

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

LUIS FERNANDO SEGALLA

ANÁLISE DE CONTROLADORES PID FRACIONÁRIOS

Joinville
2021

LUIS FERNANDO SEGALLA

ANÁLISE DE CONTROLADORES PID FRACIONÁRIOS

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Prof. Dr. Alexandro Garro Brito

Joinville
2021

Dedico este trabalho a todos que estão lutando alguma batalha mental. Você não está sozinho. Vai passar. É só química.

AGRADECIMENTOS

Agradeço a todos que fizeram parte da jornada que foi escrever este trabalho. Sem vocês este trabalho não existiria.

Tenho muito a agradecer à meus pais Emerson e Patrícia por todo o apoio e incentivo aos estudos, desde quando era apenas uma criança. Agradeço a minha irmã Beatriz por todo o carinho e amizade desde que me mudei para longe.

Tenho muito a agradecer a minha namorada Gabriella por toda a compreensão e carinho durante esse período. Agradeço por todas as horas e mais horas ouvindo reclamações e ajudando a ver o lado positivo das situações.

Por fim, tenho muito a agradecer aos meus amigos. Obrigado Breno, Arthur e Pedro por todos os anos de amizade e por estarem ao meu lado por estes mais de 10 anos. Agradeço também a Camila, ao Bruno, o Igor e a Marina por terem feito parte dessa caminhada.

"The things I once imagined would be my greatest achievements were only the first steps toward a future I can only begin to fathom."—Jace Beleren

RESUMO

Controladores PID Fracionários são uma generalização do já conceituado controlador PID. Embora seu uso ainda não seja muito comum, nos últimos anos houve esforço da comunidade de controle para pesquisar sobre o tema, suas vantagens e desvantagens em relação ao PID padrão. Tendo em vista isso, este trabalho irá abordar uma análise comparativa dessa classe de controladores em relação ao PID clássico. Os resultados serão demonstrados por meio de simulações com o *software* Octave/Matlab, incluindo um estudo de casos com o pêndulo invertido, aplicando-se métodos de otimização numérica para determinar parâmetros ótimos de ganhos para os testes. Por fim, propõe-se trazer uma discussão sobre vantagens estruturais do PID Fracionário, assim como suas desvantagens e ponderar sobre onde este melhor se encaixa dentro do panorama de sistemas de controle.

Palavras-chave: Controlador PID Fracionário. Otimização Numérica. Sistemas de Controle

ABSTRACT

Fractional Order PID Controller are a generalization of the classical PID. Even though it's use is still not as common as the classical PID, members of the control community have been studying about it, focusing in it's advantages and disadvantages when compared to it's integer counterpart. This report presents an analysis of Fractional PID controllers, by discussing some of their advantages in comparison to the tradicional PID controller. Results will be shown via Octave/Matlab simulation using numerical optimization methods to find the optimal gains for the tests. We aim to create a discussion about the advantages and disadvantages of Fractional Order PID and when it's better suit to apply it instead of a integer order PID controller.

Keywords: Fractional Order PID Controller. Numerical Optimization. Control Systems.

LISTA DE FIGURAS

Figura 1 – Comportamento da Função Gama.	15
Figura 2 – Diagrama de Blocos de um sistema de controle típico	18
Figura 3 – Simulação do Filtro de Oustaloup para diferentes N aproximando $\alpha =$ -0.5.	21
Figura 4 – Diagrama de Blocos de um sistema com realimentação.	21
Figura 5 – Representação gráfica da relação entre o FOPID e o IOPID	23
Figura 6 – Diagrama de Bode da aproximação pelo filtro de Oustaloup do termo α entre -1 e 1	24
Figura 7 – Métricas de Controle observadas pela resposta de um sistema	25
Figura 8 – Realização do sistema realimentado com o controlador fracionário	27
Figura 9 – Fluxograma com do processo de otimização do PSO para um controlador fracionário	28
Figura 10 – Resposta da planta representada pela Equação (30) ao degrau unitário	30
Figura 11 – Comparação da resposta ao degrau da planta representada pela Equação (30) com os controladores encontrados	31
Figura 12 – Comparação da resposta resposta em frequência da planta representada pela Equação (30) com os controladores encontrados	32
Figura 13 – Resposta da planta representada pela Equação (31) ao degrau unitário	33
Figura 14 – Lugar das raízes para o sistema (31) com o polo na origem e dois zeros no semiplano esquerdo.	34
Figura 15 – Resposta ao degrau da planta representada pela Equação (31) com o controlador fracionário encontrado	35
Figura 16 – Resposta em frequência da planta representada pela Equação (31) com o controlador fracionário encontrado	36
Figura 17 – Comportamento do lugar das raízes para o sistema apresentado na Equação (32) com um PID genérico	38
Figura 18 – Resposta ao degrau da planta representada pela Equação (32) com os controladores encontrados	39
Figura 19 – Resposta resposta em frequência da planta representada pela Equação (32) com os controladores encontrados	39
Figura 20 – Diagrama de corpo livre do pêndulo invertido	41
Figura 21 – Malha de controle realimentada com o distúrbio de movimentação	44
Figura 22 – Malha de controle realimentada com o controlador no caminho de realimentação	44

Figura 23 – Resposta da função de transferência do pêndulo invertido a uma perturbação de caráter impulsivo com os controladores Inteiro e Fracionário	45
Figura 24 – Comparação da resposta em frequência do para o sistema do pêndulo invertido	46
Figura 25 – Comparação com controlador inteiro e fracionário para uma perturbação impulsiva no pêndulo invertido no <i>Simulink</i>	48
Figura 26 – Comparação com controlador inteiro e fracionário para o pêndulo invertido quando a condição inicial é $\theta = 30^\circ$	49
Figura 27 – Comparação com controlador inteiro e fracionário para o pêndulo invertido quando a condição inicial é $\theta = 60^\circ$	49
Figura 28 – Comparação com controlador inteiro e fracionário para o pêndulo invertido quando a condição inicial é $\theta = 75^\circ$	50
Figura 29 – Comparação entre os controladores quando o sistema é submetidos a diversas perturbações impulsivas.	50

LISTA DE SÍMBOLOS

Γ	Letra grega Gama
λ	Letra grega Lambda
μ	Letra grega Mi
\in	Pertence
N	Conjunto dos números naturais
$R()$	Parte real de um número

LISTA DE ABREVIATURAS E SIGLAS

CF	Cálculo Fracionário
PID	Proporcional Integral Derivativo
PSO	Particle Swarm Optimization
ITAE	Integral of Time-Weighted Absolute Error
EEE	Erro em Estado Estacionário
PSS	Percentual de Sobressinal
INF	Infinito

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivo	13
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	13
1.2	Organização do Texto	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Cálculo Fracionário	15
2.2	Métodos de Otimização	17
2.2.1	Função de Custo	17
2.2.2	Algoritmo de Enxame de Partículas	18
2.3	Filtro Recursivo de Oustaloup	19
2.4	Formulação do Controlador PID Fracionário	20
3	METODOLOGIA	25
3.1	Métricas de Controle e aplicação das funções de custo	25
3.2	Implementação do sistema de controle fracionário para simulações	26
4	SIMULAÇÕES	29
4.1	Caso 1 - Sistema Marginalmente estável com polo no semiplano esquerdo	29
4.2	Caso 2 - Sistemas de Fase Não Mínima	32
4.3	Caso 3 - Sistema instável com polo no semiplano direito	37
5	ESTUDO DE CASO - PÊNDULO INVERTIDO	41
5.1	Equacionamento	41
5.2	Resultados da Otimização no espaço de Laplace	43
5.3	Simulação com Simulink	47
6	CONCLUSÕES	52
	REFERÊNCIAS	54
	APÊNDICE A	55
.1	Algoritmo para determinação dos ganhos ótimos	56
.2	Filtro Recursivo de Oustaloup	59
.3	Função de custo para o controlador inteiro	60
.4	Função de custo para o controlador fracionário	60

1 INTRODUÇÃO

Embora seja um assunto sem tanta repercussão no âmbito da engenharia, o cálculo fracionário não é necessariamente algo novo. Seu surgimento data de 1695 numa troca de cartas entre o marquês Guillaume François del'Hôpital e Gottfried Leibniz, ainda durante os primeiros anos de desenvolvimento da teoria clássica do cálculo diferencial integral. Nessas correspondências foram propostas as primeiras definições de Integrais e Derivadas de ordem não-inteira (ANDRADE et al., 2018).

A primeira utilização prática do cálculo fracionário (CF) se deu em 1823, quando o matemático norueguês Niels Abel solucionou o problema da curva tautócrona¹ através de equações fracionárias, de onde surge uma integral de ordem meia (CARVALHO; OTTONI, 2018). O CF também é utilizado em áreas da engenharia como eletrônica, robótica e controle como pode ser visto (GUTIERREZ et al., 2010), além ser capaz de modelar sistemas com características de memória infinita e não-linearidades complexas (DUARTE; MACHADO, 2006) e (TORVIK; BAGLEY, 1984).

Dentro da área de sistemas de controle, o que se chama de controle fracionário apresenta uma generalização da teoria clássica de controle, levando a uma modelagem mais detalhada do sistema quando comparada ao controle inteiro². Esta generalização é adequada ao tratar-se de sistemas que sejam muito complexos ou com características como memória, além deste tipo de controle apresentar também um desempenho mais robusto (ZHAO; XUE; CHEN, 2005).

A primeira proposta de um controlador de ordem não inteira foi feita por (PODLUBNY, 1999), Equação (1), introduzindo dois parâmetros (λ e μ) que podem ser ajustados para melhorar o desempenho em relação à sua contraparte inteira (LIU; ZHANG, 2018).

$$PI^\lambda D^\mu \quad (1)$$

Devido a algumas características de memória infinita e limitações de *software*, a única maneira de se aplicar um controlador fracionário é aproximá-lo através de um controlador de ordem inteira (VINAGRE; PODLUBNY; HERNÁNDEZ; FELIU, 2000). As expansões contínuas de frações com interpolação, ou então, identificação e ajuste de curva são algumas das técnicas que podem ser utilizadas para a aproximação do controlador fracionário (VINAGRE et al., 2000).

Para este trabalho foi escolhido o Método de Oustaloup como forma de aproximar o controlador fracionário. Esse provê uma aproximação adequada dentro de uma banda de frequências (GRANDI; TRIERWEILER, 2019). A seção 2.3 tratará de

¹ Problema de uma curva para a qual, independente do ponto inicial, objetos deslizando por ela levam o mesmo tempo para chegar ao seu ponto mais baixo.

² Teoria clássica de controle onde todos os termos e equações diferenciais são de ordem inteira.

forma mais detalhada sobre a implementação e características de implementação e desempenho do filtro.

1.1 OBJETIVO

1.1.1 Objetivo Geral

Comparar o desempenho dos controladores PID inteiros e fracionários utilizando métodos de otimização, determinando ganhos ótimos, de forma a se explorar situações onde cada tipo de controlador se encaixe melhor.

1.1.2 Objetivos Específicos

A fim de atingir o objetivo proposto acima, apresentam-se os seguintes objetivos específicos:

- Estudar um tipo de controlador de grande interesse para a comunidade de sistemas de controle;
- Simular o comportamento do controlador utilizando técnicas para representá-lo de forma fiel;
- Implementar e utilizar métodos de otimização numérica aplicados para sistemas de controle.
- Comparar o comportamento desses controladores a fim de identificar situações onde as características de cada um possam ser melhor aproveitadas.
- Aplicar o controlador fracionário para um problema conhecido de engenharia de controle, neste caso, o pêndulo invertido.

1.2 ORGANIZAÇÃO DO TEXTO

Este trabalho está dividido em mais cinco capítulos. A Fundamentação Teórica tem como objetivo estabelecer a base do que é necessário para o entendimento do trabalho, como por exemplo, conceitos sobre CF, exemplos e implementação dos métodos de otimização, o funcionamento do Filtro Recursivo de Oustaloup e como ele é implementado e finalmente como o controlador fracionário é desenvolvido e compara-se ao controlador PID clássico.

No capítulo 3 será desenvolvida a metodologia utilizada durante o trabalho. Os conceitos apresentados no capítulo de Revisão Teórica serão utilizados para desenvolver os passos a serem seguidos e o funcionamento dos algoritmos que foram utilizados no trabalho.

O capítulo de Simulações traz um estudo com algumas plantas de controle, com características específicas, a fim de se estabelecerem comparações iniciais entre o desempenho dos controladores analisados. Para validar os controladores em um último

teste, no Capítulo 5 um estudo de caso com o pêndulo invertido é utilizado como última aplicação dos controladores e validação dos métodos desenvolvidos anteriormente.

Por fim, na Conclusão, há uma última análise de tudo que foi desenvolvido no trabalho e todos os resultados obtidos com o propósito de tentar estabelecer as situações onde cada tipo de controlador pode trazer maiores benefícios.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão introduzidos os conceitos matemáticos necessários para a formulação do trabalho. Inicialmente serão discutidos os conceitos do cálculo fracionário e como este difere do cálculo integral. Em seguida será introduzida uma explicação sobre os métodos de otimização. Por fim, serão utilizados os conceitos discutidos para formular-se a base de controle que será usada no resto do trabalho.

2.1 CÁLCULO FRACIONÁRIO

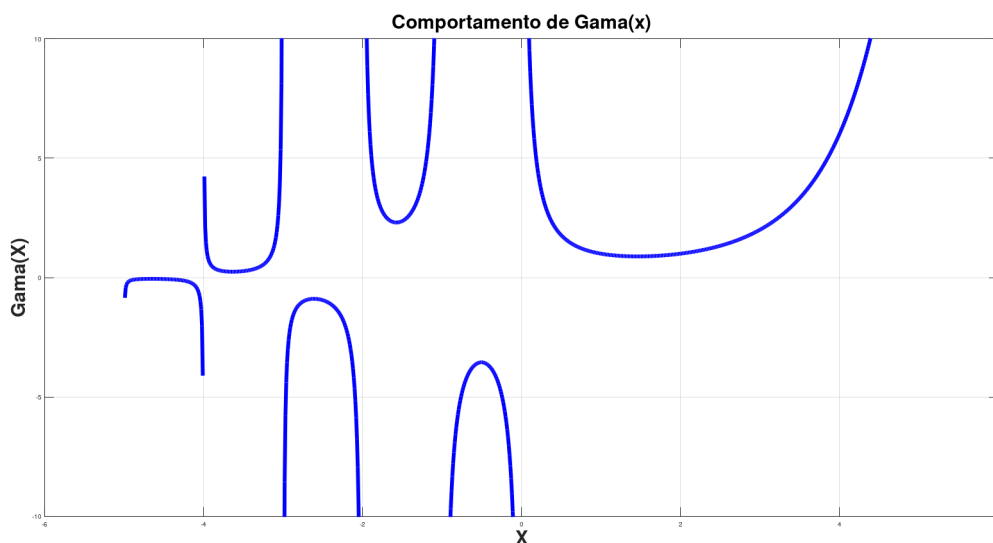
Um aspecto importante do cálculo fracionário é a existência de mais de uma definição de seus operadores de derivada e integral (ANDRADE et al., 2018). Serão demonstradas algumas das definições existentes. Antes porém se faz necessário a apresentação de algumas ferramentas matemáticas básicas que ajudarão no desenvolvimento das equações.

A primeira ferramenta apresentada é a função gama, ou função de Euler (Equação (2)), que generaliza o conceito de fatorial para qualquer valor inteiro ou não inteiro, real ou imaginário (CARVALHO; OTTONI, 2018).

$$\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt, z \neq -1, -2, -3, \dots \quad (2)$$

O comportamento da função gama pode ser visto na Figura 1:

Figura 1 – Comportamento da Função Gama.



Fonte: autoria própria (2021).

A função gama apresenta duas propriedades que serão importantes neste trabalho, representadas nas Equações (3) e (4) para o conjunto dos números naturais (N):

$$\Gamma(z + 1) = z\Gamma(z) \quad (3)$$

$$\Gamma(n + 1) = n!, \forall n \in N \quad (4)$$

A segunda função a ser tratada é a função de Mittag-Leffler, que apresenta uma generalização da exponencial. Seu papel na resolução das equações diferenciais lineares de ordem não inteira é similar à função exponencial para as equações diferenciais lineares de ordem inteira (CARVALHO; OTTONI, 2018).

