



**Centro Federal de Educação Tecnológica de Minas Gerais  
Campus Araxá**

Pedro Henrique de Oliveira Martins Quintão

**Desenvolvimento de um Pêndulo Invertido Visando Criar Aplicações  
Práticas para a Disciplina de Controle Multivariável**

Araxá-MG  
2022

Pedro Henrique de Oliveira Martins Quintão

**Desenvolvimento de um Pêndulo Invertido Visando Criar Aplicações  
Práticas para a Disciplina de Controle Multivariável**

Monografia apresentada ao Curso de Engenharia de Automação Industrial do Centro Federal de Educação Tecnológica de Minas Gerais - Campus Araxá, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Automação Industrial.

Orientador: Prof. Dr. Henrique José da Costa Avelar.

Coorientadora: Profa. Dra. Aline Fernanda Bianco Mattioli.

Araxá-MG  
2022

Pedro Henrique de Oliveira Martins Quintão

**Desenvolvimento de um Pêndulo Invertido Visando Criar Aplicações  
Práticas para a Disciplina de Controle Multivariável**

Monografia apresentada ao Curso de Engenharia de Automação Industrial do Centro Federal de Educação Tecnológica de Minas Gerais - Campus Araxá, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Automação Industrial.

**BANCA AVALIADORA**

---

Presidente e Orientador: Prof. Dr. Henrique José da Costa Avelar  
Centro Federal de Educação Tecnológica de Minas Gerais - Campus Araxá

---

Coorientadora: Profa. Dra. Aline Fernanda Bianco Mattioli.  
Centro Federal de Educação Tecnológica de Minas Gerais - Campus Araxá

---

Prof. Me. Luis Paulo Fagundes  
Centro Federal de Educação Tecnológica de Minas Gerais - Campus Araxá

---

Prof. Carlos Dias da Silva Junior  
Instituto Federal de Minas Gerais – Campus Ibirité

Araxá  
09 de fevereiro de 2022.

## RESUMO

O sistema de controle de um pêndulo invertido é uma aplicação amplamente utilizada em estudos da área de Controle de Sistemas, pois envolve diversos elementos da Teoria de Controle. O presente trabalho tem como objetivo desenvolver um protótipo que possa ser utilizado por alunos e professores como exemplificação prática para estudos da disciplina de Controle Multivariável. Como forma de aplicação de controle desse sistema, um modelo dinâmico para representação em espaço de estados foi desenvolvido e foi aplicado a ele o controle de realimentação baseado no método Regulador Linear Quadrático (RLQ). A fim de possibilitar a observação das respostas do sistema ao controle, foram executados testes via simulação computacional e posteriormente, para a comparação com o comportamento real, foram aplicados três testes a um modelo físico construído, de um robô do tipo pêndulo invertido com duas rodas, em que as variáveis a serem controladas são obtidas por meio da leitura de sensores, o processamento da lógica do controle é executado por um microcontrolador, que aciona motores acoplados às rodas em sua base. O sistema apresentou uma resposta adequada durante as simulações e ao primeiro dos três testes utilizando o modelo real, porém, para os outros dois testes, os ganhos do controlador RLQ escolhidos não foram efetivos o suficiente para garantir o controle do sistema. Ao fim foi observado que o modelo real do pêndulo invertido pode ser utilizado como material didático, uma vez que o controle da posição e do ângulo foi efetivo, porém é necessária uma melhor escolha de parâmetros para a obtenção de um ganho de realimentação mais adequado para o controle do sistema quando há uma mudança de referência de estados ou quando sofre perturbações externas.

**Palavras-Chave:** Pêndulo Invertido. Controle Multivariável. Regulador Linear Quadrático.

## ABSTRACT

The control system of an inverted pendulum is an application widely used in studies in the area of Control Systems, as it involves several elements of Control Theory. The present work aims to develop a prototype that can be used by students and professors as a practical example for studies of the discipline of Multivariable Control. As a form of control application of this system, a dynamic model for representation in state space was developed and the feedback control based on the Linear Quadratic Regulator (LQR) method was applied to it. In order to allow the observation of the system responses to the control, tests were performed via computer simulation and later, for comparison with the real behavior, three tests were applied to a built physical model, of an inverted pendulum robot with two wheels, in which the variables to be controlled are obtained by reading sensors, the control logic processing is performed by a microcontroller, which drives motors coupled to the wheels at its base. The system presented an adequate response during the simulations and the first of the three tests using the real model, however, for the other two tests, the chosen LQR controller gains were not effective enough to guarantee the control of the system. In the end, it was observed that the real inverted pendulum model can be used as teaching material, since the position and angle control was effective, but a better choice of parameters is necessary to obtain a more adequate feedback gain for the control of the system when there is a change of state reference or when it suffers external disturbances.

Keywords: Inverted Pendulum. Multivariable Control. Linear Quadratic Regulator.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Dinâmica do pêndulo invertido.....	13
Figura 2 - Diagrama elétrico de um motor de corrente contínua.....	15
Figura 3 - Diagrama de corpo livre de cada roda. ....	18
Figura 4 - Diagrama de corpo livre da haste do pêndulo invertido .....	20
Figura 5 - Diagrama de blocos do controlador RLQ.....	24
Figura 6 - Ligação em paralelo das baterias para a alimentação do microcontrolador. ....	26
Figura 7 - Ligação série-paralelo das baterias de Lítio para alimentação dos motores.....	27
Figura 8 – Diagrama do funcionamento da unidade de medida da MPU 6050.....	28
Figura 9 - Representação no espaço tridimensional do Quatérnio .....	30
Figura 10 - Encoder de efeito Hall com dupla quadratura. ....	31
Figura 11 - Leitura do encoder em ambos os sentidos de rotação.....	32
Figura 12 - Arduino Nano .....	34
Figura 13 - Driver Ponte H L298N.....	35
Figura 14 - Motor JGA25-370.....	36
Figura 15 - Roda do pêndulo invertido.....	37
Figura 16 – Tela do software Fusion 360 com o modelo 3D do chassi do pêndulo invertido. ....	37
Figura 17 - Processo de construção do chassi pela impressora 3D .....	38
Figura 18 – Primeira versão do chassi.....	39
Figura 19 - Segunda versão do chassi .....	39
Figura 20 - Versão final do chassi. ....	40
Figura 21 - Aferição da massa total do sistema. ....	42
Figura 22 - Aferição da massa com o ponto de apoio. ....	42
Figura 23 - Diagrama de corpo livre para a obtenção do centro de massa.....	43
Figura 24 - Experimento para mensurar o período de oscilação da haste do pêndulo. ....	44
Figura 25 – Deslocamento angular durante o experimento para o cálculo do período de oscilação. ....	45
Figura 26 – Comparação entre as leituras dos ângulos com e sem a utilização do processador interno DMP. ....	50
Figura 27 - Resposta simulada ao controle com um valor elevado para a matriz R. ....	52
Figura 28 - Resposta simulada ao controle com um valor elevado para o índice da matriz Q referente ao estado x. ....	53

Figura 29 - Resposta simulada ao controle com um valor elevado para o índice da matriz Q referente ao estado do ângulo. ....	54
Figura 30 - Valores para as matrizes Q e R, determinados de forma empírica. ....	55
Figura 31 - Melhores valores para a matriz Q e R.....	56
Figura 32 – Teste de estabilidade do robô sem perturbação.....	57
Figura 33 - Resposta do robô à perturbação no ângulo. ....	58
Figura 34 - Resposta do robô ao impulso no deslocamento em formato de rampa.....	59

### **LISTA DE TABELAS**

Tabela 1 - Limites máximos da leitura dos sensores da unidade inercial.....	28
Tabela 2 - Limites máximos da leitura dos sensores da unidade inercial.....	29
Tabela 3 – Identificação do sentido de rotação do encoder.....	32
Tabela 4 – Características técnicas do motor JGA25-370.....	36
Tabela 5 - Parâmetros obtidos pelas aferições e cálculos.....	46
Tabela 6 - Apresentação dos dados obtidos pelo Teste 1.....	56

### **LISTA DE QUADROS**

Quadro 1 - Especificações técnicas do Arduino Nano .....	34
--	----

## SUMÁRIO

1. INTRODUÇÃO .....	10
2. REFERENCIAL TEÓRICO .....	12
2.1. Pêndulo invertido.....	12
2.2. Representação de modelo em espaço de estados.....	13
2.3. Análise da controlabilidade do sistema .....	14
2.4. Modelo matemático do motor .....	15
2.5. Modelo matemático das estruturas do pêndulo invertido.....	17
2.5.1. Modelo matemático das rodas .....	18
2.5.2. Modelo matemático do corpo do pêndulo invertido.....	20
2.6. Regulador Linear Quadrático (RLQ).....	23
3. MATERIAIS E MÉTODOS .....	25
3.1. Materiais .....	25
3.1.1. Fontes.....	25
3.1.1.1. Baterias alcalinas.....	25
3.1.1.2. Baterias de íons de lítio .....	26
3.1.1.3. Regulador de tensão .....	26
3.1.2. Sensores .....	27
3.1.2.1. Sensor de medida inercial (IMU).....	27
3.1.2.2. <i>Encoder</i> de dupla quadratura.....	30
3.1.3. Controladores.....	33
3.1.3.1. Microcontrolador Arduino Nano.....	33
3.1.3.2. Driver (ponte H) L298N.....	34
3.1.3.3. Motor CC 6V com caixa de redução.....	35
3.1.4. Estruturas mecânicas.....	36
3.2. Métodos .....	40
3.2.1. Modelagem dos motores .....	40

3.2.2.	Obtenção do centro de massa.....	41
3.2.3.	Obtenção do momento de inércia da haste .....	43
3.2.4.	Obtenção do momento de inércia das rodas .....	45
3.2.5.	Projeto do RLQ.....	46
3.2.6.	Simulações para análise da resposta do sistema .....	48
3.2.7.	Testes para análise das respostas do comportamento real do sistema .....	48
4.	RESULTADOS .....	50
4.1.	Análise dos ângulos do sensor inercial.....	50
4.2.	Análise do controle através de simulações .....	51
4.3.	Análise do comportamento real do sistema.....	55
4.3.1.	Teste 1 – Análise da estabilidade do sistema sem impulsos ou perturbações .....	56
4.3.2.	Teste 2 – Análise da resposta do sistema à perturbação angular .....	57
4.3.3.	Teste 3 – Análise da resposta do sistema à alteração da referência de posição em formato de rampa.....	58
5.	CONCLUSÕES.....	60
	REFERÊNCIAS .....	61
	APÊNDICE A – Código utilizado no Matlab® .....	63
	APÊNDICE B – Código utilizado no arduino para cálculo da entrada do sistema.....	65

## 1. INTRODUÇÃO

O Pêndulo Invertido é uma aplicação recorrente quando se trata de disciplinas que envolvem estudos de dinâmica ou de controle moderno, como em Controle Multivariável. O sistema é amplamente utilizado para testes de algoritmos de controle, como controladores PID (Proporcional, Integral e Derivativo), redes neurais, controle fuzzy, algoritmo genético, etc., como nos informa Priyadarshi (2013). O princípio de funcionamento do pêndulo invertido e os conceitos envolvidos na sua dinâmica são utilizados como base para estudos e aplicações práticas em diversas áreas da engenharia, como em sistemas de proteção de edifícios contra abalos sísmicos; na engenharia aeroespacial, para o controle da estabilidade de foguetes, de forma a mantê-los na posição vertical; em cadeiras de rodas motorizadas; ou em meios de transporte individuais, como os veículos *segway* (PEDROSO; MODESTO, 2017).

Disciplinas do curso de Engenharia de Automação Industrial, do Centro Federal de Educação Tecnológica de Minas Gerais (CEFET – MG), em Araxá, como Sistemas de Controle de Processos Contínuos ou Sistemas de Controle de Processos Discretos, utilizam plantas de controle de variáveis como temperatura, pressão, nível, vazão etc.. Entretanto, ao se tratar da disciplina de Controle Multivariável, há uma carência de plantas para a aplicação prática, sendo as aulas integralmente teóricas. Assim, este trabalho desenvolve um modelo real de Pêndulo Invertido, com o objetivo de possibilitar que alunos e professores possam ter uma ferramenta de aplicação prática dos conteúdos vistos na referida disciplina.

Para atingir o resultado pretendido, são primordiais os seguintes objetivos específicos:

- criar um protótipo de um robô móvel de duas rodas do tipo pêndulo invertido;
- identificar as características físicas do protótipo para determinar o modelo de espaço de estados do sistema;
- Desenvolver o modelo matemático da dinâmica do sistema;
- projetar um controlador no espaço de estados para auxiliar no controle da inclinação e do deslocamento linear do pêndulo invertido;
- realizar simulações computacionais e testes com o modelo real, com o objetivo de validar os modelos matemáticos propostos;
- analisar a utilização do pêndulo invertido na disciplina de controle multivariável.

Para se compreender e analisar sistemas complexos, é necessário observar as relações entre as variáveis do sistema, a fim de se obter o modelo matemático de sua dinâmica, como afirmam Dorf e Bishop (2001). Para tanto, o presente trabalho procura apresentar os principais

conceitos sobre o tema e apresentar o desenvolvimento do modelo de controle do pêndulo em capítulos descritos a seguir.

No capítulo 2, são apresentados os principais conceitos sobre o Pêndulo Invertido, são tratadas as bases teóricas do modelo de espaço de estados, assim como da dinâmica do corpo do pêndulo invertido e, posteriormente, é descrito o modelo matemático dos motores.

Uma vez obtido o modelo de espaço de estados, é desenvolvido o controle para garantir a estabilidade do sistema por meio da análise em espaço de estados e da aplicação do indexador de desempenho, ao utilizar os conceitos do Regulador Linear Quadrático (RLQ, ou LQR do Inglês Linear Quadratic Regulator).

O capítulo 3 apresenta os materiais e componentes que compõem o robô, e esses são classificados em cinco grupos: fonte de alimentação, sensores, controladores, atuadores e estrutura mecânica. Os elementos de cada grupo são apresentados de forma detalhada com o objetivo de explicar as suas características principais e sua função no sistema. Adiante no capítulo, são desenvolvidos os modelos matemáticos do pêndulo invertido, como a modelagem dos motores, em que é definida a forma como serão calculados os coeficientes relacionados ao motor, e o modelo das estruturas mecânicas, onde é apresentado o método para o cálculo do momento de inércia total e a conclusão das matrizes do modelo espaço de estados com seus coeficientes aplicados. Em seguida, é apresentada a forma pelo qual foi projetado e aplicado o controlador RLQ ao sistema e, por fim, são exibidos os métodos e rotinas utilizados para executar as simulações e testes com o modelo real.

No capítulo 4 são disponibilizados os resultados obtidos durante o desenvolvimento do trabalho. Primeiramente, é realizado um comparativo entre dois métodos de aquisição de medida angular, de forma a comprovar a efetividade de um deles. Em seguida, são apresentados os dados da resposta simulada do sistema, durante a execução das rotinas de teste. E por fim, são apresentados os resultados obtidos pelos testes utilizando o modelo físico.

Por fim, o capítulo 5 trata das conclusões obtidas ao se observar os resultados das simulações e testes, e analisa a possibilidade da utilização do robô como material didático e elenca outras aplicações para continuação deste trabalho.

## 2. REFERENCIAL TEÓRICO

Neste capítulo são detalhados conceitos sobre o funcionamento do pêndulo invertido, são apresentados princípios da representação em espaço de estados, assim como as suas propriedades de controlabilidade. A seguir são desenvolvidos os modelos matemáticos dos motores, das rodas e do corpo do pêndulo invertido. Por fim, os conceitos do método de controle ótimo RLQ são apresentados.

### 2.1. Pêndulo invertido

O problema pêndulo invertido, de forma simplificada, consiste no controle do ângulo de inclinação de uma haste fixada à uma base móvel. O sistema deve ser capaz de se manter em equilíbrio de forma autônoma, ou seja, deve ser capaz, por meio do controle do deslocamento e velocidade da base móvel, de manter a haste na posição vertical. Para efetuar-se o controle de sistemas como esse, é necessário analisar as relações entre suas variáveis para que se possa obter a representação do modelo matemático e os métodos de solução desse sistema, como afirma Gadelha (2018).

Segundo Ogata (2010), pelo motivo de o centro de gravidade do mecanismo do pêndulo invertido estar situado acima de seu ponto de articulação, a sua posição vertical é instável, ou seja, a haste pode girar para uma das direções, a não ser que seja aplicada uma força de intensidade e sentido correta na base móvel, suficiente para mantê-la em equilíbrio.

