Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

Controle linear e não linear por realimentação de estados: Uma aplicação no pêndulo invertido

Felipe Bastos Chaves

João Monlevade, MG
2024

Felipe Bastos Chaves

Controle linear e não linear por realimentação de estados: Uma aplicação no pêndulo invertido

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.
Orientador: Prof. Dr. Márcio Feliciano Braga

Universidade Federal de Ouro Preto
João Monlevade
2024

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C512c Chaves, Felipe Bastos.
Controle linear e não linear por realimentação de estados [manuscrito]: uma aplicação no pêndulo invertido. / Felipe Bastos Chaves. - 2024.
92 f.

Orientador: Prof. Dr. Márcio Feliciano Braga.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia Elétrica .

1. Controladores programáveis. 2. Controle automático. 3. Modelos matemáticos. 4. Pêndulo. 5. Sistemas de controle por realimentação. 6. Sistemas lineares de controle. I. Braga, Márcio Feliciano. II. Universidade Federal de Ouro Preto. III. Título.

CDU 681.5

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE ENGENHARIA ELETRICA

FOLHA DE APROVAÇÃO

Felipe Bastos Chaves

Controle linear e não linear por realimentação de estados: Uma aplicação no pêndulo invertido

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia Elétrica

Aprovada em 06 de agosto de 2024.

Membros da banca

Dr. Márcio Feliciano Braga — Orientador — Universidade Federal de Ouro Preto
Ma. Anny Verly — Convidada — Universidade Federal de Ouro Preto
Dr. Rodrigo Augusto Ricco — Convidado — Universidade Federal de Ouro Preto

Márcio Feliciano Braga, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 04/09/2024.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0818 - www.ufop.br

Dedico este trabalho aos meus pais e a minha irmã, que sempre me apoiaram e me possibilitaram concluir esta longa jornada.

Agradecimentos

A Deus, pela graça de ter me permitido concluir este trabalho. Aos meus pais, Maria de Lourdes e Eduardo, os quais me deram todo o apoio necessário. Ao meu grande amigo Matheus Henrique Fernandes por todo o apoio dado durante a graduação e escrita do texto. Ao meu orientador Prof. Dr. Márcio Feliciano Braga pela ajuda no desenvolvimento, colaboração, paciência e dedicação na elaboração deste trabalho.

"E a paz de Deus, que excede todo o entendimento, guardará os vossos corações e os vossos pensamentos em Cristo Jesus".

Filipenses 4:7

Resumo

Os sistemas de controle presentes no mundo contemporâneo exigem grande precisão, são complexos, com múltiplas entradas e saídas, alguns são não lineares e variantes no tempo. A teoria clássica de controle nem sempre é adequada para lidar com tais tipos de sistemas, sendo necessário o uso da teoria moderna de controle. Com o desenvolvimento de modelos em espaço de estados, assim como a evolução do poder computacional, se tornou viável a análise de sistemas no domínio do tempo, inclusive para modelos os quais não possuem solução analítica. O objetivo desta monografia é aplicar ferramentas da teoria de controle moderno no sistema do pêndulo invertido, o qual é intrinsecamente instável. São projetados controladores para os modelos linear e não linear que descrevem a dinâmica do pêndulo invertido. O controle é implementado é por realimentação de estados e também adiciona-se um seguidor de referência na malha do sistema. Os resultados mostram a viabilidade da técnica proposta por estabilizar o sistema e permitir que uma dada referência, com o menor erro possível, seja seguida. Adicionalmente, o projeto e emprego de observadores de estado, para o caso em que a medição direta dos estados não estava disponível, foram testados. Os resultados demonstraram que o comportamento do sistema em malha fechada é idealmente idêntico ao que seria obtido se as medições dos estados estivessem disponíveis para realimentação. Por fim, foram testadas as técnicas de controle por linearização entrada-saída e o teorema de estabilidade de Lyapunov no modelo não linear. Os resultados obtidos mostraram a capacidade das técnicas em estabilizar o sistema e seguir uma dada referência, sendo que a aplicação conjunta da linearização entrada-saída e o teorema de estabilidade de Lyapunov obteve melhores resultados (rapidez na estabilização e menor esforço de controle) do que a simples aplicação da linearização entrada-saída.

Palavras-chave: Pêndulo invertido, espaço de estados, controle por realimentação de estados, seguimento de referência, controle não linear.

Abstract

The control systems present in the contemporary world require great precision, are complex, with multiple inputs and outputs, some are non-linear and time-varying. Classical control theory is not always adequate to deal with these types of systems, making it necessary to use modern control theory. With the development of state space models, as well as the evolution of computational power, the analysis of systems in the time domain has become viable, including for models that do not have an analytical solution. The objective of this monograph is to apply tools from modern control theory to the inverted pendulum system, which is intrinsically unstable. Controllers are designed for linear and nonlinear models that describe the dynamics of the inverted pendulum. The control is implemented by state feedback and a reference follower is also added to the system loop. The results show the viability of the proposed technique by stabilizing the system and allowing a given reference, with the smallest possible error, to be followed. Additionally, the design and employment of state observers, for the case where direct measurement of states was not available, were tested. The results demonstrated that the behavior of the closed-loop system is ideally identical to what would be obtained if state measurements were available for feedback. Finally, input-output linearization control techniques and Lyapunov's stability theorem were tested in the nonlinear model. The results obtained showed the techniques' ability to stabilize the system and follow a given reference, with the joint application of input-output linearization and Lyapunov's stability theorem obtaining better results (faster stabilization and less control effort) than the simple application of input-output linearization.

Keywords: Inverted pendulum, state space, state feedback control, reference tracking, nonlinear control.

Lista de ilustrações

Lista de tabelas

Lista de siglas e abreviaturas

T_p	Tempo de pico (T_p, do inglês *Peak Time*)
T_r	Tempo de subida (T_r, do inglês *Rise Time*)
T_s	Tempo de assentamento (T_s, do inglês *Settling Time*)
FT	Função de Transferência
IDE	Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*)
LGR	lugar geométrico das raízes
MIMO	múltiplas entradas e múltiplas saídas (MIMO, do inglês *multiple imput multiple output*)
MUP	Máxima Ultrapassagem Percentual
PI	Proporcional e integral
PID	Proporcional Integral e Derivativo
SNR	Relação sinal-ruído (SNR, do inglês *signal-noise relation*)
SS	Espaço de Estados (SS, do inglês *state spaces*)

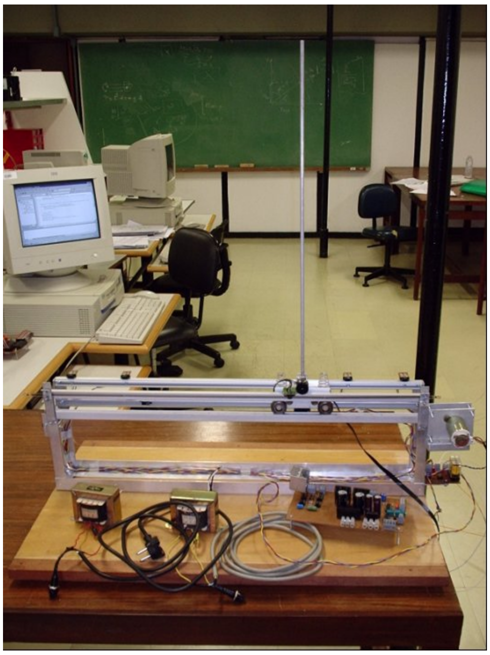
Sumário

1 Introdução

O pêndulo invertido é um sistema base a partir do qual muitos outros processos são modelados, como, por exemplo, isolamento de baixas frequências fazendo uso de conceitos do pêndulo invertido (TAKAMORI et al., 2007), controle de trajetória e navegação no ambiente de um robô móvel (RAFIKOVA, 2010), aerogeradores *offshore* (GUIMARãES, 2016), entre outros. Esse modelo se mostra valioso em uma variedade de aplicações de engenharia, incluindo, mas não se limitando a, veículos para transporte pessoal, guindastes marítimos, pesquisa robótica e a ciência do controle. De forma simples, o pêndulo invertido consiste em uma barra cilíndrica que se movimenta livremente em torno de um ponto fixo. O ponto fixo fica localizado em um carro ou outro equipamento que permita fazer um movimento horizontal. Vendramini e Silva (2010) destacam que o pêndulo invertido é caracterizado por sua dinâmica instável, o que facilita a exploração de diferentes estruturas e modalidades de controladores.

A Figura 1 ilustra um sistema de pêndulo invertido. O sistema é instável, porque o centro de gravidade está localizado acima do eixo das rodas na estrutura, fazendo com que ele tenda a se deslocar para longe da posição de equilíbrio (RIBEIRO, 2007).

Figura 1 – Exemplo de um pêndulo invertido.



Fonte: Borges (2003).

De acordo com Ogata (2010), o sistema do pêndulo invertido é descrito por equações não lineares de fase não mínima, ou seja, possui zeros instáveis no semiplano positivo. Slotine e Li (1991) destacam que sistemas não lineares com fase não mínima possuem dinâmicas internas instáveis. Essa característica representa um desafio significativo para a implementação de técnicas tradicionais de linearização e controle.

Uma forma de representar sistemas dinâmicos, dentre eles o pêndulo invertido, envolve utilizar a representação em Espaço de Estados (SS, do inglês *state spaces*), que consiste em um modelo matemático formado por um conjunto de variáveis de estado. As variáveis de estado são o conjunto mínimo de variáveis que determinam o comportamento ou a resposta futura de um sistema, considerando o estado presente, os sinais de entrada e as equações que descrevem sua dinâmica (DORF; BISHOP, 2018). A vantagem de se trabalhar com modelos em SS é que eles podem ser aplicados a sistemas não lineares, variantes no tempo, com múltiplas entradas e saídas.

A avaliação em SS engloba três categorias de variáveis: variáveis de entrada, variáveis de saída e variáveis de estado. A representação de um sistema específico em SS não é singular, contudo, o número de variáveis de estado permanece constante para todas as diversas representações em SS (OGATA, 2010).

Para se fazer um projeto de controle em SS, é necessário que o vetor de estados esteja completamente disponível para ser mensurado. Porém, em alguns sistemas de controle, não é possível instalar sensores e outros equipamentos para realizar as medições, sejam por razões econômicas ou dificuldade de instalação. Uma maneira de contornar tal problema é empregar observadores de estados (LUENBERGER, 1964 apud DALTIN, 2017). Destaca-se ainda que, a introdução do observador não afeta a estabilidade do sistema como um todo (LUENBERGER, 1971 apud DALTIN, 2017).

O presente trabalho visa, sob a lente da teoria de controle moderno, fazer a modelagem do pêndulo invertido, torná-lo estável, empregando controladores que são projetados utilizando técnicas de controle linear e não linear, e também fazê-lo atender a requisitos de desempenho específicos. Supondo que as variáveis de estado não estejam disponíveis para medição, pretende-se recorrer a um observador de estados, projetando e analisando o desempenho do sistema em malha fechada do conjunto observador-controlador sob a presença de ruídos e erros de leitura dos sensores.

De fato, há um grande interesse da sociedade científica em tratar o problema do controle associado ao pêndulo invertido. Por exemplo, Pires (2023) faz o controle do pêndulo invertido utilizando duas estratégias diferentes: realimentação de estados e controlador Proporcional Integral e Derivativo (PID). O autor enfatiza que, apesar de ambas as estratégias atingirem o objetivo, que é estabilizar o pêndulo invertido, a estratégia de realimentação de estados permite que uma ação de controle seja aplicada em um cenário real. Ao final do trabalho, é sugerido o uso de outras técnicas de controle, como o controle adaptativo.

Alves (2018) faz o estudo de um pêndulo invertido acoplado em um carro. No trabalho, faz-se a modelagem, emprega-se um controle estabilizante e também uma estratégia de controle não linear em SS. O autor faz uma abordagem lagrangeana para obter as equações não lineares do pêndulo invertido. Por fim, o autor faz uso da técnica de linearização entrada-saída e o Teorema de Estabilidade de Lyapunov. A modelagem matemática foi coerente com relação a um sistema real. Também foi satisfeita a proposta de se fazer o sistema seguir um sinal de referência para a posição do carro. Foi garantido que o pêndulo ficasse na posição vertical. Porém, o sistema fica oscilante com a adição de ruídos. Como continuação do trabalho, sugeriu-se a implementação de um controlador robusto.

Gadelha (2018) fez uma modelagem linear e não linear em SS do sistema do pêndulo invertido. O controle foi implementado por meio de lógica *fuzzy*. Foi utilizado o método de alocação de polos, no controle em SS do modelo linear do pêndulo invertido. O controle *fuzzy* demonstrou ser eficaz no tratamento de sistemas tanto lineares, como não lineares. Houve, também, a aplicação de técnicas baseadas no lugar geométrico das raízes (LGR), em que ficou demonstrado que fazer um controle baseado na lógica *fuzzy* garante resultados bem melhores. Como trabalhos futuros, a autora sugere aplicar geometrias diferentes das triangulares e trapezoidais nas funções de pertinência da lógica *fuzzy* e, para gerar um controle mais fino e preciso, aumentar a quantidade de regras proposta por Burns (2001).

Daltin (2017) tem por objetivo fazer o controle de vibrações em estruturas flexíveis com o uso do observador de estados. Nesse trabalho, foram criados modelos matemáticos, baseados no modelo matemático do pêndulo invertido, dedicados a sistemas rotacionais em estruturas flexíveis. Também, foi utilizado um observador de estados para estimar as variáveis de estado com o objetivo de diminuir o número de sensores utilizados e projetar controladores que atenuem vibrações. Os resultados obtidos incluíram o desenvolvimento e a validação experimental de modelos matemáticos para controlar a vibração de uma viga flexível. Além disso, a utilização do observador de estados demonstrou uma ótima estimativa das variáveis de estado, eliminando a necessidade de usar tacômetros e *strain gage*.

Arcolezi et al. (2017) fez a modelagem do pêndulo invertido e a inserção de um controlador proporcional na malha do sistema com o uso de recursos computacionais. Não houve um resultado vantajoso para os polos escolhidos, pois a inserção do controlador proporcional no servossistema do tipo 1 não trouxe acréscimo positivo à técnica de controle para o sistema do pêndulo invertido. Isso ocorreu porque, como característica desse ganho, já era previsto um aumento na amplitude das respostas com o aumento das oscilações, bem como uma diminuição do tempo de acomodação do sistema. No entanto, essa redução no tempo de acomodação não foi notada de forma significativa para qualquer valor do ganho proporcional.

1.1 Justificativa

O pêndulo invertido é conhecido como sendo um problema clássico da área de controle. Conforme Pires (2023) explica, existem muitas variações dele, como pêndulo invertido duplo, pêndulo invertido rotacional e o tradicional pêndulo sobre um carrinho, sendo o último o modelado empregado neste trabalho.

Pelo fato de ter duas saídas, geralmente a posição do carrinho e o ângulo da haste, a abordagem em SS é mais adequada, visto que lida melhor com sistemas com mais de uma entrada e saída. Destaca-se ainda que na teoria em SS, o controlador projetado consegue garantir, desde que algumas condições, como de controlabilidade, sejam asseguradas, que os polos determinados pelo projetista sejam de fato os polos de malha fechada do sistema, o que nem sempre é possível de ser feito em uma abordagem via LGR e métodos frequenciais, dado que se utiliza a estratégia de alocação por polos dominantes.

Outra vantagem relacionada ao SS é que caso os estados não estejam disponíveis para serem medidos, seja por estarem em um local cujo acesso é difícil ou simplesmente é inviável fazer a medição, eles podem ser estimados usando um observador de estados. Além disso, o emprego de observadores não altera a estabilidade do sistema (DALTIN, 2017).

1.2 Objetivos

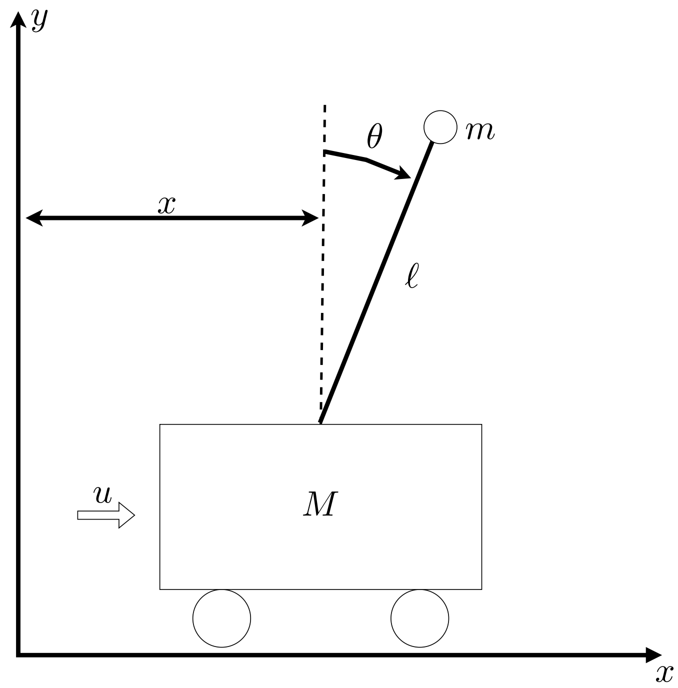
Os objetivos deste trabalho são fazer a modelagem do pêndulo invertido em SS, aplicar técnicas de controle por realimentação de estados, incluindo técnicas não lineares, e inserir um observador de estados. Para isso, os seguintes objetivos específicos devem ser cumpridos.

1.2.1 Objetivos específicos

- Modelar o pêndulo invertido em espaço de estados;
- Projetar controladores para seguimento de referência no modelo linearizado empregando observadores de estado;
- Projetar controladores via linearização entrada-saída para o modelo não linear;
- Analisar e comparar os resultados obtidos por meio das técnicas aplicadas.

A Figura 2 apresenta um diagrama esquemático de um pêndulo invertido, com todas as variáveis e forças necessárias para o pleno entendimento e modelagem do problema. Nela, representa-se a massa do pêndulo por m, a massa do carro por M, o comprimento da haste do pêndulo por ℓ, a força aplicada ao carro por u, a posição do carrinho por x e a posição angular da haste por θ.

Figura 2 – Diagrama esquemático do pêndulo invertido estudado neste trabalho.



Fonte: Autoria própria.

O principal objetivo é o de manter a haste levantada, sendo necessário mover o carrinho para que isso seja possível. A estratégia de modelagem consiste em usar as leis de Newton para obter uma representação matemática do modelo.

1.3 Estrutura do trabalho

No Capítulo 1, faz-se a introdução do trabalho, destacando as vantagens da utilização da teoria de controle moderno, além de serem apresentados a justificativa e os objetivos do trabalho.

No Capítulo 2, apresenta-se uma breve revisão de conceitos matemáticos importantes para lidar com o SS e as técnicas de controle moderno, da teoria clássica de controle, dos tipos de controladores, das técnicas que podem ser utilizadas para projetar um controlador e a apresentação do controlador Proporcional e integral (PI). Em seguida, introduz-se uma revisão sobre SS e definições. Além de discutir sobre a análise de erro em regime permanente, alguns conceitos importantes de álgebra linear para a teoria de controle, conceitos como controlabilidade e observabilidade e projetos de controladores por espaço de estados.

No Capítulo 3, é realizada a modelagem matemática do sistema do pêndulo invertido em espaço de estados. No Capítulo 4, é feita a linearização jacobiana do modelo não linear do pêndulo invertido, o projeto de controlador estabilizante, seguimento de referência e um observador de estados. No Capítulo 5, são projetados controladores não lineares, tanto para estabilização, como para seguimento de referência.

No Capítulo 6, conclui-se o trabalho, assim como se apresenta propostas para futuros trabalhos.

2 Revisão bibliográfica

Este capítulo é dedicado ao estudo de sistemas em SS e métodos avançados de controle. Começa com uma introdução à teoria do controle moderno. Em seguida, explora-se o conceito de espaço de estados e abordam-se métodos para projetar controladores que assegurem o seguimento de referência, contanto que certas condições de controlabilidade sejam atendidas. Discute-se também a aplicação de observadores de estado e os critérios necessários para sua efetiva implementação. Além disso, o capítulo trata do desenvolvimento de controladores para sistemas não lineares e os critérios que devem ser observados para sua implementação completa.

2.1 Teoria de Controle Moderno

Para o problema do pêndulo invertido, uma das possíveis abordagens utilizadas é trabalhar com a teoria em SS. Justifica-se a escolha por pelo menos dois motivos: (i) as técnicas de realimentação dos estados são bem mais sofisticadas e eficientes do que as técnicas de controle clássico e; (ii) a realização do controle de sistemas dinâmicos no espaço de estados ocorre por meio da realimentação dos próprios estados do sistema. São apresentadas, a seguir, algumas definições importantes abordadas por Ogata (2010) para o pleno entendimento do SS.

Definição 2.1 (Estado) *O estado de um sistema dinâmico é definido pelo menor conjunto de variáveis, conhecidas como variáveis de estado. Sabendo o valor dessas variáveis em um instante t_0 e a entrada para $t \geq t_0$, é possível determinar completamente o comportamento do sistema para qualquer momento $t \geq t_0$. É importante notar que o conceito de estado não se aplica apenas a sistemas físicos, mas também a sistemas biológicos, econômicos, sociais e outros.*

Definição 2.2 (Vetor de Estado) *O comportamento do sistema depende de n variáveis de estado que, quando combinadas em uma estrutura vetorial, formam o chamado vetor de estados. Um vetor de estados define o estado do sistema $x(t)$ de forma única para qualquer momento $t \geq t_0$, desde que se conheça o estado inicial em $t = t_0$ e a entrada $u(t)$ para $t \geq t_0$.*

Definição 2.3 (Espaço de estados) *O espaço n-dimensional, cujos eixos coordenados são formados pelos eixos $x_1, x_2, \cdots, x_n$, em que x_1, x_2, $\cdots$, x_n são as variáveis de estado, é denominado espaço de estados. Qualquer estado pode ser representado por um ponto no espaço de estados.*

A análise em SS abrange três tipos de variáveis utilizadas na modelagem de sistemas dinâmicos: variáveis de entrada, variáveis de saída e variáveis de estado. Embora a representação de um sistema no SS não seja única, o número de variáveis de estado permanece constante em todas as diferentes representações desse sistema no SS.

Nos sistemas de controle de tempo contínuo, os integradores são comumente utilizados como dispositivos de memória. As saídas desses integradores podem ser consideradas variáveis que definem o estado interno do sistema dinâmico. Dessa forma, as saídas dos integradores podem ser escolhidas como as variáveis de estado. O número de variáveis de estado necessário para descrever completamente a dinâmica de um sistema é igual ao número de integradores presentes no sistema.

Suponha-se um sistema MIMO com n integradores envolvidos. Consideram-se também r entradas, definidas como $u_1, u_2, \cdots, u_r$, e m saídas, dadas por $y_1, y_2, \cdots, y_m$. Definem-se as n saídas dos integradores como sendo as variáveis de estado, $x_1, x_2, \cdots, x_n$. Portanto, o sistema pode ser descrito como

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1,x_2,\cdots,x_n;u_1,u_2,\cdots,u_r;t),\\ \dot{x}_2(t) &= f_2(x_1,x_2,\cdots,x_n;u_1,u_2,\cdots,u_r;t),\\ &\vdots\\ \dot{x}_n(t) &= f_n(x_1,x_2,\cdots,x_n;u_1,u_2,\cdots,u_r;t), \end{aligned} \tag{2.1}$$

enquanto as saídas do sistema são dadas por

$$\begin{aligned} y_1(t) &= g_1(x_1,x_2,\cdots,x_n;u_1,u_2,\cdots,u_r;t),\\ y_2(t) &= g_2(x_1,x_2,\cdots,x_n;u_1,u_2,\cdots,u_r;t),\\ &\vdots\\ y_m(t) &= g_m(x_1,x_2,\cdots,x_n;u_1,u_2,\cdots,u_r;t). \end{aligned} \tag{2.2}$$

Pode-se reescrever (2.1) e (2.2) na forma matricial

$$\begin{aligned} \dot{x}(t) &= f(x,u,t),\\ y(t) &= g(x,u,t), \end{aligned} \tag{2.3}$$

em que (2.3) são chamadas, respectivamente, de equação de estado e equação de saída. Por simplicidade, neste texto, a partir de (2.3), o contador temporal t será omitido. Considerando que o sistema é invariante no tempo, o vetor de estados pode ser escrito como

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T. \tag{2.4}$$

Se o conjunto de equações (2.3) forem lineares ou então forem linearizadas ao redor de um ponto de operação usando, por exemplo, série de Taylor, e retendo os termos lineares, pode-se reescrevê-las como

$$\begin{aligned} \dot{x} &= Ax + Bu,\\ y &= Cx + Du, \end{aligned} \tag{2.5}$$

em que as matrizes $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{m\times n}$ e $D \in \mathbb{R}^{m\times r}$ são nomeadas, respectivamente, como matriz de estado, de entradas, de saídas e de transmissão direta.

A Figura 3 exemplifica um sistema em espaço de estados, além de indicar a interrelação entre os sinais presentes no sistema dinâmico. Note ainda que para o sistema da Figura 3, as matrizes podem ser variantes no tempo, indicando a versatilidade da abordagem em espaço de estados.

2.2 Análise de estabilidade

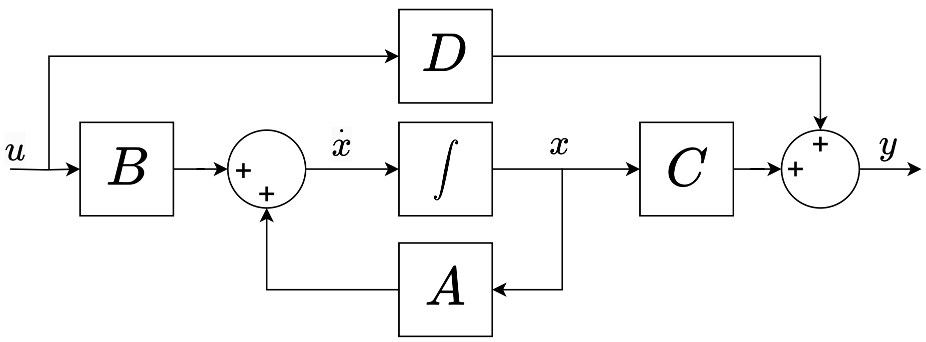
Para analisar a estabilidade de um sistema linear, pode-se verificar a localização dos polos. De acordo com Peaucelle et al. (2000), os polos de um sistema em SS correspondem aos valores próprios da matriz A de (2.5), cuja definição é apresentado a seguir.

Definição 2.4 (Autovalores) *A forma de se calcular os autovalores de uma matriz A de ordem n é a partir de*

$$|\lambda I - A| = 0, \tag{2.6}$$

conhecida como polinômio característico, em que o sinal $|\cdot|$ indica o procedimento de obter o determinante da matriz $\lambda I - A$. As raízes do polinômio característico correspondem aos autovalores de A.

Figura 3 – Diagrama de blocos de um sistema linear representado por espaço de estados.



Fonte: Autoria própria

2.2.1 Análise de erro em regime permanente

Erro em estado estacionário, e_{ss}, é caracterizado como a disparidade entre a entrada e a saída correspondente a uma entrada de teste predeterminada à medida que $t \to \infty$ (NISE, 2017). Pode ser calculado para entradas em degrau, rampa, parábola, entre outros, sendo o mais comum a resposta ao degrau unitário. Um método para se calcular o valor em regime permanente de um sistema modelado por SS é definido por

$$e_{ss} = \psi + CA^{-1}B, \tag{2.7}$$

em que a matriz $\psi \in \mathbb{R}^{m \times r}$ possui todos os elementos iguais a 1. A ordem da resultante de $CA^{-1}B$ tambem é $m \times r$. Nise (2017) explica que é uma boa alternativa empregar (2.7) a converter o SS em FT e, em seguida, calcular o erro.

