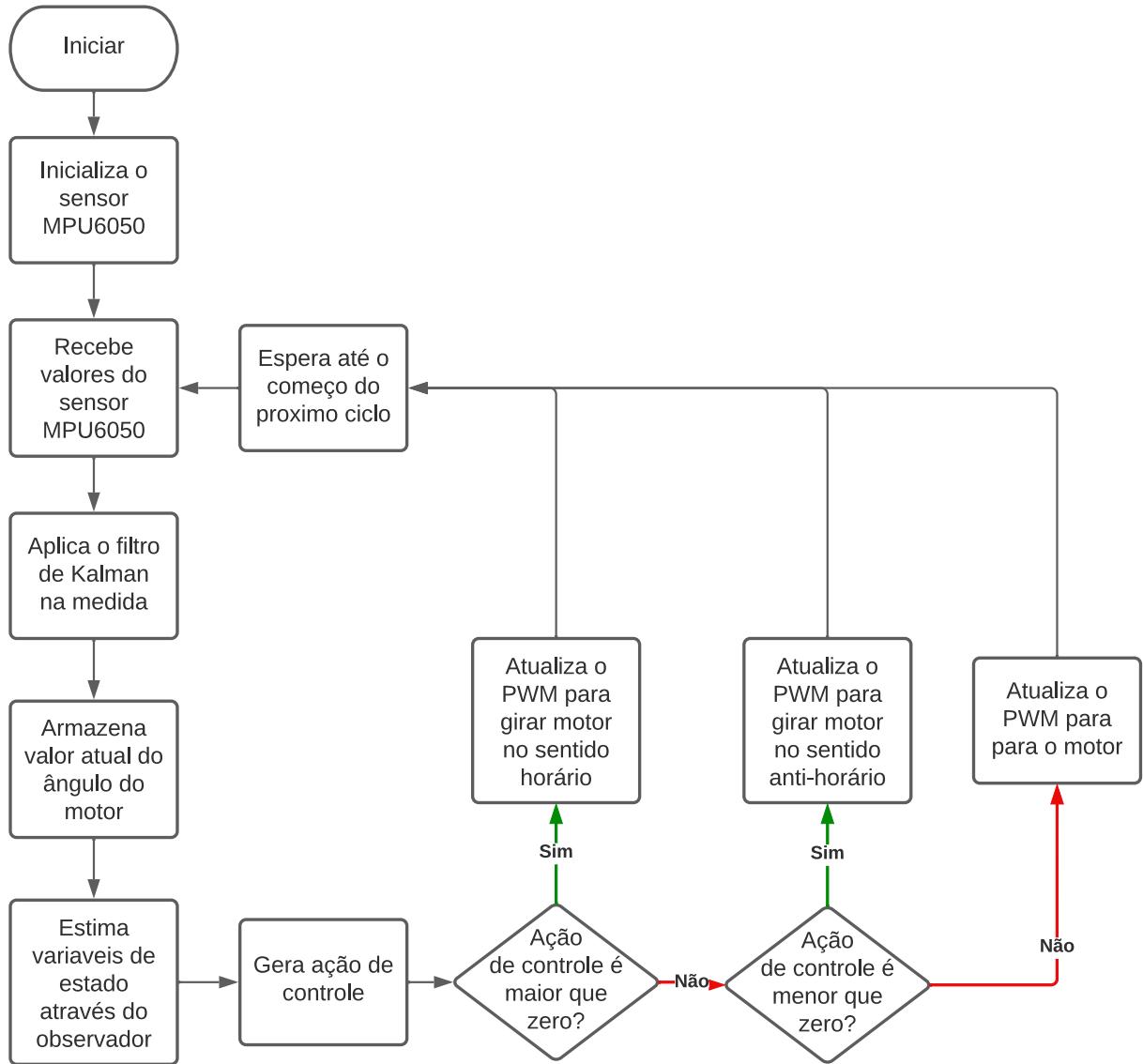
Figura 11 – Processo de medição do ângulo do motor.



Fonte: O Autor.

O programa utiliza interrupções de tempo para garantir que a aquisição dos dados e o controle do motor seja em períodos constantes. A cada ciclo o algoritmo armazena o ângulo atual do motor em uma variável, faz a comunicação com o sensor para adquirir os valores de aceleração e velocidade angular, e então filtra o sinal através de um algoritmo adaptado de uma biblioteca do filtro de Kalman para o sensor MPU-6050 (CHRISTOPHER, 2020), as variáveis de estado são então atualizadas pelo observador e utilizadas pelo controlador desenvolvido e por fim a ação de controle é enviada por PWM. O funcionamento do seguinte processo é exemplificado na Figura 12.

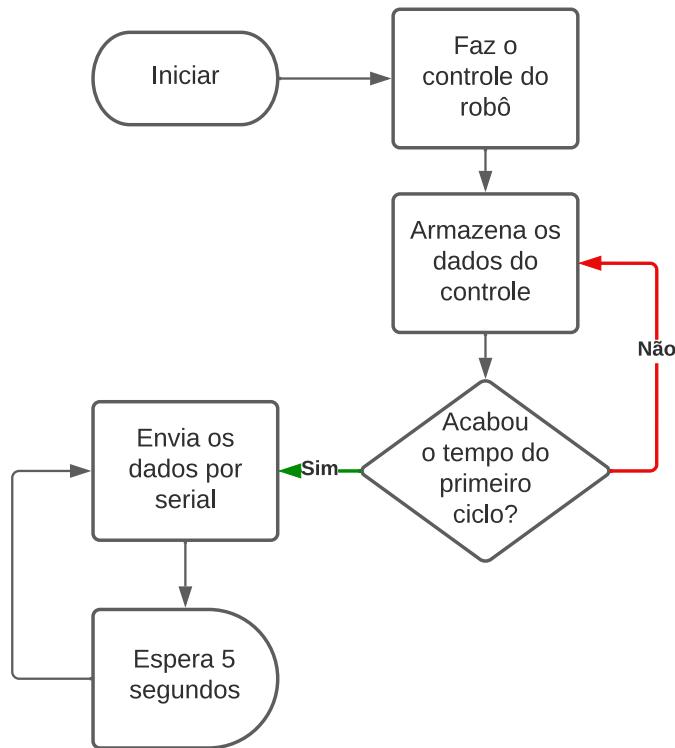
Figura 12 – Processo de controle do robô.



Fonte: O Autor.

Para examinar o comportamento do robô, o programa também tem a função de armazenar os dados e enviá-los por serial para serem analisados posteriormente através da GNU Octave. Para que isso fosse possível o programa foi dividido em dois ciclos de operação, no primeiro ciclo o robô mantém o seu comportamento normal enquanto armazena os dados obtidos em uma matriz, após um certo período de tempo ele entra no segundo ciclo, onde ele envia os dados armazenados por comunicação serial. Este funcionamento está exemplificado pelo fluxograma da Figura 13.

Figura 13 – Ciclo de trabalho do robô.



Fonte: O Autor.

4 CARACTERIZAÇÃO DO SISTEMA

Um dos pontos que mais influenciam no resultado atingido com o desenvolvimento do controlador é a precisão do modelo obtido para a planta a ser controlada. Este capítulo busca relatar os métodos utilizados para encontrar um modelo que descreva de forma satisfatória o robô auto-equilibrado.

De forma geral a caracterização de um sistema pode ser feita puramente por modelagens matemáticas e pelo entendimento do comportamento físico do sistema, esta forma é conhecida como método caixa branca. O sistema também pode ser identificado com base em resultados experimentais, caracterizando-se como método caixa preta. Ainda podem ser usados conceitos e equações teóricas unidos a resultados práticos de modo a se chegar a uma melhor aproximação, resultando assim em um método caixa cinza (ISERMANN; MÜNCHHOF, 2011; TANGIRALA, 2018).

Como o sistema a ser trabalhado é instável, é complexo chegar a um resultado por caixa preta, entretanto certos parâmetros necessários para o desenvolvimento do modelo não estão disponíveis para medição. Com isso em mente a metodologia a ser adotada aqui será a de caixa cinza, onde primeiro as equações que regem o sistema serão determinadas e por fim os parâmetros desconhecidos serão obtidos experimentalmente.

4.1 EQUAÇÕES DO SISTEMA

Esta seção foca em apresentar o desenvolvimento realizado e as considerações feitas para se chegar ao espaço de estados genérico do robô auto-equilibrado. Embora diversos modelos já estejam disponíveis na literatura, eles são complexos demais para o problema de equilíbrio, ou não consideram fatores importantes como a influência da caixa de redução e/ou o atrito entre a roda e o chão, por conta disso equações mais satisfatórias serão determinadas aqui.

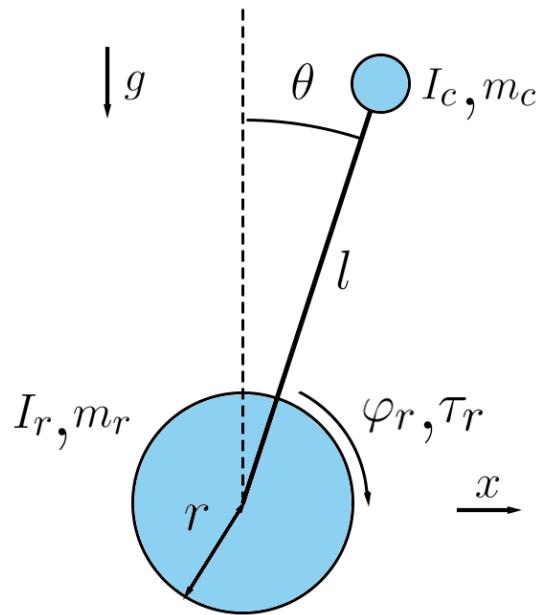
4.1.1 Equação do Robô

Para descrever o movimento do robô de maneira mais simplificada iremos partir dos seguintes pressupostos:

- Como o controle pretende apenas estabilizar o robô, podemos admitir que a movimentação será apenas bidimensional e o movimento das duas rodas serão iguais.
- O robô pode ser dividido em duas partes: O corpo do robô C e as rodas R
- Não existe deslizamento no movimento da roda

Com isso em mente podemos construir o diagrama de corpo livre da Figura 14.

Figura 14 – Diagrama de corpo livre.



Fonte: O Autor.

Sendo que:

- θ : ângulo do corpo do robô em relação ao eixo vertical
- m_c : massa do corpo do robô
- I_c : momento de inércia do corpo do robô
- φ_r : ângulo da roda
- τ_r : torque gerado pela rotação da roda
- m_r : massa da roda
- I_r : momento de inércia da roda
- x : posição da roda no eixo horizontal
- r : raio da roda
- l : distância entre o centro da roda e o centro de massa do robô
- g : aceleração da gravidade

Como não existe deslizamento:

$$x = r\varphi_r \quad (25)$$

Como visto na Seção 2.1, uma das formas de se chegar às equações que descrevem o movimento do robô é utilizando as equações de Euler-Lagrange explicadas na Seção 2.2.

A energia cinética presente no corpo T_c e a energia cinética presente na roda T_r podem ser definidas como a soma da energia cinética rotacional com a translacional:

$$T_c = \frac{I_c \dot{\theta}^2}{2} + \frac{m_c v_c^2}{2} \quad (26)$$

$$T_r = \frac{I_r \dot{\varphi}^2}{2} + \frac{m_r \dot{x}_c^2}{2} \quad (27)$$

A velocidade do corpo do robô v_c pode ser reescrita em função da velocidade horizontal do corpo \dot{x}_c e da velocidade vertical do corpo \dot{z}_c :

$$v_c = \sqrt{\dot{x}_c^2 + \dot{z}_c^2} \quad (28)$$

Como o corpo está preso à roda podemos reescrever as coordenadas x_c e z_c como:

$$x_c = x + l \sin \theta \quad (29)$$

$$x_c = l \cos \theta \quad (30)$$

Substituindo as Equações (29) e (30) na Equação (28):

$$v_c = \sqrt{\dot{x}^2 + 2\dot{x}l \cos \theta \dot{\theta} + l^2 \dot{\theta}^2} \quad (31)$$

Substituindo a Equação (31) na Equação (26):

$$T_c = \frac{I_c \dot{\theta}^2}{2} + \frac{m_c (\dot{x}^2 + 2\dot{x}l \cos \theta \dot{\theta} + l^2 \dot{\theta}^2)}{2} \quad (32)$$

Substituindo a Equação (25) na Equação (27):

$$T_r = \frac{I_r \dot{x}^2}{2r^2} + \frac{m_r \dot{x}_c^2}{2} \quad (33)$$

A energia potencial na roda é nula e no corpo é definida pela gravitacional:

$$U_c = m_c g z_c = m_c g l \cos \theta \quad (34)$$

O lagrangiano da Equação (4) é:

$$L = T_c + 2T_r - U_c = \dot{x}^2 \left(\frac{m_c}{2} + m_r + \frac{I_r}{r^2} \right) + \dot{\theta}^2 \left(\frac{I_c + m_c l^2}{2} \right) + m_c l \cos \theta (\dot{x} \dot{\theta} - g) \quad (35)$$

A energia dissipada devido ao atrito das duas rodas com o chão pode ser determinada pela Equação (7):

$$\mathcal{F} = b_r \dot{\varphi}_r^2 = \frac{b_r \dot{x}^2}{r^2} \quad (36)$$

As força externas são a força que movimenta as rodas para a coordenada x e o torque de reação gerado pelo movimento da roda para a coordenada θ :

$$Q_x = \frac{2\tau_r}{r} \quad (37)$$

$$Q_x = -2\tau_r \quad (38)$$

Através de Equação (6):

$$\begin{aligned} \ddot{x} \left(m_c + 2m_r + \frac{2I_r}{r^2} \right) + \dot{x} \frac{2b_r}{r^2} + \ddot{\theta} m_c l \cos \theta - \dot{\theta}^2 m_c l \sin \theta &= \frac{2\tau_r}{r} \\ \ddot{x} m_c l \cos \theta + \ddot{\theta} \left(I_c + m_c l^2 \right) - m_c g l \sin \theta &= -2\tau_r \end{aligned} \quad (39)$$

As equações encontradas são não lineares, entretanto podemos linearizar o sistema para facilitar o desenvolvimento do controle. Assumiremos que durante o controle do robô auto-equilibrado o ângulo θ se manterá pequeno e haverá pequenas variações no ângulo, resultado em uma velocidade $\dot{\theta}$ também pequena. Com essas considerações podemos dizer que (FRIEDLAND, 2012):

$$\begin{aligned} \sin \theta &= \theta \\ \cos \theta &= 1 \\ \dot{\theta}^2 &= 0 \end{aligned} \quad (40)$$

Linearizando a Equação (39) através da Equação (40), obtemos a equação que descreve o comportamento do robô auto-equilibrado:

$$\begin{aligned} \ddot{x} \left(m_c + 2m_r + \frac{2I_r}{r^2} \right) + \dot{x} \frac{2b_r}{r^2} + \ddot{\theta} m_c l &= \frac{2\tau_r}{r} \\ \ddot{x} m_c l + \ddot{\theta} \left(I_c + m_c l^2 \right) - m_c g l \theta &= -2\tau_r \end{aligned} \quad (41)$$

4.1.2 Equação do Motor

De acordo com Franklin *et al.* (2015), o motor de corrente contínua com caixa de redução pode ser descrito por:

$$(J_2 + J_1 n^2) \ddot{\varphi} + (b_2 + b_1 n^2) \dot{\varphi} = n \tau_m \quad (42)$$

Onde:

- φ : ângulo do eixo do motor
- J_2 : momento de inércia da engrenagem maior
- J_1 : momento de inércia da engrenagem menor
- b_2 : amortecimento da engrenagem maior
- b_1 : amortecimento da engrenagem menor

- τ_m : torque gerado pelo motor
- n : relação da caixa de redução

O torque gerado pelo motor é expresso por:

$$\tau_m = k_t i_a \quad (43)$$

Sendo k_t e i_a a constante de torque do motor e a corrente na armadura respectivamente.

A relação entre as grandezas elétricas no motor de corrente contínua é:

$$L_a \dot{i}_a + R_a i_a = V_a - \varepsilon \quad (44)$$

Sendo L_a , R_a e V_a a indutância, a resistência e a tensão na armadura respectivamente e ε a força contra eletromotriz gerada no motor.

A força contra eletromotriz é definida pela constante k_e e pela velocidade do rotor $\dot{\varphi}_m$:

$$\varepsilon = k_e \dot{\varphi}_m \quad (45)$$

Através da relação da caixa de redução chegamos a:

$$\varepsilon = k_e n \dot{\varphi} \quad (46)$$

Utilizando a Equação (46) na Equação (44) e partindo do pressuposto que a indutância na armadura é desprezível por se tratar de um motor pequeno (OOI, 2003):

$$i_a = \frac{V_a - k_e n \dot{\varphi}}{R_a} \quad (47)$$

Juntando as Equações (42), (43) e (44), resulta na equação diferencial que relaciona o ângulo do eixo do motor com a tensão aplicada na armadura do motor:

$$(J_2 + J_1 n^2) \ddot{\varphi} + \left(b_2 + b_1 n^2 + \frac{k_e k_t}{R_a} n^2 \right) \dot{\varphi} = \frac{n k_t}{R_a} V_a \quad (48)$$

A equação pode ser rescrita como:

$$\begin{aligned} J \ddot{\varphi} + B \dot{\varphi} &= K V_a \\ J &= J_2 + J_1 n^2 \\ B &= b_2 + b_1 n^2 + \frac{k_e k_t}{R_a} n^2 \\ K &= \frac{n k_t}{R_a} \end{aligned} \quad (49)$$

Ao adicionarmos uma carga ao sistema do motor, adicionamos um torque de reação τ_a , logo:

$$J\ddot{\varphi} + B\dot{\varphi} = KV_a - \tau_a \quad (50)$$

Pela segunda lei de Newton podemos determinar o torque produzido pelo motor devido à sua rotação:

$$\tau = J\ddot{\varphi} \quad (51)$$

Com as Equações (50) e (51):

$$\tau + \tau_a = KV_a - B\dot{\varphi} \quad (52)$$

Que é o torque resultante produzido pela rotação da roda.

$$\tau_r = KV_a - B\dot{\varphi} \quad (53)$$

4.1.3 Espaço de Estados do Sistema

Como o motor está preso à estrutura do robô, a relação entre o ângulo do motor e o ângulo da roda é:

$$\varphi_r = \varphi + \theta \quad (54)$$

Unindo as Equações (25), (41), (53) e (54) obtemos as equações diferenciais linearizadas que descrevem a dinâmica do robô em função da tensão aplicada ao motor:

$$\begin{aligned} & \left(r^2(m_c + 2m_r) + 2I_r \right) \ddot{\varphi} + \left(r^2(m_c + 2m_r) + 2I_r + m_{clr} \right) \ddot{\theta} + (2b_r + 2B)\varphi + 2b_r\dot{\theta} = 2KV_a \\ & m_{clr}\ddot{\varphi} + (I_c + m_{cl}(l+r))\ddot{\theta} - 2B\dot{\varphi} - m_{cgl}\theta = -2KV_a \end{aligned} \quad (55)$$

Podemos rescrever as duas equações como:

$$\mathbf{E} \begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \end{bmatrix} + \mathbf{F} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \end{bmatrix} + \mathbf{G} \begin{bmatrix} \varphi \\ \theta \end{bmatrix} = \mathbf{H}V_a \quad (56)$$

Sendo:

$$\begin{aligned} \mathbf{E} &= \begin{bmatrix} r^2(m_c + 2m_r) + 2I_r & r^2(m_c + 2m_r) + 2I_r + m_{clr} \\ m_{clr} & I_c + m_{cl}(l+r) \end{bmatrix} \\ \mathbf{F} &= \begin{bmatrix} 2b_r + 2B & 2b_r \\ -2B & 0 \end{bmatrix} \\ \mathbf{G} &= \begin{bmatrix} 0 & 0 \\ 0 & -m_{cgl} \end{bmatrix} \\ \mathbf{H} &= \begin{bmatrix} 2K \\ -2K \end{bmatrix} \end{aligned} \quad (57)$$

O espaço de estados do sistema é então definido pela relação matricial apresentada por (FRANKOVSKY *et al.*, 2017):

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \ddot{\varphi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\mathbf{E}^{-1}\mathbf{G} & -\mathbf{E}^{-1}\mathbf{F} \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{E}^{-1}\mathbf{H} \end{bmatrix} V_a \quad (58)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix}$$

4.2 DETERMINAÇÃO DOS PARÂMETROS DO SISTEMA

Após determinadas as equações que descrevem o comportamento do sistema podemos seguir para a determinação dos parâmetros do sistema de controle a ser trabalhado. Esta seção busca discutir os métodos utilizados para chegar aos parâmetros que descrevem a planta com maior eficácia.

