



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
MECATRÔNICA



Projeto de Controlador PID para Sistemas de Segunda Ordem com Atraso via Resposta em Frequência

Jhonat Heberson Avelino de Souza

Orientador: Prof. Dr. Carlos Eduardo Trabuco Dórea

Co-orientador: Prof. Dr. José Mário Araújo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Mecatrônica da UFRN como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem PEM: M018
Natal, RN, maio de 2023

Resumo

Fenômenos como vibrações mecânicas, ressonância e oscilações, podem ser descritos matematicamente por sistemas de equações diferenciais de segunda ordem, sendo estes sistemas comumente designados como sistemas de segunda ordem. Trabalhar com esse tipo de modelo, em vez dos modelos de estado de primeira ordem, traz benefícios numéricos, mas há dificuldades inerentes à determinação de seus parâmetros físicos. Os desafios são ainda mais significativos quando se considera a existência de atrasos entre as medições dos estados e os sinais de atuação, levando algumas abordagens à necessidade de uma pós-análise para determinar a estabilidade das soluções calculadas. Uma alternativa para contornar as dificuldades de medição de parâmetros é a abordagem por resposta em frequência que usa modelos baseados em receptância. É reconhecido na literatura que modelos mais precisos de receptância podem ser obtidos experimentalmente quando comparados aos modelos por equações diferenciais. Além disso, os modelos de receptância permitem tratar o atraso de forma explícita, sem recorrer a aproximações. O objetivo deste trabalho é desenvolver um método baseado em receptância para o rastreamento robusto de referências de sistemas de segunda ordem com atraso representado por receptância, usando controladores do tipo PID - Proporcional-Integral-Derivativo, a ação integral deste controlador garante o rastreamento de referências constantes, desde que o sistema em malha fechada seja estável. A utilização do Algoritmo Genético, para determinar os ganhos do controlador PID, que satisfaça as condições de robustez, quanto índice de Integral do Erro Absoluto e a busca via otimizações descritas anteriormente devem, ser aplicadas a esse tipo de controlador e analisadas para o caso monovariável. A eficácia da técnica deve ser avaliada por meio de simulações numéricas utilizando modelos disponíveis na literatura da área.

Palavras-chave: Sistemas de Segunda Ordem, Sistemas com Atraso, Matriz de Receptância, Algoritmo Genético.

Abstract

Phenomena such as mechanical vibrations, resonance, and oscillations can be mathematically described by second-order differential equation systems, which are commonly referred to as second-order systems. Working with this type of model, instead of first-order state models, brings numerical benefits, but there are inherent difficulties in determining their physical parameters. The challenges are even more significant when considering the existence of delays between state measurements and actuation signals, leading some approaches to the need for post-analysis to determine the stability of calculated solutions. An alternative to bypass the difficulties of parameter measurement is the frequency response approach that uses models based on receptance. It is recognized in the literature that more accurate receptance models can be experimentally obtained when compared to models based on differential equations. In addition, receptance models allow for the explicit treatment of delays, without resorting to approximations. The objective of this work is to develop a receptance-based method for robust tracking of second-order systems with delay represented by receptance, using PID - Proportional-Integral-Derivative controllers. The integral action of this controller ensures the tracking of constant references, provided that the closed-loop system is stable. The use of Genetic Algorithm to determine PID controller gains that satisfy robustness conditions, such as the Absolute Error Integral index, and the optimization search described earlier should be applied to this type of controller and analyzed for the single-variable case. The effectiveness of the technique should be evaluated through numerical simulations using models available in the literature of the field.

Keywords: Second Order Systems, Delayed Systems, Receptance Matrix, Genetic Algorithm.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Símbolos e Abreviaturas	vii
1 Introdução	1
2 Teoria	5
2.1 Sistemas de Segunda Ordem	5
2.1.1 Representação de Primeira Ordem	6
2.1.2 O Método da Receptância	7
2.1.3 A Matriz de Receptância	8
2.1.4 Sistema com Atraso	9
2.2 Resposta em Frequência	10
2.2.1 O Diagrama Polar, ou Diagrama de Nyquist	10
2.2.2 Critério de Estabilidade de Nyquist	11
2.3 Controlador Proporcional Integrativo Derivativo	13
2.4 Índices de desempenho	14
2.5 Robustez de Estabilidade	15
3 Formulação do Problema	17
3.1 Parâmetro de Projeto e Formulação Matemática	18
3.2 Otimização de Desempenho	20
3.3 Abordagem para sistemas instáveis	20
4 Implementação do Método Proposto	23
4.1 Solução do Problema de Otimização	23
4.2 Problemas na implementação	26
4.3 A Função Robustez	27
4.4 A Função de Restrições	30
4.5 A Função de <i>IAE</i>	32
4.6 A Função de <i>Fitness</i>	32
4.7 A Função de Variabilidade	33

5	Experimentos e Resultados	35
5.1	Experimentos estudados	35
5.1.1	Exemplo 1	35
5.1.2	Exemplo 2	36
5.1.3	Exemplo 3	37
5.2	Resultados da Otimização de Robustez	37
5.2.1	Resultado exemplo 1	37
5.2.2	Resultado exemplo 2	37
5.2.3	Resultado exemplo 3	40
5.3	Resultados da Otimização de IAE	41
5.3.1	Resultado exemplo 1	41
5.3.2	Resultado exemplo 2	42
5.3.3	Resultado exemplo 3	43
5.4	Resultados da Comparação entre as Otmização	44
5.4.1	Resultado exemplo 1	45
5.4.2	Resultado exemplo 2	45
5.4.3	Resultado exemplo 3	45
5.5	Resultados da Concorrência entre a Otimização	47
5.5.1	Resultado exemplo 1	48
5.5.2	Resultado exemplo 2	50
5.5.3	Resultado exemplo 3	51
5.6	Resultados da perturbação no sistema	51
5.6.1	Perturbação para Otimização de <i>IAE</i>	51
5.6.2	Perturbação para Concorrência entre a Otimização	52
6	Conclusão	59
	Referências bibliográficas	60

Lista de Figuras

2.1	Sistema Massa-Mola-Amortecedor	5
2.2	Exemplo de Diagrama de Nyquist para Sistema modelado por $G(s)$	11
2.3	Diagrama de Nyquist com a Parte Espelhada	12
2.4	Diagrama de blocos do PID	14
3.1	Diagrama de Nyquist Exemplo de Circunferência. M_s	19
3.2	Diagrama de Nyquist de um Sistema Genérico para Exemplo de um Caso de Instabilidade.	20
4.1	<i>Nyquist</i> com visão do sistema estável	28
4.2	<i>Nyquist</i> com visão do sistema instável	28
4.3	Simulação criada com MATLAB/Simulink®	32
5.1	Sistema massa-mola-amortecedor de um grau de liberdade	36
5.2	Diagrama de Nyquist Exemplo 5.1.1	38
5.3	Simulação do Sistema do Exemplo 5.1.1	38
5.4	Diagrama de Nyquist Exemplo 5.1.2	39
5.5	Simulação do Sistema do Exemplo 5.1.2	39
5.6	Diagrama de Nyquist Exemplo 5.1.3	40
5.7	Simulação do Sistema do Exemplo 5.1.3	40
5.8	Diagrama de Nyquist Exemplo 5.1.1	41
5.9	Simulação do Sistema do Exemplo 5.1.1	42
5.10	Diagrama de Nyquist Exemplo 5.1.2	42
5.11	Simulação do Sistema do Exemplo 5.1.2	43
5.12	Diagrama de Nyquist Exemplo 5.1.3	43
5.13	Simulação do Sistema do Exemplo 5.1.3	44
5.14	Comparação do diagrama de <i>Nyquist</i> do Exemplo 5.1.1	45
5.15	Comparação da simulação do Sistema do Exemplo 5.1.1	46
5.16	Comparação do diagrama de <i>Nyquist</i> do Exemplo 5.1.2	46
5.17	Comparação da simulação do Sistema do Exemplo 5.1.2	47
5.18	Comparação do diagrama de <i>Nyquist</i> do Exemplo 5.1.3	47
5.19	Comparação da simulação do Sistema do Exemplo 5.1.3	48
5.20	Diagrama de <i>Nyquist</i> com Método proposto do Exemplo 5.1.1	49
5.21	Simulação do Sistema com Método proposto do Exemplo 5.1.1	49
5.22	Diagrama de <i>Nyquist</i> com Método proposto do Exemplo 5.1.2	50
5.23	Simulação do Sistema com Método proposto do Exemplo 5.1.2	50
5.24	Diagrama de <i>Nyquist</i> com Método proposto do Exemplo 5.1.3	51

5.25	Simulação do Sistema com Método proposto do Exemplo 5.1.3	52
5.26	Diagrama de <i>Nyquist</i> com perturbação no sistema	53
5.27	Simulação do Sistema com perturbação no sistema	54
5.28	polos do Sistema com perturbação no sistema	55
5.29	Diagrama de <i>Nyquist</i> com perturbação no sistema	56
5.30	Simulação do Sistema com perturbação no sistema	57
5.31	polos do Sistema com perturbação no sistema	58

Lista de Tabelas

5.1	Tabela de Ganhos do PID - Otimização de Robustez	37
5.2	Tabela de Ganhos do PID - Otimização de <i>IAE</i>	41
5.3	Tabela de Variança dos Ganhos do PID - Comparação da Otimização de <i>IAE</i> com robustez	44
5.4	Tabela de Ganhos do PID com Método proposto - Comparação da Otimização de <i>IAE</i> com robustez	48

Lista de Símbolos e Abreviaturas

SISO	Single Input Single Output, veja equação (2.4), página 6
SPD	Semiplano Direito, veja equação (2.27), página 12
SPE	Semiplano Esquerdo, veja equação (2.26), página 12

Capítulo 1

Introdução

Uma diversidade de fenômenos físicos, a exemplo da ressonância em sistemas vibratórios, vibroacústicos e elétricos, é ricamente representada por meio de modelos dinâmicos de segunda ordem BALAS (1982); VANDERVELDE (1986). Os sistemas de segunda ordem são amplamente encontrados em diversas áreas, desde a engenharia até a física e a biologia, tornando seu estudo fundamental para a compreensão de uma ampla gama de fenômenos naturais e artificiais.

Modelos matemáticos que representam esses sistemas podem ser escritos como um sistema matricial de equações diferenciais de segunda ordem, onde as constantes e variáveis das equações têm relações físicas diretas com o modelo real, ou transformadas em equações diferenciais de primeira ordem. Em DATTA (2004), são apresentadas representações de primeira ordem para alguns sistemas de segunda ordem e discutidas algumas deficiências associadas ao uso do modelo original.

Devido à sua relevância prática, o estudo de sistemas de segunda ordem tem ganhado cada vez mais notoriedade na comunidade de engenharia, proporcionando soluções para diversos desafios. Uma das dificuldades na utilização de modelos matemáticos obtidos a partir de equações diferenciais de segunda ordem está na definição dos elementos das matrizes que caracterizam o sistema, como massa e elasticidade, visando simplificar o processo de modelagem desses sistemas, existe uma abordagem que utiliza a ideia de receptância, que foi proposta por RAM et al. (2009) se baseia no uso de informações experimentais do sistema para definir essas matrizes.

Nos últimos anos, alguns trabalhos utilizaram essa ideia, como base para modelagem de sistema de controle de segunda ordem com atraso, entre eles ARAÚJO e SANTOS (2018); ARAÚJO (2018*a,b*); SANTOS et al. (2018); DANTAS (2019); DANTAS et al. (2020); MARTINS et al. (2020); NUNES (2022); GONTIJO et al. (2022).

A utilização do método da receptância, é possível alocar os polos do sistema em uma posição segura no plano complexo e alcançar um desempenho desejado. Entretanto, quando há a presença de atraso de transporte no sistema, ao utilizar esse método, faz-se necessário realizar um tratamento posteriormente para garantir que o sistema continue estável em malha fechada, como visto em RAM et al. (2009) e em ARAÚJO e SANTOS (2018).

Uma abordagem foi proposta por DANTAS et al. (2020) para eliminar a análise posterior do método de receptância na presença de atraso. Eles propuseram uma abordagem

no domínio da frequência, utilizando conceitos como estabilidade de *Nyquist*, margem de ganho e função de sensibilidade. Por meio dessa abordagem, foi possível projetar um controlador robusto para sistema de segunda ordem com atraso.

Também em NUNES (2022) foi desenvolvido um método para projetar um controle utilizando realimentação derivativa de estados, que modifica as matrizes massa e amortecimento. A realimentação derivativa oferece vantagens em relação à realimentação de estados, como o uso de acelerômetros como sensores, que são mais baratos e mais aplicáveis na indústria, maior precisão na leitura dos estados do sistema ABDELAZIZ e VALÁŠEK (2004) e a possibilidade de regularização e estabilização de sistemas com matriz massa singular. Resultados deste método foram apresentados em um trabalho de Nunes et al. (2021).

No trabalho GONTIJO et al. (2022), é apresentado um método robusto para a concepção de controladores PID (Proporcional-Integral-Derivativo) para sistemas vibratórios de segunda ordem com incertezas paramétricas. Diferentemente de trabalhos correlatos baseados exclusivamente em retroalimentação proporcional e derivativa, o método proposto emprega a ação integral para atingir um erro nulo de rastreamento do ponto de ajuste na presença de distúrbios constantes relacionados aos graus de liberdade atuados, aumentando assim a flexibilidade do projeto.

Neste trabalho, é apresentado um método para o projeto de controle usando um controlador PID seguidor de referência, baseado no método proposto por DANTAS et al. (2020). A metodologia de projeto usa o critério de estabilidade de *Nyquist* como base teórica para determinar os valores dos ganhos do controlador que estabilizam a planta e atendem aos requisitos do projeto de controle.

Neste estudo, diferentemente de outros trabalhos nessa linha, um dos objetivos é alcançar a otimização do índice de Erro Absoluto Integral (*IAE*) em conjunto com a otimização da robustez do sistema. Como também, o controlador PID é introduzido na formulação do problema para sistemas de segunda ordem com atraso, o qual esse controlador é amplamente utilizado até os dias atuais, de forma que aplicação deste estudo seja mais utilizado em situações reais de controle.

O objetivo deste trabalho é apresentar uma extensão e comparação com o método proposto em DANTAS et al. (2020). Diferentemente do estudo de SANTOS et al. (2018), que utilizou técnicas de alocação de polos, a natureza do controlador PID neste trabalho não exige tais técnicas. Bem como a extensão do uso do índice de *IAE* como parâmetro de busca para Algoritmo Genético (*GA*). Isso é importante para aumentar a eficiência do sistema e garantir sua robustez, mesmo com a concorrência entre os parâmetros da otimização.

O trabalho está disposto da seguinte maneira:

- Capítulo 2: Apresenta a teoria matemática básica para a compreensão do problema abordado e desenvolvimento do trabalho.
- Capítulo 3: Apresenta o problema com equacionamento matemático e detalhes importantes sobre as vantagens do método a ser utilizado.
- Capítulo 4: Apresenta o método desenvolvido para solução do problema apresentado no Capítulo 3.
- Capítulo 5: Apresenta os resultados obtidos com a utilização do método proposto,

através de exemplos numéricos emprestados de outros trabalhos com objetivo de obter parâmetros de comparação dos resultados alcançados.

- Capítulo 6: Traz comentários conclusivos e sugestões para futuros trabalhos relacionados.

Capítulo 2

Teoria

2.1 Sistemas de Segunda Ordem

Diferentes tipos de sistemas, como os sistemas vibratórios mecânicos, são classificados como sistemas de segunda ordem. Esses sistemas são descritos por essas equações, que retratam o comportamento dinâmico do sistema. Nos trabalhos de ARAÚJO (2018b); SANTOS et al. (2018) são discutidos tais modelos com destaque para sua importância. Esses modelos são usados para sistemas que possuem n equações diferenciais, uma para cada grau de liberdade do sistema DANTAS et al. (2020).