2.3 Projeto de controlador em Espaço de Estados

Um sistema de ordem n possui um polinômio característico de ordem n, expresso por

$$q(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = 0. \tag{2.8}$$

O polinômio (2.8) possui n raízes, as quais determinam a localização dos polos em malha fechada do sistema. Logo, caso a condição de controlabilidade, abordada a seguir, seja satisfeita, é possível introduzir e relacionar n parâmetros em (2.8) e, dessa forma, ajustar os polos nas posições desejadas (NISE, 2017).

Definição 2.5 (Controlabilidade) *Um sistema é dito controlável caso exista um sinal de controle u que leve o estado inicial $x(0)$ para qualquer estado futuro x sem nenhum tipo de restrição (DORF; BISHOP, 2018). Considerando o sistema descrito por (2.5), pode-se montar uma matriz denominada matriz de controlabilidade*

$$P_c = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}. \tag{2.9}$$

Caso o posto de (2.9) seja igual à ordem n da matriz A, diz-se que o sistema é controlável.

A entrada de controle u, descrita em (2.5) e ilustrada pela Figura 3, também denominada esforço de controle, representa a quantidade de energia necessária para se efetuar o controle do sistema. Pode-se aproveitá-la para alterar os autovalores da matriz A. Dessa forma, escolhe-se uma entrada definida como

$$u = -Kx, \tag{2.10}$$

sendo x o vetor de estados descrito em (2.4) e K uma matriz de ganhos descrita como

$$K = \begin{bmatrix} k_1 & k_2 & \cdots & k_n \end{bmatrix}.$$

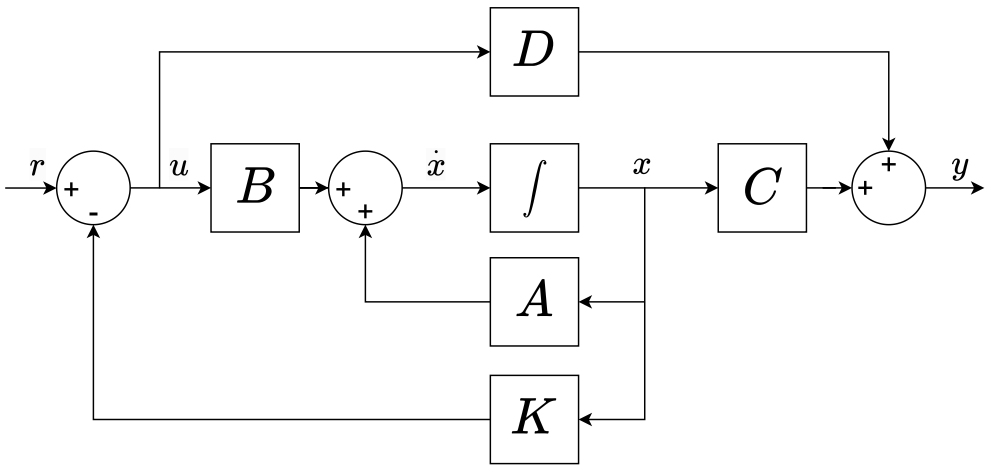
Portanto, u é descrito, alternativamente, como

$$u = -k_1x_1 - k_2x_2 - \cdots - k_nx_n. \tag{2.11}$$

Ao substituir (2.10) em (2.5), obtém-se

$$\begin{aligned} \dot{x} &= (A - BK)x \\ y &= (C - DK)x. \end{aligned} \tag{2.12}$$

Figura 4 – Diagrama de blocos de um sistema linear representado por espaço de estados com a inserção de um controlador.



Fonte: Autoria própria.

Dessa forma, é possível alocar, ou controlar, os autovalores da matriz A em qualquer posição desejada. A Figura 4 ilustra como fica o diagrama de blocos com o controlador implementado e uma entrada r.

Ogata (2010) explica que, um procedimento prático para alocar os polos do sistema em malha fechada na localização desejada é conhecido como fórmula de Ackerman. Com os polos devidamente escolhidos, monta-se o polinômio alocador característico $q(s)$

$$q(s) = (s - p_1)(s - p_2)\cdots(s - p_n). \tag{2.13}$$

O ganho K alocador de polos é obtido utilizando

$$K = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} P_c^{-1} q(A), \tag{2.14}$$

em que P_c é dado por (2.9) e $q(A)$ é o polinômio matricial obtido pela avaliação da matriz A em (2.13). A expressão (2.14) é conhecida como fórmula de Ackerman. Importante ressaltar que só é possível usar essa estratégia para sistemas com somente uma única entrada.

A escolha dos polos, pode ser feita de forma arbitrária ou então usando especificações de desempenho para a determinação deles. Na subseção seguinte, será discutido os critérios de desempenho mais comuns que orientam a geração de polos adequados a estas condições.

2.3.1 Especificações de desempenho

Pires (2023) explica que na concepção de controladores, é fundamental verificar se a resposta do sistema está em conformidade com as especificações de desempenho estabelecidas. Para excitar o sistema e observar sua resposta, empregam-se sinais de entrada variados, como um degrau, uma rampa, uma parábola, um impulso ou até mesmo ruído branco.

As especificações de desempenho são frequentemente expressas por meio de parâmetros de resposta transitória, tais como o sobressinal máximo, o tempo de acomodação e, para a resposta em regime estacionário, o erro em regime permanente. Adicionalmente, as especificações também podem ser descritas utilizando-se parâmetros de resposta em frequência (OGATA, 2010).

Têm-se a definição das principais especificações utilizadas no projeto de controladores:

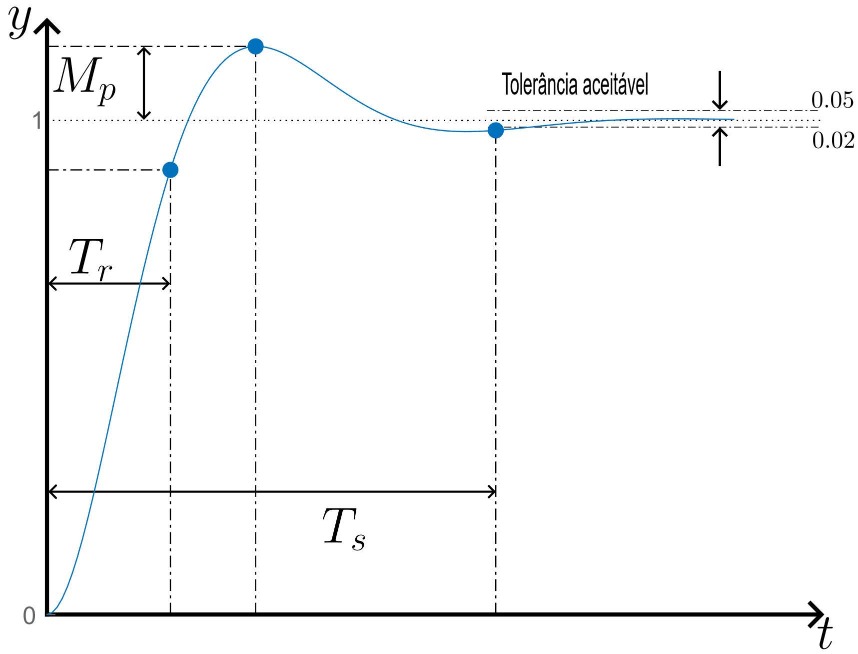
- Máximo sobressinal é o valor obtido a partir da unidade até o pico máximo da resposta, quando o sistema é excitado com um degrau unitário. Quando medido em porcentagem é definido como Máxima Ultrapassagem Percentual (MUP).
- Tempo de assentamento (T_s, do inglês *Settling Time*), é o tempo que a resposta alcança e permanece em uma determinada faixa do valor final. Geralmente, é entorno de $\pm 2\%$ ou $\pm 5\%$, sendo medido em segundos.
- Tempo de subida (T_r, do inglês *Rise Time*), é o tempo necessário para que a resposta passe pelo período compreendido entre 10% e 90% do valor final. É medido em segundos.
- Tempo de pico (T_p, do inglês *Peak Time*), é o tempo necessário para que a resposta atinja o primeiro valor máximo de sobressinal. É medido em segundos.

A Figura 5 ilustra, de forma gráfica, como são as especificações de controle mais utilizadas em projetos.

2.4 Projeto de controlador para seguimento de referência

Um controlador do tipo PI, também conhecido como servossistema do tipo 1 é um sistema que, baseado em sinais de realimentação, realiza correções podendo zerar o erro do sistema. Tal sistema é formado por vários dispositivos que verifica de forma contínua a informação atual, faz a comparação com a saída e efetua as correções necessárias para diminuir a diferença entre a referência e o sinal de saída (DINIZ et al., 2009). Caso a planta não apresente um integrador, ou seja, for classificada como uma planta de tipo 0, o fundamento principal para o projeto de um servossistema de tipo 1 envolverá a adição de um integrador no ramo direto, localizado entre o comparador e a própria planta (CARRARO, 2023). Conforme Ogata (2010) explica, a vantagem de se ter uma ação de controle integral no sistema é eliminar qualquer sinal de erro de regulação residual, desde que se tenha estabilidade no sistema. Com o controlador devidamente implementado, é possível

Figura 5 – Exemplo gráfico de especificações de desempenho de um sistema de ordem 2 à entrada degrau unitário.

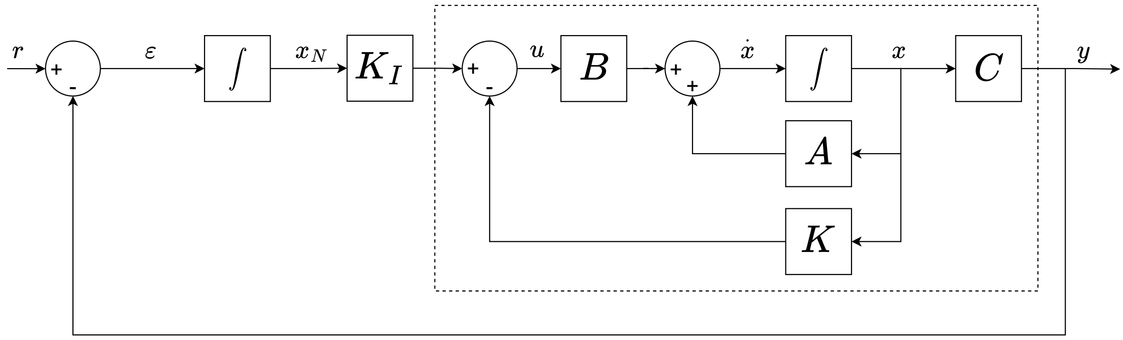


Fonte: Autoria própria.

tornar nulo o erro de regulação. A Figura 6 exemplifica como o PI pode ser implementado em um sistema descrito em SS, considerando que exista apenas uma entrada r.

Como pode ser observado na Figura 6, a parte pontilhada equivale ao controlador estabilizante projetado. Também é possível notar que se têm uma realimentação utilizada para calcular o erro ε, o qual é realimentado no sistema por um processo controlado com o uso de um integrador. Devido ao integrador, o tipo do sistema aumenta (NISE, 2017).

Figura 6 – Diagrama de blocos de um sistema linear com a inserção de um controlador e de um PI.



Fonte: Autoria própria.

A partir da análise da Figura 6, têm-se

$$
\begin{aligned}
\dot{x} &= Ax + Bu, \\
\dot{x}_N &= -Cx + r, \\
y &= Cx,
\end{aligned}
\tag{2.15}
$$

com r sendo definida como a referência desejada. Pode-se escrever (2.15) em forma de matrizes aumentadas, como

$$
\begin{aligned}
\dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r, \\
y &= \hat{C}\hat{x},
\end{aligned}
\tag{2.16}
$$

em que

$$
\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} C & 0 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} \dot{x} \\ \dot{x}_N \end{bmatrix}, \tag{2.17}
$$

enquanto o sinal de controle u é reescrito como

$$
u = -\hat{K}\hat{x}, \tag{2.18}
$$

em que $\hat{K}$ é a matriz de ganhos descrita como

$$
\hat{K} = \begin{bmatrix} K & -K_I \end{bmatrix}, \tag{2.19}
$$

com K_I sendo o ganho integral do controlador PI e K o ganho do controlador estabilizante descrito em (2.3). Efetuando a substituição de (2.18) em (2.16), obtém-se

$$
\begin{aligned}
\dot{\hat{x}} &= \begin{bmatrix} A - BK & BK_I \\ -C & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r, \\
y &= \begin{bmatrix} C & 0 \end{bmatrix} \hat{x}.
\end{aligned}
\tag{2.20}
$$

Empregando a fórmula de Ackerman (2.14), usando as matrizes $\hat{A}$, $\hat{B}$ e um conjunto adequado de polos, é possível projetar tanto o ganho K_I como a matriz de ganhos K.

2.5 Observadores de estados

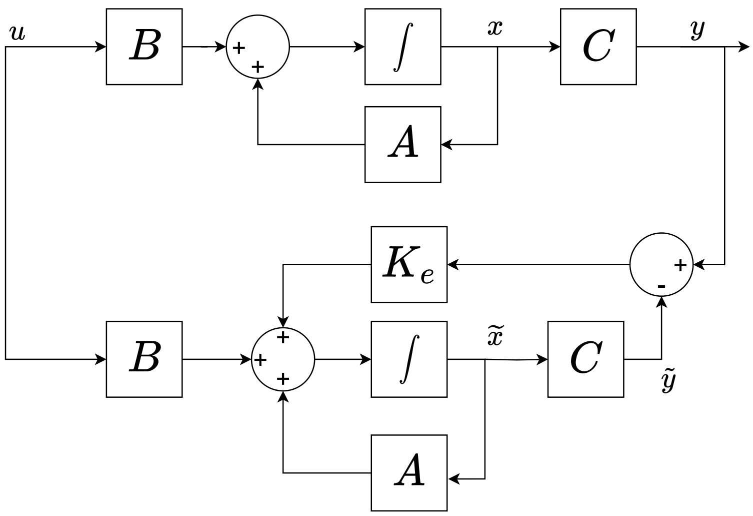
A estimativa de variáveis de estado que não são mensuráveis é denominado observação e o dispositivo que as estima é denominado observador de estado ou observador de Luenberg. Na prática, o estimador é um sistema secundário que reconstrói o vetor de estado da planta. Para verificar se é possível implementar o observador de Luenberg, é necessário que seja satisfeita a condição de observabilidade, a qual é abordada a seguir.

Definição 2.6 (Observabilidade) *Um sistema é dito observável caso exista um tempo específico T_f, o qual é finito, de forma que o estado inicial $x(0)$ seja determinado por meio* *do histórico de y dado um certo sinal de controle u. Seja um sistema descrito por (2.5), diz-se que o sistema é observável quando o posto da matriz Q, descrita a seguir,*

$$Q = \begin{bmatrix} C^T & (CA)^T & (CA^2)^T & \cdots & (CA^{n-1})^T \end{bmatrix}^T \tag{2.21}$$

é igual à ordem da matriz A.

Figura 7 – Exemplo de um sistema de controle com observador de estado.



Fonte: Autoria própria.

A Figura 7 exemplifica um sistema com observador de estados implementado e serve de base para o desenvolvimento do modelo matemático do estimador de Luenberg. A descrição matemática dos estados do observador é definida como

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + Bu + K_e(y - C\tilde{x}), \\ &= (A - K_eC)\tilde{x} + Bu + K_ey, \end{aligned} \tag{2.22}$$

em que $\tilde{x}$ são os estados observados, K_e é a matriz de ganhos do estimador e $C\tilde{x}$ são as saídas estimadas.

Para obter o erro, o qual é definido como a diferença entre os estados reais e observados, deve-se subtrair (2.22) da equação de estados descrita em (2.5), ou ainda

$$\begin{aligned} \dot{x} - \dot{\tilde{x}} &= Ax - A\tilde{x} - K_e(Cx - C\tilde{x}), \\ &= (A - K_eC)(x - \tilde{x}). \end{aligned} \tag{2.23}$$

Definindo o vetor de erro e como sendo

$$e = x - \tilde{x}. \tag{2.24}$$

Substituindo a expressão definida em (2.24) em (2.23) obtêm-se

$$\dot{e} = (A - K_eC)e. \tag{2.25}$$

De acordo com Ogata (2010), analisando (2.25), percebe-se que os autovalores da matriz $A - K_eC$ definem o comportamento dinâmico do vetor de erro. Uma matriz $A - K_eC$ estável garante que o vetor de erro se aproxime de zero a partir de qualquer condição inicial $e(0)$. Isso implica que $\tilde{x}$ irá convergir para x, não importando os valores iniciais $x(0)$ e $\tilde{x}(0)$. Portanto, se os autovalores de $A - K_eC$ forem selecionados para proporcionar estabilidade assintótica e uma taxa de convergência rápida, o vetor de erro tenderá para zero de forma eficiente. Com isso, se for possível observar completamente a planta, então se demonstra que é viável selecionar a matriz K_e de modo que $A-K_eC$ possa ter seus autovalores definidos de forma arbitrária. Isso implica que a matriz de ganho K_e do observador pode ser ajustada para alcançar a configuração desejada de $A - K_eC$.

2.6 Controle com observador de estados

Conhecido como problema dual, o desafio de criar um observador de ordem completa envolve definir a matriz de ganho K_e de tal forma que as dinâmicas do erro, conforme descritas em (2.25), sejam estabilizadas assintoticamente e com uma taxa de resposta adequada. A estabilidade assintótica e a rapidez da resposta do erro são influenciadas pelos valores próprios da matriz $A - K_eC$. Portanto, o projeto do observador de ordem completa se concentra em encontrar um K_e que resulte em valores próprios desejáveis para $A - K_eC$, o que equivale ao problema de alocação de polos discutido na Definição 2.5. Do ponto de vista matemático, é o mesmo problema. A Figura 8 ilustra com o auxílio de um diagrama de blocos como que o controle é feito com o observador implementado. Nota-se que o controle estabilizante é feito com os estados observados.

Considerando um sistema em SS descrito em (2.5), no qual as condições de controlabilidade e observabilidade são satisfeitas e ilustrado pela Figura 8, a lei de controle formada a partir dos estados observados pode ser expressa como

$$u = -K\tilde{x}. \tag{2.26}$$

Usando a lei de controle descrita em (2.26) na equação de estado descrita em (2.5) obtém-se

$$\begin{aligned} \dot{x} &= Ax - BK\tilde{x}, \\ &= (A - BK)x + BK(x - \tilde{x}). \end{aligned} \tag{2.27}$$

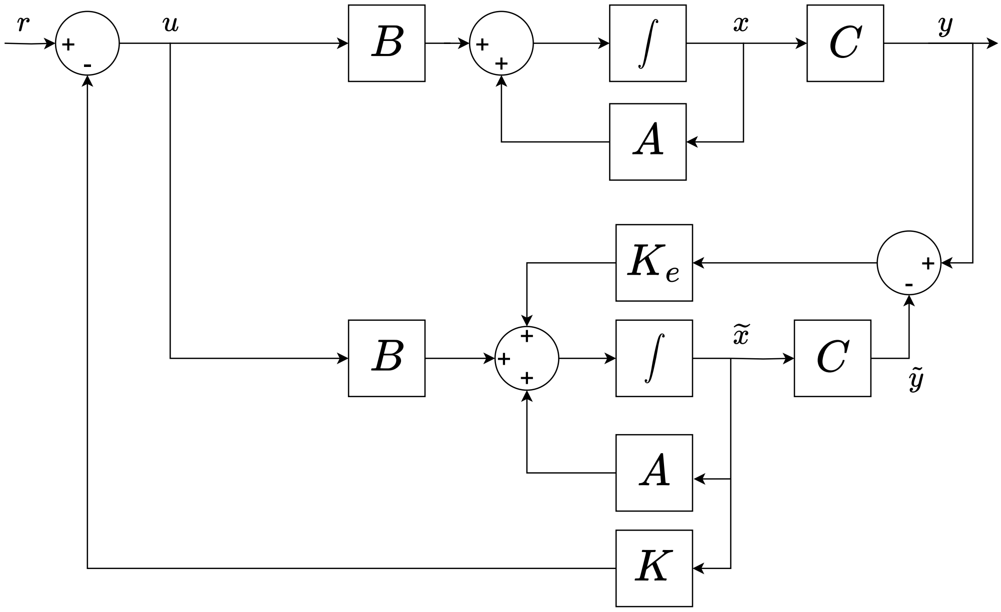
Substituindo o erro descrito em (2.24) em (2.27) obtém-se

$$\dot{x} = (A - BK)x + BKe. \tag{2.28}$$

Portanto, combinando o erro descrito em (2.25) e os estados descritos em (2.28) obtêm-se

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - K_eC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}. \tag{2.29}$$

Figura 8 – Exemplo de um sistema de controle com controlador estabilizante e o observador de estados implementado.



Fonte: Autoria própria.

Calculando os autovalores da matriz de estado de (2.29) obtêm-se o polinômio característico descrito a seguir

$$p(\lambda) = |sI - A + BK||sI - A + K_eC| = 0. \tag{2.30}$$

Analisando (2.30), nota-se que é possível escolher um conjunto de ganhos K para estabilizar o sistema e uma matriz de ganhos K_e para o observador separadamente. Esse princípio, conhecido como princípio de Separação de Luenberger, apresentado no Teorema 2.1, é muito importante na concepção de controladores e observadores. A prova do teorema encontra-se disponível em Chen (2014).

Teorema 2.1 (Princípio de Separação de Luenberger) *Pela análise de* (2.30)*, é possível observar que os ganhos K e K_e podem ser calculados de forma independente. O conceito se chama Princípio de Separação de Luenberger, ou seja, os problemas de se projetar um estimador de estados e um controlador podem ser feitos separadamente.*

Daltin (2017) explica que se o controlador e observador forem estáveis, o conjunto sistema-observador-controlador também será estável. O cálculo de K_e pode ser feito usando a fórmula de Ackerman (2.14).

2.7 Projeto de controladores MIMO

Uma forma de se calcular os ganhos é descrito por Zak et al. (2003). Considera-se que o sistema descrito por (2.5) é completamente controlável e observável. Seja uma matriz de ganhos $K \in \mathbb{R}^{r \times n}$ definida a seguir

$$K = \begin{bmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \vdots & \vdots \\ k_{r1} & \cdots & k_{rn} \end{bmatrix}, \tag{2.31}$$

e um sinal de controle do tipo

$$u = -Kx, \tag{2.32}$$

em que x é o vetor de estados descrito em (2.4).

Substituindo a entrada definida em (2.32) na equação de estado descrita em (2.5) obtêm-se

$$\dot{x} = (A - BK)x, \tag{2.33}$$

em que seus valores próprios são definidos pela seguinte equação característica

$$|sI - A + BK| = 0. \tag{2.34}$$

Os polos, previamente escolhidos pelo projetista, são apresentados a seguir

$$P = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}. \tag{2.35}$$

Portanto, o projeto por alocação de polos envolve determinar a matriz de ganhos K tal que os autovalores do sistema em malha fechada, $A - BK$, sejam iguais a P. Conforme Ogata (2010), a matriz descrita em (2.31) não é única, podendo ser escolhida de forma a maximizar a margem de estabilidade, por exemplo, ou outro critério que o engenheiro julgue ser relevante para seu projeto.

2.8 Técnicas de controle não lineares

Conforme Slotine e Li (1991) comenta, não existem um método generalizado para desenvolver controladores não lineares. Considerando essa limitação, técnicas complementares e alternativas podem ser aplicadas em cada caso em particular.

Algumas técnicas são: linearização por série de Taylor (GADELHA, 2018), controle adaptativo (YANG; LI; LI, 2013), controle robusto (BUZETTI, 2017), linearização por realimentação (CADENGUE et al., 2020), dentre outras abordagens. As técnicas abordadas neste trabalho serão: linearização entrada-saída e linearização por série de Taylor. Para o pleno entendimento das técnicas de controle não lineares, abordam-se algumas definições matemáticas, apresentadas a seguir, as quais foram citadas em Silva (2006).

Definição 2.7 (Gradiente) *É definido como gradiente o vetor de derivadas parciais de uma função escalar $h(x_1,x_2,\cdots,x_n)$ derivadas em relação às variáveis $x_1,x_2,\cdots,x_n$. É representado da seguinte forma*

$$\nabla h(x) = \left(\frac{\partial h(x)}{\partial x_1}, \frac{\partial h(x)}{\partial x_2}, \cdots, \frac{\partial h(x)}{\partial x_n}\right). \tag{2.36}$$

Definição 2.8 (Jacobiano) *É definido como jacobiano J de um conjunto de funções $f(x_1,x_2,\cdots,x_n)$ uma matriz de derivadas parciais. É representado da seguinte forma*

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \tag{2.37}$$

Definição 2.9 (Derivada de Lie) *Considere uma função escalar suave $h : \mathbb{R}^n \to \mathbb{R}$ e um conjunto de funções suave $f : \mathbb{R}^n \to \mathbb{R}^n$. A derivada de Lie da função h em relação ao campo vetorial f é descrito como*

$$L_f h = \nabla h f. \tag{2.38}$$

É possível definir derivadas de Lie de ordem superior a 1 de forma recursiva, sendo definidas como

$$\begin{aligned} L_f^0 h &= h \\ L_f^i h &= L_f(L_f^{i-1} h) = \nabla(L_f^{i-1} h) f \end{aligned} \tag{2.39}$$

É possível também calcular derivadas de campos vetoriais. Sejam dois campos vetoriais definidos como sendo f e g. Define-se $L_g L_f h$ como sendo

$$L_g L_f h = \nabla(L_f h) g \tag{2.40}$$

2.8.1 Linearização Jacobiana

Uma forma de estudar o comportamento de um sistema não linear em torno de um ponto de equilíbrio é aplicar a linearização Jacobiana, que consiste em usar a série de Taylor para obter uma aproximação linear do sistema original. Com isso, é possível analisar as propriedades do ponto de equilíbrio, como estabilidade, polos, dentre outras, usando as ferramentas da teoria de sistemas lineares (BOYCE; DIPRIMA; IÓRIO, 2002).

Considera-se novamente um sistema não linear descrito em SS apresentado a seguir

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases}, \tag{2.41}$$

em que f, g e h contém termos não lineares, sendo definidas como

$$\begin{aligned} f &= \begin{bmatrix} f_1 & f_2 & \cdots & f_n \end{bmatrix}^T, \\ g &= \begin{bmatrix} g_1 & g_2 & \cdots & g_n \end{bmatrix}^T, \\ h &= \begin{bmatrix} h_1 & h_2 & \cdots & h_m \end{bmatrix}^T. \end{aligned}$$

Aplicando o jacobiano na equação de estado descrita em (2.41) e escolhendo convenientemente um ponto de equilíbrio adequado, denominado x_0, obtêm-se as matrizes A e B presentes em (2.5), isto é

$$A = \left. \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \right|_{x=x_0}, \quad B = g(x_0). \tag{2.42}$$

2.8.2 Linearização entrada-saída

O intuito principal desse procedimento é empregar a transformação dos estados e da variável de controle para modificar a dinâmica não linear em uma dinâmica aproximadamente linear. Assim, os elementos não lineares que sobrarem poderão ser eliminados pela realimentação. Com a aplicação da linearização entrada-saída, é possível utilizar os princípios da teoria de controle linear para implementar o controle desejado (NASCIMENTO, 2009).

Considerando um sistema não linear descrito de forma genérica em (2.41). O grau relativo R de um sistema é determinado pelo número de diferenciações necessárias para que a entrada u seja evidenciada. Essa definição é uma extensão do conceito de grau relativo aplicado a sistemas lineares. Observa-se que, para possibilitar a linearização com realimentação entrada-saída, o sistema deve ter um grau relativo R menor que a ordem n do sistema. Caso R seja igual a n, o processo de linearização entrada-saída corresponde, de fato, a uma linearização entrada-estado (SLOTINE; LI, 1991). Com a entrada u evidenciada, deve-se aplicar o seguinte sinal de controle

$$u = \frac{1}{L_g L_f^{R-1} h}(-L_f^R h + v), \tag{2.43}$$

em que v é a nova entrada de controle e h é a saída do sistema, caso tenha mais de uma entrada. Com tal transformação aplicada em (2.41), o sistema será levado a uma relação linear da forma

$$y^{(R)} = v. \tag{2.44}$$

O próximo passo é escolher o sinal de controle v da forma

$$v = k_1 x_1 + k_2 x_2 + \cdots + k_n x_n, \tag{2.45}$$

o qual estabilizará o sistema.