4.2.1 Parâmetros do Motor

Os parâmetros associados ao motor serão os primeiros a serem identificados, já que eles podem ser obtidos através do comportamento do motor desacoplado do robô. O objetivo aqui é encontrar valores satisfatórios para J , B e K da Equação (55).

A função transferência do motor pode ser extraída da análise da Equação (55):

$$G_p(s) = \frac{\Phi(s)}{V_a(s)} = \frac{K}{s(Js + B)} \quad (59)$$

Como se trata de um sistema de segunda ordem, ele pode ser expresso em função da frequência natural ω_n e constante de amortecimento ζ do sistema(OGATA *et al.*, 2010):

$$G_p(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)} \quad (60)$$

Comparando a Equação (59) com a Equação (60) chegamos à seguinte relação:

$$\begin{aligned} \frac{K}{J} &= \omega_n^2 \\ \frac{B}{J} &= 2\zeta\omega_n \end{aligned} \quad (61)$$

A frequência natural e a constante de amortecimento podem ser determinadas experimentalmente com base na resposta transitória do sistema. Uma das formas é através dos valores de máximo sobressinal e tempo de pico quando o sistema em malha fechada

recebe uma entrada em degrau (OGATA *et al.*, 2010). Com isso em mente foram feitos testes de campo com o motor em vazio para diferentes entradas em degrau e determinados graficamente os valores de ω_n e ζ . Os testes foram realizado com o motor em malha fechada em conjunto com um controle proporcional, o diagrama de blocos que representa a configuração do sistema pode ser visto na ?? enquanto o resultado obtido pode ser visto na Figura 16.

Figura 15 – Diagrama de blocos para o motor em malha fechada.

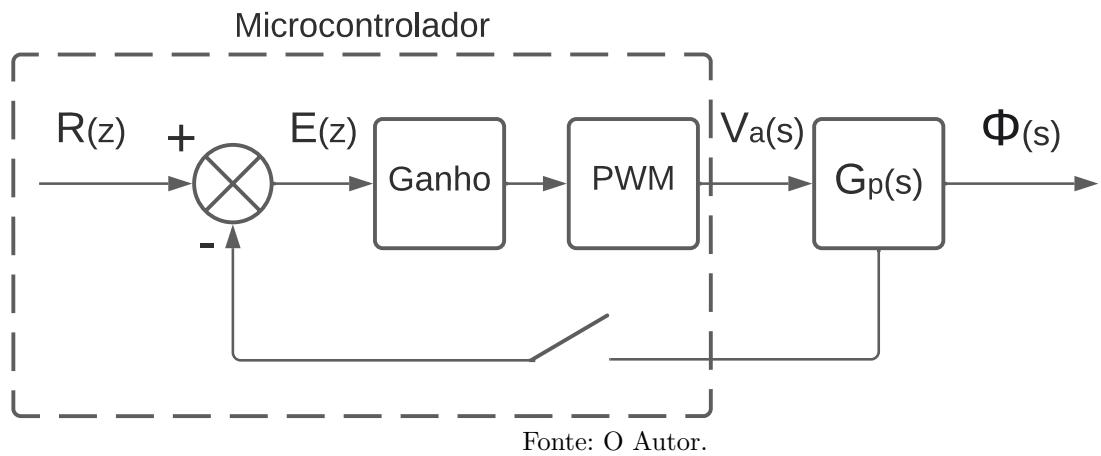
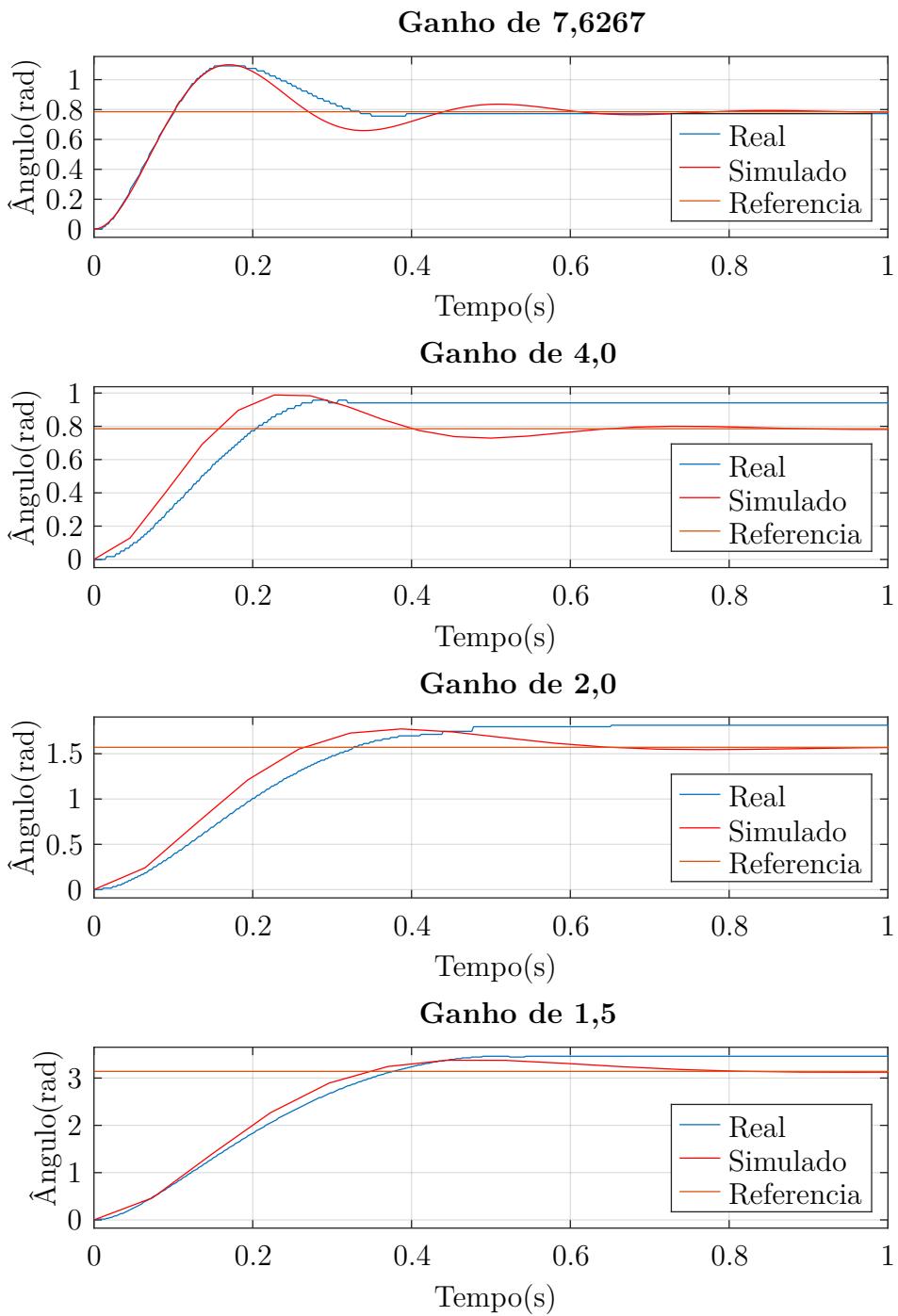


Figura 16 – Resposta do motor a diferentes entradas do tipo degrau em malha fechada.



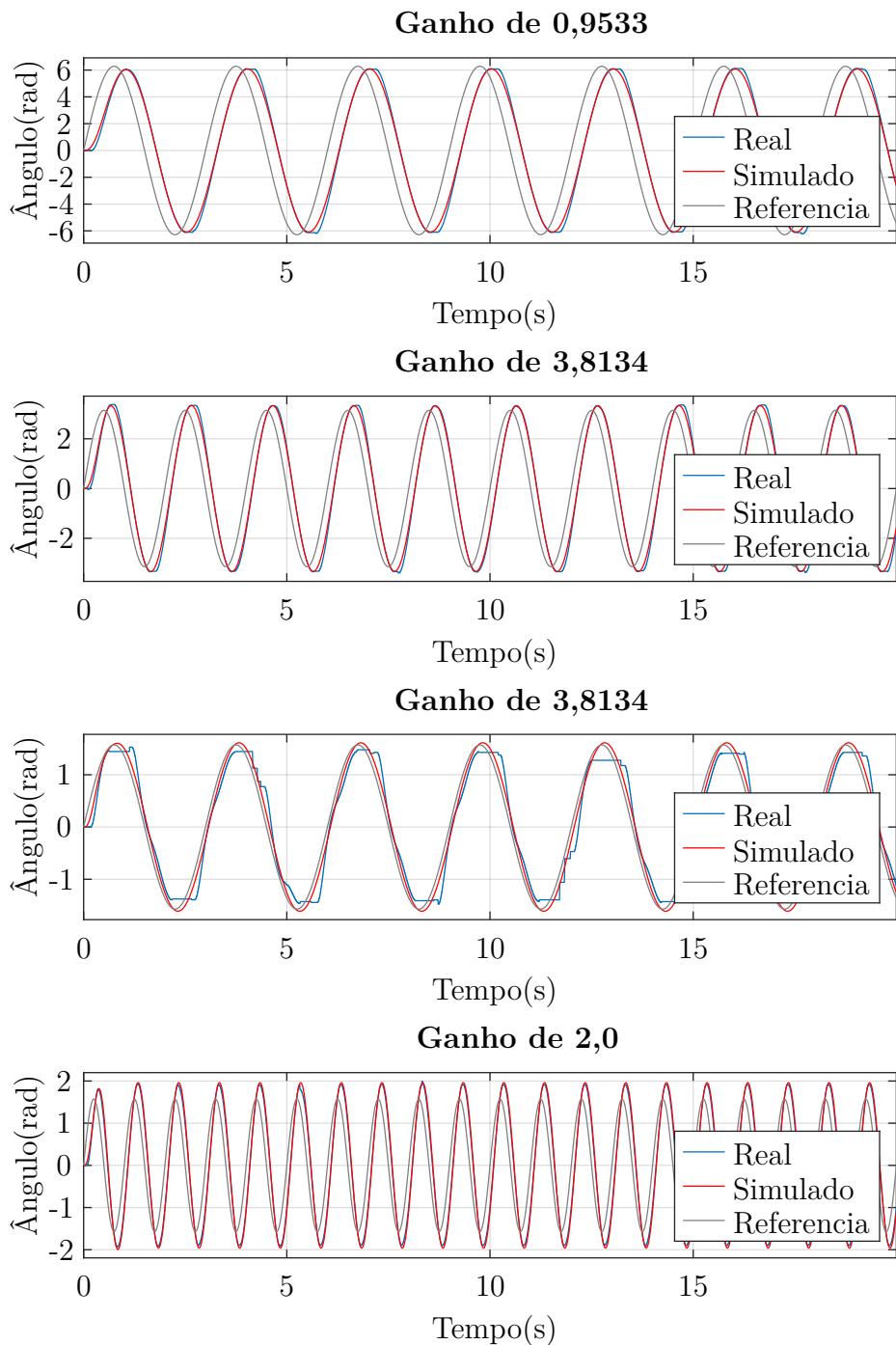
Fonte: O Autor.

Os teste mostraram que o motor apresenta comportamentos não lineares bem significativos, causados possivelmente pelo atrito estático e pela não idealidade do controle de tensão realizado pela ponte H. As não linearidades são mais presentes para baixas

tensões e perdem sua influência conforme a tensão aumenta, elas também estão associadas ao movimento do motor, devido à diferença entre o atrito estático e o atrito dinâmico. Por conta disso pode se dizer que as não linearidades estão associadas ao regime estacionário, onde a ação de controle é menor e o atrito estático é mais presente. Durante o equilíbrio do robô é esperado que o motor se mantenha em constante movimento, logo sinais transitórios como entradas senoidais e em rampa têm o potencial de descrever o comportamento esperado com mais precisão.

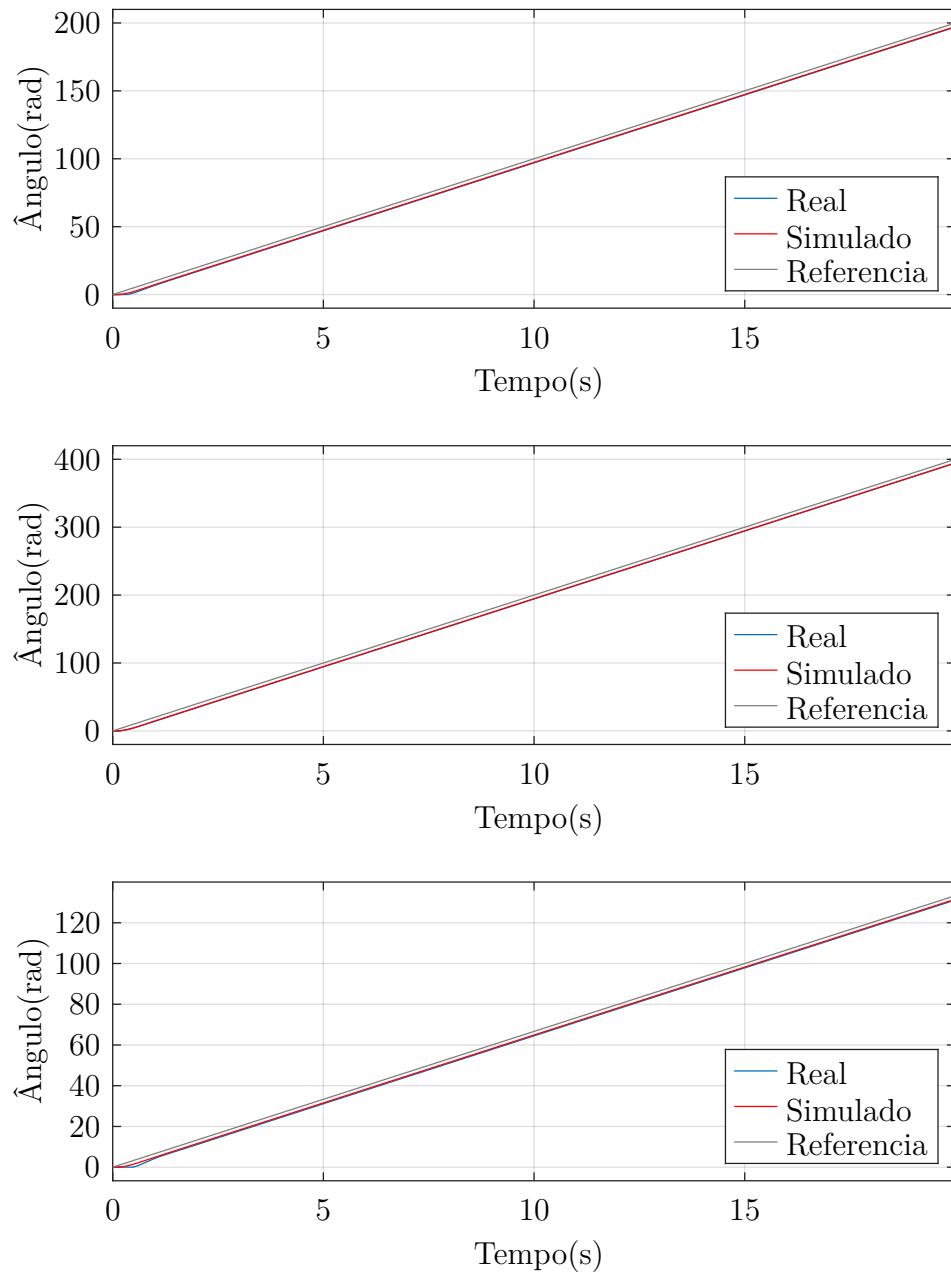
Com isso em mente foram realizados outros testes com o motor para diferentes entradas senoidais e do tipo rampa, os valores de frequência natural e constante de amortecimento obtidos foram modificados com base na resposta gráfica obtida e o resultado pode ser visto nas Figuras 17, 18 e 19.

Figura 17 – Resposta do motor a diferentes entradas senoidais em malha fechada.



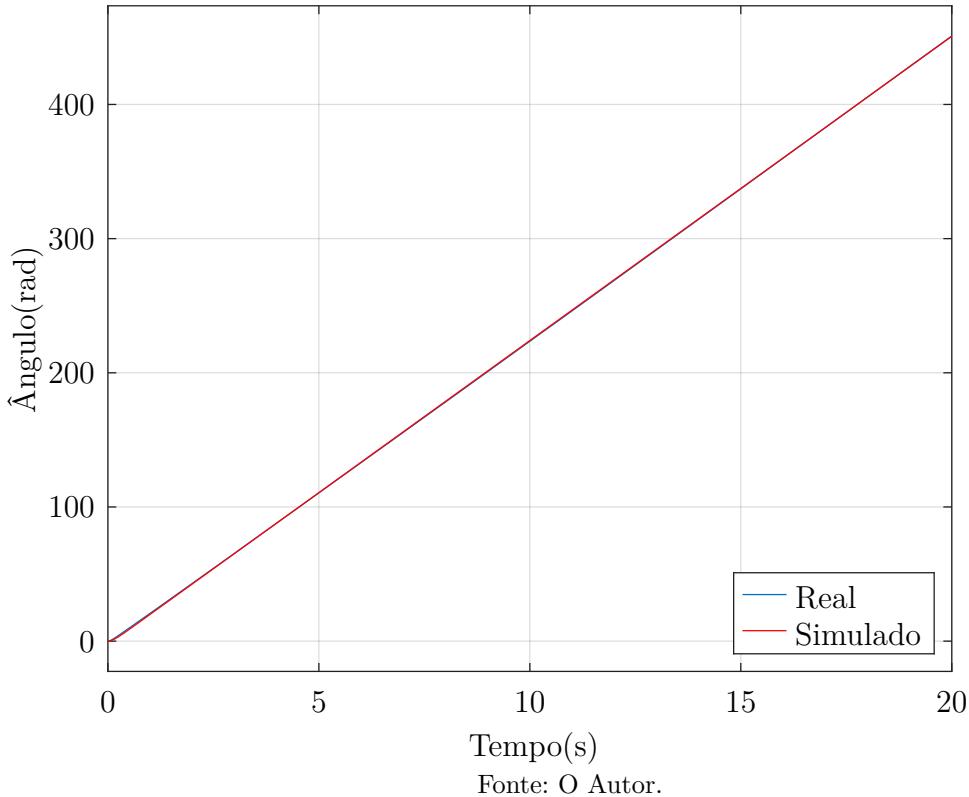
Fonte: O Autor.

