

Rafael Dias Pereira

Projeto e Implementação de um Pêndulo Invertido: Montagem e Controle

Uberlândia, MG

2024

Rafael Dias Pereira

Projeto e Implementação de um Pêndulo Invertido: Montagem e Controle

Projeto de Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Campus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Universidade Federal de Uberlândia - UFU
Faculdade de Engenharia Elétrica - FEELT

Orientador Prof. Éder Alves de Moura

Uberlândia, MG

2024

Resumo

Este trabalho aborda o desenvolvimento e controle de um sistema de pêndulo invertido, um modelo amplamente utilizado na engenharia de controle devido à sua natureza instável e complexidade dinâmica. O objetivo principal é projetar um controlador que estabilize o pêndulo em posição vertical sobre um carrinho móvel, compensando distúrbios externos e mantendo o equilíbrio. A planta pêndulo invertido foi projetada, construída, modelada e identificada (por meio de validações experimentais) para o desenvolvimento o projeto do sistema de controle. Foi adotado o formalismo de Lagrange para a obtenção do equacionamento da dinâmica do sistema, resultando em um sistema de equações diferenciais não linear. A partir do modelo não linear, realizou-se a linearização das equações de movimento ao redor do ponto de equilíbrio para viabilizar a implementação do controlador, utilizando o método do Regulador Linear Quadrático – do inglês *Linear Quadratic Regulator* (LQR). O controle físico da planta consistiu de uma interface implementada com o microcontrolador ESP32, operando como intermediário entre o sistema eletrônico e a lógica de controle, implementada no ambiente embarcado. Os resultados experimentais demonstraram a eficácia do controlador LQR em manter o pêndulo estável, comprovando a aplicabilidade da abordagem adotada. Este trabalho permitiu avaliar uma metodologia de prática e validada para o controle de sistemas instáveis, servindo de base para futuras pesquisas e aplicações em sistemas dinâmicos similares.

Palavras-chave: Pêndulo Invertido, Controle LQR, Modelagem Matemática, Linearização, Simulação, Estabilidade.

Abstract

This work addresses the development and control of an inverted pendulum system, a model widely used in control engineering due to its instability and dynamic complexity. The main objective is to design a controller that stabilizes the pendulum in a vertical position on a moving cart, compensating for external disturbances and maintaining balance. The inverted pendulum plant was designed, built, modeled, and identified (through experimental validations) for the control system project development. The Lagrange formalism was adopted to derive the dynamic equations of the system, resulting in a set of nonlinear differential equations. Based on the nonlinear model, the equations of motion were linearized around the equilibrium point to enable controller implementation using the Linear Quadratic Regulator (LQR) method. The physical control of the plant involved an interface implemented with the ESP32 microcontroller, acting as an intermediary between the electronic system and the control logic implemented in the embedded environment. Experimental results demonstrated the effectiveness of the LQR controller in keeping the pendulum stable, confirming the applicability of the adopted approach. This work allowed for the evaluation of a practical and validated methodology for controlling unstable systems, providing a foundation for future research and applications in similar dynamic systems.

Keywords: Inverted Pendulum, LQR Control, Mathematical Modeling, Linearization, Simulation, Stability.

Lista de ilustrações

| | |
|---|----|
| Figura 1 – Sistema de controle ótimo. | 18 |
| Figura 2 – Motor DC 775 utilizado no sistema | 21 |
| Figura 3 – Driver BTS7960 | 22 |
| Figura 4 – Microcontrolador ESP32 | 23 |
| Figura 5 – Encoder rotativo incremental utilizado no sistema | 24 |
| Figura 6 – Guia linear com carrinho utilizada no sistema. | 24 |
| Figura 7 – Haste para o pêndulo. | 25 |
| Figura 8 – Peças impressas em 3D para o acoplamento dos encoders. | 26 |
| Figura 9 – Diagrama de conexões do sistema de pêndulo invertido. | 27 |
| Figura 10 – Fluxograma da Lógica de Controle Implementada no ESP32 | 29 |
| Figura 11 – Visão geral do sistema completo montado. | 30 |
| Figura 12 – Pêndulo invertido. | 31 |
| Figura 13 – Modelo não linear do pêndulo invertido implementado no Simulink. | 36 |
| Figura 14 – Configuração experimental para medição da força exercida pelo motor. | 37 |
| Figura 15 – Reta de melhor ajuste traçada sobre os pontos experimentais. | 39 |
| Figura 16 – Oscilação e decaimento do pêndulo após deslocamento inicial. | 40 |
| Figura 17 – Comparação dos picos da oscilação com a curva ajustada. | 41 |
| Figura 18 – Comparação entre o sistema real e a simulação com $b = 0.0030749$ | 42 |
| Figura 19 – Comparação entre o sistema real e a simulação com $b = 0.00001$ | 43 |
| Figura 20 – Comparação entre sistema real e simulação com $c = 20$ | 44 |
| Figura 21 – Comparação entre sistema real e simulação com $c = 30$ | 45 |
| Figura 22 – Comparação entre sistema real e simulação com $c = 24$ | 45 |
| Figura 23 – Resposta do sistema a um impulso unitário por 250 ms, comparando o sistema real e o simulado. | 47 |
| Figura 24 – Sinal de controle aplicado ao motor. | 53 |
| Figura 25 – Comportamento do ângulo do pêndulo ao longo do tempo. | 53 |
| Figura 26 – Posição do carrinho ao longo do tempo. | 54 |
| Figura 27 – Velocidade angular do pêndulo ao longo do tempo. | 55 |
| Figura 28 – Velocidade do carrinho ao longo do tempo. | 55 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Comparação dos motores testados | 21 |
| Tabela 2 – Parâmetros do Sistema | 37 |
| Tabela 3 – Dados de Identificação da Força Exercida pelo Motor | 38 |
| Tabela 4 – Tabela de cálculo para a regressão linear entre tensão e força. | 38 |
| Tabela 5 – Parâmetros do sistema completo. | 46 |

Sumário

| | | |
|------------|---|-----------|
| 1 | INTRODUÇÃO | 8 |
| 1.1 | Objetivos do Trabalho | 9 |
| 1.2 | Metodologia | 9 |
| 1.2.1 | Construção da Planta Física | 10 |
| 1.2.2 | Modelagem Matemática do Sistema | 10 |
| 1.2.3 | Identificação Experimental de Parâmetros | 10 |
| 1.2.4 | Simulação e Projeto das Leis de Controle | 10 |
| 1.3 | Implementação do Sistema de Controle e Validação Experimental | 11 |
| 1.4 | Organização do Trabalho | 11 |
| 2 | REFERENCIAL TEÓRICO | 12 |
| 2.1 | Modelagem e Representação do Sistema Dinâmico | 12 |
| 2.1.1 | Teste de Controlabilidade | 14 |
| 2.1.2 | Discretização | 15 |
| 2.1.3 | Estabilidade no Espaço de Estados | 15 |
| 2.2 | Identificação Paramétrica Experimental do Modelo do Sistema | 16 |
| 2.3 | Controlador LQR | 17 |
| 3 | CONSTRUÇÃO DO DISPOSITIVO | 20 |
| 3.1 | Construção Mecânica | 20 |
| 3.2 | Diagrama de Conexões do Hardware | 26 |
| 3.3 | Algoritmo Implementado no ESP32 | 27 |
| 3.4 | Montagem Experimental | 30 |
| 4 | MODELAGEM E IDENTIFICAÇÃO | 31 |
| 4.1 | Representação do Sistema | 31 |
| 4.2 | Equações de Movimento | 32 |
| 4.2.1 | Energia Potencial e Cinética | 32 |
| 4.2.2 | Equações de Euler-Lagrange | 33 |
| 4.3 | Representação em Espaço de Estados | 34 |
| 4.4 | Implementação do Modelo Não Linear no Simulink | 35 |
| 4.5 | Identificação dos Parâmetros do Sistema | 36 |
| 4.5.1 | Medidas Diretas | 37 |
| 4.5.2 | Identificação da Conversão de Tensão para Força | 37 |
| 4.5.3 | Identificação do Coeficiente de Atrito no Eixo do Pêndulo (<i>b</i>) | 39 |
| 4.5.4 | Identificação do Coeficiente de Atrito do Carrinho com a Guia Linear (<i>c</i>) | 43 |

| | | |
|-------|--|----|
| 4.6 | Parâmetros e Verificação do Modelo Final | 46 |
| 5 | SISTEMA DE CONTROLE | 48 |
| 5.1 | Linearização do Modelo Não Linear | 48 |
| 5.2 | Implementação do Controlador LQR | 51 |
| 5.2.1 | Ajuste das Ponderações Q e R | 51 |
| 5.2.2 | Resultados Experimentais e Análise | 52 |
| 6 | CONCLUSÃO | 56 |
| 6.1 | Trabalhos Futuros | 56 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 58 |
| | ANEXO A – CÓDIGO IMPLEMENTADO NO ESP32 | 60 |

1 Introdução

O sistema de pêndulo invertido é amplamente reconhecido como um dos principais desafios no campo do controle de sistemas devido à sua natureza inerentemente instável e ao comportamento não linear. Este sistema exige um controle preciso para ser mantido em equilíbrio, servindo como uma plataforma didática ideal para o desenvolvimento e teste de técnicas avançadas de controle. Segundo [Cheng et al. \(2016\)](#), o pêndulo invertido é um sistema multivariável com forte acoplamento e instabilidade, sendo utilizado para ilustrar conceitos de estabilidade, controlabilidade e eficiência.

Aplicações práticas do pêndulo invertido incluem o desenvolvimento de robôs de equilíbrio automático e o controle de atitude de foguetes e satélites. Por exemplo, sistemas de robôs móveis autobalanceados, como apresentado por [Iacob \(2020\)](#), utilizam princípios do pêndulo invertido para manter o equilíbrio, enfrentando desafios como rejeição de distúrbios e rastreamento de referência. Além disso, o controle de atitude de veículos espaciais e a estabilização de plataformas em aeronaves utilizam métodos inspirados no controle de pêndulos invertidos, onde é necessário manter uma orientação precisa, mesmo sob perturbações externas ([COŞKUN, 2020](#)). Esses exemplos ressaltam a importância do pêndulo invertido como um modelo para o desenvolvimento de algoritmos de controle eficazes.

Diversas técnicas de controle têm sido utilizadas no controle do pêndulo invertido, sendo o Regulador Quadrático Linear – do inglês *Linear Quadratic Regulator* (LQR) – como estratégia amplamente utilizada para o controle ótimo do dispositivo. O LQR representa uma formulação de otimização o esforço de controle enquanto minimiza o erro no sistema para uma planta linear (ou linearizada) ([LENKA, 2011](#)), e é aplicado em uma grande variedade de sistemas. Conforme discutido por [Nasir, Ahmad e Rahmat \(2008\)](#), a abordagem LQR têm se mostrado eficaz ao fornecer uma resposta rápida e performática em sistemas de pêndulo invertido, sendo comparativamente superior a outras abordagens de controle lineares, como o controlador Proporcional-Integrativo-Derivativo (PID).

Assim, visando detalhar o processo o desenvolvimento em todas as suas etapas, este trabalho apresenta o projeto e a implementação de um controlador LQR para um pêndulo invertido, incluindo a construção da planta física e a modelagem matemática do sistema, tendo seus parâmetros identificados experimentalmente. A modelagem matemática do sistema foi realizada utilizando a formulação de Lagrange, com derivação de equações diferenciais ordinárias (EDOs) e representação do sistema no espaço de estados. A linearização do modelo ao redor do ponto de equilíbrio permitiu a aplicação de controladores lineares. Projetou-se o LQR e validou-se experimentalmente a eficiência do controlador

ótimo implementado.

1.1 Objetivos do Trabalho

O objetivo geral deste trabalho é desenvolver um sistema de controle utilizando a formulação LQR para manter o equilíbrio de um sistema de pêndulo invertido montado sobre um carrinho móvel. Abaixo são listados os objetivos específicos:

- **Construção da Planta Física:** Construir uma planta física do pêndulo invertido, assegurando que todos os componentes sejam precisos e bem integrados para experimentação e análise.
- **Modelagem Matemática:** Derivar o modelo matemático não-linear e identificar, experimentalmente, os parâmetros do sistema físico para obter uma representação fidedigna da dinâmica.
- **Linearização e Representação em Espaço de Estados:** Derivar uma formulação linearizada do modelo em torno do ponto de equilíbrio e sua representação em **Lagrange** e no espaço de estados, facilitando a aplicação de controladores como o LQR.
- **Desenvolvimento e Implementação do Controlador LQR:** Projetar e ajustar um controlador LQR para estabilizar o pêndulo invertido, minimizando o desvio do ponto de equilíbrio e otimizando o esforço de controle.
- **Validação Experimental:** Validar o desempenho na planta física, verificando a estabilidade e a resposta em tempo real.

Este estudo contribui para a área de controle de sistemas ao desenvolver e validar uma abordagem prática e eficaz para o controle de um sistema de pêndulo invertido.

1.2 Metodologia

Este trabalho foi desenvolvido em etapas sequenciais, que abrangem a construção da planta física do pêndulo invertido, a modelagem matemática do sistema, a linearização das equações de movimento e a implementação de um controlador adequado para estabilizar o pêndulo em sua posição vertical. A seguir, são descritas as principais fases do desenvolvimento do projeto.

1.2.1 Construção da Planta Física

A primeira etapa consistiu na construção da planta física do pêndulo invertido. Para isso, foram selecionados componentes mecânicos e eletrônicos específicos, capazes de atender às exigências de precisão e estabilidade do sistema. A estrutura foi montada com um carrinho sobre uma guia linear, permitindo o movimento horizontal controlado. Um motor DC foi utilizado para acionar o carrinho, enquanto um encoder rotacional de alta precisão foi acoplado ao eixo do pêndulo para medir seu ângulo em tempo real. O sistema foi controlado por um microcontrolador ESP32, responsável por ler os dados dos sensores e executar o algoritmo de controle.

1.2.2 Modelagem Matemática do Sistema

Com a planta construída, a próxima etapa foi a modelagem matemática do sistema. Utilizando o formalismo de Lagrange, foram derivadas as equações de movimento que descrevem a dinâmica do sistema. Essas equações representam o comportamento do pêndulo invertido em termos de suas variáveis de estado, como posição do carrinho e ângulo do pêndulo, e incluem parâmetros como massa e comprimento do pêndulo. Esse modelo foi essencial para compreender a dinâmica do sistema e fundamentar o desenvolvimento do controlador.

1.2.3 Identificação Experimental de Parâmetros

Durante o desenvolvimento, alguns parâmetros, como coeficientes de atrito e amortecimento, foram determinados experimentalmente. Esses parâmetros foram ajustados com base em observações do comportamento do sistema e foram incorporados ao modelo matemático para melhorar a precisão das simulações e a eficácia do controle.

A metodologia adotada neste projeto permitiu o desenvolvimento de um sistema de controle eficiente e estável para o pêndulo invertido. A construção cuidadosa da planta física, a modelagem precisa e a implementação do controlador LQR garantiram a estabilidade do sistema.

1.2.4 Simulação e Projeto das Leis de Controle

Para viabilizar o uso de um controlador linear, o modelo matemático foi linearizado ao redor do ponto de equilíbrio, onde o pêndulo está na posição vertical. A linearização foi realizada aplicando uma aproximação em Série de Taylor de primeira ordem, simplificando as equações de movimento para pequenas variações ao redor do ponto de equilíbrio.

Com o modelo linearizado, foi projetado um controlador Regulador Quadrático Linear (LQR) para estabilizar o pêndulo. O LQR foi escolhido por sua capacidade de

otimizar o esforço de controle enquanto minimiza o desvio do pêndulo em relação à posição vertical. As matrizes de ponderação \mathbf{Q} e R , que equilibram o controle entre a precisão da posição e o esforço do motor, foram ajustadas com base em simulações para obter uma resposta estável.

1.3 Implementação do Sistema de Controle e Validação Experimental

Antes da implementação prática, o modelo do sistema e o controlador LQR foram simulados no MATLAB/Simulink para verificar a estabilidade e a resposta dinâmica, ajustando os parâmetros do controlador. Em seguida, o controlador foi implementado digitalmente no ESP32, e o sistema foi testado na planta física, com ajustes finais para assegurar estabilidade e precisão no mundo real.

1.4 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta a fundamentação teórica necessária para o desenvolvimento do controle do sistema de pêndulo invertido. O capítulo abrange conceitos essenciais como a modelagem do sistema dinâmico, a representação em espaço de estados, os métodos de linearização e o controlador LQR, estabelecendo a base para as etapas práticas do projeto.
- O Capítulo 3 detalha o processo de construção da planta física do pêndulo invertido, incluindo a seleção dos componentes mecânicos e eletrônicos, o diagrama de conexões e a montagem experimental.
- O Capítulo 4 descreve o desenvolvimento da modelagem matemática do sistema e a identificação dos parâmetros, como coeficientes de atrito e amortecimento. Utilizando o formalismo de Lagrange, são obtidas as equações de movimento e a representação em espaço de estados.
- O Capítulo 5 aborda a linearização do sistema, o ajuste das matrizes de ponderação do controlador LQR e a apresentação dos resultados experimentais obtidos.
- O Capítulo 6 finaliza o trabalho com uma avaliação dos resultados obtidos, destacando as contribuições do projeto e sugerindo possíveis trabalhos futuros.

2 Referencial Teórico

O sistema de pêndulo invertido é um dos exemplos mais utilizados em teoria de controle, devido à sua dinâmica inerentemente instável e sensível a perturbações. Ele consiste em um pêndulo que precisa ser mantido em posição vertical sobre uma base móvel, o que exige um sistema de controle para compensar continuamente as oscilações. Este sistema representa um problema de controle instável, onde a dificuldade de estabilização aumenta com o grau de liberdade e com as variações de parâmetros não-lineares do sistema, como ressaltado por [Kumar, Singh e Das \(2013\)](#).

Este capítulo apresenta a fundamentação teórica necessária para o desenvolvimento do controle do sistema de pêndulo invertido. Serão abordados conceitos essenciais, como a modelagem de sistemas dinâmicos com base nas Equações de Euler-Lagrange, a linearização do modelo e a representação em espaço de estados. Além disso, uma seção sobre identificação experimental de parâmetros é incluída, abordando métodos como mínimos quadrados e regressão linear que serão utilizados posteriormente na identificação da planta.

