

UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
FACULDADE DE ENGENHARIA ELÉTRICA

ALEXANDRE MONTIBELLER DE OLIVEIRA

CONTROLE DE ROBÔ PÊNDULO INVERTIDO SOBRE DUAS RODAS POR ABORDAGEM ANALÍTICA E EXPERIMENTAL

TUCURUÍ

2023

ALEXANDRE MONTIBELLER DE OLIVEIRA

CONTROLE DE ROBÔ PÊNDULO INVERTIDO SOBRE DUAS RODAS POR ABORDAGEM ANALÍTICA E EXPERIMENTAL

Trabalho de Conclusão de Curso, apresentado como requisito parcial para a obtenção de grau de Bacharel em Engenharia Elétrica, pela Universidade Federal do Pará.

Orientador:
Prof. Dr. Raphael Barros Teixeira.

TUCURUÍ

2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

O48c Oliveira, Alexandre Montibeller de.
Controle de robô pêndulo invertido sobre duas rodas por abordagem analítica e experimental / Alexandre Montibeller de Oliveira. — 2023.
118 f. : il. color.

Orientador(a): Prof. Dr. Raphael Barros Teixeira
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Pará, Campus Universitário de Tucuruí, Faculdade de Engenharia Elétrica, Tucuruí, 2023.

1. Robô Pêndulo Invertido Sobre Duas Rodas. 2. Espaço de Estados. 3. Observador de Estados. 4. Ziegler-Nichols. I. Título.

CDD 620.0042

ALEXANDRE MONTIBELLER DE OLIVEIRA

CONTROLE DE ROBÔ PÊNDULO INVERTIDO SOBRE DUAS RODAS POR ABORDAGEM ANALÍTICA E EXPERIMENTAL

Trabalho de Conclusão de Curso, apresentado como requisito parcial para a obtenção de grau de Bacharel em Engenharia Elétrica, pela Universidade Federal do Pará.

Data de aprovação: 15/12/2023

Conseito: EXCELENTE

Banca Examinadora:

gov.br Documento assinado digitalmente
RAPHAEL BARROS TEIXEIRA
Data: 07/03/2024 14:25:04-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Raphael Barros Teixeira
Orientador - FEE/CAMTUC/UFPA

gov.br Documento assinado digitalmente
TELSON GALDINO PINHEIRO TEIXEIRA
Data: 01/02/2024 19:25:58-0300
Verifique em https://validar.iti.gov.br

Prof. Telson Galdino Pinheiro Teixeira
Avaliador Interno - FEE/CAMTUC/UFPA

gov.br Documento assinado digitalmente
ANDRE FELIPE SOUZA DA CRUZ
Data: 01/02/2024 15:43:50-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. André Felipe Souza da Cruz
Avaliador Interno - FEE/CAMTUC/UFPA

TUCURUÍ

2023

Dedico aos meus pais e a toda a minha família e amigos.

AGRADECIMENTOS

A jornada foi longa e bastante desafiadora. Devo agradecer à minha mãe Marilí Montibeller e ao meu pai José Ronaldo por todo o apoio e por acreditarem junto comigo no sonho de me formar. Agradeço à minha família por dar todo o suporte e incentivo para chegar onde cheguei.

Agradeço ao corpo docente e a todos os funcionários da Universidade Federal do Pará, campus de Tucuruí, por todo o empenho em tornar nossa universidade uma referência em ensino, pesquisa e extensão. Em especial, quero agradecer ao meu professor orientador, Raphael Barros Teixeira, pela dedicação e preocupação, sem as quais não seria possível concluir esta monografia.

Aos meus amigos, expresso minha admiração pelas histórias de vida, resiliência e companheirismo. Todos me proporcionaram experiências inestimáveis que marcaram minha trajetória pessoal e profissional. Em especial, quero agradecer ao Telson Galdino e Walder Júnior pela contribuição indispensável para a elaboração desta monografia e pela amizade de longa data. Também, aos meus amigos Ricardo Freitas, Matheus Gonzaga, Geisielen Seth, Nílton Gomes, Kayse Júnior e Josemar Oliveira por todos os desafios que superamos juntos nesse período da graduação e pelas experiências de vida que construímos em conjunto.

Agradeço à empresa júnior Esquadro Projetos e Consultorias por todas as experiências pessoais e profissionais que foram determinantes para fundamentar o perfil de profissional que possuo hoje. Também, ao projeto Muiraquitã, em especial à capitã Micilene Bastos, pelas experiências indescritíveis de superação e resiliência, demonstrando que com dedicação e trabalho em equipe é possível vencer desafios em condições desfavoráveis.

Apesar de não ser possível agradecer nominalmente cada pessoa que contribuiu direta ou indiretamente para que o meu sonho de formar fosse possível, faço questão de demonstrar essa gratidão sendo prestativo a quem necessitar do meu apoio sempre que possível.

"Entre os mais fortes existem os que nasceram com um dom, e aqueles que trabalham duro... E eu sou um dos que trabalhou duro!"

(Rock Lee, do anime *Naruto Shippuden*).

RESUMO

Este trabalho investigou os resultados da implementação na prática de um método analítico e um experimental de controle de posição angular de um robô pêndulo invertido sobre duas rodas construído no laboratório da UFPA (Universidade Federal do Pará), campus Tucuruí. O objetivo do controle foi manter o robô em equilíbrio vertical em regime permanente. O método analítico foi realizado por meio da modelagem do sistema por espaço de estados e projeto de controlador por realocação de polos com observador de estados. A parte teórica desse método foi desenvolvida com base nos parâmetros do robô do laboratório da UFPA. A contribuição desse trabalho foi implementar essa teoria no robô e analisar os resultados. O método experimental foi o de sintonia de controlador PID com base em resposta em frequência proposta por Ziegler-Nichols. Nesse trabalho foram apresentados os materiais e as metodologias utilizadas para adaptar a configuração física do chassi do robô, seus componentes eletromecânicos e de fixação, e o algoritmo de controle para encontrar um equilíbrio entre fazer o robô em malha aberta permanecer por mais tempo próximo do ponto de operação 0° e, em malha fechada, evitar que a posição angular do pêndulo variasse de maneira abrupta devido à aplicação de valores pequenos de torque pelos motores. Também foi apresentada a forma como ocorreu a integração do hardware do robô com a lógica de controle no Arduino por meio da plataforma IDE (Integrated Development Environment), tanto para o método analítico quanto para o experimental. O método analítico não teve o comportamento esperado, de forma que não foi eficaz no controle da posição angular do robô, sendo necessário mais pesquisa para viabilizar a aplicação da mesma. Para o método experimental de Ziegler-Nichols, foi alcançado o equilíbrio do robô em regime permanente; no entanto, ele não possibilitou um conhecimento mais aprofundado da dinâmica do sistema e a definição de requisitos de projeto mais específicos.

Palavras-chave: Robô Pêndulo Invertido Sobre Duas Rodas, Espaço de Estados, Observador de Estados, Ziegler-Nichols.

ABSTRACT

This work investigated the outcomes of implementing, in practice, an analytical and an experimental method for the angular position control of a two-wheeled inverted pendulum robot constructed in the laboratory of UFPA (Federal University of Pará), Tucuruí campus. The control objective was to maintain the robot in vertical balance in a steady state. The analytical method involved modeling the system using state-space representation and designing a controller through pole relocation with a state observer. The theoretical aspect of this method was developed based on the parameters of the UFPA laboratory robot. The contribution of this work was to implement this theory on the robot and analyze the results. The experimental method consisted of tuning a PID controller based on frequency response proposed by Ziegler-Nichols. The materials and methodologies used to adapt the physical configuration of the robot chassis, its electromechanical components, fixing elements, and the control algorithm were presented in this study. The objective was to find a balance, allowing the open-loop robot to remain closer to the 0° operating point for a longer time and, in closed-loop, preventing abrupt variations in the pendulum's angular position due to the application of small torque values by the motors. The integration of the robot's hardware with the control logic on Arduino through the Integrated Development Environment (IDE) platform was also described for both the analytical and experimental methods. The analytical method did not exhibit the expected behavior, proving ineffective in controlling the robot's angular position, necessitating further research for its potential application. As for the Ziegler-Nichols experimental method, permanent equilibrium for the robot was achieved; however, it did not provide a more in-depth understanding of the system dynamics or the definition of more specific design requirements.

Keywords: *Inverted Pendulum Robot on Two Wheels, State-Space, State Observer, Ziegler-Nichols.*

LISTA DE ILUSTRAÇÕES

Figura 1 – Robô pêndulo invertido sobre duas rodas 17
Figura 2 – Segway S-Pod 19
Figura 3 – Veículo elétrico de transporte EN-V 19
Figura 4 – Robô pêndulo invertido sobre duas rodas construído no laboratório da UFPA (Universidade Federal do Pará), campus Tucuruí 20
Figura 5 – Sistema pêndulo invertido sobre carrinho 24
Figura 6 – Diagrama de corpos pêndulo invertido sobre carrinho 24
Figura 7 – Modelo robô pêndulo invertido sobre duas rodas vista lateral 27
Figura 8 – Modelo robô pêndulo invertido sobre duas rodas vista frontal 27
Figura 9 – Vista lateral do pêndulo 29
Figura 10 – Vista frontal do pêndulo 29
Figura 11 – Vista lateral da roda 29
Figura 12 – Vista frontal da roda 29
Figura 13 – Circuito elétrico dos motores 30
Figura 14 – Simulação pêndulo robô malha aberta 33
Figura 15 – Gráfico de análise de Variáveis 33
Figura 16 – Sistema de controle com retroalimentação do estado observado 34
Figura 17 – Gráfico de análise do observador de estado 36
Figura 18 – Sistema em malha fechada 36
Figura 19 – Sistema de malha fechada com um controlador proporcional. 39
Figura 20 – Resposta oscilatória sustentada do sistema 40
Figura 21 – Resposta do robô em malha fechada para ganho 0,95 e período de amostragem de 15 ms 41
Figura 22 – Exemplo da aplicação da transformada de Fourier 42
Figura 23 – Sistema com controle PID 42
Figura 24 – Diagrama de blocos do sistema com o controlador PID projetado com o uso da regra de sintonia de resposta em frequência de Ziegler-Nichols. . 44
Figura 25 – Curva de resposta ao degrau unitário de um sistema com controlador PID projetado com o uso da regra de sintonia de resposta em frequência de ZieglerNichols 45
Figura 26 – Definição dos ângulos Roll, Pitch e Yaw. 47
Figura 27 – Exemplo para ilustrar a definição de ruído e viés. 49
Figura 28 – Medidas do giroscópio para 10 segundos de dados estacionários 49
Figura 29 – Medidas do acelerômetro para 10 segundos de dados estacionários . . . 49
Figura 30 – Histograma (azul) das medidas do giroscópio para 10 segundos de um sensor em estado de repouso e um ajuste gaussiano (vermelho) aos dados 50

Figura 31 – Histograma (azul) das medidas do acelerômetro para 10 segundos de um sensor em estado de repouso e um ajuste gaussiano (vermelho) aos dados . . . 51
Figura 32 – Exemplo de fusão de sensores. . . . 52
Figura 33 – Esquema do filtro complementar. . . . 52
Figura 34 – Face da frente do robô ajustado . . . 55
Figura 35 – Face de trás do robô ajustado . . . 55
Figura 36 – Pontos de localização de cada componente do robô pêndulo invertido móvel. . . . 55
Figura 37 – O módulo GY-521 com o sensor inercial MPU-6050 . . . 56
Figura 38 – Sistema de referência MPU . . . 57
Figura 39 – Arduino Mega 2560 Keyestudio. . . . 57
Figura 40 – Ponte H L298N . . . 58
Figura 41 – JGA25-371 DC Gearmotor com Encoder . . . 59
Figura 42 – Case de 3 baterias UR18650A M19 ligadas em série. . . . 60
Figura 43 – Bateria UR18650A M19. . . . 60
Figura 44 – Diagrama de blocos para o fluxo de sinais do robô modelo em espaço de estados . . . 61
Figura 45 – Diagrama de blocos para o fluxo de sinais do robô com controlador com estrutura de PID . . . 62
Figura 46 – Modelo genéricos de chassi de robô pêndulo invertido sobre duas rodas com 3 pisos. Não está em escala . . . 63
Figura 47 – Resposta ao degrau e lugar geométrico das raízes para diferentes valores de l . . . 64
Figura 48 – Chassi e posicionamento da distribuição de massas nos pisos . . . 65
Figura 49 – Esquema de conexão elétrica real do pêndulo móvel . . . 66
Figura 50 – Esquema elétrico do pêndulo móvel . . . 67
Figura 51 – Esquema elétrico da ponte H L298N . . . 68
Figura 52 – Sentido de rotação dos motores em relação ao eixo de coordenadas do IMU de acordo com o nível lógico aplicado nos pinos de controle 3, 4, 9 e 10 do Arduino nas portas IN1, IN2, IN3 e IN4 da Ponte H - 1° Teste 69
Figura 53 – Sentido de rotação dos motores em relação ao eixo de coordenadas do IMU de acordo com o nível lógico aplicado nos pinos de controle 3, 4, 9 e 10 do Arduino nas portas IN1, IN2, IN3 e IN4 da Ponte H - 2° Teste 70
Figura 54 – Duty cycle em função da resolução do Arduino . . . 71
Figura 55 – Não-linearidade da zona morta . . . 72
Figura 56 – Posição angular do pêndulo do robô . . . 73
Figura 57 – Sinal de Pitch estimado pelo acelerômetro e pelo filtro complementar. . . 73
Figura 58 – Fluxograma de funcionamento do robô . . . 74

Figura 59 – Fluxograma do controle do robô 75
Figura 60 – O software CoolTerm utilizado para obtenção dos sinais 77
Figura 61 – Terminal com valores enviados para a porta serial 78
Figura 62 – Representação em diagrama de blocos do sistema com controlador-observador. 80
Figura 63 – Sinal da posição do pêndulo θ para $T_s = 0,0015s$ 83
Figura 64 – Transforma de Fourier do sinal da posição do pêndulo θ 83
Figura 65 – Posição angular do pêndulo em função do tempo 86
Figura 66 – Sinais amostrados para $T_s = 0,015s$ 87
Figura 67 – Posição angular do pêndulo para $T_s = 0,010s$ e $K_{cr} = 0,72$ 88
Figura 68 – Transformada de Fourier do sinal da posição angular do pêndulo θ para $T_s = 0,010s$ e $K_{cr} = 0,72$ 89
Figura 69 – Sinais amostrados para $T_s = 0,010s$ 89
Figura 70 – Posição angular do pêndulo θ para $T_s = 0,0053s$ e $K_{cr} = 0,0475$ 90
Figura 71 – Transformada de Fourier do sinal da posição angular do pêndulo θ para $T_s = 0,0053s$ e $K_{ctr} = 0,0475$ 91
Figura 72 – Sinais amostrados $T_s = 0,0053s$ 91
Figura 73 – Sinais amostrados $T_s = 0,015s$ 92
Figura 74 – - Placas dos pisos do pêndulo móvel: placa do primeiro segundo e terceiro piso 115

LISTA DE TABELAS

Tabela 1 – Parâmetros do robô pêndulo invertido sobre rodas 32
Tabela 2 – Tabela para definição das constante do controlador pelo método ZN . . 40
Tabela 3 – Mapa de portas 66
Tabela 4 – Controle do motor segundo as entradas da ponte H. 68
Tabela 5 – Tabela de configuração do offset dos motores 72
Tabela 6 – Parâmetros do robô pêndulo invertido sobre rodas após ajustes 79
Tabela 7 – Parâmetros de diferentes controladores para $T_s = 0,015s$ 87
Tabela 8 – Parâmetros de diferentes controladores para $T_s = 0,010s$ 88
Tabela 9 – Parâmetros de diferentes controladores para $T_s = 0,0053s$ 90
Tabela 10 – Dificuldades e Soluções 94
Tabela 11 – Lista de Materiais 116
Tabela 12 – Lista de Ferramentas 117

LISTA DE QUADROS

Quadro 1 – Arranjo de Routh 43

Quadro 2 – Inicialização das variáveis 85
Quadro 3 – Implementação do Controlador-Observador 85
Quadro 4 – Aplicação do sinal de controle nos motores 86

SUMÁRIO

1 INTRODUÇÃO 17
1.1 Objetivos 19
1.2 Metodologia Aplicada 20
1.3 Estrutura da Monografia 21

2 FUNDAMENTAÇÃO TEÓRICA 23
2.1 O Pêndulo Invertido 23
2.2 Pêndulo Invertido Sobre Carrinho 23
2.3 O Robô Pêndulo Invertido Sobre Duas Rodas 27
2.4 Modelagem da Planta, Controlador e Observador pela Metodologia Analítica 28
2.4.1 Modelagem e Simulação em Malha Aberta do Sistema Pêndulo Invertido Sobre Duas Rodas 28
2.4.1.1 Diagrama de Forças do Corpo do Robô 28
2.4.1.2 Diagrama de Forças das Rodas 28
2.4.1.3 Circuito Elétrico dos Motores Elétricos DC 29
2.4.1.4 Modelagem do Sistema Por Espaço de Estados 30
2.4.1.5 Sistema em Malha Aberta 32
2.5 Controle de Sistemas Dinâmicos 35
2.6 Controlador PID 37
2.7 Método de Resposta em Frequência Ziegle-Nichols para Sintonia de Controlador PID 39
2.7.1 Exemplo da Aplicação do Método de Resposta em Frequência de Ziegle-Nichols. 42
2.8 Sensores Inerciais 45
2.8.1 Postura 47
2.8.2 Filtragem 48
2.8.2.1 Ruído e Viés de Medições de Sensores Inerciais Acelerômetro e Giroscópio 48
2.8.3 Filtragem dos Sinais do Acelerômetro e Giroscópio 51
2.8.4 Fusão de Sensores 51
2.9 Filtro Complementar 52

3 MATERIAIS E MÉTODOS UTILIZADOS 54
3.1 Materiais que Compõem o Robô Pêndulo Invertido Móvel . . . 54
3.1.1 Sensor Inercial MPU-6050 56
3.1.2 Arduino MEGA 2560 57

3.1.3 Ponte H L298N . . . 58
3.1.4 Motor DC 6V 280RPM com Encoder e Roda . . . 58
3.1.5 Case de 3 Baterias UR18650A M19 Ligadas em Série . . . 59
3.2 Diagrama do Fluxo de Sinais do Robô . . . 59
3.3 Metodologia para Design do Chassi do Robô . . . 63
3.4 Montagem do Robô . . . 65
3.5 Configuração e Integração do Hardware do Robô com os Componentes Eletrônicos . . . 68
3.6 Sensor Inercial e Filtragem . . . 72
3.7 Rotina de Controle Digital . . . 74
3.8 Comunicação e Amostragem . . . 75
3.9 Modelagem da Planta, Projeto de Controlador Observado de Acordo com a Sessão 2.4 . . . 78
3.9.1 Discretização do Controlador e Observador em Espaço de Estados . . . 80
3.10 Projeto e Implementação no Arduino o Controlador PID pela Metodologia de Resposta em Frequêdia de Ziegle-Nichols . . . 82

4 ANÁLISE DOS RESULTADOS . . . 85
4.1 Resultados da Aboragem Analítica . . . 85
4.2 Resultados da Abordamgem Experimental . . . 87

5 CONCLUSÃO . . . 93
5.1 Dificuldades Encontradas . . . 94
5.2 Trabalhos Futuros . . . 94
5.3 Compartilhamento dos Dados Experimentais Coletados . . . 95

Referências . . . 96

APÊNDICE A – CÓDIGOS MATLAB 2016A . . . 98
A.1 Códigos para construção da Figura 47 . . . 98
A.2 Códigos para montar gráfico do sinal da posição do pêndulo θ . . . 99
A.3 Códigos para montar a figura 22 . . . 100
A.4 Códigos para montar gráfico da transformada de Fourier . . . 101
A.5 Códigos para calculas as constantes do controlador PID . . . 102
A.6 Códigos para montar os gráficos dos sinais amostrados da posição angular do pêndulo θ . . . 103
A.7 Código para calcular a função de transferência discreta do controlador-observador . . . 104

APÊNDICE B – CÓDIGO ARDUINO DO PÊNDULO INVERTIDO 109

APÊNDICE C – DETALHAMENTO DOS PISOS DO ROBÔ 115

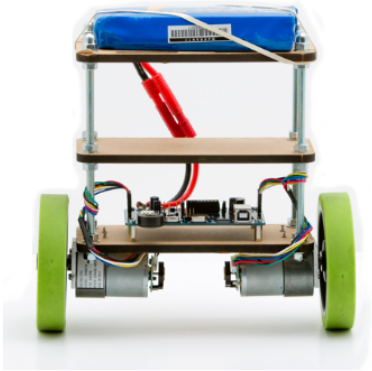
APÊNDICE D – LISTA DE MATERIAIS E FERRAMENTAS 116

1 INTRODUÇÃO

A teoria de controle de sistemas dinâmicos na engenharia tem como objetivo estudar e desenvolver métodos para controlar o comportamento de sistemas dinâmicos ao longo do tempo. Desde a antiguidade, sistemas como moinhos de vento utilizavam controles simples. Durante a Revolução Industrial, a crescente necessidade de controle para máquinas a vapor levou ao desenvolvimento de reguladores automáticos, como o criado por James Watt. Na década de 1920, a teoria de controle clássica emergiu com autores como Bode, Nyquist e Shannon. Nas décadas de 1940 e 1950, a teoria clássica possibilitou o design de sistemas de controle linear com métodos como resposta em frequência e lugar geométrico das raízes. Nesse período, o controlador PID tornou-se amplamente utilizado, e o método de Ziegler-Nichols foi estabelecido como uma das formas de sintonia experimental. Essa época também testemunhou o início da teoria de identificação de sistemas, voltada para a obtenção de modelos matemáticos a partir de dados experimentais. A década de 1950 marcou o início da teoria de controle moderno, que introduziu o espaço de estado para resolver problemas lineares e não lineares com múltiplas entradas e saídas. A partir da década de 1960, com a chegada dos computadores digitais, iniciou-se o desenvolvimento da teoria de controle digital. Atualmente, a evolução da teoria de controle inclui a crescente aplicação de inteligência artificial e aprendizado de máquina, bem como o avanço de sistemas de controle autônomos (OGATA et al., 2010).

Com o avanço dessas novas técnicas e sua aplicação em diversas áreas, o estudo de problemas modelo passou a ser incorporado em pesquisas que exploram a eficiência, propostas de melhorias e a robustez dos controladores. Um dos desafios mais estudados pela comunidade acadêmica, abrangendo diversos aspectos da teoria de controle desenvolvida até a atualidade, é o controle da posição vertical de um sistema pêndulo invertido sobre duas rodas, conforme ilustrado na figura 1 (BOUBAKER, 2012).

Figura 1 – Robô pêndulo invertido sobre duas rodas



Fonte: (BALANDUINO. . . , 2022)

Este sistema eletromecânico consiste em um corpo que possui um centro de massa apoiado sobre o eixo que liga duas rodas. Sem uma ação de controle aplicada através dos motores, ele tende a se desequilibrar e cair no chão. O controle desse sistema é um referencial, pois apresenta uma combinação de desafios teóricos e práticos que são representativos de muitos problemas de controle do mundo real. De acordo com (BOUBAKER, 2012), alguns dos desafios para controlar esse tipo de sistema são os seguintes:

a) Representar matematicamente a dinâmica desses sistemas é complexo;

b) São sistemas com múltiplas variáveis de entrada e saída;

c) Apresentam forte não linearidade, ou seja, não há uma relação linear entre as variáveis de entrada e saída;

d) São naturalmente instáveis; na ausência de um sistema de controle, não atingem o ponto de operação desejado.

Uma das aplicações mais comuns de sistemas semelhantes a um pêndulo invertido sobre duas rodas é o transporte de pessoas. A figura 2 mostra uma poltrona móvel sobre duas rodas, ideal para movimentação em ambientes fechados, como shoppings, escritórios, aeroportos, entre outros, onde seja necessário percorrer distâncias consideráveis. A figura 3 demonstra um veículo elétrico sobre duas rodas, com capacidade para duas pessoas, projetado para transporte urbano, normalmente em trajetórias de casa para o trabalho e vice-versa.

A crescente necessidade da sociedade por veículos mais compactos, baratos e elétricos fomenta a pesquisa de técnicas de controle de sistemas, como o robô pêndulo invertido, com o objetivo de tornar essa tecnologia mais acessível para todos. Uma das formas de alcançar esse objetivo é desenvolver protótipos didáticos para aplicar a teoria de controle na prática, ajustando de forma incremental a estratégia de controle até chegar a um momento em que a pesquisa desenvolva um protótipo capaz de atender às necessidades da sociedade.

Nesse contexto, a motivação para este trabalho foi aplicar uma metodologia analítica e experimental de controle para equilibrar verticalmente, em regime permanente, um robô pêndulo invertido sobre duas rodas construído no laboratório da UFPA (Universidade Federal do Pará), campus Tucuruí.

No método analítico, foram utilizados os resultados da pesquisa conduzida por (TEIXEIRA, 2023), que desenvolveu a modelagem em espaço de estados do robô mencionado acima, além do projeto do controlador pelo método de alocação de polos e do observador de estados. O controlador-observador foi testado em simulação, comprovando

Figura 2 – Segway S-Pod



Fonte: (SEGWAY, 2023)

Figura 3 – Veículo elétrico de transporte EN-V



Fonte: (MOTAVALLI, 2010)

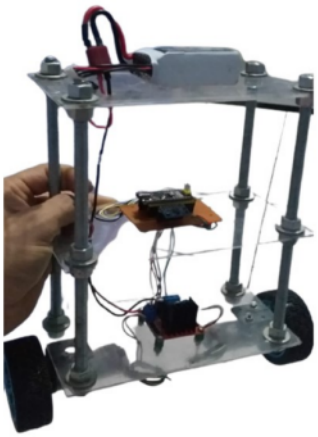
atender aos requisitos de projeto. A aplicação do método em planta real foi deixada para trabalhos futuros. A imagem do robô antes do início deste trabalho está apresentada na figura 4. A contribuição deste trabalho foi utilizar uma metodologia para adaptar a estrutura do robô, permitindo que, em malha aberta, permanecesse mais tempo próximo do ponto de operação de 0°, e em malha fechada, não variasse a posição angular do pêndulo de maneira abrupta devido a pequenas aplicações de torque pelos motores. Esses ajustes facilitaram a implementação do controlador-observador proposto por (TEIXEIRA, 2023) na planta real. No entanto, não foram alcançados os requisitos de projeto definidos, especialmente a estabilização do sistema, indicando a necessidade de continuidade da pesquisa para aplicar o método na prática.

Para o método experimental, foi realizada a sintonia do controlador PID (Proporcional, Integral e Derivativo) através do método de resposta em frequência de Ziegler-Nichols. Observou-se que, apesar de este método não fornecer uma descrição da dinâmica da planta e não permitir a definição de requisitos de projetos mais específicos, foi eficaz na estabilização da planta, mantendo-a na vertical em regime permanente e com uma resistência a perturbações externas aceitável.

1.1 Objetivos

O desafio deste projeto é manter um corpo sustentado por duas rodas na posição vertical por um longo período de tempo de forma automática. Portanto, o objetivo principal

Figura 4 – Robô pêndulo invertido sobre duas rodas construído no laboratório da UFPA (Universidade Federal do Pará), campus Tucuruí



Fonte: (TEIXEIRA, 2023)

é investigar a aplicação de uma metodologia analítica e experimental para projetar um controlador capaz de manter o robô na posição vertical em regime permanente. Para alcançar esse objetivo principal, alguns objetivos específicos devem ser realizados, sendo eles:

a) Ajustar a estrutura física do robô pêndulo invertido da figura 4 para em malha aberta permanecer mais tempo próximo do ponto de operação de 0° e, em malha fechada, não variar a posição angular do pêndulo de maneira abrupta devido a pequenas aplicações de torque pelos motores ;

b) Realizar a integração do hardware do robô com com o algorítmo de controle que deve ser implementado no Arduino, incluindo a medição precisa da posição angular do pêndulo usando o filtro complementar;

c) Coletar e gravar os dados das variáveis do sistema do robô por conexão serial via cabo;

d) Modelar a planta, projetar e discretizar controlador e observador de acordo com o método da pesquisa de (TEIXEIRA, 2023);

e) Projetar e discretizar controlador PID pelo método de sintonia experimental de reposta em frequência de Ziegler-Nichols;

f) Analisar os resultados.

1.2 Metodologia Aplicada

O primeiro passo foi desmontar o robô da figura 4 e reprojetar a configuração física da estrutura de suporte e dos componentes eletrônicos, mantendo o formato original de

paralelepípedo do chassi. O projeto dos pisos foi realizado utilizando o AutoCAD e pode ser visto no Apêndice C. Depois, foi reprojetada a interligação dos componentes eletrônicos e, por fim, veio a montagem.

O segundo passo foi realizar a integração do hardware do robô com as configurações iniciais dos componentes eletrônicos controlados pelo Arduino. Nessa etapa, foi feita a conexão do Arduino com o computador por um cabo USB e configurada a ação de controle dos pinos dos sensores inerciais e do driver de potência pela plataforma IDE. Foi realizado um processo de filtragem e fusão dos sinais medidos pelos sensores utilizando o filtro complementar para garantir que a variável medida da posição do pêndulo fosse precisa. Por fim, foram feitos ajustes de offset de comando dos motores e do offset da posição angular do pêndulo.

O terceiro passo foi coletar os dados das variáveis do sistema do robô por conexão serial via cabo. Os dados foram gravados com o auxílio de um programa chamado CoolTerm, que plotava os dados e os gravava em um arquivo de texto em formato CSV que posteriormente foi levado para o aplicativo MATLAB para os cálculos, análises e plotagem de gráficos.

O quarto passo foi definir o modelo da planta, do controlador e observador utilizando a metodologia proposta por (TEIXEIRA, 2023), levando em consideração os parâmetros do robô após os ajustes. O controlador foi discretizado e implementado no microcontrolador via código pela plataforma IDE.

O quinto passo foi projetar vários controladores com estrutura de PID utilizando a metodologia experimental de Ziegler-Nichols. Esses controladores foram implementados no microcontrolador via código pela plataforma IDE utilizando a biblioteca PID v1 como forma de discretização.

O passo final foi realizar a análise dos resultados. Nesta etapa, foram analisados os sinais de posição angular e de controle de forma qualitativa e quantitativa.

A aplicabilidade do método experimental definido por esta investigação foi validada pela estabilização bem-sucedida do equilíbrio do pêndulo no caso do controlador PID definido por Ziegler-Nichols. No caso do controlador proposto por (TEIXEIRA, 2023), foi recomendado, para trabalhos futuros, o refinamento da metodologia para que ela atinja o comportamento esperado.

1.3 Estrutura da Monografia

Este trabalho está estruturado em 5 capítulos.

No capítulo 1, introdutório, é apresentado um apanhado histórico da teoria de controle, justificativa do trabalho, aplicações práticas, objetivos e estrutura do trabalho.

No capítulo 2, é apresentada a fundamentação teórica sobre a dinâmica do robô pêndulo invertido sobre duas rodas, a metodologia analítica e experimental abordadas, e o processo de medição, filtragem e fusão de sensores para a definição da posição angular do pêndulo do robô.

No capítulo 3, são apresentados os materiais e métodos aplicados para o controle da posição angular do pêndulo do robô.

No capítulo 4, são apresentados os resultados obtidos a partir da execução das metodologias do capítulo 3.

