



UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE
E AUTOMAÇÃO - CEC AU



DIONÍZIO JÚNIO INÁCIO

DESENVOLVIMENTO DE TAREFAS PARA ROBÔ SEGWAY
CONTROLADO POR REGULADOR LINEAR QUADRÁTICO (LQR)

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO

Ouro Preto, 2022

DIONÍZIO JÚNIO INÁCIO

**DESENVOLVIMENTO DE TAREFAS PARA ROBÔ SEGWAY
CONTROLADO POR REGULADOR LINEAR QUADRÁTICO (LQR)**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Agnaldo José da Rocha Reis, Dr.

Coorientador: Prof. José Alberto Naves Cocota Júnior, Dr.

**Ouro Preto
Escola de Minas – UFOP
2022**

Dionízio Júnio Inácio

Desenvolvimento de tarefas para robô SEGWAY controlado por Regulador Linear
Quadrático (LQR)/ Dionízio Júnio Inácio. – Ouro Preto, 2022-
52 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Agnaldo José da Rocha Reis, Dr.

Monografia de Graduação em Engenharia de Controle e Automação – Universidade
Federal de Ouro Preto, 2022.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III.
Faculdade de xxx. IV. Título

CDU 02:141:005.7



ATA DA SESSÃO DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

No décimo primeiro dia do mês de janeiro de dois mil e vinte dois, realizou-se às 08 horas, de forma não presencial, por meio do aplicativo Google Meet (<https://meet.google.com/zix-ptwx-jxi>), a sessão de defesa de Trabalho de Conclusão de Curso do candidato ao grau de Engenheiro de Controle e Automação, Dionízio Júnio Inácio, intitulada “DESENVOLVIMENTO DE TAREFAS PARA ROBÔ SEGWAY CONTROLADO POR REGULADOR LINEAR QUADRÁTICO (LQR)”. A Banca Examinadora foi constituída por Agnaldo José da Rocha Reis (Orientador, UFOP/EM/DECAT), Diógenes Viegas Mendes Ferreira (UFOP/EM/DECAT) e Paulo Marcos de Barros Monteiro (UFOP/EM/DECAT). O Prof. Agnaldo abriu a sessão agradecendo a participação dos examinadores supracitados e informou que o Prof. José Alberto Naves Cocota Júnior, coorientador do trabalho, não poderia participar da defesa por problemas com sua conexão de internet. Dito isso, ele passou a palavra ao candidato, que fez a exposição do seu trabalho. Em seguida, foi realizada a arguição pelos examinadores, com a respectiva defesa do candidato. Finalizada a arguição, a Banca Examinadora, sem a presença do candidato, deliberou pela sua **Aprovação**. Nada mais havendo para constar, lavrou-se a presente ata que será assinada eletronicamente pelos membros da banca supracitada via SEI/UFOP.



Documento assinado eletronicamente por **Agnaldo Jose da Rocha Reis, PROFESSOR DE MAGISTERIO SUPERIOR**, em 11/01/2022, às 13:54, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Paulo Marcos de Barros Monteiro, PROFESSOR DE MAGISTERIO SUPERIOR**, em 11/01/2022, às 17:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Diogenes Viegas Mendes Ferreira, TECNICO DE LABORATORIO AREA**, em 12/01/2022, às 15:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0267148** e o código CRC **CE24E136**.

Mãe, te amo!

AGRADECIMENTOS

À minha amada Mãe e ao meu finado Pai, por serem fonte de inspiração para o enfrentamento de quaisquer adversidades e problemas da vida, representam a materialização do amor e inesgotável fonte de carinho, portos seguros para qualquer hora.

Ao meu querido irmão, todo o meu respeito e admiração, exemplo de busca por sabedoria.

À todos os professores que tive e terei. Em especial, Matheus (DEFIS-UFOP), Cocota (DECAT-UFOP) e Agnaldo (DECAT-UFOP), pois sem eles não conseguiria sequer vislumbrar a grandiosa beleza contida na ciência.

À todos os amigos que fiz e que farei por meu breve suspiro de vida diante da infinitude do tempo.

À mim mesmo, orgulho-me por ser quem sou, ávido por acordar amanhã e poder descobrir um pouco mais do universo que me cerca.

“1903

*A primeira vez que um homem branco observou um homem negro
Não como um animal agressivo ou força braçal desprovida de inteligência*

Desta vez, percebe-se o talento, a criatividade, a música

O mundo branco nunca havia sentido algo como o blues

Um negro, um violão e um canivete

Nasce na luta pela vida, nasce forte, nasce pungente

Pela real necessidade de existir

O que é ser Bluesman?

É ser o inverso do que os outros pensam

É ser contra a corrente Ser a própria força, a sua própria raiz

É saber que nunca fomos uma reprodução automática

Da imagem submissa que foi criada por eles

Dane-se a imagem que vocês criaram

Não sou legível, não sou entendível

Sou meu próprio Deus, meu próprio santo, meu próprio poeta

Me olhe como uma tela preta, de um único pintor

Só eu posso fazer minha arte

Só eu posso me descrever

Vocês não têm esse direito

Não sou obrigado a ser o que vocês esperam

Somos muito mais

Se você não se enquadra ao que esperam

Você é um Bluesman.”

(BB King, Baco Exu do Blues)

RESUMO

Os robôs do tipo segway são encontrados numa ampla gama de aplicações. Dentre elas, industrial, doméstica, segurança, exploração aeroespacial, pesquisa acadêmica e até mesmo como hobby. Mesclando as duas últimas aplicações citadas, faz-se presente neste trabalho o desenvolvimento de tarefas para um robô do tipo segway construído com kit LEGO Mindstorms Ev3. Utilizou-se o software LabView a fim de se programar o robô. No intuito de obter uma boa resposta transitória, empregou-se a técnica do Regulador Quadrático Linear (LQR). As tarefas implementadas foram: (i) Seguidor de linha, (ii) Seguidor de objeto colocado à frente do robô e (iii) Subida em rampa inclinada. Ainda, para comparação, maior assertividade e melhoria de desempenho, comparou-se os dados de comportamento do robô obtidos do LabView com uma simulação de controlador LQR aplicada ao Segway implementada no software Matlab. Os resultados obtidos indicam que o desempenho nas tarefas foi bastante satisfatório, confirmando a robutez e estabilidade do robô.

Palavras-chaves: Segway. Robô. Lego Mindstorms. LabView. LQR.

ABSTRACT

Segway robots are found in a wide range of applications. Among them, industrial, domestic, security, aerospace exploration, academic research and even used by hobbyists. Merging the last two applications mentioned, it is present in this work the development of tasks for a segway robot built with LEGO Mindstorms Ev3 kit. The LabView software was used to program the robot. In order to obtain a good transient response, the Linear Quadratic Regulator (LQR) technique was used. Implemented tasks were: (i) Line follower, (ii) Object follower placed in front of the robot and (iii) Climbing on an inclined ramp. Also, for comparison, greater assertiveness and improved performance, the robot behavior data obtained from LabView was compared with an LQR controller simulation applied to the Segway implemented in the Matlab software. Finally, it was found that the performance in the tasks was satisfactory, confirming the robustness and stability of the robot.

Key-words: Segway. Robô. Lego-LabView. LQR.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplos de veículos Segway do tipo pêndulo invertido. Fonte: (SILVA, 2017)	17
Figura 2 – Descrição simplificada de um sistema de controle. Fonte: (NISE, 2020)	18
Figura 3 – Filtro complementar. Fonte: (JOSEFSSON MATTIAS; PERSSON, 2017)	19
Figura 4 – Servomotor LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)	22
Figura 5 – Sensor Giroscópio e acelerômetro LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)	23
Figura 6 – Sensor ultrassônico LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)	23
Figura 7 – Sensor de cor e luz LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)	24
Figura 8 – Diagrama de Corpo Livre do sistema do Segway. Fonte: (FAN, 2013)	25
Figura 9 – Implementação do LQR. Fonte: O autor	29
Figura 10 – Switch em 0, momento inicial, timer setado em 0s. Fonte: O autor	30
Figura 11 – Switch em 1, cálculo da frequência de amostragem. Fonte: O autor	30
Figura 12 – Switch em 2 (Padrão), recalculando a frequência de amostragem e atualizando a cada iteração do looping. Fonte: O autor	31
Figura 13 – Tratamento do sinal do giroscópio com a implementação do filtro complementar. Fonte: O autor	31
Figura 14 – Tratamento do sinal dos Encoders dos motores. Fonte: O autor	32
Figura 15 – Fluxograma de comandos do seguidor de linha. Fonte: O autor	33
Figura 16 – Implementação do código do Seguidor de linha no Segway. Fonte: O autor	33
Figura 17 – Imagem do robô Segway seguindo a faixa preta. Fonte: O autor	34
Figura 18 – Fluxograma lógico do seguidor de objetos. Fonte: O autor	35
Figura 19 – Diagrama de blocos do seguidor de objetos implementado no LabView. Fonte: O autor	35
Figura 20 – Imagem do robô Segway seguindo objeto. Fonte: O autor	36
Figura 21 – Implementação do diagrama de blocos para subir a rampa. Fonte: O autor	37
Figura 22 – Imagem do robô Segway subindo uma superfície plana e inclinada. Fonte: O autor	37
Figura 23 – Aplicação de impulso com amplitude de 3° na posição angular do corpo após 10 segundos de inicialização do robô. Fonte: O autor	38
Figura 24 – Posição angular do corpo em malha aberta no MATLAB. Fonte: O autor	39
Figura 25 – Velocidade angular do corpo em malha aberta no MATLAB. Fonte: O autor	39
Figura 26 – Posição angular da roda em malha aberta no MATLAB. Fonte: O autor	40
Figura 27 – Velocidade angular da roda em malha aberta no MATLAB. Fonte: O autor	40
Figura 28 – Posição angular do corpo em malha fechada no MATLAB. Fonte: O autor	41
Figura 29 – Velocidade angular do corpo em malha fechada MATLAB. Fonte: O autor	41
Figura 30 – Posição angular da roda em malha fechada no MATLAB. Fonte: O autor	42

Figura 31 – Velocidade angular da roda em malha fechada no MATLAB. Fonte: O autor	42
Figura 32 – Variação da Posição angular do corpo do Segway durante de 20s. Fonte: O autor	43
Figura 33 – Variação da velocidade angular do corpo do segway durante 20s. Fonte: O autor	43
Figura 34 – Variação da posição angular da roda do Segway durante 20s. Fonte: O autor	44
Figura 35 – Variação da velocidade angular da roda do Segway durante 20s. Fonte: O autor	44
Figura 36 – Impulso de amplitude 3° aplicado após 10s à partir da inicialização do Segway. Fonte: O autor	45

LISTA DE TABELAS

Tabela 1 – Exibição dos Parâmetros Físicos. Fonte: O autor	26
--	----

LISTA DE SÍMBOLOS

Γ	Letra grega Gama
Λ	Letra grega Lambda
ζ	Letra grega minúscula Zeta
\in	Símbolo matemático "Pertence"
ψ	Letra grega Psi

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Motivação	15
1.2	Objetivo geral	16
1.3	Objetivos específicos	16
1.4	Justificativa do trabalho	16
1.5	Estrutura do trabalho	16
2	REVISÃO DA LITERATURA	17
2.1	Controle	17
2.1.1	<i>Sistemas de Controle</i>	17
2.1.2	<i>Sistemas Não Lineares</i>	18
2.1.3	<i>Espaço de Estados</i>	18
2.1.4	<i>Controlabilidade e observabilidade</i>	18
2.2	Filtro Complementar	19
2.3	Linearização	20
2.4	Regulador Linear Quadrático	21
3	DESENVOLVIMENTO	22
3.1	Hardware	22
3.1.1	<i>Servomotor</i>	22
3.1.2	<i>Sensor giroscópio e acelerômetro</i>	23
3.1.3	<i>Sensor Ultrassônico</i>	23
3.1.4	<i>Sensor de cor e luz</i>	24
3.2	Apresentação e elaboração da planta de controle do Segway	24
3.2.1	<i>Modelagem matemática</i>	26
3.3	Implementação do controlador no LabView	29
3.3.1	<i>Cálculo da frequência de amostragem Δt no LabView</i>	30
3.3.2	<i>Implementação do giroscópio</i>	31
3.3.3	<i>Tratamento dos Encoders</i>	32
3.4	Implementação das tarefas propostas para o Segway	32
3.4.1	<i>Implementação do código seguidor de linha</i>	32
3.4.2	<i>Implementação do código seguidor de objeto</i>	34
3.4.3	<i>Implementação do código para subir rampas</i>	36
4	RESULTADOS	38
4.1	Simulação em software numérico (MATLAB)	38

4.1.1	<i>Simulação em software numérico em malha aberta</i>	38
4.1.2	<i>Simulação em software numérico em malha fechada</i>	40
4.2	Dados obtidos do Labview	42
5	CONCLUSÃO	46
	REFERÊNCIAS	47
	APÊNDICE A – PARÂMETROS DO SISTEMA	48
	APÊNDICE B – SIMULAÇÃO EM MALHA ABERTA COM APLICAÇÃO DE IMPULSO COM AMPLITUDE 1 DE GRAU	50
	APÊNDICE C – PROJETO DO CONTROLADOR LQR	51
	APÊNDICE D – SIMULAÇÃO EM MALHA FECHADA COM APLICAÇÃO DE IMPULSO COM AMPLITUDE DE 1 GRAU	52

1 INTRODUÇÃO

1.1 Motivação

O pêndulo invertido já faz parte do imaginário humano há muito tempo, provavelmente concebido pela primeira pessoa que teve a ideia de equilibrar uma vara sobre a palma da mão aberta, algo bem involuntário e totalmente lúdico. Na mesma medida em que as inovações tecnológicas foram acontecendo, o pêndulo invertido foi ganhando propósitos para além de seu aspecto lúdico, sendo matematicamente modelado, tornando-se um importante objeto de pesquisa dentro da física e da robótica.