2.1 Modelagem e Representação do Sistema Dinâmico

Este trabalho adotará o formalismo de Lagrange para descrever o comportamento dinâmico do pêndulo invertido. As Equações de Lagrange oferecem uma maneira eficiente de descrever a dinâmica de sistemas mecânicos, especialmente aqueles sujeitos a restrições ou vínculos. Em comparação com a formulação newtoniana, o método lagrangiano permite trabalhar diretamente com coordenadas generalizadas, o que reduz o número de variáveis e elimina a necessidade de incluir as forças de vínculo explicitamente nas equações de movimento ([LEMOS, 2013](#)).

Para um sistema de n graus de liberdade, as coordenadas generalizadas q_k representam as variáveis independentes que descrevem a configuração do sistema. A função de Lagrange, ou lagrangiana L , é definida como a diferença entre a energia cinética T e a energia potencial V :

$$\mathcal{L} = T - V. \quad (2.1)$$

As Equações de Lagrange para uma coordenada q_k são derivadas usando o princípio variacional de Hamilton, que afirma que a trajetória seguida por um sistema mecânico minimiza (ou torna estacionária) a ação S , definida pela integral da lagrangiana ao longo do tempo:

$$S = \int_{t_1}^{t_2} L dt. \quad (2.2)$$

Impondo a condição de que a variação da ação seja zero ($\delta S = 0$), obtemos as Equações de Euler-Lagrange, dadas por:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = 0, \quad k = 1, \dots, n. \quad (2.3)$$

Essas equações constituem um sistema de n equações diferenciais de segunda ordem que determinam o comportamento dinâmico do sistema em função das condições iniciais (LEMOS, 2013).

Para o propósito de projeto de um sistema de controle, será adotada a representação no espaço de estados, em sua forma não-linear e linearizada. A representação no espaço de estados é uma metodologia eficaz para modelar e controlar sistemas dinâmicos, especialmente sistemas multivariáveis, pois permite descrever o comportamento do sistema ao longo do tempo por meio de variáveis de estado. Rušar et al. (2017) destaca que a modelagem em espaço de estados, aliada à linearização, simplifica o tratamento de sistemas não lineares, como o pêndulo invertido.

Em geral, para um sistema não linear, a formulação no espaço de estados é dada por:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.4)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \quad (2.5)$$

onde: $\mathbf{x} \triangleq [x_1 \ x_2 \ \dots \ x_n]^T \in \mathbb{R}^n$ representa o vetor de estados; \mathbf{u} é o vetor de entradas exógenas, $\mathbf{y} \triangleq [y_1 \ y_2 \ \dots \ y_m]^T \in \mathbb{R}^m$ é o vetor de medidas; $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$; e $\mathbf{h} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$.

Para um sistema linear ou linearizado, as equações de estado são expressas como (OGATA, 2010):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (2.6)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (2.7)$$

onde:

- \mathbf{A} é a matriz de estado, que determina a dinâmica interna do sistema,
- \mathbf{B} é a matriz de controle, que define o impacto das entradas nas variáveis de estado,
- \mathbf{C} é a matriz de saída, associando o estado à saída do sistema,
- \mathbf{D} é a matriz de feedthrough, que relaciona diretamente a entrada e a saída.

As matrizes \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} são obtidas por meio da linearização das Equações (2.4) e (2.5). A linearização é uma técnica essencial no controle de sistemas não lineares, permitindo aproximar o comportamento de um sistema ao redor de um ponto de equilíbrio.

Para linearizar uma função não linear, como as equações de movimento do pêndulo invertido, utilizamos a **expansão em Série de Taylor**. Expandimos a função $f(\mathbf{x}, \mathbf{u})$, da Equação (2.4), em torno do ponto de equilíbrio e ignoramos os termos de ordem superior (Higher-Order Terms, H.O.T.):

$$f(\mathbf{x}, \mathbf{u}) \approx f(\mathbf{x}_0, \mathbf{u}_0) + \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{u} - \mathbf{u}_0)$$

Como estamos no ponto de equilíbrio, $f(\mathbf{x}_0, \mathbf{u}_0) = 0$, e partindo da igualdade mostrada na Equação 2.4, então temos:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

onde as matrizes \mathbf{A} e \mathbf{B} são obtidas da Jacobiana de f em relação a \mathbf{x} e \mathbf{u} , respectivamente:

$$\mathbf{A} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)}, \quad \mathbf{B} = \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \quad (2.8)$$

Uma vez obtido o modelo linearizado do sistema em torno do ponto de equilíbrio $(\mathbf{x}_0, \mathbf{u}_0)$, o método indireto de [Lyapunov \(1892\)](#) permite inferir a estabilidade do sistema original a partir da análise dos autovalores da matriz associada ao modelo linearizado, estabelecendo condições para a estabilidade assintótica em torno de um ponto de equilíbrio.

As principais condições para a aplicação do método indireto de Lyapunov são:

1. Se o modelo linearizado é assintoticamente estável, ou seja, todos os autovalores da matriz \mathbf{A} estão no semiplano esquerdo do plano complexo, então o sistema original também será assintoticamente estável nas proximidades do ponto de equilíbrio $(\mathbf{x}_0, \mathbf{u}_0)$.
2. Se o modelo linearizado é instável, ou seja, ao menos um dos autovalores de A está no semiplano direito do plano complexo, o sistema original será instável nas proximidades de $(\mathbf{x}_0, \mathbf{u}_0)$.

2.1.1 Teste de Controlabilidade

Para garantir que o sistema de pêndulo invertido é controlável, realizamos o teste de controlabilidade. Este teste verifica se o sistema pode ser controlado a partir de qualquer estado inicial até qualquer estado desejado, aplicando uma entrada de controle apropriada. A análise da controlabilidade é um pré-requisito fundamental para a implementação de controladores de estado, como o controlador LQR. Sua verificação é feita construindo a matriz de controlabilidade \mathbf{MC} ([DORF; BISHOP, 2010](#)), definida como:

$$\mathbf{MC} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (2.9)$$

onde \mathbf{A} e \mathbf{B} são as matrizes de estado e entrada, respectivamente, e n representa a ordem do sistema. O sistema é considerado controlável se o posto da matriz \mathbf{MC} for igual a n , indicando que todas as dinâmicas do sistema podem ser influenciadas pela entrada de controle.

Para o cálculo, utilizamos a função `ctrb` do MATLAB, que gera automaticamente a matriz de controlabilidade a partir das matrizes \mathbf{A} e \mathbf{B} previamente definidas. Em seguida, o posto da matriz \mathbf{MC} foi determinado, obtendo-se o valor 4, que é igual à ordem do sistema. Logo, concluímos que o sistema é controlável.

2.1.2 Discretização

Em sistemas de controle digitais, os sinais e sistemas são frequentemente discretizados para serem implementados em processadores digitais. A discretização é o processo de converter um sistema contínuo em um sistema discreto, onde as variáveis de estado e o controle são avaliados em instantes de tempo específicos, separados por intervalos regulares, definidos pelo período de amostragem T_s . No [MATLAB \(2022\)](#), a função `c2d` é amplamente utilizada para realizar essa transformação em sistemas no espaço de estados.

Dado um sistema contínuo descrito no espaço de estados pelas matrizes \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} , a função `c2d` converte este sistema para um sistema discreto com um período de amostragem T_s . A sintaxe básica para essa conversão é:

```
sysd = c2d(sys, Ts, 'method');
```

em que, `sys` representa o sistema contínuo (definido no MATLAB como um objeto de espaço de estados), `Ts` é o período de amostragem, e `method` define o método de discretização. Os métodos mais comuns incluem: `'zoh'` (Zero-Order Hold), que assume que o sinal de controle é constante durante cada intervalo de amostragem; `'foh'` (First-Order Hold), que assume que o sinal de controle varia linearmente durante cada intervalo de amostragem e `'tustin'`, que aplica a transformação bilinear para a discretização.

Após a discretização, o sistema discreto resultante será descrito pelas matrizes \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d e \mathbf{D}_d , correspondentes às matrizes de estado e de entrada do sistema contínuo. Possibilitando a implementação do sistema em controladores digitais.

2.1.3 Estabilidade no Espaço de Estados

A estabilidade de um sistema dinâmico é um aspecto crítico na análise de sistemas de controle. Para sistemas representados em espaço de estados, a estabilidade é determinada pelo comportamento dos autovalores da matriz de estado \mathbf{A} . Especificamente, para um sistema linear e invariável no tempo (LTI) ([DORF; BISHOP, 2010](#)):

- O sistema é **estável** se todos os autovalores da matriz \mathbf{A} possuem parte real negativa.
- O sistema é **instável** se pelo menos um autovalor da matriz \mathbf{A} possui parte real positiva.
- O sistema é **marginalmente estável** se todos os autovalores possuem parte real não positiva, e quaisquer autovalores com parte real zero são simples (ou seja, não possuem multiplicidade superior a um).

A estabilidade é analisada resolvendo a equação característica da matriz \mathbf{A} , expressa como:

$$\det(s\mathbf{I} - \mathbf{A}) = 0, \quad (2.10)$$

sendo $s \in \mathbb{C}$ uma variável. Os autovalores da matriz de estado \mathbf{A} define os modos naturais do sistema e sua resposta ao longo do tempo. Quando todos os autovalores de \mathbf{A} têm partes reais negativas, o sistema tende a retornar ao estado de equilíbrio após qualquer perturbação, indicando uma resposta estável (DORF; BISHOP, 2010).

2.2 Identificação Paramétrica Experimental do Modelo do Sistema

A identificação experimental essencial para se determinar parâmetros difíceis de serem calculados teoricamente ou que podem variar devido ao desgaste ou alterações ambientais. Esse processo pode envolver a aplicação de estímulos controlados ao sistema e a medição das respostas, a fim de inferir os parâmetros do modelo (BROCK, 2002; AGUIRRE, 2015; LJUNG, 1998).

Uma das técnicas mais usadas para determinar a relação entre variáveis dependentes e independentes é a técnica estatística de regressão linear. No contexto do controle do pêndulo invertido, a regressão linear pode ser utilizada para estimar parâmetros como atrito ou amortecimento, com base em medições da resposta do sistema. Esses parâmetros ajustados são então empregados no modelo do sistema para melhorar a precisão do controlador, permitindo uma resposta mais fiel à dinâmica real do pêndulo invertido.

Considerando um conjunto de dados (x_i, y_i) onde x representa a variável independente e y a variável dependente, a regressão linear busca uma função na forma (AGUIRRE, 2015; LJUNG, 1998):

$$y = \beta_0 + \beta_1 x + \epsilon, \quad (2.11)$$

onde β_0 é o intercepto, β_1 é o coeficiente angular (inclinação) da reta, e ϵ representa o termo de erro associado à discrepância entre a estimativa e os dados reais.

Para determinar os coeficientes β_0 e β_1 , utiliza-se o método dos mínimos quadrados (AGUIRRE, 2015; LJUNG, 1998), que minimiza a soma dos quadrados dos erros ϵ . Esse método envolve resolver o seguinte sistema de equações normais:

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (2.12)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}, \quad (2.13)$$

onde \bar{x} e \bar{y} representam as médias dos conjuntos de dados de x e y , respectivamente.

2.3 Controlador LQR

O controlador LQR é um método de controle que visa otimizar o desempenho de sistemas dinâmicos, minimizando um índice de desempenho quadrático. Este método se aplica a sistemas representados no espaço de estados e utiliza realimentação de estado para obter uma resposta ótima do sistema.

Consideramos um sistema linear e invariante no tempo, definido pela Equação (2.6), em que o objetivo é minimizar o seguinte índice de desempenho quadrático:

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (2.14)$$

onde \mathbf{Q} e \mathbf{R} são matrizes de pesos que determinam a importância relativa dos estados e dos esforços de controle. A matriz \mathbf{Q} é geralmente uma matriz simétrica definida positiva, enquanto \mathbf{R} é uma matriz hermitiana definida positiva. O termo $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ representa o custo associado aos estados, e $\mathbf{u}^T \mathbf{R} \mathbf{u}$ representa o custo associado ao consumo de energia dos sinais de controle (OGATA, 2010; DORF; BISHOP, 2010).

A lei de controle ótimo para o controlador LQR é dada por (OGATA, 2010; DORF; BISHOP, 2010):

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t), \quad (2.15)$$

onde \mathbf{K} é a matriz de ganho de controle, determinada de modo a minimizar o índice de desempenho J . A matriz \mathbf{K} pode ser obtida resolvendo a equação de Riccati associada ao problema, a qual envolve as matrizes \mathbf{A} , \mathbf{B} , \mathbf{Q} e \mathbf{R} do sistema:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0, \quad (2.16)$$

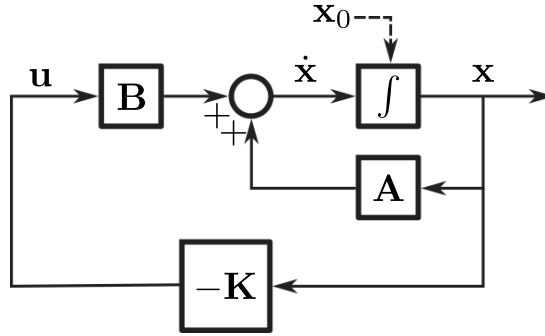
onde \mathbf{P} é uma matriz simétrica positiva definida que satisfaz essa equação de Riccati. Uma vez determinada a matriz \mathbf{P} , o ganho ótimo \mathbf{K} é calculado como:

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}. \quad (2.17)$$

A aplicação da lei de controle descrita na Equação (2.15) garante que o sistema fechado descrito na Equação (2.6) seja estável, desde que os autovalores de $\mathbf{A} - \mathbf{BK}$ tenham partes reais negativas, o que resulta na convergência do estado \mathbf{x} para zero à medida que o tempo tende ao infinito (OGATA, 2010; DORF; BISHOP, 2010).

O diagrama da lei de controle aplicada ao sistema completo e fechado pode ser visualizado na Figura 1.

Figura 1 – Sistema de controle ótimo.



Fonte: O Autor (2024).

Para o projeto de um controlador LQR discreto, utilizamos a função `dlqr` do MATLAB (2022). Esse controlador é a versão discreta do controlador LQR contínuo, mas aplicado a um sistema discreto. A função `dlqr` calcula o ganho de feedback de estado \mathbf{K}_d para o sistema.

Dado um sistema discreto nas matrizes \mathbf{A}_d e \mathbf{B}_d , o índice de desempenho a ser minimizado é dado por:

$$J = \sum_{k=0}^{\infty} (\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k), \quad (2.18)$$

onde \mathbf{Q} e \mathbf{R} são matrizes de pesos que penalizam, respectivamente, os estados e o esforço de controle, semelhantes ao LQR contínuo.

A sintaxe para calcular o ganho \mathbf{K}_d no MATLAB é:

$$[\mathbf{K}_d, \mathbf{S}, \mathbf{e}] = \text{dlqr}(\mathbf{A}_d, \mathbf{B}_d, \mathbf{Q}, \mathbf{R});$$

sendo que as matrizes \mathbf{A}_d e \mathbf{B}_d são as matrizes discretizadas do sistema (obtidas com `c2d`), \mathbf{Q} e \mathbf{R} são as matrizes de pesos, \mathbf{K}_d é a matriz de ganho ótimo do controlador LQR discreto, \mathbf{S} é a solução da equação de Riccati discreta, e \mathbf{e} são os autovalores do sistema de malha fechada.

Após a obtenção do ganho \mathbf{K}_d , a lei de controle ótima para o sistema discreto é dada por:

$$\mathbf{u}_k = -\mathbf{K}_d \mathbf{x}_k, \quad (2.19)$$

onde \mathbf{x}_k e \mathbf{u}_k são, respectivamente, o vetor de estado e o sinal de controle no instante de amostragem k .

O controlador LQR proporciona uma solução eficiente para sistemas lineares ou linearizados, permitindo minimizar o esforço de controle enquanto estabiliza o sistema. [Prasad, Tyagi e Gupta \(2014\)](#) comparam o desempenho do LQR com o PID em sistemas de pêndulo invertido e observam que o LQR é mais eficaz para estabilizar o sistema, principalmente em cenários com perturbações externas e variabilidade dos parâmetros. No entanto, o LQR exige uma modelagem precisa e linearização do sistema, o que pode limitar sua aplicação em sistemas fortemente não lineares ou sujeitos a grandes variações.

3 Construção do Dispositivo

Neste capítulo é apresentada a montagem da planta do pêndulo invertido, abordando as características elétricas, mecânicas e computacionais que garantem o funcionamento do sistema.

3.1 Construção Mecânica

Cada componente da planta foi cuidadosamente escolhido para garantir a integração eficiente com o restante do sistema. Nesta seção, serão descritos os principais componentes utilizados, assim como os critérios de seleção e a justificativa para suas escolhas.

Motor

O motor foi uma escolha fundamental no sistema, pois havia a necessidade de uma solução capaz de fornecer torque suficiente para movimentar o carro com precisão e suavidade. Três motores foram testados inicialmente:

- **Motor Akiyama AK360 12V, 7000 RPM, 5W:** Embora tenha uma alta rotação, o torque de 0,0077 Nm mostrou-se insuficiente para movimentar o carro, especialmente ao tentar vencer o atrito da guia linear. Este motor foi descartado devido à sua baixa capacidade de força.
- **Motor 25GA370, 12V, 1360 RPM, 10W:** Apesar de conseguir movimentar o carro, apresentou dificuldades em baixas velocidades devido ao atrito estático, o que resultou em movimentação lenta e irregular.
- **Motor DC 775, 12V, 12000 RPM, 100W:** Este motor foi escolhido como a melhor solução para o sistema, fornecendo torque e potência suficientes para movimentar o carro com facilidade. No entanto, devido à sua alta velocidade, foi necessário limitar a tensão aplicada a 3V, garantindo que o movimento fosse controlado e seguro. A Figura 2 ilustra fisicamente o motor selecionado.

Figura 2 – Motor DC 775 utilizado no sistema



Fonte: Mercado Livre (2024). Disponível em
<<https://produto.mercadolivre.com.br/MLB-4578031298>>.