Figura 18 – Resposta do motor a diferentes entradas do tipo rampa em malha fechada.



Fonte: O Autor.

Figura 19 – Resposta do motor a uma entrada de 5,99V em malha aberta.



Os resultados mostram que os valores determinados são válidos e descrevem o motor com precisão, os valores de ω_n e ζ determinados são:

$$\begin{aligned}\omega_n &= 15,533 \\ \zeta &= 0,2690\end{aligned}\tag{62}$$

Existem inúmeras soluções para a Equação (61) com os valores de ω_n e ζ obtidos, logo um dos parâmetros deve ser determinado por outro meio. Por simplicidade foi escolhido o K , já que ele necessita apenas da constante de torque do motor k_t , da resistência R_a e da relação de redução n , fatores que podem ser obtidos através da Tabela 1.

Mesmo em vazio o motor apresenta perdas de torque para movimentar o eixo do motor τ_p , logo, o torque exercido na carga do motor é:

$$\tau = n\tau_m - \tau_p\tag{63}$$

Utilizando a Equação (43) e a Equação (63):

$$\tau = nk_t i_a - \tau_p\tag{64}$$

Para diferentes cargas podemos considerar que k_t e τ_p serão constantes, permitindo que criemos um sistema de equações com os torques e correntes para dois diferentes testes:

$$k_t = \frac{\tau_1 + \tau_p}{ni_{a1}} \quad (65)$$

$$\tau_p = nk_t i_{a2} - \tau_2$$

A partir do sistema da Equação (65) conseguimos chegar à constante k_t :

$$k_t = \frac{\tau_1 - \tau_2}{n(i_{a1} - i_{a2})} \quad (66)$$

Utilizando a Equação (47) e considerando que para diferentes testes a tensão aplicada ao motor V_a , a resistência da armadura R_a , a relação de redução n e a constante de força contra eletromotriz serão constantes:

$$R_a = \frac{V_a + k_e n \dot{\varphi}_1}{i_{a1}} \quad (67)$$

$$k_e = \frac{V_a - R_a i_{a2}}{n \dot{\varphi}_2}$$

Resolvendo a Equação (67) com o objetivo de isolar a resistência R_a chegamos a:

$$k_t = V_a \frac{\dot{\varphi}_2 - \dot{\varphi}_1}{i_{a1} \dot{\varphi}_2 - i_{a2} \dot{\varphi}_1} \quad (68)$$

Com base nos torques, correntes e velocidades presentes na Tabela 1:

Tabela 2 – Variáveis extraídas da Tabela 1.

τ_1	$2,0 \text{ kgf}\cdot\text{cm} = 0,0057686 \text{ N}\cdot\text{m}$
τ_2	$5,2 \text{ kgf}\cdot\text{cm} = 0,014998 \text{ N}\cdot\text{m}$
$\dot{\varphi}_1$	$170 \text{ rpm} = 17,802 \text{ rad/s}$
$\dot{\varphi}_2$	$110 \text{ rpm} = 11,519 \text{ rad/s}$
i_{a1}	$0,60 \text{ A}$
i_{a2}	$1,10 \text{ A}$
n	34

Fonte: O Autor.

Através dos dados da Tabela 2 e utilizando as Equações (66) e (68) chegamos à constante de torque e à resistência da armadura do motor:

$$k_t = 0,018459 \text{ Nm/A} \quad (69)$$

$$R_a = 2,9752 \Omega \quad (70)$$

Aplicando os resultados encontrados na Equação (55) podemos determinar o valor de K :

$$K = 0,2109 \quad (71)$$

Utilizando os valores K , ω_n e ζ encontrados para o sistema na Equação (61):

$$J = 0,0066691 \quad (72)$$

$$B = 0,055734 \quad (73)$$

4.2.2 Parâmetros do Robô

Após determinado os parâmetros associados ao motor, podemos seguir para os demais parâmetros intrínsecos ao robô. As massas, tanto do corpo quanto da roda, foram medidas através de uma balança, o raio da roda e a distância da roda até o centro de massa foram medidos utilizando instrumentos de medição adequados, os valores encontrados estão dispostos na Tabela 3:

Tabela 3 – Resultado das medidas do robô.

m_c	1,013 kg
m_r	0,070 kg
r	0,04 m
l	0,09 m

Fonte: O Autor.

Existem três parâmetros restantes que não conseguem ser medidos facilmente por algum instrumento e devem ser estimados por experimentos, são eles: momento de inércia da roda, momento de inércia do corpo e constante de atrito da roda.

O momento de inércia da roda pode ser determinado através de experimentos considerando o sistema formado pela ligação da roda com o eixo do motor. Ao adicionarmos a roda ao sistema do motor, a roda produz um torque de reação e o sistema acaba sendo descrito pela Equação (50). Utilizando a segunda lei de Newton para encontrar o torque de reação da roda obtemos:

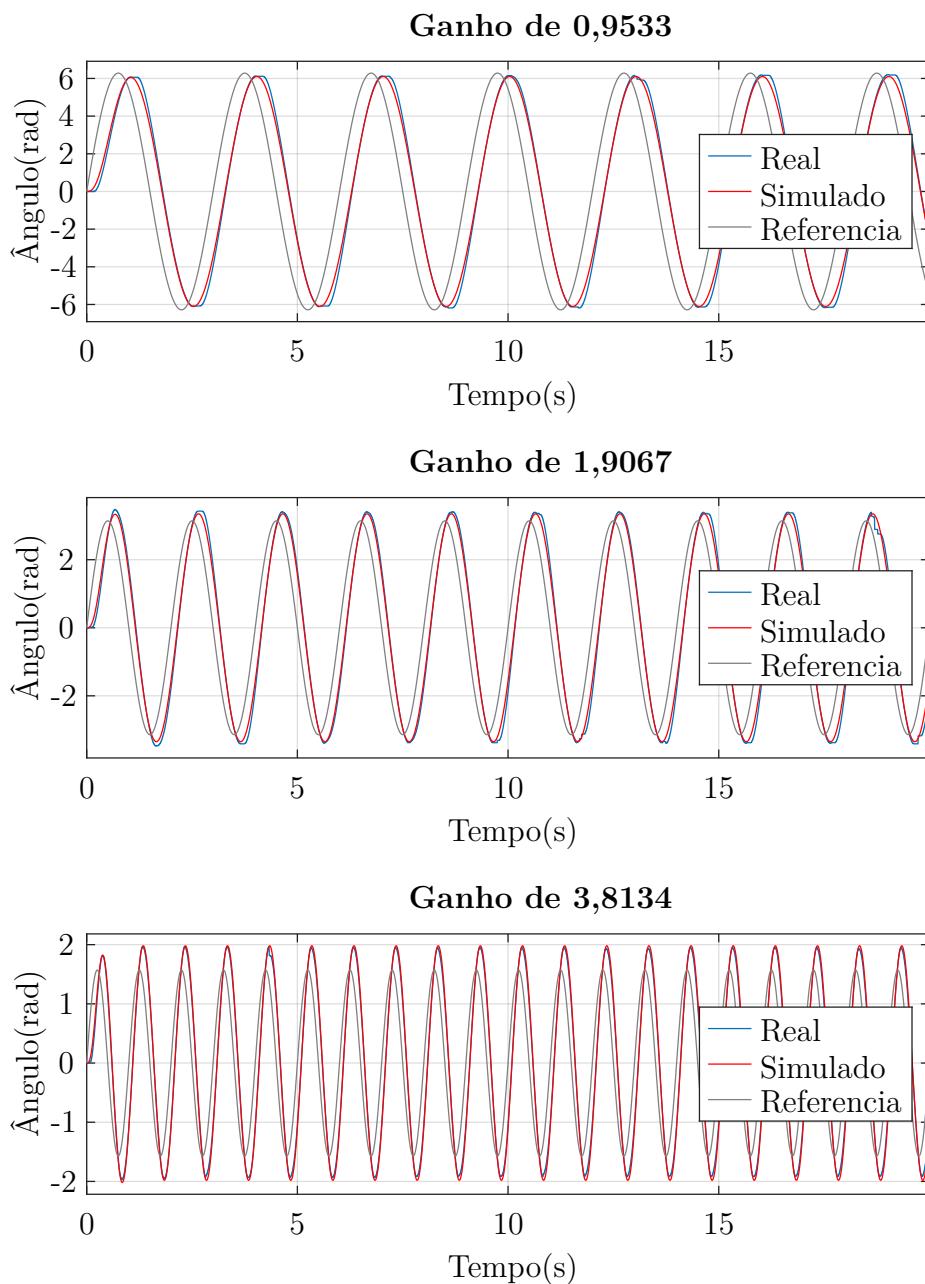
$$(J + I_r) \ddot{\varphi} + B\dot{\varphi} = KV_a \quad (74)$$

A partir da Equação (74) chegamos na função transferência do sistema com a roda:

$$G_{pr}(s) = \frac{\Phi(s)}{V_a(s)} = \frac{K}{s((J + I_r)s + B)} \quad (75)$$

Essa equação é a mesma encontrada para o motor com a adição do momento de inércia da roda. Pensando nisso foram realizados testes com entradas senoidais ao sistema motor-roda em malha fechada, através da análise gráfica foram feitos ajustes para encontrar um valor aproximado para o momento de inércia da roda. O resultado encontrado pode ser visto na Figura 20.

Figura 20 – Resposta do sistema motor-rodas a diferentes entradas senoidais em malha fechada.



Fonte: O Autor.

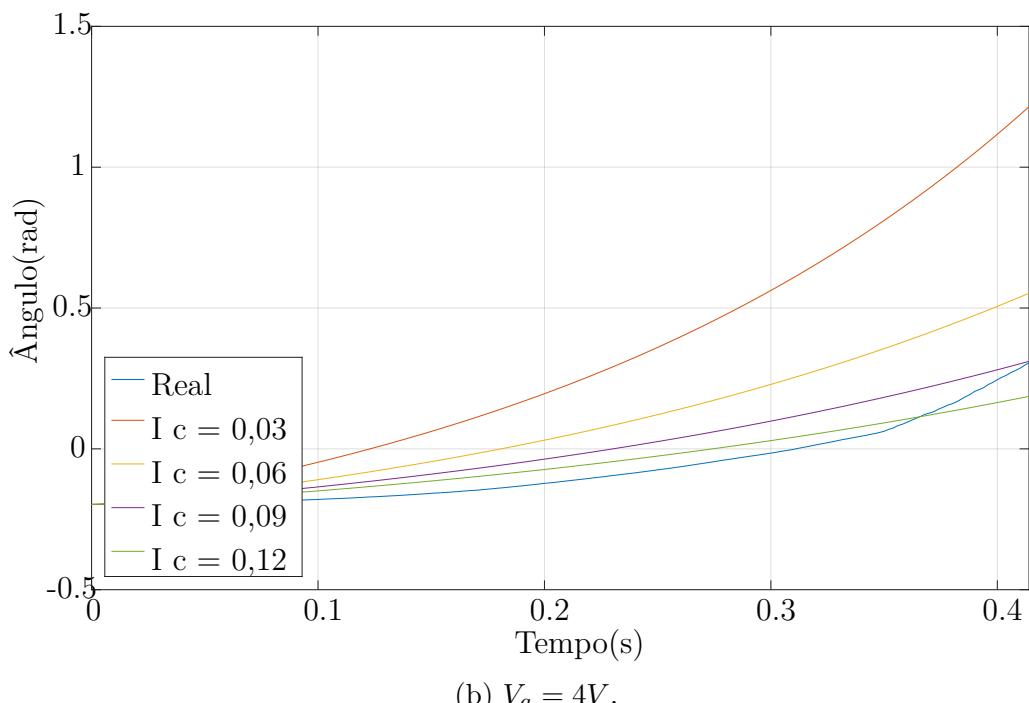
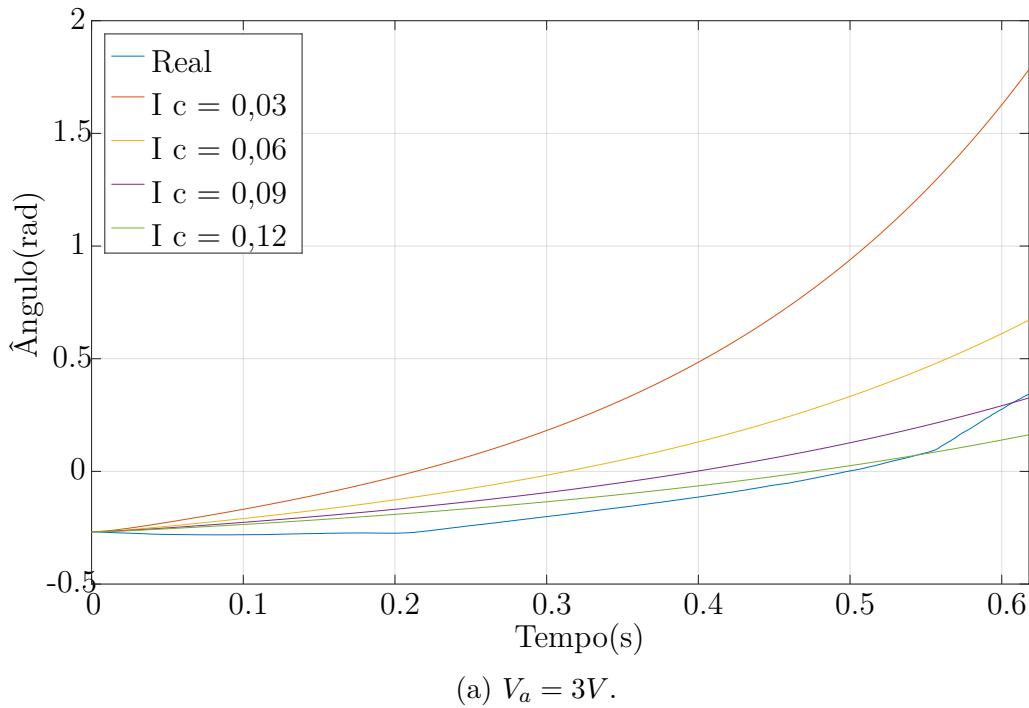
O momento de inércia encontrado para a roda é:

$$I_r = 0,0002 \text{kg} \cdot \text{m}^2 \quad (76)$$

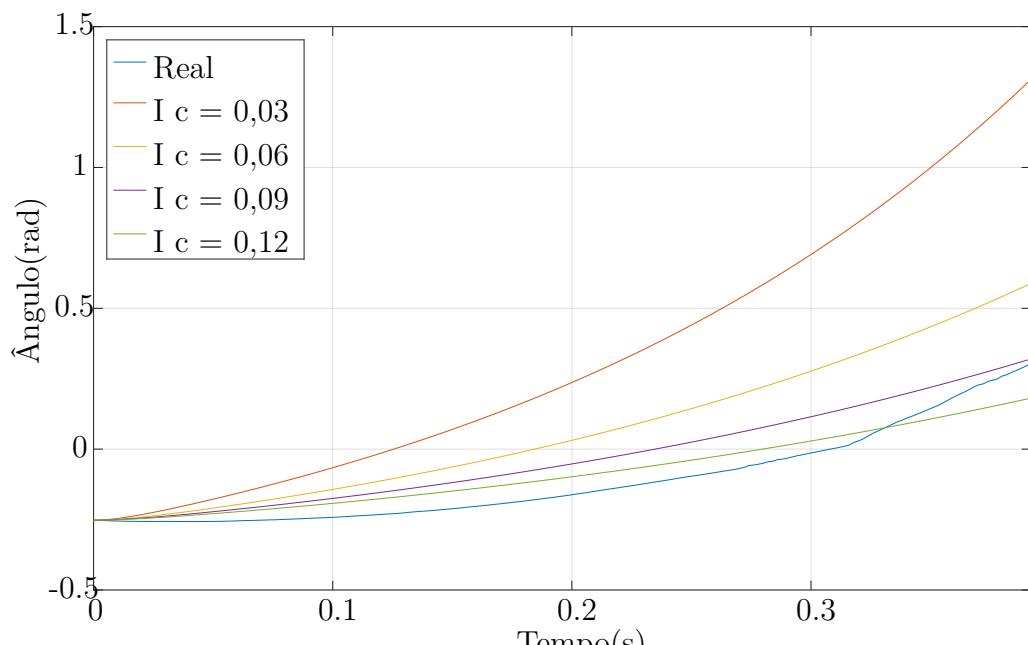
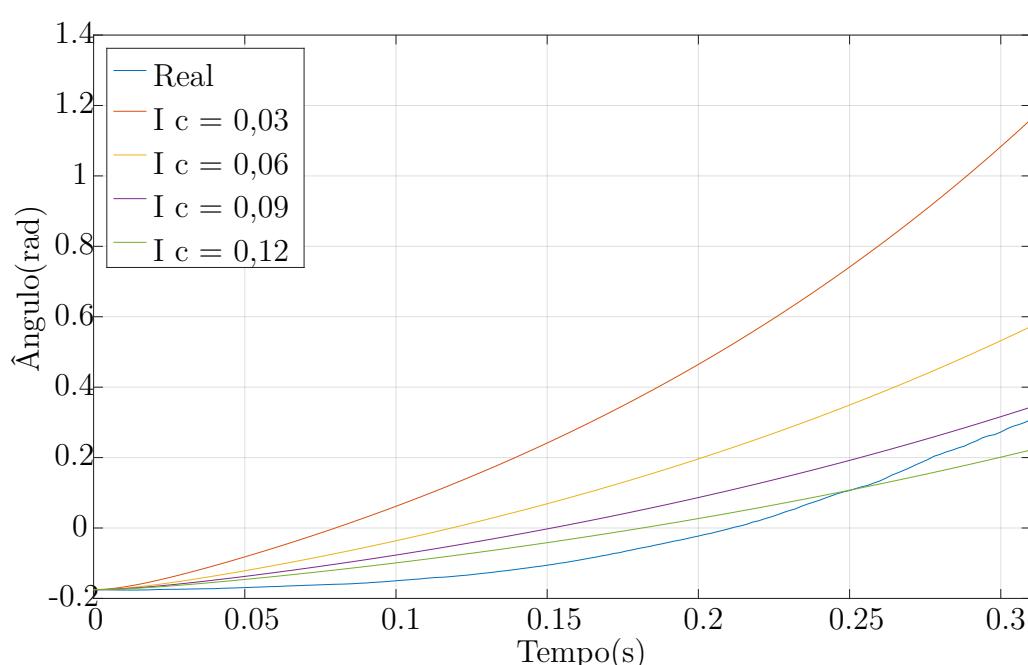
Buscando encontrar os dois parâmetros restantes, I_c e b_r , foram feitos experimentos com o robô em malha aberta para quatro diferentes entradas em degrau. Os resultados

encontrados foram comparados com a simulação para diferentes combinações dos parâmetros I_c e b_r . Como o sistema foi linearizado para pequenos ângulos, os dados considerados neste teste foram apenas os presentes na zona linear, ou seja, com ângulo θ inferior a 0,3 radianos. As Figuras 21 e 22 apresentam a resposta do ângulo do robô θ , enquanto as Figuras 23 e 24 apresentam a resposta do ângulo do motor φ .