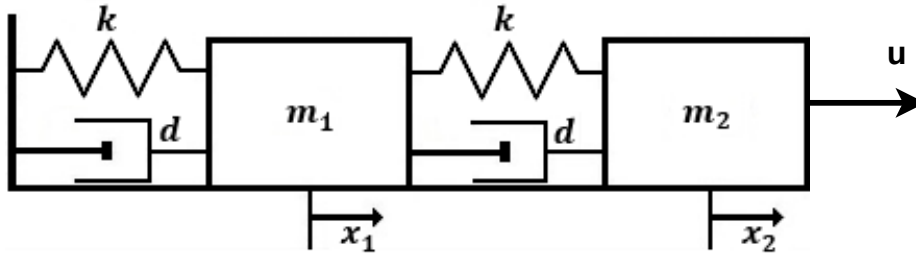


Figura 2.1: Sistema Massa-Mola-Amortecedor

Fonte: Autor

O sistema mostrado na Figura 2.1 pode ser modelado usando equações diferenciais, conforme mostrado na Equação 2.1 e 2.2, segundo as leis clássicas da mecânica de Newton.

$$m_1 \ddot{x}_1(t) + d(2\dot{x}_1(t) - \dot{x}_2(t)) + k(2x_1(t) - x_2(t)) = 0, \quad (2.1)$$

$$m_2 \ddot{x}_2(t) + d(\dot{x}_2(t) - \dot{x}_1(t)) + k(x_2(t) - x_1(t)) = u, \quad (2.2)$$

Sendo u uma força externa manipulável (entrada); m_1 e m_2 as massas dos blocos; k o coeficiente de rigidez das molas; d o coeficiente de amortecimento dos *dashpots*; x_1 e x_2 os deslocamentos dos blocos; e \dot{x}_1 , \dot{x}_2 , \ddot{x}_1 , \ddot{x}_2 respectivamente as velocidades e acelerações dos blocos.

Em forma matricial as equações 2.1 e 2.2 podem ser reescritas da seguinte maneira:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} 2d & -d \\ -d & d \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (2.3)$$

Enquanto de forma simplificada:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{B}u \quad (2.4)$$

sendo:

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 2d & -d \\ -d & d \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix} \text{ e } \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Para um sistema com n graus de liberdade e m entradas, têm-se $\mathbf{M} \in \mathfrak{R}^{n \times n}$, é uma matriz de massas, $\mathbf{D} \in \mathfrak{R}^{n \times n}$ é uma matriz de amortecimento, $\mathbf{K} \in \mathfrak{R}^{n \times n}$ é uma matriz de rigidez, $\mathbf{B} \in \mathfrak{R}^{n \times m}$ é uma matriz de controle, $\mathbf{x} \in \mathfrak{R}^n$ é o vetor de deslocamento e $\mathbf{u} \in \mathfrak{R}^m$ é um vetor de entradas.

2.1.1 Representação de Primeira Ordem

A Equação 2.4 representa o sistema de segunda ordem em sua forma original. Outra maneira de interpretar o sistema descrito em 2.4 é usar representações de primeira ordem, como espaço de estado $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$.

Definindo $\mathbf{x} = \tilde{\mathbf{x}}_1$ e $\dot{\mathbf{x}} = \tilde{\mathbf{x}}_2$, podemos escrever o sistema de segunda ordem da equação 2.4 como:

$$\dot{\tilde{\mathbf{x}}}_1 = \tilde{\mathbf{x}}_2 \quad (2.5)$$

$$\dot{\tilde{\mathbf{x}}}_2 = -\mathbf{M}^{-1}\mathbf{K}\tilde{\mathbf{x}}_1 - \mathbf{M}^{-1}\mathbf{C}\tilde{\mathbf{x}}_2 + \mathbf{M}^{-1}\mathbf{B}u \quad (2.6)$$

na forma de matrizes:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \dot{\tilde{\mathbf{x}}}_1 \\ \dot{\tilde{\mathbf{x}}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_1 \\ \tilde{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{B} \end{bmatrix} u. \quad (2.7)$$

ou de maneira abreviada:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}u$$

com:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \text{ e } \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{B} \end{bmatrix} \quad (2.8)$$

Essa representação é chamada de representação de primeira ordem padrão, enquanto outras formas de representações do sistema de segunda ordem são generalizadas em sistema descritor, que é um tipo de sistema dinâmico que pode ser representado por meio de uma equação diferencial matricial que inclui uma matriz de ganho. Ele é chamado de "descritor" porque inclui tanto as variáveis de estado quanto as de saída em uma única matriz, ao contrário dos sistemas padrão em que as variáveis de estado e saída são separadas DATTA (2004).

A motivação para a obtenção de um modelo de primeira ordem está na equação, com uma solução mais simples do que a equação de segunda ordem. No entanto, algumas dificuldades numéricas surgem ao lidar com o sistema expresso dessa forma.

Uma dificuldade numérica é a possibilidade de surgimento de raízes complexas no polinômio característico do sistema de primeira ordem, o que pode dificultar a análise da estabilidade do sistema. Por fim, a transformação de segunda ordem para primeira ordem pode levar a uma perda de precisão numérica devido a erros de arredondamento e truncamento, o que pode afetar a validade do modelo.

Outro fato importante é as propriedades disponíveis das matrizes de coeficientes \mathbf{M} , \mathbf{C} e \mathbf{K} ; como simetria, a qual em problemas de grande escala são informações importantes para representação em segunda ordem na Equação 2.7.

2.1.2 O Método da Receptância

O Método da Receptância é uma técnica utilizada na engenharia estrutural para analisar e controlar sistemas dinâmicos complexos. Ele permite a representação de um sistema de segunda ordem como um sistema de primeira ordem, o que facilita o processo de modelagem e controle desses sistemas.

O método é baseado na ideia de que um sistema de segunda ordem pode ser representado por um sistema de primeira ordem com múltiplos graus de liberdade. Cada grau de liberdade representa a resposta do sistema a uma determinada força ou excitação externa. A receptância é uma medida da relação entre a resposta do sistema e a excitação externa em cada grau de liberdade.

Para aplicar o método da receptância, é necessário medir a resposta do sistema em cada grau de liberdade para uma série de excitações externas. A partir desses dados, é possível calcular a receptância em cada grau de liberdade. Em seguida, essas receptâncias podem ser utilizadas para representar o sistema como um sistema de primeira ordem.

Uma das principais vantagens do método da receptância é que ele permite a representação de sistemas complexos com múltiplos graus de liberdade de uma forma simplificada. Isso torna mais fácil a análise e o controle desses sistemas, além de permitir a identificação de problemas que possam estar afetando o desempenho do sistema.

Por ser um método baseado unicamente em dados medidos de vibrações, não há necessidade de se determinar, ou conhecer, estas matrizes TEHRANI et al. (2010).

A Determinação da Matriz Rigidez (\mathbf{K}) e de massas (\mathbf{M}) são menos desafiadoras porque de acordo com MOTTERSHEAD e RAM (2007) os elementos finitos dessas matrizes podem ser facilmente determinados, usando métodos variacionais. No entanto, essa praticidade funciona com matriz de amortecimento (\mathbf{C}). Em algumas bibliografias o espaço de estados e os elementos da matriz de amortecimento são definidos como padrões físicos confiáveis. As vibrações levemente amortecidas podem ser aceitáveis como medida de estudos de controle ativo do modelo de amortecimento NUNES (2022).

A regra de *Rayleigh* é um método amplamente utilizado para estimar a matriz de amortecimento em um sistema dinâmico. A matriz de amortecimento é fundamental para descrever a dissipação de energia no sistema e a correta definição é importante para garantir a precisão da análise dinâmica. De acordo com a regra de *Rayleigh*, a matriz de

amortecimento pode ser aproximada como uma combinação linear das matrizes de massa e rigidez do sistema conforme apresentado na Equação 2.9.

$$C = \alpha M + \beta K \quad (2.9)$$

Onde C é a matriz de amortecimento, M é a matriz de massa, K é a matriz de rigidez, e α e β são constantes determinadas a partir das frequências naturais do sistema.

Essas constantes podem ser calculadas a partir da equação 2.10 que envolve as frequências naturais do sistema e o fator de amortecimento.

$$\frac{(\beta\omega_1^2 - \alpha\omega_2^2)}{(\omega_1^2\omega_2^2)} = \xi \quad (2.10)$$

Onde ω_1 e ω_2 são as frequências naturais do sistema, e ξ é o fator de amortecimento do sistema.

É importante notar que a regra de *Rayleigh* é uma aproximação e sua precisão pode variar dependendo das características específicas do sistema em análise. No entanto, essa técnica é comumente utilizada na prática quando não há informações precisas sobre a matriz de amortecimento ou quando é difícil medir as propriedades de amortecimento diretamente.

2.1.3 A Matriz de Receptância

Considerando o controle de um sistema de segunda ordem por realimentação de estados definido por

A saída do sistema definida como:

$$y(t) = e(t)lx(t) \quad (2.11)$$

Considere-se o monovariável, ou seja, $\mathbf{b}, \mathbf{l} \in \mathbb{R}^{1 \times n}$.

e erro do sistema é definido como:

$$e(t) = r(t) - y(t) \quad (2.12)$$

Assume-se a lei de controle por realimentação de estados como:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{b}u(t) \quad (2.13)$$

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \quad (2.14)$$

em que $\mathbf{M}, \mathbf{C}, \mathbf{K} \in \mathbb{R}^{n \times n}; k_p, k_i, k_d \in \mathbb{R}^n; \mathbf{x} \in \mathbb{R}^n, \mathbf{M} = \mathbf{M}^T, \mathbf{C} = \mathbf{C}^T, \mathbf{K} = \mathbf{K}^T; \mathbf{v}^T \mathbf{M} \mathbf{v} > 0, \mathbf{v}^T \mathbf{C} \mathbf{v} \geq 0$ e $\mathbf{v}^T \mathbf{K} \mathbf{v} \geq 0$ para qualquer $\mathbf{v} \neq \mathbf{0}, \mathbf{v} \in \mathbb{R}^n$.

Substituindo 2.14 em 2.13, assumindo $\mathbf{x}(t) = \mathbf{z}e^{st}$, sendo \mathbf{z} um vetor constante e s a

variável de *Laplace*, solução do sistema, tem-se:

$$\begin{aligned} (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} - B(k_p + \frac{k_i}{s} + k_d s)l) &= \mathbf{0} \\ (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} - Bq(s)l) &= \mathbf{0} \end{aligned} \quad (2.15)$$

é possível perceber que $q(s) = (k_p + \frac{k_i}{s} + k_d s)$ perceber que as componentes de malha fechada de rigidez e amortecimento são modificadas pela matriz de posto-1, consequência da realimentação de estados.

A Fórmula de *Sherman-Morrison*

A fórmula de *Sherman-Morrison* fornece a inversa de uma matriz com modificação de posto-1 em termos da inversa da matriz original, e como visto em VAN LOAN e GOLUB (1983) e GOLUB e VAN LOAN (2012) ela estabelece que, para uma dada matriz $\mathbf{A} \in \mathfrak{R}^{n \times n}$ quadrada e inversível e vetores colunas \mathbf{u} e $\mathbf{v} \in \mathfrak{R}^n$:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \quad (2.16)$$

Aplicando a fórmula de *Sherman-Morrison* em 2.15 com $\mathbf{A} = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K})$, $\mathbf{u} = \mathbf{B}$ e $\mathbf{v} = (k_p + \frac{k_i}{s} + k_d s)$, temos:

$$\hat{\mathbf{H}}(s) = \mathbf{H}(s) - \frac{\mathbf{H}(s)(k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)\mathbf{B}}{1 + (k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)\mathbf{B}} \quad (2.17)$$

na qual $\hat{\mathbf{H}}(s) = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + bl(k_p + \frac{k_i}{s} + k_d s))^{-1}$ é definida em RAM et al. (2009) como matriz de receptância de malha fechada e $\mathbf{H}(s) = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K})^{-1}$ como matriz de receptância de malha aberta, que, na prática, pode ser medida pela resposta em frequência $\mathbf{H}(j\omega)$.

A equação característica de 2.17 é definida como:

$$1 + (k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)\mathbf{B} = 0 \quad (2.18)$$

2.1.4 Sistema com Atraso

Quando existe um atraso no tempo entre o estado medido e a ação de controle do atuador podemos representar a lei de ação de controle da seguinte maneira:

$$u(t - \tau) = k_p e(t - \tau) + k_i \int_0^t e(t - \tau) d\tau + k_d \frac{de(t - \tau)}{dt} \quad (2.19)$$

sendo k_p, k_i e $k_d \in \mathfrak{R}^n$ e são vetores de ganho do sistema, desta forma, um sistema de controle de segunda ordem, como o da Figura 2.1, descrito pelo sistema de equações 2.4, com matrizes de massa $\mathbf{M} \in \mathfrak{R}^{n \times n}$, amortecimento $\mathbf{C} \in \mathfrak{R}^n$ e elasticidade $\mathbf{K} \in \mathfrak{R}^n$ pode

ser escrito como:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{b}u(t - \tau) \quad (2.20)$$

Aplicando-se os procedimentos da seção anterior, para obtenção da matriz de receptância do sistema de segunda ordem com atraso descrito pela equação 2.20, com $\mathbf{x}(t) = \mathbf{z}e^{st}$, sendo \mathbf{z} um vetor constante e s a variável de *Laplace*, solução do sistema, tem-se:

$$\begin{aligned} (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} - \mathbf{B}l(k_p + \frac{k_i}{s} + k_d s)e^{-\tau s}) &= \mathbf{0} \\ (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + e^{-\tau s}q(s)\mathbf{B}l)^{-1} &= \mathbf{0} \end{aligned} \quad (2.21)$$

e aplicando-se a fórmula de *Sherman-Morrison* como em 2.15

$$\hat{\mathbf{H}}(s) = \mathbf{H}(s) - \frac{\mathbf{H}(s)(k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)\mathbf{B}e^{-\tau s}}{1 + (k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)\mathbf{B}e^{-\tau s}} \quad (2.22)$$

sendo novamente, $\hat{\mathbf{H}}(s) = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + bd(k_p + \frac{k_i}{s} + k_d s)e^{-\tau s})^{-1}$ é definida como matriz de receptância de malha fechada e $\mathbf{H}(s) = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K})^{-1}$ como matriz de receptância de malha aberta. A presença do atraso na equação 2.22 não modifica a possibilidade da obtenção de $\mathbf{H}(s)$ por sua resposta em frequência, uma das motivações para o desenvolvimento deste trabalho.

A equação característica da matriz de receptância de 2.22 é dada por:

$$1 - (k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)\mathbf{B}e^{-\tau s} = 0 \quad (2.23)$$

2.2 Resposta em Frequência

Na literatura clássica de controle, como em OGATA (2009), o termo resposta A frequência é definida como a resposta de estado estacionário do sistema à entrada senoidal, o que significa que experimentalmente, o sistema é excitado com uma onda senoidal entrada e observe o comportamento da saída. Trabalho realizado em resposta a frequência é muitas vezes motivada pelo bom desempenho em lidar com a incerteza nos modelos das plantas para facilitar o uso de informações experimentais, desenhar metas e ações na presença de atrasos no transporte FRANKLIN et al. (2013).

