



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

TELSON GALDINO PINHEIRO TEIXEIRA

**MODELAGEM E CONTROLE DE ROBÔ
PÊNULO INVERTIDO SOBRE RODAS**

TUCURUÍ/PA
2023

TELSON GALDINO PINHEIRO TEIXEIRA

**MODELAGEM E CONTROLE DE ROBÔ
PÊNULO INVERTIDO SOBRE RODAS**

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

Orientador: Prof. Dr. Raphael Barros Teixeira
Universidade Federal do Pará

TUCURUÍ/PA
2023

TELSON GALDINO PINHEIRO TEIXEIRA

**MODELAGEM E CONTROLE DE ROBÔ
PÊNULO INVERTIDO SOBRE RODAS**

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

DATA DE APROVAÇÃO: 17/01/2023

CONCEITO: Excelente

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

Prof. Dr. Rafael Suzuki Bayma
Membro - UFPA - CAMTUC - FEE

Prof. Dr. André Felipe Souza da Cruz
Membro - UFPA - CAMTUC - FEE

TUCURUÍ/PA
2023

Dedico aos meus pais e a toda a minha família.

AGRADECIMENTOS

A Deus, "Porque, quando perco toda a minha força, então tenho a força de Cristo em mim."

2 Coríntios 12.10.

“Se você quiser descobrir os segredos do Universo, pense em termos de energia, frequência e vibração.” (Nikola Tesla)

RESUMO

O robô pêndulo invertido sobre rodas é uma planta clássica de sistemas de controle, não linear e instável em malha aberta. Neste trabalho é realizada a modelagem utilizando leis físicas fundamentais das partes constitutivas do robô: o pêndulo, a roda e os motores de ativação. A agregação das equações destas partes conduz a um modelo de quarta ordem que expressa as propriedades dinâmicas do sistema. A estratégia de controle adotada foi do tipo realocação de polos do modelo no espaço de estados. Em se tratando de uma técnica linear, houve a necessidade de realizar a linearização do modelo em torno do ponto de operação desejado para o sistema. O modelo linear de quarta ordem mostrou-se controlável, porém não-observável, de maneira que foi necessário excluir uma das variáveis do modelo conduzindo a uma estrutura controlável e observável de terceira ordem. O projeto de controlador e observador foram bem sucedidos, e resultados computacionais do desempenho do sistema controlado foram levantados. O trabalho apresenta ainda a descrição, projeto e implementação da planta real, a ser utilizada na implementação da estratégia de controle em trabalhos futuros.

Palavras-chave: Sistemas de Controle. Robô Pêndulo Invertido Sobre Rodas. Modelo no Espaço de Estados.

ABSTRACT

The inverted pendulum robot on wheels is a classic plant of control systems, non-linear and unstable in open loop. In this work, modeling is carried out using fundamental physical laws of the constituent parts of the robot: the pendulum, the wheel and the activation motors. The aggregation of the equations of these parts leads to a fourth-order model that expresses the dynamic properties of the system. The adopted control strategy was of the model pole reallocation in state space. Since this is a linear technique, there was a need to linearize the model around the desired operating point for the system. The fourth-order linear model proved to be controllable, but unobservable, so that it was necessary to exclude one of the variables from the model, leading to a third-order controllable and observable structure. The controller and observer design were successful, and computational results of the controlled system performance were obtained. The work also presents the description, design and implementation of the real plant, to be used in the implementation of the control strategy in future works.

Keywords: Control System. Inverted Pendulum Robot on Wheels. Model no Space of States.

LISTA DE FIGURAS

Figura 1.1 – Robô Pêndulo Invertido	13
Figura 1.2 – (a) Segway (b) Robô de teleconferência (c) Foguete	14
Figura 2.1 – Robô Pêndulo - (a) Vista Lateral (b) Vista Frontal	18
Figura 2.2 – Pêndulo - (a) Vista Lateral (b) Vista Frontal	19
Figura 2.3 – Roda - (a) Vista Lateral (b) Vista Frontal	21
Figura 2.4 – Circuito elétrico do motor cc	22
Figura 2.5 – Pêndulo Robô Malha Aberta	27
Figura 2.6 – Simulação Pêndulo Robô Malha Aberta	28
Figura 2.7 – Gráfico de análise de Variáveis	28
Figura 3.1 – Malha Aberta	32
Figura 3.2 – Malha Fechada	33
Figura 3.3 – Pêndulo Robô Malha Fechada	36
Figura 3.4 – Simulação Pêndulo Robô Malha Fechada	37
Figura 3.5 – Gráfico de análise de Variáveis Malha Fechada	37
Figura 3.6 – Malha Fechada Sem Saturação	38
Figura 3.7 – Gráfico de Análise do Sinal de Saída do Controlador	38
Figura 3.8 – Malha Fechada Com Saturação	39
Figura 3.9 – Gráfico de Análise do Sinal da Saída do Controlador Com Saturação	39
Figura 3.10 – Gráfico de Análise das Variáveis com a Saturação em Malha Fechada	40
Figura 3.11 – Medição dos Estados do Sistema por Sensores	41
Figura 3.12 – Observadores de Estado	41
Figura 3.13 – Observador de Estado	45
Figura 3.14 – Gráfico de Análise do Observador de Estado	45
Figura 4.1 – Robô Pêndulo Invertido Sobre Rodas	47
Figura 4.2 – Conexão Arduino + MPU 6050	48
Figura 4.3 – Conexão Arduino + Etapa de Potência	48
Figura 4.4 – Plataforma de Prototipagem Arduino Uno	49
Figura 4.5 – IMU MPU-6050	50
Figura 4.6 – Sistemas Microeletromecânicos	51
Figura 4.7 – Giroscópio	51
Figura 4.8 – Driver L298N	52
Figura 4.9 – Motor DC 6V 210RPM com Encoder	53
Figura 4.10 – Pinagem do Motor	53

LISTA DE TABELAS

Tabela 4.1 – Parâmetros do Robô Pêndulo Invertido Sobre Rodas	54
---	----

LISTA DE SÍMBOLOS

g	Gravidade
L	Comprimento do pêndulo
r	Raio da roda
h	Altura
m_p	Massa do pêndulo
m_r	Massa da roda
R_a	Resistência de armadura
n	Fator da caixa de redução
V_n	Tensão nominal
V_e	Tensão da força contra eletromotriz
I_a	Corrente nominal
N	RPM
K_e	Constante elétrica
T_m	Torque
K_t	Constante de torque
J_p	Inércia do pêndulo
J_r	Inércia do roda
ψ	Angulo do pêndulo
$\dot{\psi}$	Velocidade angular do pêndulo
$\ddot{\psi}$	Aceleração angular do pêndulo
θ	Angulo do roda
$\dot{\theta}$	Velocidade angular do roda
$\ddot{\theta}$	Aceleração angular do roda

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	15
1.3	Objetivos Específicos	16
1.4	Organização do texto	16
2	MODELAGEM E SIMULAÇÃO EM MALHA ABERTA	17
2.1	Modelo Matemático do Pêndulo Invertido Sobre Rodas	18
2.2	Modelo físico do pêndulo: Corpo do Robô	19
2.3	Modelo físico da roda	20
2.4	Modelo Elétrico dos Motores cc	22
2.5	Modelo dinâmico no espaço de estados	23
2.5.1	Equações de estado não lineares	26
2.6	Sistema em Malha Aberta	27
3	CONTROLE NO ESPAÇO DE ESTADOS	30
3.1	Linearização do modelo	30
3.1.1	Análise de estabilidade	32
3.2	Alocação de Polos	33
3.2.1	Controlabilidade	34
3.2.2	Projeto do Controlador - Fórmula de Ackermann	35
3.3	Projeto de Controlador por Alocação de polos	35
3.4	Simulação e Análise do Pêndulo Robô Malha Fechada	36
3.5	Observadores de Estado	40
3.6	Estratégia com modelo de ordem reduzida	43
3.6.1	Fórmula de Ackermann para o Observador	44
3.7	Requisitos de Projeto para Polos Desejados do Observador	44
3.8	Simulação e Análise do Pêndulo Robô Observador	44
4	DESCRIÇÃO DO SISTEMA PÊNDULO INVERTIDO SOBRE RODAS	47
4.1	Sistema Mecânico	47
4.2	Sistema Elétrico	47
4.3	Plataforma de Prototipagem Arduino Uno	48
4.4	Unidade de Medição Inercial (IMU)	49
4.5	Ponte H (Controle de Motores)	52
4.6	Atuadores	52
4.7	Parâmetros do sistema	53
5	CONCLUSÃO	55
	REFERÊNCIAS	56
A	ANEXO: CÓDIGOS MATLAB 2016A	58
A.1	Códigos Modelagem Pêndulo Robô	58

A.2	Códigos Projeto do Controlador	64
A.3	Códigos do Projeto Controlador Observador	69
B	ANEXO: CIRCUITOS NO SIMULINK	76
B.1	Simulação Malha Aberta	76
B.2	Simulação Malha fechada	78
B.3	Simulação Controlador Observador	81

1 INTRODUÇÃO

Robôs autônomos desempenham um papel cada vez mais importante, principalmente para atender a alta demanda por sistemas de automação na indústria e espera-se autonomia e capacidade de detectar informações sobre o ambiente, processar informações e executar ações apropriadas para uma ampla gama de tarefas, que pode ser da mais simples atividade até ao alto nível de complexidade em realizar determinado trabalho.

Um dos maiores desafios quando se trata de atuação de programação dos serviços robóticos, são as técnicas tradicionais de controle geralmente exigem um modelo matemático preciso do sistema e seu ambiente, e sendo inapropriado, é claro, causará efeitos negativos associados à atuação. O projeto de controladores robóticos por técnicas tradicionais de controle costuma ser baseada na construção de um modelo do sistema e no uso de uma técnica adequada para o controlador executar e realizar as tarefas desejadas.

Neste trabalho é realizado o estudo de modelagem e controle de um robô pêndulo invertido sobre rodas, que será referido neste texto como robô. Este é um sistema didático clássico da engenharia de controle, onde um robô busca equilibrar a posição de um pêndulo a partir do movimento unidimensional das suas rodas. Esse modelo de projeto com uma dinâmica não linear é comumente usado em pesquisas que estudam sobre a eficiência, propostas de melhorias e robustez dos controladores.

Figura 1.1 – Robô Pêndulo Invertido



Fonte: (BALANDUINO, 2022)

A Figura 1.1 apresenta um robô pêndulo invertido desenvolvido em (BALANDUINO, 2022). A complexidade dinâmica deste sistema, devido não-linearidade e a instabilidade, tem levado ao estudo de diferentes técnicas de controle visando obter o equilíbrio dinâmico. Em (GOMIDE; GUDWIN, 1994), por exemplo, é utilizada uma estratégia de controle fuzzy que manuseia informações qualitativas, e tenta replicar o conhecimento e a perspectiva do projetista visando a regulação deste sistema.

Uma abordagem como esta, faz pouco uso do modelo matemático do sistema. O qual exigiria um detalhamento minucioso da iteração das diversas partes e variáveis desta planta. A abordagem deste trabalho segue a clássica perspectiva de se determinar um modelo a partir do conhecimento da física do sistema, para então se projetar controladores

baseados na teoria de controle linear. O projeto explora conceitos abrangentes de análise com base na teoria de controle clássico e na teoria de controle moderno.

Aplicações que podem ser considerados uma variação de um pêndulo invertido e são capazes de utilizar das técnicas de controle que foram desenvolvidas para ele são os veículos de transporte pessoal de duas rodas e auto-equilíbrio fabricado pela Segway, outro tipo é o famoso robô de telepresença para acompanhar eventos e reuniões de forma remota e por último dos exemplo é o de controle de posição de um foguete no momento de lançamento, mostrado respectivamente na figura 1.2.

Figura 1.2 – (a) Segway (b) Robô de teleconferência (c) Foguete



Fonte: (SEGWAY, 2017; BOT, 2023; MALIK, 2018)

Neste trabalho também foi desenvolvido um protótipo real de um robô pêndulo invertido, a partir do qual foram medidos os parâmetros utilizados no modelo determinado e no projeto e simulação de controle, ambos realizados em ambiente computacional com linguagem MATLAB.

1.1 Motivação

Visando o objetivo de se controlar este sistema, existe a necessidade de se desenvolver um modelo matemático que explique a sua dinâmica, na qual pode ser descrita em termos de equações diferenciais. Essas equações diferenciais são obtidas pelas leis físicas fundamentais, por exemplo, as leis de Newton para sistemas mecânicos e as leis de Kirchhoff para sistemas elétricos, e para esta modelagem o sistema do robô foi dividida em 3 subsistemas que são: pêndulo, roda e motores.

O Modelo resultou em uma dinâmica no espaço de estados, não linear, de quarta ordem tendo como variáveis de estados a posição da roda θ , posição do pêndulo ψ , velocidade da roda ω e velocidade do pêndulo ϕ . O resultado da análise na simulação em malha aberta mostra que este sistema é instável, então, o projeto necessita de um controlador visando a regulação da posição do pêndulo na vertical, o que implica na necessidade de estabilização do sistema, para tal fim, a estratégia de controle investigada é

a de realocação de polos por realimentação de estados, que se baseia na ideia de que todas as variáveis de estado sejam mensuráveis e que estejam disponíveis para realimentação. A linearização do modelo do robô é necessária para o projeto deste controlador, de modo que um modelo linear foi determinado sobre o ponto de equilíbrio desejado para este sistema. Outro fator principal para a utilização deste tipo de controle é a análise de controlabilidade que consiste em avaliar se o sistema é controlável. Se este for o caso então é possível determinar o vetor de ganho tal que os autovalores da matriz de estado estejam onde se deseje.

No problema de alocação de polos é necessário o acesso (medições) de todos os estados da planta para realimentação, o que exige a necessidade da utilização observadores de estado, de modo que este dispositivo tem por objetivo estimar as variáveis de estado com base nas medidas da saída e da ação de controle. Porém, o sistema de quarta ordem mostrou-se não observável, desse modo, a estratégia de projeto foi abrir mão de uma das variáveis de estado do modelo, em que se eliminou a variável posição da roda. Este procedimento foi adotado partindo da hipótese de que a variável em questão pode estar em qualquer lugar e mesmo assim o robô continuará a equilibra-se, por consequência, formulando-se então um modelo de ordem reduzida com três variáveis de estado, que se mostrou controlável e observável, portanto, os projetos do controlador e do observador de estados foram concluídos, seguindo abordagem convencional de projetos de controladores no espaço de estados, além do mais, é feita especificação dos requisitos de polos desejados em malha fechada, visando tanto a garantia da estabilidade quanto um tempo de acomodação presumivelmente apropriado para estabilizar o sistema, contudo, no intuito de dar mais realismo a avaliação do projeto, foi inserido ainda na malha de controle uma saturação para o sinal de controle, isto por que a alimentação do sistema real possui uma fonte de alimentação limitada a 12V. Os resultados das simulações computacionais mostraram a viabilidade da estratégia adotada para o projeto de controle, uma vez que quando o robô é solto próximo ao ponto especificado, o mesmo converge para sua estabilidade na vertical, ademais, o trabalho também buscou avaliar o projeto do controlador no sistema real. No entanto os experimentos não obtiveram êxito em estabilizar o sistema real com a implementação em microcontrolador do controlador projetado, entretanto, o protótipo real mostrou-se funcional em suas partes constituintes: O pêndulo, o acionamento das rodas, o banco de bateria, a circuitaria eletrônica para acionar os motores a partir dos sinais do microcontrolador, a aquisição de sinais dos sensores e a própria estrutura física do robô.

Diante disso, este protótipo foi importante, porém para consolidar a perspectiva física do sistema, presume-se que exista a necessidade de uma melhor calibração de sensores e medição mais precisa dos parâmetros do sistema, o que deverá ser realizado em investigações futuras que busquem avaliar o projeto de controlador no sistema real.

1.2 Objetivos

O foco principal do presente trabalho é o desenvolvimento por simulação para o controle de um sistema robótico não linear que tenha a capacidade de auto controle do equilíbrio sobre rodas em determinada faixa de ajuste, por métodos de modelo matemático da engenharia de controle moderno. O objetivo final é demonstrar o desempenho de controle de acionamentos robóticos com técnicas de controle clássica, em especial o controle utilizando observador de estado e desenvolver ambientes que viabilizem a avaliação dos algoritmos de acionamento para robôs auto-equilibrantes. Tais ambientes poderão ser usados em pesquisas e no ensino de Engenharia de Controle, bem como encorajar estudantes e

profissionais a explorar a utilização das técnicas aqui apresentadas para alcançar resultados melhores em suas aplicações.

1.3 Objetivos Específicos

- Revisar a bibliografia e o estado da arte do sistema de pêndulo invertido;
- Elaborar um modelo matemático que descreve o comportamento do sistema de pêndulo invertido;
- Projetar um controlador para realizar a ação de balanceamento do robô;
- Identificar os aspectos necessários para obtenção dos parâmetros do controlador;
- Simular e analisar os resultados.

1.4 Organização do texto

Os assuntos neste trabalho estão organizados nos seguintes capítulos:

No capítulo 2 será tratada a modelagem e simulação em malha aberta do robô, o qual é dividido em 3 partes: o pêndulo caracterizando o corpo da estrutura, as rodas e dos motores. A representação do modelo é desenvolvida em espaço de estado e apresenta as equações de estado não lineares. A simulação em malha aberta é executada e o resultado da análise exibe um sistema instável e não linear, portanto o projeto tem a necessidade da utilização de uma técnica de controle dinâmico.

No capítulo 3 é realizado a linearização e definido o método de controle por alocação de polos, por conseguinte é feito o estudo de controlabilidade e posteriormente a simulação e análise em malha fechada do sistema do robô. Neste capítulo também será tratada as medições dos vetores de estado por dispositivo denominado observador de estado, no qual apresentou uma divergência de não observabilidade, o que levou a mais uma subseção com um tipo de estratégia com modelo de ordem reduzida, por fim implementado os requisitos de projeto e executada a simulação e análise do observador de estado.

O capítulo 4 apresenta a descrição física do sistema mecânico e elétrico do robô. Os principais dispositivos que compõe o circuito do projeto são identificados e exposto um breve relato das configurações das conexões e suas funcionalidades, enfim a formulação com os parâmetros que envolve o modelo do sistema.