No modelo desenvolvido neste trabalho, a tensão aplicada aos motores resulta nos controles dos torques e das velocidades de rotação. A velocidade e o sentido de giro do motor devem ser suficientes para a correção do ângulo da haste, com o objetivo de deixá-la sempre equilibrada em posição vertical, ou seja, quando há um aumento indesejado do ângulo de inclinação, para esquerda ou para a direita, os motores serão acionados em um sentido e velocidade de rotação suficientes para reduzir o esse ângulo. Sempre que houver uma ação de correção, a haste e a base do robô tendem a se deslocar em sentidos opostos. A dinâmica do sistema está apresentada na Figura 1.

Figura 1 - Dinâmica do pêndulo invertido.



Fonte: Autoria própria.

## 2.2. Representação de modelo em espaço de estados

A representação de modelo em espaço de estados faz parte da teoria de controle moderno, que tem por finalidade a abordagem de sistemas lineares e não-lineares com múltiplas entradas e saídas, além de descrever o estado do sistema no domínio do tempo. Essa abordagem, de acordo com Ogata (2010), pode facilitar a análise em diversos casos, como a questão desenvolvida neste trabalho.

Segundo Nise (2002), ao se tratar de modelamento em espaço de estados, é necessário definir seis elementos importantes, que são: variável de sistema, variáveis de estado, vetor de estados, espaço de estados, equações de estados e equação de saída. Variável de sistema é qualquer variável que responda a uma entrada ou a condições iniciais em um sistema. Variáveis de estado são definidas como a menor quantidade de funções  $x(t)$  que seja capaz de determinar o estado do sistema, sendo que os estados são variáveis calculadas no tempo em que o conhecimento de  $x$  em  $t_{(0)}$ ,  $(x_{t0})$ , juntamente com a entrada  $u_{(t)}$  para  $t \geq t_0$ , determina o comportamento do sistema para  $t \geq t_0$ ,  $(x_t)$ . Vetor de estados é o vetor composto pelas variáveis de estado e determina a dimensão desse sistema. Já o espaço de estados é o espaço n-

dimensional onde os eixos são as variáveis de estado. Por fim, equação de saída é uma equação que expressa as variáveis de saída como combinações das variáveis de estado e das entradas.

A análise no modelo de espaço de estados envolve três tipos de variáveis que estão presentes na modelagem de sistemas dinâmicos: variáveis de entrada, variáveis de saída e variáveis de estado. Essas são representadas respectivamente por  $u(t)$ ,  $y(t)$  e  $x(t)$ , como informa Vieira (2017).

O equacionamento de um modelo de sistema em espaço de estados é dado por:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (1)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (2)$$

Onde:  $A(t)$  é a matriz de estados;  $B(t)$ , a matriz de entrada;  $C(t)$ , a matriz de saída; e  $D(t)$  é a matriz de transmissão direta. Se essas matrizes variarem com o tempo, o sistema é considerado variante no tempo, do contrário, o sistema é considerado invariante no tempo. A representação em espaço de estados de um sistema invariante no tempo é apresentada a seguir, onde a equação (2) representa a equação de estado de um sistema invariante no tempo e a equação de saída para o mesmo sistema (OGATA 2010).

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3)$$

$$y(t) = Cx(t) + Du(t) \quad (4)$$

### 2.3. Análise da controlabilidade do sistema

Segundo Nise (2002), um sistema é dito controlável se for possível obter uma entrada capaz de transferir todas as variáveis de estado de um sistema de um valor inicial desejado para um estado final desejado, em caso contrário, o sistema é não controlável.

Considerando a equação no modelo espaço de estados representada pela equação (2), as matrizes  $A \in R^{n \times n}$  e  $B \in R^{n \times n_u}$  são controláveis se, para todo estado inicial  $x_{(t_0)} = x_0$  e todo estado final  $x_1 = x_{1(t_0)}$ , existir uma entrada que leve  $x_0$  a  $x_1$  em  $t \in [t_0, t_1]$ .

A matriz de controlabilidade  $\xi \in R^{n \times n_u}$  é definida por:

$$\xi = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (5)$$

O critério para que o sistema seja considerado controlável é que a matriz de controlabilidade deve possuir posto de linha pleno, ou seja, os vetores  $B \ AB \ A^2B \ \dots \ A^{n-1}B$  devem ser linearmente independentes.

Segundo Ogata 2010, a análise da controlabilidade do problema é importante pois, a sua solução pode não existir, se o sistema considerado é não controlável.

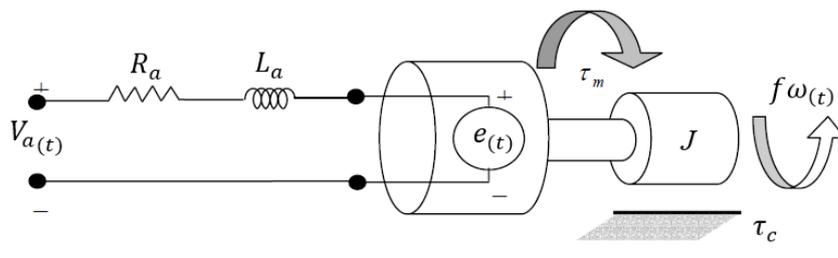
#### 2.4. Modelo matemático do motor

O eixo pivô do robô fica localizado na extremidade inferior do seu corpo e é composto por um par de rodas fixadas aos motores que, por sua vez, estão fixados na base da estrutura. Os dois motores em questão são os atuadores do sistema e são utilizados com o intuito de gerar a ação de controle, buscando, o equilíbrio do sistema.

Para desenvolver o controle do pêndulo, é necessário obter a modelagem matemática completa do seu corpo e, para isso, o modelo dos motores é fundamental. O modelo que será apresentado a seguir foi feito baseado em Ooi (2003) que apesar de ter desenvolvido um pêndulo utilizando forma, peças e equipamentos diferentes, apresenta semelhanças ao se tratar do desenvolvimento dos modelos matemáticos. Segundo o autor, o modelo a seguir, descreve o motor DC no espaço de estados e será usado para fornecer uma relação entre a entrada de tensão dos motores e o controle de torque necessário para controlar o pêndulo.

A figura 2 representa o circuito simplificado do motor DC.

Figura 2 - Diagrama elétrico de um motor de corrente contínua.



Fonte: Adaptado de Rich Chi Ooi (2003).

Onde:

$R_a$  e  $L_a$  são a resistência e indutância da armadura

$i_{a(t)}$  é a corrente na armadura

$V_{a(t)}$  é a tensão aplicada na armadura do motor

$e_{(t)}$  é a força contra eletromotriz induzida no motor

$\omega_{(t)}$  é a velocidade angular do rotor

$f$  é o coeficiente de atrito viscoso

$J$  é o momento de inércia da carga acoplada ao motor

A relação entre o torque  $\tau_m$  e a corrente de armadura  $i_{a(t)}$  é apresentada na equação (6), onde  $k_m$  é a constante de torque do motor:

$$\tau_m = k_m \cdot i_{a(t)} \quad (6)$$

A relação entre a força contra eletromotriz  $e_{(t)}$  e a velocidade angular do rotor  $\omega_{(t)}$  é representada na equação (7), onde  $k_e$  é a constante da força contra eletromotriz:

$$e_{(t)} = k_e \cdot \omega_{(t)} \quad (7)$$

Dessa forma, utilizando a lei de Kirchoff, é possível apresentar a somatória das tensões presentes no circuito da figura 2, através da equação (8):

$$\sum \text{tensões} = V_{a(t)} - R_a \cdot i_{a(t)} - L_a \frac{di_{a(t)}}{dt} - e_{(t)} = 0 \quad (8)$$

E utilizando a equação de movimento rotacional aplicado ao motor, representada pela equação (9), onde  $f\omega_{(t)}$  representa o atrito viscoso, e  $\tau_c$  representa o torque externo de perturbação, obtém-se o torque resultante  $J\dot{\omega}_{(t)}$ .

$$\sum \text{torques} = \tau_m - \tau_c - f\omega_{(t)} = J\dot{\omega}_{(t)} \quad (9)$$

Substituindo as equações (6) e (7) nas equações (8) e (9), considerando que para modelos reais  $L_a \approx 0$  e  $f \approx 0$ , e rearranjando os termos, obtém-se as equações (10) e (11):

$$i_{a(t)} = -\frac{k_e}{R_a} \omega(t) + \frac{V_{a(t)}}{R_a} \quad (10)$$

$$\frac{d\omega(t)}{dt} = \frac{k_m}{J} i_{a(t)} - \frac{\tau_c}{J} \quad (11)$$

Substituindo a equação (10) na equação (11):

$$\frac{d\omega(t)}{dt} = \frac{-k_m k_e}{JR_a} \omega(t) - \frac{k_m}{JR_a} V_{a(t)} - \frac{\tau_c}{J} \quad (12)$$

Com base na equação (12), é possível elaborar o modelo de espaço de estados da dinâmica do motor, apresentado pela equação (13):

$$\begin{aligned} \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{JR_a} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{JR_a} & \frac{1}{J} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_c \end{bmatrix} \\ y &= [1 \quad 0] \begin{bmatrix} \theta \\ \omega \end{bmatrix} + [0 \quad 0] \begin{bmatrix} V_a \\ \tau_c \end{bmatrix} \end{aligned} \quad (13)$$

Também é possível expressar o torque do motor em função da tensão aplicada e dos parâmetros do motor, substituindo a equação (10) na equação (6):

$$\tau_m = -\frac{k_m}{R_a} V_{a(t)} - \frac{k_e k_m}{R_a} \omega(t) \quad (14)$$

## 2.5. Modelo matemático das estruturas do pêndulo invertido

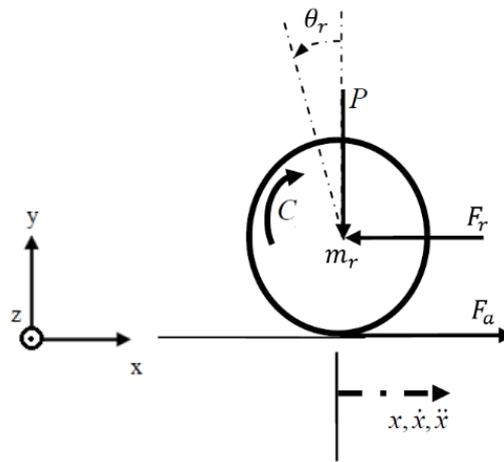
Para elaborar o modelo matemático das estruturas do pêndulo invertido, foi necessário dividi-las em duas partes: o modelo das rodas e o modelo da haste. As rodas são coaxiais e são acopladas cada uma a um motor, o que gera o seu movimento de rotação. A haste é simétrica com relação ao lado esquerdo e direito, de forma que para se obter o modelo final é necessário espelhar o modelo. Assim como a modelagem dos motores, o modelo matemático das estruturas

do pêndulo foi baseado em Ooi (2003), apesar de existirem diversas diferenciações na montagem, forma, escolha de componentes e ligações.

### 2.5.1. Modelo matemático das rodas

Para a elaboração do modelo das rodas, será tomado como ponto de partida a figura 3.

Figura 3 - Diagrama de corpo livre de cada roda.



Fonte: Adaptado de Rich Chi Ooi (2003).

Onde:

$\ddot{x}$  é a aceleração no eixo x;

$\theta_r$  é o ângulo de rotação da roda;

$m_r$  é a massa da roda;

$F_r$  é a força de reação entre a roda e a haste;

$F_a$  é a força de atrito.

Utilizando a segunda lei de Newton, aplicada ao movimento linear rotatório, obtém-se o resultado:

$$m_r \ddot{x} = F_a - F_r \quad (15)$$

$$J_r \ddot{\theta}_r = C_m - F_a r \quad (16)$$

Onde:

$P$  é a força normal da roda;

$J_r$  é o momento de inércia da roda;

$\ddot{\theta}_r$  é a aceleração angular da roda;

$C$  é o torque que o motor aplica à cada roda;

$r$  é o raio da roda.

Ao isolar a força de atrito  $F_a$  na equação (16) e substituindo-a na equação (15), obtém-se:

$$m_r \ddot{x} = \frac{C_r}{r} - \frac{J_r \ddot{\theta}_r}{r} - F_r \quad (17)$$

Considerando o torque da roda  $C_r$  igual ao torque do motor, resulta em:

$$m_r \ddot{x} = \frac{k_m}{r R_a} V_a - \frac{k_e k_m}{r R_a} \dot{\theta}_r - \frac{J_w}{r} \ddot{\theta}_r - F_r \quad (18)$$

Visto que o movimento linear tem ação no centro da roda, o movimento de rotação pode ser transformado em movimento linear por alterações em variáveis, como apresentado a seguir:

$$\ddot{\theta}_r r = \ddot{x} \rightarrow \ddot{\theta}_r = \frac{\ddot{x}}{r} \quad (19)$$

$$\dot{\theta}_r r = \dot{x} \rightarrow \dot{\theta}_r = \frac{\dot{x}}{r} \quad (20)$$

Dessa forma, a equação (18) pode ser reescrita como:

$$m_r \ddot{x} = \frac{k_m}{r R_a} V_a - \frac{k_e k_m}{r^2 R_a} \dot{x} - \frac{J_w}{r^2} \ddot{x} - F_r \quad (21)$$

Visto que o modelo da equação 21 é referente a uma única roda, e que as duas rodas são simétricas, o modelo para a segunda roda é idêntico ao apresentado. Sendo assim, o modelo resultante é a soma dos modelos das duas rodas:

$$2(m_r + \frac{J_r}{r^2})\ddot{x} = -2 \cdot \frac{k_e k_m}{r^2 R_a} \dot{x} + 2 \cdot \frac{k_m}{r R_a} V_a - (F_{r_d} + F_{r_e}) \quad (22)$$

Sendo:

$F_{r_d}$  a força de reação entre a roda direita e a haste

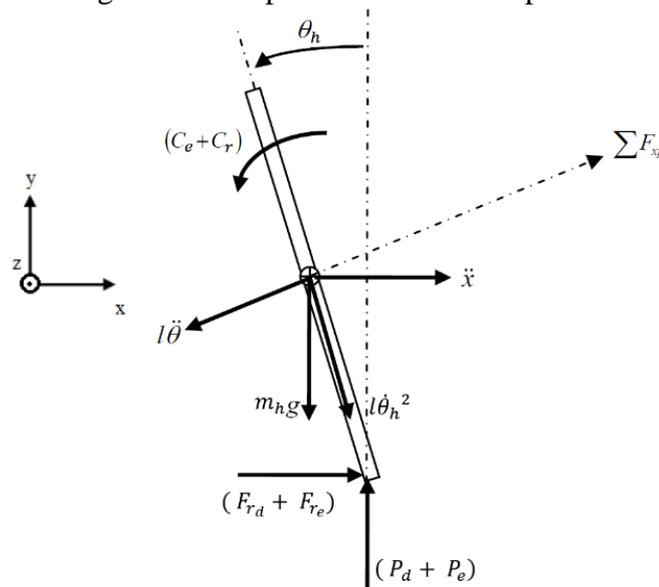
$F_{r_e}$  a força de reação entre a roda esquerda e a haste

A seguir será apresentado o modelo da haste do pêndulo invertido.

### 2.5.2. Modelo matemático do corpo do pêndulo invertido

A figura 4 ilustra o diagrama de corpo livre da haste do pêndulo:

Figura 4 - Diagrama de corpo livre da haste do pêndulo invertido



Fonte: Adaptado de Rich Chi Ooi (2003).

