



**Universidade de Brasília
Faculdade do Gama**

**Desenvolvimento de um pêndulo invertido
para ensino a distância**

Italo Marcelo Cardoso Graciano

PROJETO FINAL DE CURSO
ENGENHARIA ELETRÔNICA

Brasília
2023

**Universidade de Brasília
Faculdade do Gama**

**Desenvolvimento de um pêndulo invertido
para ensino a distância**

Italo Marcelo Cardoso Graciano

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro Eletrônico

Orientador: Prof. Dr. Guillermo Alvarez Bestard

Brasília
2023

G731d Graciano, Italo Marcelo Cardoso.
Desenvolvimento de um pêndulo invertido para ensino a distância / Italo Marcelo Cardoso Graciano; orientador Guillermo Alvarez Bestard. -- Brasília, 2023.
88 p.

Projeto Final de Curso (Engenharia Eletrônica) -- Universidade de Brasília, 2023.

1. Pêndulo invertido. 2. Laboratório de controle. 3. Sistemas embarcados. I. Bestard, Guillermo Alvarez, orient. II. Título

Universidade de Brasília
Faculdade do Gama

**Desenvolvimento de um pêndulo invertido para ensino a
distância**

Italo Marcelo Cardoso Graciano

Projeto Final de Curso submetido como requi-
sito parcial para obtenção do grau de Enge-
nheiro Eletrônico

Trabalho aprovado. Brasília, 26 de julho de 2023:

Prof. Dr. Guillermo Alvarez Bestard,
UnB/FGA
Orientador

Prof. Dr. Daniel Mauricio Muñoz
Arboleda, UnB/FGA
Examinador interno

Prof. Dr. Gilmar Silva Beserra, UnB/FGA
Examinador interno

Brasília
2023

Agradecimentos

Agradeço a Deus e a todos que me ajudaram nesta caminhada, especialmente a minha família.

Resumo

O pêndulo invertido tem sido um sistema amplamente utilizado e que encontra várias aplicações tanto no ensino quanto na indústria. Várias disciplinas como sistemas de controle, física, sistemas embarcados e outras correlatas são exercitadas durante a produção do pêndulo invertido e no seu controle, desse modo, um laboratório de pêndulo invertido pode melhorar o ensino destas disciplinas ao promover um contato prático e aplicado destes conhecimentos. Além do ensino tradicional estes laboratórios podem ser modificados para se tornarem laboratórios remotos, inserindo assim elementos de conectividade e acesso remoto da indústria 4.0 nos currículos das matérias tradicionais dos cursos de engenharia. Este trabalho tem como objetivo projetar e criar um pêndulo invertido rotacional que poderá ser utilizado como experimento de laboratório nas disciplinas de sistemas embarcados e sistemas de controle, além de criar neste experimento subsistemas que possibilitem uma futura adaptação do pêndulo invertido como laboratório remoto. Durante o desenvolvimento do projeto foram projetados e desenvolvidos componentes de estrutura, eletrônica, *software* e controle que resultaram na elaboração de um experimento do pêndulo invertido rotacional controlado por PID. O pêndulo desenvolvido possibilita a execução de experimentos presenciais, além de possuir subsistemas que ajudarão na adaptação para uso como experimento remoto em versões futuras.

Palavras-chaves: Laboratório de controle, Pêndulo invertido, Sistemas de controle, Sistemas embarcados.

Abstract

The inverted pendulum has been widely used and finds various applications in both education and industry. It involves multiple disciplines, such as control systems, physics, embedded systems, and related fields, which are actively exercised during its production and control. Consequently, an inverted pendulum laboratory can enhance the teaching of these disciplines by facilitating practical and applied learning experiences. Additionally, these laboratories can be adapted into remote setups, incorporating elements of Industry 4.0's connectivity and remote access into traditional engineering curricula. The main objective of this work is to design and create a rotational inverted pendulum for use as a laboratory experiment in embedded systems and control courses. The experiment will also include subsystems enabling future adaptation of the inverted pendulum as a remote laboratory. Throughout the project development, components for the structure, electronics, software, and control have been designed and developed, resulting in the creation of a PID-controlled rotational inverted pendulum experiment. The developed pendulum allows for conducting in-person experiments and possesses subsystems that will facilitate its adaptation for use as a remote experiment in future versions.

Key-words: Control Laboratory, Inverted Pendulum, Control Systems, Embedded Systems.

Lista de ilustrações

Figura 2.1 – Agendamento de laboratório no RELLE	19
Figura 2.2 – Modelo físico de um pêndulo invertido rotacional	20
Figura 2.3 – Controle de velocidade utilizando malha aberta	21
Figura 2.4 – Controle de velocidade utilizando malha fechada	21
Figura 2.5 – Malha de controle do controlador <i>fuzzy</i>	22
Figura 2.6 – Malha de controle com rede neural e controlador PID	23
Figura 2.7 – Diagrama geral de um controlador PID no domínio de Laplace	23
Figura 2.8 – Interface gráfica	25
Figura 2.9 – Interface gráfica de acesso ao pêndulo invertido	25
Figura 2.10–Interface de controle do laboratório remoto da Universidade de Hanôver	26
Figura 2.11–Interface de acesso aos 3 experimentos	26
Figura 2.12–Controle por ajuste do controlador PID	27
Figura 2.13–Configuração de controlador em FPGA	27
Figura 3.14–Diagrama do uso do laboratório	29
Figura 3.15–Comunicação entre usuário e o experimento	29
Figura 3.16–Vista superior do pêndulo invertido	30
Figura 3.17–Vista lateral do CAD do pêndulo invertido	30
Figura 3.18–Peça de junção abaixo da base móvel	31
Figura 3.19–Rotação indesejada da base móvel	32
Figura 3.20–Vista frontal do pêndulo invertido	32
Figura 3.21–Vista diagonal do CAD do pêndulo	33
Figura 3.22–Funcionamento do <i>encoder</i> óptico	34
Figura 3.23–Onda quadrada de saída do <i>encoder</i>	34
Figura 3.24– <i>Encoder</i> LPD3806	35
Figura 3.25–Deslocador de nível lógico de 4 canais	35
Figura 3.26–Módulo ponte H	36
Figura 3.27–Câmera rev 1.3 e cabo CSI	36
Figura 3.28–Diagrama da conexão eletrônica	37
Figura 3.29–Circuito projetado no Altium	38
Figura 3.30–Placa <i>driver</i> projetada no Altium	39
Figura 3.31–Placa furada construída	39
Figura 3.32–pinos e periféricos da MSP430F5529	40
Figura 3.33–Fluxo do experimento na MSP	41
Figura 3.34–Dados transmitidos pela MSP	42
Figura 3.35–Sentido de movimentação da base e sentido do erro de angulo da barra .	45
Figura 3.36–Roda com elástico	50

Figura 3.37–Borracha com grampos produzida	51
Figura 3.38–Roda com grampo	51
Figura 3.39–Tira de borracha produzida	52
Figura 3.40–Roda com borracha	52
Figura 3.41–Roda com borracha sobre papelão	52
Figura 4.42–Subida com motores sempre ligados	55
Figura 4.43–Subida com motores ligados abaixo da linha horizontal	56
Figura 4.44–Subida com dosagem da força	57
Figura 4.45–Subida com PID. $K_p = 2.1, K_i = 0, K_d = 100$	57
Figura 4.46–Subida com PID. $K_p = 2.2, K_i = 0, K_d = 100$	58
Figura 4.47–Gráfico do pêndulo em queda	59
Figura 4.48–Gráfico do pêndulo com função degrau	59
Figura 4.49–Função degrau no sentido horário com rodas levantadas	60
Figura 4.50–Função degrau no sentido anti-horário com rodas levantadas	60
Figura 4.51–Função degrau no sentido horário com rodas impulsionando	61
Figura 4.52–Função degrau no sentido anti-horário com rodas impulsionando	61
Figura 4.53–Velocidade das rodas levantadas	62
Figura 4.54–Velocidade das rodas impulsionando a base móvel	63
Figura 4.55–Teste de inversão com rodas levantadas	64
Figura 4.56–Teste de inversão com rodas sem alteração impulsionando a base	64
Figura 4.57–Teste de inversão com rodas com elásticos impulsionando a base	65
Figura 4.58–Teste de inversão com rodas com grampos impulsionando a base	65
Figura 4.59–Teste de inversão com rodas com borracha	66
Figura 4.60–Teste de inversão com rodas com borracha sobre papelão	66
Figura 4.61–Controle com subida manual com $K_p = 100, K_i = 0, K_d = 0$	68
Figura 4.62–Controle com subida manual com $K_p = 150, K_i = 0, K_d = 0$	69
Figura 4.63–Controle com subida manual com $K_p = 180, K_i = 0, K_d = 0$	69
Figura 4.64–Controle com subida manual com $K_p = 150, K_i = 0,001, K_d = 0$	70
Figura 4.65–Controle com subida manual com $K_p = 150, K_i = 0,0001, K_d = 0$	71
Figura 4.66–Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 0$	71
Figura 4.67–Controle com subida manual com $K_p = 150, K_i = 0,000001, K_d = 0$	72
Figura 4.68–Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 10$	73
Figura 4.69–Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 100$	74
Figura 4.70–Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 1000$	74
Figura 4.71–Controle com subida por PID com $K_p = 100, K_i = 0, K_d = 0$	75
Figura 4.72–Controle com subida por PID com $K_p = 200, K_i = 0, K_d = 0$	76
Figura 4.73–Controle com subida por PID com $K_p = 300, K_i = 0, K_d = 0$	76
Figura 4.74–Controle com subida por PID com $K_p = 500, K_i = 0, K_d = 0$	77
Figura 4.75–Controle com subida por PID com $K_p = 500, K_i = 0,001, K_d = 0$	77

Figura 4.76–Controle com subida por PID com $K_p = 500, K_i = 0,0001, K_d = 0$	78
Figura 4.77–Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 0$	78
Figura 4.78–Controle com subida por PID com $K_p = 500, K_i = 0,000001, K_d = 0$	79
Figura 4.79–Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 100$	79
Figura 4.80–Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 1000$	80
Figura 4.81–Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 10000$	80

Lista de tabelas

Tabela 3.1 – Componentes eletrônicos utilizados	33
Tabela 3.2 – Sentido de rotação dos motores controlados pela ponte H	36
Tabela 4.3 – Velocidade média das rodas	62
Tabela 4.4 – Tempos médios de subida e descida dos testes	67

Lista de abreviaturas e siglas

CA	Corrente alternada	18
CAD	<i>Computer Aided Design</i>	30
CC	Corrente contínua	18
CCS	<i>Code Composer Studio</i>	40
CSI	<i>Camera serial interface</i>	35
DC	<i>Direct Current</i>	26
DID	<i>Dynamic importance degree</i>	22
EAD	Ensino a distância	18
FGA	Universidade de Brasília Campus Gama	15
FOV	<i>Field of view</i>	36
FPGA	<i>Field Programmable Gate Array</i>	26
FSF	<i>Full State Feedback</i>	24
IFSC	Instituto Federal campus Santa Catarina	18
JTAG	<i>Joint Test Action Group</i>	26
LQR	<i>Linear Quadratic Regulator</i>	24
OSF	<i>Output Scale Factor</i>	22
PID	Controlador proporcional integral derivativo	15
PUC-Rio	Pontifícia Universidade Católica do Rio de Janeiro	18
PWM	<i>Pulse Width Modulation</i>	35
RELLE	<i>Remote Lab Learning Environment</i>	19
SCP	<i>Secure Copy Protocol</i>	43
SIRM	<i>Single input rule module</i>	22
STEM	<i>Science, Technology, Engineering e Mathematics</i>	18
UART	<i>Universal Asynchronous Receiver/Transmitter</i>	36
UERGS	Universidade Federal do Estado do Rio Grande do Sul	17
UFSC	Universidade Federal de Santa Catarina	18
USB	<i>Universal Serial Bus</i>	40
VHDL	<i>VHSIC Hardware Description Language</i>	26

Lista de símbolos

Símbolos romanos

\hat{x}	Velocidade nas direções x	20
\hat{y}	Velocidade nas direções y	20
L_p	Distância do centro de massa até o braço.....	20
r	comprimento do braço.....	20

Símbolos gregos

α	Ângulo de rotação da barra	20
$\dot{\alpha}$	Velocidade angular da barra.....	20
$\dot{\theta}$	Velocidade angular do braço	20
θ	Ângulo de rotação do braço	20

Sumário

1	Introdução	15
1.1	Objetivos gerais	15
1.2	Objetivos específicos	15
1.3	Organização do texto	16
2	Referencial teórico	17
2.1	Indústria 4.0 e Educação 4.0	17
2.1.1	Laboratórios remotos	18
2.2	Pêndulo invertido e seu controle	19
2.2.1	Controle do pêndulo invertido	21
2.2.2	Laboratórios remotos do pêndulo invertido	24
3	Metodologia	28
3.1	Requisitos funcionais	28
3.2	Requisitos não funcionais	28
3.3	Solução proposta	28
3.3.1	Solução mecânica	30
3.3.2	Solução eletrônica	33
3.3.3	Solução de <i>software</i>	42
3.3.4	Sistema de controle	43
3.4	Metodologia do desenvolvimento do projeto	46
3.4.1	Estrutura	46
3.4.2	Eletrônica	46
3.4.3	<i>Software</i>	47
3.5	Versões do pêndulo	48
3.6	Testes do pêndulo	49
3.6.1	Teste de derrapagem	50
3.6.2	Teste de estabilização	53
4	Análise dos resultados	55
4.1	Métodos para subir a barra	55
4.2	Testes do pêndulo	58
4.2.1	Testes com a barra do pêndulo	58
4.2.2	Testes de velocidade das rodas	60
4.2.3	Testes de derrapagem	63
4.2.4	Testes de estabilização	68

4.3	Mecânica	81
4.4	Eletrônica	81
4.5	Controle	82
4.6	<i>Software</i>	83
5	Conclusão	84
	Referências	86

1 Introdução

Um dos conceitos mais importantes para a Indústria 4.0 é a conectividade, tanto entre dispositivos como entre pessoas. A Indústria 4.0 trará uma revolução nos meios produtivos e na sociedade como um todo, e para estar adequado a essa nova realidade será necessário nos preparar desde agora. A atualização dos currículos para incluir os novos conceitos da Indústria 4.0 será de grande importância para que seja possível formar os trabalhadores que irão atuar nesta indústria. Laboratórios remotos são uma boa opção para esta nova educação por promoverem uma abordagem nova de experimentos clássicos das disciplinas da engenharia, incluindo assim conceitos como conectividade e trabalho remoto nos currículos dos estudantes.

Este projeto, juntamente com outros experimentos desenvolvidos no projeto Laboratório com base na Indústria 4.0 para o ensino em cursos da engenharia da FGA (Universidade de Brasília Campus Gama), tem como intuito a criação de uma plataforma de laboratórios que poderão ser utilizados nas disciplinas de engenharia eletrônica da FGA.

1.1 Objetivos gerais

Este trabalho visa desenvolver um laboratório de controle do pêndulo invertido rotacional que possa ser utilizado em disciplinas como sistemas embarcados, sistemas de controle e outras disciplinas correlatas.

1.2 Objetivos específicos

- Projetar e construir a estrutura de um pêndulo invertido rotacional.
- Projetar e desenvolver o circuito de conexão dos componentes eletrônicos, juntamente com os *firmwares* necessários para o controle do pêndulo.
- Desenvolver os códigos necessários para a comunicação, comando e exibição gráfica dos dados.
- Realizar testes de funcionamento eletromecânico utilizando um PID (Controlador proporcional integral derivativo).
- Desenvolver e testar algoritmos para subir a barra do pêndulo.

1.3 Organização do texto

Este documento apresenta as etapas de projeto, desenvolvimento e testes do pêndulo invertido sendo dividido em introdução, referencial teórico, metodologia, resultados parciais e conclusões parciais. Na introdução temos a contextualização da Indústria 4.0 e sua importância, juntamente com os objetivos deste trabalho. O referencial teórico traz os resultados das pesquisas sobre Indústria 4.0, o funcionamento do pêndulo invertido, os controladores e sistemas embarcados envolvidos, além de exemplos de implementações remotas de pêndulos invertidos. Na metodologia são apresentados os requisitos funcionais e não funcionais do projeto, a solução proposta e detalhes sobre o desenvolvimento do projeto. As seções de resultados e conclusão finalizam este trabalho apresentando os resultados finais da implementação e as mudanças e acréscimos de funcionalidades que devem ser realizadas para melhorar o laboratório.

2 Referencial teórico

Neste capítulo são trazidos os resultados das pesquisas sobre a indústria 4.0 e seus impactos, com foco nos laboratórios remotos. Este capítulo apresenta, de uma forma mais detalhada, informações sobre o funcionamento mecânico do pêndulo invertido e os controladores utilizados para estabilizá-lo, além de trazer um caso de implementação de laboratório remoto de pêndulo invertido linear.

2.1 Indústria 4.0 e Educação 4.0

O termo Indústria 4.0 se refere a um novo paradigma de produção que utilizam novas tecnologias para tornar os sistemas produtivos mais flexíveis, adaptáveis e descentralizados. Este novo modelo de produção corresponde a quarta revolução industrial, e terá impactos econômicos, tecnológicos e sociais (SANTOS; LIMA, T. M., 2018). Em (ALBERTIN et al., 2017) são destacados nove pilares tecnológicos da indústria 4.0, sendo estes internet das coisas, realidade aumentada, robôs autônomos, simulação, manufatura aditiva, *big data*, a tecnologia de nuvem, *cybersecurity* e integração horizontal e vertical de sistemas e *softwares*, sendo explicitado que essas tecnologias estão em diferentes pontos de desenvolvimento e aplicação industrial.

A indústria 4.0 vai um requerer grande esforço na formação dos trabalhadores, já que estas tecnologias sofisticadas requerem cada vez mais profissionais com habilidades específicas, portanto será necessário um investimento na capacitação dos trabalhadores que irão lidar com essas tecnologias (SANTOS; LIMA, T. M., 2018). A educação 4.0 surge como um modelo que busca inserir tecnologias digitais na educação, proporcionando habilidades para o conhecimento no século 21 e a criação de competências digitais demandadas pela indústria 4.0 (OLIVEIRA; SOUZA, 2020). Um exemplo de tecnologia da indústria 4.0 que começa a ser empregada na educação é a *big data*, que utiliza os dados coletados dos alunos para promover um ensino personalizado conforme o perfil do aluno (ALBINO, 2019).

Em (ALVES et al., 2016) é analisado a experiência de um curso de especialização para professores de engenharia e cursos das áreas de exatas na UERGS (Universidade Federal do Estado do Rio Grande do Sul). A ementa deste curso abrange as possibilidades de experimentos online voltados para a engenharia, tentando alinhar as três possibilidades de acesso a experimentos laboratoriais: simulações, realidade virtual e laboratórios remotos. Como conclusão o autor cita um desconhecimento parcial do potencial dos experimentos remotos, além de uma confusão entre laboratórios virtuais e remotos por parte dos alunos. As conclusões do trabalho anteriormente citado evidenciam que embora o conceito de laboratório remoto esteja sendo difundindo ele ainda não tem sua potencialidade totalmente

compreendida.

2.1.1 Laboratórios remotos

Laboratórios remotos utilizam a internet para promover o acesso remoto a equipamentos, bancadas e experimentos, assim combinando os requisitos dos laboratórios presenciais com a flexibilidade das simulações. O uso destes laboratórios é ainda mais proveitoso em cursos das áreas tecnológicas, pois estimula o desenvolvimento de habilidades como operação remota, diagnóstico e manutenção (SILVA, J. B. d., 2006). Esta flexibilização do uso de laboratórios é importante quando há um distanciamento geográfico do aluno com as instalações do laboratório, como no caso de disciplinas EAD (Ensino a distância). Outra vantagem importante deste modelo de laboratório é o aumento da disponibilidade do uso destes equipamentos, que podem estar funcionando 24 horas por dia durante todo o ano.

Em (ANTONIO; CARVALHO; TULHA, 2019) foi desenvolvida uma revisão sistemática sobre laboratórios remotos no Brasil. Este trabalho concluiu que a maioria dos laboratórios remotos está voltado para disciplinas STEM (*Science, Technology, Engineering e Mathematics*), e é destacado uma rede de pesquisadores (RexNet) voltada a pesquisa e disseminação de laboratórios remotos. Esta rede foi um projeto que reuniu 12 Instituições de ensino da América Latina e Europa entre 2005 e 2006 a fim de promover a partilha, harmonização e expansão das competências em experimentação remota (ALVES; FLORENZA, 2008).

No Brasil algumas instituições de ensino possuem, mantêm ou utilizam laboratórios remotos, como exemplo podemos citar o RExLAB (UFSC, s.d.) e o VISIR (PEREIRA et al., 2018). O RExLab utiliza as novas tecnologias da informação para promover atividades experimentais remotas em vários níveis do ensino. Estas atividades estão reunidas no site da UFSC (Universidade Federal de Santa Catarina), que contém práticas de painel elétrico CC (Corrente contínua) e CA (Corrente alternada), desenvolvimento em Arduino, propagação de calor e conversão de energia, sendo também possível a disponibilização de experimentos remotos criados por terceiros (UFSC, s.d.). O VISIR possibilita a montagem e medição de circuitos através de uma bancada disponibilizada pela internet, e está presente em 12 universidades ao redor do mundo, incluindo a UFSC, IFSC (Instituto Federal campus Santa Catarina) e a PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro). O VISIR instalado na UFSC foi utilizado em disciplinas de instrumentação, análise de circuitos e física por 3 instituições e por um total de 154 estudantes (PEREIRA et al., 2018).

É desejável que os laboratórios remotos disponham de um uso simples e intuitivo para possibilitar seu uso para um número maior de usuários. Para alcançar esse objetivo muitos laboratórios remotos utilizam interfaces de usuário desenvolvidas para facilitar a realização e o acompanhamento do experimento, juntamente com ferramentas de gestão de controle de acesso. Um exemplo de interface para laboratórios remotos é o RELLE (*Remote Lab Learning*

Environment), que é uma ambiente de aprendizado e repositório de laboratórios remotos desenvolvido pela RExLab (LIMA, J. P. C. de, 2016).

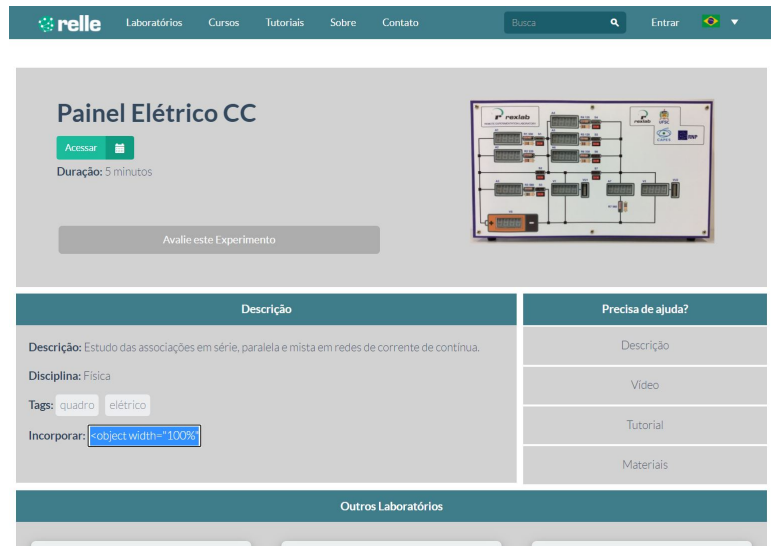


Figura 2.1 – Agendamento de laboratório no RELLE
Fonte: (UFSC, s.d.)

Na figura 2.1 é mostrada a pagina de agendamento de um experimento com painel elétrico, e vale notar a simplicidade da interface. O uso de interfaces deste tipo poupa o usuário da necessidade de saber detalhes técnicos sobre a conexão com o experimento, o que inviabilizaria o uso para muitos estudantes.

Nas referências pesquisadas os laboratórios remotos se mostraram como uma ferramenta que pode acrescentar ao aprendizado dos alunos, principalmente pela facilidade de acesso aos laboratórios remotos em comparação com os laboratórios presenciais. Os laboratórios remotos geralmente estão voltados para as disciplinas STEM, e seu uso pode beneficiar disciplinas da engenharia como: sistemas de controle, circuitos, física, química, microcontroladores etc.

2.2 Pêndulo invertido e seu controle

O pêndulo é um sistema não linear, onde a aceleração angular varia de acordo com o seno da posição. Sistemas não lineares são melhores modelos para alguns fenômenos naturais, mas apresentam maiores dificuldades na modelagem e controle (KHALIL, 2002). O pêndulo invertido proposto possui movimento rotacional em torno de uma base circular, enquanto a barra do pendulo possui um movimento circular em torno do eixo do *encoder* que é fixado na ponta do braço de rotação. Um modelo esquemático pode ser visto na figura 2.2, onde r é a distancia do centro de massa da barra até o braço do pêndulo, L_p é a distancia do centro de rotação até a ponta do braço, θ é o angulo de rotação do braço, α é o angulo de rotação do pêndulo e B é o centro de massa da barra.

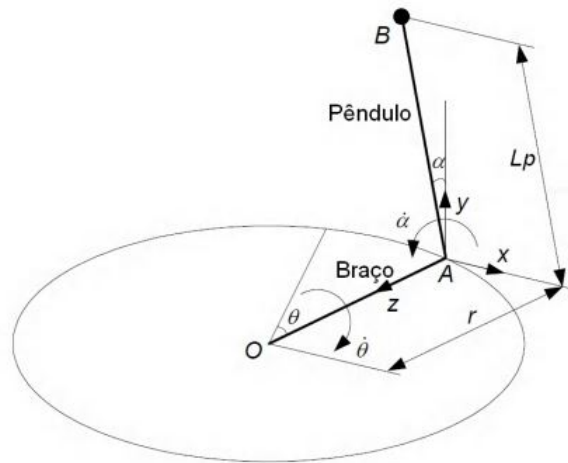


Figura 2.2 – Modelo físico de um pêndulo invertido rotacional

Fonte: (DIAO, 2006)

Em (V; Z; SRINIVASAN, 2013) os autores fazem uma análise física do problema utilizando a mecânica lagrangiana para obter as equações de espaço de estados. Esta modelagem parte primeiramente da análise da velocidade do centro de massa e do braço do pêndulo, que são calculadas pelas equações abaixo, onde \hat{x} e \hat{y} são as velocidades nas direções x e y perpendiculares ao plano de rotação do braço. Os parâmetros L_p e r representam a distância do centro de massa até o braço e o comprimento do braço, e os parâmetros α , $\dot{\alpha}$, θ e $\dot{\theta}$ representam o ângulo do pêndulo, a velocidade angular do pêndulo, o ângulo do braço e a velocidade angular do braço, respectivamente.

$$V_{CentroDeMassa} = -L_p \cos(\alpha)(\dot{\alpha})\hat{x} - L_p \sin(\alpha)(\dot{\alpha})\hat{y} \quad (2.1)$$

$$V_{Braço} = r\dot{\theta} \quad (2.2)$$

Pelas equações é possível observar que os parâmetros de construção do pêndulo que afetam a mecânica do movimento são o comprimento do braço e a distância do centro de massa até o braço, de forma que quanto maior essas distâncias maior será o momento de inércia dessas peças, portanto será mais difícil rotacioná-las.

Um parâmetro importante para o controle e movimento do projeto é o centro de massa do pêndulo. O centro de massa pode ser calculado utilizando a equação em (2.3). É importante notar que quando o corpo possui uma distribuição contínua de massa o somatório deve ser substituído por uma integral (YOUNG H. D.; FREEDMAN, 2008).

$$\vec{r}_{cm} = \frac{\sum_i m_i \vec{r}_i}{\sum_i m_i} \quad (2.3)$$

2.2.1 Controle do pêndulo invertido

Os sistemas de controle podem ser de malha aberta ou malha fechada. Em sistemas de malha aberta não existe relação entre a saída do sistema e o controle. Nestes tipo de controle o sistema não responde a mudanças externas, portanto seu funcionamento depende da calibração, e na prática só são utilizados se a relação entre entrada e saída é conhecida e não há distúrbios. Em sistemas de malha fechada a realimentação do valor de saída produz uma maior robustez a distúrbios externos e variações nos componentes do sistema. Contudo, como efeito da realimentação, sistemas de malha fechada podem apresentar instabilidade, sendo este um dos principais problemas a ser contornado no projeto e desenvolvimento deste tipo de controlador (OGATA, 2002).