Robôs que se equilibram em duas rodas já foram explorados por uma ampla gama de técnicas de controle, inclusive utilizando-se do mesmo hardware proposto neste trabalho, o kit Lego Ev3 Mindstorms. Por exemplo: [AKMAL, Jamin e Ghani \(2017\)](#) utilizaram Lógica Fuzzy, [Zometa et al. \(2012\)](#) utilizaram um modelo preditivo de controle, [Per, Ali e Olov \(2009\)](#) utilizaram um controlador Proporcional Integral Derivativo (PID) combinado a um controlador Linear Quadrático Gaussiano (LQG).

Neste trabalho a técnica de controle aplicada foi o Regulador Quadrático Linear (LQR). Conforme a teoria de controle moderno, o LQR é um controlador ótimo que minimiza a função custo, ([KANIESKI, 2010](#)). Esta função pode ser associada com a energia dos estados e a energia do controle. [Ogata \(2010\)](#), diz que uma vantagem do método de controle quadrático ótimo sobre o método de alocação é que o primeiro fornece um modo sistemático de cálculo da matriz de ganho de controle por realimentação de estado adequando-se de forma mais natural e eficaz quando aplicado à sistemas não-lineares.

Para a programação, foi utilizado o LabView (acrônimo para Laboratory Virtual Instrument Engineering Workbench) em sua versão 2016, uma linguagem gráfica elaborada conforme a direção do fluxo de dados pertencente a National Instruments, comumente utilizada na indústria, principalmente nos campos de medição e automação. Além disso, utilizou-se também o software Matlab (versão 2016) para a simulação do robô.

O conjunto de hardware utilizado foi o kit LEGO Mindstorms Ev3 da LEGO System A/S, amplamente difundido como ferramenta didático-pedagógica em laboratórios de ensino superior e até mesmo na educação infantil. Os servo-motores, sensor ultrassônico, giroscópio e sensor de cor utilizados são controlados pelo Brick (unidade programável com processador ARM 9 e sistema operacional baseado em linux pertencente ao kit Ev3), onde foi feito o upload de toda a programação do Segway.

1.2 Objetivo geral

Objetiva-se com este trabalho a implementação, observação, validação e análise do emprego da técnica de controle LQR em um robô com sistema pendular do tipo Segway.

1.3 Objetivos específicos

- Modelagem matemática e implementação da técnica de controle LQR, bem como das tarefas desenvolvidas para validação do conjunto.
- Observação e análise da dinâmica e do comportamento físico do sistema em diversos cenários aos quais o robô será submetido.
- Validação quantitativa e qualitativa da robustez e estabilidade da técnica de controle empregada.

1.4 Justificativa do trabalho

O pêndulo invertido faz-se como importante objeto de pesquisa na área de controle de sistemas, pois trata-se de um sistema não-linear, exigindo técnicas de controle complexas. O Segway vem sendo levantado como uma das alternativas de meio de transporte nos grandes centros urbanos, auxiliando no desafogo do trânsito caótico das metrópoles. Enfim, um considerável campo de pesquisa e desenvolvimento dentro da robótica.

1.5 Estrutura do trabalho

Os capítulos deste trabalho abordam de forma eficiente todo o arcabouço necessário para a compreensão do problema retratado, bem como uma estruturação condizente com o método científico, fazendo um acercamento investigativo do sistema em questão. O Capítulo 1 introduz o tema abordado por meio de uma introdução concisa sobre o objeto de estudo, além de uma breve contextualização acerca da dinâmica e modelagem envolvidas no processo. No Capítulo 2 é apresentada uma revisão bibliográfica dos métodos de linearização e de controle não linear utilizados para garantir a estabilidade do sistema apresentado no Capítulo 3. O Capítulo 3 discorre sobre a modelagem dinâmica do sistema a ser controlado, passando por toda a matematização e abordagem física do problema. Além disso, faz-se a comparação do modelo simulado com o real, evidenciando a aplicação dos métodos de controle. No Capítulo 4 são revelados os resultados obtidos, expostos os resultados tanto simulados quanto reais. No Capítulo 5 são inferidas e avaliadas todas as informações levantadas no capítulo 4 e apresentadas sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

O Segway Ev3 montado para este trabalho é considerado e tratado fisicamente como um pêndulo invertido cuja base é equilibrada sobre duas rodas dispostas no mesmo eixo, similar aos modelos presentes na figura 1.



Figura 1 – Exemplos de veículos Segway do tipo pêndulo invertido. Fonte: (SILVA, 2017)

Para que se possa controlar qualquer sistema é necessário ter o conhecimento de seu modelo dinâmico. Usualmente, sistemas de pêndulo invertido são modelados com o auxílio de métodos baseados em energia (ARÉVALO-CASTIBLANCO et al., 2018). Utilizou-se para a modelagem matemática da planta uma abordagem lagrangiana que considera coordenadas generalizadas, tornando o processo menos trabalhoso quando comparado a um sistema que se utiliza de Newton-Euler em sua abordagem.

A sua dinâmica pode ser representada por meio de equações diferenciais, denominadas equações de Euler-Lagrange, que são utilizadas para descrever o posicionamento de um sistema mecânico, utilizando o princípio da mínima ação ou princípio de Hamilton como ferramenta de cálculo. Neste capítulo são apresentados os conceitos físico-matemáticos e as técnicas de controle utilizadas na estabilização do pêndulo.

2.1 Controle

2.1.1 Sistemas de Controle

Segundo Nise (2020) um sistema de controle consiste em subsistemas (ou plantas) construídos com o objetivo de se obter uma saída desejada com um desempenho desejado, dada uma entrada específica, podendo ser representado de forma simplificada conforme a figura 2.

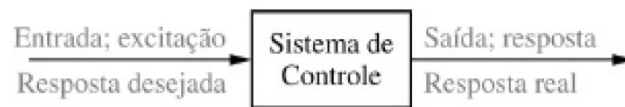


Figura 2 – Descrição simplificada de um sistema de controle. Fonte: (NISE, 2020)

2.1.2 Sistemas Não Lineares

Na prática, grande parte dos sistemas são não lineares. Além disso, não existe uma abordagem analítica totalmente generalista para avaliar seus efeitos. Segundo Dorf e Bishop (2017), um sistema linear deve contemplar as propriedades de superposição e homogeneidade. O sistema caracterizado pela seguinte relação

$$y(t) = x^2(t) \quad (2.1)$$

é tido como não linear, pois não satisfaz o princípio da superposição que por definição afirma que a resposta de saída de um sistema à soma de entradas é a soma das respostas às entradas individuais. O sistema representado pela relação

$$y(t) = mx(t) + b \quad (2.2)$$

é considerado não linear, pois o mesmo não obedece a propriedade de homogeneidade que descreve a resposta do sistema para uma multiplicação da entrada por um escalar. Especificamente, em um sistema linear, a propriedade de homogeneidade é demonstrada se, para uma entrada $r_1(t)$ que produz uma saída $C_1(t)$, uma entrada $Ar_1(t)$ produz uma saída $AC_1(t)$; isto é, a multiplicação de uma entrada por um escalar produz uma resposta que é multiplicada pelo mesmo escalar.

2.1.3 Espaço de Estados

O espaço n-dimensional, cujos eixos coordenados são formados pelos eixos de x_1, x_2, \dots, x_n , onde x_1, x_2, \dots, x_n são as variáveis de estado, é denominado espaço de estados. Qualquer estado pode ser representado por um ponto no espaço de estados. A análise no 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 (OGATA, 2010).

2.1.4 Controlabilidade e observabilidade

De acordo com Ogata (2010) um sistema é controlável no instante t_0 se for possível, por meio de um vetor de controle não limitado, transferir o sistema de qualquer estado inicial

$x(t_0)$ para qualquer outro estado, em um intervalo de tempo finito. Uma planta de ordem n cuja equação de estado é:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (2.3)$$

é completamente controlável, se a matriz

$$\mathbf{C}_M = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \dots \mathbf{A}^{(n-1)}\mathbf{B}] \quad (2.4)$$

for posto de n , na qual \mathbf{C}_M é chamada de matriz de controlabilidade. Onde, \mathbf{A} é chamada matriz de sistema e \mathbf{B} é chamada de matriz de entrada. Um sistema será dito observável no instante t_0 se, com o sistema no estado $x(t_0)$, for possível determinar esse estado a partir da observação da saída durante um intervalo de tempo finito.

2.2 Filtro Complementar

Entende-se como filtro complementar um algoritmo que tem por finalidade minimizar as falhas que um sistema pode ter quando faz-se a leitura de dois ou mais sensores para a mesma grandeza. Um exemplo recorrente acontece na combinação das leituras de um acelerômetro e um giroscópio para a medição de ângulos. Neste caso, o segundo, geralmente é impreciso em baixas frequências, gerando ruído de medição (JOSEFSSON MATTIAS; PERSSON, 2017).

O filtro complementar fornece uma estimativa do ângulo, fazendo a integração numérica do sinal obtido do giroscópio, por meio de um filtro passa-alta combinado com o sinal do acelerômetro filtrado num filtro do tipo passa-baixa conforme o esquema mostrado na figura 3.

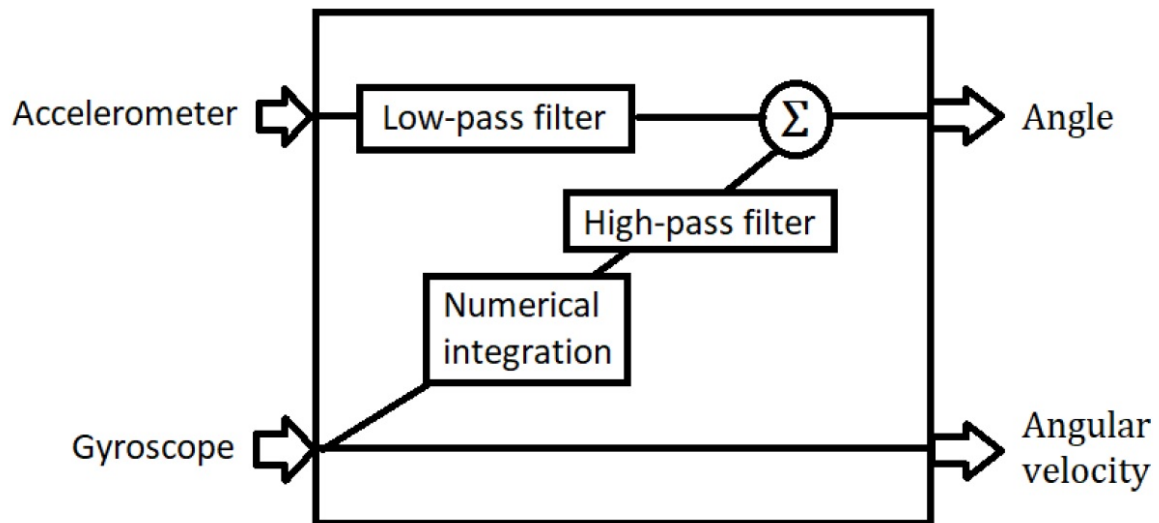


Figura 3 – Filtro complementar. Fonte: (JOSEFSSON MATTIAS; PERSSON, 2017)

A implementação de um filtro complementar pode ser feita conforme a equação 2.5.

$$\alpha = a(\alpha + gyro \, dt) + (1 - a)acel \quad (2.5)$$

Onde:

- α é o angulo medido;
- gyro é o valor da medição do giroscópio;
- accel é o valor medido pelo acelerômetro;
- a é um parâmetro que pode ser calculado pela equação 2.6.

$$a = \frac{\tau}{\tau + dt} \quad (2.6)$$

Onde: τ é a constante de tempo do sistema desejada e $dt = \frac{1}{f_s}$, é a frequência de amostragem.