Sua definição é dada pela Equação (5):

$$E_{\alpha,\beta}(x) = \sum_{n=0}^{\infty} \frac{x^n}{\Gamma(n\alpha + \beta)} \quad (5)$$

Para α e β iguais a 1 tem-se a Equação (6):

$$E_{1,1}(x) = \sum_{n=0}^{\infty} \frac{x^n}{\Gamma(n + 1)} = e^x \quad (6)$$

A definição da derivada para valores inteiros é conhecida, e de acordo com (CARVALHO; OTTONI, 2018) sua forma generalizada para ordem n é dada na Equação (7):

$$\frac{d^n f(x)}{dx^n} = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{k=0}^n (-1)^k \frac{n!}{(n-k)!k!} f(x - kh) \quad (7)$$

Por volta de 1968 Anton Karl Grünwald e Aleksey Vasilievich Letnikov, de forma independente, propuseram a Expressão (8) para a generalização da derivada de ordem não inteira (ANDRADE et al., 2018):

$$\frac{d^\alpha f(x)}{dx^\alpha} = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k \frac{\Gamma(\alpha + 1)}{\Gamma(\alpha - k + 1)k!} f(x - kh) \quad (8)$$

Como forma de demonstrar a definição da integral fracionária, inicia-se definindo uma integral inteira de ordem genérica (ANDRADE et al., 2018):

$$J^{n+1} f(x) = \frac{1}{n!} \int_{-\infty}^x (x-t)^n f(t) dt \quad (9)$$

Generalizando a ordem inteira $n + 1$ para α e substituindo o fatorial pela função Gama, obtém-se a função conhecida como integral fracionária de Riemann-Liouville (ANDRADE et al., 2018):

$$J^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_{-\infty}^x (x-t)^{\alpha-1} f(t) dt, \alpha > 0 \quad (10)$$

Com os conceitos apresentados é possível definir o Operador Fracionário Diferencial-Integral como:

$${}_{t_0}D_{t_1}^\alpha = \begin{cases} \int_{t_0}^{t_1} f(t) dt^{-\alpha} & \text{se } R(\alpha) < 0 \\ f(t) & \text{se } R(\alpha) = 0 \\ \frac{d^\alpha f(t)}{dt^\alpha} & \text{se } R(\alpha) > 0 \end{cases} \quad (11)$$

onde, baseado na parte real do valor de α , é possível realizar uma operação diferente.

A derivada de Caputo, que é o operador utilizado no resto deste trabalho, consiste de duas operações. Primeiro faz-se uma derivação inteira seguida de uma integração fracionária.

$$D^\alpha f(x) = J^v \left[\frac{d^m f(x)}{dx^m} \right], m = \lceil \alpha \rceil, v = m - \alpha \quad (12)$$

Um fator importante para a escolha da derivada de Caputo é o fato de que a derivada de valores constantes é sempre zero, o que não é sempre verdade para casos como a derivada de Riemann-Liouville. Há também o fato de que a derivada de Caputo apresenta valores iniciais fisicamente interpretáveis quando utilizada para resolução de equações diferenciais fracionárias (CARVALHO; OTTONI, 2018).

Ao aplicar-se a transformada de Laplace da derivada de Caputo, (DULĂU et al., 2017), obtém-se:

$$L\{{}_{t_0}D_t^\alpha\} = s^\alpha F(s) - {}_{t_0}D_t^{\alpha-1}f(0) \quad (13)$$

2.2 MÉTODOS DE OTIMIZAÇÃO

Otimização é constantemente utilizada na engenharia já que recursos reais como material, tempo, dinheiro e energia são sempre finitos e seu bom gerenciamento é vital para que sejam maximizados ganhos e minimizadas perdas (KOZIEL; YANG, 2011).

Métodos de otimização procuram encontrar uma solução \vec{x} que minimize o valor de $f(\vec{x})$ para uma dada função de custo específica (ARORA et al., 1995). Neste trabalho optou-se por pelo uso de otimização para determinação dos parâmetros dos ganhos dos controladores deste trabalho, ao otimizarem-se os dois controladores sob as mesmas condições, espera-se obter uma comparação justa entre os dois casos.

2.2.1 Função de Custo

A fim de otimizar-se uma função é necessário uma maneira de determinar o quanto os valores obtidos estão ajustados com o que era desejado para sua saída. Nos métodos de otimização usam o que se chama de funções de custo para este fim. Existem inúmeras formas de se definir uma função de custo, elas podem tentar otimizar

apenas um parâmetro específico. Um dos inúmeros exemplos de sua aplicação no caso de controle é o trabalho de (ZHAO et al., 2005), onde é desenvolvido uma função que otimiza parâmetros de margem de ganho e fase para um controlador específico.

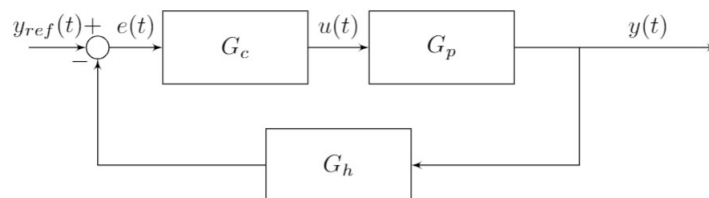
Uma classe de funções de custo muito comum na engenharia de controle são as chamadas funções com critério de desempenho integral que tentam minimizar o erro da função otimizada quando comparada com um valor de referência (DENIZ et al., 2017). Alguns exemplos deste tipo de função seriam a integral do erro absoluto (IAE), integral do erro absoluto ao quadrado (ISE), integral do erro absoluto pelo tempo (ITAE) e integral do erro absoluto ao quadrado pelo tempo (ITSE).

$$\begin{cases} IAE = \int_0^{\infty} |e(t)| dt \\ ISE = \int_0^{\infty} |e(t)|^2 dt \\ ITAE = \int_0^{\infty} t |e(t)| dt \\ ITSE = \int_0^{\infty} t |e(t)|^2 dt \end{cases} \quad (14)$$

onde $e(t)$ é definido baseado no diagrama de blocos da Figura 2 como:

$$e(t) = y_{ref}(t) - y(t) \quad (15)$$

Figura 2 – Diagrama de Blocos de um sistema de controle típico



Fonte: autoria própria (2021).

Neste trabalho o bloco G_h , que representa o sensor medindo a saída, é considerado unitário.

No Capítulo 3 as funções de custo serão discutidas mais profundamente focando em seu impacto quando deseja-se otimizar um problema de controle.

2.2.2 Algoritmo de Enxame de Partículas

Algoritmos de Enxame de Partículas (PSO - *Particle Swarm Optimization*) são uma classe de algoritmos de otimização meta-heurísticos baseado em populações, que foi desenvolvido pela observação de cardumes de peixe e bando de pássaros (BABU; CHIRANJEEVI, 2016).

O PSO consiste de um conjunto de partículas, contidas no vetor \vec{x} , dentro de um Espaço de Busca (em inglês, *search space*) N -dimensional. Cada partícula tem uma posição d_i e uma velocidade v_i local, além de ser capaz de guardar sua melhor posição P_i , depois de avaliada pela função de custo.

A cada passo, a melhor posição de todas as partículas da população é guardada em um vetor de desempenho global P_{gi} . Para que seja possível explorar o *search space*, cada partícula muda sua velocidade e posição local com base nas Equações (16) e (17):

$$v_i^{t+1} = wv_i^t + c_1r_1(p_i - x_i^t) + c_2r_2(p_g - x_i^t) \quad (16)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (17)$$

O termo w refere-se a inércia das partículas, que decresce de forma linear e é dado pela Equação (18):

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} iter \quad (18)$$

Os termos c_1 e c_2 representam as acelerações do problema e são constantes definidas antes do começo do algoritmo. O mesmo vale para os termos r_1 e r_2 que são valores aleatórios entre $[0,1]$.

O método do enxame de partículas atende às necessidades deste trabalho, sendo capaz de ser adaptado para otimizações de controladores como será demonstrado no Capítulo 3. Aspectos como custo computacional e velocidade de convergência das respostas do algoritmo estão fora do escopo de estudo, por isso não serão levados em consideração quando discutida a otimização nas simulações.

2.3 FILTRO RECURSIVO DE OUSTALOUP

Sistemas de ordem fracionária tem uma base matemática sólida, como demonstrada nas seções anteriores, e são utilizados para modelar problemas do mundo real, principalmente aqueles que apresentam requisitos como memória ou que possam apresentar características fractais¹ (ZHANG et al., 2016) e (DU et al., 2013). Tais características são vantajosas na hora de modelar este tipo de sistema, porém tornam sua implementação no mundo real impossível, já que seriam necessários recursos como memória e tempo infinito para as simulações (OUSTALOUP et al., 2000).

A solução para este problema vem da implementação de (OUSTALOUP et al., 2000) que aproxima o sistema fracionário por um sistema inteiro de alta ordem. O filtro

¹ Repetições infinitas decrescentes que seguem um padrão específico.

que aproxima um sistema de ordem fracionária α , dentro de uma faixa de frequências (ω_b, ω_h) com frequência de corte em $\sqrt{\omega_b \cdot \omega_h}$, é definido como:

$$F(\alpha) = s^\alpha = K \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k} \quad (19)$$

Para garantir a estabilidade do sistema, os polos e zeros do filtro são determinados de forma a encontrarem-se todos no eixo real do semiplano esquerdo, de acordo com as equações a seguir:

$$\omega_k = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+N+0.5(1+\alpha)}{2N+1}} \quad (20)$$

$$\omega'_k = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+N+0.5(1-\alpha)}{2N+1}} \quad (21)$$

$$K = \omega_h^\alpha \quad (22)$$

Seguindo as Equações (19) à (22) é possível garantir uma aproximação fiel de α que preserva as características especiais do sistema fracionário. O termo N nas equações corresponde à ordem de aproximação desejada no filtro; quanto maior o valor de N, melhor a resolução que se terá na resposta do sistema. Porém é possível notar que o sistema cresce com ordem $2N+1$, fazendo com que valores muito grandes de N gerem polinômios enormes, que podem causar dificuldade para serem simulados.

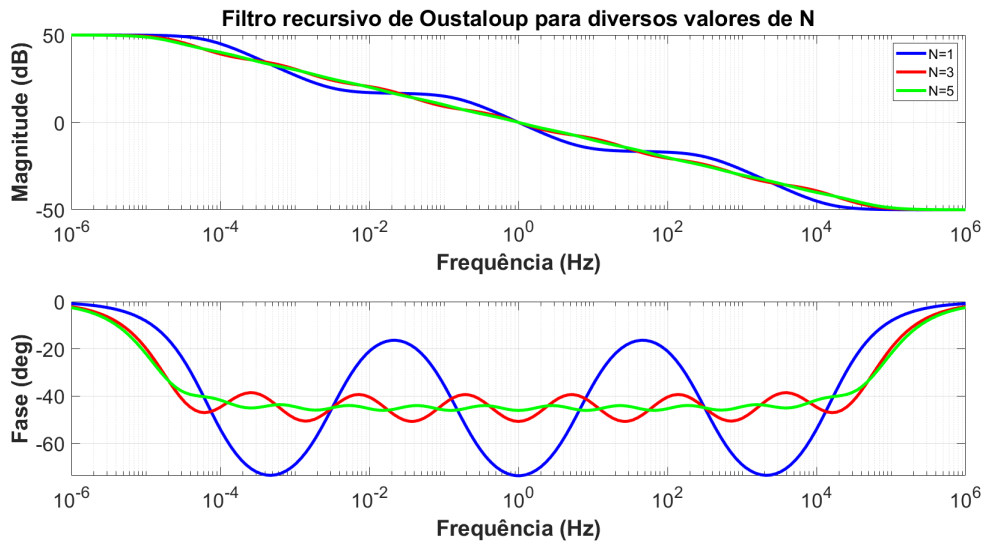
Na Figura 3 traz-se uma comparação entre a resposta em frequência de um filtro de ordem $\alpha = -0.5$, entre 10^{-5} até 10^5 Hz para alguns valores de N. Como pode ser observado na figura conforme aumenta-se o valor de N a resposta em frequência torna-se menos oscilatória dentro da faixa utilizada para construção do filtro.

O filtro de Oustaloup é uma ferramenta essencial para o presente trabalho. Sem ele seria impossível se obter uma versão implementável dos controladores fracionários dentro dos ambientes simulados do Octave/Matlab. Para os casos simulados que serão desenvolvidos nas próximas seções e capítulos, foram utilizados filtros com uma faixa de frequência entre 10^{-5} até 10^5 Hz e um $N=3$ que, de acordo com (DAS et al., 2011), já apresenta uma aproximação razoável. Valores acima destes não garantiriam maiores ganhos em termos de precisão na aproximação.

2.4 FORMULAÇÃO DO CONTROLADOR PID FRACIONÁRIO

Para entender como o PID fracionário funciona e suas características únicas é necessário primeiro compreender bem os conceitos desenvolvidos pelo PID clássico. O controlador é desenvolvido com três termos, respectivamente, proporcional, integral e derivativo. Cada um representa uma ação que será tomada perante o erro, que é

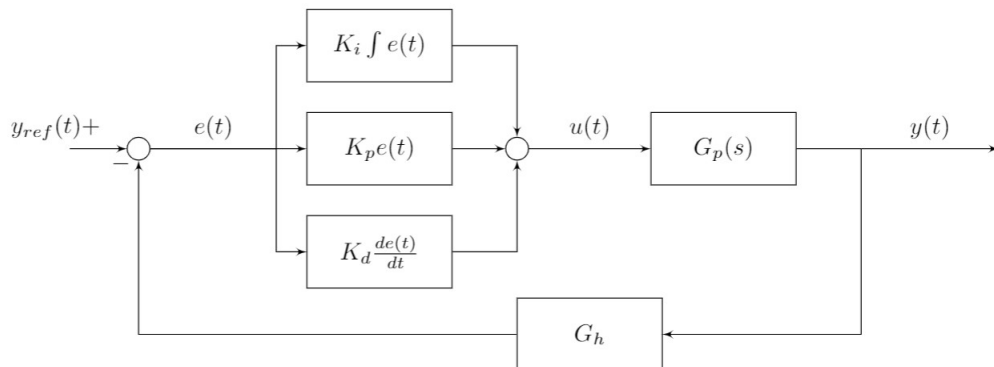
Figura 3 – Simulação do Filtro de Oustaloup para diferentes N aproximando $\alpha = -0.5$.



Fonte: autoria própria (2021).

medido quando comparada a saída do sistema com a referência desejada. A Figura 4 traz uma representação de um sistema realimentado com um controlador PID.

Figura 4 – Diagrama de Blocos de um sistema com realimentação.



Fonte: autoria própria (2021).

Dentre os sistemas analisados, encontram-se o controlador e a planta. Quando o sistema é excitado com uma entrada $r(t)$, há uma resposta $y(t)$ que representa a reação do sistema a esta excitação. Para que possamos utilizar de forma completa o controlador é necessário então que a saída seja realimentada ao sistema comparando-a com a entrada, o que gera o sinal do erro $e(t)$ (Equação (15)). Este erro é tratado pelo controlador e é sobre ele que os três termos atuam de forma a tentar reduzi-lo.

O termo proporcional age diretamente sobre o erro atual medido. Logo, seu

efeito no esforço de controle é maior quanto maior for o erro em termos instantâneos.

O segundo termo do controlador é o termo integral que age, diferentemente do termo proporcional, nos valores passados do erro. Integrando o sinal ao longo do tempo com o objetivo de eliminar erros persistentes, focando na resposta em regime estacionário.

O último termo do PID é o termo derivativo que atua sobre a derivada do erro medido. Essa ação tem como foco eliminar a tendência do erro de crescer com o tempo, tendo uma ação mais acentuada em situações onde há grandes variações no sinal medido.

