

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Campus DIVINÓPOLIS
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

Matheus Ferreira Costa

CONSTRUÇÃO E CONTROLE DE UM SISTEMA DO TIPO PÊNDULO INVERTIDO
SOBRE DUAS RODAS



Divinópolis
2019

Matheus Ferreira Costa

CONSTRUÇÃO E CONTROLE DE UM SISTEMA DO TIPO PÊNDULO INVERTIDO
SOBRE DUAS RODAS

Monografia de Trabalho de Conclusão de Curso
apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.

Áreas de integração: Controle e Mecânica.

Orientador: Prof. Dr. Luís Filipe Pereira Silva

Co-orientador: Prof. Dr. Juliano de Barros Veloso e Lima

Divinópolis
2019

Matheus Ferreira Costa

CONSTRUÇÃO E CONTROLE DE UM SISTEMA DO TIPO PÊNDULO INVERTIDO
SOBRE DUAS RODAS

Monografia de Trabalho de Conclusão de Curso
apresentada ao Colegiado de Graduação em En-
genharia Mecatrônica como parte dos requisitos
exigidos para a obtenção do título de Engenheiro
Mecatrônico.

Áreas de integração: Controle e Mecânica.

Comissão Avaliadora:

Prof. Dr. Luís Filipe Pereira Silva
CEFET/MG *Campus* Divinópolis

Prof. Dr. Wagner Custódio de Oliveira
CEFET/MG *Campus* Divinópolis

Prof. M. Lucas Silva de Oliveira
CEFET/MG *Campus* Divinópolis

Divinópolis
2019

Resumo

O sistema robotizado de pêndulo invertido se tornou largamente conhecido através do Segway nas últimas décadas, inspirando diversas pesquisas por sistemas de controle eficazes e robustos. Paralelamente, a versão autônoma do veículo também tem atraído a atenção da comunidade acadêmica por se tratar de um sistema não linear e instável em malha aberta. Dessa maneira, o trabalho objetivou-se na construção e controle um sistema do tipo pêndulo invertido sobre duas rodas. Assim, necessitou-se de estudos aprofundados sobre controladores que consigam trabalhar com sistemas instáveis. Uma das técnicas de controle amplamente utilizadas em sistemas com múltiplos estados, é o controlador LQG que advém da teoria do controle ótimo. Para um perfeito funcionamento deste controlador é necessário que a modelagem matemática do sistema seja bem feita e que a construção física não fique muito distante do modelo. Sendo assim, a primeira tarefa deste documento focou-se na obtenção do modelo que descreve o movimento do sistema. Por se tratar de um sistema no qual os sinais de controle aplicados pelo microcontrolador sejam de caráter digital, a discretização do sistema se fez necessário. Com a utilização do SolidWorks realizou-se o modelo 3D da planta e assim, os parâmetros físicos da estrutura foram estimados por este *software*. Tendo todos os parâmetros físicos e elétricos necessários, obteve-se o vetor de ganhos ótimos do LQR e do estimador LQE. Dessa maneira, adquiriu-se respostas do controlador em relação ao modelo não linear por meio do Simulink.

Palavras-chave: Segway. Pêndulo Invertido Sobre Duas Rodas. Controle Ótimo. LQG.

Lista de Figuras

1.1	Protótipo Pêndulo Invertido (PEDROSO; MODESTO, 2017).	3
1.2	Protótipo renderizado (MELO, 2017).	4
1.3	Implementação do controlador LQR (ENGIN, 2018).	4
1.4	Planta física construída (ENGIN, 2018).	5
2.1	Segway X2 SE (SEGWAY, 2019).	8
2.2	Representação elétrica dos motores.	9
2.3	Diagrama corpo livre da roda.	10
2.4	Diagrama corpo livre do chassi.	12
2.5	Estrutura de um controlador LQG.	17
2.6	Estrutura do Filtro de Kalman discreto em Espaço de Estados. Adaptado de BRUNTON <i>et al.</i> (2017).	19
3.1	Fluxograma das Atividades desenvolvidas no TCC.	22
3.2	Estrutura modelada no SolidWorks.	23
3.3	Conjunto completo do atuador: motor, acoplador e roda (BAÚ DA ELETRÔNICA, 2019).	24
3.4	<i>Shield</i> MPU6050 (INVENSENSE, 2019).	26
3.5	<i>Shield</i> MPU6050.	26
3.6	Microcontrolador ESP32 (FERNANDO K, 2019).	28
3.7	Ponte H L298N (VIDA DE SILÍCIO, 2019).	29
3.8	Sinal PWM (VIDA DE SILÍCIO, 2019).	30
3.9	Bateria Lipo Turnigy 2200 mAh 3s (MERCADO LIVRE, 2019).	31
4.1	Implementação do controlador LQR.	37
4.2	Implementação do controlador LQG.	37
4.3	Variação do ângulo da estrutura com o controlador LQR.	39
4.4	Sinal de controle do sistema com o controlador LQR.	40
4.5	Velocidade linear e angular do sistema com o controlador LQR.	40
4.6	Variação do ângulo da estrutura com o controlador LQG.	41
4.7	Sinal de controle da planta com o controlador LQG.	41
4.8	Velocidade linear e angular do sistema com o controlador LQG.	42
4.9	Distúrbio na ordem de $\pi/50$ aplicado ao estado estimado $\hat{x}_2 = \hat{\theta}$.	42
4.10	Variação do ângulo da estrutura com um distúrbio imposto ao sistema.	43
4.11	Sinal de controle com um distúrbio imposto ao sistema.	43

Lista de Tabelas

3.1	Parâmetros mecânicos da estrutura.	23
3.2	Parâmetros do motor e da roda.	25
3.3	Parâmetros do giroscópio: velocidade de giro e nível de sensibilidade.	27
3.4	Comparativo entre ESP32, ESP8266 e Arduino Uno R3	29
3.5	Consumo dos Motores CC.	31
3.6	Tabela de custo dos materiais utilizados na construção do protótipo.	32
5.1	Cronograma de atividades para o TCC2.	45

Lista de Acrônimos e Notação

LQR	Linear Quadratic Regulator (regulador linear quadrático)
LQE	Linear Quadratic Estimator (estimador linear quadrático)
LQG	Linear Quadratic Gaussian (linear gaussiano quadrático)
LTI	Linear Time-Invariant (linear invariante no tempo)

A	notação para matrizes (letras maiúsculas do alfabeto latino)
\in	símbolo "contido em"
\forall	quantificação universal ("para todos; para qualquer; para cada")
\mathbb{R}	conjunto dos números reais
\mathbb{E}	símbolo para expectância ou esperança de uma variável aleatória
\mathbf{I}	matriz identidade de dimensão apropriada
n	especialmente utilizada para representar a ordem uma matriz quadrada
\int	símbolo de integral
\sum	notação de somatório
α	símbolos utilizados para simplificar os termos do modelo (letras gregas)

Introdução

A Automação é uma tecnologia por meio da qual um procedimento ou processo é conseguido sem a interferência humana. Para realizar a mesma, é preciso que um programa de instruções seja escrito combinado a um sistema de controle que executa as tarefas. De acordo com GROOVER (2013), um processo automatizado necessita-se de energia não só para operar a condução do processo como para operar o programa e o sistema de controle. Ademais, considera-se uma grande aliada da otimização do desempenho, já que com essa tecnologia é possível conhecer indicadores que auxiliam a gestão, acelera os processos e remove trabalhos repetitivos, dispensáveis e não aceitáveis ergonomicamente.

Uma grande área em ascensão da robótica é a robótica móvel. Sendo assim, os robôs se locomovem dentro de um ambiente qualquer de diversas formas, com rodas, esteiras, pernas mecânicas, dentre outras. Estes robôs são caracterizados pela capacidade de se deslocar, podendo ser de modo guiado, semiautônomo ou totalmente autônomo (JUNG *et al.*, 2005). Algumas de suas mais variadas aplicações são: aspiração de pó, entrega de alimentos, vigilância predial, busca e salvamento, etc.

Assim sendo, neste trabalho, pretende-se o desenvolvimento de um veículo do tipo pêndulo invertido sobre duas rodas, que pode ser classificado como um robô do tipo móvel. Para a estabilização na vertical deste sistema, pretende-se a utilização de técnicas de controle robustas que irão conseguir tratar as simplificações e considerações de seu modelo matemático.

1.1 Definição do Problema

O projeto proposto tem por finalidade a construção de um protótipo que tenha uma estrutura bem projetada/construída para que as intervenções externas sejam mínimas e a elaboração de um controlador ótimo.

1.2 Motivação

O estudo da Teoria de Controle em sistemas dinâmicos e a implementação de controladores em uma planta física, complexa e com características não lineares, além da concretização

de um projeto completo é de suma importância na consolidação dos conceitos assimilados ao longo da formação acadêmica de um profissional da área de Mecatrônica.

Além de tudo, após o estudo de três disciplinas seguidas da área de controle e a vontade do autor de conseguir realizar o controle de uma planta clássica, porém com um nível de dificuldade elevada e a construção de um projeto mecatrônico, esses foram os fatores que contribuíram para que o mesmo realizasse o projeto proposto.

1.3 Objetivos do Trabalho

Aqui é descrito de forma sucinta o objetivo geral e os objetivos específicos do trabalho. Fazendo-se cumprir esses objetivos específicos, espera-se alcançar o objetivo geral.

1.3.1 Objetivo Geral

Construir um sistema denominado pêndulo invertido sobre duas rodas e controlar a posição de sua estrutura para que a mesma fique paralela com a vertical (0° ou 0 radianos) utilizando controlador ótimo.

1.3.2 Objetivos específicos

Listam-se os seguintes objetivos específicos para o projeto completo:

- desenvolver por meio do SolidWorks o conceito do protótipo;
- levantar a lista e o custo dos materiais;
- obter o modelo não linear do sistema e linearizar a partir de técnicas jacobianas;
- discretizar o modelo linear obtido;
- projetar o controlador LQG com base no modelo linear;
- avaliar a resposta do controlador aplicando ao modelo não linear;
- realizar a construção da planta física;
- validar o modelo obtido;
- realizar a calibração do sensor e testes nos componentes eletrônicos da planta;
- implementar/embarcar o controlador no sistema.

1.4 Estado da Arte

O trabalho de PEDROSO; MODESTO (2017) consistiu na construção de um protótipo projetado por meio de um *software* CAD, visto na Figura (1.1), e no desenvolvimento de um sistema de controle para esse projeto. Os autores se engajaram em controlar o sistema com um controlador PID, obtendo seus coeficientes por meio do método de sintonização de *Ziegler-Nichols*. Para a parte eletromecânica do projeto, deram ênfase maior na relação torque do motor com o tamanho do diâmetro da roda, que estão totalmente relacionados. Utilizaram como sensor a placa MPU6050, que possui embutida na mesma o giroscópio e acelerômetro. Contudo, é necessário o tratamento que provém da leitura do sinal desses dois sensores. Sendo assim, os autores optaram por utilizar o filtro complementar, que nada mais é do que um filtro de média.

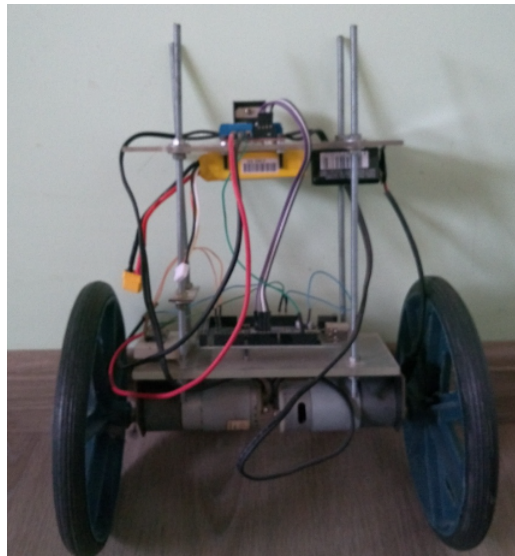


Figura 1.1: Protótipo Pêndulo Invertido (PEDROSO; MODESTO, 2017).

O artigo de MELO (2017) teve como objetivo a identificação e o controle de um veículo do tipo Segway para fins educacionais. Dessa maneira, o autor dividiu o artigo em partes como: o sistema eletrônico e mecânico, linearização do modelo matemático e discretização, implementação de filtros para a leitura dos sinais provindo do acelerômetro e giroscópio, identificação do sistema e a implementação de um controlador PD (Proporcional-Derivativo), sendo que seus coeficientes foram obtidos por meio de inspeção gráfica utilizando o lugar das raízes. O diagrama de malha fechada padrão, com o controlador em série com o sistema. Os resultados obtidos foram apenas em cima do modelo, sendo assim, não houve a implementação física ou mesmo a montagem do protótipo proposto, como visto na Figura (1.2). Os resultados obtidos na simulação foram bem satisfatórios para os critérios adotados, sendo que em malha fechada o autor conseguiu fazer com que o sistema seguisse a referência senoidal aplicada e após aplicação de um distúrbio, o sistema conseguiu se recompor rapidamente.

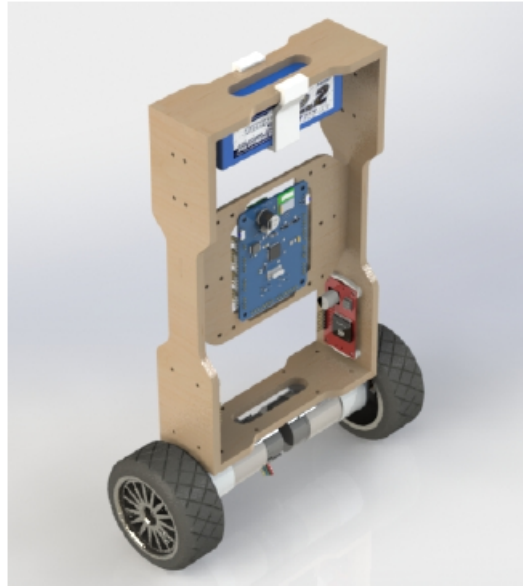


Figura 1.2: Protótipo renderizado (MELO, 2017).

Por último, o artigo de ENGIN (2018) propôs a implementação de um controlador LQR. A modelagem do sistema foi feita utilizando equações lagrangianas e, depois que encontrou uma equação não linear para o sistema, linearizou a mesma e passou para a forma de espaço de estados, chegando assim nas matrizes A , B , C e D . Aplicou a essas matrizes a propriedade de controlabilidade e observabilidade, concluindo assim, que é um sistema controlável e observável. O pêndulo invertido sobre duas rodas, é um sistema de uma única entrada e múltiplas saídas (SIMO). Dessa forma, para lidar com um tipo de sistema desses é mais simples com o LQR baseado pelo controle de velocidade como propôs o autor. Para utilizar esse controlador, é necessário encontrar um vetor de ganhos K e consequentemente, minimizará a função de custo J . Na Figura (1.3) é visto a implementação do controlador LQR.

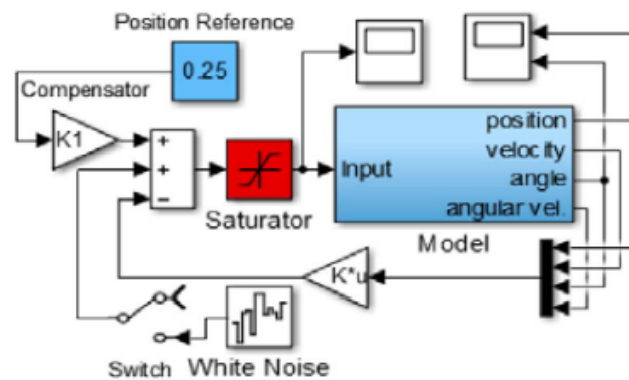


Figura 1.3: Implementação do controlador LQR (ENGIN, 2018).