O Capítulo 5 é a Conclusão, neste capítulo argumenta as conclusões obtidas a partir dos projetos de controle do robô pêndulo invertido realizados no Capítulo 3 e indica propostas de desenvolvimento de trabalhos futuros, baseando-se no que foi realizado neste presente trabalho de conclusão de curso.

2 Modelagem e Simulação em Malha Aberta

O pêndulo robô sobre rodas é um desses projetos que as pessoas olharão, mas não conseguirão enxergar toda a complexidade da engenharia envolvida por trás de uma simples massa ficar em equilíbrio sobre rodas. A princípio o raciocínio era que bastaria utilizar o acelerômetro para chegar ao equilíbrio, caso o ângulo aumente para um lado, aumentar a rotação do motor em uma direção. Se o ângulo aumentar para o outro lado, aumenta a rotação do motor no sentido contrário. Porém o equilíbrio, ou controle deste sistema, não se dá de forma tão simples, sendo necessárias técnicas de controle sofisticadas para implementá-la.

Para o controle do robô por meio de tentativas levaria muito tempo e talvez conseguiria algum resultado, mas sem uma explicação razoável do fenômeno do auto-controle de um robô com características de um pêndulo invertido sobre rodas.

O projeto de controle de sistemas dinâmicos em engenharia requer tipicamente um modelo matemático do sistema a ser controlado. Este modelo é criado para se simular algumas características de um problema real. Segundo (AGUIRRE, 2004), existem várias maneiras de se obter um modelo matemático, uma delas é modelagem caixa branca, a qual é a modelagem que considera apenas as variáveis de entrada e a física do sistema. Uma outra maneira é a identificação de sistemas conhecida como modelagem caixa preta, modelagem que considera apenas as variáveis de entrada e a saída do sistema.

Segundo (OGATA et al., 2010), Devido a sua característica instável, o pêndulo invertido é um problema clássico de dinâmica e teoria de controle, que representa um sistema não-linear, pode-se relacionar esse modelo ao controle de posição de um foguete no momento de lançamento, em que o objetivo do problema de controle é manter o foguete em sua posição vertical. Já (DORF; BISHOP, 2009), fazem analogias entre o pêndulo invertido e a situação em que uma pessoa equilibra o cabo de vassoura em sua mão ou controlar a atitude de um míssil no decorrer dos estágios iniciais do lançamento. Em um sistema com características dinâmicas desafiadoras, como é o caso do pêndulo invertido sobre rodas, o modelo precisa exibir estas propriedades, particularmente a questão da não linearidade e da instabilidade, que são inerentes a este sistema.

O pêndulo invertido é amplamente utilizado em estudos das metodologias de controle (BONAFILIA et al., 2015) devido às características instáveis e as não linearidades do sistema. Esse sistema não linear é comumente usado em pesquisas que estudam sobre a eficiência, propostas de melhorias e robustez dos controladores. Como o sistema é instável, a sua modelagem se dá tradicionalmente aplicando leis físicas sobre as suas partes fundamentais, a qual buscará então explicar a dinâmica do pêndulo, a dinâmica da roda e a dos motores elétricos responsáveis pelo acionamento do sistema.

Tendo em perspectiva os modelos tradicionalmente explorados no curso de graduação em engenharia elétrica, a representação adequada para o pêndulo é na forma de espaço de estados.

”Os modelos matemáticos podem assumir diferentes formas. Dependendo do sistema considerado e das circunstâncias particulares, um modelo matemático pode ser mais adequado que outro. Uma vez obtido o modelo matemático de um sistema, podem ser utilizadas várias ferramentas analíticas de computação para efeito de análise e síntese”(OGATA et al., 2010).

2.1 Modelo Matemático do Pêndulo Invertido Sobre Rodas

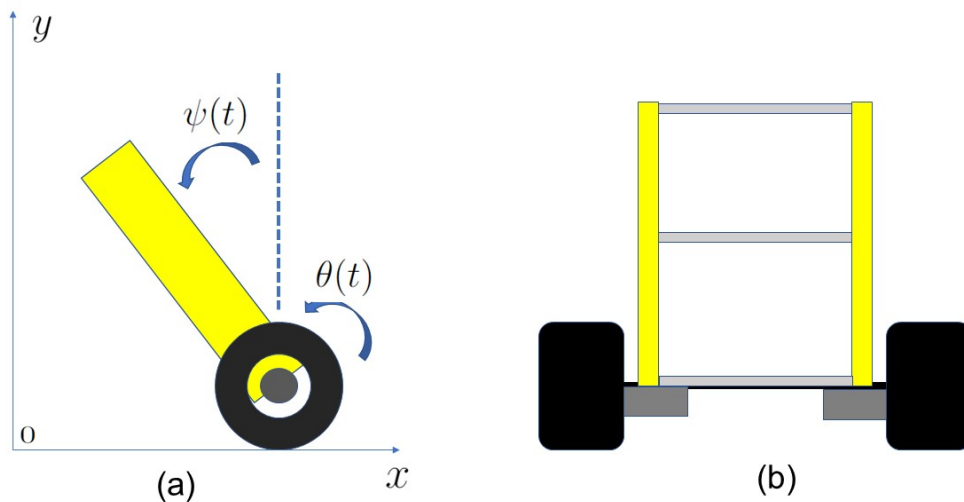
O robô equilibrista é um sistema clássico de controle dinâmico cujo objetivo é controlar a posição angular de um pêndulo invertido sobre rodas a partir da atuação de motores elétricos e da conseqüente mudança de posição das rodas.

Segundo (VENDRAMINI; SILVA, 2010), esse sistema ilustra as dificuldades práticas associadas com aplicações de sistemas de controle no mundo real, sendo portanto, de interesse para os estudos em tecnologia de controle .

O modelo matemático deste sistema é parte fundamental para a compreensão do seu comportamento dinâmico e para o projeto de controladores. Ao buscar manter o pêndulo na vertical, a estratégia de controle visa então associar o centro de massa a um ponto de equilíbrio. Em malha aberta, esta posição de equilíbrio vertical é instável, pois ao soltar o robô pêndulo o mesmo tende a cair. A Figura 2.1 apresenta uma concepção deste sistema, onde se destacam duas das principais variáveis de interesse: a posição angular da roda $\theta(t)$ e a posição angular do pêndulo $\psi(t)$. Controlar o robô pêndulo invertido sobre rodas é portanto manter $\psi(t) \approx 0$, ou seja, uma estratégia de regulação onde a posição do pêndulo(o corpo do robô) precisa se manter na vertical.

De acordo com (BURNS, 2001) é possível descrever as equações de movimento do pêndulo invertido a partir do diagrama de corpo livre do sistema, ilustrado na figura 2.1

Figura 2.1 – Robô Pêndulo - (a) Vista Lateral (b) Vista Frontal



Fonte: (Do autor 2022.)

Desse modo, a fim de analisar o comportamento dinâmico e proceder com o projeto de controladores, serão desenvolvidos modelos matemáticos do robô por espaço de estados (OGATA et al., 2010). O modelo matemático compreende uma série de equações e considerações, as quais foram acopladas para construir as equações de estado a serem utilizadas.

Nesse sentido, o modelo do robô pêndulo invertido de eixo único é modelado matematicamente a fim de conseguir o modelo em espaço de estados, que posteriormente é convertido a representação em função de transferência, buscando relacionar a saída

$\psi(t)$ com a entrada $u(t)$, sendo esta a tensão aplicada nos atuadores do sistema, motores elétricos que farão a base do pêndulo se mover buscando regular a posição.

O modelo proposto será desenvolvido a partir da física do sistema, ou seja, o equacionamento será dado pela aplicação de leis físicas envolvendo as variáveis de interesse em cada parte do robô. Neste caso, se destacam três partes principais que serão abordadas isoladamente: (i) O pêndulo ou corpo do robô, (ii) a roda ou base do robô e (iii) Os motores ou atuadores do sistema.

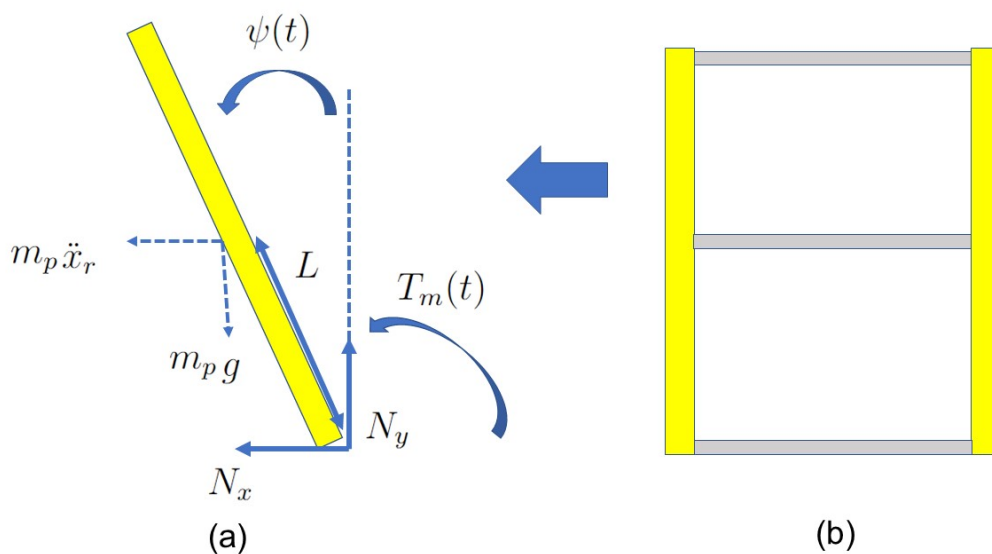
Esta abordagem simplifica a estratégia de modelagem ao focar em partes mais simples do sistema maior. O desenvolvimento das equações naturalmente irá acoplar os modelos destas três partes em uma única representação de estados do sistema.

2.2 Modelo físico do pêndulo: Corpo do Robô

A Figura 2.2 apresenta um diagrama esquemático destacando as forças que atuam sobre o pêndulo, ou seja o corpo do robô. Este diagrama é desenvolvido tendo como referencial um eixo em coordenadas cartesianas (x,y) , sob o qual as variáveis de interesse serão medidas. No diagrama são destacadas os seguintes parâmetros e variáveis:

- $\psi(t)$ - Posição angular do pêndulo;
- $T_m(t)$ - Momento angular sobre o eixo do pêndulo;
- g - Gravidade;
- m_p - Massa do pêndulo;
- \ddot{x}_r - Aceleração na direção x da roda;
- N_x - Força entre o pêndulo e a roda na direção x ;
- N_y - Força entre o pêndulo e a roda na direção y ;
- L - Distância do centro da roda ao centro do pêndulo ;

Figura 2.2 – Pêndulo - (a) Vista Lateral (b) Vista Frontal



Considerando que se trata de um sistema mecânico com movimentos rotacionais, será aplicada a Lei de Newton de rotação a qual estabelece que a soma dos momentos angulares no ponto de rotação do pêndulo é o produto da inércia J pela aceleração angular. Tomando como base a Figura 2.2, é possível verificar que existem três parcelas de momento angular anti-horário atuantes: $T_m(t)$; um conjugado devido a componente da força peso atuante sobre o centro de massa do pêndulo, dada por $T_p(t) = m_p g L \sin\psi(t)$ e uma componente que se expressa devido ao movimento horizontal do pêndulo, sendo portanto proporcional a sua velocidade de translação sobre o eixo x , dada por $T_v(t) = m_p \ddot{x}_r L \cos\psi(t)$. Isto resulta na equação 2.1 que relaciona a $\psi(t)$ com a resultante do momento angular anti-horário.

$$J\ddot{\psi}(t) = T_m(t) + m_p g L \sin\psi(t) + m_p \ddot{x}_r L \cos\psi(t) \quad (2.1)$$

Por sua vez, a dinâmica de translação do pêndulo pode ser dada pelas equações 2.2 e 2.3, que descrevem a resultante das forças atuantes nos eixos x e y , respectivamente.

$$-N_x - m_p \ddot{x}_r = m_p \ddot{x}_p \quad (2.2)$$

$$N_y - m_p g = m_p \ddot{y}_p \quad (2.3)$$

A fim de encontrar as equações do pêndulo utilizando um referencial angular, expresso em coordenadas cilíndricas, será tomada uma mudança de variável de modo a substituir as variáveis x e y do modelo. Isto é realizado expressando o centro de gravidade do pêndulo (x_p, y_p) em função de $\psi(t)$ o que resulta:

$$x_p(t) = -L \sin\psi(t) \quad (2.4)$$

$$y_p(t) = L \cos\psi(t) \quad (2.5)$$

Derivando estas equações de posição, obtém-se as equações para as velocidades do ponto central:

$$\dot{x}_p(t) = -L \dot{\psi}(t) \cos\psi(t) \quad (2.6)$$

$$\dot{y}_p(t) = -L \dot{\psi}(t) \sin\psi(t) \quad (2.7)$$

Derivando novamente para obter as acelerações, onde se omite a variável independente t para simplificar a notação, resulta:

$$\ddot{x}_p = -L \ddot{\psi} \cos\psi + L \dot{\psi}^2 \sin\psi \quad (2.8)$$

$$\ddot{y}_p = -L \ddot{\psi} \sin\psi - L \dot{\psi}^2 \cos\psi \quad (2.9)$$

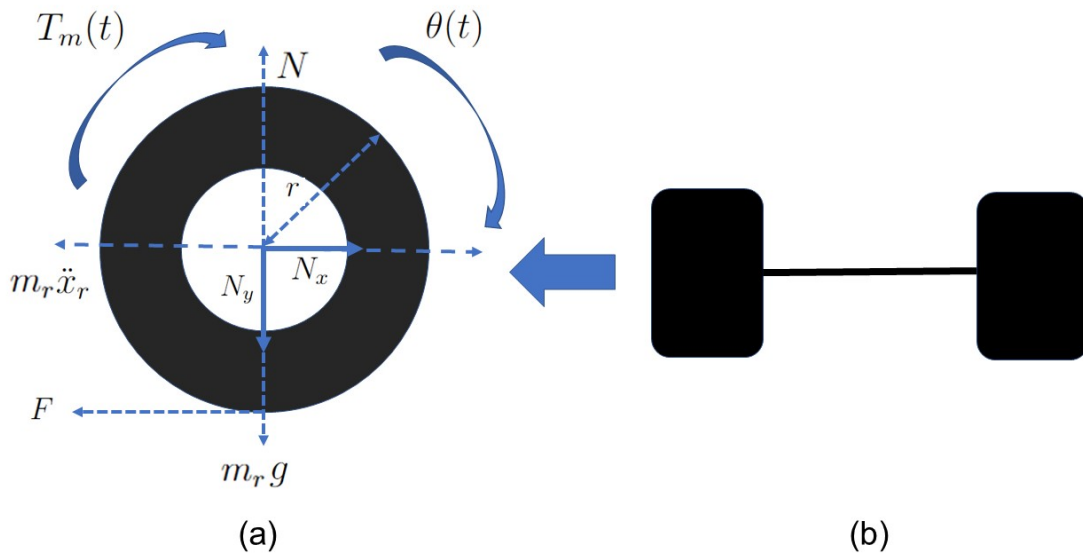
Estas relações serão substituídas posteriormente nas equações dinâmicas visando compor um modelo completo.

2.3 Modelo físico da roda

A Figura 2.3 apresenta as forças atuando sobre a roda, onde são apresentados os seguintes parâmetros e variáveis.

- $\theta(t)$ - Posição angular da roda;
- r - Raio da roda;
- g - Gravidade;
- T_m - Momento angular sobre o eixo do roda;
- N - Força normal da superfície para com as rodas;
- N_x - Força entre o pêndulo e a roda na direção x ;
- N_y - Força entre o pêndulo e a roda na direção y ;
- m_r - Massa da roda;
- \ddot{x}_r - Aceleração na direção x da roda;
- F - Fricção entre a superfície e as rodas;

Figura 2.3 – Roda - (a) Vista Lateral (b) Vista Frontal



Fonte: (Do autor 2022.)

A equação 2.10 modela a aceleração da roda em relação a resultante do momento horário em torno do centro de massa da roda.

$$T_m + Fr = J_r \ddot{\theta} \quad (2.10)$$

Sendo J_r a inércia da roda.

A aceleração \ddot{x}_r pode ser convertida do sistema de coordenadas cartesiana (x,y) para o sistema de coordenadas cilíndrica. As equações 2.11 e 2.12 são as forças atuando na roda nos eixos x e y , respectivamente.

$$N_x - m_r \ddot{x}_r - F = 0 \quad (2.11)$$

$$N - N_y - m_r g = m_r \ddot{y}_r \quad (2.12)$$

A aceleração \ddot{x}_r pode ser também convertida do sistema de coordenadas cartesiana para o sistema de coordenadas cilíndrica, utilizando:

$$\ddot{x}_r = r \ddot{\theta} \quad (2.13)$$

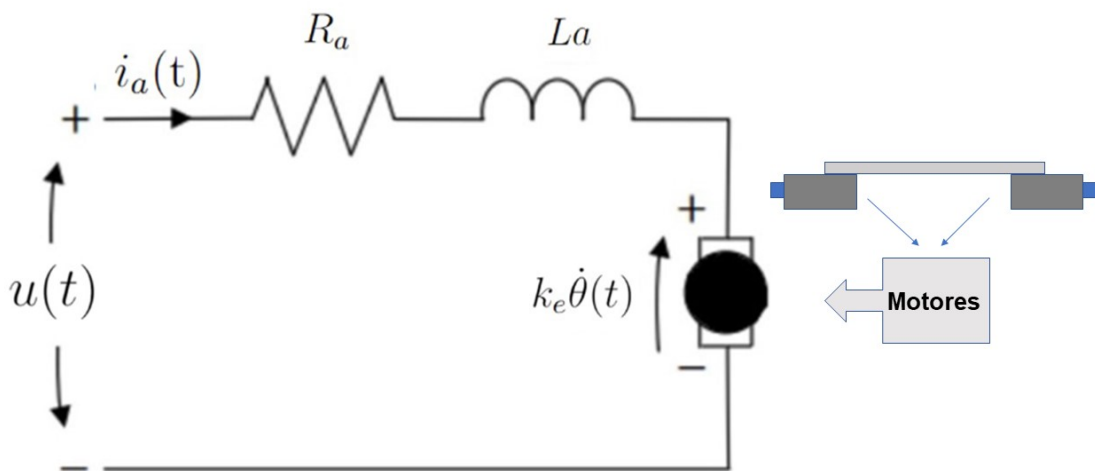
Enquanto a aceleração no sentido do eixo y é igual a 0. Relações estas que serão posteriormente embutidas no modelo geral.