Ao juntarmos os três termos como no diagrama da Figura 4, obtém-se uma equação diferencial ordinária capaz de representar o controlador PID:

$$g_c(t) = \frac{u(t)}{e(t)} = K_p e(t) + \int_0^t K_i e(t) + \frac{K_d de(t)}{dt} \quad (23)$$

Ao se aplicar a transformada de Laplace, com condições iniciais nulas, na Equação (23) chega-se na fórmula tradicional do controlador PID inteiro:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (24)$$

Para os propósitos deste trabalho a forma acima será denominada de IOPID, do inglês, *Integral Order PID*.

O controlador PID introduz dois zeros que podem ser posicionados em qualquer ponto do plano pela manipulação dos ganhos (K_p , K_i e K_d). Estes zeros tem o objetivo de movimentar os polos da planta do sistema posicionando-os de forma a satisfazer as métricas de controle desejadas pelo usuário. Há também a introdução de um polo na origem do plano s , vindo direto do integrador, que tem o objetivo de sempre zerar o erro em estado estacionário do sistema à uma entrada em degrau.

Com o conceito básico de uma malha de controle e um controlador PID inteiro definido é possível começar a explorar o controlador fracionário e sua implementação. Partindo dos conceitos discutidos na seção 2.1 passamos a interpretar operações de integração e derivação como sendo dependentes do termo fracionário (Equação (11)). Então é possível reescrever a Equação (23) de forma mais genérica dependendo dos termos não inteiros como:

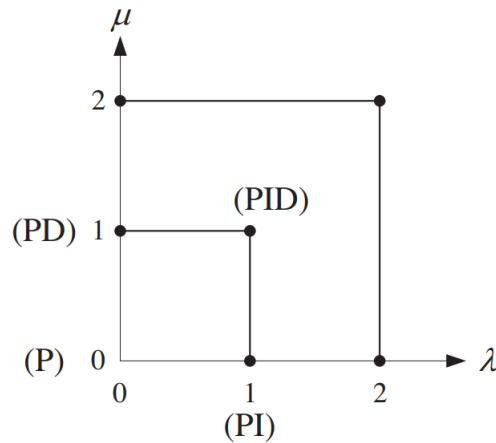
$$u(t) = K_p e(t) + K_i D^\lambda e(t) + K_d D^\mu e(t) \quad (25)$$

Aplicando a transformada de Laplace, com condições iniciais nulas, é possível então obter uma função de transferência para o controlador PID fracionário (FOPID, do inglês, *Fractional Order PID*) no domínio da frequência:

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (26)$$

Comparando as Equações (24) e (26) é possível notar que a diferença entre elas encontra-se nos termos λ e μ . Entretanto, para o caso onde ambos são iguais a 1, volta-se ao modelo original do controlador PID clássico. Isso mostra que o PID inteiro nada mais é que um subconjunto do PID fracionário (ZHAO et al., 2005), ilustrado na Figura 5.

Figura 5 – Representação gráfica da relação entre o FOPID e o IOPID



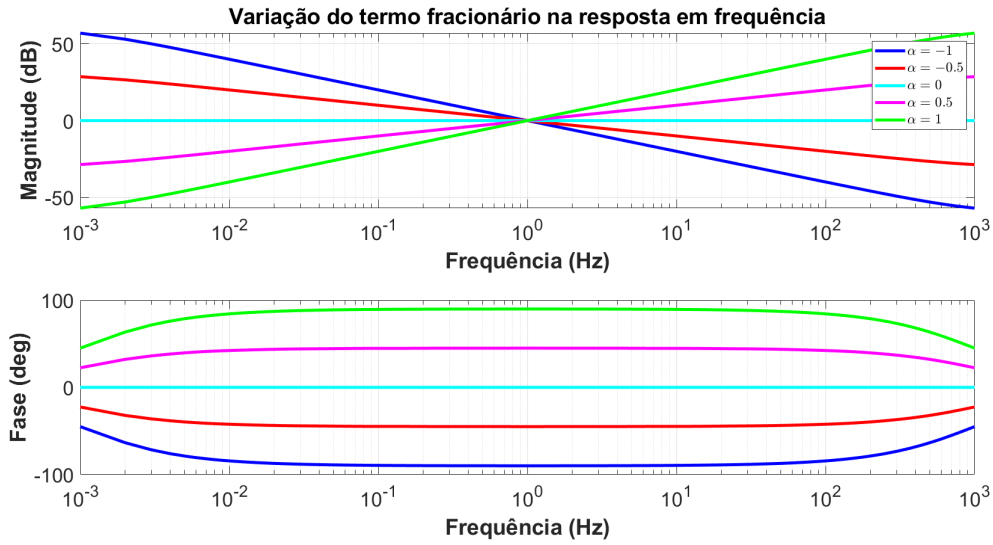
(CHIPIPOP et al., 2019)

Os dois novos termos do controlador introduzem mais liberdade na hora do ajuste dos parâmetros, permitindo, por exemplo, que se busque uma solução no espaço \mathbb{R}^5 , em vez do \mathbb{R}^3 – como feito no caso do PID clássico. Além disso, garantem-se ao controlador as características fracionárias que foram discutidas nas seções 2.1 e 2.3.

Com a adição dos termos fracionários é possível analisar como a resposta em frequência de um sistema se comporta com a variação destes termos e o que isso pode implicar para o projeto do controlador. Como o operador diferencial-integral permite que a derivada e integral fracionárias sejam obtidas apenas mudando o valor de α (Equação (11)), é possível analisarmos a resposta em frequência do sistema quando variamos o termo fracionário de -1 (um integrador puro) até 1 (um derivador puro), utilizando-se da aproximação do filtro de Oustaloup. Este resultado é apresentado na Figura 6.

O efeito do termo fracionário é então permitir que a magnitude e a fase aproximem valores intermediários do integrador e derivador. De acordo com (DORF; BISHOP, 2001) o efeito de um polo na origem (integrador puro) é o de um decaimento da magnitude de -20dB por década enquanto a fase converge para -90° . Já para um zero na origem (derivador puro) o efeito é o contrário: há um aumento de magnitude de 20dB por década e a fase converge em 90° . Neste caso, o efeito dos termos fracionários é atingir a faixa de valores entre os expoentes inteiros, por exemplo, para o caso onde α é igual a -0,5 o decaimento e a fase são, respectivamente, -10dB por década e -45° ,

Figura 6 – Diagrama de Bode da aproximação pelo filtro de Oustaloup do termo α entre -1 e 1



Fonte: autoria própria (2021)

enquanto para $\alpha=0.5$, obtém-se magnitude de 10dB por década e fase de 45° .

De acordo com (OUSTALOUP et al., 2000), para um sistema fracionário $F(\alpha)$, a magnitude e a fase para uma determinada frequência (ω_u), respectivamente, comportam-se de acordo com as Equações (27) e (28).

$$|F(j\omega_u)| = 20\alpha \left(\frac{\omega_u^2}{\omega_u^2 + \omega_b^2} - \frac{\omega_u^2}{\omega_u^2 + \omega_h^2} \right) dB \quad (27)$$

$$\angle F(j\omega_u) = \alpha \left(\arctg \left(\frac{\omega_u}{\omega_b} \right) - \arctg \left(\frac{\omega_u}{\omega_h} \right) \right) deg \quad (28)$$

Para as simulações apresentadas nas próximas seções foi utilizado o *software* Octave na sua versão 6.2.0 com o auxílio do pacote de controle na versão 3.2.0 e o *software* MATLAB na versão R2019b. Todos os algoritmos que foram implementados para este trabalho podem ser encontrados no Apêndice A.

3 METODOLOGIA

Com a base teórica estabelecida, este capítulo terá como foco desenvolver a metodologia que será usada no trabalho para avaliação dos dados obtidos, assim como para definição dos tipos de testes a serem realizados e métricas a serem observadas.

3.1 MÉTRICAS DE CONTROLE E APLICAÇÃO DAS FUNÇÕES DE CUSTO

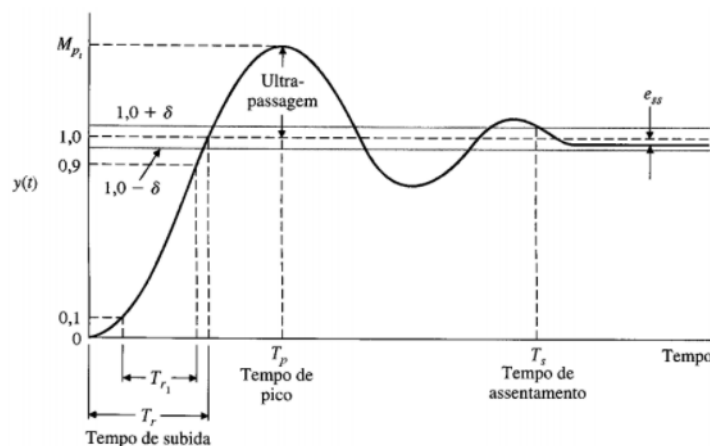
Um bom projeto de controle permite que o sistema tenha uma resposta transitória e em estado estacionário adequada quando excitado por um sinal externo (DORF; BISHOP, 2001). Para tanto, são necessárias métricas que definem o que é considerado como uma resposta aceitável dentro das limitações e requisitos do projeto em questão.

Entre algumas das métricas de controle mais comuns, e que serão utilizadas no resto deste trabalho, podem-se citar:

- Tempo de Subida (T_r) - Medida do tempo que o sistema leva para alcançar 90% do seu valor final.
- Tempo de Acomodação (T_s) - Tempo que o sistema leva para entrar em uma faixa de $\pm 2\%$ do valor final.
- Percentual de Sobressinal (PSS) - Valor máximo atingido pelo sistema quando submetido a uma excitação.
- Erro em Estado Estacionário (ESS) - Erro entre o valor atingido pelo sistema em relação a referência desejada em estado estacionário.

a Figura 7 apresenta uma visualização das métricas citadas acima.

Figura 7 – Métricas de Controle observadas pela resposta de um sistema



Fonte: (DORF; BISHOP, 2001).

Na seção 2.2.1 apresentaram-se diversos tipos de função de custo utilizadas na literatura. Junto dos conceitos discutidos anteriormente é possível então definir uma função que otimize os ganhos dos controladores, a fim de melhorar a resposta do sistema quando analisando-o pelas métricas de controle estabelecidas.

Ao considerarem-se os critérios que desejam-se otimizar é possível notar que todos relacionam-se com o erro. T_r e T_s medem a velocidade com que o erro diminui no transitório até chegar ao estado estacionário. O PSS mede o erro máximo que o sistema apresenta, geralmente, na primeira vez que o sistema tenta chegar até a referência. E por fim o ESS mede o erro já em regime estacionário para mostrar o quanto o sistema divergiu do valor de referência passado.

Tendo os requisitos necessários e as funções definidas na Equação 14 observa-se que tanto a ITAE quanto a ITSE apresentam as características necessárias, com a única diferença sendo o erro ao quadrado que aparece na segunda função. Este termo dá um foco maior ao erro absoluto, fazendo com que caso o mesmo não diminua rapidamente o valor do custo cresça ao quadrado, porém acaba tirando parte do foco do termo temporal, o que pode fazer com que funções que tenham um erro pequeno porém persistente não sejam tão afetadas. Por este motivo, optou-se pelo uso da ITAE como função de custo para as simulações deste trabalho.

3.2 IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE FRACIONÁRIO PARA SIMULAÇÕES

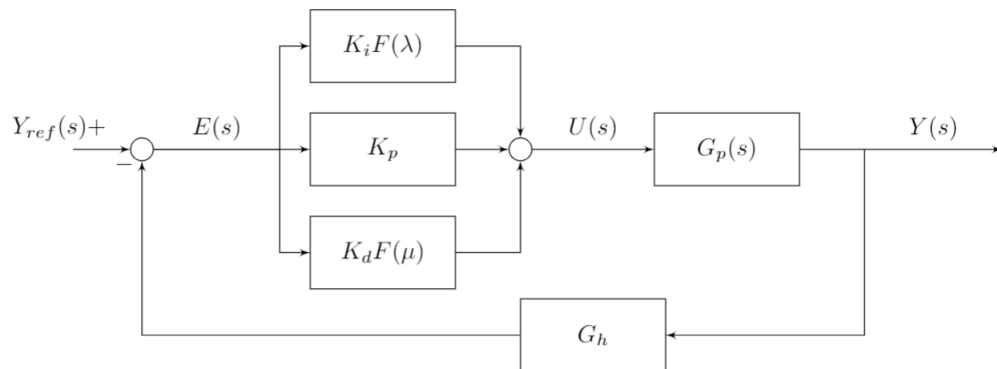
Com todos os pontos essenciais apresentados é possível agora desenvolver a metodologia que será utilizada para determinação dos controladores e simulações. O primeiro ponto a ser introduzido é como o sistema com controlador fracionário que, conforme já foi explicado anteriormente, precisa ser aproximado pelo filtro de Oustaloup é simulado. A Figura 8 traz uma representação do sistema com o filtro já multiplicando o valor dos ganhos integral e derivativo.

Os termos fracionários λ e μ são aproximados pelo filtro de Oustaloup para que possam ser simulados como um sistema de ordem inteira realimentado, dado pela Função (29). Para todos os casos de simulação apresentados neste trabalho optou-se por um grau de aproximação de $N=3$ com uma faixa de frequência entre 10^{-5} e 10^5 Hz. Com este formato é possível então utilizar as funções do Matlab/Octave como *step* e *bode* para que seja possível analisar a resposta do sistema.

$$G_{mf} = \frac{Y(s)}{R(s)} = \frac{G_p(K_p + K_i F(-\lambda) + K_d F(\mu))}{1 + G_p(K_p + K_i F(-\lambda) + K_d F(\mu))} \quad (29)$$

Com uma forma de simular o controlador fracionário e a planta é possível determinar o valor ótimo dos ganhos e dos termos fracionários baseado no algoritmo

Figura 8 – Realização do sistema realimentado com o controlador fracionário

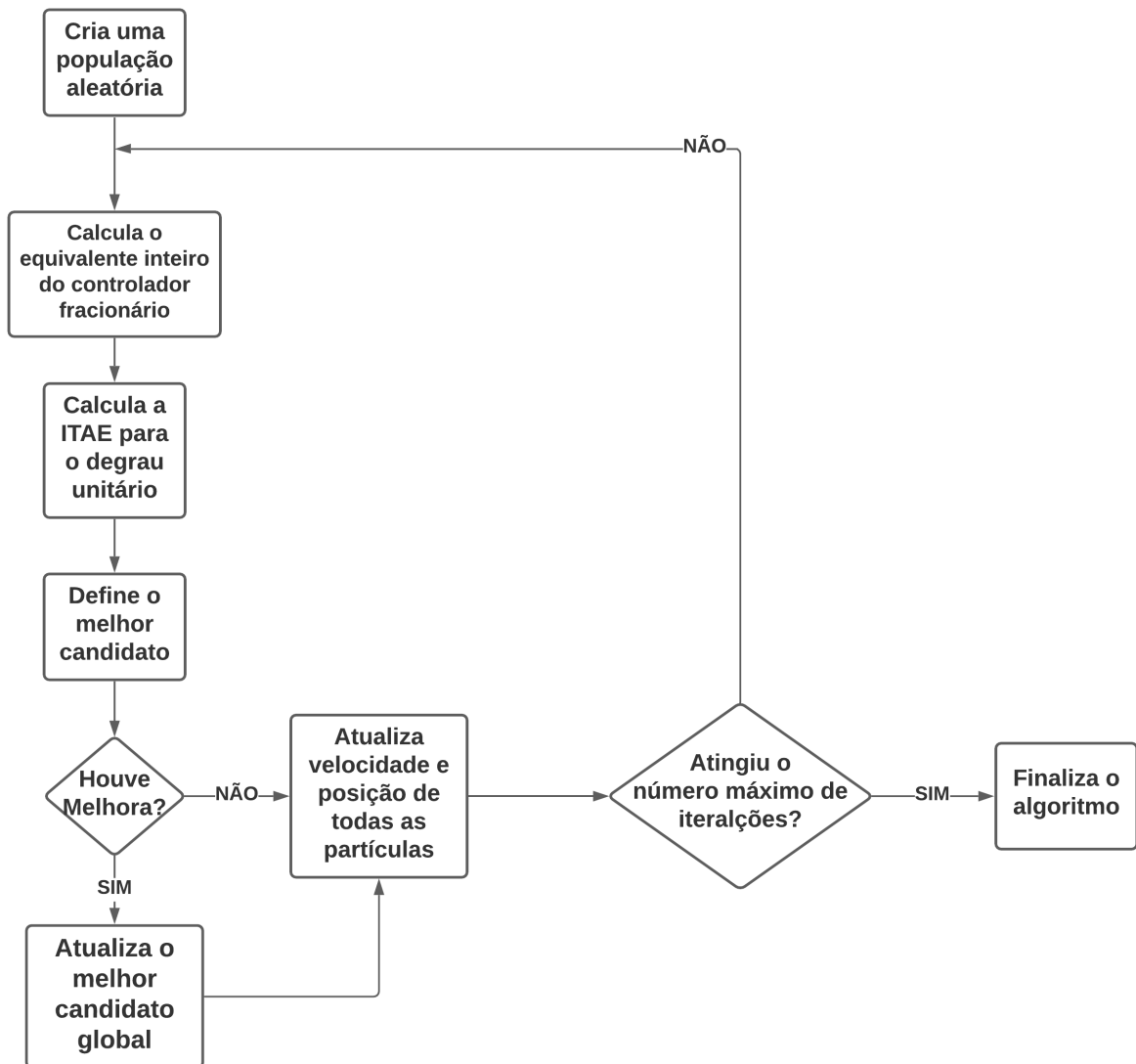


Fonte: Autoria própria 2021.

de otimização por enxame de partículas (Seção 2.2.2). Conforme foi explicado anteriormente a função de custo escolhida foi a ITAE e o erro que será medido é o da resposta do sistema ao degrau unitário já que a partir dele é possível obter todas as métricas de controle de interesse para este trabalho. Na Figura 9 há uma visualização do passo a passo do algoritmo de otimização quando selecionado o controlador fracionário para ser otimizado, enquanto no Apêndice A é possível encontrar todos os códigos implementados para o trabalho.