O filtro que foi implementado também foi o complementar, que nada mais é do que um filtro passa-baixas para o acelerômetro e um filtro passa-altas para o giroscópio. As principais variáveis que foram medidas ou estimadas foram: a posição da estrutura, θ , a velocidade angular, $\dot{\theta}$ e, velocidade da roda, ω . O protótipo montado pode ser visto na Figura (1.4).

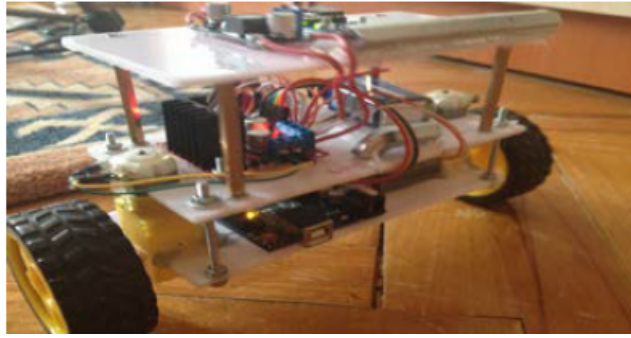


Figura 1.4: Planta física construída (ENGIN, 2018).

1.5 Organização do Documento

Este documento é dividido em quatro capítulos além das considerações finais.

O presente capítulo, introdução, apresenta a definição do problema que foi motivo de estudo e a motivação pela qual deu origem ao desenvolvimento do trabalho. Além do mais, apresenta os objetivos gerais e específicos definidos para que o projeto em questão seja realizado. Por fim, discorre sobre estado da arte do presente tema.

No segundo capítulo são apresentados os fundamentos, que consistem na revisão de literatura e fundamentação teórica, esse tem por objetivo histórica e teórica do trabalho.

O terceiro capítulo consiste na metodologia, que descreve como foi feito o projeto estrutural do protótipo, além das escolhas e justificas dos materiais que serão utilizados nesse trabalho.

O capítulo quatro inicia primeiramente expondo o modelo não linear contínuo. Em seguida, é feito a linearização e discretização do modelo para que o projeto dos controladores sejam realizados. Por fim, implementa tanto o LQR quanto o LQG no sistema não linear e analisa os resultados obtidos.

Por fim, no capítulo das considerações finais, é apresentado as conclusões desse trabalho e as tarefas restantes bem como um cronograma estimando o tempo que cada atividade gastará para ser cumprida na disciplina de TCC 2.

Fundamentos

Este capítulo tem como objetivo apresentar a parte histórica e teórica deste trabalho. A revisão de literatura fica a cargo do conteúdo histórico, descrevendo assim uma linha temporal do tema deste TCC. Na fundamentação teórica, serão descritos de forma clara as teorias mais importantes e necessárias para a realização do projeto, como por exemplo as técnicas de modelagem de sistemas e projeto de controlador ótimo.

2.1 Revisão de Literatura

Pode-se afirmar que o primeiro trabalho significativo de controle foi o regulador centrífugo construído por James Watt para o controle de velocidade de uma máquina de tear, no século XVIII. O sistema permitia realizar o controle da velocidade, controlando a injeção de combustível na máquina a vapor.

Segundo OGATA (2010), trabalhos importantes como de Minorsky, Nyquist e Hazen, datados entre as décadas de 20 e 30, contribuíram para o progresso da teoria de controle. Minorsky, em 1922, trabalhou com pilotagem de embarcações utilizando controladores automáticos e demonstrou por meio de equações diferenciais a estabilidade do sistema. Em 1932, Nyquist desenvolveu um procedimento no qual determina a estabilidade de sistemas em malha fechada com base na resposta em malha aberta. E Hazen, em 1934, introduziu o termo *servomecanismos* para sistemas de controle de posição e analisou o projeto de servomecanismos a relé.

No século XX, acontecimentos como a Segunda Guerra Mundial estimularam as pesquisas em sistemas de controle. No final dos anos 50, a teoria de controle já era bastante consolidada, sendo que o carro chefe era o método que utilizavam a resposta em frequência e com muitas aplicações industriais, como por exemplo: a utilização de controladores PID para o controle de pressão, temperatura, etc. Os métodos de resposta em frequência e do lugar das raízes, os quais são a base da teoria clássica de controle, conduziram os sistemas que são estáveis e satisfatórios para um determinado conjunto de condições. Esses sistemas são aceitáveis, porém não são ótimos no sentido literal do termo.

Entre as décadas de 60 e 80, segundo OGATA (2010), o controle ótimo de sistemas determinísticos, controle adaptativo e de aprendizagem foi altamente pesquisado. Diferentemente da teoria clássica, estes métodos se enquadram na teoria de controle moderno. Essa teoria se baseia no domínio do tempo em sistemas de equações diferenciais. Como vantagem, temos que essa teoria simplificou o projeto de sistemas de controle, já que se baseia no modelo de um sistema de controle real. Em contrapartida, a estabilidade é sensível ao erro entre o sistema real e do seu modelo. Dessa forma, quando o controlador projetado for aplicado no sistema, o mesmo poderá ficar instável.

Um dos sistemas clássicos de controle que há na literatura, é o pêndulo invertido. Este sistema possui uma única entrada e várias saídas. Dessa maneira, tratá-lo com técnicas de controle moderno tende a facilitar a estabilidade do mesmo.

Em OOI (2003) a técnica utilizada para o controle do pêndulo foi o LQR. Também desenvolveu-se um controlador por alocação de polos para a estabilização do sistema. Dessa forma, foi capaz realizar a comparação entre esses dois métodos e saber qual dos dois foi mais eficaz. Pelas simulações concluiu-se que os dois métodos atenderam, com ressalva de que o controlador LQR ofereceu maior confiabilidade.

A técnica *PID Backstepping*, que é não linear, foi utilizada por THAO *et al.* (2010) para controlar um pêndulo invertido sobre duas rodas (PIDR). Essa técnica consiste basicamente em uma estrutura de controle que possui três malhas de controle, sendo elas: 1) a malha principal, desenvolvida a partir da técnica não linear *backstepping* que manterá o pêndulo em equilíbrio; 2) a segunda malha que utiliza um PD para controlar a posição do robô e; 3) um controlador PI para o controle de movimentação.

Em JUNFENG; WANYING (2011) novamente é feito uma comparação entre o controlador ótimo LQR com o controlador por alocação de polos. Contudo, neste artigo, conclui-se que o *overshoot* foi menor por meio da técnica de alocação de polos e que o tempo de estabilização foi quase comparável ao do LQR.

No artigo de JUANG; LURRR (2013), os autores propuseram a desenvolver um protótipo utilizando o Arduino como microcontrolador e a implementação de duas técnicas de controle, sendo elas o PID e LQR na forma de PI-PD. Os resultados obtidos deixam bem claro que a técnica PI-PD foi muito mais eficaz em todos sentidos do que a utilização pura do PID. Como concluído, a estabilização por PID é marginal e as oscilações angulares excedem o limite máximo de torque que os motores podem oferecer. Já no PI-PD ou LQR a estabilidade é alcançada e ainda o controlador é capaz de compensar o desalinhamento do CG, fazendo com que o robô retorne para a posição angular inicial e que é a desejada.

Em PAULA (2014), o autor utilizou duas técnicas de controle discreto, LQR com ação integral e alocação de polos. A primeira técnica utilizada teve como objetivo a estabilização do pêndulo e a segunda o controle de velocidade dos motores. A técnica de alocação de polos além de realizar o controle dos motores, teve como objetivo interfacear o controle de ângulo e o sinal enviado para os motores.

Por último, ALVES (2018), além de projetar um controlador LQR, apresenta duas técnicas de linearização, sendo elas: a jacobina e a por realimentação. A modelagem do sistema baseou-se na equação Lagrangiana e após encontrar as equações de movimento, passou-se as mesmas para o Espaço de Estados. Os resultados obtidos foram satisfatório, contudo, ao sofrer interferências, como distúrbio e ruídos o sistema passa a ser mais oscilante.

Como dissertado acima, há vários trabalhos com diferentes resultados para o mesmo tema. Por se tratar de um sistema naturalmente instável, não linear e bastante complexo para controlar, é lógico que haverão trabalhos com resultados distintos. Entretanto, quando há domínio de um determinado assunto, é possível transformar os resultados teóricos em algo para a sociedade, como por exemplo o *Segway*, que é um veículo elétrico de transporte humano e seu funcionamento é baseado no pêndulo invertido sobre duas rodas.

O *Segway*, visto na Figura 2.1, inventado por Dean Karmen e revelado em dezembro de 2001 tem o funcionamento idêntico ao do pêndulo invertido, com adição que o mesmo está sobre duas rodas. A função básica deste veículo é o transporte de uma única pessoa. O sistema entende que deve-se locomover para frente ou mesmo frear, devido à inclinação que o passageiro impõe sobre o veículo. Hoje em dia, existem vários modelos e cada um com uma especificação diferente um do outro.



Figura 2.1: Segway X2 SE (SEGWAY, 2019).

Dessa forma, este trabalho busca a realização de um sistema semelhante ao *Segway* mas em uma versão miniatura, utilizando técnica de controle moderno e na abordagem no espaço de estados, que pode ser visto a seguir.

2.2 Fundamentação Teórica

Como já mencionado, este trabalho alinha-se na construção e controle de um veículo baseado em um pêndulo invertido sobre duas rodas. Dessa forma, será utilizado técnica de controle moderno, controle ótimo, com o modelo do sistema em espaço de estados.

2.2.1 Análise dinâmica do sistema

O sistema em estudo conforme já citado é descrito como instável e não linear. Porém para encontrar as equações de movimento que regem o sistema, é preciso escolher algum tipo de método. Para o trabalho em questão, escolheu-se representar o sistema por meio da segunda lei de Newton descrita por:

$$\sum \vec{F}_i = m \vec{a} \quad (2.1)$$

$$\sum \tau_{iz} = J \alpha_z \quad (2.2)$$

Os termos do lado esquerdo das Equações (2.1) e (2.2) representam todas as forças (F_i) e torques (τ) que atuam sobre o sistema. Já no lado direito da Equação (2.1), temos a massa total do corpo representada no centro de massa (m) e todas as acelerações lineares (a). Na Equação (2.2), as acelerações agora são as angulares representadas por (α_z) e o momento de inércia (J) é dado como sendo:

$$J = \sum m_i r_i^2$$

Equações do subsistema elétrico

O circuito elétrico que utilizará para modelar os motores CC do projeto, pode ser visto na Figura (2.2) (PAULA, 2014).

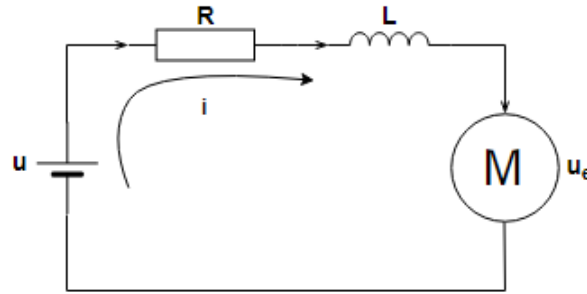


Figura 2.2: Representação elétrica dos motores.

O parâmetro u representa a tensão de controle aplicada nos motores, R e L representam a resistência e a indutância dos enrolamentos do estator respectivamente. O parâmetro u_e representa a tensão da força contra eletromotriz e é proporcional a diferença de velocidades angulares $\dot{\alpha}_w$ e $\dot{\theta}_b$ por uma constante K_e ,

$$u_e = K_e(\dot{\alpha}_w - \dot{\theta}_b)$$

O torque T_m produzido pelo motor é proporcional a corrente de enrolamento i , por meio da constante de torque do motor K_m .

$$T_1 = K_m i \quad (2.3)$$

Aplicando a Lei das Tensões de Kirchoff (KVL), pode-se estabelecer uma relação entre as tensões internas do circuito dos motores CC considerando separadamente cada componente. Para o circuito da Figura (2.2), obtém-se a seguinte equação.

$$u = Ri + L \frac{di}{dt} + K_e(\dot{\alpha}_w - \dot{\theta}_b) \quad (2.4)$$

Devido às condições de funcionamento dos motores, como por exemplo, a constante de tempo elétrica ser muito menor do que da parte mecânica, é possível ignorar a corrente de transiente da Equação (2.4). Assim, o circuito é simplificado como visto abaixo:

$$u = Ri + K_e(\dot{\alpha}_w - \dot{\theta}_b) \quad (2.5)$$

Isolando a corrente da Equação (2.5) e substituindo a mesma na equação do torque (T_1), obtemos uma nova equação para o torque de um dos motores.

$$\begin{aligned} i &= \frac{u}{R} - \frac{K_e}{R}(\dot{\alpha}_w - \dot{\theta}_b) \\ T_1 &= \frac{K_m}{R}u - \frac{K_e K_m}{R}(\dot{\alpha}_w - \dot{\theta}_b) \end{aligned} \quad (2.6)$$

A Equação (2.6) é a formulação matemática do torque de um dos motores CC que será utilizado nesse trabalho. Os termos $(\dot{\alpha}_w - \dot{\theta}_b)$ representam as rotações angulares relativas entre o rotor e o estator do motor.

Equações de movimento das rodas

As forças que atuam nas rodas do robô são apresentadas na Figura (2.3) (SCHILING, 2017). A partir dessa figura é possível obter as equações de movimento das rodas, utilizando a Segunda Lei de Newton.

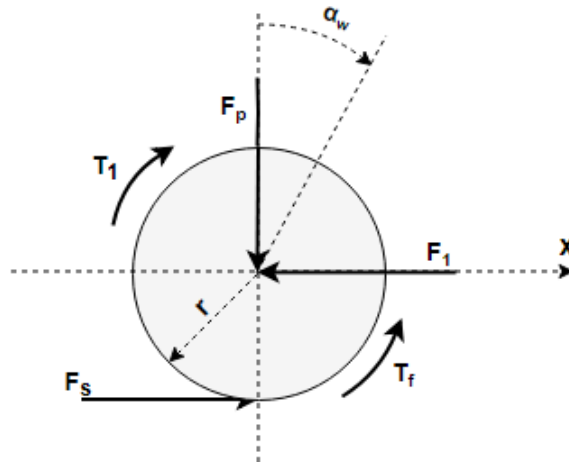


Figura 2.3: Diagrama corpo livre da roda.

Pela Figura (2.3) obtém-se as seguintes equações, como se segue

$$J_w \ddot{\alpha}_w = T_1 - rF_s - T_f \quad (2.7)$$

$$M_w \ddot{x}_w = F_s - F_1 \quad (2.8)$$

O termo J_w é o momento de inércia da roda com relação ao eixo do motor. O torque de atrito é dado por T_f , sendo proporcional a velocidade angular por uma constante de atrito b . Já a força de atrito F_s , é causada pelo contato da roda com chão, considerando que não há escorregamento.