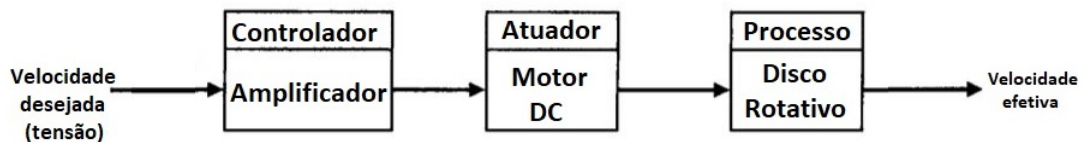


Figura 2.3 – Controle de velocidade utilizando malha aberta

Fonte: (DORF; BISHOP, 2011)

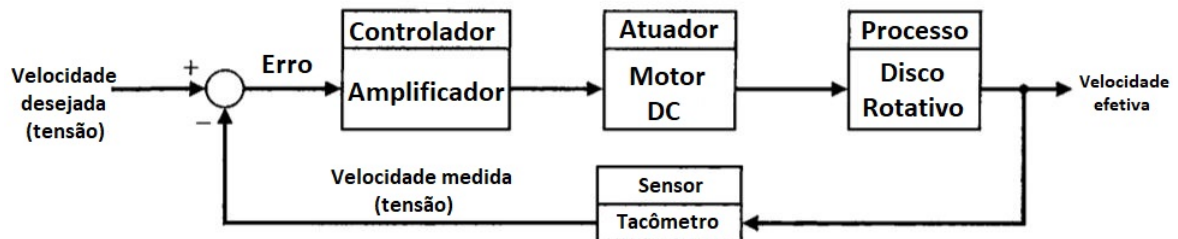


Figura 2.4 – Controle de velocidade utilizando malha fechada

Fonte: (DORF; BISHOP, 2011)

Nas figuras 2.3 e 2.4 temos exemplos de malha de controle. A diferença entre as duas é a presença do sensor, que faz a realimentação da velocidade real para gerar o erro utilizado no controle, caracterizando assim uma malha fechada.

O sistema do pêndulo invertido não é linear e é muito sensível a distúrbios, portanto sistemas de controle com malha fechada são recomendados para conseguir manter a variável do processo no valor desejado. A seguir são trazidos alguns exemplos de controladores de pêndulo invertido encontrados na literatura.

Em (BOUBAKER, 2013) é feito uma pesquisa sobre os tipos de controle utilizados no pêndulo invertido, que incluem técnicas como controle *bang-bang*, lógica *fuzzy*, redes neurais, controladores PID adaptativos, controle baseado em energia, controle híbrido, entre outros. Neste trabalho vale destacar o controlador *fuzzy* e sua estrutura simples, que geralmente não possui condição de estabilidade garantida, e o controlador PID adaptativo, que possui um resultado satisfatório no controle, mas que precisa ser constantemente ajustado. Segundo o autor técnicas de modelagem de energia, controle híbrido e controle de estrutura variável

são as metodologias de controle do pêndulo com abordagem matemática mais rigorosa, mas novas abordagens de controles híbridos tem atraído atenção por reduzir a complexidade dos controladores.

Em (YI; YUBAZAKI, 2000) foi desenvolvido um controlador baseado na lógica *fuzzy* que foi utilizado em uma simulação de controle de um pêndulo invertido linear. Ao contrário de um controlador *fuzzy* comum o controlador projetado utiliza módulos de regra de única entrada (SIRM, *Single input rule module*), que são módulos que utilizam apenas uma das quatro variáveis de estado. Além das SIRMs também foi utilizado um bloco de grau de importância dinâmica (DID, *Dynamic importance degree*) para calcular a importância relativa de cada variável de controle. A malha de controle (figura 2.5) é formada por pares de SIRM e DID, e utiliza os sinais x_1, x_2, x_3, x_4 como versões normalizadas das variáveis $\theta, \dot{\theta}, x$ e v (posição angular, velocidade angular, posição do carrinho, velocidade do carrinho). Essas versões normalizadas são utilizadas nos cálculos das SIRMs, sendo que x_1 é utilizado no cálculo das importâncias relativas de todos os blocos, e ao final o resultado é multiplicado por um fator de escala de saída (OSF, *Output Scale Factor*) antes de ser passado para a planta. Os resultados das simulações de mostraram que o controlador proposto tem uma capacidade de estabilizar uma grande variedade de pêndulos com ângulo inicial acima de 30° dentro de 9 segundos.

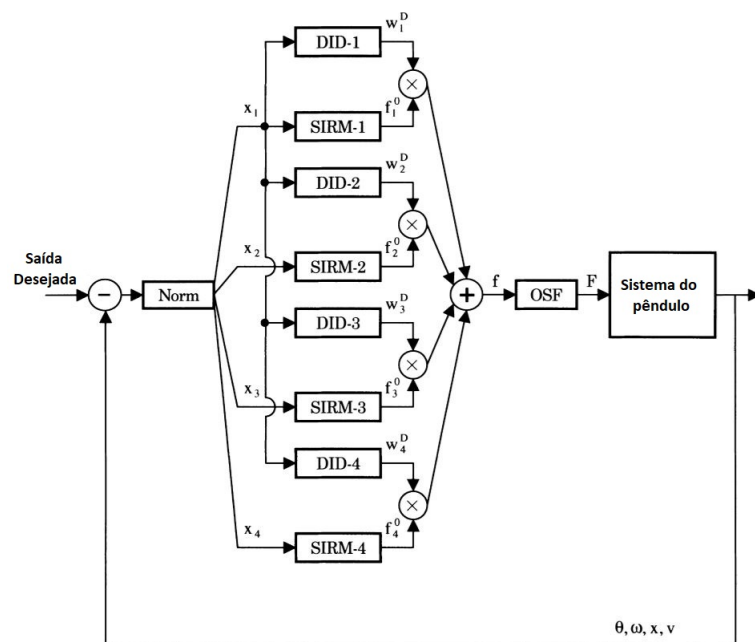


Figura 2.5 – Malha de controle do controlador *fuzzy*

Fonte: (YI; YUBAZAKI, 2000)

Em (JUNG; KIM, 2008) foi desenvolvido um controlador para um pêndulo invertido sobre 2 rodas. Este controlador utiliza a técnica de compensação de referência implementada em uma rede neural e um controlador PID para controlar a inclinação de uma barra e a posição do carrinho. Na figura 2.6 pode ser visto a malha de controle utilizada, em que a rede

neural recebe os valores do *set point* e do erro realimentado para fornecer ao controlador PID um valor compensado para minimizar o erro.

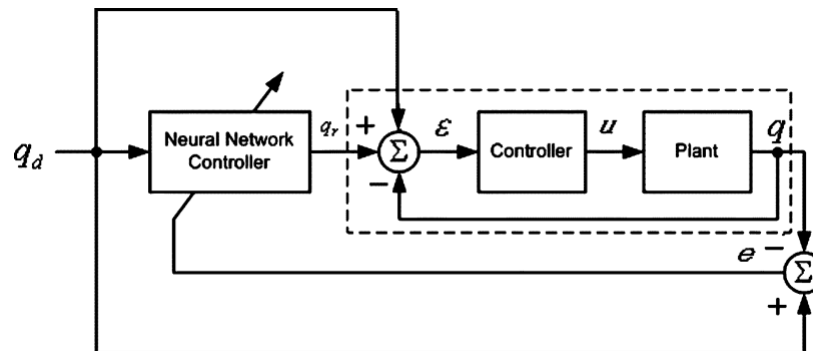


Figura 2.6 – Malha de controle com rede neural e controlador PID

Fonte: (JUNG; KIM, 2008)

Os teste apenas com o controlador PID conseguiram estabiliza a barra, mas falharam em controlar a posição do carrinho, que se movia para frente mesmo com *set point* estando na posição zero. Já a rede neural juntamente com o controlador PID conseguiu estabilizar a inclinação e a posição do sistema mesmo sem saber a dinâmica do sistema.

2.2.1.1 Controlador PID

Atualmente mais da metade dos controladores utilizados são do tipo PID. A grande utilidade desses controladores se da pelo fato da sua grande aplicabilidade, além de ser possível implementar várias técnicas de ajuste de seus parâmetros (OGATA, 2002). O controlador PID utiliza a soma dos componentes proporcional, derivativa e integrativa, que são realimentados em uma malha fechada, para melhorar a resposta transitória e o erro permanente na planta controlada.

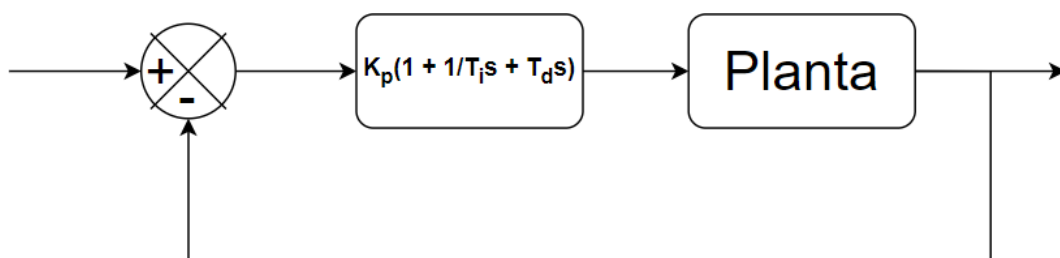


Figura 2.7 – Diagrama geral de um controlador PID no domínio de Laplace

Na figura 2.7 está sendo analisado o funcionamento de uma planta genérica com o controlador PID estando no domínio de Laplace. No domínio do tempo o controlador possui três parâmetros que controlam seu funcionamento, sendo eles:

- Parâmetro P: Corresponde ao efeito proporcional do controlador; é responsável por gerar uma resposta baseada no *set point* do sistema.

- Parâmetro I: Corresponde ao efeito integrativo do controlador; é utilizado para reduzir o erro de estado estacionário do sistema
- Parâmetro D: Corresponde ao efeito derivativo do controlador; é utilizado como uma forma de predição do sinal de saída.

É importante notar que o PID recebe o sinal do *set point* subtraído da realimentação, caracterizando este método como controle de malha fechada. A grande utilidade dos controladores PID está na sua aplicabilidade a vários sistemas de controle, especialmente em sistemas de controle em que o modelo matemático não é conhecido (OGATA, 2002). No pêndulo invertido o PID pode ser utilizado para controlar a posição do braço de rotação para gerar uma aceleração que irá compensar os distúrbios gerados e o movimento de queda do pêndulo. Em (ROSA; MILHOMEM, s.d.) foi utilizado um pêndulo invertido linear real como planta de teste para realizar um estudo comparativo entre um controlador PID e um controlador por realimentação de estados. Os resultados deste estudo apontaram o controlador PID como uma melhor opção, devido a uma resposta mais rápida.

Em (AKHTARUZZAMAN; SHAFIE, 2010) é feita uma análise comparativa entre o método de controle clássico PID e os métodos de controle moderno LQR (*Linear Quadratic Regulator*) e FSF (*Full State Feedback*) no controle de um pêndulo invertido rotacional. Neste trabalho foram utilizados 2 PIDs, cujos parâmetros foram ajustados com o método do lugar das raízes. Pelos resultados dos testes os autores concluem que o controle do pêndulo com o PID é possível mas não é robusto na prática, ao contrário dos outros 2 controladores modernos que são mais robustos a variações de parâmetros e portanto mais adequados para o controle do pêndulo.

2.2.2 Laboratórios remotos do pêndulo invertido

Como parte da pesquisa para a realização deste trabalho foram pesquisadas implementações de laboratórios remotos de pêndulo invertido. Em (sanchez2004) é descrito um ambiente para a operação remota de um pêndulo invertido de movimentação linear, cujo o objetivo é implementar um experimento em que o usuário obtenha os parâmetros de um controlador utilizando o projeto de Regulador Quadrático Linear (LQR). Neste trabalho o controlador foi implementado utilizando o Simulink rodando no servidor local, que recebe os parâmetros desejados através de uma interface gráfica.

Na figura 2.8 é mostrado a interface gráfica utilizada, que traz as opções de ajuste, os gráficos do experimento e a animação do pêndulo todas em uma única janela.

Em (ISSA et al., 2018) é descrito a implementação de um pêndulo invertido linear com acesso remoto. Foi utilizado um computador para atuar como servidor da aplicação, além de enviar os parâmetros dos controladores PID para um Arduino e transmitir as imagens do

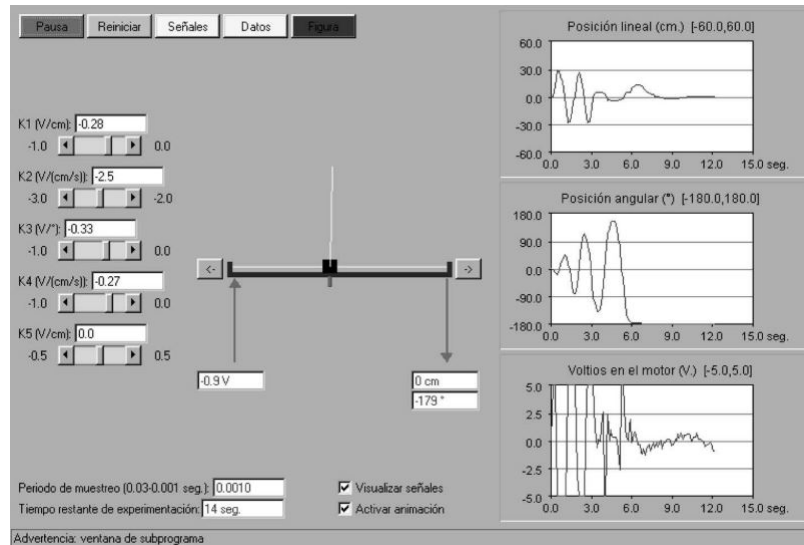


Figura 2.8 – Interface gráfica
Fonte: (SANCHEZ et al., 2004)

experimento para o usuário. Neste experimento foram utilizados dois controladores PID, um para o controle da posição linear e outro para o controle da posição angular do pêndulo.

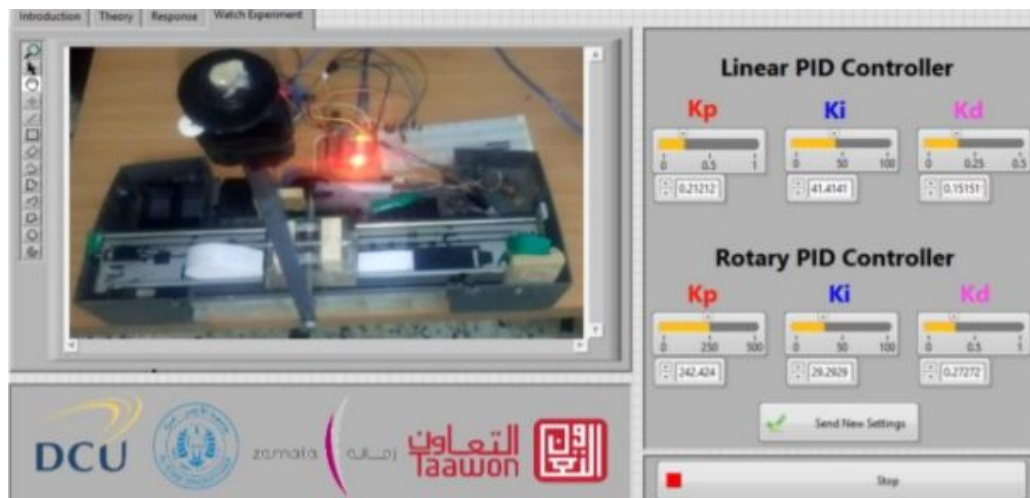


Figura 2.9 – Interface gráfica de acesso ao pêndulo invertido
Fonte: (ISSA et al., 2018)

Na figura 2.9 temos a imagem da interface gráfica utilizada, que traz em uma mesma tela tanto a escolha dos parâmetros dos controladores quanto o vídeo do experimento.

Em (ZUMSANDE et al., 2017) foi construído um pêndulo invertido linear remoto na Universidade de Hanôver para auxiliar no ensino de conceitos de mecatrônica e engenharia de controle. Este laboratório possui um sistema de agendamento, controle e visualização de resultados em uma mesma página web para implementar o laboratório remoto, resultando em uma interface intuitiva que alcançou uma boa avaliação em uma pesquisa de usabilidade relatada neste trabalho. Na figura 2.10 pode ser visto a interface gráfica de acesso e controle utilizada para o controle do pêndulo invertido. Um diferencial no uso deste laboratório foi a

utilização dos resultados do controle dos usuários em um *ranking* para motivar os alunos a melhorarem os resultados obtidos.

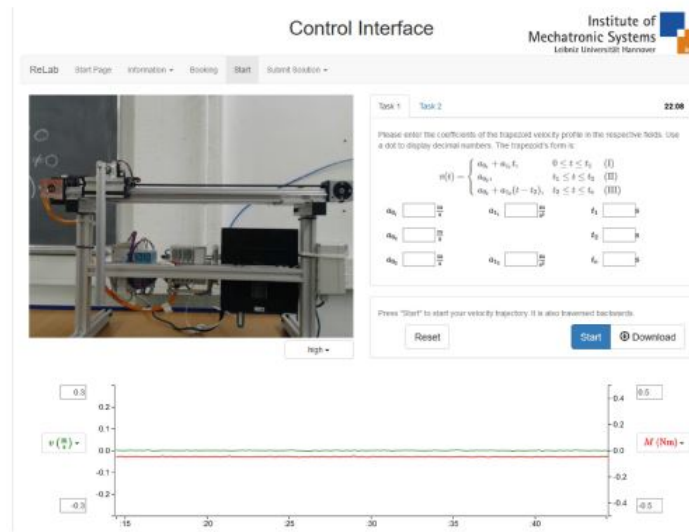


Figura 2.10 – Interface de controle do laboratório remoto da Universidade de Hanôver
Fonte: (ZUMSANDE et al., 2017)

Em (JAZIRI; CHARAABI; JELASSI, 2018) foi implementado um laboratório remoto utilizando um placa Beaglebone embarcada com Linux para reconfigurar uma FPGA (*Field Programmable Gate Array*) e assim controlar 3 possíveis experimentos: um motor DC (*Direct Current*), um robô e um pêndulo invertido. A interface de escolha do experimento pode ser vista na figura 2.11.

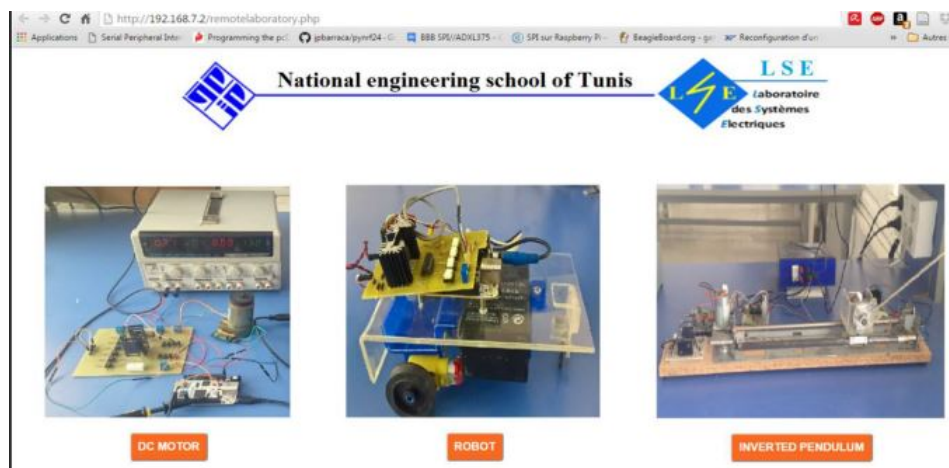


Figura 2.11 – Interface de acesso aos 3 experimentos
Fonte: (JAZIRI; CHARAABI; JELASSI, 2018)

Em um mesmo experimento é possível utilizar a interface gráfica criada para ajustar os parâmetros de um controlador PID já implementado ou desenvolver um novo controlador utilizando VHDL (*VHSIC Hardware Description Language*) e configura-lo na FPGA utilizando a conexão JTAG (*Joint Test Action Group*) com o Beaglebone, como mostram as figuras 2.12 e 2.13.

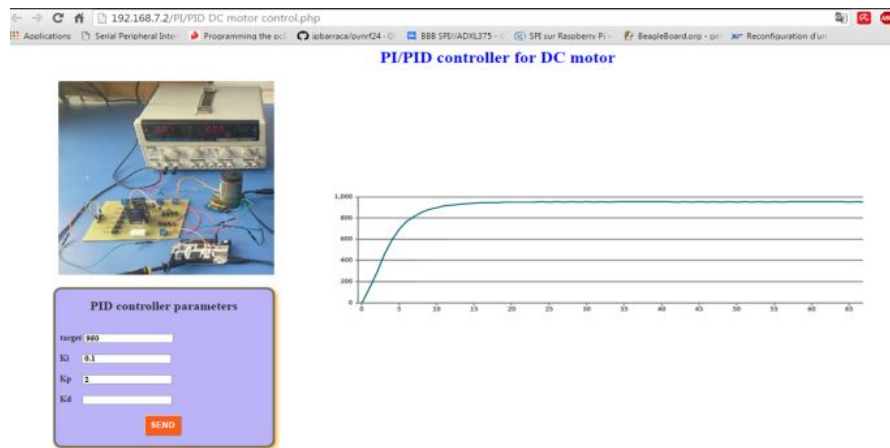


Figura 2.12 – Controle por ajuste do controlador PID

Fonte: (JAZIRI; CHARAABI; JELASSI, 2018)

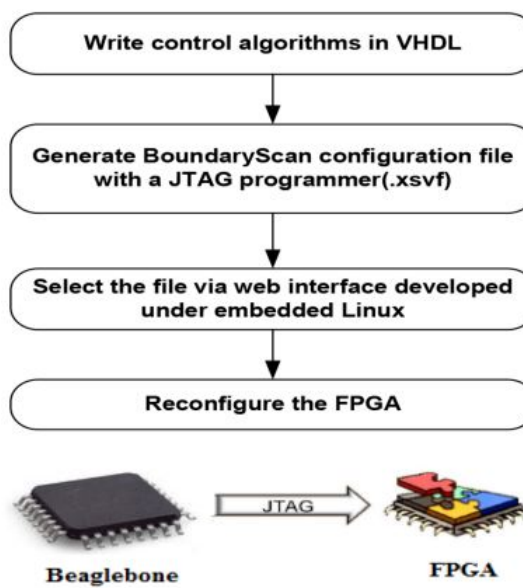


Figura 2.13 – Configuração de controlador em FPGA

Fonte: (JAZIRI; CHARAABI; JELASSI, 2018)

3 Metodologia

Neste capítulo são apresentados os requisitos funcionais e não funcionais do projeto, a solução proposta e a metodologia utilizada no seu desenvolvimento.

3.1 Requisitos funcionais

- Ter estrutura de um pêndulo invertido rotacional.
- Permitir o controle do ângulo da barra e visualização dos dados do experimento.
- Monitorar e transmitir dados de posição angular, erro de controle, velocidade angular e ciclo útil.
- Gravar dados do experimento.
- Exibir os dados obtidos na forma de gráficos.
- Permitir programação e *debug*.

3.2 Requisitos não funcionais

- Estrutura de suporte deve suportar os pesos dos componentes e os esforços envolvidos na movimentação do carrinho e do pêndulo.
- Deve ser possível mudar o microcontrolador utilizado na prática.

3.3 Solução proposta

A primeira decisão tomada no início do desenvolvimento do projeto foi o tipo de pêndulo a ser desenvolvido. Para permitir uma melhor visualização do experimento foi optado pelo pêndulo invertido de movimento circular, já que para observar o movimento de um pêndulo linear através de uma câmera seria necessário posicionar este dispositivo muito longe do pêndulo linear ou construir alguma estrutura que se movimentasse juntamente com o carro do pêndulo, o que dificultaria muito o projeto. Ao contrário do pêndulo linear no pêndulo de movimento circular a câmera pode ser posicionada sobre uma estrutura simples que rotaciona com o movimento da base em que é fixada e sempre enquadra a barra do pêndulo de forma centralizada.

Para atender os requisitos e os objetivos já descritos foi pensado uma serie de sistemas que em conjunto formarão o pêndulo. O experimento deve funcionar de acordo com o diagrama da figura 3.14 que apresenta uma ordem cronológica do acesso e uso do laboratório. Depois da conexão com a rede local o usuário deve escolher se quer ajustar os parâmetros do PID ou se prefere escrever um novo código no micro controlador (este código deve ser previamente compilado para gerar o arquivo que será enviado para o micro controlador). Após a programação será iniciado o experimento, que terá fim após um tempo pré determinado ou após o usuário encerrar o experimento, e em ambos os casos o usuário poderá baixar um arquivo com os dados do experimento.

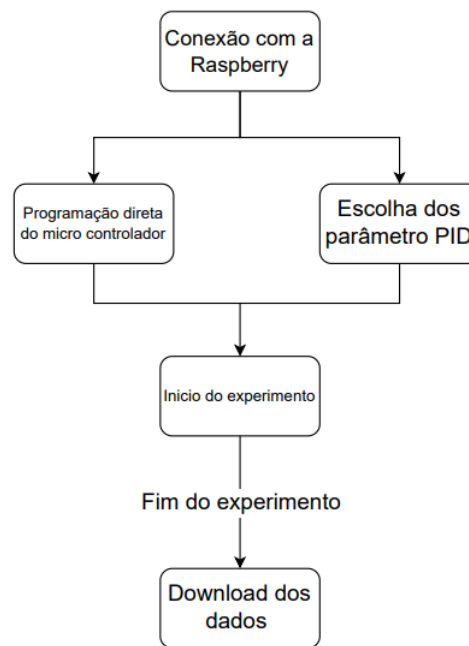


Figura 3.14 – Diagrama do uso do laboratório

A comunicação do usuário com o experimento será feita através do *software* Hamachi, que será responsável por conectar o computador do usuário a rede local na qual a Raspberry também estará conectada. A Raspberry por sua vez será responsável por escrever no microcontrolador, além de ler e enviar os dados dos sensores e da câmera para usuário. A comunicação entre os componentes do projeto pode ser visualizada no diagrama 3.15. Na versão atual deste trabalho a conexão remota e algumas outras funcionalidades necessárias para implementação completa do pêndulo como laboratório remoto são deixadas para versões futuras.

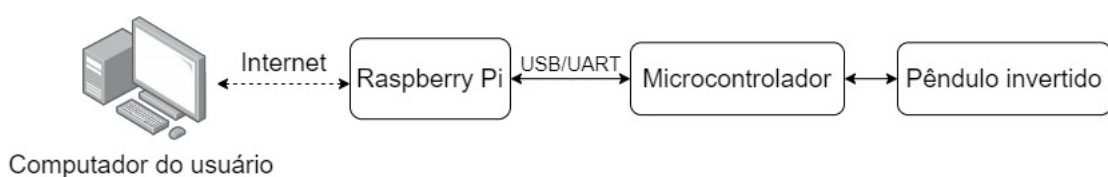


Figura 3.15 – Comunicação entre usuário e o experimento

3.3.1 Solução mecânica

Para dar suporte aos componentes eletrônicos e para implementar um pêndulo rotativo funcional foram desenhadas as peças do pêndulo utilizando CAD (*Computer Aided Design*). Na figura 3.16 é possível visualizar a vista superior do CAD do projeto do pêndulo. Como o projeto é de um pêndulo rotacional foi desenhado uma base de 38 cm de diâmetro que serve de suporte aos componentes e peças. Por cima de uma parte desta base foi colocado uma borracha com cerca de 4 mm de espessura para servir de "pista" e aumentar o atrito entre a base e a roda. A base também possui um furo central para a passagem dos cabos de alimentação, e é apoiada em cima de uma estrutura em forma de cruz que serve para sustenta-la.