Onde:

$C_e$  é o torque que o motor aplica à roda da esquerda

$C_r$  é o torque que o motor aplica à roda da direita

$J_h$  é o momento de inércia da haste

$m_h$  é a massa da haste

$\theta_h$  é o ângulo de rotação da haste

$g$  é a aceleração da gravidade

$F_{xp}$  a força perpendicular à haste do pêndulo

$F_{r_d}$  a força de reação entre a roda direita e a haste

$F_{r_e}$  a força de reação entre a roda esquerda e a haste

$P_d$  a força normal referente a roda direita e a haste

$P_e$  a força normal referente a roda esquerda e a haste

$l$  é a distância do eixo de rotação até o centro de massa

A soma das forças na direção horizontal é descrita por meio da segunda lei de movimento de Newton:

$$\sum F_x = m_h \ddot{x} \therefore m_h \ddot{x} = m_h l \dot{\theta}_h^2 \text{sen} \theta_h - m_h l \ddot{\theta}_h \text{cos} \theta_h + (F_{r_d} + F_{r_e}) \quad (23)$$

Dessa forma:

$$(F_{r_d} + F_{r_e}) = -m_h l \dot{\theta}_h^2 \text{sen} \theta_h + m_h l \ddot{\theta}_h \text{cos} \theta_h + m_h \ddot{x} \quad (24)$$

A soma das forças perpendiculares à haste do pêndulo,  $F_{xp}$  é:

$$\sum F_{xp} = m_h \ddot{x} \text{cos} \theta_h = -m_h l \ddot{\theta}_h - m_h g \text{sen} \theta_h + (P_d + P_e) \text{sen} \theta_h + (F_{r_d} + F_{r_e}) \quad (25)$$

A soma dos momentos  $M_0$  ao redor do centro de massa do pêndulo é:

$$\sum M_0 = J_h \ddot{\theta}_h = -(C_d + C_e) - (P_d + P_e) l \text{sen} \theta_h - (F_{r_d} + F_{r_e}) l \text{cos} \theta_h \quad (26)$$

Assim como feito no modelo matemático das rodas, pode-se considerar o torque que o motor aplica nas rodas como sendo o próprio valor de torque resultante do motor, ou seja:  $C_d + C_e = 2\tau_m$ . Ao realizar a substituição desses termos na equação (26), utilizando a equação (14), e rearranjando alguns componentes da equação, resulta em:

$$J_h \ddot{\theta}_h + \frac{2k_m}{R_a} V_{a(t)} - \frac{2k_e k_m \dot{\chi}}{R_a r} = -(F_{r_d} + F_{r_e}) l \text{cos} \theta_h - (P_d + P_e) l \text{sen} \theta_h \quad (27)$$

Em sequência, deve-se unir as equações de momento com as de força, sendo necessário multiplicar a equação (25) por  $-l$ , o que resulta em:

$$-m_h \ddot{x} l \cos \theta_h = m_h l^2 \ddot{\theta}_h + m_h g l \sin \theta_h - (F_{r_d} + F_{r_e}) l \cos \theta_h - (P_d + P_e) l \sin \theta_h \quad (28)$$

A multiplicação anterior foi necessária para que a substituição da equação (27) na equação (28) seja possível, obtendo assim a equação (29):

$$-m_h \ddot{x} l \cos \theta_h = m_h l^2 \ddot{\theta}_h + m_h g l \sin \theta_h + J_h \ddot{\theta}_h + \frac{2k_m}{R_a} V_{a(t)} - \frac{2k_e k_m \dot{x}}{R_a r} \quad (29)$$

E em sequência, a expressão  $(F_{r_d} + F_{r_e})$  é eliminada da dinâmica dos motores substituindo a equação (24) na equação (22), o que resulta em:

$$2(m_r + \frac{J_r}{r^2}) \ddot{x} = 2 \frac{k_m}{r R_a} V_a - 2 \frac{k_e k_m}{r^2 R_a} \dot{x} + m_h l \dot{\theta}_h^2 \sin \theta_h - m_h l \ddot{\theta}_h \cos \theta_h - m_h \ddot{x} \quad (30)$$

As equações (29) e (30) não são lineares, porém podem ser linearizadas para valores de ângulos pequenos, próximos a  $\theta_h = \pi + \theta$ , onde  $\theta \approx 0$  a partir da direção vertical.

Dessa forma, é possível aplicar as seguintes relações de transformação de linearização a fim de simplificar o modelo:

$$\cos \theta_h = 1, \quad \sin \theta_h = -\theta, \quad e \quad \dot{\theta}_h^2 = 0. \quad (31)$$

Aplicando a linearização nas equações (29) e (30), e reorganizando seus termos, obtém-se as equações que descrevem o movimento do pêndulo invertido:

$$m_h \ddot{x} l = (J_h + m_h l^2) \ddot{\theta} - \frac{2k_e k_m \dot{x}}{R_a r} - m_h g l \theta + \frac{2k_m}{R_a} V_{a(t)} \quad (32)$$

$$2 \cdot \frac{k_m}{r R_a} V_a = \left( 2m_r + m_h + \frac{2J_h}{r^2} \right) \ddot{x} + 2 \cdot \frac{k_e k_m}{r^2 R_a} \dot{x} - m_h l \ddot{\theta} \quad (33)$$

E com isso, é possível elaborar as suas matrizes em espaço de estados:

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m_h^2 g l^2}{\alpha} & \frac{2k_m k_e (m_h l r - J_h - m_h l^2)}{R r^2 \alpha} & 0 \\ 0 & \frac{m_h g l \beta}{\alpha} & \frac{2k_m k_e (r \beta - m_h l)}{R r^2 \alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2k_m (J_h + m_h l^2 - m_h l r)}{R r \alpha} \\ \frac{2k_m (m_h l - r \beta)}{R r \alpha} \end{bmatrix} V a \quad (34)$$

Onde,

$$\beta = \left(2m_r + \frac{2J_r}{r^2} + m_h\right) \quad \alpha = \left[J_h \beta + 2m_h l^2 \left(m_r + \frac{J_r}{r^2}\right)\right]$$

Observe que o modelo dado pela equação (34) possui ordem 4 e a equação de saída associada a ele é dada por:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} \quad (35)$$

Dessa forma, pelo fato de o sistema apresentar duas saídas, é considerado como um sistema de controle multivariável

## 2.6. Regulador Linear Quadrático (RLQ)

Métodos baseados na teoria de controle ótimo, como o Regulador Linear Quadrático (RLQ, ou LQR do inglês *Linear Quadratic Regulator*), possibilitam estabilizar o sistema com menor esforço de controle possível, obtendo assim boas margens de ganho e de fase (TABORDA, 2014). As definições desse controle serão apresentadas a seguir.

Dado o sistema descrito pelas equações (2) e (3), uma lei de controle capaz de estabilizá-lo é a que segue:

$$u_{(t)} = -Kx_{(t)} \quad (36)$$

No controlador RLQ, essa lei é utilizada para minimizar o índice de desempenho  $J$ :

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (37)$$

Onde  $Q$  e  $R$  são matrizes hermitianas definidas positivas ( $Q$  pode ser definida ou semidefinida positiva) ou real simétrica. O índice  $J$  mede o desempenho transitório do sistema

$x^T Q x$  e o custo do controle necessário  $u^T R u$ . Segundo Fagundes 2015, a matriz  $Q$  é uma matriz diagonal principal e cada componente da sua diagonal representa a penalização de cada estado do sistema. Já  $R$  é uma matriz quadrada de dimensão idêntica ao número de saídas do controlador, normalmente é escalar, e penaliza a saída do controlador.

As matrizes  $Q$  e  $R$  podem ser selecionadas pelos projetistas de forma a apresentar uma importância voltada ao desempenho ou ao custo do controle, assim como apresentado por Faria (2015): ao aumentar o valor da matriz  $Q$ , que representa o peso para as variáveis de estado do sistema, o resultado é um rastreamento mais preciso, enquanto ao se elevar os valores da matriz  $R$ , que representa o peso para as variáveis de entrada, o resultado será uma redução na ação de controle, ou seja, será gasto menos energia. Dessa forma, cabe ao projetista ponderar a importância de cada peso para a resposta desejada em sua aplicação específica.

Para a aplicação da lei de controle definida pela equação (36), é necessário verificar se o sistema é controlável, baseando-se nas matrizes das equações de estados do sistema. Segundo Chen (1993), o ganho  $K$  é esperado de tal forma que os autovalores de  $(A - BK)$  devem estar alocados no semi-plano esquerdo do plano- $s$ , o que torna o sistema estável para qualquer condição inicial  $x_{(0)}$ , ou seja, a resposta dos estados do sistema, quando  $t \rightarrow \infty$ , tenderá a zero.

Dito isso, define-se o objetivo do controle via RLQ, o de determinar a matriz de ganho  $K$ , chamada matriz ótima, expressa por:

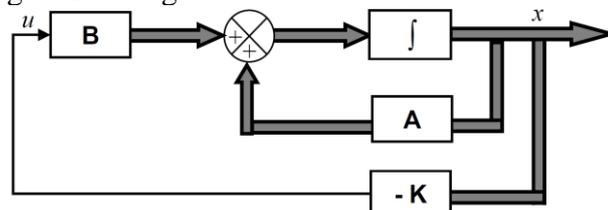
$$K = R^{-1} B^T S \quad (38)$$

Segundo Ogata (2010),  $S$  pode ser encontrado como a solução da equação algébrica matricial de Riccati, dada em sua forma reduzida pela equação (39):

$$SA + A^T S - SBR^{-1}B^T S = 0 \quad (39)$$

O diagrama em blocos para o controlador RLQ é representado a seguir:

Figura 5 - Diagrama de blocos do controlador RLQ



Fonte: Adaptado de Rich Chi Ooi (2003).

### 3. MATERIAIS E MÉTODOS

Neste capítulo serão apresentados os materiais e componentes utilizados para a construção do robô, assim como os métodos necessários para a execução dos testes, além de demonstrar as formas como esses foram executados.

#### 3.1. Materiais

Os componentes presentes no pêndulo invertido podem ser classificados em cinco grupos: fontes, sensores, controladores, atuadores e estruturas mecânicas. Adiante neste capítulo, cada um deles será mais bem detalhado.

##### 3.1.1. Fontes

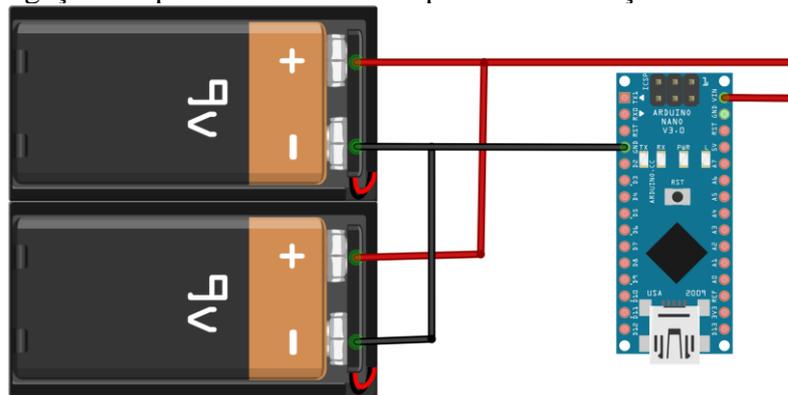
Para a alimentação do sistema, foram utilizadas as associações entre dois tipos distintos de baterias: duas baterias alcalinas ligadas em paralelo, para a alimentação do microcontrolador; e quatro baterias recarregáveis de íons de lítio (Li-ion) em uma associação série-paralelo, conectadas a um regulador de tensão para alimentação dos motores através da ponte-H. Esses componentes fazem parte do grupo das fontes.

Devido a massa das baterias ser superior aos demais componentes presentes na haste do chassi, as baterias foram posicionadas na extremidade superior do corpo do pêndulo, deslocando assim, o centro de gravidade do pêndulo para cima, o que facilita o controle do sistema.

##### 3.1.1.1. Baterias alcalinas

As fontes de alimentação escolhidas foram duas baterias alcalinas de 9V, ligadas em paralelo como a representada na figura 6. Esse tipo de bateria foi escolhido por fornecer tensão suficiente para alimentar os circuitos do microcontrolador e por apresentar um preço de mercado menor que outras opções disponíveis, como baterias de LiPo (íon de polímero). Uma desvantagem das baterias alcalinas, é que elas não são recarregáveis, ou seja, deverão ser substituídas sempre que perderem a carga. Por esse motivo, foram utilizadas duas baterias ligadas em paralelo, com o intuito de elevar a duração da capacidade de carga.

Figura 6 - Ligação em paralelo das baterias para a alimentação do microcontrolador.



Fonte: Autoria própria utilizando o *software* Fritzing.

### 3.1.1.2. Baterias de íons de lítio

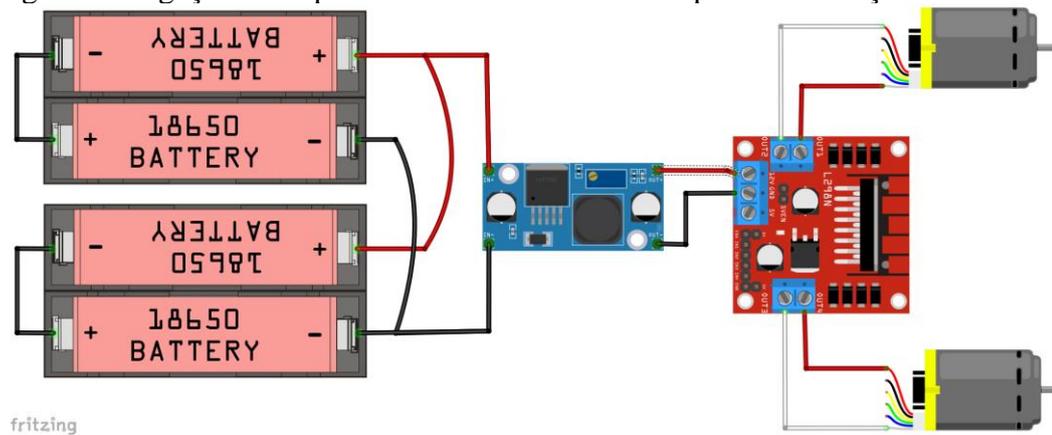
Visto que a capacidade de fornecimento de corrente das baterias alcalinas não se mostrou suficiente para a demanda dos motores, e por essas não terem a possibilidade de recarga, foram utilizadas 4 baterias Li-íon de 3800mA e 3,7 V cada, também comumente chamadas de 18650, conectadas em uma associação série-paralelo; de forma a obter uma tensão total de 7,4 V, para a alimentação dos motores. A ligação dessas baterias é apresentada na figura 7.

### 3.1.1.3. Regulador de tensão

Para o desenvolvimento do modelo matemático do sistema do pêndulo invertido, é necessário fornecer uma tensão de alimentação máxima dos motores, e essa foi definida como 6V por também ser a tensão de alimentação aconselhada pelo fabricante.

Dessa forma, um regulador de tensão ajustável LM2596, que apresenta o alcance de regulação de tensão de entrada de 1,2 V a 37 V, foi utilizado de forma a regular a tensão fornecida pelas baterias de Lítio, para o valor desejado fixo de 6V, de forma a alimentar os motores por meio da ponte H. As conexões entre as baterias e o regulador também são apresentadas pela figura 7.

Figura 7 - Ligação série-paralelo das baterias de Lítio para alimentação dos motores.



Fonte: Autoria própria utilizando o *software* Fritzing.

### 3.1.2. Sensores

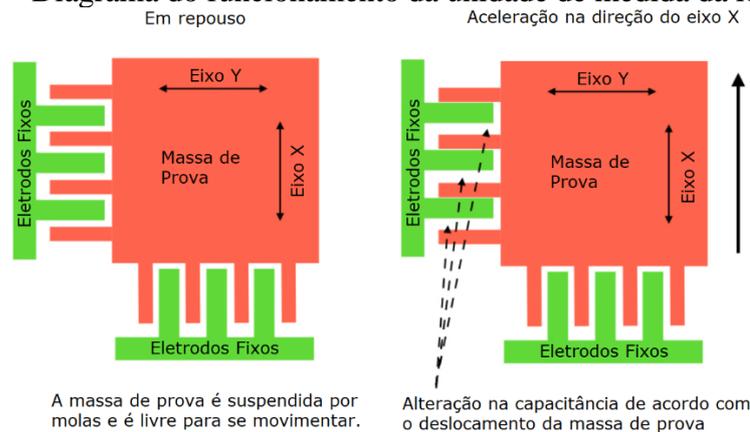
Os circuitos eletrônicos responsáveis pela aquisição dos dados das variáveis de controle, são itens que fazem parte do grupo dos sensores, são: o sensor de medida inercial e o *encoder* de dupla quadratura. O primeiro, chamado de Unidade Inercial de Medição ou IMU (do inglês *Inertial Measurement Unit*), tem a função de executar a leitura da posição angular do corpo do robô, enquanto o segundo é responsável pela aquisição dos valores da posição linear instantânea, que descreve o deslocamento do sistema, devido ao giro dos motores.

#### 3.1.2.1. Sensor de medida inercial (IMU)

O Sensor de Medida Inercial (IMU, do inglês: *Inertial Measurement Unit*) é um dispositivo que utiliza uma combinação de sensores como giroscópio e acelerômetro para realizar medições de variáveis polares de posicionamento espacial, como deslocamento, velocidade e aceleração. O modelo de IMU utilizado foi o MPU 6050 e ele se comunica com o Arduino via protocolo I2C, utilizando níveis de tensão entre 3.3 e 5V.