Para o algoritmo de otimização a população foi definida como 25 partículas que otimizam o problema durante 200 iterações. Então a cada iteração do PSO 25 possíveis soluções são criadas, para cada uma delas o termo fracionário é aproximado pelo algoritmo de Oustaloup e simula-se um degrau unitário para o sistema realimentado (Equação 29) com duração de 10 segundos e passos de simulação de 0,01 segundos. O resultado é então comparado com o degrau puro através da ITAE e o valor obtido é realimentado no algoritmo de PSO para que possam determinar o melhor candidato de cada iteração.

Figura 9 – Fluxograma com do processo de otimização do PSO para um controlador fracionário



Fonte: Autoria própria 2021.

4 SIMULAÇÕES

Neste capítulo serão apresentadas algumas simulações realizadas com o intuito de comparar o desempenho dos dois tipos de controladores em plantas teóricas com características bem definidas. O caso 1 explora uma situação marginalmente instável onde ambos os controladores são capazes de obter uma resposta estável para o sistema. Os casos 2 e 3 focam em situações naturalmente instáveis onde, por motivos que serão explorados detalhadamente em cada subseção, o controlador PID inteiro não é capaz de estabilizar a resposta da planta para nenhuma configuração dentro dos limites propostos. Esses casos servirão para reforçar os aspectos positivos do controlador PID fracionário em casos onde um controle por PID inteiro não cabe. Em todas as simulações apresentadas neste capítulo optou-se por um espaço de busca entre 0.01 até 10 para os ganhos (k_p , k_i e k_d) e 0.01 até 2 para os valores μ e λ do controlador fracionário.

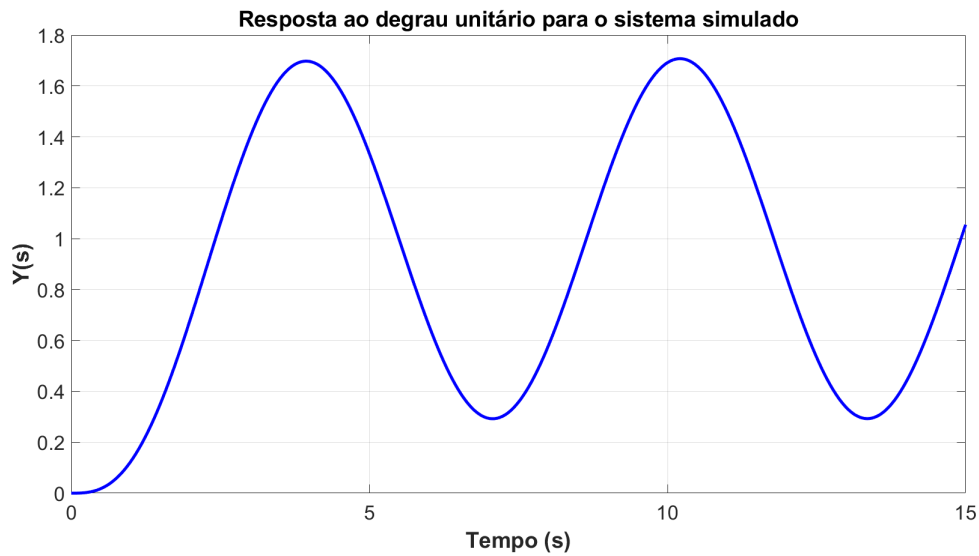
4.1 CASO 1 - SISTEMA MARGINALMENTE ESTÁVEL COM POLO NO SEMIPLANO ESQUERDO

Para um primeiro exemplo das simulações e de como foram executados os testes tem-se a planta:

$$G(s) = \frac{1}{(s^2 + 1)(s + 1)} \quad (30)$$

O sistema representado pela Equação (30) é estável (no sentido de que seu valor não diverge para infinito) sem a realimentação, porém apresenta um comportamento oscilatório muito forte graças aos seus dois polos complexos conjugados sobre o eixo imaginário, como pode ser visto na Figura 10. Quando realimentada sem um controlador, no enatando, o sistema torna-se instável oscilando indefinidamente com uma amplitude crescente.

Figura 10 – Resposta da planta representada pela Equação (30) ao degrau unitário



Fonte: Autoria própria 2021.

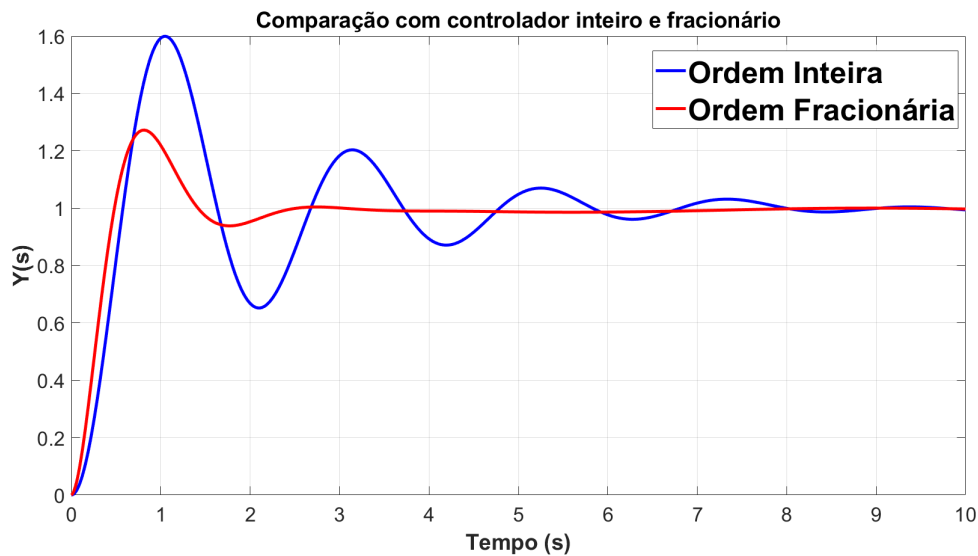
O objetivo agora é gerar dois controladores otimizados e comparar a resposta da planta quando realimentada junto deles, analisando se houve melhora na resposta. Os controladores encontrados com o método de otimização podem ser vistos na Tabela 1. Importante notar que para que houvesse uma comparação justa entre os controladores, optou-se por otimizar primeiro o controlador inteiro, e usar o valor de K_p encontrado como ganho proporcional fixo no controlador fracionário. Isso permitiria que o foco da otimização estivesse nos ganhos e índices fracionários.

Tabela 1 – Controladores otimizados para a planta da Equação (30)

	K_p	K_i	K_d	λ	μ
PID Inteiro	0,01	9,5754	9,2679	-	-
PID Fracionário	0,01	10,0	9,8734	0,7535	1,2703

Com os dois controladores definidos é possível agora simulá-los com a realimentação do sistema e aplicar o filtro de Oustaloup para aproximar o caso fracionário (Figura 8). Ambos os casos foram simulados com uma resposta ao degrau unitário e o resultado pode ser observado na figura a seguir:

Figura 11 – Comparação da resposta ao degrau da planta representada pela Equação (30) com os controladores encontrados



Fonte: Autoria própria 2021.

Analisando a Figura 11 é possível notar que ambos os controladores foram capaz de estabilizar o sistema, garantindo que o mesmo chegasse a uma resposta estável em estado estacionário. Porém o controlador fracionário obteve este resultado com tempos de subida e acomodação menores e apresentando menos sobressinal. Todos estes dados estão expostos na Tabela 2.

Tabela 2 – Métricas de controle para os controladores otimizados para a planta da Equação (30)

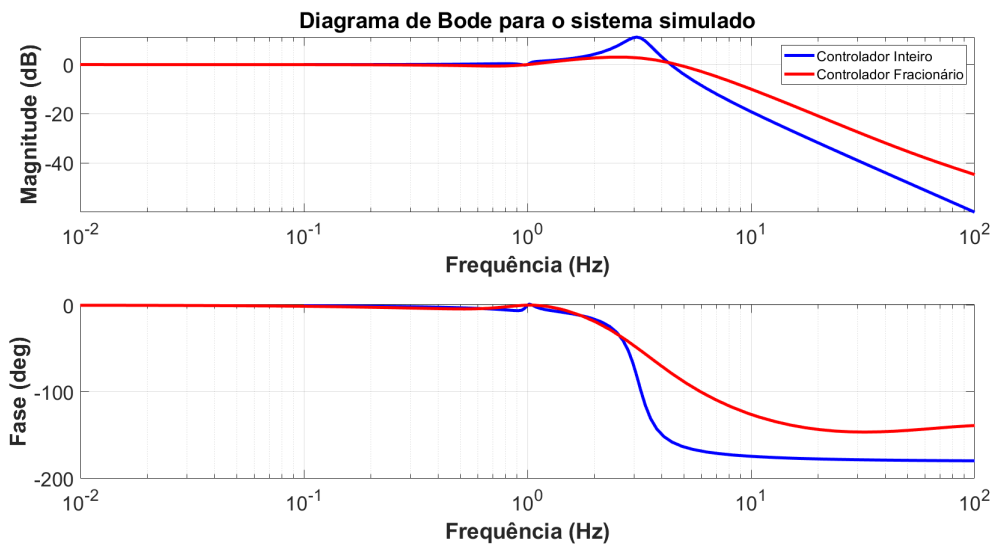
	PSS	EEE	T_r	T_s
PID Inteiro	59,5196%	0%	0,3889	7,6710
PID Fracionário	27,2186%	0%	0,3360	2,2456

A última análise de interesse a ser feita para este caso é a resposta em frequência do sistema realimentado com os controladores. Para isso será utilizada a ferramenta do diagrama de Bode para que seja possível determinar margens de ganho e fase do sistema com o intuito de entender a robustez que os controladores garantem. Na Figura 12 há uma representação da resposta em frequência dos dois sistemas e a Tabela 3 traz os resultados das margens de ganho e fase para ambos os casos.

Tabela 3 – Comparação entre margens de ganho e fase para a planta representada pela Equação (30)

Controlador	Ganho (Db)	Fase (deg)
PID Inteiro	Inf	23
PID Fracionário	127	96,8

Figura 12 – Comparação da resposta em frequência da planta representada pela Equação (30) com os controladores encontrados



Fonte: Autoria própria 2021.

Com os dados obtidos é possível concluir que o controlador fracionário, de forma geral, obteve uma resposta melhor. Seus tempos de subida e acomodação foram mais rápidos que o controlador inteiro, seu PSS foi mais baixo, houve menos oscilação e quando analisado pela resposta em frequência sua margem de fase é um pouco mais de quatro vezes maior que a proporcionada pelo controlador inteiro. Na questão da margem de ganho o controlador inteiro apresentou uma resposta melhor tendo margem de ganho infinita.

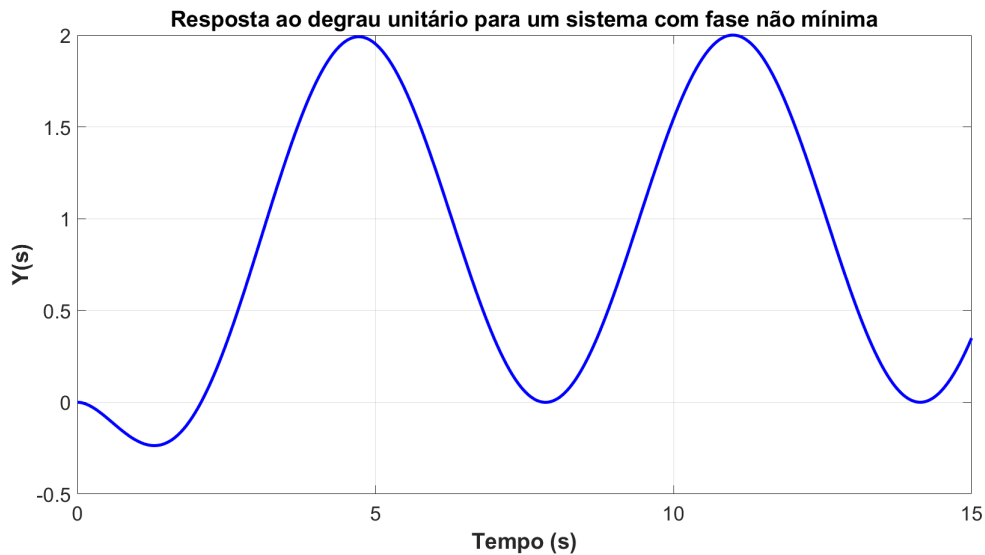
Um ponto interessante de se destacar são os ganhos obtidos em ambos os controladores. Devido a forma como a otimização funciona, levando em consideração apenas a diminuição do erro, os ganhos chegaram muito próximos dois limites superiores e inferiores do espaço de busca. Mesmo após repetidas iterações do algoritmo ainda obtiveram-se resultados nesta faixa. Isso pode implicar que o método convergiria para algum valor além dos possíveis estabelecidos, mas acabou sendo impedido pelos limites do espaço de busca.