Por fim, são apresentadas as considerações finais, dificuldades encontradas, trabalhos futuros e apêndices.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 O Pêndulo Invertido

Qualquer sistema que possui um corpo com centro de massa acima de um ponto de apoio articulado, sobre o qual o corpo pode rotacionar livremente, é um pêndulo invertido. Um bastão equilibrado na vertical, com uma extremidade apoiada na palma da mão, é um pêndulo invertido. Neste exemplo, pode-se verificar que existe um ponto de inclinação do bastão em que se consegue com maior facilidade mantê-lo em equilíbrio, ou seja, mantê-lo parado. Esse ponto é o ponto em que o bastão se encontra perfeitamente na vertical. Todavia, se não existir uma ação de controle da mão, a tendência natural é o bastão cair. Nesse caso, pode-se inferir que o ponto em que o bastão fica na vertical é um ponto de equilíbrio instável. Além disso, pode-se verificar que quanto mais o bastão inclina sobre a mão, mais difícil fica tentar mantê-lo em equilíbrio; logo, esse sistema apresenta uma dinâmica altamente imprevisível longe do ponto de equilíbrio instável. Essa imprevisibilidade pode ser definida como um sistema com características não lineares.

Tendo em vista essas características empíricas do sistema dinâmico de pêndulos invertidos, o próximo passo será buscar uma representação matemática que auxilie numa análise mais profunda do comportamento deles.

2.2 Pêndulo Invertido Sobre Carrinho

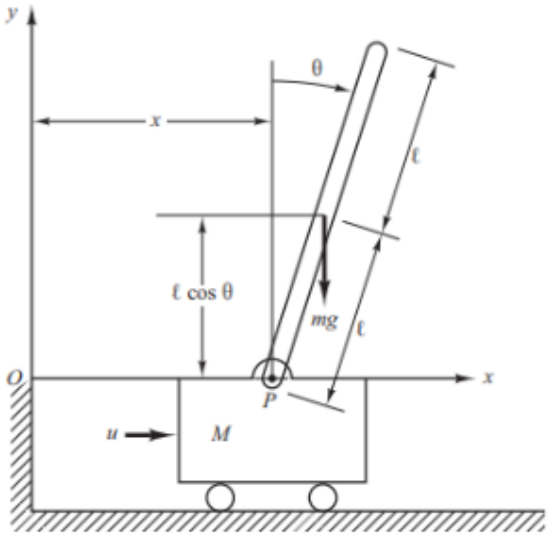
Inicialmente, será analisado um modelo mais simples de pêndulo invertido sobre carrinho que, para condições bem restritas, pode ser utilizado para compreender a dinâmica de um robô pêndulo invertido sobre duas rodas. Esse sistema, pêndulo invertido sobre carrinho, pode ser representado pela figura 5 e 6.

A partir da representação das figuras 5 e 6, é possível obter um modelo matemático que descreve o comportamento dinâmico desse sistema. Essa modelagem foi baseada na literatura (OGATA et al., 2010).

Nesta representação, temos uma haste de comprimento $2l$ e massa m ligada a um carrinho de massa M. O carrinho é limitado a um deslocamento retilíneo para a direita e para a esquerda. A haste se conecta ao carrinho por uma articulação que limita seu movimento a uma trajetória circular tanto no sentido horário quanto no anti-horário. Portanto, nota-se que este é um tipo de problema limitado a um campo bidimensional.

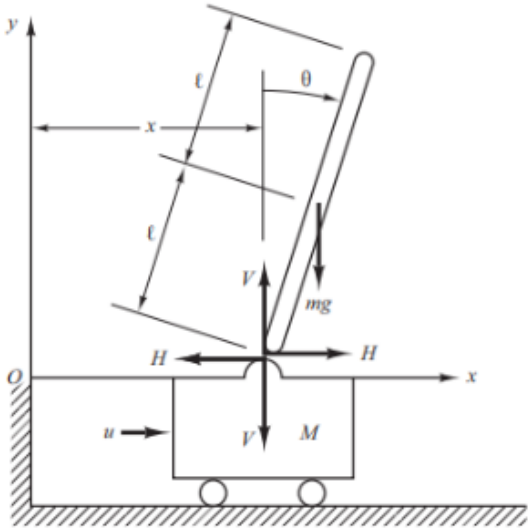
Na representação, u é uma força aplicada no carrinho. Essa força pode ser, por exemplo, proveniente de um motor elétrico. O θ é o ângulo que a haste faz com uma reta perpendicular à superfície superior do carrinho e que passa pelo ponto de referência P.

Figura 5 – Sistema pêndulo invertido sobre carrinho



Fonte: (OGATA et al., 2010)

Figura 6 – Diagrama de corpos pêndulo invertido sobre carrinho



Fonte: (OGATA et al., 2010)

Já o x é a distância entre o ponto de referência O e o ponto P. O g é a aceleração da gravidade, e V e H são as resultantes das forças aplicadas nas direções de x e y.

Outra consideração é que a distribuição da massa da haste é homogênea e, por esse motivo, seu centro de gravidade se localiza no seu centro geométrico. O ponto de centro de gravidade da haste pode ser localizado pelas coordenadas (x_G, y_G).

Então,

$$x_G = x + l \operatorname{sen} \theta \tag{1}$$

$$y_G = x + l \cos \theta \tag{2}$$

Para deduzir as equações de movimento do sistema, basta usar a segunda Lei de Newton, que determina que a soma dos torques aplicados sobre um corpo é igual à inércia de rotação do objeto vezes a aceleração angular dele.

Torque é uma forma de medir o potencial de uma força rotacionar um objeto em torno de um eixo. Quanto mais longe a força é aplicada em relação ao eixo, maior é a facilidade de uma força rotacionar esse objeto. Nesse caso, torque pode ser definido matematicamente da seguinte forma:

$$\bar{T} = \bar{F} \times \bar{d} \tag{3}$$

Onde $\bar{T}$ é o vetor torque, $\bar{F}$ é o vetor força e $\bar{d}$ é o vetor distância entre o ponto de aplicação de uma força e o eixo de rotação do objeto;

Experimentalmente, foi observado que forças que são aplicadas na mesma direção do vetor distância não geram movimento giratório. Nesse caso, apenas forças aplicadas perpendicularmente ao vetor distância geram torque.

Para calcular o módulo do torque, pode ser usado a seguinte expressão:

$$T = F \cdot d \cdot \operatorname{sen}\theta \tag{4}$$

Sendo θ o ângulo formado entre a força aplicada em relação à superfície do objeto que será rotacionado.

A direção e o sentido são dados por convenção, respeitando as regras do produto vetorial da equação 3. O fato é que o torque tem direção perpendicular simultaneamente ao vetor distância e à força, com sentido respeitando a regra da mão direita.

Tendo em vista isso, T_H, que é o módulo do torque devido à força resultante na direção de x decomposta perpendicularmente em relação à haste, pode ser dado pela seguinte equação:

$$T_H = Hl \operatorname{sen}(90 - \theta) = -Hl \cos(\theta) \tag{5}$$

T_V, que é o módulo do torque devido à força resultante na direção de y decomposta perpendicularmente em relação à haste, pode ser dado pela seguinte equação 6:

$$T_V = Vl \operatorname{sen}(\theta) \tag{6}$$

Portanto,

$$I\ddot{\theta} = T_V + T_H = Vl \operatorname{sen}(\theta) - Hl \cos(\theta) \tag{7}$$

Sendo I o momento de inércia da haste e $\ddot{\theta}$ a aceleração angular da haste.

O movimento horizontal do centro de gravidade da haste pode ser dado pela equação 8:

$$m\frac{d^2}{dt^2}(x + l \operatorname{sen}\theta) = H \tag{8}$$

O movimento vertical do centro de gravidade da haste pode ser dado pela equação 9:

$$m\frac{d^2}{dt^2}(l\cos\theta) = V - mg \tag{9}$$

O movimento horizontal do carro é descrito pela equação 10:

$$M\frac{d^2x}{dt^2} = u - H \tag{10}$$

Para uma condição de operação em que o valor de θ seja muito próximo de 0°, pode-se aproximar o valor de $\cos\theta = 1$ e $\operatorname{sen}\theta = \theta$. Considerando também que o ângulo terá uma variação muito pequena, pode-se aproximar $\dot{\theta} = 0$. Garantindo essas condições, é possível linearizar as equações 7, 8 e 9 para as seguintes:

$$I\ddot{\theta} = Vl\theta - Hl \tag{11}$$

$$m(\ddot{x} + l\ddot{\theta}) = H \tag{12}$$

$$0 = V - mg \tag{13}$$

Manipulando as equações 10 e 13, obtemos:

$$(M + m)\ddot{x} + ml\ddot{\theta} = u \tag{14}$$

E, manipulando as equações 11, 12 e 13, obtemos:

$$(I + ml^2)\ddot{\theta} + ml\ddot{x} = mlg\theta \tag{15}$$

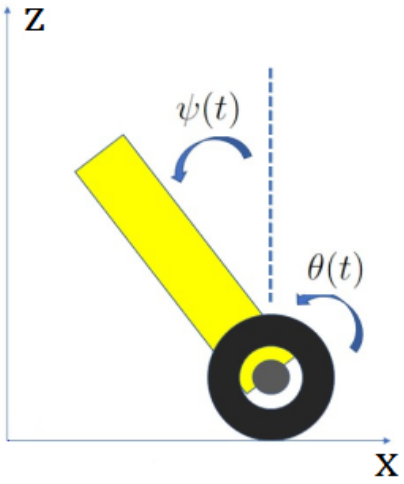
Considerando que nesse sistema a massa possa ser aproximada com uma esfera no topo da haste, o centro de gravidade é o centro da esfera. Para esse caso, o momento de inércia do pêndulo sobre seu centro de gravidade pode ser considerado pequeno e pode ser suposto que $I = 0$ na equação 15. Então, o modelo matemático para esse sistema passa a ser a equação 16.

$$ml^2\ddot{\theta} + mlx\ddot{x} = mgl\theta \tag{16}$$

As Equações 15 e 16 podem ser modificadas para determinar a equação 17 que correlaciona diretamente a entrada do sistema u com a posição angular do pêndulo θ.

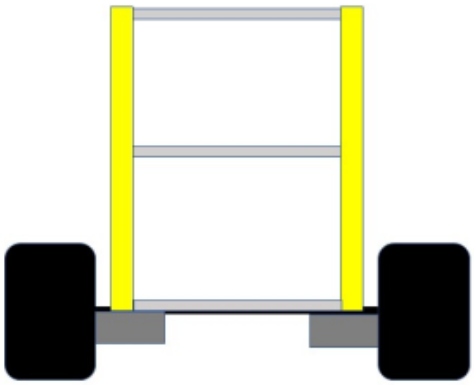
$$Ml\ddot{\theta} = (M + m)g\theta - u \tag{17}$$

Figura 7 – Modelo robô pêndulo invertido sobre duas rodas vista lateral



Fonte: Adaptado de (TEIXEIRA, 2023)

Figura 8 – Modelo robô pêndulo invertido sobre duas rodas vista frontal



Fonte: Adaptado de (TEIXEIRA, 2023)

A equação 17 será reaproveitada na seção 3.3 para determinar o melhor design para a estrutura do robô.

2.3 O Robô Pêndulo Invertido Sobre Duas Rodas

O pêndulo invertido sobre duas rodas consiste em um corpo, que podemos chamar de haste, apoiado sob um eixo que interliga duas rodas de dimensões iguais. Tanto o corpo quanto as rodas possuem liberdade para rotacionar, tanto no sentido horário quanto no sentido anti-horário. Veja a figura 7 e 8.

A dinâmica desse sistema é semelhante à do pêndulo invertido sobre carrinho. A diferença é que no caso do sistema pêndulo invertido sobre duas rodas, a base não é fixa, pois ela faz parte da haste que rotaciona em torno do eixo imaginário que liga o centro das duas rodas. Nesse caso, o torque aplicado na haste devido à movimentação do seu eixo de rotação para a direita e para a esquerda tem uma dinâmica fortemente dependente da posição angular das rodas, representada na figura 7 por $\theta(t)$, e do momento de inércia delas. O momento de inércia é a propriedade que mede quão difícil é fazer uma roda girar mais rápido ou mais devagar ou mudar sua direção de rotação. Todavia, para deslocamentos muito pequenos do eixo de rotação da haste, muitos dos comportamentos do pêndulo invertido sobre duas rodas podem ser aproximados aos do carrinho.

O sistema pêndulo invertido sobre duas rodas, na realidade, tem liberdade para rotacionar tanto em relação ao eixo Z quanto ao eixo X. Todavia, se considerarmos que as

duas rodas giram na mesma velocidade, não haverá rotação em relação ao eixo Z. Por esse motivo, uma análise bidimensional é o suficiente para descrever o modelo dinâmico desse sistema levando essa consideração restritiva.

2.4 Modelagem da Planta, Controlador e Observador pela Metodologia Analítica

Em (TEIXEIRA, 2023), foi definido um método de modelagem de um robô pêndulo invertido real, mostrado na figura 4, por meio da representação desse sistema em espaço de estados.

2.4.1 Modelagem e Simulação em Malha Aberta do Sistema Pêndulo Invertido Sobre Duas Rodas

A metodologia da modelagem resume-se em montar um diagrama de forças para as principais partes mecânicas do sistema e montar circuitos elétricos para as partes elétricas. A partir disso, definir os princípios físicos aplicáveis e as equações matemáticas correspondentes. As principais partes mecânicas do sistema são o corpo do robô e as rodas. A parte elétrica inclui os motores.

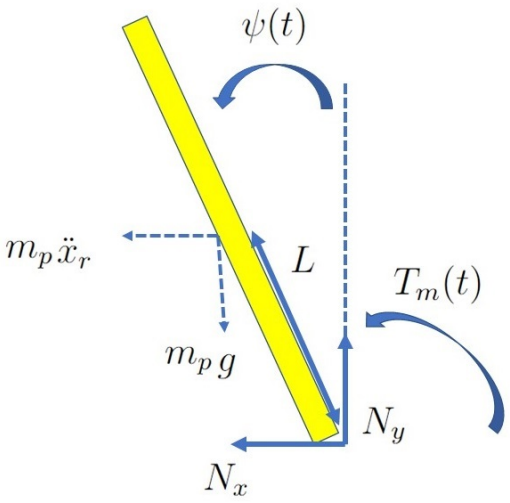
2.4.1.1 Diagrama de Forças do Corpo do Robô

A figura 9 apresenta um diagrama esquemático destacando as forças que atuam sobre o pêndulo, ou seja, o corpo do robô, e a figura 10 apresenta uma vista frontal do pêndulo. O diagrama da figura 9 é desenvolvido tendo como referencial um eixo em coordenadas cartesianas (x, y), sob o qual as variáveis de interesse serão medidas. No diagrama, são destacados os parâmetros $\psi(t)$, posição angular do pêndulo, $T_m(t)$, momento angular sobre o eixo do pêndulo, g, gravidade, m_p, massa do pêndulo, $\ddot{x}_r$, aceleração na direção x da roda, N_x, força entre o pêndulo e a roda na direção x, N_y, força entre o pêndulo e a roda na direção y e L, distância do centro da roda ao centro do pêndulo.

2.4.1.2 Diagrama de Forças das Rodas

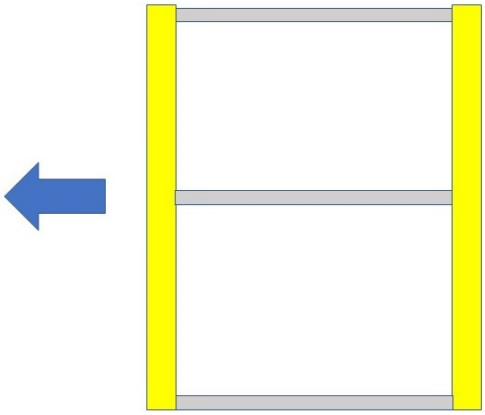
A figura 11 apresenta as forças atuando sobre a roda, e a figura 12 apresenta a vista frontal das rodas, onde $\theta(t)$ é a posição angular da roda, r, raio da roda, g, gravidade, T_m, momento angular sobre o eixo da roda, N, força normal da superfície para com as rodas, N_x, força entre o pêndulo e a roda na direção x, N_y, força entre o pêndulo e a roda na direção y, m_r, massa da roda, $\ddot{x}_r$, aceleração na direção x da roda, e F, fricção entre a superfície e as rodas.

Figura 9 – Vista lateral do pêndulo



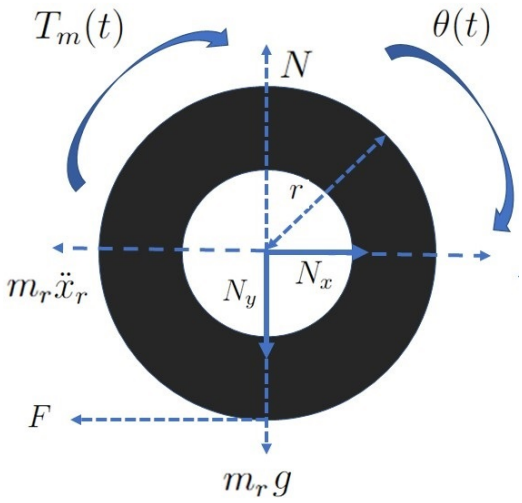
Fonte: (TEIXEIRA, 2023)

Figura 10 – Vista frontal do pêndulo



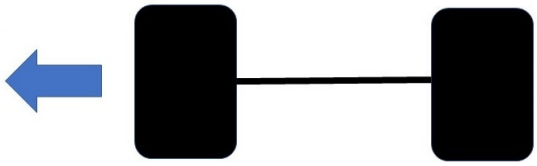
Fonte: (TEIXEIRA, 2023)

Figura 11 – Vista lateral da roda



Fonte: (TEIXEIRA, 2023)

Figura 12 – Vista frontal da roda

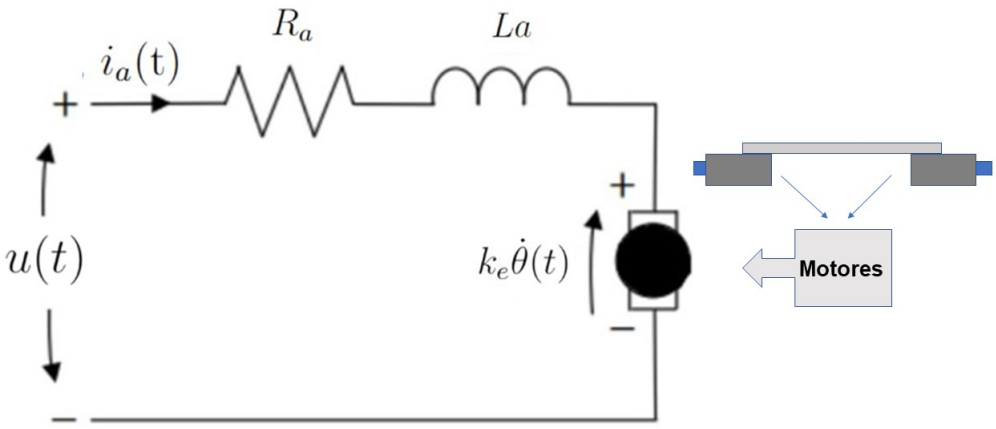


Fonte: (TEIXEIRA, 2023)

2.4.1.3 Circuito Elétrico dos Motores Elétricos DC

Os atuadores do robô serão dois motores elétricos de corrente contínua, que buscarão mantê-lo em equilíbrio a partir da atuação sobre a posição da roda. O circuito elétrico desse motores pode ser representado pelo circuito elétrico da figura 13, onde R_a é a resistência de armadura, L_a, indutância de armadura, k_e, constante da força contra eletromotriz, i_a(t), corrente de armadura, $\dot{\theta}(t)$, velocidade angular do motor, $u(t)$ tensão de entrada do motor.

Figura 13 – Circuito elétrico dos motores



Fonte: (TEIXEIRA, 2023)

2.4.1.4 Modelagem do Sistema Por Espaço de Estados

Para modelar um sistema em espaço de estados é necessário definir quais são as variáveis de estado, as variáveis de saída, de entrada e as equações de estado. As variáveis de estado são as variáveis estrategicamente escolhidas as quais as variáveis de saída possam ser definidas como uma combinação das variáveis de estado e/ou das variáveis de entrada.

As variáveis de estado escolhidas por (TEIXEIRA, 2023) em sua pesquisa foi a posição angular da roda θ, posição angular do pêndulo ψ, velocidade angular da roda ω e velocidade angular do pêndulo ϕ.

O próximo passo foi manipular as equações obtidas com base nos diagramas de força e do circuito elétrico para determinar a física da dinâmica do robô e em seguida as equações de estado que fossem compostas apenas por parâmtros fixos, as variáveis de estado θ, ψ, ω, ϕ e a variável de entrada $u(t)$.

Para respresentar o sistema do robô em espaço de estados as esquações de estado determinadas devem ser escritas de acordo com o formato matricial das Equações 18 e 19.

$$x = \begin{bmatrix} \theta \\ \psi \\ \omega \\ \phi \end{bmatrix}; \quad \dot{x} = \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\omega} \\ \dot{\phi} \end{bmatrix}; \quad f(x,u) = \begin{bmatrix} f_1(\theta,\psi,\omega,\phi,u) \\ f_2(\theta,\psi,\omega,\phi,u) \\ f_3(\theta,\psi,\omega,\phi,u) \\ f_4(\theta,\psi,\omega,\phi,u) \end{bmatrix}; \quad u = \begin{bmatrix} u(t) \end{bmatrix} \tag{18}$$

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\omega} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \\ f_3(x,u) \\ f_4(x,u) \end{bmatrix} \tag{19}$$

Observando a equação 18 temos que x é um vetor formado pelas variáveis de estado. $\dot{x}$ é a derivada do vetor x, $f(x,u)$ é a equação de estado definida como uma função que depende exclusivamente das variáveis de estado e/ou da variável de entrada u. Neste caso temos que $\dot{\theta} = f_1(x,u)$ na equação 19, ou seja, pelo formato da representação em espaço de estados a derivada da variável de estado θ deve ser igual a uma função f_1. O mesmo se aplica para os outros. Tendo em vista isso as funções determinadas na pesquisa de (TEIXEIRA, 2023) foram as seguintes:

$$f_1(x,u) = \omega \tag{20}$$

$$f_2(x,u) = \phi \tag{21}$$

$$\begin{aligned} f_3(x,u) = Lrm_p cos\psi(gLR_a m_p sin\psi + 2nk_t(u - k_e\omega)) + J_p 2nk_t(u - K_e\omega) - \\ J_p LrR_a m_p \phi^2 sin\psi / (R_a(J_p(J_r + r^2(m_p + m_r)) - L^2 r^2 m^2 cos^2\psi)) \end{aligned} \tag{22}$$

$$\begin{aligned} f_4(x,u) = (J_r(gLR_a m_p sin\psi + 2nk_t(u - k_e\omega)) + r(LrR_a m_p sin\psi(m_p(g \\ -L\phi^2 cos\psi) + gm_r) + 2nk_t(u - k_e\omega)(m_p(Lcos\psi + r) + rm_r)))/(R_a(J_p(J_r \\ +r^2(m_p + m_r)) - L^2 r^2 m_p^2 cos^2\psi)) \end{aligned} \tag{23}$$

Sendo k_t a constante de torque, n o fator da caixa de redução, J_p o momento de inércia do pêndulo, J_r o momento de inércia da roda.

Pode-se definir essa representação do sistema do robô em espaço de estados como tendo uma dinâmica de quarta ordem, por possuir quatro variáveis de estado e não linear por que as equações de estado não são lineares. Observa-se também que as esquações de estado dependem de multiplos parâmetros. Todas essas característica do sistema modelado são bastantes desafiadoras para um projeto de controle, sendo necessário uma definição precisa dos parâmetros reais do robô e cuidado no momento da discretização do controlador e observador para implementar no microprocessador. r e observador para implementar no microprocessador.

2.4.1.5 Sistema em Malha Aberta

O gráfico com os resultados da simulação do sinal das variáveis de estados θ, ψ, ω, ϕ em malha aberta pode ser visto na figura 15. Observa-se que o comportamento simulado alcançou as expectativas, pois, para o caso da posição das rodas, era esperado que oscilasse, mas que no final mantivesse sua posição inicial de 0 em condições ideais, como foi simulado. No caso da posição do pêndulo, era esperado que saísse da posição de equilíbrio instável em 0° para a posição de equilíbrio estável de 180°. No caso das velocidades, tanto da roda quanto do pêndulo, era esperado que ao final fossem zero, pois o pêndulo estaria parado. Uma ilustração da movimentação do pêndulo pode ser vista na figura 14.

Tabela 1 – Parâmetros do robô pêndulo invertido sobre rodas

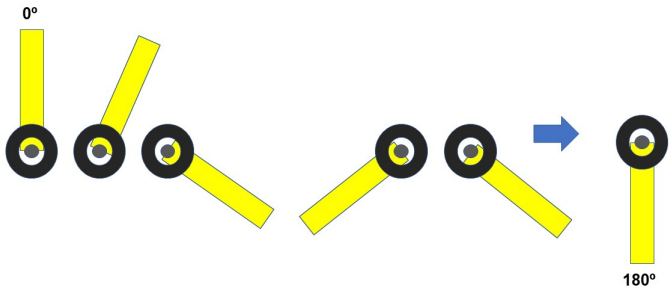
Parâmetros	**Unidade**	**Valor**
g	m/s^2	$9,81$
L	m	$0,12$
h	m	$0,21$
m_p	kg	$1,2$
J_p	kgm^2	$0,0191$
m_r	kg	$0,036$
J_r	kgm^2	$0,000017298$
r	m	$0,031$
Vn	V	12
R_a	Ω	$1,7$
I_a	A	$0,4$
N	rpm	280
T_m	Nm	$0,044$
K_e	Vs/rad	$0,39$
K_t	Nm/A	$0,11$

Fonte: (TEIXEIRA, 2023)

O gráfico com os resultados da simulação do sinal das variáveis de estados θ, ψ, ω, ϕ em malha aberta pode ser visto na figura 15. Observa-se que o comportamento simulado alcançou as espectativas pois, para o caso da posição das rodas era esperado que oscilase mas que no final mantivesse sua posição inicial de 0 em condições ideais como foi simulado. No caso da posição de pêndulo era esperado que saisse da posição de equilíbrio instável em 0° para a poisção de equilíbre estável de 180°. No caso das velocidades tanto da roda quando do pêndulo erá esperado que ao final fosse zero pois o pêndulo estaria parado. Uma ilustração da movimentação do pêndulo pode ser visto na figura 14

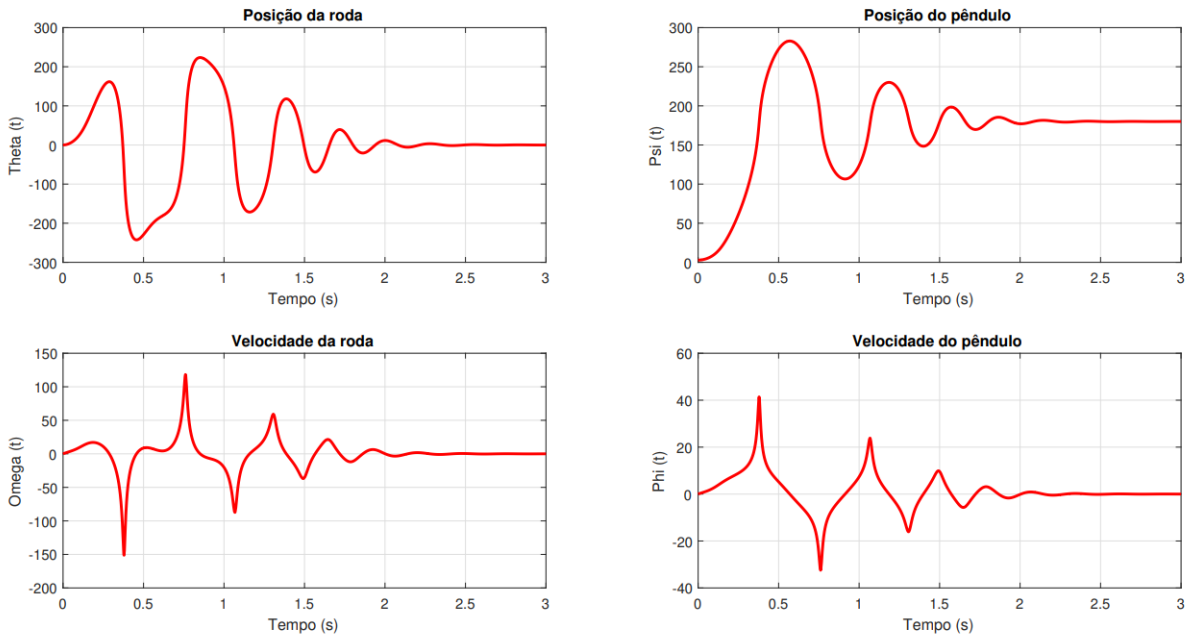
O método de controle por alocação de polos no espaço de estado determina que, se a planta for controlável, é possível escolher o ponto onde se deseja alocar os polos através da realimentação negativa das variáveis de estados multiplicadas por um ganho K. Quando não se tem acesso às medidas de todas as variáveis de estado, é possível utilizar um observador de estado que estima o valor das variáveis de estado com base nos valores

Figura 14 – Simulação pêndulo robô malha aberta



Fonte: (TEIXEIRA, 2023)

Figura 15 – Gráfico de análise de Variáveis

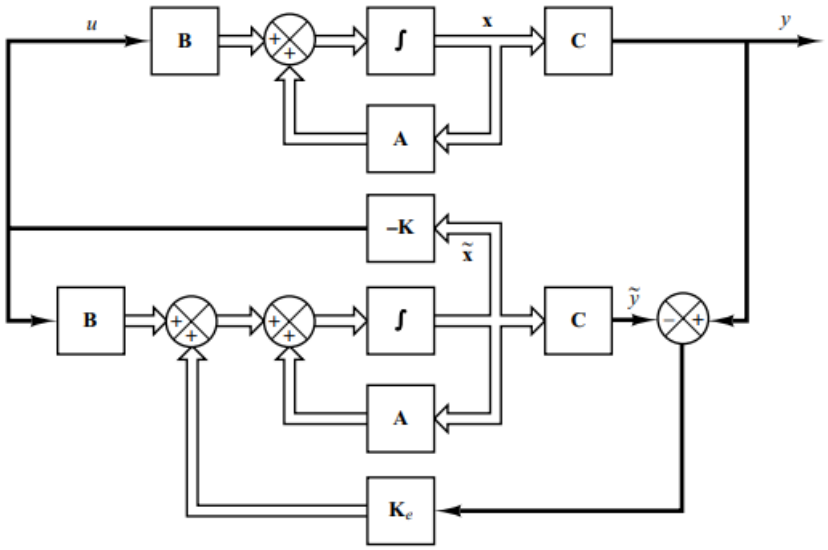


Fonte: (TEIXEIRA, 2023)

de saída e entrada da planta. Um diagrama de blocos com o controlador K em conjunto com um observador pode ser visto na figura 16.