2.3 Linearização

Conforme foi dito na subseção 2.1.2, na prática, muitos sistemas apresentam um comportamento não linear. Entretanto, podemos considerar que um sistema seja linear em torno de um ponto de operação x_0, y_0 , desde que sejam aceitas apenas pequenas variações em torno desse ponto. Quando $x(t) = x_0 + \Delta x(t)$ e $y(t) = y_0 + \Delta y(t)$, temos

$$y(t) = mx(t) + b \quad (2.7)$$

ou

$$y_0 + \Delta y(t) = mx_0 + m\Delta x(t) + b. \quad (2.8)$$

Consideradas as devidas condições de continuidade em torno do ponto de operação x_0 , podemos utilizar a expansão em Série de Taylor para encontrar um sistema linear suficientemente próximo do sistema original, permitindo a análise de características como polo, estabilidade etc (ALVES, 2010). De modo geral a Série de Taylor é expressa da seguinte forma:

$$f(x) = \sum_{k=0}^{\infty} f^{(k)}(a) \frac{(x - a)^k}{k!} \quad (2.9)$$

Representando $y(t)$ em função de $x(t)$ e aplicando a expansão em Série de Taylor, temos a equação linearizada podendo ser reescrita como

$$y(t) - y_0 = m(x(t) - x_0) \quad (2.10)$$

ou

$$\Delta y(t) = m\Delta x(t). \quad (2.11)$$

2.4 Regulador Linear Quadrático

A partir do sistema dado pela equação 2.3, Ogata (2010) verificou que o problema do regulador quadrático ótimo descrito, permite determinar a matriz de ganho \mathbf{K} do vetor de controle ótimo

$$u = -\mathbf{K}x \quad (2.12)$$

que minimiza o seguinte índice de desempenho:

$$J = \int_0^\infty (x^T \mathbf{Q}x + u^T \mathbf{R}u) dt \quad (2.13)$$

onde \mathbf{Q} é uma matriz hermitiana definida positiva (ou semidefinida positiva) ou real simétrica e \mathbf{R} é uma matriz hermitiana definida positiva ou real simétrica e determinam, respectivamente, a importância relativa do erro e o consumo de energia dos sinais de controle. Ainda, segundo Ogata (2010) esse ganho \mathbf{K} pode ser encontrado como a solução da equação diferencial matricial de Riccati, dada em sua forma reduzida pela equação 2.14:

$$-\mathbf{P}\mathbf{A} - \mathbf{A}^T \mathbf{P} - \mathbf{Q} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P} = 0. \quad (2.14)$$

3 DESENVOLVIMENTO

Neste capítulo será apresentado e especificado todo o aparato de hardware utilizado em todo o processo de desenvolvimento físico-matemático da planta, bem como toda a sua programação, implementação e simulação.

3.1 Hardware

A fim de auxiliar no controle de todo o sistema e na execução das tarefas do Segway, foram utilizados um conjunto de sensores e dois servomotores pertencentes ao kit lego Ev3. Foram eles:

3.1.1 Servomotor

O servomotor do kit LEGO Mindstorms EV3 mostrado na figura 4 utiliza um sensor de rotação interna (encoder) acoplado à sua caixa de redução que faz a medida em graus, permitindo uma medida com acurácia de 1 grau. Usando o sensor embutido, um motor se alinha com o outro na outra roda do robô para que ele possa andar em uma linha reta com a mesma velocidade nas duas rodas ou fornece sinais diferentes nas duas rodas permitindo que o robô possa girar. Possuindo as seguintes especificações:

- Feedback do encoder a um grau de acurácia;
- Velocidade máxima 160 a 170 RPM;
- Torque em movimento de 20 N/cm;
- Torque inercial de 40 N/cm.

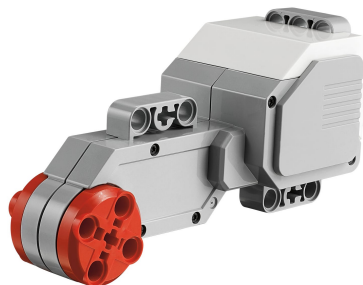


Figura 4 – Servomotor LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)

3.1.2 Sensor giroscópio e acelerômetro

O giroscópio mostrado na figura 5 mede o movimento e as mudanças na orientação de rotação do robô segway. Possuindo as seguintes especificações:

- O modo de medidas de ângulos com uma resolução de ± 3 graus;
- O modo giroscópio tem uma saída máxima de 440 graus/segundo;
- Taxa de amostragem de 1 kHz;



Figura 5 – Sensor Giroscópio e acelerômetro LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)

3.1.3 Sensor Ultrassônico

O Sensor ultrassônico mostrado na figura 6 gera ondas sonoras e lê seus ecos para detectar e medir a distância dos objetos. E possui as seguintes especificações:

- Acurácia de ± 1 cm ($\pm 0,394$ pol)
- Iluminação frontal é constante durante a emissão e pisca enquanto realiza a leitura
- Retorna true se outro ultra-som é reconhecido



Figura 6 – Sensor ultrassônico LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)

3.1.4 Sensor de cor e luz

O Sensor de cor e luz da figura 7 distingue diferentes cores e também pode detectar a ausência de luz. Ele também funciona como um sensor de luz através da detecção de diferentes intensidades. Suas especificações são:

- Reflete a luz vermelha e luz ambiente, do escuro para a luz muito brilhante.
- Detecta oito cores: Diferenciando preto e branco, ou diferenciando as cores azul, verde, amarelo, vermelho, branco ou marrom



Figura 7 – Sensor de cor e luz LEGO Mindstorms Ev3. Fonte: (LEGO System A/S, 2021)

3.2 Apresentação e elaboração da planta de controle do Segway

O Segway possui um encoder cuja função é medir a velocidade e o ângulo de rotação da roda (θ) e um sensor giroscópico que é responsável pela medição do ângulo e da velocidade angular do corpo ($\dot{\psi}$). O sensor de luminosidade teve o seu uso contemplado na segunda parte do trabalho, que consistiu em fazer o Robô Lego se locomover seguindo uma linha preta como referência. Na Figura 8 representou-se um diagrama de corpo livre para modelar este sistema, na Tabela 1 exibiu-se os valores e as unidades baseados no Sistema Internacional de Unidades (SI).

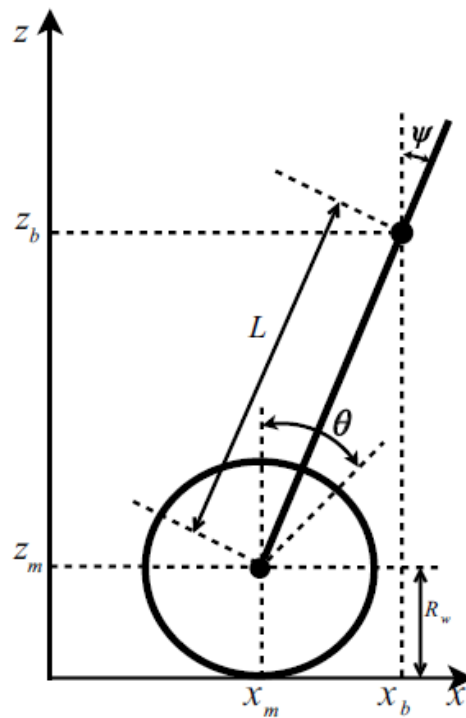


Figura 8 – Diagrama de Corpo Livre do sistema do Segway. Fonte: (FAN, 2013)

Onde de acordo com (FAN, 2013), considerou:

- L : distância do centro da roda até o centro de massa do pêndulo;
- θ : ângulo da roda;
- ψ : ângulo do pêndulo;
- M : massa do pêndulo;
- m : massa da roda;
- x_m e z_m : coordenadas do centro da roda;
- x_b e z_b : coordenadas do centro de massa do pêndulo;
- R_w : raio da roda;
- f_w : coeficiente de atrito entre a roda e o chão;
- f_m : coeficiente de atrito entre o pêndulo (corpo) e o motor;
- K_t : constante de torque do motor;
- K_b : constante da força eletromotriz;

- J_w : momento de inércia da roda;
- J_m : momento de inércia do motor;
- J_ψ : momento de inércia do pêndulo;
- i : corrente do motor DC;
- g : aceleração da gravidade;
- R_m : Resistência do motor.

Tabela 1 – Exibição dos Parâmetros Físicos. Fonte: O autor

Parâmetro	Valor	Unidade
g	9.81	$[m/s^2]$
M	0.606	$[kg]$
m	0.032	$[kg]$
R_w	0.027	$[m]$
L	0.0716	$[m]$
J_ψ	$(M * L^2)/3$	$[kg * m^2]$
J_w	$(m * R)/2$	$[kg * m^2]$
J_m	10^{-5}	$[kg * m^2]$
R_m	6.69	$[\Omega]$
K_b	0.468	$[V * s/rad]$
K_t	0.317	$[N * m/A]$
f_m	0.0022	[adimensional]
f_w	0	[adimensional]

3.2.1 Modelagem matemática

A modelagem matemática do Segway pode ser feita a partir da dinâmica de Lagrange, pois não foi necessário o desacoplamento da planta do sistema (JOSEFSSON MATTIAS; PERS-SON, 2017). Pode-se obter as Equações de Lagrange utilizando-se as coordenadas (θ) e (ψ) por 3.1, 3.2 e 3.3:

$$L_g = T_1 + T_2 - U \quad (3.1)$$

$$\frac{d}{dt} \left[\frac{\partial L_g}{\partial \dot{\theta}} \right] - \frac{\partial L_g}{\partial \theta} = F_\theta \quad (3.2)$$

$$\frac{d}{dt} \left[\frac{\partial L_g}{\partial \dot{\psi}} \right] - \frac{\partial L_g}{\partial \psi} = F_\psi \quad (3.3)$$

Onde de acordo com Josefsson Mattias; Persson (2017), F_θ e F_ψ significam as forças que dependem de θ e ψ . Logo, as equações utilizadas para o Segway são dadas por 3.4, 3.5 e 3.6:

- Energia cinética:

$$T_1 = \frac{1}{2}M(\dot{x}_b^2 + \dot{z}_b^2) + \frac{1}{2}m(\dot{x}_m^2 + \dot{z}_m^2) \quad (3.4)$$

- Energia cinética rotacional:

$$T_2 = \frac{1}{2}J_w\dot{\theta}^2 + \frac{1}{2}J_w\dot{\psi}^2 + \frac{1}{2}J_m(\dot{\theta} - \dot{\psi})^2 \quad (3.5)$$

- Energia potencial gravitacional:

$$U = mgz_m + Mgz_b \quad (3.6)$$

O sistema pode ser obtido pela representação do espaço de estados, que é desenvolvida a partir da linearização das equações 3.2 e 3.3. Essa linearização é feita a partir da consideração do fato de que o ângulo do corpo (ψ) é pequeno o suficiente, de forma que se possa supor que:

$$\sin(\psi) \rightarrow \psi$$

$$\cos(\psi) \rightarrow 1$$

$$\dot{\psi}^2 \rightarrow 0$$

Pode-se obter para as forças dependentes de (θ) e (ψ) por 3.7 e 3.8:

$$F_\theta = [(m + M)R_w + J_w + J_m]\ddot{\theta} + [MLR_w - J_m]\ddot{\psi} \quad (3.7)$$

$$F_\psi = [MLR_w - J_m]\ddot{\theta} + [ML^2 + J_\psi + J_m]\ddot{\psi} + ML_g\psi \quad (3.8)$$

Levando-se em consideração um motor de corrente contínua, foi utilizado o seguinte modelo:

$$F_\theta = K_t - f_m(\dot{\theta} - \dot{\psi}) \quad (3.9)$$

$$F_\psi = -K_t i + f_m(\dot{\theta} - \dot{\psi}) \quad (3.10)$$

Onde i = corrente do motor.

Visando a simplificação da equação, foi utilizado o seguinte modelo genérico 3.11.

$$L_m \dot{i} = v + K_b(\dot{\psi} - \dot{\theta}) - R_m i \quad (3.11)$$

No qual,

- v = tensão do motor;
- R_m = resistência do motor.