2.2.1 O Diagrama Polar, ou Diagrama de Nyquist

A representação gráfica escolhida para o desenvolvimento deste trabalho é Nyquist, também conhecido como diagrama polar. Gráfico polar de uma função de transferência senoidal $G(j\omega)$ é a magnitude de $G(j\omega)$ e o ângulo de fase de $G(j\omega)$ em coordenadas polares, onde ω varia de zero a infinito de acordo com literatura base do (Ogata 2009). Um exemplo de gráfico de Nyquist para um sistema típico de segunda ordem é mostrado na equação 2.24 é mostrada na Figura 2.2.

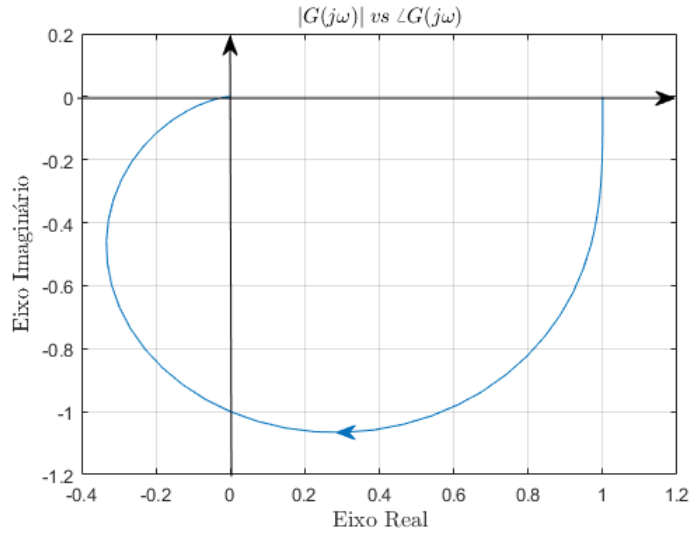


Figura 2.2: Exemplo de Diagrama de Nyquist para Sistema modelado por $G(s)$

Fonte: DANTAS (2019)

$$G(s) = \frac{1}{s^2 + s + 1} \quad (2.24)$$

O Diagrama de Nyquist, também conhecido como Diagrama Polar, é um gráfico em que o eixo radial representa a magnitude do ganho e o eixo angular representa o deslocamento de fase de um sistema de controle. Quando a curva do diagrama cruza o eixo real, há um ponto crítico de estabilidade que indica que a resposta do sistema está instável, além disso, as curvas de *Nyquist* são caracterizadas pela simetria em relação ao eixo real conforme apresentado na Figura 2.3.

O Diagrama de Nyquist é uma técnica muito útil para a análise de sistemas de controle, já que é possível avaliar a estabilidade do sistema sem precisar de uma simulação completa. É amplamente utilizado em áreas como engenharia de controle, eletrônica e telecomunicações para projetar e ajustar sistemas de controle e garantir sua estabilidade.

Métodos para obtenção das curvas são mostrados no livro de controle clássico como OGATA (2009), FRANKLIN et al. (2013) e SKOGESTAD e POSTLETHWAITE (2007). Neste trabalho, as trajetórias de Nyquist são obtidas com o auxílio do MATLAB R2020a.

2.2.2 Critério de Estabilidade de Nyquist

O critério de estabilidade de Nyquist determina a estabilidade do sistema, baseado na resposta em frequência de um sistema em malha aberta. Em um sistema no circuito fechado descrito pela seguinte função de transferência:

$$G_{mf}(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)} \quad (2.25)$$

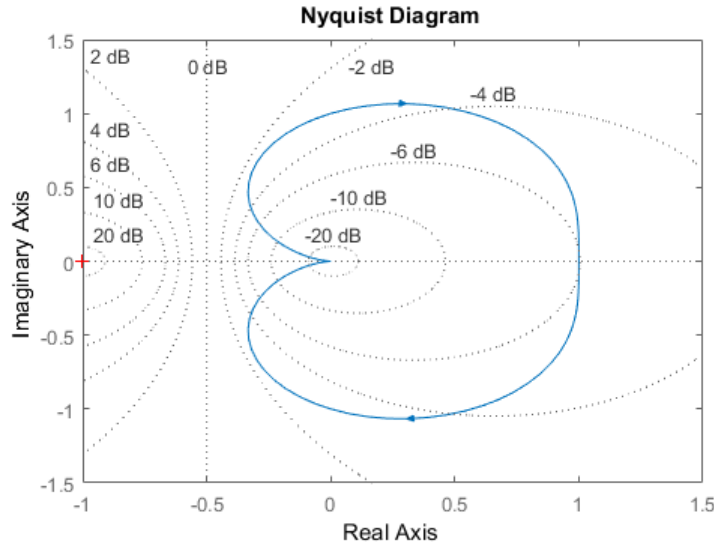


Figura 2.3: Diagrama de Nyquist com a Parte Espelhada

Fonte: DANTAS (2019)

a equação característica é dada por

$$1 + G(s) = 0 \quad (2.26)$$

e para estabilidade, todas as raízes desta equação devem ter partes reais negativas, ou seja, O semiplano esquerdo (SPE) no plano s . Esse critério de estabilidade da equação de Nyquist relaciona o número dessas raízes no semiplano direito (SPD) para com o número de voltas no ponto $(-1, 0)$ descrito na equação 2.27

$$Z = N + P \quad (2.27)$$

na qual Z representa o número de polos de malha fechada situados no SPD, N o número de envoltimentos do ponto $(-1, 0)$, no sentido horário, pelo traçado de Nyquist do ganho de malha de $G(j\omega)$ e P o número de polos de malha aberta no semi-plano direito de $G(s)$.

O número N pode ser negativo como também positivo, dependendo da orientação da volta, para cada contribuição em volta no ponto no sentido horário em N é positiva, e para cada volta no sentido anti-horário conta como uma contribuição negativa. Esse argumento é baseado no teorema do mapeamento complexo ou Teorema Cauchy, que cria uma relação a curvas em domínios funcionais com o domínio s . Segundo o OGATA (2009) a análise de estabilidade do sistema pode ser resumido para três casos possíveis:

1. Não existir nenhum envolvimento do ponto $-1 + j0$. Implicando onde o sistema será estável se não houver polos de malha aberta no SPD do plano s , e instável caso contrário;
2. Existir um ou mais envoltimentos do ponto $-1 + j0$ no sentido anti-horário. Implicando que, para o sistema estável, nesse caso, o número de envoltimentos tem que ser igual ao número de polos de malha aberta no SPD;

3. Existir um ou mais envoltimentos do ponto $-1 + j0$ no sentido horário, implicando nesse caso em instabilidade.

Baseado no teorema do mapeamento, FRANKLIN et al. (2013) resume em quatro passos básicos o processo de determinação de estabilidade de um sistema pelo critério de estabilidade de Nyquist:

1. Obter o diagrama de Nyquist do ganho de malha do sistema, geralmente representado por 2.25;
2. Avaliar o número de envoltimentos do ponto $-1 + j0$ no sentido horário (N), se o envolvimento for no sentido anti-horário o número é negativo;
3. Determinar o número de polos instáveis em malha aberta (P);
4. Calcular o número de raízes instáveis em malha fechada: $Z = N + P$

Vantagens de usar a análise de estabilidade pelo critério de estabilidade de Nyquist é que o número de polos (Z) não precisa necessariamente ser encontrado, então não há necessidade de expressar matematicamente os atrasos por meio de funções de transferência de ordem finita, eliminando o uso de aproximações, assim melhorando sua fidelidade nos resultados.

2.3 Controlador Proporcional Integrativo Derivativo

O Controlador Proporcional Integrativo Derivativo (PID) une as ações proporcional, integral e derivativa num só controlador, atuando tanto no regime transitório quanto no regime permanente. São utilizados em sistema de controle industriais, onde ele introduz um polo em $p = 0$ e dois zeros ao sistema, que dependem dos parâmetros de controlador, e tendemos ao zeros do sistema serem reais e iguais para melhorar a estabilidade GONTIJO et al. (2022).

Existem muitas maneiras de representar um PID por função de transferência, uma delas é esta:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = \frac{s^2 K_d + s K_p + K_i}{s} = K_c \frac{(s + z_1)(s + z_2)}{s} \quad (2.28)$$

Na Figura 2.4, mostra o Controlador PID, como representação de diagramas de blocos, percebe também que essa representação já está com melhorias como filtro derivativo, como objetivo de tratar variações bruscas do sinal de referência.

Na literatura de controle como OGATA (2009), o método Lugar Geométrico das Raízes (LGR), *Zigler-Nichols* são algumas técnicas de projetos de controladores PID.

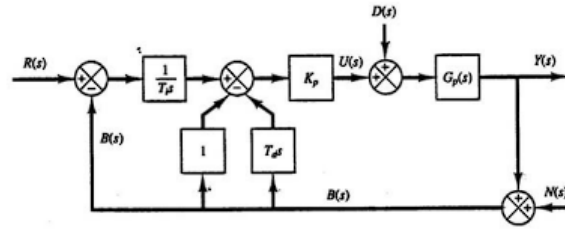


Figura 2.4: Diagrama de blocos do PID

Fonte: Autor

Os passos comum para projeto de controladores PID é:

1. Traduzir as especificações de desempenho como uma localização desejada de polos dominantes de malha fechada
2. Verificar se o objetivo não pode ser atingido com um controlador mais simples
3. Se o PID é necessário, localizar o polo na origem e os zeros de modo que a condição de ângulo seja satisfeita
4. Calcular o ganho total requerido, aplicando a condição de módulo
5. Simular o sistema com o controlador e observar o comportamento da resposta. Caso não seja satisfatório, tentar um ajuste fino dos parâmetros do controlador (K_c , z_1 e z_2)

Neste trabalho não utilizaremos soluções lineares, sim, soluções heurísticas, como o uso de Algoritmo Genético e Enxames de Partículas para encontrar os valores dos ganhos que satisfaçam as nossas restrições de estabilidade *Nyquist*.

2.4 Índices de desempenho

Os índices de desempenhos são medidas quantitativa do desempenho de um sistema, esses índices consideram fatores, como erro do sistema, e o tempo. E para cada sistema é preciso identificar qual o maior fator relevante para o sistema, a qual queremos atingir o ótimo. Dizemos que o sistema de é **ótimo** quando o índice de desempenho selecionado é minimizado, em nosso caso o algoritmo genético irá minimizá-lo.

Alguns índices de desempenhos são:

- Integral do erro ao quadrado (*ISE*)

$$ISE = \int_0^T e^2(t) dt \quad (2.29)$$

Esse índice tem empenho entre resposta subamortecidas e super-amortecida, consiste na medida do valor é dado pela integral do quadrado do erro atuante para uma malha de controle. o *ISE* tem uma maior sensibilidade aos erros grandes, porque, o

desvio em relação ao sinal de referência é grande, contribui com uma maior parcela no resultado da integral, do que erro quadrático cuja magnitude seja menor. Porém, esse índice tendem que erros pequenos persistam na resposta do sistema por um tempo maior, por exemplo, oscilações longas e de pequenas amplitudes no sinal.

- Integral do erro absoluto (*IAE*)

$$IAE = \int_0^T |e(t)| dt \quad (2.30)$$

O índice *IAE* gera respostas mais lentas que *ISE* segundo, OGATA (2009). A formulação é definida como integral do módulo do erro atuante, logo pode se notar que é atuação contrária ao *ISE*, o *IAE* não, adiciona qualquer tipo de peso ao erro citado, ao minimizar o sistema considerando *IAE* podemos afirmar que a resposta será mais lenta, contudo espera-se que o sistema de controle apresente menos oscilações na saída.

- Integral do erro absoluto vezes o tempo (*ITAE*)

$$ITAE = \int_0^T t |e(t)| dt \quad (2.31)$$

O efeito desse índice tem como principal fator é reduzir a contribuição do erro nos primeiros segundos e olhando para erro presente na resposta do regime do transi-tório. O *ITAE* é um índice mais seletivo no qual o seu valor mínimo é facilmente identificável em função do amortecimento ζ do sistema. Geralmente os sistemas que utiliza a sintonia do controlador pela minimização do *ITAE* tendem a atingir o regime permanente de forma mais rápida.

- Integral do erro ao quadrado vezes o tempo (*ITSE*)

$$ITSE = \int_0^T t e^2 dt \quad (2.32)$$

O *ITSE*, como também *ITAE* tem foca nas penalizações de oscilações persistentes, com objetivo de reduzir o tempo de acomodação.

2.5 Robustez de Estabilidade

A capacidade de um sistema de segunda ordem em manter a estabilidade e o desempenho diante de incertezas e perturbações externas é conhecida como robustez. Isso significa que um sistema robusto consegue lidar com variações em suas condições de operação e manter um comportamento estável e desejado.

Vários fatores podem influenciar a robustez de um sistema de segunda ordem, incluindo a localização dos polos do sistema no plano complexo, a razão de amortecimento e a frequência natural. Quanto mais distantes os polos do sistema estiverem do eixo imaginário, maior será a robustez, pois o sistema será menos suscetível a perturbações na

fase.

Outro fator que pode afetar a robustez de um sistema de segunda ordem é a razão de amortecimento. Sistemas com alta razão de amortecimento tendem a ser mais robustos, uma vez que o amortecimento limita o impacto das perturbações no sistema. Além disso, a frequência natural do sistema também influencia na robustez, já que sistemas com frequências naturais mais baixas tendem a ser mais robustos do que aqueles com frequências naturais mais altas.

Em síntese, é crucial garantir a robustez de um sistema de segunda ordem para que ele possa manter a estabilidade e o desempenho adequados em condições variáveis e adversas.

Considerando, sistemas que não possuem autovalores no semiplano direito (SPD), incluindo autovalores no eixo imaginário, um sistema é considerado estável em malha fechada se a curva do seu diagrama de Nyquist não cruzar (não enlaçar) o ponto $(-1 + 0j)$, segundo o critério de *Nyquist*.

Para implementar um controle robusto e com melhor desempenho, Rego et al. (2017) propõem o uso de um círculo de raio M_s^{-1} centrado em $(-1 + 0j)$ e visa fazer com que a curva do diagrama de *Nyquist* (em azul na Figura 3.1) tangencie esse círculo (em vermelho na Figura 3.1). Isso garante valores mínimos adequados de margem de ganho e margem de fase, além de uma sensibilidade apropriada do sistema, já que são utilizados valores de M_s entre 1,3 e 2, como indicado por Astrom (1995).

Capítulo 3

Formulação do Problema

O sistema matricial de equações diferenciais de 2º grau mostrado em (3.1), aparece frequentemente em uma gama variada de aplicações em análises estruturais e de vibrações como em (BALAS, 1982) e (VANDERVELDE, 1986), onde $\mathbf{M} \in \Re^{n \times n}$, é uma matriz de massas, geralmente definida positiva; $\mathbf{C} \in \Re^{n \times n}$ é uma matriz de amortecimento, geralmente, pelo menos, semi definida positiva; $\mathbf{K} \in \Re^{n \times n}$ é uma matriz de rigidez, também sendo semi definida positiva na maior parte das vezes; $\mathbf{B} \in \Re^{n \times m}$ é uma matriz de controle (também vista como matriz de influência ou de atuadores) de posto completo; $\mathbf{x} \in \Re^n$ é o vetor de estado, que para sistemas mecânicos como os da Figura 2.1, representa o deslocamento e $\mathbf{u} \in \Re^m$ é um vetor de entradas.

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{b}u(t - \tau) \quad (3.1)$$

A saída do sistema definida como:

$$y(t) = e(t - \tau)l\mathbf{x}(t) \quad (3.2)$$

Considere-se o monovariável, ou seja, $l \in \Re^{1 \times n}$.

e erro do sistema é definido como:

$$e(t) = r(t) - y(t) \quad (3.3)$$

Assume-se a lei de controle por realimentação de estados como:

$$u(t - \tau) = k_p e(t - \tau) + k_i \int_0^t e(t - \tau) d\tau + k_d \frac{de(t - \tau)}{dt} \quad (3.4)$$

em que k_p, k_i e $k_d \in \Re$ são os vetores de ganho do controlador. Substituindo a equação (3.4) na equação (3.1) e realizando os procedimentos apresentados na seção 2.1.3.