Figura 21 – Comportamento do ângulo θ quando I_c varia e $b_r=0,02 \text{ kg}\cdot\text{m}^2/\text{s}$.

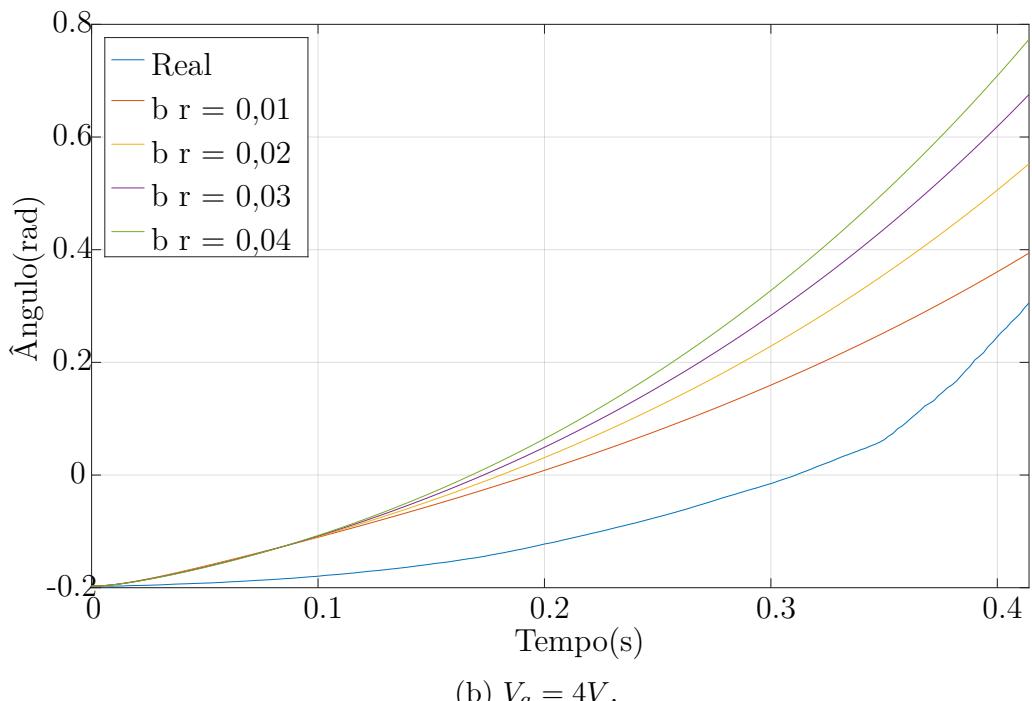
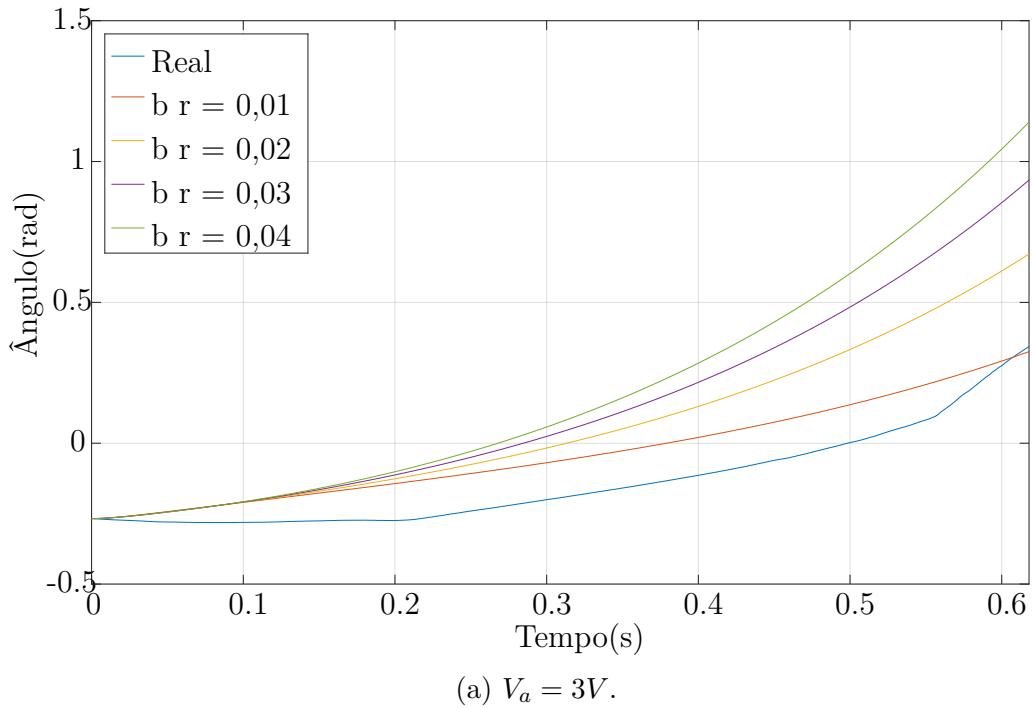


Fonte: O Autor.

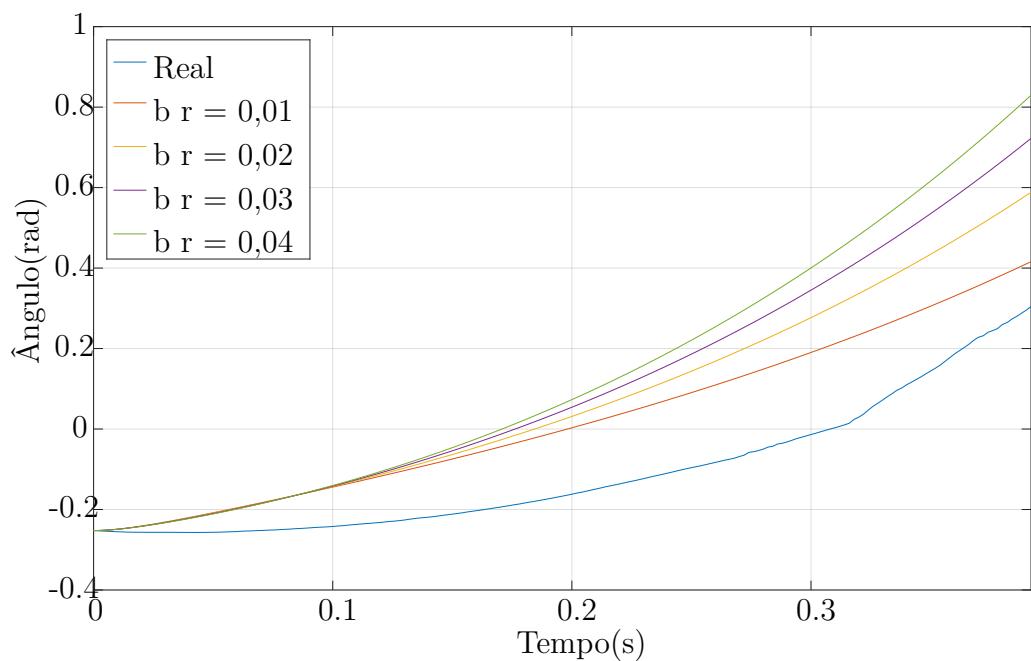
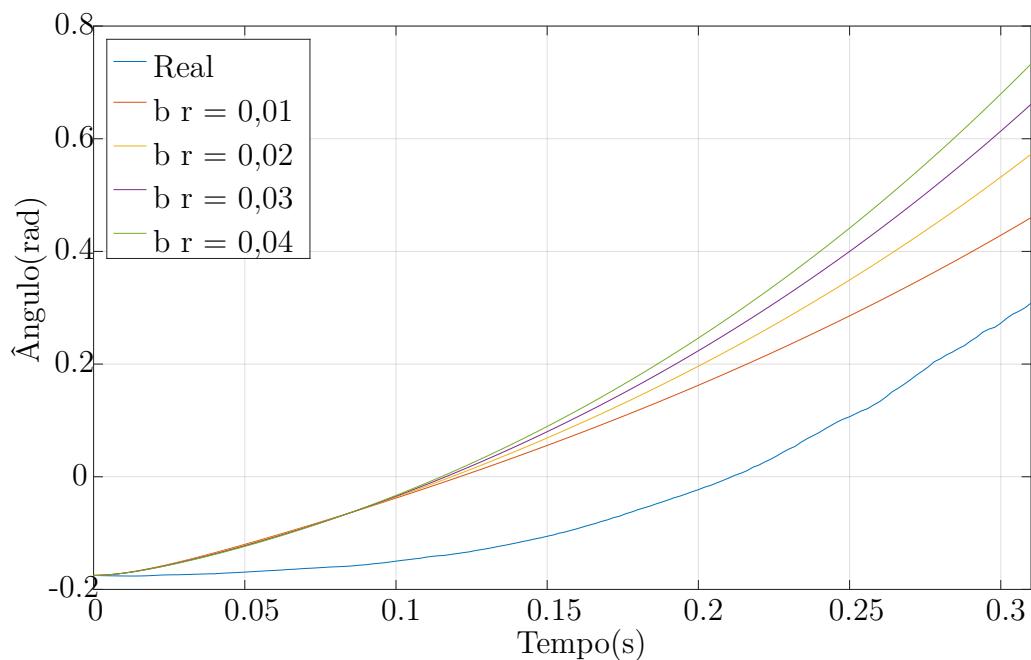
(c) $V_a = 5V$.(d) $V_a = 5,99V$.

Fonte: O Autor.

Figura 22 – Comportamento do ângulo θ conforme b_r varia e $I_c=0,06 \text{ kg}\cdot\text{m}^2$.

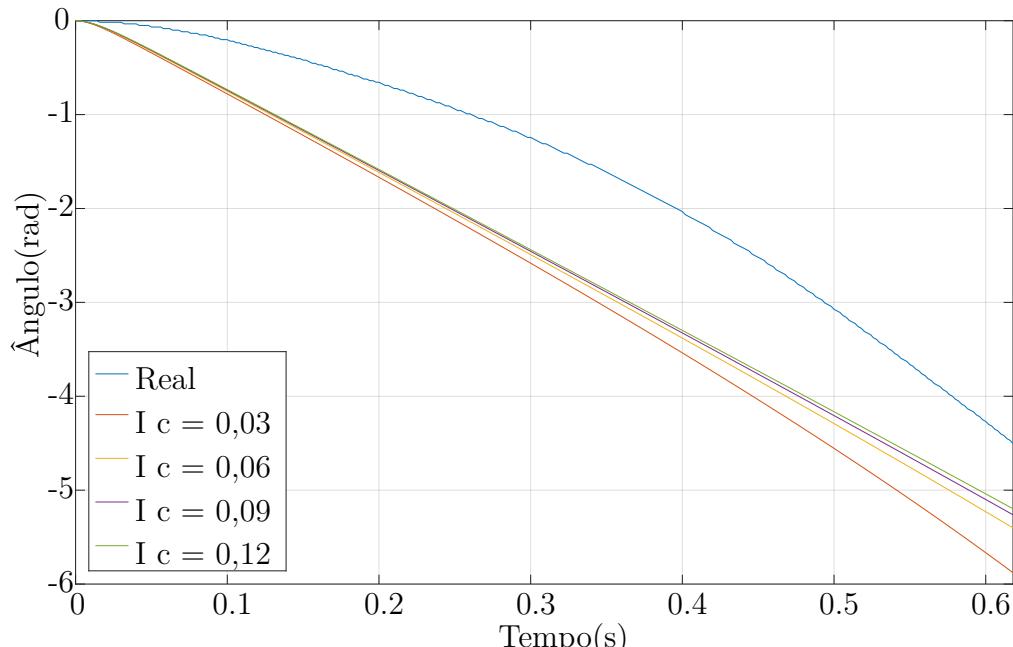


Fonte: O Autor.

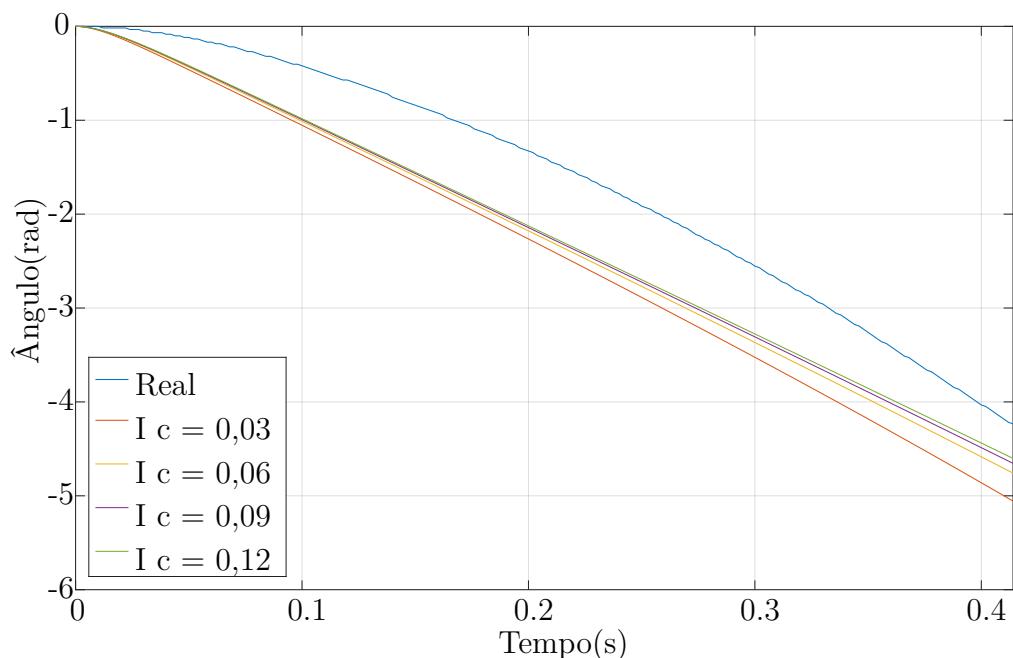
(c) $V_a = 5V$.(d) $V_a = 5,99V$.

Fonte: O Autor.

Figura 23 – Comportamento do ângulo φ conforme I_c varia e $b_r=0,02 \text{ kg}\cdot\text{m}^2/\text{s}$.

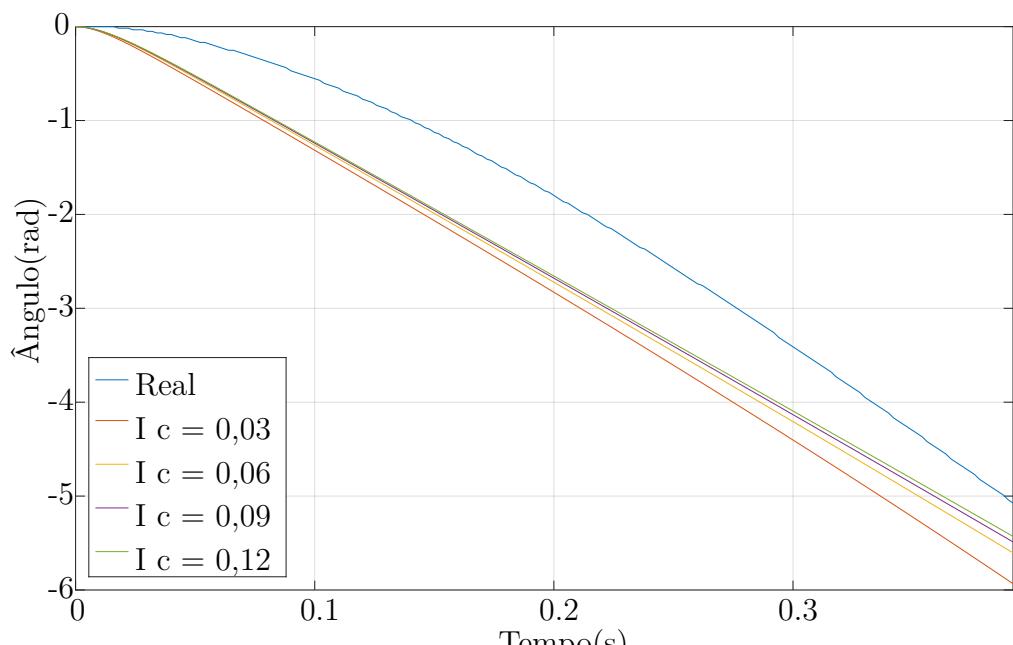
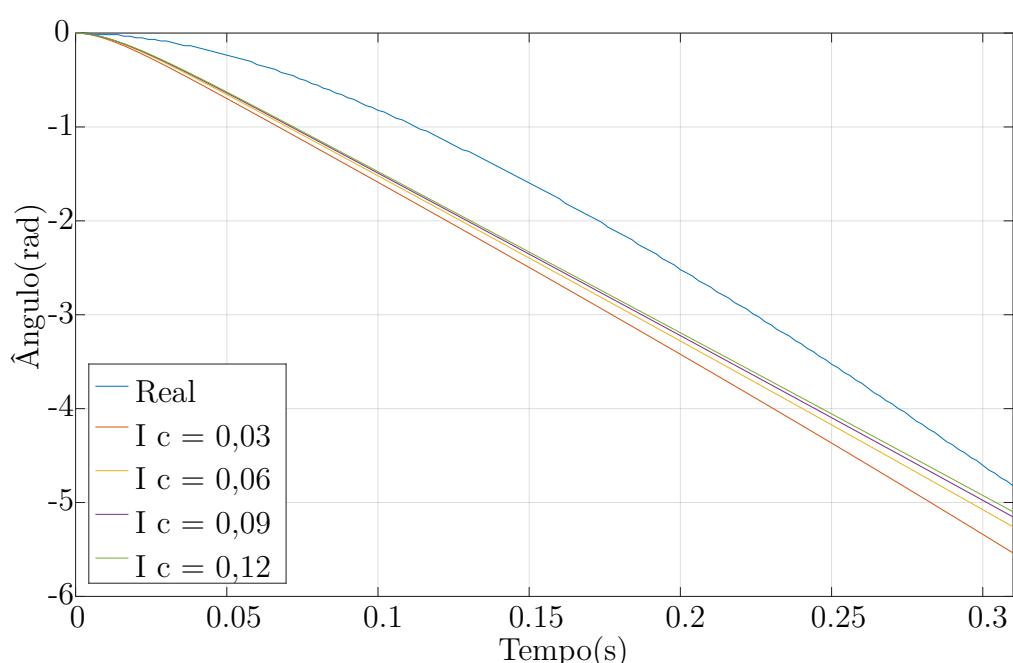


(a) $V_a = 3V$.