A MPU 6050 realiza a leitura da aceleração do corpo e da velocidade angular em três eixos, sendo que a aceleração é obtida pelo acelerômetro e a velocidade pelo giroscópio, totalizando assim, seis graus de liberdade. Um diagrama simplificado da unidade de medida interna do MPU é apresentado na figura 8.

Figura 8 – Diagrama do funcionamento da unidade de medida da MPU 6050



Fonte: Adaptado de: Martin White (2019).

Na figura 8, a massa de prova é suspensa por molas de forma a ficar livre para se mover enquanto o dispositivo sofre aceleração. Os eletrodos fixos, por sua vez, estabelecem um efeito capacitivo entre si e a massa de prova. A variação da capacitância registrada, ocasionada pela movimentação do pêndulo, é convertida por conversor A/D para valores digitais entre 0 e 32750. O giroscópio funciona de maneira semelhante, porém pelo efeito Coriolis e não pela variação da aceleração.

Os dados obtidos pelo acelerômetro e pelo giroscópio, devem ser normalizados de forma a apresentar uma sensibilidade condizente com a aplicação. Para isso, as unidades de medida da MPU apresentam divisões para representar medidas significativas. Elas alocam suas unidades de medida criando quatro níveis de sensibilidade, em que a escolha do nível da sensibilidade depende da aplicação de cada projeto. As tabelas 1 e 2 apresentam os valores de limite máximo (fundo de escala) para a aceleração e velocidade angular alocados na unidade.

Tabela 1 - Limites máximos da leitura dos sensores da unidade inercial

Fundo de escala: Aceleração	Sensibilidade
2g	16,384 LSB/g
4g	8,192 LSB/g
8g	4,096 LSB/g
16g	2,048 LSB/g

Fonte: Folha de dados da MPU 6050.

Tabela 2 - Limites máximos da leitura dos sensores da unidade inercial

Fundo de escala: Velocidade	Sensibilidade
Angular	
$\pm 250^\circ/\text{s}$	131 LSB/ $(^\circ/\text{s})$
$\pm 500^\circ/\text{s}$	65,5 LSB/ $(^\circ/\text{s})$
$\pm 1000^\circ/\text{s}$	32,8 LSB/ $(^\circ/\text{s})$
$\pm 2000^\circ/\text{s}$	16,4 LSB/ $(^\circ/\text{s})$

Fonte: Folha de dados da MPU 6050.

Neste trabalho os valores definidos para o fundo de escala foram: 2g para a aceleração, e  $\pm 250^\circ/\text{s}$  para a velocidade angular.

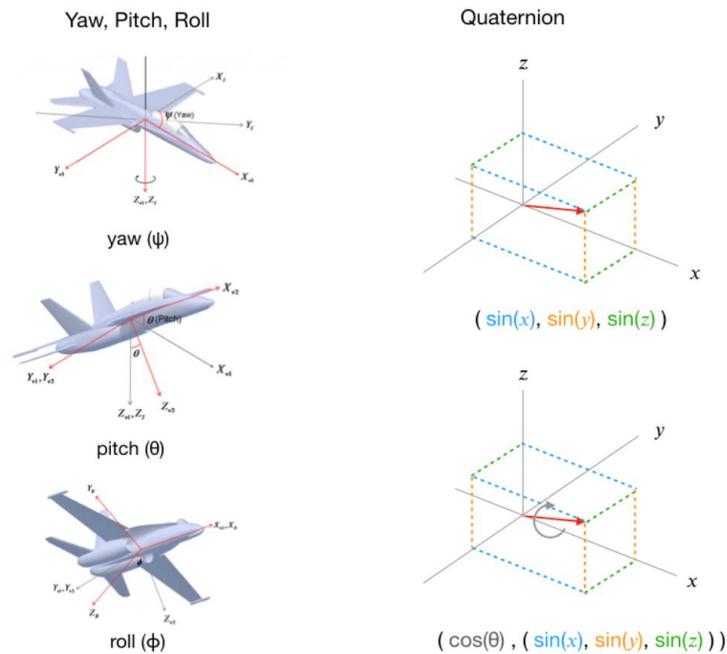
Os dados obtidos pelo acelerômetro e pelo giroscópio, quando relacionados a obtenção de valores dos ângulos, e lidos de forma independente, apresentam erros de medida significativos o suficiente, para impossibilitar a sua utilização devido a imprecisão de medida. Segundo Paula (2015), o erro do acelerômetro é relacionado a um ruído muito alto, porém com desvio nulo; enquanto o giroscópio apresenta uma leitura pouco ruidosa, porém com grande desvio. A solução para esta questão é realizar a fusão das duas leituras, com o objetivo de formar uma leitura mais precisa.

Existem diversos métodos de fusão desses dados, como o filtro de kalman ou o filtro complementar por exemplo, porém a própria MPU possui uma unidade de processamento interna que executa a fusão desses dados, denominada Processador de Movimento Digital (ou DMP, do inglês: *Digital Motion Processor*). Em testes prévios, o DMP apresentou resultados mais próximos do ângulo real do que os métodos citados acima, desta forma utilizou-se o DMP.

Essa unidade mantém um *buffer* interno que combina os dados do giroscópio e acelerômetro e calcula os valores dos ângulos instantâneos. Esses valores são apresentados em forma de vetor que representa um quatérnio, que por sua vez é uma forma de representar a orientação de um objeto no espaço tridimensional.

Os quatérnios podem ser representados por um vetor de quatro posições, onde as três últimas posições representam o ângulo de deslocamento do vetor, em relação respectivamente, aos eixos: x, y e z; enquanto a primeira posição representa um ângulo de rotação do objeto em torno do eixo com mesma direção em que o vetor aponta, como representado pela figura 9:

Figura 9 - Representação no espaço tridimensional do Quatérnio



Fonte: Adaptado de: Martin White (2019).

Onde:

$\psi$  ou *yaw* é o ângulo entre o vetor e o eixo x

$\theta$  ou *pitch* é o ângulo entre o vetor e o eixo y

$\phi$  ou *roll* é o ângulo entre o vetor e o eixo z

Dos ângulos obtidos pelo quatérnio, o único que há necessidade de ser utilizado neste trabalho é o ângulo  $\theta$  ou *pitch*, por ser o único grau de liberdade relacionado a variação da posição angular descrita pelo pêndulo.

### 3.1.2.2. Encoder de dupla quadratura

Para obter os valores de deslocamento linear do pêndulo, é necessário utilizar um sensor que meça a distância percorrida. Existem diversas formas para obtenção desses dados, sendo que a forma mais comum encontrada durante as pesquisas foi a que utiliza a contagem de um par de pulsos em quadratura defasados entre si em 90 graus. A forma de obtenção dos pulsos pode variar a cada modelo de sensor, mas o princípio de funcionamento é o mesmo: sensibilizar um elemento sensor, seja ele um receptor óptico ou um sensor de efeito hall, à medida que o eixo rotaciona.

O sensor utilizado neste trabalho é um *encoder* baseado no princípio de funcionamento do efeito hall, o qual é um transdutor que gera uma diferença de potencial a partir de uma variação no campo magnético (figura 10).

Figura 10 - Encoder de efeito Hall com dupla quadratura.



Fonte: Autoria própria.

Esse modelo de *encoder* possui dois itens principais: um disco fixado ao eixo do motor, e que por isso acompanha seu giro; e uma placa eletrônica que possui dois sensores magnéticos que captam uma variação no campo magnético sempre que um dos ímãs do disco passa por eles. Esses dois sensores são posicionados de forma a garantir uma diferença de fase de 90° entre eles, podendo assim adquirir tanto a contagem dos pulsos, quanto o sentido de rotação do eixo.

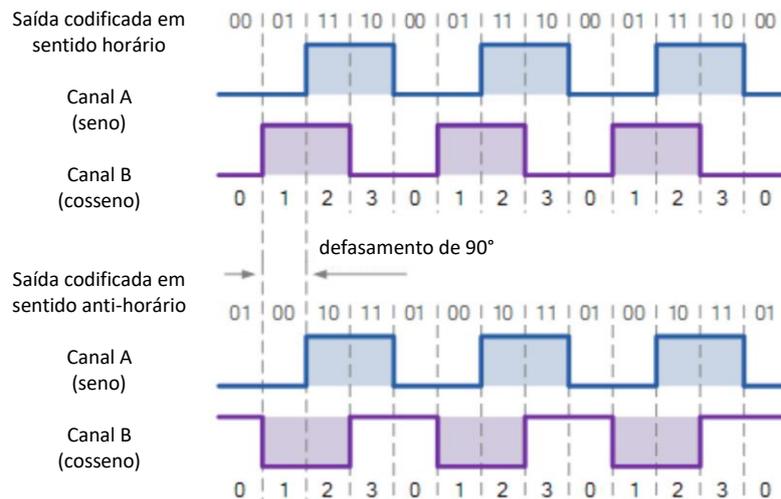
Para a medição do deslocamento, cada pulso contado equivale a uma fração do perímetro da roda, e essa fração é o deslocamento percorrido pelo robô. Para calcular essa fração da distância percorrida, foi utilizada uma relação de proporcionalidade entre o perímetro total da roda  $P = 2\pi r$  e o número total de pulsos necessários para uma revolução completa da roda. Dessa forma, com o raio  $r$  da roda = 0,0325 m, e o número de pulsos por revolução  $p = 392$  pulsos/ rev., temos que a distância percorrida a cada pulso  $\bar{p}$  é:

$$\bar{p} = \frac{2\pi r}{p} = \frac{2\pi \cdot 0,0325}{392} = 0,00052 \text{ m/pulsos} \quad (40)$$

Ao utilizar a leitura de apenas um sensor, este é denominado *encoder* incremental ou de quadratura. Embora seja eficiente para obtenção do deslocamento, não é possível obter o sentido de giro do motor e conseqüentemente das rodas. Para isso, são utilizadas as leituras dos dois sensores deslocados em 90° entre si. Este tipo de sensor é denominado *encoder* com dupla

quadratura. Dessa forma, observando a sequência da variação dos níveis lógicos dos dois sensores, é possível saber se a roda está girando em sentido horário ou anti-horário, como ilustrado pela figura 11:

Figura 11 - Leitura do *encoder* em ambos os sentidos de rotação.



Fonte: Adaptado de Eletronics Tutorials (2015)

Por existir a defasagem de 90° entre as ondas dos sinais lidos pelos sensores, existe uma sequência lógica que permite a identificação do sentido de rotação, como apresentado pela tabela 3:

Tabela 3 – Identificação do sentido de rotação do encoder			
Sequência lógica			
Sentido Horário		Sentido Anti-Horário	
Sensor 1	Sensor 2	Sensor 1	Sensor 2
0	0	0	0
0	1	1	0
1	1	1	1
1	0	0	1

Fonte: Autoria própria

### 3.1.3. Controladores

O grupo dos controladores é composto pelo microcontrolador, o módulo *bluetooth* e o driver (ponte H). O microcontrolador é o responsável por receber, através de suas entradas analógicas e digitais, a leitura das informações fornecidas pelos sensores, processar estas informações, por meio dos códigos de programação pré-carregados em sua memória, e executar as ações de controle do sistema por meio de suas saídas analógicas e digitais. Porém, este componente não tem capacidade de fornecer corrente suficiente para o acionamento de motores DC, sendo necessário utilizar um driver como interface entre o microcontrolador e o motor.

O driver, ou ponte H, é alimentado com uma tensão de entrada e conecta essa tensão ao motor conforme seus pinos de controle são energizados, os quais são comandados pelas saídas do microcontrolador utilizando sinal PWM.

A forma de comunicação padrão entre o microcontrolador e o computador é por meio de uma comunicação serial via USB, porém, o cabo conectado entre os dois equipamentos oferece uma força peso que interfere diretamente na dinâmica e controle do robô. Dessa forma, viu-se necessário abdicar-se da comunicação serial via cabo, e passou-se a utilizar o módulo *bluetooth* HC-06 para realizar a comunicação serial sem fio.

#### 3.1.3.1. Microcontrolador Arduino Nano

O microcontrolador utilizado neste trabalho foi o Arduino Nano, que é um hardware de linguagem aberta o qual possui o chip ATmega 328 como CPU. A alimentação desse hardware pode ser feita por conexão USB ou por fonte externa. Ele é responsável por armazenar toda a programação, que utiliza uma linguagem derivada da linguagem C, implementada utilizando a IDE Arduino, que é disponibilizada gratuitamente pelo próprio fabricante.

Por meio do código compilado pela IDE, e armazenado na CPU, é possível executar a leitura dos sensores e enviar os comandos para os atuadores, através das entradas e saídas analógicas e digitais do microcontrolador. Existem diversas possibilidades de escolha de microcontroladores da família Arduino, o Arduino Nano foi escolhido por ser altamente compacto, ter uma frequência de *clock* de 16MHz e quantidade de pinos I/O adequados à aplicação, além de apresentar um preço de mercado acessível. O Arduino Nano é representado pela figura 12.

Figura 12 - Arduino Nano



Fonte: Autoria Própria

As principais especificações técnicas do Arduino Nano são apresentadas pelo quadro 1.

Quadro 1 - Especificações técnicas do Arduino Nano

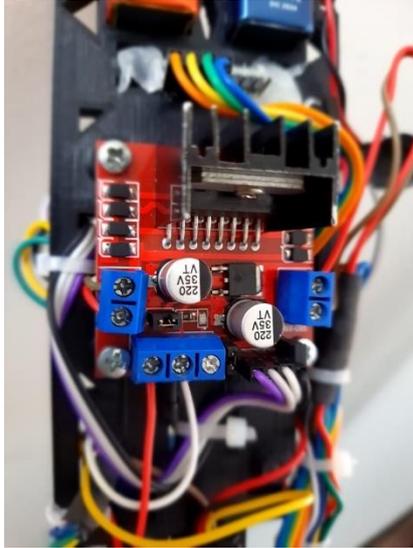
<b>Microcontrolador</b>	ATmega328
<b>Tensão operacional (nível lógico)</b>	5V
<b>Tensão de entrada (recomendada)</b>	7 – 12 V
<b>Pinos digitais E/S</b>	14 (em que 6 são PWM)
<b>Pinos de entrada Analógica</b>	8
<b>SRAM</b>	2 KB (ATmega328)
<b>EEPROM</b>	1 KB (ATmega328)
<b>Velocidade de clock</b>	16 MHz

Fonte: Folha de dados do Arduino Nano.

### 3.1.3.2. Driver (ponte H) L298N

O Driver L298N mostrado na figura 13 é um circuito eletrônico que tem como função ser a interface entre as saídas do microcontrolador e os motores de corrente contínua. Foi necessário utilizar essa interface pois o microcontrolador fornece, pelos seus pinos de saída, uma corrente de 40mA, e essa corrente é insuficiente para a alimentação dos motores. Esse driver utiliza uma configuração de transistores em ponte H, permitindo a alimentação do motor DC com tensões positivas ou negativas, cujos níveis médios de tensão aplicados aos motores podem ser regulados de 0 até  $V_{in}$ , comandados pelo microcontrolador através de um sinal PWM nos pinos de controle da Ponte H.

Figura 13 - Driver Ponte H L298N

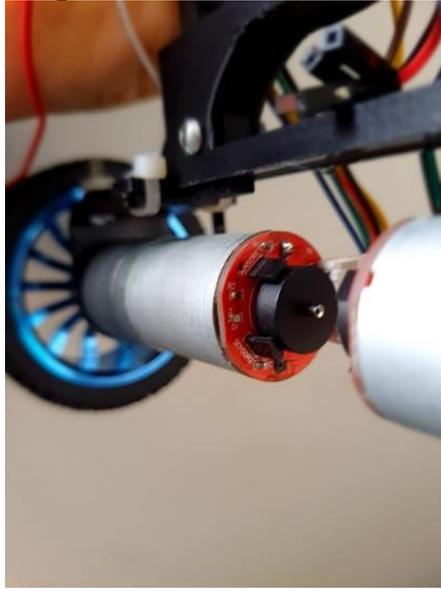


Fonte: Autoria própria

### 3.1.3.3. Motor CC 6V com caixa de redução

Um motor de corrente contínua é uma máquina elétrica que converte energia elétrica de corrente contínua em energia mecânica na forma de movimento de rotação. Essa transformação é devida as forças de atração e repulsão dos eletroímãs e ímãs permanentes presentes em seu interior. Os dois motores utilizados neste trabalho são do modelo JGA25-370 e possuem, cada um, uma caixa de redução, o que garante uma redução na velocidade de rotação em prol de um aumento considerável do torque de saída. Na Figura 14 é apresentado o conjunto motor CC 6V acoplado à caixa de redução.