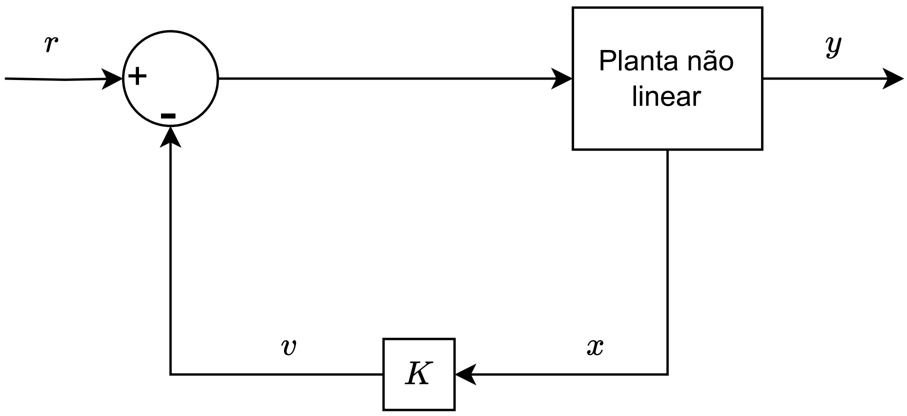
É muito importante $R \leqslant n$. É dito que o sistema descrito por (2.41) tem grau relativo R quando as seguintes condições são atendidas:

- $L_g L_f^i = 0, 0 \leqslant i \leqslant R-1;$

- $L_g L_f^{R-1} \neq 0$.

Ainda, de acordo com Silva (2006), se por acaso a saída for derivada n vezes e ainda assim não aparecer uma relação explícita entre a saída e a entrada, diz-se que o sistema não é controlável. Caso contrário, o sistema é controlável. A Figura 9 exemplifica um sistema de controle não linear com o controlador implementado.

Figura 9 – Exemplo de um sistema de controle não linear com controlador implementado.

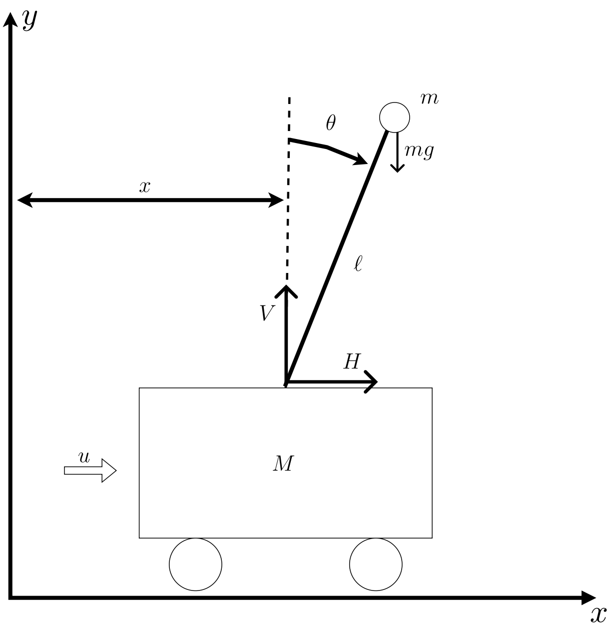


Fonte: Autoria própria.

3 Modelagem matemática do pêndulo invertido

Este capítulo dedica-se a elaboração da representação matemática do pêndulo invertido, bem como sua descrição em SS. Para o caso desta monografia, será considerado que o sistema possa se movimentar apenas no plano da página. A Figura 10 ilustra bem o conjunto de forças que atuam no sistema. Conforme Burns (2001) explica, para fazer uma descrição matemática do pêndulo invertido, tal diagrama é essencial na análise de corpo livre. O modelo utilizado neste trabalho foi o utilizado por Gadelha (2018).

Figura 10 – Diagrama de corpo livre do pêndulo invertido.



Fonte: Autoria própria.

Primeiramente, é preciso definir o vetor de estados. Conforme Taborda (2014) explica, para esse caso, define-se vetor $x \in \mathbb{R}^4$. As variáveis são:

- Deslocamento angular da haste θ, dado em rad;
- Velocidade angular da haste $\dot{\theta}$, dado em rad/s;
- Deslocamento do carrinho x, dado em m;
- Velocidade do carrinho $\dot{x}$, dado em m/s.

Vetorialmente, os estados podem ser escritos como

$$x = \begin{bmatrix} \theta & \dot{\theta} & x & \dot{x} \end{bmatrix}^T. \tag{3.1}$$

Com relação ao vetor de saídas de interesse, será definido como $y \in \mathbb{R}^2$, as quais são:

- Deslocamento angular da haste θ;
- Deslocamento do carrinho x.

Vetorialmente, as saídas podem ser escritas como

$$y = \begin{bmatrix}\theta & x\end{bmatrix}^T \cdot \tag{3.2}$$

Com relação ao sinal de controle, define-se como sendo $u \in \mathbb{R}$, e representa a força dos motores presentes no carrinho.

As considerações usadas neste trabalho são apresentadas a seguir.

Consideração 3.1

- *O centro de gravidade da haste fica no centro geométrico do carrinho.*
- *O ângulo formado entre a haste e uma linha vertical que sai do centro do carrinho como θ.*
- *As coordenadas do centro de gravidade da haste podem ser escritas como*

$$\begin{aligned} x_G &= x + \ell sen\,\theta, \\ y_G &= \ell \cos\theta. \end{aligned} \tag{3.3}$$

As equações de movimento do sistema são deduzidas utilizando o diagrama de corpo livre mostrado ilustrado na Figura 10. A rotação da barra do pêndulo em torno de seu ponto de equilíbrio é descrita como

$$I\ddot{\theta} = V\ell \mathrm{sen}\,\theta - H\ell\cos\theta. \tag{3.4}$$

Nesse caso, I indica o momento de inércia da haste com relação ao centro de gravidade. Zak et al. (2003) explica que o momento de inércia de uma haste uniforme pode ser escrito como

$$I = \frac{m\ell^2}{3} \cdot \tag{3.5}$$

Os movimentos horizontal e vertical da haste do pêndulo no centro de gravidade, representados, respectivamente, por H e V são expressos por

$$\begin{aligned} H &= m\frac{\partial^2(x + \ell \mathrm{sen}\,\theta)}{\partial t^2}, \\ V &= mg + m\frac{\partial^2(\ell\cos\theta)}{\partial t^2} \cdot \end{aligned} \tag{3.6}$$

O movimento horizontal do carro é descrito como

$$M\frac{d^2x}{dt^2} = u - H. \tag{3.7}$$

As seguintes relações são utilizadas

$$\begin{aligned} \frac{\partial^2}{\partial x^2}(\cos\theta) &= -\dot{\theta}^2\cos(\theta) - \ddot{\theta}\text{sen}\,(\theta), \\ \frac{\partial^2}{\partial x^2}(\text{sen}\,\theta) &= -\dot{\theta}^2\text{sen}\,(\theta) + \ddot{\theta}\cos(\theta). \end{aligned} \tag{3.8}$$

Para obter $\ddot{x}$, deve-se substituir a expressão H presente em (3.6) na expressão (3.7) e utilizar as relações descritas em (3.8). Para obter $\ddot{\theta}$, deve-se substituir (3.6) em (3.4) e utilizar as relações descritas em (3.8). Feito isso, obtém-se o seguinte conjunto de equações

$$\begin{aligned} \frac{m\ell^2}{3}\ddot{\theta} &= -lm(\ddot{\theta}\ell + \ddot{x}cos(\theta) - gsin(\theta)), \\ M\,\ddot{x} &= u - \ddot{x}\,m + \ell\,m\,\left(\dot{\theta}^2\,\text{sen}\,\,(\theta) - \ddot{\theta}\,\cos\,(\theta)\right). \end{aligned} \tag{3.9}$$

Resolvendo o conjunto (3.9) nas variáveis $\ddot{x}$ e $\ddot{\theta}$, obtêm-se as seguintes expressões

$$\begin{aligned} \ddot{x} &= \frac{4\,u - \text{sen}\,\,(\theta)\,\left(3\,g\,m\,\cos\,(\theta) - 4\,\ell\,m\,\dot{\theta}^2\right)}{4\,M + \frac{5\,m}{2} + \frac{3\,m\,\left(2\,\text{sen}\,(\theta)^2 - 1\right)}{2}}, \\ \ddot{\theta} &= \frac{3\left(M g\text{sen}\,(\theta) + gm\text{sen}\,(\theta) - \frac{\ell m\dot{\theta}^2\text{sen}\,(2\theta)}{2} - u\cos\,(\theta)\right)}{\ell\,(4M + 3m\text{sen}\,^2(\theta) + m)}. \end{aligned} \tag{3.10}$$

O próximo passo é escrever (3.10) em um sistema de equações diferenciais de 1° ordem, conforme Definição 2.3, sendo as variáveis de estado

$$\begin{aligned} x_1 &= \theta, \\ x_2 &= \dot{\theta}, \\ x_3 &= x, \\ x_4 &= \dot{x}. \end{aligned} \tag{3.11}$$

A seguir, efetuando a aplicação das variáveis de estado definidas em (3.11) nas expressões de $\ddot{x}$ e $\ddot{\theta}$ isoladas em (3.10), e simplificando tais expressões, obtêm-se

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_2 + g_2 u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u \end{cases} \tag{3.12}$$

em que

$$\begin{aligned}
f_2 &= \frac{3\left(2Mg\text{sen}\left(x_1\right) + 2gm\text{sen}\left(x_1\right) - \ell m x_2^2 \text{sen}\left(2x_1\right)\right)}{2\ell\left(4M + 3m\text{sen}^2(x_1) + m\right)}, \\
f_4 &= \frac{-3gm\text{sen}\left(2x_1\right) + 8\ell m x_2^2 \text{sen}\left(x_1\right)}{2\left(4M + 3m\text{sen}^2(x_1) + m\right)}, \\
g_2 &= -\frac{3\cos\left(x_1\right)}{\ell\left(4M + 3m\text{sen}^2(x_1) + m\right)}, \\
g_4 &= \frac{4}{4M + 3m\text{sen}^2(x_1) + m}.
\end{aligned} \tag{3.13}$$

As saídas de interesse são o ângulo da haste e a posição do carrinho, as quais são os estados x_1 e x_3, ou ainda

$$y = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T. \tag{3.14}$$

Com (3.12)-(3.14) determinadas, o próximo passo é escrevê-las na forma genérica mostrada em (2.41)

$$\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = f_2 + g_2 u \\
\dot{x}_3 = x_4 \\
\dot{x}_4 = f_4 + g_4 u \\
y = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T
\end{cases} \tag{3.15}$$

em que f_2, f_4, g_2, g_4 são dadas por (3.13). Os valores de M, m, g, ℓ que serão utilizados neste trabalho estão disponíveis na Tabela 1. Tais parâmetros foram extraídos de Ogata (2010, Exemplo 10.5) e, além disso, escolheu-se um passo de integração d igual a 0.1 ms, o qual também é apresentado na Tabela 1. Em todas as simulações apresentadas neste trabalho, emprega-se o método de Runge Kutta, o qual é melhor explicado no Apêndice A. Por fim, o equacionamento algébrico do pêndulo invertido encontra-se disponível no Apêndice C.

Tabela 1 – Valores dos parâmetros do pêndulo invertido.

Parâmetro	Valor	Unidade
m	0,1	Kg
M	2	Kg
g	9,8	m/s^2
ℓ	0,5	m
d	0,1	ms

Fonte: Retirado de Ogata (2010).

4 Controle do modelo linearizado do pêndulo invertido

Neste capítulo, será abordada a linearização do modelo não linear descrito por (3.15), por meio da linearização jacobiana, na seção 4.1. A seguir, são projetados controladores estabilizante, na seção 4.2, para seguimento de referência, na seção 4.3. Por fim, na seção 4.4, projeta-se um observador de estados, o qual é empregado conjuntamente a um controlador PI. Todos os resultados apresentados empregaram passo de integração igual d a 0.1 ms, conforme mostrado na Tabela 1.

4.1 Simulação e linearização do modelo não linear do pêndulo invertido

De acordo com Taborda (2014), uma das forma de se fazer a linearização de um sistema, é determinar um ponto de equilíbrio. No caso do pêndulo invertido, conforme Alves (2018) explica, existem vários pontos, porém somente dois pontos são, de fato, relevantes, que são a posição para baixo, ou seja, $\theta = 180°$ e a posição para cima, ou seja, $\theta = 0°$. No caso deste trabalho, sera definido o ponto de equilíbrio como sendo

$$x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.1}$$

Aplicando a linearização jacobiana, discutida na subseção 2.8.1, em (3.15) e usando o ponto de equilíbrio (4.1), obtêm-se as matrizes

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{3\,(M\,g+g\,m)}{4\,M\,\ell+\ell\,m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{3\,g\,m}{4\,M+m} & 0 & 0 & 0 \end{bmatrix}, \quad & B &= \begin{bmatrix} 0 \\ -\frac{3}{4\,M\,\ell+\ell\,m} \\ 0 \\ \frac{4}{4\,M+m} \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, & D &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{4.2}$$

Substituindo os valores da Tabela 1 nas matrizes (4.2), têm-se

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 15{,}2444 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0{,}3630 & 0 & 0 & 0 \end{bmatrix}, \quad & B &= \begin{bmatrix} 0 \\ -0{,}7407 \\ 0 \\ 0{,}4938 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, & D &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{4.3}$$

Os autovalores da matriz A de (4.3), conforme discutido na seção 2.2, informam os polos do sistema descrito pelas matrizes (4.3). Usando o comando *eigvals*, do *python*, obtêm-se os valores próprios

$$\lambda = \begin{bmatrix} 0 & 0 & P & -P \end{bmatrix}, \tag{4.4}$$

em que

$$P = \sqrt{\frac{3\,g\,(M+m)}{\ell\,(4\,M+m)}} = 3{,}9044.$$

Pela análise dos autovalores em (4.4), nota-se a presença de quatro polos: dois na origem, um no semieixo negativo e outro no semieixo positivo. Portanto, o sistema é instável.

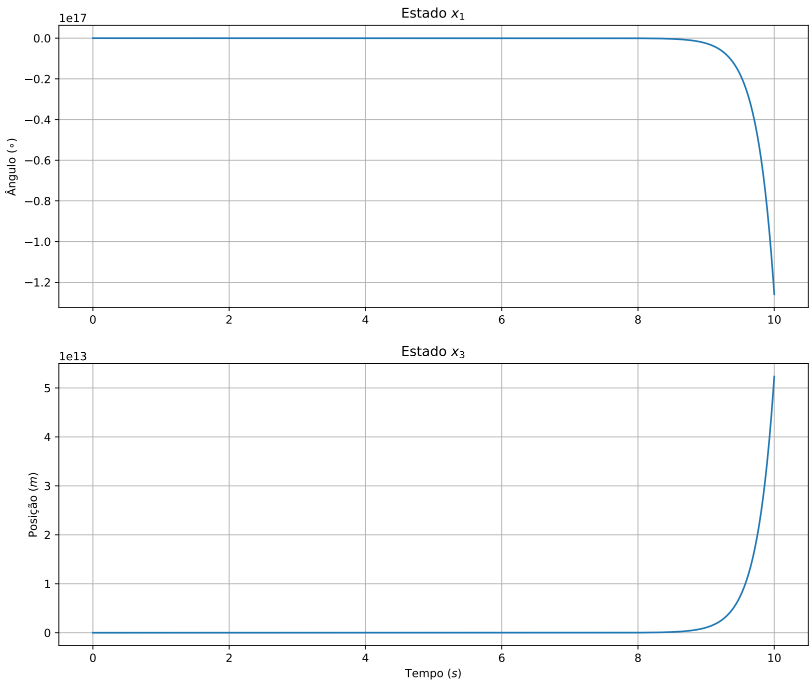
Todas as simulações desenvolvidas neste trabalho, com exceção do observador de estados, foram feitas em *Python*. A linguagem de programação *Python* é dinâmica e orientada a objetos, a qual pode ser utilizada para desenvolver qualquer aplicação, seja científica ou não (COELHO, 2007). A IDE utilizada será o *Google Colab*.

A escolha por trás da escolha do *Colab* é enumerada a seguir:

- Aceleração de GPU gratuita;
- Bibliotecas pré-instaladas: Todas as principais bibliotecas *Python*, como *scipy*, *numpy*, *Matplotlib*, entre outras, estão pré-instaladas e prontas para uso;
- Funciona de maneira semelhante ao *Google Docs*, permitindo que desenvolvedores usem e compartilhem notebooks entre si sem precisar baixar, instalar ou executar qualquer coisa além de um navegador;
- O *Google Colab* permite dividir blocos de texto e código no mesmo arquivo, facilitando comentários, compilação modularizada, elaboração simultânea de relatórios, clareza e organização dos estudos;
- Os *notebooks* do *Google Colab* são armazenados no *Google Drive*, permitindo assim acesso em qualquer máquina, inclusive compartilhar com outros usuários.

Em um primeiro momento, será analisada a resposta ao degrau unitário do sistema descrito pelas matrizes (4.3), com condições iniciais nulas, ou seja, $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$. Ao examinar detalhadamente a Figura 11, fica evidente que a resposta do sistema sem nenhum tipo de controlador manifesta um crescimento infinito, o que visualmente confirma a instabilidade inerente ao sistema. Os dados da simulação numérica encontram-se disponíveis no Apêndice D.

Figura 11 – Resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) sem controladores.



Fonte: Autoria própria.

4.2 Projeto de controlador estabilizante

Como demonstrado por (4.4) e pela Figura 11, o sistema do pêndulo invertido é instável em malha aberta, necessitando da inserção de um controlador para torná-lo estável. Antes de iniciar o projeto, é necessário verificar se sistema descrito pelas matrizes (4.3) é controlável. Aplicando o conceito de controlabilidade, visto na Definição 2.5, nas matrizes A e B presentes em (4.3), obtém-se a matriz de controlabilidade dada por

$$P_c = \begin{bmatrix} 0 & -0{,}7407 & 0 & -11{,}2922 \\ -0{,}7407 & 0 & -11{,}2922 & 0 \\ 0 & 0{,}4938 & 0 & 0{,}2689 \\ 0{,}4938 & 0 & 0{,}2689 & 0 \end{bmatrix},$$

cujo posto vale 4, comprovando que o sistema é de fato controlável.

Portanto, deve-se efetuar a escolha de novos polos para o pêndulo invertido. No total, serão escolhidos 3 conjuntos. No caso específico do conjunto de polos 1, os polos complexos foram projetados para atender determinados requisitos de desempenho, escolhidos a partir de projetos anteriores. Estas especificações estão descritas a seguir

$$T_s = 2s, \quad MUP = 15\%. \tag{4.5}$$

Ogata (2010) explica as especificações (4.5) podem ser traduzidas em raízes dominantes

$$r_1,\, r_2 = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}, \tag{4.6}$$

em que

$$\begin{aligned} \zeta &= \sqrt{\frac{\log\left(\frac{MUP}{100}\right)^2}{\pi^2 + \log\left(\frac{MUP}{100}\right)^2}}, \\ \omega_n &= \frac{4}{\zeta T_s}. \end{aligned} \tag{4.7}$$

Portanto, aplicando o conjunto (4.6)-(4.7) nas especificações de desempenho descritas em (4.5), obtêm-se as raízes

$$r_1,r_2 = -2 \pm 3{,}3120j. \tag{4.8}$$

Os outros dois polos do conjunto 1 foram escolhidos arbitrariamente, de forma a ficarem distantes, no semiplano esquerdo s de (4.8). Com relação aos outros conjuntos de polos, foram escolhidos aleatoriamente, de forma a ficarem distantes da origem no semiplano esquerdo do plano s. Sendo, os três conjuntos de polos expressos como

$$P_1 = -\begin{bmatrix} 2 \pm 3{,}3120j & 6 & 8 \end{bmatrix}, \tag{4.9}$$

$$P_2 = -\begin{bmatrix} 3 & 4 & 5 & 6 \end{bmatrix}, \tag{4.10}$$

$$P_3 = -\begin{bmatrix} 6 & 8 & 10 & 12 \end{bmatrix}. \tag{4.11}$$

Com relação às matrizes de ganhos K_1, K_2, K_3 associadas aos polos descritos em (4.9)-(4.11), elas foram calculados a partir de (2.14), obtendo-se, K_1, K_2, K_3 são iguais a

$$\begin{aligned} K_1 &= -\begin{bmatrix} 247{,}1744 & 61{,}1786 & 98{,}9792 & 55{,}3179 \end{bmatrix}, \\ K_2 &= -\begin{bmatrix} 214{,}2912 & 55{,}7082 & 49{,}5918 & 47{,}1122 \end{bmatrix}, \\ K_3 &= -\begin{bmatrix} 1192{,}1596 & 299{,}8653 & 793{,}4694 & 5376{,}898 \end{bmatrix}. \end{aligned} \tag{4.12}$$

Tem-se os dados da resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com três controladores diferentes projetados com três conjuntos de polos diferentes. Para os três casos, foram consideradas condições iniciais nulas, ou seja, $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$. Os algoritmos desenvolvidos para as simulações numéricas encontram-se disponíveis no Apêndice E.

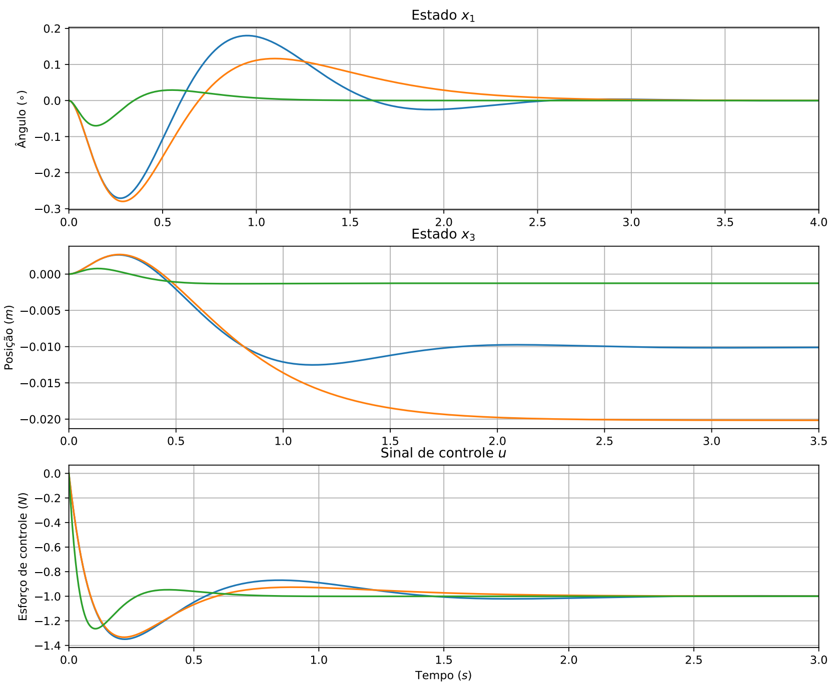
A partir da análise da Figura 12 e da Tabela 2, é possível observar que tanto T_s quanto o MUP se aproximam bastante do tempo designado tanto para o ângulo θ como a posição x, para o conjunto de polos P_1. Além disso, o valor em regime permanente do sistema se aproxima de zero para a segunda saída e se torna nulo para a primeira saída. Por outro lado, o esforço de controle u atinge um valor mínimo de $-0{,}8446$ em $0{,}8435s$.

Com referência ao conjunto de polos descrito em (4.10) e considerando o ângulo θ, ao examinar os dados apresentados na Figura 12 e na Tabela 2, é observável que a parte

Tabela 2 – Dados da resposta ao degrau unitário da posição da haste, estado x_1, e da posição do carrinho, estado x_3, do sistema descrito pelas matrizes (4.3) com o controlador estabilizante implementado, com condições iniciais $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$.

Conjunto de polos	Sinal	T_s (s)	MUP	T_r (s)	Valor final	Erro
P_1 (4.9)	x_1	2,4149	0	0	0°	1°
	x_3	2,3989	23,9513	0,3078	$-0{,}0101$ m	1,0101 m
P_2 (4.10)	x_1	2,6472	–	0	0°	1°
	x_3	1,9490	0	0,9231	$-0{,}0202$ m	1,0202 m
P_3 (4.11)	x_1	1,3236	0	0	0°	1°
	x_3	0,9821	5,2818	0,2164	$-0{,}0013$ m	1,0013 m

Figura 12 – Resposta ao degrau unitário e esforço de controle do sistema descrito pelas matrizes (4.3) com controlador estabilizante implementado, com os conjuntos P_1 (4.9) em azul, P_2 (4.10) em laranja e P_3 (4.11) em verde, degrau unitário em vermelho, com condições iniciais $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$.



Fonte: Autoria própria.

real dos polos está mais afastada do semieixo negativo. O T_s é aproximadamente 9,6194% maior em relação ao valor anterior, enquanto o MUP não se altera significativamente, permanecendo elevado. Com relação a posição x, nota-se que o T_s é reduzido em cerca de 16,2261% em relação ao valor anterior, ao passo que a MUP torna-se nulo.

No contexto do esforço de controle em magnitude, ele é ligeiramente superior ao do primeiro conjunto de polos, com valor de $-0{,}9015$ o qual ocorre em $0{,}8972s$. Isso sugere que é requerida uma quantidade maior de energia para conduzir o sistema ao estado estacionário em comparação com o caso anterior.

No que diz respeito ao conjunto de polos descrito em (4.11) e considerando o ângulo θ, ao analisar os dados representados na Figura 12 e na Tabela 2, verifica-se que a parte real das raízes está ainda mais afastada no semieixo negativo em comparação com o segundo conjunto. O T_s é reduzido em aproximadamente 50% em relação ao conjunto anterior, enquanto o MUP permanece significativamente elevado. No contexto da posição x, observa-se que o T_s é reduzido em 49,6101% em relação ao valor anterior, é consideravelmente menor. Além disso, o esforço de controle é ligeiramente superior, com valor de pico de $-0{,}9224$ e ocorre em $0{,}3964s$. Em outras palavras, é necessário ainda mais energia para alcançar o estado de equilíbrio.

Para todos os três cenários, é possível observar que o sistema converge de maneira estável, com mínimas flutuações, à medida que os polos se afastam da origem. À medida que os polos dominantes se deslocam cada vez mais para o semieixo negativo, as oscilações que ocorrem antes de atingir o estado permanente diminuem significativamente, até quase desaparecerem por completo. Por outro lado, o esforço de controle é aumentado com o deslocamento dos polos em malha fechada para valores cuja parte real é cada vez menores. Dentre os conjuntos (4.9)-(4.11), o que mais se aproxima das especificações descritas em (4.5) é o conjunto (4.11). Portanto, conclui-se que o objetivo de estabilizar o estado x_1 é satisfeito.

4.3 Projeto de controlador para seguimento de referência

Para a inserção do PI, é preciso analisar novamente os requisitos de controlabilidade, dado que as matrizes do sistema são alteradas. Como o objetivo é manter o ângulo θ com um valor nulo em regime permanente, não será implementado um seguimento de referência para o estado x_1, somente para a posição do carrinho, representada pelo estado x_3. Como será utilizado somente uma saída, no caso, o estado x_3, a matriz C de (4.3) será alterada, sendo escrita como

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4.13}$$

Com a nova matriz de saída definida, é preciso substituir as matrizes A e B de (4.3) e a matriz C definida em (4.13) em (2.17). Desta forma, as matrizes aumentadas podem ser escritas como

$$\hat{A} = \begin{bmatrix} 0 & 1{,}0 & 0 & 0 & 0 \\ 15{,}2444 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1{,}0 & 0 \\ -0{,}363 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1{,}0 & 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ -0{,}7407 \\ 0 \\ 0{,}4938 \\ 0 \end{bmatrix},$$
$$\hat{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.14}$$

Com as novas matrizes $\hat{A}$ e $\hat{B}$ definidas em (4.14), é preciso verificar se é controlável o sistema aumentado. Recorrendo a (2.9), a nova matriz de controlabilidade é descrita como

$$P_c = \begin{bmatrix} 0 & -0{,}7407 & 0 & -11{,}2922 & 0 \\ -0{,}7407 & 0 & -11{,}2922 & 0 & -172{,}143 \\ 0 & 0{,}4938 & 0 & 0{,}2689 & 0 \\ 0{,}4938 & 0 & 0{,}2689 & 0 & 4{,}0986 \\ 0 & 0 & -0{,}4938 & 0 & -0{,}2689 \end{bmatrix},$$

cujo posto é igual a 5. Como a ordem de $\hat{A}$ é igual a 5, conclui-se que o sistema é controlável e, portanto, é possível implementar o PI.