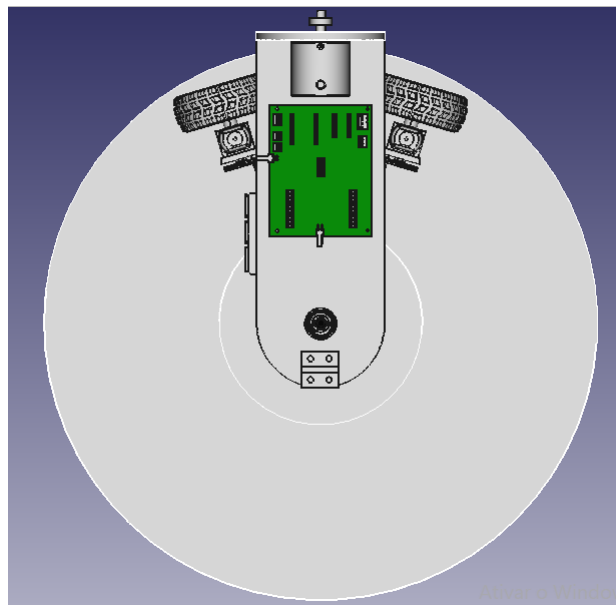


Figura 3.16 – Vista superior do pêndulo invertido

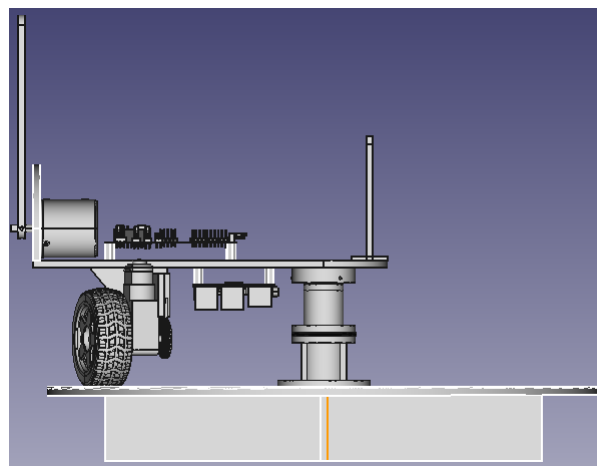


Figura 3.17 – Vista lateral do CAD do pêndulo invertido

Parafusada acima desta base temos o tubo de sustentação da base móvel. Esta estrutura foi projetada de forma a sustentar e permitir a rotação da base móvel, além de

possibilitar a passagem dos cabos de alimentação. Este tubo é composto por 3 peças: a parte inferior do tubo, o anel deslizante (*slip ring*) e a parte superior do tubo. Acima do tubo temos um rolamento flangeado que é fixado ao tubo através de uma bucha roscada (os cabos de alimentação passam por dentro da bucha roscada) e de uma porca. Em volta do rolamento flangeado foi fixado uma peça em formato de rosca que serve como interface mecânica para unir o rolamento flangeado com a base rotativa (o rolamento foi fixado a peça utilizando prensagem, a base é parafusada a esta peça utilizando 3 parafusos). A peça de junção pode ser visto na figura 3.18.

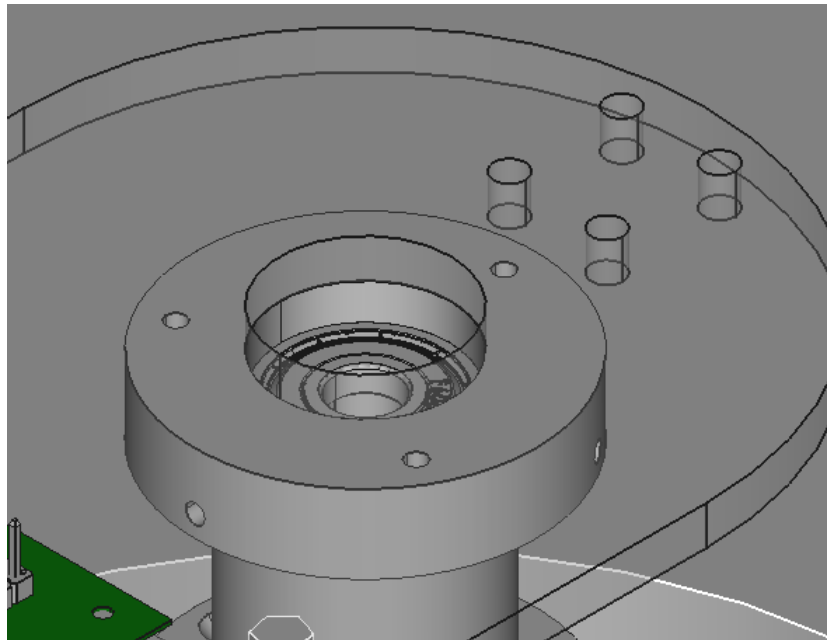


Figura 3.18 – Peça de junção abaixo da base móvel

A base móvel tem formato retangular com um semicírculo em uma das pontas, onde foi feito um furo circular para a passagem dos cabos. Esta base foi desenhada para sustentar toda a eletrônica do projeto, o que inclui o *encoder* rotacional, a Raspberry, a câmera, as placas eletrônicas e os demais componentes.

Em uma primeira versão do pêndulo invertido (versão que foi defendida do TCC 1) o pêndulo contava apenas com um motor, e portanto havia apenas um *encoder* óptico com roda furada e as estruturas de sustentação que são necessárias ao motor e ao *encoder*. Após a montagem e uma série de testes foi identificado um problema de movimentação indesejada no pêndulo durante a realização dos experimentos. Esta movimentação consistia em uma rotação indesejada em torno do eixo do braço do pêndulo (figura 3.19), o que causava erros de leitura do ângulo da barra do pêndulo e instabilidade no controle.

Para resolver este problema foi necessário utilizar 2 motores para gerar um conjunto de apoio com 2 pontos, e assim impedir a rotação indesejada. Com isso foi construído uma única estrutura de fixação que fixa os motores e os dois *encoders* utilizados nesta versão do pêndulo.

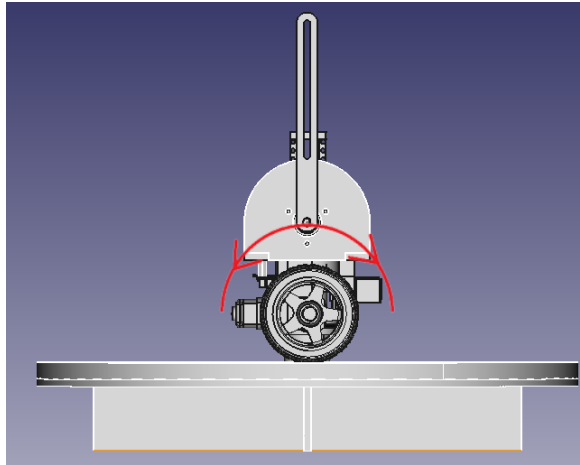


Figura 3.19 – Rotação indesejada da base móvel

Para servir como pêndulo foi desenhado uma barra com um corte no meio na qual pode ser colocado pesos em diferentes alturas para obter sistemas com diferentes respostas dinâmicas. A barra está fixada a base móvel através do eixo do *encoder* rotacional, que serve como eixo de rotação do pêndulo. O *encoder* por sua vez é fixado a uma estrutura semicircular que é encaixada na base móvel. Na figura 3.20 pode ser vista em maior detalhe a barra juntamente com a peça em formato semicircular que prende o *encoder* rotacional a base móvel.

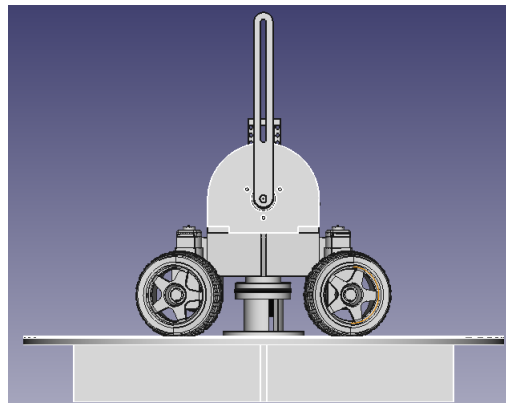


Figura 3.20 – Vista frontal do pêndulo invertido

Na figura 3.21 temos a vista diagonal do projeto CAD do pêndulo, onde pode ser visto o suporte da câmera da Raspberry. Este suporte foi desenhado para poder fixar a câmera da Raspberry em diferentes alturas.

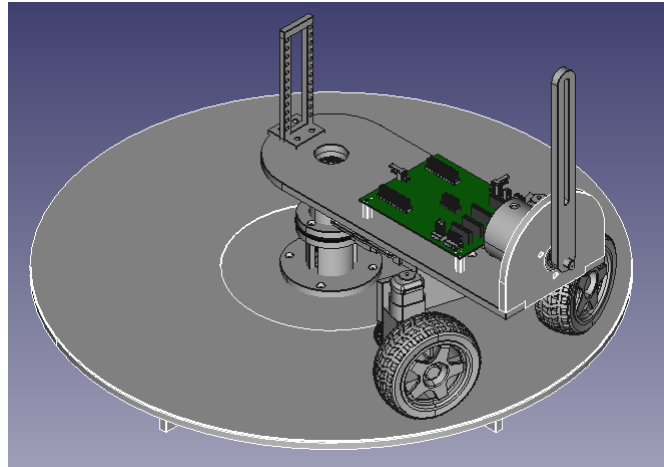


Figura 3.21 – Vista diagonal do CAD do pêndulo

3.3.2 Solução eletrônica

Para a implementação das funcionalidades do projeto foram necessários um conjunto de componentes eletrônicos listados na tabela 3.1 e descritos nos parágrafos seguintes.

Componente	Modelo
2 Módulos encoder óptico	MH-sensor-series
Endoder óptico incremental	LPD3806
Slip ring	8 canais 2 A
2 Motores DC	3-6v com redução
Deslocador de nível lógico	4 canais
Módulo ponte H	TB6612FNG
Placa de desenvolvimento com microcontrolador MSP430F5529	F5529
Raspberryberry pi	3B
Câmera	Rev 1.3 (70° FOV)
Fonte de alimentação	5V 5A

Tabela 3.1 – Componentes eletrônicos utilizados

Para realizar a medição da velocidade do carrinho foi escolhido um módulo *encoder* com sensor ótico para realizar a contagem dos passos da roda furada que acompanha este módulo.

Como pode ser visto na figura 3.22 a interrupção do feixe de luz gera um sinal de saída digital (SILVA, C. W. d., 2007). O sinal quadrado de saída do sensor pode ser lido para calcular a velocidade de rotação da roda furada. Nesta versão do experimento são utilizadas 2 sensores, cada um em um pino do microcontrolador com interrupção.

Para medir o ângulo da barra um *encoder* óptico incremental foi incorporado ao projeto. Esse tipo de sensor gera duas ondas quadradas defasadas em 90°, assim é possível detectar a direção da rotação comparando dois valores consecutivos da saída do *encoder*.

O *encoder* escolhido para o projeto foi o LPD3806 (figura 3.24), que possui tensão de

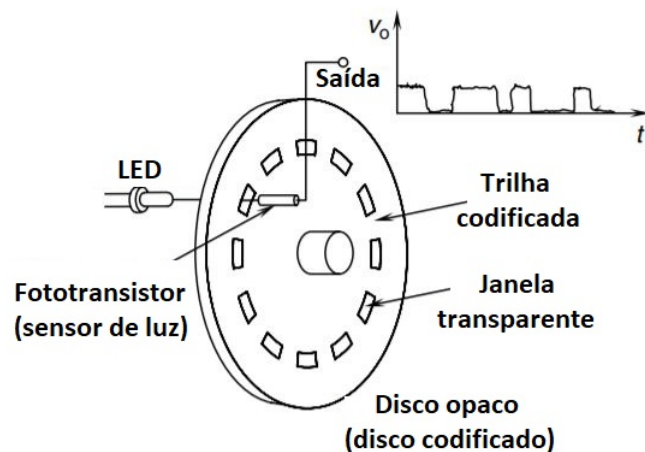


Figura 3.22 – Funcionamento do *encoder* óptico

Fonte: (SILVA, C. W. d., 2007)

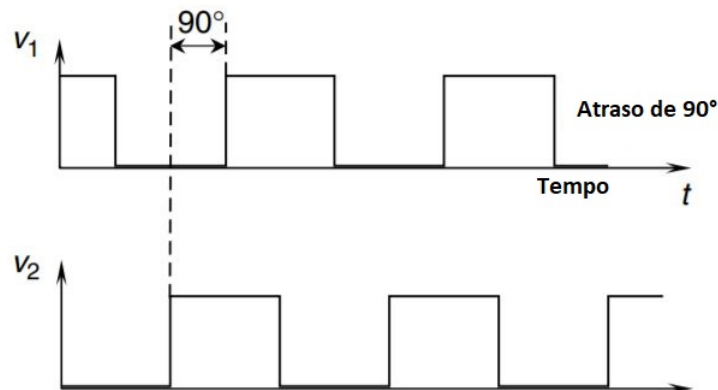


Figura 3.23 – Onda quadrada de saída do *encoder*

Fonte: (SILVA, C. W. d., 2007)

alimentação de 5V, frequência de operação de 20 kHz e fornece 2400 pulsos por rotação e portanto tem $0,15^\circ$ como menor escala na leitura do ângulo. Por sua tensão de operação ser maior do que 3.3 V será necessário utilizar um dispositivo para modificar as tensões de saída para poder haver uma conexão entre o sensor e o microcontrolador.

Um deslocador de nível lógico é um dispositivo que tem como função transformar os níveis de tensão lógica entre dispositivos, portanto utilizando um deslocador é possível estabelecer uma conexão entre os pinos de dispositivos que possuem tensões diferentes. O uso do deslocador de nível lógico é justificado para converter os níveis de tensão do *encoder* LPD3806 que possui tensão lógica de 5 V para tensões mais baixas para que possa haver comunicação com micro controladores que utilizem a tensão de 3,3 V. O deslocador escolhido para o projeto possui 4 canais (figura 3.25), mas só serão utilizados 2 canais.

Para o controle do motor é necessário um dispositivo que consiga fornecer correntes mais elevadas do que aquelas disponíveis através dos pinos dos microcontroladores ou da Raspberry. O dispositivo escolhido foi o módulo ponte H baseado no componente



Figura 3.24 – Encoder LPD3806
Fonte: (ORANGE, 2014)

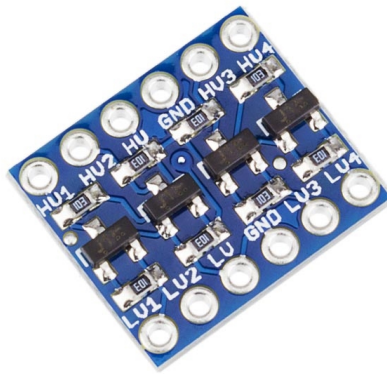


Figura 3.25 – Deslocador de nível lógico de 4 canais
Fonte: (POLOLU, s.d.)

TB66122FNG que é um *driver* de corrente para motor DC.

Neste componente são utilizados um sinal de PWM (*Pulse Width Modulation*) para controlar a potência e dois sinais de entrada para controlar o sentido de rotação dos motores (o controle de sentido rotação pode ser vistos na tabela 3.2). Este módulo tem capacidade de fornecer até 1.2 A para dois motores DC simultaneamente, podendo fornecer uma corrente de pico de até 2 A por 20 ms. Embora este módulo tenha 2 pontes H independentes essas pontes são usadas em conjunto de forma a terem o mesmo sentido de rotação e a mesma potência.

Para registrar a realização do experimento é necessário uma câmera, de preferência que possua uma boa conexão com a Raspberry utilizada. Para isso foram pesquisados vários modelos de câmeras que possuíssem comunicação com a Raspberry via interface serial de câmera (CSI; *Camera serial interface*). Dos modelos pesquisados haviam variações tanto

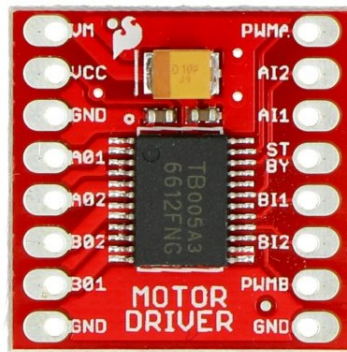


Figura 3.26 – Módulo ponte H

Fonte: (TOSHIBA, s.d.)

INA2	INA1	Motores
0	0	Parados
0	1	Sentido horário
1	0	Sentido Anti-horário
1	1	Parados

Tabela 3.2 – Sentido de rotação dos motores controlados pela ponte H

no número de megapixels da câmera quanto no ângulo de abertura (FOV; *Field of view*). Para uma boa visualização do projeto é desejável que o usuário consiga ver toda a barra do pêndulo na tela, sendo assim a distância da câmera para o pêndulo varia de acordo com o FOV e quanto maior o FOV menor deve ser esta distância. Dentre os modelos viáveis foi escolhido a câmera Rev 1.3, que possui 5 megapixel e um FOV de 70°.



Figura 3.27 – Câmera rev 1.3 e cabo CSI

Fonte: (RASPCAM)

Para construir a eletrônica do pêndulo invertido foi montado um circuito exemplificado na figura 3.28 onde pode ser visto que todos os sensores, atuadores e componentes intermediários estão conectados diretamente a MSP, com a Raspberry estando responsável apenas por programar e se comunicar com a MSP (ambas as funcionalidades executadas pela UART; *Universal Asynchronous Receiver/Transmitter*) e pelo uso da câmera.

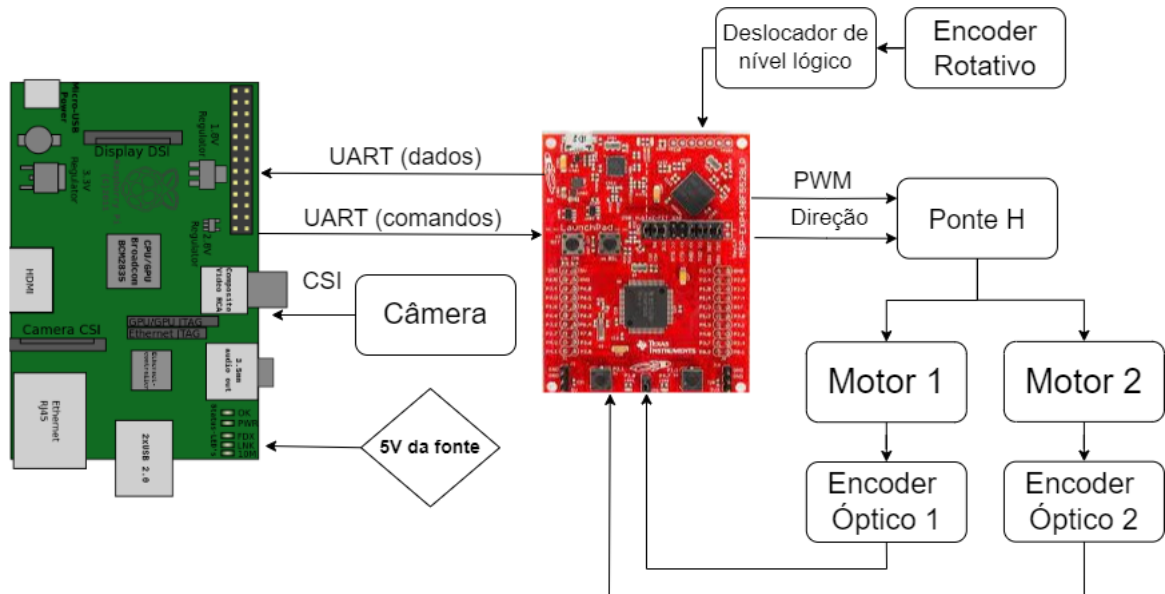


Figura 3.28 – Diagrama da conexão eletrônica

Para conectar todos os componentes e implementar a solução eletrônica como um todo foi projetado uma placa de circuito utilizando o *software* Altium Design. Primeiramente foi construído um esquemático do circuito do projeto, que além dos componentes já descritos também incluiu todos os conectores necessários, como pode ser visto na figura 3.29. Foram utilizados conectores do tipo kk da Molex, barras de pinos macho e barras de pinos fêmea e este circuito foi projetado para que a ponte h e o deslocador de nível lógico possam ser facilmente substituídos e que um microcontrolador qualquer possa ser utilizado nas barras de 20 pinos laterais. Para se adequar ao uso de diferentes micro controladores a placa foi projetada para operar com tensão de 3,3 V ou 5 V, sendo que esta escolha de tensão é realizada conectando determinados pinos da placa utilizando *jumpers*. Após o desenvolvimento do esquemático os componentes foram posicionados no projeto de PCB também utilizando o Altium. Este posicionamento, juntamente com o esquemático serviram de base para a construção de uma placa furada que é utilizada na montagem final deste projeto. A placa tem dimensões 7 cm x 9 cm e é fixada a base móvel utilizado espaçadores de nylon, e é conectada aos componentes externos a ela por cabos com conectores do tipo kk fêmea nas pontas. Nas figuras 3.29, 3.30 e 3.31 podem ser vistos o circuito projetado, a placa resultante e a placa furada construída.

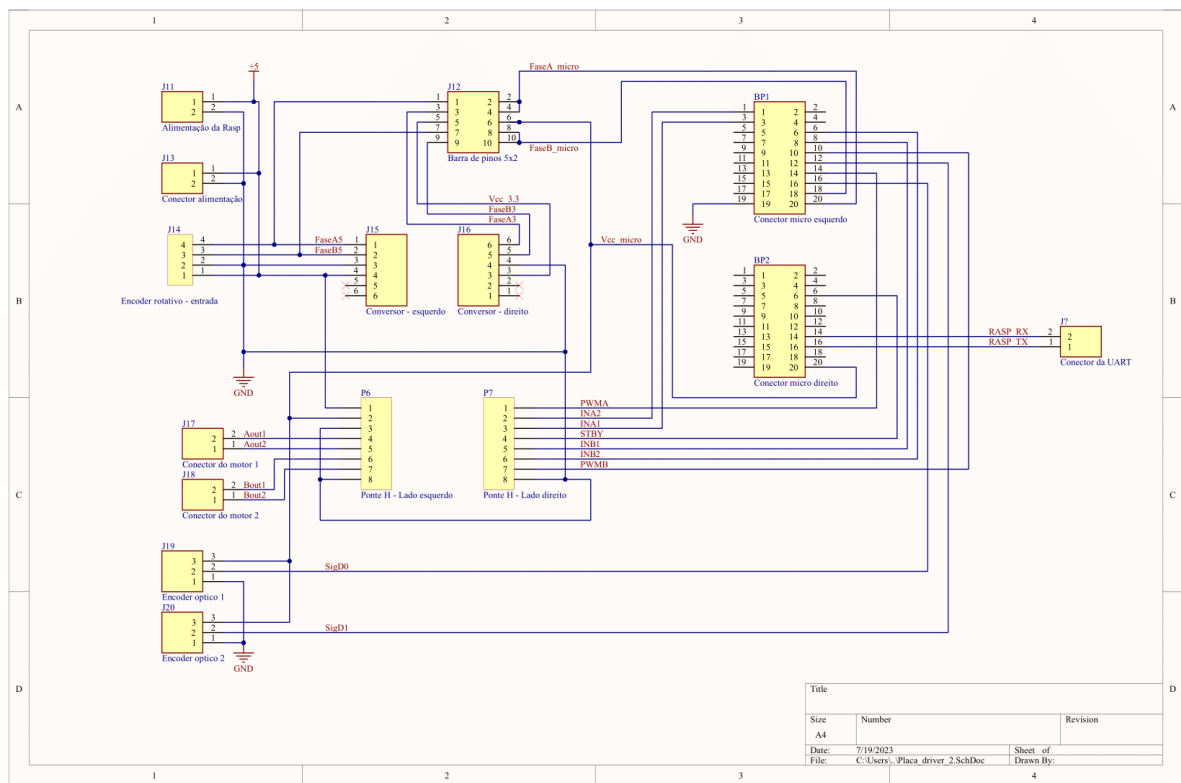


Figura 3.29 – Circuito projetado no Altium

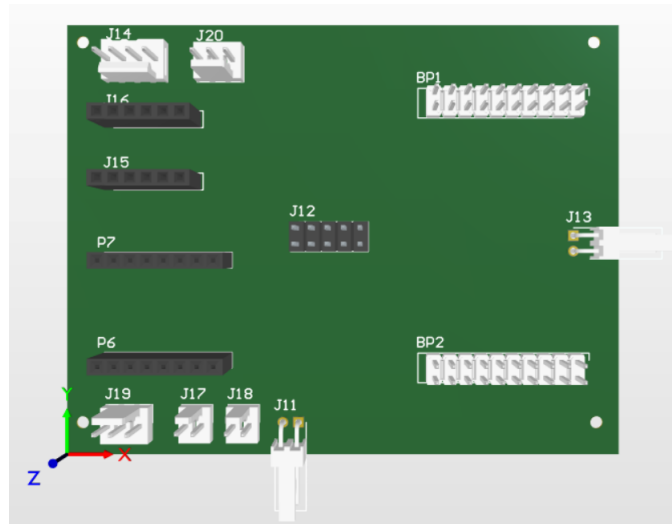


Figura 3.30 – Placa *driver* projetada no Altium

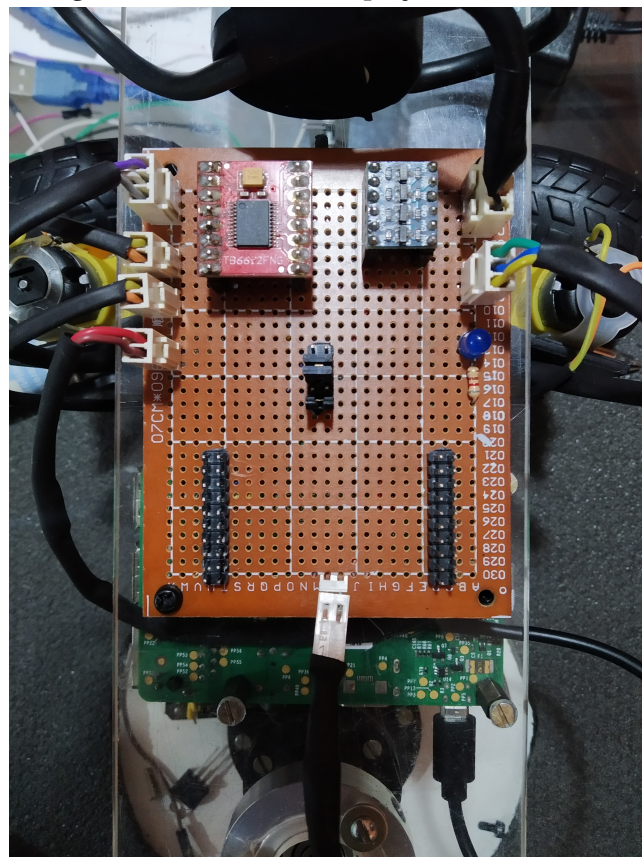


Figura 3.31 – Placa furada construída

A placa construída possui trilhas de alimentação para todos os componentes, com excessão da Raspberry e da MSP. A alimentação da Raspberry é feita pela entrada USB (*Universal Serial Bus*) micro utilizando um cabo que foi adaptado para ter um conector do tipo Molex kk em uma ponta e um conector micro na outra. Há também uma conexão entre a Raspberry e a MSP que utiliza as portas USB das duas. Essa conexão é responsável por alimentar a MSP, e quando solicitado utilizar o ez-FET para reprogramar ou monitorar as variáveis do programa (modo *debug* do CCS; *Code Composer Studio*).

3.3.2.1 MSP430F5529LP

O projeto demanda um microcontrolador para implementar o algoritmo de controle e também ler os valores dos sensores e comandar o motor, para isso podem ser utilizados vários modelos de microcontroladores, desde que seja criado uma placa ou algum outro tipo de suporte que faça conexão com as barras de pinos macho deixados na placa *driver*. Para esta versão do pêndulo foi escolhido o microcontrolador MSP430F5529 da Texas Instruments. Além de possuir todos os periféricos necessários para o controle e monitoramento do pêndulo o controlador escolhido possui o emulador eZ-FET embutido. O eZ-FET é um emulador embutido que possibilita a programação, avaliação e *debug*, o que permite avaliar o código enquanto a aplicação está executando sem usar nenhum *hardware* adicional (TEXAS INSTRUMENTS, 2013).

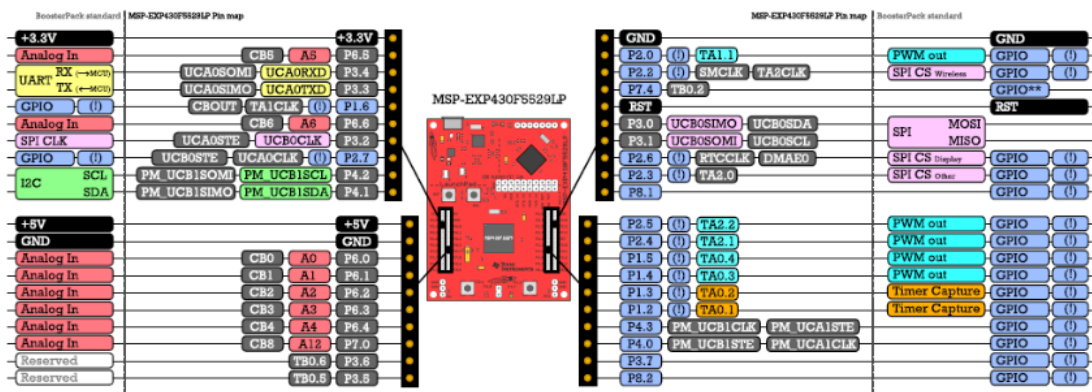


Figura 3.32 – pinos e periféricos da MSP430F5529

Fonte: (EMBEDDED LAB, s.d.)