Considerando:

$$V(s) = q(s)R(s), \text{ com } q(s) = (k_p + \frac{k_i}{s} + k_d s)$$

$$U(s) = -(k_p + \frac{k_i}{s} + k_d s)e^{-\tau s}lX(s) + V(s) \quad (3.5)$$

$$(Ms^2 + Cs + K + e^{-\tau s}q(s)lB)X(s) = V(s) \quad (3.6)$$

$$x(s) = (Ms^2 + Cs + K + e^{-\tau s}q(s)lB)^{-1}V(s) \quad (3.7)$$

aplicando a fórmula de *Sherman-Morrison*, chega-se a matriz de receptância dada por:

$$\hat{\mathbf{H}}(s) = \mathbf{H}(s) - \frac{\mathbf{H}(s)(k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)Be^{-\tau s}}{1 + (k_p + \frac{k_i}{s} + k_d s)l\mathbf{H}(s)Be^{-\tau s}} \quad (3.8)$$

Trabalhos relacionados ao controle em malha fechada aplicados a sistemas de segunda ordem têm ganhado cada vez mais notoriedade devido a sua relevância prática.

3.1 Parâmetro de Projeto e Formulação Matemática

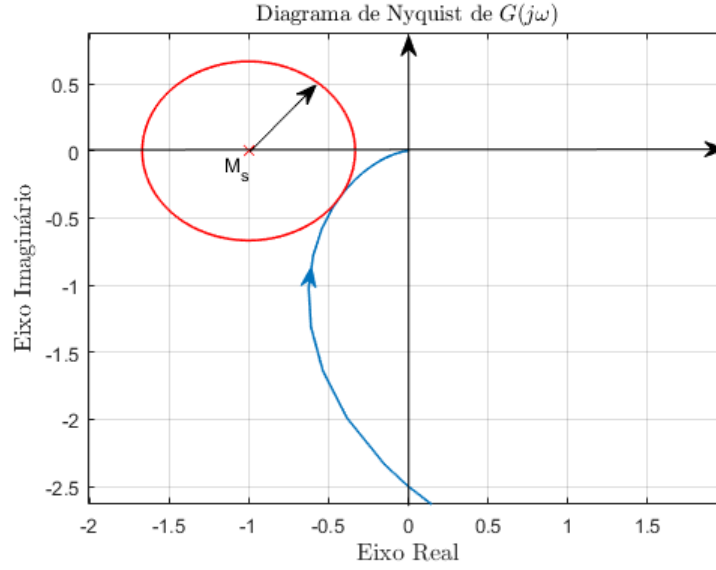
Margens de ganho e de fase são usualmente utilizadas como parâmetros de projeto de controladores por resposta em frequência. As relações entre a resposta temporal e essas margens fornecem boas referências, ainda que dadas de maneira indireta, para o controle de sistemas via resposta em frequência. Em algumas referências essas margens são chamadas de margens de estabilidade, muito embora para casos com sistemas com mais de um grau de liberdade essas não sejam medidas de estabilidade muito precisas.

Circunferência M_s

Dadas as discussões anteriores sobre estabilidade com a definição do critério de estabilidade de Nyquist, uma conclusão para sistemas estáveis em malha aberta, ou seja, sem nenhum polo de malha aberta no SPD ($P = 0$), é que não se deve haver nenhum envolvimento do ponto $(-1, 0)$ pelo diagrama de Nyquist do sistema. Assim, surge a ideia de se definir como parâmetro de projeto uma medida que garanta estabilidade visando atender a este critério. (SKOGESTAD e POSTLETHWAITE, 2007) em uma discussão acerca de critérios de picos máximos trazem a ideia da relação entre o chamado pico máximo da função de sensibilidade e a menor distância entre o ponto $(-1, 0)$ e o diagrama de Nyquist do ganho de malha do sistema. Baseado nesse conceito, define-se uma circunferência de raio M_s^{-1} centrada no ponto $-1 + j0$ delimitando a região onde o traçado de *Nyquist* é permitido, assim, garantindo uma distância segura do ponto $(-1, 0)$, ilustrando a ideia de sistema robusto.

Definição do problema de otimização

A figura 3.1 mostra o diagrama de Nyquist de um sistema genérico (traçado em azul) e uma circunferência (em vermelho) representando a circunferência M_s . À medida que o ganho do sistema aumenta o diagrama de Nyquist invade a área da circunferência e se aproxima do ponto $-1 + j0$ chegando próximo da fronteira da instabilidade. Com essa ideia, define-se como parâmetro de projeto que o sistema tenha uma distância mínima

Figura 3.1: Diagrama de Nyquist Exemplo de Circunferência. M_s

Fonte: DANTAS (2019)

do ponto $(-1, 0)$ definida pelo raio dessa circunferência. A solução apresentada neste trabalho visa garantir uma estabilidade robusta, a fim de garantir que mesmo sujeito a alterações de parâmetros o sistema seja estável. Para isso deve-se manter o ponto $(-1, 0)$ a uma distância segura de $L(j\omega)$, o que pode ser interpretado como o traçado de $L(j\omega)$ fora de uma circunferência centrada em $(-1, 0)$ cujo raio seria a distância mínima desejada com o cuidado de não apresentar soluções nas quais as curvas de $L(j\omega)$ estejam muito afastadas do círculo, o que levaria a respostas muito lenta. Dessa forma, busca-se fazer com que a curva de $L(j\omega)$ tangencie o círculo M_s estabelecido e então o seguinte problema de otimização pode ser formulado:

$$\min_{k_p, k_i, k_d} h(k_p, k_i, k_d) = \left(\min_{\omega_i} |L(j\omega) + 1| - M_s^{-1} \right)^2 \quad (3.9)$$

$$\begin{aligned} s.a. \quad L(j\omega_i) &= \left(k_p + \frac{k_i}{j\omega_i} + k_d j\omega_i \right) l \mathbf{H}(j\omega_i) B e^{-j\omega_i \tau} \\ \text{Re} \{L(j\omega_i)\} &\geq -1 + M_s^{-1} \quad \forall \quad \omega_i / \text{Im} \{L(j\omega_i)\} = 0 \end{aligned} \quad (3.10)$$

sendo que o subscrito i indica o i -ésimo valor de frequência com ω variando de zero a um valor elevado o suficiente para que $|L(j\omega)| \simeq 0$. A equação (3.9) estabelece o problema de minimização da função genérica $h(k_p, k_i, k_d)$, que por ser uma função quadrática terá seu valor mínimo para $(\min_{\omega_i} |L(j\omega) + 1| = M_s^{-1})$, e como $(\min_{\omega_i} |L(j\omega) + 1|)$ é a menor distância entre $L(j\omega)$ e o ponto $(-1, 0)$, a igualdade significa que o valor mínimo para o problema dado pela equação (3.9) se dará num ponto tangente a circunferência de raio M_s^{-1} .

3.2 Otimização de Desempenho

O problema de otimização definido matematicamente pela equação (3.9) estabelece a busca por um conjunto de vetores de ganho de controlador que garantam que a curva de Nyquist do sistema de segunda ordem não adentre a região delimitada pela circunferência centrada no ponto $(-1, 0)$, de raio M_s^{-1} . A restrição dada pela equação (3.10) garante que soluções como a de um conjunto de solução aleatória mostrado na figura 3.2 não sejam obtidas pela busca, pois como ilustrado, a curva de Nyquist da figura 3.2 enlaça o ponto $(-1, 0)$, ilustrando instabilidade MARTINS et al. (2020).

O problema de otimização dado pela equação (3.9) não garante, todavia que o desempenho da resposta seja considerada diretamente, apesar da ideia de robustez, não se garante respostas rápidas tão quanto desejadas somente com a definição da região “proibida” para o traçado da curva de Nyquist de $L(j\omega)$. Em alguns trabalhos na literatura o desempenho do sistema é abordado utilizando-se técnicas de ajuste de curva, sintonia de ganhos do controlador, integral do erro como índice de *IAE* a qual iremos abordar neste trabalho apresentados na seção de 2.4, esses são itens relacionados a resposta temporal EUZÉBIO e BARROS (2015, 2013).

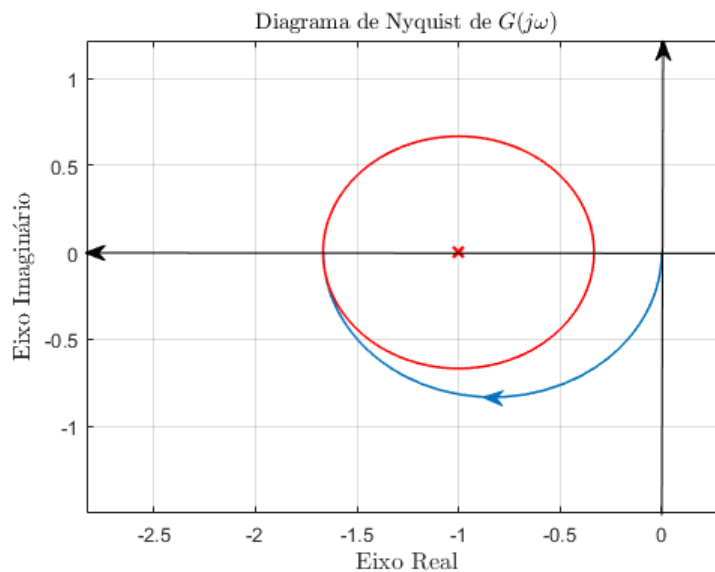


Figura 3.2: Diagrama de Nyquist de um Sistema Genérico para Exemplo de um Caso de Instabilidade.

Fonte: DANTAS (2019)

3.3 Abordagem para sistemas instáveis

Neste trabalho também aplicaremos os métodos testado para sistema de controle estáveis para sistemas instáveis, porém isso traz algumas complicações nas aplicações das funções de restrições.

Pelo critério de *Nyquist*, todas as raízes precisam ser reais e negativas, mas quando temos um sistema instável significa que temos raízes no SPD, e para tornar esse sistema estável, precisamos forçar que o número de voltas no ponto $(-1,0)$ seja igual ao número de raízes semi plano direito, assim tornando o sistema estável. Para isso nosso algoritmo escolhe de forma heurística os ganhos do controlador PID, para atender essa condição de voltas, igual o número de polos de malha aberta no SPD, logo no algoritmo precisar ficar monitorando o número de voltas e quantidade de polos, para atender essa condição.

Esta problemática, de impor o número de voltas adequadas do diagrama de *Nyquist*, é complexo e que neste trabalho visamos estudar para tentar resolver. Inicialmente precisamos implementar uma técnica que consiga contar o número de voltas no ponto $(-1,0)$ no algoritmo 5, desenvolvemos uma maneira de realizar essa contagem.

Capítulo 4

Implementação do Método Proposto

Este capítulo tem como objetivo apresentar o método desenvolvido para alcançar os resultados apresentados no Capítulo 5. A busca por uma solução para o problema de otimização descrito no Capítulo 3 é feita usando um algoritmo genético, um algoritmo estocástico baseado em população que procura soluções aleatoriamente por meio de mutação e cruzamento entre os membros da população. Este algoritmo pode ser utilizado para solucionar problemas de otimização, tanto suaves como não suaves, com restrições lineares ou não lineares MATHWORKS. (2019); GOLDBERG e Holland (1988).

4.1 Solução do Problema de Otimização

Uma das alternativas para solucionar problemas de otimização, como o descrito na equação (3.9), é o uso do algoritmo genético (GA), um método para solução de problemas de otimização com ou sem restrições baseado no processo de seleção natural que imita a evolução biológica. O GA é um pioneiro na solução de problemas do tipo abordado neste trabalho MATHWORKS. (2019). O algoritmo funciona repetitivamente alterando a população de soluções individuais. A cada etapa, o algoritmo genético escolhe aleatoriamente indivíduos da população atual e os utiliza como pais para gerar filhos para a geração seguinte. Ao longo de sucessivas gerações, a população evolui até chegar a uma solução ótima. O que é perfeitamente adequado ao objetivo de procura para o problema descrito pela equação (3.10).

O algoritmo genético pode ser utilizado para resolver problemas que não são adequados aos algoritmos de otimização padrão, incluindo aqueles em que a função objetivo é descontínua, não diferenciável, estocástica ou altamente não linear. A função objetivo neste caso é abordada pela curva de Nyquist do sistema para os valores desejados de **kp**, **ki** e **kd**.

Algoritmo genético desenvolvido

No trabalho apresentado, optou-se por desenvolver o algoritmo genético devido à facilidade de otimização dos hiperparâmetros e obtenção de melhores resultados com variações dos métodos principais do GA, tais como seleção, cruzamento e mutação. SOUZA (2020)

Os passos do algoritmo genético implementado foram:

1. Inicialização: Uma população inicial de soluções é gerada aleatoriamente de **kp**, **ki** e **kd**.
2. Seleção: Os indivíduos são ordenados de forma que os ganhos do controlador que tiveram melhor resultados estão no início da população e esses são selecionados para reprodução, esse tipo de seleção é denominado de elitista.
3. Cruzamento: Os ganhos do PID selecionados são usadas como pais para produzir novos indivíduos (filhos).
4. Mutação: Alguns dos indivíduos produzidos são submetidos a pequenas alterações aleatórias (mutações).
5. Avaliação: A nova geração de soluções é avaliada em termos de sua aptidão com função de otimização, em nosso trabalho utilizamos robustez do sistema, integral do erro absoluto e concorrência entre eles.
6. Variabilidade Genética: Implementamos esse calculo a fim de verificar a variação dos resultados da função de otimizando, reduzindo o tempo de otimização desnecessários quando não tem ganhos significativos a cada interação.
7. Substituição: A nova geração de soluções é substituída pela antiga, formando a nova população.
8. Repetição: O processo é repetido sucessivas vezes até que se atinja uma solução ótima ou se alcance um critério de parada.
9. Resultado: A melhor solução encontrada é retornada como resultado do algoritmo genético.

Essas variações no comportamento do passo de seleção, como seleção por elitismo, seleção randômica, seleção por roleta, bem como variações no método de cruzamento, como cruzamento em dois pontos ou uniforme, e a mutação gaussiana ou uniforme, podem trazer grandes ganhos na otimização do GA, dependendo da formulação do problema e de como ele se comporta no espaço de amostra SOUZA (2020).

O objetivo é seguir as etapas do Algoritmo 1 para encontrar os valores ideais de K_p , K_i e K_d que resolvem o problema de otimização específico apresentado na seção 5. Para utilizar essa função, é necessário fornecer matrizes que descreva o sistema de segunda ordem (**M**, **C**, **K**, **B**, **I**), o atraso do sistema τ , as frequências em que o sistema atua ω , a circunferência M_s e o nome do arquivo de simulação do sistema **Simulink**.

O programa também utiliza funções personalizadas, como uma função de restrição para descrever as limitações impostas pelas equações (3.10), uma função de robustez que avalia a capacidade do sistema em malha fechada de lidar com variações, conforme definida pela equação (3.9), e uma função de *IAE* que calcula a integral do erro absoluto no sistema em malha fechada, conforme definido pela equação (2.31), e uma função *Fitness* que calcula o valor a ser minimizado pelo algoritmo genético. Dependendo do valor da variável *alfa*, a robustez do sistema ou o *IAE* em malha fechada podem ser otimizados, bem como a combinação desses parâmetros no sistema.