2.4 Modelo Elétrico dos Motores cc

Os atuadores do pêndulo serão dois pequenos motores elétricos de corrente contínua, que buscarão mantendo-o em equilíbrio a partir da atuação sobre a posição da roda. Sendo assim para utiliza-los no modelo matemático é necessário estabelecer uma relação entre a tensão aplicada sobre eles e o torque entregue dos eixo dos motores às rodas.

A equação 2.14 representa a dinâmica elétrica do circuito de armadura de um motor CC no domínio do tempo, obtida pela aplicação da lei de Kirchoff das tensões.

Figura 2.4 – Circuito elétrico do motor cc



Fonte: (SCHILING, 2017)

$$u(t) = R_a i_a(t) + k_e \dot{\theta}(t) + L_a \frac{di}{dt} \quad (2.14)$$

Onde R_a é a resistência de armadura, L_a é a indutância de armadura, k_e é a constante da força contra eletromotriz, $i_a(t)$ é a corrente de armadura, $\dot{\theta}(t)$ é a velocidade angular do motor conforme mostrado na figura 2.4

Aqui será feita uma consideração simplificadora, que consiste em tomar a constante de tempo elétrica do motor muito menor do que a constante de tempo mecânica, de modo que sua indutância possa ser desprezada, o que resulta na equação 2.15.

$$u(t) = R_a i_a(t) + k_e \dot{\theta}(t) \quad (2.15)$$

O torque do motor precisa superar a inércia da carga acoplada em seu eixo. Essa variável é proporcional à $i_a(t)$, tendo a constante de torque k_t e o Fator da caixa de redução n como fatores de proporcionalidade. O torque é então dado por:

$$T_m(t) = n k_t i_a(t) \quad (2.16)$$

Desta maneira, necessita-se agora encontrar os valores para as constantes de torque do motor k_t e da força contra eletromotriz k_e , sendo que estas podem ser estimadas através

das equações 2.17 e 2.18.

$$k_t = \frac{T_m}{I_a} \quad (2.17)$$

$$V_e = \omega k_e \quad (2.18)$$

Onde T_m e I_a são o torque nominal e a corrente de armadura nominal, respectivamente. Ainda V_e e ω são a tensão da força contra eletromotriz e velocidade angular. Estes valores serão obtidos posteriormente da folha de dados do motor CC utilizado neste trabalho, a qual fornece a velocidade em *rpm*, dessa forma a equação 2.18 deve ser readequada.

$$V_e = \frac{2\pi N}{60} k_e \quad (2.19)$$

Novamente aplicando a lei de Kirchoff no circuito, no intuito de encontrar a tensão de armadura, sendo a tensão nominal V_n menos a corrente de armadura nominal I_a multiplicada pela resistência de armadura R_a .

$$V_e = V_n - I_a R_a \quad (2.20)$$

resultando então:

$$k_e = \frac{(V_n - I_a R_a) 60}{2\pi N} \quad (2.21)$$

Desta maneira é possível determinar as constantes de torque k_t e força contra eletromotriz k_e utilizando os parâmetros do fabricante.

Os motores serão ativados por modulação PWM (Pulse Width Modulation), o que significa que são controlados por tensão, portanto a corrente necessita ser substituída da equação de torque. Isto pode ser feito utilizando a mesma lógica da equação 2.20, que nos dá a equação 2.22.

$$I_a = \frac{u - k_e \dot{\theta}}{R_a} \quad (2.22)$$

Assim sendo, a equação 2.23 descreve a equação de torque para cada motor, sendo que a mesma deve ser multiplicada por dois quando substituída nas demais equações posteriormente, tendo em vista que serão utilizados dois motores no eixo das rodas do robô.

$$T_m = \frac{nk_t u}{R_a} - \frac{nk_t k_e \dot{\theta}}{R_a} \quad (2.23)$$

2.5 Modelo dinâmico no espaço de estados

Tendo conseguido os modelos parciais das partes constituintes do sistema, procede-se então buscando acoplá-las para gerar uma única representação em espaço de estados. Partindo das equações de momento 2.1 e 2.10, no pêndulo e na roda, respectivamente, são encontradas as equações não lineares para as acelerações angulares $\ddot{\psi}$ e $\ddot{\theta}$. Iniciando então pela equação de momento no pêndulo.

É necessário encontrar a equação para a aceleração \ddot{x}_r na direção do eixo x , para que $\ddot{\psi}$ possa ser isolada e as demais variáveis conhecidas. Dessa maneira, para se determinar a equação de \ddot{x}_r as equações 2.2 e 2.11, correspondente as forças atuando no eixo x , foram utilizadas, onde o desenvolvimento pode ser acompanhado nas equações a seguir.

Isolando N_x da equação 2.2,

$$N_x = -m_p \ddot{x}_r - m_p \ddot{x}_p \quad (2.24)$$

e aplicando em 2.11, resulta:

$$-m_p \ddot{x}_r - m_p \ddot{x}_p - m_r \ddot{x}_r - F = 0 \quad (2.25)$$

Agora, isolando \ddot{x}_r :

$$\ddot{x}_r = \frac{m_p \ddot{x}_p + F}{(-m_p - m_r)} \quad (2.26)$$

Em \ddot{x}_r se faz necessário conhecer F . Para tal foi utilizada a equação 2.10, onde F pode ser isolado e substituído na equação 2.26

$$F = \frac{J_r \ddot{\theta} - T_m}{r} \quad (2.27)$$

$$\ddot{x}_r = \frac{m_p \ddot{x}_p + \frac{J_r \ddot{\theta} - T_m}{r}}{(-m_p - m_r)} \quad (2.28)$$

A aceleração angular $\ddot{\theta}$ também precisa ser eliminada da equação 2.28 para determinar \ddot{x}_r . Isso é feito através da equação 2.13.

$$\ddot{x}_r = \frac{m_p \ddot{x}_p r^2 + J_r \ddot{x}_r - T_m r}{r^2(-m_p - m_r)} \quad (2.29)$$

Assim foi possível chegar em uma equação para \ddot{x}_r .

$$\ddot{x}_r = \frac{m_p \ddot{x}_p r^2 - T_m r}{r^2(-m_p - m_r) - J_r} \quad (2.30)$$

Por fim \ddot{x}_r como também \ddot{x}_p podem ser substituídos na equação para $\ddot{\psi}$, que então pode ser trabalhada como:

$$T_m + m_p g L \sin \psi + \frac{m_p \ddot{x}_p r^2 - T_m r}{r^2(-m_p - m_r) - J_r} L \cos \psi = J_p \ddot{\psi} \quad (2.31)$$

e então

$$\begin{aligned} & T_m(r^2(-m_p - m_r) - J_r) + m_p g L \sin \psi (r^2(-m_p - m_r) \\ & - J_r) + m_p (m_p \ddot{x}_p r^2 - T_m r) L \cos \psi = J_p (r^2(-m_p - m_r) - J_r) \ddot{\psi} \end{aligned} \quad (2.32)$$

Agora, substituindo a equação para $\ddot{x}_p(t)$, resulta:

$$\begin{aligned} & T_m(r^2(-m_p - m_r) - J_r) + m_p g L \sin \psi (r^2(-m_p - m_r) - J_r) \\ & + (m_p (m_p (-L \ddot{\psi} \cos \psi + L \dot{\psi}^2 \sin \psi) r^2 - T_m r) L \cos \psi = J_p (r^2(-m_p - m_r) - J_r) \ddot{\psi} \end{aligned} \quad (2.33)$$

Isolando, no lado direito os termos com $\psi''(t)$

$$T_m(r^2(-m_p - m_r) - J_r) + m_p g L \sin \psi (r^2(-m_p - m_r) - J_r) \quad (2.34)$$

$$+ m_p^2 r^2 L^2 \dot{\psi}^2 \sin \psi - T_m m_p r L \cos \psi = J_p (r^2(-m_p - m_r) - J_r) \ddot{\psi} + (L^2 r^2 m^2 \cos^2 \psi) \ddot{\psi}$$

É possível escrever:

$$(T_m((r^2(-m_p - m_r) - J_r) - m_p r L \cos \psi) + m_p g L \sin \psi (r^2(-m_p - m_r) - J_r) \quad (2.35)$$

$$+ m_p^2 r^2 L^2 \dot{\psi}^2 \sin \psi \cos \psi) / (J_p (r^2(-m_p - m_r) - J_r) + L^2 r^2 m^2 \cos^2 \psi) = \ddot{\psi}$$

Ainda substituindo na equação de torque do motor T_m na equação 2.36 pode-se finalizar a equação para a aceleração angular do pêndulo $\ddot{\psi}$.

$$\ddot{\psi} = (J_r (g L R_a m_p \sin \psi + 2 n k_t (u - k_e \dot{\theta})) + r (L r R_a m_p \sin \psi (m_p (g - \quad (2.36)$$

$$L \dot{\psi}^2 \cos \psi) + g m_r) + 2 n k_t (u - k_e \dot{\theta}) (m_p (L \cos \psi + r + r m_r)))$$

$$/ (R_a (J_p (J_r + r^2 (m + p + m_r)) - L^2 r^2 m_p^2 \cos^2 \psi))$$

Com a finalidade de encontrar a segunda equação não linear que descreve a aceleração angular da roda, foi utilizada a equação 2.28 que parte da ideia de isolar \ddot{x}_r , igualando-a a equação 2.13, onde é possível isolar $\ddot{\theta}$ na equação 2.38.

$$r \ddot{\theta} = \frac{m_p \ddot{x}_p + \frac{J_r \ddot{\theta} - T_m}{r}}{(-m_p - m_r)} \quad (2.37)$$

$$\ddot{\theta} = \frac{m_p \ddot{x}_p r - T_m}{r^2 (-m_p - m_r) - J_r} \quad (2.38)$$

Então faz se necessário substituir \ddot{x}_p na equação 2.38

$$\ddot{\theta} = \frac{m_p r (-L \ddot{\psi} \cos \psi + L \dot{\psi}^2 \sin \psi) - T_m}{r^2 (-m_p - m_r) - J_r} \quad (2.39)$$

Tomando então a aceleração angular $\ddot{\psi}$ a partir da equação 2.1.

$$\ddot{\psi} = \frac{T_m + m_p g L \sin \psi + m_p \ddot{x}_r L \cos \psi}{J_p} \quad (2.40)$$

Aplicando em 2.39:

$$\ddot{\theta} = \frac{m_p r (-L (\frac{T_m + m_p g L \sin \psi + m_p \ddot{x}_r L \cos \psi}{J_p}) \cos \psi + L \dot{\psi}^2 \sin \psi)}{r^2 (-m_p - m_r) - J_r} \quad (2.41)$$

Expandindo:

$$\ddot{\theta} = \frac{m_p r (-L \cos \psi (T_m + m_p g L \sin \psi + m_p \ddot{x}_r L \cos \psi) + J_p L \dot{\psi}^2 \sin \psi) - J_p T_m}{J_p (r^2 (-m_p - m_r) - J_r)} \quad (2.42)$$

Onde se substituiu \ddot{x}_r da equação 2.13. Isolando agora $\ddot{\theta}$ isolado, a equação do torque do motor pode ser substituída para então finalizar a equação não linear da aceleração angular $\ddot{\theta}$.

$$\ddot{\theta} = \frac{m_p r (-L \cos \psi (T_m + m_p g L \sin \psi + m_p \ddot{\theta}_r L \cos \psi) + J_p L \dot{\psi}^2 p \sin \psi - J_p T_m)}{J_p (r^2 (m_p + m_r) + J_r)} \quad (2.43)$$

$$\ddot{\theta} = \frac{m_p r L \cos \psi T_m + m_p^2 r L^2 g \cos \psi \sin \psi + L^2 r^2 m_p^2 \ddot{\theta} L \cos^2 \psi - J_p m_p r L \dot{\psi}^2 \sin \psi + J_p T_m}{J_p (r^2 (m_p + m_r) + J_r)} \quad (2.44)$$

$$\ddot{\theta} = \frac{m_p r L \cos \psi T_m + m_p^2 r L^2 g \cos \psi \sin \psi - J_p m_p r L \dot{\psi}^2 \sin \psi + J_p T_m}{J_p (r^2 (m_p + m_r) + J_r) - L^2 r^2 m_p^2 L \cos^2 \psi} \quad (2.45)$$

Assim sendo a equação 2.45 pode ser rearranjada resultando na resposta para aceleração angular das rodas $\ddot{\theta}$.

$$\ddot{\theta} = \frac{L r m_p \cos \psi (g L R_a m_p \sin \psi + 2 n k_t (u - k_e \dot{\theta})) + J_p 2 n k_t (u - K_e \dot{\theta}) - J_p L r R_a m_p \dot{\psi}^2 \sin \psi}{(R_a (J_p (J_r + r^2 (m_p + m_r)) - L^2 r^2 m_p^2 \cos^2 \psi))} \quad (2.46)$$

2.5.1 Equações de estado não lineares

Tendo finalizado o modelo matemático das equações não lineares, resultou em duas equações de segunda ordem. Visando gerar um modelo em espaço de estados, serão feitas as seguintes escolhas para as variáveis de interesse do pêndulo:

- θ - Posição da roda;
- ψ - Posição angular do pêndulo;
- $\omega = \dot{\theta}$ - Velocidade da roda;
- $\phi = \dot{\psi}$ - Velocidade angular do pêndulo;

Desta forma, o vetor de estados fica definido como $\mathbf{x} = [\theta \ \psi \ \omega \ \phi]^T$. A estrutura em espaço de estados é então definida na forma geral

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\omega} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}, u) \\ f_2(\mathbf{x}, u) \\ f_3(\mathbf{x}, u) \\ f_4(\mathbf{x}, u) \end{bmatrix} \quad (2.47)$$

Onde:

$$f_1(\mathbf{x}, u) = \omega \quad (2.48)$$

$$f_2(\mathbf{x}, u) = \phi \quad (2.49)$$

$$f_3(\mathbf{x}, u) = L r m_p \cos \psi (g L R_a m_p \sin \psi + 2 n k_t (u - k_e \omega)) + J_p 2 n k_t (u - K_e \omega) - J_p L r R_a m_p \phi^2 \sin \psi / (R_a (J_p (J_r + r^2 (m_p + m_r)) - L^2 r^2 m_p^2 \cos^2 \psi)) \quad (2.50)$$

$$f_4(\mathbf{x}, u) = (J_r(gLR_a m_p \sin\psi + 2nk_t(u - k_e\omega)) + r(LrR_a m_p \sin\psi(m_p(g - L\phi^2 \cos\psi) + gm_r) + 2nk_t(u - k_e\omega)(m_p(L\cos\psi + r) + rm_r)))/(R_a(J_p(J_r + r^2(m + p + m_r)) - L^2r^2m_p^2\cos^2\psi)) \quad (2.51)$$

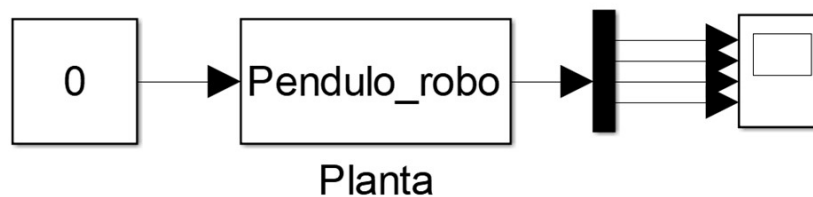
O que define o modelo em espaço de estados do sistema, que se apresenta como uma dinâmica de quarta ordem não-linear.

2.6 Sistema em Malha Aberta

De posse do modelo em espaço de estados é desejável verificar o comportamento dinâmico deste sistema. Para isso este modelo foi codificado e simulado em ambiente Matlab/Simulink de acordo com a Figura 2.5. Nesta simulação foram utilizados parâmetros medidos e identificados a partir do protótipo real do robô pêndulo invertido, desenvolvido no escopo deste trabalho. O sistema real, bem como a medição dos parâmetros e seus respectivos valores é apresentado detalhadamente no capítulo 4.

Como a simulação do sistema é em malha aberta, o sinal de entrada foi zerado, sendo simulado o comportamento dinâmico dado pela resposta à seguinte posição inicial: $\mathbf{x}_0 = [0 \ \psi_0 \ 0 \ 0]^T$. Ou seja, foi simulada uma situação em que o pêndulo é solto próximo ao ponto de equilíbrio do robô em 0° . Os resultados das simulações são apresentados na Figura 2.7.

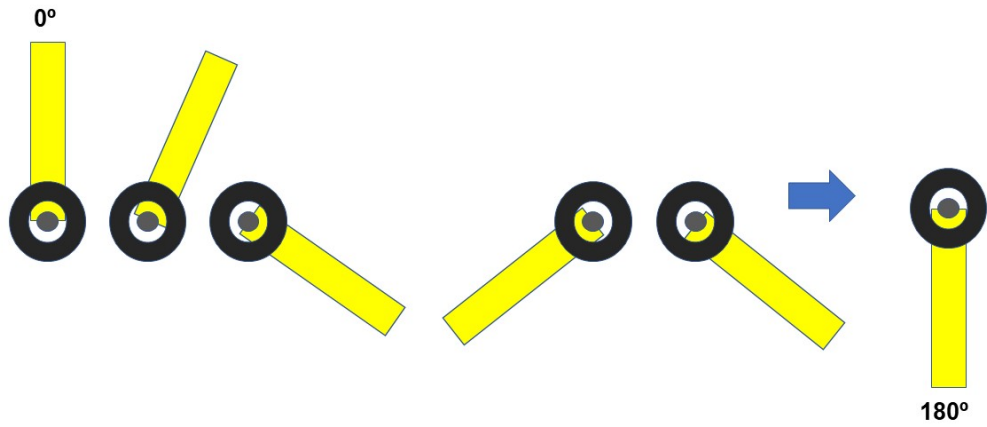
Figura 2.5 – Pêndulo Robô Malha Aberta



Fonte: (Do Autor, 2022.)