A expressão da força de atrito (F_s) é obtida substituindo torque (T_1) da Equação (2.6) na Equação (2.7). Sendo assim, obtém-se a seguinte equação

$$F_s = -\frac{J_w}{r} \ddot{\alpha}_w - \frac{T_f}{r} + \frac{K_m}{rR} u - \frac{K_m K_e}{rR} (\dot{\alpha}_w - \dot{\theta}_b) \quad (2.9)$$

Considerando que não há escorregamento entre a roda e o solo, ou seja, a roda gira sem deslizar, pode-se expressar em termos do deslocamento no eixo x a rotação da roda, como se segue:

$$\ddot{x}_w = \ddot{\alpha}_w r$$

Assim, a expressão para a força (F_1), ou seja, a de um dos motores, é obtida fazendo a substituição da Equação (2.9) na Equação (2.8).

$$F_1 = -\left(M_w + \frac{J_w}{r^2}\right) \ddot{x}_w - \left(\frac{K_m K_e}{r^2 R}\right) \dot{x}_w + \left(\frac{K_m K_e}{rR}\right) \dot{\theta}_b + \left(\frac{K_m}{rR}\right) u - \frac{T_f}{r} \quad (2.10)$$

Na Equação (2.11), o termo $K_m K_e$ pode ser escrito como sendo uma constante K . Logo, a expressão para a força do motor 1 pode ser reescrita como sendo

$$F_1 = -\left(M_w + \frac{J_w}{r^2}\right) \ddot{x}_w - \left(\frac{K}{r^2 R}\right) \dot{x}_w + \left(\frac{K}{rR}\right) \dot{\theta}_b + \left(\frac{K_m}{rR}\right) u - \frac{T_f}{r}. \quad (2.11)$$

Equações de movimento da estrutura

Na Figura (2.4), é apresentado o diagrama de corpo livre da parte estrutural do pêndulo contendo todas as forças que atuam sobre o mesmo (BONAFILIA *et al.*, 2013).

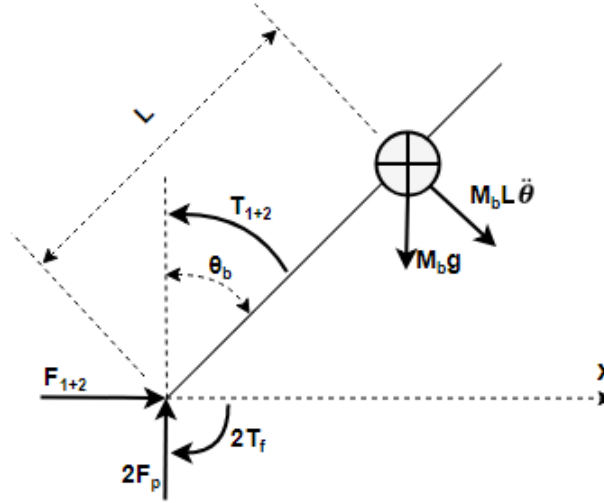


Figura 2.4: Diagrama corpo livre do chassi.

A equação de movimento do pêndulo com relação ao eixo horizontal, pode ser descrita como:

$$M_b \ddot{x}_b = (F_1 + F_2) - M_b L \cos(\theta_b) \ddot{\theta}_b \quad (2.12)$$

O último termo da Equação (2.12) é em relação a força tangencial na horizontal. O termo F_1 representa a força do motor da roda esquerda e a força F_2 representa a do motor da roda direita. Considerando que os motores são iguais, pode-se afirmar que tanto a força e o torque de cada motor são iguais.

$$F_1 + F_2 = 2F_1$$

$$T_1 + T_2 = 2T_1$$

Como o propósito do trabalho é estabilizar a estrutura na vertical, espera-se que o mesmo tenha pequenas variações. Sendo assim, pode-se simplificar as variáveis das equações em termos genéricos, tais como:

$$\begin{aligned} \ddot{x} &= \ddot{x}_b = \ddot{x}_w \\ \ddot{\theta} &= \ddot{\theta}_b \end{aligned}$$

Substituindo a expressão de F_1 na Equação (2.12) e re-ajeitando os termos, obtém-se a primeira equação de movimento do sistema como um todo em termos da aceleração linear \ddot{x} e aceleração angular $\ddot{\theta}$.

$$\begin{aligned} \left(2\frac{J_w}{r^2} + 2M_w + M_b\right) \ddot{x} &= 2\left(\frac{b}{r^2} - \frac{K}{r^2 R}\right) \dot{x} - (M_b L \cos(\theta)) \ddot{\theta} \\ &+ 2\left(\frac{K}{rR} - \frac{b}{r}\right) \dot{\theta} + 2\frac{K_m}{rR} u \end{aligned} \quad (2.13)$$

A Equação (2.14) refere-se a expressão de movimento em relação ao centro de massa (CM). Os termos que estão multiplicados por 2 na Equação (2.14) são referentes as forças de reação e de atrito que estão aplicadas em cada roda, tendo assumido que os motores são iguais.

$$J_b \ddot{\theta} = -2F_1 L \cos(\theta) + 2F_p L \sin(\theta) - 2T_1 + 2T_f \quad (2.14)$$

Com relação ao eixo perpendicular, a equação referente a esse eixo é descrita como:

$$M_b \cos(\theta) \ddot{x} = 2F_1 \cos(\theta) - 2F_p \sin(\theta) - M_b L \ddot{\theta} + M_b g \sin \theta \quad (2.15)$$

Analisando as Equações (2.14) e (2.15), percebe-se termos iguais. Sendo assim, possibilita-se que as mesmas sejam igualadas afim de encontrar uma segunda equação de movimento para o sistema como um todo.

$$(J_b + M_b L^2) \ddot{\theta} = -M_b L \cos(\theta) \ddot{x} + M_b g L \sin(\theta) + 2T_f - 2T_1 \quad (2.16)$$

Os torques T_f e T_1 da Equação (2.16) podem ser descritos por:

$$\begin{aligned} T_1 &= \frac{K_m}{R} u - \frac{K}{R} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \\ T_f &= b \left(\dot{\theta} - \frac{\dot{x}}{r} \right) \end{aligned}$$

Assim, a Equação (2.16) pode ser reescrita e assim, obtém-se uma outra equação de movimento para o sistema todo em termos da aceleração angular da estrutura $\ddot{\theta}$ e da roda \ddot{x} , como sendo:

$$\begin{aligned} (J_b + M_b L^2) \ddot{\theta} &= -M_b L \cos(\theta) \ddot{x} + 2 \left(\frac{K}{rR} - \frac{b}{r} \right) \dot{x} + 2 \left(b - \frac{K}{R} \right) \dot{\theta} \\ &\quad + M_b g L \sin(\theta) - 2 \frac{K_m}{R} u \end{aligned} \quad (2.17)$$

2.2.2 Abordagem do sistema em Espaço de Estados

A representação no espaço de estados garante uma maior facilidade de soluções de um modelo matemático e de um sistema físico, já que a mesma pode relacionar um conjunto de variáveis de entradas, saídas e de estados por meio de equações diferenciais de primeira ordem. A representação fornece de maneira prática as soluções do sistema, por adotar notações matriciais quando o sistema dinâmico é linear e invariante no tempo.

Caso contínuo no tempo

No caso contínuo no tempo, um sistema representado no espaço de estados de acordo com HESPANHA (2009) pode ser definido pelas seguintes equações:

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t)\end{aligned}\tag{2.18}$$

os sinais u , y e x das Equação (2.18) são chamados de entrada, saída e de estados respectivamente e definidos por:

$$u : [0, \infty] \rightarrow \mathbb{R}^k, \quad x : [0, \infty] \rightarrow \mathbb{R}^n, \quad y : [0, \infty] \rightarrow \mathbb{R}^m$$

Sendo o sistema representado pela Equação (2.18) linear e as matrizes $A(t)$, $B(t)$, $C(t)$ e $D(t)$ constantes $\forall t \geq 0$, o sistema é chamado de linear invariante no tempo (LTI). Assim, pode-se simplificar a representação no caso contínuo, como visto abaixo.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{2.19}$$

As dimensões das matrizes A , B , C e D das Equações (2.18) e (2.19) são as mesmas e são definidas como:

$$\begin{aligned}A &\in \mathbb{R}^{n \times n} & B &\in \mathbb{R}^{k \times m} \\ C &\in \mathbb{R}^{l \times n} & D &\in \mathbb{R}^{l \times m}\end{aligned}$$

Caso discreto no tempo

As aplicações de forma geral são feitas utilizando microcontroladores ou algum dispositivo digital. Sendo assim, para aplicações desse tipo, faz-se necessário a discretização do sistema.

De acordo com CHEN (1999), a representação discreta do sistema no espaço de estados são dados pelas equações abaixo.

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C_d x_k + D_d u_k\end{aligned}\tag{2.20}$$

Sendo que a variável $t = kT$ para $k = 0, 1, 2, \dots$, e o termo T é o período de amostragem do sistema. O período é escolhido de forma a ser o menor possível, ou seja, o máximo que o microcontrolador consegue trabalhar.

As matrizes A_d , B_d , C_d , D_d são descritas como se segue e a solução para as matrizes A_d e B_d pode ser vista em (CHEN, 1999, p. 91).

$$A_d = e^{AT} \quad B_d = \left(\int_0^T e^{A\tau} d\tau \right) B \quad C_d = C \quad D_d = D\tag{2.21}$$

2.2.3 Linearização

Uma gama de sistemas físicos reais conhecidos e estudados são de natureza não linear. Assim, para que as técnicas clássicas de controle ou mesmo algumas de controle moderno sejam utilizadas nos mesmos é preciso tratá-los por aproximações linearizantes.

Segundo OGATA (2010), um sistema para ser denominado de não linear é preciso que o princípio da superposição falhe ao ser aplicado nele. Dessa forma, caso tente obter uma resposta do sistema aplicando duas entradas simultâneas considerando as mesmas individualmente e somando-se os resultados, essa resposta não será válida.

Um sistema representado no espaço de estados é definido por HESPANHA (2009) da seguinte forma:

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= g(x, u)\end{aligned}\tag{2.22}$$

sendo $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ e $u \in \mathbb{R}^k$.

Ponto de equilíbrio

Para se obter uma aproximação linear de um sistema não linear, é preciso, primeiramente, determinar os pontos de equilíbrio do sistema. A determinação dos pontos de equilíbrio é feito quando se iguala o termo diferencial da Equação (2.22).

Pela definição de HESPANHA (2009), têm-se que um par de pontos $(x^{eq}, u^{eq}) \in \mathbb{R}^n \times \mathbb{R}^k$ que pode ser chamado de ponto de equilíbrio de (2.22) caso $f(x^{eq}, u^{eq}) = 0$. A solução de (2.22) é visto abaixo.

$$u(t) = u^{eq}, \quad x(t) = x^{eq}, \quad y(t) = y^{eq} := g(x^{eq}, u^{eq}), \quad \forall t \geq 0 \tag{2.23}$$

Linearização em torno de um ponto de equilíbrio

De acordo com HESPANHA (2009), a linearização do sistema em série de Taylor pode ser representado por matrizes Jacobianas, conforme descrito a seguir.

$$\begin{aligned}A &:= \frac{\partial f(x^{eq}, u^{eq})}{\partial x}, & B &:= \frac{\partial f(x^{eq}, u^{eq})}{\partial u}, \\ C &:= \frac{\partial g(x^{eq}, u^{eq})}{\partial x}, & D &:= \frac{\partial g(x^{eq}, u^{eq})}{\partial u}\end{aligned}\tag{2.24}$$

Assim, o sistema linear invariante no tempo (LTI), no espaço de estados, linearizado em torno de um ponto de equilíbrio (x^{eq}, u^{eq}) , é dado por:

$$\begin{aligned}\dot{\delta x} &= A\delta x + B\delta u \\ \delta y &= C\delta x + D\delta u\end{aligned}\tag{2.25}$$

Sendo que os termos $\delta\dot{x}(t)$, $\delta y(t)$ e $\delta u(t)$ para todo $t \geq 0$ é descrito como se segue.

$$\delta\dot{x}(t) := \dot{x}(t) - \dot{x}^{eq}, \quad \delta y(t) := y(t) - y^{eq}, \quad \delta u(t) := u(t) - u^{eq} \quad (2.26)$$

2.2.4 Realimentação de Estados

Uma das formas de se implementar controladores por meio da representação no espaço de estados é por meio da realimentação de estados, no qual utiliza-se os próprios estados do sistema com um fator multiplicativo (ganho) para gerar o sinal de controle. Para ter acesso a todos os estados da planta deve-se ter sensores para cada um dos estados do sistema. Como na maioria das aplicações reais isso não ocorre, pois implica em um custo elevado, faz-se necessário a implementação de um sistema que irá estimar os estados (observador) e assim, o sinal de controle passará a ser com relação a esses estados estimados. Contudo, para realizar a construção de um observador ou mesmo fazer o controle do sistema, deve-se, primeiramente, testar o sistema com métodos matemáticos que informam tais capacidades. Dito isso, os critérios são apresentados a seguir para um sistema discreto, tal como:

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C_d x_k + D_d u_k \end{aligned} \quad (2.27)$$

Controlabilidade

De acordo com HESPANHA (2009), para saber se o sistema é controlável ou não, utiliza-se as matrizes A_d e B_d da equação de estados de (2.27). O critério para determinar a controlabilidade é visto abaixo.

$$\mathbb{C} := \begin{bmatrix} B_d & A_d B_d & A_d^2 B_d & \dots & A_d^{n-1} B_d \end{bmatrix}_{n \times (kn)} \quad (2.28)$$

A partir do critério (2.28) pode-se determinar se o sistema em questão poderá ou não ser controlável. Segundo OGATA (2010), para que o sistema seja controlável é preciso que o determinante da matriz (2.28) seja diferente de 0, ou seja, que tenham vetores linearmente independentes (LI) e assim, posto n .

Observabilidade

Como não haverá medições diretas de todos os estados, será necessário a utilização de um observador como já mencionado. O mesmo irá estimar o valor dos estados que faltam e assim a lei de controle da realimentação será baseado neles.

Contudo, nem todos os sistemas são observáveis, dessa forma é preciso garantir que o sistema deste trabalho seja observável. De acordo com HESPANHA (2009), o critério no qual testa-se o sistema será observável ou não é visto em (2.29).

$$\mathbb{O} := \begin{bmatrix} C_d \\ C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{n-1} \end{bmatrix}_{(kn) \times n} \quad (2.29)$$

A análise da matriz (2.29) é dual ao critério de controlabilidade. Para que o sistema seja observável é preciso que os vetores sejam LI, ou seja, que tenham posto (*rank*) igual a n .

2.2.5 Controlador Linear Gaussiano Quadrático (LQG)

De acordo com BRUNTON *et al.* (2017), o controlador LQG é constituído por um estimador quadrático linear (LQE) ou mais popularmente denominado de Filtro de Kalman, ou seja, um observador de estados no qual os estados estimados são realimentados através de ganhos ótimos projetados por meio do regulador linear quadrático LQR. A Figura (2.5) apresenta como deve ser o *design* de um controlador LQG.

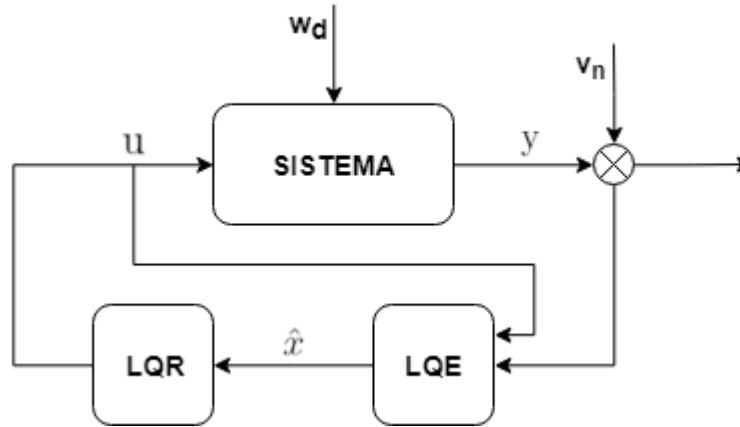


Figura 2.5: Estrutura de um controlador LQG.