Algoritmo 1: Função de Busca

Entrada: M ; C ; K ; B ; I ; τ ; ω ; M_s e **simulink**
Saída: gain = [K_p K_i K_d];

```

1 populacao = random(n);
2 objetivo = "execute";
3 execucao = 0;
4 variabilidade = 1;
5 evolucao = 0;
6 alfa = random(0,1);
7 n = 100;
8 repita
9   para geracao=1 até geracao=n faça
10    avalicao = []
11    se execucao > 0 então
12     | populacao(2:n,:) = random(n-1);
13    fim
14    para i = 1 até populacao=n faça
15     |  $L \leftarrow LFunction(M, C, K, B, I, \tau, \omega, populacao(i, 1:3))$ ;
16     |  $robustez \leftarrow RobustezFunction(L(s), M_s)$ ;
17     |  $restrictions \leftarrow RestrictionsFunction(L(s), 0)$ ;
18     |  $IAE \leftarrow IaeFunction(populacao, simulink)$ ;
19     |  $ff \leftarrow FitnessFunction(IAE, robustez, alfa)$ ;
20     |  $avaliacao(i, 1:end) = [ff, restrictions]$ ;
21     |  $i = i + 1$ ;
22    fim
23    populacao = [populacao avaliacao];
24    populacao  $\leftarrow AssortmentFunction(populacao)$ ;
25    populacao  $\leftarrow CrossoverFunction(populacao(1:n, 1:end-2))$ ;
26     $L \leftarrow LFunction(M, C, K, B, I, \tau, \omega, populacao(1, 1:3))$ ;
27     $robustez \leftarrow RobustezFunction(L(s), M_s)$ ;
28     $restrictions \leftarrow RestrictionsFunction(L(s), 0)$ ;
29     $IAE \leftarrow IaeFunction(populacao, simulink)$ ;
30     $ff \leftarrow FitnessFunction(IAE, robustez, alfa)$ ;
31    se ( $ff < 0.4 \ \&\& \ restrictions < 0.9$ ) || ( $execucao >$ 
32     | 2 &&  $Restrictions < 0.9$ ) então
33     | objetivo = "fim";
34     | melhorIndividuo = populacao(1,:);
35     | interromper;
36    fim
37    VariabilidadeFunction(variabilidade, evolucao, ff);
38    geracao = geracao + 1;
39  fim
40 até (objetivo  $\neq$  "fim");

```

População inicial

A primeira etapa na configuração de um algoritmo genético consiste em criar a população inicial. Esse termo, assim como todos os outros relacionados ao algoritmo, é inspirado na analogia com a teoria da evolução biológica. No contexto de otimização, ele se refere ao conjunto inicial de soluções possíveis para o problema específico em questão.

Neste trabalho, optou-se pelo desenvolvimento de um conjunto aleatório de indivíduos para a população. Inicialmente, criamos uma população com cem indivíduos, onde o primeiro *locus* corresponde a K_p , seguido por K_i , e K_d . Ao tentar resolver o problema descrito pela equação (3.9).

4.2 Problemas na implementação

Nesta seção, discutiremos alguns problemas que surgiram durante o desenvolvimento e implementação das teorias apresentadas neste trabalho. Esses problemas resultaram em obstáculos que exigiram mais esforço para serem resolvidos, incluindo dados contraditórios relacionados ao problema de "*detuning*" e "*curva de Nyquist infinita*".

Procedimento baseado em "*detuning*"

A restrição estabelecida pela equação (3.10) tem como objetivo garantir a estabilidade do sistema diante da resposta obtida pelo algoritmo genético. Em alguns casos, como em sistemas sem amortecimento em que os polos de malha aberta estão localizados no eixo imaginário, pode ser desafiador trabalhar com a resposta em frequência devido às discontinuidades que esses pontos geram na relação entre os planos s e $j\omega$). Essa dificuldade pode se refletir na solução retornada pela busca do algoritmo genético, manifestando-se como um mínimo local ou até mesmo como uma solução inexistente para o problema definido com as restrições apresentadas.

Com base na ideia apresentada em Santos (SANTOS et al., 2018) para sistemas com atraso variável e motivado por essa dificuldade, é gerada uma população inicial em que todos os indivíduos possuem soluções estáveis.

Para gerar a população inicial, segue-se o seguinte procedimento. Inicialmente, define-se um conjunto de indivíduos da população inicial $[K_p \ K_i \ K_d]$, sem restrições quanto à sua definição. Uma sugestão é utilizar o vetor calculado pelo método proposto em (MOTTERSHEAD e RAM, 2007) para sistemas sem atraso, ou simplesmente um vetor com ganho unitário. Em seguida, verifica-se a estabilidade do sistema com o ganho de malha $[K_p \ K_i \ K_d]$ estabelecido no passo anterior. Caso o sistema seja instável, multiplica-se o vetor $[K_p \ K_i \ K_d]$ por um valor $0 \leq \alpha \leq 1$ e novamente verifica-se a estabilidade com o novo ganho $\alpha[K_p \ K_i \ K_d]$. Repete-se esse processo, reduzindo gradativamente o valor de α , até atingir a estabilidade. A partir daí, constrói-se o conjunto de indivíduos da população inicial conforme o algoritmo 2, essa solução para esse problema não é tratado neste trabalho.

Algoritmo 2: População Inicial

Entrada: $L(s) = (k_p + \frac{k_i}{s} + k_d s)lH(s)Be^{-\tau s}$;
Saída: populacao

```

1 populacao = [];
2 para i = 1 até i = 100 faça
3      $\alpha = 1 - \frac{(i-1)}{100}$ ;
4     se LFunction( $M, C, K, B, l, \tau, \omega$ , populacao(i, 1:3)) é estável então
5         | populacao=[populacao  $\alpha[Kp \ Ki \ Kd]$ ];
6     fim
7     i = i + 1;
8 fim
```

Erros numéricos causam Nyquist infinito

Além disso, devido à natureza do controlador integrativo, para frequências altas, o cálculo do diagrama de *Nyquist*, devido natureza aleatória da população, pode ocorrer problema numéricos de valores infinitos para o ganho integrativo do PID. Com isso, a depender dos valores de frequência ω , o algoritmo pode ter duas visões diferentes do gráfico de *Nyquist*, e como função de restrição é modelada com base nesse diagrama, o algoritmo pode ter interpretação errada.

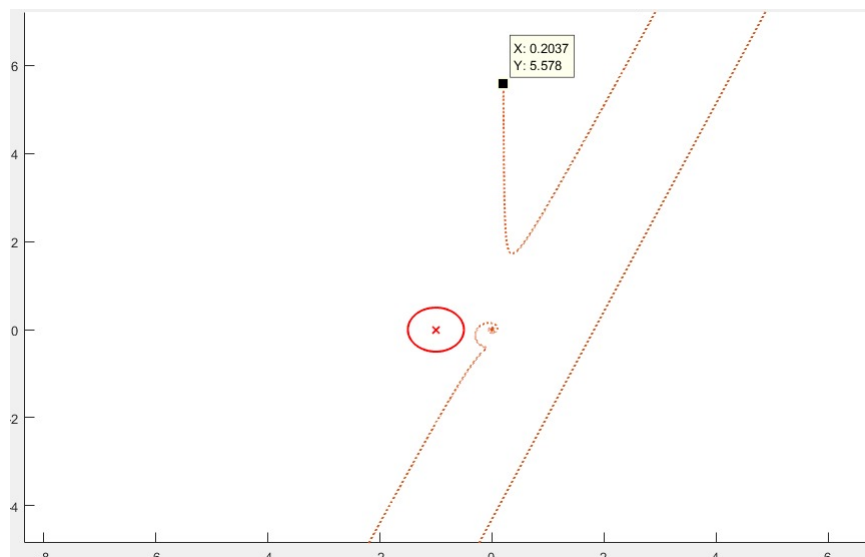
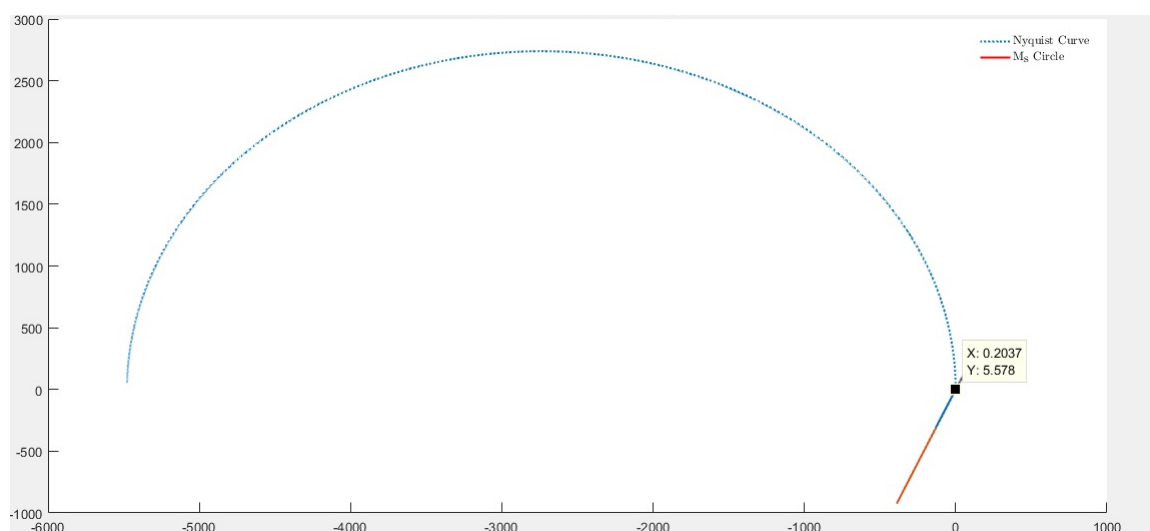
Por exemplo, quando utilizamos o sistema do exemplo 5.1.3, do capítulo 5 o método resultou nos seguintes ganhos do controlador PID [7.9574 14.8018 3.5920] respectivamente Kp, Ki, Kd na figura 4.1, o qual as frequências ω tem uma variação de $[10^{-1} \ 10^3]$ podemos observar que o diagrama não envolve o ponto de instabilidade $-1 - j\omega$, e com base na definição de estabilidade de *Nyquist* o sistema é estável em malha aberta. Mas, para os mesmo valores dos ganhos, com a variação do ω de $[10^{-6} \ 10^3]$, quando observamos a figura 4.2, o gráfico envolve o ponto de instabilidade, assim quando ocorre um diagrama que tem casos infinitos, o algoritmo não consegue enxergar que o sistema é instável, errando a solução.

Como solução inicial, introduzimos uma perturbação na ação integrativa, com objetivo de evitar valores infinitos para cálculo do diagrama de *Nyquist* conforme a função L exposta no algoritmo 3.

4.3 A Função Robustez

A função de robustez descrita no Algoritmo 4 representa um dos problemas de otimização abordados neste trabalho, definido pela equação (3.9). Essa função é utilizada como medida de avaliação dos indivíduos da população e tem como objetivo garantir que a menor distância entre o ponto $-1 + j0$ e o diagrama de *Nyquist* de $L(j\omega)$ seja igual ao raio da circunferência M_s , um parâmetro de projeto definido na resposta em frequência.

A função L apresentada no Algoritmo 3 calcula os valores de $L(j\omega_i)$ e os armazena nos vetores Re e Im , correspondendo às partes real e imaginária de $L(j\omega_i)$, respectivamente. O vetor d é utilizado para armazenar as distâncias entre o ponto $(-1, 0)$ e os pontos ao

Figura 4.1: *Nyquist* com visão do sistema estávelFigura 4.2: *Nyquist* com visão do sistema instável

Algoritmo 3: Função L

Entrada: \mathbf{M} ; \mathbf{C} ; \mathbf{K} ; \mathbf{B} ; \mathbf{l} ; τ ; ω e *populacao***Saída:** L

```

1  $K_p = \text{populacao}(1);$ 
2  $K_i = \text{populacao}(2);$ 
3  $K_d = \text{populacao}(3);$ 
4 para  $i = 1$  até  $i = \text{comprimento}(\omega)$  faça
5    $\beta = j \times \omega(i);$ 
6    $L(i) = (K_p + K_i/\beta + 0.0001) + \beta \times K_d) \times \mathbf{l} \times (\mathbf{M} \times \beta^2 + \mathbf{C} \times \beta + \mathbf{K})^{-1} \times \mathbf{B} \times e^{-\tau \times \beta};$ 
7    $i = i + 1;$ 
8 fim
```

longo da curva $L(j\omega_i)$ que foram armazenados nos vetores Re e Im . Finalmente, a função de robustez é a medida de avaliação para cada indivíduo da população (que é um candidato a solução do problema de otimização) e o seu valor é utilizado na avaliação realizada pelo Algoritmo Genético (GA).

Algoritmo 4: Função Objetivo

Entrada: L e M_s **Saída:** robustez

- 1 $d = \sqrt{(\text{Re}(L) + 1)^2 + \text{Im}(L)^2}$;
 - 2 $\text{robustez} = (\min(d) - M_s^{-1})^2$;
-

4.4 A Função de Restrições

O algoritmo 5 apresenta a função de restrição responsável por estabelecer as limitações do problema de otimização, definidas pelas equações (3.10). O vetor "*cross*" armazena os pontos de intersecção entre a curva de Nyquist de $L(j\omega)$ e o eixo real. A variável "*restrictions*" é utilizada para representar a restrição não linear estabelecida pela inequação (3.10) e é restrita a valores menores ou iguais a zero.

A função de restrição envia os valores de *restrictions* de volta para o Algoritmo Genético e avalia o indivíduo da população com a função fitness. A busca é concluída quando as mudanças entre as gerações se tornam insignificantes e o melhor indivíduo avaliado pelas funções descritas é apresentado como resposta. Os resultados obtidos com o algoritmo desenvolvido são apresentados no capítulo 5.

Algoritmo 5: Função de Restrição

```

Entrada: L; Z;
Saída: restrictions
1 contadorcw = 0;
2 contadorccw = 0;
3 cross = 0;
4 se  $\text{Re}(L(1)) < -1 \&\& \text{abs}(\text{Im}(L(1))) < 0.2$  então
5   | xcrosses = [xcrosses Re(Ls(1))];
6   | se  $\text{Im}(L(2)) < \text{Im}(L(1))$  então
7   |   | contadorccw = contadorccw + 1;
8   |   fim
9   | senão
10  |   | contadorcw = contadorcw + 1;
11  |   fim
12 fim
13 para i = 1 até comprimento(L) - 1 faça
14   | se  $\text{Im}(L(i)) \times \text{Im}(L(i+1)) < 0$  então
15   |   | se  $\text{Re}(L(i)) \times \text{Re}(L(i+1))/2 < -1$  então
16   |   |   | cross = [cross Re(L(i)) + Re(L(i+1))]/2];
17   |   |   | se  $\text{Im}(L(i+1)) < \text{Im}(L(i))$  então
18   |   |   |   | contadorccw = contadorccw + 1;
19   |   |   |   fim
20   |   |   | senão
21   |   |   |   | contadorcw = contadorcw + 1;
22   |   |   |   fim
23   |   |   fim
24   |   fim
25   | i = i + 1;
26 fim
27 para i = 1 até comprimento(L) - 1 faça
28   | se  $-\text{Im}(L(\text{comprimento}(L) - (i - 1))) \times -\text{Im}(L(\text{comprimento}(L) - i)) < 0$ 
29   |   | então
30   |   |   | se
31   |   |   |   |  $\text{Re}(L(\text{comprimento}(L) - (i - 1))) + \text{Re}(L(\text{comprimento}(L))) / 2 < -1$ 
32   |   |   |   | então
33   |   |   |   |   | cross = [cross Re(L(comprimento(L) - (i - 1))) + Re(L(comprimento(L) - i))]/2];
34   |   |   |   |   | se
35   |   |   |   |   |   |  $-\text{Im}(L(\text{comprimento}(L) - (i - 1))) > -\text{Im}(L(\text{comprimento}(L) - i))$ 
36   |   |   |   |   |   | então
37   |   |   |   |   |   |   | contadorccw = contadorccw + 1;
38   |   |   |   |   |   |   fim
39   |   |   |   |   | senão
40   |   |   |   |   |   | contadorcw = contadorcw + 1;
41   |   |   |   |   fim
42   |   |   |   fim
43   |   |   fim
44   |   fim
45   | i = i + 1;
46 fim
47 restrictions =  $-\min(\text{cross}) \times \text{contadorcw} + (Z - \text{contadorccw})^2$ ;

```

4.5 A Função de *IAE*

A função de Integral do Erro Absoluto é um indicador que avalia o desempenho do sistema conforme descrito na seção 2.4. Esse indicador considera o erro do sistema, e para calculá-lo, a maneira mais simples é executar a simulação do sistema de segunda ordem que está sendo estudado.