4.2 CASO 2 - SISTEMAS DE FASE NÃO MÍNIMA

Sistemas de Fase Não Mínima são aqueles que apresentam zeros no semiplano direito, como visto na Função (31), usada de exemplo para este caso. Esse tipo de sistema não é instável por natureza, mas apresenta uma característica única de começar sua resposta a um sinal indo para a direção oposta da referência, como pode ser visto no exemplo da Figura 13.

$$G(s) = \frac{s - 1}{(s^2 + 1)(s + 1)} \quad (31)$$

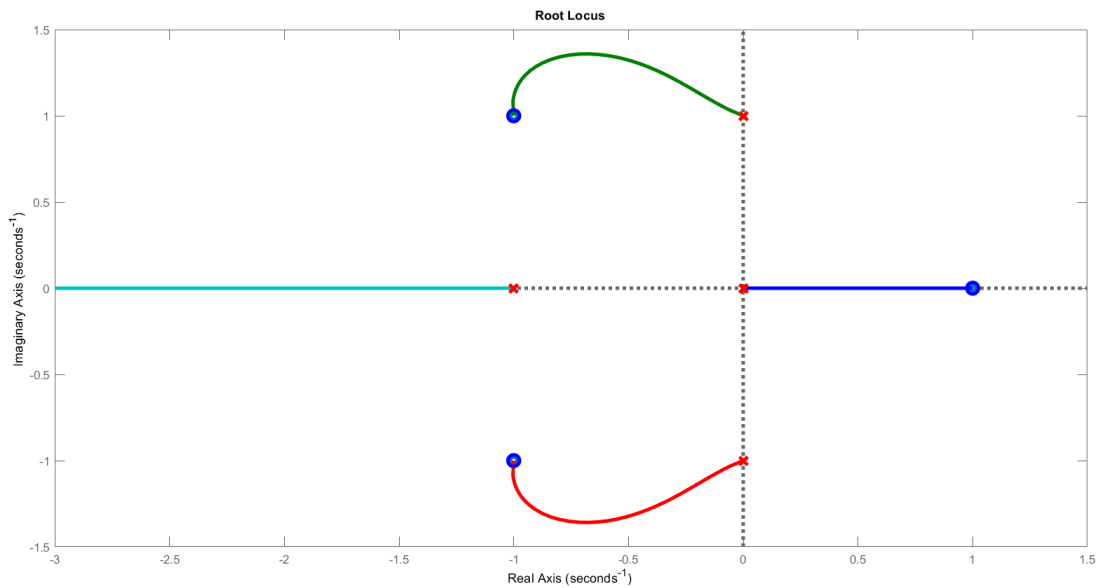
Figura 13 – Resposta da planta representada pela Equação (31) ao degrau unitário



Fonte: Autoria própria 2021.

Outro ponto que torna a análise deste sistema interessante é o comportamento do lugar das raízes quando introduzido o controlador. O PID naturalmente introduz um polo na origem e dois zeros que são posicionados, conforme os ganhos, a fim de atingir as métricas de controle desejadas. De acordo (DORF; BISHOP, 2001) o lugar das raízes sobre o eixo real está sempre à esquerda de uma quantidade ímpar de polos e zeros, logo, quando se introduz um polo na origem há uma região do lugar das raízes do sistema que se localiza inteiramente no semiplano direito, Figura 14, tornando o sistema naturalmente instável. Os dois zeros introduzidos pelo PID não são capazes de estabilizar o sistema, porque para qualquer configuração deles o polo na origem sempre irá se mover na direção do zero no semiplano direito.

Figura 14 – Lugar das raízes para o sistema (31) com o polo na origem e dois zeros no semiplano esquerdo.



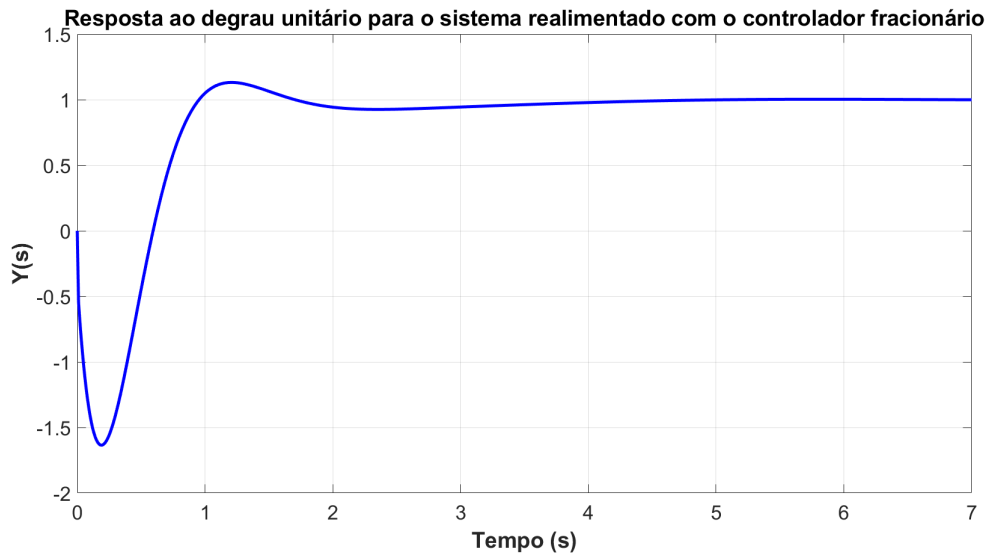
Fonte: Autoria própria 2021.

Como para este caso qualquer otimização feita com o controlador PID inteiro ainda resultaria numa planta instável, como foi demonstrado acima, optou-se por determinar apenas o controlador PID fracionário para analisar se o mesmo conseguiria então estabilizar o sistema. Os ganhos otimizados encontram-se na Tabela 4 e a resposta do sistema realimentado com o controlador é demonstrada na Figura 15:

Tabela 4 – Controladores otimizados para a planta da Equação (31)

	K_p	K_i	K_d	λ	μ
PID Fracionário	1,0897	0,5923	1,1943	1,0736	1,7238

Figura 15 – Resposta ao degrau da planta representada pela Equação (31) com o controlador fracionário encontrado



Fonte: Autoria própria 2021.

O controlador fracionário neste caso foi capaz de encontrar uma resposta estável para o sistema. Como pode ser visto na Tabela 5, há um PSS de cerca de 13%, que pode ser considerado relativamente baixo quando comparado ao sistema de malha aberta, por exemplo. Seu tempo de acomodação ficou por volta de 3 segundos, um tempo bom considerando que sem a realimentação o sistema demorava cerca de 8 segundos para completar seu primeiro período.

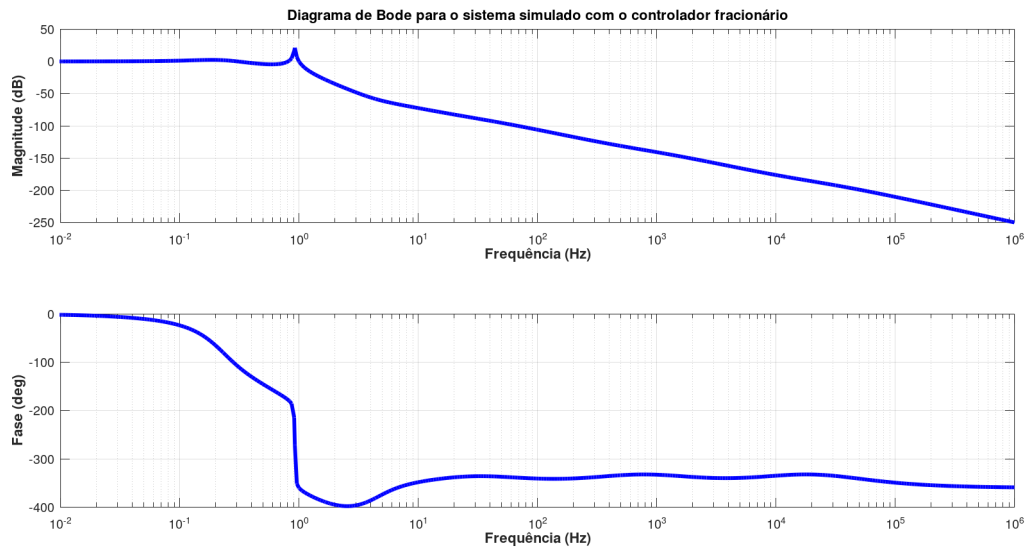
Tabela 5 – Métricas de controle para os controladores otimizados para a planta da Equação (31)

	PSS	EEE	T_r	T_s
PID Fracionário	13,1126%	0%	0,2687	3,0954

O maior problema que o controlador fracionário apresentou foi um valor próximo de -1.6 no início da resposta, porém essa é uma característica da própria planta, mesmo que fosse possível determinar um controlador inteiro para este caso ainda haveria esta 'inversão' de sentido inicial. O valor de -1.6 é relativamente alto, porém ao considerarmos que a resposta do sistema é rápida, já que em 3s o valor já estabilizou com ESS igual a zero, essa inversão passa a pesar pouco contra o controlador fracionário.

Outra análise que pode ser feita para este caso é da sua resposta em frequência para que seja possível a determinação das margens de ganho e fase do sistema que podem ajudar a esclarecer melhor a robustez do sistema. Os resultados são apresentados na Figura 16 e na Tabela 6.

Figura 16 – Resposta em frequência da planta representada pela Equação (31) com o controlador fracionário encontrado



Fonte: Autoria própria 2021.

Tabela 6 – Comparação entre margens de ganho e fase para a planta representada pela Equação (31)

Controlador	Ganho (Db)	Fase (deg)
PID Fracionário	0,538	2,31

Com os resultados obtidos é possível concluir que o controlador fracionário é sim capaz de estabilizar um sistema de fase não mínima em malha fechada. Nas condições apresentadas, ele pode vir a ser uma opção quando trabalhando com este tipo de sistema, ainda mais, quando o PID inteiro não apresentar uma resposta dentro das métricas desejadas.

4.3 CASO 3 - SISTEMA INSTÁVEL COM POLO NO SEMIPLANO DIREITO

Para estes testes a planta selecionada pode ser vista na Equação (32):

$$G(s) = \frac{1}{(s^2 + 1)(s - 1)} \quad (32)$$

O sistema apresentado é naturalmente instável devido ao polo localizado no semiplano direito que faz com que a resposta a qualquer sinal de referência cresça indefinidamente. A estratégia para controlar este sistema seria introduzir zeros capazes de eliminar o efeito deste polo instável, porém como existem também os dois polos marginalmente instáveis a tarefa torna-se impossível para o PID inteiro. Isso tudo porque para qualquer configuração dos zeros do controlador as assíntotas do lugar das raízes do sistema sempre estarão no semiplano direito.

De acordo com (DORF; BISHOP, 2001) polos de malha aberta do sistema movem-se em direção aos zeros de malha aberta quando o excitados por um ganho K, este movimento é chamado de lugar das raízes e representa o lugar geométrico do polos de malha fechada. Quando há um número de polos maior que o número de zeros é impossível que se formem pares com todos, logo, alguns dos polos procuram zeros no infinito do plano. Estes polos seguem as assíntotas de convergência, que partem de um ponto σ_a chamado de centroide (Equação (33)) com um ângulo θ_a (Equação (34)).

$$\sigma_a = \frac{\sum \text{polos} - \sum \text{zeros}}{N_{\text{polos}} - N_{\text{zeros}}} \quad (33)$$

$$\theta_a = \frac{2q + 1}{N_{\text{polos}} - N_{\text{zeros}}} 180^\circ, q = 0, 1, \dots, (N_{\text{polos}} - N_{\text{zeros}} - 1) \quad (34)$$

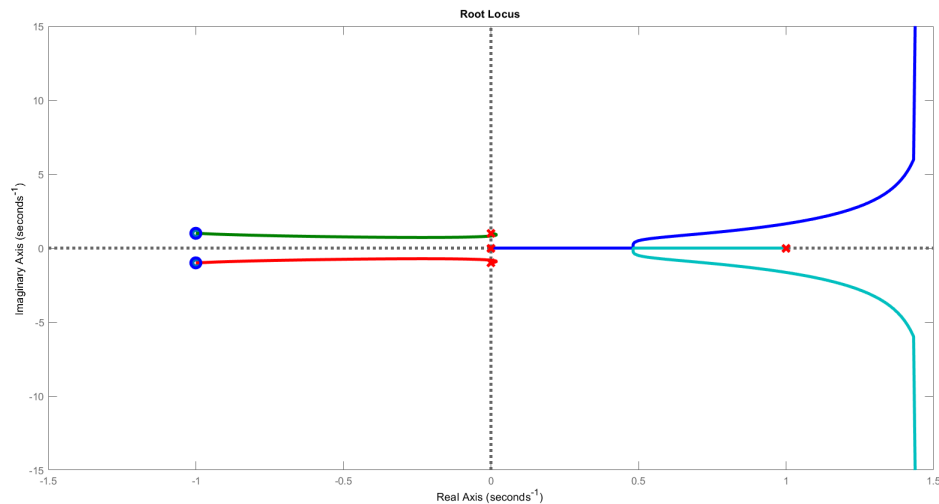
Seguindo a Equação (33) com os polos da Equação (32) mais o integrador e dois zeros genéricos provenientes do controlador, é possível desenvolver a Equação (35):

$$\sigma_a = \frac{(1 + 1i - 1i + 0) - (z_1 + z_2)}{4 - 2} \quad (35)$$

Para que as assíntotas de convergência estejam localizadas no semiplano esquerdo é necessários que os valores de z_1 e z_2 tenham parte real positiva, o que indicaria que os zeros do controlador encontram-se no semiplano direito. Que por sua vez introduziria dois zeros de fase não mínima ao sistema. A Figura 17 mostra um exemplo com do

comportamento do lugar das raízes para um controlador PID que tem um polo na origem e dois zeros complexos conjugados em $1 \pm 1j$.

Figura 17 – Comportamento do lugar das raízes para o sistema apresentado na Equação (32) com um PID genérico



Fonte: Autoria própria 2021.

Conforme foi demonstrado, nenhuma configuração do PID inteiro será capaz de encontrar uma resposta estável para o sistema da Equação (32). Com isso então este caso será focado apenas na determinação do controlador PID fracionário que possa vir a estabilizar e controlar este sistema. A Tabela 7 traz o controlador encontrado pelo método de otimização enquanto a Figura 18 mostra uma a resposta do sistema com o controlador fracionário determinado.

Tabela 7 – Controlador otimizado para a planta da Equação (32)

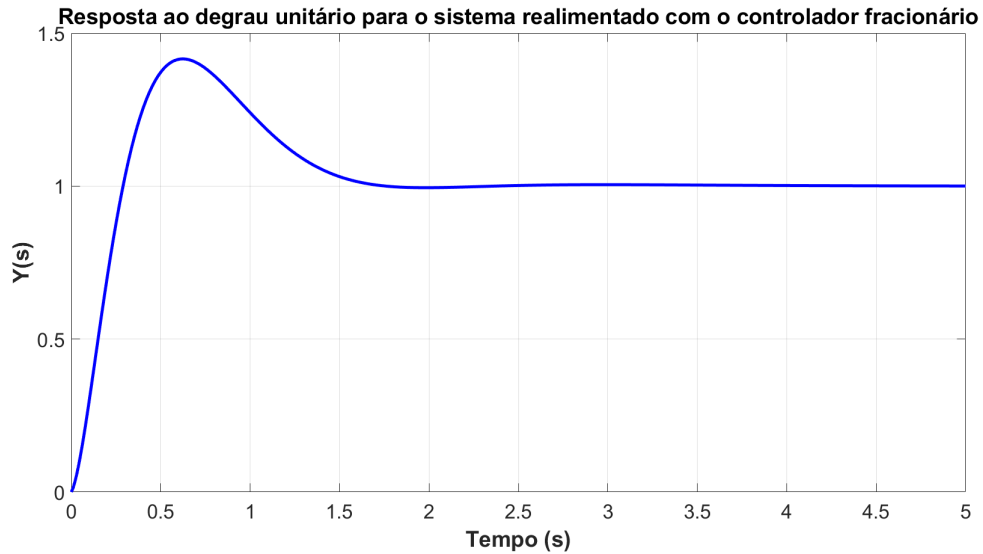
	Kp	Ki	Kd	λ	μ
PID Fracionário	7,5467	6,3198	9,6035	1,0413	1,6046

O controlador PID fracionário conseguiu estabilizar a planta e as métricas de controle avaliadas podem ser observadas na Tabela 2.