O modelo do robô de quarta ordem, conforme mencionado por (TEIXEIRA, 2023), é controlável, mas não é observável. Neste trabalho, considerando apenas a posição do pêndulo ϕ como conhecida, é imprescindível que seja possível observar os outros estados. No entanto, foi constatado que um modelo mais simples de terceira ordem, obtido suprimindo a variável de estado posição das rodas θ, é controlável e observável. A supressão dessa variável faz sentido, pois as equações de estado 21, 23 e 23 não dependem dessa variável; logo, a expectativa é que sua supressão não deva alterar a dinâmica do sistema. As técnicas

Figura 16 – Sistema de controle com retroalimentação do estado observado



Fonte: (OGATA et al., 2010)

de controle por alocação de polos em espaço de estados demandam que as equações de estado sejam lineares. Uma técnica de linearização muito eficiente é a expansão em série de Taylor. Essa técnica preconiza que é possível determinar uma equação linear que se aproxima muito bem da equação não linear em uma região delimitada ou ponto de operação. Para o caso do robô, é possível encontrar uma equação linear que se aproxima bastante da dinâmica não linear para a região próxima do ponto de operação de 0°. A linearização foi realizada no Matlab utilizando a função jacobian(). Assim, foram determinadas as Matrizes A', B', C' e D' de acordo com a equação 27 da estrutura convencional de um sistema linear e invariante no tempo no espaço de estados, conforme as Equações 24 e 25.

$$\dot{x} = Ax + Bu \tag{24}$$

$$y = Cx + Du \tag{25}$$

$$x = \begin{bmatrix} \psi \\ \omega \\ \phi \end{bmatrix} \tag{26}$$

$$A' = \begin{bmatrix} 0 & 0 & 1 \\ 2056,83 & -18,01 & 0 \\ 555,26 & -4,34 & 0 \end{bmatrix} \quad B' = \begin{bmatrix} 0 \\ 46,66 \\ 11,23 \end{bmatrix} \quad C' = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad D' = \begin{bmatrix} 0 \end{bmatrix} \tag{27}$$

Para os requisitos de projeto, buscou-se especificar um par de polos complexos conjugados como sendo os polos dominantes, para os quais foi projetado $t_s = 1,5s$ e $\zeta = 0,8$. Esses requisitos resultam na alocação dos polos para $s_{1,2} = -2,67 \pm j2$. O terceiro polo foi definido como $s_3 = -60$, resultando nos seguintes ganhos do controlador:

$$K = [78,8977 \quad -0,6242 \quad 6,8045] \tag{28}$$

Para determinar os ganhos do observador, utilizou-se a abordagem convencional de tomar polos mais rápidos que os polos do controlador da planta. Isso é necessário para que as variáveis observadas convirjam mais rapidamente para as variáveis reais do sistema e para que estas sejam antecipadas apropriadamente para a atuação do controlador. Os polos do observador serão quatro vezes mais rápidos que os polos do controlador. Isso é garantido simplesmente multiplicando a parte real dos polos de projeto do controlador por um fator de 4. Assim, os polos determinados foram $s_{1,2} = -10.6667 \pm j2$ e $s_3 = -240$. Desse modo, o vetor de polos desejados foi especificado como:

$$s = [s_1, s_2, s_3] \tag{29}$$

Com os polos definidos, procede-se então com a determinação dos ganhos do observador, diretamente pela fórmula de Ackermann, que é codificada em Matlab como $L = \text{acker}(A', C', s)'$, então:

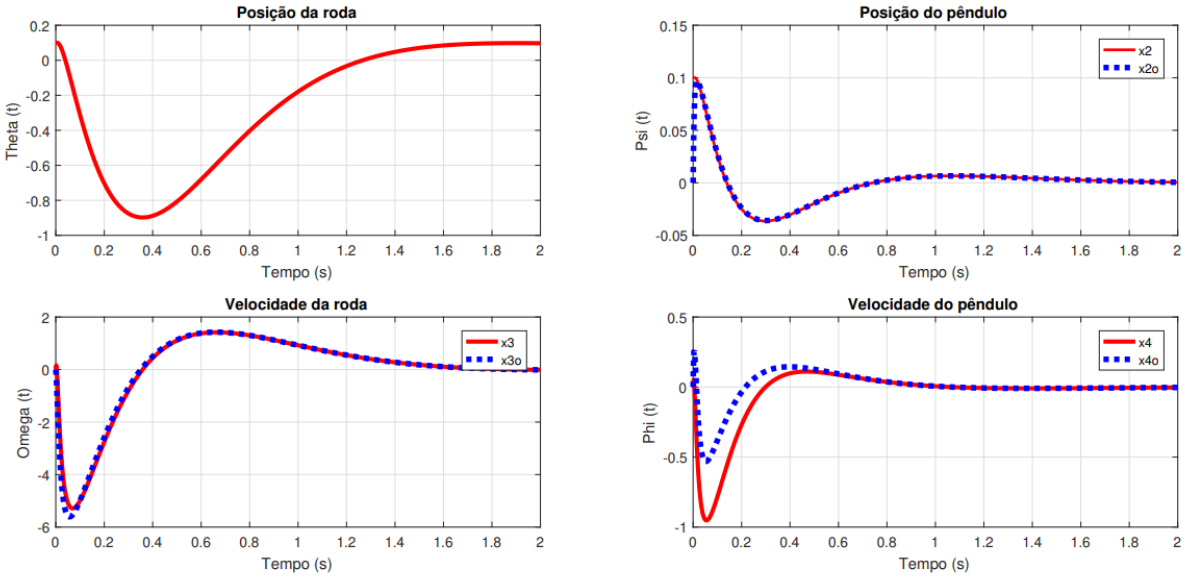
$$L = \begin{bmatrix} 243,32 & -909,83 & 1410,20 \end{bmatrix} \tag{30}$$

O resultado da simulação do sistema com o controlador e observador definidos foi o da figura 17.

2.5 Controle de Sistemas Dinâmicos

O controle de sistemas dinâmicos lida com a gestão e regulação de sistemas que evoluem ao longo do tempo. O objetivo dele é garantir que o sistema atinja um desempenho desejado ou mantenha seu comportamento dentro de limites específicos). Exemplos comuns de variáveis controladas são temperatura, pressão, vazão, tensão, corrente, posição, velocidade e inclinação. Existem dois tipos de configuração de sistemas de controle. Sistemas em malha fechada e em malha aberta. Os sistemas em malha aberta são um tipo de sistema de controle em que a saída do sistema não é usada para realimentação e ajuste contínuo do processo. Nesse tipo de sistema, as ações de controle são pré-programadas e

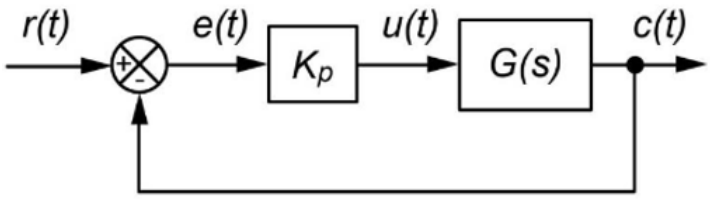
Figura 17 – Gráfico de análise do observador de estado



Fonte: (TEIXEIRA, 2023)

não dependem do feedback ou das condições reais da saída. Um exemplo disso são fornos elétricos mais simples, pois inicialmente é defina uma temperatura desejada e o tempo de cozimento, mas o forno não faz ajustes com base na temperatura real do forno ou nas condições de cozimento. Já os sistemas em malha fechada são do tipo de sistema de controle em que a saída do sistema é continuamente monitorada e usada como feedback para ajustar o comportamento do sistema, a fim de manter a saída o mais próxima possível de um valor de referência desejado. Um exemplo desse sistema em malha fechada é o do controle de posição de pêndulo invertido sobre duas rodas, pois para que seja possível fazer o controle da posição angular do pêndulo para mantê-lo sempre na vertical é necessário medir continuamente sua posição e compara essa posição com o valor desejado que é 0°. Um diagrama de blocos de um sistema em malha fechada pode ser representado da seguinte forma:

Figura 18 – Sistema em malha fechada



Fonte: adaptado de (OGATA et al., 2010)

A figura 18 traz a representação de um sistema com realimentação, onde $G(s)$ é a planta, ou o sistema a ser controlado; $r(t)$ é o sinal de referência, também chamado de

setpoint; $c(t)$ é a variável controlada, ou de saída; e $u(t)$ é o sinal de controle, ou variável manipulada.

Nos sistemas de malha fechada, a variável controlada (que é o valor real medido) é comparada com um valor de referência (que é o valor desejado), e dessa comparação (valor desejado − valor real) é obtido o erro $e(t)$, que é o sinal de entrada do controlador que gera a ação de controle. Isso implica que o estado atual do sistema interfere na ação de controle futura. Sempre que há erro não nulo, esse estado define para o controlador que uma ação de controle deve processar o erro para eliminá-lo ou minimizá-lo.

2.6 Controlador PID

Segundo (ÅSTRöM; HäGGLUND, 1995), o controlador PID é um mecanismo de controle em malha fechada amplamente utilizado em sistemas de controle industrial. Conforme (OGATA et al., 2010) na área dos sistemas de controle de processos, sabe-se que os esquemas básicos de controle PID provaram sua utilidade, conferindo um controle satisfatório, embora em muitas situações eles possam não proporcionar um controle ótimo. O controlador PID calcula o erro $e(t)$ entre um valor medido $c(t)$ e um valor desejado $r(t)$, gerando uma ação de controle $u(t)$ de acordo com a figura 18. Essa ação de controle em tempo contínuo possui uma lógica em sua forma padrão com base na equação 31.

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\, d\tau + K_d \cdot \frac{de(t)}{dt} \tag{31}$$

No domínio da frequência, aplicando a transformada de Laplace na equação 31 e considerando condições iniciais nulas, obtém-se a equação 32.

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \tag{32}$$

Onde K_p é o ganho proporcional, T_i o tempo integrativo e T_d o tempo derivativo.

A equação 31 expressa que a ação de controle será ponderada por três constantes, uma que pondera o erro proporcional, outra que pondera o somatório dos erros e outra que pondera a taxa de variação do erro. Isso quer dizer que a constante K_p dá um peso proporcional ao erro, a constante K_i um peso ao acúmulo de erros passados e a constante K_d peso a uma expectativa de crescimento ou decréscimo do erro no futuro. O processo de selecionar parâmetros K_p, K_i e K_d do controlador que garantam dada especificação de desempenho é conhecido como sintonia do controlador.

De acordo com (ÅSTRöM; HäGGLUND, 1995), existem diversas formas de sintonizar um controlador PID para alcançar o controle automático desejado. Quando o modelo matemático da planta é conhecido, as constantes podem ser determinadas de forma analítica pelo método de resposta em frequência. Porém, existem casos em que

os parâmetros determinados de forma analítica não atendem aos requisitos de projeto em plantas reais, pois a solução definida está restrita a considerações ideais do sistema. Nesse caso, podem ser aplicados métodos de sintonia experimental, que podem ser guiados pelo modelo matemático para determinar uma sintonia inicial do PID. Quando não se conhece o modelo matemático da planta por motivos como o sistema ser muito complexo de representar matematicamente, envolver muitas variáveis, ter comportamento não linear ou ser de natureza instável, uma alternativa para a sintonia inicial do controlador PID pode ser utilizar um método experimental. Os métodos mais conhecidos incluem o método de resposta ao degrau de Ziegler-Nichols, o método de resposta em frequência de Ziegler-Nichols, o método de Chien-Hrones-Reswick, o método de Cohen-Coon, o método da integral do erro e a sintonia por relé. Os métodos de sintonia experimentais são úteis para definir parâmetros do controlador PID que podem proporcionar um controle estável com requisitos de projeto predefinidos pelos seus criadores. Em muitos casos, esses métodos não proporcionam um controle ótimo, mas podem ser utilizados como ponto de partida para o refinamento do controle.

De acordo com (TEIXEIRA et al., 2010), o método de Ziegler-Nichols pode ser aplicado na sintonia do controlador PID para sistemas com resposta ao degrau sigmoideal (curva ou função que se assemelha a uma forma de "S") em malha aberta ou que em malha fechada possa ser induzida uma resposta oscilatória sustentada através do aumento gradual de um ganho K na malha direta. O método de sintonia de controlador PID de Chien-Hrones-Reswick, de Cohen-Coon e integral do erro são aplicáveis quando a resposta ao degrau do sistema em malha aberta é sigmoideal ou quando pode ser definido um modelo matemático de primeira ordem que simule de forma aproximada a dinâmica da planta. O método de sintonia por relé pode ser aplicado para sistemas nos quais pode ser induzida uma resposta oscilatória em malha fechada com o auxílio de um ganho K em paralelo com um relé na malha direta.

Aproveitando o modelo matemático em espaço de estados de terceira ordem do robô visto na sessão 2.4, foi constatado que ele não se estabiliza em malha aberta, logo não apresenta uma resposta sigmoideal e não é possível criar um modelo matemático de primeira ordem que simule esse sistema de maneira aproximada. Portanto, os métodos de Chien-Hrones-Reswick, de Cohen-Coon e integral do erro, bem como o método de resposta ao degrau em malha aberta de Ziegler-Nichols, não são aplicáveis.

Foi realizado um experimento ao fechar a malha do robô real com uma constante K na malha direta, a qual foi sendo aumentada gradativamente até um ponto em que foi observada uma oscilação constante. Portanto, o segundo método de Ziegler-Nichols é aplicável conforme o resultado experimental.

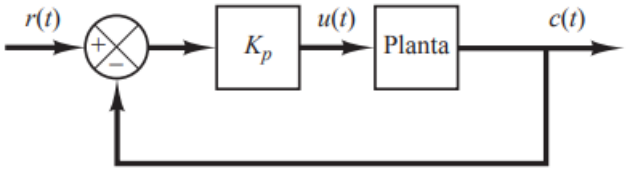
De acordo com (PINTO, 2014), Aström e Háglund propuseram um método de sintonia de controladores que utiliza "relés"em malha fechada. O objetivo é provocar

oscilações controladas e limitadas no processo para estimar a resposta em frequência da planta a partir dessas oscilações. Este método é conceitualmente semelhante ao proposto por Ziegler e Nichols para obter o ganho crítico e o período crítico. Uma vantagem é que este método é mais controlado, pois limita a amplitude da perturbação no processo. No entanto, neste trabalho, não foi necessário adotar esse método específico, uma vez que foi possível fazer o robô oscilar sem recorrer ao uso de um relé na malha direta, e a partir da resposta, determinar o ganho e o período crítico.

2.7 Método de Resposta em Frequência Ziegle-Nichols para Sintonia de Controlador PID

Tendo em vista que o método de resposta em frequência de Ziegler-Nichols é o mais adequado para os objetivos deste trabalho, nesta seção será explicado como ele funciona. De acordo com (OGATA et al., 2010), esse método é útil quando o modelo matemático da planta é desconhecido, mas também pode ser aplicado quando é conhecido. O método sugere valores para K_p, T_i e T_d que podem proporcionar uma operação estável do sistema. Contudo, o sistema resultante pode exibir um sobressinal elevado na resposta ao degrau, o que não é desejável. Nesse caso, é preciso realizar uma série de sintonias finas até que um resultado aceitável seja obtido. Esse método é adequado apenas para sistemas nos quais, quando colocados em malha fechada com um ganho K na malha direta, é possível induzir uma resposta oscilatória sustentada através do aumento gradativo do ganho K. O sistema em malha fechada pode ser representado pela figura 19, e a resposta oscilatória sustentada pode ser observada na figura 20.

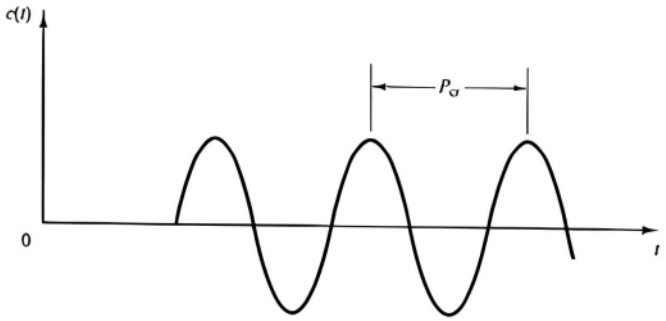
Figura 19 – Sistema de malha fechada com um controlador proporcional.



Fonte: (OGATA et al., 2010)

Chegando no ponto de oscilação sustentada o K é definido como o ganho crítico do sistema K_{cr} e o perído de oscilação é o período crítico P_{cr}. Segundo (PINTO, 2014), tendo essas duas informações basta usar a Tabela 2 proposta por Ziegle-Nichols para determinar K_p, T_i e T_d, usando um critério de desempenho uma razão de declínio do sobressinal igual a um quarto 1/4. Na prática esse teste pode levar o processo a variar fora de uma região segura, podendo causar instalilidade do sistema. Por isso é necessário ter o cuidado de garantir qual é a região segura antes de começar os testes.

Figura 20 – Resposta oscilatória sustentada do sistema



Fonte: (OGATA et al., 2010)

Tabela 2 – Tabela para definição das constante do controlador pelo método ZN

Tipo de controle	k_p	T_i	T_d
P	0,5 k_{cr}	∞	0
PI	0,45 k_{cr}	$\frac{1}{1,2}P_{cr}$	0
PID	0,6 k_{cr}	0,5 P_{cr}	0,125 P_{cr}

A equação padrão do controlador PID pode ser vista na equação 32. Para transformar essa equação no formato que será implementado no microcontrolador a seguinte equação é determinada:

$$C(s) = K_p + \frac{K_p}{T_i s} + K_p T_d s \tag{33}$$

Portanto as equações para definir as constantes do controlador PID em tempo contínuo requeridas são:

$$K_p = K_p \tag{34}$$

$$K_i = \frac{K_p}{T_i} \tag{35}$$

$$K_d = K_p T_d \tag{36}$$

Para determinar o valor das constantes em tempo discreto basta ajustar os ganhos de tempo contínuo acrescentando o período de amostragem T_s do controlador de acordo com as Equações 37, 38 e 39.

$$K_{pd} = K_p \tag{37}$$

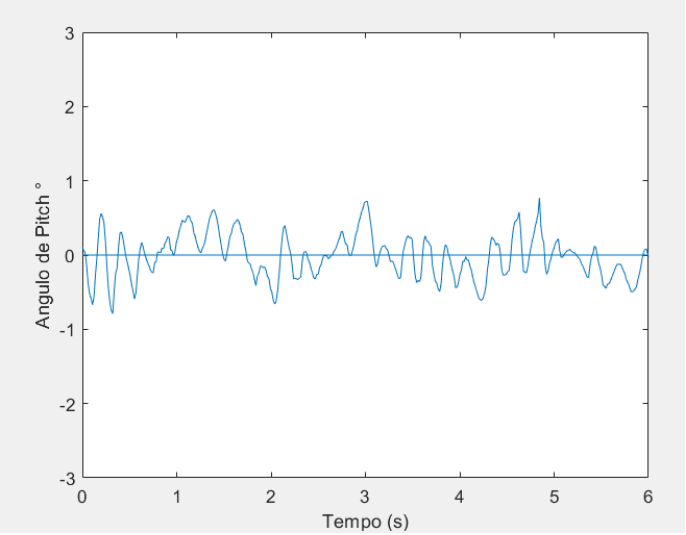
$$K_{id} = \frac{K_p}{T_i} T_s \tag{38}$$

$$K_{dd} = \frac{K_p T_d}{T_s} \tag{39}$$

Uma das maiores dificuldades de implementação do método de resposta em frequência de Ziegle-Nichols é determinar o P_{cr} pois em uma situação ideal a resposta oscilatória sustentada do sistema tem formato padrão como da figura 20, a qual é possível determinara facilmente o período do sinal. Todavia na realidade o sinal coletado normalmente não é tão regular podendo se parecer com a resposta da figura 21. Por essa figura é possível observar uma certa periodicidade, porem a forma se encontra muito longe da ideal da figura 20. Nesse caso para obter o período crítico P_{cr} basta imaginar que o sinal da figura 21 é o resultado da soma de várias funções senoidais com valores de frequência das mais variadas, porém deve haver uma dessas funções senoidais com contribuição mais significativa para a construção do sinal. A figura 22 exemplifica esse conceito.

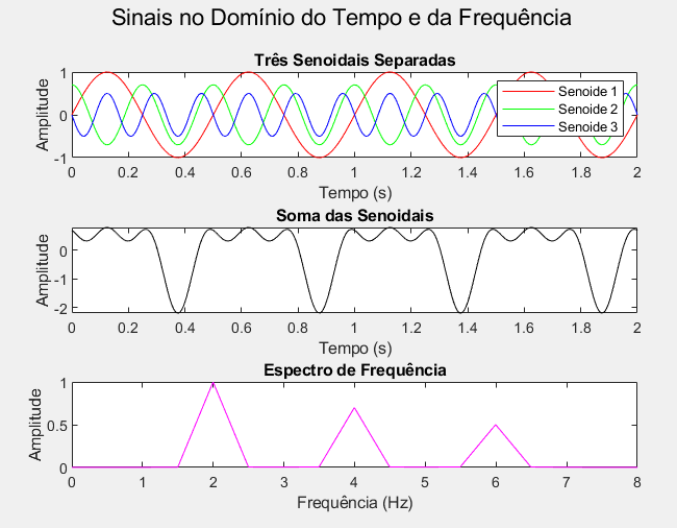
Pela figura 22 tem-se que o sinal preto é formado pela soma das funções senoidais vermelha, verde e azul. Visualmente é possível determinar que a função que mais tem contribuição na contrução do sinal preto é a da linha vermelha por possuir maior amplitude. Na situação em que se tem apenas o sinal preto é possível determinar a frequência das funções senoidais que o compõe utilizando a transformada de Fourier. Essa transforma retona o gráfico com a linha lilás. Nesse grafico é possível determinar que existem 3 frequências de funções senoidais que são dominantes na formação do sinal preto, uma na frequência 2 Hz, outra em 4 Hz e outra em 6 Hz. Além disso o gráfico determina qual é a amplitide de cada um desses sinais. Nesse caso o sinal com maior amplitude em 2 HZ é justamente a amplitude do sinal vermelho. Tendo determinado a frequência desse sinal vermelho é possível determinar o período dele.

Figura 21 – Resposta do robô em malha fechada para ganho 0,95 e período de amostragem de 15 ms



Fonte: Do Autor

Figura 22 – Exemplo da aplicação da transformada de Fourier



Fonte: Autor

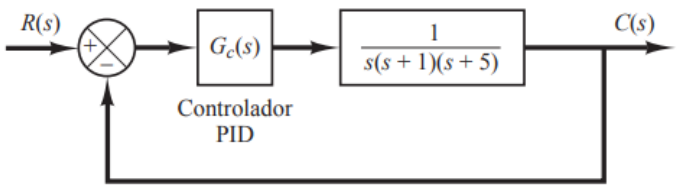
Portanto é possível determinar o período critico de uma resposta real de um sistema utilizando a transformada de Fourrier para determiara a frequência da senoide dominante na formação do sinal e a partir dessa frequência determinar o período crítico do sinal completo.

2.7.1 Exemplo da Aplicação do Método de Resposta em Frequência de Ziegle-Nichols.

De acordo com (OGATA et al., 2010), considerando um sistema de controle mostrado na figura 23 no qual um controlador PID é utilizado para controlar o sistema, a função de transferência desse controlador pode ser expressada pela equação 40.

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \tag{40}$$

Figura 23 – Sistema com controle PID



Fonte: (OGATA et al., 2010)

Será aplicado a regra de sintonia de resposta em frequência de Ziegler-Nichols na determinação dos parâmetros Kp, Ti e Td. Em seguida, será obtido a curva de resposta ao degrau unitário.

Como a planta tem um integrador, utilizamos o de resposta em frequêndia de sintonia de Ziegler-Nichols. Fazendo $T_i = \infty$ e $T_d = 0$, obtemos a função de transferência de malha fechada é da equação 41.

$$\frac{C(s)}{R(s)} = \frac{K_p}{s(s+1)(s+5) + K_p} \tag{41}$$

O valor de Kp que torna o sistema marginalmente estável, de modo que ocorram oscilações sustentadas, pode ser obtido pelo uso do critério de estabilidade de Routh. Uma vez que a equação característica do sistema em malha fechada é o da equação 44.

$$s^3 + 6s^2 + 5s + K_p = 0 \tag{42}$$

o arranjo de Routh pode ser visto no quadro 1.

Quadro 1 – Arranjo de Routh

s^3	1	5
s^2	6	K_p
s^1	$\frac{30 - K_p}{6}$	
s^0	K_p	

Examinando os coeficientes da primeira coluna da tabela de Routh, determinamos que oscilações sustentadas existirão se $K_p = 30$, pois pra valores maiores de 30 vai haver uma mudança de sinal, e cada mundaça de sinal indica o número de polos com parte real positiva. Portanto, o valor crítico K_{cr} é:

$$K_{cr} = 30 \tag{43}$$

Com o ganho K_p igual a $K_{cr} = 30$, a equação característica resulta em:

$$s^3 + 6s^2 + 5s + 30 = 0 \tag{44}$$

Para encontrar a frequência da oscilação sustentada, substituímos $s = j\omega$ na equação característica, como segue:

$$j\omega^3 + 6(j\omega)^2 + 5j\omega + 30 = 0 \tag{45}$$

Ou

$$6(5-\omega^2)+j\omega(5-\omega^2)=0 \tag{46}$$

A partir da qual determinamos a frequência da oscilação sustentada como $\omega^2=5$ ou $\omega=\sqrt{5}$. Logo, o período de oscilação sustentada é:

$$P_{cr}=\frac{2\pi}{\omega}=\frac{2\pi}{\sqrt{5}}=2,8099 \tag{47}$$

Referindo-nos à Tabela 2, determinamos K_p, T_i, e T_d como segue:

$$\begin{aligned} K_p &= 0,6K_{cr}=18 \\ T_i &= 0,5P_{cr}=1,405 \\ T_d &= 0,125P_{cr}=0,35124 \end{aligned}$$

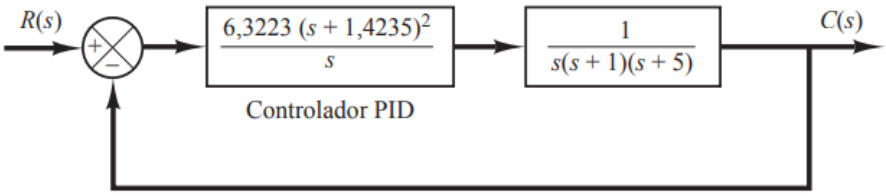
A função de transferência do controlador PID é, portanto,

$$G_c(s)=K_p\left(1+\frac{1}{T_i s}+T_d s\right)=18\left(1+\frac{1}{1,405s}+0,35124s\right) \tag{48}$$

$$G_c(s)=\frac{6,3223(s+1,4235)^2}{s} \tag{49}$$

O controlador PID tem um polo na origem e um zero duplo em $s=-1.4235$. Um diagrama de blocos do sistema de controle com o controlador PID projetado é mostrado na figura 23

Figura 24 – Diagrama de blocos do sistema com o controlador PID projetado com o uso da regra de sintonia de resposta em frequência de Ziegler-Nichols.



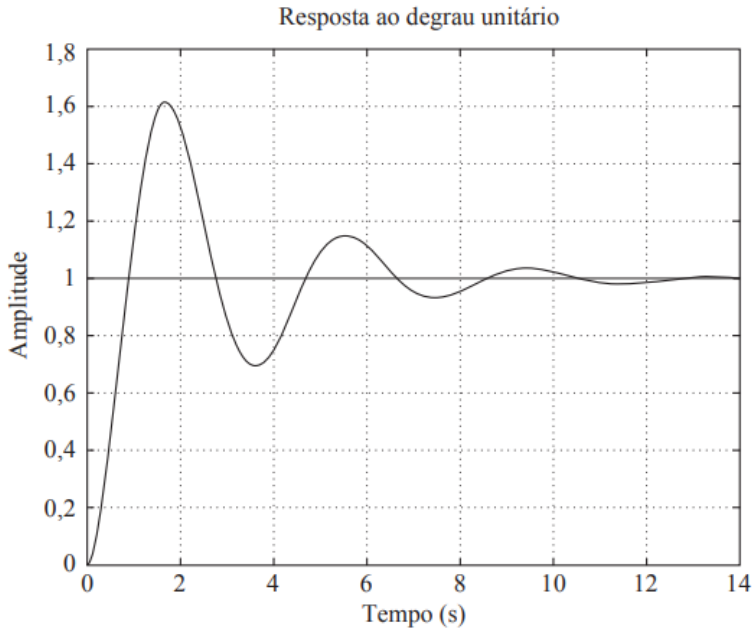
Fonte: (OGATA et al., 2010)

Em seguida, vamos examinar a resposta do sistema ao degrau unitário. A função de transferência C(s)/R(s) é dada por:

$$\frac{C(s)}{R(s)}=\frac{6,3223s^2+18s+12,811}{s^2+6s^3+11,3223s^2+18s+12,811} \tag{50}$$

A resposta ao degrau unitário desse sistema obtida com o MATLAB foi da figura 25

Figura 25 – Curva de resposta ao degrau unitário de um sistema com controlador PID projetado com o uso da regra de sintonia de resposta em frequência de ZieglerNichols



Fonte: (OGATA et al., 2010)

A resposta ao degrau unitário da figura 25 demonstra que foi possível estabilizar o sistema usando um controlador PID sintonizado pelo método de ZN. Inclusive, observa-se um decaimento do primeiro sobressinal para o segundo de aproximadamente 1/4, corroborando a proposta do método. No entanto, a resposta do sistema apresenta um sobressinal máximo de aproximadamente 62%, valor que é considerado excessivo para muitos projetos. Portanto, esse método normalmente é utilizado para realizar uma sintonia inicial e, posteriormente, realizar uma sintonia fina utilizando outros métodos para otimizar esse resultado.

2.8 Sensores Inerciais

Tanto para implementação na prática dos métodos de controle analíticos quanto os experimentais necessitam de medidas precisas da variável que se deseja controlar, que no caso desse trabalho é a posição angular do pêndulo. Essas medidas são feitas através de sensores que normalmente convertem defirentes grandezas físicas em sinais elétricos interpretáveis por microprocessadores. Os sensores que serram utilizados nesse trabalho serão sensores inerciais, também conhecidos como sensores de inércia, que são dispositivos eletrônicos utilizados para medir a aceleração e a orientação de um objeto em relação ao espaço tridimensional. Eles possuem aplicações diversas, desde sistemas de navegação em veículos e aeronaves até dispositivos de realidade virtual, drones, robótica e até mesmo em

dispositivos de controle de movimento em eletrônicos de consumo, como smartphones e tablets(KADAM; PAWAR, 2020).

A a popularização dessas aplicações é atribuída ao avanço dos sistemas micro-eletromecânicos (MEMS), e à sua integração com sensores inerciais acessíveis, como acelerômetros e giroscópios. Estes sensores agora são produzidos e disponibilizados em dispositivos conhecidos como Unidades de Medição Inercial (IMUs), normalmente compostos por um sensor acelerômetro e giroscópio (KADAM; PAWAR, 2020). A figura 37 demonstra o IMU utilizado nesse trabalho.

Os sensores inerciais consistem em dois tipos principais. Acelerômetros e giroscópios. Os acelerômetros são capazes de detectar acelerações, quando submetidos a forças de qualquer espécie, enquanto os giroscópios percebem variações angulares e rotações. No projeto de robô pêndulo invertido sobre duas rodas será necessário utilizar sensores inerciais para medir a inclinação do pêndulo. Para obter um valor preciso dessa inclinação será necessário passar por três etapas de processamento dos sinais dos sensores. Cálculo de postura, filtragem e fusão de sensores.

Segundo (KOK; HOL; SCHöN, 2017) o cálculo de postura é uma forma de usar os sinais dos sensores para determinar as coordenadas de posição e orientação (geralmente referidas como pose) de um sistema. Para o robô pêndulo invertido esse cálculo é feito para medir a orientação do pêndulo, ou seja, o seu valor de inclinação θ.

Entre as abordagens mais comuns de cálculo de postura estão (KOK; HOL; SCHöN, 2017):

a) Ângulos de Euler;

b) Ângulos de Rolagem, Inclinação e Guinada – Roll, Pitch e Yaw;

c) Matriz dos Cossenos Diretores;

d) Quaternions.

Dentre esses métodos o de ângulos de Rolagem, Inclinação e Guinada – Roll, Pitch e Yaw é a abordagem que tem menor custo computacional, com algoritmo de fácil compreensão e implementação, além de ser adequado para medir a variável de interresse utilizando um sensor acelerômetro.

Os ruídos e bias medidos pelos sensores podem distorcer o valor calculado da postura do sistema. Para resolver esse problema é usado a técnica de filtragem dos sinais e fusão de sensores. A técnica de filtragem consiste em eliminar componentes de frequência que tornam a medição da variável de interesse menos precisa. A técnica de fusão de sensores consiste em utilizar mais de um sensor para estimar um valor mais preciso da

variável de interesse. Para o robô essas técnicas foram utilizadas para estimar o valor de inclinação do pêndulo θ utilizando um sensor acelerômetro e giroscópio (KOK; HOL; SCHöN, 2017).