Ao usar o MATLAB/Simulink®, uma ferramenta que simula sistemas de controle de forma mais precisa em relação a sistemas reais, é possível medir o erro do sistema considerando o atraso do mesmo, o que é um dos objetivos deste trabalho. A figura 4.3 apresenta a simulação desenvolvida que utiliza as matrizes de estado do sistema, considera o atraso e está acoplada ao controlador PID. Nessa simulação, o *IAE* é calculado e o valor resultante é salvo em uma variável com o mesmo nome.

O Pseudocódigo 6 emprega a simulação mencionada para calcular o índice *IAE* para cada indivíduo da população, sendo que cada indivíduo é composto por K_p , K_i e K_d . As demais variáveis necessárias para a execução da simulação são carregadas no *Workspace* do software MATLAB®. A função importa os parâmetros do controlador para a simulação e a executa, utilizando as variáveis do *Workspace*. Após a conclusão da simulação, a função retorna o valor do *IAE*.

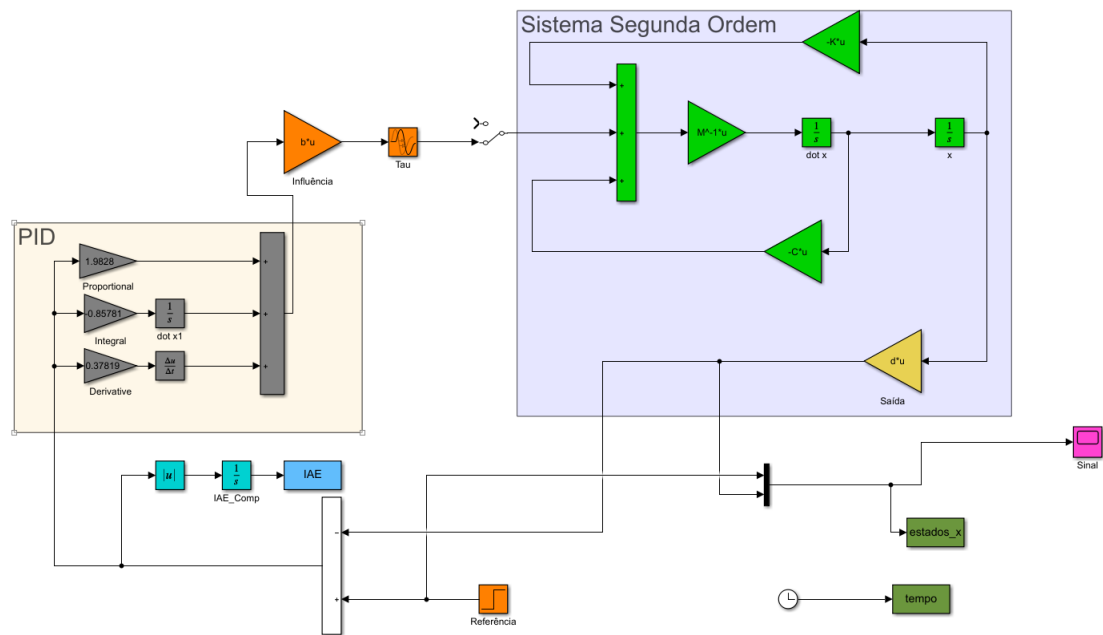


Figura 4.3: Simulação criada com MATLAB/Simulink®

4.6 A Função de *Fitness*

A função *Fitness* recebe como entrada dois parâmetros: os índices de robustez e *IAE*, resultantes das funções descritas nas seções 4.3 e 4.5, respectivamente. O índice de robustez mede o quão robusto é o sistema, ou seja, o quão improvável é que ele se torne

Algoritmo 6: Função IAE**Entrada:** populacao e simulink**Saída:** IAE

```

1 Kp = populacao(1);
2 Ki = populacao(2);
3 Kd = populacao(3);
4 setParam(strcat(simulink,"/Proportional"),"Gain",num2str(Kp));
5 setParam(strcat(simulink,"/Integral"),"Gain",num2str(Ki));
6 setParam(strcat(simulink,"/Derivative"),"Gain",num2str(Kd));
7 sim(simulink);
8 IAE = IAE(length(IAE));

```

instável. Por sua vez, o *IAE* mede a área abaixo do gráfico do erro do sistema, sendo que o erro representa a distância entre o sistema e o sinal desejado.

Entretanto, é possível notar que as definições dessas duas medidas podem entrar em conflito, já que quando se deseja minimizar o erro do sistema, ou seja, ter um tempo de acomodação baixo, pode-se acabar com um sistema menos robusto, com altos valores de *Overshoot*. Como solução, foi criado um parâmetro chamado alfa, que pondera esses dois critérios na função de fitness. O valor de alfa é um número entre zero e um, permitindo escolher se dar maior importância ao índice de robustez ou ao *IAE*, bem como decidir se ambos têm a mesma importância na otimização do algoritmo de busca. O algoritmo 7 apresenta essa lógica.

Algoritmo 7: Função Fitness**Entrada:** Robustez, IAE, e alfa**Saída:** ff

```

1 ff = alfa × robustez + (1 - alfa) × IAE;

```

4.7 A Função de Variabilidade

O objetivo da Função de Variabilidade Genética, apresentada no Algoritmo 8, é solucionar o problema de iteração do Algoritmo Genético quando não houver melhora no valor da Função *Fitness* após várias iterações. Com isso, evita-se a execução desnecessária do programa quando a solução estiver presa em um mínimo local, no qual não haverá avanços significativos com a continuidade das execuções do *GA*.

A função avalia o valor da função objetivo em dez execuções. Caso haja uma baixa variância entre esses valores, indica que o algoritmo não está apresentando evolução na otimização e que o custo computacional de continuar executando o algoritmo proposto neste trabalho não resultará em ganhos significativos. Com isso, a função interrompe a execução do *GA* e retorna o melhor indivíduo para aquela execução.

Algoritmo 8: Função de Variabilidade

Entrada: variabilidade, evolucao, e **ff**

```
1 evolucao(variabilidade) = ff;  
2 variabilidade = variabilidade + 1;  
3 se tamanho(evolucao) > 10 então  
4   |   variabilidade = 1;  
5   |   evolucao = evolucao(2:end);  
6   |   se var(evaluation) < 0.0000001 então  
7   |   |   objetivo = "fim";  
8   |   |   melhorIndividuo = populacao(1,:);  
9   |   |   interromper;  
10  |   fim  
11 fim
```

Capítulo 5

Experimentos e Resultados

O objetivo deste estudo é analisar a eficácia do controlador PID em sistemas de segunda ordem com atraso estáveis em malha aberta. Para a otimização do PID, são utilizados dois parâmetros de busca.

A primeira otimização se baseia no critério de robustez do sistema, e seus resultados são apresentados na seção 5.2. A segunda otimização é baseada no índice de *IAE*, buscando alcançar uma maior eficiência do sistema, e seus resultados estão descritos na seção 5.3.

Podemos encontrar na seção 5.4 uma comparação entre os resultados obtidos quando otimizamos a robustez do sistema e o índice de Integral do Erro Absoluto.

Podemos inferir que esses parâmetros de avaliação do sistema são mutuamente exclusivos, o que significa que é difícil obter um sistema altamente robusto e eficiente ao mesmo tempo. Quando minimizamos o índice *IAE*, a robustez do sistema pode ser comprometida. Na seção 5.5, tentamos encontrar um equilíbrio para otimizar tanto a robustez quanto o índice de *IAE*, a fim de obter um controlador ideal que possa manter a estabilidade e o tempo de acomodação mesmo em situações de perturbações no sistema.

5.1 Experimentos estudados

Os exemplos que serão estudados nesta seção são baseados em literatura de sistemas de controle, como descrito em OGATA (2009). Abordaremos sistemas de controle de segunda ordem com atraso e detalharemos os resultados obtidos com o método proposto, além de compará-los com trabalhos já realizados, conforme descrito em DANTAS (2019).

5.1.1 Exemplo 1

Na Figura 5.1, é apresentado um exemplo clássico de aplicação do sistema massa-mola-amortecedor de um grau de liberdade. O problema de controle por realimentação de estados é definido para $\mathbf{M} = 1$, $\mathbf{C} = 0,01$, $\mathbf{K} = 5$, $\mathbf{B} = 1$ e $\tau = 0,1$ como no Exemplo 2 de (RAM et al., 2009). O objetivo é encontrar valores de \mathbf{K}_p , \mathbf{K}_i e \mathbf{K}_d , de forma que o diagrama de Nyquist de $L(j\omega)$ esteja completamente fora da área delimitada pela circunferência de raio M_s^{-1} , que é um parâmetro de projeto definido. A definição de M_s

estabelece margens de ganho e de fase para o sistema, e valores comuns de M_s estão dentro do intervalo de $1,22 \leq M_s \leq 1,667$ SKOGESTAD e POSTLETHWAITE (2007).

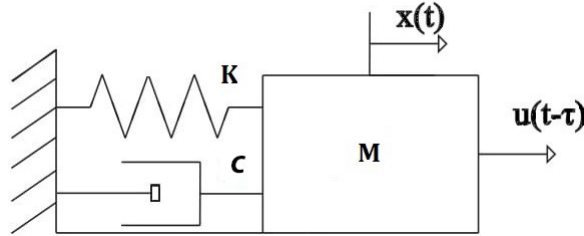


Figura 5.1: Sistema massa-mola-amortecedor de um grau de liberdade

O método proposto neste trabalho foca nos parâmetros de projeto baseados no diagrama de Nyquist do ganho de malha do sistema $L(j\omega)$, considerando os conceitos de estabilidade segundo os critérios de Nyquist. O objetivo da solução é obter um sistema relativamente robusto, tendo como meta $M_s = 1,66$.

5.1.2 Exemplo 2

Considerando um exemplo prático para demonstrar a aplicação do método apresentado, conforme visto em (RAM et al., 2011), podemos tomar as matrizes de massa, amortecimento e elasticidade como:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$$

Considerando o método apresentado em (RAM et al., 2011), vamos utilizar como exemplo prático uma matriz de entradas $\mathbf{B}' = \mathbf{I}' = [0 \ 1]$ e uma constante de atraso $\tau = 5$. Nesse caso, estamos lidando com um sistema não amortecido com autovalores de malha aberta sobre o eixo imaginário do plano s . Quando isso ocorre, pode ser difícil trabalhar com a resposta em frequência, uma vez que há uma descontinuidade nas frequências que coincidem com os autovalores do sistema no eixo imaginário. Para lidar com essa situação, a literatura de sistemas de controle com base na teoria do mapeamento sugere fazer uma pequena alteração no contorno do plano s , de modo a evitar os polos sobre o eixo $j\omega$. Em outras palavras, é necessário deslocar levemente os polos para a esquerda do eixo, o que pode ser conseguido por meio de uma pequena alteração em um dos elementos da matriz de amortecimento OGATA (2009). Nesse exemplo, a matriz \mathbf{C} é considerada.

$$\mathbf{C} = \begin{bmatrix} 1 & -0,9999 \\ -0,9999 & 1 \end{bmatrix}$$

Novamente a busca pela solução do problema descrito pela equação (3.9) é realizada para o círculo $M_s = 1,6667$.

5.1.3 Exemplo 3

Neste caso específico, foi feita uma pequena modificação do exemplo 5.1.2, com o objetivo de introduzir uma perturbação no sistema.

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$$

Para uma matriz de entradas $\mathbf{B}' = \mathbf{I}' = [0 \ 1]$, e uma constante de atraso $\tau = 0,5$.

5.2 Resultados da Otimização de Robustez

O método apresentado aqui trata dos parâmetros de design com base no gráfico de Nyquist da função de transferência em malha aberta do sistema $L(j\omega)$. Ele utiliza os critérios de estabilidade de Nyquist para alcançar uma solução que satisfaça um sistema razoavelmente robusto, com a circunferência $M_s = 1,6667$.

Os valores dos vetores de ganho descobertos utilizando o algoritmo genético (método proposto) com atraso estão apresentados na Tabela 5.1.

Exemplo	kp	ki	kd	robustez
1	1,4160	1,5225	0,8718	0,1536
2	0,1774	0,2229	0,4433	0,0001
3	-0,2400	1,0990	1,6015	0,3148

Tabela 5.1: Tabela de Ganhos do PID - Otimização de Robustez

5.2.1 Resultado exemplo 1

Ao aplicar o algoritmo genético ao 5.1.1 e definir a robustez como parâmetro de otimização, a Figura 5.2 é gerada. É possível observar que o diagrama está tangente à circunferência M_s^{-1} , que tem como objetivo garantir a robustez do sistema.

A Figura 5.3 apresenta uma simulação que demonstra um sistema estável e robusto, de acordo com a teoria descrita na seção 2.5, a qual utiliza o diagrama de *Nyquist*.

5.2.2 Resultado exemplo 2

Ao considerar o exemplo 5.1.2, que envolve um sistema mais complexo, a Figura 5.4 mostra o diagrama de Nyquist, que também evidencia a robustez do sistema. Isso é possível observar pelo fato do diagrama não incluir o ponto $-1 + j0$.

Utilizando um valor de $\tau = 5$, foi possível obter uma medida de robustez mais flexível. Ao comparar com os resultados do trabalho anterior DANTAS (2019), a resposta simulada mostrada na Figura 5.5 apresentou um tempo de acomodação médio de sessenta segundos, sendo que, para este trabalho, o sinal de referência foi definido no nível um. Já no trabalho de DANTAS (2019), o sinal de referência foi definido em zero.