Tabela 8 – Métricas de controle para os controladores otimizados para a planta da Equação (32)

	PSS	EEE	T_r	T_s
PID Fracionário	41,5628%	0%	0,2118	1,5611

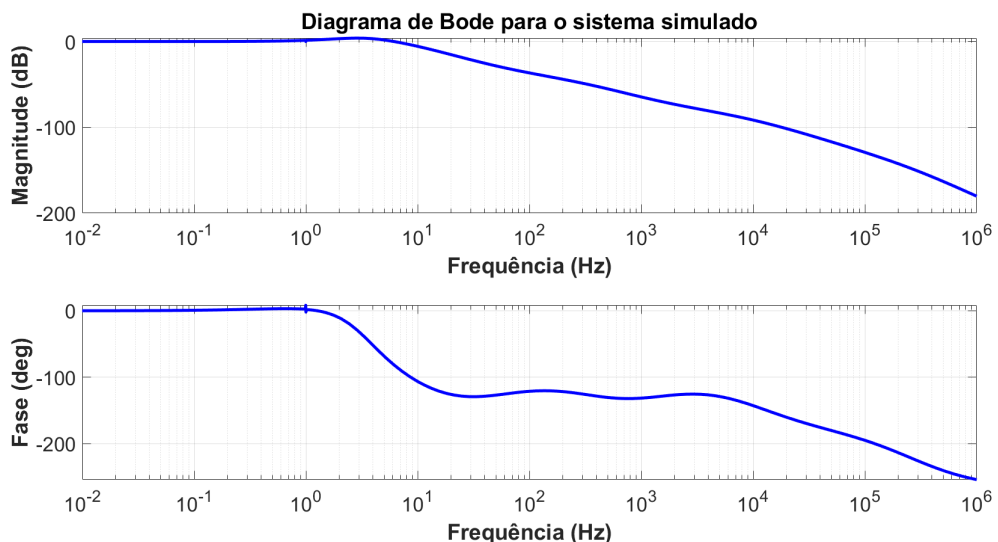
Figura 18 – Resposta ao degrau da planta representada pela Equação (32) com os controladores encontrados



Fonte: Autoria própria 2021.

Para os interesses do trabalho, a última análise a ser feita é a da resposta em frequência do sistema, na Figura 19 mostra a resposta com o controlador fracionário e a Tabela 9 mostra as margens de ganho e fase para o caso analisado.

Figura 19 – Resposta resposta em frequência da planta representada pela Equação (32) com os controladores encontrados



Fonte: Autoria própria 2021.

O controlador fracionário foi capaz de controlar este sistema naturalmente instável, diferentemente do PID inteiro, como foi demonstrado. Com o controlador

Tabela 9 – Margens de ganho e fase para a planta representada pela Equação (32)

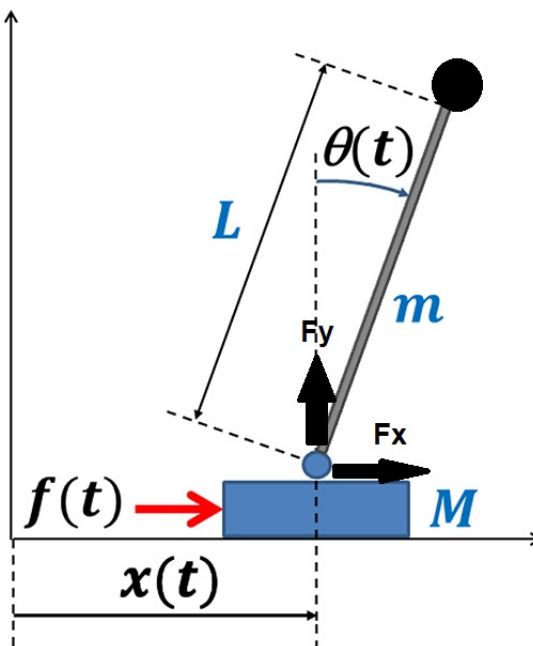
Controlador	Ganho (Db)	Fase (deg)
PID Fracionário	117	98,4

fracionário a resposta apresentou métricas de controle razoáveis quando comparadas com outros valores obtidos durante este trabalho. A análise em frequência também garantiu margens largas que mostram a robustez do sistema a variações.

5 ESTUDO DE CASO - PÊNULO INVERTIDO

O pêndulo invertido pode ser considerado um dos casos clássicos para estudos e ensino de controle (BOUBAKER, 2012), o problema consiste em equilibrar um pêndulo apoiado apenas pela sua base sem que o mesmo caia mantendo sempre um ângulo desejado, a Figura 20 traz uma representação do caso.

Figura 20 – Diagrama de corpo livre do pêndulo invertido



Fonte: Adaptado de (MORASSO et al., 2019)

O objetivo com este estudo de caso é aplicar o que foi desenvolvido até agora em um problema real onde há solução pelo PID inteiro, como foi demonstrado por (MORASSO et al., 2019) e (BOUBAKER, 2012), e comparar o desempenho do controlador PID fracionário para esta mesma situação. Ainda serão utilizados os métodos de otimização e toda a metodologia desenvolvida no Capítulo 3, porém desta vez a solução será comparada no domínio do tempo com a resolução das equações diferenciais que regem o problema.

5.1 EQUACIONAMENTO

Inicialmente é necessário conseguir estabelecer alguma relação entre a força aplicada pelo atuador e o movimento angular do pêndulo. Para isso serão utilizados conceitos da mecânica e as leis de Newton para translação e rotação, porém antes de iniciar o equacionamento é preciso definir os parâmetros físicos do problema:

- J - Momento de Inércia
- M_p - Massa do pêndulo
- M_a - Massa do atuador
- L - Comprimento total do pêndulo
- θ - Ângulo entre o pêndulo e o eixo vertical
- f - Força aplicada pelo atuador

Tendo definidos os parâmetros também são necessárias algumas considerações, neste caso:

- O movimento está contido apenas no plano XY.
- Não há movimento na vertical, ou seja, $y(t)=0$.
- A massa concentrada na ponta do pêndulo é consideravelmente maior que a massa da barra.

Tendo os parâmetros físicos e as considerações definidas pode-se, então iniciar o equacionamento. O primeiro passo é a definição da posição de controle (x_c, y_c) , neste caso localizada na ponta do pêndulo a uma distância L da base de apoio:

$$\begin{aligned}x_c &= L \sin(\theta) + x(t) \\y_c &= L \cos(\theta)\end{aligned}\tag{36}$$

Com as equações da posição definidas é possível deriva-las duas vezes a fim de encontrar a aceleração para o eixo X e Y no ponto de controle:

$$\begin{aligned}\ddot{x}_c &= L(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) + \ddot{x} \\ \ddot{y}_c &= -L(\ddot{\theta} \sin(\theta) - \dot{\theta}^2 \cos(\theta))\end{aligned}\tag{37}$$

Então definem-se as equações para a força aplicada pelo atuador na base do pêndulo:

$$\begin{aligned}F_x &= M_p \ddot{x}_c = M_p [L(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) + \ddot{x}] \\ F_y &= M_p \ddot{y}_c - M_p g = -M_p L(\ddot{\theta} \sin(\theta) - \dot{\theta}^2 \cos(\theta)) - M_p g\end{aligned}\tag{38}$$

Estabelecidas as relações de translação é necessário descrever o movimento rotacional do pêndulo. Para isso serão utilizados os conceitos das leis de Newton para a rotação:

$$\sum T = L(F'_x - F'_y) = J\ddot{\theta}\tag{39}$$

Aqui F'_x e F'_y são as componentes da força que efetivamente geram torque no sistema, estas podem ser definidas conforme as equações a seguir:

$$\begin{aligned}F'_x &= F_x \cos(\theta) \\ F'_y &= F_y \sin(\theta)\end{aligned}\tag{40}$$

Combinando as Equações 38 e 40 e substituindo-as na Equação 39 obtém-se a Equação 41, após uma manipulação algébrica das variáveis:

$$\ddot{\theta}(J - M_p L^2) = M_p L \ddot{x} \cos(\theta) + M_p g L \sin(\theta) \quad (41)$$

Utilizando a componente da força horizontal da Equação 38 e a Equação 41 chega-se no sistema:

$$\begin{cases} F_x = M_p [L(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) + \ddot{x}] \\ \ddot{\theta}(J - M_p L^2) = M_p L \ddot{x} \cos(\theta) + M_p g L \sin(\theta) \end{cases} \quad (42)$$

Como o objetivo é obter uma relação apenas da força aplicada pelo atuador F_x com o ângulo do pêndulo θ pode-se isolar o termo $M\ddot{x}$ que está presente em ambas as equações. O resultado então é a Equação 43:

$$(J - M_p L^2 \sin^2(\theta)) \ddot{\theta} = F_x L \cos(\theta) - M_p L^2 \dot{\theta}^2 \sin(\theta) \cos(\theta) + M_p g L \sin(\theta) \quad (43)$$

Da maneira como está apresentada, a Equação 43 é não linear, o que impede que a determinação de uma função de transferência para que seja possível aplicar o método de otimização utilizado até agora no trabalho. A solução encontrada é, então, linearizar a Equação 43 ao redor de θ igual a zero, que corresponderia ao pêndulo completamente na vertical, ou seja, na posição de equilíbrio desejada para o problema.

$$\begin{cases} \theta \approx 0 \\ \dot{\theta}^2 \approx 0 \\ \sin(\theta) \approx \theta \\ \cos(\theta) \approx 1 \end{cases} \quad (44)$$

Tendo as condições de linearização definidas e utilizando-as na Equação 43 obtém-se sua versão linearizada:

$$J \ddot{\theta} - M_p g L \theta = F_x L \quad (45)$$

A partir da Equação 45 é possível com a aplicação da transformada de Laplace, usando condições iniciais nulas, obter uma função de transferência:

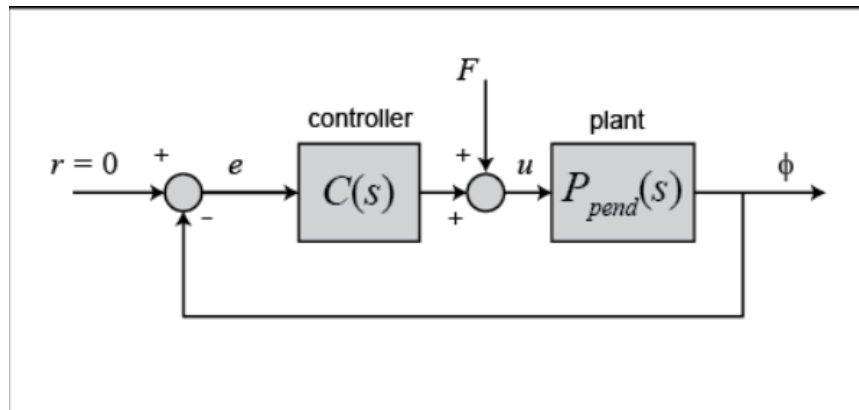
$$\frac{\theta(s)}{F_x(s)} = \frac{L}{J s^2 - M_p g L} \quad (46)$$

5.2 RESULTADOS DA OTIMIZAÇÃO NO ESPAÇO DE LAPLACE

Com a função de transferência definida é possível então determinar os ganhos dos controladores. Porém para este caso não será utilizado o sistema realimentado padrão usado nas seções anteriores. O objetivo do problema, no ponto de vista de controle, é garantir que o pêndulo sempre volte para a posição de controle (θ igual a 0) quando sofrer qualquer tipo de perturbação. Logo, a referência de entrada será zero

porém serão introduzidas perturbações que movimentam o pêndulo, como pode ser visto no modelo do diagrama de blocos desenvolvido por (MICHIGAN et al.,) na Figura 21:

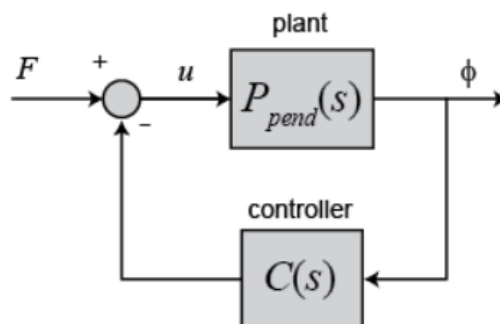
Figura 21 – Malha de controle realimentada com o distúrbio de movimentação



Fonte: (MICHIGAN et al.,)

É possível então rearranjar os blocos do sistema de forma a se obter a seguinte configuração:

Figura 22 – Malha de controle realimentada com o controlador no caminho de realimentação



Fonte: (MICHIGAN et al.,)

Para o sistema da Figura 22 uma perturbação move a planta do sistema, porém o controlador atua de forma a zerar a saída da mesma trazendo novamente θ para a posição de equilíbrio. Os seguintes parâmetros físicos foram definidos a fim de simular o problema:

- $L = 0,3 \text{ m}$
- $M_p = 0,2 \text{ Kg}$
- $g = 9,81 \text{ m/s}^2$
- $J = 0,006 \text{ kg.m}^2$

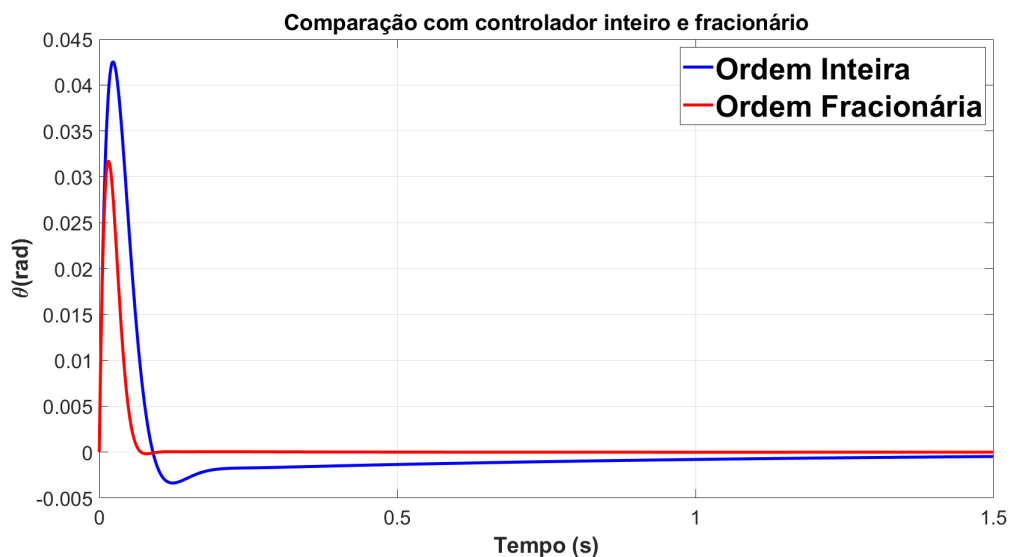
Os resultados dos controladores otimizados podem ser encontrados na Tabela 10. É importante ressaltar que para este estudo o espaço de busca dos ganhos foi modificado de 0 até 500, pois observou-se que ganhos de no máximo 10 não estavam trazendo uma resposta adequada em ambos os casos. Como forma de se obter uma comparação mais justa entre os dois controladores e conseguir analisar melhor o efeito dos termos fracionários, optou-se por utilizar o mesmo valor de K_p , neste caso otimizado para o caso inteiro, no controlador fracionário.

Tabela 10 – Ganhos determinados para o problema do pêndulo invertido

	K_p	K_i	K_d	λ	μ
PID Inteiro	500	500	15.4044	-	-
PID Fracionário	500	500	31,4229	0,0164	0,9031

Com os controladores definidos é possível então simular o sistema, para isso será utilizada então uma perturbação impulsiva de magnitude mil como forma de avaliar a resposta do modelo linearizado. Os resultados são então mostrados na figura a seguir:

Figura 23 – Resposta da função de transferência do pêndulo invertido a uma perturbação de caráter impulsivo com os controladores Inteiro e Fracionário



Fonte: Autoria própria 2021.

Para este caso não é possível aplicar as mesmas métricas de controle obtidas com a resposta ao degrau unitário, porém ainda conseguem-se obter algumas informações sobre o sistema a fim de comparar o desempenho dos dois controladores. Estas medidas são mostradas na Tabela 11. Onde $\text{Max } \theta$ representa o ângulo máximo do pêndulo e T_p representa o tempo até o primeiro pico.