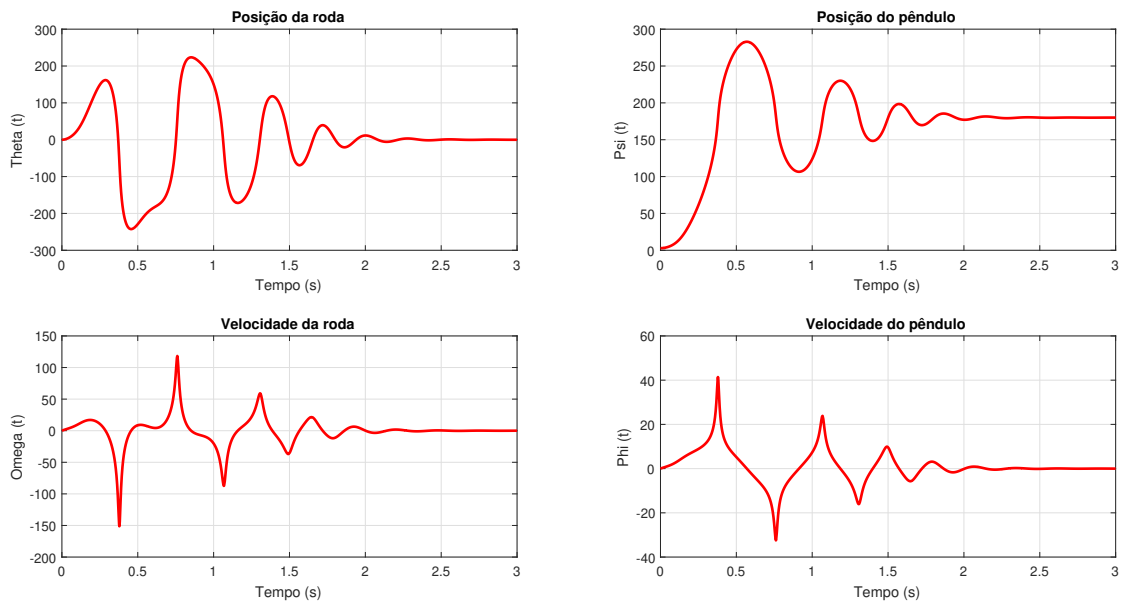
Como esperado, a posição angular do pêndulo robô tende a 180° , exibindo uma oscilação amortecida. A figura 2.6 exibe a queda do pêndulo robô.

Figura 2.6 – Simulação Pêndulo Robô Malha Aberta



Fonte: (Do Autor,2022.)

Figura 2.7 – Gráfico de análise de Variáveis



Fonte: (Do Autor, 2022.)

A figura 2.7 apresenta os gráficos das variáveis de estado θ , ψ , ω e ϕ , o qual demonstra que a única variável que não obteve seu amortecimento em 0° foi a posição do pêndulo, que passa a se acomodar em 180° , o que corresponderia a uma *queda* do pêndulo. Este comportamento, caracteriza qualitativamente, que o sistema é instável em malha aberta, o que constitui um novo desafio do pêndulo robô. Buscando atingir o equilíbrio

na posição vertical ou ainda a estabilidade dinâmica do sistema em torno de \boldsymbol{x}_0 se faz necessária a utilização de alguma técnica de controle dinâmico.

3 Controle no espaço de estados

A representação em espaço de estados permite que se projete um controlador com base na realimentação de forma que os estados do sistema sejam regulados em algum valor de referência constante, no caso um ponto de equilíbrio, e também com essa estratégia de controle deseja-se que o sistema tenha polos que forneça o desempenho transitório requisitado.

No caso de um sistema não linear, como o robô, o modelo necessita ser linearizado, tendo em vista que o projeto de controladores se baseia em modelos lineares. O objetivo do controle a ser realizado é equilibrar o corpo do robô na vertical, de forma que o sistema pode ser linearizado dentro de uma pequena variação de movimento em relação ao eixo vertical, onde o ângulo do pêndulo pode ser admitido como $\psi = 0$, a velocidade angular do pêndulo $\dot{\psi} = 0$ e a velocidade angular da roda $\dot{\phi} = 0$.

Os métodos de controle clássicos são amplamente utilizados na indústria pelo baixo custo, eficiência, ampla aplicabilidade e facilidade de projeto. Para o presente trabalho, deseja-se utilizar tais métodos para projetar um controlador capaz de posicionar o pêndulo invertido na posição desejada.

3.1 Linearização do modelo

De acordo com (AMBA, 2015), a técnica de linearização utiliza a expansão em Séries de Taylor a fim de encontrar um modelo linear que aproxime do sistema não linear sobre um determinado ponto de equilíbrio, de forma que seja possível analisar as características desse ponto, como estabilidade, polos, etc.

Como visto, o modelo do sistema robô pêndulo invertido sobre rodas resultou em uma dinâmica não linear de quarta ordem. Além disso o comportamento dinâmico qualitativo é de um sistema instável, se considerar a necessidade de regular a posição do pêndulo na vertical, com $\psi = 0$. Neste caso existe a necessidade de se projetar um controlador que altere a dinâmica natural do sistema visando garantir estabilidade sobre este ponto de equilíbrio.

As estratégias de controle convencionais envolvem tipicamente a utilização de um modelo linear cujo comportamento se aproxime da dinâmica do sistema nas proximidades do ponto de equilíbrio \mathbf{x}_0 desejado, de modo que os sinais também operem com pequenas amplitudes.

A obtenção deste modelo linear parte do modelo não linear do sistema, que pode ser expresso genericamente como:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \quad (3.1)$$

No caso do modelo do pêndulo robô 3.1 representa o conjunto das quatro equações de estado do sistema. Como $\mathbf{f}(\mathbf{x}, u)$ é uma estrutura vetorial de funções não lineares, então, se a condição de operação normal corresponde a $\vec{\mathbf{x}}$ e $\vec{\mathbf{u}}$, pode-se expandi-la por uma série de Taylor multivariável, em torno do ponto de operação desejado.

$$\dot{x} = f(x_0, u_0) + \frac{\partial f(x, u)}{\partial x}(x - x_0) + \frac{\partial f(x, u)}{\partial u}(u - u_0) + \frac{1}{2!} \frac{\partial^2 f(x, u)}{\partial x^2}(x - x_0)^2 + \frac{1}{2!} \frac{\partial^2 f(x, u)}{\partial u^2}(u - u_0)^2 + \dots \quad (3.2)$$

Sobre esta estrutura pode-se considerar que, quando o sistema está operando no equilíbrio o estado não varia, ou seja $f(x_0, u_0) = 0$. Além disso, uma vez que se busca uma aproximação linear, os termos com ordem maior ou igual a 2 (não-lineares) serão desprezados nesta aproximação polinomial. De modo que se pode tomar uma aproximação linear dada por:

$$\dot{x} = \frac{\partial f(x, u)}{\partial x}(x - x_0) + \frac{\partial f(x, u)}{\partial u}(u - u_0) \quad (3.3)$$

Considerando ainda, no caso particular do sistema robô pêndulo que $x_0 = 0$ e $u_0 = 0$, 3.3 recai na estrutura convencional de um sistema linear e invariante no tempo no espaço de estados, ou seja:

$$\dot{x} = Ax + Bu \quad (3.4)$$

Sendo A e B as matrizes jacobianas do sistema, com coeficientes constantes

$$A = \frac{\partial f(x, u)}{\partial x} \quad B = \frac{\partial f(x, u)}{\partial u} \quad (3.5)$$

que expandidas a partir das funções de estado, variáveis de estados e entrada individuais podem ser determinadas como:

$$A = \begin{bmatrix} \frac{\partial f_1(x, u)}{\partial x_1} & \frac{\partial f_1(x, u)}{\partial x_2} & \frac{\partial f_1(x, u)}{\partial x_3} & \frac{\partial f_1(x, u)}{\partial x_4} \\ \frac{\partial f_2(x, u)}{\partial x_1} & \frac{\partial f_2(x, u)}{\partial x_2} & \frac{\partial f_2(x, u)}{\partial x_3} & \frac{\partial f_2(x, u)}{\partial x_4} \\ \frac{\partial f_3(x, u)}{\partial x_1} & \frac{\partial f_3(x, u)}{\partial x_2} & \frac{\partial f_3(x, u)}{\partial x_3} & \frac{\partial f_3(x, u)}{\partial x_4} \\ \frac{\partial f_4(x, u)}{\partial x_1} & \frac{\partial f_4(x, u)}{\partial x_2} & \frac{\partial f_4(x, u)}{\partial x_3} & \frac{\partial f_4(x, u)}{\partial x_4} \end{bmatrix} \quad (3.6)$$

e

$$B = \begin{bmatrix} \frac{\partial f_1(x, u)}{\partial u} \\ \frac{\partial f_2(x, u)}{\partial u} \\ \frac{\partial f_3(x, u)}{\partial u} \\ \frac{\partial f_4(x, u)}{\partial u} \end{bmatrix}. \quad (3.7)$$

Para a linearização do sistema pêndulo-robô o modelo não linear foi declarado em linguagem Matlab utilizando o pacote simbolic, de forma que as operações de linearização foram realizadas sobre a estrutura algébrica do modelo, de maneira analítica. O código deste processo de linearização é apresentado no Anexo A.1.

O código com a seguinte sequência: (i) Declaração das variáveis e parâmetros simbólicos, (ii) declaração das equações não-lineares, (iii) determinação das matrizes A e B simbólicas utilizando o operador jacobiano, (iv) substituição dos parâmetros e do ponto de

equilíbrio nas matrizes simbólicas visando obter as matrizes A e B numéricas, que foram determinadas como:

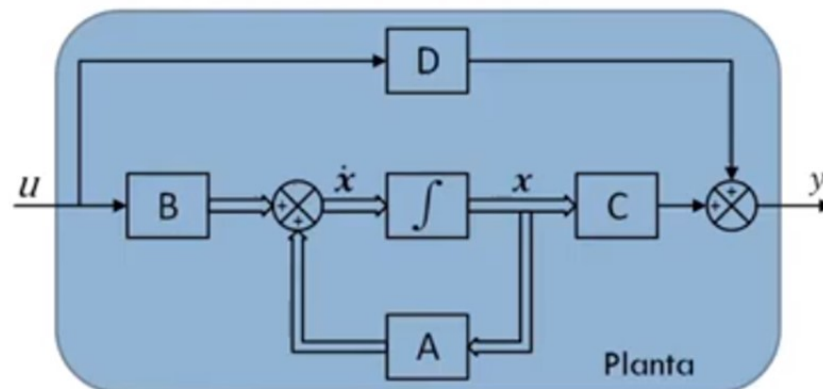
$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2056.83 & -18.01 & 0 \\ 0 & 555.26 & -4.34 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 46.66 \\ 11.23 \end{bmatrix}. \quad (3.8)$$

Estas matrizes parametrizam o modelo linear do sistema no ponto de operação desejado. Sistema representado em espaço de estado:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.9)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}u \quad (3.10)$$

Figura 3.1 – Malha Aberta



Fonte: (CARDOSO, 2020a)

A figura 3.1 é uma representação gráfica por diagrama de blocos em espaço de estado, no qual o sistema é em malha aberta.

3.1.1 Análise de estabilidade

Uma vez definido o modelo linear pode-se proceder com a análise de estabilidade. Isto é realizado determinando os autovalores da matriz \mathbf{A} que correspondem aos polos do sistema. No MATLAB é possível determinar os autovalores da matriz \mathbf{A} com a função $eig(\mathbf{A})$, o que resulta:

$$\text{eig}(\mathbf{A}) = [0 \quad -33,5866 \quad 17,4211 \quad -1,8471] \quad (3.11)$$

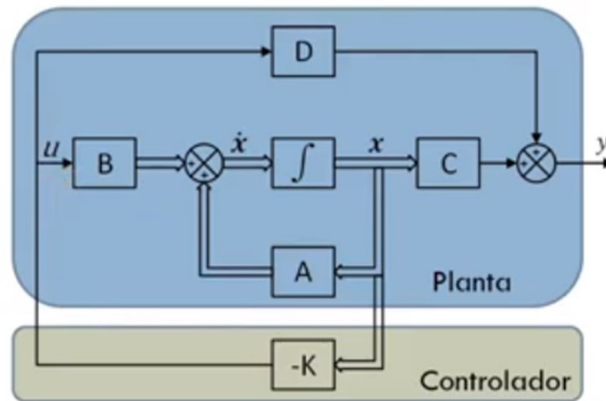
O sistema exibe 4 polos reais, sendo um polo na parte real positiva o que o caracteriza como um sistema instável. Comprova-se portanto que sem a presença do controlador o robô não tem como se equilibrar na posição desejada.

3.2 Alocação de Polos

Segundo (OGATA et al., 2010), deve-se admitir, que todas as variáveis de estado são mensuráveis e estão disponíveis para realimentação. Caso o sistema seja completamente controlável então os polos do mesmo em malha fechada podem ser alocados em qualquer lugar desejado.

O método de alocação de polos é um método de projeto de controlador no qual é possível determinar as posições dos polos de malha fechada no plano complexo através de um ganho de controlador \mathbf{K} . Os polos descrevem o comportamento de sistemas de dinâmicas lineares. Através do uso da realimentação é feita uma tentativa de mudar esse comportamento de uma forma que seja mais favorável ao objetivo esperado. Assim pode-se decidir onde os polos de malha fechada devem estar e então mantê-los ali através do projeto de um sistema de controle por realimentação com o método de alocação de polos onde o ganho \mathbf{K} do controlador coloca os polos em seus locais desejados.

Figura 3.2 – Malha Fechada



Fonte: (CARDOSO, 2020a)

A figura 3.2 é uma representação gráfica por diagrama de blocos em espaço de estado, a qual apresenta o bloco de ganhos com realimentação dos estado, caracterizando o sistema em malha fechada.

Partindo então para a formulação do projeto do controlador, define-se a lei de controle como sendo:

$$u = -[k_1 \ k_2 \ \dots \ k_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = -\mathbf{K}\mathbf{x} \quad (3.12)$$

Substituindo 3.12 em 3.4, resulta:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} \quad (3.13)$$

Isolando \mathbf{x} à direita, resulta no modelo dinâmico em malha fechada:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \quad (3.14)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (3.15)$$

A solução de 3.14 é:

$$\dot{\mathbf{x}}(t) = e^{(\mathbf{A}-\mathbf{B}\mathbf{K})t} \mathbf{x}(0) \quad (3.16)$$

Os polos do sistema em malha fechada serão portanto os autovalores de $(\mathbf{A} - \mathbf{B}\mathbf{K})$. Assim, se o sistema for controlável, a estratégia de controle por realocação de polos mostra (OGATA et al., 2010) que pode-se determinar o valor \mathbf{K} tal que os autovalores de $(\mathbf{A} - \mathbf{B}\mathbf{K})$ estejam em qualquer posição do plano complexo. Desta forma é possível especificar polos de malha fechada estáveis e que atendam a especificações estritas de resposta transitória.

3.2.1 Controlabilidade

O sistema é dito controlável no instante em t_0 , se for possível, por meio de um vetor de controle $u(t)$ não restrito, transferir de qualquer estado inicial $x(t_0)$ para qualquer outro estado num intervalo de tempo finito.

A verificação da controlabilidade é dada a partir do posto ou do determinante da matriz de controlabilidade, a qual é definida por:

$$\mathcal{C} = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \quad (3.17)$$

Onde n é a ordem do sistema.

Para o sistema robô pêndulo a matriz de controlabilidade é determinada no Matlab pelo comando $ctrb(A,B)$, o que resulta:

$$\mathcal{C} = \begin{bmatrix} 0 & 46,66 & -840,41 & 38245,30 \\ 0 & 11,23 & -202,36 & 9883,04 \\ 46,66 & -840,41 & 38245,30 & -1105119,20 \\ 11,23 & -202,36 & 9883,04 & -278240,22 \end{bmatrix} \quad (3.18)$$

A qual possui posto $\mathcal{C} = 4$ e determinante $\det(\mathcal{C}) \neq 0$, de forma que o sistema é controlável.

3.2.2 Projeto do Controlador - Fórmula de Ackermann

Uma maneira de calcular o vetor de ganhos \mathbf{K} é por meio da fórmula de Ackermann, a qual é definida por (OGATA et al., 2010):

$$\mathbf{K} = [0 \ 0 \ \dots \ 1]C^{-1}\phi(A) \quad (3.19)$$

Onde $\phi(s)$ é o polinômio característico cujas raízes são os polos desejados de malha fechada (s_1, s_2, \dots, s_n) , ou seja:

$$\phi(s) = (s - s_1)(s - s_2)(s - s_n) \quad (3.20)$$

Ackermann expressa explicitamente que no caso de sistema não controlável a inversa da matriz de controlabilidade não vai existir e com isso não será possível cálculo do vetor de ganho \mathbf{K} .

3.3 Projeto de Controlador por Alocação de polos

Considerando agora o projeto do controlador por alocação de polos para o sistema pêndulo-robô, tomando o modelo linear do sistema, inicia-se pela especificação dos requisitos de projeto.

O primeiro e mais relevante dos requisitos é a garantia da estabilidade do sistema, o que será atingido simplesmente por especificar os polos desejados de malha fechada com parte real negativa.

Um segundo requisito diz respeito ao tempo de acomodação t_s do sistema em uma resposta de regulação. Ou seja, o tempo entre o sistema transitar de um ponto inicial nas vizinhanças do ponto de equilíbrio até o sistema se regular em torno deste.