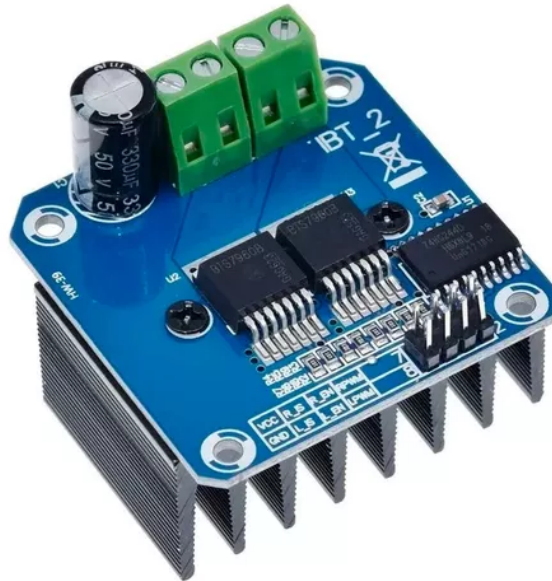
Tabela 1 – Comparação dos motores testados

| Motor | Características |
|---------------|--|
| Akiyama AK360 | Tensão: 12V |
| | Potência: 5W |
| | Torque: 0.0077 Nm |
| | Resultado: Não movimentou o carro |
| 25GA370 | Tensão: 12V |
| | Potência: 10W |
| | Torque: 0.039 Nm |
| | Resultado: Movimentou com dificuldades |
| DC 775 | Tensão: 12V |
| | Potência: 100W |
| | Torque: 0.79 Nm |
| | Resultado: Funcionou perfeitamente |

Driver do Motor

Para o controle do motor DC 775 foi utilizado o módulo de driver BTS7960, que atende à necessidade de alta potência e suporta correntes elevadas, necessárias para o funcionamento do sistema. Esse módulo é amplamente utilizado em aplicações de robótica e mecânica, sendo uma escolha popular entre estudantes e hobbystas pelo seu excelente custo-benefício. A Figura 3 mostra o driver BTS7960 fisicamente.

Figura 3 – Driver BTS7960



Fonte: Mercado Livre (2024).

O módulo BTS7960 é baseado em dois drivers BTS7960, cada um representando metade de uma ponte H. Em conjunto, eles formam uma ponte H completa, com capacidade de corrente contínua de até 43 A, o que o torna ideal para o controle de motores que demandam alta potência. Suas principais especificações são:

- Controle de direção e velocidade via PWM;
- Tensão de alimentação: 5 - 45 V;
- Tensão lógica: 3,3 - 5,3 V;
- Corrente de alimentação do módulo: 3 mA;
- Corrente contínua máxima: 43 A.

A escolha desse driver BTS7960 se deveu à sua capacidade de fornecer uma corrente suficientemente elevada para o motor DC 775, que é responsável pelo movimento do carrinho na planta do pêndulo invertido.

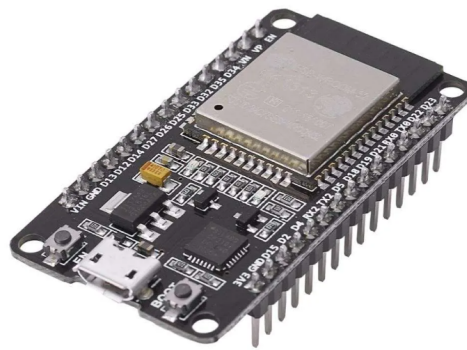
Microcontrolador

O microcontrolador ESP32 foi escolhido para este projeto devido à sua alta capacidade de processamento, suporte para multitarefa e facilidade de programação. Ele é equipado com um processador dual-core que opera em uma faixa de clock entre 80 MHz e 240 MHz, tornando-o apto para executar múltiplas tarefas, como o controle do

motor, a leitura de sensores e a comunicação com outros dispositivos. Sua arquitetura também permite a execução eficiente de processos simultâneos (multitasking).

Além da alta velocidade de processamento, o ESP32 oferece flexibilidade para o desenvolvimento de firmware. Ele é compatível com a IDE do Arduino, facilitando a programação mesmo para usuários iniciantes, e oferece uma ampla gama de bibliotecas que simplificam a implementação de protocolos de comunicação e o controle de periféricos. A Figura 4 mostra o microcontrolador ESP32.

Figura 4 – Microcontrolador ESP32



Fonte: Mercado Livre (2024).

Principais características do ESP32:

- Processador dual-core, com clock ajustável entre 80 MHz e 240 MHz;
- Capacidade para multitarefa, ideal para controle de sistemas dinâmicos complexos;
- Programação facilitada via IDE do Arduino;
- Ampla conectividade com periféricos e suporte a protocolos de comunicação como I2C, SPI, UART e Wi-Fi.

Encoders

Para a medição da posição do carro e do pêndulo, foram utilizados encoders rotativos incrementais com resolução de 2000 passos por revolução, resultando em uma precisão angular de 0,09 graus por passo. A escolha desse tipo de encoder foi baseada em sua alta resolução e facilidade de integração ao sistema, pois foram utilizados tanto como eixos da polia e do pêndulo quanto para a medição das variáveis, simplificando o sistema mecânico. A Figura 5 mostra a imagem do encoder rotativo incremental selecionado.

Figura 5 – Encoder rotativo incremental utilizado no sistema



Fonte: AliExpress (2024).

Guia Linear com Carrinho

O carro da planta se move ao longo de uma guia linear com carrinho, um trilho deslizante comumente utilizado em sistemas de CNC e impressoras 3D. O fato de a guia já vir com um carrinho integrado simplificou a montagem, eliminando a necessidade de acoplar suportes adicionais. A Figura 6 apresenta a imagem da guia linear utilizada.

Figura 6 – Guia linear com carrinho utilizada no sistema.



Fonte: AliExpress (2024).

Pêndulo

Para o pêndulo, foi escolhida uma haste linear metálica com 30 cm de comprimento e 6 mm de diâmetro. A escolha do metal como material se deu por sua resistência mecânica e por apresentar uma distribuição de massa uniforme ao longo de todo o seu comprimento, o que simplifica a modelagem matemática do sistema, pois a distribuição uniforme da massa permite que o momento de inércia da haste seja calculado de forma precisa, utilizando modelos matemáticos tradicionais para objetos alongados. A Figura 7 apresenta a imagem da haste utilizada.

Figura 7 – Haste para o pêndulo.



Fonte: Mercado Livre (2024).

Outros Componentes Mecânicos

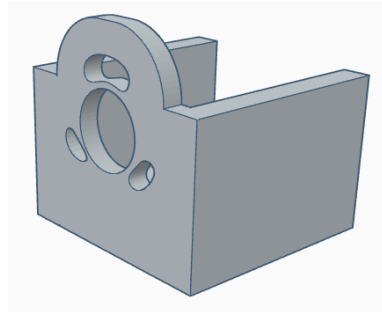
Adicionalmente, o sistema mecânico inclui um conjunto de polias e correia GT2, que foi escolhido por sua precisão no controle do movimento linear. A correia GT2, de 20mm, garantiu uma movimentação precisa e suave do carro, enquanto a polia foi acoplada diretamente aos encoders, facilitando a medição precisa da posição.

Peças Impressas em 3D

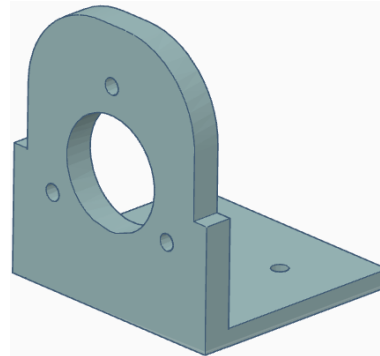
Peças customizadas foram impressas em 3D para suportar os componentes mecânicos e eletrônicos, como o acoplamento do encoder ao pêndulo e os suportes para os encoders e motores. A impressão 3D foi utilizada para permitir a fabricação de peças sob medida, que não estavam disponíveis comercialmente, ajustando o sistema às necessidades específicas da planta. As peças foram modeladas no software *Fusion 360*, da Autodesk, que possibilitou um design adaptado às exigências da planta. A Figura 8 a seguir mostra dois exemplos das peças impressas, utilizadas como suportes para os encoders:

Figura 8 – Peças impressas em 3D para o acoplamento dos encoders.

(a) Suporte do encoder para a posição.



(b) Suporte do encoder para o ângulo.



Fonte: O Autor (2024).

3.2 Diagrama de Conexões do Hardware

O diagrama de conexões do sistema é mostrado na Figura 9, onde detalhamos a ligação elétrica de alimentação e o envio de sinais entre os componentes principais.

No centro do esquema está o microcontrolador ESP32, que desempenha o papel de controlador do sistema, recebendo dados dos sensores, processando as informações e enviando comandos de controle ao motor.

Conexão do ESP32 com os Encoders

O ESP32 fornece alimentação de 5V para os encoders de posição e do pêndulo, permitindo a leitura dos dados de ambos os sensores. Os encoders estão conectados ao ESP32 por meio das linhas de sinal CLK e DT, responsáveis por enviar as informações de contagem de pulsos que representam a posição angular dos eixos. Essa conexão permite ao ESP32 monitorar a posição do carrinho e do pêndulo em tempo real.

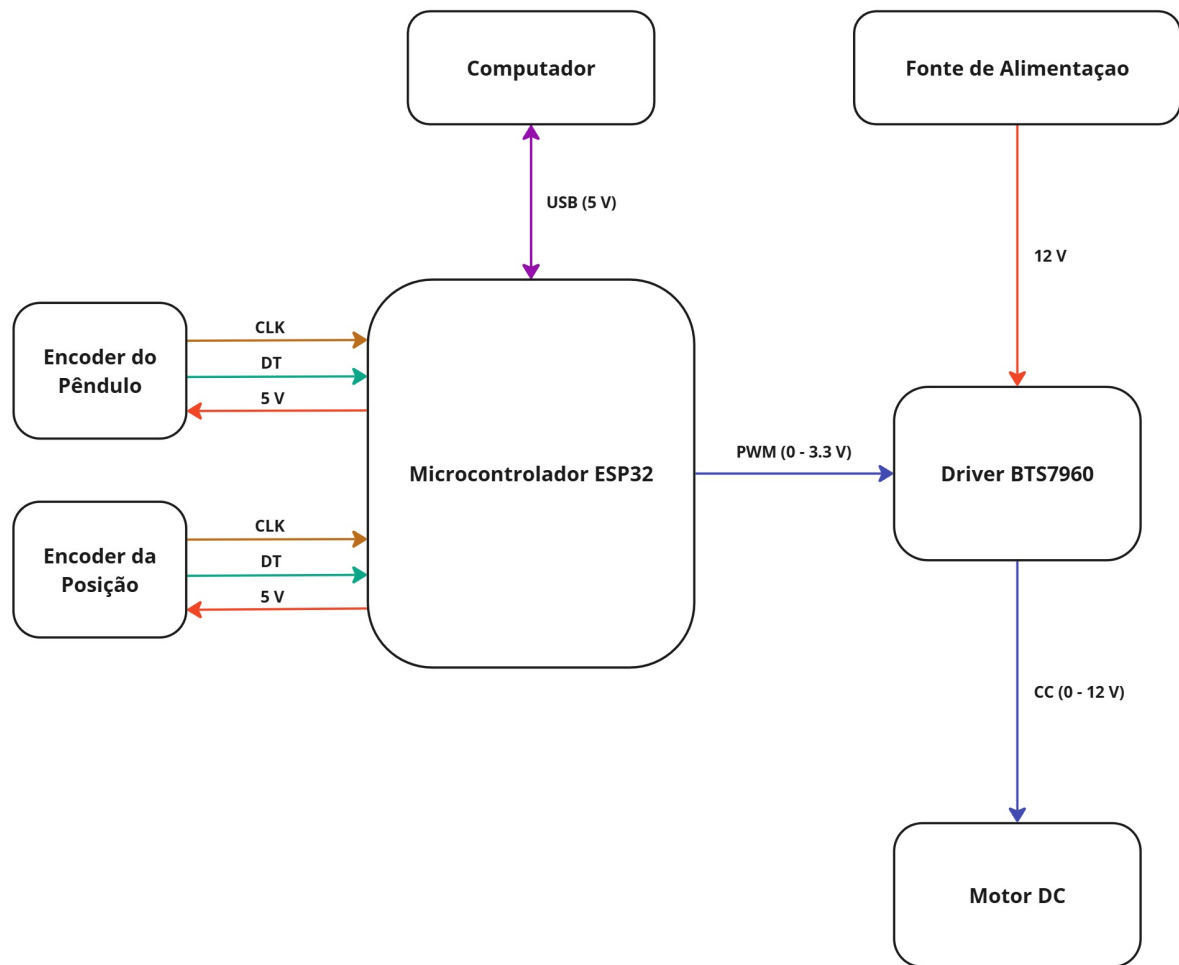
Conexão com o Computador

O ESP32 está conectado ao computador via USB. Esse link permite que o microcontrolador envie dados ao computador para monitoramento em tempo real do sistema e também possibilita o upload do firmware. Além disso, a conexão USB fornece uma alimentação de 5V ao ESP32, garantindo sua operação.

Controle do Driver do Motor

O ESP32 se conecta ao driver BTS7960 do motor DC, enviando um sinal PWM, que varia entre 0 a 3.3V, e atua como comando de controle para a tensão fornecida ao

Figura 9 – Diagrama de conexões do sistema de pêndulo invertido.



Fonte: O Autor (2024).

motor.

Alimentação do Motor e Driver

O driver BTS7960 recebe alimentação externa de uma fonte DC de 12V, capaz de fornecer até 100W de potência. Essa fonte permite ao driver controlar o motor DC 775, ajustando a tensão de saída de 0 a 12V para fornecer a potência adequada ao sistema, conforme necessário para mover o carrinho e manter o pêndulo em equilíbrio.

3.3 Algoritmo Implementado no ESP32

O algoritmo implementado digitalmente no ESP32 realiza a inicialização e leitura dos sensores, a atualização das variáveis de estado e o envio do sinal de controle ao driver do motor.

Inicialização e Leitura dos Sensores

Ao iniciar o programa, o ESP32 configura os encoders do sistema para capturar as posições do pêndulo e do carrinho. A leitura dos sensores é feita continuamente com interrupções no microcontrolador, assegurando que as variações de posição sejam capturadas em tempo real. A contagem dos pulsos é processada diretamente pela biblioteca `ESP32Encoder`, que traduz as mudanças angulares para os valores de posição do carrinho e ângulo do pêndulo.

Atualização das Variáveis e Controle Digital

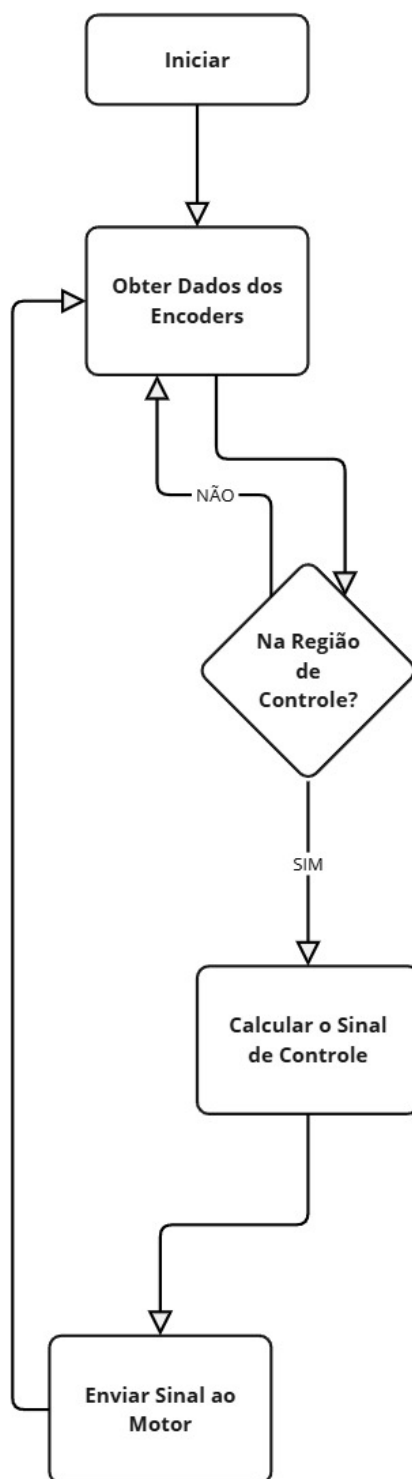
A implementação do controle ocorre em uma estrutura de amostragem digital discreta, com uma frequência de atualização de 50 Hz, correspondente a um período de amostragem de 20 ms. Essa taxa foi escolhida para ser suficientemente rápida para implementar o controle do sistema, garantindo uma resposta adequada sem comprometer a estabilidade do algoritmo e evitando a amplificação de ruídos indesejados no sinal de controle. A Figura 10 mostra o digrama da implementação digital.

Em cada ciclo de amostragem, o algoritmo verifica se o sistema está dentro da região de controle definida. Se estiver, as operações de controle são executadas, calculando a entrada de controle necessária para estabilizar o pêndulo. Esse sinal de controle é então convertido para um valor de PWM de 8 bits (variando de 0 a 255), que representa a amplitude da tensão que será aplicada ao motor por meio do driver.

Envio do Sinal de Controle

Após o cálculo do sinal de controle, o ESP32 envia o valor PWM ao driver do motor para manter o pêndulo em equilíbrio. O fluxo contínuo de leitura de sensores, cálculo do controle e envio do sinal ao driver permite a implementação digital completa do sistema de controle da planta de pêndulo invertido.

Figura 10 – Fluxograma da Lógica de Controle Implementada no ESP32

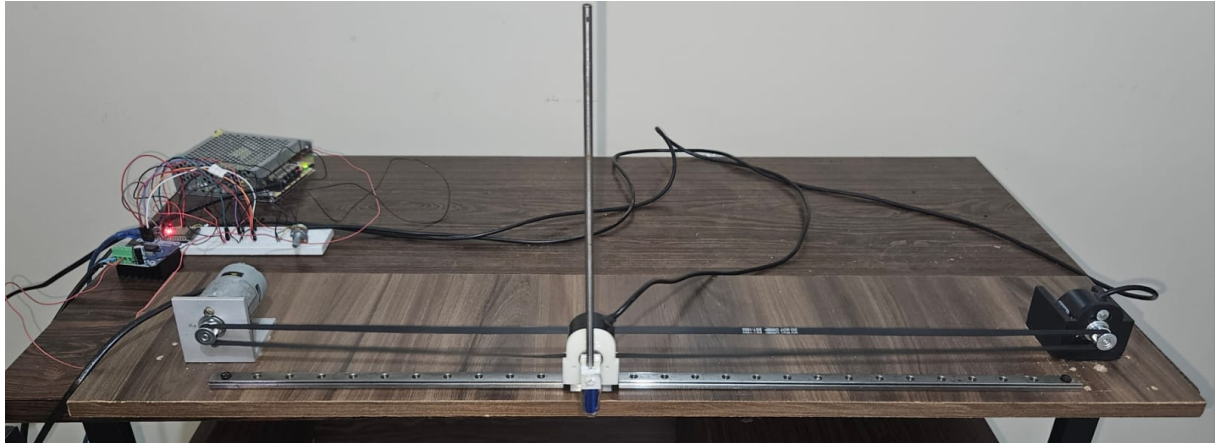


Fonte: O Autor (2024).