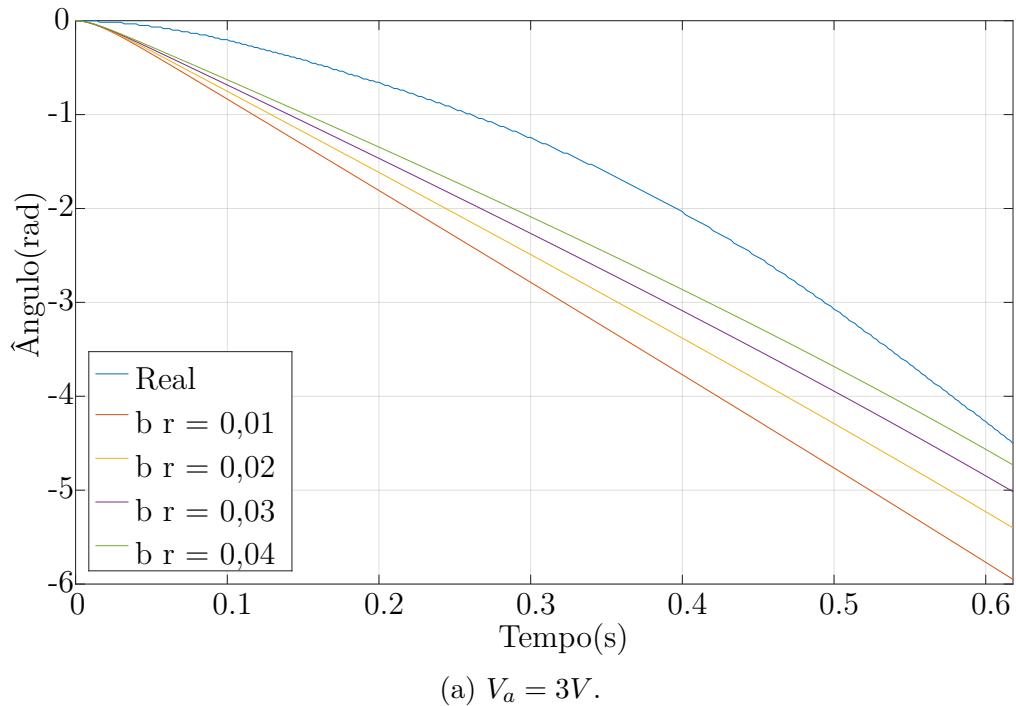
(b) $V_a = 4V$.

Fonte: O Autor.

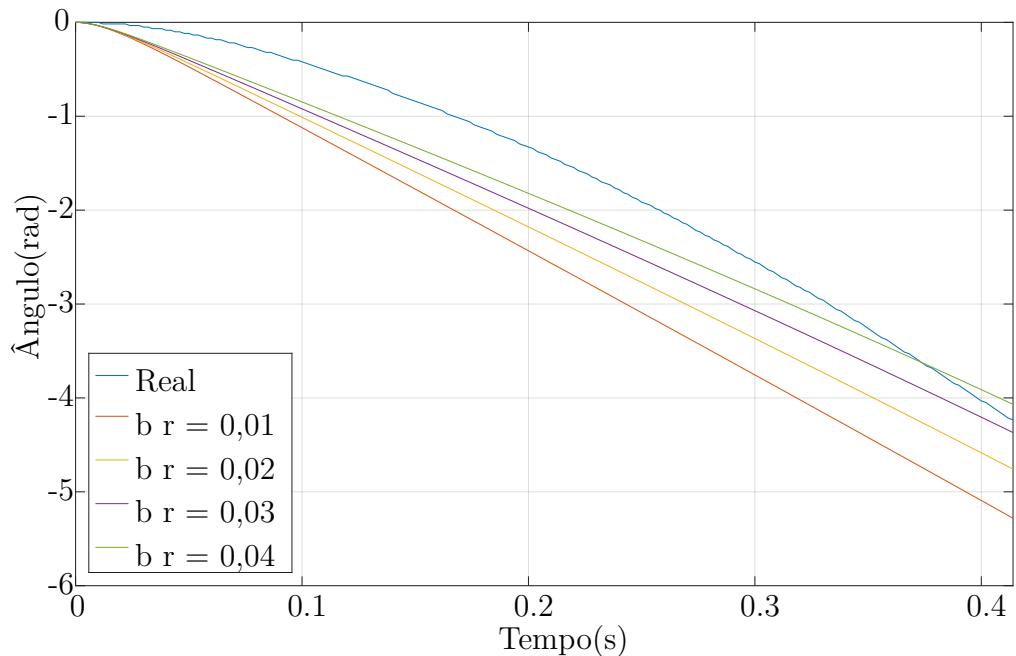
(c) $V_a = 5V$.(d) $V_a = 5,99V$.

Fonte: O Autor.

Figura 24 – Comportamento do ângulo φ conforme b_r varia e $I_c=0,06 \text{ kg}\cdot\text{m}^2$.

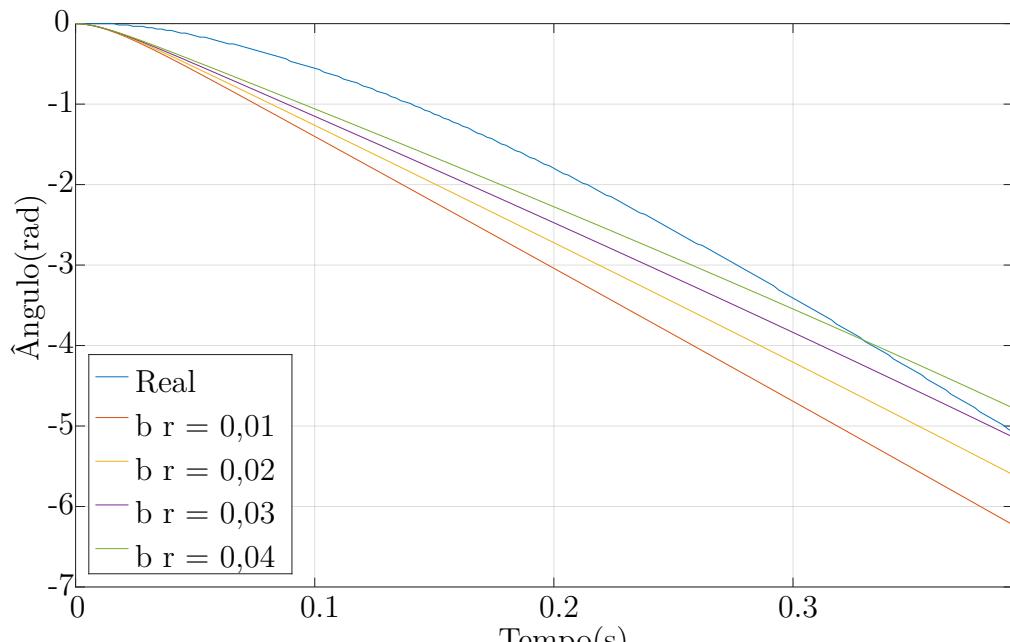
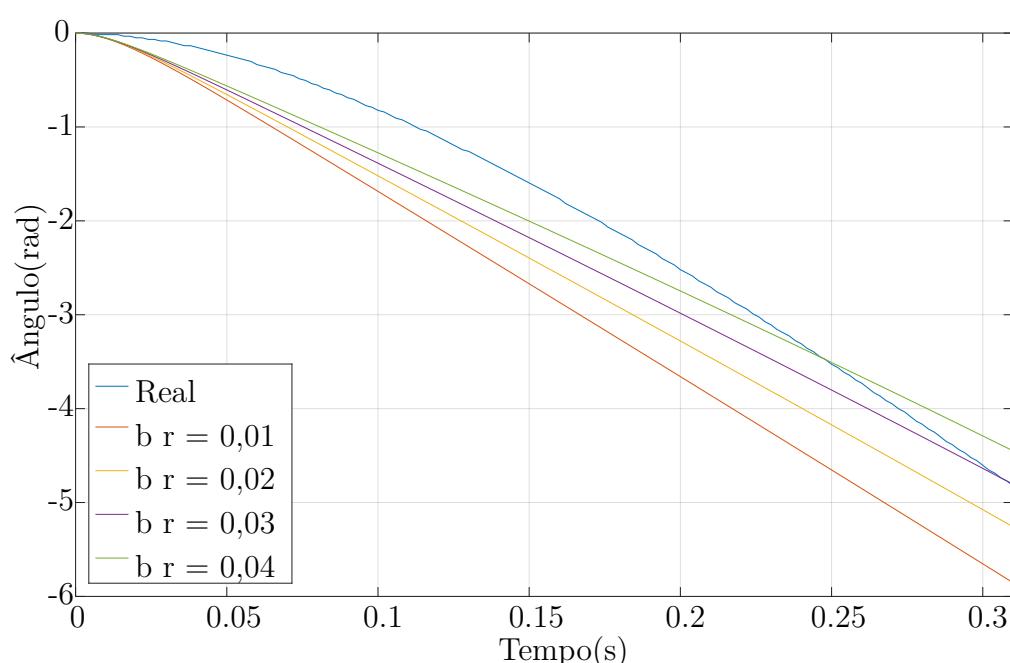


(a) $V_a = 3V$.



(b) $V_a = 4V$.

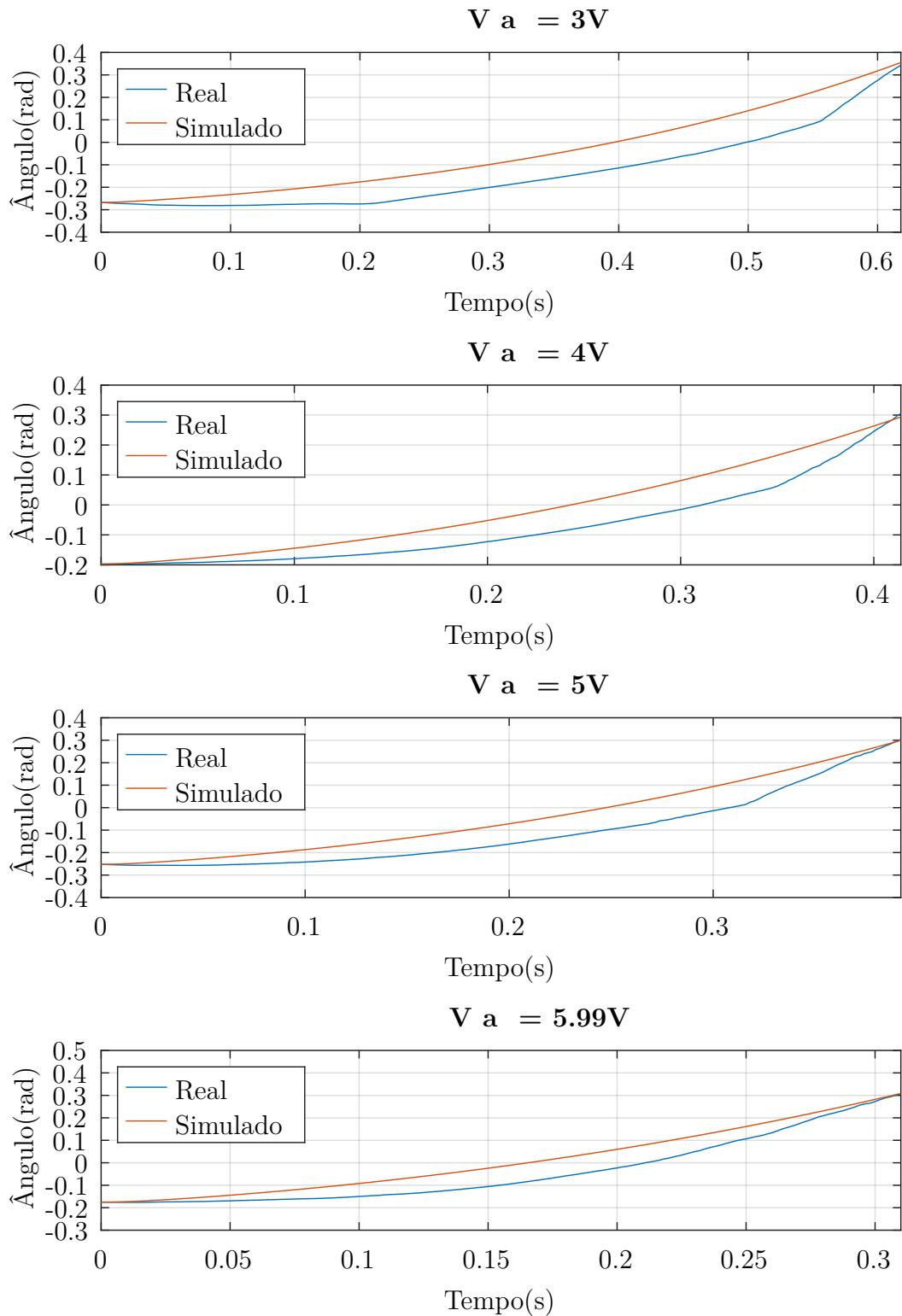
Fonte: O Autor.

(c) $V_a = 5V.$ (d) $V_a = 5,99V.$

Fonte: O Autor.

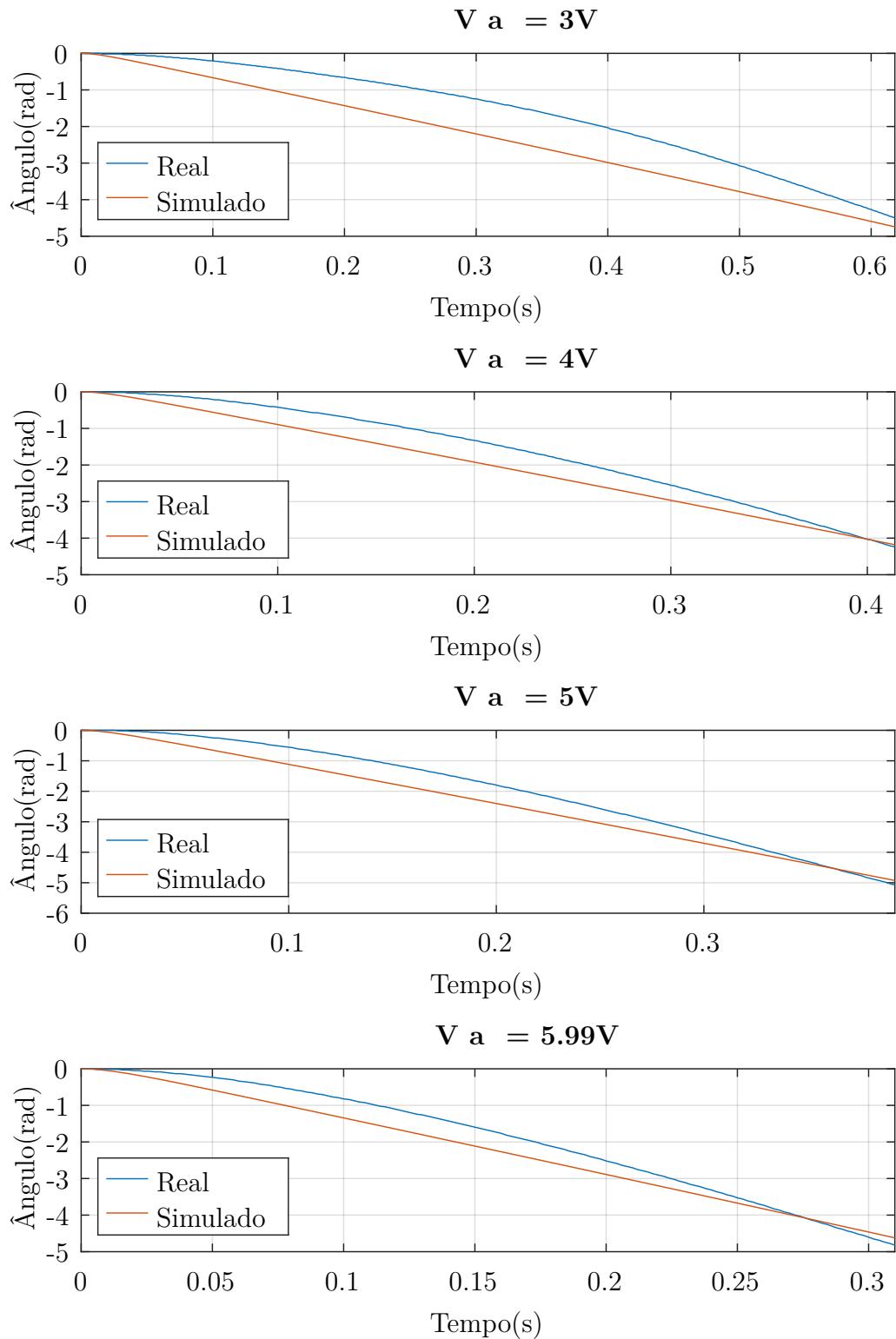
É perceptível que o resposta do ângulo do motor φ está mais fortemente associada à constante de atrito b_r , enquanto a resposta do ângulo do robô θ está relacionada com mais intensidade ao momento de inércia do corpo I_c . Um atrito maior gera uma menor variação do ângulo φ enquanto um momento de inércia maior gera uma maior variação do ângulo θ . Os valores que mais aproximam o sistema simulado ao real foram escolhidos com isso em mente e o resultado alcançado pode ser visto nas Figuras 25 e 26

Figura 25 – Resposta de θ com os parâmetros que mais se aproximam do real.



Fonte: O Autor.

Figura 26 – Resposta de φ com os parâmetros que mais se aproximam do real.



Fonte: O Autor.

Os parâmetros escolhidos foram:

$$\begin{aligned} I_c &= 0,11 \text{kg} \cdot \text{m}^2 \\ b_r &= 0,03 \text{kg} \cdot \text{m}^2/\text{s} \end{aligned} \quad (77)$$

A Tabela 4 apresenta todos os parâmetros determinados para o sistema do robô.

Tabela 4 – Parâmetros do robô auto-equilibrado.

K	0,2109
B	0,055734
m_c	1,013 kg
I_c	0,11 kg·m ²
m_r	0,070 kg
I_r	0,0002 kg·m ²
b_r	0,03 kg·m ² /s
r	0,04 m
l	0,09 m
g	9,80665 m/s ²

Fonte: O Autor.

Utilizando os dados da Tabela 4 e as Equações (57) e (58) chegamos ao espaço de estado do robô auto-equilibrado.

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \ddot{\varphi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -20,9 & -85,5 & -29,01 \\ 0 & 7,963 & 3,474 & 0,8681 \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -213,8 \\ 9,86 \end{bmatrix} V_a \quad (78)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix}$$

Considerando que as variáveis de estado e a entrada do sistema são:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix} \\ \mathbf{u} &= V_a \end{aligned} \quad (79)$$

Então as matrizes do espaço de estado são:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -20,9 & -85,5 & -29,01 \\ 0 & 7,963 & 3,474 & 0,8681 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 \\ 0 \\ -213,8 \\ 9,86 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\end{aligned}\tag{80}$$

5 CONTROLE

Após encontrado um modelo que descreve o comportamento do sistema de forma satisfatória, podemos seguir para o desenvolvimento do controle, que inclui desde a análise do sistema e do seu comportamento até o desenvolvimento do controlador e do observador que serão utilizados no controle. Este capítulo busca apresentar todos os passos tomados para se chegar ao controle final que estabiliza o robô.

Tanto a análise dos resultados, a criação do controle e as simulações foram feitas com o auxílio da plataforma GNU Octave, que conta com um pacote de controle que simplifica os cálculos e a criações de gráficos.