Figura 14 - Motor JGA25-370.



Fonte: Autoria própria

As características técnicas do conjunto motor com caixa de redução escolhido para este trabalho são apresentadas no Tabela 4:

Tabela 4 – Características técnicas do motor JGA25-370

	<b>Sem carga</b>	<b>Com Carga</b>	<b>Rotor Bloqueado</b>
<b>Corrente (A)</b>	0,6	1,1	3,2
<b>Rotação (rad/s)</b>	17,80	11,52	0
<b>Torque (Nm)</b>	2	5,2	10
<b>Resistência (<math>\Omega</math>)</b>	1,1		

Fonte: Folha de dados do motor JGA25-370

#### 3.1.4. Estruturas mecânicas

O grupo das estruturas mecânicas é constituído pelas rodas e pelo chassi do robô, onde são fixados todos os demais componentes que compõem o sistema do pêndulo invertido.

O modelo das rodas utilizadas no robô é apresentado pela figura 5, e seu raio é de 3,25 cm.

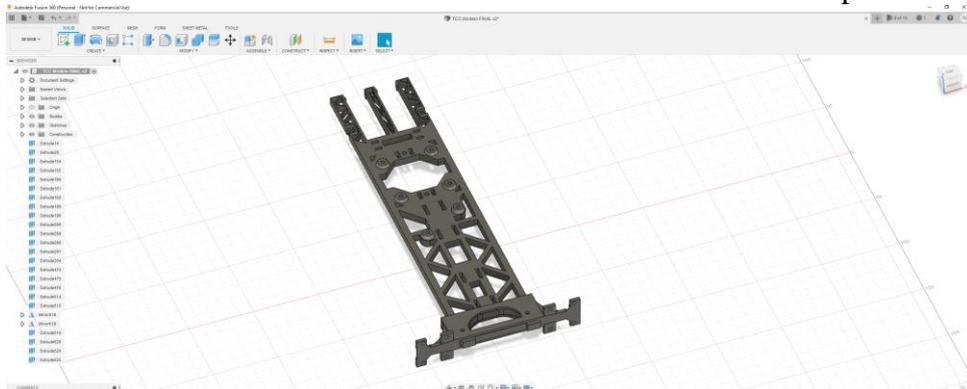
Figura 15 - Roda do pêndulo invertido.



Fonte: Autoria própria

O chassi e as estruturas da versão final do corpo do pêndulo foram projetados através do *software Fusion 360* (figura 16) e foram impressos utilizando uma impressora 3D pelo método de Modelagem por Deposição Fundida ou *Fused Deposition Modeling (FDM)*. O material utilizado para impressão do chassi é o Acrilonitrila Butadieno Estireno (ABS). Uma parte do processo de confecção é registrada pela figura 17.

Figura 16 – Tela do software Fusion 360 com o modelo 3D do chassi do pêndulo invertido.



Fonte: Autoria própria utilizando o software: Fusion 360.

Figura 17 - Processo de construção do chassi pela impressora 3D



Fonte: Autoria própria.

Houve, no total, três tentativas de construção do corpo do pêndulo. A primeira, como protótipo, um chassi foi fabricado utilizando materiais como acrílico e plástico, porém o seu desempenho foi inferior ao chassi impresso pela impressora 3D, pois não foi possível posicionar os componentes de forma simétrica, o que dificultaria o cálculo do momento de inércia. O método de confecção do corpo do pêndulo por impressão 3D, foi escolhido devido a simplicidade, nível de detalhamento e precisão que a modelagem tridimensional oferece. Na segunda tentativa foi feito um chassi que distribuía os componentes de forma mais simétrica, por meio da impressão 3D. Porém, houve algumas alterações de componentes, como o motor por exemplo, e com isso foi necessário a confecção da terceira versão.

As figuras 18, 19 e 20 apresentam respectivamente, a primeira versão do chassi, a segunda versão, e a versão final.

Figura 18 – Primeira versão do chassi



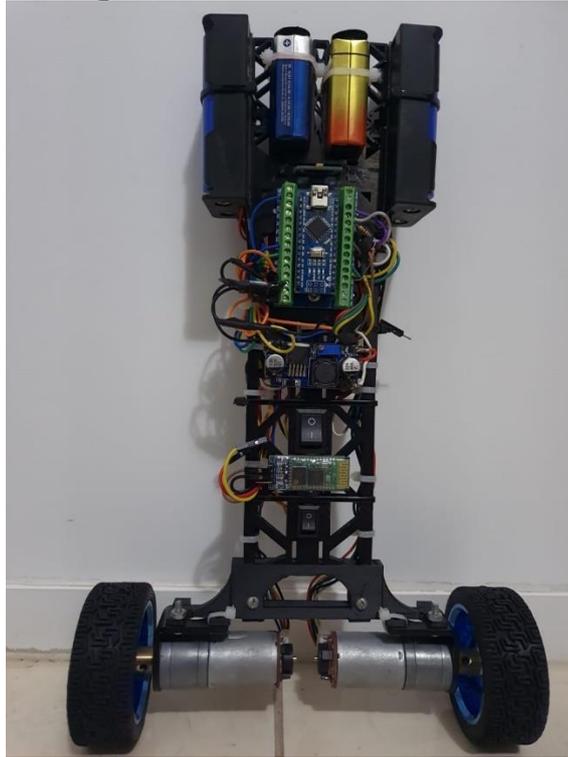
Fonte: Autoria própria.

Figura 19 - Segunda versão do chassi



Fonte: Autoria própria.

Figura 20 - Versão final do chassi.



Fonte: Autoria própria.

## 3.2. Métodos

Descritos os componentes utilizados no projeto, passaremos agora ao estudo dos mesmos e do modelamento do pêndulo invertido.

### 3.2.1. Modelagem dos motores

Os dados para a modelagem dos motores, foram obtidos utilizando os parâmetros de especificações técnicas dos motores apresentados anteriormente pelo quadro 2. Além desses, é necessário calcular o coeficiente mecânico  $K_m$  e a constante de força contra-eletromotriz  $K_e$ .

A equação (41) é utilizada para calcular o coeficiente mecânico  $K_m$ :

$$\tau_m = K_m \cdot i \quad (41)$$

Onde:

$\tau_m$  é o torque que o motor imprime;

$K_m$  é o coeficiente mecânico.

Dessa forma, utilizando os dados referentes ao rotor bloqueado, presentes no quadro 1, obtém-se  $K_m = 0,3065 \text{ Nm/A}$

Para o cálculo de  $K_e$ , a equação utilizada foi a equação (42):

$$V_a = R \cdot i + K_e \cdot \omega \quad (42)$$

Onde:

$V_a$  é a tensão aplicada ao motor

$R$  é a resistência do motor

$i$  é a corrente que circula no motor

$K_e$  é o coeficiente elétrico

$\omega$  é a velocidade angular do motor

Utilizando os dados de motor com carga, presentes na tabela 1, foi obtido o valor de  $K_e = 0,4158 \text{ Vs/rad}$ .

### 3.2.2. Obtenção do centro de massa

Na análise do sistema, com o intuito de obter o seu modelo matemático e calcular o momento de inércia, é necessário conhecer diversas medidas relacionadas aos seus aspectos construtivos, tais como massa, centro de gravidade e dimensões de cada uma das partes que o compõe.

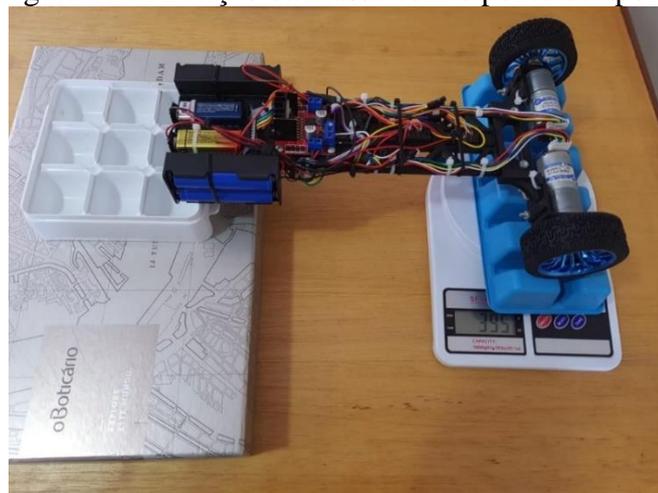
Uma balança de precisão foi utilizada para aquisição dos valores de massa e para o cálculo do centro de massa. O método utilizado para obtenção do centro de massa foi inspirado em Bushnell (2014), que consiste em aferir a massa total do sistema (figura 21) e repousar o robô horizontalmente com a sua extremidade inferior em uma balança, enquanto a sua extremidade superior é apoiada em um apoio fixo (figura 22). Dessa forma, a partir do diagrama de corpo livre da figura 23, utilizando a somatória das forças com relação ao eixo  $y$  e o momento de inércia resultante no ponto B, obteve-se  $l$ , que representa a distância entre o eixo de rotação e o centro de massa conforme apresentado pelas equações (43) e (44).

Figura 21 - Aferição da massa total do sistema.



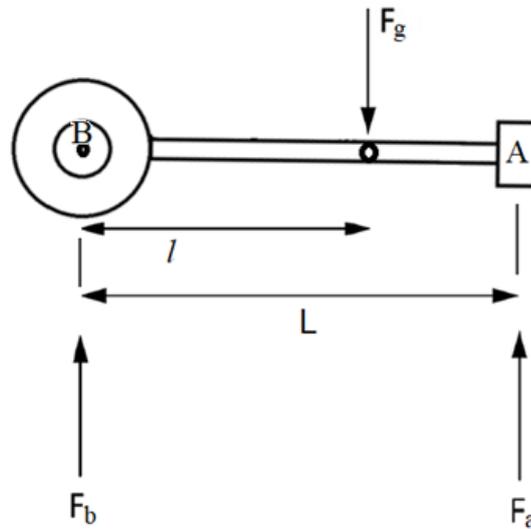
Fonte: Autoria própria.

Figura 22 - Aferição da massa com o ponto de apoio.



Fonte: Autoria própria.

Figura 23 - Diagrama de corpo livre para a obtenção do centro de massa.



Fonte: Autoria própria

$$\sum F_y = 0 \rightarrow F_b + F_a - F_g = 0 \rightarrow F_a = -F_b + F_g = -0,395 + 0,650 = 0,255m \quad (43)$$

$$\sum M_B = 0 \rightarrow F_b l + F_g h = 0 \rightarrow l = \frac{F_a}{F_g} = \frac{0,255}{0,650} \cdot 0,285 = 0,11 \text{ m} \quad (44)$$

### 3.2.3. Obtenção do momento de inércia da haste

Devido à complexidade de forma e de dimensões da haste, o cálculo do momento de inércia resultante da haste, utilizando medições convencionais, seria muito trabalhoso. Pensando nisso, o método utilizado para obtenção do momento de inércia da haste foi inspirado por Bushnell (2014), que consiste em adquirir o valor do momento de inércia total da haste, tomando como ponto de partida a equação que descreve o movimento oscilatório de um pêndulo simples, apresentada pela equação:

$$T = 2\pi \sqrt{\frac{J_h}{m_h g l}} \quad (45)$$

Onde:

$T$  é o período de oscilação da haste;

$J_h$  é o momento de inércia da haste;

$l$  é a distância entre o eixo de rotação e o centro de massa;

$m_h$  é a massa da haste;  
 $g$  é a aceleração da gravidade.

Colocando  $J_h$  em evidência, é possível obter o momento de inércia da haste, e a equação (44) é rearranjada da seguinte forma:

$$J_h = \frac{m_h g l T^2}{(2\pi)^2} \quad (46)$$

Para a obtenção da massa, a haste foi pesada em uma balança de precisão. A distância  $l$  foi adquirida através do experimento da seção anterior, e o período de oscilação da haste foi obtido pendurando-a de ponta-cabeça com o eixo dos motores apoiados em um suporte e foi solta de forma a exercer o movimento semelhante a oscilação de um pêndulo simples. A figura 26 ilustra o experimento.

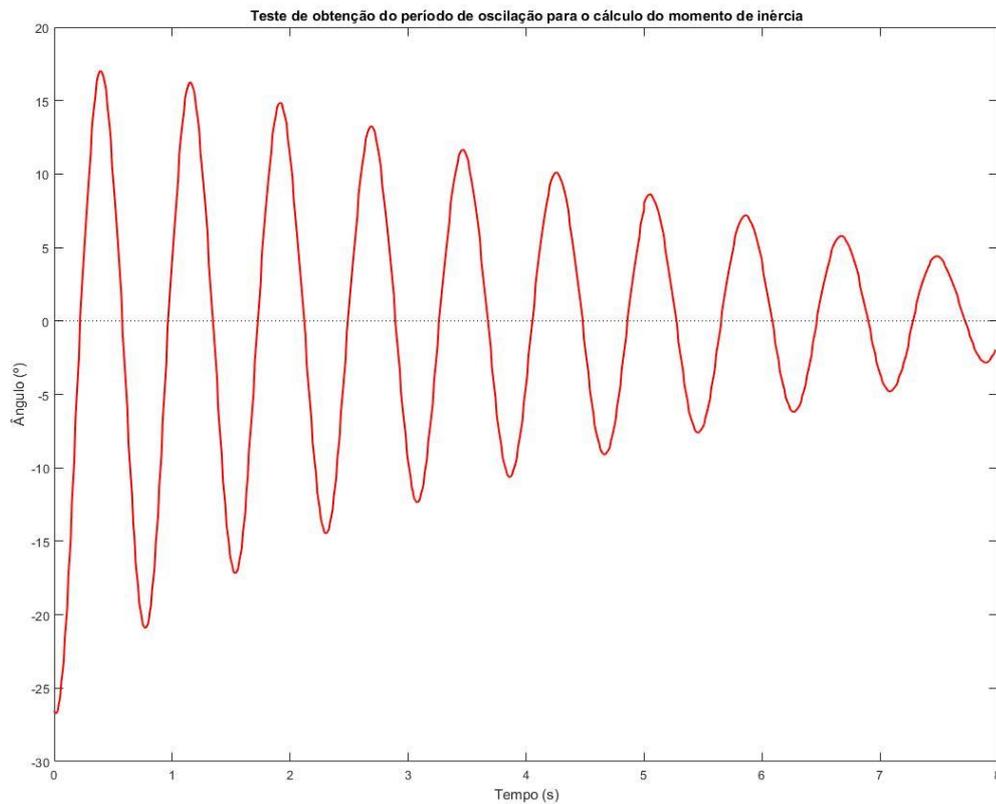
Figura 24 - Experimento para mensurar o período de oscilação da haste do pêndulo.



Fonte: Autoria própria.

A variação angular durante o experimento foi registrada por meio da leitura do sensor de medida inercial, e esses dados foram utilizados para o cálculo do período de oscilação. Dessa forma, foi calculada a média entre os períodos entre cada valor máximo de ângulo, resultando no período  $T = 0,77$  s. A figura 27 apresenta a trajetória angular descrita pela haste.

Figura 25 – Deslocamento angular durante o experimento para o cálculo do período de oscilação.



Fonte: Autoria própria.

Com todos os dados necessários em mãos, o momento de inércia da haste foi calculado utilizando a equação (46).

### 3.2.4. Obtenção do momento de inércia das rodas

Para o cálculo do momento de inércia das rodas, essas foram consideradas como um cilindro que gira em seu próprio eixo, que também é o eixo de rotação do pêndulo invertido. A equação para o cálculo do momento de inércia das rodas é apresentada pela equação (47).

$$J_r = \frac{m_h \cdot r^2}{2} \quad (47)$$

Onde:

$J_r$  é o momento de inércia das rodas;

$m_h$  é a massa das rodas;

$r$  é o raio das rodas;

Dessa forma, os valores obtidos pelas aferições e cálculos são apresentados pela tabela 5.

Tabela 5 - Parâmetros obtidos pelas aferições e cálculos.

Parâmetro	Valor	Unidade
Momento de inércia da roda	3,38e-5	Kg.m <sup>2</sup>
Momento de inércia da haste	12,5e-3	Kg.m <sup>2</sup>
Massa da roda	0,64e-2	Kg
Massa da haste	0,7	Kg
Raio da roda	3,25e-2	m
Distância até o centro de massa	10,2e-2	m

Fonte: Autoria própria.

Em posse desses valores, foi possível calcular os coeficientes das matrizes A e B que constituem o modelo do pêndulo invertido, representados na forma de espaço de estados pela equação (48).