Para filtragem e fusão de sensores, as técnicas mais comuns são (KOK; HOL; SCHöN, 2017):

a) Filtro Complementar;

b) Filtro de Kalman

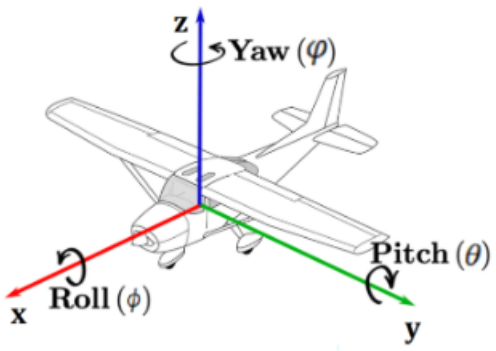
O Filtro de Kalman é capaz de calcular o erro de cada medida a partir das medidas anteriores, eliminando o erro e fornecendo o valor real do ângulo. No entanto, ele tem um custo de processamento bastante elevado e é bastante complexo, razão pela qual foi decidido utilizar o filtro complementar que é igualmente útil para o caso, mais simples e ideal para implementação com Arduino.

As próximas seções tratam do cálculo de postura, do processo de filtragem e da fusão dos sensores.

2.8.1 Postura

O método dos ângulos de Rolagem, Inclinação e Guinada – Roll, Pitch e Yaw usa como referencial fixo os eixos X, Y e Z do sistema de coordenadas cartesiano para representar a orientação de um objeto em um espaço tridimensional. De acordo com a figura 26 o ângulo de rolamento *Roll* se refere à rotações em torno do eixo de rolagem x, o ângulo de inclinação *Pitch* se refere à rotações em torno do eixo de inclinação y, e o ângulo de guinada *Yaw* se refere à rotações em torno do eixo de guinada z. Esses três ângulos juntos permitem uma descrição completa da orientação espacial de um objeto em relação a um sistema de coordenadas cartesiano fixo.

Figura 26 – Definição dos ângulos Roll, Pitch e Yaw.



Fonte: Adaptado de (DEGOND; DIEZ; NA, 2021)

Os ângulos de *Pitch*, no caso desse trabalho se refere a posição angular do pêndulo θ, pode ser obtido das medidas instantâneas de aceleração do sensor inercial do tipo acelerômetro segundo a equação 51:

$$\theta = \tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \tag{51}$$

A_x, A_y e A_z são as medidas das acelerações nos eixos x, y e z do sistema de referência do sensor.

Uma observação é que o ângulo de *Yaw* não pode ser calculado da medição direta do acelerômetro, porque o eixo z é paralelo à ação da gravidade, e qualquer rotação no plano xy não produz efeito sobre o Az. Nas aplicações onde é necessário o cálculo do *yaw*, faz-se a associação do sensor inercial com outros sensores, como os magnetômetros.

2.8.2 Filtragem

Segundo (KOK; HOL; SCHöN, 2017) os sinais de sensores acelerômetro e de giroscópio normalmente apresentam ruído e viés. Por esse motivo é necessário que os sinais medido pelos sensores passem por uma etapa de filtragem antes de serem utilizadas para fazer os cálculos. O ruído é uma interferência indesejada ou aleatória que é adicionada a um sinal que torna as medições menos precisas ou confiáveis. O viés refere-se a um deslocamento sistemático ou erro sistemático em um sinal ou medição. Ele é uma constante que é adicionada ao sinal ou à medição, geralmente devido a imperfeições ou calibração incorreta do sistema. A figura 27 exemplifica o que são esses conceitos onde em um jogo de dardos em que os arremessos se concentram no centro do alvo é possível comparar com um sinal que não possui ruído ou viés. Já quando os arremessos se concentram em um ponto que não é o alvo central é possível comparar com um sinal sem ruído com viés. Já quando os arremessos são muito dispersos é possível comparar com um sinal com ruído sem viés. Na situação em que os arremessos possuem uma dispersão considerável e uma concentração também considerável em um ponto que não é o centro do alvo seria como um sinal com ruído e viés.

2.8.2.1 Ruído e Viés de Medições de Sensores Inerciais Acelerômetro e Giroscópio

As medições de sensores inerciais e seus erros podem ser exemplificados pelo experimento realizado segundo (KOK; HOL; SCHöN, 2017). Na figura 28 com a medição do giroscópio e a figura 29 com a medição do acelerômetro para cerca de 10 segundos de dados estacionários coletados com um smartphone Sony Xperia Z5 Compact.

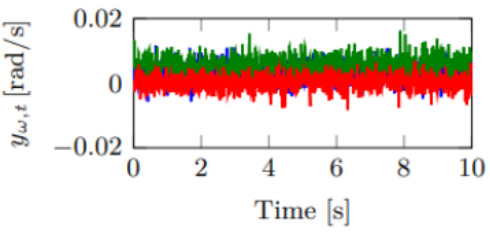
Observando a medição do giroscópio na figura 28, caso fosse um medidor ideal a resposta que se esperaria seria 3 linhas, a primeira azul (para giro em torno do eixo X), a

Figura 27 – Exemplo para ilustrar a definição de ruído e viés.



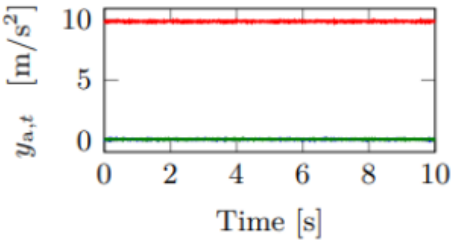
Fonte: Do autor

Figura 28 – Medidas do giroscópio para 10 segundos de dados estacionários



Fonte: (KOK; HOL; SCHöN, 2017).

Figura 29 – Medidas do acelerômetro para 10 segundos de dados estacionários



Fonte: (KOK; HOL; SCHöN, 2017).

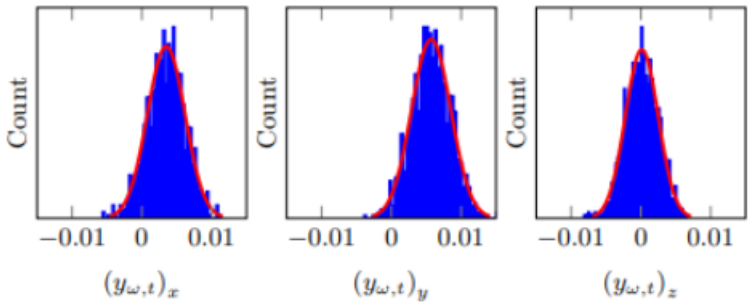
segunda verde (para giro em torno do eixo Y) e a terceira vermelha (para giro em torno do eixo Z) medindo 0 rad/s, haja vista que a medição foi feita com sensor parado. Porém pode-se verificar que existe uma variação em relação ao valor esperado, o que denota a presença de ruído, além disso observa-se que o sinal verde e o azul estão deslocados para cima, o que denota que possuem viés.

Observando a medição do acelerômetro na figura 29, caso fosse um medidor ideal a resposta que se esperaria seria 3 linhas, a primeira azul (aceleração na direção de X), a segunda verde (aceleração na direção de Y) e a terceira vermelha (aceleração na direção de Z), medindo respectivamente 0 m/s^2, 0 m/s^2, devido o sensor estar parado, e 9,81

m/s^2 devido aceleração da gravidade. Devido a escala do gráfico não é possível verificar visualmente, mas existe variação em relação ao valor esperado, o que denota a presença de ruído, além disso observa-se que os sinais estão ligeiramente deslocados para cima o que denota que existe viés.

Uma outra forma de visualizar o ruído e viés do giroscópio é através da função densidade de probabilidade do sinal medido. Veja imagem 30.

Figura 30 – Histograma (azul) das medidas do giroscópio para 10 segundos de um sensor em estado de repouso e um ajuste gaussiano (vermelho) aos dados



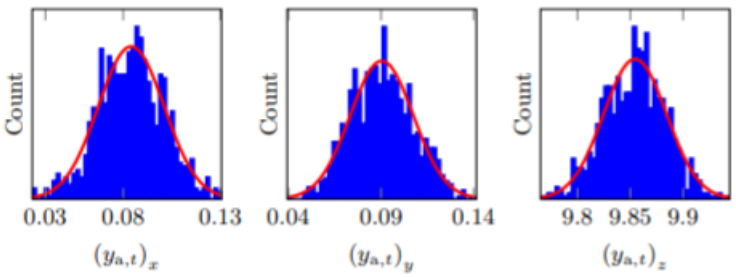
Fonte: (KOK; HOL; SCHöN, 2017)

A resposta para um sensor giroscópio ideal deveria ser apenas uma linha vertical (ou componente) centrada no ponto zero para os três eixos. Todavia o que se observa são várias linhas verticais para valores diferentes do esperado, o que denota a presença de ruído na medição. Além disso, observa-se um deslocamento do histograma para a direita, mais evidente no eixo X e Y. Isso denota que o sinal medido possui viés. Além disso, um ponto importante é notar que os histogramas se ajustam a uma curva Gaussiana, logo pode-se aproximar que o ruído do sistema também possui um comportamento Gaussiano.

Fazendo uma análise semelhante para o sensor acelerômetro temos que na figura 31 a resposta para um sensor acelerômetro ideal deveria ser apenas uma linha vertical (ou componente) centrada no ponto zero para os eixos X e Y e 9.81 para o eixo Z. Todavia, de maneira semelhante ao que ocorreu com o sensor giroscópio, existe ruído na medição. Também se observa o deslocamento para a direita dos histogramas, mais evidente nos eixos X e Y, denotando a presença de viés na medição. Além disso, o ruído também apresenta um comportamento Gaussiano.

Portanto esse experimento evidencia que é necessário passar esses sinais por um etapa de filtragem para que seja possível medir os valores de aceleração e de velociade angular mais precisos.

Figura 31 – Histograma (azul) das medidas do acelerômetro para 10 segundos de um sensor em estado de repouso e um ajuste gaussiano (vermelho) aos dados



Fonte: (KOK; HOL; SCHöN, 2017)

2.8.3 Filtragem dos Sinais do Acelerômetro e Giroscópio

De acordo com (AGUIRRE, 2004), Normalmente, utiliza-se um giroscópio porque ele proporciona uma medição estável e sem ruído da velocidade angular. Sabemos que se conhecemos a velocidade angular, então podemos conhecer a posição. Isso é feito através da operação de integração. O problema ao realizar uma integração numérica no valor fornecido pelo giroscópio é que, à medida que o tempo avança, a leitura estimada do giroscópio contém uma desvio cada vez maior do valor real, resultado da aproximação numérica. Isso acontece por que mesmo parado o giroscópio dificilmente mede um valor nulo de velociade angular, medindo resquícios que ao longo de tempo tende a se somar com a operação de integração. Nesse caso as estimativas de θ utilizando o giroscópio possuem propriedades desejáveis em altas frequências. Portanto, frequências mais baixas podem ser filtradas usando um filtro passa-alta. Também costuma-se usar um acelerômetro que permite conhecer diretamente o ângulo de inclinação. No entanto, a leitura fornecida é altamente ruidosa e sensível a vibrações imperceptíveis que não permitem obter uma leitura estável. Nesse caso o acelerômetro têm propriedades desejáveis em baixas frequências. Portanto, frequências mais altas podem ser filtradas usando um filtro passa-baixa. Mesmo após o processo de filtragem as estimativas de θ podem diferir entre os sensores. Nesse caso para se obter uma medida mais precisa com base na duas medições é feita uma fusão deles.

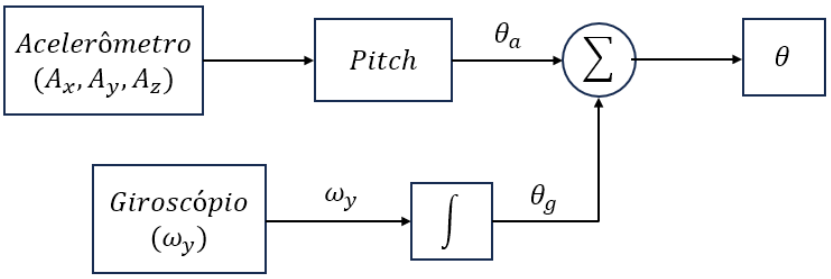
2.8.4 Fusão de Sensores

Segundo (AGUIRRE, 2004) as medições da variável de interesse realizada por sensores diferentes podem apresentar pequenas divergências, uma técnica para estimar essa variável de maneira mais precisa é a fusão de sensores. Essa técnica consiste em fazer uma média aritimética ou ponderada das estimativas da variável de interesse de cada um dos sensores. A média aritimética é aplicada quando a confiabilidade de cada sensor é igual. Quando um dos sensores é mais confiável que outros é dado um peso maior com

média ponderada.

Nesse trabalho a determinação da posição angular do pêndulo do robô será com base em um sensor do tipo acelerômetro e outro giroscópio. O sinal do acelerômetro é usado para medir a posição angular do pêndulo θ por meio do cálculo do angulo de Pitch. O sinal do giroscópio é usado para medir a posição angular do pêndulo integrando o sinal de velocidade angular medido. Ao fundir esses dois sinais é estimado o valor da posição angular do pêndulo com maior precisão. Veja a figura 32 do fluxo dessa técnica de fusão de sensores:

Figura 32 – Exemplo de fusão de sensores.

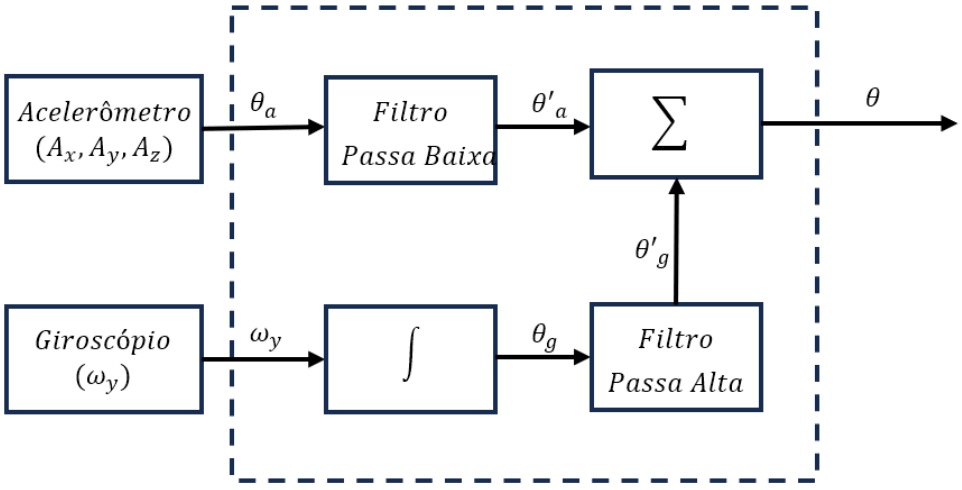


Fonte: Do autor

2.9 Filtro Complementar

De acordo com (AGUIRRE, 2004) o filtro complementar é uma técnica para determinar o valor da variável de interesse unindo uma etapa de filtragem e uma etapa de fusão de sensores. Veja a figura 33

Figura 33 – Esquema do filtro complementar.



Fonte: Adaptado de (AGUIRRE, 2004)

Os dois coeficientes necessários para o correto funcionamento do filtro são obtidos através da relação descrita pela equação 52.

$$\alpha = \frac{\tau}{\tau + \Delta t} \tag{52}$$

Onde α é o coeficiente do filtro passa-altas. Para obter o coeficiente do filtro passa-baixas, realiza-se a subtração de $1 - \alpha$, pois o filtro complementar tem a característica de que a soma dos coeficientes dos filtros deve ser igual a unidade. τ é a constante de tempo dos filtros e Δt representa o intervalo de tempo entre a captura de uma leitura e a próxima.

O filtro complementar utilizado nesse trabalho foi o da equação 53 que se demonstrou bastante eficaz na aplicação de (MESQUITA et al., 2014).

$$\theta = 0,98(\theta + \omega \cdot \Delta t) + 0,02 \cdot \theta_a \tag{53}$$

Onde θ é o ângulo estimado, ω é a velocidade angular fornecida pelo giroscópio e θ_a é o ângulo obtido a partir da informação fornecida pelo acelerômetro.

3 MATERIAIS E MÉTODOS UTILIZADOS

Aqui serão apresentados os materiais e métodos utilizados para alcançar os objetivos específicos definidos na introdução. Esse capítulo se divide em:

a) Materiais que compõem o robô pêndulo invertido móvel;

b) Diagrama do fluxo de sinais do robô;

c) Metodologia para design do chassi do robô;

d) Montagem do robô;

e) Configuração e integração do hardware do robô com os componentes eletrônicos;

f) Sensor inercial e filtragem;

g) Rotina de controle digital;

h) Comunicação e amostragem;

i) Modelagem da planta, discretização do controlador e observador definidos pela metodologia analítica;

j) Projeto e implementação no Arduino o controlador PID pela metodologia de resposta em frequência de Ziegle-Nichols;

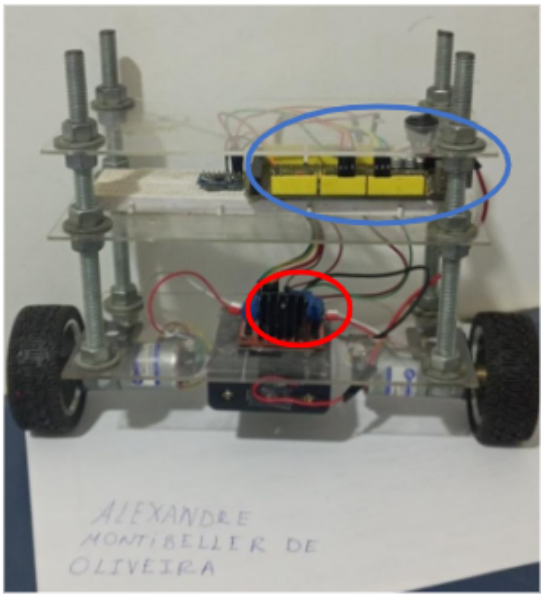
3.1 Materiais que Compõem o Robô Pêndulo Invertido Móvel

O robô pêndulo invertido móvel construído para esse trabalho pode ser observado na figura 34 e 35. A lista de materiais e ferramentas utilizadas para construir o robô pode ser encontrada no apêndice D.

Foi convencionado que a face da frente do robô seria a parte em que aparece um dissipador preto do componente eletrônico chamado de ponte H L298N que aparece circulado em vermelho na figura 34. Essa convenção será importante na hora de definir a postura do sistema e na implementação do algoritmo de controle no Arduino Mega 2560 circulado em azul na figura 34.

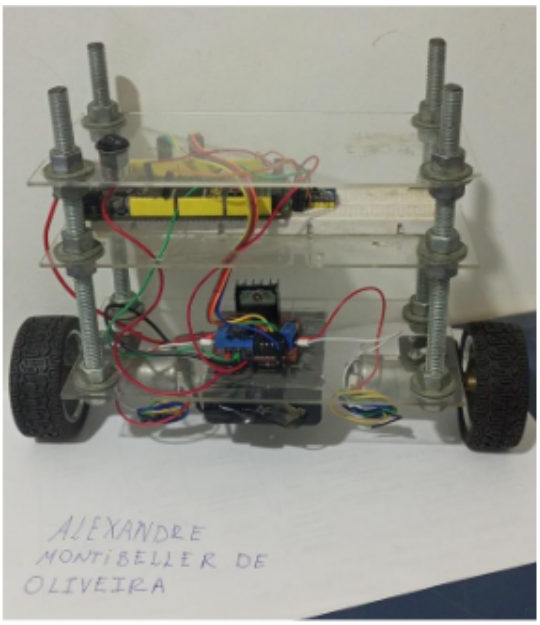
Como é possível observar na figura 34 e 35 o robô é basicamente uma estrutura em formato de paralelepípedo apoiado sobre duas rodas com liberdade para girar tanto para frente quanto para trás. A escolha da estrutura dele foi definida com base em exemplos de referências bibliográficas e limitada pelo custo total dos componentes.

Figura 34 – Face da frente do robô ajustado



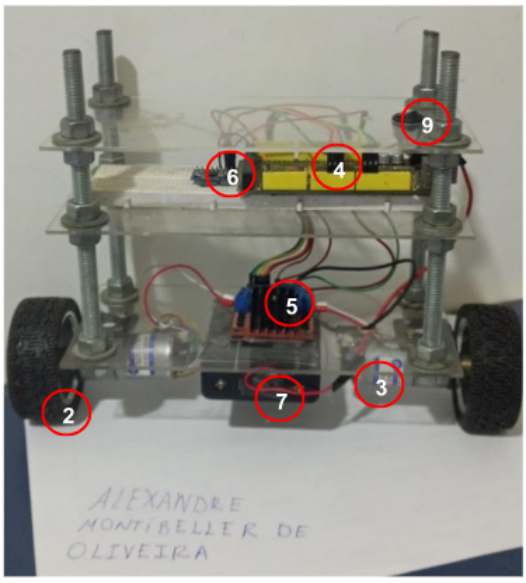
Fonte: Acervo do autor

Figura 35 – Face de trás do robô ajustado



Fonte: Acervo do autor

Figura 36 – Pontos de localização de cada componente do robô pêndulo invertido móvel.



Fonte: Acervo do autor

Os componentes principais que compõem o robô são enumerados logo em seguida e correspondem aos números dos componentes indicados na figura 36.

1. O chassi do robô, onde são fixadas as outras partes;
2. As rodas;

3. Motores de corrente contínua 6V 280RPM com Encoder e redutores;
4. A placa Arduino UNO Mega 2560;
5. A ponte H L298N;
6. O sensor inercial MPU-6050;
7. Case com 3 baterias Li-ion UR18650 2150 mAh 3,6v ligadas em série;
8. As conexões eletro-mecânicas entre as partes.
9. Botão liga e desliga.;

Em seguida será feito um detalhamento maior de cada componente.

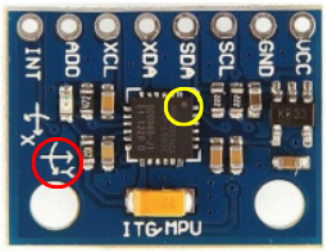
3.1.1 Sensor Inercial MPU-6050

O sensor será o componente responsável por medir o grau de inclinação do rôbo.

O sensor inercial utilizado é o MPU-6050, embarcado no módulo GY-521. O MPU6050 tem acelerômetro de 3 eixos e giroscópio de 3 eixos, e cada eixo possui um conversor analógico-digital (ADC) de 16 bits. A escala do acelerômetro pode ser programada com as faixas de $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$, e o sinal do giroscópio com as faixas de ± 250, ± 500, ± 1000, $\pm 2000/s$ (INVENSENSE. . . , 2012).

A comunicação do sensor é feita através de barramento serial I2C, na taxa de 400kHz, pelos pinos SDA (serial data) e SCL (serial clock) (INVENSENSE. . . , 2012).

Figura 37 – O módulo GY-521 com o sensor inercial MPU-6050

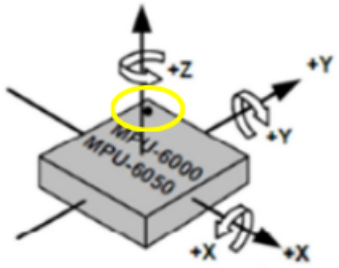


Fonte: (MóDULO. . . , 2023)

O sistema de coordenadas desse sensor pode ser ilustrado pela figura 38.

Na figura 37 tem um círculo vermelho que indica a direção e sentido do eixo Y em relação ao MPU, e a partir dessa orientação é possível definir os outros eixos cartesianos. Também na Figura 37 tem um círculo amarelo que destaca um ponto preto no canto de um circuito integrado que marca o mesmo ponto de referência destacado com o círculo amarelo na Figura 38. Essa correlação é uma segunda maneira de definir a orientação dos eixos cartesianos em relação ao MPU.

Figura 38 – Sistema de referência MPU



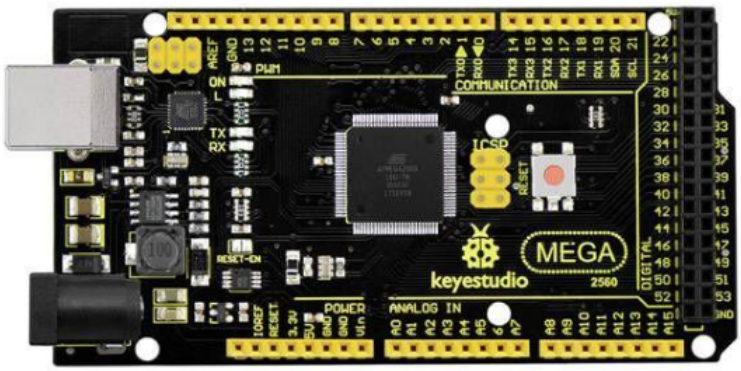
Fonte: Adaptado de (MERINO, 2015)

Essa orientação será importante no momento de definir a posição em que o MPU ficará no chassi do robô. Essa posição influenciará no cálculo da postura do robô. Comparando o chassi do robô com o avião da figura 26 temos que eixo X está saindo pela parte da frente do avião e o eixo Y saindo pela asa esquerda. Nesse caso o MPU deve ser posicionado no chassi do robô de tal forma que o seu eixo X saia pala parte da frente do robô e o eixo Y pela lateral esquerda.

3.1.2 Arduino MEGA 2560

O Arduino é o componente responsável por realizar os cálculos de controle do robô. O sistema de controle está embarcado numa placa Arduino MEGA 2560, baseada no MCU (microcontroller unit) ATmega 2560-16UA. A placa tem 54 entradas e saídas digitais, 15 das quais podem fornecer saída PWM, e 16 entradas analógicas com conversor analógico-digital de 10 bits. Possui memória Flash de 256 KB, SRAM de 8 KB e EEPROM de 4 KB. O clock é de 16 MHz.

Figura 39 – Arduino Mega 2560 Keyestudio.

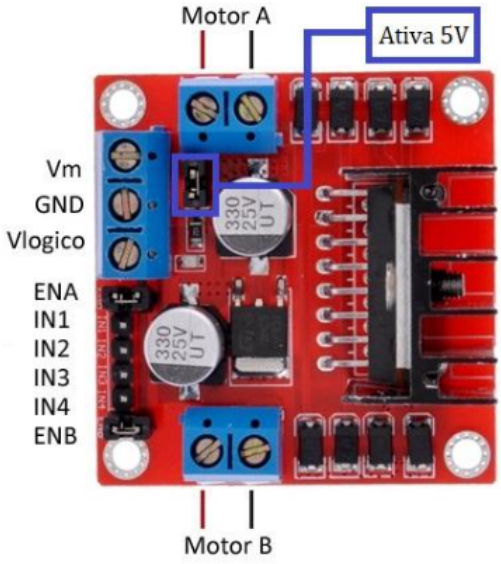


Fonte: (KEYESTUDIO...,)

3.1.3 Ponte H L298N

A ponte H é o componente resposável por acionar os motores a partir de sinais de comando do microcontrolador. Os sinais lógicos de saída do sistema de controle alimentam uma ponte H L298N por 2 pares de conexões. Esses sinais são tipo PWM, e sendo sinais de entrada do circuito lógico de controle do chaveamento da ponte H, definem o sinal PWM de potência na saída da ponte H, para o acionamento de 2 moto-redutores.

Figura 40 – Ponte H L298N



Fonte: (MOTOR...,)

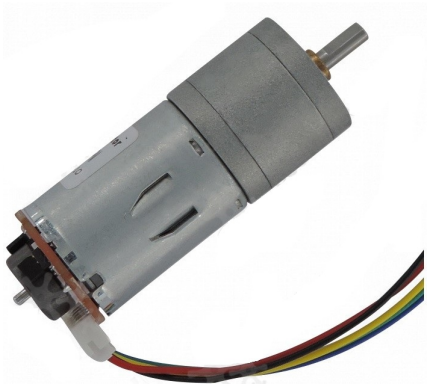
Na figura 40 o borne Vm é o da entrada da fonte de alimentação dos motores que pode variar de 7 á 35 V. Nesse caso em que a fonte de alimentação é maior que 5 V é necessário usar um jumper no ponto indicado por um quadrado azul para habilitar o regulador de tensão para sair 5V no terminar Vlogico. Caso contrário basta tirar o jumper e alimentar o terminal Vlogico com 5 V.

O terminal GND é o terra, Vlogico é a tensão Vm retificada para 5 V normalmente utilizada para alimentar o microcontrolador. Os terminais de controle IN1, IN2, IN3, IN4 são os que recebem o sinal PWM de controle do Arduino. Os terminais ENA e ENB também podem ser habilitados para controlar a velocidade dos motores recebendo o sinal PWM, porém isso fica a critério do projetista. Caso queira habilitar basta tirar os jumpers e por os cabos de controle que vem do Arduino. Do contrário basta por os jumpers. Os outros terminais são os de alimentação dos motores.

3.1.4 Motor DC 6V 280RPM com Encoder e Roda

Os motores são os componente resposáveis por fazer a correção da posição do pêndulo do robô de acordo com o acionamento da ponte H. Este motor opera com uma

Figura 41 – JGA25-371 DC Gearmotor com Encoder



Fonte: (Open Impulse,)

faixa de tensão entre 6 V e 24 V, possuindo uma voltagem nominal de 12 V. A sua velocidade de rotação livre atinge 280 RPM quando alimentado com 12 V, e a corrente de funcionamento livre nesse mesmo ponto é de 0,4 A. o torque é de 0.044 N · m. A relação de engrenagens é de 1:21.3.

3.1.5 Case de 3 Baterias UR18650A M19 Ligadas em Série

A alimentação de todo o circuito é feita por uma bateria de polímero de lítio (Li-Po) de 3 células de 3,7 V cada de acordo do a figura 43 , totalizando 11,1 V. Foi utilizado em case de acordo com a figura 42 para facilitar a conexão desconexão das baterias no robô facilinado o momento de carregar elas com um carregador universal. O circuito é fechado por um interruptor no polo positivo da bateria.

Essas baterias podem ser encontradas em aparelhos eletrônicos que viraram sucata, principalmente notebooks. Normalmente lojas que prestam serviços de conserto de aparelhos eletrônicos vendem essas baterias a um preço acessível.

3.2 Diagrama do Fluxo de Sinais do Robô

Os sinais de entrada, saída, de controle e referência do sistema do robô podem ser vistos no diagrama de fluxo de sinais de um sistema modelado pelo espaço de estados na figura 44. O fluxo do sinal começa com a medição das acelerações, A_x, A_y, A_z na direção de X, Y e Z e da velocidade angular ω_y no eixo Y pelo sensor acelerômetro e giroscópio embarcado na MPU-6050. A comunicação entre o MPU-6050 e o Arduino é estabelecida pelo protocolo de comunicação I2C. As medidas de aceleração são usadas para o cálculo de ψ_a pelo cálculo de Pitch e o sinal da velocidade angular ω_y é direcionado para o filtro complementar que realiza os cálculos e determina a posição angular medida ψ_m. Quando o robô é posto na vertical deveria medir 0°, porém na prática sempre é medido um valor que é chamado de ψ_{offset}. Subtraindo esse valor é definido o valor da posição angular do

Figura 42 – Case de 3 baterias UR18650A M19 ligadas em série.



Fonte: (SUPORTE. . . ,)

Figura 43 – Bateria UR18650A M19.