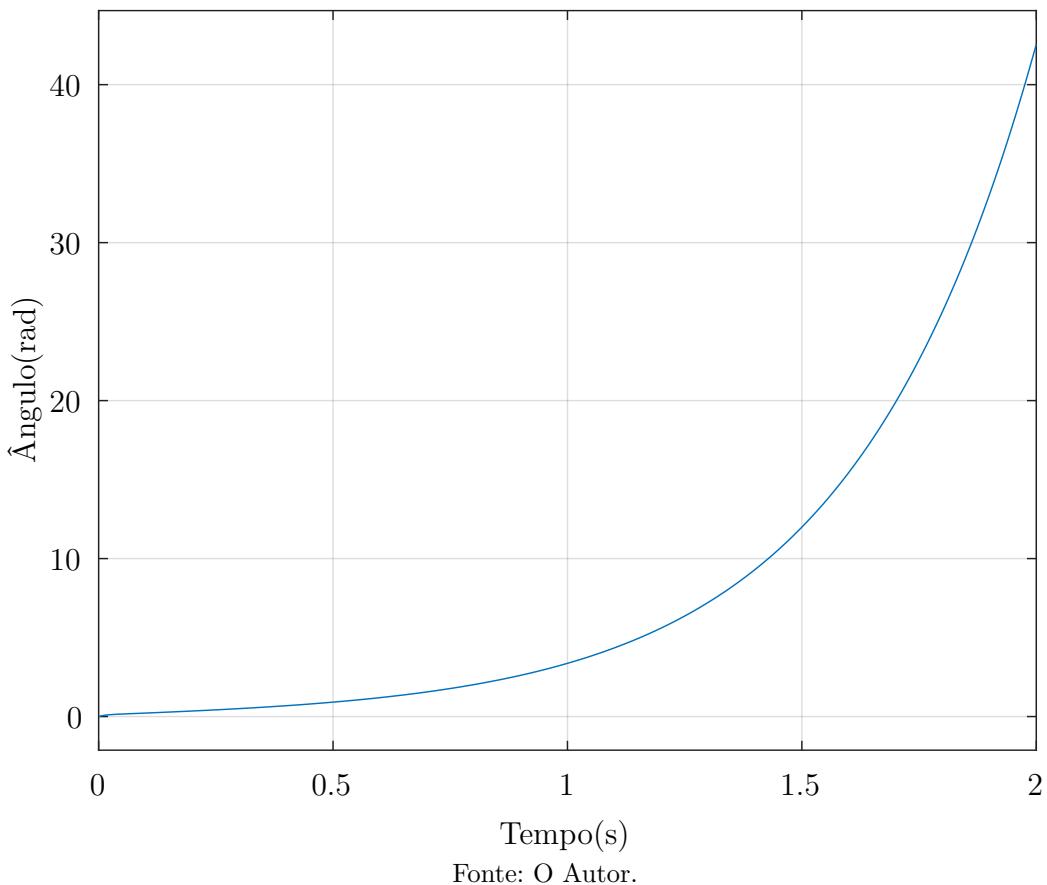
Como visto anteriormente, o objetivo deste trabalho é desenvolver um controle que mantenha o robô estável na posição de equilíbrio, em outras palavras, manter o ângulo θ do robô o mais próximo possível de zero, mesmo quando o sistema for exposto a perturbações do ambiente. Com isso em mente fica claro que a primeira característica a ser analisada é a estabilidade do sistema, uma das maneiras é através dos autovalores do sistema e outra através da resposta do sistema no tempo, ambas alternativas serão tratadas aqui.

Os autovalores, ou polos do sistema, são:

$$\mathbf{p} = \begin{bmatrix} 0 \\ -84,327 \\ 2,5371 \\ -2,8429 \end{bmatrix} \quad (81)$$

Como um dos autovalores é maior que zero, o sistema é instável em malha aberta. Ao simularmos o comportamento do sistema a uma entrada em impulso temos o mesmo resultado, o sistema se desestabiliza e as variáveis de estado tendem ao infinito, na prática, o robô atinge o chão e para. O resultado da simulação pode ser visto na Figura 27.

Figura 27 – Resposta do sistema ao impulso.



Para que o sistema se torne estável todos os autovalores devem ser modificados de maneira que eles sejam menores que zero, o que pode ser feito através da técnica de realimentação de estados.

Como visto em Seção 2.3, o primeiro passo é verificar a estabilidade e controlabilidade do sistema. Realizando os testes utilizando os valores da Equação (80) verificamos que o sistema é completamente controlável e completamente observável.

O equipamento responsável por implementar o controle no robô é o microcontrolador, o qual trabalha de forma digital, logo existe a necessidade de discretizar o sistema a fim de se desenvolver um controlador digital que irá gerar o comportamento desejado na planta de controle.

A transformação do espaço de estados pode ser feita através da GNU Octave, mas primeiro é necessário definir qual o tempo de amostragem do sistema. Foi escolhido um tempo de 2 milissegundos, valor pequeno o suficiente para permitir que controles mais rápidos sejam executados, mas também que garante a consistência na execução das funções do microcontrolador. Com esse valor, o resultado obtido para o espaço de estados discreto é:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1 & -3,981 \cdot 10^{-5} & 0,001838 & -5,49 \cdot 10^{-5} \\ 0 & 1 & 6,571 \cdot 10^{-6} & 0,002002 \\ 0 & -0,03885 & 0,8426 & -0,05341 \\ 0 & 0,0158 & 0,006391 & 1,002 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} -0,0004045 \\ 1,878 \cdot 10^{-5} \\ -0,3935 \\ 0,01833 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{82}$$

Como o sistema é completamente controlável, a realimentação de estados nos permite realocar os autovalores do sistema para quaisquer outros autovalores através do vetor de realimentação \mathbf{K} , entretanto o problema é determinar quais seriam os autovalores que gerariam o comportamento desejado para o sistema. Existem diversas maneiras de se chegar a este resultado, mas aqui a técnica escolhida para determinar o vetor de realimentação do sistema foi o regulador linear quadrático, visto que ele permite determinar um ganho que estabiliza o sistema através da escolha das matrizes \mathbf{Q} e \mathbf{R} .

A matriz \mathbf{Q} está relacionada às variáveis de estado, portanto é uma matriz 4x4, enquanto \mathbf{R} está relacionado às entradas do sistema, logo é um escalar. Para facilitar a escolha da matriz \mathbf{Q} se convém admitir que:

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \tag{83}$$

Quando a matriz \mathbf{Q} assume este formato a minimização do índice de desempenho faz com que as variáveis de estados x_i sejam minimizadas a uma proporção q_i (PHILLIPS; NAGLE; CHAKRABORTTY, 2015). Em termos práticos, isso nos diz que conforme maior for o valor de q_i em relação às outras parcelas, mais importância o controle dará para o estado correspondente, ou seja, mais rapidamente esse estado tenderá a zero. Por exemplo, conforme aumentarmos q_1 , mais intensamente o controle irá tentar fazer com que o ângulo do motor φ se estabilize em zero, dando prioridade a ele sobre os outros estados.

R é responsável por restringir a ação de controle penalizando esforços excessivamente intensos. De modo geral, um R menor gera um controle mais abrupto com gasto maior de energia e vice-versa

Como demonstrado por Castrucci, Bittar e Sales (2018), o valor do vetor de realimentação \mathbf{K} pode ser determinado pelo algoritmo de repetição mostrado na Figura 28,

o N deve ser um valor suficientemente grande para ocorrer a convergência da solução da equação de Riccati:

Figura 28 – Algoritmo para a determinação do ganho \mathbf{K} .

```

Iniciar  $F = Q;$ 
Para  $k$  de 1 até  $N$ 
     $W = R + B_d^T F B_d;$ 
     $P = F - F B_d W^{-1} B_d^T F;$ 
     $F = A_d^T P A_d + Q;$ 
fim para;
 $K = W^{-1} B_d^T F A_d$ 
```

Fonte: Castrucci, Bittar e Sales (2018).

Independentemente do valor que escolhermos para \mathbf{Q} e \mathbf{R} o sistema será estabilizado, entretanto, certos valores podem gerar um comportamento que não é replicável na prática, devido às não linearidades intrínsecas ao robô, portanto utilizaremos como parâmetro a resposta do sistema a um estado inicial $\mathbf{x}(0)$ de:

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0,15 \\ 0 \\ 0 \end{bmatrix} \quad (84)$$

Que é o mesmo que analisar a resposta do robô quando ele é iniciado aproximadamente $8,6^\circ$ inclinado em relação à vertical. O comportamento desejado é:

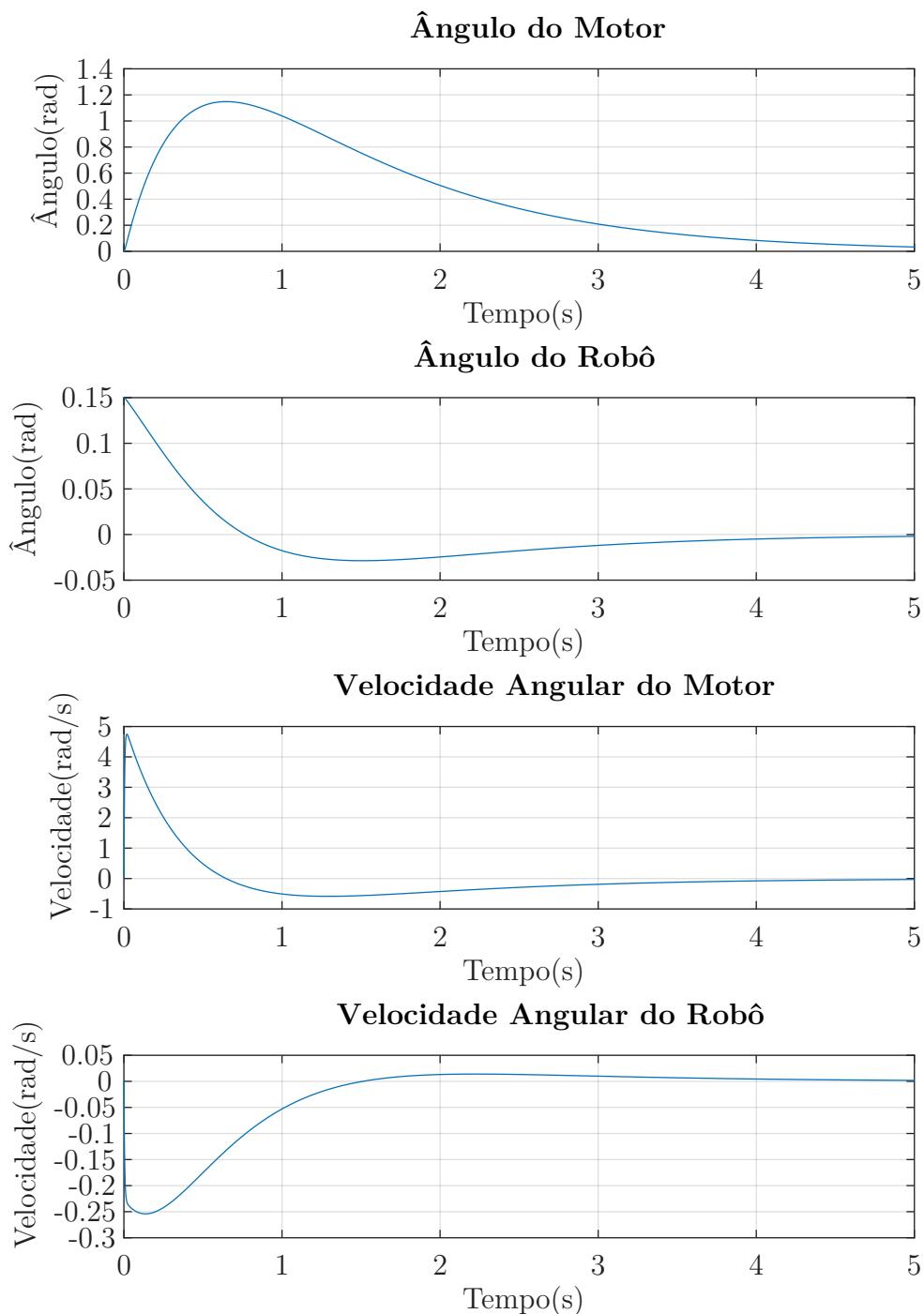
- O ângulo do robô se mantenha pequeno o suficiente para que o modelo linearizado seja próximo do real.
- A ação de controle esperada não seja grande demais, de maneira que não sature os atuadores(motores).
- A ação de controle esperada não seja pequena demais, já que os testes de campo mostraram que o motor não responde linearmente a tensões pequenas.
- A velocidade de resposta esperada pelo motor esteja dentro das capacidades mecânicas dele.

Primeiro assumimos que:

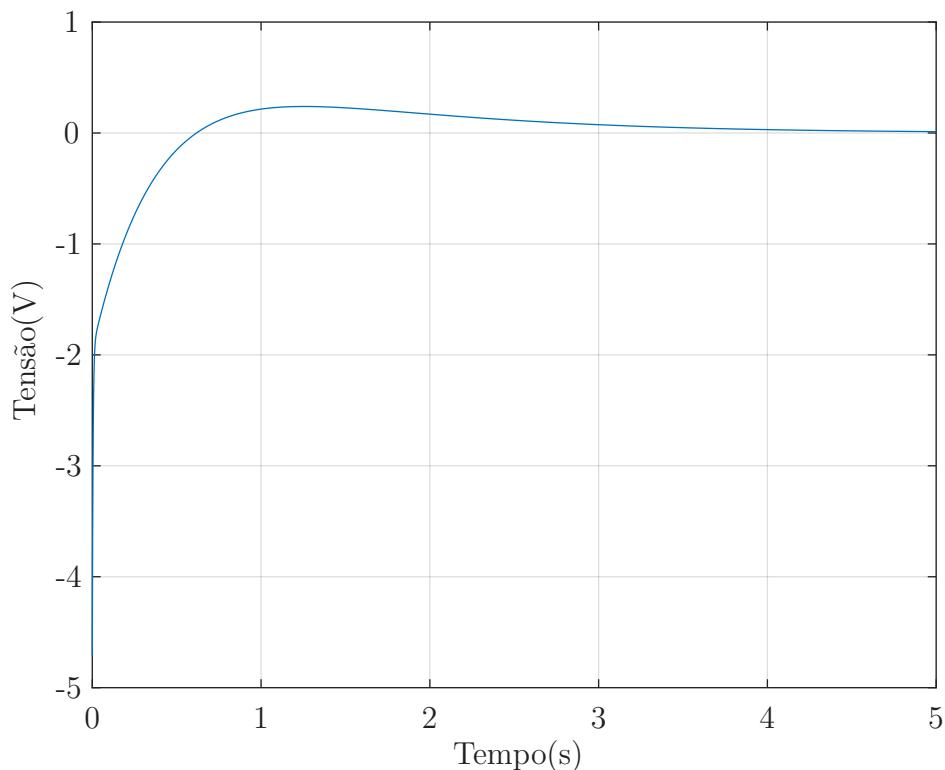
$$\mathbf{R} = 1$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (85)$$

A simulação feita com o controle gerado pode ser vista na Figura 29.

Figura 29 – Resposta do primeiro controle ao estado inicial $\mathbf{x}(0)$.

(a) Variáveis de estado.



(b) Ação de controle.

Fonte: O Autor.

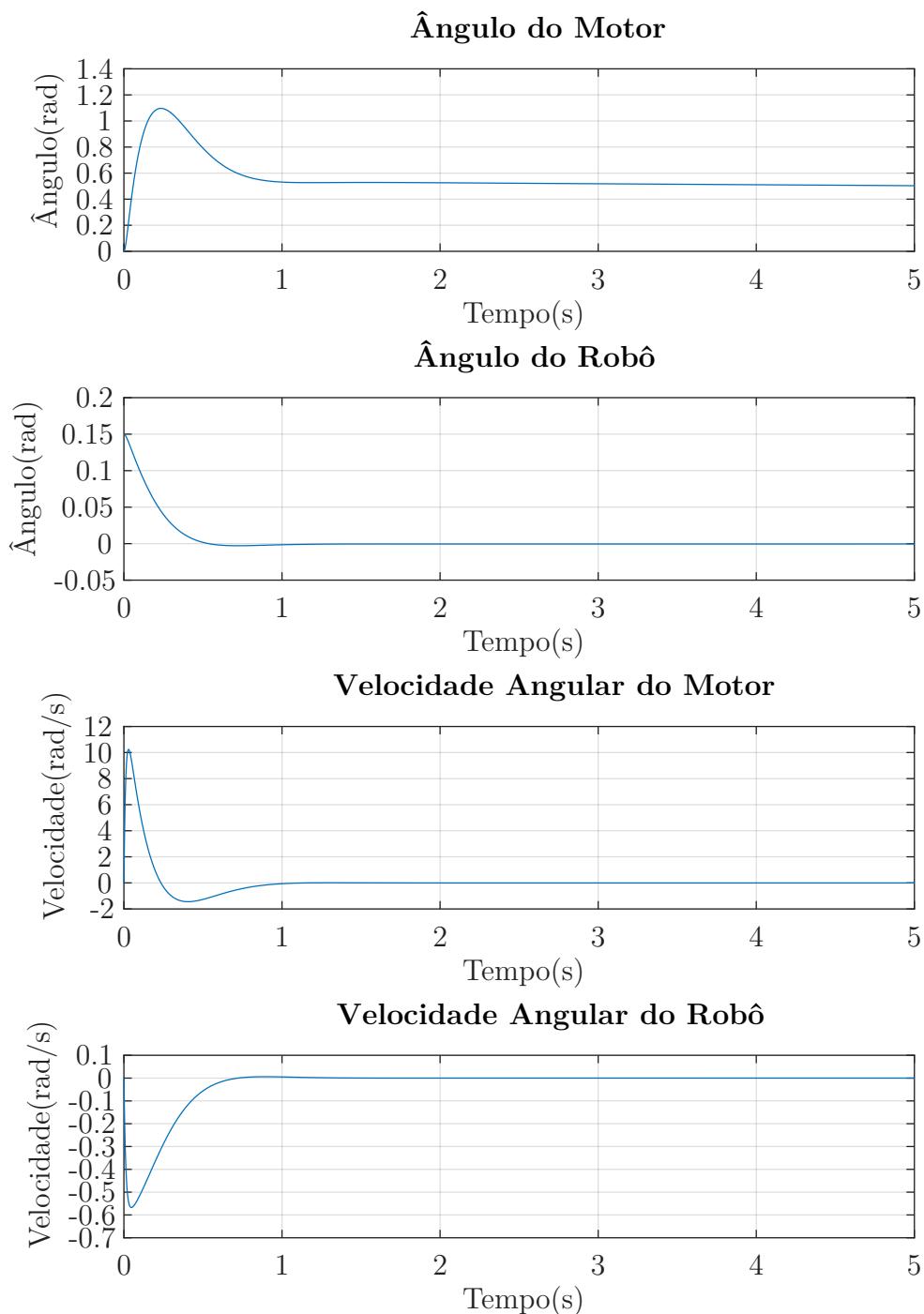
O resultado encontrado apresenta uma resposta muito rápida do motor que não pode ser realizada na prática, além de gerar uma ação de controle pequena, portanto, os valores devem ser modificados. O \mathbf{R} deve ser reduzido para aumentar a ação de controle, enquanto os valores da matriz \mathbf{Q} podem ser ajustados com base na importância dos estados no comportamento do robô.