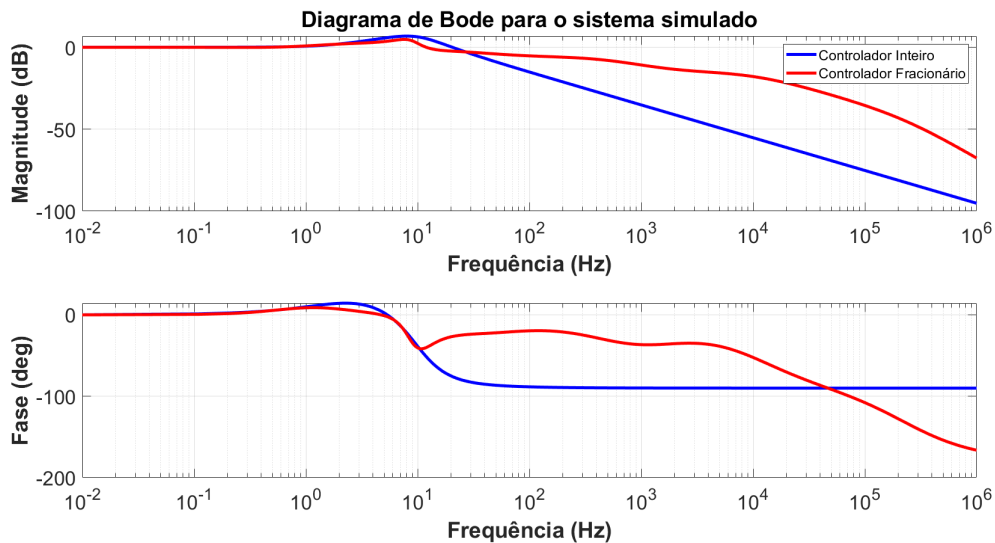
Tabela 11 – Métricas de controle para os controladores otimizados para o pêndulo invertido

	Max θ	EEE	T_p	T_s
PID Inteiro	0,043 rad	0%	0,1	1,0
PID Fracionário	0,033 rad	0%	0,05	0,1

Os resultados para ambos os controladores aqui foram parecidos e satisfatórios, o controlador fracionário conseguiu um ângulo máximo, tempo de pico e tempo de acomodação menores que o controlador inteiro, porém estas diferenças ainda não são tão gritantes ainda mais considerando ângulos tão pequenos e tempos tão curtos. Dentro dos espaço de Laplace, com a equação linearizada, ambos os controladores foram capazes de oferecer respostas adequadas ao problema, mostrando-se capazes de satisfazer os requisitos de controle.

A última análise a ser feita nesta etapa é em relação a resposta em frequência dos controladores. Suas respostas podem ser encontradas na Tabela 12 e na Figura 24.

Figura 24 – Comparação da resposta em frequência do para o sistema do pêndulo invertido



Fonte: Autoria própria 2021.

Tabela 12 – Comparação entre margens de ganho e fase para o sistema do pêndulo invertido

Controlador	Ganho (Db)	Fase (deg)
PID Inteiro	Inf	104
PID Fracionário	Inf	140

Mais uma vez é notável que ambos os controladores foram capazes de obter

respostas adequadas na análise em frequência. Todos estes resultados já eram de certa forma esperados uma vez que é sabido que o controlador PID inteiro consegue controlar este sistema, como foi comentado no início do capítulo. Uma vez que o controlador inteiro é um subconjunto do controlador fracionário então é razoável assumir que a versão fracionária também seria capaz de controlar e oferecer bom desempenho.

5.3 SIMULAÇÃO COM SIMULINK

Como forma de validar os controladores em um teste mais completo o objetivo agora é simular os mesmos no domínio do tempo utilizando as equações completas que descrevem o sistema. Para isso será utilizada a ferramenta de simulação do Matlab, o *Simulink*, que permite através de uma linguagem interpretada estabelecer as relações entre as equações diferenciais que representam o problema e também partes como o controlador e os sinais de entrada.

Inicialmente é necessário que sejam definidas as equações que serão utilizadas dentro da simulação, neste caso são aquelas definidas na subseção 5.1, com a adição também de uma equação para que se estabeleça a relação entre a força aplicada pelo atuador e a posição da base onde o pêndulo está apoiado. A equação pode ser facilmente deduzida baseada no somatório de forças na horizontal, como pode ser visto a seguir:

$$\sum F = f(t) + F_x(t) + b\dot{x} = M_a\ddot{x} \quad (47)$$

Como o interesse é apenas no termo \ddot{x} , isola-se ele dando origem à:

$$\ddot{x} = \frac{f(t) + F_x(t) + b\dot{x}}{M_a} \quad (48)$$

Agora então é possível estabelecer o conjunto de equações que rege o movimento do pêndulo mais atuador. Sendo estas colocadas a seguir:

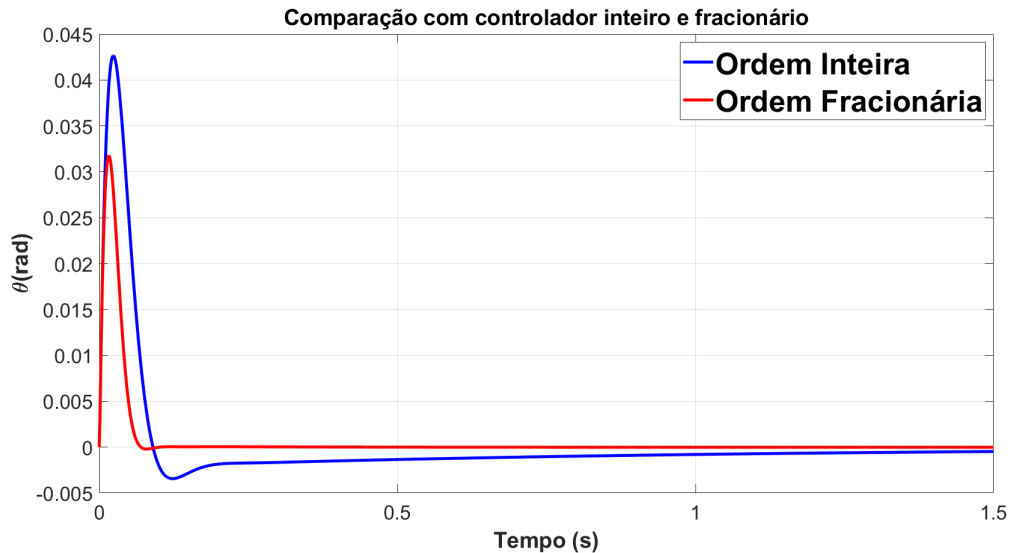
$$\begin{cases} F_x(\theta, \dot{\theta}, \ddot{\theta}, \ddot{x}) = M_p[L(\ddot{\theta}\cos(\theta) - \dot{\theta}^2\sin(\theta)) + \ddot{x}] \\ \ddot{\theta}(\theta, \ddot{x}) = \frac{M_pL\ddot{x}\cos(\theta) + M_pgL\sin(\theta)}{(J - M_pL^2)} \\ \ddot{x}(f, F_x, \dot{x}) = \frac{f(t) + F_x(t) + b\dot{x}}{M_a} \end{cases} \quad (49)$$

É possível notar que todas as equações do Conjunto 49 são acopladas, além de não lineares tornando sua solução possível apenas por métodos numéricos. A ferramenta do *Simulink* entra nesta etapa já que é possível estabelecer de forma fácil e intuitiva o conjunto como uma série de blocos conectados e suas saídas podem ser facilmente realimentadas umas nas outras. Para os propósitos de controle, o interesse está na variável de estado da posição angular do pêndulo θ .

Após a resolução do conjunto de equações realimentadas com os controladores obtidos anteriormente (Tabela 10) obtiveram-se as respostas a um distúrbio impulsivo

vistas na Figura 25:

Figura 25 – Comparação com controlador inteiro e fracionário para uma perturbação impulsiva no pêndulo invertido no *Simulink*

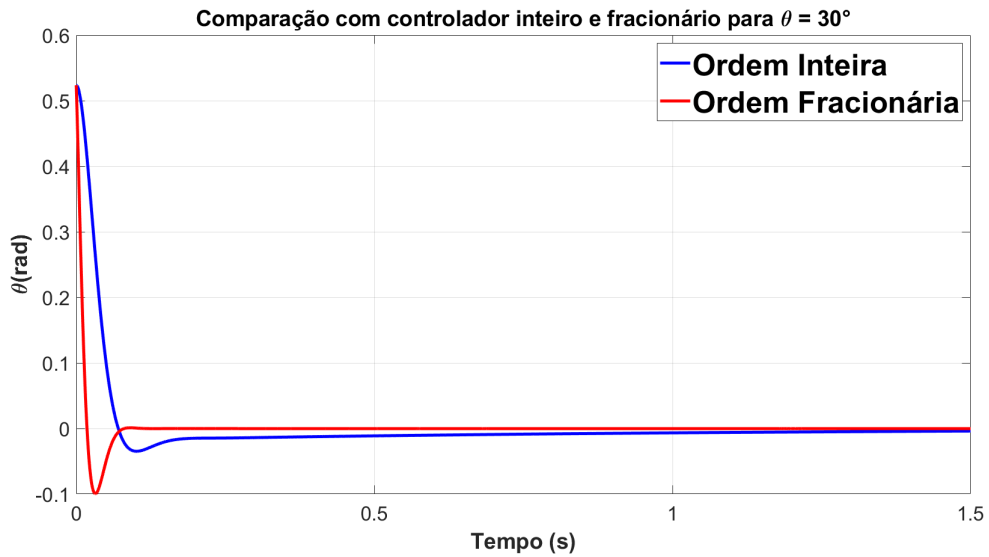


Fonte: Autoria própria 2021.

Analisando a Figura 25 é possível ver que tanto o modelo linearizado estava correto em sua previsão do comportamento do pêndulo como também os controladores foram capazes de controlar o pêndulo mesmo quando submetidos uma simulação com as equações completas. Novamente é notável que o controlador fracionário conseguiu uma resposta mais rápida e menos oscilatória quando comparada com o controlador inteiro.

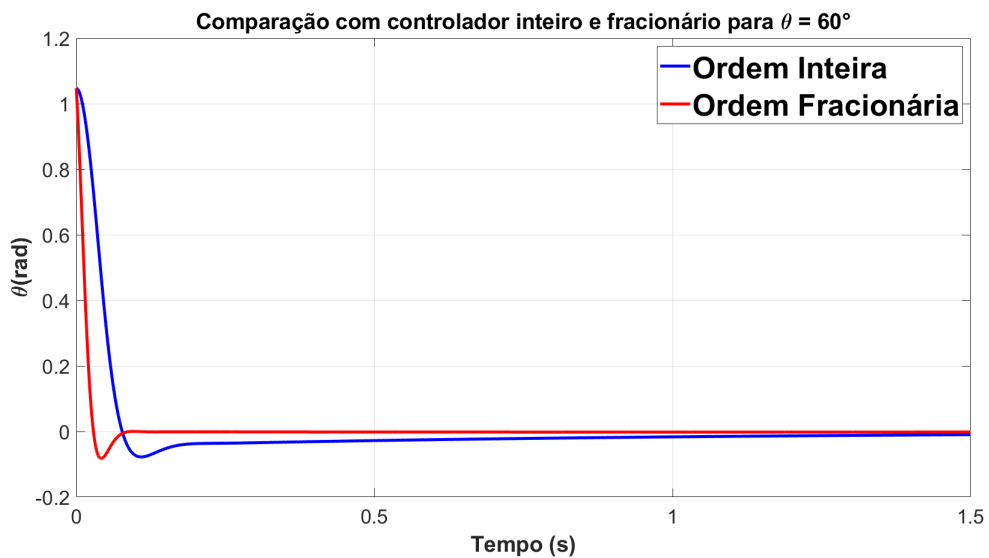
Há também a possibilidade de mudarem-se as condições iniciais do modelo para analisar se os controladores ainda são capazes de manter o pêndulo estabilizado, mesmo que longe da condição de linearização para a qual eles foram desenvolvidos. As Figuras 26, 27 e 28 trazem uma comparação da resposta do sistema realimentada com os controladores, mas desta vez com condição inicial de θ igual a 30° , 60° e 75° sem a utilização da resposta impulsiva para gerar movimento.

Figura 26 – Comparação com controlador inteiro e fracionário para o pêndulo invertido quando a condição inicial é $\theta = 30^\circ$



Fonte: Autoria própria 2021.

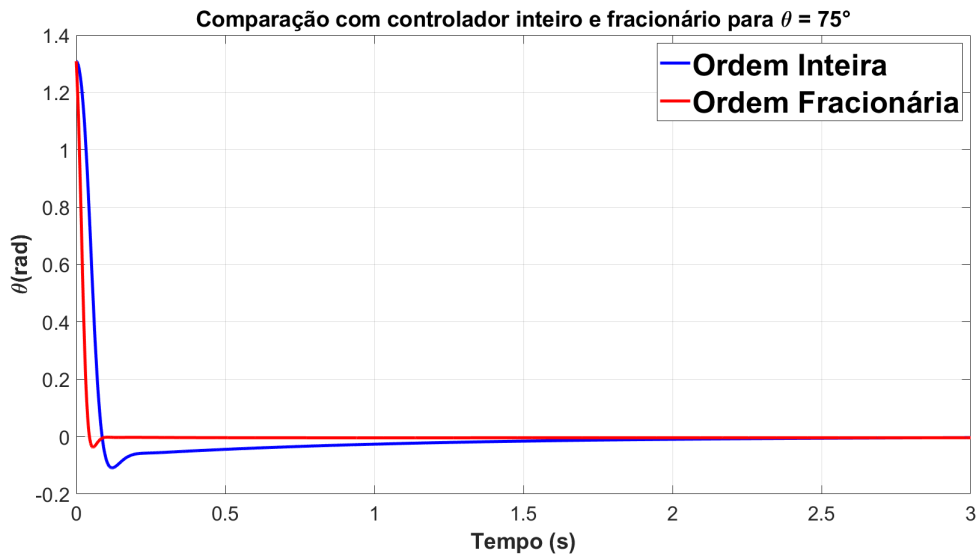
Figura 27 – Comparação com controlador inteiro e fracionário para o pêndulo invertido quando a condição inicial é $\theta = 60^\circ$



Fonte: Autoria própria 2021.

Como é possível observar pelos resultados observados ambos os controladores conseguiram estabilizar o pêndulo na posição desejada. Em todas as condições iniciais o controlador fracionário manteve a estabilidade do sistema além de apresentar uma resposta mais rápida para estabilização porém também oscilou mais que a contraparte inteira.

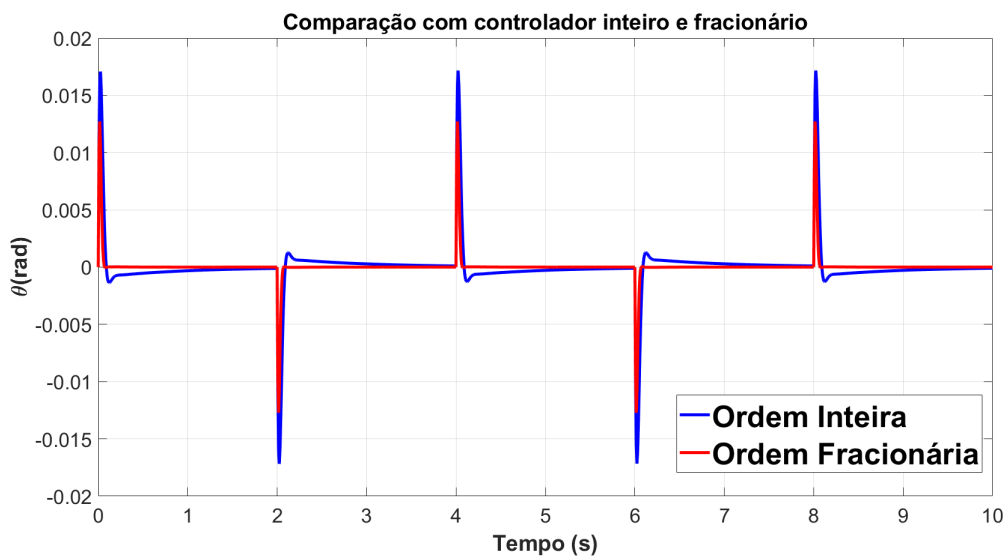
Figura 28 – Comparação com controlador inteiro e fracionário para o pêndulo invertido quando a condição inicial é $\theta = 75^\circ$



Fonte: Autoria própria 2021.

O último teste que será realizado tem como objetivo avaliar a robustez do sistema a diversas perturbações impulsivas durante o movimento. Tais perturbações terão sentidos opostos separados sempre por dois segundos, dando tempo para o pêndulo começar a se estabilizar, para testar se o sistema consegue se adaptar e manter o pêndulo na posição vertical. Os resultados são apresentados na Figura 29.