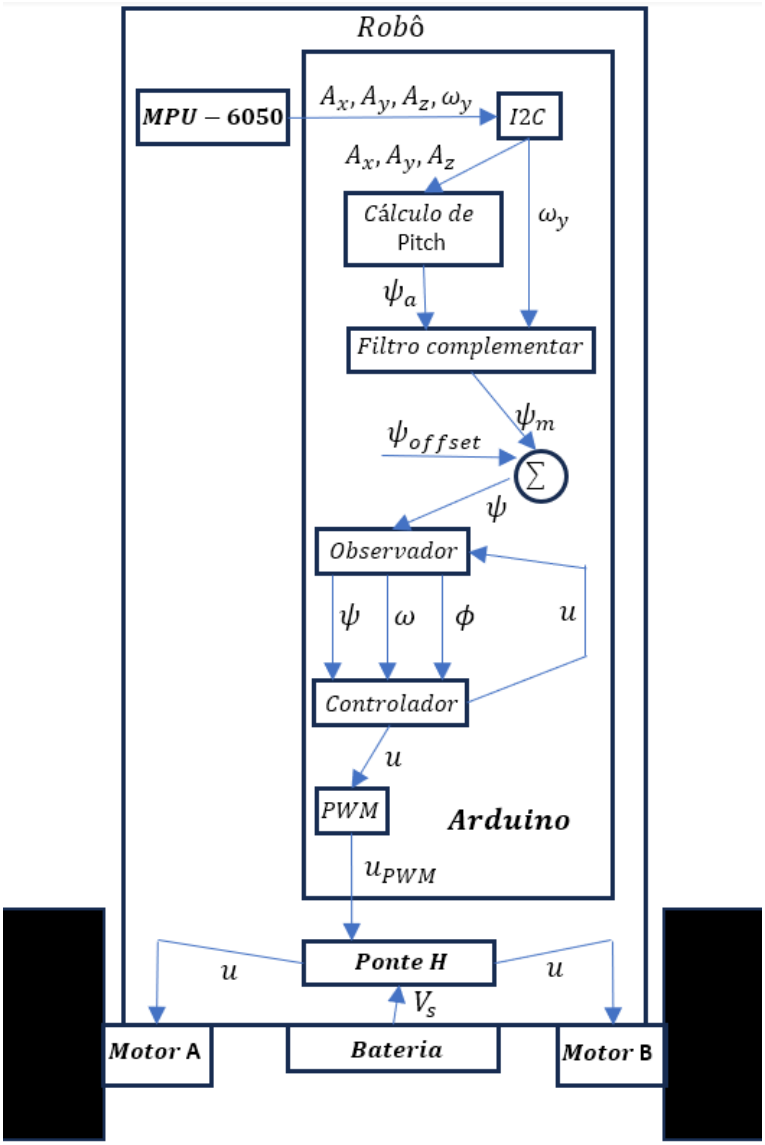


Fonte: (CéLULA. . . ,).

pêndulo ψ já ajustado. O observador recebe o sinal ψ e o valor de tensão u aplicado no motor e estima o valor das variáveis de estado ω e ϕ. O controlador recebe os sinais dos três estados e calcula o valor de tensão que varia entre -12 e 12 V que deve ser aplicado nos motores. u é convertido para um sinal de tensão u_{PWM} em formato de PWM que irá variar com uma tensão média entre -5 e 5 V. A ponte H recebe o sinal de controle u_{PWM} e converte para um sinal u novamente de acordo com a lógica da equação 56 que será detalhada nas próximas sessões. Com aplicação de tensão u nos motores eles fazem a posição angular do pêndulo variar fechando o ciclo com uma nova medição pela MPU.

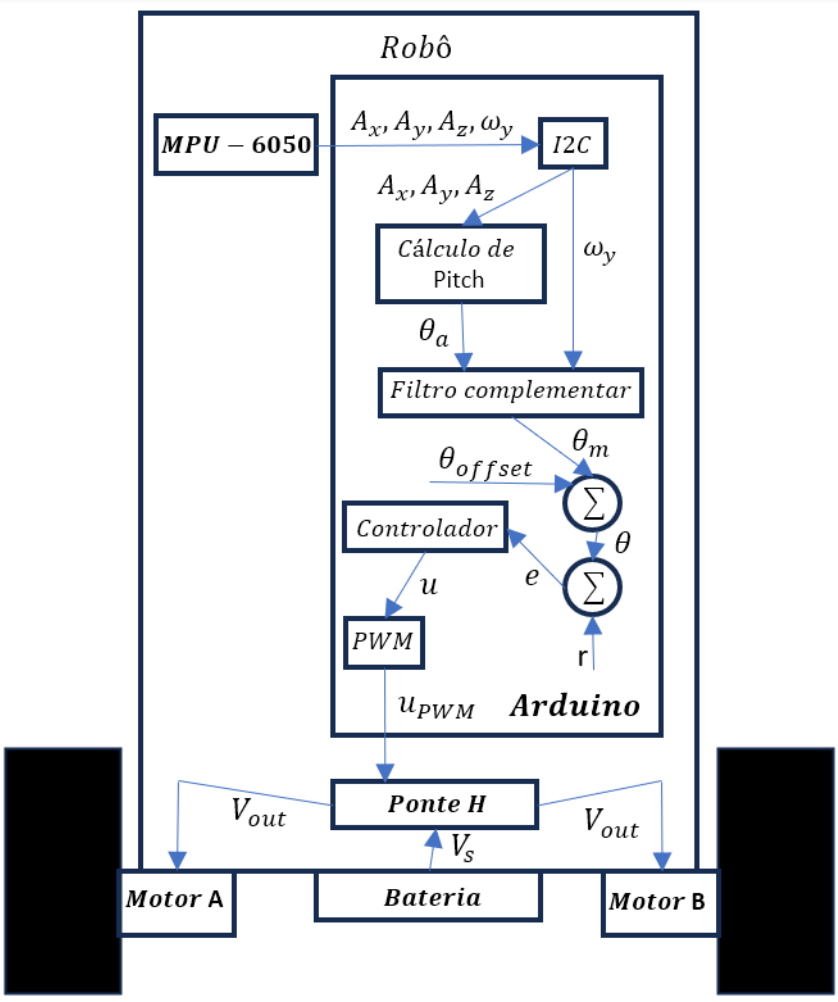
De forma semelhante a figura 45 tem o fluxo de sinais do robô para controladores com a estrutura de PID. O fluxo do sinal é o mesmo que o do modelo da figura 44, diferindo quando o sinal chega no controlador, também com a mudança de que a posição do pêndulo é denotada por θ. Esse sinal é comparado com o sinal de referência r que é 0° gerando o sinal de erro e. Esse sinal é mandado para o controlador que calcula o valor de tensão u delimitado entre -5 e 5 V que é convertido em um sinal u_{PWM} em formato PWM que é enviado para a ponte H que determina o valor de tensão aplicado nos motores V_{out} de acordo com a lógica da equação 56 que será detalhada nas próximas sessões. V_{out} é enviado para os motores que fazem a posição do pêndulo variar e ser medido novamente pelo sensor MPU-6050.

Figura 44 – Diagrama de blocos para o fluxo de sinais do robô modelo em espaço de estados



Fonte: Do autor

Figura 45 – Diagrama de blocos para o fluxo de sinais do robô com controlador com estrutura de PID

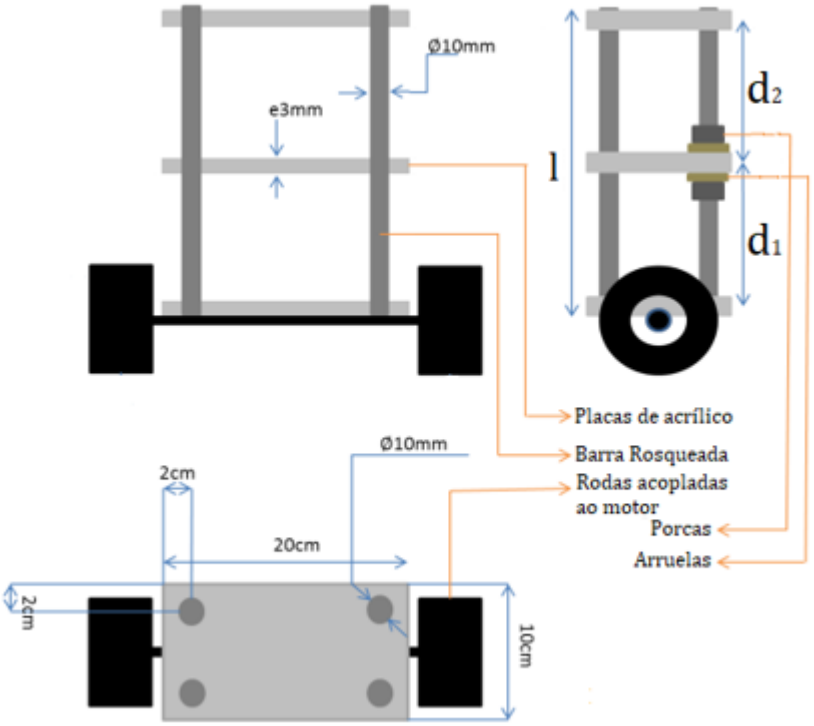


Fonte: Do autor

3.3 Metodologia para Design do Chassi do Robô

Para a integração das partes, foi necessário avaliar o posicionamento dos dispositivos, bem como as características geométricas que influenciam na dinâmica do pêndulo móvel. Foi adotado um modelo de chassi com 3 pisos de acrílico de 3 mm de espessura, 10 cm de largura e 20 cm de comprimento, interligados por barras rosqueados de 10 mm de diâmetro nominal e fixados com porcas M12 e arruelas para fixação com diâmetro interno de 15 mm e externo de 30 mm. Um esquema mais detalhado dos furos feitos nos pisos para passagem das barras rosqueadas, fixação dos suportes dos motores e de cabos de conexão elétrica pode ser visto no apêndice C. Os moto-redutores são fixados no piso inferior, através de suportes que vem junto com o kit motor, rodas e conjunto de fixação. Um esquema genérico desse modelo de chassi pode ser observado na figura 46:

Figura 46 – Modelo genéricos de chassi de robô pêndulo invertido sobre duas rodas com 3 pisos. Não está em escala



Fonte: Adaptado de (SILVA; OUTROS, 2017)

Pelo modelo genérico de chassi da figura 46, existe liberdade para definir o comprimento total do pêndulo l e a distância entre os pisos d_1 e d_2, além de liberdade para escolher em que piso cada um dos componentes do robô irá ser posicionado. Nesse caso, o objetivo será definir uma configuração de chassi e de posicionamento dos componentes do robô de forma que em malha aberta permanecer mais tempo próximo do ponto de operação de 0°, e em malha fechada, não variar a posição angular de maneira abrupta devido a pequenas aplicações de torque pelos motores. Esse ajuste é necessário para facilitar o

projeto de controladores tanto pela metodologia analítica quanto pela experimental. Uma forma de analisar o comportamento dinâmico correlacionado ao comprimento do robô e com a distribuição das massas dos seus componentes nos pisos do seu chassi é através do modelo simplificado da equação 17, que considera o pêndulo móvel composto de uma massa M, maior e concentrada na parte inferior, e uma massa m, menor e concentrada na parte superior.

Tomando a equação 17:

$$Ml\ddot{\theta} = (M+m)g\theta - u \tag{54}$$

Aplicando a transformada de Laplace e simplificando a equação, obtém-se a equação 56.

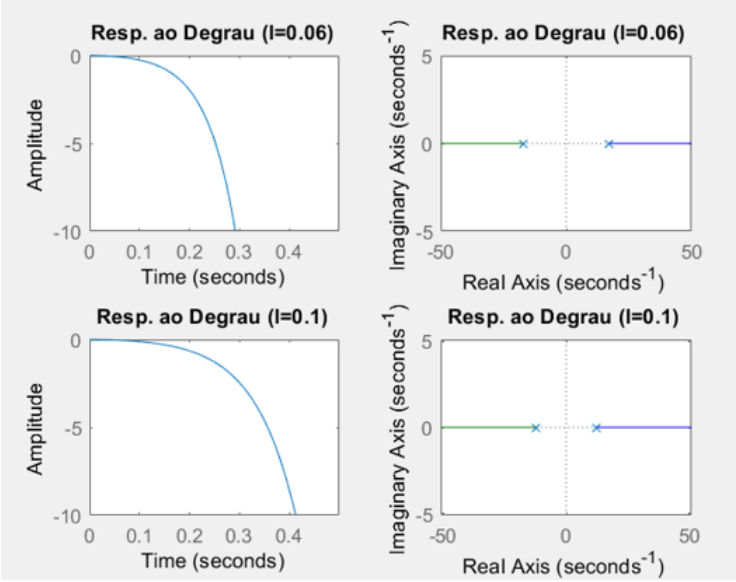
$$\frac{\theta(s)}{U(s)} = \frac{1}{-(Ml)s^2 + (M+m)g} \tag{55}$$

Para fins de análise será adotando para $M = 0.5Kg$ e $m = 0.25Kg$.

Para avaliar o impacto da escolha de l será escolhido dois valores, $l_1 = 0.06m$ e $l_2 = 0.1m$

A resposta ao degrau e o lugar geométrico das raízes para os dois comprimentos adotados é o mostrado na figura 47.

Figura 47 – Resposta ao degrau e lugar geométrico das raízes para diferentes valores de l



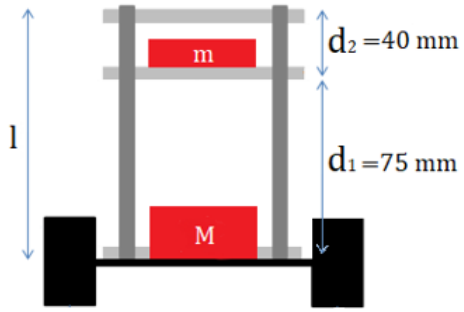
Fonte: Do autor

É possível notar que o aumento do comprimento determina um aumento do tempo de equilíbrio em malha aberta. Também se observa a aproximação do polo real positivo

da origem, o que determina a redução do grau de instabilidade. Todavia foi verificado experimentalmente que se os polos se aproximam muito da origem o sistema tende a ficar mais oscilatório em malha fechada com ação de controle proporcional nos motores, fazendo com que a posição do pêndulo varie de maneira abrupta para uma região longe da zona de estabilidade próxima de 0°. O que ocorreu foi que pequenas aplicações de torque pelos motores faziam com que o sistema se desestabilizasse mais rápido para valores maiores de l. Nesse caso foi necessário haver uma dosagem entre a resposta em malha aberta e fechada pra determinar o valor de l adequado. Devido a essas verificações, a bateria que possui um peso significativo foi posta no piso inferior junto com os motores para compor um conjunto de massa M maior na parte inferior do chassi e o restante dos componentes no segundo piso para compor o conjunto de massa m. Para aumentar o comprimento l do pêndulo foi definido uma distância maior entre o primeiro e segundo piso. Para concentrar o restante da massa do robô na extremidade superior foi definido uma pequena distância entre os pisos dois e três de acordo com a figura 48.

Por fim foi determinado que a melhor resposta encontrada que dosa a relação entre resposta em malha fechada e aberta foi a distância $d_1 = 75\,\text{mm}$ e $d_2 = 40\,\text{mm}$.

Figura 48 – Chassi e posicionamento da distribuição de massas nos pisos



Fonte: Adaptado de (SILVA; OUTROS, 2017)

3.4 Montagem do Robô

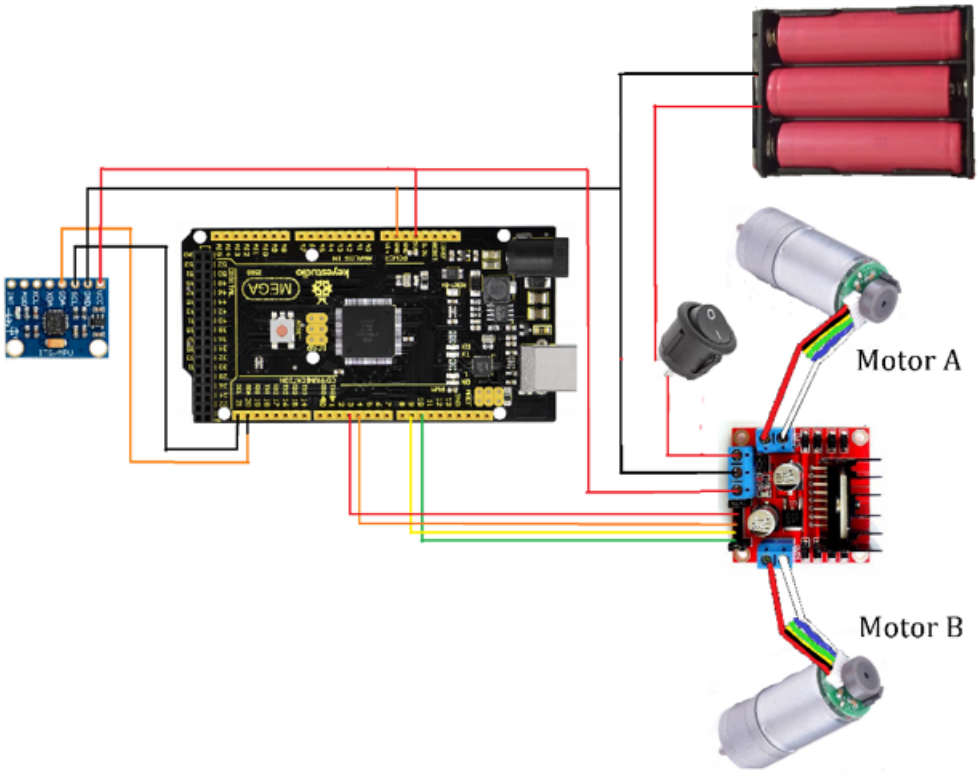
Tendo em vista as conclusões da sessão anterior a configuração da posição de cada componente do robô é exatamente a mostrada na figura 36.

A fixação da protoboard, da ponte H e das baterias nos pisos de acrílico foi feita com fita adesiva dupla face. O Arduino também foi fixado no protoboard da mesma forma. A fixação do restante dos componentes já foi abordada na sessão anterior. Nessa configuração foi escolhida a opção de pôr a ponte H na parte de cima do primeiro piso para ficar mais próxima das baterias e dos motores diminuindo assim o caminho da transferência de potência. Além disso a escolha de posicionar o sensor MPU mais longe dos motores foi para que o campo magnético dos mesmos não gerasse ruídos na medição. O botão de liga e

desliga foi posicionado na parte de cima do terceiro piso para ficar de mais fácil acesso. As ligações elétricas foram todas feitas com o auxílio da protoboard e cabos jumpers. Além disso para melhor fixação da conexão dos motores com a ponte H foi usado um terminal elétrico tipo pino isolado da mesma cor que os cabos da conexão.

As conexões elétricas entre os componentes são apresentadas na figura 49 que ilustra um esquema de conexão elétrica real, na figura 50 que apresenta o projeto elétrico elabora no aplicativo online EasyEAD e no mapa de portas interligadas entre Arduino, MPU-6050 e ponte H descrito na tabela 3.

Figura 49 – Esquema de conexão elétrica real do pêndulo móvel



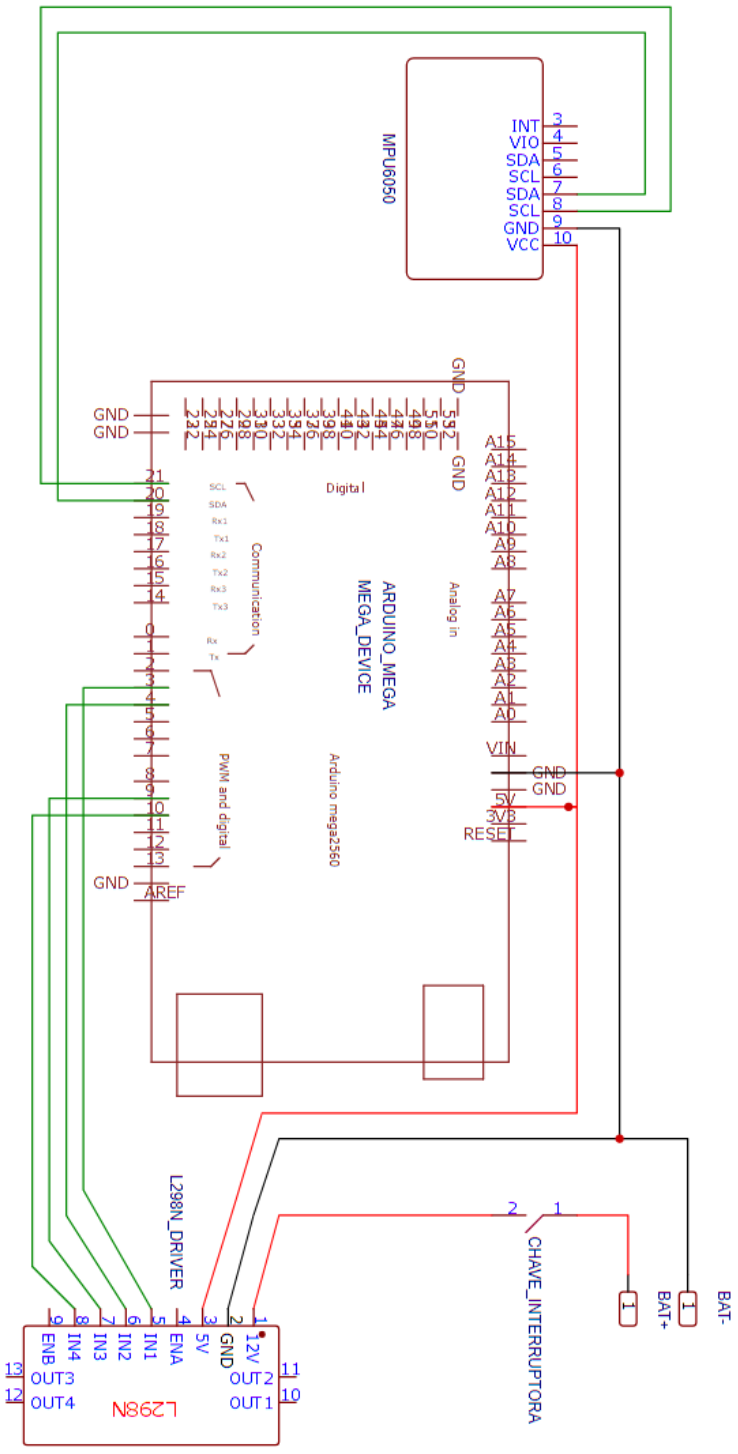
Fonte: Da autor

Tabela 3 – Mapa de portas

Dispositivo	**Pino de Origem**	**Pino Arduino**	**Descrição**
MPU-6050	SDA	20	Dados
	SCL	21	Clock
	VCC	5V	Alimentação 5 V
	GND	GND	GND
Ponte-H	IN1	3	Controle motor1
	IN2	4	Controle motor1
	IN3	9	Controle motor2
	IN4	10	Controle motor2

Fonte: Do autor

Figura 50 – Esquema elétrico do pêndulo móvel



Fonte: Do autor

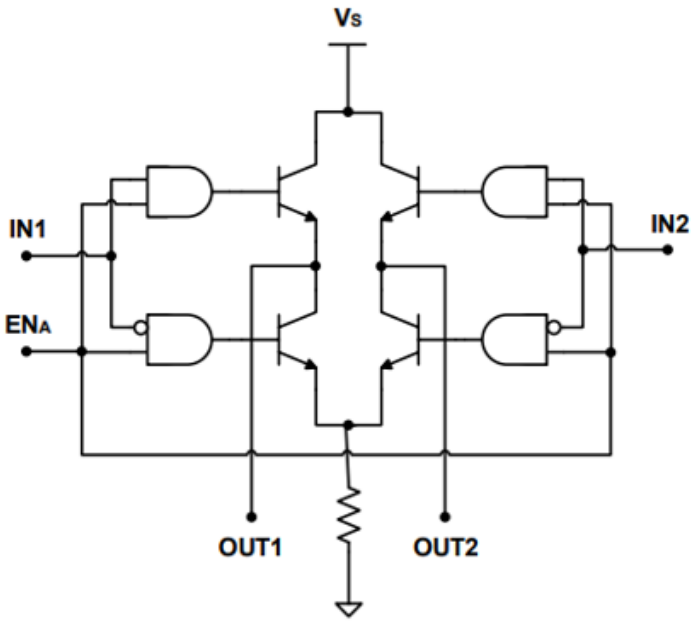
O último teste após a montagem foi um teste de equilíbrio. O pêndulo móvel foi colocado na posição vertical e solto para verificar se ele mantinha a posição sem ação motora. Como era de se esperar, foi verificado que ele não sustenta a posição e que a geometria e material das rodas não fornece auxílio adicional no equilíbrio vertical.

3.5 Configuração e Integração do Hardware do Robô com os Componentes Eletrônicos

Após a montagem, foram realizados testes para identificação da pinagem que determina os sentidos de rotação dos motores. Primeiramente será necessário entender como que o Arduino controla a ponte H para determinar o sentido de giro e velocidade dos motores.

A figura 51 mostra o esquemático da ponte H L298 para um motor, onde os pinos IN1 e IN2 são os pinos de controle, e os pinos OUT1 e OUT2 são os pinos de saída para o motor. Há também um pino de ativação ENA, que é jumpeado na placa (veja figura 40) para nível alto. Para que haja tensão nos pinos de saída, um pino de controle precisa estar em nível lógico alto, enquanto o outro deve estar em nível baixo. Se os dois pinos estiverem em nível lógico alto, a ponte não fornece tensão para os motores. O quadro 4 demonstra as condições de operação.

Figura 51 – Esquema elétrico da ponte H L298N



Fonte: (STMicroelectronics,)

Tabela 4 – Controle do motor segundo as entradas da ponte H.

IN1	IN2	OUT
Baixo	Baixo	Eixo livre
Baixo	Alto	Direto
Alto	Baixo	Reverso
Alto	Alto	Frenagem Forçada

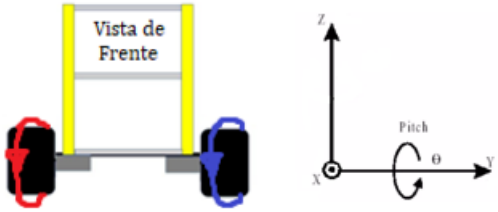
Fonte: (STMicroelectronics,)

Neste caso o Arduino será o componente responsável por mandar o sinal de nível alto e baixo para os pinos IN1 e IN2 da ponte H. A tabela 4 demonstra o efeito das combinações de nível lógico alto e baixo.

Neste caso para fazer o teste de sentido de giro do motor A, que é o motor direito do robô, será necessário mandar os sinais teste para a portas IN1 e IN2 da ponte H. Pelo quadro 1 temos que IN1 está ligado a porta 3 do Arduino e IN2 na portar 4. Nesse caso basta configurar o Arduino para mandar os sinais lógicos da porta 3 e 4 como HIGH ou LOW e observar a resposta. Esse mesmo raciocínio pode ser aplicado para o motor B. Usando o código do apêndice B na parte "Teste de motores" para configurar os sinais lógicos das portas 3, 4, 9 e 10 temos a resposta da figura 52.

Figura 52 – Sentido de rotação dos motores em relação ao eixo de coordenadas do IMU de acordo com o nível lógico aplicado nos pinos de controle 3, 4, 9 e 10 do Arduino nas portas IN1, IN2, IN3 e IN4 da Ponte H - 1° Teste

IN1= 3	IN2 = 4
Auto	Baixo
IN3= 9	IN4 = 10
Auto	Baixo



Fonte: Do autor

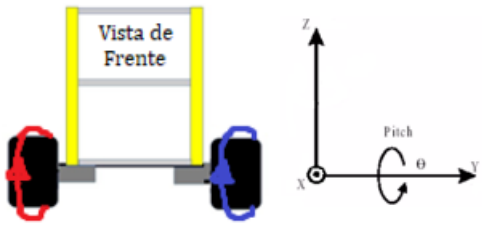
O resultado do primeiro teste foi que para os sinais lógicos definidos o sentido de giro das duas rodas coincidiu com sentido de giro positivo em relação ao eixo Y do sistema de coordenadas do IMU. Nesse caso pode-se dizer que ambas as rodas estão girando no sentido horário.

O resultado do segundo teste foi que o sentido de giro das duas rodas foi o contrário de sentido de giro positivo em relação ao eixo Y do sistema de coordenadas do IMU. Nesse caso pode-se dizer que ambas as rodas estão girando no sentido anti-horário. Para os outros testes não foi observado movimento das rodas. Isso já era esperado pois para esses testes o resultado esperado era eixo livre e frenagem forçada.

Até esse momento foi possível ligar e desligar os motores, mas não controlar a sua velocidade, de forma que desligados permaneciam parados e ligado giravam na velocidade máxima. Para fazer o controle da velocidade foi necessário regular a tensão entregue aos motores. Para fazer isso basta aplicar uma técnica de ligar e desligar o sinal em nível lógico alto de tal forma que a tensão entregue ao motor seja limitada ao tempo que ele

Figura 53 – Sentido de rotação dos motores em relação ao eixo de coordenadas do IMU de acordo com o nível lógico aplicado nos pinos de controle 3, 4, 9 e 10 do Arduino nas portas IN1, IN2, IN3 e IN4 da Ponte H - 2° Teste

IN1= 3	IN2 = 4
Baixo	Auto
IN3= 9	IN4 = 10
Baixo	Auto



Fonte: Do autor

permanecer ligado ou desligado. Essa técnica é chamada de modulação por largura de pulso (PWM). A lógica é que para um período fixo é possível estipular o tempo que o nível lógico alto permanecerá ligado e desligado. A tensão média entregue ao motor será a porcentagem entre o tempo que o nível lógico permanece ligado em relação ao tempo de um período completo de controle. Essa relação é chamada de duty cycle. Veja a equação 56.

$$V_{\text{out}} = \frac{V_s \cdot t_{\text{on}}}{t_{\text{on}} + t_{\text{off}}} = \frac{V_s \cdot t_{\text{on}}}{T} = V_s \cdot d \tag{56}$$

- V_{out}: Tensão média de saída
- V_s: Tensão da fonte de tensão
- t_{on}: Tempo que o sinal lógico permanece ligado
- t_{off}: Tempo que o sinal lógico permanece desligado
- T: Período do sinal
- d: Duty Cycle

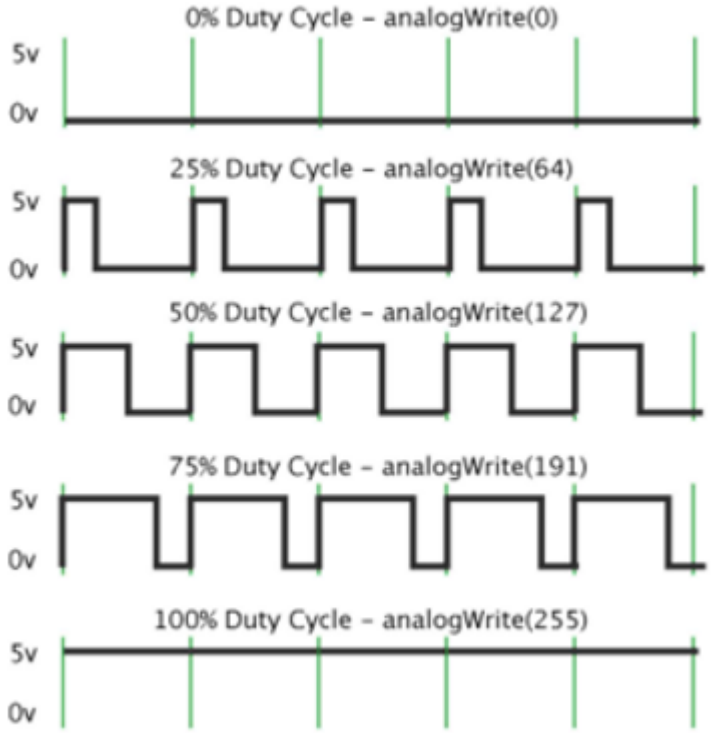
Pela relação acima, se o duty cycle for, por exemplo, 50%, ou seja, o sinal lógico de nível alto ficará metade do período ligado e outra metade desligado, a tensão entregue ao motor será de 50% da tensão da fonte.

O sinal de controle, definido pelo Arduino do tipo PWM, possui amplitude de 5 V e resolução de 8 bits (0 a 255) para a definição do Duty Cycle. A relação entre o

duty cycle e a resolução é linear. Portanto, para definir a resolução que será aplicada no Arduino, basta multiplicar o duty cycle pela resolução máxima. Por exemplo, se quiser que o Arduino gere um sinal PWM com duty cycle de 50%, basta determinar 50% de 255, o que seria 127.