Regulador Linear Quadrático (LQR)

O LQR possui como principal característica a garantia de estabilidade em malha fechada, um alto grau de robustez do sistema controlado proporcionado ao mesmo margem de ganho infinita e margem de fase $\pm 60^\circ$ (FERNANDES, 2014). A principal desvantagem desse controlador é a necessidade de saber todos os estados da planta. Entretanto, como na maioria das aplicações reais não se faz a medição de todos os estados possíveis, é necessário projetar um observador que irá por meio do valor da saída atual, estimar os próximos estados do sistema.

Para encontrar o ganho ótimo do LQR, considera-se o seguinte sistema no tempo discreto.

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C_d x_k \end{aligned} \quad (2.30)$$

Assume-se que $x(0) = x_0$ e $k = 0, 1, 2, \dots, k_f$. De acordo com SAGE (1977), a função de custo a ser minimizada é dada por:

$$J = \frac{1}{2} \|x_{k_f}\|_S^2 + \frac{1}{2} \sum_{k=0}^{k_f-1} \{ \|x_k\|_Q^2 + \|u_k\|_R^2 \} \quad (2.31)$$

A matriz $Q \in \mathbb{R}^{n \times n}$ é uma matriz positiva semi-definida e $R \in \mathbb{R}^{m \times m}$ é uma matriz positiva definida. Essas matrizes são responsáveis pela ponderação dos estados e o sinal de controle do sistema. De acordo com MATOS (2008), essas matrizes são escolhidas da seguinte forma:

$$\begin{aligned} Q &= C_d^T C_d \\ R &= \rho \mathbf{I}_{m \times m}, \quad \rho > 0. \end{aligned}$$

Conforme SAGE (1977), a equação Hamiltoniana é dada como se segue.

$$H_k = \frac{1}{2} x_k^T Q x_k + \frac{1}{2} u_k^T R u_k + \lambda_{k+1}^T [A_d x_k + B_d u_k] \quad (2.32)$$

Caso o par (A_d, B_d) seja controlável e o par (A_d, C_d) observável, a lei de controle ótimo que minimiza a equação de custo (2.31) é dada por:

$$u_k = -K x_k \quad (2.33)$$

A matriz $K \in \mathbb{R}^{m \times n}$ de ganhos ótimos do controlador LQR é definida como se segue

$$K = (R + B_d^T P B_d)^{-1} B_d^T P A_d \quad (2.34)$$

A matriz simétrica $P \in \mathbb{R}^{n \times n}$ é conseguida após solucionar a equação de Riccati de forma recursiva como segue:

$$P_{k+1} = Q + A_d^T P_k A_d - A_d^T P_k B_d (R + B_d^T P_k B_d)^{-1} B_d^T P_k A_d \quad (2.35)$$

Estimador Linear Quadrático (LQE)

A Figura (2.6) apresenta a estrutura do estimador (filtro de Kalman) no tempo discreto.

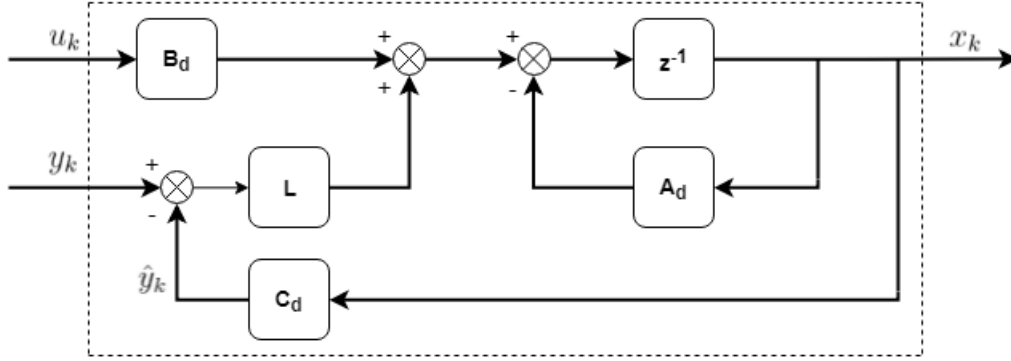


Figura 2.6: Estrutura do Filtro de Kalman discreto em Espaço de Estados. Adaptado de BRUNTON *et al.* (2017).

O ganho do estimador ótimo quadrático linear se baseia no desempenho do observado projetado com a presença de erro de medição e ruído de processo. Logo, a cada tempo que passa o ganho do filtro de Kalman vai sendo atualizado. Para realizar o projeto desse estimador, primeiramente, considera-se o seguinte sistema.

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k + w_k \\ y_k &= C_d x_k + v_k \end{aligned} \quad (2.36)$$

em que o ruído w_k e o erro de medição v_k são sequências aleatórias de ruído branco gaussiano com média zero (SAGE, 1977, p. 231). Sendo assim, temos:

$$\mathbb{E}\{w_k\} = 0 \quad \text{e} \quad \mathbb{E}\{v_k\} = 0$$

em que $\mathbb{E}\{\}$ é conhecido como expectância ou valor esperado. A correlação por se tratar de ruído branco é zero

$$\text{cov}\{w_i w_j^T\} = 0 \quad \text{e} \quad \text{cov}\{v_i v_j^T\} = 0 \quad \text{para} \quad i \neq j$$

e a covariância é definida por

$$\text{cov}\{w_k w_k^T\} = Q_o \quad \text{e} \quad \text{cov}\{v_k v_k^T\} = R_o$$

A principal tarefa da otimização é determinar uma matriz de ganhos que minimizará a variância da estimação do erro, que é denotado por G_k :

$$G_k = \mathbb{E}\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\} \quad (2.37)$$

Segundo BRUNTON *et al.* (2017), um estimador produz uma estimação dos estados x_k conhecida como \hat{x}_k caso conheça a saída medida y_k provinda de um sensor e o sinal de entrada u_k . Caso o sistema seja observável, é possível construir o filtro a partir do ganho L como se segue

$$\begin{aligned}\hat{x}_{k+1} &= A_d \hat{x}_k + B_d u_k + L_k (y_k - \hat{y}_k) \\ \hat{y}_k &= C_d \hat{x}_k\end{aligned}\tag{2.38}$$

sendo que a saída \hat{y}_k é a predição da expectativa da saída do sensor baseada no estados estimados \hat{x}_k . Manipulando a Equação (2.38), encontra-se uma expressão para \hat{x}_{k+1} .

$$\begin{aligned}\hat{x}_{k+1} &= (A_d - LC_d) \hat{x}_k + B_d u_k + L y_k \\ &= (A_d - LC_d) \hat{x}_k + \begin{bmatrix} B_d & L \end{bmatrix} \begin{bmatrix} u_k \\ y_k \end{bmatrix}\end{aligned}\tag{2.39}$$

assim, as matrizes A_f e B_f do filtro de Kalman são:

$$A_f = (A_d - LC_d) \quad B_f = \begin{bmatrix} B_d & L \end{bmatrix}\tag{2.40}$$

De acordo com KIM *et al.* (2013), o ganho L do filtro de Kalman é um vetor pertencente aos números reais e de dimensão $n \times 1$ e é definido pela seguinte expressão:

$$L = A_d S_k C_d^T (C_d S_k C_d^T + R_o)^{-1}\tag{2.41}$$

A matriz S é obtida quando soluciona-se a equação de Riccati para a equação de diferenças. A expressão que precisa-se solucionar e que fará com que o ganho L seja ótimo é vista abaixo:

$$S_{k+1} = Q_o + A_d S_k A_d^T - A_d S_k C_d^T (C_d S_k C_d^T + R_o)^{-1} C_d S_k A_d^T\tag{2.42}$$

em que Q_o é uma matriz positiva definida e $R_o > 0$ são especificados como parâmetros de projeto, considerando a condição inicial $S_0 = 0$.

Metodologia

Inicialmente, realizou-se a revisão de literatura e a fundamentação teórica a fim de obter informações a respeito de trabalhos que já foram desenvolvidos na área e, conhecer as limitações e facilidades pertinentes do projeto. Assim, atentou-se no conhecimento e domínio sobre o sistema, tais como seus pontos de operação e os desafios da implementação no sistema físico. Em seguida, foi necessário realizar o projeto estrutural da planta no SolidWorks. Dessa maneira, com a concepção da estrutura do sistema foi possível aferir algumas propriedades mecânicas importantes para a modelagem matemática. Após a obtenção do modelo não linear através de EDOs, foi necessário a realização da linearização e discretização do sistema. Em posse do modelo linear e na forma de espaço de estados, projetou-se os ganhos do controlador LQR e do estimador LQE e que quando fundidos, torna-se o controlador LQG. Sendo assim, simulou o sistema não linear juntamente do controlador obtido utilizando o MATLAB e o Simulink como *softwares*.

A Figura (3.1) apresenta todos os passos que pretende percorrer para a construção de um protótipo e o controle do mesmo. Dessa maneira, precisou-se dividir as tarefas entre TCC1 e TCC2 como visto no fluxograma de atividades.

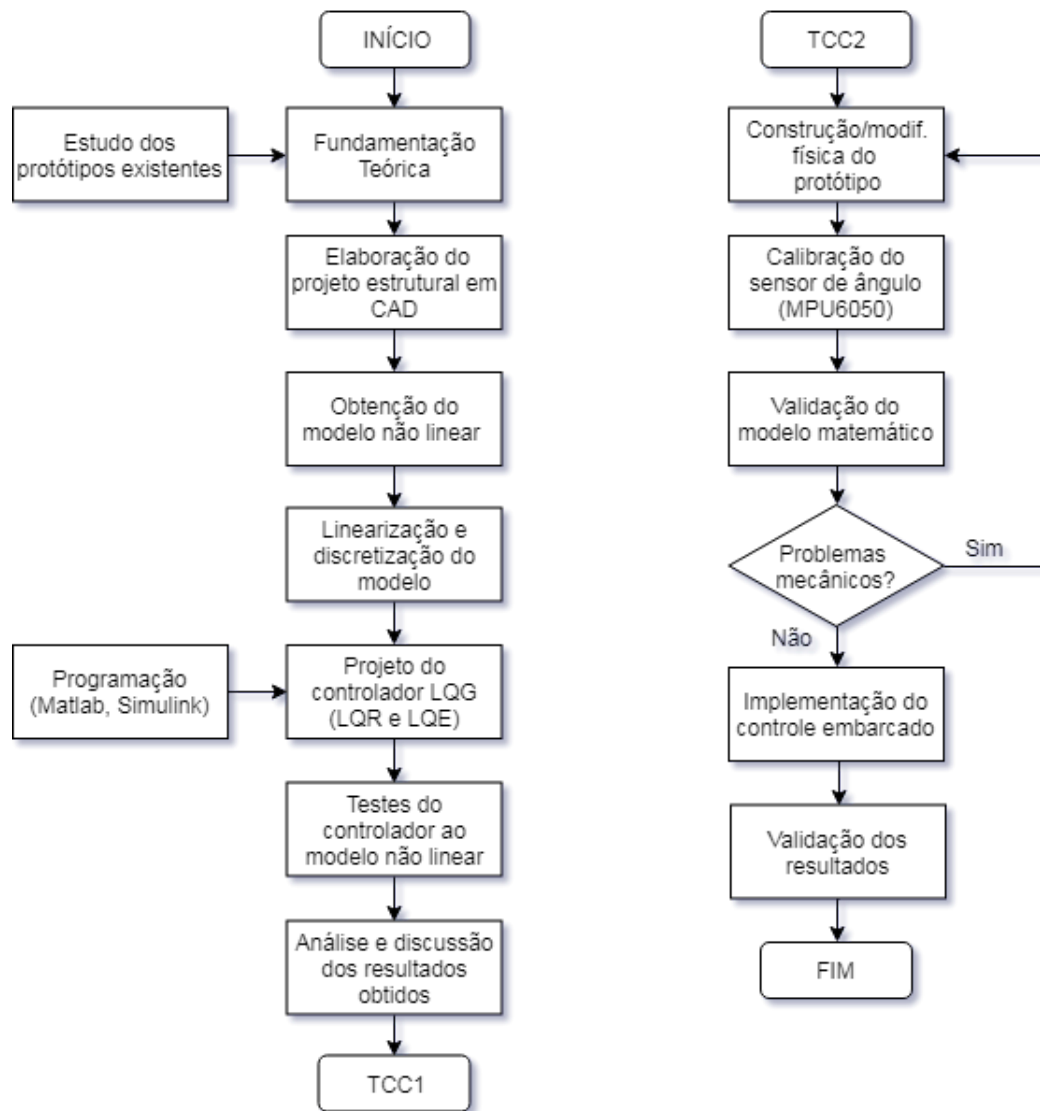


Figura 3.1: Fluxograma das Atividades desenvolvidas no TCC.

3.1 Estrutura Adotada

A Figura (3.2) apresenta o conceito inicial da estrutura do protótipo. Esse tipo de formato com andares para acomodar melhor os componentes eletrônicos aparece frequentemente nas literaturas abordadas. Em um primeiro momento não se preocupou com as dimensões exatas do protótipo e assim, modelou-o de acordo com os componentes que ficarão em cada andar. A única consideração a ser feita e que será avaliada posteriormente, é a posição que deve ficar o centro de massa (CM). A posição do CM está ligada diretamente na estabilidade do sistema, facilitando ou dificultando o controle da estrutura.

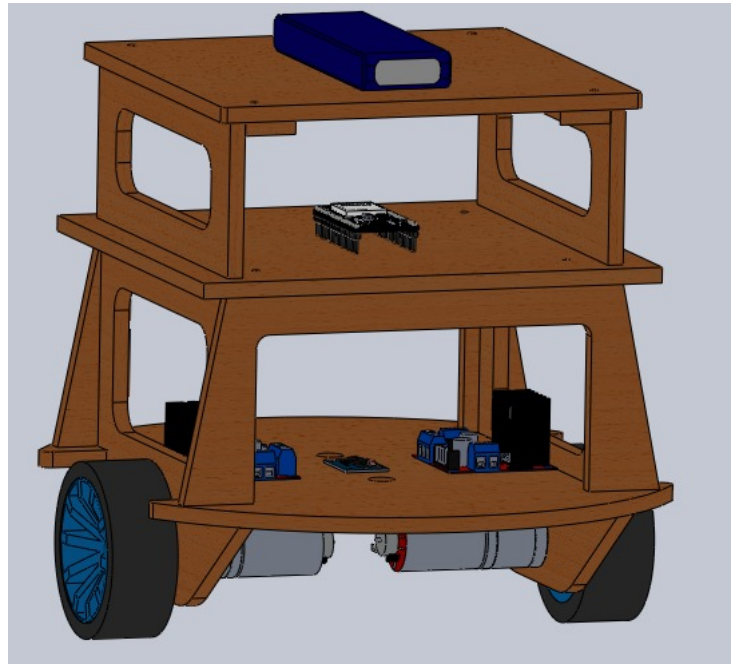


Figura 3.2: Estrutura modelada no SolidWorks.