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0001 & 0. & 0.0000 \\ 0.0000 & 1.0011 & -0.0041 & 0.0100 \\ 0.0000 & 0.0049 & 0.1511 & 0.0000 \\ 0.0000 & 0.2128 & -0.6284 & 1.0011 \end{bmatrix} \cdot \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0.0004 \\ 0.0003 \\ 0.0664 \\ 0.0491 \end{bmatrix} \cdot V \quad (48)$$

### 3.2.5. Projeto do RLQ

De posse do modelo dinâmico do sistema, apresentado no capítulo 2 e 3, e dos valores das variáveis adquiridos pelos sensores, foi possível implementar o controlador RLQ, introduzido no capítulo 2.

As equações do modelo dinâmico, que descrevem o funcionamento do sistema no espaço de estados estão no domínio contínuo, porém para aplicar o algoritmo do controle pelo microcontrolador tais equações devem ser discretizadas. Neste trabalho, para a discretização foi utilizado o método *Zero Order Hold* (ZOH).

Utilizando como base as equações (2) e (3) apresentada no capítulo 2, o modelo simplificado que descreve o espaço de estados no domínio discreto é:

$$x_{(k+1)} = A_d x_k + B_d u_k \quad (49)$$

$$y_k = C_d x_k + D_d u_k \quad (50)$$

Nas equações (49) e (50),  $A_d$ ,  $B_d$ ,  $C_d$ ,  $D_d$  são, respectivamente, a matriz discreta de estados, a matriz discreta de entradas, a matriz discreta de saídas e a matriz discreta de transmissão direta, definidas por:

$$A_d = e^{AT} \rightarrow e^{AT} = \mathcal{L}^{-1}\{(sI - A)^{-1}\}_{t=T} \quad (51)$$

$$B_d = \left( \int_{\tau=0}^T e^{A\tau} d\tau \right) \cdot B \quad (52)$$

$$C_d = C \quad (53)$$

$$D_d = D \quad (54)$$

Onde  $T = 0,01s$  é o período de amostragem. Para realizar a transformação de estado contínuo para o discreto, foi utilizado o software Matlab pois esse possui ferramentas e funções específicas para esse tipo de transformação, a função utilizada para tal foi a `sys_d = c2d(sys, Ts, 'zoh')`, onde `sys` é a equação de espaço de estados no tempo contínuo, `Ts` é a taxa de amostragem e `'zoh'` é a definição do método utilizado, o *Zero Order Hold*.

Os cálculos para obtenção do vetor de ganho  $K$  do RLQ também foram feitos utilizando o Matlab, por meio da função `K = rlqd(Ad, Bd, Cd, Dd, Rd)`, onde  $A_d$ ,  $B_d$ ,  $C_d$  e  $D_d$  são as matrizes discretas no espaço de estados e  $R_d$  é o equivalente discreto para o fator multiplicador  $R$  para otimizar o custo de controle do ganho do RLQ. Como foi apresentado no capítulo 2, o vetor  $K$  é utilizado para o cálculo da saída do sistema, que no caso deste trabalho é o valor da tensão aplicada por PWM aos motores.

Os valores obtidos do vetor de ganho  $K$  são apresentados no capítulo 4, como discussão dos resultados obtidos para diversos valores de  $Q$  e  $R$  em simulação pelo Matlab® assim como em testes utilizando o robô real.

A entrada  $u$  do sistema foi calculada por meio de um algoritmo carregado no microcontrolador utilizando as matrizes de espaço de estados e ganhos do controlador e observador discretizados, conforme o código presente no apêndice B.

### 3.2.6. Simulações para análise da resposta do sistema

Após obter o modelo matemático da resposta dinâmica do sistema; identificar os parâmetros dos motores e da estrutura do chassi do robô; transformar os modelos do domínio contínuo para o discreto; e apresentar a forma de obtenção do vetor de ganho de controle  $K$  para o cálculo da saída do sistema através do controle RLQ; foram feitas simulações utilizando a ferramenta Simulink do software Matlab® para observação do comportamento do sistema.

Uma rotina foi desenvolvida e implementada na simulação, para observar a resposta dos estados do sistema, quando é aplicada a esse, uma variação na referência do estado da posição do sistema. Essa rotina é dividida em cinco fases:

Primeira fase: Após inicialização do sistema, aguardar 10 segundos sem referencial de posição adicional.

Segunda fase: Aplicar à entrada do sistema, um sinal adicional na referência de posição com o formato de rampa ascendente que varia de 0cm a 30cm durante 20 segundos;

Terceira fase: Manter a referência de posição em 30 cm por mais 20 segundos;

Quarta fase: Aplicar um sinal em formato de rampa descendente que varia de 30cm a 0cm ao longo de 20 segundos;

Quinta fase: Manter o sistema sem referencial de posição adicional por mais 10 segundos.

Foram feitas diversas simulações utilizando essa rotina, variando-se os valores da matriz  $Q$  e do  $R$  para observar o comportamento do sistema diante a variação do ganho de controle RLQ a fim de se obter valores para um ganho ótimo nos experimentos práticos.

Os gráficos com as respostas da simulação e as análises da dinâmica do sistema serão apresentados posteriormente no capítulo 4.

### 3.2.7. Testes para análise das respostas do comportamento real do sistema

Após realizadas as simulações, observando o comportamento do sistema com diversos valores para o ganho RLQ, foram selecionados os valores para a matriz  $Q$  e o  $R$ , os quais

apresentaram a melhor resposta para o sistema. Em posse desses valores, foram executados três testes para a confirmação do modelo físico do robô.

O primeiro teste trata-se de observar a resposta do sistema sem nenhum impulso ou perturbação externa; dessa forma, o robô deve permanecer em posição vertical sem se deslocar no espaço.

O segundo teste consiste em analisar a robustez do controle e a sua estabilidade a variações bruscas e aleatórias no ângulo. Para tal, foi introduzida uma perturbação angular no sistema, ao empurrar manualmente a haste para um dos lados. Espera-se que o sistema se estabilize após a perturbação, de forma a retornar para a posição vertical, parado na sua posição linear inicial.

Por fim, o terceiro teste é um algoritmo que executa uma rotina de variação da referência de posição idêntica à da simulação apresentada anteriormente, a fim de se obter a resposta real no modelo e compará-la a resposta simulada; sendo assim, é esperado que a resposta do sistema seja semelhante a que foi apresentada na seção que trata das simulações.

Os valores utilizados para o ganho RLQ ótimo, e os dados da resposta do experimento são apresentados no capítulo 4.

## 4. RESULTADOS

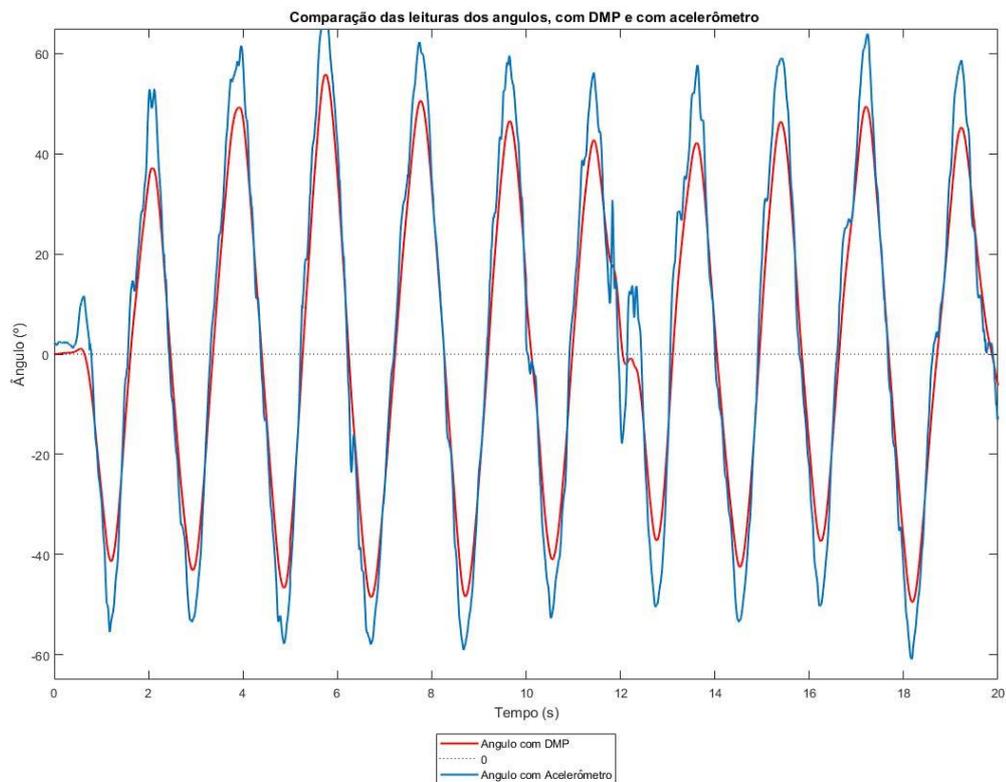
### 4.1. Análise dos ângulos do sensor inercial

Como apresentado no capítulo 3, os valores de ângulo obtidos utilizando-se somente o acelerômetro do MPU 6050, são muito ruidosos, foi necessário utilizar o processador interno DMP (*Digital Motion Processor*) do módulo.

Para analisar a precisão, confiabilidade e efetividade dos dados obtidos através da leitura do ângulo pelo processador interno da MPU 6050 foi realizado um teste que consiste em variar manualmente o ângulo de inclinação do corpo do robô entre dois ângulos iguais com sentidos diferentes durante um período de tempo.

Esse teste foi realizado com o objetivo coletar os valores de ângulo obtidos utilizando o DMP e utilizando somente o acelerômetro para comparação dos dois métodos. Dessa forma, os dados obtidos foram coletados e são apresentados em forma de gráfico na figura 28.

Figura 26 – Comparação entre as leituras dos ângulos com e sem a utilização do processador interno DMP.



Fonte: Autoria própria utilizando o software Matlab.

Nota-se que os dados referentes a medição de ângulo com a utilização do DMP apresentam muito menos ruído em comparação aos dados obtidos pelo acelerômetro.

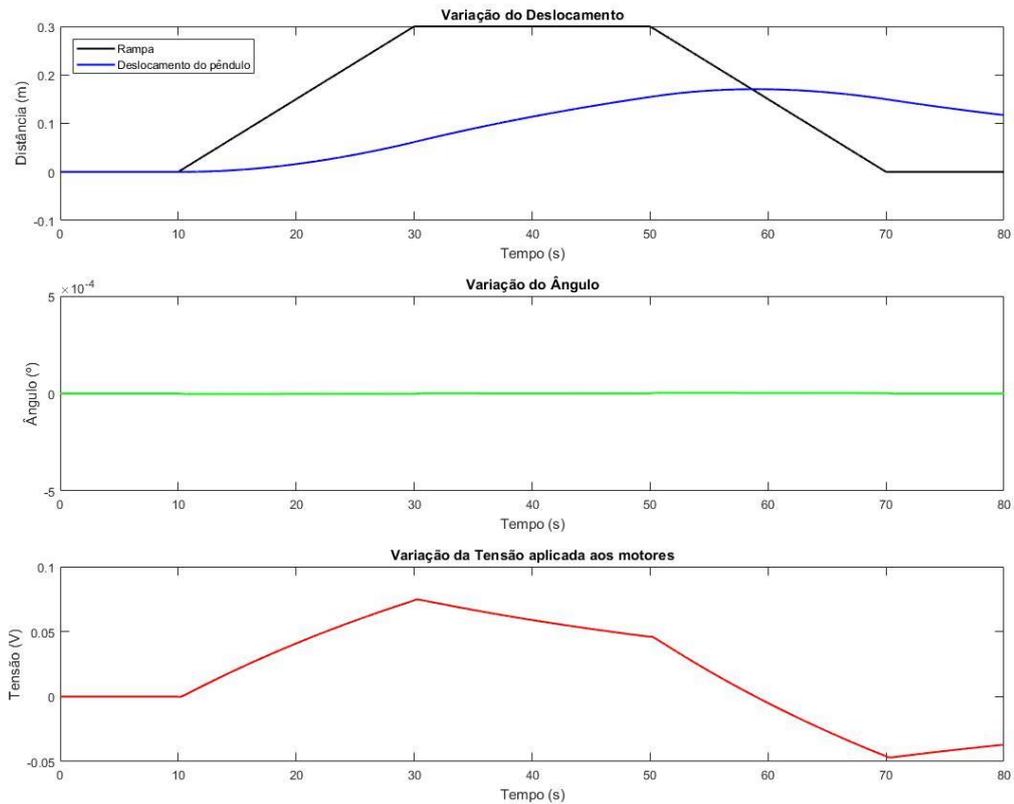
#### **4.2. Análise do controle através de simulações**

A seguir são apresentados, os gráficos das respostas da dinâmica das variáveis do sistema, obtidas por meio de simulação.

Cada figura foi obtida utilizando-se valores diferentes para o ganho RLQ, e possuem três gráficos que seguem uma ordem padrão de posicionamento, sendo que o primeiro descreve a resposta do deslocamento do robô em comparação às rampas aplicadas ao referencial da posição; o segundo descreve a variação do ângulo; e o terceiro a tensão aplicada aos motores. Todos os gráficos têm 8000 amostras e uma taxa de amostragem de 0,01 segundos por amostra.

Foi observado que quanto maior o valor escolhido para R, menor é tensão aplicada aos motores, isto vai contra ao objetivo real do sistema de controle, que é fazer com que todos os estados do sistema convirjam para zero no menor tempo possível, ou seja, o valor de R é inversamente proporcional à velocidade de resposta da tensão aplicada aos motores. A figura 29 representa os estados do sistema para um peso R com um valor alto.

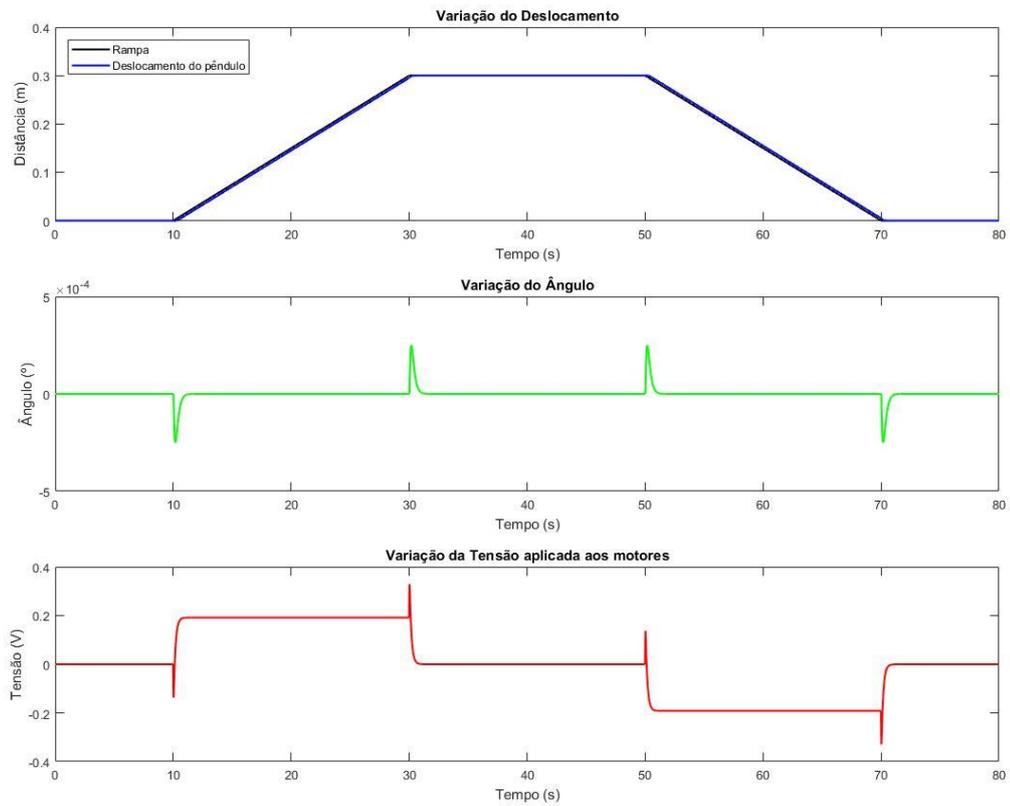
Figura 27 - Resposta simulada ao controle com um valor elevado para a matriz R.



Fonte: Autoria própria utilizando o software Matlab.

A figura 30 representa a resposta do sistema para um valor alto do peso da matriz Q referente ao estado de posição do robô. Esse aumento resultou em um alto ganho que reduziu o tempo de acomodação da posição do robô. Porém, essa ação da redução do tempo de acomodação é limitada à resposta de velocidade de rotação dos motores, quanto maior esse limite, maior será a possibilidade de aumento do ganho de peso Q referente ao estado x e menor será o tempo de acomodação do sistema.