3.4 Montagem Experimental

A Figura 11 ilustra o sistema de pêndulo invertido montado, onde estão visíveis os principais componentes mecânicos e elétricos da planta.

Figura 11 – Visão geral do sistema completo montado.



Fonte: O Autor (2024).

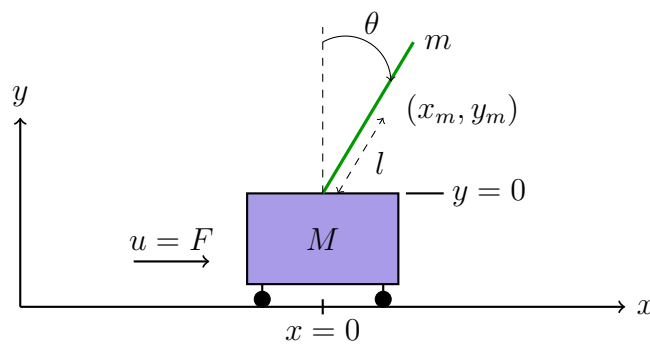
4 Modelagem e Identificação

Este capítulo apresenta a modelagem matemática do sistema de pêndulo invertido sobre um carrinho móvel, incluindo a definição de variáveis, análise das forças e torques atuantes e a formulação das equações de movimento. Para descrever a dinâmica do sistema, aplicamos o formalismo de Lagrange, que considera as energias cinética e potencial envolvidas. A representação final será expressa em termos das variáveis de estado, o que nos permitirá realizar uma análise em espaço de estados e desenvolver técnicas de controle nas etapas subsequentes.

4.1 Representação do Sistema

A representação física do sistema de pêndulo invertido é mostrada na Figura 12, onde um carrinho de massa M que pode se mover horizontalmente, sobre o qual é montado um pêndulo de massa m e comprimento l , onde o ponto de articulação se encontra na extremidade superior. O ângulo θ representa a inclinação do pêndulo em relação à vertical. Esse sistema é sujeito a forças externas, como a força F aplicada ao carrinho e os efeitos de atrito, representados pelos coeficientes b e c para o pêndulo e o carrinho, respectivamente.

Figura 12 – Pêndulo invertido.



Fonte: O Autor (2024).

A definição das variáveis é descrita por:

- $M[kg]$: massa do carrinho
- $m[kg]$: massa do pêndulo
- $b[kg \cdot m^2/s^{-1}]$: coeficiente de atrito do pêndulo
- $c[kg \cdot s^{-1}]$: coeficiente de atrito do carrinho

- $l[m]$: comprimento do pêndulo (distância do ponto de articulação ao centro de massa)
- $I[kg \cdot m^2]$: momento de inércia do pêndulo em relação ao centro de massa
- $F[N]$: força aplicada ao carrinho
- $x[m]$: posição horizontal do carrinho
- $\theta[rad]$: ângulo do pêndulo em relação à vertical
- $g[m \cdot s^{-2}]$: aceleração da gravidade

4.2 Equações de Movimento

Para determinar as equações de movimento do sistema, aplicamos a formulação de Lagrange, a qual é baseada na diferença entre a energia cinética total e a energia potencial total do sistema (Equação (2.1)). A fim de descrever as variáveis principais e suas relações, é necessário definir as propriedades e parâmetros envolvidos, conforme segue.

As coordenadas do centro de massa do pêndulo apresentado na Figura X, (x_m, y_m) , são expressas por:

$$x_m = x + l \sin \theta \quad (4.1)$$

$$y_m = l \cos \theta \quad (4.2)$$

4.2.1 Energia Potencial e Cinética

A energia potencial U do sistema é relacionada à altura do pêndulo em relação ao nível de referência. Ela pode ser expressa como:

$$U = mgy_m \quad (4.3)$$

Substituindo y_m pela Equação (4.2), a energia potencial torna-se:

$$U = mgl \cos \theta \quad (4.4)$$

A energia cinética total T do sistema é a soma das energias cinéticas translacional e rotacional. A energia cinética translacional T_T , dada pela soma das energias do carrinho e do pêndulo, pode ser expressa por:

$$T_T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}_m^2 + \dot{y}_m^2) \quad (4.5)$$

Substituindo as expressões para \dot{x}_m e \dot{y}_m das Equações (4.1) e (4.2), obtemos:

$$T_T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{x}\dot{\theta} \cos \theta \quad (4.6)$$

A energia cinética rotacional T_R está presente apenas no pêndulo, uma vez que o carrinho não realiza rotação. Esta energia é dada por:

$$T_R = \frac{1}{2} I \dot{\theta}^2 \quad (4.7)$$

Somando T_T e T_R para obter a energia cinética total, substituímos as Equações (4.6) e (4.7):

$$T = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m l^2 \dot{\theta}^2 + m l \dot{x} \dot{\theta} \cos \theta + \frac{1}{2} I \dot{\theta}^2 \quad (4.8)$$

$$= \frac{1}{2} (M + m) \dot{x}^2 + \frac{1}{2} (I + m l^2) \dot{\theta}^2 + m l \dot{x} \dot{\theta} \cos \theta \quad (4.9)$$

Assim, o Lagrangiano \mathcal{L} é dado por:

$$\mathcal{L} = \frac{1}{2} (M + m) \dot{x}^2 + \frac{1}{2} (I + m l^2) \dot{\theta}^2 + m l \dot{x} \dot{\theta} \cos \theta - m g l \cos \theta \quad (4.10)$$

4.2.2 Equações de Euler-Lagrange

A partir do Lagrangiano, aplicamos as equações de Euler-Lagrange (ver Equação (2.3)) para cada uma das variáveis generalizadas x e θ , conforme segue:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{x}} \right) - \frac{\partial \mathcal{L}}{\partial x} = F_x \quad (4.11)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = F_\theta \quad (4.12)$$

onde F_x representa as forças externas que atuam na direção x , enquanto F_θ representa as forças na direção angular θ . Neste sistema, as forças externas são dadas por:

$$F_x = F - c \dot{x} \quad (4.13)$$

$$F_\theta = -b \dot{\theta} \quad (4.14)$$

Substituindo as Equações (4.13) e (4.14) nas Equações (4.11) e (4.12), e resolvendo as derivadas, obtemos as seguintes equações diferenciais não lineares que descrevem a dinâmica do sistema:

$$(M + m) \ddot{x} + m l \ddot{\theta} \cos \theta - m l \dot{\theta}^2 \sin \theta = F - c \dot{x} \quad (4.15)$$

$$(I + m l^2) \ddot{\theta} + m l \ddot{x} \cos \theta - m g l \sin \theta = -b \dot{\theta} \quad (4.16)$$

Essas equações representam o modelo matemático completo do sistema de pêndulo invertido em sua forma não linear.

4.3 Representação em Espaço de Estados

Para a análise em controle e simulações, é conveniente reescrever essas equações na forma de espaço de estados, o que permite expressar o sistema em termos de variáveis de estado e suas respectivas derivadas de primeira ordem.

O primeiro passo para formular o modelo de espaço de estados é expressar as variáveis de interesse, isto é, as variáveis do sistema que caracterizam seu comportamento dinâmico. Para tanto, definimos as variáveis de estado como:

$$\begin{aligned} x_1 &= x & (\text{posição do carrinho}) \\ x_2 &= \theta & (\text{ângulo do pêndulo}) \\ x_3 &= \dot{x} & (\text{velocidade do carrinho}) \\ x_4 &= \dot{\theta} & (\text{velocidade angular do pêndulo}) \end{aligned}$$

Com essas definições, podemos reescrever o vetor de estado do sistema como $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$. A partir das Equações (4.15) e (4.16), buscamos isolar \ddot{x} e $\ddot{\theta}$ em termos das variáveis de estado e das forças atuantes no sistema.

Para começar, isolamos $\ddot{\theta}$ na Equação (4.15):

$$\ddot{\theta} = \frac{F - c\dot{x} + ml\dot{\theta}^2 \sin \theta - (M + m)\ddot{x}}{ml \cos \theta}$$

Em seguida, substituímos essa expressão na Equação (4.16) para eliminar $\ddot{\theta}$, obtendo uma nova expressão que depende apenas de \ddot{x} :

$$(I + ml^2) \frac{F - c\dot{x} + ml\dot{\theta}^2 \sin \theta - (M + m)\ddot{x}}{ml \cos \theta} + ml\ddot{x} \cos \theta - mgl \sin \theta = -b\dot{\theta}$$

Após simplificar essa expressão, obtemos a fórmula para \ddot{x} :

$$\ddot{x} = \frac{bml\dot{\theta} \cos \theta - m^2 l^2 g \sin \theta \cos \theta + (I + ml^2) (F - c\dot{x} + ml\dot{\theta}^2 \sin \theta)}{m^2 l^2 \sin^2 \theta + Mml^2 + (M + m) I} \quad (4.17)$$

Agora, repetindo o processo, isolamos \ddot{x} da Equação (4.16) e substituímos a expressão resultante na Equação (4.15):

$$\ddot{x} = \frac{-b\dot{\theta} + mgl \sin \theta - (I + ml^2) \ddot{\theta}}{ml \cos \theta}$$

Ao substituir essa expressão na Equação (4.15), obtemos:

$$(M + m) \frac{-b\dot{\theta} + mgl \sin \theta - (I + ml^2) \ddot{\theta}}{ml \cos \theta} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F - c\dot{x}$$

Simplificando, obtemos a expressão para $\ddot{\theta}$:

$$\ddot{\theta} = -\frac{Fml \cos \theta - c\dot{x}ml \cos \theta + m^2 l^2 \dot{\theta} \sin \theta \cos \theta + (M + m)(b\dot{\theta} - mgl \sin \theta)}{m^2 l^2 \sin^2 \theta + Mml^2 + (M + m)I} \quad (4.18)$$

Com as expressões obtidas para \ddot{x} (Equação (4.17)) e $\ddot{\theta}$ (Equação (4.18)), podemos formular o modelo em espaço de estados. Substituindo as variáveis de estado definidas anteriormente, temos:

$$\begin{aligned} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= \frac{bmlx_4 \cos x_2 - m^2 l^2 g \sin x_2 \cos x_2 + (I + ml^2)(F - cx_3 + mlx_4^2 \sin x_2)}{m^2 l^2 \sin^2 x_2 + Mml^2 + (M + m)I} \\ \dot{x}_4 &= -\frac{Fml \cos x_2 - cx_3 ml \cos x_2 + m^2 l^2 x_4 \sin x_2 \cos x_2 + (M + m)(bx_4 - mgl \sin x_2)}{m^2 l^2 \sin^2 x_2 + Mml^2 + (M + m)I} \end{aligned}$$

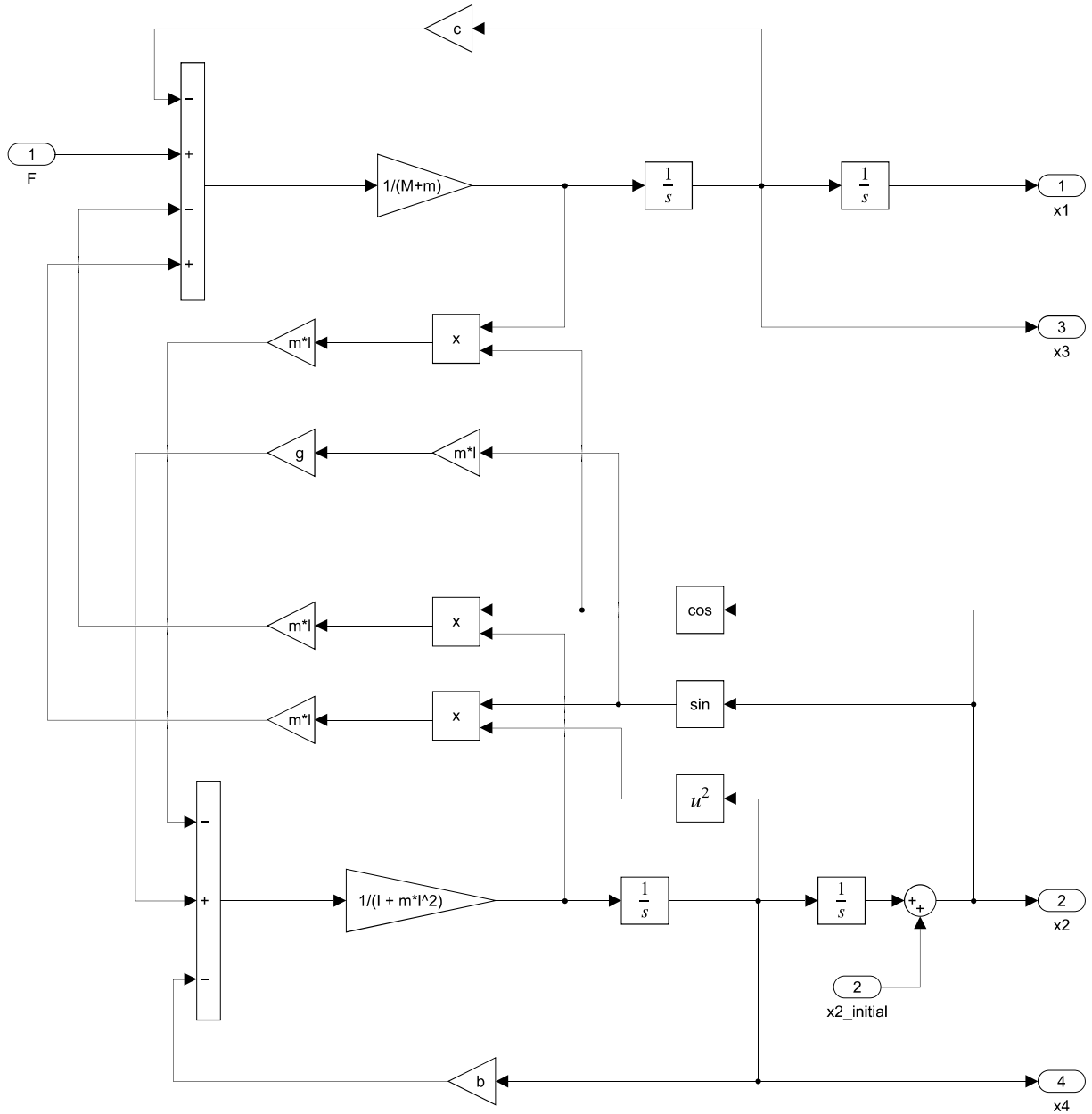
Assim, é possível definir a formulação não linear no espaço de estado, ver Equação (2.4), como:

$$\dot{\mathbf{x}} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{bmlx_4 \cos x_2 - m^2 l^2 g \sin x_2 \cos x_2 + (I + ml^2)(F - cx_3 + mlx_4^2 \sin x_2)}{m^2 l^2 \sin^2 x_2 + Mml^2 + (M + m)I} \\ -\frac{Fml \cos x_2 - cx_3 ml \cos x_2 + m^2 l^2 x_4 \sin x_2 \cos x_2 + (M + m)(bx_4 - mgl \sin x_2)}{m^2 l^2 \sin^2 x_2 + Mml^2 + (M + m)I} \end{bmatrix} \quad (4.19)$$

4.4 Implementação do Modelo Não Linear no Simulink

Para validar o comportamento do modelo não linear do pêndulo invertido sobre o carrinho, foi desenvolvida uma simulação no software Simulink, conforme ilustrado na Figura 13. No modelo, a entrada do sistema é representada pela variável u , que corresponde à força aplicada ao carrinho para controlar o movimento do pêndulo. Os estados de saída são x_1 , x_2 , x_3 , e x_4 , que representam, respectivamente, a posição do carro, o ângulo do pêndulo em relação à posição vertical, a velocidade do carro e a velocidade angular do pêndulo.

Figura 13 – Modelo não linear do pêndulo invertido implementado no Simulink.



Fonte: O Autor (2024).

4.5 Identificação dos Parâmetros do Sistema

Nesta seção, é apresentada a metodologia utilizada para identificar os parâmetros da planta que não puderam ser medidos diretamente, em especial o atrito do carrinho com a guia linear (parâmetro c) e o atrito do pêndulo no eixo (parâmetro b). Esses parâmetros são fundamentais para descrever com precisão a dinâmica do sistema, uma vez que influenciam diretamente na dissipação de energia e na resposta do sistema ao controle. Além disso, foi realizada a identificação da conversão de tensão para força, necessária para descrever o efeito da entrada de controle no movimento do carro.