Figura 29 – Comparação entre os controladores quando o sistema é submetido a diversas perturbações impulsivas.



Fonte: Autoria própria 2021.

Novamente ambos os controladores foram capazes de estabilizar o sistema mesmo quando submetidos às perturbações, porém o controlador fracionário apresentou uma resposta mais rápida estabilizando o pêndulo em ângulo zero sem mesmo oscilar ao redor da referência.

6 CONCLUSÕES

Com todos os capítulos expostos chega-se então nas considerações finais deste trabalho. O objetivo geral era estudar e comparar o desempenho de um controlador clássico e muito utilizado na engenharia, o controlador PID, com um tipo de controlador que não é visto na graduação e ainda tem usos muito restritos no âmbito da engenharia e controle. Para que fosse possível então desenvolver este trabalho primeiro necessitou-se estudar toda a teoria que dizia respeito ao controlador fracionário, voltando às raízes matemáticas do problema e estudando uma versão do cálculo que também não é abordada na graduação. Como forma de determinar os parâmetros do controlador optou-se pelo uso de métodos de otimização, primeiramente por não ser possível aplicar técnicas do controle clássico neste tipo de problema e também porque dentro da engenharia métodos de otimização são um tópico de grande interesse.

Tendo em vista tudo que foi desenvolvido no trabalho é possível afirmar que os objetivos foram cumpridos. Nos primeiros capítulos o texto trouxe uma extensa discussão sobre como desenvolve-se um controlador fracionário desde a matemática até sua implementação numérica. Foram desenvolvidos algoritmos de otimização para aplicação focada num problema de engenharia, além de todo o estudo sobre como estas técnicas funcionam e como é possível as implementar numericamente. Por fim trouxeram-se as comparações entre os dois modelos de controlador de interesse. O Capítulo 4 trouxe uma série de casos onde a aplicação de um controlador PID clássico era complexa, e muitas vezes impraticável, como foram os casos mostrados nas seções 4.2 e 4.3. Nestes foi possível ver que o controlador fracionário é capaz de resolver problemas complexos que fogem do escopo de um controlador PID padrão, além de apresentar parâmetros de controle dentro de limites considerados bons.

O último ponto de interesse deste trabalho era a aplicação do controlador fracionário em um problema conhecido, que possuísse solução com um PID inteiro e que permitisse explorar ou o controlador fracionário para um caso um pouco mais complexo. Para isso optou-se pelo estudo do pêndulo invertido, um problema que, inicialmente, parece ser simples, porém, traz grandes dificuldades em sua resolução devido às equações não lineares acopladas do modelo. Para sua resolução, utilizou-se da ferramenta de simulação *simulink* que por sua vez também gerou uma nova carga de estudos sobre modelagem e resolução numérica de equações não lineares. Mas ao fim foi possível novamente simular o sistema com os dois controladores, mostrando que a modelagem desenvolvida durante na Seção 5.1 estava coerente e também que novamente o controlador fracionário era superior trazendo uma resposta com menos

variação e que atingia o controle desejado.

Em relação a trabalhos futuros ainda há muito o que ser explorado dentro do tema de controladores fracionários. Há a possibilidade de estudar outras técnicas de determinação do mesmo, com funções de custo customizadas. Pode-se ainda explorar dentro das simulações como modelar plantas com características fracionárias comparando o desempenho do controlador inteiro e do controlador fracionário com a mesma. Partindo para uma aplicação mais prática ainda há muito o que se estudar sobre a implementação deste tipo de controlador em um ambiente real, aplicações dele dentro do mundo digital e como o mesmo se comporta no domínio Z. Todas essas aplicações podem ser exploradas futuramente, seja pelo autor do trabalho ou por pessoas que venham a se interessar pelo tema.

REFERÊNCIAS

ANDRADE, A. M. F. de; LIMA, E. G.; DARTORA, C. A. Uma introdução ao cálculo fracionário e suas aplicações em circuitos elétricos. **Revista Brasileira de Ensino de Física**, v. 40, n. 3, p. e3314–2 – e3314–8, 09 2018.

ARORA, J. et al. Global optimization methods for engineering applications: a review. **Structural optimization**, v. 9, n. 3-4, p. 137–159, 1995.

BABU, F. A. S.; CHIRANJEEVI, S. B. Implementation of fractional order pid controller for an avr system using ga and aco optimization techniques. **IFAC-PapersOnLine**, v. 49, n. 1, p. 456–461, 2016.

BOUBAKER, O. The inverted pendulum: A fundamental benchmark in control theory and robotics. In: **International Conference on Education and e-Learning Innovations**. [S.l.: s.n.], 2012. p. 1–6.

CARVALHO, M. D.; OTTONI, J. E. Introdução ao Cálculo Fracionário com aplicações. **Revista de Matemática de Ouro Preto**, v. 5, n. 1, p. 50–77, 2018.

CHIPIPOP, B. et al. Designing of fractional order pid controller for hard-to-be-control plant with time delay. In: **PROCEEDINGS OF THE 2019 JOINT INTERNATIONAL CONFERENCE ON DIGITAL ARTS, MEDIA AND TECHNOLOGY WITH ECTU NORTHERN SECTION CONFERENCE ON ELECTRICAL, ELECTRONICS, COMPUTER AND TELECOMUNICATIONS ENGINEERING (ECTI DAMT-NCON)**. IEEE, 2019. p. 73–76. Disponível em: <https://doi.org/10.1145/268437.268460>.

DAS, S. et al. Revisiting oustaloup's recursive filter for analog realization of fractional order differintegrators. In: **PROCEEDINGS 2011 INTERNATIONAL CONFERENCE ON ENERGY, AUTOMATION AND SIGNAL**. IEEE, 2011. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/document/6147190>.

DENIZ, F. N. et al. Tuning of fractional order pid controllers based on integral performance criteria using fourier series method. **IFAC-PapersOnLine**, Elsevier, v. 50, n. 1, p. 8561–8566, 2017.

DORF, R. C.; BISHOP, R. H. **Sistemas de Controle Modernos**. 1 Lake St Upper Saddle River, United States: Addison Wesley Longman, 2001.

DU, M.; WANG, Z.; HU, H. Measuring memory with the order of fractional derivative. **Scientific reports**, Nature Publishing Group, v. 3, n. 1, p. 1–3, 2013.

DUARTE, F. B.; MACHADO, J. T. Fractional dynamics in the describing function analysis of nonlinear friction. **IFAC Proceedings Volumes**, Elsevier, v. 39, n. 11, p. 218–223, 2006.

DULĂU, M.; GLIGOR, A.; DULĂU, T.-M. Fractional order controllers versus integer order controllers. **Procedia Engineering**, Elsevier, v. 181, p. 538–545, 2017.

GRANDI, G. L.; TRIERWEILER, J. O. Tuning of fractional order pid controllers based on the frequency response approximation method. **IFAC-PapersOnLine**, v. 52, n. 1, p. 982–987, 2019.

GUTIERREZ, R. E.; ROSARIO, J. M.; MACHADO, J. T. Fractional order calculus: basic concepts and engineering applications. **Mathematical problems in engineering**, v. 2010, p. 0–19, 2010.

KOZIEL, S.; YANG, X. **Computational optimization, methods and algorithms**. Verlag Berlin Heidelberg: Springer, 2011.

LIU, L.; ZHANG, S. Robust fractional-order pid controller tuning based on bode's optimal loop shaping. **Complexity**, v. 2018, p. 0–14, 2018.

MICHIGAN, U. of; UNIVERSITY, C. M.; DETROIT MERCY, U. of. **Inverted Pendulum: PID Controller Design**. Disponível em: <https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=ControlPID>.

MORASSO, P. et al. Stabilization of a cart inverted pendulum: Improving the intermittent feedback strategy to match the limits of human performance. **Frontiers in Computational Neuroscience**, v. 13, p. 16, 2019. Disponível em: <https://www.frontiersin.org/article/10.3389/fncom.2019.00016>.

OUSTALOUP, A. et al. Frequency-band complex noninteger differentiator: characterization and synthesis. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, IEEE, v. 47, n. 1, p. 25–39, 2000.

PODLUBNY, I. Fractional order systems and $\pi\lambda d\mu$ -controllers. **IEEE Transactions on Automatic Control**, v. 44, p. 208–214, 1999.

TORVIK, P.; BAGLEY, R. On the appearance of the fractional derivative in the behavior of real materials. **Journal of Applied Mechanics**, v. 51, p. 4, 06 1984.

VINAGRE, B. et al. Some approximations of fractional order operators used in control theory. **Fractional Calculus Applied Analysis (FCAA)**, v. 3, p. 1–16, 01 2000.

ZHANG, L. et al. Fractional-order modeling and state-of-charge estimation for ultracapacitors. **Journal of Power Sources**, Elsevier, v. 314, p. 28–34, 2016.

ZHAO, C.; XUE, D.; CHEN, Y. A fractional order pid tuning algorithm for a class of fractional order plants. In: **PROCEEDINGS IEEE INTERNATIONAL CONFERENCE MECHATRONICS AND AUTOMATION**. IEEE, 2005. v. 1, p. 216 – 221. Disponível em: <https://ieeexplore.ieee.org/document/1626550>.

APÊNDICE A

.1 ALGORITMO PARA DETERMINAÇÃO DOS GANHOS ÓTIMOS

```

clc; clear; close all;
s = tf('s');

%-----%
%Par metros do Filtro
N = 3;
wb = 10^-5;
wh = 10^5;

%Par metros de simula o
t = 0:0.05:10;
vecSim = ones(length(t),1);

%planta
G_p = (-s+1)/((s^2+1)*(s+1))
G_f = 1

nVar = 3
%Numero de variaveis de decisao
if (nVar == 5)
    costFunction = @(x,G_p) ITAEOustaloup_Control(x,G_p,G_f,N,wb,wh);
    varMin = [0.01 0.01 0.01 0.01 0.01];%m nimo valor que uma
        vari vel pode assumir
    varMax = [ 10 10 10 2 2];%m ximo valor que uma
        vari vel pode assumir
end

if (nVar == 3)
    costFunction = @(x,G_p) ITAE_Control(x,G_p,G_f);
    varMin = [ 0.01 0.01 0.01];%m nimo valor que uma
        vari vel pode assumir
    varMax = [10 10 10];%m ximo valor que uma vari vel pode
        assumir
end
varSize = [1 nVar];%Matriz de vari veis de decis o
maxVelocity = 0.1*(varMax-varMin);
minVelocity = -maxVelocity;

%Constriction coefficients
kappa = 1;

```

```

phi1 = 2.05;
phi2 = 2.05;
phi = phi1 + phi2;
chi = 2*kappa / abs(2-phi-sqrt(phi*phi -4*phi));

%%Par metros do problema
maxIteration = 400;%m ximo de itera es
nPop          = 25;%tamanho da popula o
w             = chi;%coeficiente de in rcia
wDamp         = 1;%Fator de amortecimento do coeficiente de in rcia
c1            = chi*phi1;%coeficiente pessoal de acelera o
c2            = chi*phi2;%coeficiente social de acelera o

%%Inicializa o

%Template da struct particle
empty_particle.Position = [];
empty_particle.Velocity = [];
empty_particle.Cost     = [];
empty_particle.Best.Position = [];
empty_particle.Best.Cost     = [];

%cria o vetor da popula o
particle = repmat(empty_particle,nPop,1);%vetor de empty_particle

%Inicializa o melhor resultado global
GlobalBest.Cost = inf;

%Inicializa a popula o
for i=1:nPop
    %gera uma solu o aleat ria dentro dos limites do dom nio
    particle(i).Position = unifrnd(varMin,varMax,varSize);

    %inicializa a velocidade
    particle(i).Velocity = zeros(varSize);

    %Avalia a solu o encontrada
    particle(i).Cost = costFunction(particle(i).Position,G_p); %TAE

    %atualiza a melhor resposta particular
    particle(i).Best.Position = particle(i).Position;
    particle(i).Best.Cost     = particle(i).Cost;

    %atualiza a melhor resposta global
    if (particle(i).Best.Cost < GlobalBest.Cost)

```

```

    GlobalBest = particle(i).Best;
end

end

bestCosts = zeros(maxIteration,1);%vetor para guardar oos melhores valores
em cada itera o

tic%inicia a medi o de tempo do programa

%%Itera o (main loop)
for it=1:maxIteration

for i=1:nPop
%atualiza a velocidade
particle(i).Velocity = w*particle(i).Velocity + c1*rand(varSize).*(
particle(i).Best.Position - particle(i).Position) + c2*rand(varSize)
.*(GlobalBest.Position - particle(i).Position);

%avalia a velocidade dentro dos limites do problema
particle(i).Velocity = max(particle(i).Velocity , minVelocity);
particle(i).Velocity = min(particle(i).Velocity , maxVelocity);

%atualiza a posi o
particle(i).Position = particle(i).Position + particle(i).Velocity;

%avalia a posi o dentro dos limites do problema
particle(i).Position = max(particle(i).Position , varMin);
particle(i).Position = min(particle(i).Position , varMax);

%avalia a nova posi o
particle(i).Cost = costFunction(particle(i).Position ,G_p); %ITAE

%atualiza o melhor resultado pessoal
if (particle(i).Cost < particle(i).Best.Cost)
particle(i).Best.Position = particle(i).Position;
particle(i).Best.Cost = particle(i).Cost;

%atualiza o melhor resultado global
if (particle(i).Best.Cost < GlobalBest.Cost)
GlobalBest = particle(i).Best;
end

end

end

end

```

```

%atualiza o melhor valor de cada iterao
bestCosts(it) = GlobalBest.Cost;
disp(['Itera o :_' num2str(it), '_Melhor_custo_=' num2str(bestCosts(it))])

%amortecendo o coeficiente de inrcia
w = w * wDamp;

end

toc%finaliza a medi o de tempo do programa

%%Resultados
figure;
plot(bestCosts, 'linewidth', 3);
semilogy(bestCosts, 'linewidth', 3);
xlabel('Itera es ');
ylabel('Fun o de custo');
title('Fun o de custo com o passar das itera es ');
grid on;

```

.2 FILTRO RECURSIVO DE OUSTALOUPI

```

function G = oustaloupLuis(frac, N, w_B, w_H)
    pkg load control;
    s = tf('s');

    w_B = w_B;
    w_h = w_H;

    K = w_H^frac;

    num = 1;
    den = 1;

    for i = -N:N

        wk1 = w_B*( (w_H/w_B)^((i+N+0.5*(1-frac))/(2*N + 1)) );
        wk2 = w_B*( (w_H/w_B)^((i+N+0.5*(1+frac))/(2*N + 1)) );

        num = num*(s + wk1);
        den = den*(s + wk2);
    endfor

    G = ((K*num/den));
end

```

.3 FUNÇÃO DE CUSTO PARA O CONTROLADOR INTEIRO

```

function out=ITAE_Control(x,G_ma,Gf)
    pkg load control;
    s = tf('s');

    G_c = (x(3)*s*s + x(1)*s + x(2)) / s;
    G_MF = minreal((G_ma*G_c) / (1 + G_ma*G_c*Gf));

    t = 0:0.05:10;
    respDegrau = step(G_MF,t);
    degrau      = ones(length(t),1);
    erro        = abs(degrau - respDegrau);
    tt          = t';
    eat         = tt.*erro;
    ITAE        = trapz(tt,eat);
    out         = ITAE;
end

```

.4 FUNÇÃO DE CUSTO PARA O CONTROLADOR FRACIONÁRIO

```

function out=ITAEOustaloup_Control(x,G_ma,Gf,N,wb,wh)
    pkg load control;
    s = tf('s');

    ousta1 = oustaloupLuis(-x(4),N,wb,wh);
    ousta2 = oustaloupLuis(x(5),N,wb,wh);

    G_c = minreal(x(1) + x(2)*ousta1 + x(3)*ousta2);

    G_MF = minreal((G_ma*G_c) / (1 + G_ma*G_c*Gf));

    t = 0:0.05:10;
    respDegrau = step(G_MF,t);
    degrau      = ones(length(t),1);
    erro        = abs(degrau - respDegrau);
    tt          = t';
    eat         = tt.*erro;
    ITAE        = trapz(tt,eat);
    out         = ITAE;
end

```