A programação da MSP foi feita utilizando o Code Composer Studio (CCS) como compilador e *debugger* durante o desenvolvimento do *firmware* da MSP. O CCS é um ambiente de desenvolvimento integrado (*integrated development environment*; IDE) para os microcontroladores e processadores da Texas Instruments (TEXAS INSTRUMENTS, s.d.).

O *firmware* foi feito para conseguir ler os sensores, controlar os motores e se comunicar com a Raspberry. Inicialmente foi feita a configuração das portas e periféricos da MSP, sendo que foram utilizados 11 pinos, 1 módulo de comunicação e 3 *timers*. Após a configuração foram escritas as funções de interrupção do *encoder* rotativo, dos *encoders*

ópticos, da comunicação UART e dos *timers*. Dentre essas interrupções a mais importante para o projeto é a interrupção do *encoder* rotativo, já que é esta é a interrupção utilizada para medir o ângulo da barra, e portanto o algoritmo das interrupções foi desenvolvido para que esta interrupção possa ser executada mesmo enquanto outra interrupção está sendo processada.

O programa desenvolvido funciona da seguinte maneira: primeiro as configurações são realizadas, depois o programa fica esperando o usuário enviar um comando de ajuste dos parâmetros do PID ou de início de experimento. Quando o comando de início é detectado o programa reinicia as variáveis do programa, inicia o algoritmo de subida e quando a barra já está suficientemente perto dos 180° o algoritmo do PID tenta controlar a barra enquanto ela estiver acima da linha horizontal. Se a barra cair abaixo da linha horizontal o controlador PID para, os motores são desligados e o programa volta novamente a esperar um comando de início de experimento. O funcionamento explicado anteriormente está ilustrado na figura 3.33, que não apresenta a reprogramação dos parâmetros do PID porque esta mudança de parâmetros pode acontecer em qualquer momento do fluxo do programa.

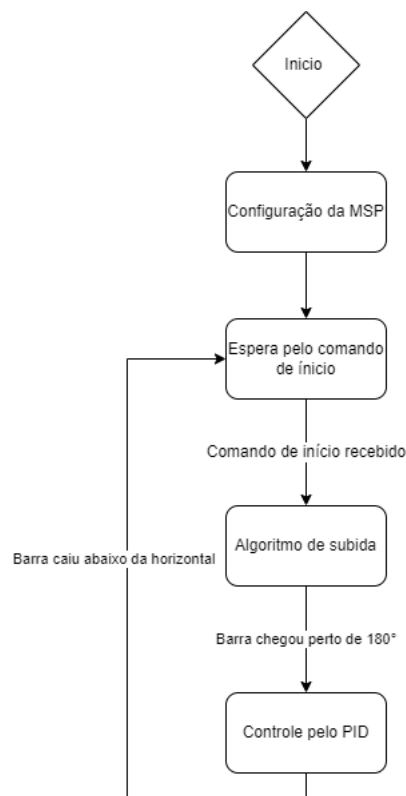


Figura 3.33 – Fluxo do experimento na MSP

3.3.2.2 Raspberry Pi

Uma das necessidades principais do projeto é comandar o microcontrolador escolhido, além de se comunicar para comandar o experimento e salvar os dados, e para isso é utilizado a Raspberry Pi 3B. A Raspberry Pi 3B é um computador de placa única que possui 40 pinos

de propósito geral além de possuir periféricos de comunicação UART e comunicação com a internet, sendo possível ser utilizada como intermediária da comunicação da MSP com o computador do usuário.

Para implementar a leitura, armazenamento de dados e conexão com a *internet* foi necessário desenvolver algumas funcionalidades na Raspberry. Primeiro foi desenvolvido o algoritmo para ler e salvar os dados da execução do experimento provenientes da MSP. Este algoritmo primeiro habilita a UART nos pinos 8 e 10 e configura os demais parâmetros necessários para estabelecer a comunicação entre os dois dispositivos, depois entra em um modo de espera pela próxima transmissão. O modo de espera foi projetado para esperar até que todos os dados de um comando completo sejam transmitidos (13 *bytes* para a transmissão no modo de subida e 37 *bytes* para o modo de controle do PID; o *data frame* de cada modo pode ser visto na figura 3.34), e quando a recepção do *frame* estiver completa estes dados são salvos como uma nova linha em um arquivo de texto.

Modo	Char de modo	Tempo	Angulo	Saída do PID	Erro de controle	Passos do motor 1	Passos do motor 2	Total de bytes
Controle do PID	1 byte - "<"	4 Bytes	8 Bytes	8 Bytes	8 Bytes	4 Bytes	4 Bytes	37
Subida	1 byte - ">"	4 Bytes	8 Bytes	-	-	-	-	13
Parar experimento	1 byte - "="	-	-	-	-	-	-	1

Figura 3.34 – Dados transmitidos pela MSP

Outra funcionalidade desenvolvida para comunicação foi o algoritmo para comandar o experimento, onde a Raspberry utiliza a UART para enviar os parâmetros do PID e controlar o fluxo do experimento enviando comandos de início, fim e reinício de experimento.

Para poder programar e também utilizar o *debugger* Ez-fet da MSP (o *debugger* utiliza a porta USB da MSP) foi instalado, na Raspberry (servidor) e no computador utilizado no teste e desenvolvimento (cliente) o *software* Virtual Here. Este *software* permite a criação de uma porta USB virtual entre servidor e cliente, desde que estejam na mesma rede. O Virtual Here foi utilizado para conseguir acessar os dados da memória da MSP utilizando o software CCS no modo *debugger* e também para programar a MSP, além de possibilitar que futuros usuários possam fazer o carregamento de outro programa sem a necessidade de utilizarem a conexão USB da MSP presencialmente.

3.3.3 Solução de *software*

Uma das necessidades do projeto é a possibilidade de traçar gráficos utilizando os dados gerados em testes como forma de monitorar o comportamento do experimento realizado. Para esse monitoramento é importante que sejam traçados gráficos de vários tipos de grandezas correlacionadas com o tempo. Os gráficos foram pensados de forma a trazer simultaneamente uma ou mais grandezas com relação ao tempo, o que facilita a visualização e a validação do algoritmo de subida e controle e a implementação e funcionamento do sistema do pêndulo como um todo. Com este propósito foram pensados os gráficos de posição

angular da barra e velocidade angular da barra, erro de controle e a saída do PID associada e de PWM do motor e velocidade angular da roda (também será possível traçar o gráfico com apenas uma grandeza em relação ao tempo). Este programa de gráficos será realizado em um computador externo e utilizará os dados experimentais transferidos da Raspberry através do protocolo de transferência de dados SCP (*Secure Copy Protocol*).

Outra funcionalidade importante é a possibilidade de gerar testes automáticos. Realizar várias vezes um teste garante uma maior confiança nos resultados do método de controle utilizado. Os testes automáticos foram criados principalmente para a validação do algoritmo de subida autônoma da barra, mas podem também serem utilizados na validação de metodologias de controle. Durante a realização do teste automático o pêndulo sobe a barra e tenta controlar utilizando uma abordagem de controle já programada, e ao final do controle (quando foi passado um tempo determinado ou quando a barra cai abaixo da linha horizontal) os dados deste teste são salvos na Raspberry. Este procedimento será utilizado repetidas vezes para gerar uma base de dados com vários testes, e em seguida os resultados serão portados para um computador externo utilizando SCP. No computador externo devem ser calculados alguns parâmetros estatísticos usando os testes, sendo estes:

- Média de tempo que o controlador conseguiu controlar a barra.
- Média de tempo que o algoritmo de subida demora pra colocar a barra em posição vertical.
- Média de velocidade da barra quando o pêndulo trocou do algoritmo de subida para o algoritmo de controle.

3.3.4 Sistema de controle

A execução de um experimento do pêndulo de invertido é dividida em duas partes: primeiro levantar a barra até a posição de equilíbrio vertical e depois estabilizar a barra nesta posição. Nesta seção deste documento é descrita as metodologias de controle proposta para cada uma das duas partes do experimento.

3.3.4.1 Controle PID

O controlador do pêndulo foi desenvolvido para ser embarcado como *firmware* na MSP, e portanto deveria ser um controlador PID digital. As próximas equações foram baseadas nas equações da seção de implementação digital apresentadas no livro *PID controllers: theory, design, and tuning* (°ASTRÖM; HÄGGLUND, 1995) e apresentam as aproximações feitas para cada ação de controle que no fim serão somadas resultando na equação de saída do PID implementado. A ação proporcional é implementada simplesmente substituindo a

variável continua pela sua versão amostrada (equação (3.1)), onde $P[n]$ e $e[n]$ são a ação proporcional e o erro de controle (ambos na amostra n) e K_p é a constante proporcional.

$$P[n] = K_p \cdot e[n] \quad (3.1)$$

A ação integral foi obtida a partir da equação de diferença retrógrada e foi escrita utilizando um somatório das amostras dos erros, onde T é o período de amostragem e $\frac{K}{T_i}$ foi trocado pelo ganho K_i . Nas equações abaixo os termos $I[n+1]$, $I[n]$ e $e[n+1]$ são a ação integral em $n+1$, a ação integral em n e o erro em $n+1$, e posteriormente a equação (3.2) é rescrita na forma da equação (3.3) para representar esta ação na amostra atual. Na equação (3.4) o termo do somatório realiza a soma dos erros de controle desde a primeira amostra ($m=1$) até a amostra atual ($m=n$), sendo que esta soma é multiplicada pela constante de integração K_i .

$$I[n+1] = I[n] + \frac{KT}{T_i} \cdot e[n+1] \quad (3.2)$$

$$I[n] = TK_i \cdot e[n] + TK_i \cdot e[n-1] + \dots + TK_i \cdot e[0], K_i = \frac{K}{T_i} \quad (3.3)$$

$$I[n] = TK_i \cdot \sum_{m=1}^n e[m] \quad (3.4)$$

A ação derivativa foi obtida a partir da equação de diferença retrógrada sem utilizar o termo que limita o ganho derivativo, assim apenas o termo da variação do erro e seus coeficientes aparecem na equação a seguir, onde o produto KT_d foi substituído por K_d . Na equação (3.5) o termo $D[n]$ é o valor da ação derivativa, que é calculada pela diferença entre o erro atual e o erro da amostra anterior ($e[n] - e[n-1]$) multiplicada pela constante de derivação K_d .

$$D[n] = K_d \cdot (e[n] - e[n-1]), K_d = KT_d \quad (3.5)$$

As três ações calculadas anteriormente são somadas para formar a equação (3.6) que mostra a saída do controlador, que é atualizada a cada iteração de cálculo do PID.

$$SAIDA_{PID}[n] = K_p e[n] + K_i T \sum_{m=1}^n e[m] + \frac{K_d}{T} (e[n] - e[n-1]) \quad (3.6)$$

A saída do controlador PID implementado pode ter valores fracionários e negativos, o que não é possível de aplicar aos motores, já que a variável de PWM só aceita valores inteiros e positivos. Para contornar este problema o sinal do resultado do cálculo do PID é utilizado para selecionar o sentido de rotação dos motores, assim um sinal positivo ativa os motores em no sentido positivo e o sinal negativo ativa os motores no sentido oposto, como pode ser

visto na figura 3.35. Já para o ciclo de trabalho do PWM é utilizado o valor calculado pelo PID arredondado e em módulo.

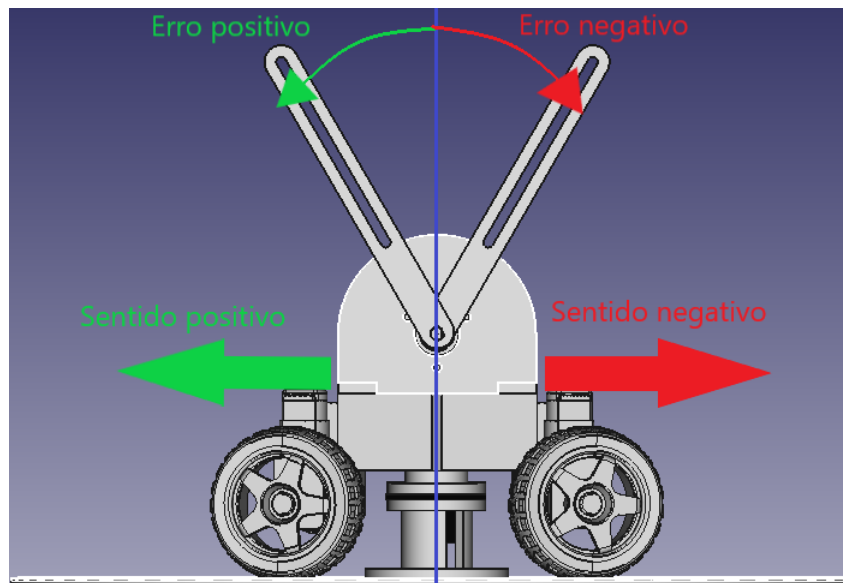


Figura 3.35 – Sentido de movimentação da base e sentido do erro de ângulo da barra

3.3.4.2 Métodos para subir a barra

Na situação de controle ideal a barra do pêndulo começaria com velocidade zero e na posição vertical, assim o controle do PID entraria no comando do experimento sem ter que compensar nenhuma energia cinética no momento inicial do controle (muitas modelagens do pêndulo e de controladores partem de uma posição vertical sem nenhuma velocidade angular). Para subir a barra foram testados diversos tipos de algoritmo e os resultados foram comparados para ver qual mais se aproximava da solução ideal.

Primeiramente foi testado um algoritmo de subida que utilizava 100% do PWM mudando o sentido de rotação dos motores quando a barra parava de subir. Para tentar reduzir a velocidade em 180° foram realizadas algumas modificações no algoritmo anterior. Primeiro os motores foram limitados a estarem ligados apenas enquanto a barra do pêndulo estivesse abaixo da linha horizontal, reduzindo pela metade o tempo de ativação dos motores. Outra modificação adicionada neste algoritmo foi a de limitação a potência dos motores conforme a barra alcançasse um ponto mais alto, ou seja, quanto mais alto a barra for menos potência os motores vão ter quando forem ligados novamente, sendo que existe um limite mínimo para este valor. Neste algoritmo também foi adicionado uma constante de ajuste experimental que multiplica o valor final de PWM para tentar minimizar a velocidade no ponto de equilíbrio.

Outro método de subida testado foi a subida utilizando um PID. Nesta abordagem a velocidade do motor é controlada pelo PID e o sentido de rotação dos motores é alterado cada vez que a barra do pêndulo para de subir (o mesmo algoritmo de troca de sentido

dos motores dos outros algoritmos de subida). Para que o cálculo do erro seja sempre igual independente do sinal do ângulo, quando o valor do ângulo da barra muda de sinal o valor do *setpoint* também é alterado (se o ângulo é positivo: $setpoint = 180^\circ$; se o ângulo é negativo: $setpoint = -180^\circ$). Como esse algoritmo requer a especificação dos parâmetros do PID foi utilizado o método de alterar uma constante enquanto as outras ficam fixas para testar várias combinações de K_p , K_i e K_d . Os resultados obtidos são discutidos mais a frente no texto.

3.4 Metodologia do desenvolvimento do projeto

O desenvolvimento do experimento do pêndulo invertido possui 3 frentes: estrutura, eletrônica e *software*. Por se tratar de um projeto que envolveu o desenvolvimento de várias frentes simultaneamente muitas modificações foram sendo realizadas durante a sua execução, e portanto uma modificação em uma determinada frente muitas vezes demandava uma alteração nas outras frentes para garantir compatibilidade. Esta seção deste trabalho documenta a metodologia utilizada no desenvolvimento de cada frente do projeto.

3.4.1 Estrutura

O desenvolvimento mecânico do experimento visa criar a estrutura do pêndulo rotacional, na qual a base rotativa se movimenta impulsionado por um conjunto roda e motor fixado a sua extremidade. Para este desenvolvimento foi utilizada o *software* Freecad para projetar o CAD das peças mecânicas do pêndulo, sendo que algumas destas peças passaram por várias modificações a fim de remover alguns efeitos indesejados como folgas e má fixações que foram observadas durante a produção e montagem do pêndulo. Foram levadas em conta a sustentação dos componentes eletrônicos, a fabricação e montagem do pêndulo e o tipo de material utilizado nas peças. Após o projeto de um componente mecânico está peça era fabricada (por impressão 3D, corte a laser, fabricação no torno, etc.) e testada para verificar se a solução proposta funcionava bem e se adequava a estrutura já existente. Este processo de projeto, fabricação e teste levou algumas peças a serem repensadas e reprojatadas várias vezes durante todas as versões do pêndulo até a atual.

3.4.2 Eletrônica

Os componentes e conexões necessários para o funcionamento do projeto foram pensados a partir das funcionalidades necessárias para o experimento remoto, e a partir destas funcionalidades foram estabelecidos critérios de escolha que guiaram a pesquisa e a compra dos componentes. Boa parte dos critérios dos componentes foi baseado nas necessidades e compatibilidades da Raspberry, já que seu uso já estava pré estabelecido pela possibilidade de conexão com computadores utilizando protocolo SSH e por sua disponibilidade e custo benefício.

O ângulo da barra do pêndulo é uma variável extremamente importante para o controle do pêndulo, sendo assim necessita de uma medição mais precisa, e para atingir essa precisão um *encoder* óptico incremental foi incorporado ao projeto. Para medir a velocidade do motor foi pesquisado um sensor que pudesse ser acoplado a roda ou ao motor de forma a medir as rotações do eixo. O critério de escolha do motor se baseou na tensão de alimentação, que deveria ser próximo a tensão de alimentação de 5 V da fonte, sendo que o motor deveria possuir torque suficiente para movimentar a base a uma velocidade razoável. Com base na tensão do motor escolhido e na necessidade de controle por PWM foi escolhido uma ponte H para funcionar como intermediário e fornecer corrente ao motor. A escolha da fonte para o sistema se baseou primeiramente na alimentação da Raspberry, que pode ser alimentada pela porta micro USB ou diretamente pelos pinos através das conexões de 5 V. Outro fator de escolha pela alimentação de 5 V foi a tensão de alimentação do motor escolhido, que pode ser de 3,3 V até 6 V, mas que apresenta potencias maiores em alimentações maiores. Como a fonte de alimentação fica por debaixo da base e os componentes eletrônicos ficam por cima da parte móvel há a necessidade de passar os fios de alimentação de baixo para cima sem que estes fios fiquem sujeitos a torção quando a parte móvel estiver girando. Para atender a esse requisito foram pesquisados componentes que pudessem servir como conexão entre a fonte e a parte superior e que possuísse algum mecanismo de rotação. Os critérios de escolha da câmera fora principalmente a compatibilidade com a Raspberry e um baixo custo de aquisição em relação aos vários modelos de câmeras disponíveis no mercado.

Com a decisão e compra dos componentes os subsistemas da parte eletrônica foram testados e a realização destes testes ajudou a formar as estruturas iniciais do código geral do pêndulo. Para este teste todos os componentes foram montados na estrutura, e foram conectados através de uma *protoboard* que foi fixada temporariamente na parte móvel. O circuito do teste foi o mesmo circuito exibido na figura 3.28.

3.4.3 *Software*

Os códigos deste trabalho foram desenvolvidos para controlar e monitorar o pêndulo e visualizar os dados obtidos dos testes, sendo que a frente de *software* é muito dependente das escolhas tomadas pela frente de eletrônica e por isso foi sendo modificada paralelamente com as modificações da eletrônica.

Para controlar o pêndulo foi utilizada a linguagem C na IDE CCS (ambiente de desenvolvimento do fabricante da MSP) para desenvolver o *firmware* da MSP começando primeiro com os algoritmos de leituras de sensores passando por várias etapas de modificações, incremento de funções e testes até chegar ao estado atual que tem a leitura de sensores, comando dos motores, algoritmo de controle e comunicação UART desenvolvidas no mesmo código. A recepção de dados e o comando dos testes e experimentos foi realizado usando *scripts* em C que eram feitos no computador e passados para a Raspberry, compilados e

executados utilizando o SSH. Para visualizar os dados foram desenvolvidos no VSC algoritmos em Python para possibilitar a leitura dos dados e a exibição de gráficos. O código para mostrar os dados foi acessado em um arquivo Batch para desenvolver um executável no Windows que acesse a Raspberry, salve os dados do ultimo teste no computador e mostre os gráficos deste ultimo teste.

3.5 Versões do pêndulo

O projeto do pêndulo invertido foi sendo construído e melhorado de forma contínua desde sua primeira versão apresentada em trabalhos anteriores. As modificações foram sendo realizadas a medida em que eram descobertos novos problemas durante os testes do pêndulo.

Na primeira versão do pêndulo foi construído uma versão mais simples que serviu de protótipo. Nesta versão a estrutura mecânica tinha alguns componentes improvisados e a Raspberry controlava o experimento diretamente, não tendo nenhum microcontrolador entre a Raspberry e os sensores e atuadores. Este projeto apresentava alguns problemas como:

- O tubo de sustentação ainda não tinha sido fabricado e teve que ser improvisado com um tubo de PVC.
- Não era possível apertar o parafuso que segurava a barra do pêndulo (a barra foi fabricada em um plástico que não aceitava bem rosca), o que produzia uma folga entre a barra e o eixo de rotação.
- A base móvel se apoiava diretamente em cima do anel deslizante (esforços mecânicos no eixo de rotação do anel deslizante podem danificar a peça).
- A Raspberry e a *protoboard* utilizada nos testes não estavam bem fixadas a base móvel.
- A base móvel se apoiava somente em uma roda, o que causava uma inclinação indesejada da base móvel em torno do eixo da roda.

Visando resolver os problemas anteriormente citados foram feitas varias modificações mecânicas no projeto. Entre elas temos:

- Foi projetado e fabricado em 3D um tubo de sustentação que apoiasse a base móvel e o anel deslizante de forma adequada.
- Foi projetada uma barra do pêndulo em alumínio, o que possibilita apertar bem um parafuso para eliminar possíveis folgas entre a barra e o eixo.

- Foi fabricada uma peça de alumínio em formato de rosca para servir como peça intermediária de fixação entre o rolamento flangeado e a base móvel, retirando assim o esforço mecânico sobre o anel deslizante.
- A base móvel foi refeita em acrílico para possibilitar uma melhor visão da barra e dos outros componentes.
- O circuito do pêndulo foi implementado utilizando uma placa furada para melhorar a organização do circuito e torna-lo menos suscetível a ruídos ou desconexões causados pela movimentação ou vibração dos componentes eletrônicos.
- A Raspberry e a placa furada foram fixadas a base móvel utilizando espaçadores de nylon.
- Foi desenvolvida uma estrutura de suporte com 2 motores e 2 rodas para eliminar os efeitos da rotação indesejada.

3.6 Testes do pêndulo

Para caracterizar e ilustrar o comportamento do experimento construído foram realizados alguns testes utilizando o experimento já montado. O primeiro teste foi feito com o objetivo de observar o comportamento da barra do pêndulo sem influência de nenhuma energia dos motores, para isso a barra do pêndulo foi posicionada sem peso adicional em 180° (no referencial utilizado a posição de 0° corresponde ao pêndulo apontando para baixo). Com a barra na posição vertical se iniciou o processo de amostragem do ângulo a cada 50 ms, e depois a barra foi solta para obter os dados da posição angular enquanto oscilava até parar novamente na posição de repouso em 0° .

Para visualizar aceleração que os motores exercem na barra durante a subida foi realizado um teste que consistiu em utilizar um degrau de PWM com a barra ainda na posição de repouso. Este degrau foi mantido até a barra se estabilizar perto da posição de repouso inicial e o gráfico obtido foi utilizado para visualizar o máximo ganho de ângulo da barra com uma única inversão dos motores.

Para caracterizar a resposta das rodas a entrada de PWM foram desenvolvidos 2 testes. O primeiro teste foi feito para observar a resposta das rodas ao degrau de PWM e para isso foi utilizado uma função degrau com 100% do PWM, ou seja, toda a potência do motor. O teste com a função degrau foi realizado invertendo o sentido de rotação e também com as rodas levantadas e com as rodas impulsionando a base móvel, e no total foram feitos 4 testes deste tipo.

O segundo teste consistiu em zerar o PWM e aumentar este valor, que vai de 0 a 1000, em 10 a cada 2 segundos enquanto eram medidas as velocidades de cada roda. Este teste

foi realizado tanto com as rodas suspensas para testar a velocidade dos motores sem carga quanto com as rodas impulsionando a base móvel.

3.6.1 Teste de derrapagem

Um dos problemas observados após a montagem do pendulo e após alguns testes tentando estabiliza-lo é a derrapagem das rodas durante os testes, o que acontecia principalmente quando os motores já estavam girando e invertiam o sentido de rotação. Para tentar solucionar este problema foram testadas e analisadas algumas possíveis soluções para aumentar o atrito das rodas com a borracha que cobre a base. A fim de visualizar o impacto das possíveis soluções foram realizados testes de velocidade com inversão de sentido em 5 abordagens para as rodas: apenas as rodas que já eram utilizadas, as rodas com elásticos (figura 3.36), as rodas com uma borracha com grampos (figura 3.38), rodas com borracha de câmara de ar (figura 3.40) e rodas com borracha de câmara de ar sobre uma superfície de papelão (figura 3.41).



Figura 3.36 – Roda com elástico



Figura 3.37 – Borracha com grampos produzida

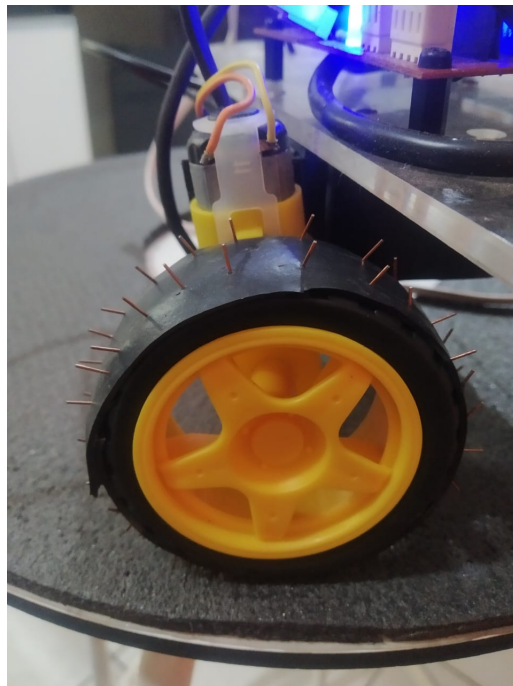


Figura 3.38 – Roda com grampo



Figura 3.39 – Tira de borracha produzida

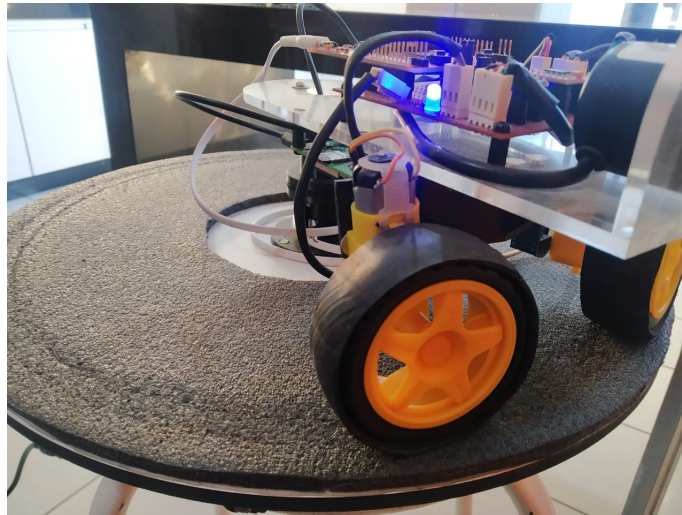


Figura 3.40 – Roda com borracha

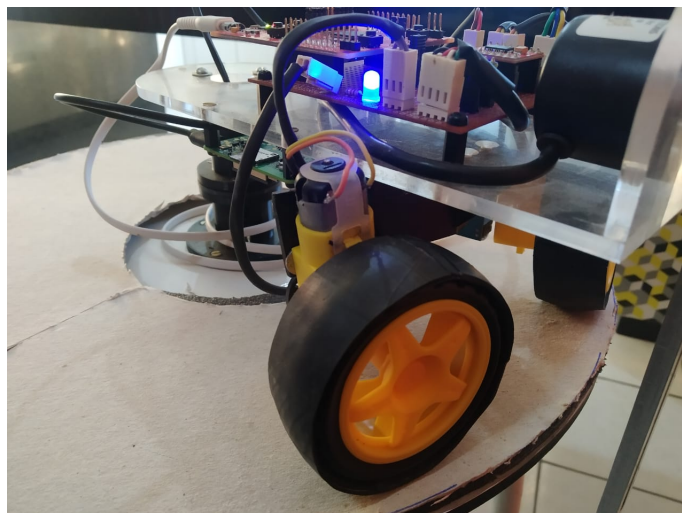


Figura 3.41 – Roda com borracha sobre papelão

No teste com elásticos foram utilizados 3 elásticos de látex de 2mm nas rodas e para o teste com grampos foram confeccionadas 2 tiras de borracha com grampos (figura 3.37), cujo intuito é atravessar a borracha da base e travar a roda para que ela não apresente derrapamento. Para o teste com a borracha foi produzido 2 tiras de borracha semelhantes a anterior, mas sem grampos, e no ultimo teste essas tiras foram utilizadas por cima de um papelão fixado a base onde as rodas se apoiam. Em todos os testes foram obtidos dados de velocidade das rodas que foram utilizados para calcular alguns parâmetros e analisar os efeitos das modificações no atrito das rodas.