Como o sistema possui 4 polos a serem realocados, buscou-se especificar um par de polos complexos conjugados como sendo os polos dominantes, para os quais foi projetado $t_s = 1,5s$. A estes polos foi atribuído ainda um amortecimento de $\zeta = 0,8$. Utilizando a relação conhecida para o tempo de acomodação de um sistema de segunda ordem, dada por:

$$t_s = \frac{4}{\zeta\omega_n} \quad (3.21)$$

onde ω_n é a frequência natural do polo, que pode então ser calculada como:

$$\omega_n = \frac{4}{\zeta t_s} = 3,33 \text{ rad/s} \quad (3.22)$$

Assim, o par de polos dominantes $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$ foi determinado como: $s_{1,2} = -2,67 \pm j2$. Os demais polos foram atribuídos como polos reais a esquerda dos polos dominantes, de modo que a sua dinâmica transitória fosse irrelevante comparada ao par complexo conjugado, os valores escolhidos livremente no contexto deste projeto foram: $s_3 = -60$ e $s_4 = -85$. De modo que o vetor de polos desejados foi especificado como:

$$s = [s_1, s_2, s_3, s_4] \quad (3.23)$$

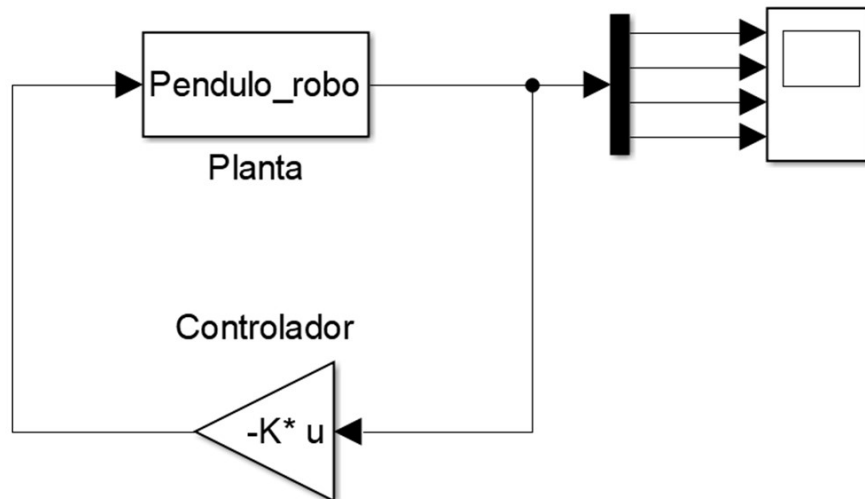
Com os polos definidos procede-se então com a determinação dos ganhos do controlador, diretamente pela fórmula de Ackermann, que é codificada em matlab como: $\mathbf{K} = \text{acker}(A,B,s)$. O que resultou no seguinte projeto para o controlador:

$$\mathbf{K} = [-20,2423 \quad 657,2796 \quad -10,6779 \quad 56,1239] \quad (3.24)$$

3.4 Simulação e Análise do Pêndulo Robô Malha Fechada

Com a utilização do software Matlab/Simulink para a simulação do sistema em malha fechada que é mostrado na Figura 3.3 , o qual apresenta o bloco da planta que está sendo realimentada pelo bloco do controlador e na saída possui o demultiplexador com o scope para gerar gráficos de análise das variáveis.

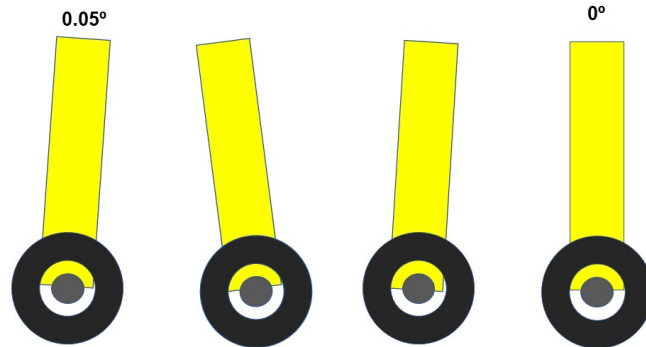
Figura 3.3 – Pêndulo Robô Malha Fechada



Fonte: (Do autor, 2022.)

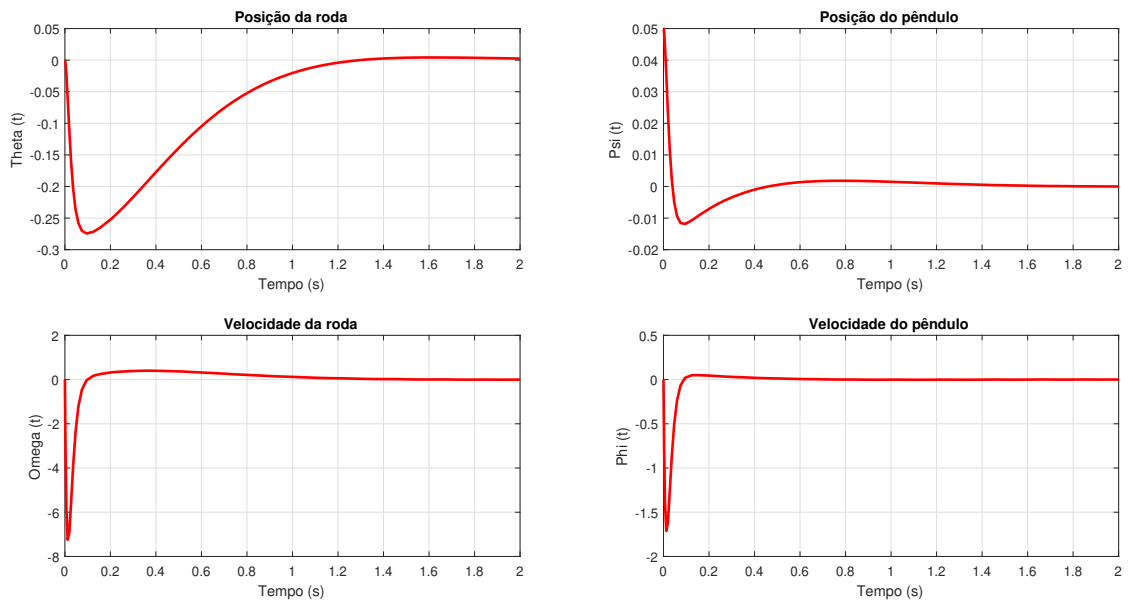
As representações do sistema do pêndulo robô foi simulada para uma resposta, em que o pêndulo é solto próximo ao ponto de inércia de equilíbrio do robô em $0,05^\circ$. Os resultados das simulações são apresentados na Figura 3.4 Como esperado, o corpo do pêndulo robô tende a posição de controle de equilíbrio na vertical em 0° , conforme os requisitos de projeto para gerar os ganhos de realimentação do sistema.

Figura 3.4 – Simulação Pêndulo Robô Malha Fechada



Fonte: (Do Autor, 2022.)

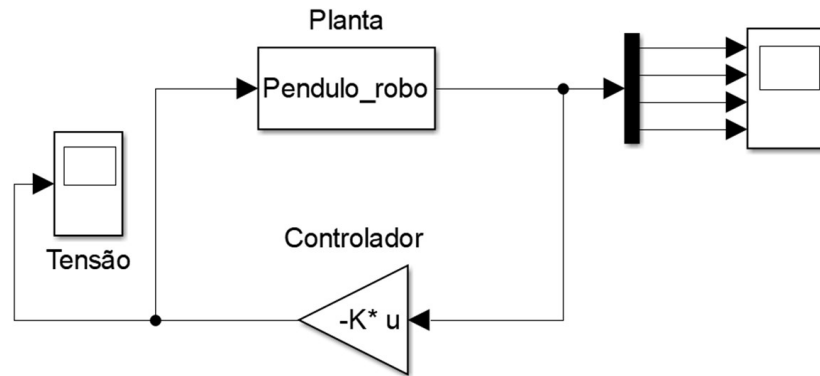
Figura 3.5 – Gráfico de análise de Variáveis Malha Fechada



Fonte: (Do Autor, 2022.)

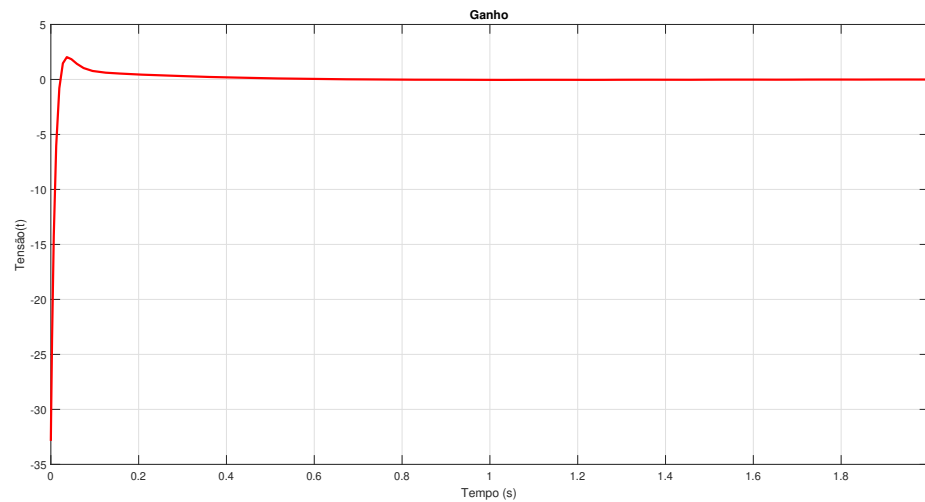
Os gráficos da Figura 3.5 convergem para o ponto de equilíbrio do pêndulo robô que é a vertical em 0° , e também é notório que o tempo de acomodação de $t_s = 1,5s$ foi atendido conforme requisitos de projeto. Entretanto, na análise do pêndulo robô em malha fechada é possível observar, a tensão de trabalho requerida pelo sistema vista através da implementação do scope na saída do controlador Figura 3.6 que é de aproximadamente $32,8V$ conforme mostrado na Figura 3.7.

Figura 3.6 – Malha Fechada Sem Saturação



Fonte: (Do Autor, 2022.)

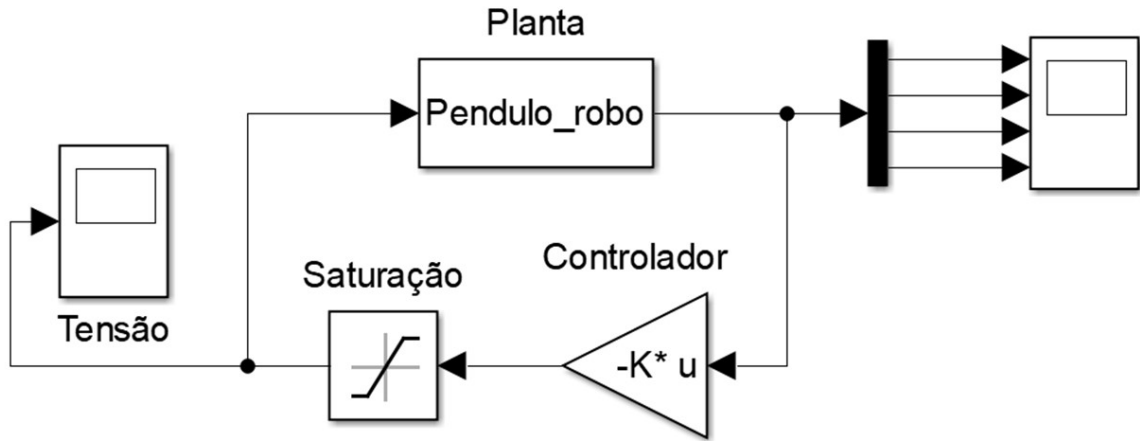
Figura 3.7 – Gráfico de Análise do Sinal de Saída do Controlador



Fonte: (Do Autor, 2022.)

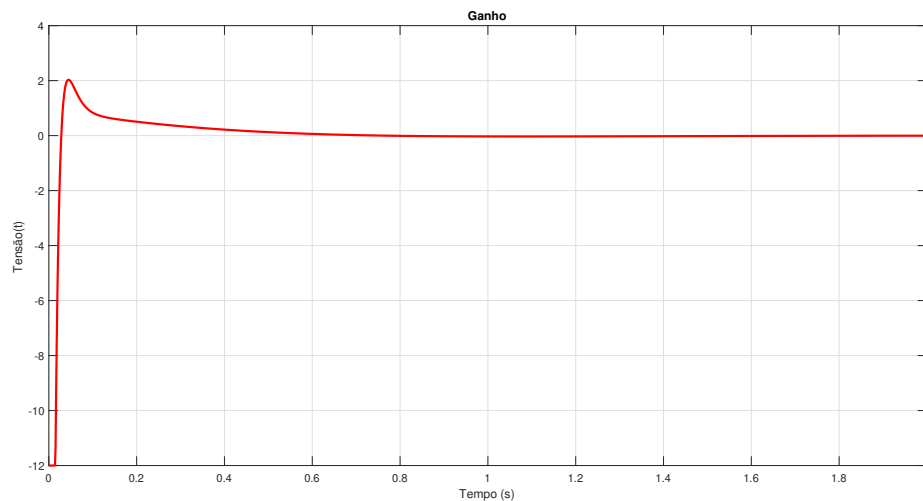
Diante disso, utilizou-se o bloco de saturação no sistema para a parametrização da tensão de alimentação em $+12V$ e $-12V$ devido a projeção do sistema físico da planta com alimentação de uma bateria de $12V$ conforme mostrado na Figura 3.8.

Figura 3.8 – Malha Fechada Com Saturação



Fonte: (Do Autor, 2022.)

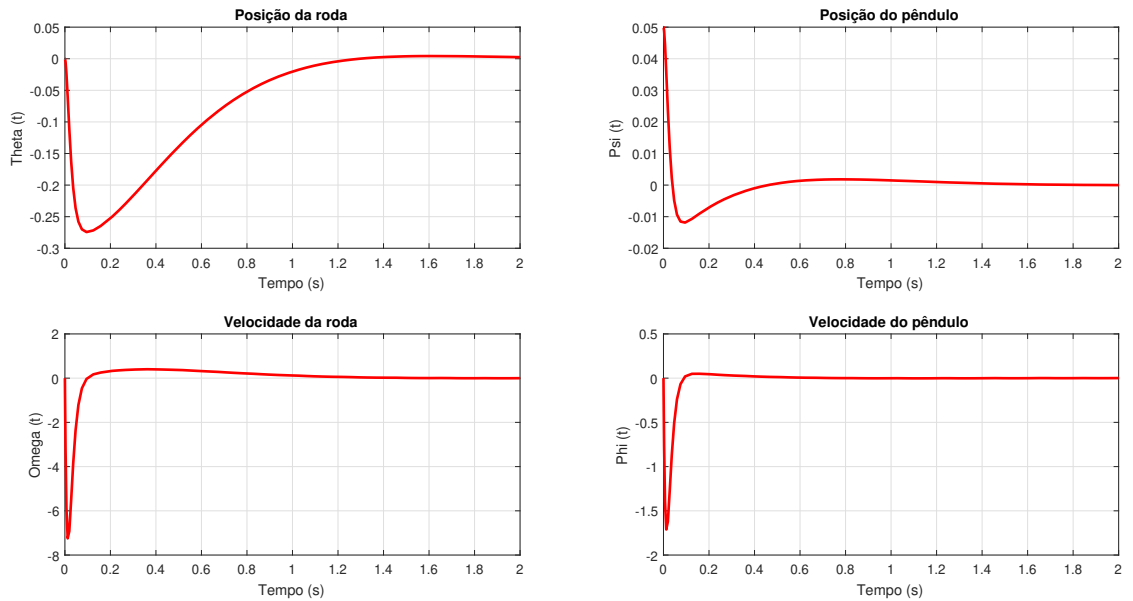
Figura 3.9 – Gráfico de Análise do Sinal da Saída do Controlador Com Saturação



Fonte: (Do Autor, 2022.)

É perceptível observar a tensão na Figura 3.9, em que, mesmo com o acréscimo do bloco de saturação na saída do controlador, o sistema do robô, continua convergindo para 0, porém é esperado que o mesmo tenha sua velocidade reduzida.

Figura 3.10 – Gráfico de Análise das Variáveis com a Saturação em Malha Fechada



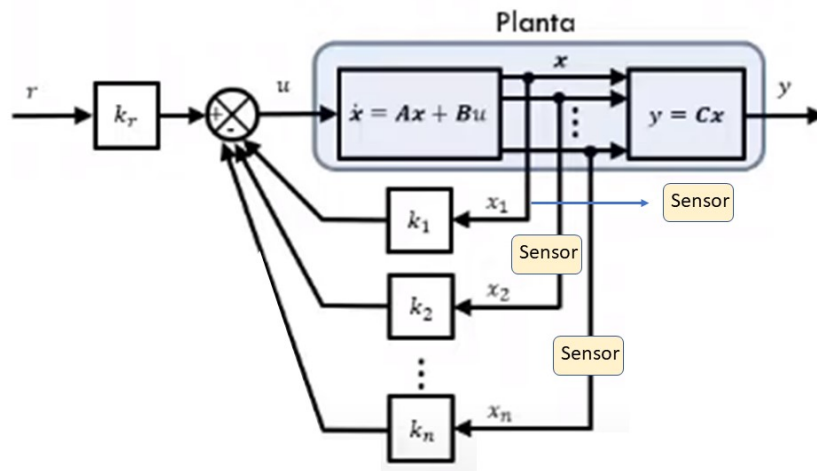
Fonte: (Do Autor, 2022.)

Os gráficos da figura 3.10 mostram a convergência para o ponto de equilíbrio do pêndulo robô, após a implementação do bloco de saturação na saída do controlador.

3.5 Observadores de Estado

No problema de alocação de polos é necessário o acesso (medições) de todos os estados da planta para realimentação. Segundo (OGATA et al., 2010), para a realização das medições é possível utilizar a estrutura de controle da medição dos estados por sensores, entretanto, se este não for o caso, é preciso estimar as variáveis de estado não disponíveis. Denomina-se esta estimativa de observação. Os observadores de estado estimam as variáveis de estado com base nas medidas das variáveis de saída e das variáveis de controle.

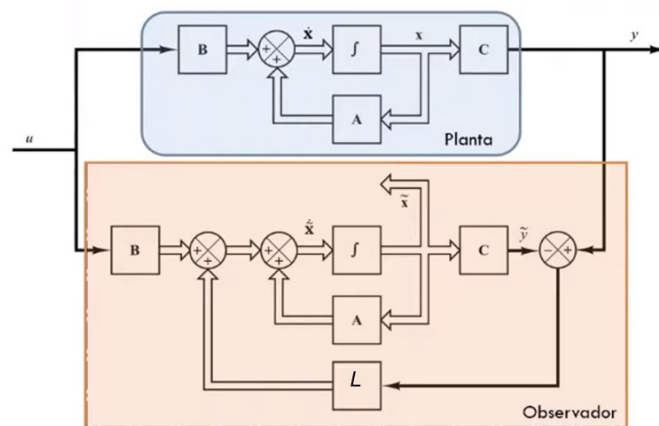
Figura 3.11 – Medição dos Estados do Sistema por Sensores



Fonte: (CARDOSO, 2020b)

Para o tipo de medição dos estados através de sensores Figura 3.11, por muitas vezes, o custo, ou mesmo a disponibilidade de um sensor necessário para a instrumentação torna a abordagem inviável. Entretanto, uma possível solução para contornar a necessidade de medição dos estados para implementação de controladores é o uso dos observadores de estado.