Como as velocidades angulares irão se estabilizar junto aos seus respectivos ângulos, os valores de q_3 e q_4 podem ser modificados para 0. Para que o robô não saia da região linear é importante que o controle responda mais rapidamente ao ângulo do robô, logo q_2 deve ser consideravelmente maior que q_1 , escolha que também reduz a velocidade de resposta esperada pelo motor. Com base nisso os novos valores são:

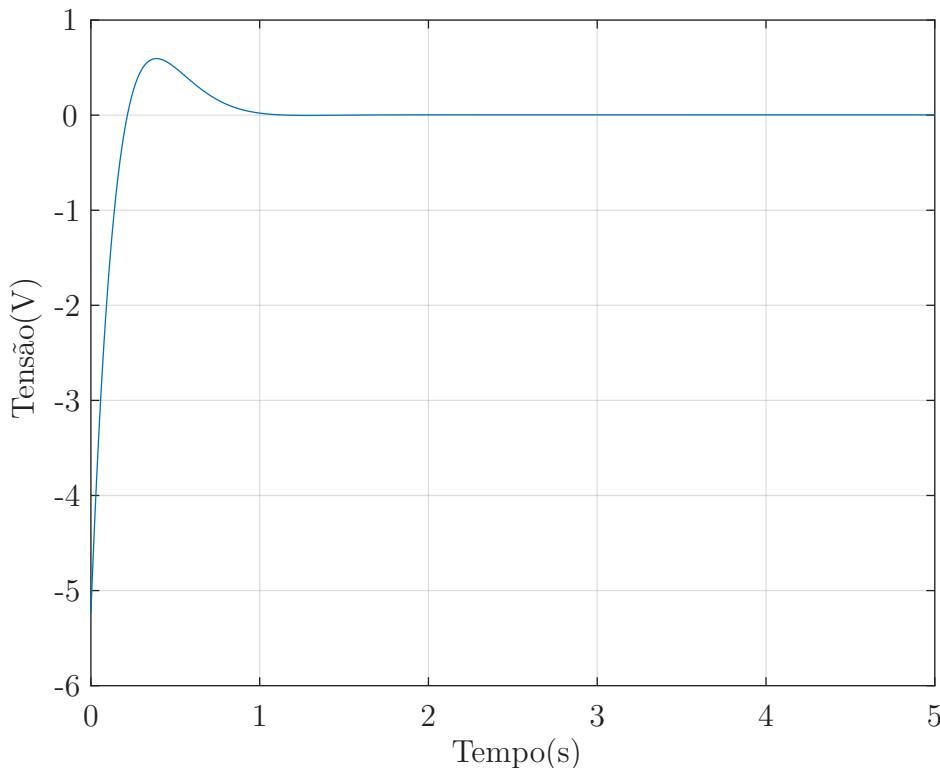
$$\mathbf{R} = 0.1$$

$$\mathbf{Q} = \begin{bmatrix} 0,0001 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (86)$$

O resultado obtido com esses novos valores pode ser visto na Figura 30.

Figura 30 – Resposta do segundo controle ao estado inicial $x(0)$.

(a) Variáveis de estado.



(b) Ação de controle.

Fonte: O Autor.

É perceptível pela resposta apresentada que o sobressinal do ângulo do robô é bem pequeno, característica que garante que o sistema não irá sair da região linear tão facilmente. O controle desenvolvido tem um tempo de acomodação do ângulo do motor bem grande, entretanto, o robô se manterá estável. Desempenhos melhores para o ângulo do motor arriscam desestabilizar o robô por requerirem respostas mais rápidas, que o sistema não consegue entregar.

O vetor de realimentação encontrado para o controle é:

$$\mathbf{K} = [0,031131 \quad 34,938 \quad 0,21227 \quad 5,5981] \quad (87)$$

A ação de controle por realimentação de estados é feita através dos valores das variáveis de estado do sistema, entretanto apenas duas das variáveis estão sendo medidas (o ângulo do motor e o ângulo robô). De modo a determinar as outras duas variáveis (velocidade angular do robô e do motor) um observador de estados será implementado.

Como os dois estados que conseguem ser medidos não apresentam necessidade de serem filtrados, a implementação de um observador de estado de ordem plena é redundante, logo se optou pelo desenvolvimento de um observador de ordem reduzida, que permite com que o cálculo e a implementação do observador sejam simplificados.

Com base nos espaço de estado do sistema os valores das matrizes da Equação (22) são:

$$\begin{aligned}
\mathbf{A}_{aa} &= \begin{bmatrix} 1 & -3,981 \cdot 10^{-5} \\ 0 & 1 \end{bmatrix} \\
\mathbf{A}_{ab} &= \begin{bmatrix} 0,001838 & -5,49 \cdot 10^{-5} \\ & 6,571 \cdot 10^{-6} & 0,002002 \end{bmatrix} \\
\mathbf{A}_{ba} &= \begin{bmatrix} 0 & -0,03885 \\ 0 & 0,0158 \end{bmatrix} \\
\mathbf{A}_{bb} &= \begin{bmatrix} 0,8426 & -0,05341 \\ 0,006391 & 1,002 \end{bmatrix} \\
\mathbf{B}_a &= \begin{bmatrix} -0,0004045 \\ 1,878 \cdot 10^{-5} \end{bmatrix} \\
\mathbf{A}_b &= \begin{bmatrix} -0,3935 \\ 0,01833 \end{bmatrix}
\end{aligned} \tag{88}$$

A escolha do ganho \mathbf{L} do observador pode ser facilmente feita pelo método de alocação de polos. Ao escolhermos polos estáveis e suficientemente rápidos o observador não irá ter problema em aproximar as variáveis de estados. Como visto em Seção 2.4 os polos do observador devem ser mais rápidos que os polos do sistema. Tomaremos como base os polos do sistema com o controlador:

$$\mathbf{z}_c = \begin{bmatrix} 0,84494 \\ 0,99997 \\ 0,98979 + 0,0069076i \\ 0,98979 - 0,0069076i \end{bmatrix} \tag{89}$$

A constante de tempo de um polo discreto é definida como sendo (PHILLIPS; NAGLE; CHAKRABORTTY, 2015):

$$\tau_z = \frac{-T}{\ln r} \tag{90}$$

Sendo T o período de aquisição do sinal e r o módulo do polo.

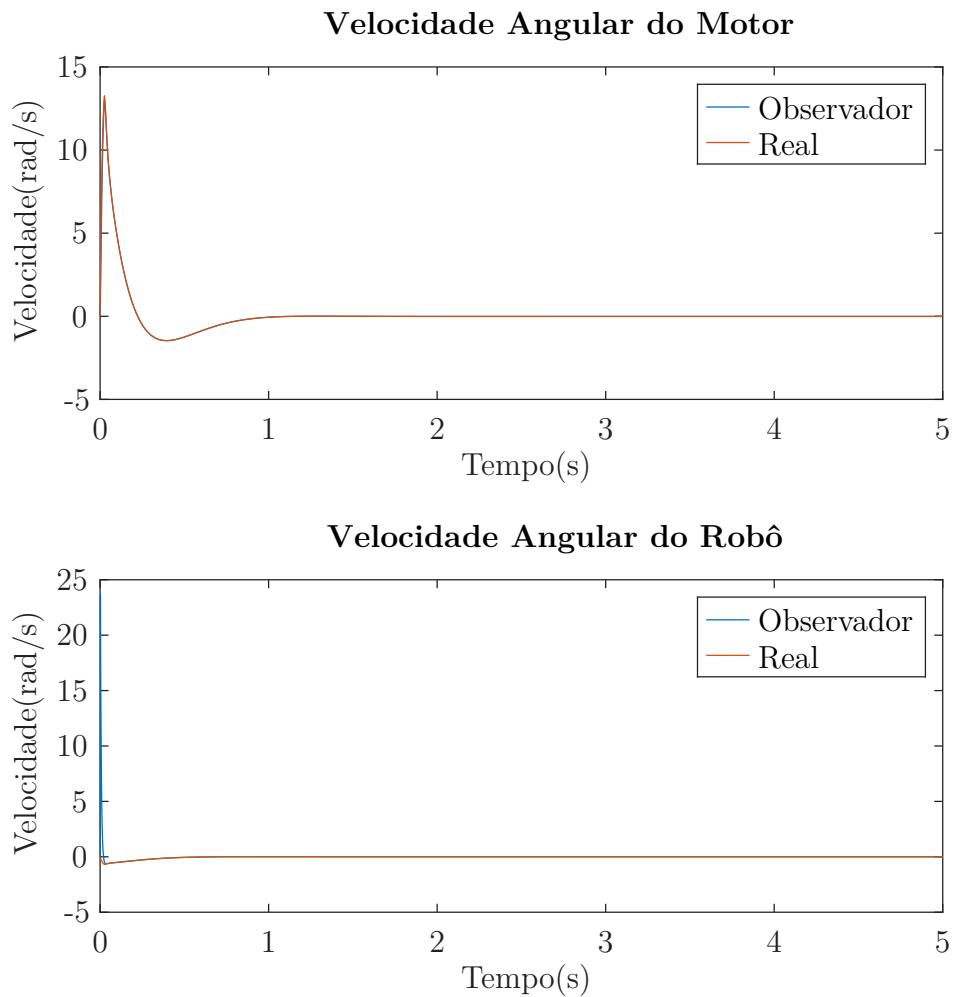
A constante de tempo mais rápida está no polo 0,84494 com um valor de 0,01187, assumindo uma constante de tempo duas vezes menor e fazendo o processo inverso chegamos a um polo em 0,71392. Alocando ambos os polos do observador neste valor determinamos o ganho do observador:

$$\mathbf{L} = \begin{bmatrix} 53,456 & -7,1241 \\ -11,069 & 158,64 \end{bmatrix} \tag{91}$$

6 RESULTADOS E DISCUSSÃO

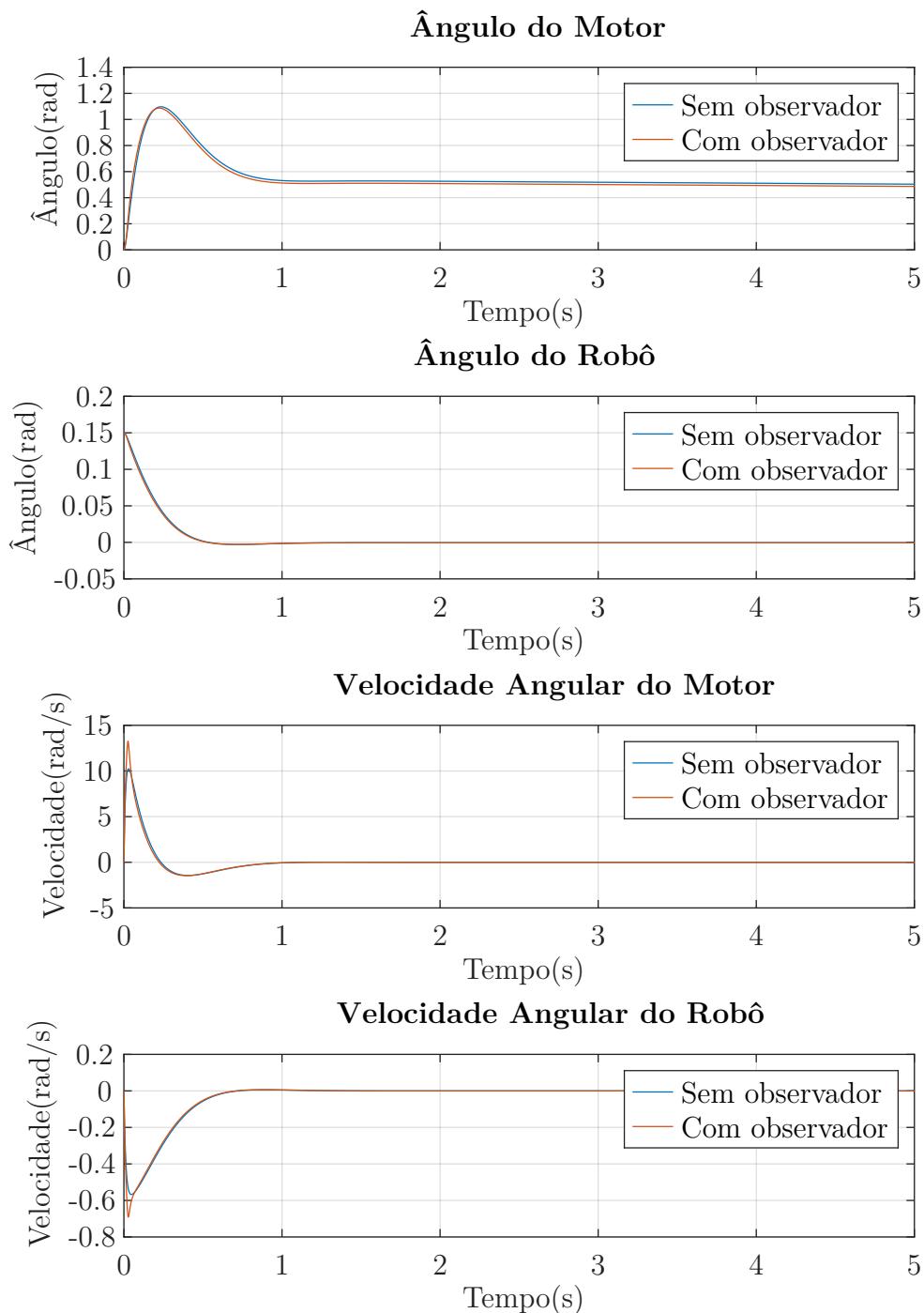
Finalizado o desenvolvimento do controle foram feitos testes para verificar a qualidade do controle desenvolvido. Este capítulo visa apresentar e discorrer sobre os resultados obtidos.

Com o intuito de simular a diferença da adição do observador de estados no sistema, foi desenvolvido um algoritmo no GNU Octave. O código é baseado nas equações de diferença do sistema e do observador, o algoritmo determina a ação de controle $u(k)$ através da realimentação dos valores de $y(k)$ e $\hat{x}_b(k)$, e a partir desse valor determina todas as outras variáveis utilizando as equações do sistema e do observador. Também foi adicionado uma saturação à ação de controle a fim de proporcionar que o comportamento da simulação se aproxime mais ao do sistema real. O resultado encontrado pode ser visto na Figura 31.

Figura 31 – Simulação do controle com observador $\mathbf{x}(0)$.

(a) Variáveis de estado estimadas pelo observador.

Fonte: O Autor.



- (b) Diferença no comportamento do sistema em simulação quanto ao uso do observador de estados.

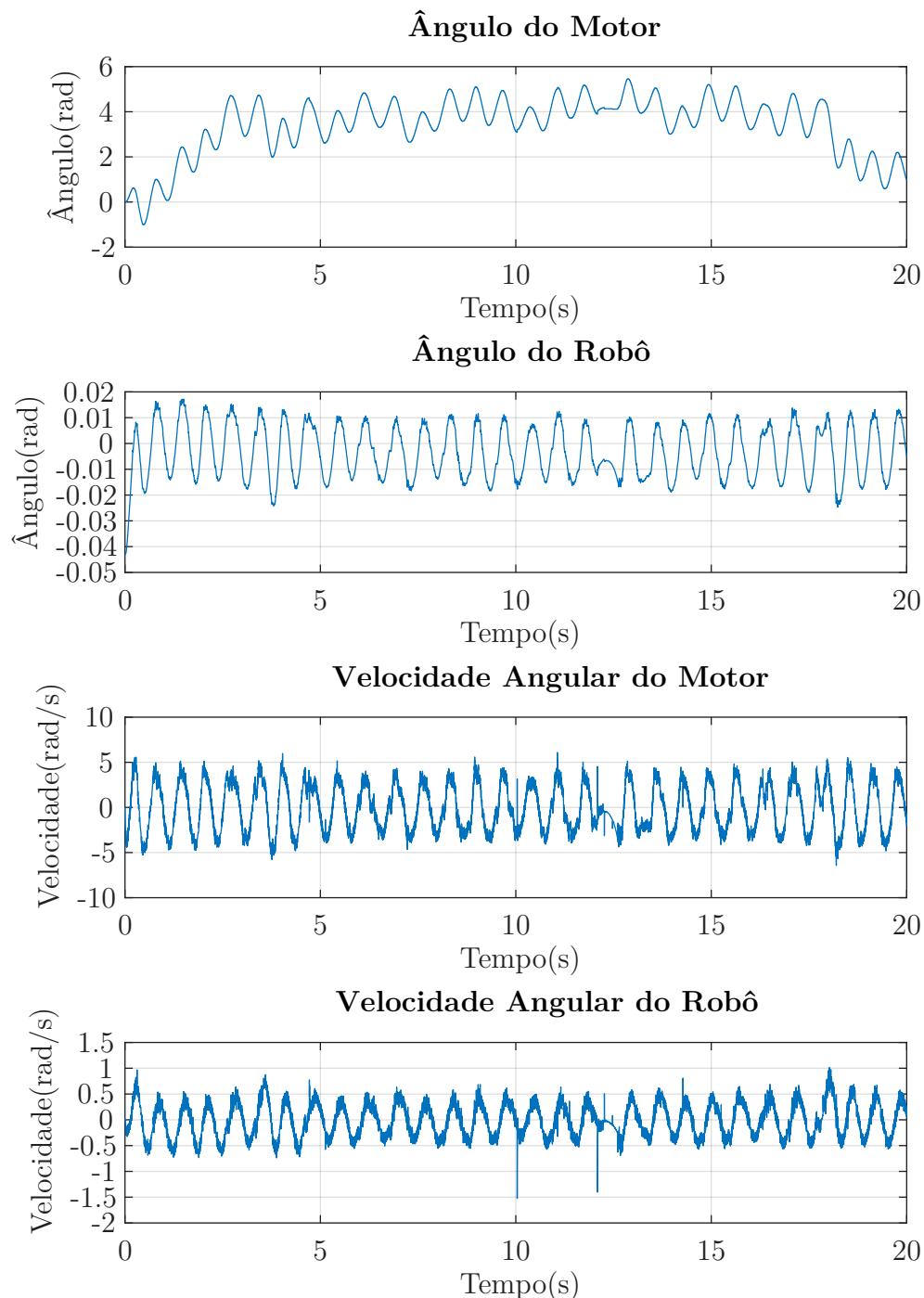
Fonte: O Autor.

O resultado mostra que o observador consegue se estabilizar na medida real rapidamente, sem apresentar problemas para acompanhar as mudanças causadas pelo controle,

logo, o conjunto controlador e observador consegue estabilizar o sistema com sucesso.

O controle desenvolvido foi implementado na planta real com o objetivo de validar os resultados simulados, o resultado obtido pode ser visto na Figura 32.

Figura 32 – Resultado prático.