Com auxílio do SolidWorks, foi possível mensurar algumas unidades da estrutura. A Tabela (3.1) apresenta os valores dessa variáveis.

Tabela 3.1: Parâmetros mecânicos da estrutura.

Parâmetro	Valor	Unidade	Descrição
M_b	1.017	kg	Massa total do protótipo
L	0.096331	m	Distância do eixo dos motores ao CG da estrutura
J_b	0.0054715	$kg.m^2$	Momento de inércia da estrutura

3.2 Recursos Utilizados

No momento em que se inicia o planejamento de um projeto de engenharia, é de suma importância realizar a listagem dos recursos que serão utilizados, bem como a justificativa de cada. Desta forma, segue abaixo a listagem dos *softwares* e materiais que foram ordenados da seguinte maneira: atuadores, transdutor de orientação e eletrônica.

3.2.1 Softwares

Os *softwares* aqui listados foram utilizados para a concepção do conceito 3D do projeto e programação do código para o controle do sistema. São eles: SolidWorks e o MATLAB com a utilização do Simulink.

SolidWorks

O SolidWorks é um *software* de CAD 3D desenvolvido inicialmente pela SolidWorks Corporation que funciona no sistema operativo Windows. Basicamente, é utilizado para modelagem de peças sólidas e para simulação de processos. A versão 2016, em que a concepção do conceito deste projeto foi realizado, possui funções de otimização de simulação e de modelos de sólidos básicos pré-feitos. Além dessas funções de otimização, o também *software* é capaz de fornecer as propriedades mecânicas, tais como: momento de inércia de massa, centro de massa, etc.

MATLAB

O MATLAB é um *software* interativo de alta performance voltado para o cálculo numérico. Realiza a análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente fácil, sendo programado com linguagem matemática ao invés da programação tradicional.

O *software* conta com inúmeras extensões, mas com destaque para o Simulink. Essa extensão conta com uma programação baseada em diagrama de blocos e com esse *toolbox* é possível realizar as simulações dos circuitos de uma forma que fique mais simples e de fácil visualização.

3.2.2 Atuadores

A Figura 3.3 mostra os dispositivos responsáveis por acionarem um dos lados do pêndulo: motor, acoplador e roda.



Figura 3.3: Conjunto completo do atuador: motor, acoplador e roda (BAÚ DA ELETRÔNICA, 2019).

A partir da folha de dados do fabricante é possível obter algumas características dos motores. Essas características são listadas como se segue:

- Tensão de operação: $6 V_{cc}$;

- Máxima velocidade: 210 RPM;
- Máxima eficiência: 2.0 kg.cm/170 RPM/2.0 W/0.60 A;
- Máximo torque: 5.2 kg.cm/110 RPM/3.1 W/1.10 A;
- Encoder integrado;

Com relação a pinagem e como estão dispostos os cabos, pode-se afirmar que: 2 fios são para a alimentação do motor e os outros 4 fios para o encoder (2 fios de alimentação em 3.3V e 2 fios para sinal).

Algumas propriedades mecânicas, como o momento de inércia e massa da roda, constante de torque e da força contra eletromotriz do motor podem ser vistas na Tabela (3.2).

Tabela 3.2: Parâmetros do motor e da roda.

Parâmetro	Valor	Unidade	Descrição
R_m	7	Ω	Resistência elétrica do motor
K_m	0.3922	Nm/A	Constante de torque do motor
K_e	0.1404	Vs/rad	Constante da força contra eletromotriz
M_w	0.0844	kg	Massa de cada roda
r	0.325	m	Raio da roda
J_r	0.004457	$kg.m^2$	Momento de inércia da roda

3.2.3 Transdutor de Orientação

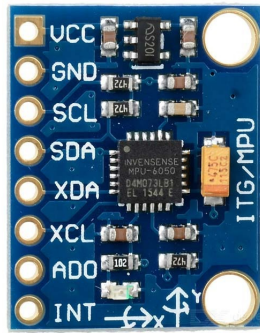
O dispositivo para medir a orientação da estrutura será composto por dois sensores, acelerômetro e giroscópio, que estão embutidos na placa MPU6050.

Placa MPU6050

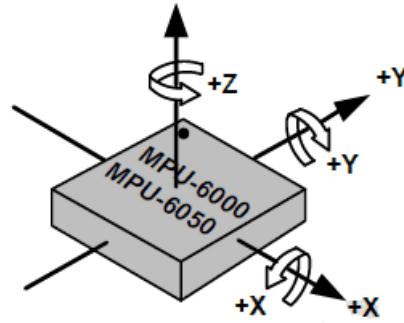
Essa *shield* possui dois sensores, acelerômetro e giroscópio, dos quais serão fundidos a fim de obter uma melhor precisão da medição da orientação da estrutura do pêndulo. A Figura (3.4) apresenta a placa física e um modelo que representa todos os 6 eixos, 3 para o acelerômetro e 3 para o giroscópio, o que faz com a mesma possua 6 graus de liberdade (*6DOF*). A lista mostrada logo abaixo apresenta algumas das características dessa placa.

- Tensão de operação: $3 \sim 5 V_{cc}$;
- Modelo: GY-521;
- Conversor AD: 16 bits;
- Comunicação: I2C;
- Tempo de acomodação: 10 ms (máx); e

- Dimensões (CxLxA): 21 x 17 x 3 mm;



(a)



(b)

Figura 3.4: *Shield* MPU6050 (INVENSENSE, 2019).

O acelerômetro é um equipamento que é utilizado para mensurar sua aceleração própria. A aceleração própria é diferente daquela estabelecida através da relação entre velocidade e tempo. Sendo que esta considera a sensação de peso medida em um dado referencial. Quando considera-se o acelerômetro colocado em uma superfície plana, a medição será de aproximadamente $9.81m/s^2$. Na maioria dos casos, a aceleração é medida em força g, que é basicamente a aceleração sentida como peso. A Figura (3.5) mostra a inclinação de algum objeto utilizando dois eixos.

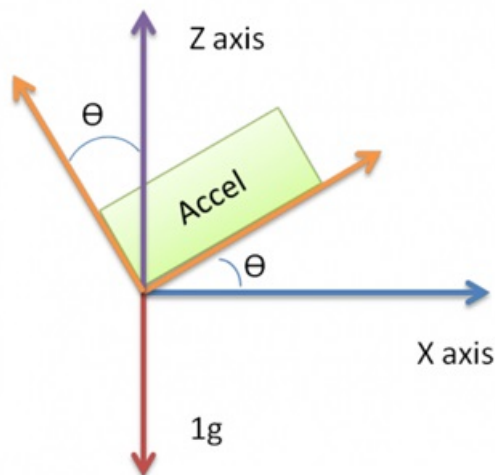


Figura 3.5: *Shield* MPU6050.

A medição da orientação pode ser feita de três maneiras: utilizando apenas um eixo, dois e três eixos. A forma mais simples é a primeira, porém a mais eficiente em termos de

precisão é a última. As equações são mostradas logo abaixo:

$$\begin{aligned}\theta &= \sin^{-1}(x) \\ \theta &= \arctan\left(\frac{x}{z}\right) \\ \theta &= \arctan\left(\frac{x}{\sqrt{z^2 + y^2}}\right)\end{aligned}\tag{3.1}$$

O giroscópio por sua vez, é um dispositivo utilizado para manter ou medir a orientação. Um giroscópio mecânico normalmente consiste de um disco rotativo, em que os eixos ligados a ele são capazes de se deslocarem livremente em qualquer orientação.

O valor lido da MPU6050 sem nenhum tratamento pode ser chamado de valor bruto já que não representa nenhuma grandeza física e para realizar a conversão para um valor usual, é necessário saber/configurar o quão rápido o sensor está girando em graus por segundo (dps). A Tabela (3.3), com conteúdo encontrado no *datasheet* do fabricante, relaciona cada velocidade de giro do sensor com a sua sensibilidade, sendo que essas constantes são chamadas de ganho e devem ser divididas por 1000.

Tabela 3.3: Parâmetros do giroscópio: velocidade de giro e nível de sensibilidade.

Faixa (°/s)	Sensibilidade LSB/(°/s)
±250	131
±500	65.5
±1000	32.8
±2000	16.4

3.2.4 Eletrônica

Nessa etapa do projeto será descrito de forma breve os itens que compõe o tópico de eletrônica.

ESP32

O ESP32, Figura (3.6), é um microcontrolador de baixo custo, com alta performance, com módulo Wi-Fi e *Bluetooth* já previamente embutido na placa. O seu funcionamento é muito semelhante ao Arduino, bastando apenas conectar um cabo USB, que o mesmo irá alimentá-lo e por fim, programar o código que deseja.

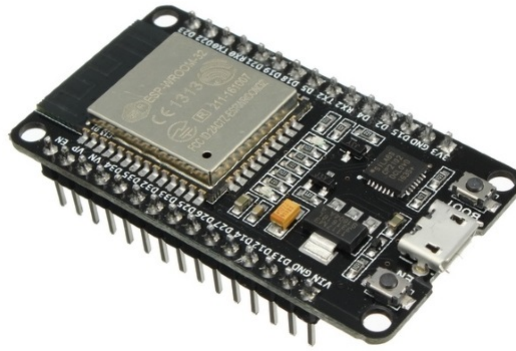


Figura 3.6: Microcontrolador ESP32 (FERNANDO K, 2019).

A programação deste microcontrolador pode ser feita através da própria IDE do Arduino, facilitando ainda mais a migração para o ESP32. A partir do *datasheet* da placa, é possível conhecer as características principais que a compõe, como visto abaixo:

- Wi-Fi embutido: padrão 802.11 B/G/N, operando na faixa de 2.4 a 2.5 GHz;
- *Bluetooth* v4.2 BR / EDR e BLE (*Bluetooth Low Energy*);
- Microprocessador dual core Tensilica Xtensa 32-bit LX6;
- Clock ajustável de 80MHz até 240MHz;
- Tensão de operação: 3.3 Vcc;
- Memória SRAM de 512 KB e ROM de 448 KB;
- Corrente máxima por pino: 12 mA; e
- Massa: 7g.

A escolha da utilização desse microcontrolador ao invés do Arduino ou mesmo algum outro mais potente como o *Raspberry Pi*, se deu pelos seguintes motivos: baixo custo de aquisição, quantidade de GPIO correspondentes para o projeto, Wi-Fi e *Bluetooth* embutido no chip e velocidade de processamento elevada, podendo ainda utilizar os dois núcleos simultaneamente. A Tabela (3.4), mostra a comparação entre alguns parâmetros do ESP32 com o seu predecessor, o ESP8266 e com o Arduino UNO R3.

A partir da Tabela (3.4), fica claro o porquê da escolha pela utilização do ESP32 ao invés de seus concorrentes. Uns dos itens que foram considerados e não estão na tabela é o preço, que é uns dos fatores determinantes. O ESP8266 e o Arduino UNO são um pouco mais baratos que o ESP32, contudo não são completos como visto.

Tabela 3.4: Comparativo entre ESP32, ESP8266 e Arduino Uno R3

	ESP32	ESP8266	ARDUINO UNO
Núcleos	2	1	1
Arquitetura	32 Bits	32 Bits	8 Bits
Clock	160 MHz	80 MHz	16 MHz
Wi-Fi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512 KB	160 KB	2 KB
Flash	16 MB	16 MB	32 KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

Driver Motor CC

A Figura (3.7) apresenta o drive do motor CC que será utilizado, que é a Ponte H L298N. Com essa placa será possível controlar o sentido de giro e realizar o controle da velocidade dos dois motores por meio de sinais PWM (*Pulse Width Modulation*). As características dessa placa se encontram na folha de dados do fabricante e algumas importantes a serem consideradas são listadas abaixo:

- Tensão de operação: 4~35 V_{cc} ;
- Corrente de operação máxima: 2 A por canal ou 4 A máx;
- Tensão lógica: 5 V;
- Corrente lógica: 0~36 mA;
- Dimensões: 43 x 43 x 27 mm; e
- Massa: 30g;

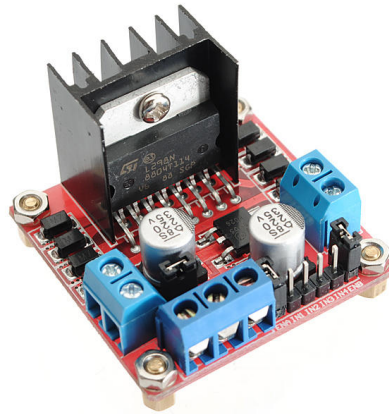


Figura 3.7: Ponte H L298N (VIDA DE SILÍCIO, 2019).

Como mencionado, com essa placa realiza-se duas coisas importantes, mudança no sentido de giro e alteração da velocidade dos motores. O primeiro caso é realizado mudando o sentido da corrente, pois os motores CC giram a partir de um campo magnético que é provocado por uma corrente que passa em suas bobinas.

Para controlar a velocidade do motor, é necessário recorrer ao PWM, que nada mais é do que uma técnica que obtém resultados analógicos por meio digitais. A técnica consiste em ondas quadradas de alta frequência das quais é possível controlar a porcentagem que o tempo em nível alto ou baixo. Esse tempo é chamado de *Duty Cycle* e ao utilizar desse recurso, o valor médio de tensão aplicado aos terminais do motor é alterado.

A partir da Figura (3.8), é possível obter a equação do *Duty Cycle* (DC) e da tensão média que será de fato o valor aplicado aos motores.

$$\begin{aligned} DC(\%) &= (x/T) * 100 \\ V_{medio} &= V_{max} * DC \end{aligned} \quad (3.2)$$

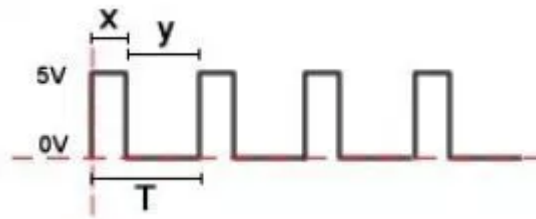


Figura 3.8: Sinal PWM (VIDA DE SILÍCIO, 2019).

Bateria

A escolha da bateria de Lipo de 2200mAh, como visto na Figura (3.9), passou por alguns critérios, como o tempo de funcionamento e capacidade de fornecimento de energia. As características mais importantes encontradas no *datasheet* do fabricante, é visto abaixo:

- Tensão de carga: 3 células / 11.1 V_{cc} ;
- Capacidade: 2200 mAh;
- Descarga: 20-30 C;
- Dimensões (CxLxA): 105 x 35 x 25 mm;
- Massa: 180g (incluindo cabo, plug e rótulo)



Figura 3.9: Bateria Lipo Turnigy 2200 mAh 3s (MERCADO LIVRE, 2019).

O item mais relevante a considerar é se a mesma consegue fornecer a capacidade necessária de energia para o sistema. A relação de capacidade de uma bateria está ligada diretamente com a taxa de descarga, que nesse caso é de 20-30C. O cálculo é simples, e pode ser realizado da seguinte maneira:

$$\text{Capacidade (A)} = 2.2\text{Ah} * 20\text{C} = 44\text{A} \quad (3.3)$$

Assim, pela Equação (3.3), essa bateria suporta fornecer até 44A em teoria quando solicitada. Geralmente, se trabalha com uma margem de 50% para preservá-la.