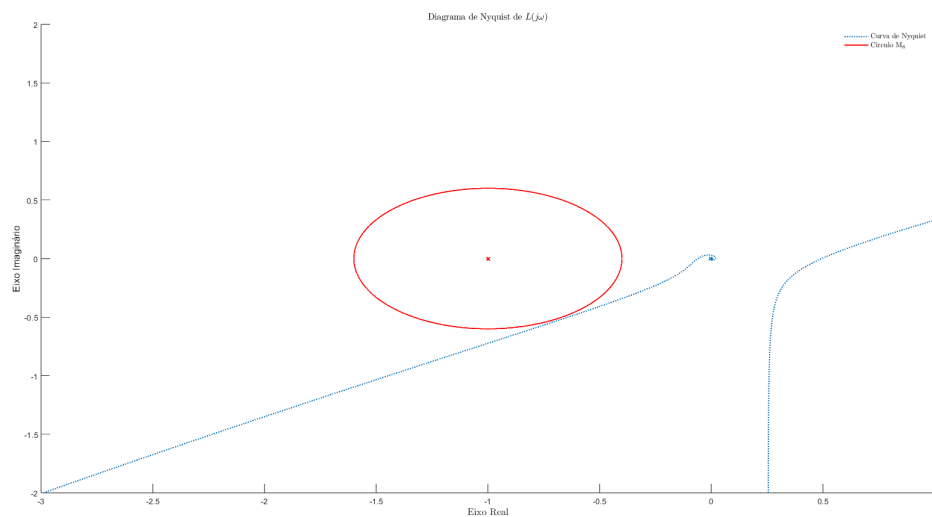


Figura 5.2: Diagrama de Nyquist Exemplo 5.1.1

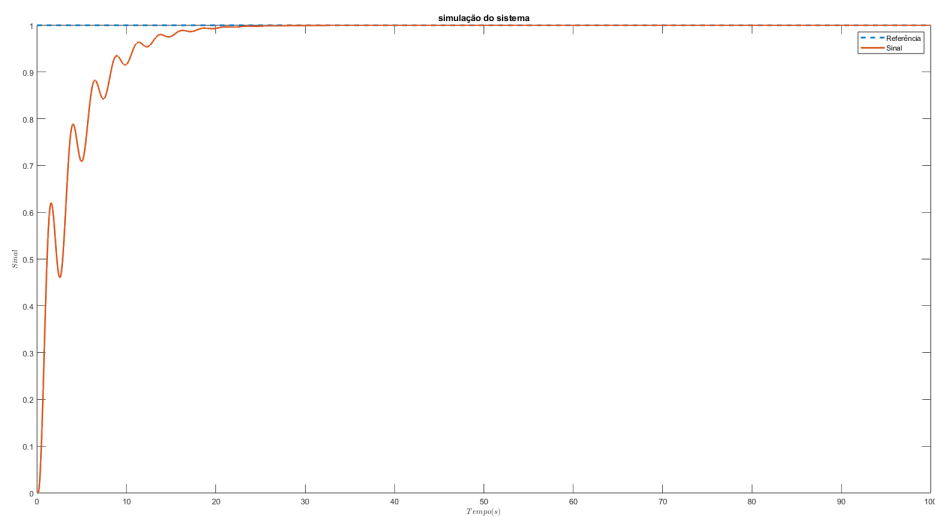


Figura 5.3: Simulação do Sistema do Exemplo 5.1.1

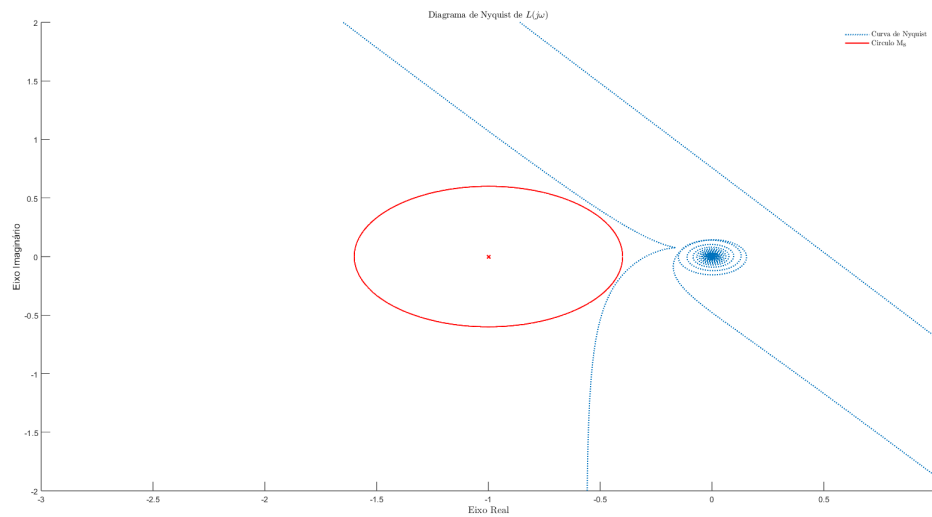


Figura 5.4: Diagrama de Nyquist Exemplo 5.1.2

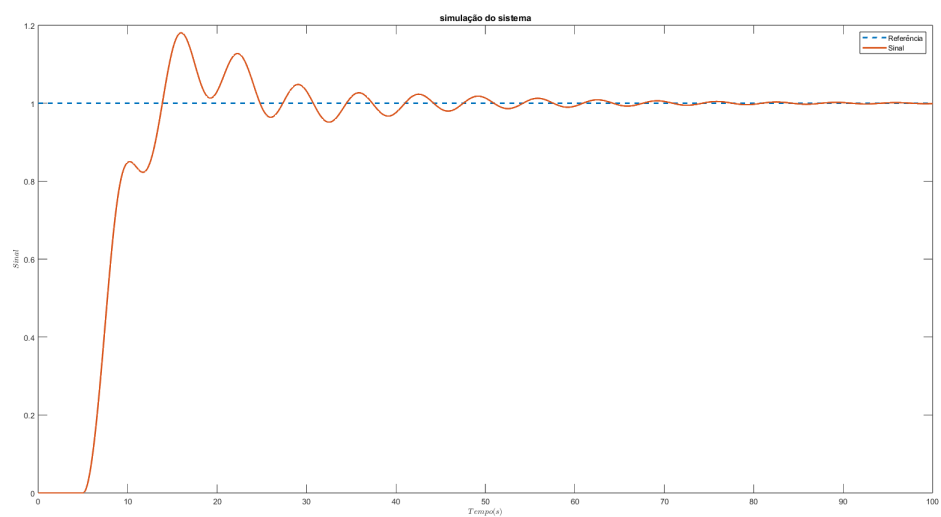


Figura 5.5: Simulação do Sistema do Exemplo 5.1.2

5.2.3 Resultado exemplo 3

No exemplo 5.1.3, houve uma alteração na matriz **C** em relação ao exemplo 5.1.2. O gráfico de Nyquist resultante é apresentado na Figura 5.6, evidenciando a robustez do sistema de segunda ordem com atraso. A simulação do sistema no exemplo 5.1.3 foi realizada no MATLAB/Simulink[®], e o tempo de acomodação foi ainda mais reduzido, em torno de trinta segundos, conforme evidenciado na Figura 5.7.

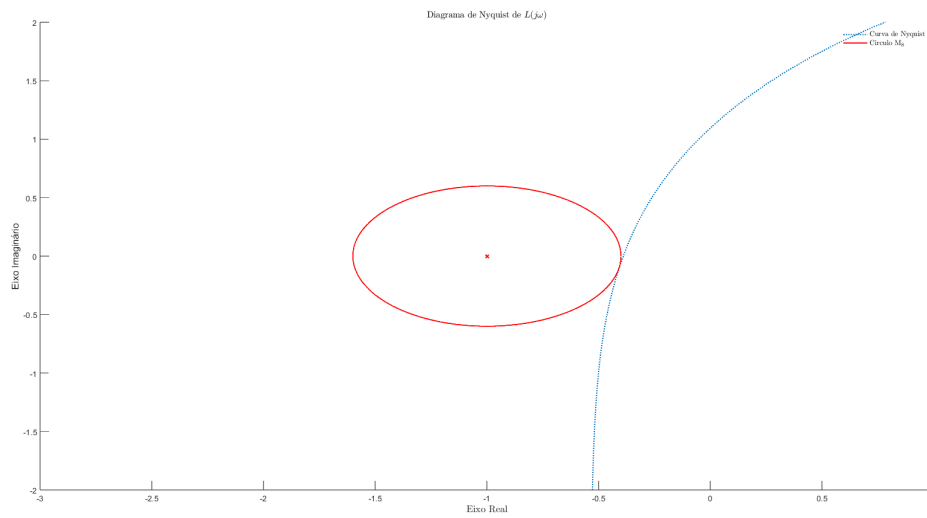


Figura 5.6: Diagrama de Nyquist Exemplo 5.1.3

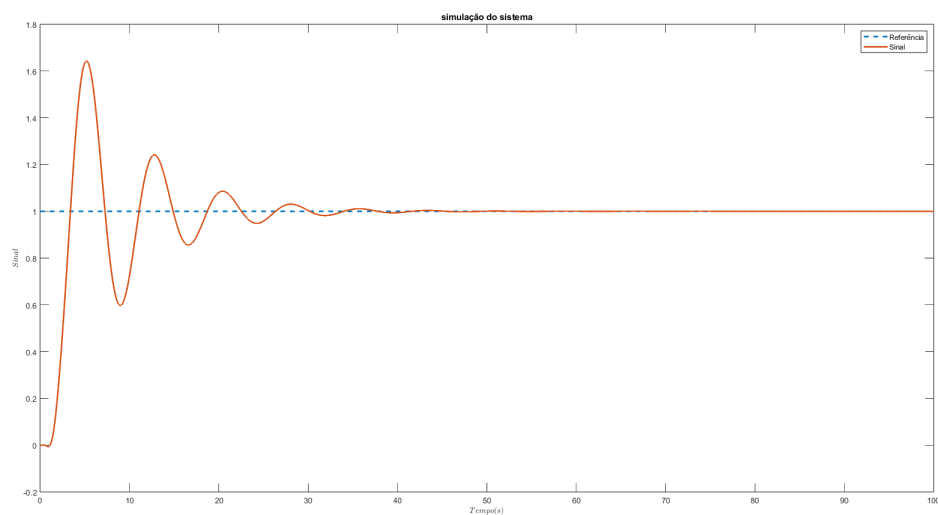


Figura 5.7: Simulação do Sistema do Exemplo 5.1.3

5.3 Resultados da Otimização de IAE

Os resultados da otimização com o índice Integral do Erro Absoluto são apresentados neste trecho, em comparação aos resultados teóricos discutidos na seção 5.2. Espera-se que essa abordagem resulte em um desempenho mais eficiente, com um tempo de acomodação reduzido, uma vez que não é necessário que o sistema de controle seja robusto. No entanto, o critério de estabilidade de *Nyquist* ainda foi usado para garantir a estabilidade do sistema, com a circunferência $M_s = 1,6667$. A Tabela 5.2 fornece os valores dos ganhos obtidos pelo algoritmo genético.

Exemplo	kp	ki	kd	IAE
1	1,5467	4,7074	2,8071	1,0622
2	0,5146	0,2539	0,4016	7,3368
3	1,7283	1,0606	1,9955	2,2072

Tabela 5.2: Tabela de Ganhos do PID - Otimização de IAE

5.3.1 Resultado exemplo 1

Utilizando o algoritmo genético (GA), foi realizada a otimização do índice *IAE* por meio de simulação computacional do exemplo 5.1.1. Como resultado, o gráfico 5.8 demonstrou que o sistema não deve obrigatoriamente tangenciar a circunferência que representa a sua robustez.

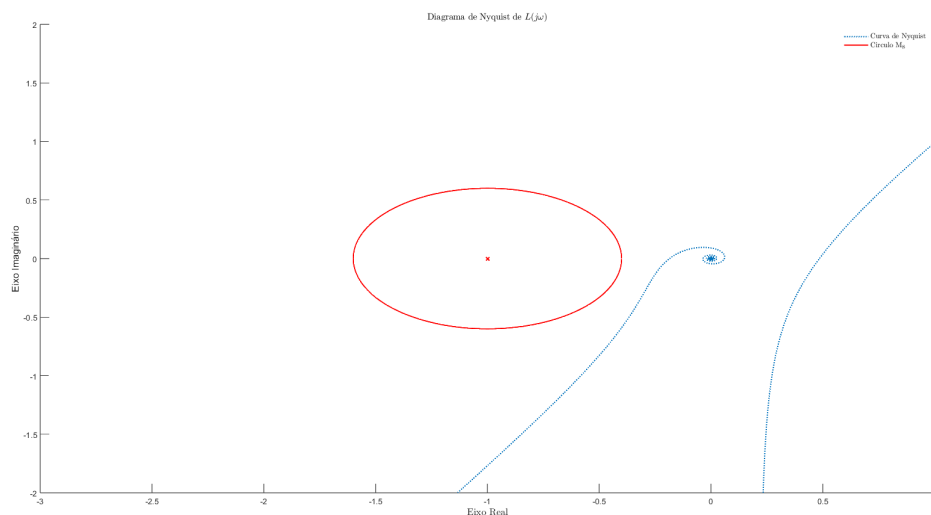


Figura 5.8: Diagrama de Nyquist Exemplo 5.1.1

Ao examinarmos o gráfico 5.9, podemos notar que o sistema alcança o estado estacionário em poucos segundos, com um tempo de acomodação menor do que o apresentado na seção 5.2, o que é um resultado desejado neste projeto.

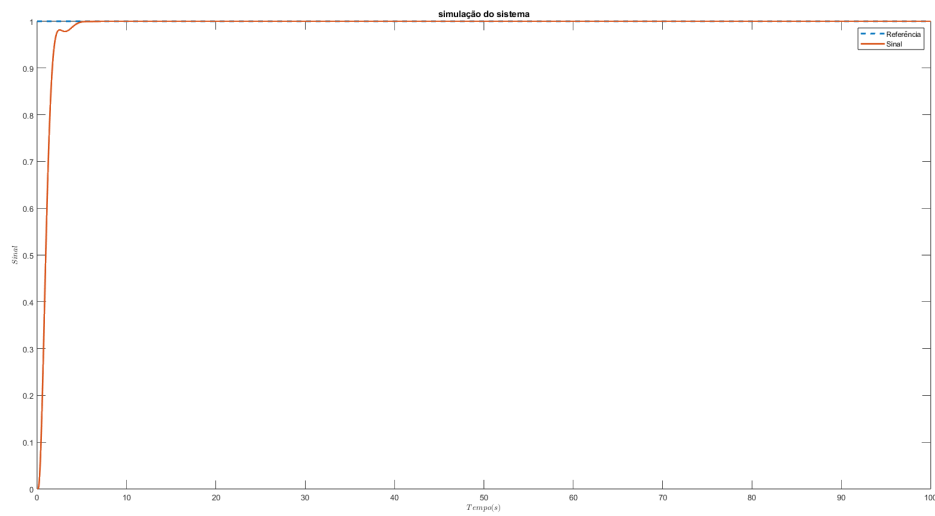


Figura 5.9: Simulação do Sistema do Exemplo 5.1.1

5.3.2 Resultado exemplo 2

No exemplo 5.1.2, que é mais complexo e exigiu mais esforço para encontrar uma solução adequada, a Figura 5.10 mostra que o diagrama passa pela circunferência M_s , o que não torna o sistema instável, como indicado pela simulação na Figura 5.11. No entanto, a otimização da Integral do Erro Absoluto tornou o sistema mais eficiente, apresentando menor *overshoot* e tempo de acomodação.