Para a seleção dos polos, foram definidos três conjuntos distintos. No primeiro conjunto, foram selecionados os quatro polos iniciais, conforme estabelecido em (4.9), e o quinto polo será determinado por cinco vezes o valor da parte real do polo definido pelas especificações de desempenho (4.5). Para os demais conjuntos, os primeiros quatro polos seguem os mesmos definidos em (4.9)-(4.11), enquanto o último é cinco vezes o valor do polo mais à esquerda no semieixo negativo. De modo, que os conjuntos de polos necessários para o controlador PI são

$$P_{I_1} = -\begin{bmatrix} 2 \pm 3{,}3120j & 6 & 8 & 10 \end{bmatrix}, \tag{4.15}$$
$$P_{I_2} = -\begin{bmatrix} 3 & 4 & 5 & 6 & 30 \end{bmatrix}, \tag{4.16}$$
$$P_{I_3} = -\begin{bmatrix} 6 & 8 & 10 & 12 & 60 \end{bmatrix}. \tag{4.17}$$

Com relação aos ganhos do controlador estabilizante e o ganho do controlador PI, ambos serão calculados ao mesmo tempo usando (2.14). Portanto, a matriz de ganhos K e o ganho K_I, mostrados no formado mostrado em (2.19), para os 3 conjuntos de polos mencionados em (4.15)-(4.17) são iguais a

$$\begin{aligned} \begin{bmatrix} K_1 & K_{I_1} \end{bmatrix} &= -\begin{bmatrix} 858{,}9605 & 228{,}8245 & 652{,}1584 & 286{,}5367 & -989{,}7921 \end{bmatrix}, \\ \begin{bmatrix} K_2 & K_{I_2} \end{bmatrix} &= -\begin{bmatrix} 1885{,}5361 & 491{,}5372 & 1462{,}9592 & 640{,}1058 & -1487{,}7551 \end{bmatrix}, \\ \begin{bmatrix} K_3 & K_{I_3} \end{bmatrix} &= -\begin{bmatrix} 19184{,}078 & 5162{,}8228 & 23407{,}3469 & 7549{,}8342 & -47608{,}1633 \end{bmatrix}. \end{aligned} \tag{4.18}$$

Tem-se os dados da resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com três controladores diferentes projetados com três conjuntos de polos diferentes. Todo o desenvolvimento dos gráficos e do cálculo dos valores presentes na Tabela 3 foram feitos usando *python*. Para os três casos, foram consideradas condições iniciais nulas, ou seja, $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. Os algoritmos das simulações numéricas encontram-se disponíveis no Apêndice F.

Tabela 3 – Dados da resposta ao degrau unitário da posição da haste, estado x_1 e da posição do carrinho, estado x_3, do sistema descrito pelas matrizes (4.3) com o controlador PI implementado, com condições iniciais $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.

Conjunto de polos	Sinal	T_s (s)	MUP	T_r (s)	Valor final	Erro
P_{I_1} (4.15)	x_1	2,5395	–	0	$0°$	$1°$
	x_3	2,4250	21,8987	0,3350	$1\ m$	$0\ m$
P_{I_2} (4.16)	x_1	2,6885	–	0	$0°$	$1°$
	x_3	1,9854	0	0,9254	$1\ m$	$0\ m$
P_{I_3} (4.17)	x_1	1,3442	–	0	$0°$	$1°$
	x_3	1,0012	5,2504	0,2175	$1\ m$	$0\ m$

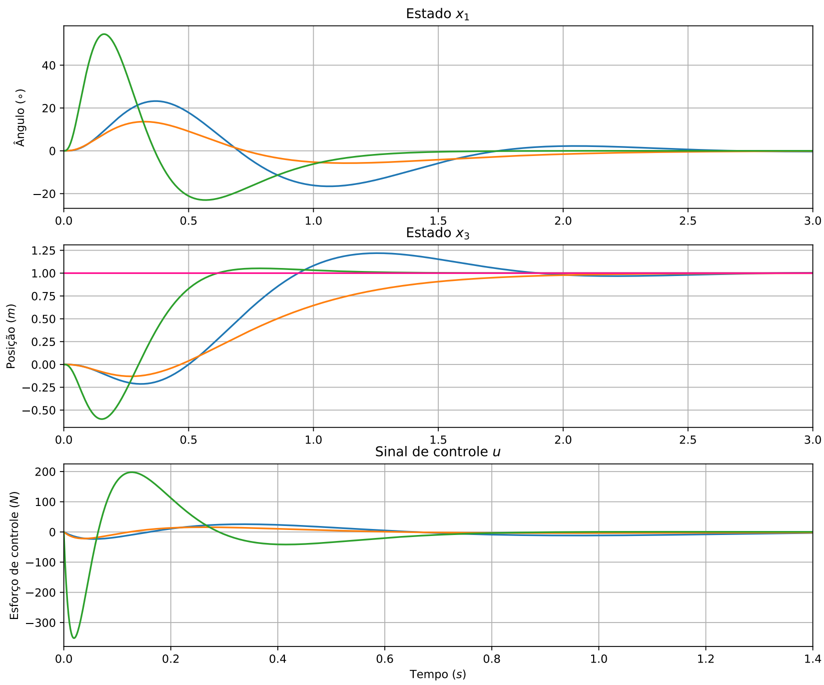
Com relação ao controlador PI, considerando o ângulo θ, usando os dados da Tabela 3 e comparando com os dados apresentados na Tabela 2, é possível notar que todas as outras especificações de desempenho estão praticante idênticas.

Com relação ao controlador PI, considerando a posição x usando os dados da Tabela 3 e comparando com os dados apresentados na Tabela 2, é possível notar que, com exceção do valor final e do erro, todas as outras especificações de desempenho estão praticante idênticas. A única diferença está presente no erro, que é nulo e no valor final, o qual é unitário quando o PI é implementado. Como tem-se uma resposta ao degrau unitário, comprovamos que tal controlador realmente segue a entrada. A Figura 13 ilustra a resposta ao degrau unitário do sistema descrito pelas matrizes (4.3), além de reforçar o fato comprovado anteriormente. O sinal de controle u também está representado.

Com relação ao esforço de controle, nota-se que o sinal gerado a partir do conjunto de polos (4.17) é consideravelmente maior que os dos outros polos. Com relação ao esforço de controle, têm-se picos de 25,6528 em 0,3408s para o conjunto de polos P_{I_1} descrito em (4.15), 15,7944 em 0,2610s para o conjunto de polos P_{I_2} descrito em (4.16) e 197,5419 em 0,1305s para o conjunto de polos P_{I_1} descrito em (4.15). Observa-se que o sistema, para atingir o erro nulo, necessita empregar mais energia com o PI implementado.

Com relação as condições de desempenho descritas em (4.5), o que mais se aproxima é o conjunto de polos (4.17). Pode-se observar que os estados x_1 e x_3 convergem de maneira satisfatória e o carrinho segue a referência do degrau unitário para os três conjunto de polos. Portanto, o objetivo de estabilizar o estado x_1 e o estado x_3 seguir o sinal de referência são satisfeitos.

Figura 13 – Resposta ao degrau unitário e esforço de controle do sistema descrito pelas matrizes (4.3) com controlador seguidor de referência implementado, com os conjuntos (4.15) em azul, (4.16) em laranja e (4.17) em verde e degrau unitário em rosa, com condições iniciais $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.



Fonte: Autoria própria.

4.4 Projeto conjunto de controlador seguidor de referência e observador de estados

Até o dado momento, os projetos foram feitos supondo que os estados estavam todos disponíveis para serem mensurados. Porém, conforme explica Tran, Pepitone e Choi (2019), em sistemas MIMO, é bastante difícil ou até mesmo impossível obter informações de todos os estados de um sistema, portanto, é importante o projeto de um estimador de estados, para efetuar o controle do pêndulo invertido. Conforme Ellis (2002) explica, o uso de sensores pode causar algumas desvantagens, por exemplo:

- No geral, sensores são bem caros, o que aumenta bastante o custo do projeto de um sistema de controle;
- As ligações associadas causam a redução da confiabilidade do sistema de controle;
- É simplesmente impossível medir alguns sinais, em razão de insalubridade, restrição na movimentação presente entre o sensor e controlador, dentre outros motivos;

- É possível a indução de erros por parte dos sensores, como erros cíclicos, ruídos estocásticos e uma flexibilidade limitada.

Neste trabalho, será considerado que não é possível mensurar nenhum estado do pêndulo invertido, sendo necessário projetar um observador para estimar todos os estados. Primeiramente, deve-se modificar a matriz C de (4.3), de forma a capturar x_1, x_2, x_3 e x_4. Portanto, o sistema descrito pelas matrizes (4.3) será reescrito como

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 15{,}2444 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0{,}3630 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -0{,}7407 \\ 0 \\ 0{,}4938 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{4.19}$$

O próximo passo é verificar se o sistema descrito pelas matrizes (4.19) é observável. Usando o conceito de observabilidade apresentado na Definição 2.6, monta-se a matriz de observabilidade descrita como

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix},$$

em que

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 15{,}2444 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0{,}3630 & 0 & 0 & 0 \end{bmatrix}, \qquad Q_2 = \begin{bmatrix} 15{,}2444 & 0 & 0 & 0 \\ 0 & 15{,}2444 & 0 & 0 \\ -0{,}3630 & 0 & 0 & 0 \\ 0 & -0{,}3630 & 0 & 0 \\ 0 & 15{,}2444 & 0 & 0 \\ 232{,}3931 & 0 & 0 & 0 \\ 0 & -0{,}3630 & 0 & 0 \\ -5{,}5332 & 0 & 0 & 0 \end{bmatrix},$$

cujo posto é igual a 4. Portanto, é possível observar todos os estados do sistema descrito pelas matrizes (4.3).

É importante ressaltar que, para os testes a seguir, foi utilizado apenas o conjunto de polos descritos em (4.9). Os outros apresentam resultados semelhantes. O critério utilizado para a escolha dos polos do observador foi o mesmo usado por Tran, Pepitone e Choi (2019), em que, os polos do observador valem 10 vezes a parte real dos polos do controlador estabilizante. Portanto, os polos P_e do observador são iguais a

$$P_e = -\begin{bmatrix} 20 \pm 3{,}3120j & 60 & 80 \end{bmatrix}. \tag{4.20}$$

A matriz de ganhos K_e foi calculada a partir do comando *place*, do *Matlab*, usando as transpostas das matrizes A e C de (4.19) e os polos descritos em (4.20) como argumentos. Portanto, K_e é igual a

$$K_e = \begin{bmatrix} 20 & -2{,}3120 & 0 & 0 \\ 18{,}5564 & 20 & 0 & 0 \\ 0 & 0 & 60 & 1 \\ -0{,}3630 & 0 & 0 & 80 \end{bmatrix}. \tag{4.21}$$

As condições iniciais são adaptados dos valores descritos em Choi et al. (2009) cujos erros iniciais para o ângulo da haste de $e_1(0) = 0{,}3\ rad = 17{,}1887°$ e para a posição de $e_3(0) = 0{,}5\ m$. Portanto, e_0 é igual a

$$e_0 = \begin{bmatrix} 0{,}3 & 0 & 0{,}5 & 0 \end{bmatrix}. \tag{4.22}$$

Com relação ao erro de estimação, ele foi calculado simplesmente como sendo a diferença entre a saída real e observada. Todo o desenvolvimento foi feito em *Matlab*, pois ele permite a construção de blocos para representar os sistemas. Os dados das simulações numéricas encontram-se disponíveis no Apêndice G.

Tabela 4 – Dados da resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com controlador seguidor de referência e observador de estados implementados.

Estado	T_s (s)	MUP (%)	T_r (s)	Valor final
x_1	2,6140	–	0	0°
x_3	1,8684	16,3076	0,4822	1 m

Fonte: Autoria própria.

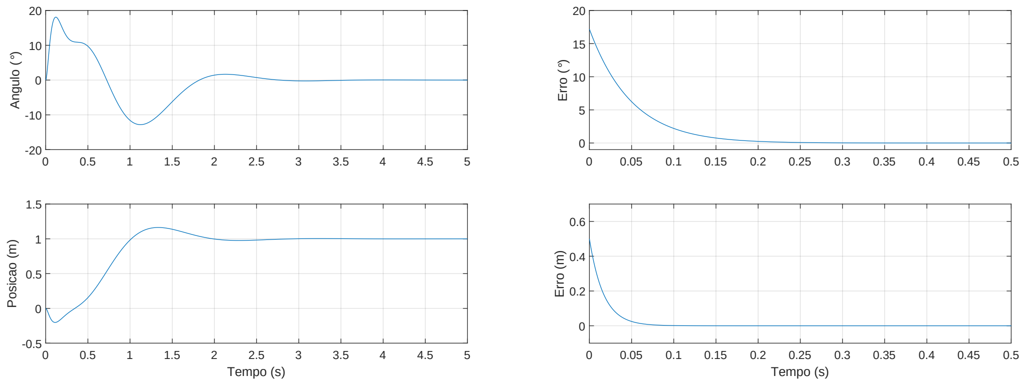
A partir da análise da Figura 14, bem como a Tabela 4 e comparando com os dados presentes na Tabela 3 , com relação a posição da haste, nota-se um T_s ligeiramente superior e formas de onda bem diferentes, porém, em ambos os casos o estado x_1 converge para um valor nulo. Com relação a posição do carrinho, nota-se que o T_s é consideravelmente menor, bem como o MUP. As formas de onda são bem parecidas. Em ambos os casos, o estado x_3 converge para a referência, que é o degrau unitário. Pode-se concluir que o observador de estados implementado conseguiu satisfazer plenamente o objetivo de estimar os estados e fazer o controle do pêndulo invertido.

4.4.1 Projeto conjunto de controlador seguidor estabilizante e observador de estados e fontes de erros

Os dados apresentados na Figura 14 foram gerados levando em consideração que o sistema não apresenta ruídos, e os sensores utilizados são ideais. Porém, não é possível

Figura 14 – Resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com controlador seguidor de referência e observador de estados implementados e os erros entre as saídas reais e estimadas.

(a) Posição da haste e posição do carrinho (b) Erro



Fonte: Autoria própria.

obter tal cenário na prática. Uma fonte de erros advém dos sensores e equipamentos de medição. Mesmo se for utilizado os melhores sensores e equipamentos do mercado, os dados registrados podem apresentar erro, sejam erros de exatidão ou de precisão (CABRAL, 2004). Outra fonte poderosa de erros são ruídos provenientes de incertezas nas constantes do sistemas. Para ilustrar o cenário descrito, serão feitas duas interferências:

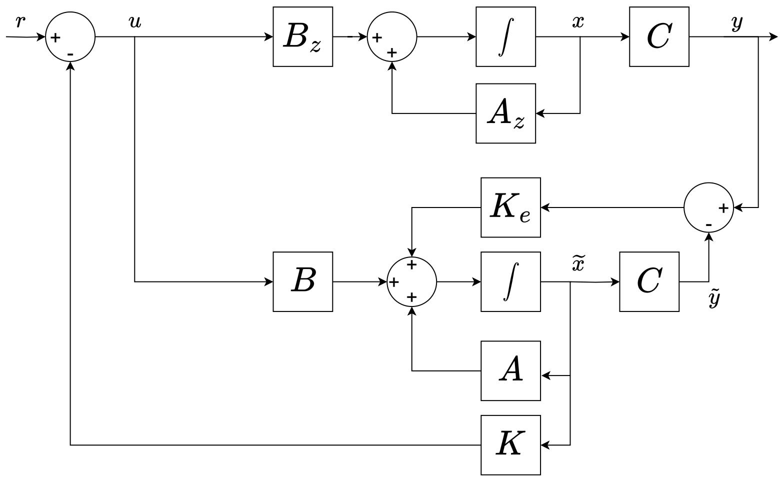
1°. As medidas de massa do carrinho, do pêndulo e o comprimento da haste sofrerão um decréscimo de 5% em seu valor, para simular o erro presente na medição;

2°. Ruído branco gaussiano adicionado à saída real do sistema para simular erros na leitura dos sensores.

4.4.1.1 Simulação de erro presente na medição das constantes do modelo

Para este caso, as matrizes do sistema, utilizadas na simulação do pêndulo invertido desenvolvida em *Simulink/Matlab*, são alteradas para ilustrar os erros de medição das variáveis massa e comprimento da haste da planta real. O controlador utilizado não foi alterado, dado que o objetivo é simular o efeito que a variação dos parâmetros tem no controle ao utilizar um modelo ligeiramente diferente da planta real. O estimador será recalculado de modo que os polos do observador foram escolhidos para variar entre 0,5 e 10 vezes a parte real dos polos do controlador estabilizante. Somente serão apresentados os resultados para o conjunto de polos descritos em (4.9). Os outros conjuntos de polos citados na Tabela 2 seguem a mesma lógica. A Figura 15 ilustra um sistema que apresenta erros de medição. As matrizes com subscrito z indicam onde foram introduzidos os valores com erros. Os dados das simulações numéricas encontram-se disponíveis no Apêndice H.

Figura 15 – Exemplo de um sistema com erros de medição.



Fonte: Autoria Própria.

Tabela 5 – Dados da resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com erros de medição e polos do observador variando entre 0,5 e 10 vezes os polos do controlador.

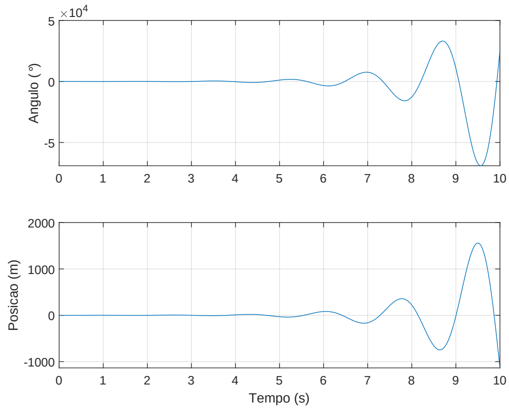
Ganho dos polos do estimador	Sinal	T_s (s)	T_r (s)	Valor final
0,5	x_1	–	0	∞
	x_3	–	–	∞
5	x_1	6,3463	0	0°
	x_3	3,7989	0,7869	1 m
10	x_1	2,6880	0	0°
	x_3	2,0176	0,5165	1 m

A partir da análise da Figura 16 e da Tabela 5, nota-se que, apesar dos erros de medição, a estabilidade não é afetada para ganhos a partir de 5 vezes. Com os ganhos valendo 0,5 vezes os polos do controlador estabilizante, o sistema descrito pelas matrizes (4.3) tornou-se instável, o que era de se supor, dado que o observador não converge em tempo hábil para que as observações dos estados possam ser utilizadas pelo controlador estabilizante. Com ganhos de 5 vezes, ambos os estados apresentaram menos oscilações antes de chegar no regime permanente que os ganhos de 10 vezes.

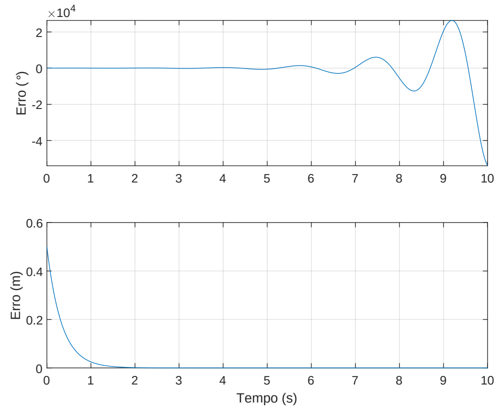
Ogata (2010) explica que a escolha de polos do observador é fundamental até mesmo para evitar a instabilidade da planta toda. A matriz de ganhos K_e, de certa forma, serve como um sinal que faz a correção do modelo da planta. Isso faz com que incertezas, como erros de medição, presentes na planta sejam consideradas. Se tais incertezas envolvidas forem bastante significativas, o sinal de realimentação que passa por K_e será igualmente elevada. Porém, se muito ruído de medida e distúrbios estiverem contaminando o sinal de saída, o sinal de saída é pouco confiável e a realimentação que passa por

Figura 16 – Resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com controlador seguidor de referência e estimador de estados implementados e erros de medição adicionados. Polos do observador ajustados para 0,5 vezes os polos do controlador estabilizante, em azul, 2 vezes os polos do controlador estabilizante, em verde e 5 vezes os polos do controlador estabilizante, em vermelho .

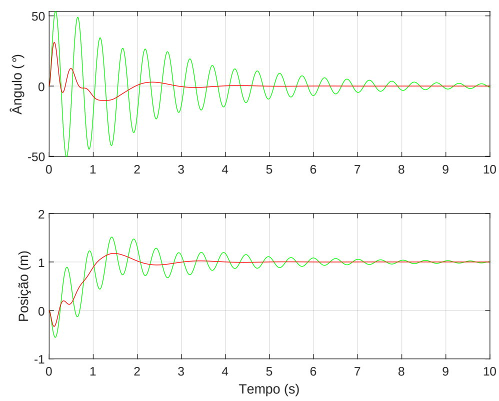
(a) Posição da haste e posição do carrinho.



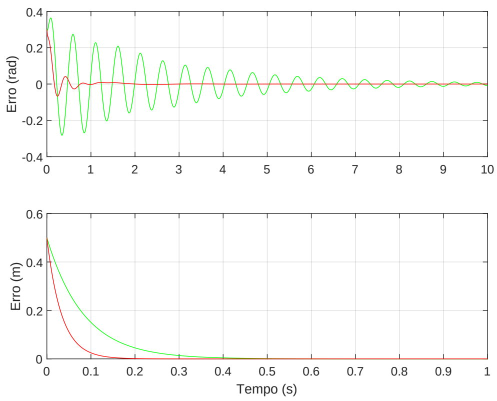
(b) Erro.



(c) Posição da haste e posição do carrinho.



(d) Erro.



Fonte: Autoria própria.

K_e deverá ser pequeno. Portanto, durante o cálculo da matriz K_e, tais efeitos relacionados com a saída devem ser cuidadosamente examinados.

Portanto, conclui-se que é possível contornar incertezas nas medições dos parâmetros do pêndulo invertido com um bom projeto de um estimador de estados.

4.4.1.2 Simulação de ruídos na leitura dos sensores

Ruído branco gaussiano foi adicionado à saída do sistema. Considera-se que o ganho dos polos do observador será igual a 10 vezes a parte real dos polos do controlador. Para simular diferentes amplitudes do ruído, o ganho é variado entre 10^{-4} e 10^{-2}.

Conforme Bortoni (2012) explica, a SNR pode ser expressa como

$$SNR = 20log\frac{P_{sinal}}{P_{ruido}}. \tag{4.23}$$

De acordo com Oppenheim e Willsky (2010), a potência média de um sinal pode ser calculada como

$$P = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} |x(t)|^2, \tag{4.24}$$

em que t_2 e t_1 são os limites do intervalo de tempo do sinal. Aplicando (4.23) e (4.24) aos ganhos entre 10^{-4} e 10^{-2} e também ao sinal observado, obtêm-se os dados da Tabela 6. Nota-se que, à medida que os polos são alocados mais distantes no semieixo negativo, o SNR aumenta, o que indica que o ruído está sendo filtrado e o sinal resultante é o mais próximo possível do sinal ideal. Fora esses resultados, têm-se os dados da resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com o observador de estados implementado e erros na medição dos parâmetros. Os dados das simulações numéricas encontram-se disponíveis no Apêndice I.

Tabela 6 – Dados do SNR do sistema descrito pelas matrizes (4.3) com controlador seguidor de referência, observador de estados implementados e fontes de ruídos.

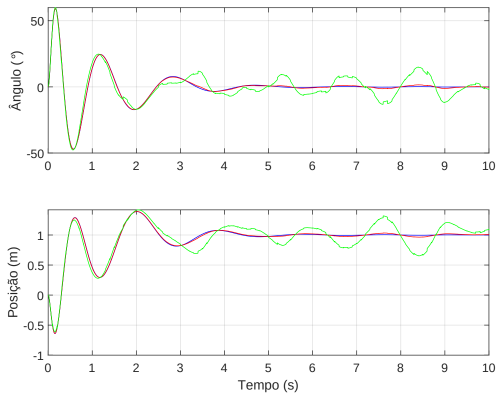
Polos	Ganho do ruído	x_1 (dB)	x_3 (dB)
$-\begin{bmatrix}1 \pm 3{,}3j & 3 & 4\end{bmatrix}$	10^{-4}	186,3698	218,9859
	10^{-3}	146,3618	178,9879
	10^{-2}	108,6811	139,1190
$-\begin{bmatrix}4 \pm 3{,}3j & 12 & 16\end{bmatrix}$	10^{-4}	173,2954	218,8429
	10^{-3}	133,3101	178,8439
	10^{-2}	98,1174	138,8962
$-\begin{bmatrix}10 \pm 3{,}3j & 30 & 40\end{bmatrix}$	10^{-4}	166,9210	218,8568
	10^{-3}	127,0598	178,8576
	10^{-2}	94,8839	138,8976

Tem-se os dados da resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com o observador de estados implementado e ruídos adicionados. Os dados das simulações numéricas encontram-se disponíveis no Apêndice I.

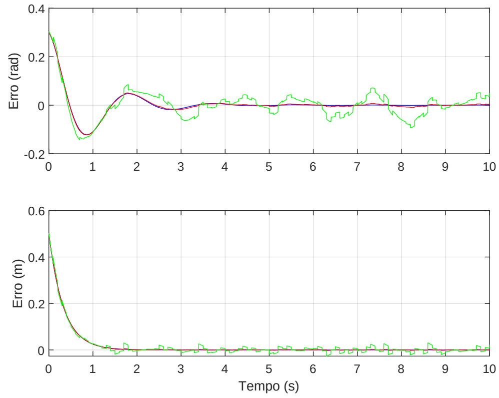
A partir da análise da Figura 17 e da Tabela 6, nota-se maior presença de ruídos com o conjunto de polos para um ganho dos polos do observador a 5 vezes os polos do controlador estabilizante, o que dificulta bastante na visualização da forma de onda. Para um ganho dos polos do observador a 0,5 vezes os polos do controlador estabilizante, a percepção do ruído é menor. Ogata (2010) explica que uma escolha dos polos do observador mais lentos que os polos do controlador estabilizante faz com que o observador atue como uma espécie de filtro, removendo parte do ruído e tornando o sinal resultante mais limpo. Esse fato fica claro ao comparar $P_e = -\begin{bmatrix}1 \pm 3{,}3j & 3 & 4\end{bmatrix}$ e $P_e = -\begin{bmatrix}10 \pm 3.3{,}j & 30 & 40\end{bmatrix}$. Com $P_e = -\begin{bmatrix}10 \pm 3{,}3j & 30 & 40\end{bmatrix}$, os sinais estão fortemente dominados pelo ruído, sendo

Figura 17 – Resposta ao degrau unitário do sistema descrito pelas matrizes (4.3) com controlador seguidor de referência e estimador de estados implementados. Ruído adicionado e polos do observador alocados entre 0,5 e 5 vezes os polos do controlador estabilizante. Posição da haste e posição do carrinho à esquerda e erro à direita. Ruído com ganho de 10^{-4}, em azul, 10^{-3} em vermelho e 10^{-2} em verde.