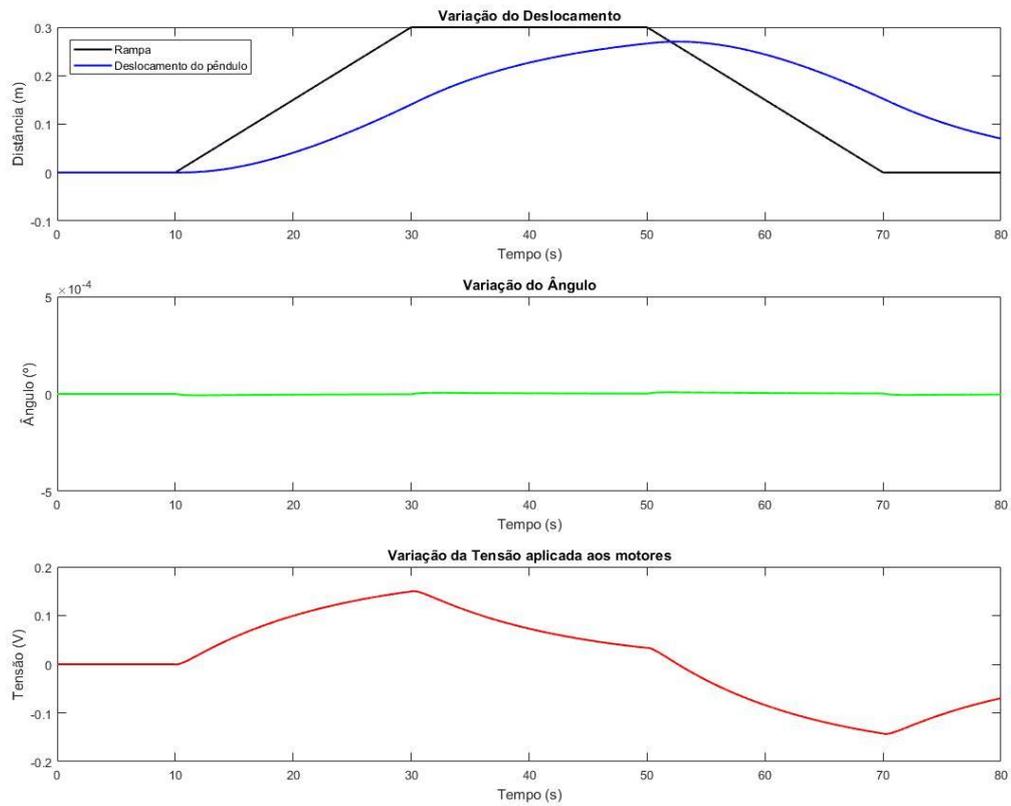
Figura 28 - Resposta simulada ao controle com um valor elevado para o índice da matriz Q referente ao estado x.



Fonte: - Autoria própria utilizando o software Matlab.

A Figura 31 mostra que com o aumento do peso da matriz Q referente ao ângulo de inclinação, resulta em um aumento significativo no tempo de acomodação da resposta dos outros estados.

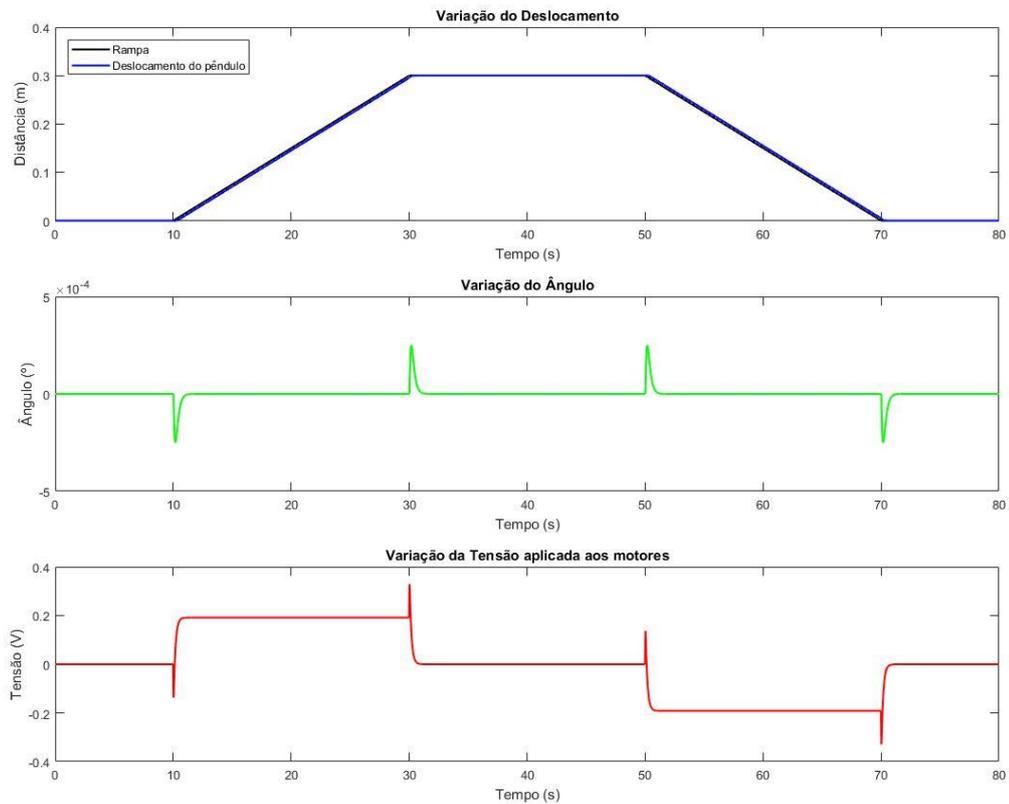
Figura 29 - Resposta simulada ao controle com um valor elevado para o índice da matriz Q referente ao estado do ângulo.



Fonte: Autoria própria utilizando o software Matlab.

Como os elementos da matriz Q e do R são determinados de forma empírica, foram realizados testes com valores diferentes através da simulação, a fim de se obter uma resposta mais próxima do ideal para o controle do robô. Essa resposta é apresentada pela figura 32

Figura 30 - Valores para as matrizes Q e R, determinados de forma empírica.



Fonte: Autoria própria utilizando o software Matlab.

Mais à frente, na próxima seção serão apresentados testes com esses valores ideais, porém com o modelo real do robô.

#### 4.3. Análise do comportamento real do sistema

A seguir, serão apresentados os resultados dos três testes, comentados na seção 3.2.6, utilizando o modelo real do pêndulo invertido. Para a execução dos testes, foram utilizados os valores dos pesos da matriz Q e de R que mais se aproximaram de um ganho RLQ ideal para a resposta do sistema, esses valores estão presentes na figura 33.

Figura 31 - Melhores valores para a matriz Q e R.

```

Qd =
    500     0     0     0
     0    100     0     0
     0     0     1     0
     0     0     0    10

Rd =
    1.0e-07 *

```

Fonte: Autoria própria

#### 4.3.1. Teste 1 – Análise da estabilidade do sistema sem impulsos ou perturbações

A figura 34 apresenta os dados obtidos durante o Teste 1. Ao analisar os resultados, foi observado que apesar de apresentar comportamento oscilatório, os valores escolhidos para o ganho do controlador RLQ foram suficientes para o sistema manter-se em posição vertical durante todo o tempo de teste, como esperado.

É esperado que a resposta apresente oscilações, devido à dificuldade de obtenção de parâmetros que um modelo real apresenta, como folga no eixo dos motores, coeficiente de atrito das rodas, saturação da fonte de tensão ou imprecisões e ruídos na aquisição de dados obtidos pelos sensores.

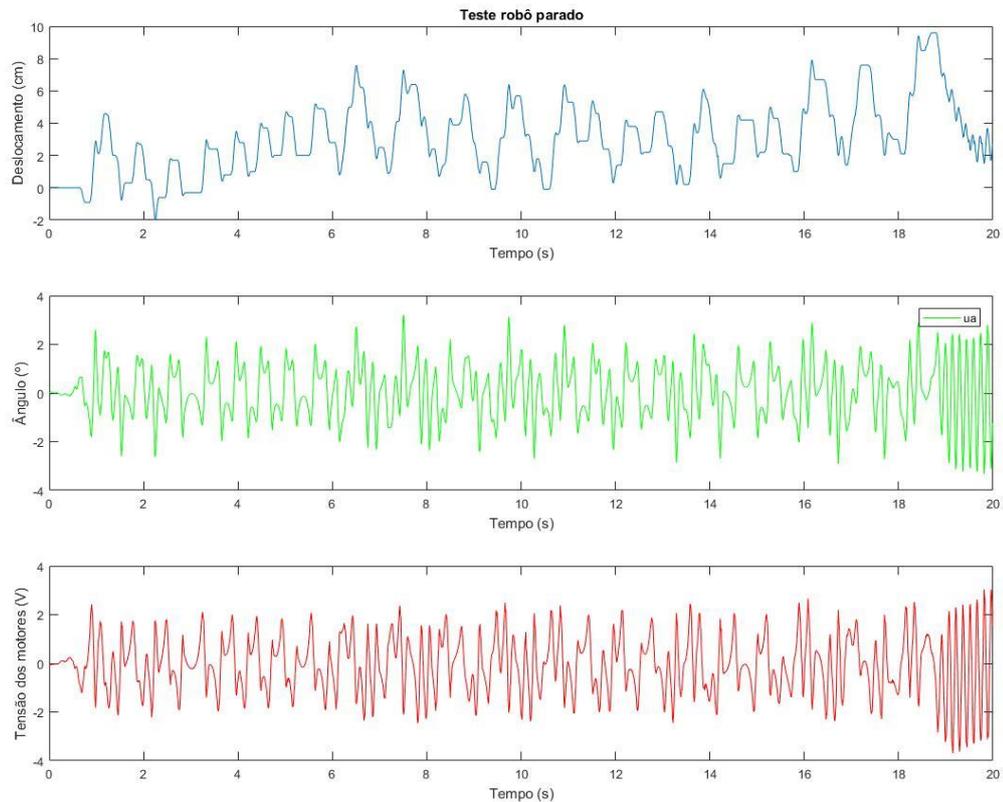
A tabela 6 apresenta o valor máximo, mínimo e a média do deslocamento linear durante o teste.

Tabela 6 - Apresentação dos dados obtidos pelo Teste 1.

	Máximo	Mínimo	Média
Deslocamento (cm)	9,6	-2	3,06
Ângulo (°)	3,2	-3,34	0,04
Tensão (V)	3,04	-3,69	-0,07

Fonte: Autoria própria.

Figura 32 – Teste de estabilidade do robô sem perturbação.



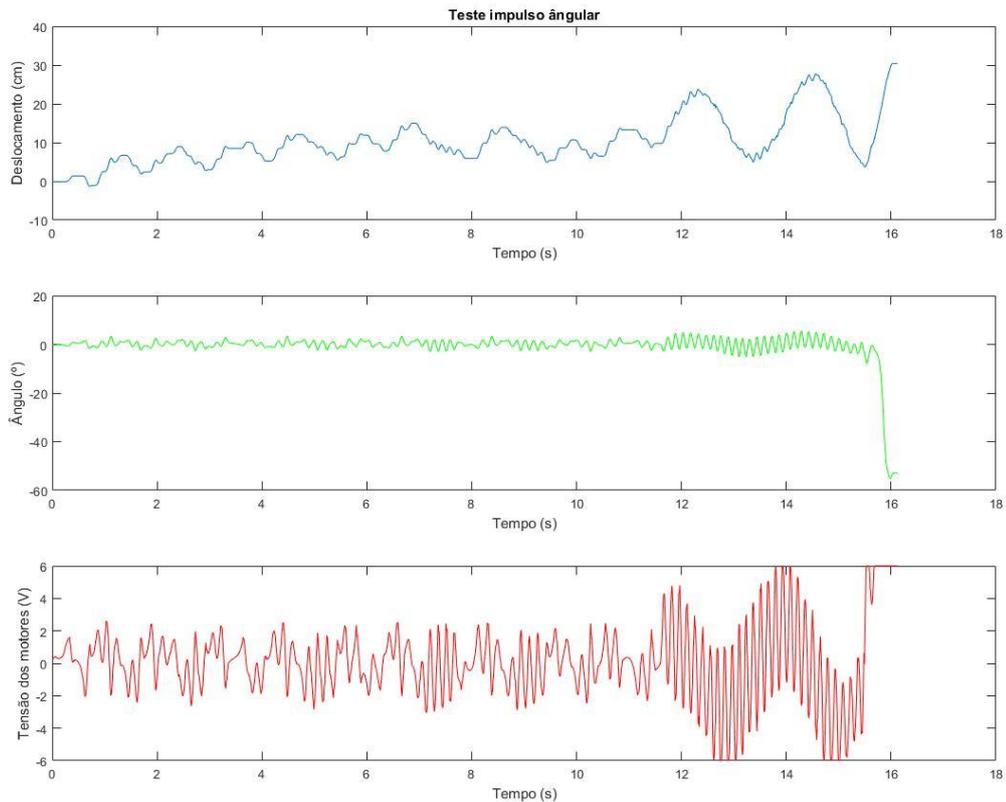
Fonte: Autoria própria utilizando o software Matlab.

#### 4.3.2. Teste 2 – Análise da resposta do sistema à perturbação angular

A resposta do sistema ao Teste 2 é apresentada na figura 35. O impulso no ângulo durante o teste foi aplicado no instante de tempo igual a 11,6 segundos.

Ao analisar os resultados obtidos, observa-se que apesar de o sistema ficar estável durante o Teste 1, os valores escolhidos para calcular o ganho do controlador RLQ não foram suficientes para manter a estabilidade do controle durante o teste 2, uma vez que ao aplicar a perturbação no ângulo, o sistema entrou em oscilação até cair.

Figura 33 - Resposta do robô à perturbação no ângulo.



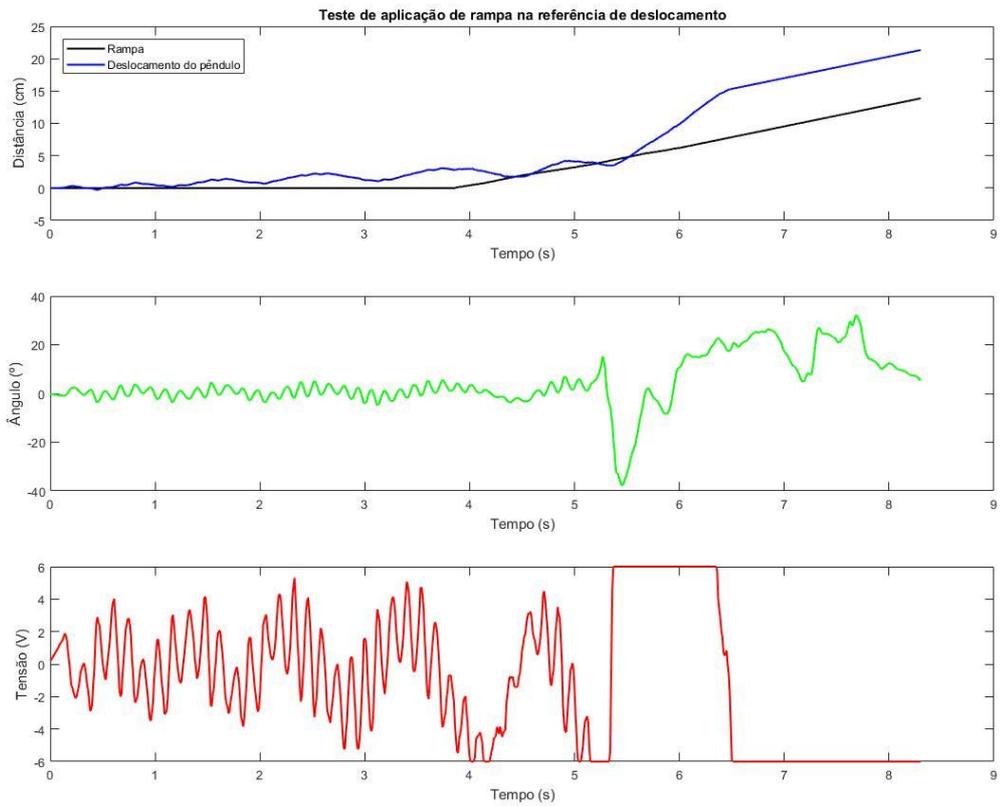
Fonte: Autoria própria utilizando o software Matlab.