Figura 3.12 – Observadores de Estado



Fonte: (CARDOSO, 2020b)

O observador de estado é um dispositivo (ou um programa de computador) que

estima (observa) as variáveis de estado com base nas medidas da saída e da ação de controle figura 3.12. Assim como o projeto do controlador, o do observador baseia-se no modelo linear da planta, dado por:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.25)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}u \quad (3.26)$$

A dinâmica do observador, que rege o vetor de estados observado $\tilde{\mathbf{x}}$, segue a do sistema, sendo corrigida sempre que houver um erro entre a saída do sistema y e a saída do próprio observador \tilde{y} , o qual é ponderado pelo ganho do observador \mathbf{L} . Esta estrutura é dada por:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}u + \mathbf{L}(y - \tilde{y})$$

Substituindo \tilde{y} :

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}u + \mathbf{L}(y - \mathbf{C}\tilde{\mathbf{x}}). \quad (3.27)$$

Rearranjando o modelo:

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{x}} + \mathbf{B}u + \mathbf{L}y \quad (3.28)$$

Subtraindo 3.27 de 3.25, considerando 3.26

$$\dot{\mathbf{x}} - \dot{\tilde{\mathbf{x}}} = \mathbf{A}\mathbf{x} - \mathbf{A}\tilde{\mathbf{x}} - \mathbf{L}(\mathbf{C}\mathbf{x} - \mathbf{C}\tilde{\mathbf{x}}) \quad (3.29)$$

Agrupando os termos:

$$\dot{\mathbf{x}} - \dot{\tilde{\mathbf{x}}} = (\mathbf{A} - \mathbf{L}\mathbf{C})(\mathbf{x} - \tilde{\mathbf{x}}) \quad (3.30)$$

$$\dot{\mathbf{x}} - \dot{\tilde{\mathbf{x}}} = \mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}) - \mathbf{L}\mathbf{C}(\mathbf{x} - \tilde{\mathbf{x}}) \quad (3.31)$$

Essa equação mostra a dinâmica do erro de observação, definido por:

$$\mathbf{e}_x = \mathbf{x} - \tilde{\mathbf{x}}$$

A equação 3.30 pode ser reescrita como:

$$\dot{\mathbf{e}}_x = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}_x \quad (3.32)$$

A dinâmica do vetor de erros é determinada pelos autovalores da matriz $(\mathbf{A} - \mathbf{L}\mathbf{C})$. Se $(\mathbf{A} - \mathbf{L}\mathbf{C})$ é estável, o vetor de erros convergirá para zero independente do valor de $\mathbf{e}(0)$.

Se o sistema for observável, pode-se determinar \mathbf{L} de forma a se escolher arbitrariamente os autovalores de $(\mathbf{A} - \mathbf{L}\mathbf{C})$, e com isso, a velocidade de convergência do observador.

Um sistema é dito observável no instante t_0 , se com o sistema em uma condição inicial $\mathbf{x}(t_0)$, pode-se determinar $\mathbf{x}(t)$ a partir das medidas de saída $y(t)$ e da entrada $u(t)$ durante um intervalo de tempo finito a partir de t_0 .

A verificação da observabilidade é dada a partir do posto ou do determinante da matriz de observabilidade, a qual é definida por:

$$\mathbf{O}_M = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \quad (3.33)$$

Onde n é a ordem do sistema.

Para o sistema robô pêndulo a matriz de observabilidade é determinada no Matlab pelo comando $obsv(A,C)$, o que resulta:

$$O_M = \begin{bmatrix} 0 & 1,00 & 0 & 0 \\ 0 & 0 & 0 & 1,00 \\ 0 & 555,26 & -4,34 & 0 \\ 0 & -8920,89 & 78,12 & 555,26 \end{bmatrix} \quad (3.34)$$

A qual possui posto $O_M = 3$ e determinante $det(O_M) = 0$, de forma que o sistema é não observável.

Este resultado impõe um novo desafio para o projeto de controladores por realocação de polos, para o sistema pêndulo invertido, uma vez que não é possível reconstruir o vetor de estados a partir de medições de alguma variável particular.

3.6 Estratégia com modelo de ordem reduzida

Neste caso a estratégia de projeto seguida neste trabalho partirá da premissa de que a posição angular das rodas θ , não interfere nos objetivos da estratégia de controle, uma vez que o robô pode estar em qualquer lugar, e ainda assim manter-se na vertical. Assim, optou-se, como estratégia de projeto, suprimir esta variável do modelo dinâmico do sistema, visando o projeto do controlador. As novas variáveis de estado são:

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

O modelo utilizado passou a ser então um de terceira ordem, para o qual:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 2056.83 & -18.01 & 0 \\ 555.26 & -4.34 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 46.66 \\ 11.23 \end{bmatrix} \quad \mathbf{C} = [1 \ 0 \ 0] \quad \mathbf{D} = [0] \quad (3.35)$$

Novamente a matriz de observabilidade é determinada no matlab por $obsv(A,C)$, o que resulta:

$$O_M = \begin{bmatrix} 1,00 & 0 & 0 \\ 0 & 0 & 1,00 \\ 555,26 & -4,34 & 0 \end{bmatrix} \quad (3.36)$$

A qual possui posto $O_M = 3$ e determinante $det(O_M) \neq 0$, de forma que o sistema é observável.

Este novo modelo também resultou controlável, sendo realizado um novo projeto de controlador, seguindo os passos do primeiro. O que obteve como resultado:

$$\mathbf{K} = [78.8977 \ -0.6242 \ 6.8045] \quad (3.37)$$

3.6.1 Fórmula de Ackermann para o Observador

Uma maneira de calcular o vetor de ganhos \mathbf{L} é a fórmula de Ackermann.

$$\mathbf{L} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \mathbf{O}_M^{-1} \phi(A) \quad (3.38)$$

Onde $\phi(s)$ é o polinômio característico formado a partir dos autovalores desejados para o observador, isto é:

$$\phi(s) = (s - s_1)(s - s_2)(s - s_n)$$

3.7 Requisitos de Projeto para Polos Desejados do Observador

O sistema é observável, então, a relevância dos requisitos é determinar \mathbf{L} de maneira a se escolher de forma arbitrária os autovalores de $(\mathbf{A} - \mathbf{L}\mathbf{C})$, e com isso, obter a velocidade de convergência do observador.

Partindo então para o projeto do observador, utilizou-se a abordagem convencional de se tomar polos mais rápidos que os polos do controlador da planta, o que é necessário para que as variáveis observadas convirjam para as variáveis reais do sistema, e ainda para que estas sejam antecipados apropriadamente para a atuação do controlador. Os polos do observador serão quatro vezes mais rápidos que os polos do controlador. Isso é garantido simplesmente multiplicando a parte real dos polos do projeto do controlador por um fator de 4.

Assim, o par de polos dominantes $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$ foi determinado como: $s_{1,2} = -10.6667 \pm j2$. O outro polo foi atribuído como polo real a esquerda dos polos dominantes, o valor escolhido livremente no contexto deste projeto foi: $s_3 = -240$. De modo que o vetor de polos desejados foi especificado como:

$$s = [s_1, s_2, s_3] \quad (3.39)$$

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

$$\mathbf{L} = [243.32 \quad -909.83 \quad 1410.20] \quad (3.40)$$

3.8 Simulação e Análise do Pêndulo Robô Observador

Tendo concluído o projeto do controlador e do observador, procede-se então com a simulação e avaliação do projeto no sistema pêndulo robô.

Para tanto, foi implementado em Matlab/Simulink o código para a simulação do sistema completo. Note que o bloco do sistema implementa o seu modelo não linear completo, com as 4 variáveis de estados. Seguindo a estratégia de controle adotada, apenas 3 destas variáveis foram observadas, de modo que o controlador ignorou a dinâmica da posição da roda. A Figura 3.13 apresenta o diagrama simulink completo da estratégia.

controlador, \mathbf{K} , o projeto do observador tem a finalidade de calcular o vetor de constante, \mathbf{L} , de maneira que a resposta transitória do observador seja mais rápida que a resposta da velocidade malha controlada, na Figura 3.14 demonstra a velocidade do pêndulo $\dot{\psi}$ do observador(x4o) apresenta uma proporção de 4 vezes mais rápido em relação ao controlador(x4) a fim de resultar em uma estimação atualizada rapidamente do vetor de estado. além disso, é notório verificar que a velocidade da roda $\dot{\theta}$ e a posição do pêndulo ψ foram estimadas conforme os requisitos de projeto. Entretanto, devido a uma estratégia de projeto de ordem reduzida verificada na seção 3.6, a posição da roda θ foi suprimida do observador, porém na análise do gráfico, é evidente que houve a convergência para o sua posição inicial. Diante disso, o robô pêndulo invertido sobre rodas convergiu para a posição na vertical em 0° conforme a modelagem e controle do projeto.

4 Descrição do Sistema Pêndulo Invertido Sobre Rodas

Esta seção fornece uma descrição de todos os componentes, tanto mecânicos e eletrônicos utilizados para realizar o projeto de controle do robô.

4.1 Sistema Mecânico

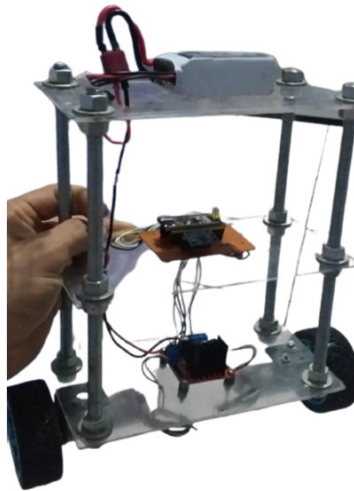
A Figura 4.1 apresenta o design geral do robô. A estrutura é composta por três bases de acrílico separadas cerca de 10 cm entre si, tipo estante, unidos por quatro hastes parafusos roscados localizados nos quatro cantos de cada uma das bases.

Na base de acrílico, localizada acima dos dois motores e entre as duas rodas está posicionado o estágio de potência, formado pelo driver *l298n*.

Na base intermediária, 10cm mais alta, está o estágio de controle. Ele contém a placa Arduino e o giroscópio e acelerômetro MPU6050.

Por fim, na base superior encontra-se o circuito de alimentação com uma bateria tipo LiPo de 12V.

Figura 4.1 – Robô Pêndulo Invertido Sobre Rodas

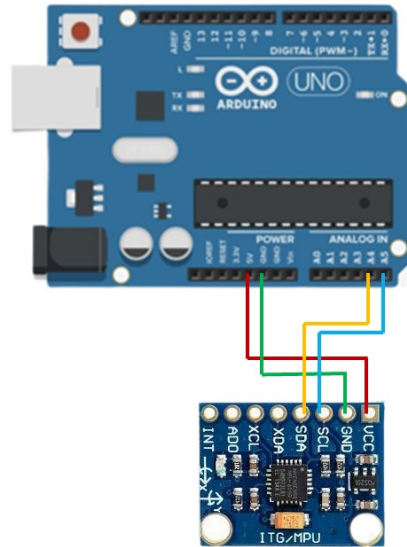


Fonte: (Do autor 2022.)

4.2 Sistema Elétrico

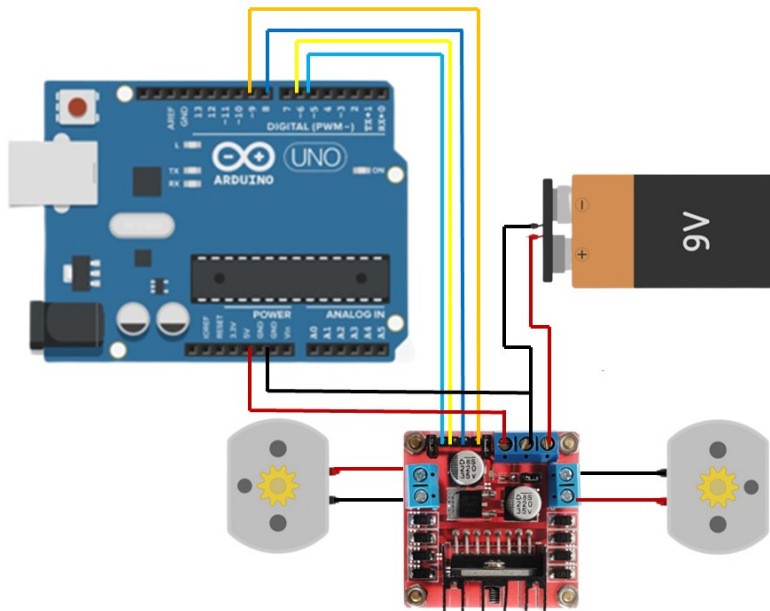
Nesta seção, é feita uma descrição dos componentes eletrônicos que compõem o circuito do robô. Na Figura 4.2 o sensor de medição de ângulo como a pinagem de alimentação de energia e transmissão de dados para controle dos motores que acionam as rodas. Na Figura 4.3 relata a configuração das conexões com o microcontrolador Arduino e também a etapa do sistema de potência.

Figura 4.2 – Conexão Arduino + MPU 6050



Fonte: (Do autor 2022.)

Figura 4.3 – Conexão Arduino + Etapa de Potência



Fonte: (Do autor 2022.)

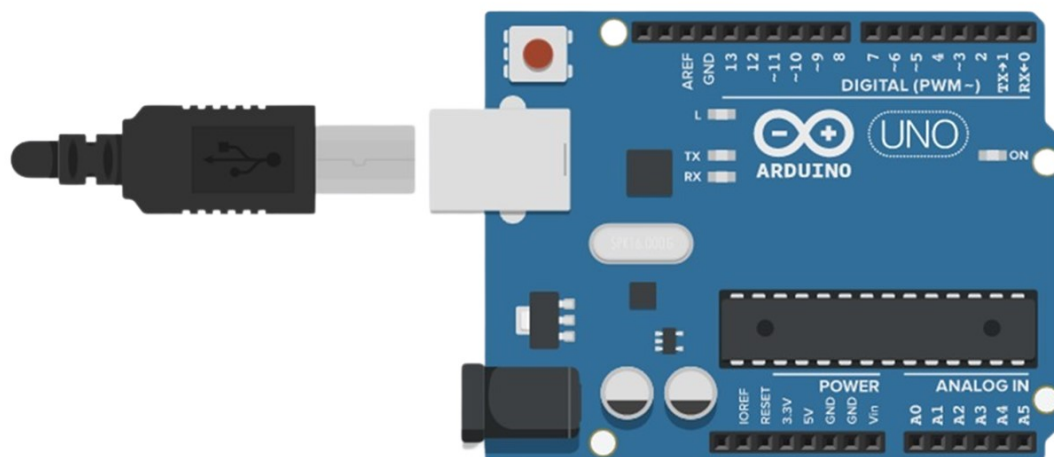
4.3 Plataforma de Prototipagem Arduino Uno

Arduino é uma plataforma de prototipagem com o hardware livre, Figura 4.4, fundamentada em uma placa com um microcontrolador e um ambiente de desenvolvimento integrado, para projetar e facilitar o uso da eletrônica em projetos de circuitos eletrônicos multidisciplinares. Pertencentes ao tipo de programação de código aberto.

Há uma diversidade dos tipos de placas Arduino, dependendo dos requisitos, com mais entradas e saídas, com Wi-Fi, para aplicações de miniaturização, etc.

Para este projeto foi utilizado o modelo de placa Arduino UNO, que é a placa básica, o mais difundido e de acordo com suas características.

Figura 4.4 – Plataforma de Prototipagem Arduino Uno



Fonte: (Do autor 2022.)

A placa Arduino Uno é baseada no processador ATmega 328. Possui 14 entradas e saídas digitais (das quais 6 fornecem saída PWM), 6 pinos de entradas analógicas, clock de 16 MHz, conexão USB, alimentação, um cabeçalho ICSP e um botão de reinicialização. O microcontrolador tem a opção de ser alimentado através do Conexão USB ou com uma fonte de alimentação externa através do conector Jack e também através do pin chamado Vin. A faixa de tensão recomendada de A alimentação DC é de 7 a 12 volts (normalmente utiliza-se uma fonte de 9V / 1 A).

O Arduino Uno também disponibiliza uma série de facilidades para comunicação com um computador, outros Arduinos ou outros microcontroladores. O ATmega328 oferece comunicação serial UART TTL (5V), que está disponível em pinos digitais 0 (RX) e 1 (TX). Um ATmega16U2 na placa usa esta comunicação serial sobre USB e aparece como uma porta COM virtual no software no computador. O software Arduino através da comunicação serial permite a transmissão e recepção dados de texto simples a serem enviados a placa Arduino. Os LEDs RX e TX na placa pisca quando os dados são transmitidos através do chip USB para conexão serial e USB ao computador. O ATmega328 também é compatível com Comunicação I2C (biblioteca Wire) e SPI (usando a biblioteca SPI). A placa possui um multifusível reinicializável que protege as portas USB do seu computador contra curtos-circuitos e sobrecorrentes e suas dimensões são 6,9 e 5,3 cm (SILVEIRA, 2018).

4.4 Unidade de Medição Inercial (IMU)

A Unidade de Medição Inercial ou IMU, Figura 4.5, é um dispositivo eletrônico que possui a capacidade de medir e registrar informações obtidas sobre velocidade, orientação

e efeitos das forças gravitacionais.

No caso deste projeto, o circuito é montado com o IMU MPU-6050, com a finalidade de almejar leituras precisas sobre o ângulo de rotação do robô.

Figura 4.5 – IMU MPU-6050



Fonte: (NAWAZI, 2022)

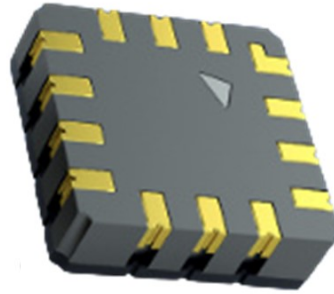
O sensor MPU-6050 usa uma combinação de acelerômetro e giroscópios para obter dados de sistemas de 6 graus de liberdade, que podem ser estendidos até 9 incluindo magnetômetros.