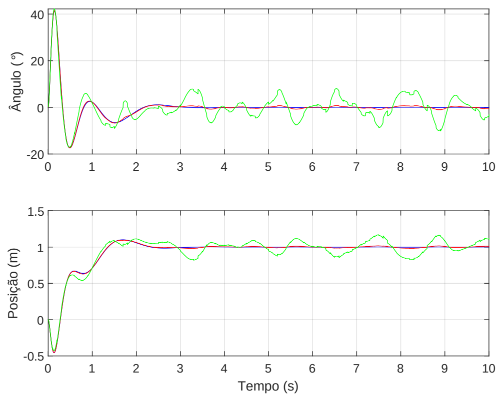
(a) Estados x_1 e x_3, $P_e = -\begin{bmatrix}1 \pm 3{,}3j & 3 & 4\end{bmatrix} \cdot$



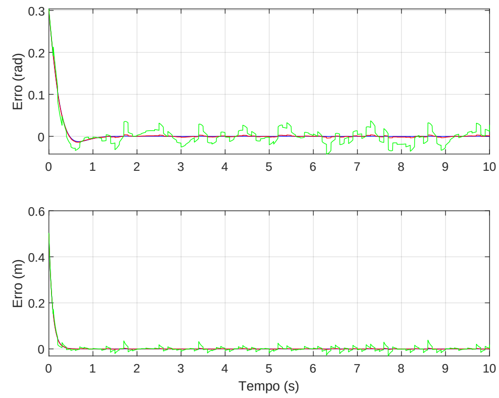
(b) Erro com $P_e = -\begin{bmatrix}1 \pm 3{,}3j & 3 & 4\end{bmatrix} \cdot$



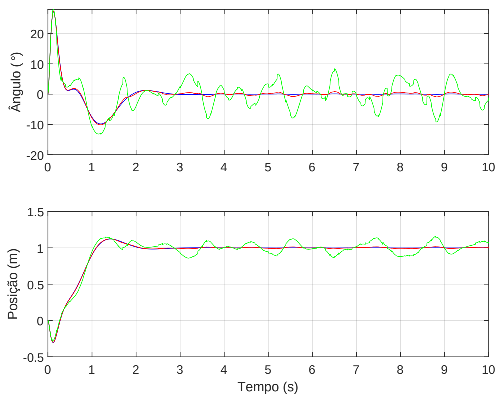
(c) Estados x_1 e x_3, $P_e = -\begin{bmatrix}4 \pm 3{,}3j & 12 & 16\end{bmatrix} \cdot$



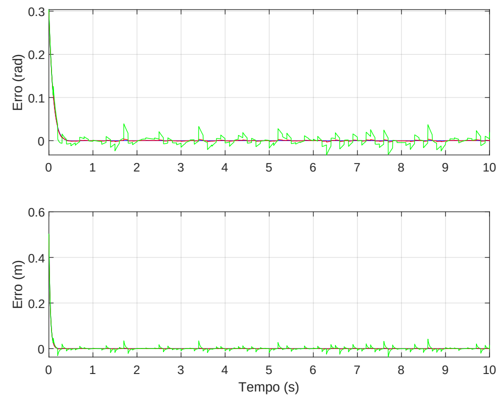
(d) Erro com $P_e = -\begin{bmatrix}4 \pm 3{,}3j & 12 & 16\end{bmatrix} \cdot$



(e) Estados x_1 e x_3, $P_e = -\begin{bmatrix}10 \pm 3{,}3j & 30 & 40\end{bmatrix} \cdot$



(f) Erro com $P_e = -\begin{bmatrix}10 \pm 3{,}3j & 30 & 40\end{bmatrix} \cdot$



Fonte: Autoria própria.

difícil identificar a forma de onda, diferentemente do que ocorre para o outro conjunto de polos. Portanto, pode-se dizer que, para este caso, foi satisfeito o objetivo de se projetar o controle com observador de estados e ruído adicionado, cabendo ao projetista definir a melhor configuração de polos para projeto a depender do objetivo final pretendido: melhor estimação dos estados ou melhor rejeição de ruído.

5 Controle do modelo não linear do pêndulo invertido

Este capítulo dedica-se a discutir o controle para o modelo não linear do pêndulo invertido, descrito por (3.15).

Será projetado controladores não lineares por meio da linearização entrada-saída e usando linearização entrada-saída e a análise da dinâmica zero, usando a teoria de estabilidade de Lyapunov. Utilizando os valores da Tabela 1 em (3.15) obtêm-se

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_2 + g_2 u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u \\ y = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T \end{cases} \tag{5.1}$$

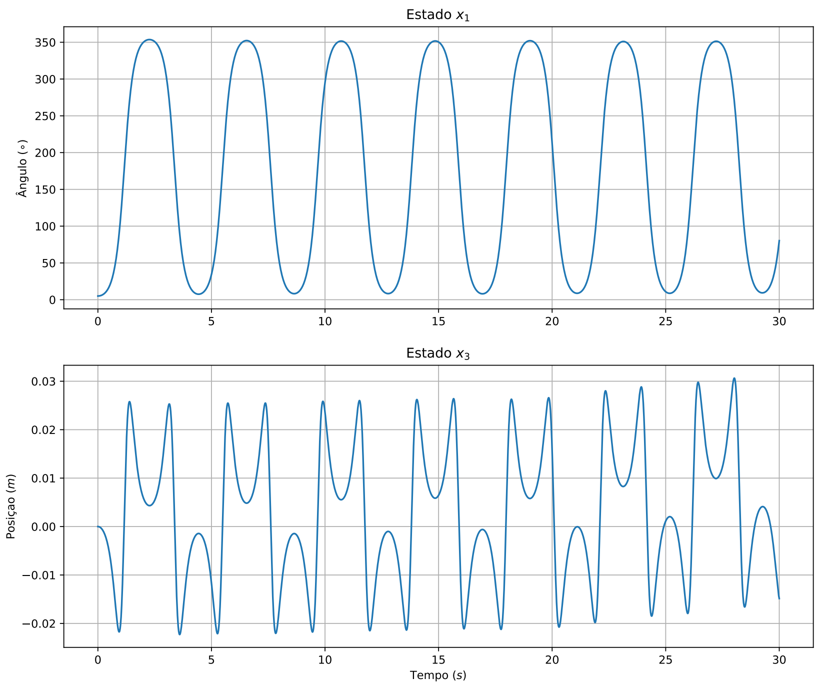
em que

$$\begin{aligned} f_2 &= \frac{3\left(-0{,}25x_2^2 \mathrm{sen}\,(2x_1) + 49\mathrm{sen}\,(x_1)\right)}{1{,}5\mathrm{sen}^2(x_1) + 8{,}5}, \quad & g_2 &= -\frac{6\cos(x_1)}{1{,}5\mathrm{sen}^2(x_1) + 8{,}5}, \\ f_4 &= \frac{2x_2^2 \mathrm{sen}\,(x_1) - 14{,}7\mathrm{sen}\,(2x_1)}{2\left(1{,}5\mathrm{sen}^2(x_1) + 8{,}5\right)}, & g_4 &= \frac{4}{1{,}5\mathrm{sen}^2(x_1) + 8{,}5}. \end{aligned} \tag{5.2}$$

Em um primeiro momento, será analisada a solução numérica de (5.1). Como condições iniciais, define-se um ângulo inicial $\theta = 5° = 0{,}0873\ rad$, ou seja, $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 \end{bmatrix}$. O esforço de controle, conforme Gadelha (2018) usou, será considerado $u = 0$. A Figura 18 ilusta as formas de onda para o ângulo θ e a posição x. Nota-se uma característica oscilatória do sistema descrito por (5.1). Os dados das simulações numéricas encontram-se disponíveis no Apêndice J.

Foi constatado, por meio das simulações que o controlador projetado para o modelo linear gera resultados satisfatórios somente para uma faixa bastante restrita de valores, entre $0° < \theta < 40°$. Para valores fora dessa faixa, o sistema descrito por (5.1) torna-se instável. A explicação é que o modelo linear representa bem a não linearidade do modelo não linear dentro de uma faixa de operação. À medida que o estado x_1 começa a sair da região de linearização, o controlador não é suficiente para estabilizar o processo, sendo necessário implementar o controlador não linear ou então fazer outra linearização.

Figura 18 – Formas de onda da resposta dinâmica do sistema descrito por (5.1) sem nenhum tipo de controlador, com condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 \end{bmatrix}$.



Fonte: Autoria própria.

5.1 Linearização entrada-saída

Hassanzadeh, Mobayen e Harifi (2008) explicam que o objetivo principal é manter a haste levantada, ou seja, o estado x_1 convergir para um valor nulo. Por isso, a saída x_1 será utilizada.

Primeiramente, deve-se derivar a saída até encontrar uma relação explícita entre y e o sinal de controle u. Portanto, derivado o estado x_1 de (5.1), obtêm-se

$$\begin{aligned} y &= x_1, \\ \dot{y} &= \dot{x}_1 = x_2, \\ \ddot{y} &= \dot{x}_2 = f_2 + g_2 u. \end{aligned} \tag{5.3}$$

Como é necessário derivar duas vezes para se obter a relação explícita entre y e u, diz-se que o grau relativo R do sistema descrito por (5.1) é de 2 e, portanto, é parcialmente linearizável e também é controlável. O próximo passo é calcular o valor de u, de forma a mudar a dinâmica do sistema, de não linear para linear. Como foi visto na subseção 2.8.2, a linearização entrada-saída leva a relação linear

$$\ddot{y} = v, \tag{5.4}$$

em que v é a nova entrada de controle linear.

Igualando a expressão de $\ddot{y}$ presente em (5.3) e (5.4) obtêm-se

$$v = f_2 + g_2 u, \tag{5.5}$$

e finalmente, isolando u obtêm-se

$$u = \frac{v - f_2}{g_2}, \tag{5.6}$$

ou ainda

$$u = \frac{Mg\text{sen}\,(x_1) + gm\text{sen}\,(x_1) - \frac{\ell m x_2^2 \text{sen}\,(2x_1)}{2} - \frac{\ell v\left(4M + 3m\text{sen}^{\,2}(x_1) + m\right)}{3}}{\cos\,(x_1)}. \tag{5.7}$$

Substituindo os valores da Tabela 1 em (5.7)

$$u = \frac{0{,}025v\cos\,(2x_1) - 1{,}375v - 0{,}025x_2^2\text{sen}\,(2x_1) + 20{,}58\text{sen}\,(x_1)}{\cos\,(x_1)}. \tag{5.8}$$

Com a expressão de u calculada em (5.8), o próximo passo é substituí-la em (5.1), ou seja

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = v \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = 9{,}8\tan\,(x_1) - \frac{0{,}67v}{\cos\,(x_1)} \\ y = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T \end{cases} \tag{5.9}$$

A entrada de controle v apresentada em (5.4), definida como sendo v_{estab}, é expressa como

$$v_{estab} = -247{,}1744x_1 - 61{,}1786x_2 - 98{,}9792x_3 - 55{,}3179x_4, \tag{5.10}$$

a qual foi obtida usando os polos descritos em (4.9) e a matriz de ganhos K_1 descrita em (4.12). Os outros ganhos referentes aos conjuntos P_2 e P_3 seguem a mesma lógica. Portanto, substituindo a entrada v_{estab} definida em (5.10) na entrada u definida em (5.8) obtém-se a seguinte lei de controle

$$u = \frac{v_{estab} - f_2}{g_2}. \tag{5.11}$$

Como condições iniciais, será considerado

$$x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 \end{bmatrix}. \tag{5.12}$$

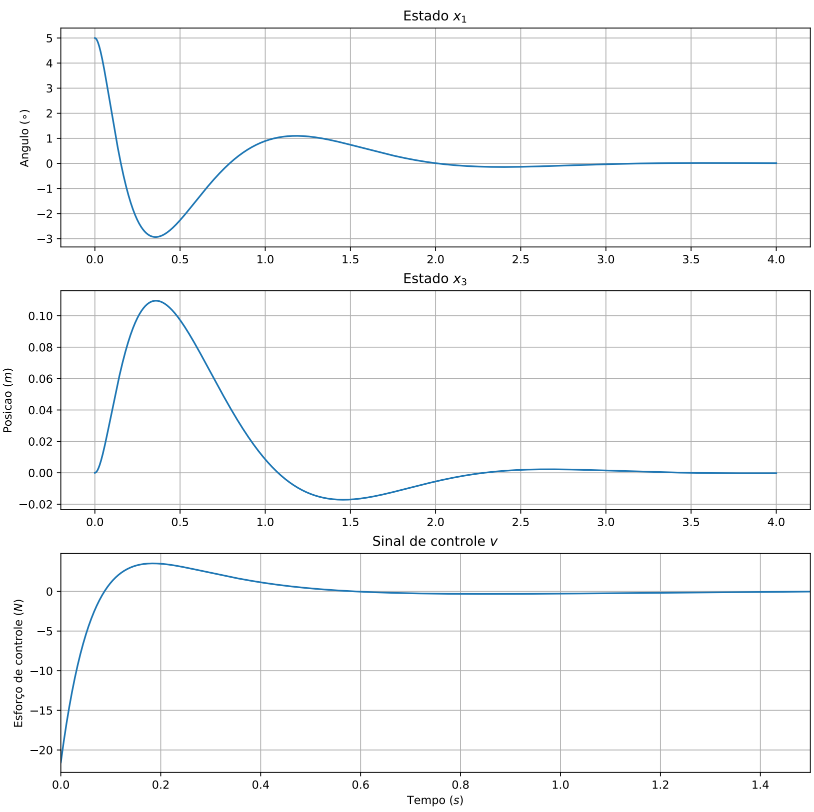
Os resultados das simulações numéricas encontram-se disponíveis no Apêndice K. Têm-se os dados das simulações do sistema descrito por (5.2) com o controlador estabilizante implementado.

Pela análise da Figura 19, bem como a Tabela 7, nota-se que o sistema descrito por (5.1) converte sem muitas oscilações para os estados x_1 e x_3, e o esforço de controle também converte para um valor nulo sem muitas oscilações. Pode-se concluir que o projeto do controlador não linear do pêndulo invertido conseguiu fazer a posição do carrinho e a haste do pêndulo atingir um valor nulo em regime permanente.

Tabela 7 – Resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), empregando o controlador estabilizante projetado por linearização entrada-saída.

Variável	T_s (s)	MUP	T_r (s)	Valor final
x_1	0,6351	8,7266	0	0°
x_3	0,9187	10,9595	0	0 m
v	2,4413	352,9081	0,0073	0 N

Figura 19 – Formas de onda da resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), com controlador estabilizante implementado. Condições iniciais dadas por (5.12).



Fonte: Autoria própria.

5.1.1 Seguimento de referência para o modelo não linear

Para fazer uso de um controlador PI, primeiramente deve-se adicionar uma nova variável de estado, referente ao erro. Da mesma forma que foi feito para o modelo linear, será projetado um seguimento de referência somente para o estado x_3. Levando em consideração todas essas informações, as novas variáveis de estado podem ser escritas como

$$\begin{aligned} x_1 &= \theta, \\ x_2 &= \dot{\theta}, \\ x_3 &= x, \\ x_4 &= \dot{x}, \\ x_5 &= x_r. \end{aligned} \tag{5.13}$$

Com as novas variáveis declaradas, o sistema descrito por (5.1) pode ser reescrito como

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_2 + g_2 u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u \\ \dot{x}_5 = -x_3 + 1 \\ y = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T \end{cases} \tag{5.14}$$

em que f_2, f_4, g_2, g_4 são dadas por (5.2). A entrada de controle u é dada por (5.7). O próximo passo é obter uma expressão para a entrada de controle v, a qual é expressa como

$$v_{P_I} = -858{,}9605x_1 - 228{,}8245x_2 - 652{,}1584x_3 - 286{,}5367x_4 + 989{,}7921x_5. \tag{5.15}$$

Portanto, substituindo a entrada v definida em (5.15) na entrada u definida em (5.8) obtém-se a seguinte lei de controle

$$u = \frac{v_{P_I} - f_2}{g_2}, \tag{5.16}$$

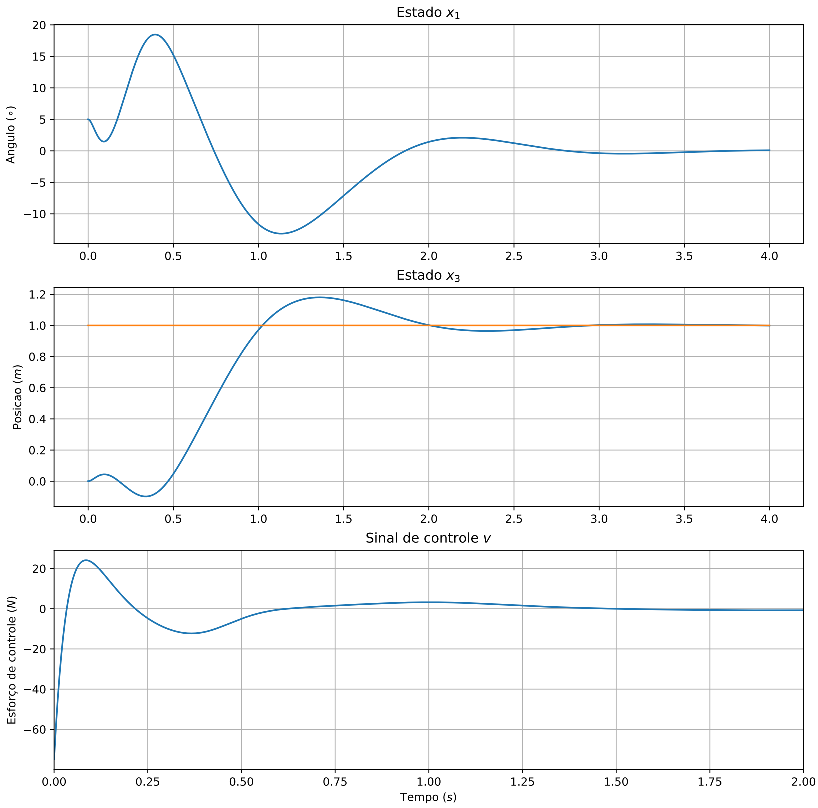
a qual foi obtida com o auxílio da matriz de ganhos K_1 e o ganho K_{I_1} descritos em (4.18) e ajustados durante as simulações. Com relação às condições iniciais, será considerado que $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$. Os dados das simulações encontram-se disponíveis no Apêndice L. Têm-se os dados das simulações do sistema descrito por (5.1) com o seguimento de referência implementado.

Tabela 8 – Resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), empregando o controlador seguidor de referência projetado por linearização entrada-saída.

Variável	T_s (s)	MUP	T_r (s)	Valor final
x_1	2,5167	32,2449	0	0°
x_3	2,6522	18,0349	0,4177	1 m
v	4,1293	2416,1147	0,0006	0 N

Pela análise da Figura 20, bem como a Tabela 8, nota-se que, de fato, a posição do carrinho segue a referência do degrau unitário e a posição da haste converge para um

Figura 20 – Formas de onda da resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), com controlador seguidor de referência implementado, com condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$. Degrau unitário em laranja. Parâmetros descritos na Tabela 1.



Fonte: Autoria própria.

valor nulo, indicando que a proposta de se estabilizar ambos os estados foi satisfeita. Com relação as especificações de desempenho dos estados e do sinal de controle v, o modelo com controlador estabilizante foi mais ágil e com menor sobressinal, se comparado com o controlador PI implementado.

5.2 Seguimento de referência considerando relação entre esforço de controle e especificações de desempenho

Os polos serão escolhidos de forma a se ter um custo benefício entre esforço de controle e especificações de desempenho, ou seja, achar um equilíbrio entre bom desempenho, mas sem gastar muita energia. No geral, tal relação é bem difícil de se otimizar.

Após muitos testes com vários conjuntos, foram selecionados 3 conjuntos de polos, a saber

$$P_1 = -\begin{bmatrix} 1 & 2 & 3 & 4 & 8 \end{bmatrix}, \tag{5.17}$$

$$P_2 = -\begin{bmatrix} 2 \pm 3{,}3j & 0{,}9 & 1 & 2 \end{bmatrix}, \tag{5.18}$$

$$P_3 = -\begin{bmatrix} 2 \pm 3{,}3j & 0{,}7 & 0{,}8 & 10 \end{bmatrix} \cdot \tag{5.19}$$

O sinal de controle v foi obtido por meio da alocação de polos do modelo linear e posteriormente ajustado por tentativa e erro com a ajuda das simulações. Têm-se os dados do sistema descrito por (5.14), bem como as formas de ondas. As condições iniciais serão as mesmas usadas anteriormente, ou seja, $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$. As simulações numéricas encontram-se disponíveis no Apêndice M.

Tabela 9 – Dados da resposta dinâmica das posições da haste (estado x_1), do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), com controlador seguidor de referência implementado.

Conjunto de polos	Sinal	T_s (s)	MUP	T_r (s)	Valor final
P_1 (5.17)	x_1	1,4659	8,7266	0	$0°$
	x_3	5,5728	$8{,}066 \times 10^{-6}$	2,9322	$1\ m$
	Esforço de controle	2,9979	324,2257	0,0074	$0\ N$
P_2 (5.18)	x_1	0,5601	8,7266	0	$0°$
	x_3	8,0129	$5{,}061 \times 10^{-5}$	4,399	$1\ m$
	Esforço de controle	1,8772	717,6577	0,027	$0\ N$
P_3 (5.19)	x_1	1,4834	8,7266	0	$0°$
	x_3	8,3909	$6{,}9215 \times 10^{-6}$	5,2656	$1\ m$
	Esforço de controle	2,6756	208,8472	0,0135	$0\ N$

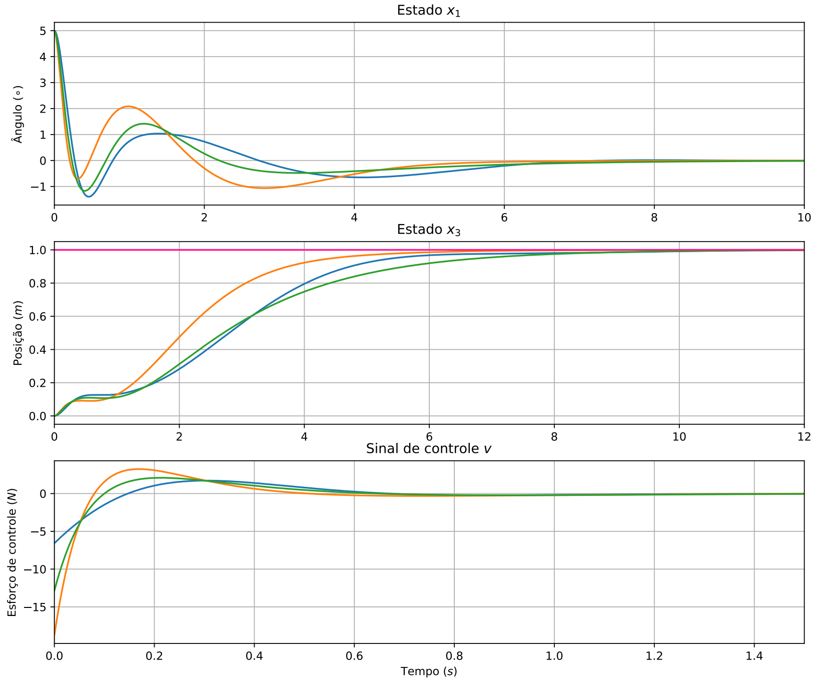
A partir da análise da Figura 21, bem como a Tabela 9, nota-se que, conforme os polos são alocados mais distantes da origem no semieixo negativo, o tempo até chegar no regime permanente diminui, porém, é sacrificado o esforço de controle, visto que se torna significativo em comparação aos outros. Porém, com polos escolhidos mais perto da origem no semieixo negativo e até mesmo com polos escolhidos baseados em especificações de desempenho, o esforço de controle fica menor, porém o sistema fica menos ágil. Nesse caso, fica a critério do projetista quais são suas prioridades com relação ao projeto.

5.3 Linearização entrada-saída e dinâmica zero

Uma outra forma de se fazer o projeto de um controlador para o modelo não linear do pêndulo invertido é utilizando a linearização entrada-saída em conjunto com o conceito de dinâmica interna e teorema de estabilidade de Lyapunov. Tal estratégia é abordada por Alves (2018).

A necessidade de realizar duas diferenciações para estabelecer uma relação direta com a entrada u indica que o sistema carro-pêndulo tem um grau relativo igual a 2. Isso

Figura 21 – Formas de onda da resposta dinâmica das posições da haste (estado x_1), do carrinho (estado x_3) e esforço de controle do sistema descrito por (5.2) com controlador seguidor de referência implementado, com os conjuntos (5.17), em azul, (5.18), em laranja, e (5.19), em verde, e degrau unitário, em rosa, com condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$.



Fonte: Autoria própria.

significa que sua dinâmica interna também é caracterizada por um grau 2, refletindo a complexidade e o comportamento do sistema. Para este caso, será utilizada uma entrada de controle v da forma

$$v = -k_1 x_1 - k_2 x_2. \tag{5.20}$$

Uma forma de se analisar a dinâmica interna do pêndulo invertido é usando a dinâmica zero. Conforme Amba (2015) e Alves (2018) explicam, tal dinâmica ocorre quando os estados x_1 e x_2, já linearizados, com o auxílio de uma entrada de controle v, são conduzidos a zero. Tal relação é descrita como

$$\begin{aligned} \dot{x}_3 &= x_4, \\ \dot{x}_4 &= 0. \end{aligned} \tag{5.21}$$

A análise de (5.21) revela que a dinâmica zero do sistema é caracterizada por polos situados na origem, o que indica uma condição de instabilidade. Para assegurar a estabilidade do sistema, torna-se essencial a implementação de um controlador. Esse controlador é projetado com base no Teorema de Estabilidade de Lyapunov e incorpora conceitos de sistemas singularmente perturbados. Conforme Alberto (2006) e Amba (2015) explicam, ao definir o ganho da lei de controle linearizante, expresso em (5.7), busca-se assegurar que o sistema apresente duas dinâmicas distintas: uma lenta, relacionada ao movimento do carro, e outra rápida, vinculada à oscilação do pêndulo. Essa configuração caracteriza o sistema como singularmente perturbado. Tal abordagem permite que as dinâmicas do sistema sejam estabilizadas de maneira independente, utilizando controladores específicos para cada uma delas, garantindo assim uma resposta adequada às diferentes exigências de comportamento do sistema. O teorema a seguir será utilizado para conceber um controlador. Encontra-se disponível em Silva (2006).

Teorema 5.1 (Critério de estabilidade de Lyapunov) *Considere o sistema descrito como*

$$\dot{x} = f(x)$$
$$x(t_0) = x_0,$$

em que $f : D \to \mathbb{R}^n$ *é chamada aplicação Lipschitz,* $D \subset \mathbb{R}$ *é um aberto. Considerando* $\bar{x} \ \epsilon \ D$ *um ponto de equilíbrio e* $V : D \to \mathbb{R}$ *uma função cuja derivada de ordem* 1 *são contínuas. Serão admitidas as seguintes proposições:*

- $V(x)$ *é definida localmente positiva ao redor de* $\bar{x}$*;*
- $\dot{V}(x)$ *é semidefinida localmente negativa ao redor de* $\bar{x}$*.*

O ponto $\bar{x}$ *é considerado ponto de equilíbrio estável. Caso* $\dot{V}(x)$ *seja definida negativa localmente, considera-se que a estabilidade é assintótica.*

O Teorema de Lyapunov estabelece que, se for possível identificar uma função $V(x)$ estritamente positiva cuja derivada temporal seja não positiva em um ponto de equilíbrio específico, então a estabilidade desse ponto é assegurada nas proximidades. Portanto, para criar uma função de controle que regule a dinâmica ao redor do ponto de equilíbrio zero, define-se uma nova variável de controle, u_1, que é essencial para o sistema. u_1 pode ser escrita como

$$u_1 = u + v_2, \tag{5.22}$$

em que a nova entrada v_2 serve para representar a lei de controle linearizante da dinâmica zero e u é a lei de controle linearizante descrita em (5.7).