Um exemplo gráfico da relação entre o duty cycle e a resolução do Arduino encontra-se na figura 54:

Figura 54 – Duty cycle em função da resolução do Arduino

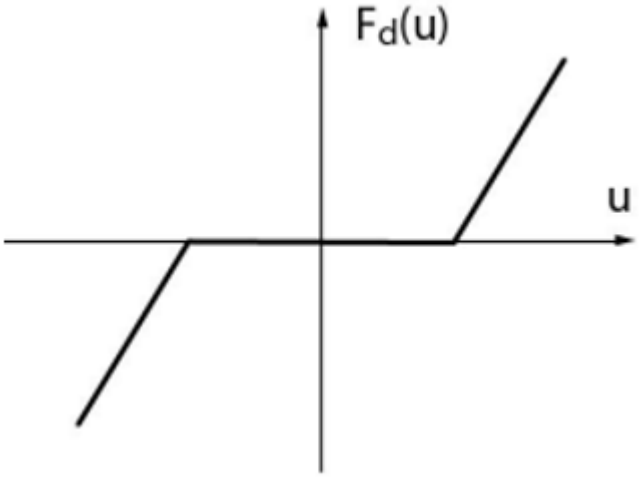


Fonte: (MESQUITA et al., 2014)

Tendo em vista isso para aplicar o teste de velocidade primeira deve ser verificado se as portas do Arduino escolhidas são do tipo digital PWM. Em seguida basta usar o código do Apêndice B na parte "Teste de motores" e definir valores de resoluções entre 0 e 255 nas portas de devem estar no nível lógico alto enquanto a outra permanece no nível lógico baixo.

Um problema que pode dificultar o controle do sistema é a resposta dos motores numa região de trabalho chamada de zona morta. A zona morta é a denominação do comportamento não-linear da dinâmica de motores, principalmente quando operam em baixas velocidades e alternando o sentido de rotação, quando a tensão necessária para iniciar movimento do eixo precisa ser suficiente para superar a inércia e o torque de fricção estático (KARA; EKER, 2004). A figura 55 representa uma curva de torque x tensão para um motor CC de imãs permanentes típico.

Figura 55 – Não-linearidade da zona morta



Fonte: (KARA; EKER, 2004))

Nesse trabalho, considerando-se que os motores utilizados no robô são de qualidade equivalente, adotou-se uma solução mais simples e intuitiva, que foi o ajuste de um offset no duty cycle do PWM do sinal de controle, para garantir reação imediata dos moto-redutores a partir de sinal de controle não nulo. O teste para encontrar o offset do duty cycle foi feito com o motor sem carga, ou seja, rodas sem contato com o chão. O ajuste do offset foi feito para os dois sentidos de rotação, de forma independente para cada motor, obtendo dois valores para cada motor, para cada sentido de rotação. Esses valores estão disponíveis na tabela 5

Tabela 5 – Tabela de configuração do offset dos motores

Roda	**Sentido**	**Offset (PWM)**	**u_Offset (V)**
Direta	Anti-horário	14	0.27
Esquerda	Anti-horário	18	0.35
Direita	Horário	15	0.29
Esquerda	Horário	15	0.29

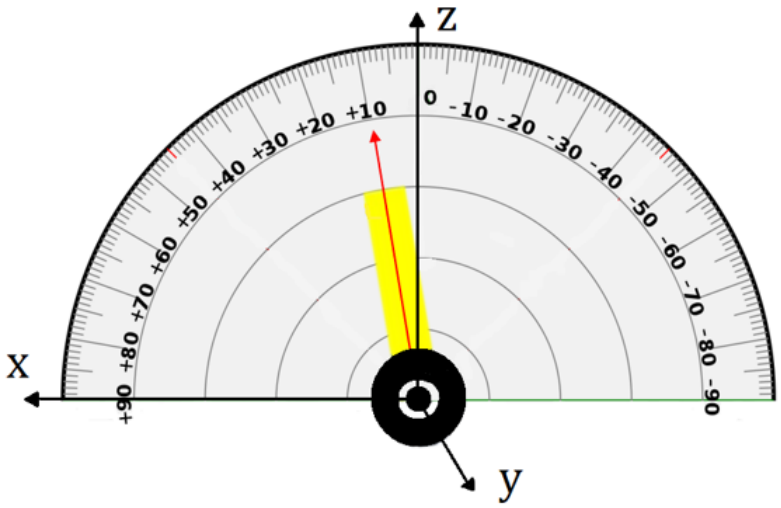
Fonte: Autor

Em seguida foi verificado se o IMU estava se comunicando e se o sinal estava coerente, recebendo dados via monitor serial do Arduino IDE. Foi feita a leitura dos acelerômetros e dos giroscópios, atestando funcionamento.

3.6 Sensor Inercial e Filtragem

A posição angular θ do pêndulo do robô pode ser representada pela figura 56 tendo como sistema de coordenadas a mesma do sensor IMU.

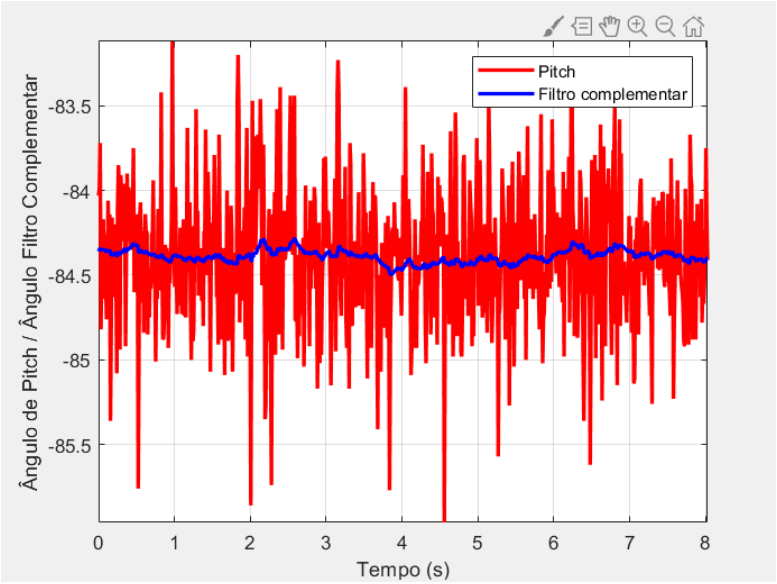
Figura 56 – Posição angular do pêndulo do robô



Fonte: Autor

Para a estimativa θ foi usado o filtro complementar. O teste de sinal do sensor foi realizado com o pêndulo em condição estática. Os sinais θ estimados pelo acelerômetro e pelo filtro complementar foram transmitidos via cabo USB para a porta serial e plotados no MATLAB. A figura 57 mostra os sinais.

Figura 57 – Sinal de Pitch estimado pelo acelerômetro e pelo filtro complementar.



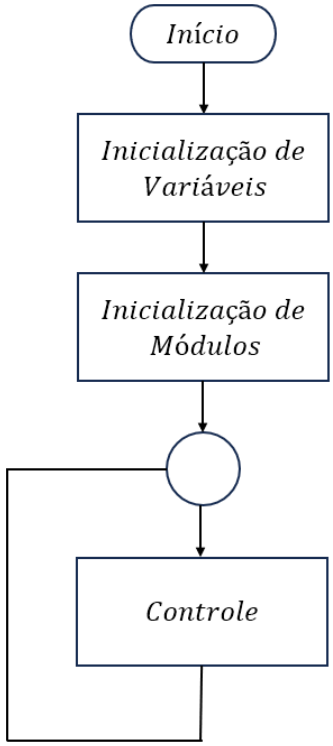
Fonte: Autor

O sinal de Pitch estimado pelo acelerômetro tem média de -84.38° com desvio padrão de 0.39°, oscilando numa faixa de 1.57°. A estimativa pelo filtro complementar tem média de -84.38°, com desvio padrão de 0,03° e oscila numa faixa de 0.1°. Por esses resultados pode-se afirmar que o angulo estimado pelo filtro complementar é mais preciso.

3.7 Rotina de Controle Digital

O controle do sistema do pêndulo móvel está descrito nos fluxogramas da figura 58 e figura 59. O detalhamento de cada bloco de função dos diagramas pode ser conferido no código presente no apêndice B.

Figura 58 – Fluxograma de funcionamento do robô



Fonte: Do autor

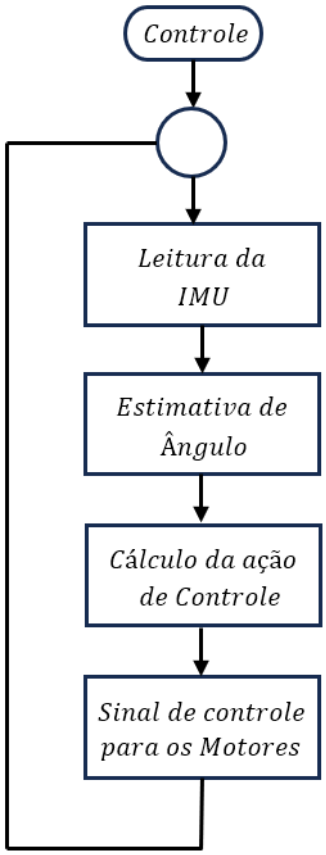
Na figura 58, na etapa de inicialização de variáveis, inclui-se inicialização de variáveis e objetos, entre elas: variáveis de leitura do IMU, de estimativa e de tempo e iteração; objeto do filtro e entre outros.

Na inicialização de módulos, estão incluídos todos os dispositivos que recebem parâmetros iniciais e seus métodos, como comunicação I2C. Em controle, estão as etapas do loop de controle, que estão detalhadas na figura 59.

A saída do controlador PID, que se refere a etapa de cálculo da ação de controle, está limitada entre -5 e 5 V, que é o valor da amplitude do pulso PWM enviado para as portas lógicas da ponte H. Para enviar o sinal PWM, esse sinal de controle é escalado para o intervalo de -210 a 210. Esse escalonamento é feito pela equação 57 de conversão do sinal de uma escala de -5 até 5 para -210 até 210:

$$comando = ((u_{teta} + 5) * (OUT_{MAX} - OUT_{MIN})/10) + OUT_{MIN} \tag{57}$$

Figura 59 – Fluxograma do controle do robô



Fonte: Do autor

- *comando*= Saída do sinal de comando do controlador na resolução de-255 até 255;
- u_{teta}= Sinal de controle dentro da escala de-5 até 5 V;
- OUT_{MAX}= 210;
- OUT_{MIN}= -210.

A escala de -210 até 210 foi determinada devido a soma da resolução do sinal de controle com o sinal de offset da zona morta dos motores não poder ser superior a resolução de 255. Essas características podem ser observadas em detalhes na rotina do Apêndice B.

3.8 Comunicação e Amostragem

Foi necessário amostrar o sinal da variável controlada θ e o sinal de controle $u(t)$. Esses sinais forma transmitidos via cabo USB, salvos em arquivos de texto e importados no MATLAB.

O Arduino é equipado com uma porta USB que permite conectá-lo a um computador por meio de um cabo USB padrão. Este cabo é usado para transferir dados e fornecer

energia ao Arduino. Ao conectar o Arduino ao computador pela primeira vez, geralmente é necessário instalar drivers USB específicos. Esses drivers permitem que o computador reconheça o Arduino como um dispositivo USB. Uma vez que o driver USB está instalado e o Arduino está conectado, o Arduino cria uma porta serial virtual no computador. Essa porta serial virtual é usada para enviar e receber dados entre o computador e o Arduino. Para programar o Arduino, utiliza-se a IDE (Ambiente de Desenvolvimento Integrado) do Arduino no computador. A IDE do Arduino é uma aplicação de software que permite escrever, compilar e carregar programas (sketches) para o Arduino. Para escrever um código no Arduino com o IDE basta clicar no botão "Carregar"(Upload) para enviar o código para o Arduino. A IDE se comunica com o Arduino através da porta serial virtual criada quando o Arduino está conectado via USB. O Arduino e o computador podem trocar informações serialmente usando a porta serial virtual. É possível usar funções como Serial.begin(), Serial.print(), e Serial.read() no código do Arduino para enviar e receber dados. A IDE do Arduino também possui uma ferramenta chamada "Monitor Serial"que permite visualizar os dados enviados pelo Arduino e os dados que o Arduino envia de volta para o computador. Isso é útil para depuração e visualização de resultados.

Um dos pontos mais importantes para ser avaliado em projeto de controle digital é definir qual que é a frequência de amostragem adequada para controlar o sistema, pois uma alta taxa de amostragem permite capturar variações sutis das variáveis de entrada e saída, permitindo que o controlador responda mais rápido a distúrbios do sistema além de ser mais preciso na captura dos dados, porém amostragem muito alta pode gerar atrasos de comunicação, pois quando os dados de sensores são transmitidos para o controlador e, em seguida, o controlador envia comandos para os atuadores, pode haver um atraso significativo entre a aquisição dos dados e a ação de controle. Isso pode levar a respostas mais lentas do sistema, gerar instabilidade e oscilações indesejadas. Já para uma baixa frequências de amostragem o atraso de comunicação é reduzido, o sistema fica mais estável, todavia amostragem muito baixa pode resultar na perda de informações importantes, especialmente em sistemas dinâmicos rápidos, resposta mais lenta a mudanças no sistema ou distúrbios, e o controle preciso pode ser prejudicado em sistemas que exigem uma resposta rápida a mudanças nas variáveis do sistema.

Tendo em vista isso foram realizados testes da resposta do sistema a diferentes frequências de amostragem que variaram entre 15 ms e 5 ms. Para fazer esse teste foi implementado um controlador proporcional usando o código do Apêndice B com um valor suficiente para fazer o robô oscilar por pelo menos 5 segundos sem cair. Inicialmente ajustado o Arduino na velocidade padrão de 9600 bauds, o ciclo de processamento era da ordem de 10 a 20 ms, o que determinava uma lenta reação do sistema ao erro de posição angular do pêndulo.

Para melhorar o tempo de execução, a velocidade de sinalização foi ajustada para

115200 bauds. Isso permitiu ajustar a frequência de amostragem para valores de até 3 ms.

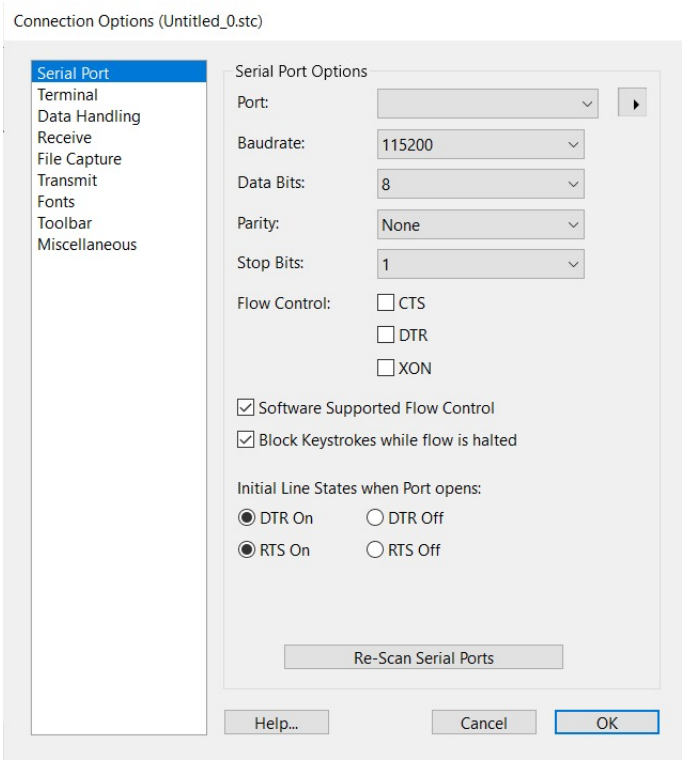
Na excursão de 15 ms para 5 ms a resposta foi piorando de tal forma que o robô acabava passando menos tempo em pé que o necessário para fazer o teste. Para valores menores que 5 ms o pêndulo oscilar de maneira muito brusca de tal maneira que acabava passando menos tempo em pé, ou seja, ficou menos estável.

Já com o código definido, o envio dos dados para a porta serial foi formatado no código do Apêndice B para que as variáveis de θ e sinal de controle fossem impressos no formato csv – comma separated values. Isso foi feito para facilitar que o MATLAB organizasse as variáveis em colunas. Além dessas duas variáveis, também foram enviados o número da amostra e o período do loop do algoritmo. O formato csv enviado para a porta serial foi:

$$< Númerodaamostra >, < teta >, < sinaldecontrole >, < dt >$$

Na plataforma do IDE não foi possível coletar os dados direto do monitor serial para um bloco de notas. Por esse motivo foi necessário utilizar um aplicativo chamado de **CoolTerm** para fazer esse trabalho. A configuração desse aplicativo se encontra na figura 60.

Figura 60 – O software CoolTerm utilizado para obtenção dos sinais

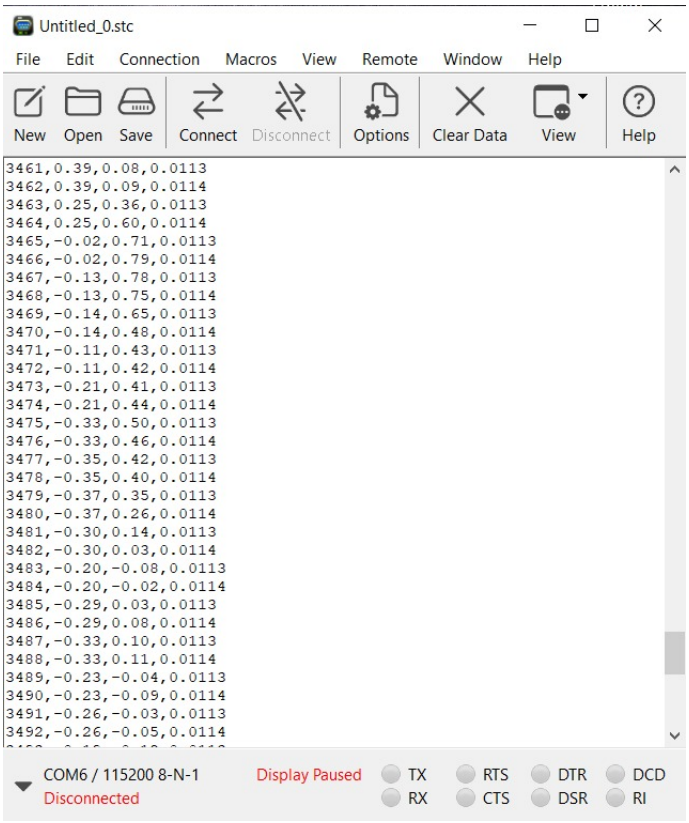


Fonte: Do autor

A figura 61 mostra o terminal do CoolTerm com as variáveis no formato programado

no código do apêndice B.

Figura 61 – Terminal com valores enviados para a porta serial



Fonte: Do autor

O aplicativo recebe os dados do Arduino e cria um bloco de notas com as variáveis no formato de CSV.

Após a implementação da comunicação, foi realizado um teste para aferir o offset do IMU na posição vertical, que se deve tanto à fabricação do dispositivo quanto à montagem no chassi. Esse offset precisa ser subtraído do valor do setpoint no controlador. O pêndulo foi colocado na posição vertical, transmitindo o θ sem ação motora, em 0°. O valor de ângulo lido no terminal é o valor de offset, que após sucessivas leituras ficou ajustado em 2,75°.

3.9 Modelagem da Planta, Projeto de Controlador Observado de Acordo com a Sessão 2.4

Primeiramente foi tentado conferir se a planta de quarta ordem era observável com os parâmetros após os ajustes do robô. Os parâmetros formam os da tabela 6. O resultado foi não observável. Partindo para o cálculo da planta pelo método de ordem reduzida foi constatado que era observável de acordo com o código A.7.

Tabela 6 – Parâmetros do robô pêndulo invertido sobre rodas após ajustes

Parâmetros	**Unidade**	**Valor**
g	m/s^2	$9,81$
L	m	$0,0575$
h	m	$0,115$
m_p	kg	$1,26$
J_p	kgm^2	$0,0059$
m_r	kg	$0,036$
J_r	kgm^2	$0,000017298$
r	m	$0,031$
Vn	V	12
R_a	Ω	$1,7$
I_a	A	$0,4$
N	rpm	280
T_m	Nm	$0,044$
K_e	Vs/rad	$0,39$
K_t	Nm/A	$0,11$

Fonte: (TEIXEIRA, 2023)

As matrizes da representação em espaço de estados calculadas podem ser vistas nas equações 58 e 59.

$$x = \begin{bmatrix} \psi \\ \omega \\ \phi \end{bmatrix} \tag{58}$$

$$A' = \begin{bmatrix} 0 & 0 & 1,0000 \\ 662,8912 & -7,9363 & 0 \\ 372,7013 & -3,4177 & 0 \end{bmatrix} \quad B' = \begin{bmatrix} 0 \\ 20,5570 \\ 8,8527 \end{bmatrix} \quad C' = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad D' = \begin{bmatrix} 0 \end{bmatrix} \tag{59}$$

Para determinar a localização dos polos do sistema foi calculado os autovalores da matriz A usando a função $eig()$ no Matlab. O resultado foi da equação 60.

$$eig(A') = \begin{bmatrix} -22,8946 & 16,7622 & -1,8040 \end{bmatrix} \tag{60}$$

Neste caso o sistema é instável por possuir um polo positivo.

Para calcular os polos conjugados do sistema com controlador foi definido um tempo de acomodação Ts=0.1 s e $\zeta = 0.7448$ e outro polo mais distante em -60. A partir desses polos foi definido os ganhos do controlador K mostrado na equação 61.

$$K = [910,1184, -96,8923, 239,9129]; \tag{61}$$

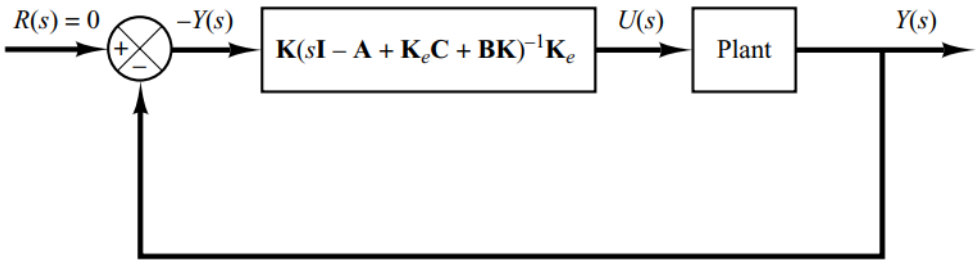
Fazendo o cálculo da matriz A em malha fechada com o ganho K a mesma se torna A_c, ou matriz A com controlador. A matriz $A_c = A - B * K$. Calculando os autovalores de A_c foi obtido os polos projetados, logo o sistema se tornou estável. Como não se tem acesso a todos os estados nesse trabalho será necessário projetar um observador. Esse observador também terá 3 polos, sendo eles os mesmos do controlador multiplicado por 4 para que convirja mais rápido. Os ganhos do observado neste caso será o da equação 62

$$L = [190,899940, -13050] \tag{62}$$

3.9.1 Discretização do Controlador e Observador em Espaço de Estados

Pra que seja possível discretizar o controlador e observador será necessário determinar a função de transferência em malha fechada do conjunto controlador-observador. segundo (OGATA et al., 2010) com as matrizes A, B, C e D da planta, o ganho K do controlador e o ganho L do observador definidos é possível calcular a função de transferência do controlador-observador a partir da operação mostrada na figura 62

Figura 62 – Representação em diagrama de blocos do sistema com controlador-observador.



Fonte: (OGATA et al., 2010)

Sendo K_e representado pelo (OGATA et al., 2010) o L modelado nesse trabalho. Aplicando esse transformação é obtido a seguinte função da equação 63.

$$C(s) = \frac{U(s)}{E(s)} = \frac{-9,015 \times 10^7 s^2 - 7,684 \times 10^8 s + 3,048 \times 10^{10}}{s^3 + 332,1 s^2 + 2,152 \times 10^4 s + 7,949 \times 10^8} \tag{63}$$

Sendo $C(s)$ a função de transferência do controlador-observador, $U(s)$ a saída do controlador que é um sinal de tensão para os motores e $E(s)$ o erro da posição angular medida com a referência de 0°. Para discretizar essa função de transferência é usada a transformada Z pelo método de Tustin com período de amostragem de 0.15 ms utilizando o comando c2d() no Matlab resultando na função de transferência da equação 65.

$$C(Z) = \frac{5,284 z^3 + 122,2 z^2 + 148 z + 31,05}{z^3 + 2,999 z^2 + 2,998 z + 0,9993} \tag{64}$$

Para implementar essa equação no Arduino será necessário isolar o valor atual de U(Z) em função de valores dele mesmo atrasado, das constantes e de E(Z) e seu atrasos. Para facilitar será substituído os valores das constantes por letras:

$$\frac{U(Z)}{E(Z)} = \frac{b_3 z^3 + b_2 z^2 + b_1 z + b_0}{z^3 + a_1 z^2 + a_1 z + a_0} \tag{65}$$

$$U(z) \cdot (z^3 + a_1 z^2 + a_1 z + a_0) = (b_3 z^3 + b_2 z^2 + b_1 z + b_0) \cdot E(z) \tag{66}$$

$$\begin{aligned} U(z) \cdot & (z^3 + a_1 z^2 + a_1 z + a_0) \\ & = (b_3 z^3 + b_2 z^2 + b_1 z + b_0) \cdot E(z) \end{aligned} \tag{67}$$

Multiplicando os dois lados da equação 67 pelo operador de atraso Z^-3 tem-se:

$$U(z) \cdot (1 + a_1 z^{-1} + a_1 z^-2 + a_0 z^-3) = (b_3 + b_2 z^{-1} + b_1 z^{-2} + b_0 z^-3) \cdot E(z) \tag{68}$$

$$\begin{aligned} U(z) = & -U(z) a_1 z^{-1} - U(z) a_1 z^{-2} - U(z) a_0 z^{-3} \\ & + E(z) b_3 + E(z) b_2 z^{-1} + E(z) b_1 z^{-2} + E(z) b_0 z^{-3} \end{aligned} \tag{69}$$

Em termos computacionais z^{-3}, z^{-2}, z^{-1} equivale a [n-3], [n-2], [n-1], ou seja, três amostras atrás a amostra atual, duas e uma respectivamente, logo a equação 69 munipulada será:

$$\begin{aligned} U(n) + U(n-1) a_1 + U(n-2) a_1 + U(n-3) a_0 \\ = E(n) b_3 + E(n-1) b_2 + E(n-2) b_1 + E(n-3) b_0 \end{aligned} \tag{70}$$

$$\begin{aligned} U(n) = & -U(n-1) a_1 - U(n-2) a_1 - U(n-3) a_0 \\ & + E(n) b_3 + E(n-1) b_2 + E(n-2) b_1 + E(n-3) b_0 \end{aligned} \tag{71}$$

$$\begin{aligned} U(n) = & -U(n-1) \cdot 2,998 - U(n-2) \cdot 2,998 - U(n-3) \cdot 0,993 \\ & + E(n) \cdot 5,284 + E(n-1) \cdot 122,2 + E(n-2) \cdot 148 + E(n-3) \cdot 31,05 \end{aligned} \tag{72}$$

A equação 72 é a que será aplica no algoritmo de controle no Arduino, considerando no primeiro instante de tempo, ou seja, n=0, os valores de U(n-3), U(n-2), U(n-1), E(n-3), E(n-2), E(n-1) todos iguais a zero. No segundo instante de tempo n=1, os valores de

U(n-3), U(n-2), E(n-3), E(n-2), todos iguais a zero. No segundo instante de tempo n=2, os valores de U(n-3) e E(n-3) serão iguais a zero. A parir de n=3 todos os valores da equação 72 serão conhecidos. O método de interação numérica usada para implementar a equação de diferenças 72 foi a de Euler. Com esse método a equação 72 fica da seguinte maneira:

$$\begin{aligned} U[0] = U[1] + T \cdot (&-2.998 \cdot U[1] - 2.998 \cdot U[2] - 0.993 \cdot U[3] \\ &+ 5.284 \cdot E[0] + 122.2 \cdot E[1] + 148 \cdot E[2] + 31.05 \cdot E[3]) \end{aligned} \tag{73}$$

Onde o valor de T é o mesmo usado como período de amostragem da transformada Z.

3.10 Projeto e Implementação no Arduino o Controlador PID pela Metodologia de Resposta em Frequêdia de Ziegle-Nichols

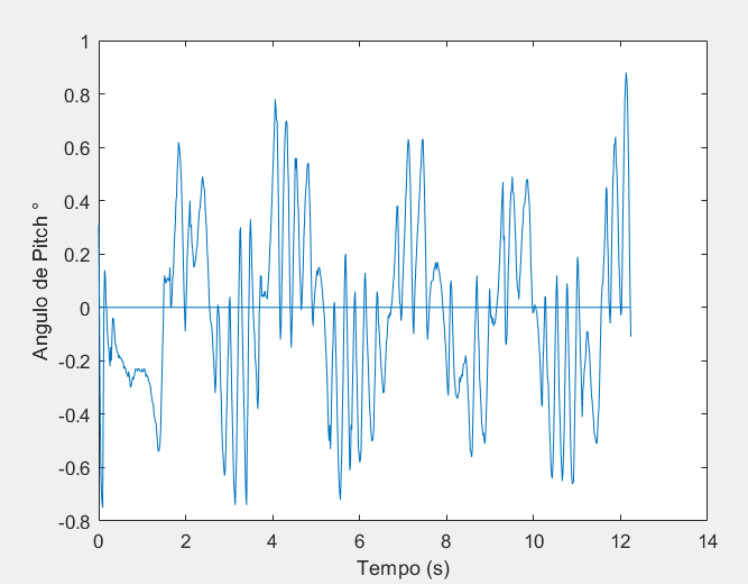
O primeiro passo para definir como foi feita a amostragem foi definir qual a melhor taxa de amostragem de forma que a estabilização da posição angular do robô seja mais fácil. Uma forma de determinar isso foi amostrar o sistema com um período de amostragem entre 5 ms a 15 ms. Para verificar a estabilização do sistema foi usado um controlador PID com seus parâmetros definidos pela metodologia de sintonia experimental de Ziegler-Nichols.

A aplicação da metodologia proposta será exemplificada aqui para um período de amostragem de $T_s = 0,015s$. O restante será apresentado apenas os resultados no capítulo 4.

O primeiro passo foi fechar a malha do sistema e aumentando o valor de K_p gradativamente no controlador PID digital implementado com a biblioteca PID v1 na plataforma IDE até um valor em que o ângulo do pêndulo θ apresentasse um formato periódico constante. O ideal seria que as rodas movimentassem para frente e para trás de maneira periódica sem que houvesse o deslocamento do centro de massa do pêndulo. Porém não foi possível manter o centro de massa parado sem que o sistema se instabilizasse de tal forma que impossibilitasse a coleta dos dados. O que foi possível fazer foi obter um movimento de vai e vêm das rodas com um deslocamento pequeno do centro de massa. Seguindo essa estratégia foi definido um valor de $K_{cr} = 1,02$. A resposta do sistema a essa constante retorna um sinal da posição do pêndulo θ de acordo com a figura 63.

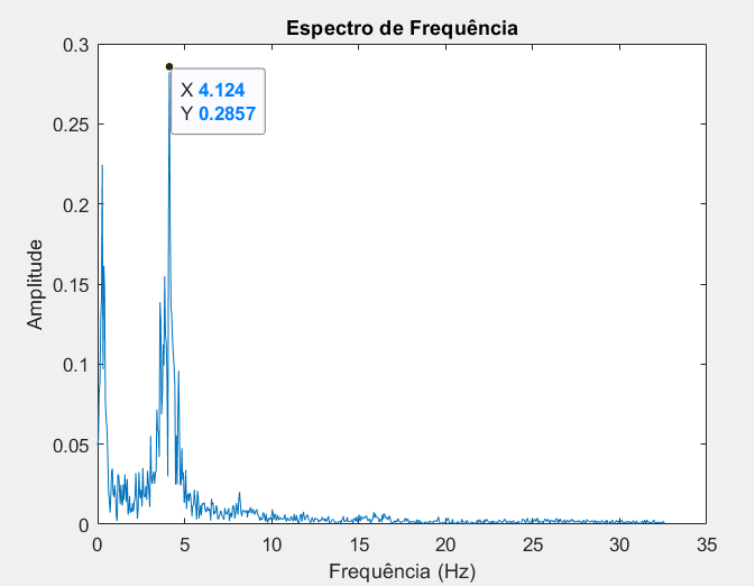
Pela figura 63 é possível observar uma certa periodicidade, porém a forma se encontra muito longe de uma senoide ideal. Como discutido na sessão 2.7, será aplicada a transformada de Fourier para determinar a componente de frequência com maior contribuição na formação do sinal para determinar o período crítico P_{cr}. O resultado é a figura 64.