Sem considerarmos a indutância do motor pelo fato de a mesma ser pequena, tem-se que em 3.12:

$$i = \frac{v + K_b(\dot{\psi} - \dot{\theta})}{R_m} \quad (3.12)$$

Realizando-se a substituição da equação 3.12 em 3.9 e 3.10, obtêm-se 3.13, 3.14 e 3.15

$$F_\theta = \alpha v - (f_w + \beta)\dot{\theta} + \beta\dot{\psi} \quad (3.13)$$

$$F_\psi = -\alpha v + \beta\dot{\theta} - \beta\dot{\psi} \quad (3.14)$$

$$\alpha = \frac{K_t}{R_m}, \quad \beta = f_m + \frac{K_t K_b}{R_m} \quad (3.15)$$

Fazendo a igualdade da equação 3.7 com a 3.13 e a equação 3.8 com a 3.14, obtêm-se a representação do sistema em espaço de estados:

$$\dot{x} = E_p^{-1} A_p x + E_p^{-1} B_p u, \quad \theta = Cx \quad (3.16)$$

Cujo:

$$E_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (m+M)R_m^2 + J_w + J_m & MLR_w - J_m \\ 0 & 0 & MLR_w - J_m & ML^2 + J_\psi + J_m \end{bmatrix} \quad (3.17)$$

$$A_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -f_w - \beta & \beta \\ 0 & ML_g & \beta & -\beta \end{bmatrix} \quad (3.18)$$

$$B_p = \begin{bmatrix} 0 \\ 0 \\ \alpha \\ -\alpha \end{bmatrix} \quad (3.19)$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.20)$$

3.3 Implementação do controlador no LabView

Em conformidade com o que foi mencionado anteriormente em 2.4, os valores obtidos para a matriz \mathbf{Q} e a constante \mathbf{R} foram feitos por ponderação através do processo de tentativa e erro, similar aos métodos utilizados em (JOSEFSSON MATTIAS; PERSSON, 2017). Sendo assim, os valores encontrados foram, respectivamente:

$$\mathbf{Q} = \begin{bmatrix} 17566.8 & 0 & 0 & 0 \\ 0 & 905.2 & 0 & 0 \\ 0 & 0 & 137.4 & 0 \\ 0 & 0 & 0 & 68.8 \end{bmatrix} \quad (3.21)$$

$$\mathbf{R} = 0.05; \quad (3.22)$$

Para o cálculo da matriz de ganho estático \mathbf{K} do controlador LQR (3.23) foi utilizado o software numérico MATLAB, onde foram passados os valores das matrizes \mathbf{Q} e a constante \mathbf{R} conforme o apêndice C, análogo à abordagem encontrada em Mathworks® (2021), atingindo satisfatoriamente as características desejadas.

$$\mathbf{K}_{LQR} = \begin{bmatrix} 878.34 & 45.26 & 6.87 & 3.44 \end{bmatrix} \quad (3.23)$$

É necessário que se multiplique os ganhos da matriz \mathbf{K}_{LQR} por $\pi/180$ para converter os ganhos que estavam em radianos para graus nessa planta de controle.

$$\mathbf{K}_{LQR} = \begin{bmatrix} 15.33 & 0.79 & 0.12 & 0.06 \end{bmatrix} \quad (3.24)$$

O controlador LQR foi implementado no software LabView 2016, é possível observar os valores da matriz de ganho 3.24 sendo aplicados à planta conforme à Figura 9.

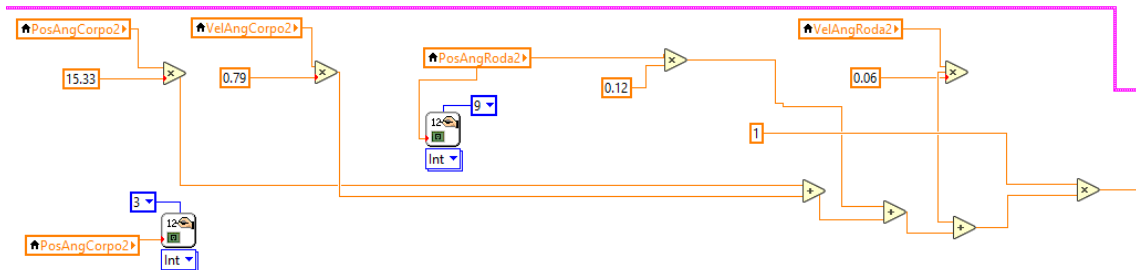


Figura 9 – Implementação do LQR. Fonte: O autor

3.3.1 Cálculo da frequência de amostragem dt no LabView

As imagens das Figuras 10, 11 e 12 mostram a implementação no LabView do cálculo da frequência de amostragem mostrada na seção 2.2. nas configurações do Switch em 0, em 1 e em 2, este último sendo o padrão mantido após as duas primeiras iterações do sistema de controle. Todas as implementações abaixo foram feitas no Software LABView.

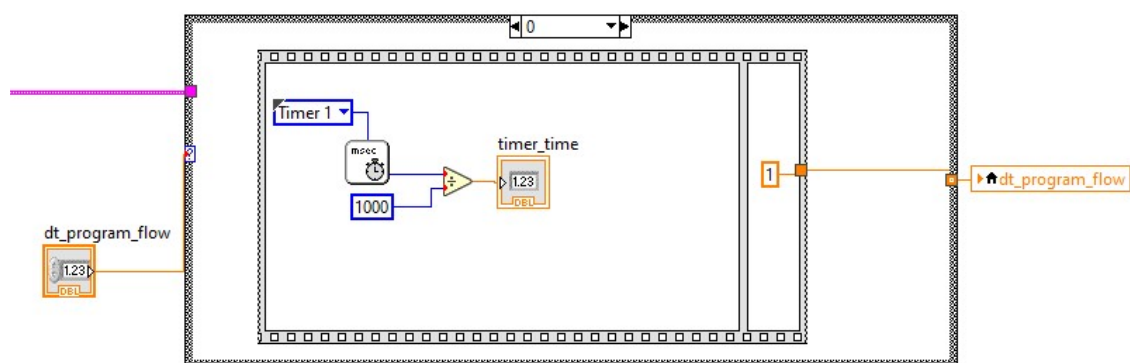


Figura 10 – Switch em 0, momento inicial, timer setado em 0s. Fonte: O autor

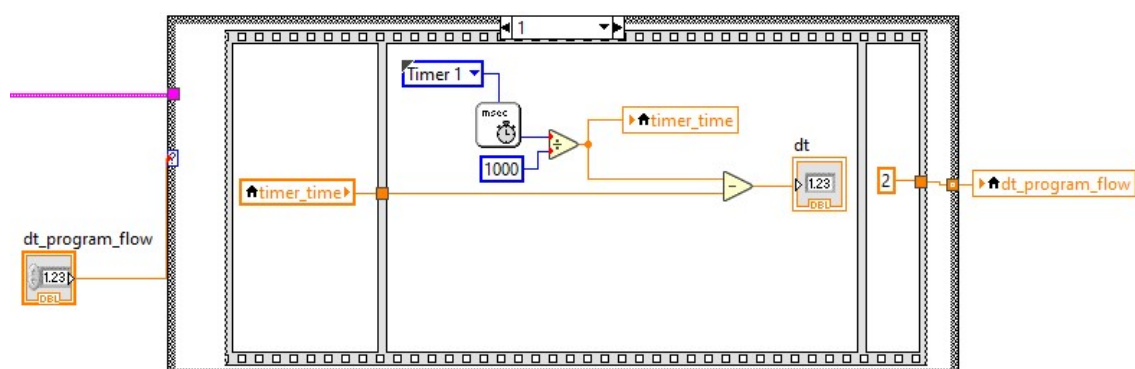


Figura 11 – Switch em 1, cálculo da frequência de amostragem. Fonte: O autor

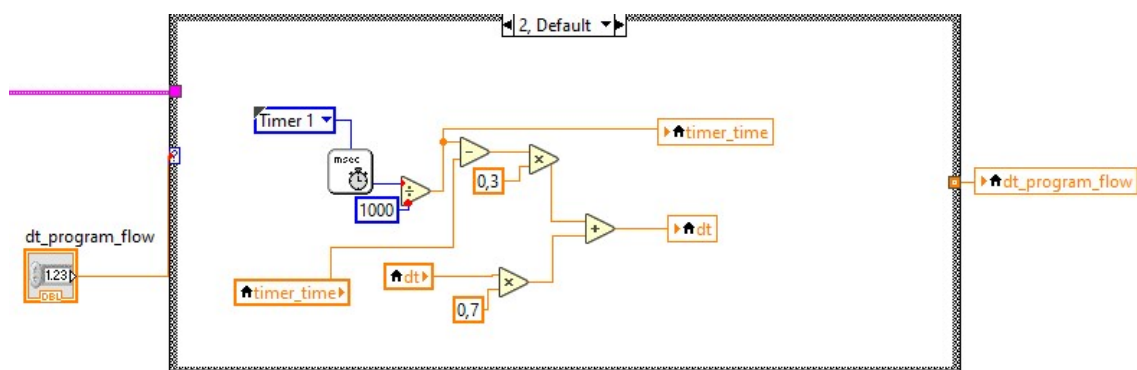


Figura 12 – Switch em 2 (Padrão), recalculando a frequência de amostragem e atualizando a cada iteração do looping. Fonte: O autor

3.3.2 Implementação do giroscópio

Utilizou-se um filtro complementar no sinal do giroscópio e do acelerômetro para auxiliar no desempenho do robô durante o seu equilíbrio e trajetória. A obtenção, processo de filtragem e saída do sinal devidamente tratado antes de seguir para uso no bloco de controle podem ser vistos na figura 13.

A estrutura elaborada segue a equação proposta para implementação do filtro conforme 2.5. Onde, a variável denominada e correspondente ao valor da medição no giroscópio, o parâmetro a e a constante de tempo τ são aplicados.

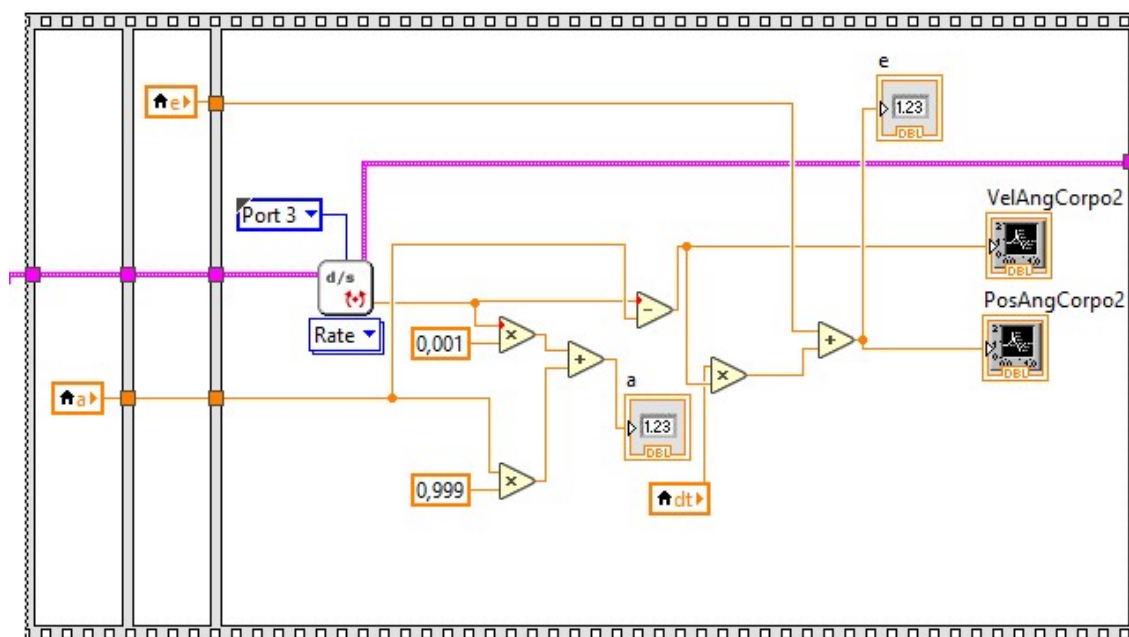


Figura 13 – Tratamento do sinal do giroscópio com a implementação do filtro complementar. Fonte: O autor

3.3.3 Tratamento dos Encoders

Para o tratamento dos sinais dos encoders dos motores foi feita a implementação da figura 14 no LabView.

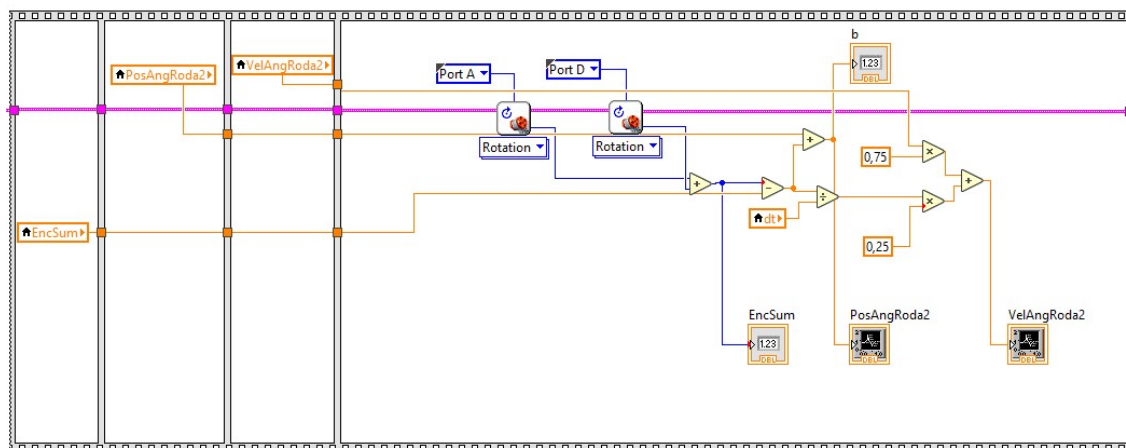


Figura 14 – Tratamento do sinal dos Encoders dos motores. Fonte: O autor

3.4 Implementação das tarefas propostas para o Segway

3.4.1 Implementação do código seguidor de linha

Uma das propostas deste trabalho foi a implementação do código do robô seguidor de linha no Segway, para fazer com que ele se locomova guiado por uma faixa preta em uma superfície seguindo a trajetória da mesma. Na lógica do seguidor de linha foi feita a duplicação da implementação do controlador aplicando-o em cada roda, visando melhorar o desempenho do robô ao longo do trajeto. O robô utiliza o sinal dos sensores de luz e cor como referência para acionar os motores conforme cada situação possível de acordo com a figura 15:

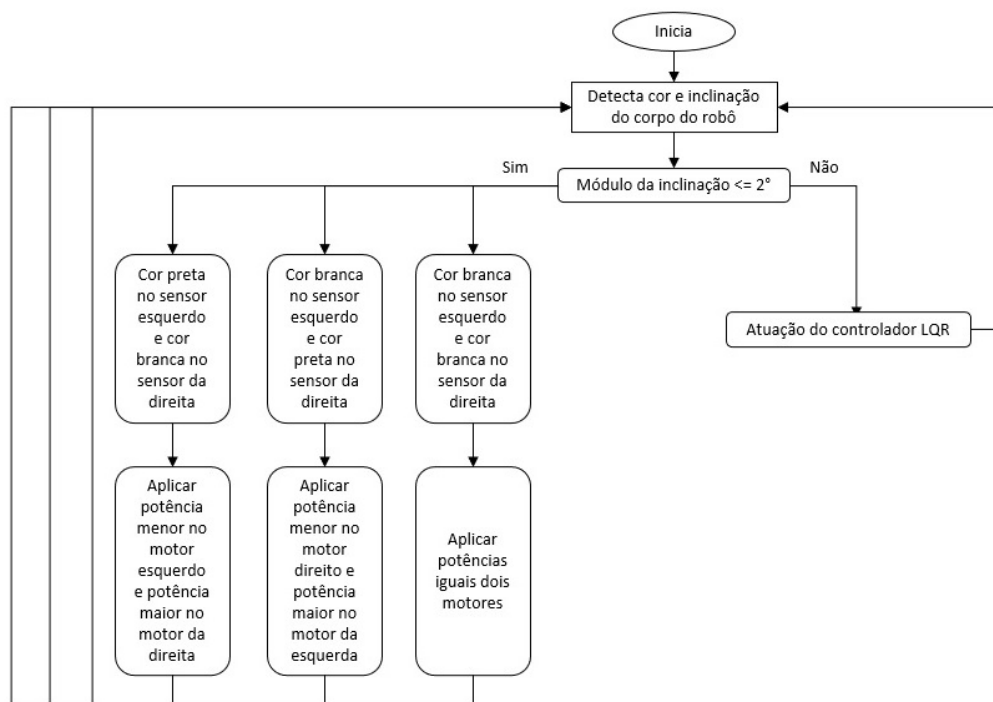


Figura 15 – Fluxograma de comandos do seguidor de linha. Fonte: O autor

Na Figura 16, tem-se a ilustração desse diagrama de blocos no LabView e na Figura 17 uma imagem do Segway seguindo uma faixa preta em uma bancada;

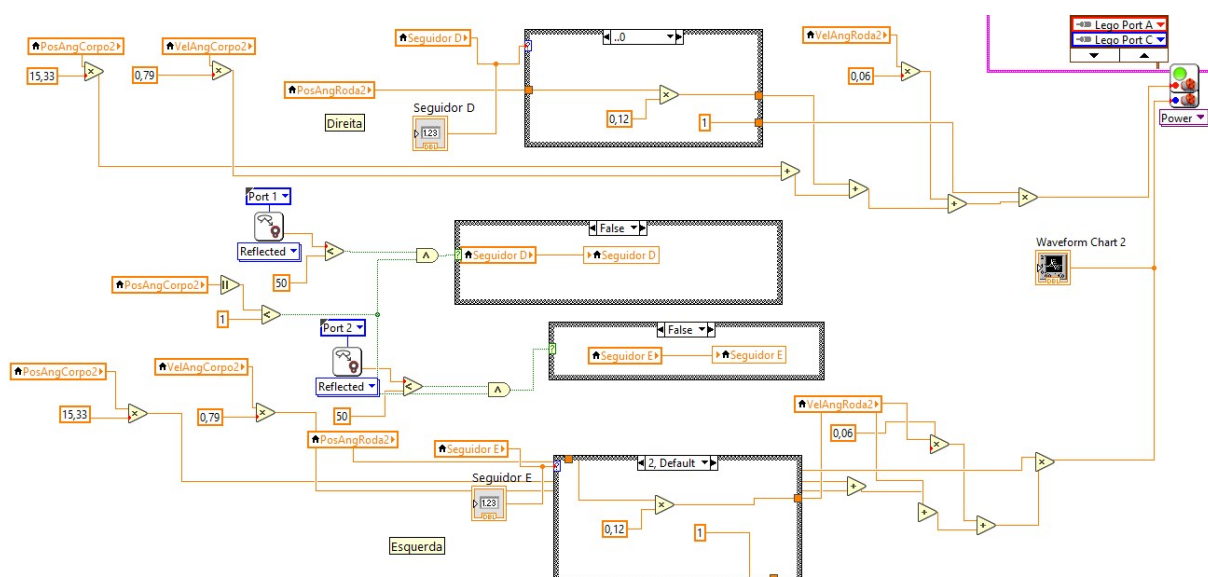


Figura 16 – Implementação do código do Seguidor de linha no Segway. Fonte: O autor

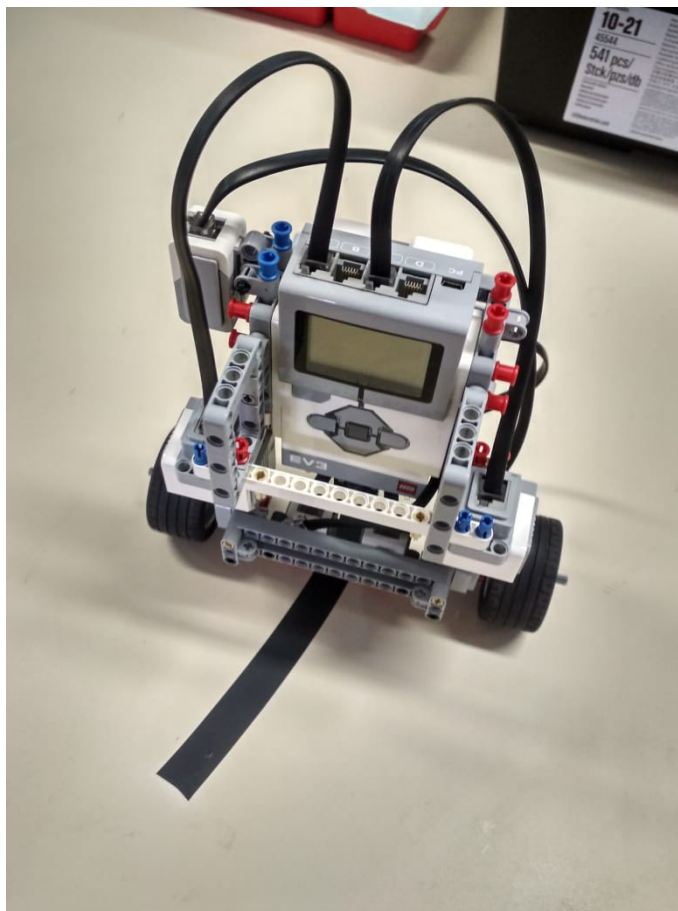


Figura 17 – Imagem do robô Segway seguindo a faixa preta. Fonte: O autor

3.4.2 Implementação do código seguidor de objeto

Nesta etapa também foi feita a utilização de um controlador por roda com a mesma finalidade do código seguidor de linha. Na figura 18 encontra-se o fluxograma contendo os sinais e as respostas de acordo com cada situação para coordenar as ações do segway.

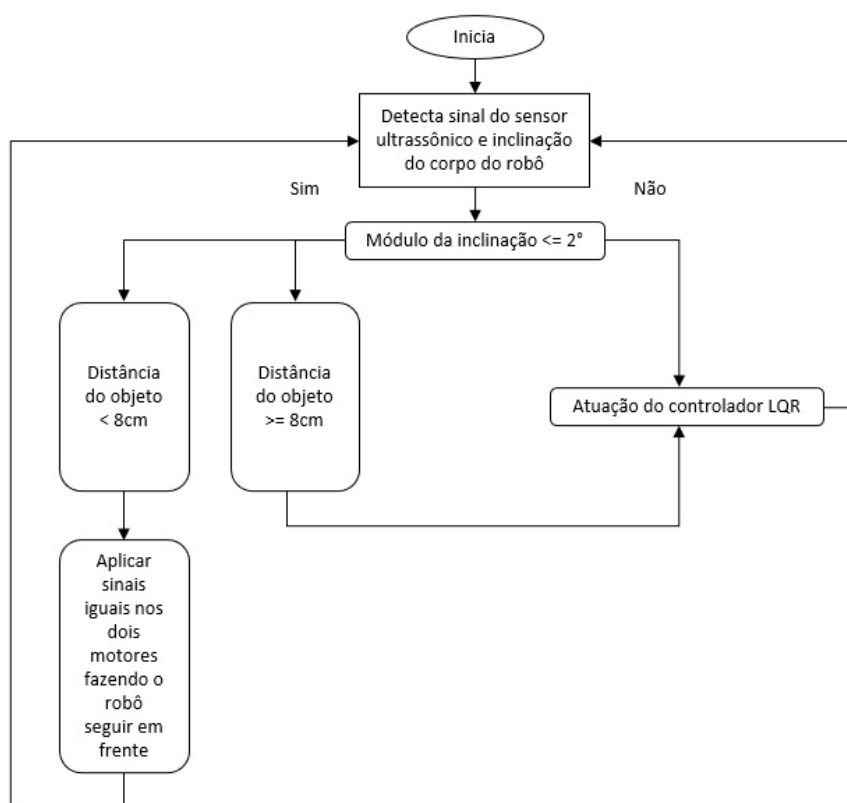


Figura 18 – Fluxograma lógico do seguidor de objetos. Fonte: O autor

O código que permite o Segway seguir objetos, é mostrado na Figura 19.

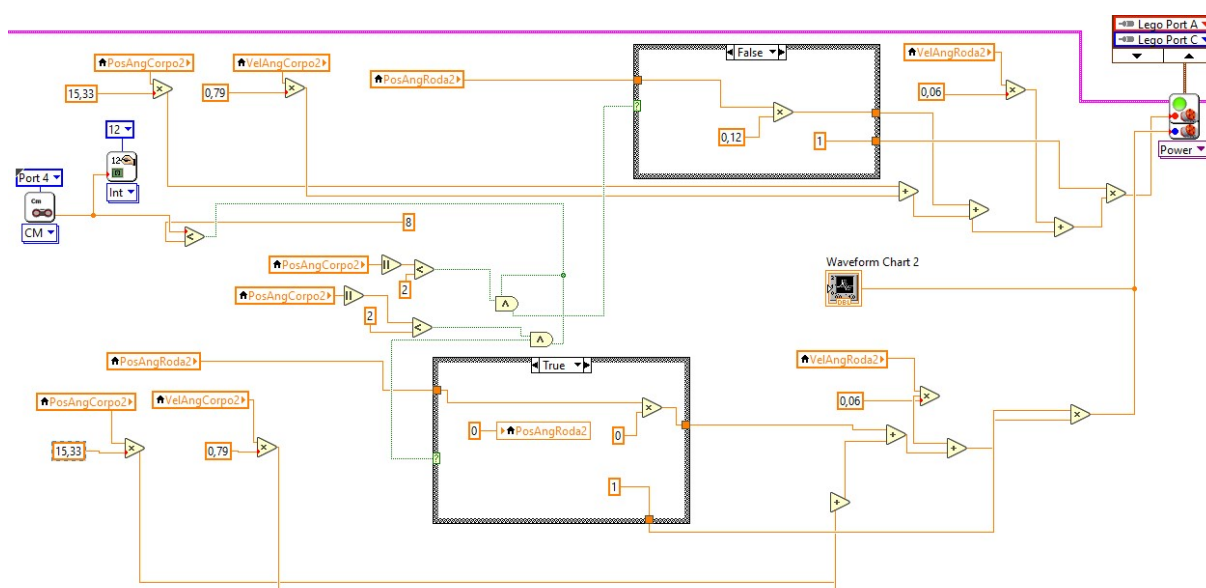


Figura 19 – Diagrama de blocos do seguidor de objetos implementado no LabView. Fonte: O autor

Na Figura 20 pode-se visualizar o robô Segway seguindo um objeto (livro).