A derrapagem percebida durante os testes era maior quando o motor já estava ligado e invertia o sentido de rotação, por isso foi desenvolvido o mesmo teste anterior, mas invertendo os motores a cada 1 segundo para obter gráficos que melhor exemplifiquem a derrapagem quando os motores são invertidos, o que acontece muitas vezes quando a barra do pêndulo está em torno do ponto de equilíbrio.

3.6.2 Teste de estabilização

Após a realização dos testes de derrapagem foi escolhido a borracha sobre a superfície de papelão para ser utilizado durante os testes de controle, que tem como objetivo obter parâmetros de PID que melhor estabilizem a barra para cima. Foram realizados 2 tipos de testes: o primeiro com a barra do pêndulo sendo colocada na posição vertical manualmente (tentando assim minimizar a velocidade angular da barra quando o PID assumia o controle), e o segundo com a barra sendo levantada pelo algoritmo de subida PID com os parâmetros $K_p = 2,1, K_i = 0, K_d = 100$ que foi descrito na seção 3.3.4.2. Nos dois tipos de teste os parâmetros do PID foram ajustados por tentativa e erro seguindo o seguinte método:

- Todos os parâmetros são zerados e o período de controle do PID é colocado em 10 ms.
- Os valores de K_p são aumentados gradativamente e são testados no controle.
- É escolhido o valor que produziu melhores resultados nos gráficos em relação ao tempo máximo com a barra estabilizada, número de inversões do PID e saturação dos motores.
- O valor de K_p é mantido no melhor resultado anterior, mas podendo ser reduzido, enquanto os valores de K_i são aumentados gradativamente.
- É escolhido um valor de K_i de acordo com os critérios anteriores.
- O valor de K_d é aumentado gradativamente enquanto os valores de K_p e K_i são mantidos constantes.
- Um valor de K_d é escolhido e por fim temos todos os parâmetros do PID.

Para o teste com a subida manual da barra primeiro os valores dos parâmetros eram ajustados, depois a barra era colocada manualmente na posição de 180° e então era solta, sendo que o PID de controle da barra é ligado quando a barra esta a pouco menos de 1° de distancia da posição desejada de 180° . Para o teste com subida pelo PID os testes foram realizados da mesma forma que para a subida manual. Os valores foram testados e ajustados um de cada vez, começando pelo K_p e terminando com o K_d .

4 Análise dos resultados

Nesta seção são apresentados os resultados e as análises dos desenvolvimentos e testes realizados nas frentes do projeto.

4.1 Métodos para subir a barra

Para visualizar os resultados dos métodos de subida de barra foram traçados gráficos onde a linha vermelha é a velocidade angular e a linha azul é a posição angular da barra. O gráfico de subida do algoritmo com 100% do PWM mudando o sentido de rotação dos motores pode ser visto em 4.42, nele é possível ver que o algoritmo sobe a barra até a posição de 180°, mas ainda com muita velocidade angular.

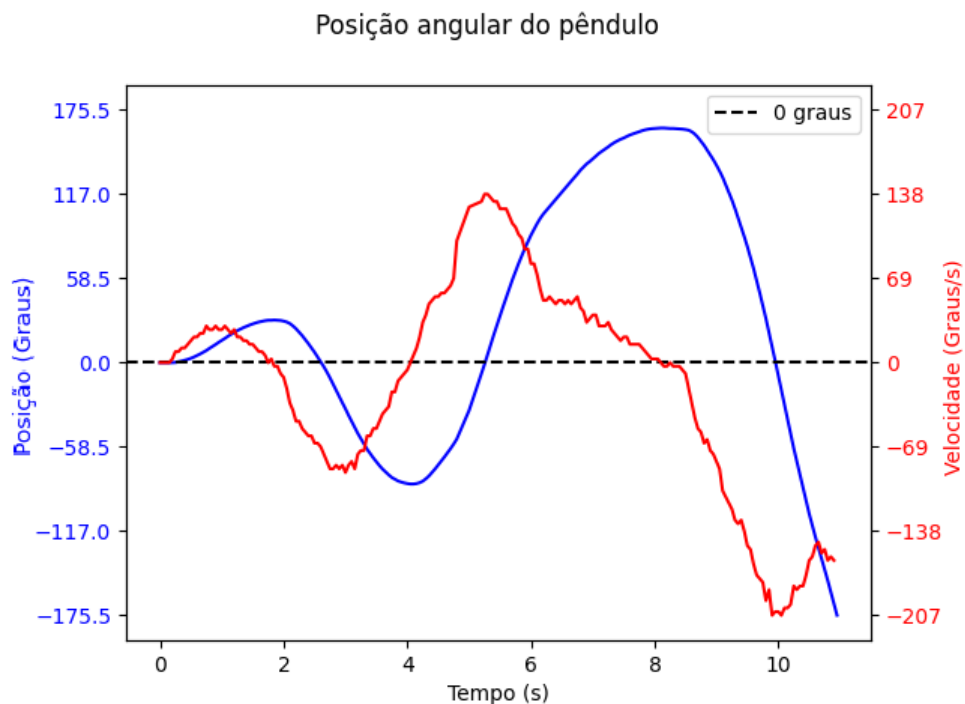


Figura 4.42 – Subida com motores sempre ligados

A partir das modificações do algoritmo anterior limitando o tempo em que os motores ficam ligados foi testado e obtido o gráfico da figura 4.43 em que a velocidade no ponto próximo a 180° diminuiu, mas ainda tem um valor de cerca de 42 graus/s, o que é relativamente rápido.

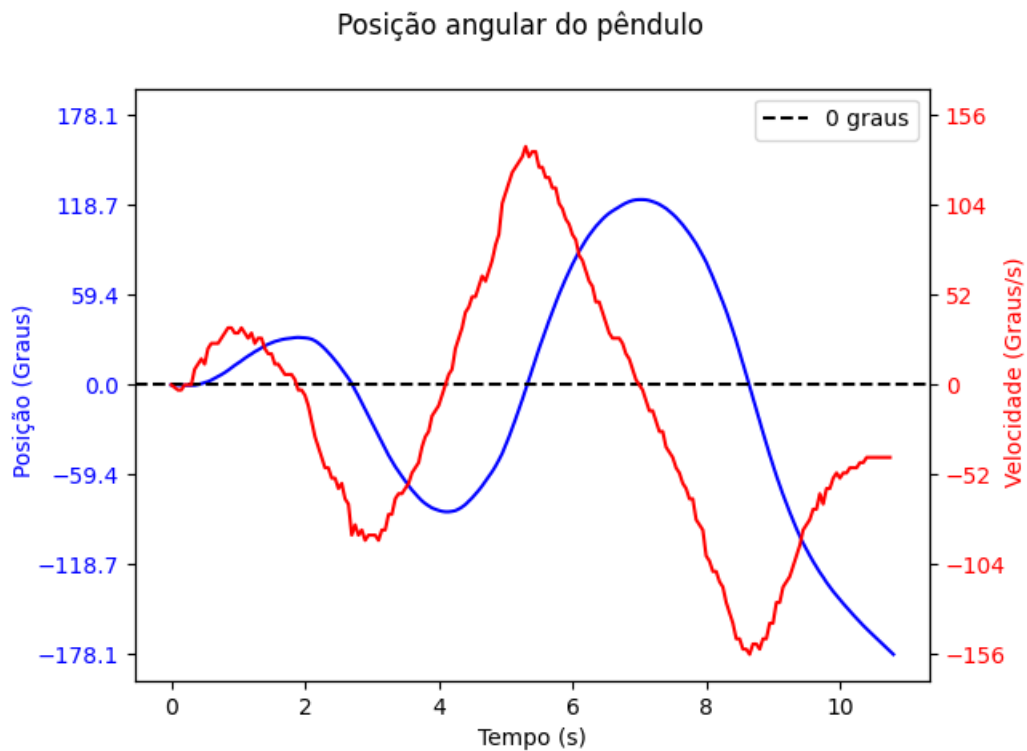


Figura 4.43 – Subida com motores ligados abaixo da linha horizontal

Na figura 4.44 pode ser visto o gráfico do teste de subida que utilizou o algoritmo configurado com um limite mínimo de PWM de 600 e uma constante de percentual de 75%. Comparando os gráficos das duas estratégias descritas anteriormente pode ser visto que o algoritmo com limitação de PWM colocou a barra perto dos 180° com menos velocidade, mas ao custo de ter um maior tempo e mais oscilações da barra.

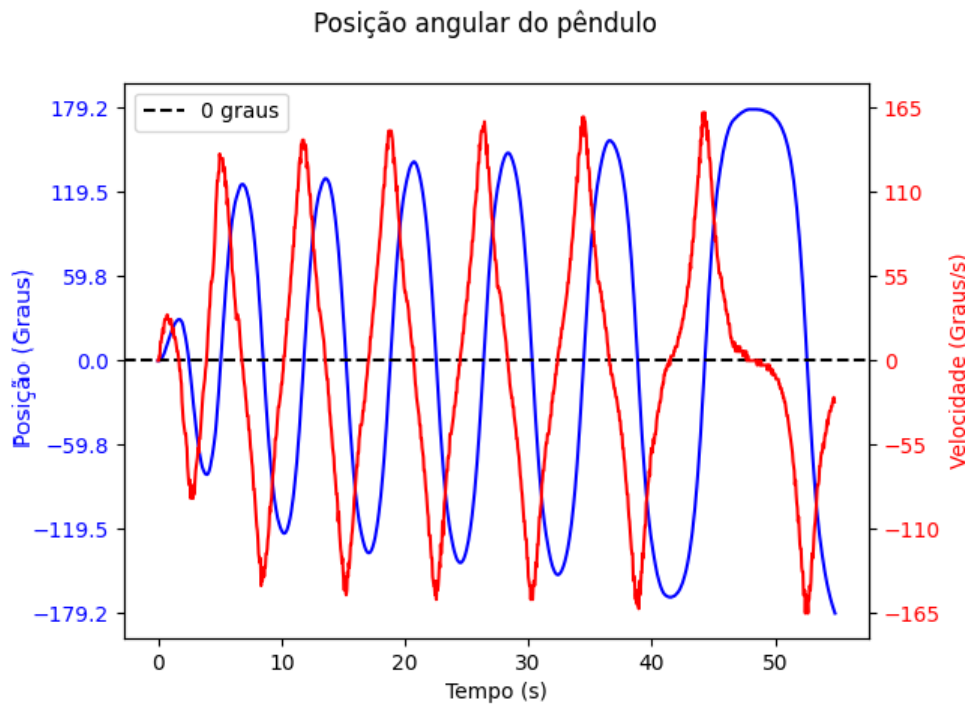


Figura 4.44 – Subida com dosagem da força

Os resultados da subida de barra utilizando o PID são apresentados a seguir. Como a barra inicia embaixo e o *setpoint* é acima o erro começa já muito grande e o integrador do PID satura muito rapidamente, o que causa uma subida de com muita aceleração e muita velocidade quando a barra chega no alto. Por conta disso K_i foi colocado em zero, a parte proporcional do PID foi utilizada com valores baixos e K_d foi colocado em valores mais altos. Dos vários ajustes de parâmetros e testes realizados os melhores resultados foram obtidos com valores em torno de $K_p = 2,1$, $K_i = 0$, e $K_d = 100$.

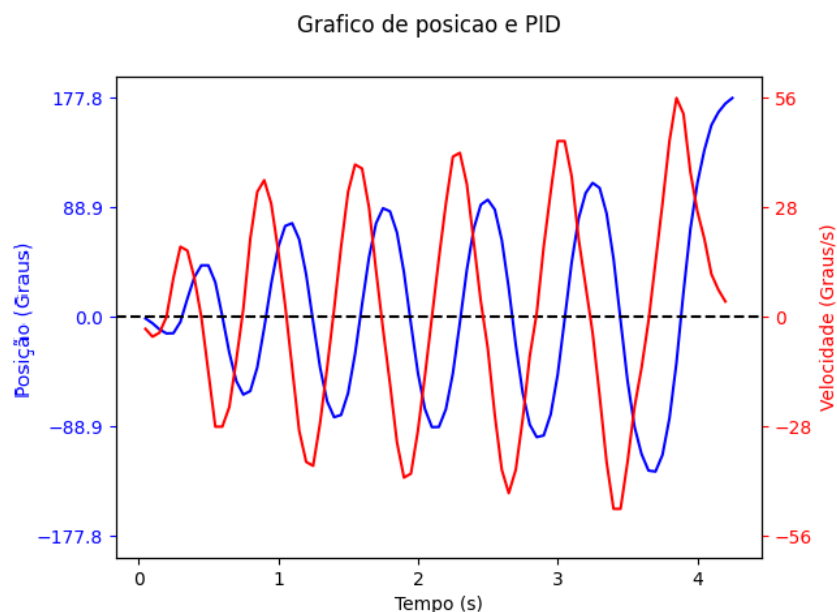


Figura 4.45 – Subida com PID. $K_p = 2.1$, $K_i = 0$, $K_d = 100$

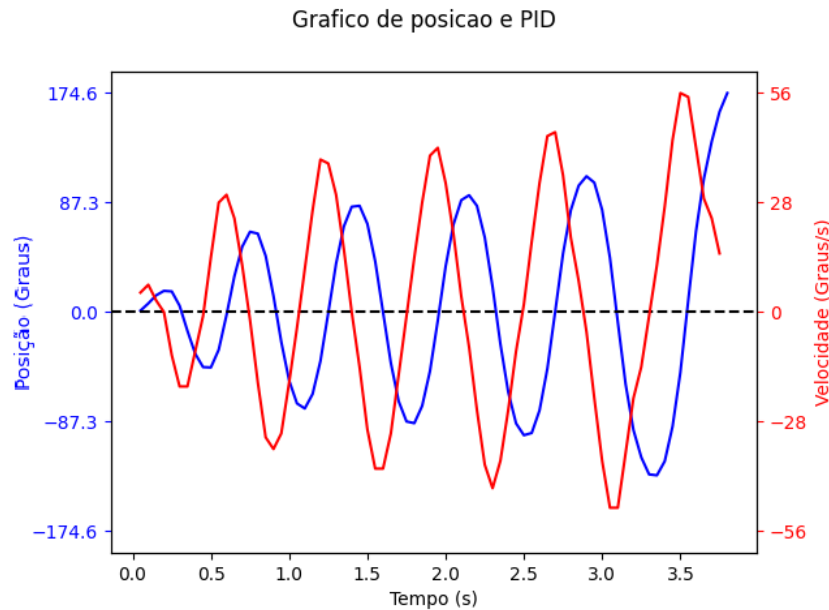


Figura 4.46 – Subida com PID. $K_p = 2.2, K_i = 0, K_d = 100$

Nos gráficos 4.45 e 4.46 pode ser visto que o algoritmo de subida também colocou a barra na vertical com relativa baixa velocidade, e em comparação aos outros algoritmos obteve o melhor resultado.

4.2 Testes do pêndulo

Os resultados dos testes são apresentados e discutidos mais detalhadamente nesta seção, tendo em vista que estes mesmos resultados são utilizados de forma mais geral nas análises de cada frente de desenvolvimento.

4.2.1 Testes com a barra do pêndulo

No gráfico da figura 4.47 pode ser visto o gráfico resultante do teste com a barra caindo sem influência do motor, e pode ser visto que o pêndulo depois de solto começa a oscilar de acordo com uma cossenoide amortecida, sendo que este é o comportamento esperado para um pêndulo amortecido.

Com os dados gerados no teste anterior foi possível calcular a frequência natural do pêndulo sem atuação. Esta frequência foi obtida calculando a média de tempo entre três cruzamentos por zeros sucessivos e resultou em um período de oscilação de 645 ms, o que resulta numa frequência de 1,55 Hz. Como o controle implementado é digital é importante fazer uma comparação entre a frequência natural do sistema e a frequência de controle para garantir que esta frequência é minimamente adequada. O PID utilizado tem o período de controle ajustável, mas nos testes foi utilizado um período de no mínimo 10 ms que resulta

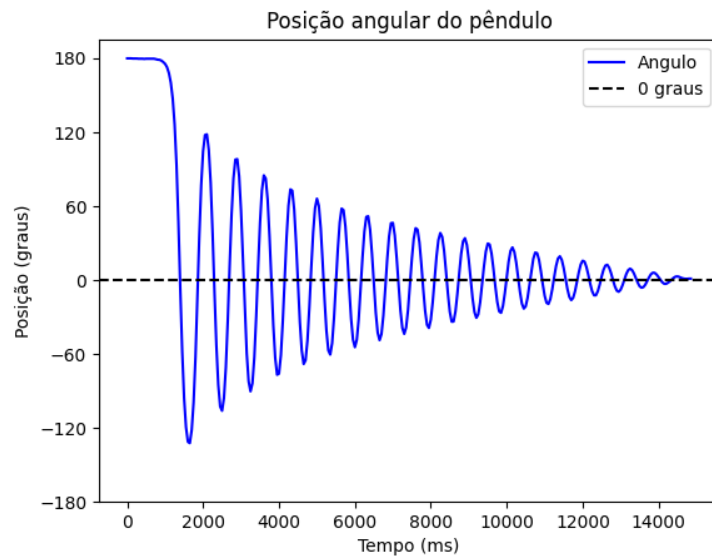


Figura 4.47 – Gráfico do pêndulo em queda

numa frequência de 100 Hz, ou seja 64 vezes a frequência natural, o que é suficiente para o controle.

Do teste com função degrau de PWM nos motores e barra em posição inicial em repouso foi obtido o gráfico da figura 4.48. Neste gráfico pode ser visto que o degrau de PWM, que é ativado após a linha tracejada, promoveu uma mudança de posição que gera uma oscilação que é dissipada com o passar do tempo, restando apenas um ruído. Deste teste foi aferido o ganho angular máximo de um único impulso, esta medição é feita observando o máximo que o ângulo do pêndulo chega depois da aplicação do degrau, com esse ângulo máximo sendo de 22°.

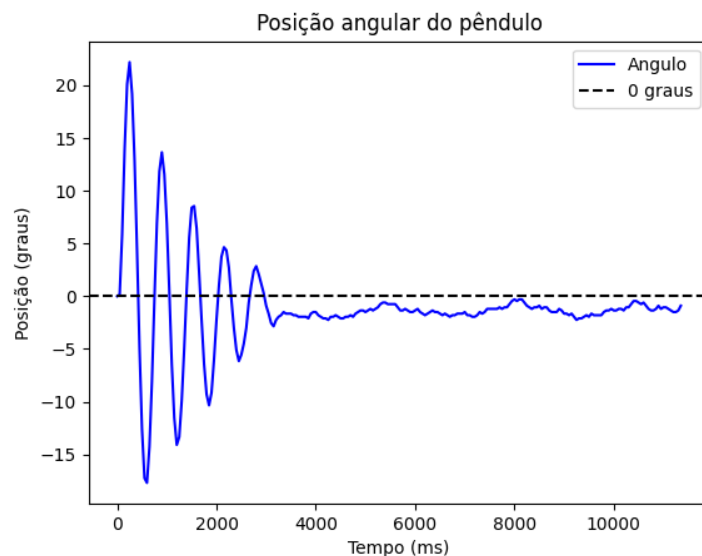


Figura 4.48 – Gráfico do pêndulo com função degrau

4.2.2 Testes de velocidade das rodas

Durante os testes de velocidade da roda com função degrau os motores foram ativados com 1 segundo de duração do experimento e os dados coletados resultaram nos gráficos das figuras 4.49 até 4.52 e na tabela 4.3.

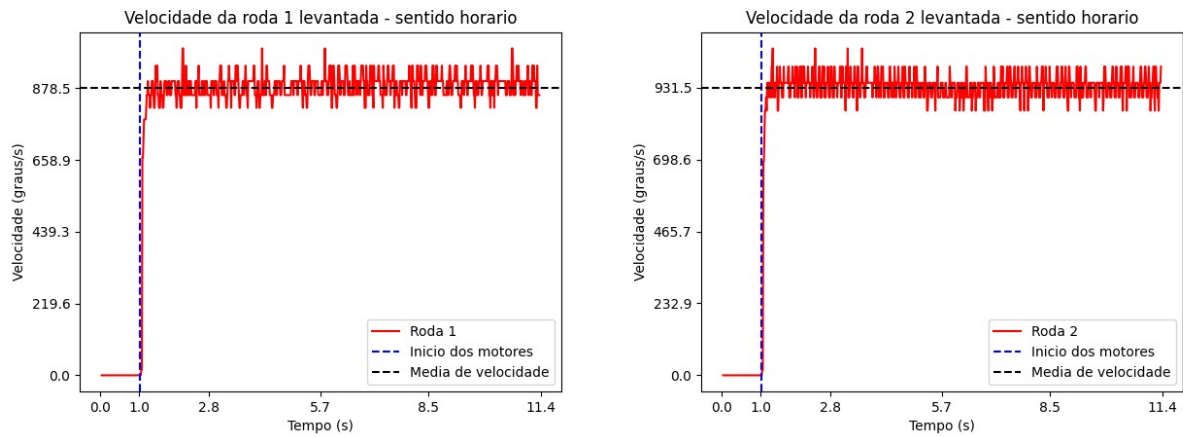


Figura 4.49 – Função degrau no sentido horário com rodas levantadas

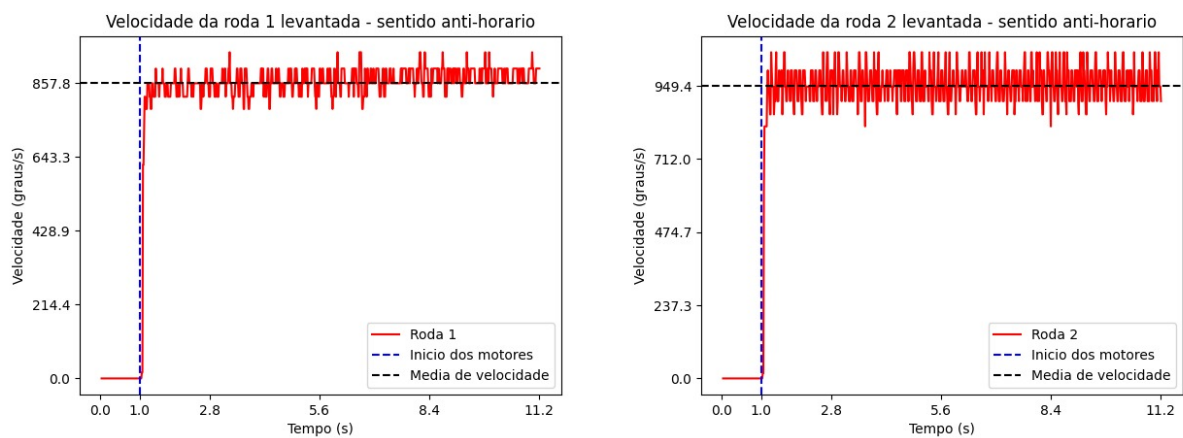


Figura 4.50 – Função degrau no sentido anti-horário com rodas levantadas

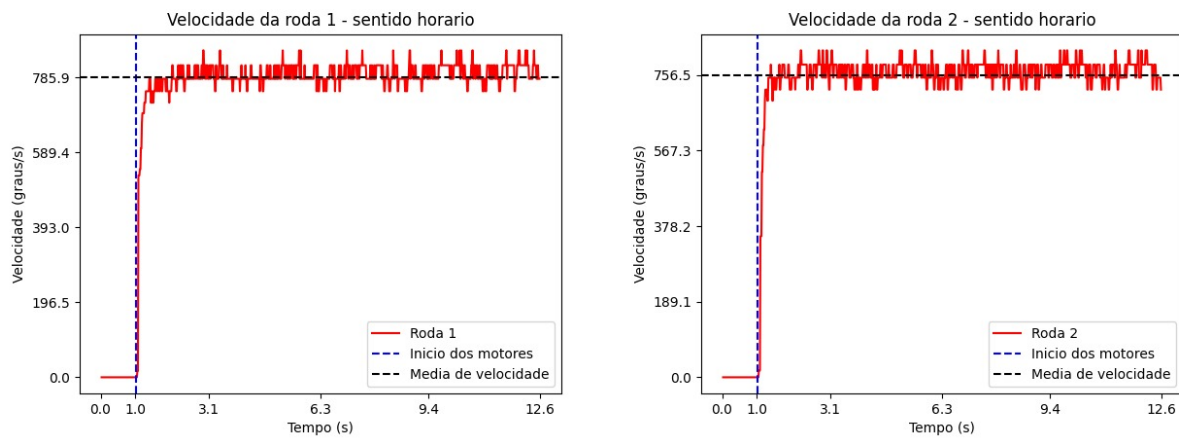


Figura 4.51 – Função degrau no sentido horário com rodas impulsionando

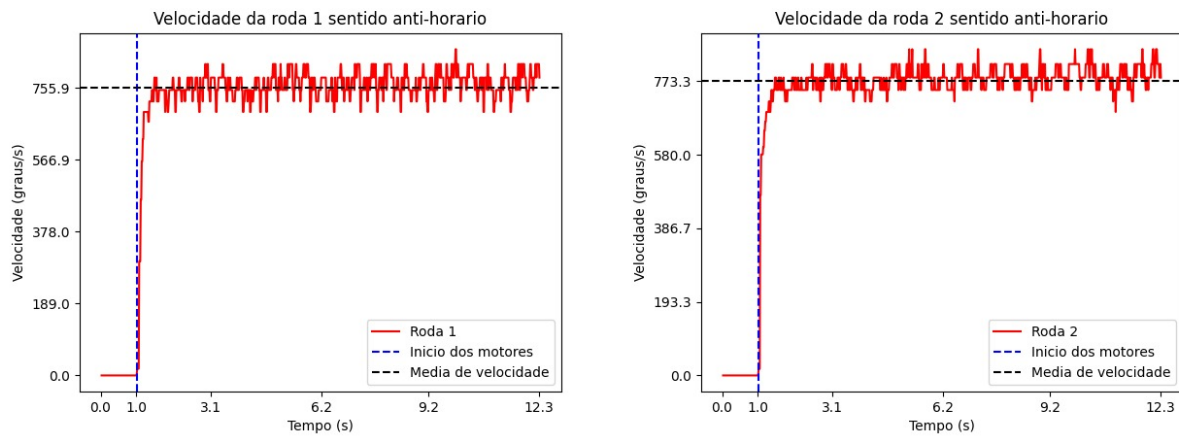


Figura 4.52 – Função degrau no sentido anti-horário com rodas impulsionando

Nos gráficos do teste pode ser visto que os motores não ficam em uma velocidade fixa, mas eles tem uma oscilação de velocidade em torno da média de cada teste, e pelos resultados compilados na tabela 4.3 é possível ver que os motores não possuem diferenças de atuação significativas com relação ao sentido de rotação. Outra informação importante obtida por estes testes é a diferença de velocidade de rotação entre os motores, já que os dados das rodas levantadas, ou seja, os dados de velocidade em que os motores trabalharam de forma independente mostram uma diferença significativa nas velocidades médias das rodas. A partir dos dados dos testes com os motores impulsionando a base móvel foi obtida o tempo de resposta dos motores, que demoram cerca de 200 ms para acelerar as rodas até a velocidade média.

Velocidade média Roda 1	Levantada	Impulsionando
Horário	878,5	785,9
Anti-horário	857,8	755,9
Velocidade média Roda 2	Levantada	Impulsionando
Horário	931,5	756,5
Anti-horário	949,4	773,3

Tabela 4.3 – Velocidade média das rodas

O teste com PWM progressivo é discutido a seguir. Pode ser visto na figura 4.53 que ambos os motores possuem valores de banda morta diferentes, mas os dois motores movem juntamente a base o que acaba resultando num único valor de banda morta como pode ser visto na figura 4.54. A partir destes testes foi obtido o valor de PWM de 400 (equivalente a 40%) como banda morta dos motores em conjunto, e este valor foi adicionado a saída do PID para compensar o efeito da banda morta.