Com relação ao tempo de funcionamento, basta conhecer o valor de consumo de cada dispositivo que compõe o sistema. Para esse projeto, pode-se considerar apenas o consumo dos motores, Tabela (3.5), já que a corrente drenada pelos outros eletrônicos são irrelevantes perante aos atuadores. Em teoria e considerando tudo em perfeitas condições, é possível obter 1 (uma) hora de funcionamento utilizando o torque máximo ou 110 min usando a máxima eficiência dos motores em descarga contínua.

Tabela 3.5: Consumo dos Motores CC.

Dispositivo	Máx. Eficiência	Máx. Torque
Motores	1.2 A	2.2 A

3.2.5 Custo dos Materiais

A concepção deste projeto só será possível quando todos os equipamentos descritos acima forem de fato adquiridos. Contudo, é necessário realizar uma pesquisa de mercado e avaliar os melhores preços, ou mesmo aproveitar equipamentos. A Tabela (3.6), enumera o custo de cada dispositivo que será utilizado no projeto considerando o frete.

Tabela 3.6: Tabela de custo dos materiais utilizados na construção do protótipo.

Descrição	Un.	Qtd.	Valor Un.	Valor Total	Adquirido (S/N)
ESP32	pç	1	R\$ 62,90	R\$ 62,90	N
Kit Motor CC	pç	2	R\$ 119,90	R\$ 239,80	S
Driver L298 H	pç	1	R\$ 19,90	R\$ 19,90	S
MPU6050	pç	1	R\$ 16,90	R\$ 16,90	S
Bateria Lipo 2200mAh	pç	1	R\$ 90,00	R\$ 90,00	N
Carregador Lipo Imax B6	pç	1	R\$ 110,00	R\$ 110,00	N
Estrutura MDF	-	1	R\$ 45,00	R\$ 45,00	N
Miscelaneas	-	1	R\$ 58,00	R\$ 58,00	-
TOTAL				R\$ 632,50	-

O item denominado de Miscelaneas da lista de materiais remete aqueles materiais que não há necessidade de contagem, como parafusos, porcas, jumpers, etc. Para encontrar seu valor, considerou-se 10% do valor total de toda a lista.

Projeto de Controladores

Este capítulo abordará a realização do projeto dos controladores e a análise dos resultados alcançados sobre o modelo não linear do sistema físico. Sendo assim, primeiramente, é necessário encontrar o modelo do pêndulo invertido sobre duas rodas utilizando a teoria vista no Capítulo 2. O código feito no MATLAB e utilizado para o projeto dos controladores é visto no Apêndice A.

4.1 Modelo Não Linear Contínuo no Tempo

As EDOs do sistema surgiram a partir das deduções realizadas no Capítulo 2 como já mencionado. A partir das Equações (2.13) e (2.17), vistas logo abaixo, por meio de manipulações algébricas gerou-se duas EDOs. E de forma lógica as outras duas foram obtidas.

$$\left(2\frac{J_w}{r^2} + 2M_w + M_b\right)\ddot{x} = 2\left(\frac{b}{r^2} - \frac{K}{r^2R}\right)\dot{x} - (M_bL \cos(\theta))\ddot{\theta} + 2\left(\frac{K}{rR} - \frac{b}{r}\right)\dot{\theta} + 2\frac{K_m}{rR}u$$

$$(J_b + M_bL^2)\ddot{\theta} = -M_bL \cos(\theta)\ddot{x} + 2\left(\frac{K}{rR} - \frac{b}{r}\right)\dot{x} + 2\left(b - \frac{K}{R}\right)\dot{\theta} + M_bgL \sin(\theta) - 2\frac{K_m}{R}u$$

Analisando o sistema em espaço de estados como todo, visualiza-se que o mesmo poderá ter até 4 estados: x , \dot{x} , θ , $\dot{\theta}$. Contudo, para uma solução mais genérica, as variáveis serão tratadas como sendo:

$$\begin{aligned} x_1 &= x & x_3 &= \theta \\ x_2 &= \dot{x} & x_4 &= \dot{\theta} \end{aligned}$$

O modelo não linear da planta será composto por quatro equações diferenciais contínuas no tempo que descrevem o movimento do sistema. Para simplificar e melhorar a visualização das EDOs, utilizou-se de letras gregas para agregar termos em comum. As 4 equações são

vistas logo abaixo:

$$f_1 = \dot{x}_1 = x_2 \quad (4.1)$$

$$f_2 = \dot{x}_2 = \left(\frac{\omega}{r\Omega}\right) x_2 - \left(\frac{\omega}{\Omega}\right) x_4 + \left(\frac{\psi}{\Omega}\right) + \left(\frac{\Delta}{\Omega}\right) u \quad (4.2)$$

$$f_3 = \dot{x}_3 = x_4 \quad (4.3)$$

$$f_4 = \dot{x}_4 = \left(\frac{v}{\Omega}\right) x_4 - \left(\frac{v}{r\Omega}\right) x_2 + \left(\frac{\zeta}{\Omega}\right) + \left(\frac{\Phi}{\Omega}\right) u \quad (4.4)$$

sendo que os termos implícitos nas Equações (4.2) e (4.4) são definidos como visto abaixo:

$$\omega = 2r(R_mb - K_m K_e)[J_b + M_b L^2 + M_b L r \cos(x_3)]$$

$$\psi = (-M_b^2 g L^2 r^2) \sin(x_3) \cos(x_3)$$

$$\Delta = 2K_m r[J_b + M_b L^2 + M_b L r \cos(x_3)]$$

$$v = 2(R_mb - K_m K_e)[2J_w + M_b L r \cos(x_3) + r^2(2M_w + M_b)]$$

$$\zeta = M_b g L(2J_w + 2M_w r^2 + M_b r^2) \sin(x_3)$$

$$\Phi = -2K_m[2J_w + M_b L r \cos(x_3) + r^2(2M_w + M_b)]$$

$$\Omega = R_m[2(J_w + M_w r^2)(J_b + M_b L^2) + J_b M_b r^2 + M_b^2 L^2 r^2(1 - \cos^2(x_3))]$$

A equação que define a saída do sistema é dada por:

$$g_1 = x_3 \quad (4.5)$$

4.2 Modelo Linear Discreto no Tempo

Os controladores que serão utilizado serão lineares, assim é preciso que o modelo que é não linear se torne linear. A técnica para linearizar e obter as matrizes do espaço de estados pode ser vista na Equação (2.24). Dessa forma, para aplicar esse método nas funções de movimento, (4.1) a (4.4) e na função de saída, (4.5), define-se os pontos de equilíbrio (x^{eq}, u^{eq}) como sendo:

$$\begin{aligned} x^{eq} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ u^{eq} &= 0 \end{aligned} \quad (4.6)$$

Aplicando o método de linearização visto em (2.24) em torno do ponto de equilíbrio (4.6), encontra-se as seguintes matrizes lineares:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \sigma & \tau & -r\sigma \\ 0 & 0 & 0 & 1 \\ 0 & -\lambda & \Gamma & r\lambda \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \alpha \\ 0 \\ \beta \end{bmatrix} \quad (4.7)$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \quad D = 0 \quad (4.8)$$

Os termos implícitos das matrizes de estados A e B são definidos como se segue:

$$\begin{aligned} \delta &= R_m[2(J_w + M_w r^2)(J_b + M_b L^2) + J_b M_b r^2] \\ \sigma &= \frac{2(R_m b - K_m K_e)[J_b + M_b L^2 + M_b L r]}{\delta} \\ \tau &= \frac{-M_b^2 g L^2 r^2}{2(J_w + M_w r^2)(J_b + M_b L^2) + J_b M_b r^2} \\ \lambda &= \frac{2K_m r[2J_w + M_b L r + r^2(2M_w + M_b)]}{r\delta} \\ \Gamma &= \frac{M_b g L(2J_w + 2M_w r^2 + M_b r^2)}{\delta} \\ \alpha &= \frac{2K_m r(J_b + M_b L^2 + M_b L r)}{\delta} \\ \beta &= \frac{-2K_m[2J_w + M_b L r + r^2(2M_w + M_b)]}{\delta} \end{aligned}$$

Antes de iniciar o projeto dos controladores, encontrando os ganhos ótimos tanto do LQR quanto do estimador LQE, é necessário encontrar os valores numéricos das matrizes A e B. Dessa maneira, substitui os valores dos parâmetros da estrutura, motor e da roda que podem ser vistos nas Tabelas (3.1) e (3.2). Após a substituição de cada parâmetro que aparece nos termos das A e B, temos como solução o seguinte resultado:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.5560 & -10.0814 & 0.1807 \\ 0 & 0 & 0 & 1 \\ 0 & 6.0751 & 18.6727 & -1.9744 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1.7259 \\ 0 \\ -18.8569 \end{bmatrix} \quad (4.9)$$

Em posse das matrizes A, B e C testou-se a controlabilidade e a observabilidade, par (A,B) e (A,C) respectivamente. Sendo assim, utilizando o método visto em (2.28) e (2.29) ou os comandos do MATLAB (*ctrb* e *obsv*) foi possível avaliar o posto das matrizes, que no caso devem ser igual a 4. Os resultados encontrados foram os seguintes:

$$\begin{aligned} \text{rank}(\text{ctrb}(A,B)) &= 4 \\ \text{rank}(\text{obsv}(A,C)) &= 3 \end{aligned}$$

Diante do resultado, presumiu-se que não era possível obter todos os estados do sistema. Contudo, percebeu-se que para este projeto não está preocupado com a posição linear do protótipo, o que faz com que o estado x seja desnecessário. Dessa forma, trabalhará agora com um subsistema que tem apenas 3 estados: x_2 , x_3 e x_4 .

Discretização das Matrizes Lineares

Com as matrizes numéricas A e B é possível realizar a discretização. Utilizando a forma vista em (2.21) ou mesmo usando o comando do MATLAB (*c2d*) utilizando uma taxa de amostragem T_s de 15ms, que é um tempo suficientemente grande para que o sensor retorne a medida do ângulo lida e processada e o código seja compilado. Dessa forma, as matrizes discretas A_d e B_d encontradas foram:

$$A_d = \begin{bmatrix} 0.9918 & -0.1503 & 0.0015 \\ 0.0007 & 1.0020 & 0.0148 \\ 0.0895 & 0.2694 & 0.9730 \end{bmatrix} \quad B_d = \begin{bmatrix} 0.0255 \\ -0.0021 \\ -0.2777 \end{bmatrix} \quad (4.10)$$

De (2.21) é visto que as matrizes $C_d = C$ e $D_d = D$. Com as matrizes A_d , B_d e C_d testou-se agora a controlabilidade e observabilidade do subsistema. Agora, o resultado esperado do *rank* é que seja igual a 3. Testando por meio do MATLAB, o resultado foram os seguintes:

$$\text{rank}(\text{ctrb}(A_d, B_d)) = 3$$

$$\text{rank}(\text{obsv}(A_d, C_d)) = 3$$

Dessa maneira, agora é possível realizar o projeto do controlador ótimo e aplicá-lo ao modelo não linear.

4.3 Projeto dos Controladores

Com as 4 matrizes do espaço de estados linearizadas e discretas, é possível obter os valores de ganhos ótimos do LQR e o ganho do estimador, bem como sua implementação, que será feita utilizando o Simulink.

4.3.1 Controlador LQR

O controlador LQR é uma técnica de controle ótimo como já mencionado. Dessa maneira, busca-se minimizar a função de custo J , que tem como elementos as matrizes Q que ponderam os estados x e a matriz R que pondera o sinal de controle. Sendo assim, em um primeiro momento, escolheu-se as matrizes da seguinte maneira:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = 1 \quad (4.11)$$

Dessa forma, para obter o vetor de ganhos do LQR, utilizou-se o comando *dlqr* do MATLAB passando como parâmetros as matrizes A_d , B_d , Q e R . Este comando irá retornar tanto o vetor de ganhos ótimos K_r quanto a solução da equação de Riccati P .

$$K_r = [\ 0.5721 \quad -5.0411 \quad -0.9402 \] \quad (4.12)$$

$$P = \begin{bmatrix} 42.4936 & -80.6491 & 1.2268 \\ -80.6491 & 389.8724 & 12.9644 \\ 1.2268 & 12.9644 & 4.4682 \end{bmatrix} \quad (4.13)$$

A Figura (4.1) mostra a implementação do regulador feita no Simulink.

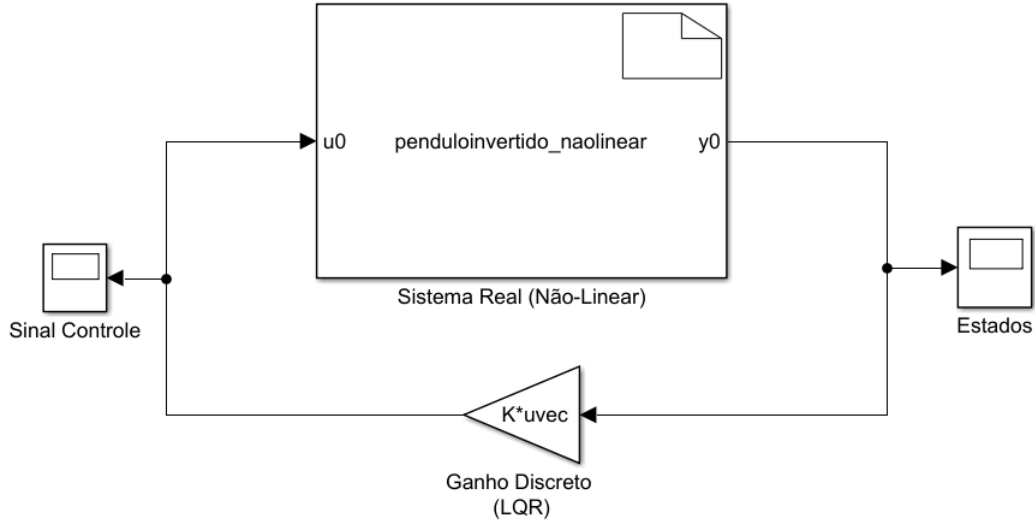


Figura 4.1: Implementação do controlador LQR.

4.3.2 Controlador LQG

Como já mencionado anteriormente, o controlador LQG é composto além dos ganhos ótimos do LQR quanto de um observador de estados, que nesse caso é o filtro de Kalman (LQE). Dessa forma, precisa-se encontrar um vetor de ganhos ótimo L que minimize a função de variância de erros vista em (2.37). A Figura (4.2) mostra a implementação do LQG com o filtro de Kalman em espaço de estado.

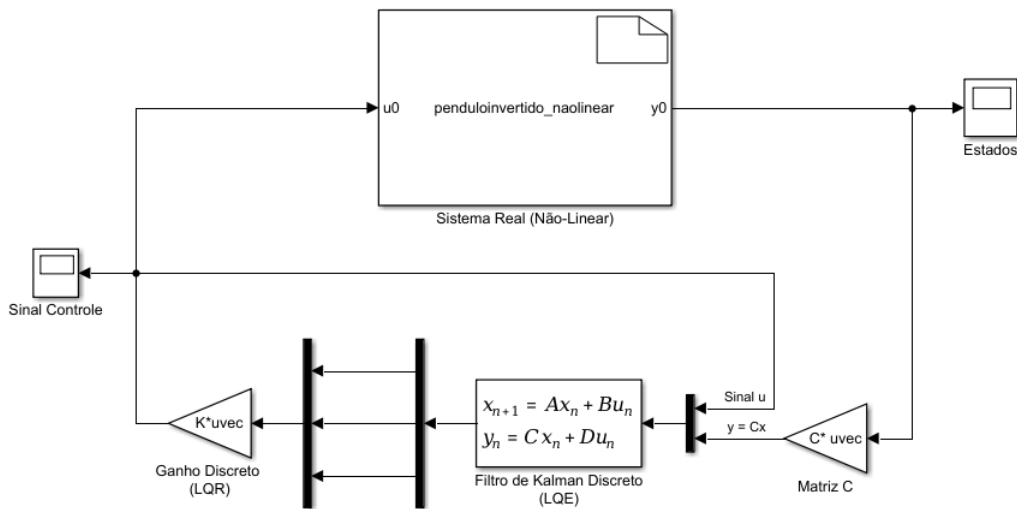


Figura 4.2: Implementação do controlador LQG.