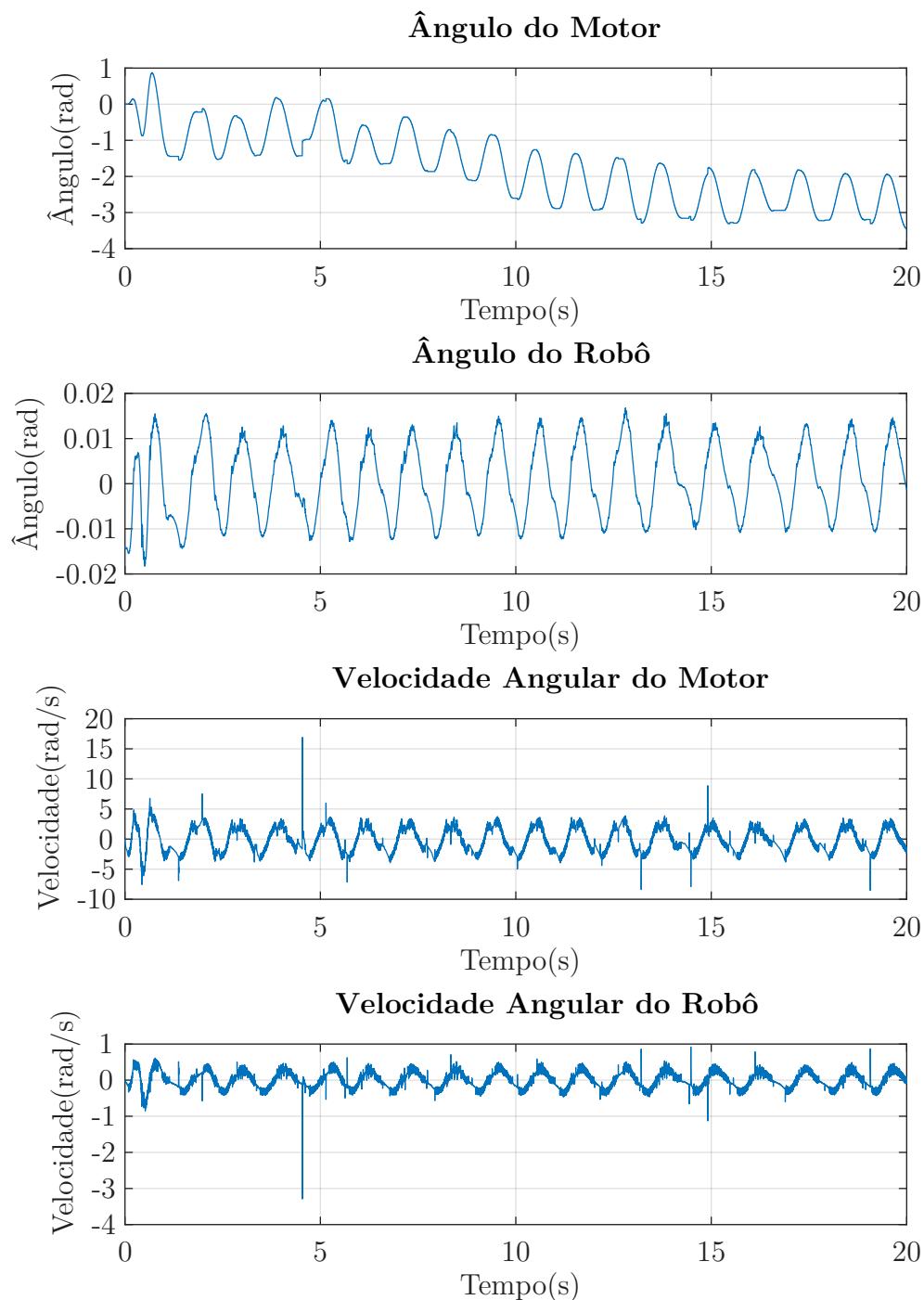
Fonte: O Autor.

Embora o controle criado mantenha o robô estabilizado, a frequência de oscilação do robô está bem elevada, isso é devido a não termos dado nenhuma importância à velocidade angular do robô através da matriz \mathbf{Q} . Adicionando um valor de $q_4=0,5$ e refazendo todos os cálculos chegamos a:

$$\begin{aligned}\mathbf{K} &= \begin{bmatrix} 0,031041 & 34,784 & 0,21666 & 5,9625 \end{bmatrix} \\ \mathbf{L} &= \begin{bmatrix} 57,861 & -6,2975 \\ -10,432 & 162,57 \end{bmatrix}\end{aligned}\tag{92}$$

Utilizando estes valores no controle obtemos o resultado da Figura 33.

Figura 33 – Resultado prático após o ajuste.



Fonte: O Autor.

Com este controle o robô consegue se manter estável, mas com uma frequência de oscilação bem menor, que melhora a resposta do robô a distúrbios externos. O objetivo proposto foi alcançado, manter o robô estabilizado próximo ao ponto de equilíbrio dele.

Os resultados práticos divergem do simulado de tal forma que não é possível estar fazendo uma comparação direta entre os dois resultados, isso se devem principalmente por conta das não linearidades do motor que fazem com que o sistema não seja linear para baixas tensões, ou seja, conforme o robô se aproxima do equilíbrio o motor começa a parar de girar, por conta disso o comportamento do robô acaba sendo uma senoide, esse resultado é comum e pode ser visto, por exemplo, em Ooi (2003), Juang e Lum (2013) e Wasif *et al.* (2013). Entretanto, se as não linearidades do motor fossem consideradas, o comportamento poderia ser ainda menos oscilatório.

Outra consideração importante a ser feita é o controle do ângulo do motor. É perceptível pelos resultados obtidos que por conta de alguma interferência no teste feito o robô acabou saindo da posição inicial. Entretanto, como o ângulo do motor tem uma importância bem pequena o sistema foca em se manter estabilizado e não retorna para a posição inicial de imediato, entretanto como visto anteriormente uma importância maior poderia causar a instabilidade do sistema real. Essa é uma limitação que poderia ser resolvida com um modelo não linear ou com um método de controle diferente.

Uma característica que chama atenção no observador é a quantidade de ruído que ele apresenta, os valores das variáveis de estado são refletidos na ação de controle, logo a ação de controle é afetada por esse ruído. Por exemplo, existe um momento próximo de 5 segundos em que ocorre um ruído bem grande nas variáveis observadas, o que acaba causando um pico na ação de controle. Para evitar essas variações seria ideal fazer uma filtragem do sinal do observador.

7 CONCLUSÃO

Este trabalho teve êxito em identificar o espaço de estados linearizado do robô auto-equilibrado e desenvolver um controle que conseguisse solucionar a instabilidade do sistema. Ainda que o objetivo tenha sido alcançado, o comportamento do robô se mostrou oscilatório e o controle desenvolvido apresenta dificuldade em controlar a posição do robô, características que seriam problemáticas caso outros comportamentos fossem esperados do sistema, como o controle de posição ou a locomoção em terreno irregulares. Entretanto, esta é uma limitação do uso do controle linear em um sistema que apresenta diversas não linearidades, para que resultados melhores sejam alcançados é necessário considerar as não linearidades no desenvolvimento do modelo matemático e utilizar métodos que consigam trabalhar com essas características com mais eficiência.

Embora todo o estudo e desenvolvimento do controle tenha sido feito em volta do robô auto-equilibrado, a metodologia utilizada pode ser aplicada em qualquer sistema, desde que o espaço de estados do sistema seja identificado de maneira correta. As táticas aqui mostradas poderão ser utilizadas para estabilizar o sistema, entretanto devem ser observadas as limitações intrínsecas ao controle linear, como foi visto aqui.

Trabalhos futuros devem buscar encontrar melhores maneiras de descrever a planta de controle. Poderiam ser consideradas outras características da planta na modelagem, como, por exemplo, o atrito estático no motor ou o chaveamento realizado pela ponte H. É recomendado que seja feito o controle não linear da planta ou que outros métodos de linearização sejam utilizados, como é o exemplo da linearização por realimentação (CHAN; STOL; HALKYARD, 2013; FRANKLIN *et al.*, 2015), a fim de buscar resultados melhores. Como foi encontrado um controle que estabiliza a planta, podem ser utilizados métodos de caixa preta para se chegar a uma melhor aproximação linear do comportamento do sistema, abrindo a possibilidade de um estudo envolvendo identificação de sistemas e a procura por uma melhor representação da planta.

Outra sugestão para trabalhos futuros é adicionar novas função ao robô, tentando fazer controle de posição, velocidade, trajetória e também buscando aperfeiçoar a resposta do robô a pertubações, permitindo com que ele trabalhe em ambientes mais hostis, com terrenos irregulares e interferências. Entretanto, seria necessário um aprimoramento no *hardware* e no *software* do robô para que ele consiga fazer as medições e o controle de forma eficiente enquanto realiza todas as suas outras novas funções.

REFERÊNCIAS

- ADEEL, Umar; ALIMGEER, KS; INAM, Omair; HAMEED, Ayesha; QURESHI, Mehmood; ASHRAF, Mehmood. Autonomous dual wheel self balancing robot based on microcontroller. **Journal of Basic and Applied Scientific Research**, v. 3, n. 1, p. 843–848, 2013.
- AGUIRRE, L. **Controle de Sistemas Amostrados**. [S.l.]: Independently Published, 2019. ISBN 9781799052081.
- BAÚ DA ELETRÔNICA. **Motor DC 6V com Encoder 210 RPM**. Disponível em: <https://www.baudaelectronica.com.br/motor-dc-6v-com-encoder-210-rpm.html>. Acesso em: 5 nov. 2022.
- CASTRUCCI, Plinio de Lauro; BITTAR, Anselmo; SALES, Roberto Moura. **Controle Automático**. 2. ed. Rio de Janeiro: LTC, 2018.
- CHAN, Ronald Ping Man; STOL, Karl A; HALKYARD, C Roger. Review of modelling and control of two-wheeled robots. **Annual reviews in control**, Elsevier, v. 37, n. 1, p. 89–103, 2013.
- CHRISTOPHER, Roche. **KalmanFilterPython**. 2020. Disponível em: <https://github.com/rocheparadox/Kalman-Filter-Python-for-mpu6050>. Acesso em: 5 nov. 2022.
- DELTA COMPUTER SYSTEMS. **Quadrature Encoder Fundamentals**. Disponível em: https://deltamotion.com/support/webhelp/rmc/tools/index.htm#t=Controller_Features%5C%2FTransducer_Basics%5C%2FQuadrature_Fundamentals.htm. Acesso em: 5 nov. 2022.
- DORF, Richard C; BISHOP, Robert H. **Modern control systems**. 14. ed. Harlow: Pearson Education Limited, 2022.
- DYNAPAR. **Quadrature Encoder Overview**. Disponível em: https://www.dynapar.com/Technology/Encoder_Basics/Quadrature_Encoder/. Acesso em: 5 nov. 2022.

FILIPEFLOP. **Acelerômetro e Giroscópio 3 Eixos 6 DOF MPU-6050.** Disponível em: <https://www.filipeflop.com/produto/acelerometro-e-giroscopio-3-eixos-6-dof-mpu-6050/>. Acesso em: 5 nov. 2022.

FILIPEFLOP. **Driver Motor Ponte H L298n.** Disponível em: <https://www.filipeflop.com/produto/driver-motor-ponte-h-l298n/>. Acesso em: 5 nov. 2022.

FILIPEFLOP. **Raspberry Pi Pico.** Disponível em: <https://www.filipeflop.com/produto/raspberry-pi-pico/>. Acesso em: 5 nov. 2022.

FRANKLIN, Gene F; POWELL, J David; EMAMI-NAEINI, Abbas; POWELL, J David. **Feedback control of dynamic systems.** 7. ed. Harlow: Pearson Education Limited, 2015.

FRANKOVSKY, P; DOMINIK, L; GMITERKO, A; VIRGALA, I; KURYLO, P; PERMINOVA, O. Modeling of two-wheeled self-balancing robot driven by DC gearmotors. **International Journal of Applied Mechanics and Engineering**, v. 22, n. 3, 2017.

FRIEDLAND, Bernard. **Control system design: an introduction to state-space methods.** New York: Courier Corporation, 2012.

GOLDSTEIN, Herbert; POOLE, Charles; SAFKO, John. **Classical Mechanics.** 3. ed. Harlow: Pearson Education Limited, 2014.

GRASSER, Felix; D'ARRIGO, Aldo; COLOMBI, Silvio; RUFER, Alfred C. JOE: a mobile, inverted pendulum. **IEEE Transactions on industrial electronics**, IEEE, v. 49, n. 1, p. 107–114, 2002.

HSU, Chun-Fei; SU, Chien-Ting; KAO, Wei-Fu; LEE, Bore-Kuen. Vision-based line-following control of a two-wheel self-balancing robot. In: IEEE. 2018 International Conference on Machine Learning and Cybernetics (ICMLC). [S.l.: s.n.], 2018. v. 1, p. 319–324.

INVENSENSE INC. **MPU-6000 and MPU-6050 Product Specification Revision 3.4.** California, 2013.

ISERMANN, Rolf; MÜNCHHOF, Marco. **Identification of dynamic systems: an introduction with applications.** [S.l.]: Springer, 2011. v. 85.

JUAN RINCÓN PASAYE, J.; ALBERTO BONALES VALENCIA, J.; JIMÉNEZ PÉREZ, Félix. Tilt measurement based on an Accelerometer, a Gyro and a Kalman Filter to control a self-balancing vehicle. *In:* 2013 IEEE International Autumn Meeting on Power Electronics and Computing (ROPEC). [S.l.: s.n.], 2013. P. 1–5.

JUANG, Hau-Shiue; LUM, Kai-Yew. Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. *In:* IEEE. 2013 10th IEEE International Conference on Control and Automation (ICCA). [S.l.: s.n.], 2013. P. 634–639.

KADIR, Herdawatie Abdul. **Modelling and control of a balancing robot using digital state space approach.** 2005. Tese (Doutorado) – Universiti Teknologi Malaysia.

KIM, Jaekyung; YOO, Jinhyeok; BAE, Jisu; KIM, Changhyun; CHOI, Junwon. Two-Wheeled Self-Balancing Robot Based On LQ-Servo with Reduced Order Observer. *In:* IEEE. 2015 Fourth International Conference on Information Science and Industrial Applications (ISI). [S.l.: s.n.], 2015. P. 6–9.

KUNG, Fabian. A tutorial on modelling and control of two-wheeled self-balancing robot with stepper motor. **Applications of Modelling and Simulation**, v. 3, n. 2, p. 64–73, 2019.

MERCADO LIVRE. **Conversor Dc-dc 12a Fonte Ajustável 5v 9v 12v 24v Step Down.** Disponível em: https://produto.mercadolivre.com.br/MLB-1083204788-conversor-dc-dc-12a-fonte-ajustavel-5v-9v-12v-24v-step-down-_JM?quantity=1. Acesso em: 5 nov. 2022.

NIKITA, T; PRAJWAL, KT. PID Controller Based Two Wheeled Self Balancing Robot. *In:* IEEE. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). [S.l.: s.n.], 2021. P. 1–4.

NISE, Norman S. **Control systems engineering.** 7. ed. New Jersey: John Wiley & Sons, 2015.

NOLTE, David D. **Introduction to modern dynamics: Chaos, networks, space and time.** 2. ed. Oxford: Oxford University Press, 2019.

OGATA, Katsuhiko *et al.* **Modern control engineering**. 5. ed. Harlow: Pearson Education Limited, 2010.

OOI, Rich Chi. Balancing a two-wheeled autonomous robot. **University of Western Australia**, v. 3, 2003.

PALANISAMY, Naveen Prabu. **Filtering of IMU Data Using Kalman Filter**. 2016. Tese (Doutorado) – California State University, Sacramento.

PHILLIPS, Charles L; NAGLE, H Troy; CHAKRABORTTY, Aranya. **Digital control system analysis and design**. 4. ed. Harlow: Pearson Education Limited, 2015.

QINGCHENG, Wang; JIAN, Fang. Fuzzy immune pd algorithm applied in the self-balancing two-wheeled robot. In: IEEE. 2014 8th International Conference on Future Generation Communication and Networking. [S.l.: s.n.], 2014. P. 112–115.

RASPBERRY PI LTD. **Raspberry Pi Pico C/C++ SDK**: Libraries and tools for C/C++ development on RP2040 microcontrollers. Cambridge, 2022.

RASPBERRY PI LTD. **Raspberry Pi Pico Datasheet**: An RP2040-based microcontroller board. Cambridge, 2022.

ROMLAY, MRM; AZHAR, MI; TOHA, SF; RASHID, MM. Two-wheel Balancing Robot; Review on Control Methods and Experiment. **International Journal of Recent Technology and Engineering (IJRTE)**, v. 7, n. 68, p. 106–12, 2019.

RUAN, Xiaogang; CAI, Jianxian. Fuzzy backstepping controllers for two-wheeled self-balancing robot. In: IEEE. 2009 International Asia Conference on Informatics in Control, Automation and Robotics. [S.l.: s.n.], 2009. P. 166–169.

RUAN, Xiaogang; CHEN, Jing. H robust control of self-balancing two-wheeled robot. In: IEEE. 2010 8th World Congress on Intelligent Control and Automation. [S.l.: s.n.], 2010. P. 6524–6527.

RUAN, Xiaogang; LI, Wangbo. Ultrasonic sensor based two-wheeled self-balancing robot obstacle avoidance control system. In: IEEE. 2014 IEEE International Conference on Mechatronics and Automation. [S.l.: s.n.], 2014. P. 896–900.

- RUAN, Xiaogang; ZHAO, Jianwei. The PWM Servo and LQR Control of a Dual-wheel Upright Self-balancing Robot. *In: IEEE. 2008 International Symposiums on Information Processing.* [S.l.: s.n.], 2008. P. 586–590.
- TANGIRALA, Arun K. **Principles of system identification: theory and practice.** [S.l.]: Crc Press, 2018.
- WARDYOYO, Adik S; HENDI, S; SEBAYANG, Darwin; HIDAYAT, Imam; ADRIANSYAH, Andi. An investigation on the application of fuzzy and PID algorithm in the two wheeled robot with self balancing system using microcontroller. *In: IEEE. 2015 International Conference on Control, Automation and Robotics.* [S.l.: s.n.], 2015. P. 64–68.
- WASIF, Ammar; RAZA, Danish; RASHEED, Waqas; FAROOQ, Zubair; ALI, Syed Qaseem. Design and implementation of a two wheel self balancing robot with a two level adaptive control. *In: IEEE. EIGHTH International Conference on Digital Information Management (ICDIM 2013).* [S.l.: s.n.], 2013. P. 187–193.
- WENXIA, Sun; WEI, Chen. Simulation and debugging of LQR control for two-wheeled self-balanced robot. *In: IEEE. 2017 Chinese Automation Congress (CAC).* [S.l.: s.n.], 2017. P. 2391–2395.
- WU, Junfeng; ZHANG, Wanying. Design of fuzzy logic controller for two-wheeled self-balancing robot. *In: IEEE. PROCEEDINGS of 2011 6th international forum on strategic technology.* [S.l.: s.n.], 2011. v. 2, p. 1266–1270.
- XIAOGANG, Ruan; JIANG, Liu; HAIJIANG, Di; XINYUAN, Li. Design and LQ control of a two-wheeled self-balancing robot. *In: IEEE. 2008 27th Chinese Control Conference.* [S.l.: s.n.], 2008. P. 275–279.
- XU, Changkai; LI, Ming; PAN, Fangyu. The system design and LQR control of a two-wheels self-balancing mobile robot. *In: IEEE. 2011 International Conference on Electrical and Control Engineering.* [S.l.: s.n.], 2011. P. 2786–2789.

APÊNDICE A – Descrição

Com o intuito de facilitar o acesso aos arquivos do projeto foi criado um repositório no github, o qual pode ser acessado através deste link:https://github.com/LuanArantes/Robo_Autoequilibrado, lá é possível encontrar:

- Códigos e dados utilizados para a modelagem e análise do projeto.
- O programa implementado no microcontrolador.
- Vídeos do funcionamento do sistema.
- Gráficos associados ao trabalho.