Figura 63 – Sinal da posição do pêndulo θ para $T_s = 0,0015s$



Fonte: Do autor

Figura 64 – Transforma de Fourier do sinal da posição do pêndulo θ



Fonte: Do autor

Pela figura 64 temos que a frequência da função senoidal com maior amplitude é de 4.124 Hz. Nesse caso o período crítico $P_{cr} = 1/4.124$ s. Portanto $P_{cr} = 0{,}2425$ s.

De posse dos valores de $K_{ctr} = 1.02$ e $P_{cr} = 0,2425$ basta utlizar a tabela 2 e as equações 37, 38, e 39 para determinar os valores das constantes K_p, K_i e K_d. Para exemplificar será feito o cálculo para um controlador PID resultando nas equações 74, 75 e 76.

$$K_p = 0,6 \cdot K_{cr} = 0,6 \cdot 1,02 = 0,6120 \tag{74}$$

$$K_i = \frac{K_p}{T_i} = \frac{K_p}{2/P_{cr}} = \frac{K_p \cdot P_{cr}}{2} = 5,047 \tag{75}$$

$$K_d = K_p \cdot T_d = K_d \cdot 0,0125 \cdot P_{cr} = 0,0185 \tag{76}$$

Por fim foi aplicado as constantes das equações 74, 75 e 76 em uma biblioteca da plataforma IDE que discretizou essas constantes que inicialmente são contínuas para um sistema discreto implementado no Arduino. A resposta do sistema pode ser vista na figura 66 do capítulo 4.

4 ANÁLISE DOS RESULTADOS

4.1 Resultados da Aboragem Analítica

Uma das contribuições da metodologia analítica foi a definição de um fluxo de sinais para o robô, ilustrado na figura 45. Além disso, outra conquista significativa foi a efetiva implementação do algoritmo de controle no Arduino, conforme apresentado nos quadros 2, 3 e 4.

Quadro 2 – Inicialização das variáveis

```
// Condições inicais para controlador em espaço de estados
/*
  #define T 0.15  // Período de amostragem em segundos

  double U[4] = {0, 0, 0, 0};  // Inicialização de U(n), U(n-1), U(n-2), U(n-3)
  double E[4] = {0, 0, 0, 0};  // Inicialização de E(n), E(n-1), E(n-2), E(n-3)

*/
```

Fonte: Do Autor

Quadro 3 – Implementação do Controlador-Observador

```
/*
    void atualizarSaida(double erro) {
    // Atualizar arrays de entrada
    for (int i = 3; i > 0; i--) {
      E[i] = E[i-1];
    }
    E[0] = erro;

    // Calcular saída usando a equação discretizada
    U[0] = U[1] + T * (-2.998 * U[1] - 2.998 * U[2] - 0.993 * U[3] +
    5.284 * E[0] + 122.2 * E[1] + 148 * E[2] + 31.05 * E[3]);

     // Limitar a saída entre 12 e -12
     if (novaSaida > 12) {
       novaSaida = 12;
     } else if (novaSaida < -21) {
       novaSaida = -12;
     }

    // Atualizar arrays de saída
    for (int i = 3; i > 0; i--) {
      U[i] = U[i-1];
    }
  }

*/
```

Fonte: Do autor

Quadro 4 – Aplicação do sinal de controle nos motores

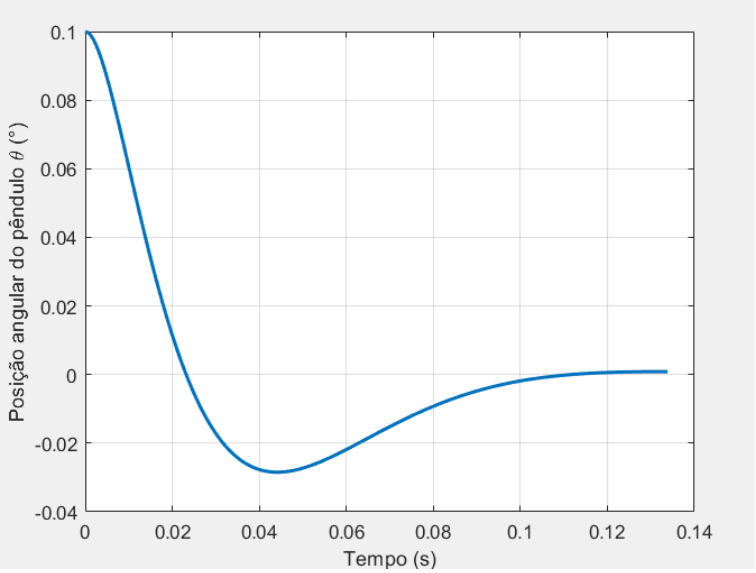
```
/* Cáculo da tensão em PWM do Controlador-Observador
   u_pwm = ((u[0] + 12) * (OUT_MAX - OUT_MIN) / 24) + OUT_MIN;
   u_pwm = abs(u_pwm);
*/
```

Fonte: Do autor

Os quadros 2, 3 e 4 apresentam as três partes do porcesso de implemntação do agorítmo de controle do controlador-obsevador discretizado. O quadro 2 apresenta a inicialização das variáveis, enquanto o quadro 3 descreve a implementação do controlador-observador e o quadro 4 detalha como o sinal de controle é aplicado nos motores.

A resposta esperada do sistema pode ser visto na figura 65. Nessa figura a posição do pêndulo parte de 0.01°, tem um overshoot de aproximadamente 3% e acomodação em aproximadamente em 0.1 s.

Figura 65 – Posição angular do pêndulo em função do tempo



Fonte: Do autor

A resposta do sistema á implementação do controlador-observador discreto não atingiu os requisitos de projeto desejados, haja vista que o sistema não se estabilizou. Dessa forma é necessário mais pesquisa para viabilizar a implementação desse método.

4.2 Resultados da Abordamgem Experimental

Para período de amostragem $T_s = 0,015s$, $K_{cr} = 1,02$ e $P_{cr} = 0,2425$, os valores das constantes dos controladores de acordo com a tabela 2 e as equações 37, 38, e 39 podem ser visto na tabela 7.

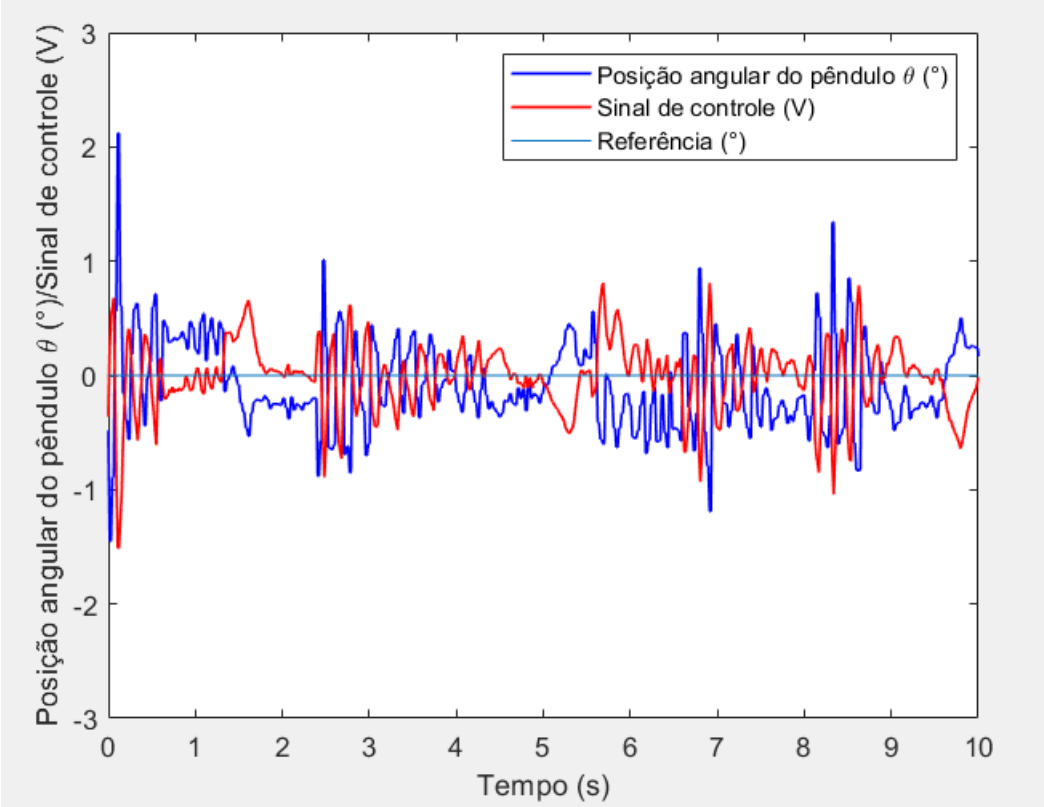
Tabela 7 – Parâmetros de diferentes controladores para $T_s = 0,015s$

Tipo de controlador	K_p	K_i	K_d
P	0,5100		
PI	0,4590	2,2713	
PID	0,6120	5,0474	0,0186

Fonte: Do autor

O controlador do tipo P e PI não obteve exito na estabilização do sistema. A resposta do sinal de controle do controlador PID e posição angular do pêndulo θ pode ser vista na figura 66.

Figura 66 – Sinais amostrados para $T_s = 0,015s$



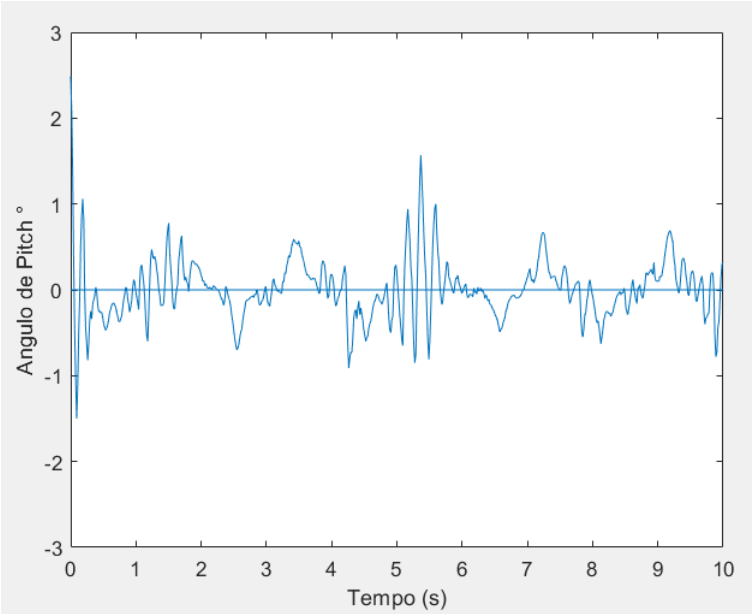
Fonte: Do autor

Desconsiderando dados de 1,5 s que foi o tempo que o sistema levou para se estabilizar, a média da posição angular deu -0,0707°, o desvio padrão foi de 0,43° e o desvio máximo do ponto de referência foi de 2,12°. A média mostra que apesar do desvio o

sistema trabalha circundando o ponto de equilíbrio de 0°, o que pode ser considerado uma boa resposta. O desvio padrão demostra que apesar de alguns desvios consideráveis do sinal, no padrão ela tende a permanecer dentro de uma faixa de -0,43° e 0,43°. O desvio máximo calculado foi de 2,12°, o que quer dizer que o controle não foi eficaz para manter o sistema dentro de uma faixa linear em torno do ponto zero, mas por outro lado, o sinal de controle conseguiu corrigir esse desvio, resultando em uma estabilização do sistema, mas não atendendo requisitos de projeto mais rigorosos como do método analítico.

Para amostragem com $T_s = 0,010s$ foi definido o $K_{cr} = 0,72$. Para essa constante o sinal da posição angular do pêndulo θ foi o da figura 67.

Figura 67 – Posição angular do pêndulo para $T_s = 0,010s$ e $K_{cr} = 0,72$



Fonte: Do autor

A transformada de Fourier para de sinal da figura 67 resulta na figura 68:

Pela figura 68 temos que a frequência da função senoidal com maior amplitude é de 4,195 Hz . Nesse caso o período crítico $P_{cr} = 1/4,195$ s. Portanto $P_{cr} = 0,2383$ s.

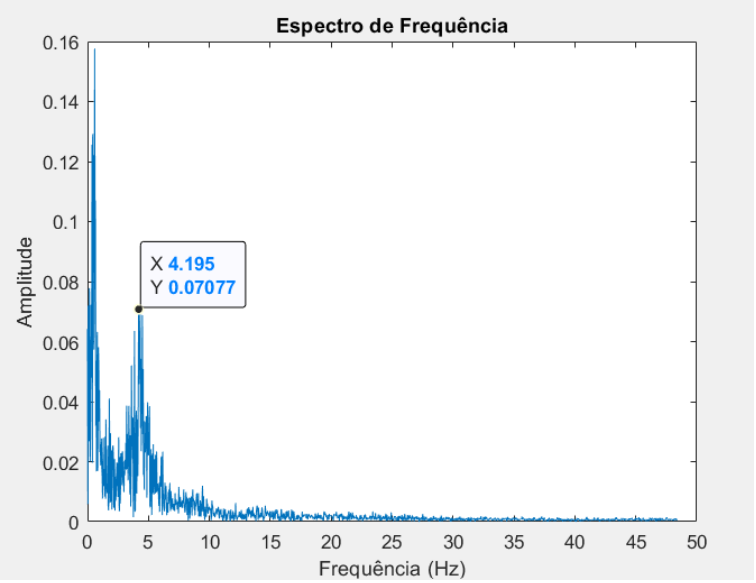
Para período de amostragem $T_s = 0,010s$, $K_{cr} = 0,72$ e $P_{cr} = 0,2383$, os valores das constantes dos controladores podem ser visto na tabela 8.

Tabela 8 – Parâmetros de diferentes controladores para $T_s = 0,010s$

Tipo de controlador	K_p	K_i	K_d
P	0,3600		
PI	0,3240	1,6316	
PID	0,4320	3,6257	0,0129

Fonte: Do autor

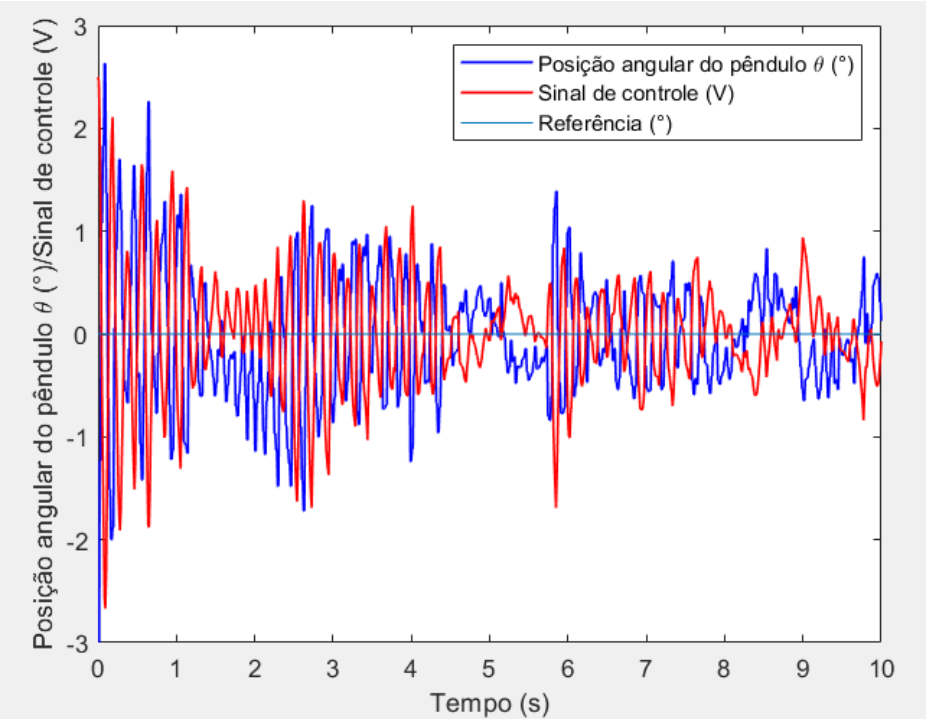
Figura 68 – Transformada de Fourier do sinal da posição angular do pêndulo θ para $T_s = 0,010s$ e $K_{cr} = 0,72$



Fonte: Do autor

O controlador do tipo P e PI não obteve exito na estabilização do sistema. A resposta do sinal de controle do controlador PID e posição angular do pêndulo θ pode ser vista na figura 69.

Figura 69 – Sinais amostrados para $T_s = 0,010s$



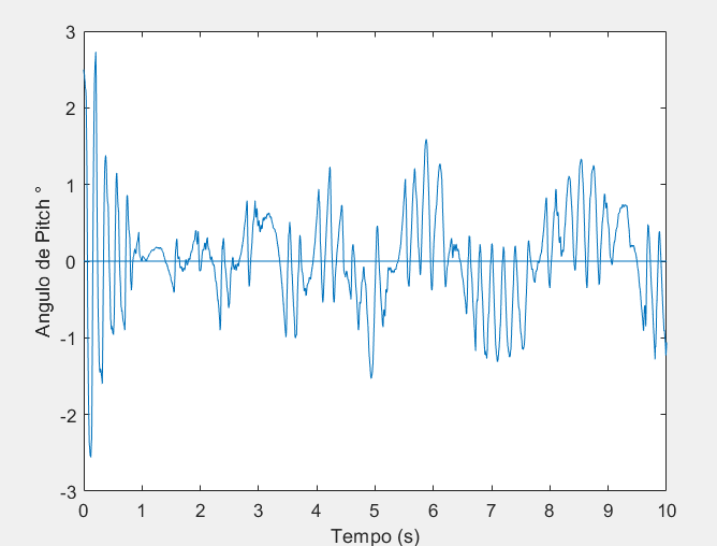
Fonte: Do autor

Desconsiderando dados de 1,5 s que foi o tempo que o sistema levou para se

estabilizar, a média da posição angular deu -0,0228°, o desvio padrão foi de 0,5743° e o desvio máximo do ponto de referência foi de 2.26°. Apesar de visivelmente essa segunda resposta ter mais desvios que a primeira, na média ela trabalhou mais próxima do ponto de operação desejado de 0°. O desvio padrão demostra que no padrão ele trabalha dentro de uma faixa de -0,5743° e 0,5743°. O desvio máximo calculado foi de 2,26°, o que quer dizer que o controle não foi eficaz para manter o sistema dentro de uma faixa linear em torno do ponto zero, mas por outro lado, o sinal de controle conseguiu corrigir esse desvio, resultando em uma estabilização do sistema, mas não atendendo requisitos de projetos mais rigorosos como do método analítico.

Para amostragem com $T_s = 0,0053s$ foi definido $K_{cr} = 0,0475$. Para essa constante o sinal da posição angular do pêndulo θ foi o da figura 70.

Figura 70 – Posição angular do pêndulo θ para $T_s = 0,0053s$ e $K_{cr} = 0,0475$



Fonte: Do autor

A tranformada de Forier do sinal da figura 70 foi o da figura 71:

Pela figura 71 temos que a frequência da função senoidal com maior amplitude é de 4,568 Hz . Nesse caso o período crítico $P_{cr} = 1/4,568$ s. Portanto $P_{cr} = 0,2189s$

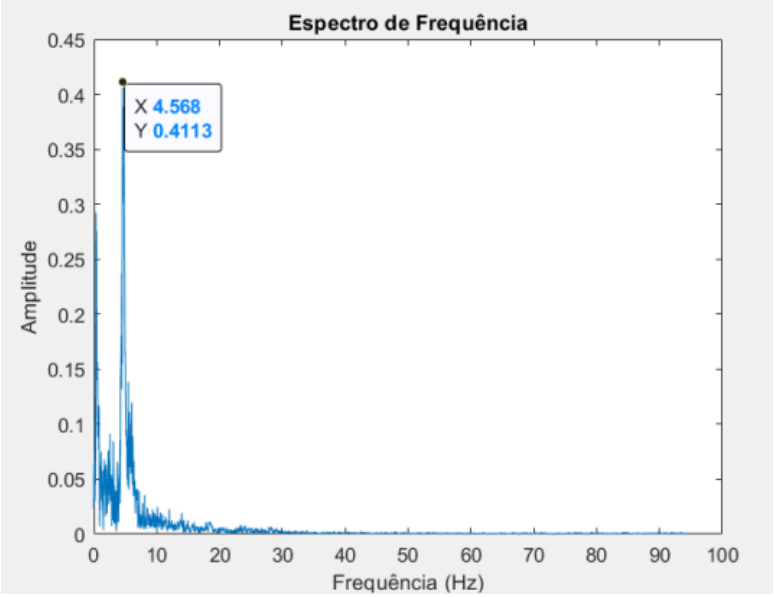
Para período de amostragem $T_s = 0,0053s$, $K_{cr} = 0,475$ e $P_{cr} = 0,2189$, os valores das constantes dos controladores podem ser visto na tabela 9.

Tabela 9 – Parâmetros de diferentes controladores para $T_s = 0,0053s$

Tipo de controlador	K_p	K_i	K_d
P	0,2375		
PI	0,2137	1,1718	
PID	0,2850	2,6039	0,0078

Fonte: Do autor

Figura 71 – Transformada de Fourier do sinal da posição angular do pêndulo θ para $T_s = 0,0053s$ e $K_{ctr} = 0,0475$

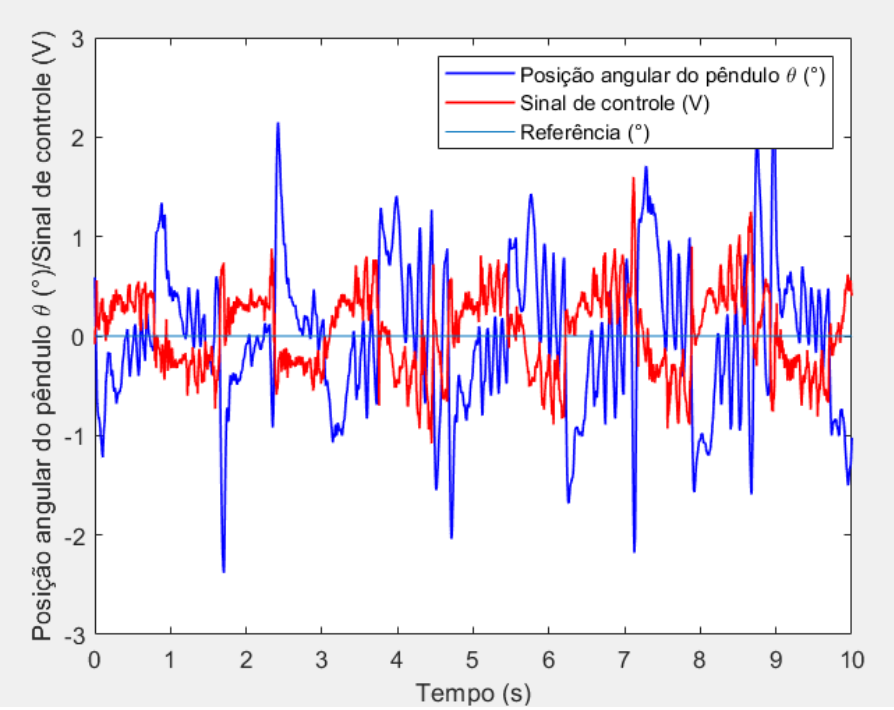


Fonte: Do autor

.

O controlador do tipo P e PI não obteve exito na estabilização do sistema. A resposta do sinal de controle do controlador PID e posição angular do pêndulo θ pode ser vista na figura 72.

Figura 72 – Sinais amostrados $T_s = 0,0053s$

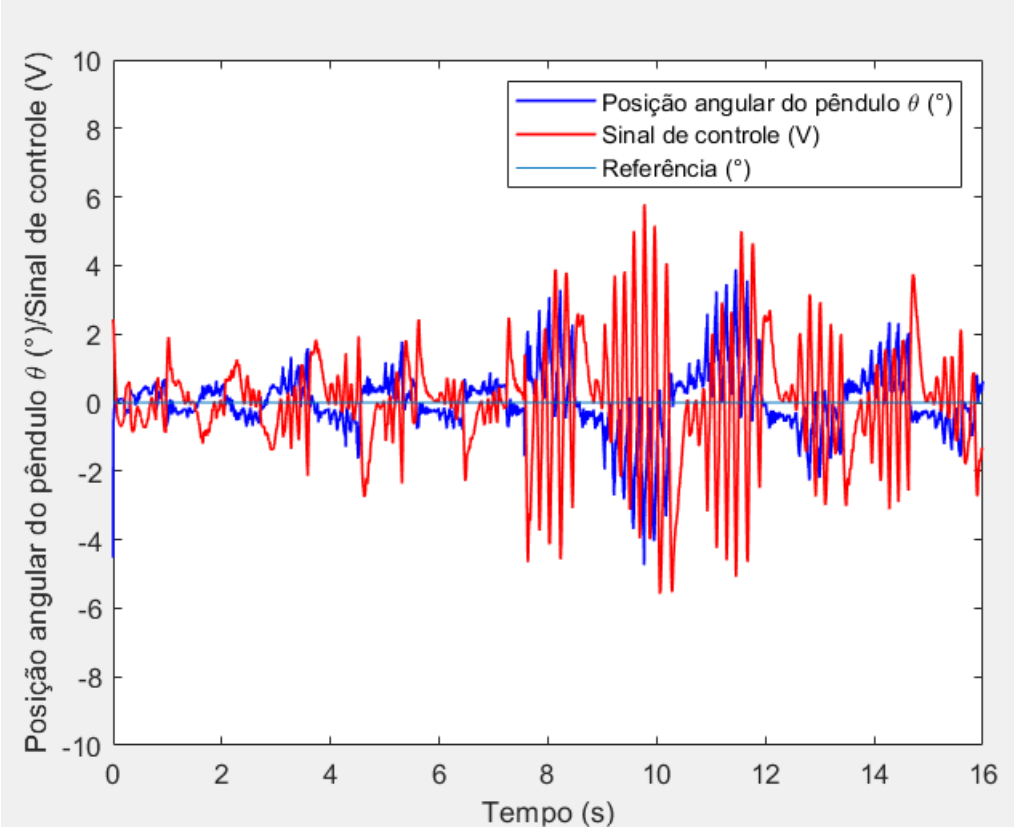


Fonte: Do autor

Desconsiderando dados de 1,5 s que foi o tempo que o sistema levou para se estabilizar, a média da posição angular deu -0,0171°, o desvio padrão foi de 0,6688° e o desvio máximo do ponto de referência foi de 2,38°. Apesar de visivelmente essa terceira resposta ter mais desvios que a primeira e a segunda, na média ela trabalhou mais próxima do ponto de operação desejado de 0° que todas. O desvio padrão demostra que no padrão ela trabalhou dentro de uma faixa de -0,6688° e 0.6688°, sendo o maior entre as três respostas. O desvio máximo calculado foi de 2,26°, o que quer dizer o controle não foi eficaz para manter o sistema dentro de uma faixa linear em torno do ponto zero, mas por outro lado, o sinal de controle conseguiu corrigir esse desvio, resultando em uma estabilização do sistema, mas não atendendo requisitos de projeto mais rigorosos como do método analítico.

A resposta a uma perturbação do controlardor PID com período de amostragem de 15 ms pode ser visto na figura 73

Figura 73 – Sinais amostrados $T_s = 0,015s$



Fonte: Autor

A figura 73 demonstra que o controlador projetado tem uma certa resistência a perturbaçoes externas pois em aproximadamente 8 s foi aplicado uma perturbação e em aproximadamente 12 s houve uma estabilização. Todavia essa estabilização se encontra muito longe da ideal, pois o sistema esta trabalhando muito longe da zona linear.

5 CONCLUSÃO

Tendo em vista que a abordagem analítica não alcançou o comportamento esperado, é necessário realizar uma revisão minuciosa da modelagem do sistema do robô e validar esse modelo em uma planta real. Isso permitirá identificar os motivos pelos quais o controlador-observador projetado não atendeu aos requisitos de projeto definidos. Essa revisão é sugerida para trabalhos futuros.

Quanto à abordagem experimental, foi possível atingir a estabilização do sistema para todos os períodos de amostragem testados, sendo o período de $T_s = 0,015s$ a melhor resposta devido ao menor desvio padrão. Isso indica que, entre os três sinais amostrados no padrão, ele opera mais próximo do valor de referência, denotando um controle mais adequado. Importa destacar que o estudo aprofundado do funcionamento dos sensores, a implementação de processos de filtragem, fusão de sensores, ajuste da estrutura física e da configuração física dos componentes, bem como o ajuste do offset de acionamento dos motores, foram de suma importância para atingir o objetivo principal, que era manter o robô na vertical em regime permanente de forma autônoma. Contudo, apesar da estabilização do sistema, a resposta apresenta sobressinais indesejados, o que, para sistemas de controle que exigem uma garantia de trabalho dentro de uma faixa de segurança mais rigorosa, não seria atendido.

Além disso, o projeto de controlador por metodologia experimental não permite uma definição de requisitos de projeto mais rigorosos, servindo apenas para uma sintonia inicial com o objetivo de estabilizar o sistema. É necessário ter cautela na utilização desse método, visto que a técnica exige que o sistema seja induzido a trabalhar muito próximo de uma faixa de instabilidade. Para este projeto, tal situação não gera muitos prejuízos caso o robô venha a se desestabilizar e cair, mas em casos de aplicação no transporte de pessoas seria inadmissível.

Por fim, foi alcançado um avanço significativo na jornada do projeto de controle do robô, tornando viável futuramente a definição de requisitos de projeto mais rigorosos. A intenção é que este protótipo didático possa evoluir em um futuro próximo para um projeto que beneficie a sociedade.

5.1 Dificuldades Encontradas

Na tabela 10 encontra-se uma lista das principais dificuldades e das soluções definidas nesse trabalho para cada uma delas.

Tabela 10 – Dificuldades e Soluções

Dificuldades	**Soluções**
Dificuldade de compreender como seria a implementação do controlador-observador de forma prática no robô.	Transformar o controlador-observador em espaço de estados para uma função de transferência a qual foi discretizada e implementada no código do Arduino.
Saber se era possível estabilizar o robô da figura 4 com um controlador PID.	Método de Ziegle-Nichols afirma que é possível estabilizar um sistemas que apresentem resposta oscilatória sustentada em malha fechada com aplicação de um ganho proporcional na malha direta.
Definir o algoritmo que deveria ser aplicado no Arduino para medir a posição angular do pêndulo com precisão e que possibilitasse fazer testes de malha fechada com um controlador proporcional na malha direta.	Adaptação do código de (MESQUITA et al., 2014) que demonstrou medir a posição angular do pêndulo com uma boa precisão devido à aplicação de filtro complementar e possibilitou fazer o testes necessários.
Mesmo com o algoritmo adaptado de (MESQUITA et al., 2014) e implementado, o robô não apresentava resposta oscilatória sustentada em malha fechada em resposta ao aumento gradativo do ganho proporcional na malha direta.	Foi necessário projetar a estrutura do robô e a configuração dos componentes eletrônicos de acordo com a metodologia da sessão 3.3 e definir um offset de controle dos motores de acordo com a tabela 5.
Foi encontrada dificuldades para definir os ganhos do controlador PID a partir dos parâmetros K_{cr}, P_{cr}, T_i, T_d.	Definição das equações 74, 75 e 76 que calculam as constantes para um controlador em tempo contínuo e as equações 37, 38, 39 para o tempo discreto.
Mesmo após a aplicação correta da metodologia de Ziegle-Nichols, não estava sendo ajustado o tempo de atualização dos cálculos do controlador para que fosse mais rápido que o loop do algoritmo de controle.	Ajustar esse tempo na parte do código $robot.SetSampleTime(T_s)$ para valores menores ou iguais ao período de amostragem experimentados.