Para os requisitos do robô será suficiente usar o sistema de 6 graus de liberdade que inclui 3 acelerômetros dispostos em um ortogonal e 3 giroscópios também dispostos ortogonalmente.

- Acelerômetro

Os acelerômetros MEMS (“Sistemas Microeletromecânicos”), Figura 4.6, são de tamanhos reduzidos e podem ser incluídos facilmente nas IMUs. A principal vantagem desses dispositivos MEMS é que eles podem ser criados usando técnicas de fabricação microeletrônica em um chip de silício, tecnologia necessária para o sistema de condicionamento, aquisição e comunicação a um preço viável e tamanho moderado (ALSINA, 2016).

Figura 4.6 – Sistemas Microeletromecânicos

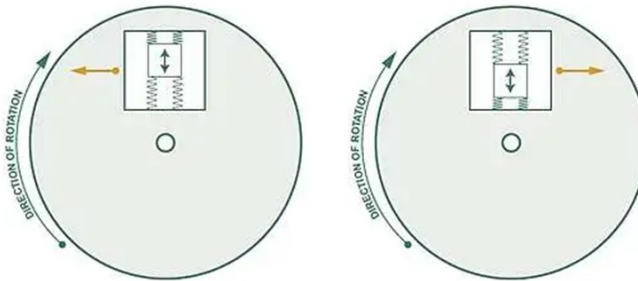


Fonte: (ALSINA, 2016)

- Giroscópio

Os sensores giroscópicos incluídos na IMU também são MEMS, sua operação é baseada em uma pequena massa que varia sua posição variando velocidade angular, o dispositivo converte essas mudanças em um sinal mensurável (ALSINA, 2016).

Figura 4.7 – Giroscópio

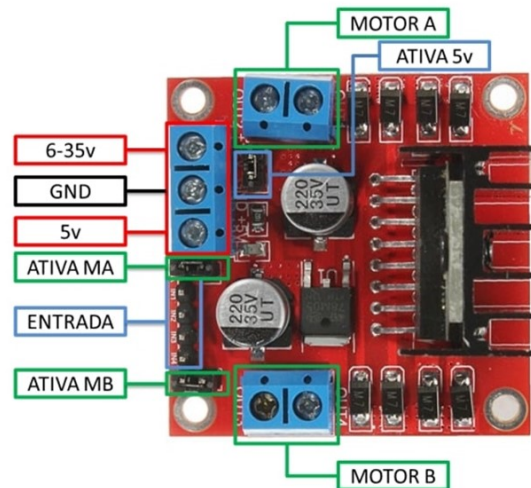


Fonte: (EDN, 2016)

4.5 Ponte H (Controle de Motores)

Para o controle e alimentação dos dois motores, o driver ou controlador L298N conforme Figura 4.8. Este módulo tem como base o chip L298N que permite controlar dois motores DC ou um motor de passo bipolar até 2 amperes.

Figura 4.8 – Driver L298N



Fonte: (THOMSEN, 2018)

O drive permite controlar o sentido de rotação e a velocidade por meio de sinais TTL através do microcontrolador. O módulo possui todos os componentes necessários para funcionar sem necessidade de elementos adicionais, incluindo diodos de proteção e um Regulador LM7805 que fornece 5V para a parte lógica do L298N integrado. Possui jumpers de seleção para habilitar cada uma das saídas do módulo (*A* e *B*) para cada um dos dois motores. A saída *A* é composta através de OUT1 e OUT2 e saída *B* através de OUT3 e OUT4. Os pinos de habilitação são ENA e ENB respectivamente.

Quando o jumper de seleção 5V está ativo, o módulo permite uma alimentação entre 6V a 12V DC. Como o regulador está ativo, o pino marcado com +5V terá uma tensão de 5V DC. Esta tensão pode ser usada para alimentar a parte de controle do módulo, seja um microcontrolador ou um Arduino, desde que o consumo não seja superior a 500 mA.

Quando o jumper de seleção 5V está inativo, o módulo permite uma fonte de alimentação entre 12V a 35V DC.

4.6 Atuadores

Para a correção da posição do robô e o desvio do ângulo, é necessário exercer um torque adequado no ponto de rolagem, isso será obtido com o uso de motores.

No projeto foram utilizados dois motores de corrente contínua, um para a rotação de cada roda. São motores DC 6V 210RPM com encoder, Figura 4.9, e possui 6 pinos para ligação do motor conforme a Figura 4.10.

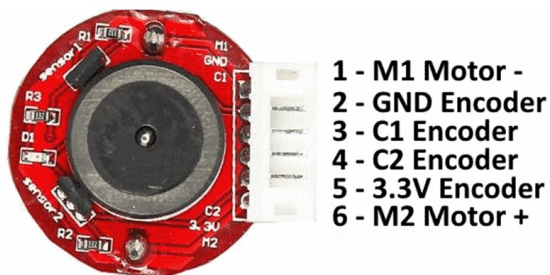
Esses motores são controlados por saídas digitais por largura de pulso microcontrolador PWM. O sentido de rotação desses motores também é controlável de acordo com a disposição da alimentação, mudando o sentido para o inverter a polaridade de seus terminais.

Figura 4.9 – Motor DC 6V 210RPM com Encoder



Fonte: (LIMA, 2019)

Figura 4.10 – Pinagem do Motor



Fonte: (CIA, 2016)

4.7 Parâmetros do sistema

Os parâmetros do robô projetado são dados pela tabela 4.1 e foram calculados a partir das equações do modelo matemático conforme apresentado no capítulo 2 e obtidos no respectivo datasheet. O modelo do sistema é formulado com os seguintes parâmetros: O corpo do robô pode ser visto como um corpo retangular de massa uniforme, e dessa maneira a inercia pode ser calculada usando a mecânica clássica. Onde m_p é a massa do pêndulo, l e h são a largura e altura, respectivamente.

$$J_p = \frac{1}{12}m_p l^2 + \frac{1}{3}m_p h^2 \quad (4.1)$$

Da mesma maneira como calculado a inercia do pêndulo, a inercia da roda pode ser calculada através da mecânica clássica, porem agora utilizando a formula para o círculo.

$$J_r = \frac{m_r r^2}{2} \quad (4.2)$$

Tabela 4.1 – Parâmetros do Robô Pêndulo Invertido Sobre Rodas

Parâmetros	Unidade	Valor
g	m/s^2	9,81
L	m	0,12
h	m	0.21
m_p	kg	1,2
J_p	kgm^2	0,0191
m_r	kg	0,036
J_r	kgm^2	0,000017298
r	m	0,031
V_n	V	12
R_a	Ω	1,7
I_a	A	0,4
N	rpm	280
T_m	Nm	0.044
K_e	Vs/rad	0,39
K_t	Nm/A	0,11

Os valores que envolve a massa, largura, raio e altura foram coletados com o auxilio de uma balança de cozinha digital e uma trena. Outro equipamento utilizado nas medições foi o multímetro para mostrar o valores de resistência de armadura dos motores cc.

Por conseguinte, o sistema pode ser testado, de forma que todas as suas partes estão completas.

5 Conclusão

Quando o assunto diz respeito a sistema de controle é inegável as suas diversas aplicações e a importância de sua existência e do seu correto funcionamento que contribuem para todos os aspectos da sociedade moderna. O objetivo deste trabalho foi projetar e implementar um robô capaz de representar e controlar um sistema de pêndulo invertido sobre rodas, explorando os recursos oferecidos pelo software Matlab 2016a. O projeto proporcionou grandes desafios, devido apresentar um sistema não linear de quarta ordem e instável, e também mostrou-se não observável, assim sendo, a estratégia de projeto seguida foi o modelo de ordem reduzida que proporcionou resultados satisfatórios.

Diante disso, o proposto oferecido para este projeto foi alcançado, entretanto, o trabalho apresenta ainda a descrição e implementação da planta real, a ser utilizada no desenvolvimento da estratégia de controle em trabalhos futuros. .

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à identificação de sistemas—Técnicas lineares e não-lineares aplicadas a sistemas reais**. [S.l.]: Editora UFMG, 2004. Citado na página 17.
- ALSINA, G. **Acelerômetros triaxiais MEMS**. [S.l.], 2016. Disponível em: <<https://www.diarioelectronico hoy.com>>. Citado 2 vezes nas páginas 50 e 51.
- AMBA, A. J. Feedback linearization, sliding mode and swing up control for the inverted pendulum on a cart. 2015. Citado na página 30.
- BALANDUINO. **Robô Pêndulo Invertido**. [S.l.], 2022. Disponível em: <<https://balanduino.tkjelectronics.dk/>>. Citado na página 13.
- BONAFILIA, B.; GUSTAFSSON, N.; NYMAN, P.; NILSSON, S. Self-balancing two-wheeled robot. **Web**.< <http://sebastiannilsson.com/wpcontent/uploads/2013/05/Self-balancing-two-wheeled-robotreport.pdf>, 2015. Citado na página 17.
- BOT, P. **Robô de Telepresença**. [S.l.], 2023. Disponível em: <<https://pluginbot.ai/robotelepresenca/>>. Citado na página 14.
- BURNS, R. **Advanced control engineering**. [S.l.]: Elsevier, 2001. Citado na página 18.
- CARDOSO, R. **Sistemas de Controle 2 -Alocação de polos**. [S.l.], 2020. Disponível em: <https://www.youtube.com/watch?v=9KrvCj0ufNA&list=PLopvEOWJbKA0HWNsL0eqINamKHCSaDREb&index=11&ab_channel=RAFAELCARDOSO>. Citado 2 vezes nas páginas 32 e 33.
- CARDOSO, R. **Sistemas de Controle 2 -Observador de estado**. [S.l.], 2020. Disponível em: <https://www.youtube.com/watch?v=chS_W6pRspg&list=PLopvEOWJbKA0HWNsL0eqINamKHCSaDREb&index=16&ab_channel=RAFAELCARDOSO>. Citado na página 41.
- CIA, A. e. **Como usar motor DC com encoder no Arduino**. [S.l.], 2016. Disponível em: <<https://www.arduinoecia.com.br/motor-dc-com-encoder-arduino/>>. Citado na página 53.
- DORF, R.; BISHOP, R. Sistemas de controle modernos. 11. **Ed. Rio de Janeiro: LCT**, 2009. Citado na página 17.
- EDN. **Detecção inercial de precisão do giroscópio MEMS em ambientes hostis e de alta temperatura**. [S.l.], 2016. Disponível em: <<https://www.edn.com/mems-gyroscope-precision-inertial-sensing-in-harsh-high-temperature-environments/>>. Citado na página 51.
- GOMIDE, F. A. C.; GUDWIN, R. R. Modelagem, controle, sistemas e lógica fuzzy. **SBA controle & Automação**, v. 4, n. 3, p. 97–115, 1994. Citado na página 13.
- LIMA, P. E. M. d. Plataforma de teste para sistemas de controle baseada em pêndulo invertido sobre duas rodas. 2019. Disponível em: <<https://bdm.unb.br/handle/10483/23552>>. Citado na página 53.

MALIK, T. **Elon Musk diz que a SpaceX tentará pousar um foguete com um 'balão de festa gigante'**. [S.l.], 2018. Disponível em: <<https://www.space.com/40313-spacex-rocket-landing-giant-party-balloon.html>>. Citado na página 14.

NAWAZI, F. **Interface MPU6050 Sensor de acelerômetro e giroscópio com Arduino**. [S.l.], 2022. Disponível em: <<https://www.circuits-diy.com/interface-mpu6050-accelerometer-and-gyroscope-sensor-with-arduino/>>. Citado na página 50.

OGATA, K. et al. **Modern control engineering**. [S.l.]: Prentice hall Upper Saddle River, NJ, 2010. v. 5. Citado 6 vezes nas páginas 17, 18, 33, 34, 35 e 40.

SCHILING, D. E. Modelagem e controle de um robô de eixo único em sistema de pêndulo invertido. 2017. Disponível em: <<https://bibliodigital.unijui.edu.br:8443/xmlui/handle/123456789/4810>>. Citado na página 22.

SEGWAY. **Diciclo Segway**. [S.l.], 2017. Disponível em: <<http://urbanwheels.co.nz/>>. Citado na página 14.

SILVEIRA, A. L. M. da. **Arduino Uno R3 - ATmega328P 5V 16MHz**. [S.l.], 2018. Disponível em: <<http://www.um.pro.br/arduino/index.php?c=ArduinoUno>>. Citado na página 49.

THOMSEN, A. **Motor DC com Driver Ponte H L298N**. [S.l.], 2018. Disponível em: <<https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>>. Citado na página 52.

VENDRAMINI, G.; SILVA, P. S. da. Controle de um pêndulo invertido sobre uma plataforma móvel utilizando pid e mfac (model-free adaptive control). In: **Proceeding of the 9th Brazilian Conference on Dynamics, Control and Their Applications (DINCON'10)**. [S.l.: s.n.], 2010. p. 405–414. Citado na página 18.

A ANEXO: CÓDIGOS MATLAB 2016a

A.1 Códigos Modelagem Pêndulo Robô

```

clear, clc

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

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

% Equações de estado:
f1 = omega;
f2 = phi ;
f3 = (L*r*mp*cos(psi)*(g*L*Ra*mp*sin(psi) + 2*n*Kt*(u-Ke*omega)) + Jp*(2*n*Kt*(u-Ke*omega) - L*r*Ra*mp*(phi^2)*sin(psi)))/(Ra*(Jp*(Jr+(r^2)*(mp+mr)) - (L*r*mp*cos(psi))^2));
f4 = (Jr*(g*L*Ra*mp*sin(psi) + 2*n*Kt*(u-Ke*omega)) + r*(L*r*Ra*mp*sin(psi)*(mp*(g-L*(phi^2)*cos(psi)) + g*mr) + 2*n*Kt*(u-Ke*omega)*(mp*(L*cos(psi)+r)+r*mr)))/(Ra*(Jp*(Jr+(r^2)*(mp+mr)) - (L*r*mp*cos(psi))^2));

% Modelo dinâmico não-linear:
dx = [f1;
      f2;
      f3;
      f4;
      ];

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

% Ponto de operação:
theta = 0;
psi = 0;
omega = 0;
phi = 0;

% Parâmetros:
g = 9.81; % Gravidade;
L = 0.12; % Comprimento do pêndulo;
r = 0.031; % Raio da roda;
h = 0.21; % Altura;
mp = 1.2; % Massa do pêndulo;
mr = 0.036; % Massa da roda;
Ra = 1.7; % Resistência da armadura;
n = 1/21.3; % Fator da caixa de redução;
Vn = 12; % Tensão nominal;
Ian = 0.4; % Corrente nominal;
N = 280; % ?
Ke = (Vn-Ian*Ra)*60/(2*pi*N); % Constante elétrica;

```

```
Tm = 0.044; % Torque;
Kt = Tm/Ian; % Constante de torque;
Jp = (1/12)*(mp*L^2) + (1/3)*(mp*h^2); % Inércia do pêndulo;
Jr = mr*r^2/2; % Inércia da roda;
```

```
% Substitui os valores nas matrizes:
```

```
A = double(vpa(subs(A),6));
B = double(vpa(subs(B),6));
C = [ 1 0 0 0];
D = 0;
```

```
[num,den] = ss2tf(A,B,C,D);
C = eye(4);
G = tf(num,den)
```

```
save ModeloLinear A B C D G
```

```
% Código que implementa uma s-function para um
% sistema dinâmico Pêndulo:

% Função de iteração e configuração:
function[sys,x0,str,ts] = Pendulo_rob(t,x,u,flag)

    switch flag

        case 0

            str=[];
            ts=[0 0];
            s=simsizes;
            s.NumContStates = 4;
            s.NumDiscStates = 0;
            s.NumOutputs = 4;
            s.NumInputs = 1;
            s.DirFeedthrough = 0;
            s.NumSampleTimes = 1;

            sys = simsizes(s);

            x0 = [0; 0.05; 0; 0];

        case 1
            sys = EqEstados(t,x,u);

        case 3
            sys = Saidas(t,x);

        case{2 4 9}
            sys=[];

        otherwise
            error(['unhandled flag =',num2str(flag)]) ;

    end
end
```

```
function dx = EqEstados(~,x,u)

% Vetor de estados:
% x = [theta; % Posição da roda;
%      psi; % Posição do pêndulo;
%      omega; % Velocidade da roda;
%      phi]; % Velocidade do pêndulo;

% Estados:
theta = x(1);
psi = x(2);
omega = x(3);
```

```

phi = x(4);

% Parâmetros:
g = 9.81; % Gravidade;
L = 0.12; % Comprimento do pêndulo;
r = 0.031; % Raio da roda;
h = 0.21; % Altura;
mp = 1.2; % Massa do pêndulo;
mr = 0.036; % Massa da roda;
Ra = 1.7; % Resistência da armadura;
n = 1/21.3; % Fator da caixa de redução;
Vn = 12; % Tensão nominal;
Ian = 0.4; % Corrente nominal;
N = 280; % ?
Ke = (Vn-Ian*Ra)*60/(2*pi*N); % Constante elétrica;
Tm = 0.044; % Torque;
Kt = Tm/Ian; % Constante de torque;
Jp = (1/12)*(mp*L^2) + (1/3)*(mp*h^2); % Inércia do pêndulo;
Jr = mr*r^2/2; % Inércia da roda;

% Equações de Estados:
dtheta = omega;
dpsi = phi ;
domega = (L*r*mp*cos(psi)*(g*L*Ra*mp*sin(psi) + 2*n*Kt*(u-Ke*omega)) + Jp*(2*n*Kt*(u-Ke*omega) - L*r*Ra*mp*(phi^2)*sin(psi)))/(Ra*(Jp*(Jr+(r^2)*(mp+mr)) - (L*r*mp*cos(psi))^2));
dphi = (Jr*(g*L*Ra*mp*sin(psi) + 2*n*Kt*(u-Ke*omega)) + r*(L*r*Ra*mp*sin(psi)*(mp*(g-L*(phi^2)*cos(psi)) + g*mr) + 2*n*Kt*(u-Ke*omega)*(mp*(L*cos(psi)+r)+r*mr)))/(Ra*(Jp*(Jr+(r^2)*(mp+mr)) - (L*r*mp*cos(psi))^2));

% Variação dos estados:
dx = [dtheta; dpsi; domega; dphi];
end

function ys = Saidas(~,x)
    ys = x;
end