Velocidade das rodas e PWM

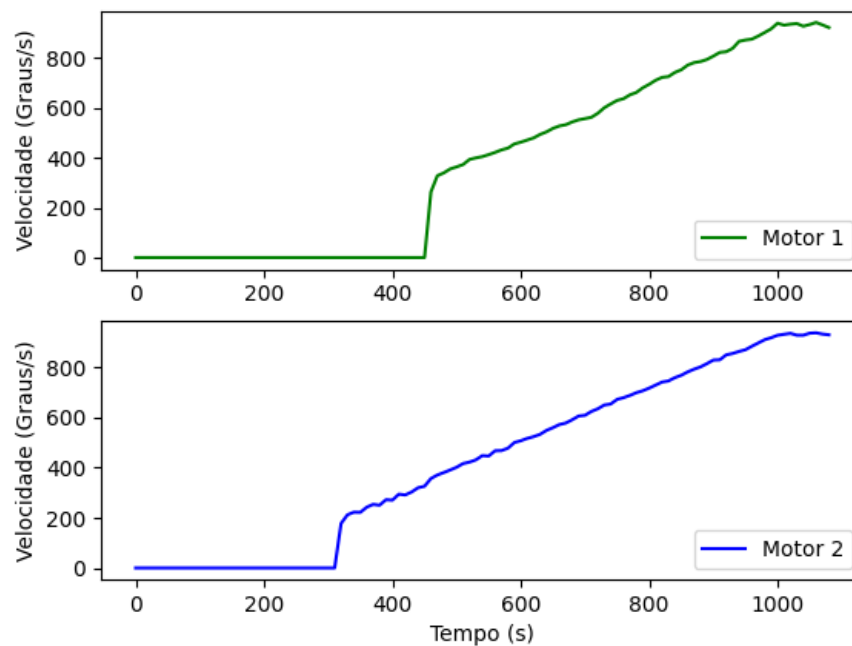


Figura 4.53 – Velocidade das rodas levantadas

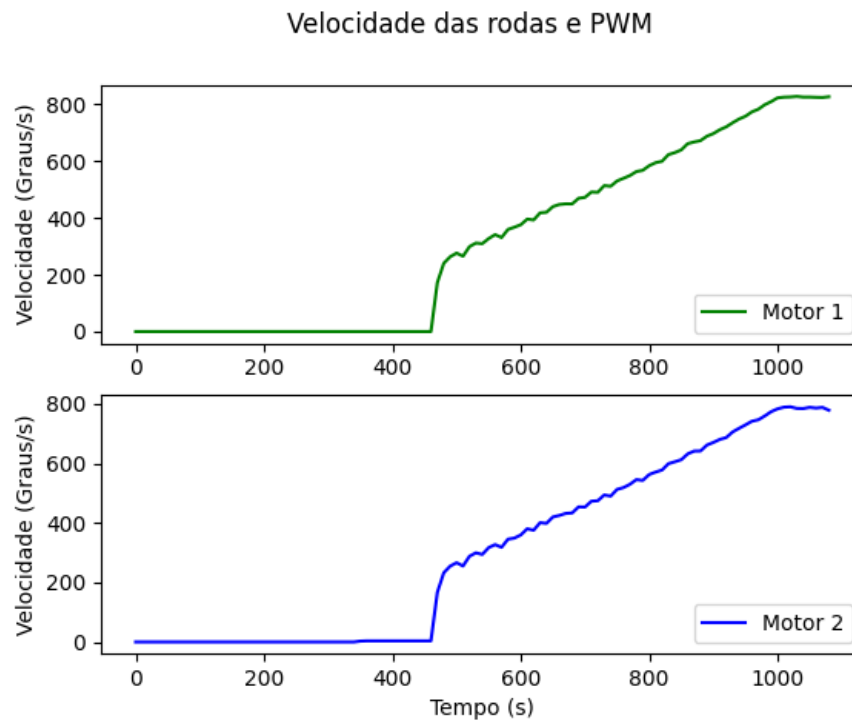


Figura 4.54 – Velocidade das rodas impulsionando a base móvel

4.2.3 Testes de derrapagem

Nas figuras com os testes invertendo os motores a velocidade é apresentada em módulo e as linhas pontilhadas indicam o momento em que houve uma inversão de sentido dos motores, além disso são marcados os pontos de velocidade mínima e máxima (triângulos verdes para baixo e para cima respectivamente) que serão usados para calcular parâmetros de avaliação de atrito. A mudança de sentido de rotação primeiro causa uma desaceleração na roda e depois uma aceleração no outro sentido, o que pode ser visto nos gráficos como uma súbita queda da velocidade após a linha pontilhada, seguida por um aumento em curto tempo.

O primeiro teste deste tipo foi feito com as rodas suspensas para observar a velocidade média e o tempo de resposta a inversão de sentido de rotação. Na figura 4.55 pode ser visto os testes com as rodas levantadas. A velocidade média das duas rodas ficaram acima de 900 graus por segundo e o tempo até a velocidade estabilizar na velocidade média é entorno de 100 ms para as duas rodas.

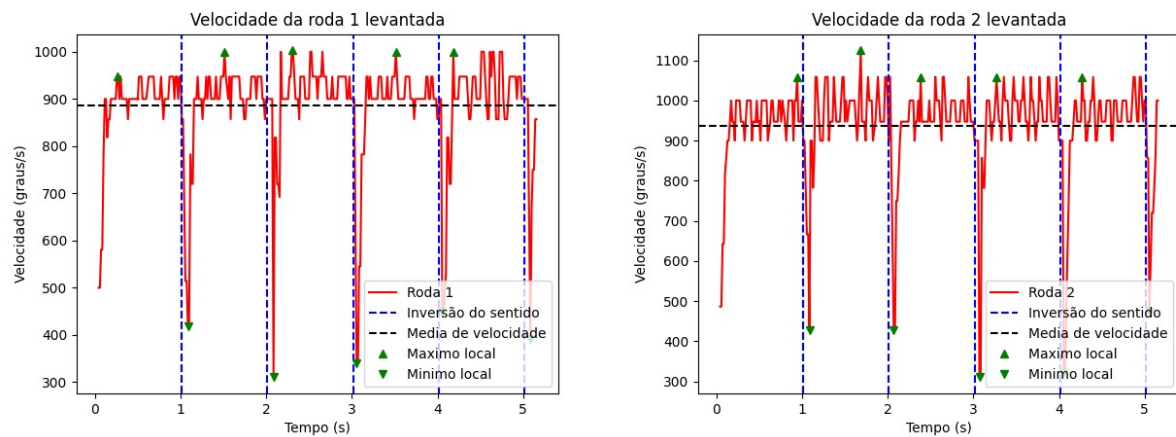


Figura 4.55 – Teste de inversão com rodas levantadas

O teste com os motores impulsionando a base móvel sem nenhuma alteração nas rodas resultou em uma grande diminuição da velocidade média (ambas as rodas ficaram entre 900 e 800 graus por segundo) e num grande aumento do tempo de estabilização (cerca de 250 ms) como pode ser visto na figura 4.56.

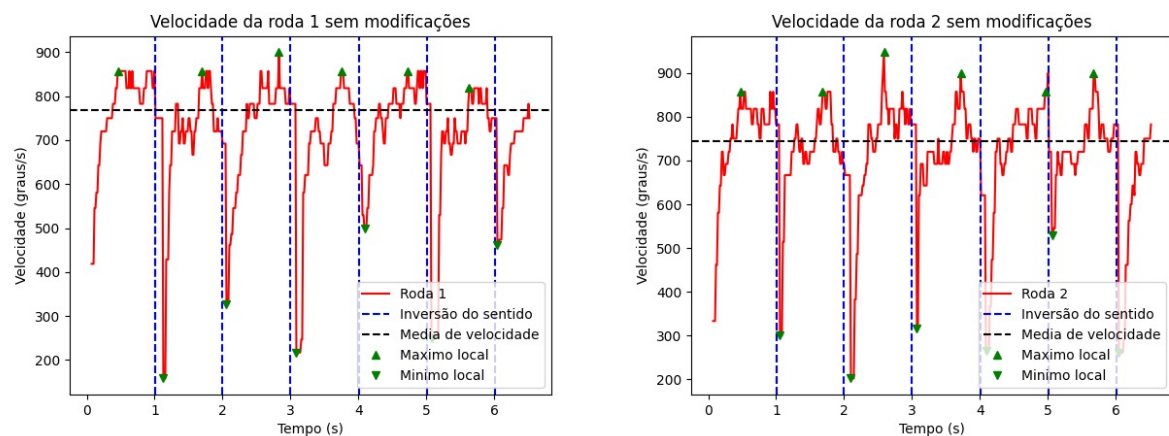


Figura 4.56 – Teste de inversão com rodas sem alteração impulsionando a base

Nos testes utilizando elásticos os resultados foram muito parecidos com os resultados da roda sem nenhuma modificação, o que reforça a não eficácia deste método na tentativa de eliminar a derrapagem das rodas. Os gráficos resultante deste teste pode ser visto na figura 4.57.

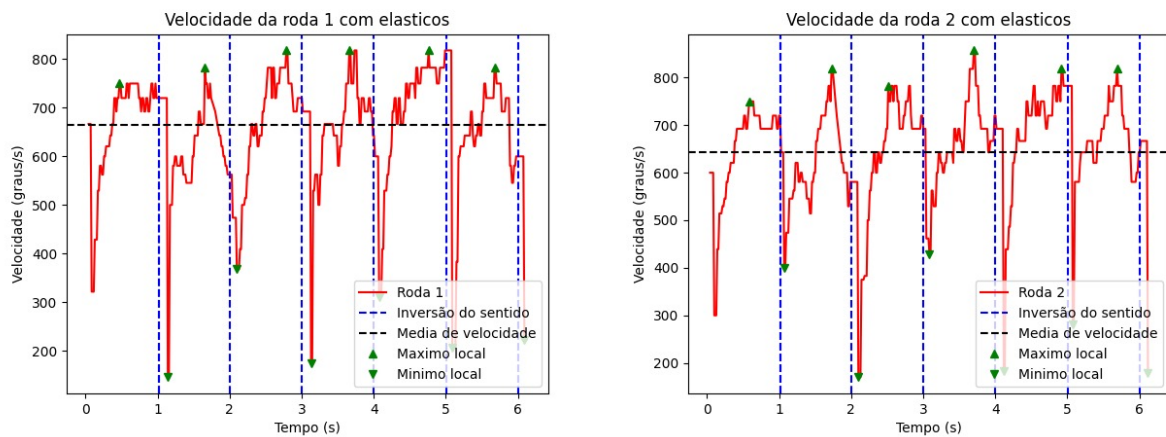


Figura 4.57 – Teste de inversão com rodas com elásticos impulsionando a base

Os testes com as rodas modificadas com a borracha com grampos que apresentou resultados mais diferentes dentre os testes. Na figura 4.58 pode ser visto que este tipo de abordagem gerou uma velocidade média menor e mais instável do que os outros testes, já que apresenta quedas significativas de velocidade entre 2 inversões de sentido.

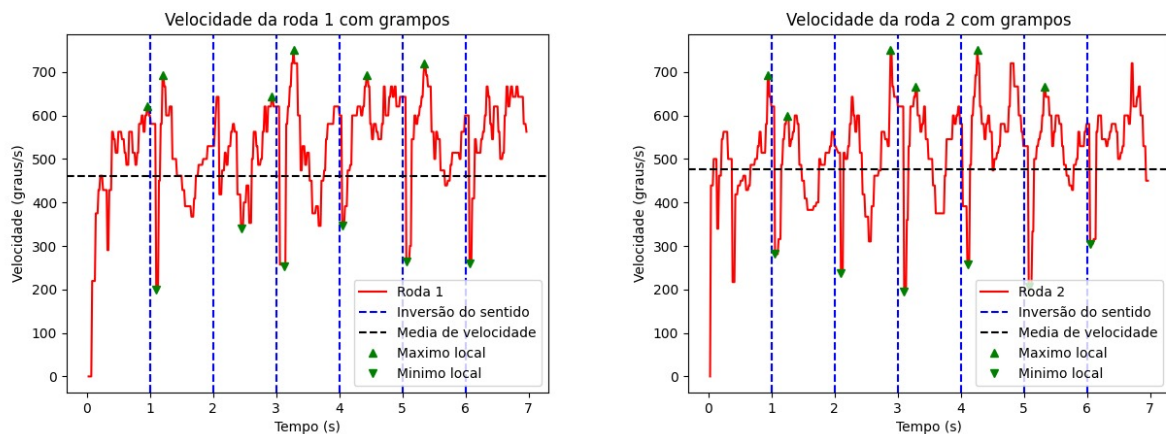


Figura 4.58 – Teste de inversão com rodas com grampos impulsionando a base

Na figura 4.59 temos os gráficos do teste apenas com a borracha sobre os pneus e na figura 4.60 temos o gráfico de velocidade das rodas com tiras de borracha sobre o papelão.

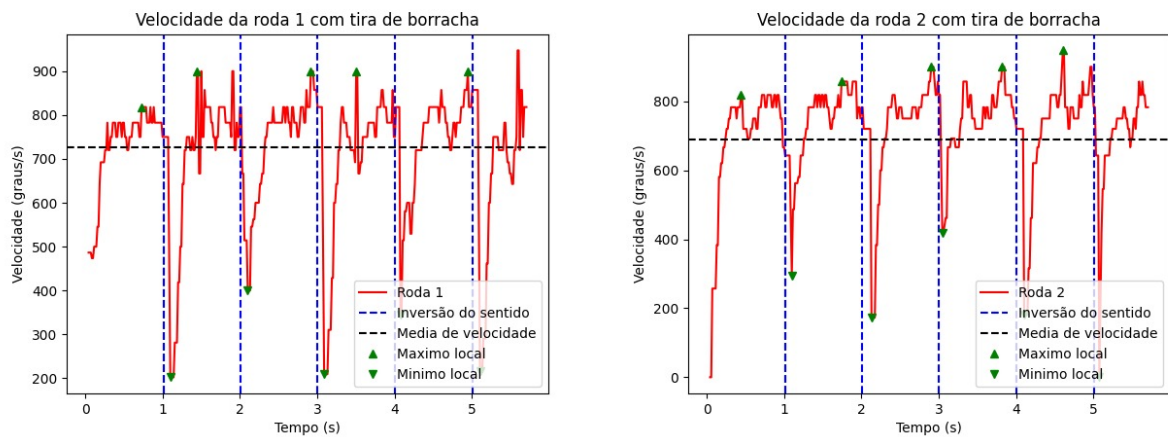


Figura 4.59 – Teste de inversão com rodas com borracha

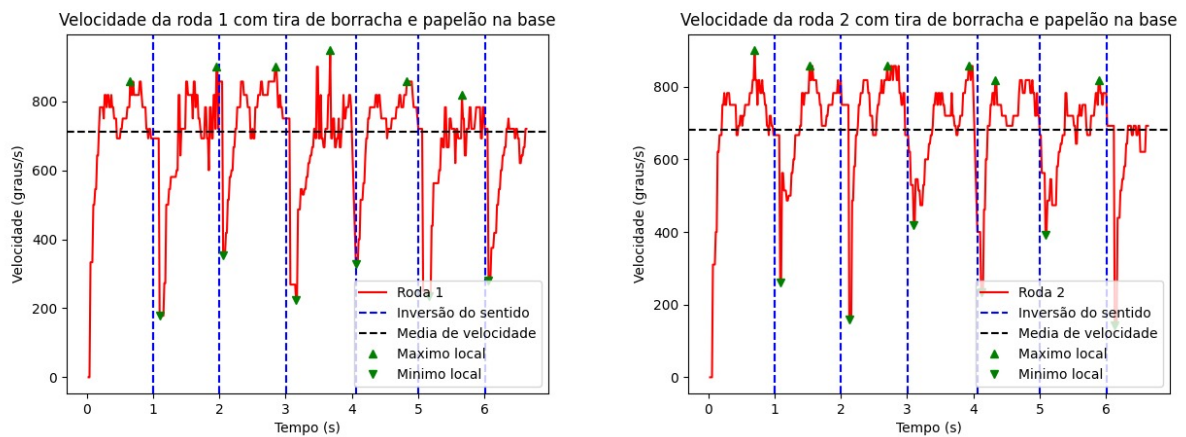


Figura 4.60 – Teste de inversão com rodas com borracha sobre papelão

O intuito destes testes era escolher o método que melhor minimizasse a derrapagem ao aumentar o atrito, e para escolher entre os métodos foi utilizada uma abordagem numérica baseada nos dados obtidos dos testes. Para isso foram utilizados o tempo entre a inversão e o ponto de mínimo e entre a inversão e o ponto de máximo como parâmetros de verificação de menor derrapagem, considerando que quanto maiores estes dois parâmetros maior o atrito nas rodas. A tabela 4.4 traz os tempos médios de cada teste para as duas rodas, sendo que o tempo entre a inversão e o ponto de menor velocidade é chamado de tempo de descida e o tempo entre a inversão e o ponto máximo é chamada tempo de subida.

Pelos resultados apresentados na tabela 4.4 pode ser visto que o método que mais aumentou o tempo de subida, na soma dos valores das duas rodas, foi a borracha sobre a superfície de papelão.

Tipo de teste	Roda levantada		Roda sem alterações		Elásticos	
Numero da roda	Roda 1	Roda 2	Roda 1	Roda 2	Roda 1	Roda 2
Tempo de subida	343 ms	500 ms	677 ms	679 ms	650 ms	686 ms
Tempo de descida	60 ms	54 ms	76 ms	67 ms	100 ms	90 ms
Tipo de teste	Grampo		Borracha		Borracha + papelão	
Numero da roda	Roda 1	Roda 2	Roda 1	Roda 2	Roda 1	Roda 2
Tempo de subida	522 ms	488 ms	703 ms	698 ms	760 ms	670 ms
Tempo de descida	183 ms	79 ms	88 ms	86 ms	88 ms	98 ms

Tabela 4.4 – Tempos médios de subida e descida dos testes

4.2.4 Testes de estabilização

Nas figuras 4.61 a 4.70 são apresentados os gráficos resultantes dos testes de estabilização da barra com subida manual. O caso ideal destes testes seria que a barra ficasse em torno de 180° e que o PID ajustasse os valores abaixo das linhas de saturação (linhas tracejadas verdes em 1000 e -1000; essa faixa em que não há saturação será chamada de faixa de saturação na continuação deste documento) e que a velocidade ficasse oscilando em valores baixos até estabilizar em 0 novamente. Esse caso produziria um gráfico em que a posição da barra oscilaria um pouco em torno de 180° , a velocidade da barra em torno de 0 e o PID oscilaria entre valores positivos e negativos, que decresceriam em módulo com o tempo, dentro da faixa de saturação. Os ajustes de K_p no início do teste podem ser vistos nas figura 4.61 a 4.63.

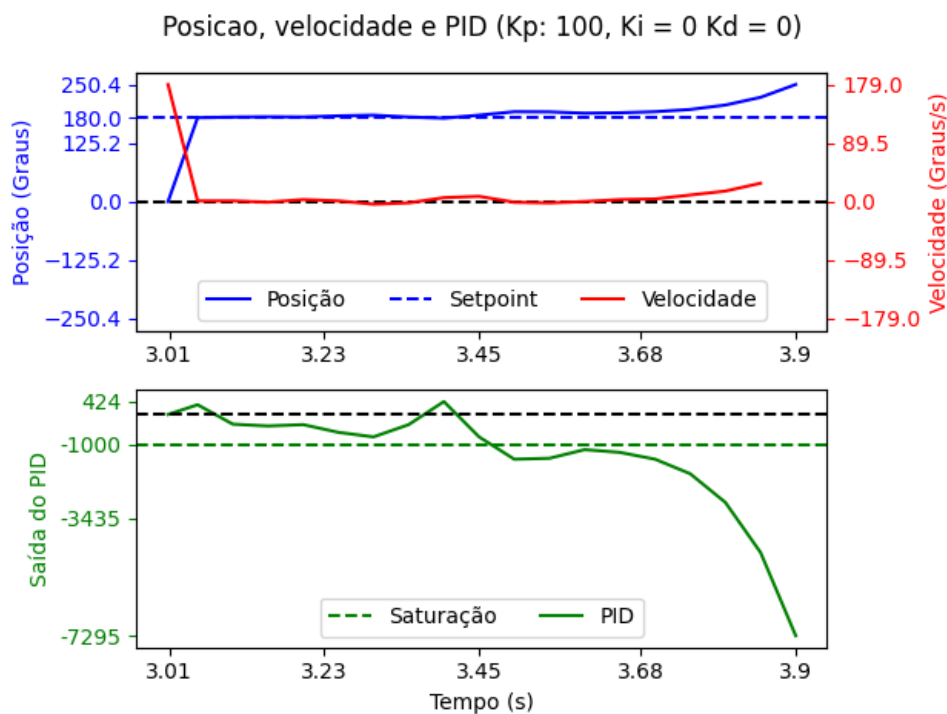


Figura 4.61 – Controle com subida manual com $K_p = 100, K_i = 0, K_d = 0$.

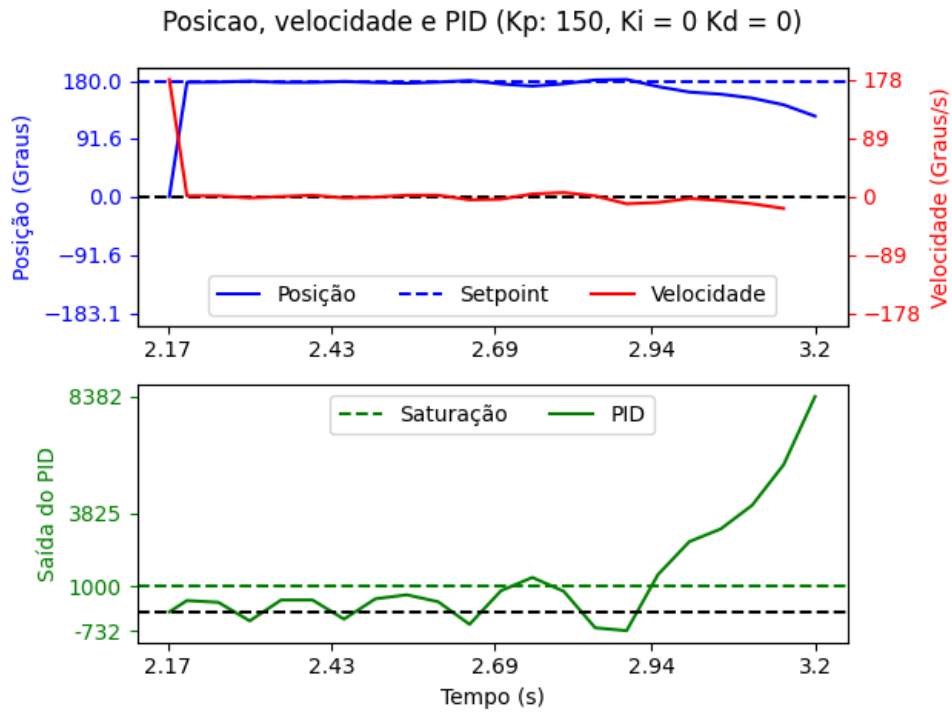


Figura 4.62 – Controle com subida manual com $K_p = 150, K_i = 0, K_d = 0$.

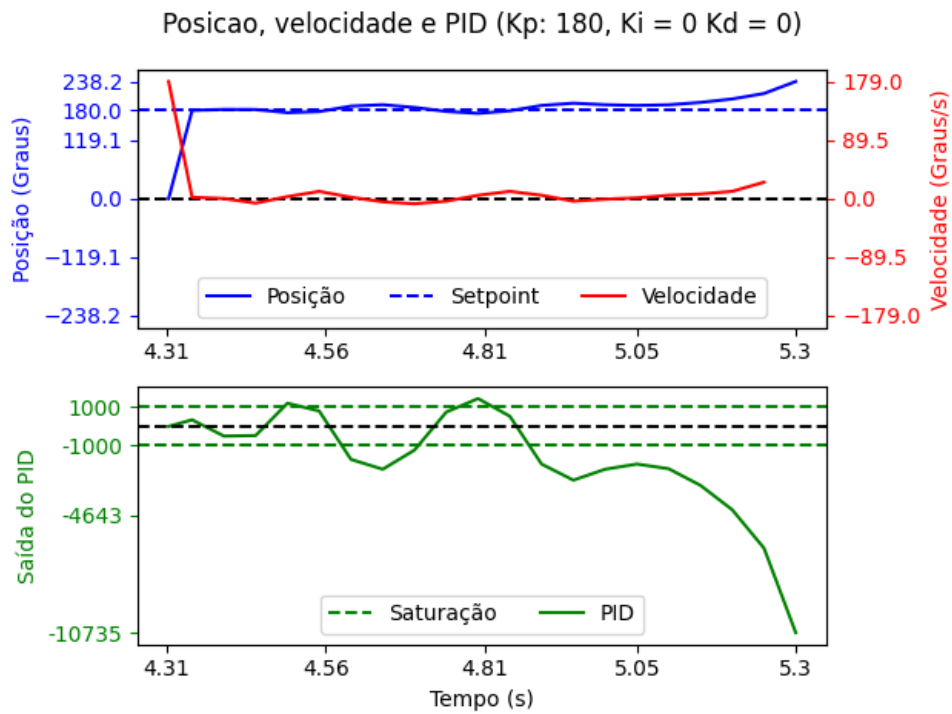


Figura 4.63 – Controle com subida manual com $K_p = 180, K_i = 0, K_d = 0$.

Com $K_p = 100$ o PID resultou em pouca saturação, mas também em poucas inversões de sentido, já o $K_p = 150$ produziu mais oscilações dentro da faixa de saturação, enquanto o $K_p = 180$ produziu oscilações que saíram muito da faixa de saturação. Da figura 4.64 a 4.67 são apresentados os gráficos de ajuste do parâmetro K_i com K_p fixo em 150.

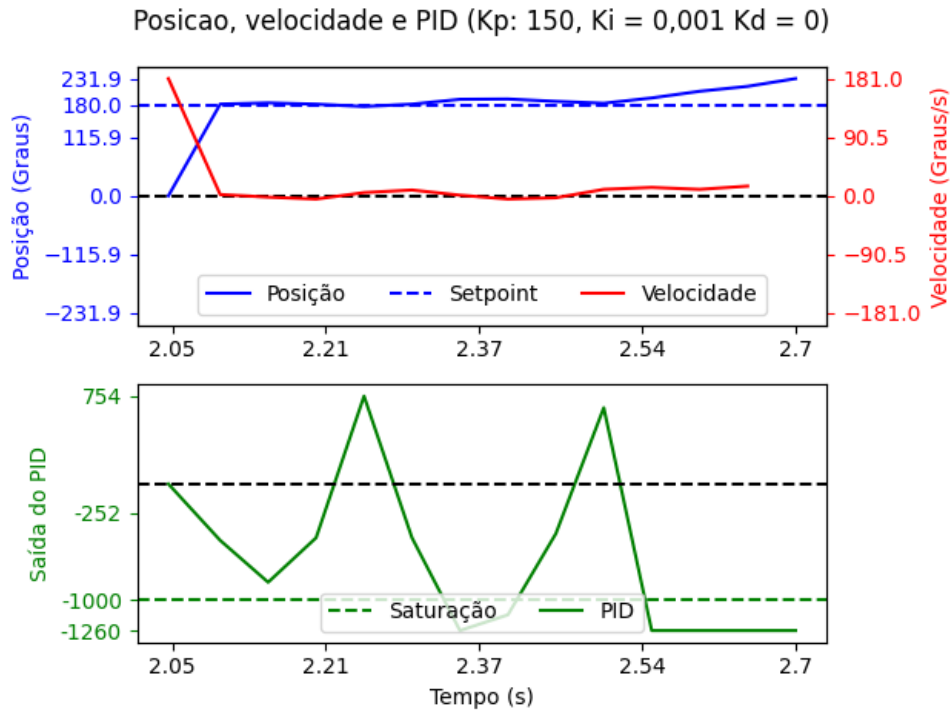


Figura 4.64 – Controle com subida manual com $K_p = 150$, $K_i = 0,001$, $K_d = 0$.