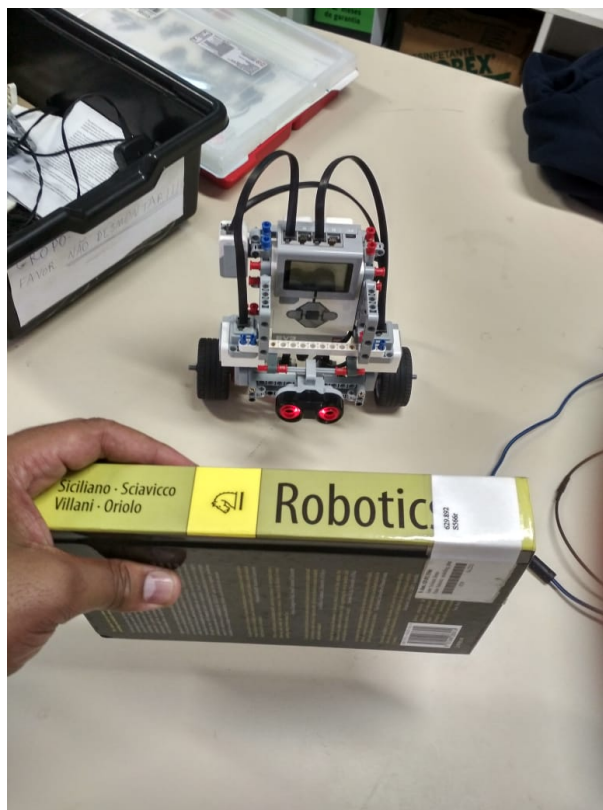


Figura 20 – Imagem do robô Segway seguindo objeto. Fonte: O autor

3.4.3 Implementação do código para subir rampas

Implementou-se também um código para fazer o Segway subir rampas, no caso, próximas a 15° de inclinação, sempre mantendo o equilíbrio. Seguiu-se a mesma ideia usada em 3.4.2 de se utilizar um controlador para cada roda, minimizando possíveis problemas em decorrência do movimento de spin do robô, ocasionado pela imperfeição da superfície. Além disso foi colada uma lixa na rampa, pois a mudança brusca no ângulo do plano de deslocamento causou sucessivas derrapagens. A Figura 21 exibe essa implementação. Na Figura 22, mostra-se o robô Segway subindo uma superfície inclinada.

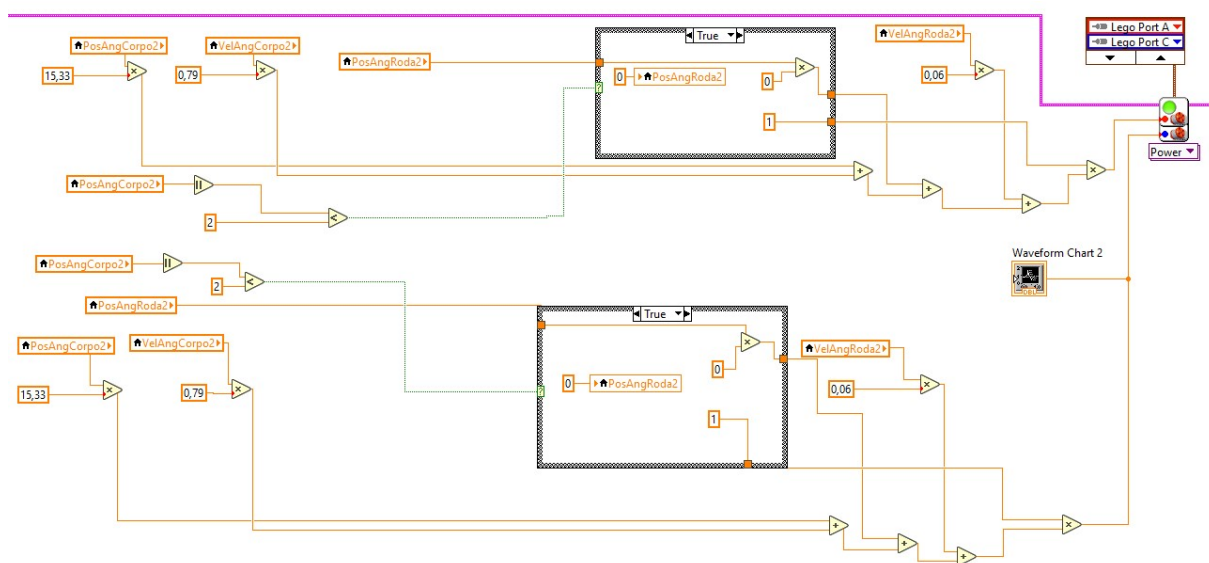


Figura 21 – Implementação do diagrama de blocos para subir a rampa. Fonte: O autor

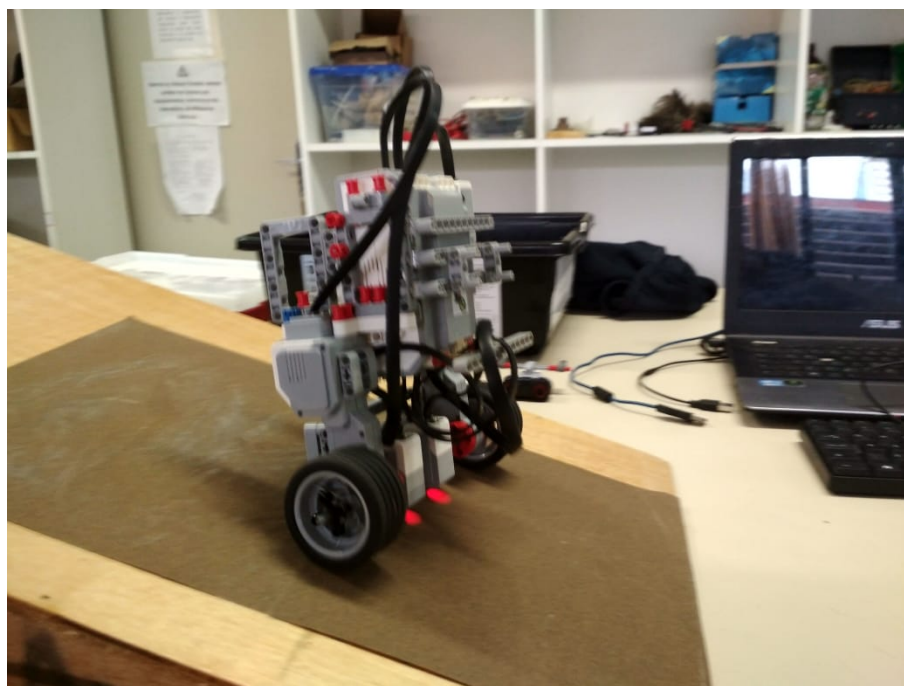


Figura 22 – Imagem do robô Segway subindo uma superfície plana e inclinada. Fonte: O autor

4 RESULTADOS

Nesta etapa são apresentados os resultados das simulações realizadas no software numérico MATLAB, bem como os dados obtidos a partir LabView do robô Segway enquanto estava em funcionamento. O Segway permaneceu por 20 segundos em operação estável e foi possível a coleta dos dados em tempo real através da porta USB no LabView. Com o objetivo de obter um resultado e poder compará-lo à simulação teórica, aplicou-se um impulso de amplitude de 3 graus no instante de 10 segundos, para possibilitar primeiramente o equilíbrio e a estabilidade do robô para posteriormente se aplicar o distúrbio conforme a figura 23 abaixo:

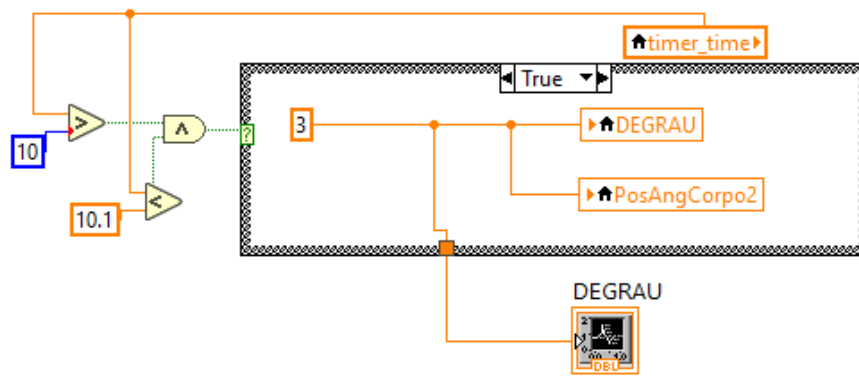


Figura 23 – Aplicação de impulso com amplitude de 3° na posição angular do corpo após 10 segundos de inicialização do robô. Fonte: O autor

4.1 Simulação em software numérico (MATLAB)

Para fins de comparação foi desenvolvida uma simulação do pêndulo no software MATLAB, em malhas aberta e fechada conforme os apêndices B e D, respectivamente, com o intuito de antever possíveis problemas que poderiam ocorrer no projeto prático. Permitindo assim, uma análise mais precisa, quantitativamente e qualitativamente do projeto proposto. O código do controlador LQR simulado em software pode ser visto no apêndice C. Além disso, os parâmetros do sistema encontram-se no apêndice A. Assim como no projeto real, a simulação conta com a aplicação de um impulso de amplitude de 3 graus na posição angular do corpo no instante de tempo igual a 10 segundos após inicialização da planta.

4.1.1 Simulação em software numérico em malha aberta

A fim de evidenciar o impacto da ausência de malha fechada no sistema as imagens 24 e 25 representam as variações da posição angular e da velocidade angular do corpo do Segway ao longo do tempo.

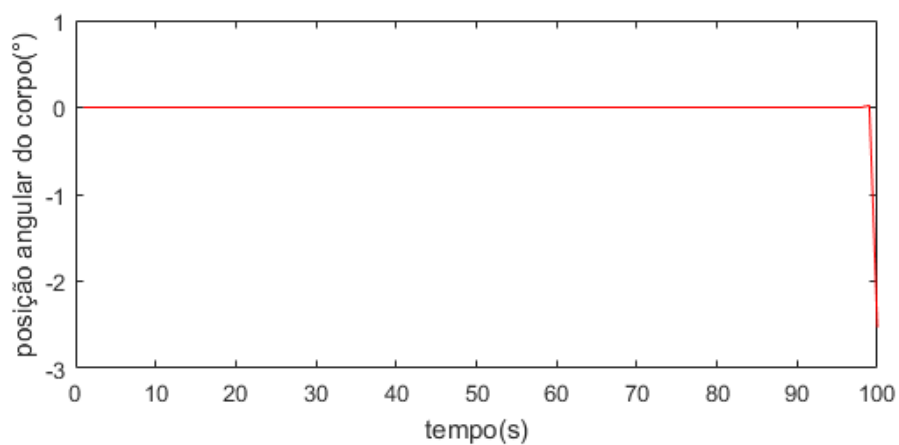


Figura 24 – Posição angular do corpo em malha aberta no MATLAB. Fonte: O autor

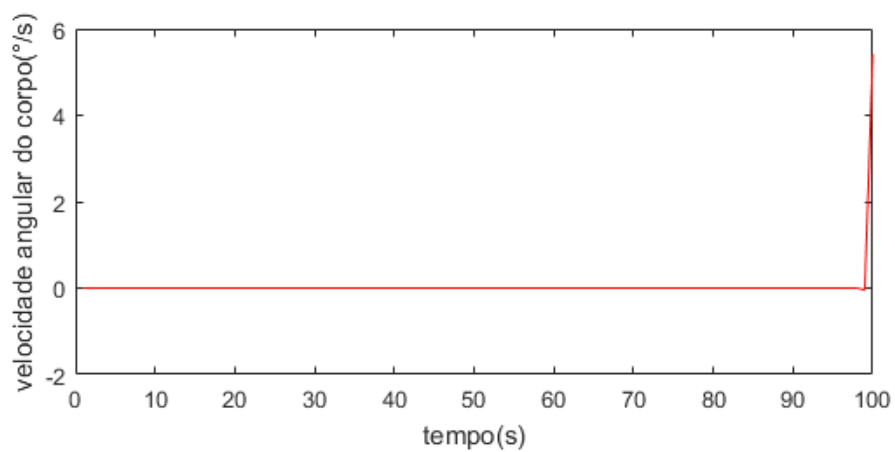


Figura 25 – Velocidade angular do corpo em malha aberta no MATLAB. Fonte: O autor

As imagens 26 e 27 reforçam os problemas inferíveis a partir das duas figuras anteriores, só que desta vez, exibem as variações de posição e velocidade angular ao longo do tempo para as rodas do robô, respectivamente:

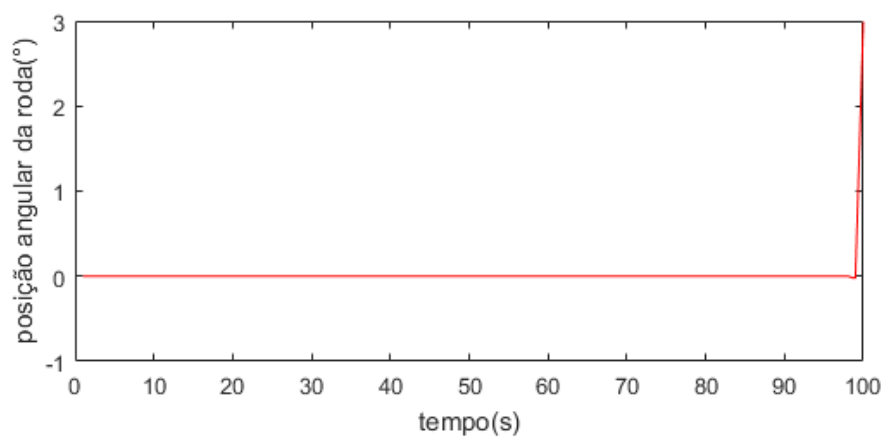


Figura 26 – Posição angular da roda em malha aberta no MATLAB. Fonte: O autor

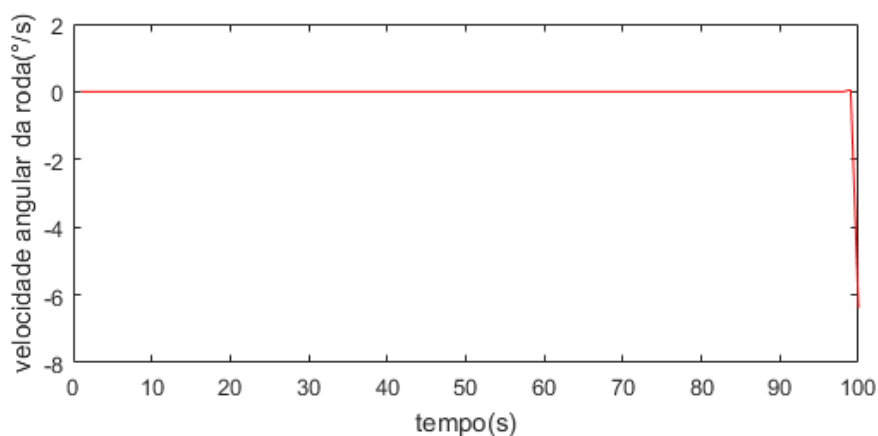


Figura 27 – Velocidade angular da roda em malha aberta no MATLAB. Fonte: O autor

4.1.2 Simulação em software numérico em malha fechada

Com o fechamento da malha e nova aplicação da simulação foram obtidos os seguintes resultados para posição angular e velocidade angular do corpo do Segway, conforme as figuras 28 e 29:

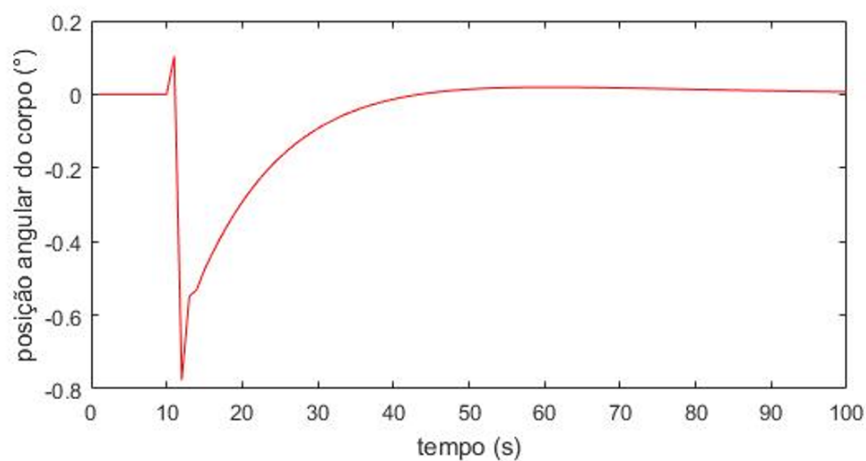


Figura 28 – Posição angular do corpo em malha fechada no MATLAB. Fonte: O autor

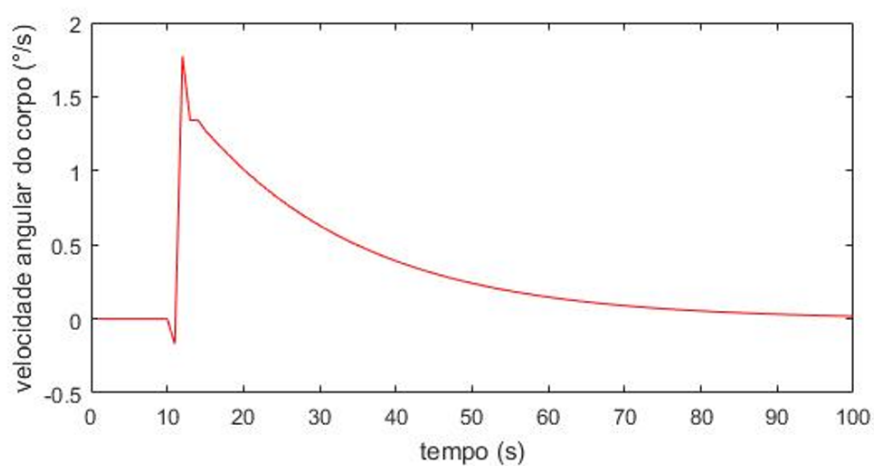


Figura 29 – Velocidade angular do corpo em malha fechada MATLAB. Fonte: O autor

Ainda em malha fechada, as figuras 30 e 31 representam o comportamento da posição e velocidade angulares da roda do Segway, conforme variação do tempo:

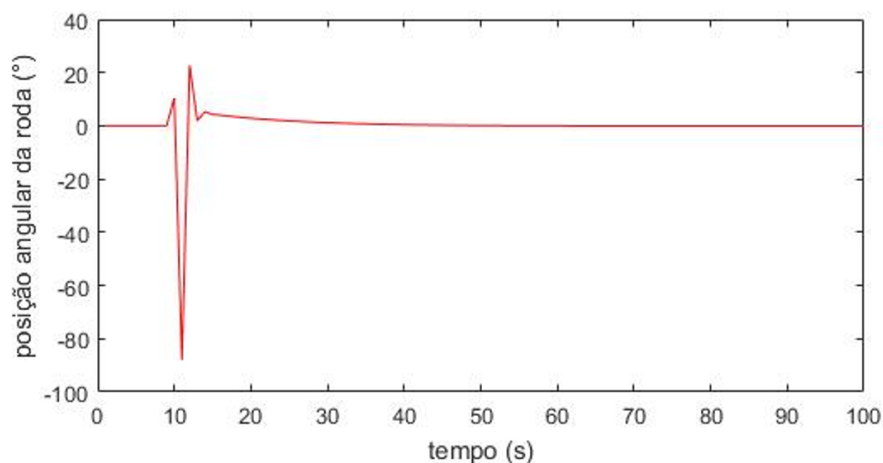


Figura 30 – Posição angular da roda em malha fechada no MATLAB. Fonte: O autor

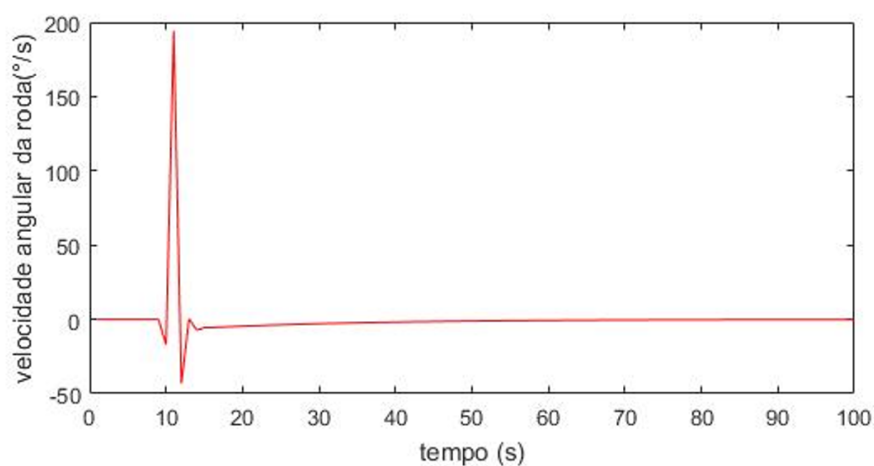


Figura 31 – Velocidade angular da roda em malha fechada no MATLAB. Fonte: O autor

4.2 Dados obtidos do Labview

Conforme proposto no início deste capítulo serão apresentados abaixo os dados obtidos a partir da execução do diagrama de blocos construído no Labview. Faz-se necessário observar que a implementação necessita de 5 segundos para inicialização após o upload do código no hardware do kit LEGO Mindstorms EV3. As figuras 32 e 33 ilustram respectivamente, a variação da posição e da velocidade angular do corpo do Segway ao longo de um tempo de 20s.

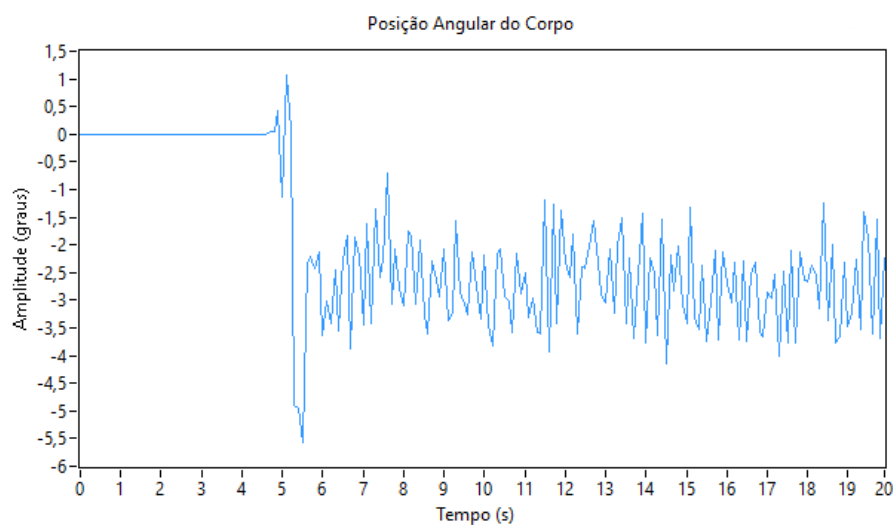


Figura 32 – Variação da Posição angular do corpo do Segway durante de 20s. Fonte: O autor

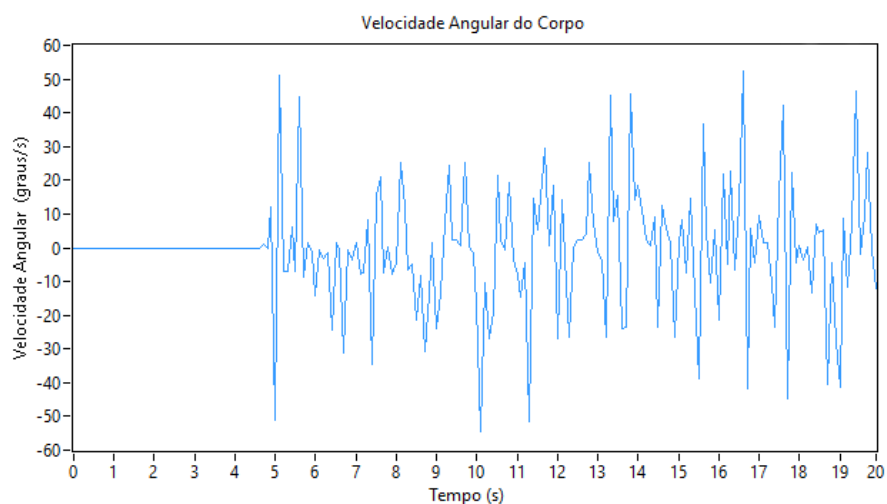


Figura 33 – Variação da velocidade angular do corpo do segway durante 20s. Fonte: O autor

As figuras 34 e 35 ilustram respectivamente, a variação da posição e da velocidade angular da roda do Segway ao longo de um tempo de 20s.

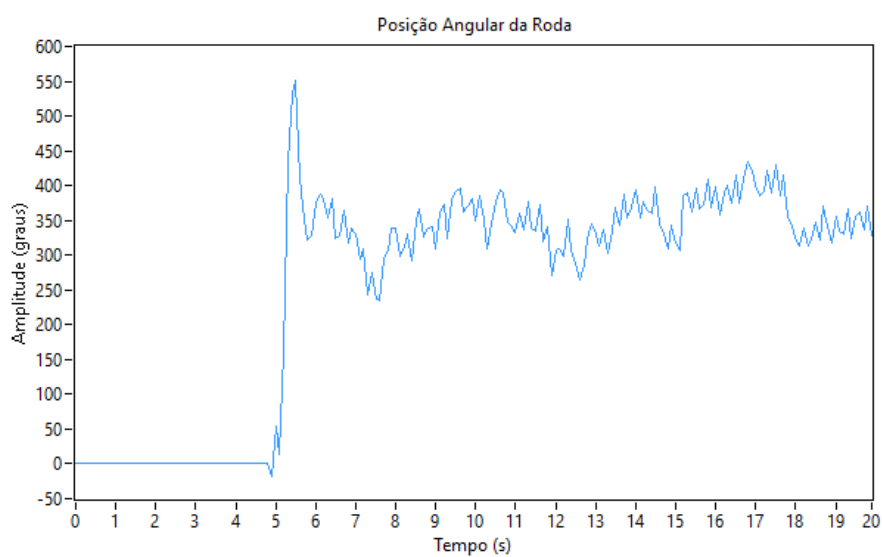


Figura 34 – Variação da posição angular da roda do Segway durante 20s. Fonte: O autor

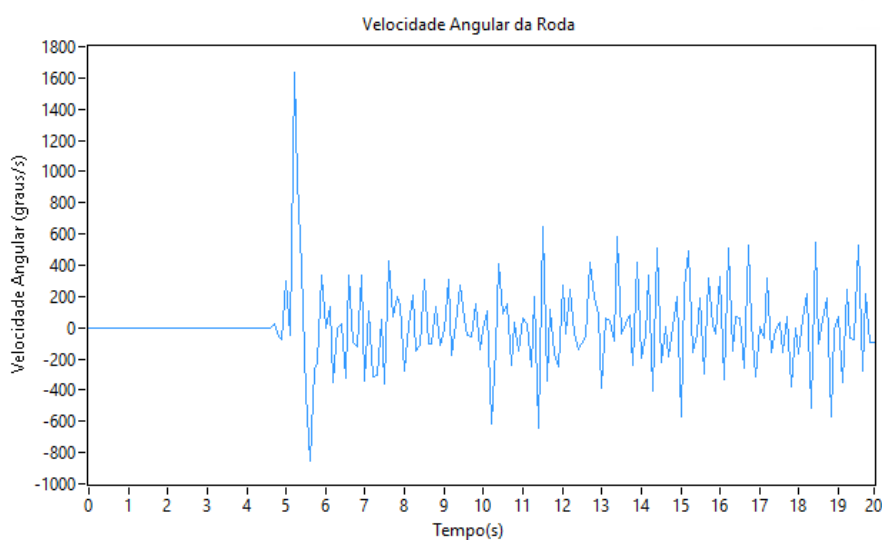


Figura 35 – Variação da velocidade angular da roda do Segway durante 20s. Fonte: O autor

Na figura seguinte (36) nota-se o pulso do impulso de amplitude 3 graus aplicado 10 segundos após a inicialização do Segway.