#### 4.3.3. Teste 3 – Análise da resposta do sistema à alteração da referência de posição em formato de rampa.

A resposta do sistema ao Teste 3 é apresentada pela figura 36. Ao analisar a resposta, observa-se que, assim como observado durante o Teste 2, os valores obtidos para cálculo do ganho RLQ não foram suficientes para manter a estabilidade do sistema durante o Teste 3. Ao aplicar a rampa no instante de tempo 10 segundos, o sistema acompanha a variação na referência do deslocamento, e corrige a sua posição até próximo ao instante 14 segundos. Após isso, o sistema entra em instabilidade devido à saturação do valor da tensão.

Como o sistema necessitou de uma tensão acima do limite fornecido, ao atingir a saturação, não conseguiu corrigir a posição em tempo suficiente, aumentando a sua instabilidade até não conseguir mais ficar na posição vertical.

Figura 34 - Resposta do robô ao impulso no deslocamento em formato de rampa.



Fonte: Autoria própria utilizando o software Matlab.

## 5. CONCLUSÕES

O presente trabalho propôs desenvolver o controle do modelo real de um robô móvel do tipo pêndulo invertido de duas rodas, e com o objetivo de utilizá-lo como material didático na disciplina de controle multivariável. Para tal, o modelo da dinâmica do sistema foi levantado e obtidas as equações do sistema no espaço de estados. Posteriormente foi desenvolvido e aplicado um controlador, utilizando o método de controle RLQ, com o intuito de controlar o sistema de forma a minimizar os estados de deslocamento e ângulo instantâneos do robô.

O sistema proposto foi validado por meio de simulações computacionais utilizando o software Matlab e pela implementação do controle em um modelo real do robô, o qual teve o chassi projetado por meio do software Fusion 360 e confeccionado por impressão 3D. O robô construído possui componentes eletrônicos capazes de adquirir a leitura dos dados dos estados do sistema e de executar o processamento do controle em uma velocidade compatível com o período de amostragem.

Para a aplicação do controle, foi observado que os sensores e o microcontrolador possuem velocidade de aquisição de dados e processamento condizentes com o necessário para a aplicação desenvolvida. Além disso, foi necessário executar testes nos sensores de medição do ângulo e do deslocamento, a fim de analisar a precisão e confiabilidade na aquisição dos dados. Todos os sensores apresentaram valores adequados para a aplicação proposta.

Como apresentado pelos resultados das simulações e do Teste 1, o controle RLQ conseguiu estabilizar o ângulo em uma posição fixa, o que possibilita que ele seja utilizado como material didático em práticas na disciplina de controle multivariável. Porém, ao analisar os resultados dos Testes 2 e 3 vê-se necessário a melhoria na escolha dos valores das matrizes para calcular o ganho do controlador RLQ, de forma a minimizar as incertezas e ruídos que um sistema real apresenta em comparação à simulação.

Durante as pesquisas para o desenvolvimento deste trabalho, foram encontrados diversos métodos de controle diferentes do utilizado neste trabalho, tais como: controle fuzzy; PID; LQG; alocação de polos; Lyapunov, dentre outros. Propõe-se, através dos resultados obtidos e da descoberta de aplicações que utilizam métodos de controle alternativos, o desenvolvimento de trabalhos que visam estudar a resposta do modelo físico, utilizando-se dos demais tipos de controladores, também com o objetivo de observar as diferenças entre as respostas do sistema aos controles aplicados e realizar comparações entre os resultados obtidos.

## REFERÊNCIAS

DORF, R.; BISHOP, R. H. **Sistemas de controle moderno**. 11. ed. Rio de Janeiro: LTC, 2009. 752 p.

Electronics Tutorials. **Position Sensors**. Programming Robots, 2015 Disponível em: [https://www.electronics-tutorials.ws/io/io\\_2.html](https://www.electronics-tutorials.ws/io/io_2.html) . Acesso em: 29/01/2022.

FAGUNDES, R dos S. **Robô pêndulo invertido**. Universidade Federal do Rio Grande do Sul (TCC). Porto Alegre, 2015.

FARIA, Lucas M. **Modelagem e controle do pêndulo invertido rotacional ou pêndulo de Furuta**. Centro Federal de Educação Tecnológica de Minas Gerais (TCC). Belo Horizonte, 2015.

GADELHA, Nayana Freitas. **Controle de um pêndulo invertido utilizando lógica fuzzy**. Universidade Federal Rural de Semiárido (TCC). Caraúbas, 2018.

HOWARD, B. **MinSeg™ state-space RLQ controller, PID controller, and motor controller development**. Appendix A – MinSeg Measurements. Washington, 2014. Disponível em: <http://dairyroadsolutions.com/htmlMinSeg/indexMatlab.html> . Acesso em 20/01/2022.

NISE, N. **Engenharia de sistemas de controle**. 3. ed. Rio de Janeiro: LTC, 2002. 693 p.

OGATA, K. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Pearson, 2010.

OOI, Rich Chi. **Balancing a Two-Wheeled Autonomous Robot**. The University of Western Australia (Final Year Thesis). 2003.

PAULA, Fabio Oliveira de. **Sensores IMU – Uma Abordagem Completa – Parte 2**. Laboratório Mobilis Computação Móvel, UFOP, 2015 Disponível em: <http://www2.decom.ufop.br/imobilis/sensores-imu-uma-abordagem-completa-parte-2/>. Acesso em: 29/01/2022.

PEDROSO, C. C. de S.; MODESTO, E. L. P. **Sistema de controle de pêndulo invertido**. Universidade Tecnológica Federal do Paraná (TCC). Curitiba, 2017.

PRIYADARSHI, Prassana. **Optimal Controller Design for Inverted Pendulum System: An Experimental Study**. National Institute of Technology (Master Thesis). Rourkela, 2013.

TABORDA, Renan A. **Análise de desempenho de controladores aplicados ao sistema pêndulo invertido**. Universidade Tecnológica Federal do Paraná (TCC). Toledo, 2014.

VIEIRA, Iago S. **Modelagem dinâmica e controle de atitude de um quadricóptero**. Centro Federal de Educação Tecnológica de Minas Gerais (TCC). Araxá, 2017.

White, Martin. **The MPU6050 Explained**. Programming Robots, 2019 Disponível em: <https://mjwhite8119.github.io/Robots/mpu6050> . Acesso em: 29/01/2022.

## APÊNDICE A – Código utilizado no Matlab®

```

clear all, close all, clc

% % -----Parâmetros:-----
mr = 0.0640;      % Massa das rodas (kg)
mh = 0.6500;      % Massa da haste (kg)
l = 0.1100;      % Distância entre o eixo de rotação e o centro
de massa (m)
g = 9.8100;      % Aceleração da gravidade (m/s^2)
Jh = 9.7e-3;     % Momento de inércia da haste (Kgm^2)
Jr = 3.38e-5;    % Momento de inércia da roda (Kgm^2)
r = 0.0325;     % Raio da roda (m)
km = 0.3065;    % Coeficiente mecânico (Nm/A)
ke = 0.4158;    % Coeficiente elétrico (V*s/rad)
Ra = 1.1000;    % Resistência da armadura (ohm)

beta = 2*mr+mh+2*Jr/(r*r);
alpha = Jh*beta+2*mh*l*l*(mr+(Jr/(r*r)));

a22 = mh^2*g*l^2/alpha;
a23 = 2*km*ke*(mh*l*r-Jh-mh*l^2)/(Ra*r^2*alpha);
a42 = mh*g*l*beta/alpha;
a43 = 2*km*ke*(r*beta - mh*l)/(Ra*r^2*alpha);

b21 = 2*km*(Jh + mh*l^2 - mh*l*r)/(Ra*r*alpha);
b41 = 2*km*(mh*l-r*beta)/(Ra*r*alpha);

A = [ 0,      0,      1,      0;
      0,      0,      0,      1;
      0,      a22,    a23,    0;
      0,      a42,    a43,    0];
B = [ 0;      0;      b21;      b41];

% x1 = x
% x2 = teta
% x3 = v = dx
% x4 = w = dteta
% dx = A*x + B*u

% Sairas
C = [ 1, 0, 0, 0
      0, 1, 0, 0];

D = 0;

sys = ss(A,B,C,D); %Matriz espaço de estados

% Condição inicial:
x0 = [0;0*pi/180;0;0]; % [xc ap dxc/dt dap/dt]

```

```

eig(A); % Sistema instável

Sc = ctrb(sys); rank(Sc)
So = obsv(sys); rank(So)

% Discretização:
Ts = 0.01;
sys_d = c2d(sys, Ts, 'zoh');
Ad = sys_d.a
Bd = sys_d.b % GRAFICO 2 (Qx >>>>)
Qd = diag([999999, 1, 1, 1]);
Rd = 1;

Cd = sys_d.c
Dd = sys_d.d

% GRAFICO 1 (R>>>>)
Qd = diag([100, 100, 1, 1]);
Rd = 999;

% GRAFICO 2 (Qx >>>>)
% Qd = diag([999999, 1, 1, 1]);
% Rd = 1;

% GRAFICO 3 (Qtheta >>>>)
% Qd = diag([1, 999999, 1, 1]);
% Rd = 1;

% GRAFICO 4 (Melhor resultado durante a simulação)
% Qd = diag([5000, 10, 1, 1])
% Rd = 0.1

N = zeros(length(B), 1);
[K_d, ~, ~] = dlqr(Ad, Bd, Qd, Rd, N) %Cálculo do ganho LQR
Rd = Rd*eye(2);
[Kobv, ~, ~] = dlqr(Ad', Cd', Qd, Rd) %Calculo do ganho do
observador

```

## APENDICE B – Código utilizado no arduino para cálculo da entrada do sistema.

```

#define deg2rad 0.017453292
#define min_angle_threshold 3*deg2rad
#define max_angle_threshold 50*deg2rad
#define u_threshold 6
#define PWM_max 255
#define x_min 0

float PWM_normalize = PWM_max/u_threshold //Constante para normalização da saída para
valor PWM.
double bias = 0; // Utilizado para a calibração do angulo 0
double pref = 0; // Utilizado para alterar a referência de distância
float u = 0; // entrada u
float PWM = 0; // Tensão aplicada aos motores.
/*x Vector:*/
float x1, x2, x3, x4; // Estados x1 = distância; x2 = ângulo; x3 = velocidade linear;
x4 = velocidade angular

/*=====
-----Matrix-----
=====
*/
//Declaração das Matrizes A, B, C e ganhos K e Kobv (Discretizados).
float Ad11 = 1.000000, Ad12 = 0.000047, Ad13 = 0.005133, Ad14 = 0.000000;
float Ad21 = 0.000000, Ad22 = 1.002087, Ad23 = -0.001423, Ad24 = 0.010007;
float Ad31 = 0.000000, Ad32 = 0.007512, Ad33 = 0.217906, Ad34 = 0.000047;
float Ad41 = 0.000000, Ad42 = 0.416975, Ad43 = -0.228730, Ad44 = 1.002087;

float Bd11 = 0.000380;
float Bd21 = 0.000111;
float Bd31 = 0.061131;
float Bd41 = 0.017878;

float Cd11 = 1.000000, Cd12 = 0.000000, Cd13 = 0.000000, Cd14 = 0.000000;
float Cd21 = 0.000000, Cd22 = 1.000000, Cd23 = 0.000000, Cd24 = 0.000000;

float K_d11 = -1.481611, K_d12 = 580.017822, K_d13 = -25.259948, K_d14 = 90.072923;

float Kobv11 = 0.765564, Kobv12 = 0.000004, Kobv13 = 0.000000, Kobv14 = 0.000023;
float Kobv21 = 0.000042, Kobv22 = 0.595779, Kobv23 = 0.004897, Kobv24 = 0.838020;

/*KobvT:*/
//Kobv transposto
float KobvT11 = Kobv11, KobvT12 = Kobv21;
float KobvT21 = Kobv12, KobvT22 = Kobv22;
float KobvT31 = Kobv13, KobvT32 = Kobv23;
float KobvT41 = Kobv14, KobvT42 = Kobv24;

/*=====
-----Operations-----
=====*/

volatile float y1 = 0; // distância atual
float y2 = bias; // Angulo atual

/*Ad*x*/
float Adx1, Adx2, Adx3, Adx4;

/*Bd*u*/
float Bdu1, Bdu2, Bdu3, Bdu4;

/*Cd*(x-y)*/

```

```

float Cdx_y1, Cdx_y2;

/*Kobv*(Cd*x)-y*/
float KoCd1, KoCd2, KoCd3, KoCd4;

void exec_controll() {

    y1 = get_position() + pref;
    y2 = get_angle() - bias;

    /*-----Ad*x Multiplication-----*/

    Adx1 = Ad11 * x1 + Ad12 * x2 + Ad13 * x3 + Ad14 * x4;
    Adx2 = Ad21 * x1 + Ad22 * x2 + Ad23 * x3 + Ad24 * x4;
    Adx3 = Ad31 * x1 + Ad32 * x2 + Ad33 * x3 + Ad34 * x4;
    Adx4 = Ad41 * x1 + Ad42 * x2 + Ad43 * x3 + Ad44 * x4;

    /*-----Bd*u Multiplication-----*/

    Bdu1 = Bd11 * u;
    Bdu2 = Bd21 * u;
    Bdu3 = Bd31 * u;
    Bdu4 = Bd41 * u;

    /*----- (Cd*x)-y Multiplication-----*/

    Cdx_y1 = Cd11 * x1 + Cd12 * x2 + Cd13 * x3 + Cd14 * x4 - y1;
    Cdx_y2 = Cd21 * x1 + Cd22 * x2 + Cd23 * x3 + Cd24 * x4 - y2;

    /*-----KobvT*(Cd*x)-y Multiplication-----*/

    KoCd1 = -KobvT11 * Cdx_y1 + -KobvT12 * Cdx_y2;
    KoCd2 = -KobvT21 * Cdx_y1 + -KobvT22 * Cdx_y2;
    KoCd3 = -KobvT31 * Cdx_y1 + -KobvT32 * Cdx_y2;
    KoCd4 = -KobvT41 * Cdx_y1 + -KobvT42 * Cdx_y2;

    /*-----Ad*x Multiplication-----*/

    Adx1 = Ad11 * x1 + Ad12 * x2 + Ad13 * x3 + Ad14 * x4;
    Adx2 = Ad21 * x1 + Ad22 * x2 + Ad23 * x3 + Ad24 * x4;
    Adx3 = Ad31 * x1 + Ad32 * x2 + Ad33 * x3 + Ad34 * x4;
    Adx4 = Ad41 * x1 + Ad42 * x2 + Ad43 * x3 + Ad44 * x4;

    /*-----Get x-----*/

    x1 = Adx1 + Bdu1 + KoCd1;
    x2 = Adx2 + Bdu2 + KoCd2;
    x3 = Adx3 + Bdu3 + KoCd3;
    x4 = Adx4 + Bdu4 + KoCd4;

    u = -(K_d11 * x1 + K_d12 * x2 + K_d13 * x3 + K_d14 * x4); //Calculo da entrada.

    PWM = u * PWM_normalize; //Normalização do valor da entrada dos motores para
valores // de PWM de 0 a 255.

    if (u > u_threshold) //Limite para valor max de tensão = 6v e PWM = 255
    {
        PWM = PWM_max;
        u = u_threshold;
    }
    if (u < -u_threshold)
    {
        u = -u_threshold;
        PWM = -PWM_max;
    }
}

```

```
    if ((y2 > max_angle_threshold || y2 < -max_angle_threshold )) // Rotina para
desligar
    {
        PWM = 0; // os motores se o ang-
                // ulo for maior que um
                //valor pré-determinado
    }
    if (y1 < x_min && y1 > -x_min)
    {
        y1 = 0;
    }
    else
    {
    }
    start_motors(PWM); //Acionamento dos motores
}
```



---

Emitido em 17/02/2022

**CÓPIA DO TRABALHO Nº 16/2022 - DELMAX (11.57.05)**

**(Nº do Protocolo: NÃO PROTOCOLADO)**

*(Assinado digitalmente em 17/02/2022 15:21 )*

**HENRIQUE JOSE AVELAR**

*PROFESSOR ENS BASICO TECN TECNOLOGICO*

*DELMAX (11.57.05)*

*Matrícula: 392234*

Para verificar a autenticidade deste documento entre em <https://sig.cefetmg.br/documentos/> informando seu número: **16**, ano: **2022**, tipo: **CÓPIA DO TRABALHO**, data de emissão: **17/02/2022** e o código de verificação: **aedd17102e**