4.5.1 Medidas Diretas

Para a identificação completa da planta, alguns parâmetros foram medidos diretamente, sendo eles demonstrados a seguir pela Tabela 2

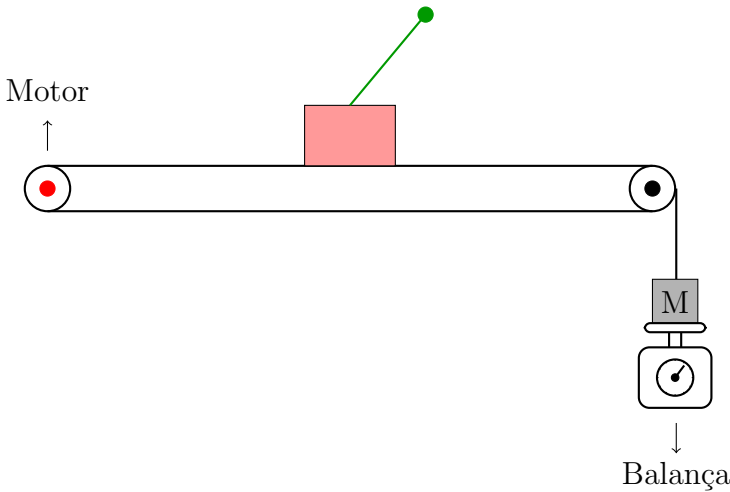
Tabela 2 – Parâmetros do Sistema

| Símbolo | Descrição | Valor |
|---------|---|---------------------------------------|
| M | Massa do carrinho | 0.2 kg |
| m | Massa do pêndulo | 0.075 kg |
| l | Distância ao centro de massa do pêndulo | 0.147 m |
| I | Momento de inércia do pêndulo | $5.402 \times 10^{-4} \text{ kg m}^2$ |
| g | Aceleração gravitacional | 9.81 m/s^2 |

4.5.2 Identificação da Conversão de Tensão para Força

Para determinar a relação entre a tensão aplicada ao motor e a força resultante exercida sobre o carrinho, foi realizada uma série de experimentos. Primeiramente, uma corda foi acoplada à polia da guia linear, com uma massa pendurada na vertical, tangenciando a polia do encoder de posição. Embaixo dessa massa, uma balança foi posicionada para medir o peso aparente da massa em diferentes tensões aplicadas ao motor. A cada tensão aplicada, a força exercida pelo motor fazia a massa ser parcialmente levantada, e a balança registrava o peso aparente reduzido. A partir dessas leituras, foi possível calcular a força efetiva exercida pelo motor. A Figura 14 ilustra o sistema experimental utilizado.

Figura 14 – Configuração experimental para medição da força exercida pelo motor.



Fonte: O Autor (2024).

Inicialmente, a massa pendurada foi de 1427 g. A cada tensão aplicada ao motor, registrou-se a massa levantada parcial. A força correspondente a cada tensão foi obtida multiplicando-se a massa levantada (convertida em kg) pela aceleração gravitacional $g = 9.81 \text{ m/s}^2$. Os dados coletados são apresentados na Tabela 3.

Tabela 3 – Dados de Identificação da Força Exercida pelo Motor

| Tensão (V) | Massa Levantada (kg) | Força (N) |
|------------|----------------------|-----------|
| 0.60 | 0.267 | 2.62 |
| 1.05 | 0.527 | 5.17 |
| 1.51 | 0.784 | 7.69 |
| 2.01 | 0.984 | 9.65 |
| 2.56 | 1.239 | 12.15 |

Para determinar a relação exata entre a tensão V e a força F , aplicamos o método de regressão linear, assumindo que a relação pode ser descrita pela equação da reta

$$F = k \cdot V \quad (4.20)$$

onde k representa a inclinação da reta. A Tabela 4 apresenta os valores intermediários para o cálculo do coeficiente de inclinação k .

Tabela 4 – Tabela de cálculo para a regressão linear entre tensão e força.

| x (Tensão em V) | y (Força em N) | xy | x^2 | y^2 | |
|-------------------|------------------|-------|-------|--------|--------|
| 0.60 | 2.62 | 1.57 | 0.36 | 6.86 | |
| 1.05 | 5.17 | 5.43 | 1.10 | 26.73 | |
| 1.51 | 7.69 | 11.61 | 2.28 | 59.15 | |
| 2.01 | 9.65 | 19.40 | 4.04 | 93.18 | |
| 2.56 | 12.15 | 31.12 | 6.55 | 147.73 | |
| Σ | 7.73 | 37.29 | 69.13 | 14.34 | 333.65 |

Conforme demonstrado na Equação (2.12), o coeficiente k é calculado pela fórmula:

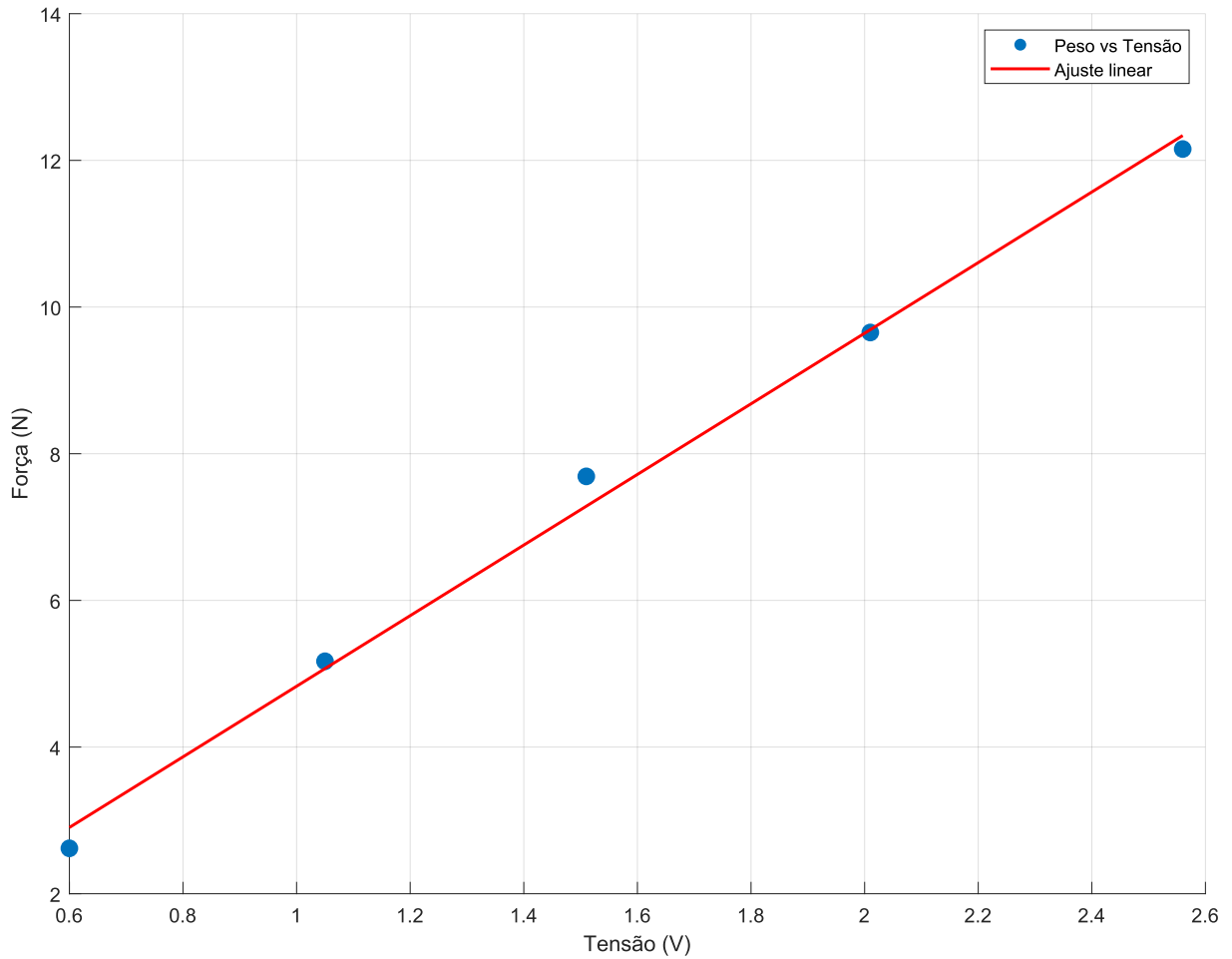
$$k = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} = \frac{5 \cdot 69.13 - (7.73)(37.29)}{5 \cdot 14.34 - (7.73)^2} = 4.81 \text{ N/V}$$

Dessa forma, a equação da relação entre tensão e força é dada por:

$$F = 4.81 \cdot V \quad (4.21)$$

A Figura 15 apresenta a reta de melhor ajuste sobreposta aos pontos experimentais.

Figura 15 – Reta de melhor ajuste traçada sobre os pontos experimentais.



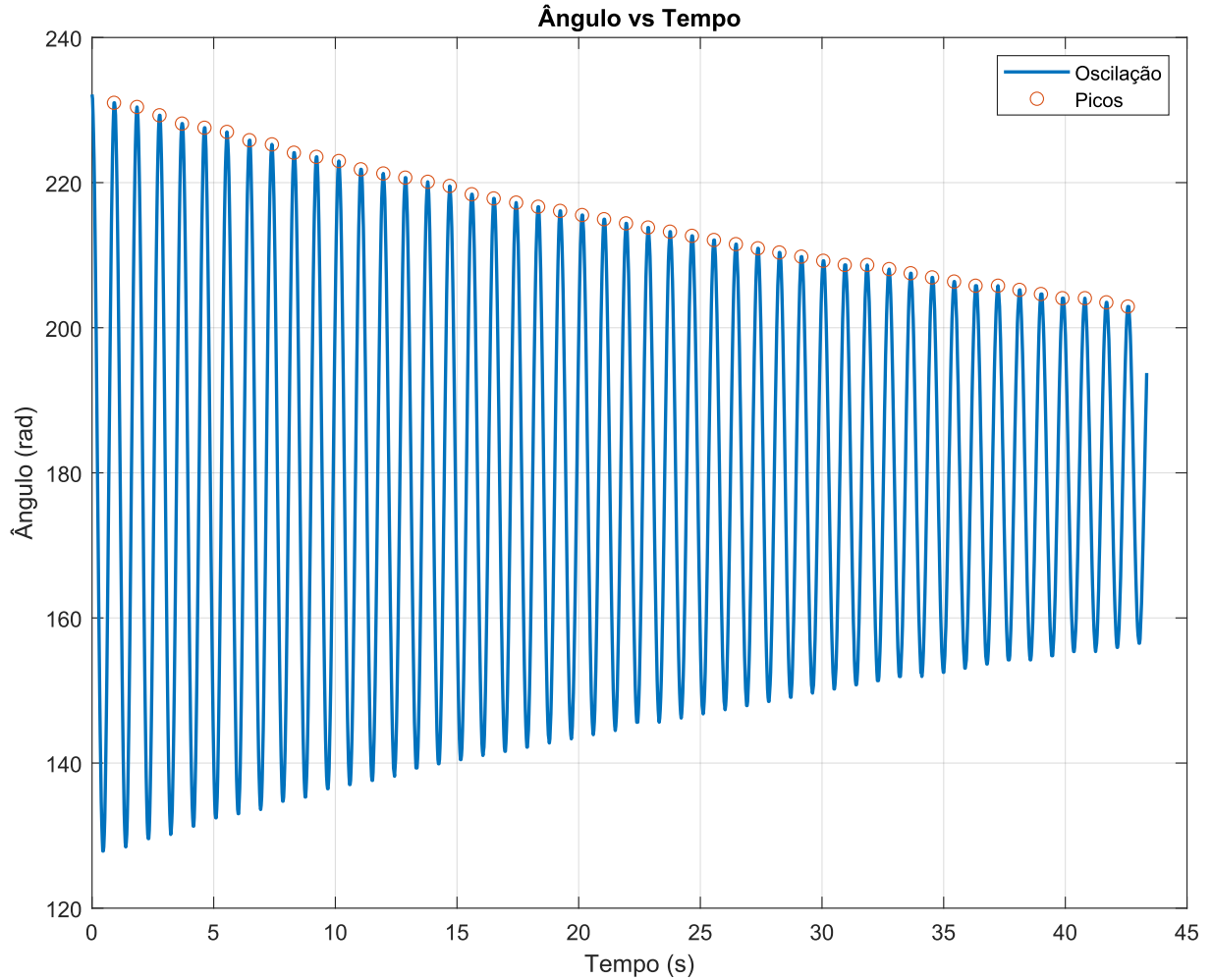
Fonte: O Autor (2024).

A relação obtida para a conversão entre a tensão aplicada ao motor e a força gerada permitirá que o sistema de controle utilize a tensão aplicada ao motor para calcular diretamente a força necessária para mover o carrinho.

4.5.3 Identificação do Coeficiente de Atrito no Eixo do Pêndulo (b)

Para determinar o coeficiente de atrito viscoso b no eixo do pêndulo, foi realizado um experimento de oscilação livre. Inicialmente, o pêndulo foi posicionado em um ângulo de $\theta = 180^\circ$ (4.05 radianos) e então liberado sem aplicação de torque adicional. A resposta do sistema foi registrada para observar o comportamento de oscilação e seu respectivo decaimento, como ilustrado na Figura 16.

Figura 16 – Oscilação e decaimento do pêndulo após deslocamento inicial.



Fonte: O Autor (2024).

Para modelar o comportamento de decaimento observado, aproximamos a resposta do sistema por uma função exponencial da forma:

$$y = Ae^{-bx} \quad (4.22)$$

onde A representa a amplitude inicial da oscilação e b é a taxa de decaimento, que relaciona-se diretamente com o coeficiente de atrito.

Tomando o logaritmo natural de ambos os lados, obtemos uma relação linear em função de x :

$$\ln(y) = \ln(A) - bx. \quad (4.23)$$

Assim, esta expressão pode ser reescrita no formato de uma função linear (ver Equação (2.11)):

$$y = m - bx \quad (4.24)$$

onde: $m = \ln(A)$ é o termo de amplitude e b representa a taxa de decaimento.

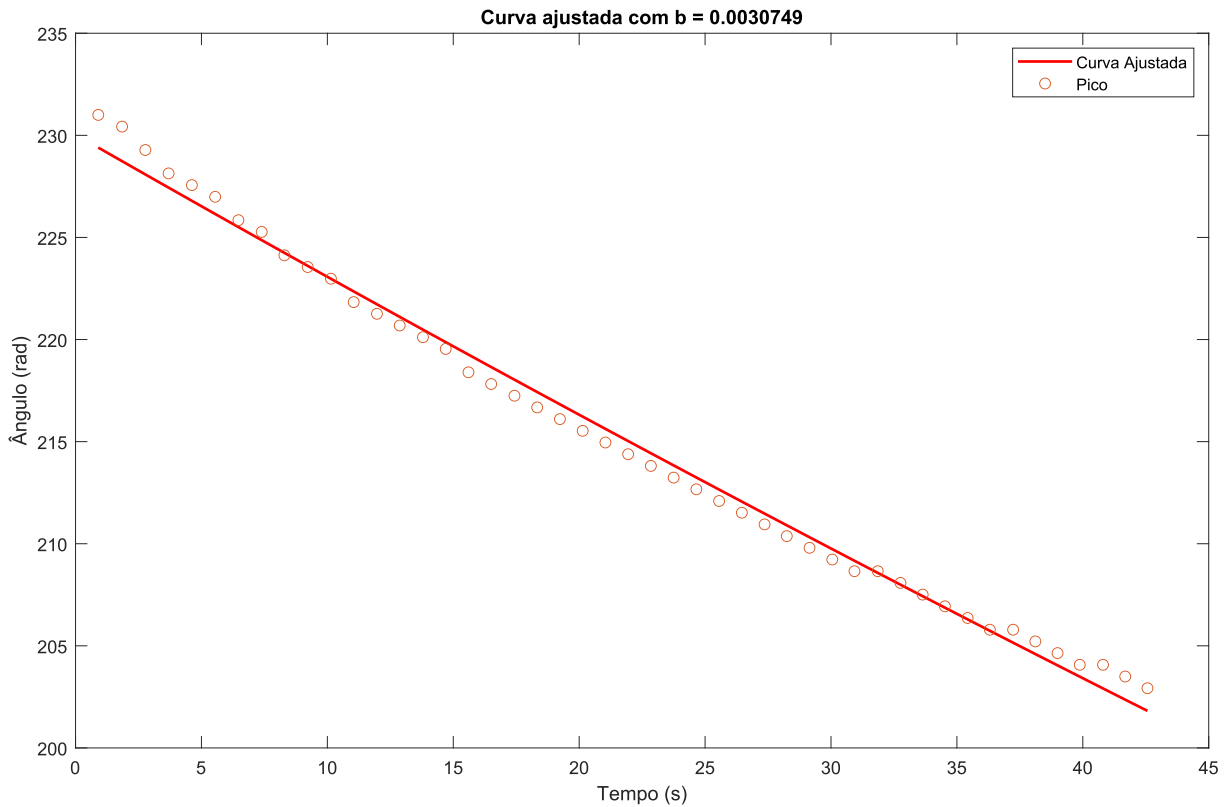
Utilizando o método dos mínimos quadrados para ajustar uma reta aos pontos transformados, obtém-se os seguintes valores para m e b :

$$m = 4.0148, \quad (4.25)$$

$$b = -0.0030749. \quad (4.26)$$

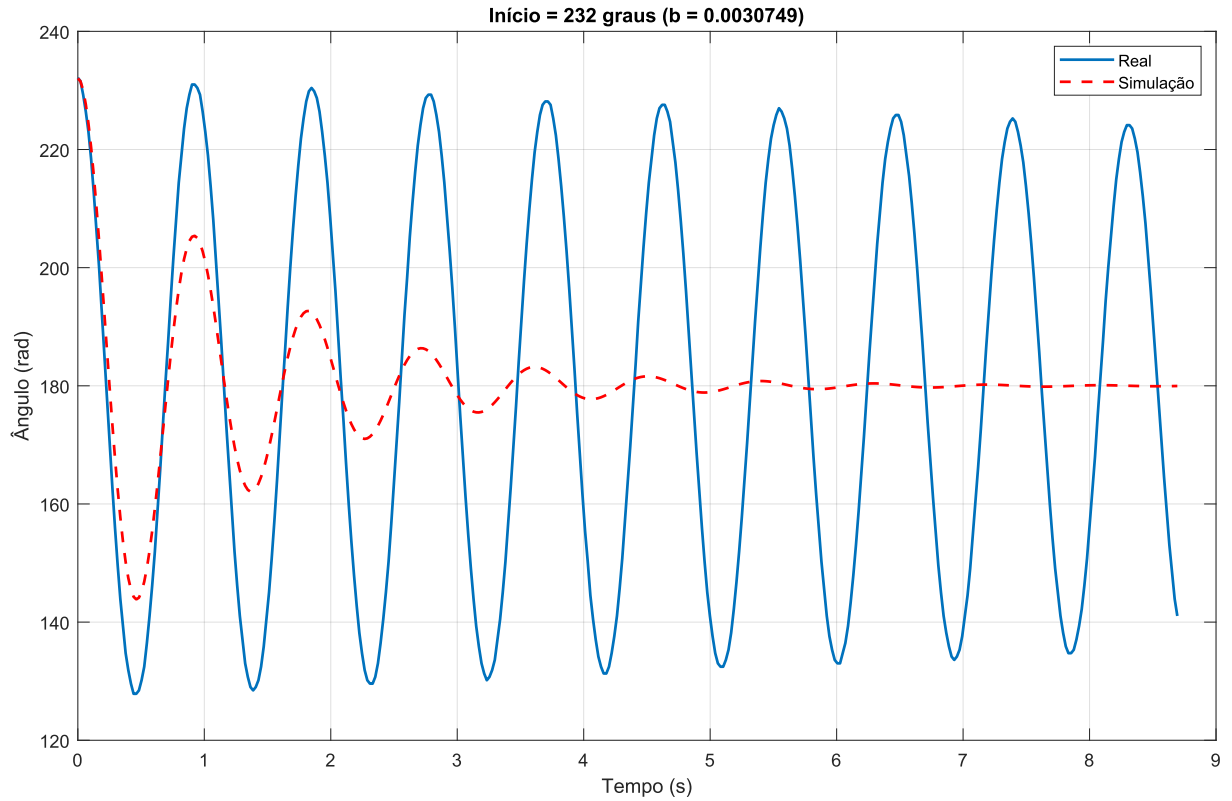
A Figura 17 mostra a curva de ajuste exponencial sobreposta aos picos observados da oscilação, indicando um bom ajuste inicial.