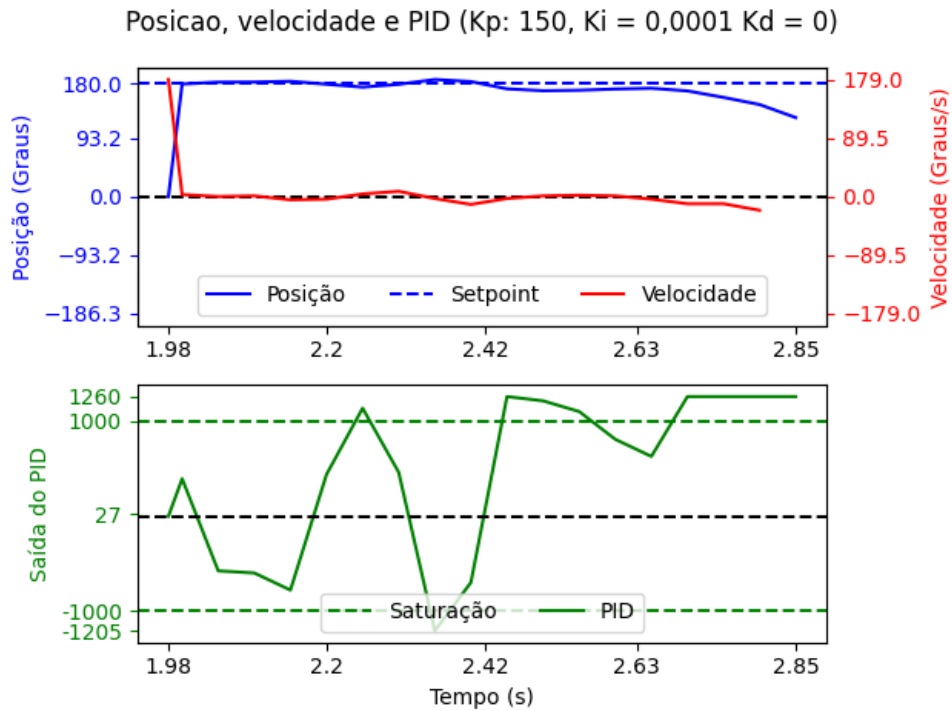


Figura 4.65 – Controle com subida manual com $K_p = 150, K_i = 0,0001, K_d = 0$.

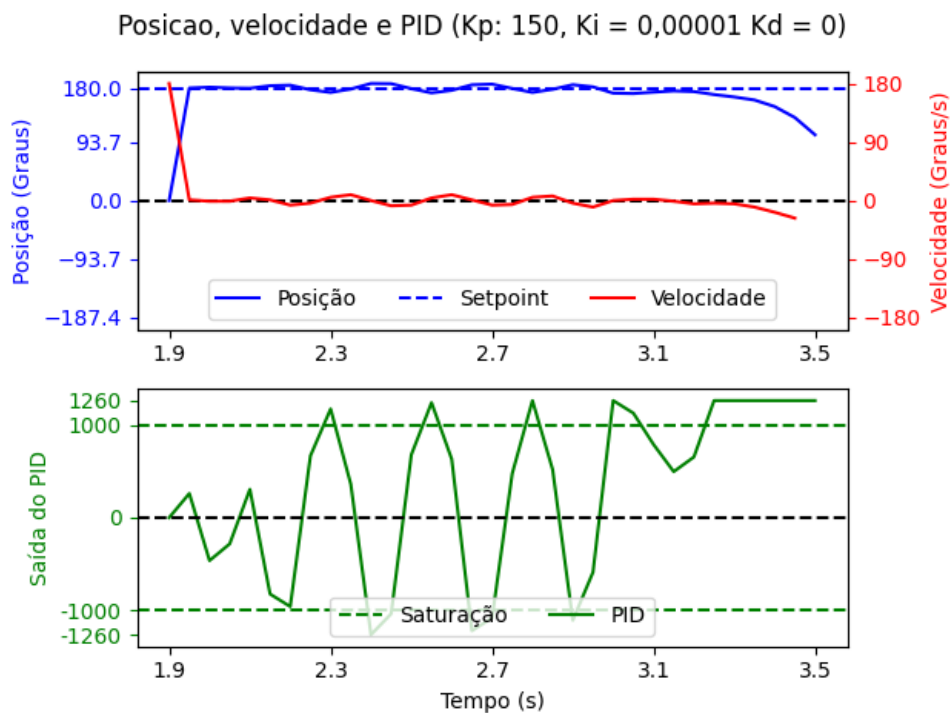


Figura 4.66 – Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 0$.

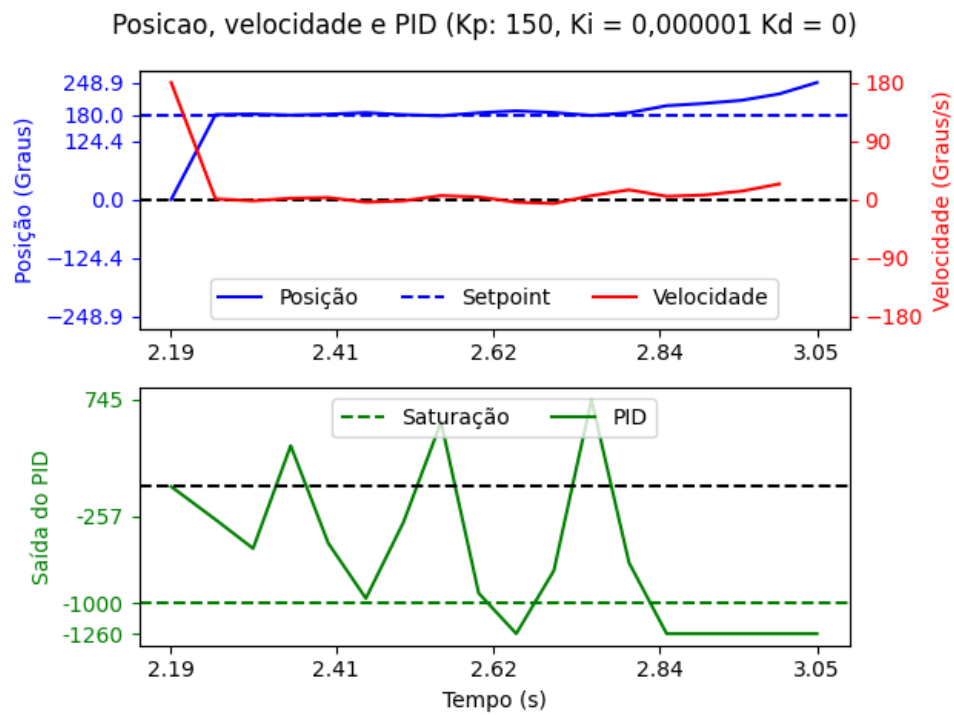


Figura 4.67 – Controle com subida manual com $K_p = 150, K_i = 0,000001, K_d = 0$.

Dos testes de ajuste de K_i o que se comportou melhor dentre todos testados foi $K_i = 0,00001$, que produziu várias oscilações que escapavam pouco da faixa de saturação. Os gráficos com ajuste do parametro K_d são mostrados nas figuras 4.68 a 4.70 a seguir.

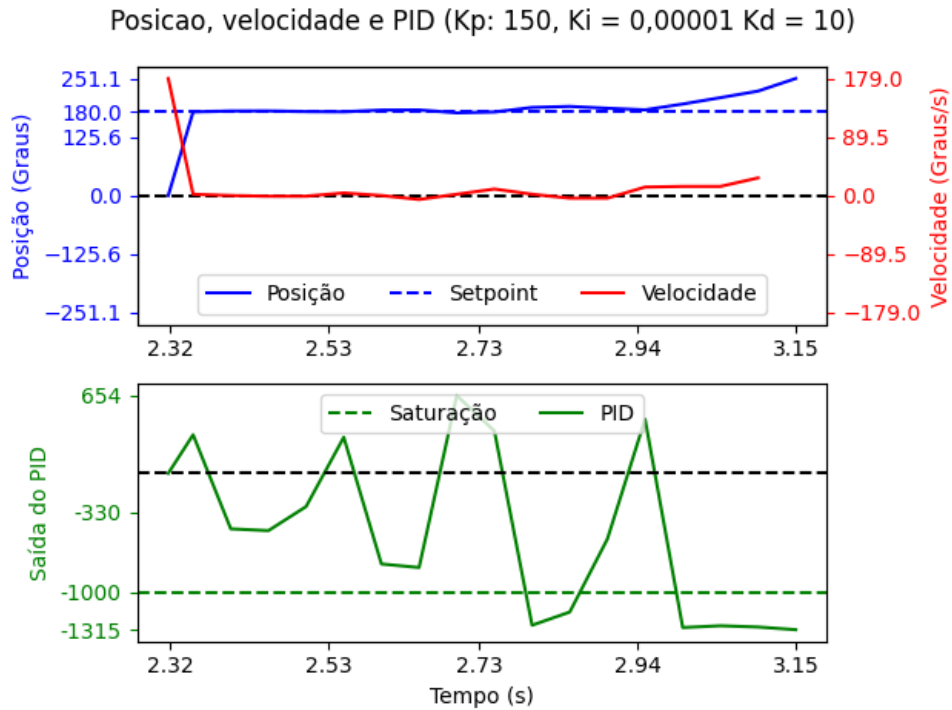


Figura 4.68 – Controle com subida manual com $K_p = 150$, $K_i = 0,00001$, $K_d = 10$.

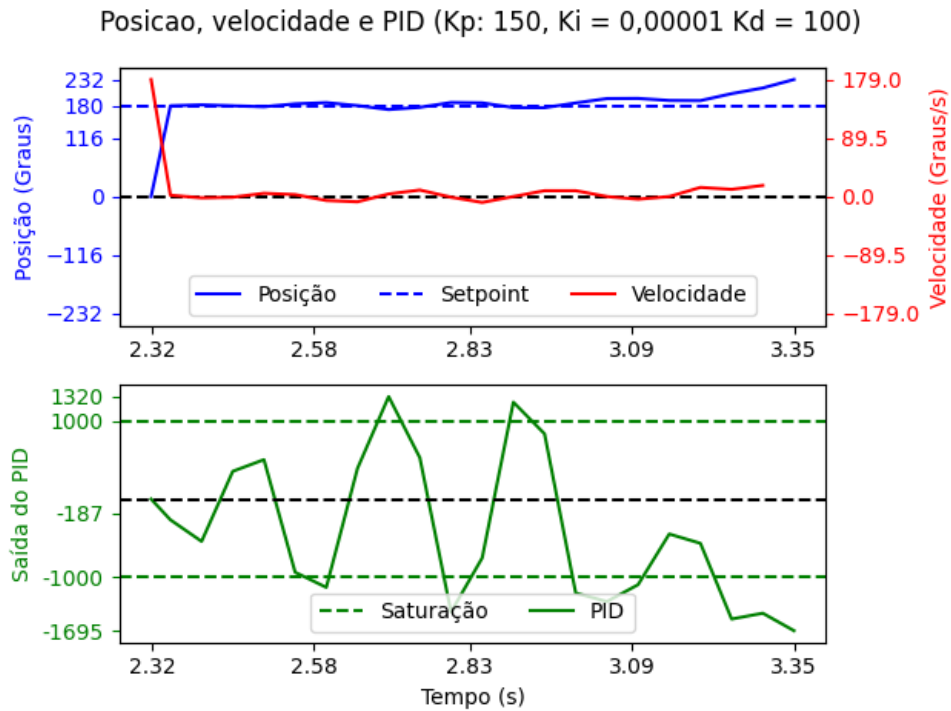


Figura 4.69 – Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 100$.

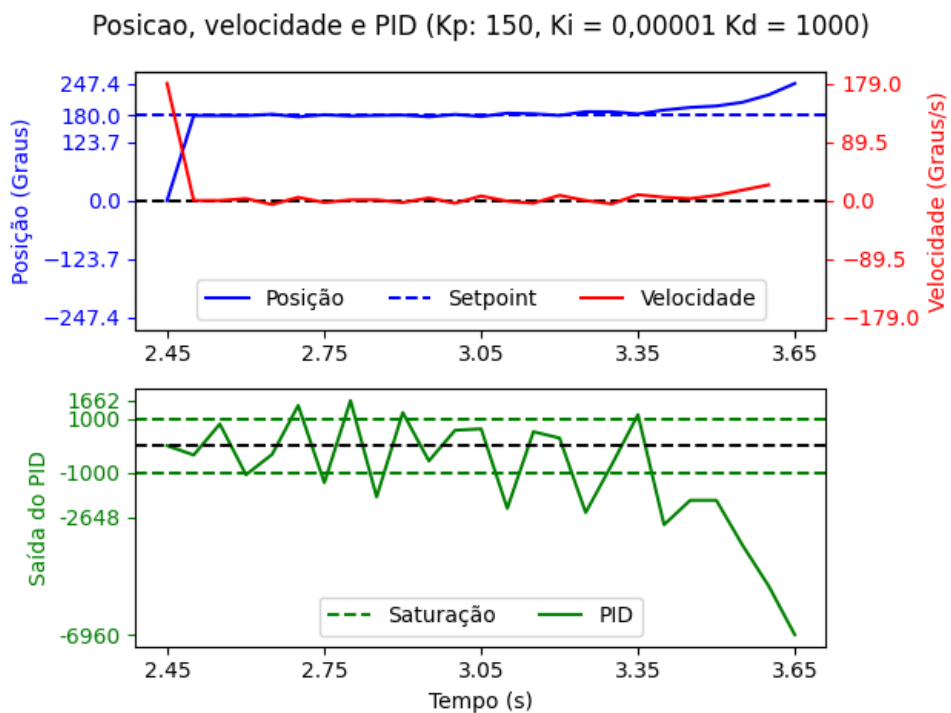


Figura 4.70 – Controle com subida manual com $K_p = 150, K_i = 0,00001, K_d = 1000$.

Para o K_d o melhor resultado foi obtido com $K_d = 1000$, e portanto o melhor PID encontrado nestes testes foi o PID $K_p = 150, K_i = 0,00001, K_d = 1000$, cujo gráfico pode ser visto na figura 4.70.

OS gráficos do teste de estabilização com subida de barra com PID podem ser vistos nas figuras de 4.71 a 4.81, onde a linha tracejada azul indica o momento de troca entre o PID de subida e o PID de estabilização testado.

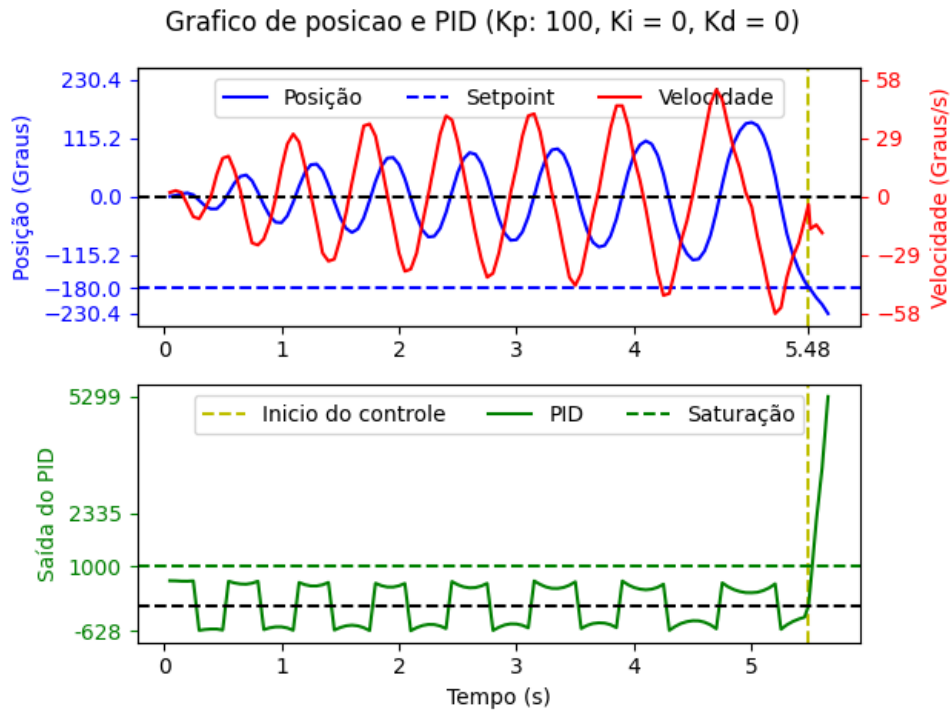


Figura 4.71 – Controle com subida por PID com $K_p = 100, K_i = 0, K_d = 0$.

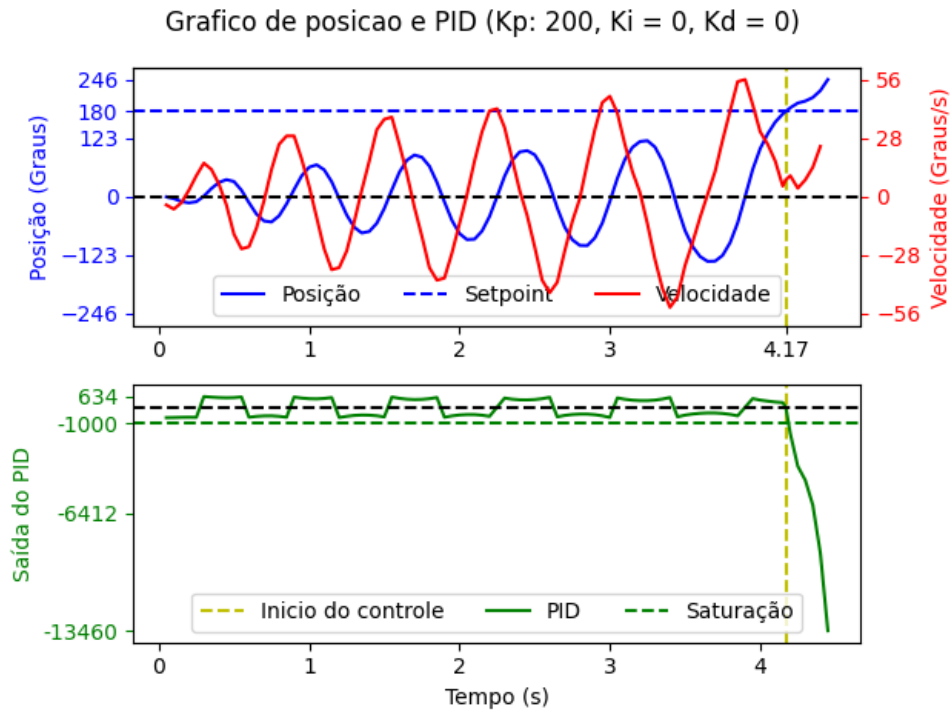


Figura 4.72 – Controle com subida por PID com $K_p = 200, K_i = 0, K_d = 0$.

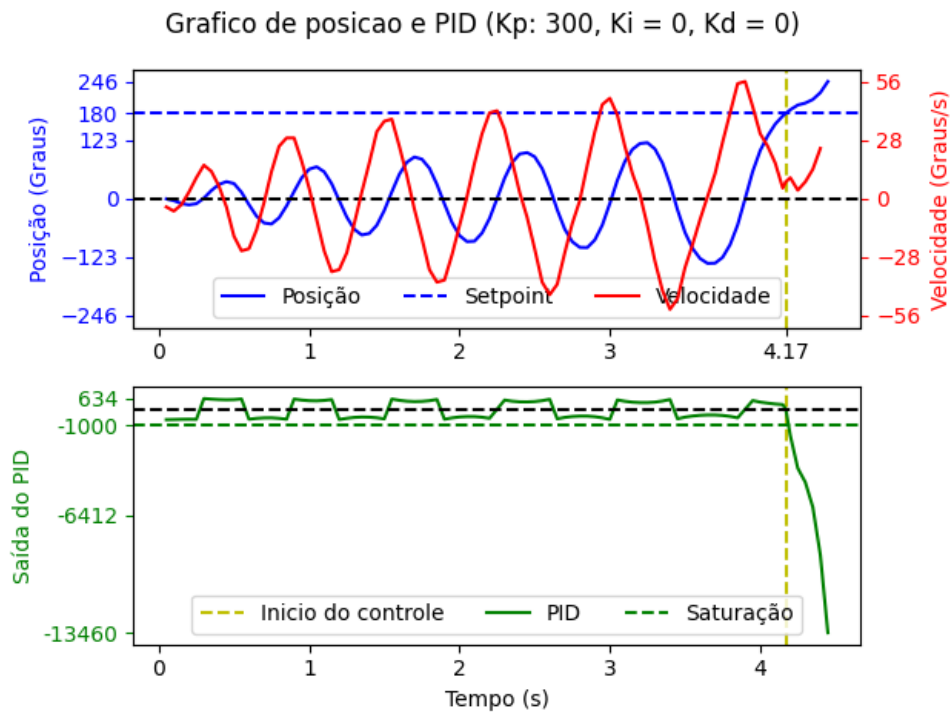


Figura 4.73 – Controle com subida por PID com $K_p = 300, K_i = 0, K_d = 0$.

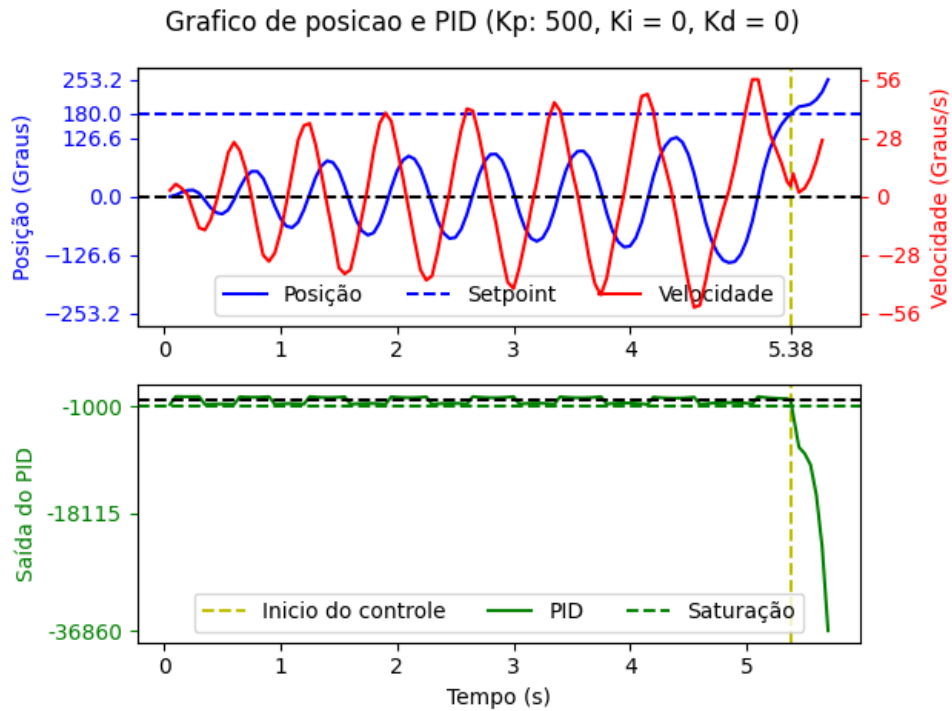


Figura 4.74 – Controle com subida por PID com $K_p = 500, K_i = 0, K_d = 0$.

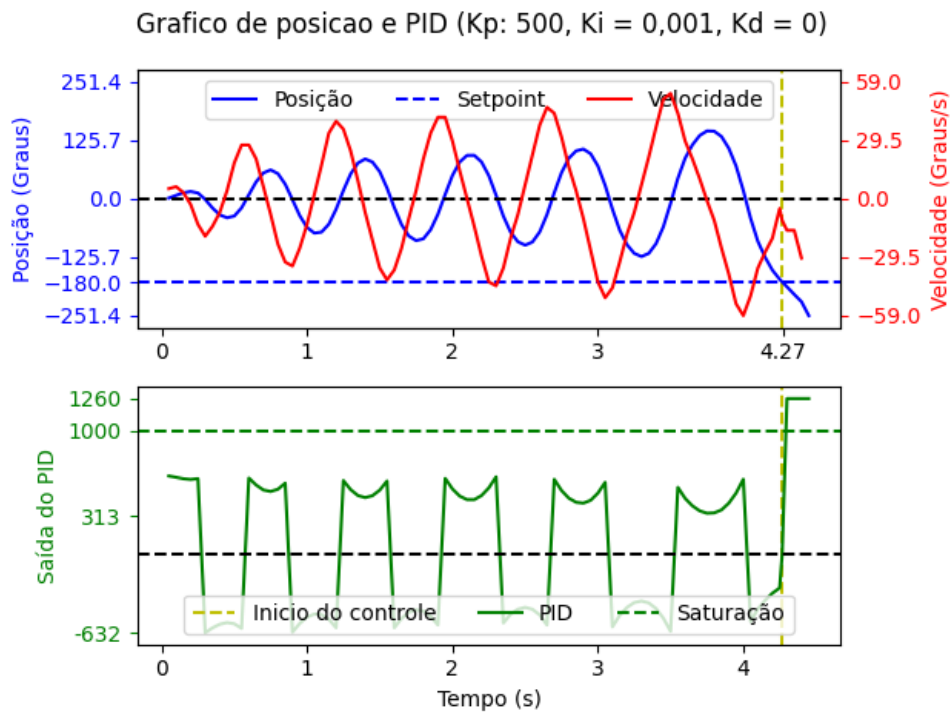


Figura 4.75 – Controle com subida por PID com $K_p = 500, K_i = 0,001, K_d = 0$.

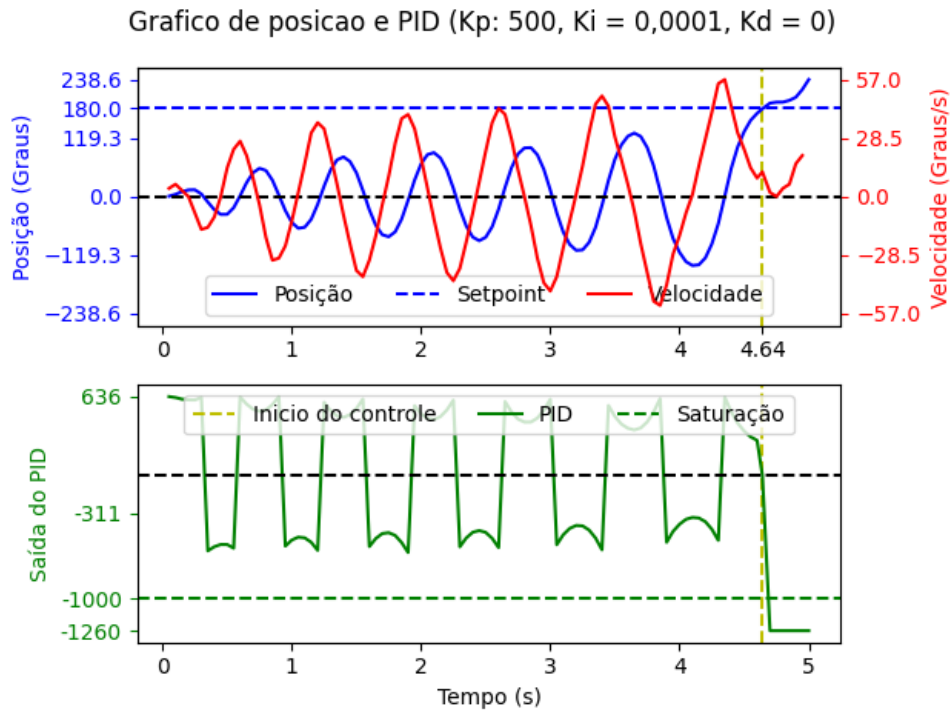


Figura 4.76 – Controle com subida por PID com $K_p = 500, K_i = 0,0001, K_d = 0$.

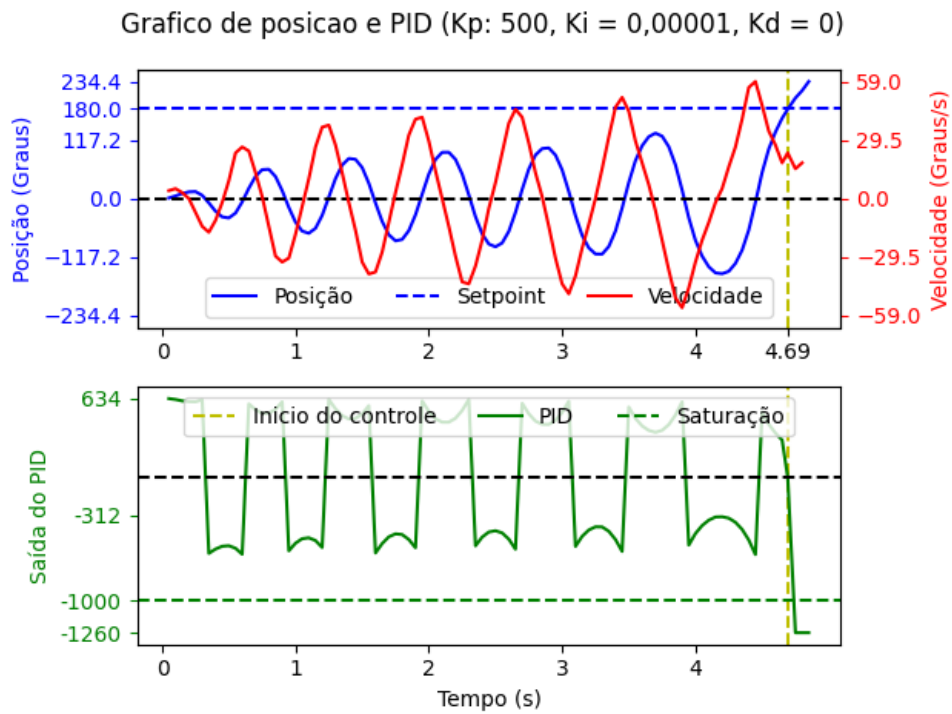


Figura 4.77 – Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 0$.