Efetuando a substituição do esforço de controle definido em (5.22) em (5.2), obtêm-se

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = v + g_2 v_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u + g_4 v_2. \end{cases} \tag{5.23}$$

É necessário criar v_2 como sendo uma função candidata de Lyapunov $V(x)$ com base nos estados x_3 e x_4 presentes na dinâmica interna. Para isso, considera-se

$$\begin{aligned} e_3 &= x_3 - x_{3d}, \\ e_4 &= x_4 - x_{4d}, \end{aligned} \tag{5.24}$$

em que e_3 e e_4 representam os erros referentes à posição e a velocidade do carrinho. Neste caso, x_{3d} significa a posição de referência do carrinho e x_{4d} significa a velocidade de referência do carrinho. Será considerado x_{4d} como sendo um valor nulo. Considera-se também as constantes c_1 e c_2, as quais são positivas e não nulas. Desta forma, a função candidata de Lyapunov pode ser escrita como

$$V(x) = \frac{c_1 e_3(x)^2}{2} + \frac{c_2 e_4(x)^2}{2}. \tag{5.25}$$

Calculado a derivada da função definida em (5.25) obtêm-se

$$\begin{aligned} \dot{V} &= c_1 e_3 \dot{e}_3 + c_2 e_4 \dot{e}_4 \\ &= c_1 e_3 \dot{e}_3 + c_2 x_4 \dot{x}_4. \end{aligned} \tag{5.26}$$

Têm-se que $\dot{e}_3 = e_4 = x_4$, é possível reescrever (5.26) como sendo

$$\begin{aligned} \dot{V} &= c_1 e_3 x_4 + c_2 e_4 \dot{x}_4 \\ &= x_4(c_1 e_3 + c_2 \dot{x}_4). \end{aligned} \tag{5.27}$$

Para assegurar que $\dot{V}(x)$ seja definida negativa, garantindo que a dinâmica interna seja estável, considera-se $c_3 > 0$, de forma que

$$\dot{V} = -c_3 x_4^2 \tag{5.28}$$

Isolando $\dot{x}_4$ a partir da igualdade entre (5.27) e (5.28), têm-se que

$$\dot{x}_4 = \frac{-c_3 x_4 - c_1 e_3}{c_2}. \tag{5.29}$$

Feito isso, igualando a expressão de $\dot{x}_4$ com a expressão de $\dot{x}_4$ presente em (5.2), obtêm-se

$$v_2 = \frac{g_2(-c_1 e_3 - c_3 x_4 - c_2 f_4) - c_2 g_4 (v - f_2)}{c_2 g_2 g_4}. \tag{5.30}$$

Com a expressão (5.30) calculada, o próximo passo é substituí-la em (5.22). Feito isso, obtêm-se

$$u_1 = u + \frac{g_2(-c_1 e_3 - c_3 x_4 - c_2 f_4) - c_2 g_4 (v - f_2)}{c_2 g_2 g_4}. \tag{5.31}$$

Contudo, Amba (2015) explica que a expressão encontrada em (5.31) faz o sinal de controle v, que é responsável pela linearização da saída x_1, ser eliminado. Portanto, por causa disso, a diferença entre u e v_2 foi utilizada, resultando no sinal de controle u_1 definido por

$$u_1 = u - \frac{g_2(-c_1 e_3 - c_3 x_4 - c_2 f_4) - c_2 g_4 (v - f_2)}{c_2 g_2 g_4}. \tag{5.32}$$

Substituindo o controlador (5.32) em (5.1) obtêm-se

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_2 + g_2 u_1 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_4 + g_4 u_1 \\ y = \begin{bmatrix} x_1 & x_3 \end{bmatrix}^T \end{cases} \tag{5.33}$$

em que f_2, f_4, g_2, g_4 são dados por (5.2).

Alves (2018) explica que para valores pequenos de k_1 e k_2, o sistema é incapaz de manter o pêndulo na posição invertida. Foi observado também que o parâmetro k_1 afeta o tempo necessário para o pêndulo se estabilizar, enquanto o k_2 influencia a intensidade do impulso necessário para alcançar a posição desejada do carro. As constantes c_1 e c_3 são responsáveis por influenciar a resposta transiente do sistema, ao passo que c_2 é a chave para o controle do tempo de acomodação. Observou-se ainda que valores de c_1 superiores a 84 induzem o sistema a um estado de oscilações amplificadas. Portanto, seguindo estes critérios, as constantes k_1, k_2, c_1, c_2 e c_3, após testes e ajustes, foram

$$k_1 = 100, \quad k_2 = 20, \quad c_1 = 10, \quad c_2 = 1, \quad c_3 = 20. \tag{5.34}$$

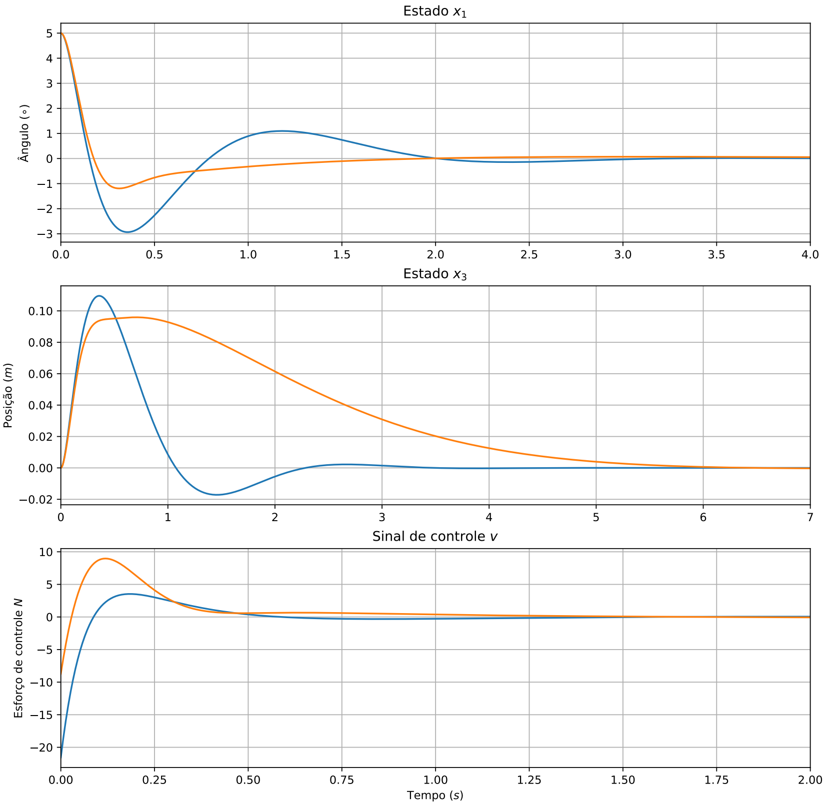
Nas simulações foi considerado que o pêndulo possui ângulo inicial de 0,0873 rad e a posição inicial nula para o carrinho. Portanto, $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 \end{bmatrix}$. Em um primeiro momento, será considerado que $x_{3d} = 0$, porém, também serão exploradas outras posições de referência para o carrinho. Serão comparados os resultados desta estratégia com o obtido por meio da linearização entrada-saída. Os dados das simulações numéricas encontram-se disponíveis no Apêndice N.

Tabela 10 – Resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), empregando o controlador estabilizante projetado por linearização entrada-saída e a linearização entrada-saída e dinâmica zero. Condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 \end{bmatrix}$.

Controlador	Sinal	T_s (s)	MUP (%)	T_r (s)	Valor final
(5.11)	x_1	0,6351	8,7266	0	$0°$
	x_3	0,9187	10,9595	0	$0\ m$
	Esforço de controle	2,4412	2156,9978	0	$0\ N$
(5.32)	x_1	0,3517	8,7266	0	$0°$
	x_3	3,5128	9,5924	0	$0\ m$
	Esforço de controle	5,9814	872,6674	0	$0\ N$

Pela análise da Figura 22, bem como a Tabela 10, nota-se que o modelo descrito em (5.33) é bem mais ágil para estabilizar o estado x_1, porém é bem mais lento para

Figura 22 – Formas de onda da resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), empregando o controlador estabilizante projetado por linearização entrada-saída, em azul, e a linearização entrada-saída e dinâmica zero, em laranja. Condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 \end{bmatrix}$.



Fonte: Autoria própria.

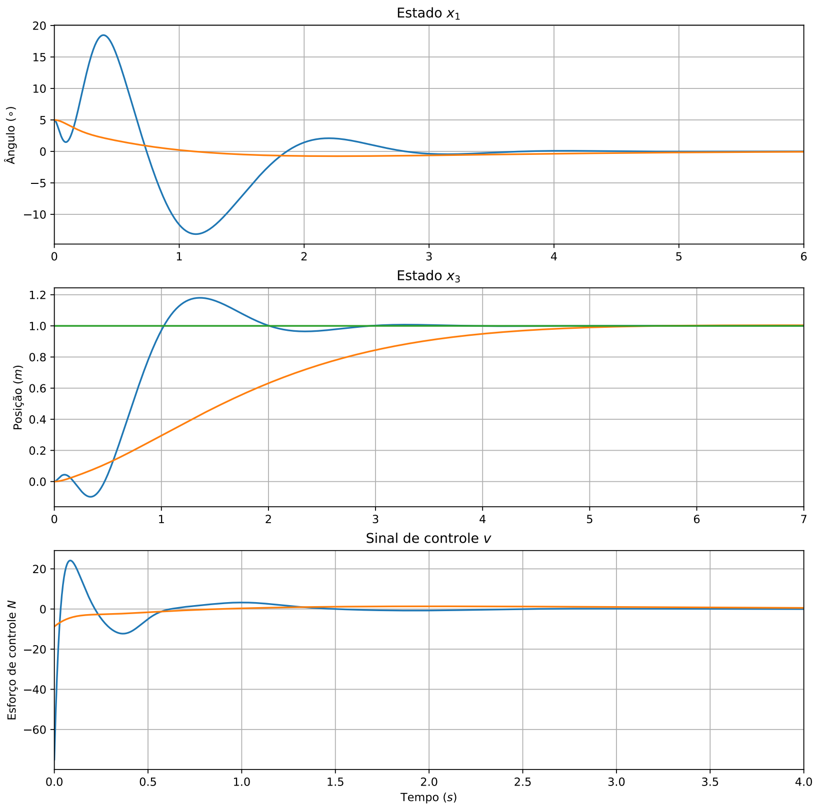
estabilizar o estado x_3. Com relação ao esforço de controle, nota-se que (5.33) tem picos menores que (5.9), indicando um menor gasto de energia por parte do controlador.

Foi calculado a potência média dos esforços de controle com os controladores (5.11) e (5.32), cujos valores são, respectivamente, iguais a 0.2258 W e 0.2484 W. Pode-se dizer que o controlador projetado a partir da análise da dinâmica zero mostra-se menos eficiente do ponto de vista de gasto de energia, o que pode ser interessante caso o projetista queira uma solução que use menos recursos. Pode-se dizer que ambas as estratégias conseguem estabilizar o estado x_1 e x_3.

5.3.1 Seguimento de referência e controlador por dinâmica zero

É possível fazer o carrinho seguir uma referência, como rampa, onda quadrada ou o degrau unitário . Para isso, serão utilizadas as posições de referência presentes em (5.24). Nesse caso, será fixado o valor de $x_{3d} = 1$ e será mantido $x_{4d} = 0$. Com relação as condições iniciais, serão $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$. Os dados das simulações numéricas encontram-se disponíveis no Apêndice O.

Figura 23 – Formas de onda da resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), empregando o controlador para seguimento de referência projetado por linearização entrada-saída (5.16), em azul, e a linearização entrada-saída e dinâmica zero (5.32), em laranja, e degrau unitário, em verde. Condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$.



Fonte: Autoria própria.

A partir da análise da Figura 23, bem como a Tabela 11, nota-se que o controlador por dinâmica zero é mais ágil para estabilizar o estado x_1, porém, mais lento para estabilizar o estado x_3. O esforço de controle, da mesma forma que ocorre no caso do controlador estabilizante, também apresenta picos bem menores.

Tabela 11 – Resposta dinâmica das posições da haste (estado x_1) e do carrinho (estado x_3) e esforço de controle do modelo não linear descrito por (5.1), empregando o controlador para seguimento de referência projetado por linearização entrada-saída e a linearização entrada-saída e dinâmica zero. Condições iniciais $x_0 = \begin{bmatrix} 0{,}0873 & 0 & 0 & 0 & 0 \end{bmatrix}$.

Controlador	Sinal	T_s (s)	MUP (%)	T_r (s)	Valor final
(5.16)	x_1	2,5167	32,2449	0	0°
	x_3	1,8638	9,0174	0,8870	1 m
	Esforço de controle	4,1293	7495,7204	0	0 N
(5.32)	x_1	0,6544	8,7266	0	0°
	x_3	4,1851	0,1988	2,7322	1 m
	Esforço de controle	6,9560	134,3592	0,2075	0 N

Pode-se concluir que ambas as estratégias conseguem atingir o objetivo, que é o de estabilizar o estado x_1. A vantagem do controlador (5.33) é que ele apresenta menos oscilações antes de chegar no regime permanente e um menor gasto de energia.

Foi calculado a potência média dos esforços de controle com os controladores (5.16) e (5.33), cujos valores são, respectivamente, iguais a 2.5024 e 0.2027. Pode-se dizer que o controlador projetado a partir da análise da dinâmica zero mostra-se menos eficiente do ponto de vista de gasto de energia, o que pode ser interessante caso o projetista queira uma solução que use menos recursos. Pode-se dizer que ambas as estratégias conseguem estabilizar o estado x_1 e x_3.

6 Conclusão e trabalhos futuros

A alocação de polos se revela como uma ferramenta indispensável e altamente eficaz no controle de sistemas dinâmicos complexos. Ao longo deste trabalho, foram explorados os fundamentos teóricos da técnica. Por meio da análise das abordagens tradicionais e modernas para a alocação de polos, pode-se compreender como a escolha estratégica das posições dos polos influencia diretamente a estabilidade, desempenho e resposta dinâmica dos sistemas.

Pela análise obtida tanto pela Figura 12 como pela Tabela 2, observa-se que o sistema descrito pelas matrizes (4.3) chega mais rápido ao regime permanente e sem grandes oscilações à medida que os polos dominantes do sistema se encontram mais afastados da origem no semiplano negativo. Se os polos forem alocados distantes o bastante, o sistema pode chegar até mesmo sem nenhuma oscilação, além de ficar mais ágil. Nota-se ainda que o esforço de controle também fica tão rápido e sem oscilações quanto os polos dominantes do sistema são afastados da origem no semiplano negativo, no entanto, necessitando de um maior consumo de energia.

Com o PI devidamente implementado na malha do sistema, o erro em regime permanente foi anulado e o valor final atingindo, comprovado tanto pela Figura 13, como pelos dados presentes na Tabela 3. Como ficou comprovado nas simulações, implementar um PI na malha requer muito mais energia do que implementar somente um controle estabilizante.

Com o observador de estados implementado, o controle foi feito de forma satisfatória, mesmo com o erro de estimação inicial não nulo, conforme evidenciado na Figura 14 e na Tabela 4. Além disso, a configuração estudada lidou bem com erros de medição, comprovado pela Figura 16 e pela Tabela 5, assim como com ruídos, indicado pela Figura 17, com polos cuidadosamente escolhidos.

Os controladores não lineares projetados para o sistema descrito por (5.2) conseguiram estabilizar tanto a posição do carrinho, como o ângulo da haste e também fazer o carrinho seguir uma referência, graças ao PI, apesar de o pêndulo invertido ser um sistema parcialmente linearizável. Essas afirmações são corroboradas pelos resultados mostrados nas figuras 19 e 20.

Com relação ao controlador projetado pela análise da dinâmica interna, ele se mostrou bem eficiente, inclusive sendo mais eficiente do ponto de vista de gasto de energia quando comparado com o modelo obtido por linearização entrada-saída. Portanto, pode-se concluir que o propósito do projeto foi alcançado com sucesso.

O modelo não linear de fato representa mais fielmente o modelo real, no geral, têm especificações de desempenho melhores que o modelo linear e também consegue ter um esforço de controle menor. Especial destaque é feito para o controlador projetado pela análise da dinâmica interna que apresentou resultados superiores.

É notável como a alocação de polos, bem como a linearização entrada-saída se adaptam à era da automação e inovação tecnológica, fornecendo meios para aprimorar o controle de sistemas cada vez mais complexos. No entanto, é fundamental reconhecer que, apesar de sua eficácia, a alocação de polos não é uma abordagem única para todos os problemas e deve ser avaliada cuidadosamente em relação a outras técnicas de controle disponíveis.

6.1 Proposta de trabalhos futuros

Como sugestão de futuros trabalhos e até mesmo uma forma de continuação desta monografia, sugere-se verificar a possibilidade de projetar um controlador de referência para o estado x_1. É interessante também testar outras estratégias de controle, como controle adaptativo e também lógica *fuzzy*, implementando também um controle para a posição do carrinho. Por fim, recomenda-se a aplicação prática dos controladores projetados em um sistema real, o que permitirá uma avaliação mais criteriosa da versatilidade e qualidade dos resultados obtidos via simulação.

Referências

ALBERTO, L. F. C. *Caracterização e estimativas da área de atração de sistemas dinâmicos não lineares.* Tese (Dissertação de Mestrado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006. Disponível em: <https://www.teses.usp.br/teses/disponiveis/livredocencia/18/tde-26042010-100000/pt-br.php>.

ALVES, R. G. *Controle de um pêndulo invertido utilizando técnica de linearização por realimentação.* 2018. 77 p. Monografia (Graduação em Engenharia de Controle e Automação) - Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto. Disponível em: <https://monografias.ufop.br/handle/35400000/1651>.

AMBA, A. J. Feedback linearization, sliding mode and swing up control for the inverted pendulum on a cart. 2015.

ARCOLEZI, H. H. et al. Análise da inserção do ganho proporcional no método de controle por servossistema do tipo 1 no sistema do pêndulo invertido. I congresso das engenharias do estado de Mato Grosso, Mato Grosso, 2017.

ARRIETA, A. R. A.; ALVIZ, D. C. F.; COBA, A. C. N. Implementación del método de perfil de gota para la medición de la tensión interfacial en matlab®, utilizando un nuevo sistema de coordenadas de referencia. *Avances: Investigacion en Ingeniería*, Facultad de Ingeniería (Seccional Bogotá), v. 8, n. 1, p. 40–46, 2011. Disponível em: <https://dialnet.unirioja.es/servlet/articulo?codigo=6684781>.

BORGES, D. G. A. *Concepção e controle de um processo pêndulo invertido.* 2003. Disponível em: <http://www2.ene.unb.br/gaborges/pesquisa/controle/pendulo/index.htm>.

BORTONI, R. Amplificadores de potência. *Encontro de Sistemas de Áudio-STUDIO R/-SELENIUM Disponível em http://www. dee. ufcg. edu. br/˜ gutemb/AmplificadorPotencia. pdf Acesso em*, v. 9, 2012. Disponível em: <https://studior.com.br/amp_avan.pdf>.

BOYCE, W.; DIPRIMA, R.; IÓRIO, V. de M. *Equações diferenciais elementares e problemas de valores de contorno.* Livros Técnicos e Científicos, 2002. Disponível em: <https://books.google.com.br/books?id=LMXtAAAACAAJ>.

BURNS, R. *Advanced Control Engineering.* Elsevier Science, 2001. Disponível em: <https://books.google.com.br/books?id=DovwVQu6ImsC>.

BUZETTI, A. S. *Projeto de controle robusto chaveado com falhas nos sensores.* Dissertação (Dissertação (mestrado)) — Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira, Ilha Solteira, 2017.

CABRAL, P. Erros e incertezas nas medições. *IEP – Instituto Electrotécnico Português*, 2004. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/59707784/Erros_Incertezas20190613-54353-1a8wqwh-libre.pdf?1560460297=&response-content-disposition=inline%3B+filename%3DErros_Incertezas.pdf&Expires=1724805417&Signature=edyInw6450OoWuRfwm7~mYE0T~OnMuQTmSZ7LsfLEuWP7SmoZNua4hu-NAt9GHCsx1tcR70CGJA0DOmrS16decei96emFwA6rGV94eW8NuKaPFIeekzPrTNFU4VxQ3hyaR~4xO2Lgtjkz8nyIUhFzgTH-uBLptMJ-g1IO3SgdFBSdXlhMiW4RvUDQuQmOvKht9

i7fCuk7sHGfzG~7~OO-UNUsm94wXBD5G5ry4WENBH83w1AQkIVLIOXy3NGWdZ9lt lUMt8gaPWjqJC-p9X5ZmAZnOiETr-HkXzFwke0d0RWKfPclcRSrSQNxLsNPdU-Xv7~-TQYcuDTsUilCig5CA___&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>.

CADENGUE, L. S. et al. Linearização por realimentação e aprendizagem por reforço para o controle do sistema de posicionamento do rov / feedback linearization and reinforcement learning for controlling the positioning system of a rov. *Brazilian Applied Science Review*, v. 4, n. 3, p. 1523–1534, May 2020. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BASR/article/view/10702>. doi: 10.34115/basrv4n3-062.

CARRARO, S. A. *Estudo e projeto de controle automático para um sistema aeropêndulo*. Ilha Solteira: [s.n.], 2023. 72 p. Trabalho de conclusão de curso (Graduação em Engenharia Elétrica). Disponível em: <http://hdl.handle.net/11449/244547>.

CHEN, C. *Linear System Theory and Design*. [S.l.]: Oxford University Press, 2014. (The Oxford Series in Electrical and Computer Engineering Series).

CHOI, N.-S. et al. Application to stabilizing control of nonlinear mobile inverted pendulum using sliding mode technique. *Journal of Ocean Engineering and Technology*, Korean Society of Ocean Engineers, v. 23, n. 2, p. 1–7, 2009. Disponível em: <https://www.joet.org/upload/pdf/HOGHC7_2009_v23n2_1.pdf>.

COELHO, F. *Computação Científica com Python*. Flávio Codeço Coelho, 2007. Disponível em: <https://books.google.com.br/books?id=yHw25UKduJIC>.

DALTIN, D. C. *Controle de vibrações em estruturas flexíveis utilizando observador de estados*. Dissertação (Mestrado) — Universidade Estadual Paulista. Faculdade de Engenharia, Bauru, 2017. Dissertação (Mestrado). Disponível em: <http://hdl.handle.net/11449/150854>.

DINIZ, I. S. et al. Controle de um pêndulo invertido auxiliado por computador integrando pid e servossistema multivariável. v. 8, 2009. Disponível em: <https://www.researchgate.net/profile/Ivando-Diniz/publication/283056127_CONTROLE_DE_UM_PENDULO_INVERTIDO_AUXILIADO_POR_COMPUTADOR_INTEGRANDO_PID_E_SERVOSSISTEMA_MULTIVARIAVEL/links/5638e27708ae4bde5021e11c/CONTROLE-DE-UM-PENDULO-INVERTIDO-AUXILIADO-POR-COMPUTADOR-INTEGRANDO-PID-E-SERVOSSISTEMA-MULTIVARIAVEL.pdf>.

DORF, R.; BISHOP, R. *Sistemas de Controle Modernos*. LTC, 2018. Disponível em: <https://books.google.com.br/books?id=i9r7zwEACAAJ>.

ELLIS, G. *Observers in Control Systems: A Practical Guide*. Elsevier Science, 2002. Disponível em: <https://books.google.com.br/books?id=hO6YDG8iVpUC>.

FILHO, F. *Algoritmos numéricos*. LTC, 2007. Disponível em: <https://books.google.com.br/books?id=83y5MgAACAAJ>.

GADELHA, N. d. F. *Controle de um pêndulo invertido utilizando lógica fuzzy*. 2018. 74 p. Monografia (Graduação em Engenharia Mecânica). Disponível em: <https://repositorio.ufersa.edu.br/handle/prefix/1980>.

GUIMARãES, P. V. B. *Controle Semiativo de Modelo de Pêndulo Invertido para Aerogeradores Offshore*. 101 p. Dissertação (Mestrado) — Universidade de Brasília, Faculdade UnB Gama/FT, Brasília, DF, 2016. Dissertação de Mestrado em Integridade de Materiais da Engenharia, Publicação 031A/2016.

HASSANZADEH, I.; MOBAYEN, S.; HARIFI, A. Input-output feedback linearization cascade controller using genetic algorithm for rotary inverted pendulum system. *American Journal of Applied Sciences*, v. 5, n. 10, p. 1322–1328, 2008. Disponível em: <https://www.researchgate.net/profile/Saleh-Mobayen/publication/26516902_Input-Output_Feedback_Linearization_Cascade_Controller_Using_Genetic_Algorithm_for_Rotary_Inverted_Pendulum_System/links/0c96052557ddad6f3d000000/Input-Output-Feedback-Linearization-Cascade-Controller-Using-Genetic-Algorithm-for-Rotary-Inverted-Pendulum-System.pdf>.

LUENBERGER, D. An introduction to observers. *IEEE Transactions on automatic control*, IEEE, v. 16, n. 6, p. 596–602, 1971.

LUENBERGER, D. G. Observing the state of a linear system. *IEEE transactions on military electronics*, IEEE, v. 8, n. 2, p. 74–80, 1964.

NASCIMENTO, T. P. *Controle de trajetória de robôs móveis omni-direcionais: uma abordagem multivariável*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia, 2009. Disponível em: <http://www.ppgee.eng.ufba.br/esp/teses/ac64916ca6ef3bdd2f466bffdd30af7e.pdf>.

NISE, N. *Engenharia de Sistemas de Controle*. 6. ed. LTC, 2017. Disponível em: <https://books.google.com.br/books?id=7LH0zwEACAAJ>.

OGATA, K. *Modern Control Engineering*. 5ª. ed. Englewood Cliffs, NJ: Prentice-Hall, Inc., 2010.

OPPENHEIM, A.; WILLSKY, A. *Sinais e Sistemas*. Pearson Universidades, 2010. Disponível em: <https://books.google.com.br/books?id=ZOg9bwAACAAJ>.

PEAUCELLE, D. et al. A new robust d-stability condition for real convex polytopic uncertainty. *Systems & Control Letters*, Elsevier, v. 40, n. 1, p. 21–30, 2000.

PIRES, L. da S. *Técnicas de controle para robôs tipo pêndulo invertido*. João Monlevade: [s.n.], 2023. 54 p. Monografia (Graduação em Engenharia Elétrica). Disponível em: <http://www.monografias.ufop.br/handle/35400000/5600>.

RAFIKOVA, E. *Controle de um robô móvel através de realimentação de estados utilizando visão estereoscópica*. 142 p. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Campinas, SP, 2010. Tese (Doutorado). Disponível em: <https://hdl.handle.net/20.500.12733/1613619>.

RIBEIRO, R. Implementação de um sistema de controle de um pêndulo invertido. Universidade Federal de Itajubá, p. 17–23, 2007. Tese (mestrado). Disponível em: <https://repositorio.unifei.edu.br/jspui/handle/123456789/3386>.

ROCHA, A. F. *Integração associativa em métodos Runge-Kutta*. 79 p. Dissertação (Mestrado) — Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2016. Dissertação (Mestrado em Física).

SAIZAKI, P. M. *Modelagem matemática da hidratação da soja*. 2022. Monografia.

SILVA, G. V. M. da. Controlo nao linear. *Escola superior de tecnologia de Setúbal*, 2006. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/48447248/BINA_sistemas_de_controle_nao_lineares-libre.pdf?1472588398=&response-content-disposition=inline%3B+filename%3DCONTROLO_NAO_LINEAR.pdf&Expires=1723300879&Signature=HQz-~pXnxF1fpAH8n6rEuiejuX7I45PjEUfuH-kgrF9i-ebab7xDu9LVQGjoWE5lsrsze-j3ieOO2vbpC8R0aBMWARrIYYLPiMsyIrXK47vY9AjaRcfR42WroRKhSD1DTg~gDClChqBpm-9053LoWT48xTljAHINwf-naVQgBq~iYxocB554bBYZSztNSAP9LXQeX89lUSV1lEbdiMic74gMVuUDlJUW4Obo1LgVaHIUxVg1r3jDzTNl5qPGRwMPU4sAwUADmd7-sdYCupbpwhgkmUpaEVcxSRLHNg7lMUek6mKVRcq054i81WFMxIi8dOz6F61tBsMhYOHRxhWAOA___&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>.