Figura 17 – Comparação dos picos da oscilação com a curva ajustada.



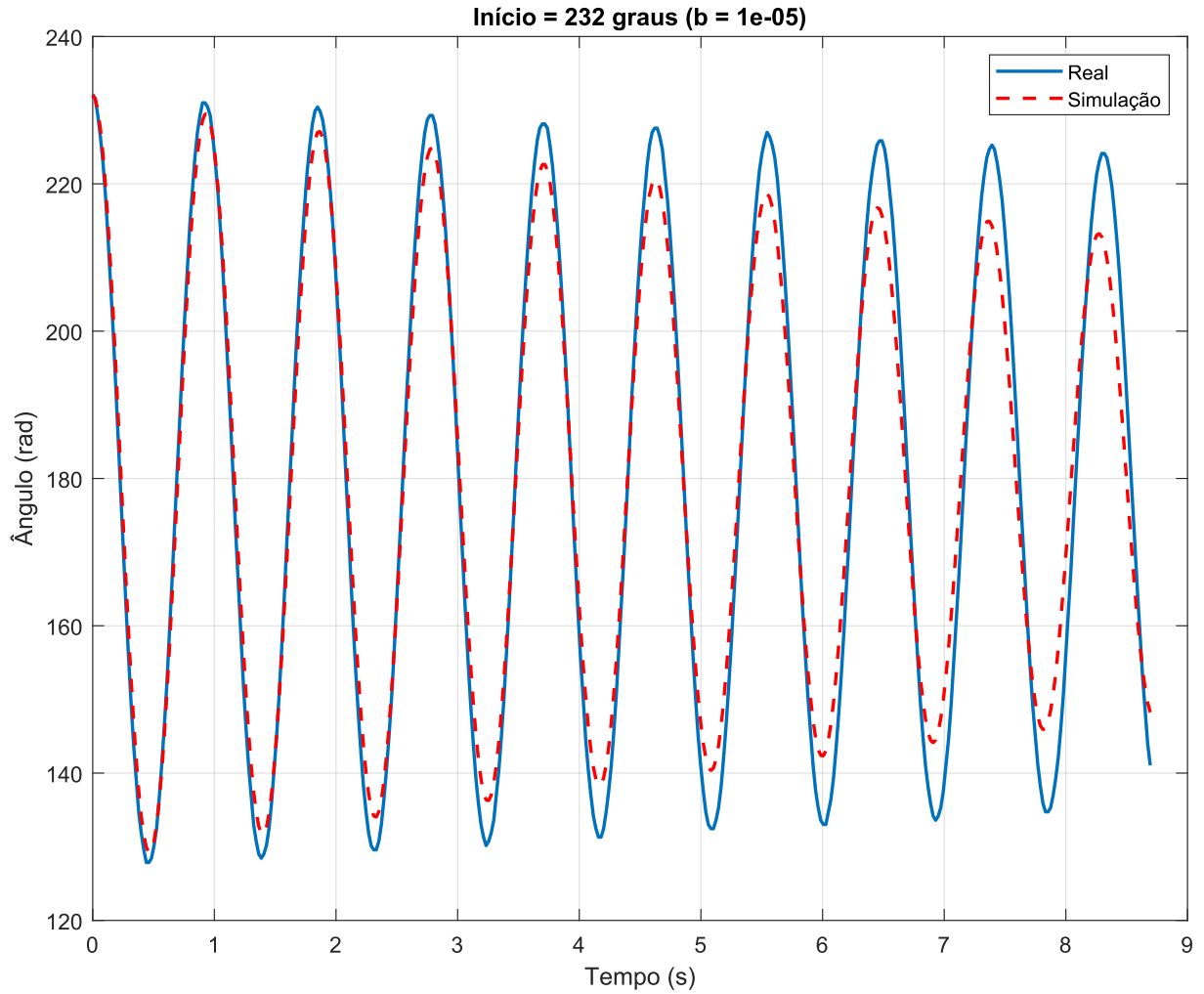
Fonte: O Autor (2024).

Para verificar a adequação desse coeficiente no modelo, foi realizada uma simulação com o sistema modelado usando $b = 0.0030749$. A Figura 18 ilustra a resposta comparativa entre a simulação e o sistema real.

Figura 18 – Comparação entre o sistema real e a simulação com $b = 0.0030749$.

Fonte: O Autor (2024).

Observou-se, contudo, que a resposta simulada apresentava pequenas discrepâncias em relação à oscilação real do sistema, provavelmente devido a acoplamentos e não-linearidades do sistema físico. Para otimizar o ajuste, foi realizada uma série de simulações com diferentes valores de b . Ao ajustar para um valor de $b = 1 \times 10^{-5}$, a resposta simulada mostrou-se mais próxima do comportamento observado experimentalmente, conforme mostrado na Figura 19.

Figura 19 – Comparação entre o sistema real e a simulação com $b = 0.00001$.

Fonte: O Autor (2024).

Este ajuste final de b foi considerado satisfatório para a modelagem do atrito no eixo do pêndulo, permitindo uma representação mais precisa do comportamento dinâmico do sistema.

4.5.4 Identificação do Coeficiente de Atrito do Carrinho com a Guia Linear (c)

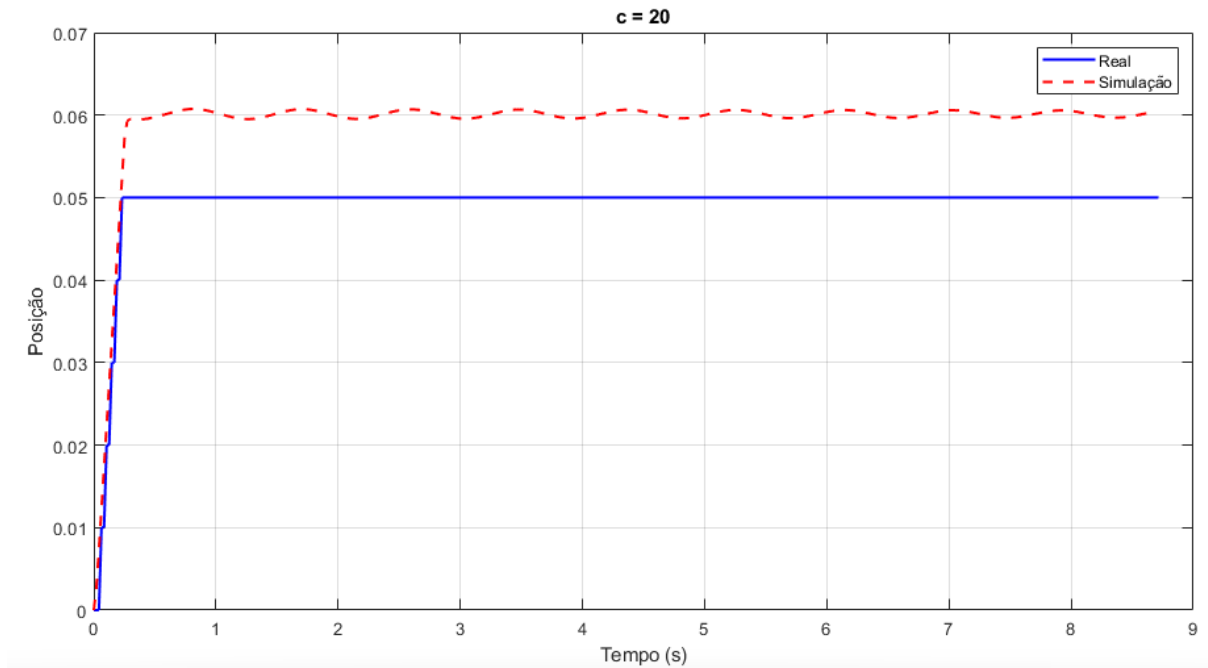
Para identificar o coeficiente de atrito c entre o carrinho e a guia linear, foi realizado um experimento que consistiu na aplicação de um sinal de entrada degrau de 1V no motor por um período de 250 ms, enquanto a posição do carrinho foi registrada ao longo do tempo. Esse sinal foi escolhido para analisar o comportamento transitório do sistema, que é diretamente influenciado pelo atrito entre o carrinho e a guia. A posição final do carrinho ao término do degrau foi de aproximadamente 0.05 metros. Os dados experimentais foram exportados para o MATLAB para análise e comparação com simulações do modelo.

Para validar o valor de c , o sistema foi simulado no *MATLAB/Simulink* com uma entrada degrau equivalente. Com base na conversão de tensão para força identificada anteriormente (Equação (4.21)), onde 1V equivale a 4.81 N, a simulação foi configurada com uma entrada de degrau de 4.81 N aplicada durante 250 ms. Dessa forma, a simulação representaria fielmente as condições do experimento real.

A simulação foi realizada para três valores distintos de c para encontrar o que melhor reproduzisse o deslocamento final do carrinho em 0.05 metros, de acordo com os dados experimentais.

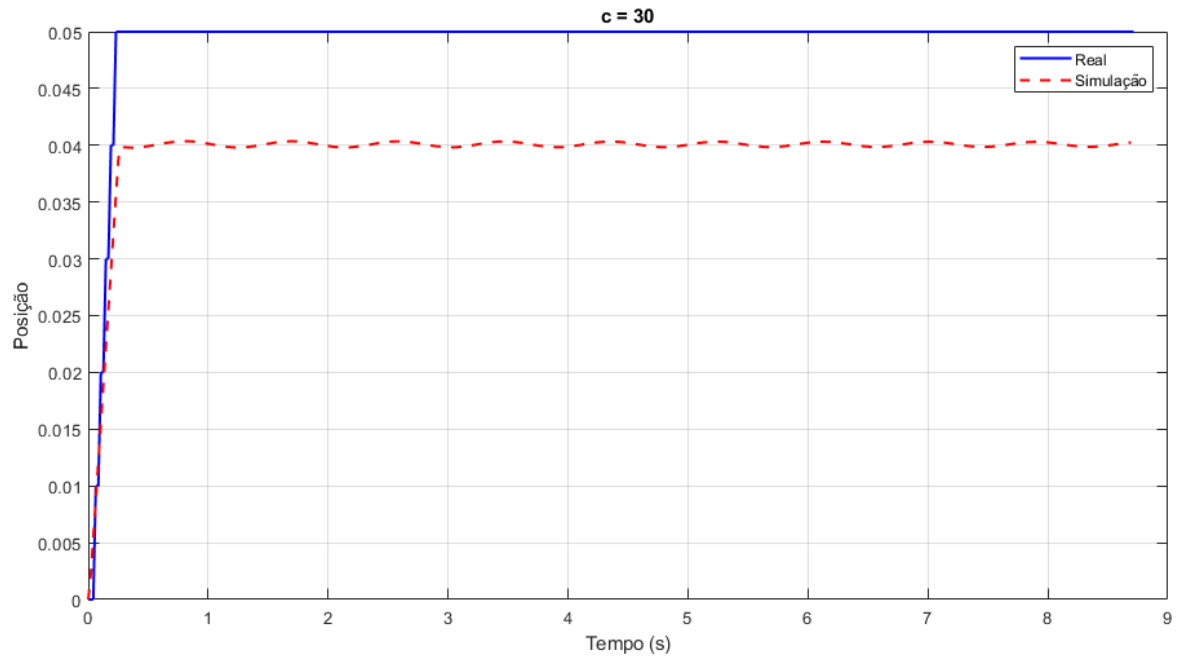
1. Simulação com $c = 20$: Nesse caso, o sistema simulado apresentou um deslocamento final superior ao do sistema real, alcançando aproximadamente 0.06 metros, como mostrado na Figura 20.

Figura 20 – Comparação entre sistema real e simulação com $c = 20$.



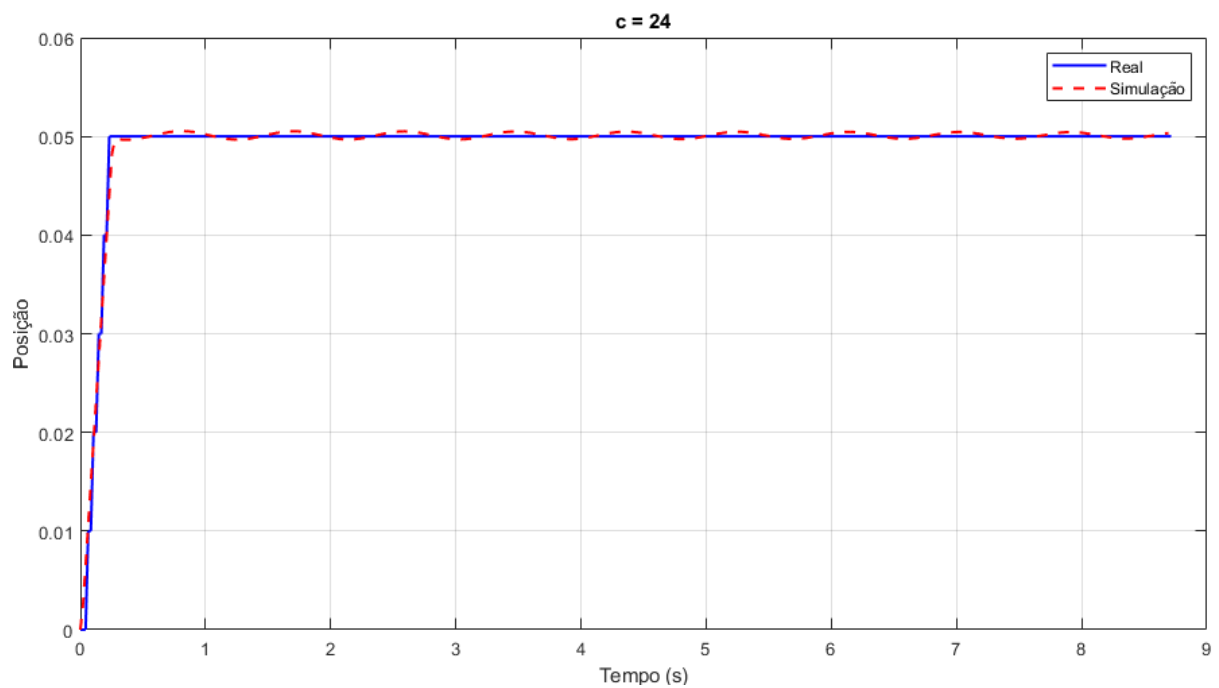
Fonte: O Autor (2024).

2. Simulação com $c = 30$: Com o valor $c = 30$, o deslocamento final do carrinho foi inferior ao observado experimentalmente, alcançando cerca de 0.04 metros, conforme ilustrado na Figura 21.

Figura 21 – Comparação entre sistema real e simulação com $c = 30$.

Fonte: O Autor (2024).

3. Simulação com $c = 24$: Ao ajustar o coeficiente de atrito para $c = 24$, o deslocamento final do carrinho na simulação foi aproximadamente 0.05 metros, coincidindo com o valor registrado no experimento real, conforme mostrado na Figura 22.

Figura 22 – Comparação entre sistema real e simulação com $c = 24$.

Fonte: O Autor (2024).

4.6 Parâmetros e Verificação do Modelo Final

A Tabela 5 apresenta os parâmetros completos da planta, incluindo os identificados experimentalmente, para o modelo do sistema desenvolvido.