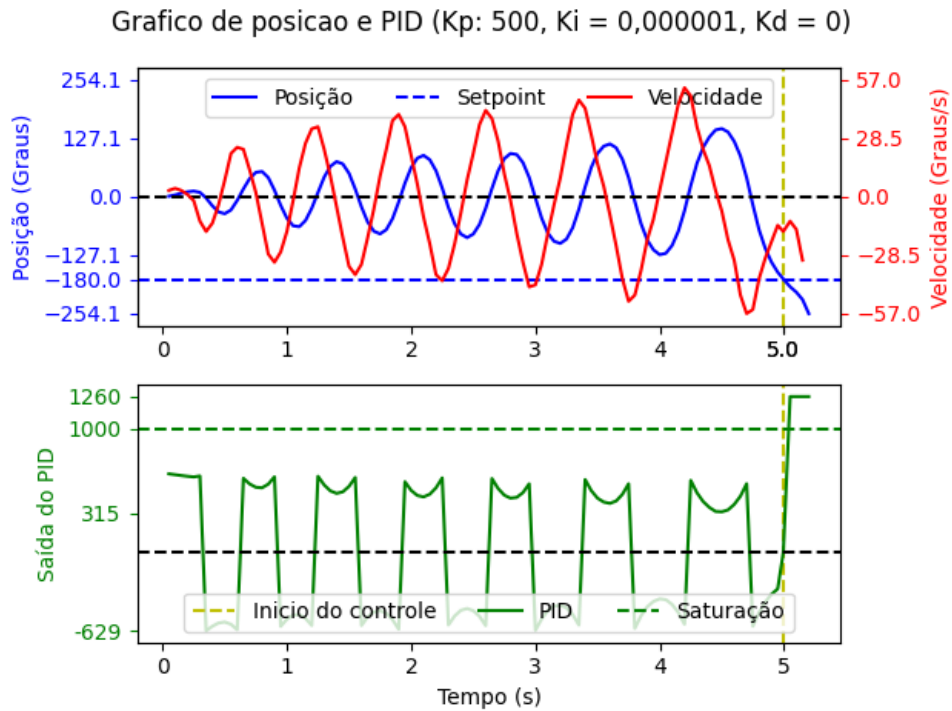


Figura 4.78 – Controle com subida por PID com $K_p = 500, K_i = 0,000001, K_d = 0$.

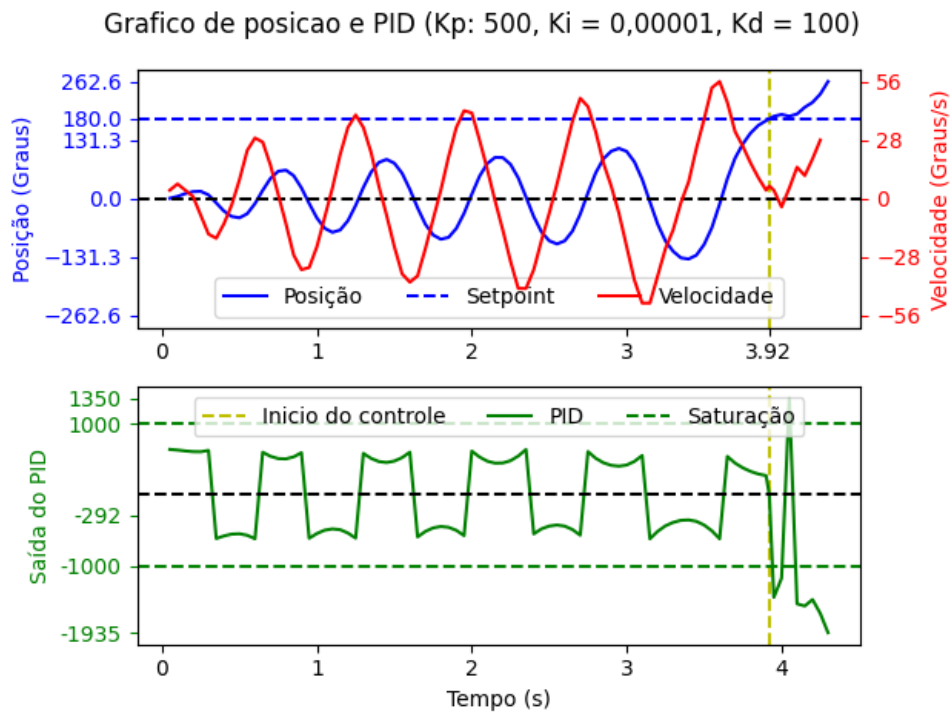


Figura 4.79 – Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 100$.

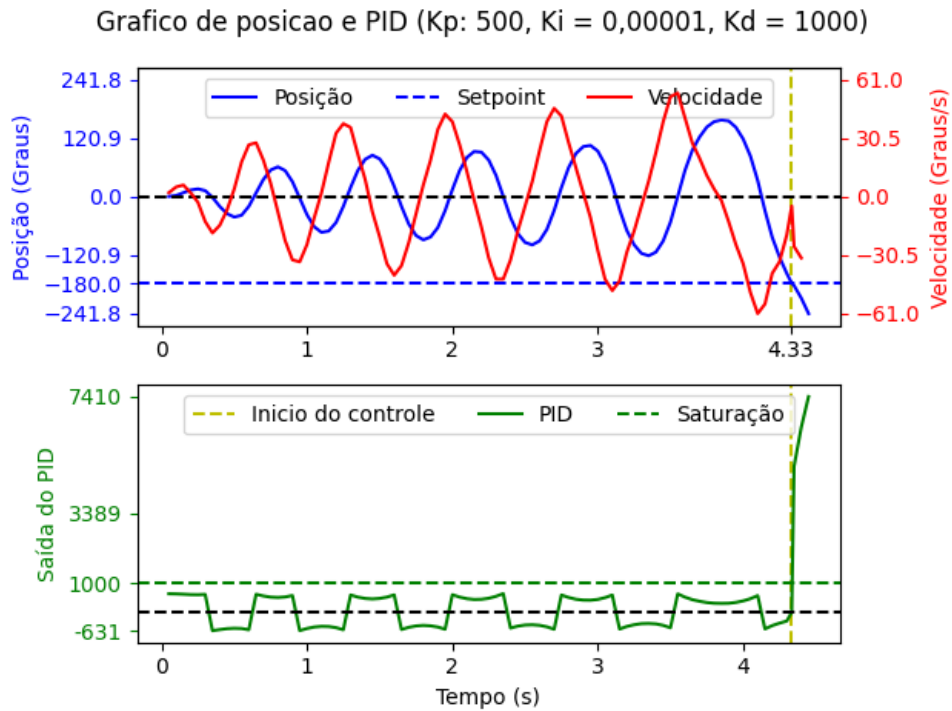


Figura 4.80 – Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 1000$.

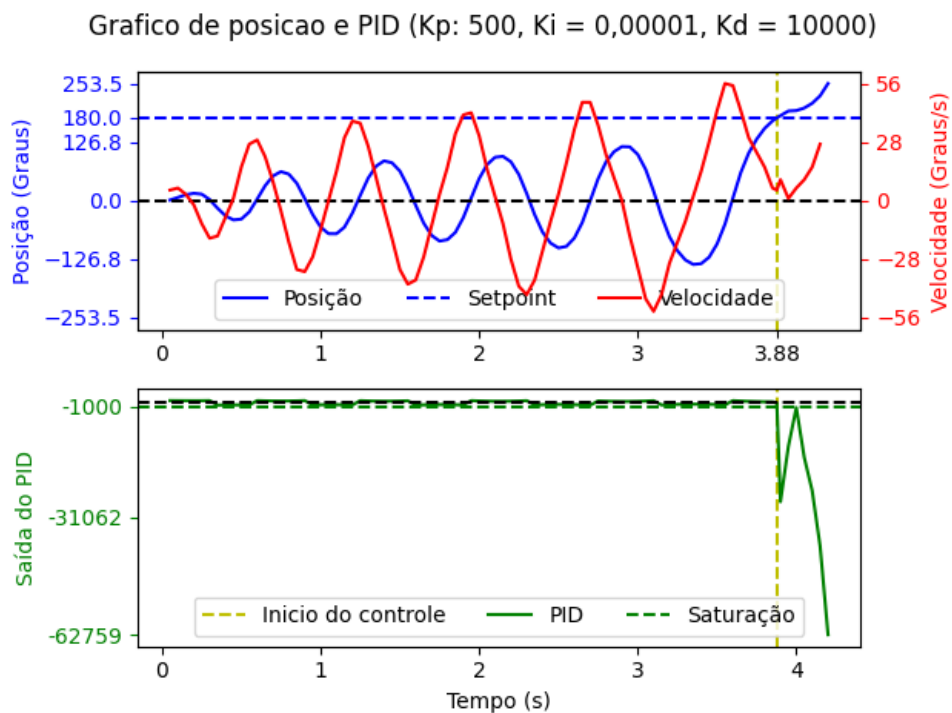


Figura 4.81 – Controle com subida por PID com $K_p = 500, K_i = 0,00001, K_d = 10000$.

Nos gráficos dos testes para ajuste é possível ver que a velocidade inicial da barra após o algoritmo de subida não é constante nem zero, e por conta disto o parâmetro K_p teve de ser ajustado com um valor bem maior do que o valor com teste com subida manual, para assim tentar compensar a velocidade angular da barra quando o PID de estabilização assume o controle. Um efeito desse aumento de K_p pode ser visto nas curvas de saída do PID, que após a mudança do PID de subida pro PID de controle, apresentam uma rápida saturação. O melhor resultado de PID para estes testes pode ser visto na figura 4.79.

4.3 Mecânica

Os resultados do projeto e elaboração da mecânica resultaram numa estrutura que cumpre bem as funções de movimento e sustentação propostas sem gerar muitas oscilações indesejadas que poderiam prejudicar a eletrônica e principalmente o controle. A base móvel implementada, que serve como braço de rotação do pêndulo e sustentação da eletrônica, possui um bom espaço para fixar os componentes eletrônicos atuais, além de ter pouco peso o que facilita a movimentação por parte dos motores. O tubo de sustentação projetado sustenta e permite que a base móvel rotacione ao mesmo tempo em que os cabos de alimentação passam através delas pelo *slip ring*.

Um problema mecânico encontrado durante os testes foi o baixo atrito entre a roda e a superfície da base móvel, e embora novos métodos tenham sido testados e utilizados para minimizar o problema, o pouco peso da base móvel e o atrito entre a superfície das rodas e dos materiais utilizados no teste não foram suficientes para eliminar o problema da derrapagem. Outro problema encontrado foi a folga da caixa de redução dos motores, que possuem alguns graus de folga quando o motor começa a inverter o sentido de rotação. Essa folga é especialmente prejudicial durante o controle, pois enquanto a barra está em torno do ponto de equilíbrio o motor inverte o sentido várias vezes respondendo ao PID que tende a oscilar para manter a barra parada neste ponto.

4.4 Eletrônica

A eletrônica construída permitiu monitorar e controlar o pêndulo, além de também permitir a troca de informações entre a MSP e a Raspberry durante a execução dos testes. Embora os componentes escolhidos e montados no pêndulo tenham implementado as funções principais com sucesso o projeto apresenta alguns problemas e algumas características que podem vir a ser um problema em versões futuras do projeto. A primeira característica que vale ressaltar é a precisão do *encoder* que mede o ângulo da barra, que consegue medir com uma precisão de $0,15^\circ$, que pode ser insuficiente ou indesejado a depender das alterações realizadas. A precisão também é um problema nos *encoders* das rodas, que tem uma precisão muito baixa (sua menor escala mede 18°) e dificultou a medição de velocidade angular das

rodas prejudicando a geração dos gráficos e atrapalhando na análise do derrapamento. Para análise de derrapamento das rodas, e também para poder obter uma correlação numérica entre a potência demandada dos motores e a real aceleração que a base móvel experimenta, seria necessário colocar um sensor de posição ou velocidade angular no eixo de rotação da base móvel, o que não foi incluído nesta versão do pêndulo.

Também foi medido uma pequena queda de tensão de cerca de 200 mV no *slip ring* quando a corrente demandada pela eletrônica subia (quando os motores estavam ativos e invertendo os sentidos de rotação frequentemente), o que não causa uma falha dos sistemas da eletrônica, mas se a queda de tensão aumentar pode fazer a Raspberry ler caracteres aleatórios na UART, inviabilizando a comunicação. Na comunicação UART também foi observado que as vezes alguns valores estavam errados, principalmente quando os motores estavam ligados e operando, o que pode ser indício de um mal contato do cabo ocasionado pela vibração.

Um dos problemas mais graves da eletrônica é a baixa potência dos motores. Nos testes de controle os motores não conseguem fazer a velocidade angular da barra inverter o sentido depois de um certo ângulo, ou seja, depois que a barra já passou de um certo ângulo os motores não tem potência para deslocar a base rápido o suficiente para voltar a barra ao *setpoint*. Outro efeito da potência percebido nos testes foi a incapacidade de compensar velocidades angulares mais altas resultantes de algum método de subida, que pode ser visto na baixa duração de tempo da estabilização da barra quando a velocidade angular depois da subida não está próxima a 0.

4.5 Controle

O controlador PID digital implementado na MSP funciona bem e com uma frequência alta, mas apesar disso a planta implementada apresenta alguns desafios de controle. O primeiro deles é o derrapamento das rodas, que acabam por colocar um elemento de banda morta, já que nem toda a energia mecânica do motor é transferida imediatamente para a base móvel, criando assim um atraso entre a aplicação do sinal de PWM e seu efeito na movimentação do pêndulo. Outro efeito de banda morta visualizado nos testes é causado pelas folgas da caixa de redução dos motores utilizados. Esta folga tem amplitude significativa e é especialmente prejudicial quando os motores invertem o sentido de rotação, portanto esse fenômeno atrapalha bastante quando a barra está perto do ponto de equilíbrio e a saída do PID inverte o sentido de rotação dos motores várias vezes. Os motores também apresentam baixa potência, e mesmo tendo 2 a potência obtida não é o suficiente para responder ao PID e compensar os movimentos da barra, principalmente quando a barra já está em um ângulo relativamente afastado do ponto de equilíbrio ou quando a velocidade angular da barra é alta.

4.6 Software

Os testes automáticos implementados possibilitaram a produção de uma base de dados de vários testes que foram úteis para o desenvolvimento e aprimoramento dos algoritmos de controle, principalmente durante o desenvolvimento do algoritmo de subida. Os testes automáticos podem ser utilizados para realizar múltiplos testes e assim validar ou uma estratégia de controle, podendo também serem adaptados em versões futuras para serem utilizados como método de sintonia de PID que utilizem abordagens de testes iterativos.

O algoritmo de plotagem permitiu a visualização dos dados obtidos nos experimentos com o pêndulo, e estes gráficos foram utilizados tanto na implementação e melhoria da estratégia de controle quanto no próprio desenvolvimento do pêndulo (foi utilizado um controle PID padrão para verificar o funcionamento de todas as variáveis e componentes do pêndulo). Exemplos destes gráficos podem ser vistos na seção de análise de resultados de controle.

Os códigos desenvolvidos conseguiram implementar as funções esperadas, com eles é possível controlar e monitorar o pêndulo além de transmitir e criar gráficos para analisar os resultados. Embora os códigos funcionem bem é importante fazer algumas observações pontuais. Primeiro: é preciso estar conectado a mesma rede de internet para conseguir acessar a Raspberry, portanto quando não é possível acessar a rede em que a Raspberry está conectada é necessário algum *software* que estabeleça conexão entre o computador e a rede (*softwares* como o Hamachi). Segundo: o código Python utilizado para criar os gráficos é sensível a dados errados, portanto se o arquivo de texto transmitido contiver erros nos dados (alguns erros são perceptíveis, por exemplo dados de tempo com números negativos) esses erros serão colocados no gráfico, e se uma linha do texto estiver incompleta, o que geralmente acontece por uma finalização inesperada da comunicação entre Raspberry e MSP, o algoritmo apresenta erros e não mostra o gráfico.

5 Conclusão

Os desafios encontrados no desenvolvimento deste laboratório foram importantes para um aprendizado sobre sistemas de controle e sua integração e dependência da mecânica e eletrônica utilizadas no projeto. A montagem e os testes realizados levaram a um maior conhecimento dos erros e acertos de concepção e desenvolvimento do pêndulo.

A eletrônica produzida para controlar, obter e transmitir os dados do experimento (conexão dos componentes na placa, confecção dos cabos, programação do *firmware* da MSP, desenvolvimento do código da recepção de dados e comando na Raspberry) funciona bem e cumpre as funções projetadas para ela, embora ainda existam pequenos erros. Uma sugestão para melhorar a eletrônica é substituir a placa furada atual pela PCB (que foi projetada no desenvolvimento deste trabalho) para garantir uma maior confiabilidade nas conexões entre os componentes e diminuir possíveis ruídos ou mal funcionamentos.

A frente de *software*, que neste projeto compreende a conexão entre computador do usuário e Raspberry e os algoritmos em Python para plotagem de gráficos funcionaram bem durante os testes realizados. Como sugestão de melhoria dessas frentes é possível apontar o desenvolvimento de uma interface gráfica para unir a recepção de dados, a exibição dos gráficos e o comando do experimento numa plataforma só para facilitar o uso por parte do usuário final. Também é recomendável procurar algum método para estabelecer a conexão das redes do computador e do usuário que não utilize um *software* de terceiros, para assim também poder conectar o computador ao pêndulo na mesma interface gráfica que comandaria todo o experimento.

Devido as não linearidades do tipo banda morta causadas pela folga das caixas de redução dos motores e pela derrapagem das rodas, e também pela baixa potência dos motores, os diferentes PIDs testados foram incapazes de estabilizar a barra por mais de um segundo. Como sugestão de melhoria para versões futuras é recomendado trocar os dois motores com rodas por um motor mais potente (um motor de passo é uma boa escolha) posicionado no centro da base fixa e conectado a base móvel diretamente por seu eixo de rotação, de forma que o motor aplique o torque diretamente a base móvel. Essa modificação eliminaria a necessidade de rodas, portanto eliminando a derrapagem e seus efeitos sobre o controle e também, dependendo da potência e torque do motor escolhido, eliminaria a necessidade da caixa de redução reduzindo a praticamente zero a banda morta por folgas de engrenagens.

Este trabalho implementou o laboratório do pêndulo invertido em várias frentes, e embora o controlador escolhido não tenha tido sucesso em compensar as não linearidades decorrentes dos motores utilizados, realizar a substituição mecânica anteriormente indicada

ou utilizar um método de controle mais robusto a banda morta que o PID deve permitir estabilizar a barra por mais tempo.

Referências

- °ASTRÖM, K. J.; HÄGGLUND, T. **PID controllers: theory, design, and tuning**. Research Triangle Park, Nc Instrument Society Of America, 1995. Citado na p. 43.
- AKHTARUZZAMAN, M.; SHAFIE, A. A. Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis. In: **2010 IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS AND AUTOMATION**. IEEE, ago. 2010. P. 1342–1347. ISBN 978-1-4244-5140-1. DOI: [10.1109/ICMA.2010.5589450](https://doi.org/10.1109/ICMA.2010.5589450). Disponível em: <http://ieeexplore.ieee.org/document/5589450/>. Citado na p. 24.
- ALBERTIN, M. R.; ELIENESIO, M. L. L.; SANTOS AIRES, A. dos; PONTES, H. L. J.; JUNIOR, D. P. A. Principais inovações tecnológicas da indústria 4.0 e suas implicações na manufatura. **XXIV Simpósio De Engenharia De Produção**, São Paulo, p. 0–13, 2017. Citado na p. 17.
- ALBINO, R. **As Principais Metodologias E Ferramentas Na Educação 4.0: Aplicação Interdisciplinar**. 2019. P. 80. ISBN 978-85-94294-26-5. Citado na p. 17.
- ALVES, G. R.; FLORENZA, J. R. Demonstração dos Resultados dos Projetos ALFA LEAL e RexNet. **Revista Técnico Científica do IFSC**, 2008. Citado na p. 18.
- ALVES, G. R. d. C.; SILVA, J. B. d.; PAVANI, A. M. B.; CARDOSO, D. C.; LIMA, D. A.; TAKAHASHI, E. K.; ROQUE, G. R.; TEMPORÃO, G. P.; NERI, H. G. F.; PEREIRA, J.; COSTA, L. A. C. d.; MENDES, L. A.; MOURA, R. M. d.; GEDRAITE, R.; BILESSIMO, S. M. S.; UHLMANN, T. S.; LIMA, V. A. P. Desafios da Educação em Engenharia. **ABENGE**, p. 10–47, 2016. Citado na p. 17.
- ANTONIO, M.; CARVALHO, G. D. E.; TULHA, C. Uso de Laboratórios Remotos na educação a distância no Brasil : uma revisão sistemática T. **Informática na Educação: teoria e prática**, n. 1, p. 195–209, 2019. Citado na p. 18.
- BOUBAKER, O. The inverted pendulum benchmark in nonlinear control theory: A survey. **International Journal of Advanced Robotic Systems**, v. 10, 2013. ISSN 17298806. DOI: [10.5772/55058](https://doi.org/10.5772/55058). Citado na p. 21.
- DIAO, X. ME 452 Course Project II: Rotary Inverted Pendulum. **Scribd**. [s.d.]. Disponível em: <https://pt.scribd.com/document/240436439/Inverted-Pendulum#>. Acesso em 21/07/2023, 2006. Citado na p. 20.
- DORF, R. C.; BISHOP, R. H. **Modern Control Systems**. 12. ed.: Prentice Hall, 2011. Citado na p. 21.

- EMBEDDED LAB. Tinkering TI MSP430F5529. Disponível em: <https://embedded-lab.com/blog/tinkering-ti-msp430f5529/>. Acessado em: 20/05/2023, s.d. Citado na p. 40.
- ISSA, A.; EL SHORAFI, M.; O. A. AQEL, M.; BRABAZON, D.; YOUNG, P. Remote Computer Based Learning System for Inverted Pendulum Lab Experiment. In: **2018 INTERNATIONAL CONFERENCE ON PROMISING ELECTRONIC TECHNOLOGIES (ICPET)**. IEEE, out. 2018. P. 113–117. ISBN 978-1-5386-5697-6. DOI: [10.1109/ICPET.2018.00027](https://ieeexplore.ieee.org/document/8531240/). Disponível em: [<https://ieeexplore.ieee.org/document/8531240/>](https://ieeexplore.ieee.org/document/8531240/). Citado nas pp. 24, 25.
- JAZIRI, I.; CHARAABI, L.; JELASSI, K. Remote web-based control laboratories using embedded Linux and field-programmable gate array. **Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering**, v. 232, n. 9, p. 1146–1154, out. 2018. ISSN 0959-6518. DOI: [10.1177/0959651818776542](https://doi.org/10.1177/0959651818776542). Disponível em: <http://journals.sagepub.com/doi/10.1177/0959651818776542>. Citado nas pp. 26, 27.
- JUNG, S.; KIM, S. S. Control experiment of a wheel-driven mobile inverted pendulum using neural network. **IEEE Transactions on Control Systems Technology**, v. 16, n. 2, p. 297–303, 2008. ISSN 10636536. DOI: [10.1109/TCST.2007.903396](https://doi.org/10.1109/TCST.2007.903396). Citado nas pp. 22, 23.
- KHALIL, H. K. **Non linear systems**. 3. ed.: Prentice Hall, 2002. Citado na p. 19.
- LIMA, J. P. C. de. **Desenvolvimento de servidores para laboratórios remotos baseado no paradigma de dispositivos inteligentes**. TCC(graduação) - Engenharia da Computação, Universidade Federal de Santa Catarina, Campus Araranguá., p. 89, 2016. Citado na p. 19.
- OGATA, K. **Modern control engineering**. 4. ed.: Aeeizh, 2002. Citado nas pp. 21, 23, 24.
- OLIVEIRA, K. K. d. S.; SOUZA, R. A. C. de. Habilitadores da transformação digital em direção à Educação 4.0. **Revista Novas Tecnologias na Educação**, v. 18, n. 1, 2020. ISSN 1679-1916. DOI: [10.22456/1679-1916.106012](https://doi.org/10.22456/1679-1916.106012). Citado na p. 17.
- ORANGE. LPD3806-100BM Encoder Datasheet. Disponível em: <https://datasheetspdf.com/pdf/1422931/ORANGE/LPD3806-100BM-G5-24C/1>. Acesso em: 28/09/2021, 2014. Citado na p. 35.
- PEREIRA, J.; SILVA, I. N. da; ALVES, J. B. d. M.; SILVA, J. B. da; ALVES, G. Implantação e Utilização do Laboratório Remoto VISIR em Instituições de Ensino Técnico, Tecnológico e Superior. **Revista Tecnologias na Educação**, p. 1–10, 2018. ISSN 1984-4751. Citado na p. 18.
- POLOLU. Bidirectional 4 Channel Logic Level Shifter Datasheet. Disponível em: <https://www.pololu.com/product/2595>. Acesso em: 28/09/2021, s.d. Citado na p. 35.

- ROSA, A. D.; MILHOMEM, R. Modelagem e controle de estabilização e posição de um pêndulo invertido. Disponível em: <https://tinyurl.com/2s4amxfv>. Acesso em: 29 de setembro de 2021, s.d. Citado na p. 24.
- SANCHEZ, J.; DORMIDO, S.; PASTOR, R.; MORILLA, F. A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum. **IEEE Transactions on Education**, v. 47, n. 3, p. 321–329, ago. 2004. ISSN 0018-9359. DOI: 10.1109/TE.2004.825525. Disponível em: <<http://ieeexplore.ieee.org/document/1323144/>>. Citado na p. 25.
- SANTOS, B.; LIMA, T. M. Indústria 4.0: desafios e oportunidades. **Revista Produção e Desenvolvimento**, May, 2018. Citado na p. 17.
- SILVA, C. W. d. **Sensors and actuators : control system instrumentation**. CRC Press, 2007. Citado nas pp. 33, 34.
- SILVA, J. B. d. **A utilização da experimentação remota como suporte para ambientes colaborativos de aprendizagem**. Tese (doutorado em Engenharia de Gestão do Conhecimento) - Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, 2006. Citado na p. 18.
- TEXAS INSTRUMENTS. Code Composer Studio™ integrated development environment (IDE). Disponível em: <https://www.ti.com/tool/CCSTUDIO>. Acessado em: 20/05/2023, s.d. Citado na p. 40.
- TEXAS INSTRUMENTS. eZ-FET lite Revision 1.10 User's Guide, 2013. Acessado em: 20/05/2023. Disponível em: <<https://tinyurl.com/yj7xtfxf>>. Citado na p. 40.
- TOSHIBA. TB6612FNG Datasheet. Disponível em: <https://tinyurl.com/3ukv8ewp>. Acesso em: 28/09/2021, s.d. Citado na p. 36.
- UFSC. **RELLE**. s.d. Ambiente de aprendizado com Experimentos Remotos. Disponível em: <http://relle.ufsc.br/>. Acesso em: 28/09/2021. Citado nas pp. 18, 19.
- V, V.; Z, J.; SRINIVASAN, A. Real time modelling and balance controller design for a rotary inverted pendulum – using LabVIEW. **Industrial Science**, 2013. Citado na p. 20.
- YI, J.; YUBAZAKI, N. Stabilization fuzzy control of inverted pendulum systems. **Artificial Intelligence in Engineering**, v. 14, n. 2, p. 153–163, 2000. ISSN 09541810. DOI: 10.1016/S0954-1810(00)00007-8. Citado na p. 22.
- YOUNG H. D.; FREEDMAN, R. A. **Física 1**. São Paulo: Addison Wesley, 2008. P. 403. Citado na p. 20.
- ZUMSANDE, J.; BOSSELMANN, S.; DAGEN, M.; ORTMAIER, T. Design and Development of a Remote Lab for Hands-On Education in Mechatronics and Control Engineering. Disponível em: https://laris.univ-angers.fr/_resource/IFAC2017/IFAC_Paper_4212.pdf. Acessado em: 21/07/2023, 2017. Citado nas pp. 25, 26.