5.2 Trabalhos Futuros

Para melhoria dos resultados encontrados nesse trabalho fica como sugestão revisar minunciosamente a metodologia analítica e a aplicação dela na prática para descobrir os pontos que devem ser ajustados para viabilizar a mesma. A utilização de alguma técnica de identificação dos parâmetros dos motores DC e de outro parâmetros construtivos do

robô também seria um bom ponto a ser melhorado. Outro ponto seria a utilização de uma comunicação via bluetooth para evitar interferências do cabo na resposta do sistema e utilizar o encoder dos motores para medir a velocidade angular das rodas ω. Além disso fica a sugestão para investigação de outras técnicas de controle como por exemplo o controle Fuzzie, um controlador PID para controle da posição angular do pêndulo e outro da velocidade das rodas ou controle que considere três graus de liberdade.

5.3 Compartilhamento dos Dados Experimentais Coletados

Na referência (OLIVEIRA,) é possível acessar os dados experimentais coletados que compõem os resultados desta monografia.

Referências

AGUIRRE, L. A. *Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais*. [S.l.]: Editora UFMG, 2004. Citado 2 vezes nas páginas 51 e 52.

BALANDUINO - Robô Pêndulo Invertido. 2022. <https://balanduino.tkjelectronics.dk/>. Citado na página 13. Citado na página 17.

BOUBAKER, O. The inverted pendulum: A fundamental benchmark in control theory and robotics. In: *2012 International Conference on Education and e-Learning Innovations, ICEELI 2012*. Sousse, Tunisia: [s.n.], 2012. Citado 2 vezes nas páginas 17 e 18.

CéLULA UR18650A Sanyo 2200mAh - Kit 50 unidades. Acessado em 6 de dezembro de 2023. Disponível em: <https://produto.mercadolivre.com.br/MLB-1566673875-celula-ur18650a-sanyo-2200mah-kit-50-unidades-_JM>. Citado na página 60.

DEGOND, P.; DIEZ, A.; NA, M. *Estados Topológicos em Volume em um Novo Modelo de Dinâmica Coletiva*. Trabalho de Conclusão de Curso — Université de Toulouse, Janeiro 2021. Citado na página 47.

INVENSENSE INC. MPU-6000 and MPU-6050 Product Specification. [S.l.]: [s.n], 2012. Sunnyvale, CA. Citado na página 56.

KADAM, S. R.; PAWAR, S. N. Development of cost effective motion capture system based on arduino. In: IEEE. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. [S.l.], 2020. Citado na página 46.

KARA, T.; EKER, I. Nonlinear modeling and identification of a dc motor for bidirectional operation with real time experiments. *Energy Conversion and Management*, 2004. Citado 2 vezes nas páginas 71 e 72.

KEYESTUDIO MEGA 2560 Compatible Board Advanced. Acessado em 6 de dezembro de 2023. Disponível em: <https://wiki.keyestudio.com/Ks0342_Keyestudio_MEGA_2560_Compatible_Board_Advanced>. Citado na página 57.

KOK, M.; HOL, J. D.; SCHöN, T. B. Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, v. 11, n. 1-2, p. 1–153, 2017. Citado 6 vezes nas páginas 46, 47, 48, 49, 50 e 51.

MERINO, E. M. *Diseño y control de un sistema electromecánico inestable de dos grados de libertad: aplicación al caso de un péndulo invertido*. Dissertação (Trabajo de Conclusión de Curso) — Universitat Politècnica de Catalunya, 2015. Citado na página 57.

MESQUITA, A. L. A. et al. A methodology for the transient behavior of horizontal axis hydrokinetic turbines. *Energy Conversion and Management*, v. 87, p. 1261 – 1268, 2014. ISSN 0196-8904. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0196890414005433>. Citado 3 vezes nas páginas 53, 71 e 94.

MOTAVALLI, J. *G.M. EN-V: Sharpening the Focus of Future Urban Mobility*. 2010. Published on The New York Times, March 24, 2010. Disponível em: <https://archive.nytimes.com/wheels.blogs.nytimes.com/2010/03/24/g-m-en-v-sharpening-the-focus-of-future-urban-mobility/>. Citado na página 19.

MOTOR DC Arduino com Ponte H L298N. Acessado em 6 de dezembro de 2023. Disponível em: <https://www.makerhero.com/blog/motor-dc-arduino-ponte-h-l298n/>. Citado na página 58.

MóDULO Acelerômetro 3 Eixos MPU6050. 2023. Acessado em 6 de dezembro de 2023. Disponível em: <https://curtocircuito.com.br/modulo-acelerometro-3-eixos-mpu6050.html>. Citado na página 56.

OGATA, K. et al. *Modern control engineering*. [S.l.]: Prentice hall Upper Saddle River, NJ, 2010. v. 5. Citado 12 vezes nas páginas 17, 23, 24, 34, 36, 37, 39, 40, 42, 44, 45 e 80.

OLIVEIRA, A. M. de. *Dados Experimentais*. Acessado em 6 de dezembro de 2023. Disponível em: <https://github.com/alexandrebeller/TCC_ALEXANDRE_UFPA.git>. Citado na página 95.

Open Impulse. *JGA25-371 DC Gearmotor with Encoder*. Acessado em 6 de dezembro de 2023. Disponível em: <https://www.openimpulse.com/blog/products-page/25d-gearmotors/jga25-371-dc-gearmotor-encoder-201-rpm-12-v-2/>. Citado na página 59.

PINTO, J. E. M. G. *Aplicação prática do método de sintonia de controladores PID utilizando o método do relé com histerese*. 116 p. Dissertação (Dissertação) — Universidade Federal do Rio Grande do Norte, Natal, 2014. Citado 2 vezes nas páginas 38 e 39.

SEGWAY. *Segway S-Pod*. 2023. Disponível em: <https://www.segway.com/segway-s-pod/>. Citado na página 19.

SILVA, G. D.; OUTROS. *Projeto, modelagem, simulação, construção e controle de um veículo de duas rodas pêndulo invertido*. 2017. Trabalho de Conclusão de Curso. Citado 2 vezes nas páginas 63 e 65.

STMicroelectronics. *L298 Dual Full-Bridge Driver*. <https://www.st.com/en/motor-drivers/l298.html>. Acessado em 6 de dezembro de 2023. Citado na página 68.

SUPORTE Case para 3 Baterias 18650 em Série com Rabicho. Acessado em 6 de dezembro de 2023. Disponível em: <https://produto.mercadolivre.com.br/MLB-3816666290-suporte-case-para-3-baterias-18650-em-serie-com-rabicho-_JM>. Citado na página 60.

TEIXEIRA, E. H. C. G. et al. *Controles Típicos de Equipamentos e Processos Industriais*. [S.l.]: Editora Blucher, 2010. Citado na página 38.

TEIXEIRA, T. G. P. *Modelagem e Controle De Robô Pêndulo Invertido Sobre Rodas*. Monografia de Graduação em Engenharia Elétrica — Universidade Federal do Pará, 2023. Citado 13 vezes nas páginas 18, 19, 20, 21, 27, 28, 29, 30, 31, 32, 33, 36 e 79.

ÅSTRöM, K. J.; HäGGLUND, T. *PID Controllers: Theory, Design, and Tuning*. S.l.: ISA-The Instrumentation, Systems and Automation Society, 1995. Citado na página 37.

APÊNDICE A – CÓDIGOS MATLAB 2016a

A.1 Códigos para construção da Figura 47

```
clear,clc,close all

g=9.81;

M = 0.5;
m = 0.25;

l1 = 0.05;
l2 = 0.1;


num1 = [1];
den1=[(-M*l1) 0 (M+m)*g];
G1 = tf(num1,den1);

num2 = [1];
den2=[(-M*l2) 0 (M+m)*g];
G2 = tf(num2,den2);

subplot(2,2,1)
step(G1)
title('Resp. ao Degrau (l=0.06)')
xlim([0 0.5])
ylim([-10 0])

subplot(2,2,2)
rlocus(G1)
title('Resp. ao Degrau (l=0.06)')
xlim([-50 50])

subplot(2,2,3)
step(G2)
title('Resp. ao Degrau (l=0.1)')
xlim([0 0.5 ])
```

```
ylim([-10 0])

subplot(2,2,4)
rlocus(G2)
title('Resp. ao Degrau (l=0.1)')
```

A.2 Códigos para montar gráfico do sinal da posição do pêndulo θ

```
    clear, clc

Dados = load('Controlador_P_0.475_ts_0.0053_filtro_complementar_ver01.txt');
y = Dados(:,3);
u = Dados(:,2);
t = Dados(:,4);



y_plot= y(1:2000);
u_plot=u(1:2000);

Ts = mean(t); % Tempo de amostragem
N = length(y_plot);  % Número total de amostras
t = linspace(0, (N-1)*Ts, N);



plot(t, y_plot); % 'o-' especifica marcadores de ponto e uma linha
hold on;

% Adicionar uma linha horizontal no valor zero
line([min(t), max(t)], [0, 0]);

% Adicionar título e legendas
%title('Gráfico de Exemplo');
xlabel('Tempo (s)');
ylabel('Angulo de Pitch °');

ylim([-3, 3]);
xlim([0, 10]);

% Adicionar legenda
```

```
%legend('Série de Dados');
hold off;
```

A.3 Códigos para montar a figura 22

```
clear, clc

% Parâmetros das senoidais
A1 = 1;     % Amplitude da primeira senoide
A2 = 0.7;   % Amplitude da segunda senoide
A3 = 0.5;   % Amplitude da terceira senoide

f1 = 2;     % Frequência da primeira senoide (Hz)
f2 = 4;     % Frequência da segunda senoide (Hz)
f3 = 6;     % Frequência da terceira senoide (Hz)

phi1 = 0;   % Fase da primeira senoide
phi2 = pi/2; % Fase da segunda senoide
phi3 = pi;   % Fase da terceira senoide

% Vetor de tempo
tempo = linspace(0, 2, 1000); % 1000 pontos entre 0 e 2 segundos
Ts = tempo(2) - tempo(1);     % Tempo de amostragem

% Equações senoidais
y1 = A1 * sin(2 * pi * f1 * tempo + phi1);
y2 = A2 * sin(2 * pi * f2 * tempo + phi2);
y3 = A3 * sin(2 * pi * f3 * tempo + phi3);

% Soma das senoidais
soma = y1 + y2 + y3;

% Calcular a Transformada de Fourier
N = length(tempo);
frequencias = linspace(0, 1/(2*Ts), N/2+1);
transformada = fft(soma)/N;
amplitude_espectro = 2*abs(transformada(1:N/2+1));

% Plotar as senoidais e a soma
```

```
figure;

subplot(3, 1, 1);
plot(tempo, y1, 'r', tempo, y2, 'g', tempo, y3, 'b');
title('Três Senoidais Separadas');
xlabel('Tempo (s)');
ylabel('Amplitude');
legend('Senoide 1', 'Senoide 2', 'Senoide 3');

subplot(3, 1, 2);
plot(tempo, soma, 'k');
title('Soma das Senoidais');
xlabel('Tempo (s)');
ylabel('Amplitude');

subplot(3, 1, 3);
plot(frequencias, amplitude_espectro, 'm');
title('Espectro de Frequência');
xlabel('Frequência (Hz)');
ylabel('Amplitude');

% Ajustar o eixo x para exibir até 5 Hz
xlim([0 8]);

sgtitle('Sinais no Domínio do Tempo e da Frequência');
```

A.4 Códigos para montar gráfico da transformada de Fourier

```
clc; clear; close all

ensaio = load('Controlador_P_0.475_ts_0.0053_filtro_complementar_ver01.txt');
teta = ensaio(:,3);
time = ensaio(:,4);
u = ensaio(:,2);

n = length(time);
dt = mean(time);

tempo = [0:dt:(n-1)*dt]';
```

```
fs = 1/dt;
df = fs/n;
f = 0:df:(n/2-1)*df;

XT= fft(teta,n)*2/n;
plot(f,abs(XT(1:n/2)));

title('Espectro de Frequência');
xlabel('Frequência (Hz)');
ylabel('Amplitude');
```

A.5 Códigos para calculas as constantes do controlador PID

```
Kcrt = 0.475;

Pcrt = 1/4.568

% P

Kp = 0.5*Kcrt;

% PI

%Kp2 = 0.45*Kcrt
%Ki2 = Kp2*(1.2/Pcrt)

%PID

Kp3 = 0.6*Kcrt
Ki3 = Kp3*(2/Pcrt)
Kd3 = Kp3*(0.125*Pcrt)
```

A.6 Códigos para montar os gráficos dos sinais amostrados da posição angular do pêndulo θ

```
clear, clc

Dados = load('Controlador_P_0.475_ts_0.0053_filtro_complementar_ver01.txt');
y = Dados(:,3);
u = Dados(:,2);
t = Dados(:,4);



y_plot= y(1:2000);
u_plot=u(1:2000);

Ts = mean(t); % Tempo de amostragem
N = length(y_plot);  % Número total de amostras
t = linspace(0, (N-1)*Ts, N);



plot(t, y_plot); % 'o-' especifica marcadores de ponto e uma linha
hold on;

% Adicionar uma linha horizontal no valor zero
line([min(t), max(t)], [0, 0]);

% Adicionar título e legendas
%title('Gráfico de Exemplo');
xlabel('Tempo (s)');
ylabel('Angulo de Pitch °');

ylim([-3, 3]);
xlim([0, 10]);

% Adicionar legenda
%legend('Série de Dados');
hold off;
```

A.7 Código para calcular a função de transferência discreta do controlador-observador

```
clear, clc

% Símbolos
syms g Jp mp L  % Parâmetros do pêndulo;
syms Jr mr r    % Parâmetros da roda;
syms Ra Kt Ke n % Parâmetros do motor;
syms u psi phi omega %theta % Variáveis;

% Modelo de ordem reduzida no Espaço de Estado:
% Vetor de estados:
x = [  psi;     % Posição do pêndulo;
     omega;     % Velocidade da roda;
       phi];    % Velocidade do pêndulo;

% Equações de estado:
f2 = phi;

f3 = (L*r*mp*cos(psi)*(g*L*Ra*mp*sin(psi) + 2*n*Kt*(u-Ke*omega)) + ...
       Jp*(2*n*Kt*(u-Ke*omega) -L*r*Ra*mp*(phi^2)*sin(psi)))/...
       (Ra*(Jp*(Jr+(r^2)*(mp+mr)) -(L*r*mp*cos(psi))^2));

f4 = (Jr*(g*L*Ra*mp*sin(psi) + 2*n*Kt*(u-Ke*omega)) + ...
      r*(L*r*Ra*mp*sin(psi)*(mp*(g-L*(phi^2)*cos(psi)) + g*mr) + ...
      2*n*Kt*(u-Ke*omega)*(mp*(L*cos(psi)+r)+r*mr)))/...
      (Ra*(Jp*(Jr+(r^2)*(mp+mr)) -(L*r*mp*cos(psi))^2));

% Modelo dinâmico não-linear:
dx = [f2;
      f3;
      f4];

% Linearização (Jacobiano):
A = jacobian(dx,x);
B = jacobian(dx,u);

% Ponto de operação:
```

```
psi = 0;
omega = 0;
phi = 0;

% Parâmetros:
g = 9.81;               % Gravidade;
L = 0.0575;               % Comprimento do pêndulo;
r = 0.031;              % Raio da roda;
h = 0.115;               % Altura;
mp =  1.26;              % Massa do pêndulo;
mr = 0.036;             % Massa da roda;
Ra = 1.7;               % Resistência da armadura;
n = 1/21.3;             % Fator da caixa de redução;
Vn = 12;                % Tensão nominal;
Ian = 0.046;              % Corrente nominal;
N = 201;                % RPM
Ke = (Vn-Ian*Ra)*60/(2*pi*N);   % Constante elétrica;
Tm = 0.25965;                     % Torque;
Kt = Tm/Ian;                    % Constante de torque;
Jp = (1/12)*(mp*L^2) + (1/3)*(mp*h^2);  % Inércia do pêndulo;
Jr = mr*r^2/2;                  % Inércia da roda;
%
% Substitui os valores nas matrizes:
A = double(vpa(subs(A),6));
B = double(vpa(subs(B),6));
C = [1 0 0 ];
D = 0;

% Cálculo da funsão de transferência da planta
[num,den] = ss2tf(A,B,C,D);
C = eye(3);
G = tf(num,den);

save ModeloLinear3 A B C D G

%

clear, clc
```

```
load('ModeloLinear3.mat')

% AnÃ¡lise de Observabilidade:
C = [1 0 0];
Ob = obsv(A,C);
% O sistema Ã© observÃ¡vel.
det(Ob)
%% Análise de controlabilidade

% Construir a matriz de controlabilidade
Co = ctrb(A, B);

% Verificar se o sistema é controlável
if rank(Co) == size(A, 1)
    disp('O sistema é controlável.');
else
    disp('O sistema não é controlável.');
end

%%
% Polos desejados do controlador:
ts = 0.1;
qsi = 0.7448;
wn = 4/(ts*qsi);
p1 = -qsi*wn+1i*wn*sqrt(1-qsi^2);
p2 = -qsi*wn-1i*wn*sqrt(1-qsi^2);
p3 = -150;

polos = [p1, p2, p3];

%Ganho do controlador
K = acker(A,B,polos);

% Cálculo das matrizes em malha fechada
Ac = A-B*K;
Bc = B;
Cc = C;
Dc = D;
```

```
% Plote da posição angular do pêndulo a condições iniciais

WIP = ss(Ac,Bc,Cc,Dc);

x0 = [0.1;0;0];

[y,t] = initial(WIP,x0);

plot(t,y(:,1),'linewidth',1.8)
grid

xlabel('Tempo (s)');
ylabel('Posição angular do pêndulo \theta (°)');

%% Plote da resposta do sistema para condições iniciais
%   para as três vairáveis de estado
subplot(2,2,1),
plot(t,y(:,1),'linewidth',1.8)
grid
subplot(2,2,2),
plot(t,y(:,2),'linewidth',1.8)
grid
subplot(2,2,3),
plot(t,y(:,3),'linewidth',1.8)
grid
subplot(2,2,4),
plot(t,y(:,4),'linewidth',1.8)
grid


%% Análise de Observabilidade:

% Definição da matriz C:
C = [1 0 0];

% Construir a matriz de observabilidade
Os = obsv(A, C);

% Verificar se o sistema é observável
```

```
if rank(Os) == size(A, 1)
    disp('O sistema é observável.');
else
    disp('O sistema não é observável.');
end

%%

% Polos desejados do observador:
ts = 0.1;
qsi = 0.7448;
wn = 4/(ts*qsi);
po1 = -4*qsi*wn + 1i*wn*sqrt(1-qsi^2);
po2 = -4*qsi*wn - 1i*wn*sqrt(1-qsi^2);
po3 = -2*p3;

Po = [po1 po2 po3];

%% Projeto do observador:

L = acker(A',C',Po)'

%% Malha fechada controlador observador

Ac_o = A-B*K-L*C
Bc_o = L
Cc_o = K
Dc_o = 0

%%
syms s

[numC,denC] = ss2tf(Ac_o,Bc_o,Cc_o,Dc_o)

ControladorS = tf(numC,denC)

Ts = 0.15
ControladorZ = c2d(ControladorS,Ts,'tustin')
```

APÊNDICE B – CÓDIGO ARDUINO DO PÊNDULO INVERTIDO

```
//Inclusão das Bibliotecas
#include <PID_v1.h>
#include <Wire.h>

// Condições inicais para controlador em espaço de estados
/*
  #define T 0.15  // Período de amostragem em segundos

  double U[4] = {0, 0, 0, 0};  // Inicialização de U(n), U(n-1), U(n-2), U(n-3)
  double E[4] = {0, 0, 0, 0};  // Inicialização de E(n), E(n-1), E(n-2), E(n-3)

*/

//Direção I2C da IMU
#define MPU 0x68

//Razões de Conversão
#define A_R 16384.0
#define G_R 131.0

//Conversão de Radianos para Graus 180/PI
#define RAD_A_DEG = 57.295779

//O MPU-6050 fornece os valores em inteiros de 16 bits.
//Valores não refinados
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

//Ângulos
float Pitch[2]; // Ângulo de Pitch
float Gy[2];
float TetaM[2]; // Posição angular do pêndulo Theta medido

// Pinos do driver
int IN1=3;
int IN2=4;
int IN3=9;
```

```
int IN4=10;

// Tempo de sistema
uint32_t timer;
// Variável de tempo de loop
float dt;
// Contador de amostras
unsigned long n = 1;

double Teta, sal, sald;
//CONFIGURAÇÃO DO LOOP PID
//Definir as variáveis que vamos conectar
double sp, setpoint, offset=-2.54, input, u_pwm, erro, u;
//Definir os parâmetros de ajuste

double Kp=0.2850, Ki=2.6039 , Kd=0.0078;

//Especificar os vínculos dos parâmetros de ajuste
PID robot(&erro,&u,&sp,Kp,Ki,Kd,DIRECT);

void setup(){

  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(115200);

  //Configuração dos pinos do driver
  pinMode (IN1,OUTPUT); //Input 1 conectada al pin 3
  pinMode (IN2,OUTPUT); //Input 2 conectada al pin 4
  pinMode (IN3,OUTPUT); //Input 3 conectada al pin 9
  pinMode (IN4,OUTPUT); //Input 4 conectada al pin 10

  //CONFIGURAÇÃO DO LOOP PID
  delay(3000);
  Serial.println(sp);
  setpoint = TetaM[1]; //variavel de entrada PID
```

```
  sp=0; //setpoint PID
  robot.SetMode(AUTOMATIC); //ativa o PID
  robot.SetSampleTime(5); // Intervalo entre os cálculos
  robot.SetOutputLimits(-5,5);

  timer = micros();

}

void loop() {
  //Ler os valores do Acelerômetro do IMU
  Wire.beginTransmission(MPU);
  Wire.write(0x3B); //Solicitar o registro 0x3B - correspondente ao AcX
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true); //A partir do 0x3B, são solicitados 6 registros
  AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
  AcY=Wire.read()<<8|Wire.read();
  AcZ=Wire.read()<<8|Wire.read();

  // Calcula o delta t [s]
  dt = (double)(micros() - timer) / 1000000;
  timer = micros();

  // O ângulos de pitch é calculados
  Pitch[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) +
  pow((AcZ/A_R),2)))*RAD_TO_DEG;

  //Ler os valores do Giroscópio
  Wire.beginTransmission(MPU);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,4,true);
  //Ao contrário do acelerômetro, são solicitados apenas 4 registros.
  GyX=Wire.read()<<8|Wire.read();
  GyY=Wire.read()<<8|Wire.read();

  //Ler os valores do Giroscópio
  Gy[1] = GyY/G_R;
```

```
 //Aplicar o Filtro Complementar
 TetaM[1] = 0.985 *(TetaM[1]+Gy[1]*0.010) + 0.015*Pitch[1];

 //Configuração do Loop de Controle PID.
 input= TetaM[1];
 Teta = input - offset;
 erro = Teta;


 /*
     void atualizarSaida(double erro) {
     // Atualizar arrays de entrada
     for (int i = 3; i > 0; i--) {
       E[i] = E[i-1];
     }
     E[0] = erro;

     // Calcular saída usando a equação discretizada
     U[0] = U[1] + T * (-2.998 * U[1] - 2.998 * U[2] - 0.993 * U[3]
     + 5.284 * E[0] + 122.2 * E[1] + 148 * E[2] + 31.05 * E[3]);

       // Limitar a saída entre 12 e -12
       if (novaSaida > 12) {
         novaSaida = 12;
       } else if (novaSaida < -21) {
         novaSaida = -12;
       }

     // Atualizar arrays de saída
     for (int i = 3; i > 0; i--) {
       U[i] = U[i-1];
     }
   }

 */

robot.Compute();

 const int OUT_MAX = 210;
```

```
const int OUT_MIN = -210;

if(abs(erro)>0.01){
if(erro<0.01){
sal=10;
sald=4;
digitalWrite(9,LOW);
digitalWrite(3,LOW);
}

if(erro>0.01)
{
sal=9;
sald=3;
digitalWrite(10,LOW);
digitalWrite(4,LOW);
}
u_pwm = ((u + 5) * (OUT_MAX - OUT_MIN) / 10) + OUT_MIN;
u_pwm = abs(u_pwm);

/* Cáculo da tensão em PWM do Controlador-Observador
   u_pwm = ((u[0] + 12) * (OUT_MAX - OUT_MIN) / 24) + OUT_MIN;
   u_pwm = abs(u_pwm);
*/


if (erro<0.01){

  analogWrite(sal, u_pwm+14);//Roda direita para ângulo negativo
  analogWrite(sald, u_pwm+18);//Roda esquerda para ângulo negativo
  }
  else{
  analogWrite(sal, u_pwm+15);//Roda direita para ângulo positivo
  analogWrite(sald, u_pwm+15);//Roda esquerda para ângulo positivo
  }


}else{
digitalWrite(10,LOW);
```

```
 digitalWrite(9,LOW);
 digitalWrite(4,LOW);
 digitalWrite(3,LOW);
 }

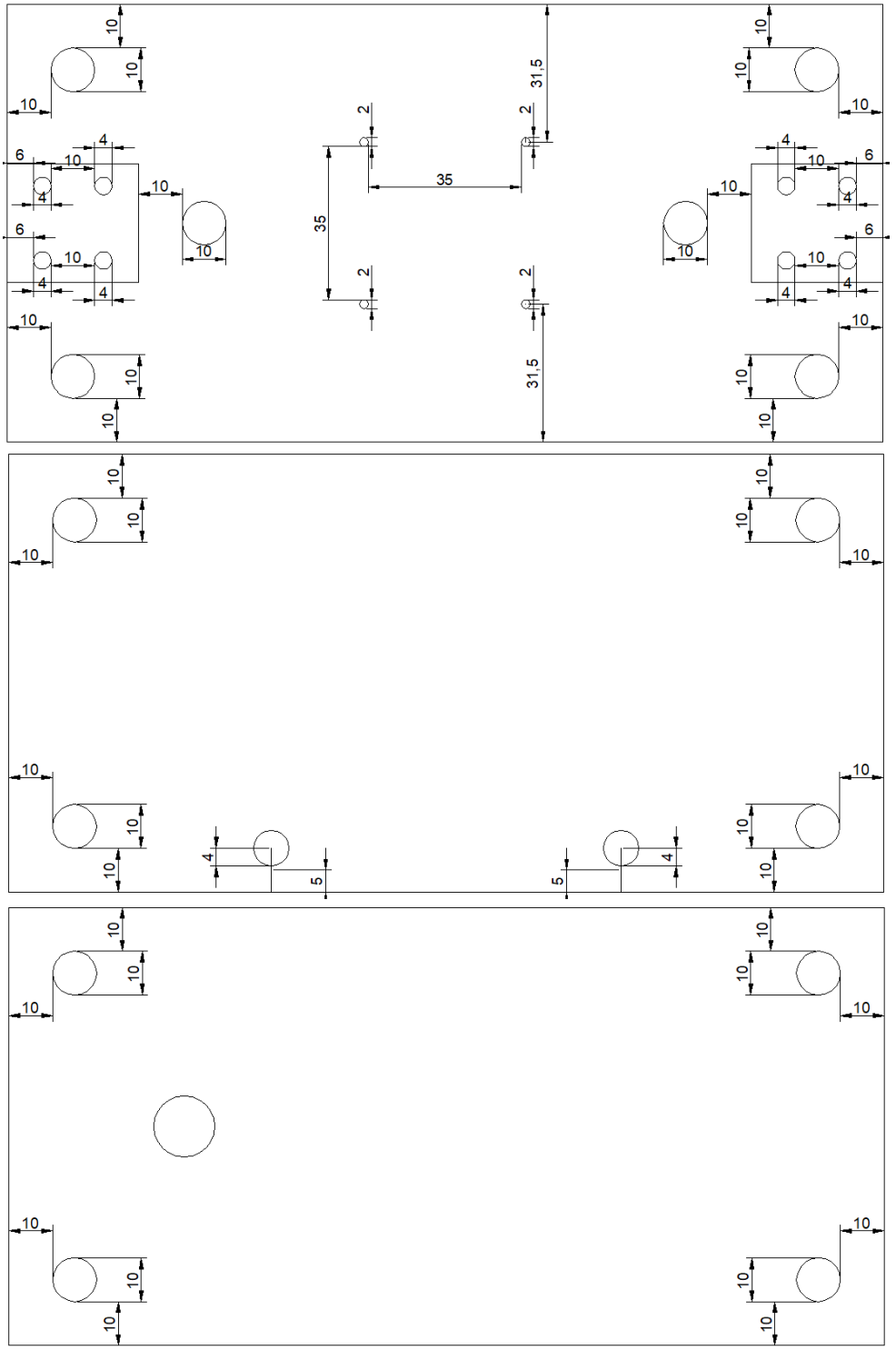
 // Exibir os valores no monitor serial.
 //Serial.println(String(n) + "," + String(u)
 + "," + String(Teta) + "," + String(dt, 4));
 Serial.println(String(n) + "," + String(Teta)
 + "," + String(u) + "," + String(dt, 4));

 delay(2); // Ts = 15 ms = delay de 12
 // Ts = 10 ms = delay de 7// Ts = 5 ms = delay de 2

 n++;
}
```

APÊNDICE C – DETALHAMENTO DOS PISOS DO ROBÔ

Figura 74 – - Placas dos pisos do pêndulo móvel: placa do primeiro segundo e terceiro piso



Fonte: Do autor

APÊNDICE D – LISTA DE MATERIAIS E FERRAMENTAS

Tabela 11 – Lista de Materiais

Material	**Quantidade**
Porcas M12 para fixação com diâmetro interno rosqueado 10 mm	12
Parafusos rosqueados M12 com 23 cm de altura e 10 mm de diâmetro	4
Arruelas para fixação com diâmetro interno de 15 mm e externo de 30 mm	12
Placas de acrílico 10 cm x 20 cm x 3 mm de espessura	3
Protoboard 830 Furos	1
Terminal elétrico tipo pino isolado brando 0.25 mm	2
Terminal elétrico tipo pino isolado vermelho 0.25 mm	2
Chave gangorra redonda Liga/Desliga	1
Arduino Mega 2560	1
Sensor inercial MPU 6050	1
Ponte H L298N	1
Kit Motor JGA25-371 DC 6V 280RPM com Encoder e Roda com Base de fixação com parafusos	1
Suporte case para 3s Baterias 18650 em série com molas e fios	1
Baterias UR18650A M19A	3
Jumper Premium 40p x 20cm - Macho / Macho	1
Jumper Premium 40p x 20cm - Fêmea / Fêmea	1
Kit Jumper Premium 10p x 20cm - Macho / Fêmea	1
Fita Dupla Face 3m Fixa Forte Extrema 24mm X 2m	1
Tubo de Solda Estanho 1mm 63x37 Cobix 22g	1
Cabo USB para Arduino	1
Computador com configurações mínimas para executar o programa Arduino IDE	1

Fonte: Do autor

Tabela 12 – Lista de Ferramentas

Ferramenta	**Quantidade**
Chave Combinada 14mm	1
Alicate Universal 9 Tramontina	1
Alicate de Corte Diagonal 6 Tramontina	1
Alicate Crimpador de Terminais 0,25 a 6mm	1
Multímetro Digital Minipa Et-1002	1
Ferro De Solda 60w	1
Carregador Universal de Bateria Lithium De 3,7v 18650	1
Chave Sextavada Para Eletrônica 1mm em L	1
Mini Kit Ferramentas De Precisão Fenda Philips	1
Régua de 30 cm	1

Fonte: Do autor