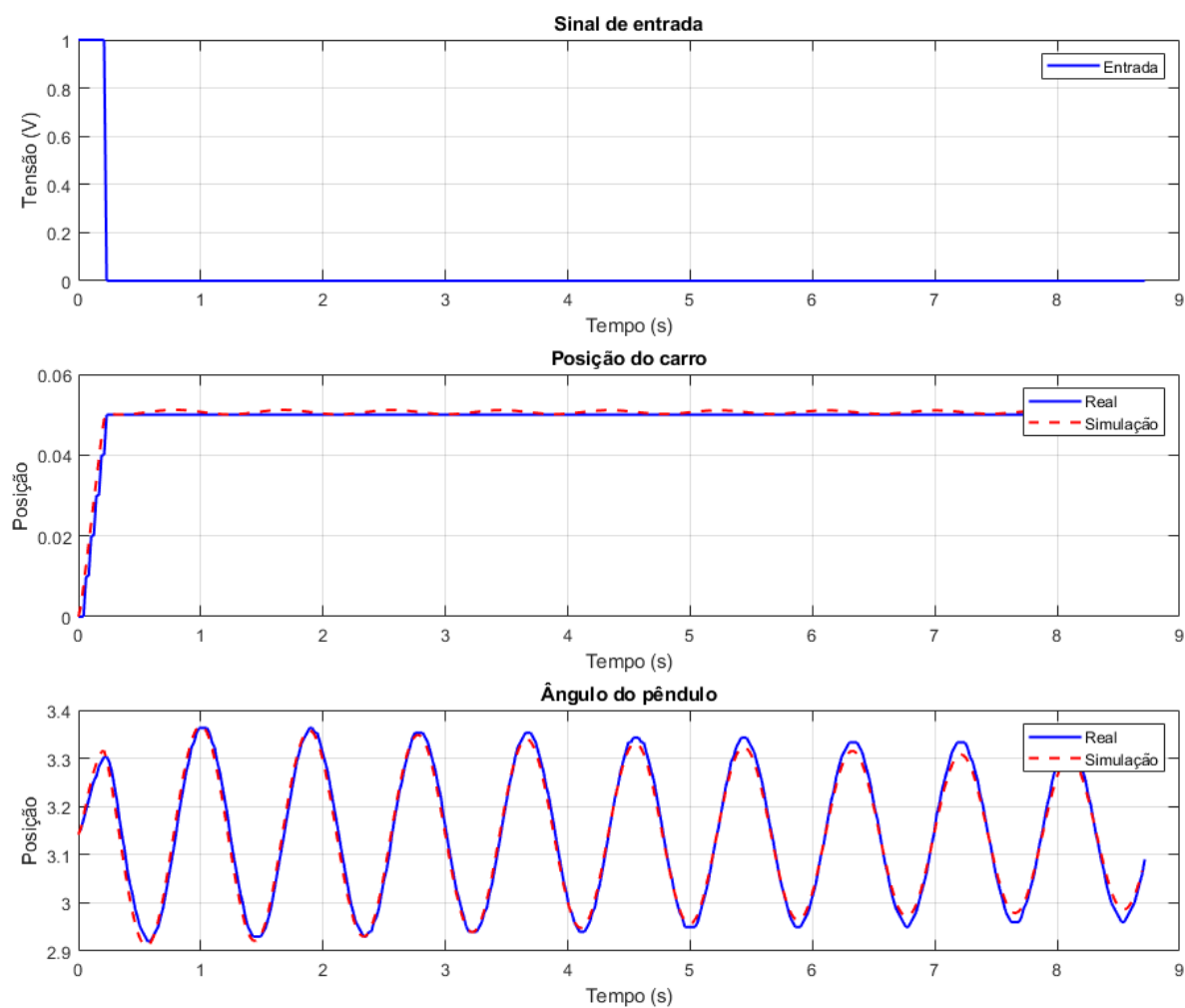
Tabela 5 – Parâmetros do sistema completo.

| Símbolo | Descrição | Valor |
|---------|--|--|
| M | Massa do carrinho | 0.2 kg |
| m | Massa do pêndulo | 0.075 kg |
| l | Distância ao centro de massa do pêndulo | 0.147 m |
| I | Momento de inércia do pêndulo | $5.402 \times 10^{-4} \text{ kg m}^2$ |
| g | Aceleração gravitacional | 9.81 m/s^2 |
| b | Coefficiente de amortecimento do pêndulo | $1 \times 10^{-5} \text{ kg m}^2/\text{s}$ |
| c | Coefficiente de atrito do carrinho | 24 kg/s |
| k | Conversão tensão-força | 4.81 N/V |

Fonte: O Autor (2024)

Para validar o modelo, foi aplicada uma entrada impulso unitário de tensão por 250 ms, de modo a comparar a resposta do sistema simulado com a resposta do sistema físico. A Figura 23 apresenta o comportamento do sinal de entrada em tensão (V), a posição do carrinho e o ângulo do pêndulo ao longo do tempo. A linha sólida representa a resposta do sistema real, enquanto a linha pontilhada corresponde à resposta obtida pelo modelo simulado. Observa-se que o modelo simulado reproduz de forma satisfatória tanto o deslocamento do carrinho quanto o comportamento do ângulo do pêndulo, indicando que os parâmetros identificados capturam com precisão as características dinâmicas do sistema.

Figura 23 – Resposta do sistema a um impulso unitário por 250 ms, comparando o sistema real e o simulado.



Fonte: O Autor (2024).

5 Sistema de Controle

Este capítulo aborda o desenvolvimento da estratégia de controle para estabilizar o sistema de pêndulo invertido. Serão apresentados os passos para linearizar o modelo do pêndulo invertido ao redor do ponto de equilíbrio usando a série de Taylor e a matriz Jacobiana. Serão descritas a construção das matrizes de estado \mathbf{A} e \mathbf{B} , a verificação de estabilidade e controlabilidade, e a implementação do controlador LQR em tempo discreto. Por fim, os resultados experimentais do sistema controlado serão analisados.

5.1 Linearização do Modelo Não Linear

O método da série de Taylor, combinado com a matriz Jacobiana, permite obter um modelo linearizado no espaço de estados (ver Seção 2.1). Na sequência, são apresentados os passos para obter linearização do sistema não linear (ver Equações (2.4) e (2.5)):

1. **Definir o Ponto de Equilíbrio:** O sistema deve ser linearizado em torno do ponto de equilíbrio onde o pêndulo está na posição vertical ($\theta = 0$) e o carro está em repouso ($\dot{x} = 0, \dot{\theta} = 0$). Portanto, o ponto de operação aproximado é definido como $(\mathbf{x}_0, u_0) = ([0, 0, 0, 0], 0)$.

2. **Construção das Matrizes \mathbf{A} e \mathbf{B} :**

- **Matriz \mathbf{A} :** A matriz \mathbf{A} contém as derivadas parciais de cada função f_i , conforme a Equação (2.8), em relação a cada variável de estado x_j , avaliadas no ponto de equilíbrio, e é dada por:

$$\mathbf{A} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, u_0)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \bigg|_{(\mathbf{x}_0, u_0)}$$

Após resolver as derivadas parciais, obtemos a matriz \mathbf{A} como:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m^2 l^2 g}{I(M+m)+Mml^2} & -\frac{(I+ml^2)c}{I(M+m)+Mml^2} & \frac{bml}{I(M+m)+Mml^2} \\ 0 & -\frac{mgl(M+m)}{I(M+m)+Mml^2} & \frac{cml}{I(M+m)+Mml^2} & -\frac{(M+m)b}{I(M+m)+Mml^2} \end{bmatrix}$$

- **Matriz \mathbf{B} :** A matriz \mathbf{B} contém as derivadas parciais das funções f_i em relação à entrada u , também avaliadas no ponto de equilíbrio. No sistema de pêndulo invertido (Figura 12), a entrada u corresponde à força F aplicada ao carrinho. Assim, temos $u = F$,

dada por:

$$\mathbf{B} = \left. \frac{\partial f}{\partial F} \right|_{(x_0, F_0)} = \left. \begin{bmatrix} \frac{\partial f_1}{\partial F} \\ \frac{\partial f_2}{\partial F} \\ \frac{\partial f_3}{\partial F} \\ \frac{\partial f_4}{\partial F} \end{bmatrix} \right|_{(x_0, F_0)}$$

Após resolver as derivadas parciais, obtemos a matriz B como:

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ -\frac{ml}{I(M+m)+Mml^2} \end{bmatrix}$$

3. Modelo Linearizado no Espaço de Estados:

A partir da Equação 2.6, onde $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$ é o vetor de estados, com x_1 representando a posição do carro, x_2 o ângulo do pêndulo em relação à posição vertical, x_3 a velocidade do carro e x_4 a velocidade angular do pêndulo, e F é a força aplicada ao carrinho, responsável por controlar o movimento do sistema e estabilizar o pêndulo, podemos substituir nas matrizes de \mathbf{A} e \mathbf{B} , obtendo:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m^2 l^2 g}{I(M+m)+Mml^2} & -\frac{(I+ml^2)c}{I(M+m)+Mml^2} & \frac{bml}{I(M+m)+Mml^2} \\ 0 & -\frac{mgl(M+m)}{I(M+m)+Mml^2} & \frac{cml}{I(M+m)+Mml^2} & -\frac{(M+m)b}{I(M+m)+Mml^2} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ -\frac{ml}{I(M+m)+Mml^2} \end{bmatrix} F$$

Assim, chegamos ao modelo linearizado do sistema em torno do ponto de equilíbrio, onde o pêndulo está na posição vertical.

Para aplicar esse modelo com a tensão V como entrada de controle, utilizamos a relação de conversão entre a tensão e a força, conforme descrito anteriormente pela Equação (4.20), obtendo-se:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m^2 l^2 g}{I(M+m)+Mml^2} & -\frac{(I+ml^2)c}{I(M+m)+Mml^2} & \frac{bml}{I(M+m)+Mml^2} \\ 0 & -\frac{mgl(M+m)}{I(M+m)+Mml^2} & \frac{cml}{I(M+m)+Mml^2} & -\frac{(M+m)b}{I(M+m)+Mml^2} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{(I+ml^2)k}{I(M+m)+Mml^2} \\ -\frac{mlk}{I(M+m)+Mml^2} \end{bmatrix} V$$

4. Parametrização das Matrizes \mathbf{A} e \mathbf{B} :

Após substituir os parâmetros específicos do sistema, apresentados na Tabela 5, obtemos as matrizes A e B , conforme:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ 0 & -2.5226 & -109.7143 & 0.0002 \\ 0 & 62.4961 & 555.9846 & -0.0057 \end{bmatrix} \quad (5.1)$$

e

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 21.9886 \\ -111.4286 \end{bmatrix} \quad (5.2)$$

5. Verificação de Estabilidade:

Conforme visto na Equação (2.10), a estabilidade de um sistema no espaço de estados pode ser determinada calculando os autovalores da matriz \mathbf{A} e verificando se a parte real de cada autovalor é positiva ou negativa. Para a matriz obtida em (5.1), definindo λ como a matriz de autovalores, temos que:

$$\lambda = \begin{bmatrix} 0 \\ -109.8332 \\ 7.1276 \\ -7.0145 \end{bmatrix} \quad (5.3)$$

onde um dos polos está localizado no semiplano direito ($\lambda_3 = 7.1276$) e, portanto, o sistema em malha aberta é instável.

6. Verificação de Controlabilidade:

Para assegurar que o sistema de pêndulo invertido seja controlável, realizamos a verificação de controlabilidade calculando a matriz de controlabilidade \mathbf{MC} . Esta matriz é construída de acordo com a Equação (2.9), sendo definida como:

$$\mathbf{MC} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (5.4)$$

onde \mathbf{A} e \mathbf{B} representam as matrizes de estado e entrada do sistema, e n é a ordem do sistema (número de estados). Para que o sistema seja considerado controlável, o posto da matriz \mathbf{MC} deve ser igual a n , indicando que a entrada de controle pode influenciar todas as dinâmicas do sistema.

Após realizar as operações necessárias, a matriz de controlabilidade \mathbf{MC} obtida é:

$$\mathbf{MC} = 10^8 \times \begin{bmatrix} 0 & 0.0000 & -0.0000 & 0.0026 \\ 0 & -0.0000 & 0.0001 & -0.0136 \\ 0.0000 & -0.0000 & 0.0026 & -0.2910 \\ -0.0000 & 0.0001 & -0.0136 & 1.4910 \end{bmatrix} \quad (5.5)$$

O posto da matriz \mathbf{MC} é 4, igual à ordem do sistema, o que confirma que o sistema é controlável e que todas as dinâmicas podem ser influenciadas pela entrada de controle.

5.2 Implementação do Controlador LQR

Para implementar o controlador LQR em tempo discreto, o sistema contínuo foi discretizado no MATLAB usando a função `c2d`, como ilustrado no código 2.1.2. A discretização foi realizada com um tempo de amostragem de $T_s = 0.02$ s, que é adequado para capturar a dinâmica do sistema sem perda significativa de informações. As matrizes de estado discretas resultantes, \mathbf{A}_d e \mathbf{B}_d , são:

$$\mathbf{A}_d = \begin{bmatrix} 1.0000 & -0.0003 & 0.0081 & -0.0000 \\ 0 & 1.0114 & 0.0605 & 0.0201 \\ 0 & -0.0205 & 0.1107 & -0.0003 \\ 0 & 1.1022 & 4.5292 & 1.0113 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} 0.0024 \\ -0.0121 \\ 0.1782 \\ -0.9077 \end{bmatrix} \quad (5.6)$$

onde \mathbf{A}_d é a representação discreta da matriz \mathbf{A} , e \mathbf{B}_d é a representação discreta da matriz \mathbf{B} , correspondendo à dinâmica do sistema após a discretização.

5.2.1 Ajuste das Ponderações \mathbf{Q} e \mathbf{R}

A escolha dos parâmetros de ponderação \mathbf{Q} e \mathbf{R} influencia diretamente a resposta do sistema. O ajuste dessas matrizes foi realizado com base em experimentos práticos, buscando priorizar a precisão no controle da posição do carro, representada pelo estado x_1 , ao mesmo tempo em que se mantém o pêndulo na posição vertical.

Para atender a esses requisitos, atribuímos pesos maiores para o estado de posição do carro, resultando em uma resposta rápida com menor variação de posição. O peso relativo para o ângulo do pêndulo foi menor, pois a prioridade era garantir que o sistema focasse mais no controle da posição do carro.

As ponderações \mathbf{Q} e \mathbf{R} selecionadas foram:

$$\mathbf{Q} = \begin{bmatrix} 40 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad \text{e} \quad \mathbf{R} = [0.001]$$

onde: \mathbf{Q} é uma matriz diagonal de pesos para os estados, onde os valores refletem a prioridade de cada estado no controle. O valor mais alto, 40, foi atribuído ao estado de posição do carro x_1 , pois notou-se que a resposta do sistema é mais rápida com o peso da posição sendo maior, enquanto os demais estados têm valores menores. \mathbf{R} é um escalar que penaliza a magnitude da entrada de controle, limitando a agressividade das ações de controle para evitar oscilações excessivas.

As matrizes \mathbf{Q} e \mathbf{R} foram aplicadas na função `dlqr` do MATLAB, como mostrado no código 2.3. A função `dlqr` resolve a Equação de Riccati associada ao sistema discretizado, fornecendo o vetor de ganhos \mathbf{K}_d , que minimiza o custo total de controle e permite uma estabilização eficiente do sistema.

O ganho \mathbf{K}_d resultante para o sistema é:

$$\mathbf{K}_d = \begin{bmatrix} -18.7855 & -20.2044 & -13.6020 & -2.9104 \end{bmatrix}$$

A seguinte matriz \mathbf{e} representa os autovalores em malha fechada em regime discreto do sistema:

$$\mathbf{e} = \begin{bmatrix} 0.9120 + 0.0668i \\ 0.9120 - 0.0668i \\ 0.8907 + 0.0000i \\ 0.0013 + 0.0000i \end{bmatrix},$$

onde todos os autovalores estão dentro do círculo de raio unitário, evidenciando a estabilidade do sistema.

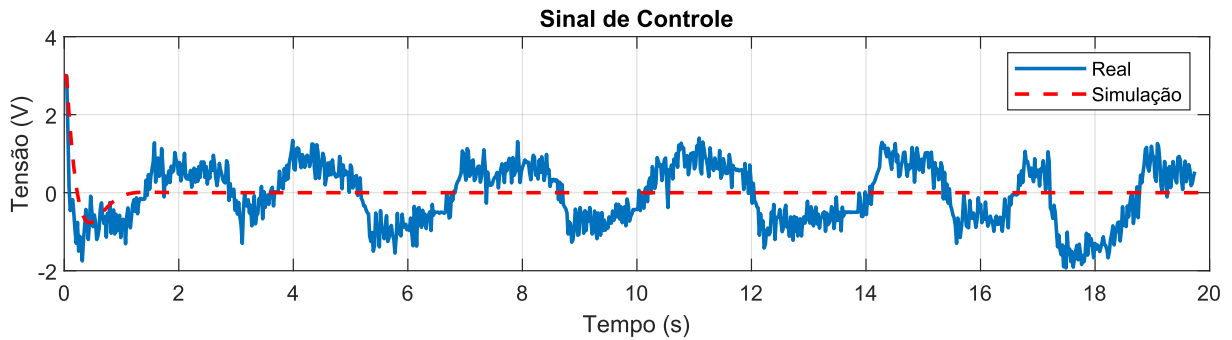
5.2.2 Resultados Experimentais e Análise

Com o controlador LQR implementado no ambiente MATLAB e no microcontrolador ESP32 (Anexo A), o sistema foi testado para verificar a capacidade de estabilização do pêndulo em torno da posição vertical, mesmo com um deslocamento inicial. Inicialmente, o pêndulo foi posicionado a 11.4° (0.2 rad) da posição vertical e, em seguida, foi acionado o controle automático para o retorno ao ponto de equilíbrio (0°). Os gráficos a seguir mostram o comportamento do sistema para cada variável monitorada, comparando entre a simulação do LQR aplicado ao modelo não linear com a planta física.

Sinal de Controle - Tensão no Motor

O gráfico da Figura 24 mostra o comportamento do sinal de controle, ou seja, a tensão aplicada ao motor ao longo do tempo. Observa-se que, no instante inicial, o controlador aplica uma tensão elevada de cerca de -3 V para compensar o deslocamento do pêndulo. Após essa resposta inicial, o controle se estabiliza abaixo de 1.5 V em módulo, conforme o pêndulo retorna à posição de equilíbrio.

Figura 24 – Sinal de controle aplicado ao motor.

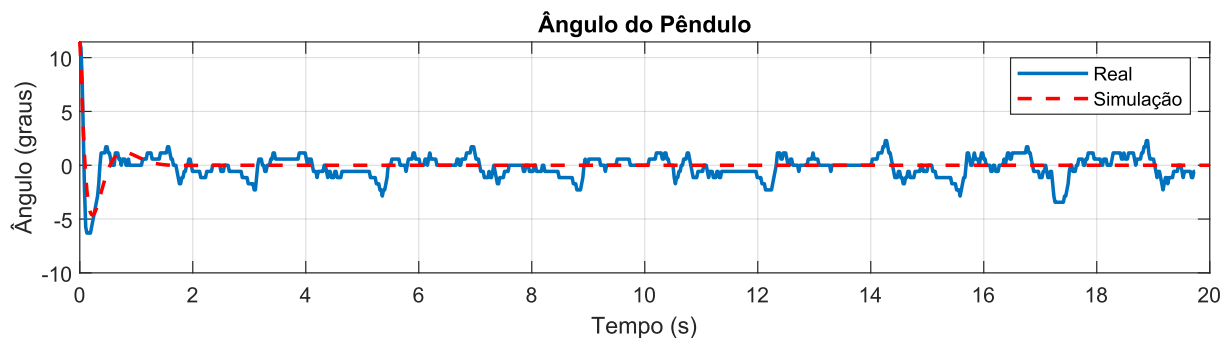


Fonte: O Autor (2024).