```

```
clear,clc

load('DataMA.mat')

t = DadosMA(1,:);
x1 = DadosMA(2,:);
x2 = DadosMA(3,:);
x3 = DadosMA(4,:);
x4 = DadosMA(5,:);

figure(1)
subplot(2,2,1)
plot(t,x1*180/pi,'r','linewidth',2)
title('Posição da roda')
xlabel('Tempo (s)')
ylabel('Theta (t)')
xlim([0,3])
grid on

subplot(2,2,2)
plot(t,x2*180/pi,'r','linewidth',2)
title('Posição do pêndulo')
xlabel('Tempo (s)')
ylabel('Psi (t)')
xlim([0,3])
grid on

subplot(2,2,3)
plot(t,x3,'r','linewidth',2)
title('Velocidade da roda')
xlabel('Tempo (s)')
ylabel('Omega (t)')
xlim([0,3])
grid on

subplot(2,2,4)
plot(t,x4,'r','linewidth',2)
title('Velocidade do pêndulo')
xlabel('Tempo (s)')
ylabel('Phi (t)')
xlim([0,3])
grid on
```

A.2 Códigos Projeto do Controlador

```
clear, clc

load('ModeloLinear.mat')
%%
% Polos desejados do Controlador
ts = 1.5;
qsi = 0.8;
wn = 4/(ts*qsi);
p1 = -qsi*wn+1i*wn*sqrt(1-qsi^2);
p2 = -qsi*wn-1i*wn*sqrt(1-qsi^2);
p3 = -60;
p4 = -85;
%% Fórmula de Ackermann

polos = [p1, p2, p3, p4];

K = acker(A,B,polos)

%%

Ac = A-B*K;
Bc = B;
Cc = C;
Dc = D;
WIP = ss(Ac,Bc,Cc,Dc)

x0 = [0;0.1;0;0];

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

subplot(311)
plot(t,y(:,1),'linewidth',1.8)
grid
subplot(312)
plot(t,y(:,2),'linewidth',1.8)
grid
subplot(313)
plot(t,y(:,3),'linewidth',1.8)
grid
```

```
clear,clc

load('DataMF.mat')

t = DadosMF(1,:);
x1 = DadosMF(2,:);
x2 = DadosMF(3,:);
x3 = DadosMF(4,:);
x4 = DadosMF(5,:);

figure(1)
subplot(2,2,1)
plot(t,x1,'r','linewidth',2)
title('Posição da roda')
xlabel('Tempo (s)')
ylabel('Theta (t)')
xlim([0,2])
grid on

subplot(2,2,2)
plot(t,x2,'r','linewidth',2)
title('Posição do pêndulo')
xlabel('Tempo (s)')
ylabel('Psi (t)')
xlim([0,2])
grid on

subplot(2,2,3)
plot(t,x3,'r','linewidth',2)
title('Velocidade da roda')
xlabel('Tempo (s)')
ylabel('Omega (t)')
xlim([0,2])
grid on

subplot(2,2,4)
plot(t,x4,'r','linewidth',2)
title('Velocidade do pêndulo')
xlabel('Tempo (s)')
ylabel('Phi (t)')
xlim([0,2])
grid on
```

```
clear,clc
```

```
load('DataMF_tensao.mat')
```

```
t = DadosMF(1,:);  
x1 = DadosMF(2,:);
```

```
figure(1)  
subplot(1,1,1)  
plot(t,x1,'r', 'linewidth',2)  
title('Ganho')  
xlabel('Tempo (s)')  
ylabel('Tensão(t)')  
xlim([0,2])  
grid on
```

```
clear,clc
```

```
load('DataMF_saturacao.mat')
```

```
t = DadosMF(1,:);  
x1 = DadosMF(2,:);
```

```
figure(1)  
subplot(1,1,1)  
plot(t,x1,'r', 'linewidth',2)  
title('Ganho')  
xlabel('Tempo (s)')  
ylabel('Tensão(t)')  
xlim([0,2])  
grid on
```

A.3 Códigos do Projeto Controlador Observador

```

clear, clc

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

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

% Equações de estado:
f2 = phi;

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

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

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

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

% Ponto de operação:
psi = 0;
omega = 0;
phi = 0;

% Parâmetros:
g = 9.81; % Gravidade;
L = 0.12; % Comprimento do pêndulo;
r = 0.031; % Raio da roda;
h = 0.21; % Altura;
mp = 1.2; % Massa do pêndulo;
mr = 0.036; % Massa da roda;
Ra = 1.7; % Resistência da armadura;
n = 1/21.3; % Fator da caixa de redução;
Vn = 12; % Tensão nominal;
Ian = 0.4; % Corrente nominal;
N = 280; % RPM
Ke = (Vn-Ian*Ra)*60/(2*pi*N); % Constante elétrica;
Tm = 0.044; % Torque;
Kt = Tm/Ian; % Constante de torque;

```

```
Jp = (1/12)*(mp*L^2) + (1/3)*(mp*h^2); % Inércia do pêndulo;  
Jr = mr*r^2/2; % Inércia da roda;
```

```
% Substitui os valores nas matrizes:
```

```
A = double(vpa(subs(A),6));
```

```
B = double(vpa(subs(B),6));
```

```
C = [1 0 0];
```

```
D = 0;
```

```
[num,den] = ss2tf(A,B,C,D);
```

```
C = eye(3);
```

```
G = tf(num,den);
```

```
save ModeloLinear3 A B C D G
```

```
clear, clc

load('ModeloLinear3.mat')
%%
% Polos desejados:
ts = 1.5;
qsi = 0.8;
wn = 4/(ts*qsi);
p1 = -qsi*wn+1i*wn*sqrt(1-qsi^2);
p2 = -qsi*wn-1i*wn*sqrt(1-qsi^2);
p3 = -60;

polos = [p1, p2, p3];

K =acker(A,B,polos)

%%

Ac = A-B*K;
Bc = B;
Cc = C;
Dc = D;

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

x0 = [0.1;0;0];

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

subplot(311),
plot(t,y(:,1),'linewidth',1.8)
grid
subplot(312),
plot(t,y(:,2),'linewidth',1.8)
grid
subplot(313),
plot(t,y(:,3),'linewidth',1.8)
grid

%% Análise de Observabilidade:
C = [1 0 0];
Ob = obsv(A,C);
% O sistema é observável.
det(Ob)

%%

% Polos desejados do observador:
ts = 1.5;
qsi = 0.8;
wn = 4/(ts*qsi);
po1 = -4*qsi*wn + 1i*wn*sqrt(1-qsi^2);
po2 = -4*qsi*wn - 1i*wn*sqrt(1-qsi^2);
po3 = -4*60;
```

```
Po = [po1 po2 po3];
```

```
%% Projeto do observador:
```

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

```
clear, clc
load('DadosControle.mat')
Dados = ans;

%%
t = Dados(1,:);
x1 = Dados(2,:);
x2 =Dados(3,:);
x3 =Dados(4,:);
x4 = Dados(5,:);
x2o = Dados(6,:);
x3o =Dados(7,:);
x4o =Dados(8,:);

figure(1)
subplot(2,2,1)
plot(t,x1,'r', 'linewidth',2)
title('Posição da roda')
xlabel('Tempo (s)')
ylabel('Theta (t)')
xlim([0,2])
grid on

subplot(2,2,2)
plot(t,x2,'r', 'linewidth',2)
title('Posição do pêndulo')
xlabel('Tempo (s)')
ylabel('Psi (t)')
xlim([0,2])
grid on
hold on
plot(t,x2o,'-b', 'linewidth',2)
legend('x2','x2o')
hold off

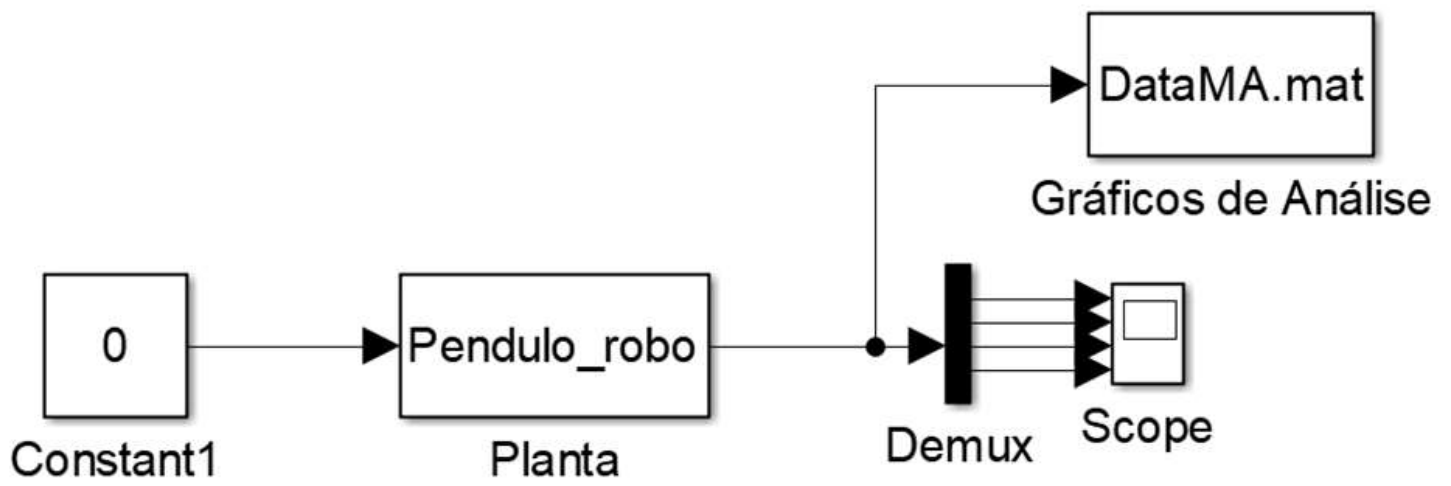
subplot(2,2,3)
plot(t,x3,'r', 'linewidth',2)
title('Velocidade da roda')
xlabel('Tempo (s)')
ylabel('Omega (t)')
xlim([0,2])
grid on
hold on
plot(t,x3o,'b', 'linewidth',2)
legend('x3','x3o')
hold off

subplot(2,2,4)
plot(t,x4,'r', 'linewidth',2)
title('Velocidade do pêndulo')
xlabel('Tempo (s)')
ylabel('Phi (t)')
xlim([0,2])
grid on
```

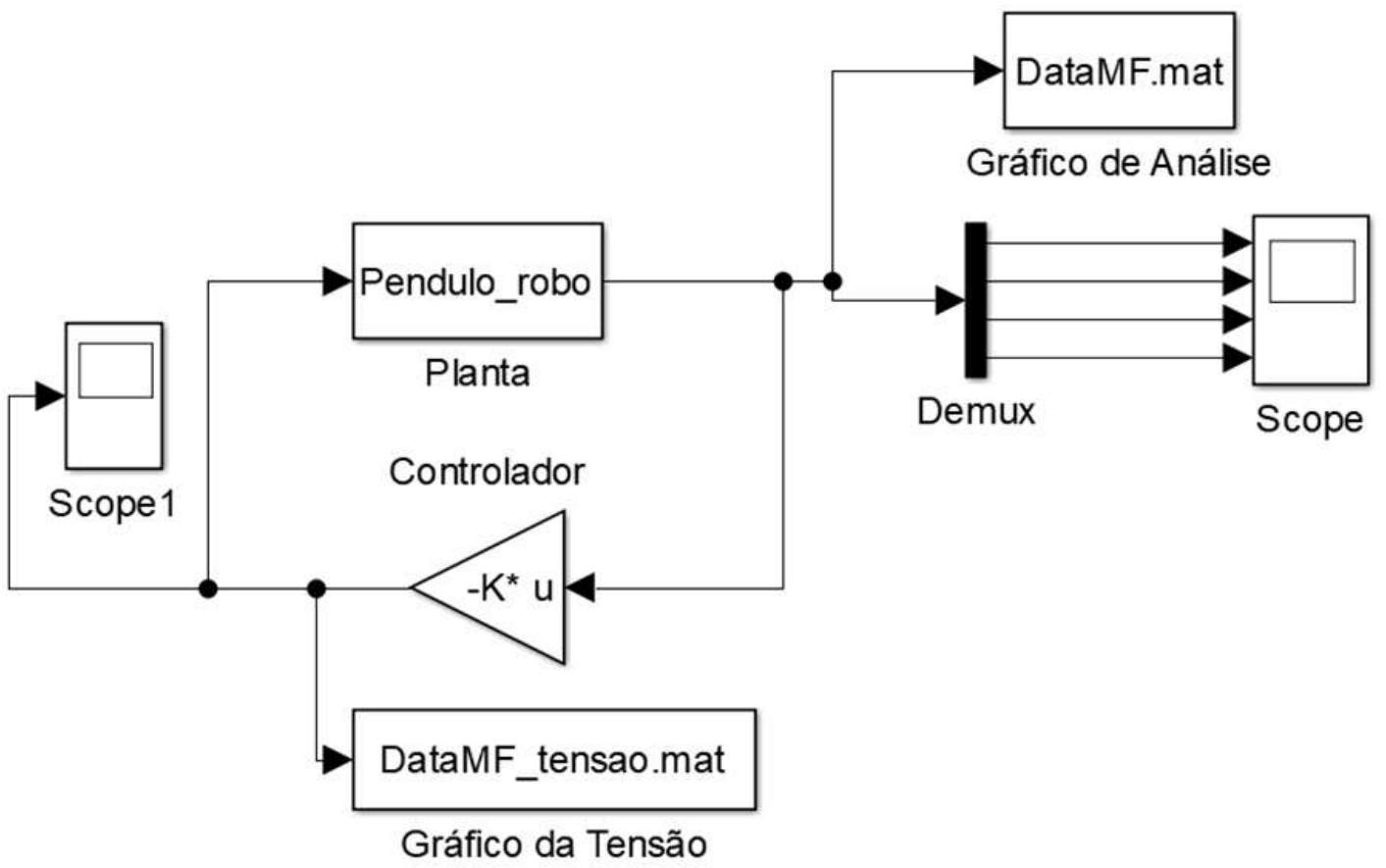
```
hold on
plot(t,x4o,'b', 'linewidth',2)
legend('x4','x4o')
hold off
```

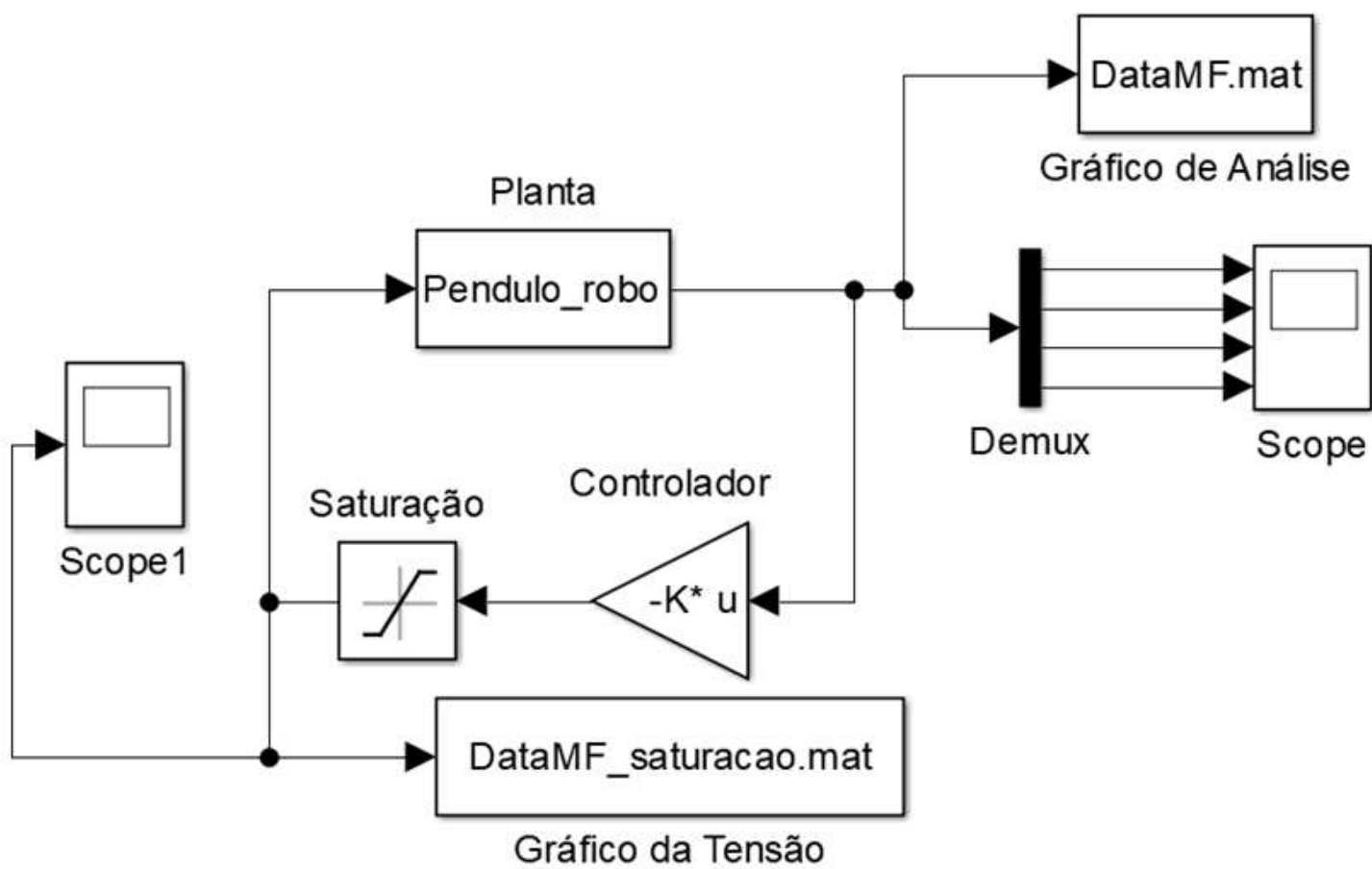
B ANEXO: CIRCUITOS NO SIMULINK

B.1 Simulação Malha Aberta



B.2 Simulação Malha fechada





B.3 Simulação Controlador Observador