Filtro de Kalman

Primeiramente, para projetar um filtro de Kalman, é necessário encontrar a matriz de covariância de processo Q_o e R_o , que geralmente é definido como um número escalar $\rho > 0$. Para efeitos de testes no modelo, escolheu-se esses parâmetros da seguinte maneira:

$$Q_o = 0.1\mathbf{I}_{n \times n} \quad R_o = 1 \quad (4.14)$$

Após feito essas escolhas, utilizou-se o comando do MATLAB *dlqe*, que retornou o ganho ótimo L do filtro, a matriz da solução de Riccati S , o erro de covariância Z e os polos do estimador E . Os parâmetros necessários precisa passar para este comando são: A_d e C_d do sistema bem como as matrizes de covariância escolhidas Q_o e R_o . O resultado que o *software* proporcionou foi o seguinte:

$$L = \begin{bmatrix} 0.5721 \\ -5.0411 \\ -0.9402 \end{bmatrix} \quad S = \begin{bmatrix} 0.9702 & -0.0442 & 0.7936 \\ -0.0442 & 0.1341 & 0.5272 \\ 0.7936 & 0.5272 & 3.1667 \end{bmatrix} \quad (4.15)$$

$$Z = \begin{bmatrix} 0.9685 & -0.0390 & 0.8141 \\ -0.0390 & 0.1183 & 0.4648 \\ 0.8141 & 0.4648 & 2.9216 \end{bmatrix} \quad E = (0.9023, 0.9695 \pm 0.0273i) \quad (4.16)$$

Assim, basta colocar o filtro no padrão de espaço de estados (2.40) e realizar a implementação como visto na Figura (4.2). As matrizes do filtro são A_f , B_f , C_f e D_f e são definidas como se segue

$$\begin{aligned} A_f &= (A_d - LC_d) & B_f &= [B_d \quad L] \\ C_f &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & D_f &= 0 * [B_d \quad L] \end{aligned} \quad (4.17)$$

A matriz C_f é definida como identidade porque necessita-se que na saída do filtro estejam presente todos os estados do sistema. A matriz D_f é completa por zeros e possui a mesma dimensão de B_f .

4.4 Implementação dos Controladores ao Modelo Não Linear

Nesta etapa, será discutido sobre os resultados encontrados após a implementação dos controladores LQR e LQG encontrados anteriormente. Utilizou-se o Simulink para simular os resultados no modelo não linear da planta como visto em (4.1) e (4.2).

4.4.1 Resposta ao controlador LQR

Primeiramente, para efeito de comparação, simulou-se apenas o LQR, ou seja, realimentou os estados por um fator K_r que é o ganho ótimo encontrado para esse regulador. Como já mencionado, esse controlador necessita-se de todos os estados disponíveis. Logo, na prática fica inviável dependendo do processo que pretende-se controlar.

A Figura (4.3) apresenta o comportamento da planta em relação ao ângulo da estrutura. Para efeitos de simulação, iniciou-se o sistema com um angulação ($\pi/20$ radianos) para avaliar o quão rápido é a resposta deste controlador. Como visto na figura, o sistema entrou em regime permanente em aproximadamente 330ms (**Alterar, está errado os 2%, TEM QUE SER EM RELAÇÃO A 0.16**) e em 600ms já estava em seu ponto de equilíbrio (0°).

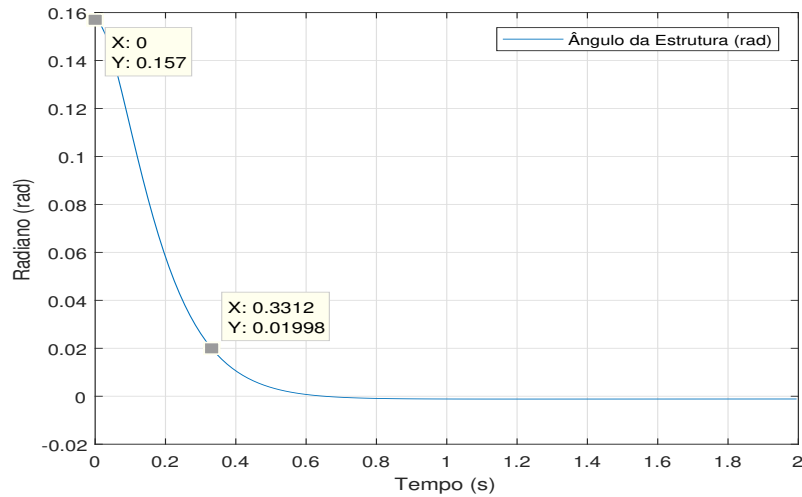


Figura 4.3: Variação do ângulo da estrutura com o controlador LQR.

Com um tempo tão baixo para se acomodar, é esperado que o sinal de controle seja alto inicialmente. A Figura (4.4) apresenta estes dados. Como visto, no início estava em 0.8 e foi diminuindo rapidamente com o passar do tempo. No mesmo tempo em que a saída da planta está em 0° , o valor do sinal de controle é praticamente nulo.

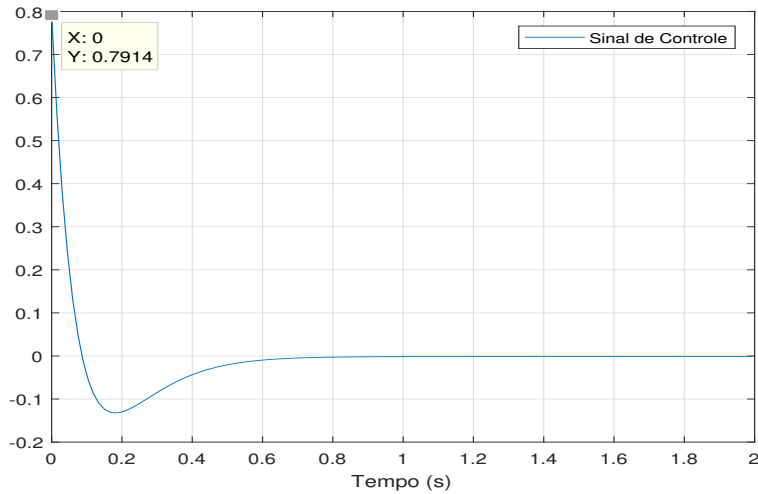


Figura 4.4: Sinal de controle do sistema com o controlador LQR.

A Figura (4.5) mostra a dinâmica dos outros dois estados estimados pelo LQE. Em laranja é visto a velocidade angular, que é coerente com a posição da estrutura. Já em azul é visto a velocidade linear do protótipo. A partir dela conclui que foi necessário para o pêndulo variar a sua posição linear para equilibrar-se.

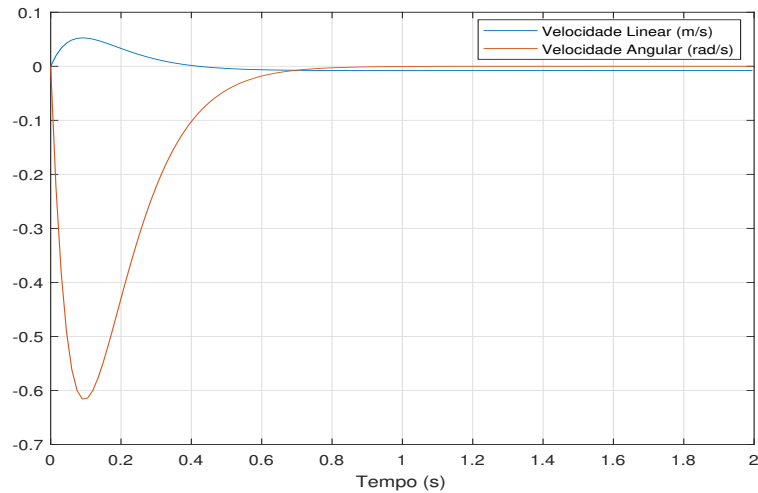


Figura 4.5: Velocidade linear e angular do sistema com o controlador LQR.

4.4.2 Resposta ao controlador LQG

Agora, com o uso do controlador LQG, fará as mesmas avaliações que foram feitas anteriormente. Contudo, comparará os resultados obtidos com cada controlador, avaliando suas vantagens e desvantagens. Para este controlador iniciou o ângulo da estrutura em $\pi/20$ radianos também.

A Figura (4.6) detalha a variação do ângulo da estrutura. É perceptível que com esse controlador houveram oscilações e o tempo de acomodação (1.2s) foi aproximadamente 4 vezes maior que apenas com o uso do LQR. Este fato pode ser devido à diminuição da

margem de fase e de ganho do LQG em relação ao LQR, que possui margem de ganho infinita e de fase $> 60^\circ$, segundo às literaturas. Contudo, como já mencionado, a maior vantagem desse controlador é que não é necessário que todos estados estejam disponíveis, sendo assim, é muito mais viável implementar o LQG em sistemas reais do o LQR.

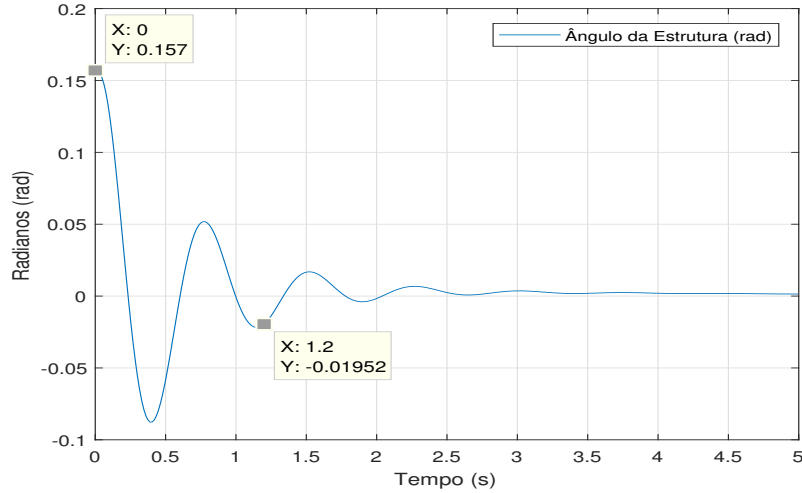


Figura 4.6: Variação do ângulo da estrutura com o controlador LQG.

Devido uma demora maior de acomodação do sistema com o uso do LQG, espera-se que o sinal de controle seja mais baixo do que o do LQR. A Figura (4.7) mostra a dinâmica do sinal e comprova este fato. A escolha do valor da matriz R também influencia diretamente no valor de controle que está disponível o sistema utilizar. Contudo, para efeito de comparação fez-se com que as matrizes Q e R fossem as mesmas. O valor máximo que o sinal de controle chegou foi de 0.3757, praticamente a metade do controlador anterior.

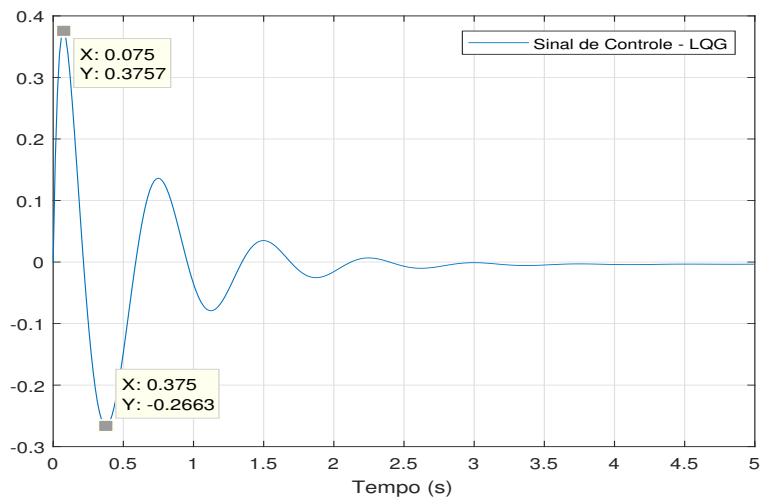


Figura 4.7: Sinal de controle da planta com o controlador LQG.

A Figura (4.8) apresenta a variação dos sinais dos outros estados estimados. O valor máximo que a velocidade angular atingiu neste controlador foi praticamente o dobro em

relação a velocidade angular utilizando apenas o LQR. Por razão da não acomodação da estrutura, é possível reparar que essa velocidade variou bastante ao longo do tempo.

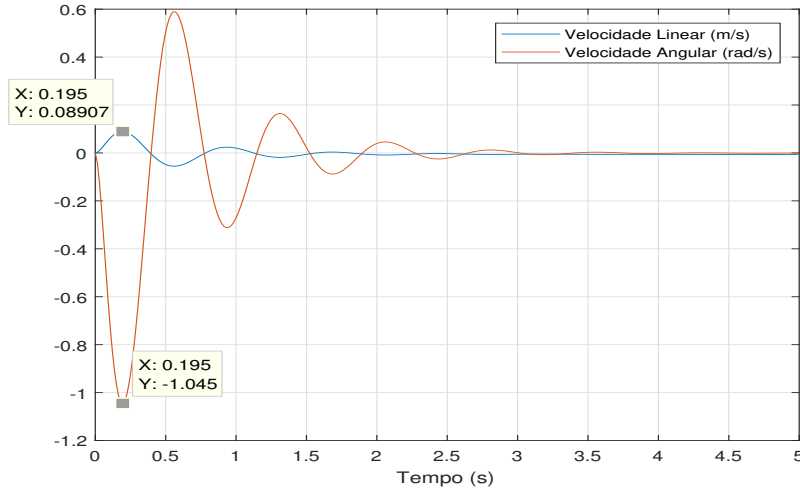


Figura 4.8: Velocidade linear e angular do sistema com o controlador LQG.

4.4.3 Resposta do sistema com aplicação de um distúrbio

Com o intuito de testar os controladores, aplicou-se um tipo de distúrbio não muito comum, pois mudou um pouco o valor do estado estimado \hat{x}_3 que agora com o subsistema é chamado de \hat{x}_2 . A Figura (4.9) apresenta como foi aplicado o distúrbio no estado.

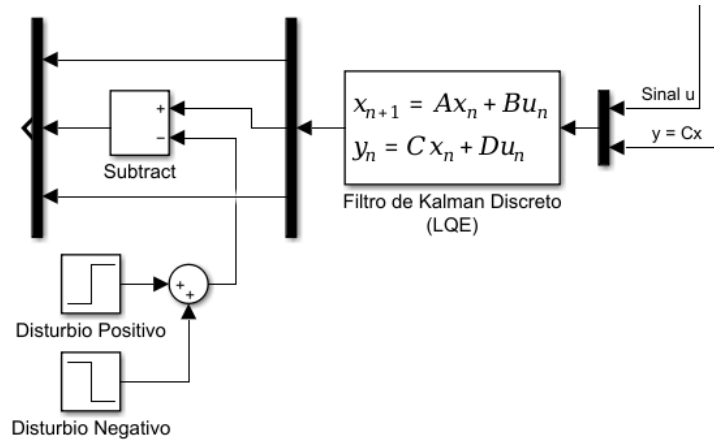


Figura 4.9: Distúrbio na ordem de $\pi/50$ aplicado ao estado estimado $\hat{x}_2 = \hat{\theta}$.