SILVA, M. D. da. Aplicação da ferramenta google colaboratory para o ensino da linguagem python. In: SBC. *Anais da IV Escola Regional de Engenharia de Software*. 2020. p. 67–76. Disponível em: <https://sol.sbc.org.br/index.php/eres/article/view/13717>.

SLOTINE, J.; LI, W. *Applied Nonlinear Control*. Prentice Hall, 1991. Disponível em: <https://books.google.com.br/books?id=cwpRAAAAMAAJ>.

TABORDA, R. A. *Análise de desempenho de controladores aplicados ao sistema pêndulo invertido*. Toledo: [s.n.], 2014. 58 p. Trabalho de Conclusão de Curso (Graduação).

TAKAMORI, A. et al. Inverted pendulum as low-frequency pre-isolation for advanced gravitational wave detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, v. 582, n. 2, p. 683–692, 2007. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168900207018554>.

TRAN, A.; PEPITONE, P.; CHOI, J. State space and energy based control for an autonomous self-rising inverted pendulum. 12 2019. doi: 10.13140/RG.2.2.36425.70243.

VENDRAMINI, G.; SILVA, P. d. Controle de um pêndulo invertido sobre uma plataforma móvel utilizando pid e mfac (model-free adaptive control). p. 405–414, 2010.

YANG, C.; LI, Z.; LI, J. Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models. *IEEE Transactions on Cybernetics*, v. 43, n. 1, p. 24–36, 2013. doi: 10.1109/TSMCB.2012.2198813.

ZAK, S. H. et al. *Systems and control*. [S.l.]: Oxford University Press New York, 2003. v. 198.

A Runge Kutta

Considerando a equação de estado descrita em (2.3), alguns modelos tratados neste texto são não lineares, cuja solução analítica pode ser difícil de obter. Portanto, pode-se usar um método numérico para obter as formas de onda de todos os estados presentes em (2.5). Portanto, o método de Runge Kutta, apresentado a seguir, foi o escolhido para ser utilizado neste trabalho. Saizaki (2022) e Arrieta, Alviz e Coba (2011) explicam que o método de Runge Kutta é o utilizado pelo comando *ode*45 do *Matlab*, simulações do *Simulink* e também pelo comando *solve_ivp*, presente no módulo *scipy.integrate*, do *python*

Runge Kutta é um método numérico bastante difundido como uma forma de se resolver equações diferencias de ordem 2 e 4. É muito competente em encontrar a solução de problemas com valor inicial, especialmente equações ou sistemas de equações que não possuem solução analítica (GADELHA, 2018).

Conforme Rocha (2016) comenta, a fórmula de Runge Kutta é essencialmente uma média ponderada dos valores de $f(t,x)$ em diferentes pontos do intervalo $t_n \leq t \leq t_{n+1}$ e um passo d. Tal fórmula é expressa por (A.1), descrita como

$$x_{n+1} = x_n + d\frac{k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}}{6}, \tag{A.1}$$

em que

$$\begin{aligned} k_{n1} &= f(t_n,x_n), \\ k_{n2} &= f(t_n + \frac{1}{2}d,x_n + \frac{1}{2}dk_{n1}), \\ k_{n3} &= f(t_n + \frac{1}{2}d,x_n + \frac{1}{2}dk_{n2}), \\ k_{n4} &= f(t_n + d,x_n + dk_{n3}). \end{aligned}$$

A soma de k_{n1}, k_{n2}, k_{n3}, k_{n4} pode ser vista como uma média ponderada dos coeficientes angulares, em que k_{n1} corresponde ao coeficiente angular no início do intervalo, k_{n2} corresponde ao coeficiente angular na metade do intervalo usando a primeira estimativa, k_{n3} corresponde ao coeficiente angular na metade do intervalo usando a segunda estimativa e k_{n4} corresponde ao coeficiente angular no final do intervalo.

A importância de tal método é que, conforme Saizaki (2022) e Arrieta, Alviz e Coba (2011) explicam, o método de Runge Kutta é o utilizado pelo comando *ode*45, do *Matlab*, simulações do *Simulink* e também pelo comando *solve_ivp*, presente no módulo *scipy.integrate*, do *python*. O método em questão é bastante usado na obtenção das soluções das equações diferencias características do SS.

B Integração numérica

No presente trabalho, alguns sinais não possuem uma forma analítica de serem expressas, sendo necessário um método numérico para calcular a integral de uma curva. O método escolhido para resolver este problema foi a regra de Simpson 3/8.

Filho (2007) explica que a 2° regra de Simpson consiste em aproximar uma função $f(x)$ por meio de um polinômio interpolador. O procedimento para aproximar a área de $f(x)$ é expresso por

$$\int_a^b f(x)dx \approx \frac{3d}{8}\sum_{i=0}^{m} c_i y_i, \tag{B.1}$$

em que y_i são os pontos da curva, d é o passo de integração, $c_0 = c_m = 1$, $c_i = 2$ caso i seja múltiplo de 3, e $c_i = 3$ caso i não seja múltiplo de 3. O método numérico de integração (B.1) será empregado para computar a potência média de um sinal neste trabalho. A biblioteca *scipy.integrate*, do *Python*, possui uma implementação desta regra, por meio a função *simpson*. Uma implementação da regra de Simpson 3/8 implementada em *Matlab* é apresentada a seguir

```
function I = regra_simpson(t,u)
%Implementacao da regra de simpson 3/8
%vetor de tempo t
%sinal de interesse u
h = t(2) - t(1);
soma = u(1) + u(end);%Soma o primeiro e ultimo elemento
for c = 2:(length(u)-1)
    %Percorre o vetor da 2 posicao ate a penultima
    if mod(c-1,3) == 0
        soma = soma + 2*u(c);
    else
        soma = soma + 3*u(c);
    end
end
I = soma*3*h/8;
end
```

C Cálculo das expressões matemáticas do pêndulo invertido

```
import numpy as np
from numpy.linalg import matrix_rank, eigvals
from numpy import block, eye, round, zeros
from numpy import array, zeros_like, conj
import sympy as sy
from sympy import Symbol, symbols, Eq, solve, Matrix
from sympy import diff, Subs, simplify, latex
#Calculo das expressoes do pendulo invertido
o,o1,O2,m,M,g,l = sy.symbols(['o','o1','O2','m','M','g','l'])
X2,u = sy.symbols(['X2','u'])
x1,x2,x3,x4 = symbols(['x1','x2','x3','x4'])
I = m*l*l/3
o,o1 = [x1,x2]
derivada_cosseno = -o1*o1*sy.cos(o)-O2*sy.sin(o)
derivada_seno = -o1*o1*sy.sin(o)+O2*sy.cos(o)
H,V = [m*X2 + m*l*derivada_seno, m*g + m*l*derivada_cosseno]
#Equacoes do pendulo invertido
e1 = Eq(I*O2,simplify(V*l*sy.sin(o) - H*l*sy.cos(o)))
e2 = Eq(M*X2,simplify(u-H))
#Acha o valor das expressoes de \ddot{x} e \ddot{\theta}
sol = sy.solve([e1,e2],[X2,O2])
#Simplifica, para ficar melhor de visualizar
O2,X2 = [simplify(sol[O2]),simplify(sol[X2])]
#Representacao nao linear do pendulo invertido
xdot_nao_linear = sy.Matrix([[x2],[O2],[x4],[X2]])
#Separa xdot_nao_linear nas funcoes F(x) e G(x)
F = simplify(Subs(xdot_nao_linear,u,0),100)
G = simplify(Subs(xdot_nao_linear-F,u,1),100)
#Linearizacao ja  cobiana
A = F.jacobian([x1,x2,x3,x4]).subs([(x1,0),(x2,0),(x3,0),(x4,0)])
B = G.subs([(x1,0),(x2,0),(x3,0),(x4,0)])
#Calcular as matrizes A e B com os valores usados no trabalho
#Converte para float
A = array(A.subs([(m,0.1),(M,2),(g,9.8),(l,0.5)]),dtype = float)
B = array(B.subs([(m,0.1),(M,2),(g,9.8),(l,0.5)]),dtype = float)
#Calcula os autovalores da matriz A
Polos = eigvals(A)
```

D Simulações lineares sem controladores do pêndulo invertido

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#!pip install control
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
import control as ct
from scipy.signal import lsim, lti
from numpy.linalg import matrix_rank
from numpy import transpose, array, pi, arange, ones_like, eye
from matplotlib.pyplot import plot, grid, title, figure
from matplotlib.pyplot import xlabel, ylabel, savefig, subplot, xlim
m,M,g,l = [0.1,2,9.8,0.5]
A = array([[0, 1, 0, 0], [3*(2*M*g + 2*g*m)/(2*l*(4*M + m)), 0, 0, 0],
    [0, 0, 0, 1], [-3*g*m/(4*M + m), 0, 0, 0]])
B = array([[0], [-3/(l*(4*M + m))], [0], [4/(4*M + m)]])
C = array([[1,0,0,0],[0,0,1,0]])
D = np.zeros_like(C@B)
#monta a matriz de controlabilidade e observabilidade
Pc = ct.ctrb(A,B)
Q = ct.obsv(A,eye(4))
#Verifica o posto
rank = array([matrix_rank(Pc),matrix_rank(Q)])
#Monta o sistema
sys = lti(A,B,C,D)
tempo,h = [10,1e-4]
t = arange(0,tempo+h,h)
u = ones_like(t)
x0 = array([0,0,0,0])
t,y,x = lsim(sys, U = u, X0 = x0,T = t)
figure(figsize=(12,10))
subplot(2,1,1),plot(t,y[:,0]*180/pi)
title('Estado $x_1$')
ylabel('Angulo $(\\circ)$')
grid(True)
subplot(2,1,2),plot(t,y[:,1])
title('Estado $x_3$')
xlabel('Tempo $(s)$')
ylabel('Posicao $(m)$')
grid(True)
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/angulo_sem_controladores2.pdf')
```

E Simulações lineares com controlador estabilizante do pêndulo invertido

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from scipy.signal import lsim, lti, place_poles
from numpy import transpose, array, pi, arange, ones_like
from matplotlib.pyplot import plot, grid, title, figure
from matplotlib.pyplot import xlabel, ylabel, savefig, subplot, xlim
m,M,g,l = [0.1,2,9.8,0.5]
A = array([[0, 1, 0, 0], [3*(2*M*g + 2*g*m)/(2*l*(4*M + m)), 0, 0, 0],
    [0, 0, 0, 1], [-3*g*m/(4*M + m), 0, 0, 0]])
B = array([[0], [-3/(l*(4*M + m))], [0], [4/(4*M + m)]])
C = array([[1,0,0,0],[0,0,1,0]])
D = np.zeros_like(C@B)
r1 = -2+3.3119598956507454j
r2 = -2-3.3119598956507454j
#Define polos
p1 = array([r1,r2,-6,-8]);
p2 = -array([3,4,5,6])
p3 = -array([6,8,10,12])
#Monta os ganhos
K1 = place_poles(A,B,p1).gain_matrix
K2 = place_poles(A,B,p2).gain_matrix
K3 = place_poles(A,B,p3).gain_matrix
#Monta o sistema
sys1 = lti(A-B@K1,B,C,D)
sys2 = lti(A-B@K2,B,C,D)
sys3 = lti(A-B@K3,B,C,D)
tempo,h = [6,1e-4]
t = arange(0,tempo+h,h)
u = ones_like(t)
x0 = array([0,0,0,0])
t,y,X1 = lsim(sys1, U = u, X0 = x0,T = t)
t,y,X2 = lsim(sys2, U = u, X0 = x0,T = t)
t,y,X3 = lsim(sys3, U = u, X0 = x0,T = t)
u1 = -transpose(K1@transpose(X1))
u2 = -transpose(K2@transpose(X2))
u3 = -transpose(K3@transpose(X3))
figure(figsize=(12,10))
subplot(3,1,1),plot(t,X1[:,0]*180/pi),plot(t,X2[:,0]*180/pi),plot(t,X3
   [:,0]*180/pi)
```

```
title('Estado $x_1$')
xlim((0,4))
ylabel('Angulo $(\\circ)$')
grid(True)
subplot(3,1,2),plot(t,X1[:,2]),plot(t,X2[:,2]),plot(t,X3[:,2])
title('Estado $x_3$')
xlim((0,3.5))
ylabel('Posicao $(m)$')
grid(True)
subplot(3,1,3),plot(t,u1),plot(t,u2),plot(t,u3)
title('Sinal de controle $u$')
xlim((0,2.5))
ylabel('Esforco de controle $(N)$')
grid(True)
xlabel('Tempo $(s)$')
xlim((0,3))
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/estados_conjuntos.pdf')
```

F Simulações lineares com controlador PI do pêndulo invertido

```
#Na primeira vez q for rodar, descomentar essas linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from scipy.signal import lsim, lti, place_poles
from numpy import transpose, array, pi, arange, ones_like, block,
   zeros
from matplotlib.pyplot import plot, grid, title, figure
from matplotlib.pyplot import xlabel, ylabel, savefig, subplot, xlim
m,M,g,l = [0.1,2,9.8,0.5]
A = array([[0, 1, 0, 0], [3*(2*M*g + 2*g*m)/(2*l*(4*M + m)), 0, 0, 0],
    [0, 0, 0, 1], [-3*g*m/(4*M + m), 0, 0, 0]])
B = array([[0], [-3/(l*(4*M + m))], [0], [4/(4*M + m)]])
C = array([0,0,1,0])
D = array([0])
Atau = block([[A,zeros((4,1))],[-C,0]])
Btau = block([[B],[0]])
r1 = -2+3.3119598956507454j
r2 = -2-3.3119598956507454j
#Define polos
p1 = array([r1,r2,-6,-8,-10]);
p2 = -array([3,4,5,6,30])
p3 = -array([6,8,10,12,60])
#Monta os ganhos
K1 = place_poles(Atau,Btau,p1).gain_matrix
K2 = place_poles(Atau,Btau,p2).gain_matrix
K3 = place_poles(Atau,Btau,p3).gain_matrix
#Monta o sistema
Br = block([[zeros((4,1))],[1]])
Ctau = block([[1,0,0,0,0],[0,0,1,0,0]])
sys1 = lti(Atau-Btau@K1,Br,Ctau,Ctau@Br)
sys2 = lti(Atau-Btau@K2,Br,Ctau,Ctau@Br)
sys3 = lti(Atau-Btau@K3,Br,Ctau,Ctau@Br)
tempo,h = [7,1e-4]
t = arange(0,tempo+h,h)
u = ones_like(t)
x0 = array([0,0,0,0,0])
t,y,X1 = lsim(sys1, U = u, X0 = x0,T = t)
t,y,X2 = lsim(sys2, U = u, X0 = x0,T = t)
t,y,X3 = lsim(sys3, U = u, X0 = x0,T = t)
u1 = -transpose(K1@transpose(X1))
```

```
u2 = -transpose(K2@transpose(X2))
u3 = -transpose(K3@transpose(X3))
figure(figsize=(12,10))
subplot(3,1,1),plot(t,X1[:,0]*180/pi),plot(t,X2[:,0]*180/pi),plot(t,X3
    [:,0]*180/pi)
title('Estado $x_1$')
xlim((0,3.0))
ylabel('Angulo $(\\circ)$')
grid(True)
subplot(3,1,2),plot(t,X1[:,2]),plot(t,X2[:,2]),plot(t,X3[:,2]),plot(t,
    ones_like(t),color = 'deeppink')
title('Estado $x_3$')
xlim((0,3.0))
ylabel('Posicao $(m)$')
grid(True)
subplot(3,1,3),plot(t,u1),plot(t,u2),plot(t,u3)
title('Sinal de controle $u$')
xlim((0,2.5))
ylabel('Esforco de controle $(N)$')
grid(True)
xlabel('Tempo $(s)$')
xlim((0,1.4))
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/estados_conjuntos_PI.pdf')
```

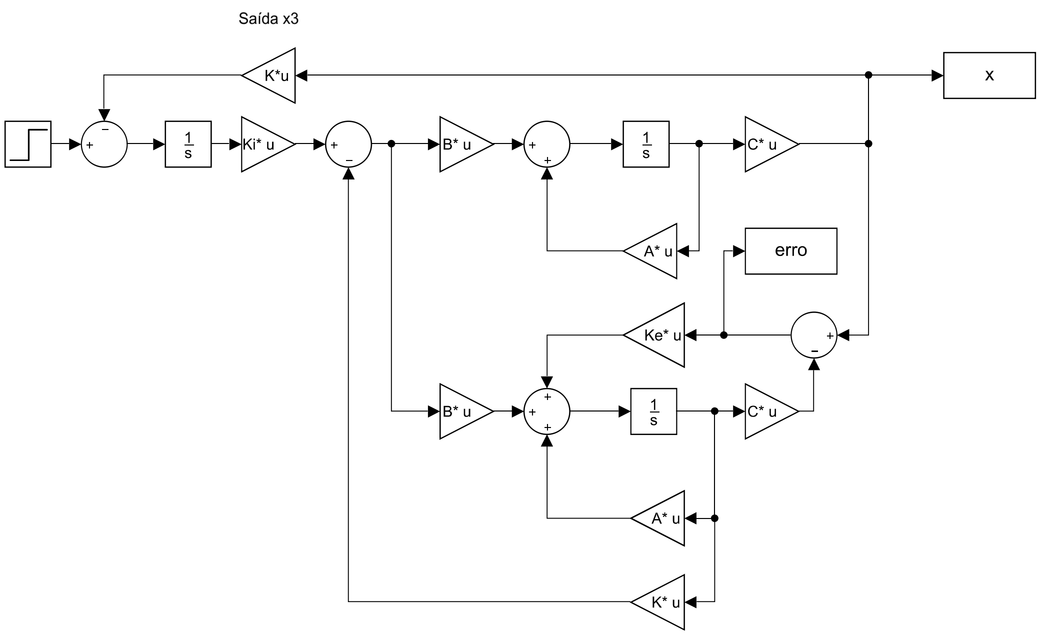
G Simulações lineares com controlador seguidor de referência e observador de estados do pêndulo invertido

```
clear all
clc
close all

m = 0.1;
M = 2;
l = 0.5;
g = 9.8;

A = [0 1 0 0;3*(M*g+m*g)/(4*M*l+l*m) 0 0 0;0 0 0 1;-3*m*g/(4*M+m) 0 0
    0];
B = [0 -3/(4*M*l + l*m) 0 4/(4*M+m)]';
C = eye(4);
D = 0;
%C = [1 0 0 0;0 0 1 0];
%D = [0 0 0 0]';
[zeta,wn] = zeta_wn(15,2);
r1 = -zeta*wn + j*wn*sqrt(1-zeta^2);

a = 10;%fator q multiplica a parte real dos polos
Po = [a*real(r1) + j*imag(r1) a*real(r1) - j*imag(r1) -6*a -8*a];

Ke = place(A',C',Po)';
Atau = [A zeros(4,1);-[0 0 1 0] 0];
Btau = [B;0];
K = place(Atau,Btau,[r1 r1' -6 -8 -10]);
Ki = -K(end);
K = K(1:4);




tempo = 6;
h = 1e-4;
x0 = 0;
sim('observador_PI');

x1 = x(:,1);
x3 = x(:,3);
```

```
e1 = erro(:,1);
e3 = erro(:,3);
t = tout;

%%especificacoes de desempenho. Mudar tempo p 100s

fprintf('Especificacoes de x1')
stepinfo(x1,t)

fprintf('Especificacoes de x3')
stepinfo(x3,t)


% %Salvar graficos
% subplot(2,1,1),plot(t,x1*180/pi)
%
% grid on
% ylabel('Angulo (\circ)')
% subplot(2,1,2),plot(t,x3)
% grid on
% ylabel('Posicao (m)')
% xlabel('Tempo (s)')
%
% %Salvar graficos
% set(gcf,'Units','inches');
%
% screenposition = get(gcf,'Position');
%
% set(gcf,...
%     'PaperPosition',[0 0 screenposition(3:4)],...
%     'PaperSize',[screenposition(3:4)]);
%
% print -dpdf -painters obsv_ordem_completa_PI_estados2
%
% figure
% subplot(2,1,1),plot(t,e1*180/pi)
% xlim([0,1/2])
% ylim([-1 20])
% ylabel('Erro (\circ)')x
% grid on
%
% subplot(2,1,2),plot(t,e3)
% xlim([0,1/2])
% ylim([-.1 0.7])
% grid on
```

```
% ylabel('Erro (m)')
% xlabel('Tempo (s)')
%
% set(gcf,'Units','inches');
%
% screenposition = get(gcf,'Position');
%
% set(gcf,...
%     'PaperPosition',[0 0 screenposition(3:4)],...
%     'PaperSize',[screenposition(3:4)]);
%
% print -dpdf -painters obsv_ordem_completa_PI_erros2
%
```

Figura 24 – Modelo no *simulink* do pêndulo invertido com controlador PI e observador de estados.



Fonte: Autoria própria.

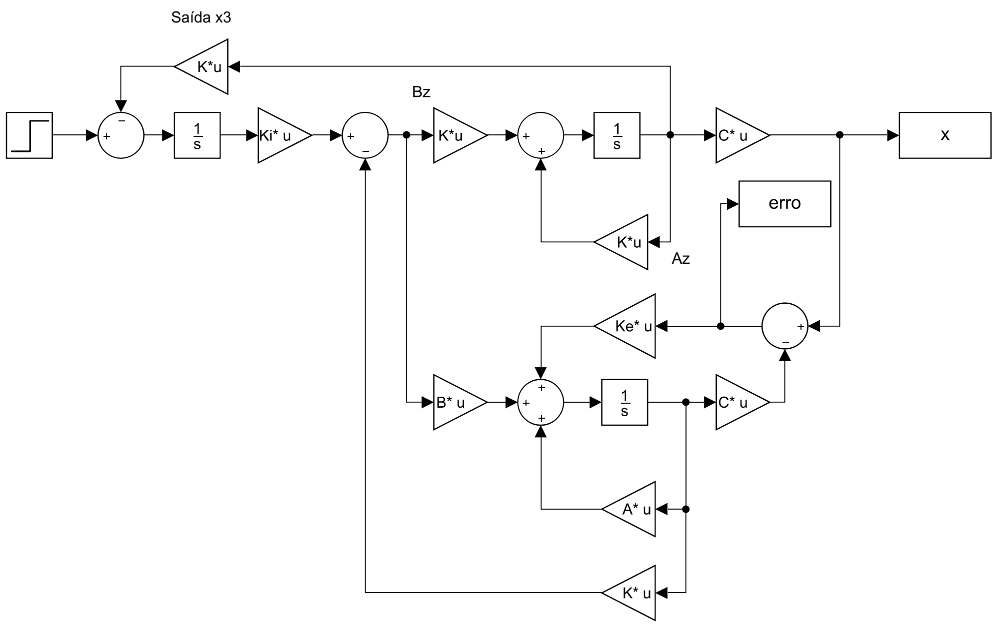
H Simulações lineares com controlador estabilizante, observador de estados e erros presentes nas medições dos parâmetros

```
clear all
clc
close all

m = 0.1;
M = 2;
l = 0.5;
g = 9.8;

A = [0 1 0 0;3*(M*g+m*g)/(4*M*l+l*m) 0 0 0;0 0 0 1;-3*m*g/(4*M+m) 0 0
    0];
B = [0 -3/(4*M*l + l*m) 0 4/(4*M+m)]';
C = eye(4);

[zeta,wn] = zeta_wn(15,2);
r1 = -zeta*wn + j*wn*sqrt(1-zeta^2);

a = 5;%fator q multiplica a parte real dos polos
Po = [a*real(r1) + j*imag(r1) a*real(r1) - j*imag(r1) -6*a -8*a];

Ke = place(A',C',Po)';

Atau = [A zeros(4,1);-[0 0 1 0] 0];
Btau = [B;0];
K = place(Atau,Btau,[r1 r1' -6 -8 -10]);
Ki = -K(end);
K = K(1:4);


x0 = [0.3 0 0.5 0];

fator = -5/100;
m = 0.1*(1+fator);
M = 2*(1+fator);
l = 0.5*(1+fator);
g = 9.8;

Az = [0 1 0 0;3*(M*g+m*g)/(4*M*l+l*m) 0 0 0;0 0 0 1;-3*m*g/(4*M+m) 0 0
     0];
```

```
Bz = [0 -3/(4*M*l + l*m) 0 4/(4*M+m)]';

tempo = 100;
h = 1e-4;
sim('observador_simulacao_erros_completo_PI')

%informacoes.
%Mudar tempo p 100s, para melhorar precisao
fprintf('Estado x1\n')
disp(stepinfo(x(:,1),tout))

fprintf('Estado x3\n')
disp(stepinfo(x(:,3),tout))

% %Descricao dos graficos
% subplot(2,1,1),plot(tout,x(:,1)*180/pi)
% grid on
% ylabel('Angulo (\circ)');
% subplot(2,1,2),plot(tout,x(:,3))
% %title('Estados')
% grid on
% ylabel('Posicao (m)');
% xlabel('Tempo (s)')
%
% %Salvar graficos
% set(gcf,'Units','inches');
%
% screenposition = get(gcf,'Position');
%
% set(gcf,...
%     'PaperPosition',[0 0 screenposition(3:4)],...
%     'PaperSize',[screenposition(3:4)]);
%
% print -dpdf -painters obsv_ordem_completa_PI_estados_erro_cinco
%
%
%
% figure
% subplot(2,1,1),plot(tout,erro(:,1)*180/pi)
% %xlim([0 1])
% %title('Erro')
% grid on
% ylabel('Erro (\circ)');
% subplot(2,1,2),plot(tout,erro(:,3))
% grid on
% %ylim([-0.05 0.5+0.05])
% %xlim([0 1])
```

```
% ylabel('Erro (m)');
% xlabel('Tempo (s)')
%
% %Salvar graficos
% set(gcf,'Units','inches');
%
% screenposition = get(gcf,'Position');
%
% set(gcf,...
%     'PaperPosition',[0 0 screenposition(3:4)],...
%     'PaperSize',[screenposition(3:4)]);
%
% print -dpdf -painters obsv_ordem_completa_PI_erro_erro_cinco
```

Figura 25 – Modelo no *simulink* do pêndulo invertido com controlador seguidor de referência e erros de medição.



Fonte: Autoria própria.