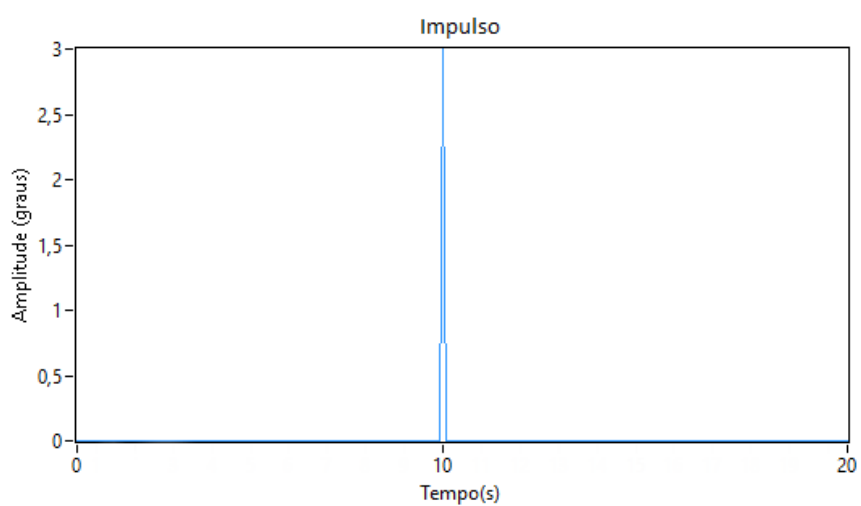


Figura 36 – Impulso de amplitude 3° aplicado após 10s à partir da inicialização do Segway.
Fonte: O autor

5 CONCLUSÃO

Para verificar-se a estabilidade do sistema, que é o objetivo principal deste projeto, pôde-se conferir graficamente, pelo LabView, em tempo real a resposta da velocidade angular em relação à variação de posição ao longo do tempo, que estabilizou-se em malha fechada por meio do uso do controlador LQR. Reforçando assim, a confiabilidade no modelo matemático tornando-o coerente e aplicável enquanto técnica de controle para o sistema real.

A robustez e estabilidade do sistema fizeram com que após a aplicação de distúrbio em forma de impulso no instante de tempo 10s, o robô retomasse o seu equilíbrio e a referência, evidenciando a robustez do projeto. Com isso, pode-se dizer que os resultados obtidos na implementação do LQR no Segway foram satisfatórios, atendendo então aos requisitos desejados no projeto. Além disso, o Segway mostrou-se promissor enquanto veículo não tripulado, podendo ser operado remotamente ou mesmo autonomamente em implementações mais complexas.

Além disso, o sistema mostrou um desempenho satisfatório e eficiente quando submetido às três tarefas propostas no capítulo 3. Obtendo êxito em suas execuções, apesar da necessidade de restrições e/ou adaptações descritas em 3.4.1, 3.4.2 e 3.4.3, devido à limitação de hardware e material disponível, uma vez que o kit LEGO EV3, apesar de apresentar componentes de alta qualidade, ainda assim, constitui um sistema majoritariamente para fins hobbistas e casuais.

Para trabalhos futuros, seria desejável a utilização de hardwares exclusivamente voltados para atividades acadêmicas e/ou profissionais, por exemplo, o dispositivo embarcado myRIO desenvolvido pela National Instruments. Tal hardware, permitiria maior usabilidade dos recursos disponibilizados pelo LabView, não havendo limitações quanto ao uso de bibliotecas exclusivamente desenvolvidas para kit LEGO Mindstorms EV3. Além disso, no tratamento dos sinais do giroscópio, faz-se como alternativa no mínimo igualmente eficiente, a utilização de um filtro de Kalman.

REFERÊNCIAS

- AKMAL, M.; JAMIN, N.; GHANI, N. A. Fuzzy logic control for two wheeled ev3 lego robot. *2017 IEEE Conference on Systems, Process and Control(ICSPPC 2017)*, 15–17 December 2017, Melaka, Malaysia, 2017. n. 17630460, p. 134–139, 2017. Citado na página 15.
- ALVES, R. G. *Conrtrole de um pêndulo invertido utilizando técnica de linearização por realimentação*. Monografia de graduação em engenharia de controle e automação — Universidade Federal de Ouro Preto, 2010. Citado na página 20.
- ARÉVALO-CASTIBLANCO, M. F. et al. Controlador lqr y smc aplicado a plataformas penbdulares. *Revista Iberoamericana de Informática Industrial*, 2018. p. 374–383, 2018. Citado na página 17.
- DORF, R. C.; BISHOP, R. H. *Modern Control Systems*. Davis, CA, USA: Pearson, 2017. Citado na página 18.
- FAN, Z. L. . C. Y. . L. *Advanced Control of Wheeled Inverted Pendulum Systems*. London, U.K.: Springer-Verlag London, 2013. Citado 2 vezes nas páginas 9 e 25.
- JOSEFSSON MATTIAS; PERSSON, J. S. A. W. J. N. T. *Self - Balancing Lego Robot*. 2017. Lund University. Disponível em: <<http://alexanderskafte.com/static/lego.pdf>>. Acesso em: 16 jul. 2019. Citado 4 vezes nas páginas 9, 19, 26 e 29.
- KANIESKI, J. M. *Modelagem e Controle LQR Aplicado a um Condicionador de Energia*. Dissertação (Dissertação de Mestrado) — Universidade Tecnológica Federal do Paraná, 2010. Citado na página 15.
- LEGO System A/S. *Página Inicial\LEGO.comBR*. 2021. Disponível em: <https://www.lego.com/pt-br/>. Acesso em: 05 novembro 2021. Citado 4 vezes nas páginas 9, 22, 23 e 24.
- Mathworks®. *Linear-Quadratic Regulator (LQR) design - MATLAB lqr mathworks.com*. 2021. Disponível em: <https://www.mathworks.com/help/control/ref/lqr.html#f3-411186>. Acesso em: 10 julho 2020. Citado na página 29.
- NISE, N. S. *Engenharia de Sistemas Controle*. New York, NY, USA: LTC, 2020. Citado 3 vezes nas páginas 9, 17 e 18.
- OGATA, K. *Engenharia de Controle Moderno*. 5. ed. USA: Pearson, 2010. Citado 3 vezes nas páginas 15, 18 e 21.
- PER, J.; ALI, P.; OLOV, R. Two wheeled balancing robot. 2009. 2009. Citado na página 15.
- SILVA, G. D. *Projeto, Modelagem,, Simulação, Construção e Controle de um Veículo de Duas Rodas Pêndulo Invertido*. Monografia de graduação em engenharia mecatrônica — Universidade Federal de Uberlândia, 2017. Citado 2 vezes nas páginas 9 e 17.
- ZOMETA, P. et al. Implementation aspects of model predictive control for embeeded systems. *2012 American Control Conference*, 2012. p. 1205–1210, 2012. Citado na página 15.

APÊNDICE A – PARÂMETROS DO SISTEMA

```

1  % Modelagem do Segway;
2  clear; clc;
3  g=9.81;
4  m=0.023;
5  R=0.06;
6  Jw=m*R^2/2;
7  M=0.636;
8  W=0.15;
9  D=0.06;
10 H=0.22;
11 L=H/2;
12 Jv=M*L^2/3;
13 Jo=M*(W^2+D^2)/12;
14 Jm=1*10^-5;
15 Rm=6.69;
16 Kb=0.468;
17 Kt=0.317;
18 n=1;
19 fm=0.0022;
20 fw=0;
21 alpha=n*Kt/Rm;
22 beta=n*Kt*Kb/Rm + fm;
23 E=[(2*m+M)*R^2+2*Jw+2*n^2*Jm    M*L*R-2*n^2*Jm
24     M*L*R-2*n^2*Jm              M*L^2+Jv+2*n^2*Jm];
25 A1=[0 0 1 0;0 0 0 1];
26 A1(3,2)=-g*M*L*E(1,2)/det(E);
27 A1(4,2)=g*M*L*E(1,1)/det(E);
28 A1(3,3)=-2*[(beta+fw)*E(2,2)+beta*E(1,2)]/det(E);
29 A1(4,3)=2*[(beta+fw)*E(1,2)+beta*E(1,1)]/det(E);
30 A1(3,4)=2*beta*[E(2,2)+E(1,2)]/det(E);
31 A1(4,4)=-2*beta*[E(1,1)+E(1,2)]/det(E);
32 B1(3)=alpha*[E(2,2)+E(1,2)]/det(E);
33 B1(4)=-alpha*[E(1,1)+E(1,2)]/det(E);
34 B1=B1';
35 B1(:,2)=B1;
36 C1=eye(4);

```

```
37 D1=[0 0 0 0;0 0 0 0]';
38 %Inicia o setup da simulacao;
39 AngRoda=0;
40 AngCorpo=0;
41 VelAngRoda=0;
42 VelAngCorpo=0;
43 vel1=0;
44 vel2=0;
45 AngRot=0;
46 VelAngRot=0;
47 x1=[AngRoda AngCorpo VelAngRoda VelAngCorpo]';
48 x2=[AngRot VelAngRot]';
49 u=[vel1 vel2]';
```

APÊNDICE B – SIMULAÇÃO EM MALHA ABERTA COM APLICAÇÃO DE IMPULSO COM AMPLITUDE 1 DE GRAU

```

1 %Gera sinal de impulso nos estados x;
2 x1=zeros(4,1000);
3 x1(2,10)=1*pi/180; % Gera um impulso de 1 grau no angulo do
   corpo do Segway%
4 %Simula em malha aberta%
5 dt=0.01;
6 for i=1:1000
7     X(:,i)=A1*x1(:,i)+B1*u;
8     if i>9
9         x1(:,i+1)=X(:,i);
10    end
11 end
12 subplot(2,2,1)
13 plot(X(1,:));
14 ylabel('velangcorpo');
15 xlabel('time');
16 subplot(2,2,2)
17 plot(X(2,:));
18 ylabel('posangcorpo');
19 xlabel('time');
20 subplot(2,2,3)
21 plot(X(3,:));
22 ylabel('velangroda');
23 xlabel('time');
24 subplot(2,2,4)
25 plot(X(4,:));
26 ylabel('posangroda');
27 xlabel('time');

```

APÊNDICE C – PROJETO DO CONTROLADOR LQR

```
1 %Projeta o controlador
2 Q = [0.6      0      0      0
3       0    35000      0      0
4       0      0    0.001      0
5       0      0      0    40];
6
7
8 R = eye(2,2)*10000;
9 [Klqr,S,e]=lqr(A1,B1,Q,R);
```

APÊNDICE D – SIMULAÇÃO EM MALHA FECHADA COM APLICAÇÃO DE IMPULSO COM AMPLITUDE DE 1 GRAU

```

1 %Gera sinal de impulso nos estados x;
2 x1=zeros(4,1000);
3 x1(2,10)=1*pi/180; % Gera um impulso de 1 grau no angulo do
   corpo do Segway
4 %Simula em malha fechada
5 dt=0.01;
6 for i=1:1000
7     X(:,i)=A1*x1(:,i)+B1*u;
8     if i>9
9         x1(:,i+1)=x1(:,i)+X(:,i)*dt;
10        %Atualiza Estado do Segway
11    end
12    u=-Klqr*[x1(:,i)+X(:,i)*dt]; % Multiplica o ganho pelo
   estado atual do Segway.
13    tensao(i,:)=u;
14 end
15 figure();
16 subplot(2,2,1)
17 plot(X(1,:))
18 ylabel('velangcorpo');
19 xlabel('time');
20 subplot(2,2,2)
21 plot(X(2,:))
22 ylabel('posangcorpo');
23 xlabel('time');
24 subplot(2,2,3)
25 plot(X(3,:))
26 ylabel('velangroda');
27 xlabel('time');
28 subplot(2,2,4)
29 plot(X(4,:))
30 ylabel('posangroda');
31 xlabel('time');

```