A Figura (4.9) mostra como foi realizado o distúrbio no LQG. Contudo, aplicou-se este mesmo método para o controlador LQR. Basicamente, aplicou-se uma pequena variação na ordem de $\pi/50$ no estado estimado \hat{x}_2 , tanto positivo quanto negativo em tempos diferentes: 10s aplicou-se o positivo e 0.5s depois o negativo. Dessa forma, será avaliada as respostas tanto do ângulo da estrutura quanto do sinal de controle para os dois controladores.

A Figura (4.10) apresenta a variação da posição angular da estrutura para os dois controladores. Até o tempo de 10s, a dinâmica da posição angular já foi discutida. Sendo assim, a

partir de $t=10s$, é visível que a posição angular para o controlador LQR tem um pico maior, chegando até 0.05 rad e acomodando após 0.68s. Para o controlador LQG, a variação da posição angular foi menor, porém demorou 1.16s para entrar em regime de acomodação.

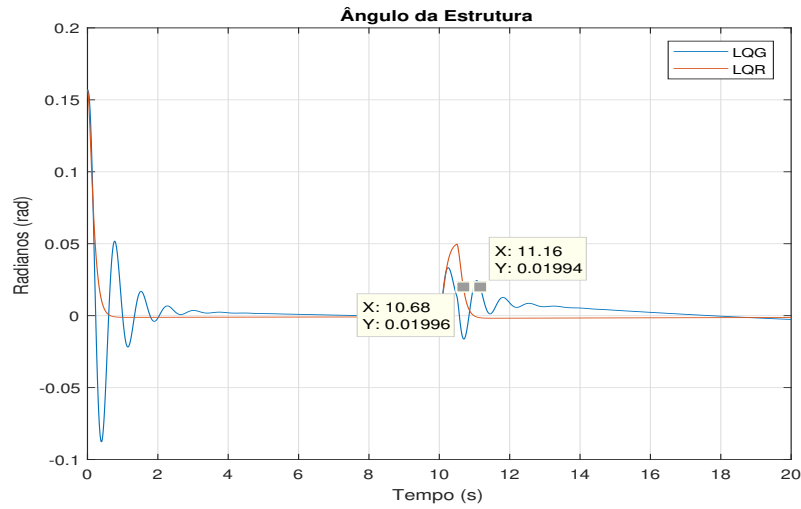


Figura 4.10: Variação do ângulo da estrutura com um distúrbio imposto ao sistema.

O sinal de controle com o distúrbio em $t=10s$ é o mesmo para os dois controladores em termos escalares, como visto na Figura (4.11). Porém, o sinal de controle do LQR varia muito pouco em contrapartida do sinal de controle do LQG que possui uma cauda e que faz com que seja zerado apenas após $t=16s$.

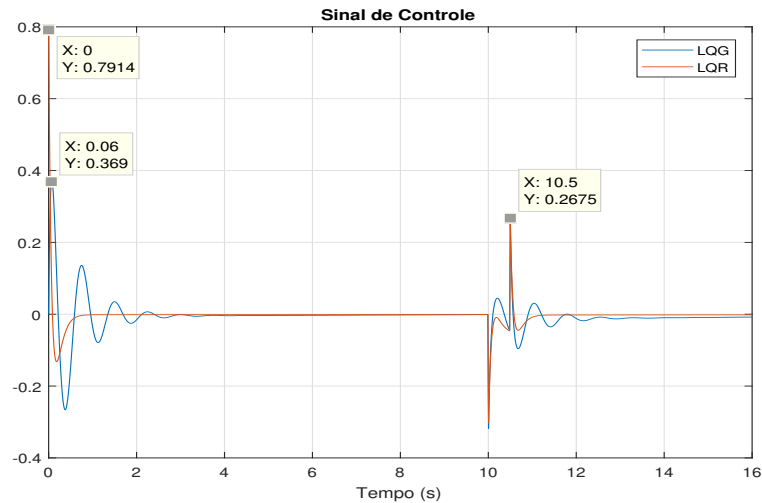


Figura 4.11: Sinal de controle com um distúrbio imposto ao sistema.

Considerações Finais

Este capítulo apresenta as conclusões acerca do projeto deste documento bem como as tarefas e um cronograma que se pretende seguir para a realização de um protótipo físico.

5.1 Conclusões

Deste documento, conclui-se que o tema abordado é de grande importância na academia e que possui uma variedade de pesquisas já realizadas em relação a este sistema. A técnica de modelagem utilizada é bastante usual na literatura, tendo em vista que agora precisa-se apenas dar continuidade e realizar a construção da planta física para observar e corrigir as diferenças entre o real e o modelo matemático encontrado.

Com relação as técnicas de controle ótimo, há vários estudos sobre o tema, o que assegurou uma maior certeza na escolha do controlador desenvolvido. O projeto do controlador LQG por meio de ajuda de *software*, como o MATLAB, mostrou ser fácil de se realizar e ao visualizar os resultados que foram obtidos com relação ao modelo não linear contínuo, apresentou-se bastante eficaz ao lidar com este tipo de sistema.

As atividades que foram realizadas e que estão documentadas neste texto, são de comum acordo com o que se propôs para esta parcial (TCC 1). As tarefas concluídas podem ser vistas na lista a seguir:

- desenvolver por meio do SolidWorks o conceito do protótipo;
- levantar a lista e o custo dos materiais;
- obter o modelo não linear do sistema e linearizar a partir de técnicas jacobianas;
- discretizar o modelo linear obtido;
- projetar o controlador LQG com base no modelo linear;
- avaliar a resposta do controlador aplicando ao modelo não linear;

Assim, pretende-se dar continuidade no projeto com o objetivo de realizar a construção física da planta e controlá-la. As atividades restantes para a realização do objetivo principal serão feitas no semestre seguinte, em TCC 2.

5.2 Continuidade para TCC 2

Este documento apresentou o projeto para a construção de um pêndulo invertido sobre duas rodas e que pretende-se dar continuidade no semestre seguinte como TCC 2. Dessa forma, as tarefas a serem cumpridas bem como um cronograma com o tempo estimado que cada atividade levará para ser alcançada podem ser vistos abaixo.

1. Construção Física do Protótipo:

- 1.1 Compra dos materiais eletrônicos necessários;
- 1.2 Calibração da MPU6050, testes dos motores com e do microcontrolador ESP32;
- 1.3 Construção/montagem da planta física;
- 1.4 Validação do modelo do sistema.

2. Aplicação dos Controladores:

- 2.1 Implementação do controlador à planta física;
- 2.2 Validação dos controladores por meio das especificações;
- 2.3 Aplicar melhorias ao projeto (caso seja possível).

3. Relatório do TCC2:

- 3.1 Escrita do relatório final de TCC2;
- 3.2 Defesa do TCC2.

Tabela 5.1: Cronograma de atividades para o TCC2.

Atividade (↓) Mês.(→)	<i>Julho</i>	<i>Agosto</i>	<i>Setembro</i>	<i>Outubro</i>	<i>Novembro</i>
1.1	✓				
1.2		✓			
1.3	✓	✓			
1.4			✓		
2.1			✓	✓	
2.2				✓	
2.3				✓	
3.1		✓	✓	✓	✓
3.2					✓

Apêndice A

Códigos

```
%% PROGRAMA PÊNDULO INVERTIDO - TCC 1 - 2019.1

clc; clear all; close all

%% Definição dos Parâmetros
syms g Jb Mb L Jw Mw r Rm b Ke Km

g = 9.81;
Jb = 0.00547151;    Mb = 1.017;    L = 0.096331;
Jw = 0.00445738;    Mw = 0.0844;    r = 0.325;
Rm = 7;    b = 0.002;    Ke = 0.1404;    Km = 0.3922;

% Momentos de Inércia deslocado
Jb_barra = Jb + Mb*L^2;
Jw_barra = Jw + Mw*r^2;

%% Parâmetros linearizados das matrizes A, B, C, D
den = Rm*(2*Jb_barra*Jw_barra + Jb*Mb*r^2);

sigma = (2*(Rm*b - Ke*Km)*(Jb + Mb*L^2 + Mb*L*r))/den;
tau = (-Mb^2*g*L^2*r^2)*Rm/den;
lambda = (2*(Rm*b - Ke*Km)*(2*Jw + 2*Mw*r^2 + Mb*L*r + Mb*r^2))/(r*den);
psi = (Mb*g*L*(2*Jw + 2*Mw*r^2 + Mb*r^2))/den;
alfa = (2*Km*r*(Jb + Mb*L^2 + Mb*L*r))/den;
beta = (-2*Km*(2*Jw + 2*Mw*r^2 + Mb*r^2 + Mb*L*r))/den;
```

```

%% Definição das Matrizes A, B, C, D
A = [ sigma      tau      -r*sigma;
      0          0          1;
      -lambda    psi      r*lambda];
B = [alfa; 0; beta];
C = [0 1 0];
D = zeros(size(C,1),size(B,2));

%% Construção do Sistema em Espaço de Estados no Tempo Contínuo
sys = ss(A,B,C,D);
Sc = ctrb(sys);
So = obsv(sys);

% Escolha das matrizes Q e R:
Q = [1  0  0;      % Considerando Q = C'C
     0  1  0;
     0  0  1;];
R = 1;

%% Construção do Sistema em Espaço de Estados no Tempo Discreto
Ts = 0.015;
sysd = c2d(sys,Ts);

% Matrizes discretas: Ad e Bd, Cd = C e Dd = D
Ad = sysd.a;
Bd = sysd.b;

% LQR Discreto
[Krd,Pd,Erd] = dlqr(Ad,Bd,Q,R);

% Sistema aumentado inserindo distúrbio e ruído
Qo = .1*eye(3); % matriz de covariância do distúrbio
Ro = 1;         % (escalar) covariância do ruído de medição

% Construção do LQE - Filtro de Kalman
[Ld,Sd,Z,Ed] = dlqe(Ad,Qo,C,Qo,Ro); % Comando LQE

sysKFd = ss(Ad-Ld*C,[Bd Ld],eye(3),0*[Bd Ld]);

```

Referências

- ALVES, R. G. *Controle de um pêndulo invertido utilizando técnica de linearização por realimentação*. Ouro Preto, MG, 2018.
- BAÚ DA ELETRÔNICA. Acesso: 25 de Abril de 2019, <http://www.baudaeletronica.com.br/motor-dc-6v-210rpm-com-encoder-e-roda.html>.
- BONAFILIA, B. R.; GUSTAFSSON, N.; NYMAN, P.; NILSSON, S. N. Self-balancing two-wheeled robot. In: *Anais...* [S.l.: s.n.], 2013.
- BRUNTON, S. L.; DURIEZ, T.; NOACK, B. R. *Machine Learning Control - Taming Non-linear Dynamics and Turbulence*. [S.l.]: Springer International Publishing, 2017.
- CHEN, C.-T. *Linear System Theory and Design*. 3. ed..ed. New York: Oxford University Press, 1999. 334p.
- ENGIN, M. Embedded LQR controller design for self-balancing robot. In: MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO), 2018., Budva, Montenegro. *Anais...* IEEE, 2018. p.1–4.
- FERNANDES, P. B. *Controladores Robustos do Tipo LQG/LTR de Orde Reduzida para Sistemas MIMO com Saídas Independentes de seus Modos Não Dominantes*. 2014. — Universidade Federal do Pará, Belém, PA.
- FERNANDO K. Acesso: 30 de Abril de 2019, <https://www.fernandok.com/2017/11/introducao-ao-esp32.html>.
- GROOVER, M. P. *Automação industrial e sistemas de manufatura*. 3. ed..ed. São Paulo: Pearson Prentice Hall: Prentice-Hall, Inc, 2013. 581p.
- HESPANHA, J. P. *Linear Systems Theory*. 1. ed..ed. Princeton, New Jersey: Princeton University Press, 2009. 280p.
- INVENSENSE. *MPU-6000 and MPU-6050 Product Specification*. Acesso: 2 de Maio de 2019, https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf.

- JUANG, H.; LURRR, K. Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL AND AUTOMATION (ICCA), 2013. *Anais...* [S.l.: s.n.], 2013. p.634–639.
- JUNFENG, W.; WANYING, Z. Research on Control Method of Two-wheeled Self-balancing Robot. In: FOURTH INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTATION TECHNOLOGY AND AUTOMATION, 2011. *Anais...* [S.l.: s.n.], 2011. v.1, p.476–479.
- JUNG, C. R.; OSÓRIO, F. S.; KELBER, C. R.; HEINEN, F. J. Computação Embarcada: projeto e implementação de veículos autônomos inteligentes. , [S.l.], 2005.
- KIM, H. S.; SOHN, J. W.; JEON, J.; CHOI, S.-B. Reduction of the Radiating Sound of a Submerged Finite Cylindrical Shell Structure by Active Vibration Control. *Sensors (Basel, Switzerland)*, [S.l.], p.2132–2147, 2013.
- MATOS, E. J. de. *Projeto de Controladores Robustos do tipo LQG/LTR para Sistema Multivariável Instável com uso de Pré-Compensador Dinâmico*. 2008. — Universidade Federal do Pará, Belém, PA.
- MELO, J. V. *Identificação e Controle de um Veículo Segway para fins Educacionais*. Porto Alegre, RS: SBAI, 2017. 1-6p.
- MERCADO LIVRE. Acesso: 10 de Maio de 2019, <https://produto.mercadolivre.com.br/MLB-699859307-lipo-bateria-2200-mah-3s-25c-turnigy-aero-drone-heli-JM?quantity=1>.
- OGATA, K. *Engenharia de Controle Moderno*. 5. ed..ed. São Paulo: Pearson Prentice Hall, 2010. 809p.
- OOI, R. C. *Balancing a Two-Wheeled Autonomous Robot*. Western Australia, 2003.
- PAULA, A. R. de. *Modelagem e controle do pêndulo invertido sobre duas rodas*. Fortaleza, CE, 2014.
- PEDROSO, C. C. S.; MODESTO, E. L. P. *Sistema de Controle de Pêndulo Invertido*. Monografia (Bacharel em Mecatrônica Industrial), Universidade Tecnológica Federal do Paraná UTFPR, Curitiba, PR, 2017.
- SAGE, A. P. *Optimum Systems Control*. 2. ed..ed. Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1977. 427p.
- SCHILING, D. E. *Modelagem e Controle de um Robô de Eixo Único em Sistema de Pêndulo Invertido*. Ijuí, RS, 2017.

- SEGWAY. Acesso: 20 de Abril de 2019, <http://www.segway.com/products/professional/segway-x2-se>.
- THAO, N. G. M.; NGHIA, D. H.; PHUCAND, N. H. A PID backstepping controller for two-wheeled self-balancing robot. In: INTERNATIONAL FORUM ON STRATEGIC TECHNOLOGY 2010. *Anais...* [S.l.: s.n.], 2010. p.76–81.
- VIDA DE SILÍCIO. *Ponte H L298N Controlando a velocidade de um motor DC com PWM*. Acesso: 25 de Abril de 2019, <https://portal.vidadesilicio.com.br/ponte-h-l298n-controle-velocidade-motor>.