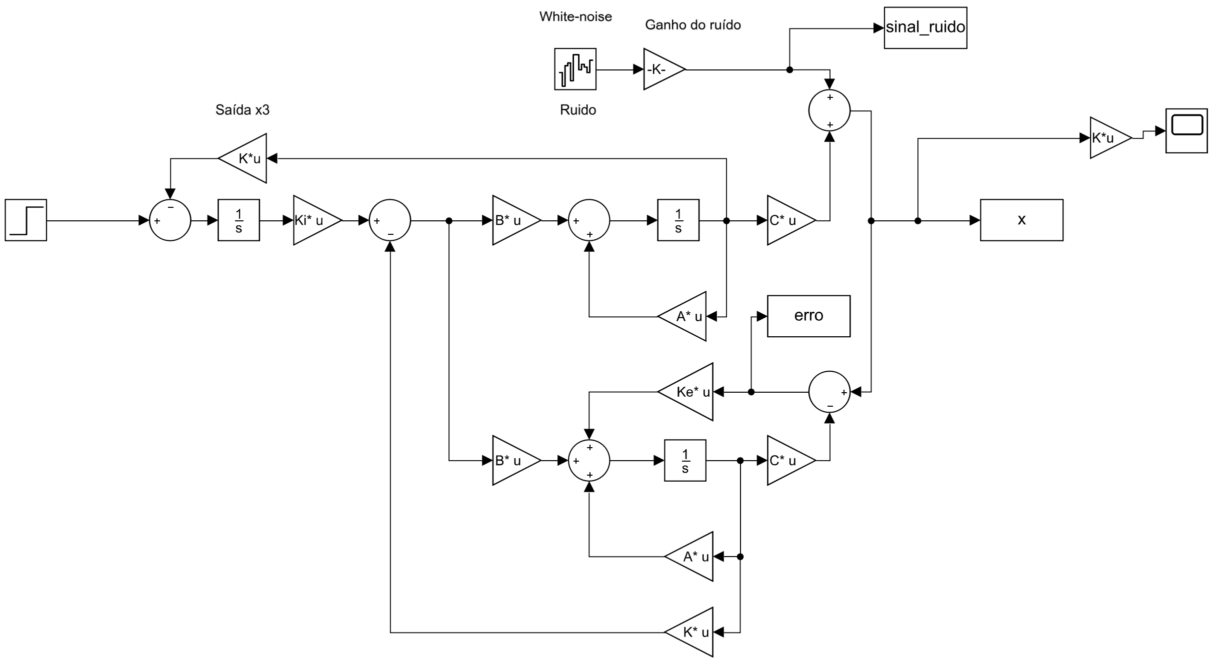
I Simulações lineares com controlador seguidor de referência, observador de estados e ruídos presentes nas saídas

```
clear all
clc
close all
m = 0.1;
M = 2;
l = 0.5;
g = 9.8;

A = [0 1 0 0;3*(M*g+m*g)/(4*M*l+l*m) 0 0 0;0 0 0 1;-3*m*g/(4*M+m) 0 0
    0];
B = [0 -3/(4*M*l + l*m) 0 4/(4*M+m)]';

C = eye(4);
%D = [0 0 0 0]';
[zeta,wn] = zeta_wn(15,2);
r1 = -zeta*wn + j*wn*sqrt(1-zeta^2);

a = 2;%fator q multiplica a parte real dos polos
Po = [a*real(r1) + j*imag(r1) a*real(r1) - j*imag(r1) -6*a -8*a];

Ke = place(A',C',Po)';

P = [r1 r1' -6 -8];

Atau = [A zeros(4,1);-[0 0 1 0] 0];
Btau = [B;0];
K = place(Atau,Btau,[r1 r1' -6 -8 -10]);
Ki = -K(end);
K = K(1:4);




x0 = [0.3 0 0.5 0];

ruido = 10^-2;
```

```
tempo = 20;
h = 1e-4;

sim('observador_simulacao_ruidos_completo_PI')

x1 = x(:,1);
x3 = x(:,3);
e1 = erro(:,1);
e3 = erro(:,3);
% %Descricao dos graficos
% subplot(2,1,1),plot(tout,x(:,1)*180/pi)
% grid on
% ylabel('Angulo (\circ)');
% subplot(2,1,2),plot(tout,x(:,3))
%
% grid on
% ylabel('Posicao (m)');
% xlabel('Tempo (s)')

% %Salvar graficos
% set(gcf,'Units','inches');
%
% screenposition = get(gcf,'Position');
%
% set(gcf,...
%     'PaperPosition',[0 0 screenposition(3:4)],...
%     'PaperSize',[screenposition(3:4)]);
%
% print -dpdf -painters
    obsv_ordem_completa_estados_ruido_2_polos_cinco_PI
%
%
%
% figure
% subplot(2,1,1),plot(tout,erro(:,1)*180/pi)
% %xlim([0 2])
% grid on
% ylabel('Erro (\circ)');
% subplot(2,1,2),plot(tout,erro(:,3))
% grid on
% %xlim([0 2])
% ylabel('Erro (m)');
% xlabel('Tempo (s)')
%
% %Salvar graficos
% set(gcf,'Units','inches');
%
```

```
% screenposition = get(gcf,'Position');
%
% set(gcf,...
%     'PaperPosition',[0 0 screenposition(3:4)],...
%     'PaperSize',[screenposition(3:4)]);
%
% print -dpdf -painters
   obsv_ordem_completa_erro_ruido_2_polos_cinco_PI


x1 = x(:,1);



x3 = x(:,3);
t = tout;

Pruido = regra_simpson(t,sinal_ruido.^2)/(t(end)-t(1));

Px1 = regra_simpson(t,x1.^2)/(t(end)-t(1));

Px3 = regra_simpson(t,x3.^2)/(t(end)-t(1));

SNR_x1 = 20*log10(Px1/Pruido)


SNR_x3 = 20*log10(Px3/Pruido)

%%informacoes.
%%Mudar tempo p 300s, para melhorar precisao
% fprintf('Estado x1 real\n')
% disp(stepinfo(x_real(:,1),tout))
%
% fprintf('Estado x1 obsv\n')
% disp(stepinfo(x_obsv(:,1),tout))
%
%
% fprintf('Estado x3 real\n')
% disp(stepinfo(x_real(:,2),tout))
%
% fprintf('Estado x3 obsv\n')
% disp(stepinfo(x_obsv(:,2),tout))
```

Figura 26 – Modelo no *simulink* do pêndulo invertido com controlador seguidor de referência e adição de ruídos.



Fonte: Autoria própria.

J Simulações não lineares sem controladores do pêndulo invertido

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from scipy.integrate import solve_ivp
from numpy import transpose, array, pi, arange, ones_like, sin, cos
from matplotlib.pyplot import plot, grid, title, figure
from matplotlib.pyplot import xlabel, ylabel, savefig, subplot, xlim
def pendulo_n_linear(t,x):
m,M,g,l = [0.1,2,9.8,0.5]
x1,x2,x3,x4 = x
f2 = 3*(2*M*g*sin(x1) + 2*g*m*sin(x1) - l*m*x2**2*sin(2*x1))/(2*l*(4*M
     + 3*m*sin(x1)**2 + m))
f4 = (-3*g*m*sin(2*x1) + 8*l*m*x2**2*sin(x1))/(2*(4*M + 3*m*sin(x1)**2
     + m))
g2 = -3*cos(x1)/(l*(4*M + 3*m*sin(x1)**2 + m))
g4 = 4/(4*M + 3*m*sin(x1)**2 + m)
u = 0
xdot = [x2,f2+g2*u,x4,f4+g4*u]
return xdot
tempo,h = [30,1e-4]
t = arange(0,tempo+h,h)
iter = [0,tempo]
x0 = array([5*pi/180,0,0,0])
sol = solve_ivp(pendulo_n_linear,t_span = iter,y0 = x0,t_eval=t,
    method = 'RK45')
t,y = sol.t,(sol.y)
x1,x2,x3,x4 = y
#Plota os graficos
figure(figsize=(12,10))
subplot(2,1,1),plot(t,x1*180/pi)
title('Estado $x_1$')
grid(True)
ylabel('Angulo $(\\circ)$')
subplot(2,1,2),plot(t,x3)
title('Estado $x_3$')
grid(True)
ylabel('Posicao $(m)$')
xlabel('Tempo $(s)$')
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/simulacao_angulo_nao_linear2.pdf')
```

K Simulações não lineares com controlador estabilizante do pêndulo invertido

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from scipy.integrate import solve_ivp
from numpy import transpose, array, pi, arange, ones_like, sin, cos
from matplotlib.pyplot import plot, grid, title, figure
from matplotlib.pyplot import xlabel, ylabel, savefig, subplot, xlim
def pendulo_n_linear(t,x,k1,k2,k3,k4):
m,M,g,l = [0.1,2,9.8,0.5]
x1,x2,x3,x4 = x
f2 = 3*(2*M*g*sin(x1) + 2*g*m*sin(x1) - l*m*x2**2*sin(2*x1))/(2*l*(4*M
    + 3*m*sin(x1)**2 + m))
f4 = (-3*g*m*sin(2*x1) + 8*l*m*x2**2*sin(x1))/(2*(4*M + 3*m*sin(x1)**2
    + m))
g2 = -3*cos(x1)/(l*(4*M + 3*m*sin(x1)**2 + m))
g4 = 4/(4*M + 3*m*sin(x1)**2 + m)
v = -(k1*x1 + k2*x2 + k3*x3 + k4*x4)
u = (v-f2)/g2
xdot = [x2,f2+g2*u,x4,f4+g4*u]
return xdot
tempo,h = [4,1e-4]
t = arange(0,tempo+h,h)
iter = [0,tempo]
x0 = array([5*pi/180,0,0,0])
k1,k2,k3,k4 = [247.17439707,61.17861094,98.97921195,55.31791641]
sol = solve_ivp(pendulo_n_linear,t_span = iter,y0 = x0,t_eval=t,
   method = 'RK45', args = (k1,k2,k3,k4))
t,y = sol.t,sol.y
x1,x2,x3,x4 = y
u = -(k1*x1 + k2*x2 + k3*x3 + k4*x4)
#Plota os graficos
figure(figsize=(12,12))
subplot(3,1,1),plot(t,x1*180/pi)
title('Estado $x_1$')
grid(True)
ylabel('Angulo $(\\circ)$')
subplot(3,1,2),plot(t,x3)
title('Estado $x_3$')
grid(True)
ylabel('Posicao $(m)$')
```

```
subplot(3,1,3),plot(t,u)
xlim((0,1.5))
title('Sinal de controle $v$')
grid(True)
ylabel('Esforco de controle $(N)$')
xlabel('Tempo $(s)$')
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/simulacao_angulo_nao_lineadr2.pdf')
```

L Simulações não lineares do pêndulo invertido com controlador PI

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from scipy.integrate import solve_ivp
from numpy import transpose, array, pi, arange, ones_like, sin, cos
from matplotlib.pyplot import plot, grid, title, figure
from matplotlib.pyplot import xlabel, ylabel, savefig, subplot, xlim
def pendulo_n_linear(t,x,k1,k2,k3,k4,k5):
m,M,g,l = [0.1,2,9.8,0.5]
x1,x2,x3,x4,x5 = x
f2 = 3*(2*M*g*sin(x1) + 2*g*m*sin(x1) - l*m*x2**2*sin(2*x1))/(2*l*(4*M
    + 3*m*sin(x1)**2 + m))
f4 = (-3*g*m*sin(2*x1) + 8*l*m*x2**2*sin(x1))/(2*(4*M + 3*m*sin(x1)**2
    + m))
g2 = -3*cos(x1)/(l*(4*M + 3*m*sin(x1)**2 + m))
g4 = 4/(4*M + 3*m*sin(x1)**2 + m)
v = -(k1*x1 + k2*x2 + k3*x3 + k4*x4 - k5*x5)
u = (v-f2)/g2
xdot = [x2,f2+g2*u,x4,f4+g4*u,x3-1]
return xdot
tempo,h = [4,1e-4]
t = arange(0,tempo+h,h)
iter = [0,tempo]
x0 = array([5*pi/180,0,0,0,0])
k1,k2,k3,k4,k5 =
   [858.96050648,228.82445929,652.15837605,286.53668893,-989.7921195]
sol = solve_ivp(pendulo_n_linear,t_span = iter,y0 = x0,t_eval=t,
   method = 'RK45', args = (k1,k2,k3,k4,k5))
t,y = sol.t,(sol.y)
x1,x2,x3,x4,x5 = y
u = -(k1*x1 + k2*x2 + k3*x3 + k4*x4 - k5*x5)
#Plota os graficos
figure(figsize=(12,12))
subplot(3,1,1),plot(t,x1*180/pi)
title('Estado $x_1$')
grid(True)
ylabel('Angulo $(\\circ)$')
subplot(3,1,2),plot(t,x3,t,ones_like(t))
title('Estado $x_3$')
grid(True)
ylabel('Posicao $(m)$')
```

```
subplot(3,1,3),plot(t,u)
xlim((0,2))
title('Sinal de controle $v$')
grid(True)
ylabel('Esforco de controle $(N)$')
xlabel('Tempo $(s)$')
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/simulacao_angulo_nao_lineaddr2.pdf'
    )
```

M Simulações não lineares do pêndulo invertido considerando relação entre esforço de controle e especificações de desempenho

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from numpy import pi, transpose, sin, cos, arange
from numpy import zeros_like, array, ones_like
from scipy.signal import place_poles
from scipy.integrate import solve_ivp, simpson
from matplotlib.pyplot import plot, grid, xlabel, ylabel, title
from matplotlib.pyplot import figure, savefig, subplot, xlim
def pendulo_n_linear_custo(t,x,K):
m,M,g,l = [0.1,2,9.8,0.5]
k1,k2,k3,k4,k5 = -K
x1,x2,x3,x4,x5 = x
f2 = 3*(2*M*g*sin(x1) + 2*g*m*sin(x1) - l*m*x2**2*sin(2*x1))/(2*l*(4*M
    + 3*m*sin(x1)**2 + m))
f4 = (-3*g*m*sin(2*x1) + 8*l*m*x2**2*sin(x1))/(2*(4*M + 3*m*sin(x1)**2
    + m))
g2 = -3*cos(x1)/(l*(4*M + 3*m*sin(x1)**2 + m))
g4 = 4/(4*M + 3*m*sin(x1)**2 + m)
v = -(k1*x1 + k2*x2 + k3*x3 + k4*x4 + k5*x5)
u = (v-f2)/g2
xdot = [x2,f2+g2*u,x4,f4+g4*u,1-x3]
return xdot
m,M,g,l = [0.1,2,9.8,0.5]
A = array([[0, 1, 0, 0, 0],[3*(2*M*g + 2*g*m)/(2*l*(4*M + m)), 0, 0,
   0, 0],[0, 0, 0, 1, 0],[-3*g*m/(4*M + m), 0, 0, 0, 0],[0, 0, -1, 0,
   0]])
B = array([[0],[-3/(l*(4*M + m))],[0],[4/(4*M + m)],[0]])
r1 = -2 + 3.311959895650745j;
r2 = -2 - 3.311959895650745j;
P1 = array([r1,r2,-0.9,-1,-2])
P2 = -array([1,2,3,4,8])
P3 = array([-0.7,-0.8,-10,r1,r2])
K1 = place_poles(A,B,P1).gain_matrix
K2 = place_poles(A,B,P2).gain_matrix
K3 = place_poles(A,B,P3).gain_matrix
tempo,h = [50,1e-4]
t = arange(0,tempo+h,h)
```

```
x0 = array([5*pi/180,0,0,0,0])
iter = array([0,tempo])
sol = solve_ivp(pendulo_n_linear_custo,t_span = [0,tempo],y0 = x0,
    t_eval=t, method = 'RK45', args = (K1))
t,X1 = sol.t,np.transpose(sol.y)
sol = solve_ivp(pendulo_n_linear_custo,t_span = [0,tempo],y0 = x0,
    t_eval=t, method = 'RK45', args = (K2))
t,X2 = sol.t,np.transpose(sol.y)
sol = solve_ivp(pendulo_n_linear_custo,t_span = [0,tempo],y0 = x0,
    t_eval=t, method = 'RK45', args = (K3))
t,X3 = sol.t,np.transpose(sol.y)
u1 = (K1@(X1.T)).T
u2 = (K2@(X2.T)).T
u3 = (K3@(X3.T)).T
figure(figsize=(12,10))
subplot(3,1,1),plot(t,X1[:,0]*180/pi),plot(t,X2[:,0]*180/pi),plot(t,X3
    [:,0]*180/pi)
title('Estado $x_1$')
xlim((0,10))
ylabel('Angulo $(\\circ)$')
grid(True)
subplot(3,1,2),plot(t,X1[:,2]),plot(t,X2[:,2]),plot(t,X3[:,2]),plot(t,
    ones_like(t),color = 'deeppink')
title('Estado $x_3$')
xlim((0,12))
ylabel('Posicao $(m)$')
grid(True)
subplot(3,1,3),plot(t,u1),plot(t,u2),plot(t,u3)
title('Sinal de controle $v$')
xlim((0,2.5))
ylabel('Esforco de controle $(N)$')
grid(True)
xlabel('Tempo $(s)$')
xlim((0,1.5))
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/estados_n_linear_conjuntos_custo.
    pdf')
#Calcular a potencia do esforco de controle
pot_media = simpson([(u1**2).T,(u2**2).T,(u3**2).T],x = t)/(t[-1]-t
    [0])
print(f'A potencia media para o conjunto 1 vale {pot_media[0]}')
print(f'A potencia media para o conjunto 2 vale {pot_media[1]}')
print(f'A potencia media para o conjunto 3 vale {pot_media[2]}')
```

N Simulações não lineares do pêndulo invertido com linearização entrada-saída e dinâmica zero

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from numpy import pi, transpose, sin, cos, arange, array, zeros_like
from scipy.integrate import solve_ivp, simpson
from matplotlib.pyplot import plot, grid, xlabel, figure, title
from matplotlib.pyplot import ylabel, savefig, subplot, show, xlim
def pendulo_n_linear(t,x):
m,M,g,l = [0.1,2,9.8,0.5]
x1,x2,x3,x4 = x
f2 = 3*(2*M*g*sin(x1) + 2*g*m*sin(x1) - l*m*x2**2*sin(2*x1))/(2*l*(4*M
    + 3*m*sin(x1)**2 + m))
f4 = (-3*g*m*sin(2*x1) + 8*l*m*x2**2*sin(x1))/(2*(4*M + 3*m*sin(x1)**2
    + m))
g2 = -3*cos(x1)/(l*(4*M + 3*m*sin(x1)**2 + m))
g4 = 4/(4*M + 3*m*sin(x1)**2 + m)
k1,k2,k3,k4 = [247.17439707,61.17861094,98.97921195,55.31791641]
v = -(k1*x1 + k2*x2 + k3*x3 + k4*x4)
u = (v-f2)/g2
xdot = [x2,f2+g2*u,x4,f4+g4*u]
return xdot
def pendulo_alves(t,x):
x1,x2,x3,x4 = x
m,M,g,l = [0.1,2,9.8,0.5]
f2 = (3*(- l*m*cos(x1)*sin(x1)*x2**2 + g*m*sin(x1) + M*g*sin(x1)))/(M*
   l + l*m + 3*M*l*cos(x1)**2 + 3*M*l*sin(x1)**2 + 3*l*m*sin(x1)**2);
f4 = (3*l*m*x2**2*sin(x1)**3 - 3*g*m*cos(x1)*sin(x1) + l*m*x2**2*sin(
   x1) + 3*l*m*x2**2*cos(x1)**2*sin(x1))/(M + m + 3*m*sin(x1)**2 + 3*M
   *cos(x1)**2 + 3*M*sin(x1)**2);
g2 = - (3*(l*m*cos(x1)*sin(x1)*x2**2 + cos(x1) - g*m*sin(x1) - M*g*sin
   (x1)))/(M*l + l*m + 3*M*l*cos(x1)**2 + 3*M*l*sin(x1)**2 + 3*l*m*sin
   (x1)**2) - (3*(- l*m*cos(x1)*sin(x1)*x2**2 + g*m*sin(x1) + M*g*sin(
   x1)))/(M*l + l*m + 3*M*l*cos(x1)**2 + 3*M*l*sin(x1)**2 + 3*l*m*sin(
   x1)**2);
g4 = (3*cos(x1)**2 + 3*sin(x1)**2 + 3*l*m*x2**2*sin(x1)**3 - 3*g*m*cos
   (x1)*sin(x1) + l*m*x2**2*sin(x1) + 3*l*m*x2**2*cos(x1)**2*sin(x1) +
    1)/(M + m + 3*m*sin(x1)**2 + 3*M*cos(x1)**2 + 3*M*sin(x1)**2) -
   (3*l*m*x2**2*sin(x1)**3 - 3*g*m*cos(x1)*sin(x1) + l*m*x2**2*sin(x1)
```

```
    + 3*l*m*x2**2*cos(x1)**2*sin(x1))/(M + m + 3*m*sin(x1)**2 + 3*M*
    cos(x1)**2 + 3*M*sin(x1)**2);
k1,k2,c1,c2,c3,e1,e3 = [100,20,10,1,20,x3,x4]
v = -k1*x1 - k2*x2
u = (v-f2)/g2
v3 = (g2*(-c1*e1-c3*x4-c2*f4) - c2*g4*(v-f2))/(c2*g2*g4)
u1 = u - v3
xdot = [x2,f2+g2*u1,x4,f4+g4*u1]
return xdot
tempo,h = [500,1e-4]
t = arange(0,tempo+h,h)
x0 = [5*pi/180,0,0,0]
#calcular os valores do alves
sol = solve_ivp(pendulo_alves,t_span=[0,tempo],t_eval=t,y0 = x0,
    method='RK45')
x_alves = sol.y
x1_alves,x2_alves,x3_alves,x4_alves = x_alves
v_alves = -100*x1_alves - 20*x2_alves
#calcular os meus valores
sol = solve_ivp(pendulo_n_linear,t_span=[0,tempo],t_eval=t,y0 = x0,
    method='RK45')
x_meu = sol.y
x1_meu,x2_meu,x3_meu,x4_meu = x_meu
v_meu = -(247.17439707*x1_meu + 61.17861094*x2_meu + 98.97921195*
    x3_meu + 55.31791641*x4_meu)
figure(figsize = (12,12))
subplot(3,1,1),plot(t,x1_meu*180/pi,t,x1_alves*180/pi)
title('Estado $x_1$')
xlim((0,4))
ylabel('Angulo $(\\circ)$')
grid(True)
subplot(3,1,2),plot(t,x3_meu,t,x3_alves)
title('Estado $x_3$')
xlim((0,7))
grid(True)
ylabel('Posicao $(m)$')
subplot(3,1,3),plot(t,v_meu,t,v_alves)
title('Sinal de controle $v$')
xlabel('Tempo $(s)$')
grid(True)
ylabel('Esforco de controle $N$')
xlim((0,2))
pot_media = simpson([v_alves**2,v_meu**2],x = t)/(t[-1]-t[0])
print(f'A potencia media para o caso do alves vale {pot_media[0]}')
print(f'A potencia media para o meu caso vale {pot_media[1]}')
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/sinal_estados_conjunto_cocota.pdf')
```

O Simulações não lineares do pêndulo invertido com linearização entrada-saída, dinâmica zero e controlador PI

```
#Na primeira vez q for rodar, descomentar 3 prim. linhas
#from google.colab import drive#Montar a pasta no drive
#drive.mount('/content/drive')
import numpy as np
from numpy import pi, transpose, sin, cos, arange, array, ones_like
from scipy.integrate import solve_ivp, simpson
from matplotlib.pyplot import plot, grid, xlabel, figure, title
from matplotlib.pyplot import ylabel, savefig, subplot, show, xlim
def pendulo_n_linear_PI(t,x):
m,M,g,l = [0.1,2,9.8,0.5]
x1,x2,x3,x4,x5 = x
f2 = 3*(2*M*g*sin(x1) + 2*g*m*sin(x1) - l*m*x2**2*sin(2*x1))/(2*l*(4*M
     + 3*m*sin(x1)**2 + m))
f4 = (-3*g*m*sin(2*x1) + 8*l*m*x2**2*sin(x1))/(2*(4*M + 3*m*sin(x1)**2
     + m))
g2 = -3*cos(x1)/(l*(4*M + 3*m*sin(x1)**2 + m))
g4 = 4/(4*M + 3*m*sin(x1)**2 + m)
k1,k2,k3,k4,k5 = [858.9605,228.8245,652.1584,286.5367,-989.7921]
v = -(k1*x1 + k2*x2 + k3*x3 + k4*x4 + k5*x5)
u = (v-f2)/g2
xdot = [x2,f2+g2*u,x4,f4+g4*u,1-x3]
return xdot
def pendulo_alves_PI(t,x):
x1,x2,x3,x4 = x
m,M,g,l = [0.1,2,9.8,0.5]
f2 = (3*(- l*m*cos(x1)*sin(x1)*x2**2 + g*m*sin(x1) + M*g*sin(x1)))/(M*
    l + l*m + 3*M*l*cos(x1)**2 + 3*M*l*sin(x1)**2 + 3*l*m*sin(x1)**2);
f4 = (3*l*m*x2**2*sin(x1)**3 - 3*g*m*cos(x1)*sin(x1) + l*m*x2**2*sin(
    x1) + 3*l*m*x2**2*cos(x1)**2*sin(x1))/(M + m + 3*m*sin(x1)**2 + 3*M
    *cos(x1)**2 + 3*M*sin(x1)**2);
g2 = - (3*(l*m*cos(x1)*sin(x1)*x2**2 + cos(x1) - g*m*sin(x1) - M*g*sin
    (x1)))/(M*l + l*m + 3*M*l*cos(x1)**2 + 3*M*l*sin(x1)**2 + 3*l*m*sin
    (x1)**2) - (3*(- l*m*cos(x1)*sin(x1)*x2**2 + g*m*sin(x1) + M*g*sin(
    x1)))/(M*l + l*m + 3*M*l*cos(x1)**2 + 3*M*l*sin(x1)**2 + 3*l*m*sin(
    x1)**2);
g4 = (3*cos(x1)**2 + 3*sin(x1)**2 + 3*l*m*x2**2*sin(x1)**3 - 3*g*m*cos
    (x1)*sin(x1) + l*m*x2**2*sin(x1) + 3*l*m*x2**2*cos(x1)**2*sin(x1) +
     1)/(M + m + 3*m*sin(x1)**2 + 3*M*cos(x1)**2 + 3*M*sin(x1)**2) -
    (3*l*m*x2**2*sin(x1)**3 - 3*g*m*cos(x1)*sin(x1) + l*m*x2**2*sin(x1)
```

```
    + 3*l*m*x2**2*cos(x1)**2*sin(x1))/(M + m + 3*m*sin(x1)**2 + 3*M*
    cos(x1)**2 + 3*M*sin(x1)**2);
k1,k2,c1,c2,c3,e1,e3 = [100,20,10,1,20,x3-1,x4]
v = -k1*x1 - k2*x2
u = (v-f2)/g2
v3 = (g2*(-c1*e1-c3*x4-c2*f4) - c2*g4*(v-f2))/(c2*g2*g4);
u1 = u - v3;
xdot = [x2,f2+g2*u1,x4,f4+g4*u1]
return xdot
tempo,h = [50,1e-4]
t = arange(0,tempo+h,1e-4)
iter = array([0,tempo])
x0 = [5*pi/180,0,0,0]
#calcular os valores do alves
sol = solve_ivp(pendulo_alves_PI,t_span=[0,tempo],t_eval=t,y0 = x0,
    method='RK45')
x_alves = sol.y
x1_alves,x2_alves,x3_alves,x4_alves = x_alves
v_alves = -100*x1_alves - 20*x2_alves
#calcular os meus valores
x0 = np.hstack((x0,0))
sol = solve_ivp(pendulo_n_linear_PI,t_span=[0,tempo],t_eval=t,y0 = x0,
     method='RK45')
x_meu = sol.y
x1_meu,x2_meu,x3_meu,x4_meu,x5_meu = x_meu
v_meu = -(858.9605*x1_meu + 228.8245*x2_meu + 652.1584*x3_meu +
    286.5367*x4_meu - 989.7921*x5_meu)
figure(figsize = (12,12))
subplot(3,1,1),plot(t,x1_meu*180/pi,t,x1_alves*180/pi)
title('Estado $x_1$')
xlim((0,6))
ylabel('Angulo $(\\circ)$')
grid(True)
subplot(3,1,2),plot(t,x3_meu,t,x3_alves,t,ones_like(t))
title('Estado $x_3$')
xlim((0,7))
grid(True)
ylabel('Posicao $(m)$')
subplot(3,1,3),plot(t,v_meu,t,v_alves)
title('Sinal de controle $v$')
xlabel('Tempo $(s)$')
grid(True)
ylabel('Esforco de controle $N$')
xlim((0,4))
#Calcular a potencia do esforco de controle
pot_media = simpson([v_alves**2,v_meu**2],x = t)/(t[-1]-t[0])
print(f'A potencia media para o caso do alves vale {pot_media[0]}')
```

```
print(f'A potencia media para o meu caso vale {pot_media[1]}')
savefig('/content/drive/MyDrive/2023.1/TCC/Felipe_Chaves/tcc_miktek/[
    TCC UFOP] Felipe Chaves/Figuras/sinal_estados_conjunto_cocota_PI.
    pdf')
```