Ângulo do Pêndulo

A Figura 25 apresenta o comportamento do ângulo do pêndulo ao longo do tempo. O zero para o ângulo foi definido na posição vertical, ou seja, o topo do movimento do pêndulo, conforme mostrado na Figura 12. Neste contexto, a posição π representa o pêndulo apontando diretamente para baixo, sua posição natural de repouso. Observa-se que o controlador LQR estabiliza rapidamente o ângulo do pêndulo em torno de 0 graus, com oscilações inferiores a 3 graus. Essa condição de estabilidade é alcançada em menos de 0.5 s, evidenciando a eficiência do controlador LQR na correção do deslocamento inicial.

Figura 25 – Comportamento do ângulo do pêndulo ao longo do tempo.

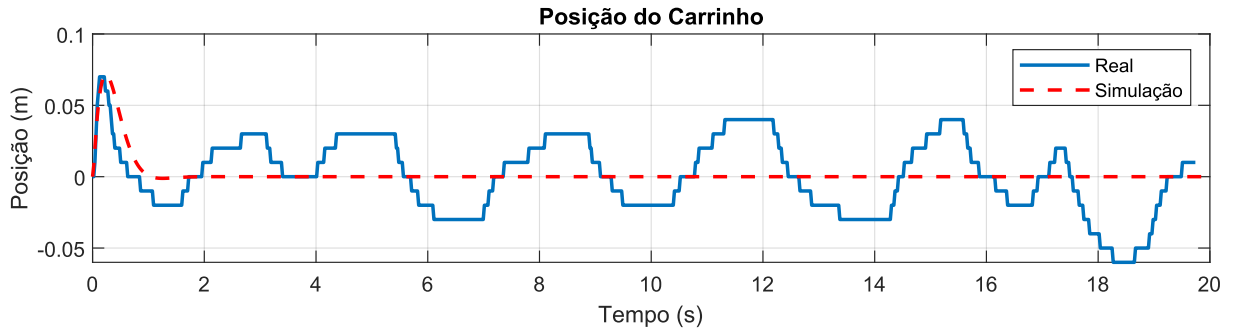


Fonte: O Autor (2024).

Posição do Carrinho

A posição do carrinho em relação ao ponto de equilíbrio é mostrada na Figura 26. A resposta mostra um deslocamento máximo de 0.07 m durante a fase inicial de estabilização, enquanto o pêndulo se realinha verticalmente. Após essa oscilação inicial, a posição do carrinho se estabiliza com variações inferiores a 0.03 m.

Figura 26 – Posição do carrinho ao longo do tempo.



Fonte: O Autor (2024).

Velocidade Angular do Pêndulo e Velocidade do Carrinho

As velocidades do pêndulo e do carrinho foram obtidas a partir das medidas de posição e ângulo, calculando a taxa de variação em um sistema discreto. Esse método permite estimar a velocidade a partir das amostras discretas de posição $x[k]$ e ângulo $\theta[k]$ ao longo do tempo.

Para a velocidade angular do pêndulo, cada valor de velocidade $\omega[k]$ foi calculado pela diferença entre os ângulos consecutivos dividida pelo intervalo de amostragem T_s :

$$\omega[k] = \frac{\theta[k] - \theta[k-1]}{T_s} \quad (5.7)$$

onde $\theta[k]$ representa o ângulo do pêndulo no instante k e T_s é o tempo entre amostras. Esse cálculo fornece uma aproximação da velocidade angular média entre duas amostras consecutivas, permitindo observar a taxa de variação do ângulo ao longo do tempo.

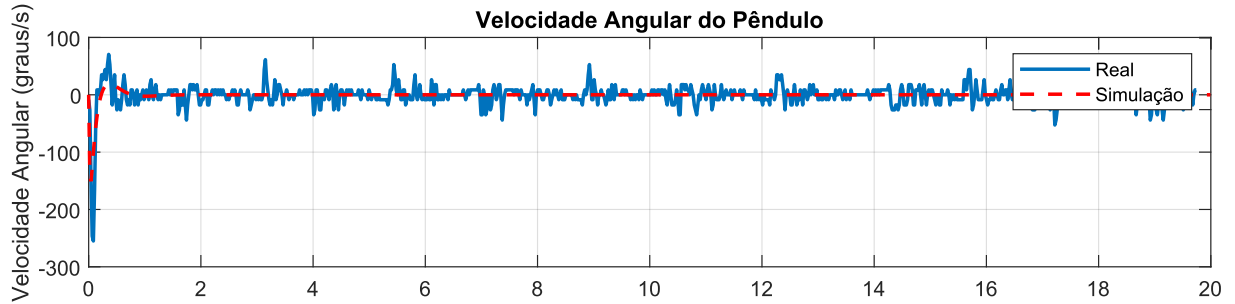
Para a velocidade linear do carrinho, o mesmo método foi aplicado às amostras de posição $x[k]$, resultando na velocidade $v[k]$:

$$v[k] = \frac{x[k] - x[k-1]}{T_s} \quad (5.8)$$

onde $x[k]$ é a posição do carrinho no instante k . Assim como no cálculo da velocidade angular, essa abordagem fornece uma estimativa da velocidade média do carrinho entre as amostras consecutivas, permitindo monitorar sua taxa de deslocamento ao longo do tempo.

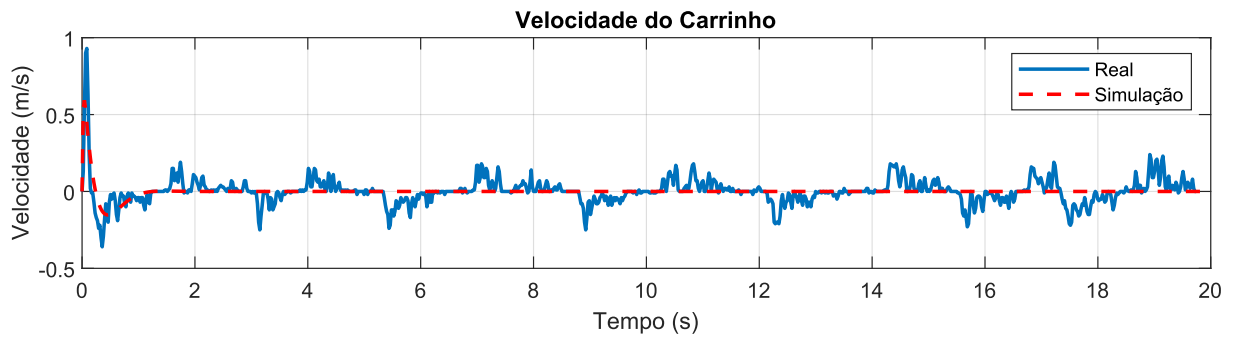
Os gráficos nas Figuras 27 e 28 mostram a resposta das velocidades angular e linear com base nessas estimativas. Observa-se que ambas as velocidades apresentam um comportamento coerente com as variações de posição e ângulo, alcançando valores próximos de 0 após a ação inicial do controlador.

Figura 27 – Velocidade angular do pêndulo ao longo do tempo.



Fonte: O Autor (2024).

Figura 28 – Velocidade do carrinho ao longo do tempo.



Fonte: O Autor (2024).

Os resultados obtidos demonstram a capacidade do controlador LQR em estabilizar o pêndulo invertido na posição vertical, mesmo a partir de uma condição inicial de desequilíbrio. O sinal de controle inicialmente aplicado foi suficiente para contrabalançar o ângulo de 11.4° , e o sistema estabilizou as variáveis de posição e ângulo em menos de 0.5 s. As oscilações das variáveis permaneceram dentro de limites aceitáveis, mostrando a eficácia do ajuste realizado nas matrizes \mathbf{Q} e \mathbf{R} para controlar as oscilações e garantir o rápido retorno ao equilíbrio.

6 Conclusão

Este trabalho apresentou o desenvolvimento, modelagem e controle de um sistema de pêndulo invertido, um modelo amplamente utilizado em engenharia de controle devido à sua natureza instável e dinâmica complexa. A partir da construção de uma planta física e da implementação de um controlador Regulador Quadrático Linear (LQR), foi possível estabilizar o pêndulo na posição vertical sobre um carrinho móvel. A modelagem matemática foi realizada utilizando a formulação de Lagrange e a linearização das equações de movimento em torno do ponto de equilíbrio permitiu a aplicação de técnicas de controle linear.

O controlador LQR, projetado com base no modelo linearizado, demonstrou ser eficaz em manter o equilíbrio do pêndulo, compensando perturbações externas e minimizando o esforço de controle. A simulação inicial no ambiente MATLAB/Simulink permitiu ajustes e validações preliminares, enquanto os testes experimentais na planta física confirmaram a precisão e a estabilidade do sistema sob condições reais de operação. A identificação experimental de parâmetros, como atrito e amortecimento, contribuiu para melhorar a precisão do modelo matemático, garantindo que ele representasse com fidelidade a dinâmica observada na planta física.

Este estudo contribui para a área de controle de sistemas ao demonstrar uma metodologia prática e validada para o controle de sistemas instáveis, evidenciando a viabilidade do controlador LQR para o pêndulo invertido. No entanto, este trabalho se concentrou na estabilização do pêndulo na posição vertical e no controle em torno do ponto de equilíbrio.

6.1 Trabalhos Futuros

Para futuros trabalhos, diversas extensões podem ser exploradas para expandir as capacidades do sistema de pêndulo invertido:

- **Controle de Swing-Up:** Uma limitação do controlador LQR é que ele opera apenas em torno do ponto de equilíbrio. O controle de *swing-up* permitiria estabilizar o pêndulo partindo de uma posição inferior, aplicando forças controladas para levantar o pêndulo até a posição vertical. Essa abordagem é mais complexa, pois envolve controle não linear, exigindo a implementação de técnicas como controle de energia ou estratégias híbridas combinando o *swing-up* com a estabilização por LQR.
- **Rastreamento de Trajetória (Trajectory Tracking):** Outra extensão relevante é o

controle de rastreamento de trajetória, que visa controlar não apenas o ângulo do pêndulo, mas também a posição e o movimento do carrinho de acordo com uma trajetória desejada. Esse tipo de controle é aplicável em situações onde o carrinho precisa seguir uma rota específica enquanto mantém o pêndulo em equilíbrio, como em sistemas de transporte autônomo. O rastreamento de trajetória requer uma abordagem mais complexa, integrando o controle LQR com técnicas de controle preditivo ou adaptativo.

- **Implementação de Controladores Adaptativos e Não Lineares:** Expandir o trabalho com controladores adaptativos ou não lineares pode aumentar a faixa de estabilidade do sistema e melhorar o desempenho em condições de grandes perturbações ou variações nos parâmetros. Esses controladores poderiam ajustar automaticamente seus parâmetros em tempo real para responder a mudanças nas características do sistema, oferecendo maior robustez em comparação ao controlador LQR tradicional.

Essas extensões abririam novas possibilidades de aplicação para o sistema de pêndulo invertido e permitiriam explorar um controle ainda mais sofisticado para sistemas instáveis, sendo uma base sólida para futuras pesquisas e desenvolvimentos na área de controle de sistemas dinâmicos.

Referências Bibliográficas

AGUIRRE, L. *Introdução à Identificação de Sistemas*. [S.l.: s.n.], 2015. ISBN 978-85-423-0079-6. Citado 2 vezes nas páginas 16 e 17.

BROCK, S. Identification of the parameters in inverted pendulum model [dc motor control]. p. 63–68, 2002. Citado na página 16.

CHENG, H. X. et al. Optimal control for single inverted pendulum based on linear quadratic regulator. v. 44, p. 02064, 2016. Citado na página 8.

COŞKUN, S. Non-linear control of inverted pendulum. *Çukurova University Journal of the Faculty of Engineering and Architecture*, v. 35, n. 1, p. 27–38, 2020. Citado na página 8.

DORF, R. C.; BISHOP, R. H. *Modern Control Systems*. 13. ed. [S.l.]: Pearson, 2010. Citado 5 vezes nas páginas 14, 15, 16, 17 e 18.

IACOB, C.-G. Linear and angular position control of a custom built stepper motor driven self-balancing robot. *Proceedings of the International Conference on System Theory, Control and Computing (ICSTCC)*, 2020. Citado na página 8.

KUMAR, R.; SINGH, R. B.; DAS, J. Modeling and simulation of inverted pendulum system using matlab: Overview. *International Journal of Mechanical and Production Engineering*, v. 1, n. 4, p. 52–56, 2013. Citado na página 12.

LEMO, N. *Mecânica Analítica*. Editora Livraria da Física, 2013. ISBN 9788588325241. Disponível em: <<https://books.google.com.br/books?id=XpOS2eOdDR4C>>. Citado 2 vezes nas páginas 12 e 13.

LENKA, N. Modeling and controller design for an inverted pendulum system. 2011. Citado na página 8.

LJUNG, L. *System identification*. 2. ed. Philadelphia, PA: Prentice Hall, 1998. (Prentice Hall information and system sciences series). Citado 2 vezes nas páginas 16 e 17.

LYAPUNOV, A. M. *The General Problem of the Stability of Motion*. Kharkov, Russia: University of Kharkov, 1892. v. 55. Originally published as a doctoral dissertation at the University of Kharkov. Citado na página 14.

MATLAB. *version R2022a*. Natick, Massachusetts: The MathWorks Inc., 2022. Citado 2 vezes nas páginas 15 e 18.

NASIR, A.; AHMAD, M.; RAHMAT, M. Performance comparison between lqr and pid controller for an inverted pendulum system. *International Conference on Power Control and Optimization*, 2008. Citado na página 8.

OGATA, K. *Modern Control Engineering*. 5. ed. [S.l.]: Prentice Hall, 2010. Citado 3 vezes nas páginas 13, 17 e 18.

PRASAD, L. B.; TYAGI, B.; GUPTA, H. O. Optimal control of nonlinear inverted pendulum system using pid controller and lqr: Performance analysis without and with disturbance input. *International Journal of Automation and Computing*, Springer, v. 11, n. 6, p. 647–657, 2014. Citado na página [19](#).

RUŠAR, L. et al. State-space predictive control of inverted pendulum model. 2017. Citado na página [13](#).

ANEXO A – Código Implementado no ESP32

```

1  #include <ESP32Encoder.h>
2
3  #define CLK_1 14 // CLK ENCODER 1 (position MEASUREMENT)
4  #define DT_1 27 // DT ENCODER 1
5  #define CLK_2 26 // CLK ENCODER 2 (angle MEASUREMENT)
6  #define DT_2 25 // DT ENCODER 2
7  #define MOTOR_CLOCKWISE_PIN 12
8  #define MOTOR_COUNTERCLOCKWISE_PIN 32
9  #define ENCODER_ENABLE_PIN 33
10 #define MOTOR_ENABLE_PIN 13
11 #define ENCODER_PULSES_PER_REV 2048
12 #define SAMPLE_TIME 20
13 #define MAX_INPUT 3
14
15 typedef struct states_t {
16     float position = 0;
17     float angle = 0;
18     float linear_speed = 0;
19     float angular_speed = 0;
20 };
21
22 states_t states;
23
24 ESP32Encoder encoder_angle;
25 ESP32Encoder encoder_position;
26
27 float lqr_controller() {
28     float K_1 = -18.7855, K_2 = -20.2044, K_3 = -13.6020, K_4 = -2.9104;
29     float input = -(K_1 * states.position + K_2 * states.angle + K_3 *
30
31     return input;
32 }
33
34 void setup() {
35     //encoders setup
36     encoder_angle.attachHalfQuad(DT_2, CLK_2);
37     encoder_angle.setCount(0);
38     encoder_position.attachHalfQuad(DT_1, CLK_1);
39     encoder_position.setCount(0);
40
41     //motor setup

```

```
42  pinMode(MOTOR_CLOCKWISE_PIN, OUTPUT);
43  pinMode(MOTOR_COUNTERCLOCKWISE_PIN, OUTPUT);
44  pinMode(MOTOR_ENABLE_PIN, OUTPUT);
45  pinMode(ENCODER_ENABLE_PIN, OUTPUT);
46
47  digitalWrite(MOTOR_ENABLE_PIN, HIGH);
48  digitalWrite(ENCODER_ENABLE_PIN, HIGH);
49  analogWrite(MOTOR_CLOCKWISE_PIN, 0);
50  analogWrite(MOTOR_COUNTERCLOCKWISE_PIN, 0);
51
52  Serial.begin(115200);
53 }
54
55 float calc_angle_from_count(long encoder_angle_count) {
56     if (encoder_angle_count < 0)
57         return float(encoder_angle_count % ENCODER_PULSES_PER_REV +
58             ENCODER_PULSES_PER_REV) * 2 * 3.141592 /
59             ENCODER_PULSES_PER_REV - 3.141592;
60
61     return float(encoder_angle_count % ENCODER_PULSES_PER_REV) * 2 *
62         3.141592 / ENCODER_PULSES_PER_REV - 3.141592;
63 }
64
65 void motor_write_pwm(float input) {
66
67     digitalWrite(MOTOR_COUNTERCLOCKWISE_PIN, LOW);
68     digitalWrite(MOTOR_CLOCKWISE_PIN, LOW);
69
70     if (input == 0) {
71         digitalWrite(MOTOR_ENABLE_PIN, LOW);
72         return;
73     }
74
75     digitalWrite(MOTOR_ENABLE_PIN, HIGH);
76     int input_pwm = (int)abs(input / 12 * 255);
77
78     analogWrite((input < 0) ? MOTOR_CLOCKWISE_PIN : MOTOR_COUNTERCLOCKWISE_PIN,
79         input_pwm);
80 }
81
82 float position_reference = 0;
83 float position_integral_error = 0;
84
85 void loop() {
86     long encoder_angle_count = -encoder_angle.getCount();
87     long encoder_position_count = encoder_position.getCount();
88     static float input = 0;
89     float dt = 0.001 * SAMPLE_TIME;
90
```

```
91  if (millis() - time_elapsed > SAMPLE_TIME) {
92      time_elapsed = millis();
93
94      states.angular_speed = (calc_angle_from_count(encoder_angle_count) -
95                             states.angle) / dt;
96      states.linear_speed = (float(encoder_position_count) * 0.00001953125 -
97                             states.position) / dt;
98      states.position = float(encoder_position_count) * 0.00001953125;
99      states.angle = calc_angle_from_count(encoder_angle_count);
100
101      input = lqr_controller();
102
103      if (input > MAX_INPUT) input = MAX_INPUT;
104      if (input < -MAX_INPUT) input = -MAX_INPUT;
105
106      if (abs(states.angle) > 0.3 || abs(states.position) > 0.25) input = 0;
107
108      motor_write_pwm(input);
109  }
110 }
```