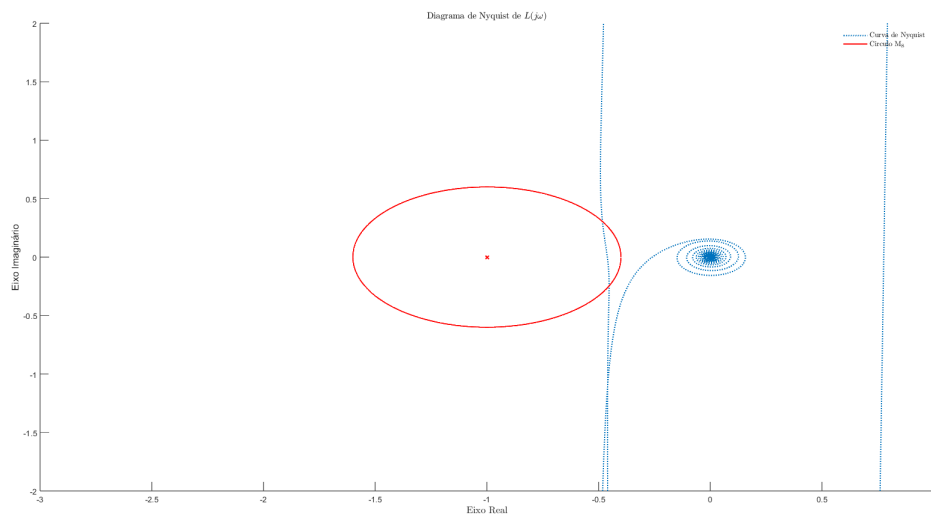


Figura 5.10: Diagrama de Nyquist Exemplo 5.1.2

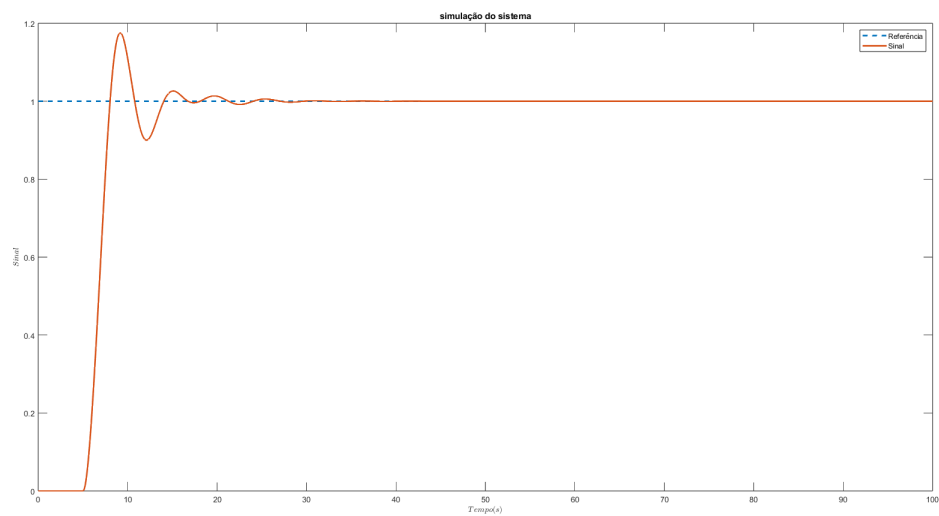


Figura 5.11: Simulação do Sistema do Exemplo 5.1.2

5.3.3 Resultado exemplo 3

Através da otimização desse parâmetro no exemplo 5.1.3, foi possível observar que o *Nyquist* na Figura 5.12 se afastou mais da circunferência de robustez. Na Figura 5.13, que apresenta o resultado da simulação do sistema para esse exemplo, pode-se notar que o tempo de acomodação é menor em comparação à busca pela robustez apresentada na seção 5.2.3 e até mesmo no trabalho original DANTAS (2019).

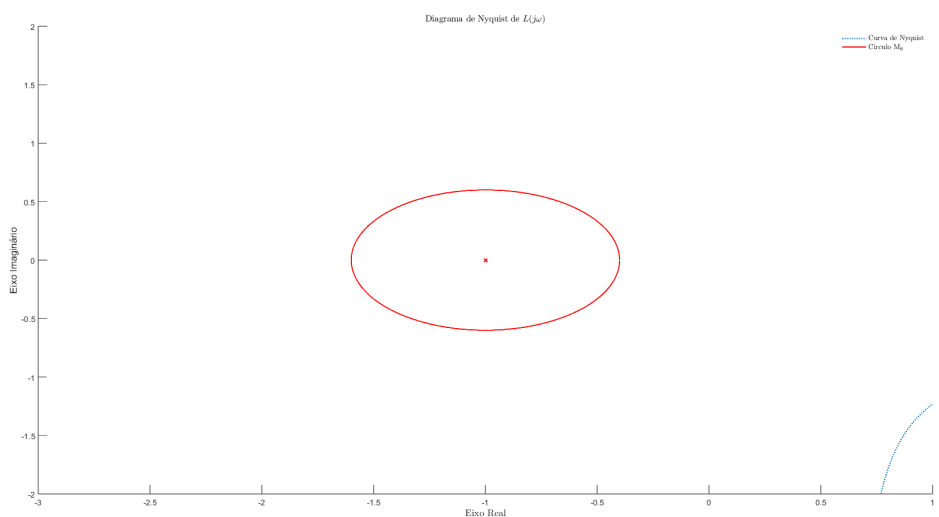


Figura 5.12: Diagrama de Nyquist Exemplo 5.1.3

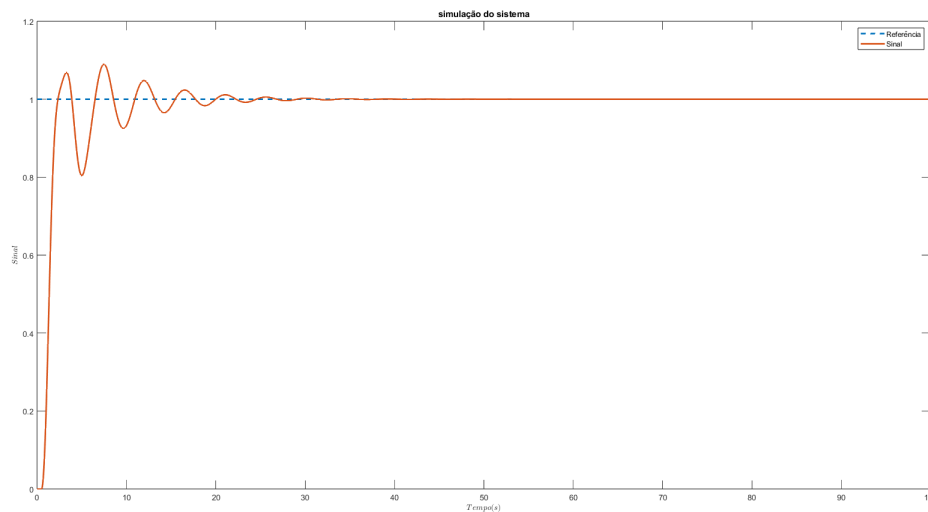


Figura 5.13: Simulação do Sistema do Exemplo 5.1.3

5.4 Resultados da Comparação entre as Otimização

Nesta seção, faremos uma análise comparativa dos parâmetros otimizados diretamente e observaremos os benefícios de realizar a busca pelo critério de robustez ou pelo índice de *IAE*. A ideia inicial é comparar os gráficos de *Nyquist* e as simulações do sistema, sobrepondo os resultados. Além disso, realizaremos uma simulação com perturbação no sistema para validar sua robustez.

Com o objetivo de comparar os ganhos resultantes de cada parâmetro otimizado, efetuamos o cálculo da variância entre o ganho obtido a partir da otimização da robustez e aquele proveniente do índice que considera o módulo do erro. Os resultados dessa comparação foram apresentados na Tabela 5.3, permitindo-nos avaliar como cada um desses parâmetros afeta os ganhos.

Exemplo	Δk_p	Δk_i	Δk_d	Robustez	IAE
1	0,0085	5,0717	1,8727	0,1536	1,0622
2	0,0569	0,0004	0,0008	0,0001	7,3368
3	1,9370	0,0007	0,0776	0,3148	2,2072

Tabela 5.3: Tabela de Variança dos Ganhos do PID - Comparação da Otimização de *IAE* com robustez

Após analisar a Tabela 5.3, é perceptível que no exemplo 5.1.2, a variação nos ganhos é baixa, o que era esperado devido ao comportamento semelhante no diagrama de *Nyquist*. No entanto, para os exemplos 5.1.1 e 5.1.3, há uma variação maior, especialmente no ganho integral e proporcional.

5.4.1 Resultado exemplo 1

A Figura 5.14 apresenta a comparação dos diagramas para o exemplo 5.1.1, sendo que o diagrama que enfatiza a robustez do sistema tangencia a referência M_s , enquanto que o diagrama otimizado para o IAE está próximo à circunferência e apresenta comportamento semelhante a outro diagrama.

A respeito da Figura 5.15, percebemos que o sistema estabilizado pelo critério do IAE é mais eficiente, apresentando um tempo de acomodação mais rápido, enquanto o sistema com ênfase na robustez tem uma resposta transitória mais lenta.

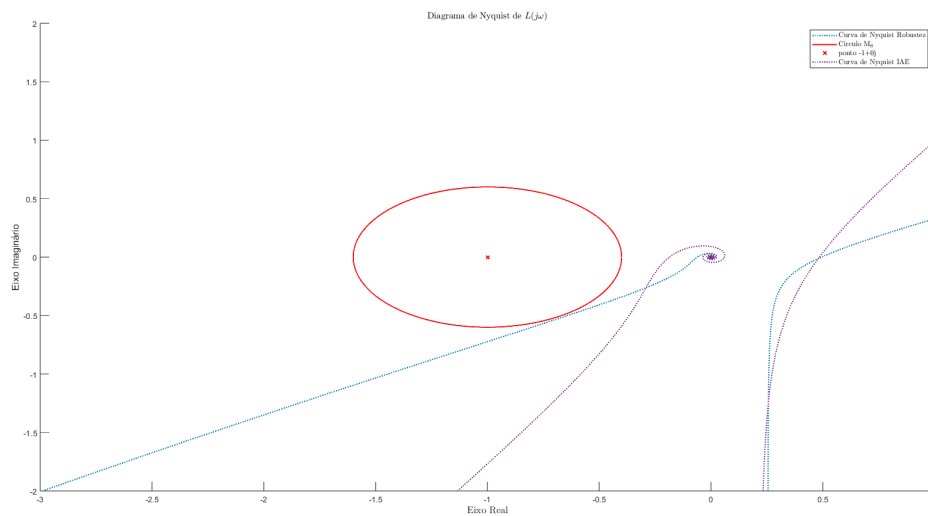


Figura 5.14: Comparação do diagrama de *Nyquist* do Exemplo 5.1.1

5.4.2 Resultado exemplo 2

No caso do exemplo 5.1.2, por se tratar de um sistema mais complexo e, teoricamente, com maior dificuldade na sintonia dos ganhos do controlador, a Figura 5.16 apresenta o diagrama de *Nyquist*. Neste caso, é possível observar que, para o índice considerado, o *Nyquist* intersecta a circunferência, enquanto que, para a busca pela robustez, isso não ocorre, como esperado. Além disso, conseguimos obter um comportamento semelhante no diagrama, circulando o ponto $0 + 0j$.

A simulação do sistema na Figura 5.17 apresenta o comportamento desejado quando otimizado com o índice de Integral do Erro Absoluto, sendo notável que o sistema é mais eficiente, enquanto que o sistema otimizado com ênfase na robustez tem uma resposta transitória mais lenta.

5.4.3 Resultado exemplo 3

No exemplo 5.1.3, podemos observar na Figura 5.18 que há uma maior distância do diagrama em relação a ambas as otimizações. A partir da simulação do sistema na F5.19,

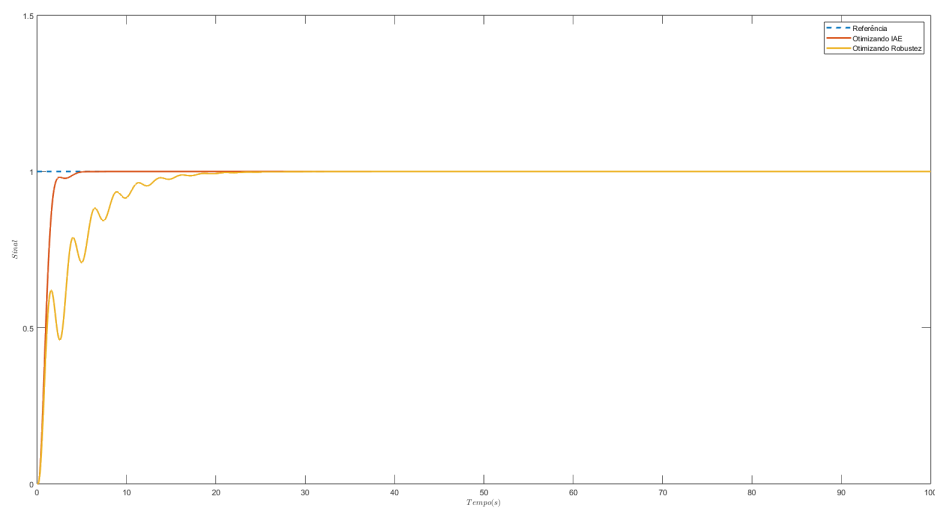


Figura 5.15: Comparação da simulação do Sistema do Exemplo 5.1.1

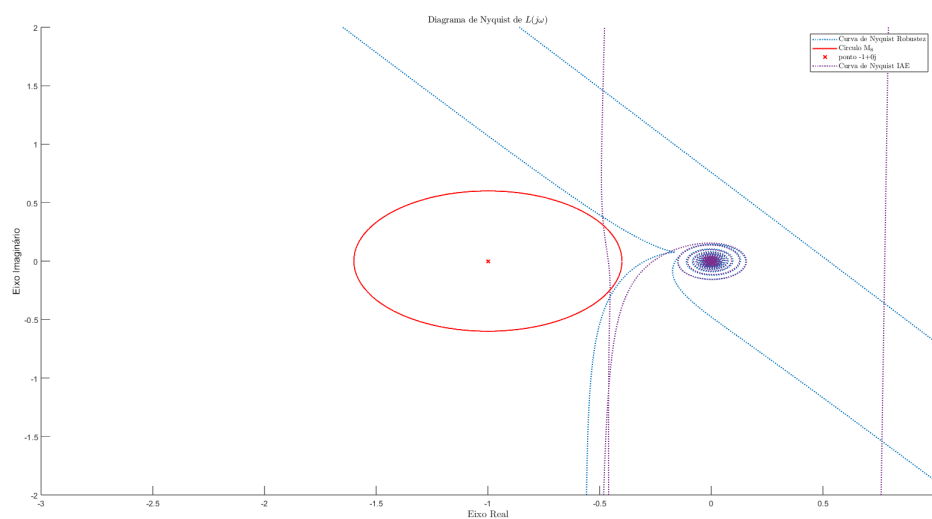


Figura 5.16: Comparação do diagrama de *Nyquist* do Exemplo 5.1.2

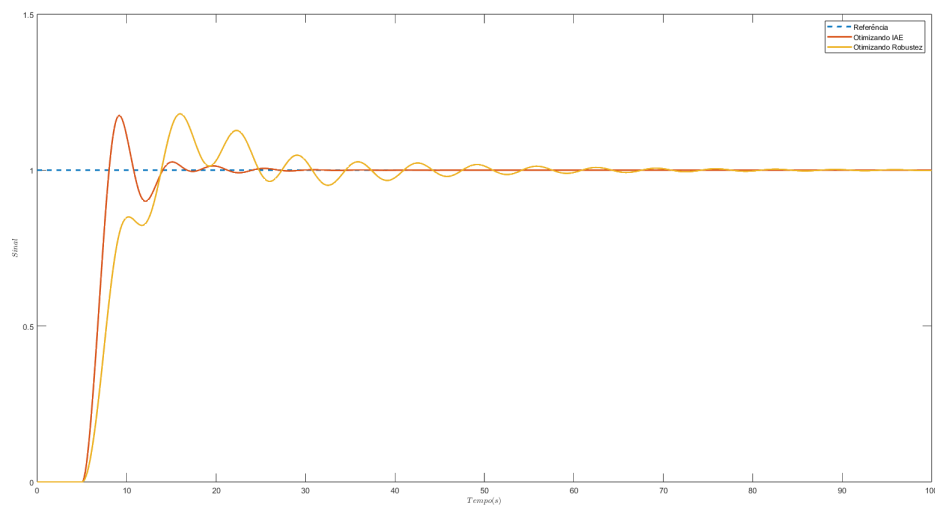


Figura 5.17: Comparação da simulação do Sistema do Exemplo 5.1.2

verificamos que a otimização do índice de *IAE* apresentou um tempo de acomodação menor em comparação com a otimização do critério de robustez.

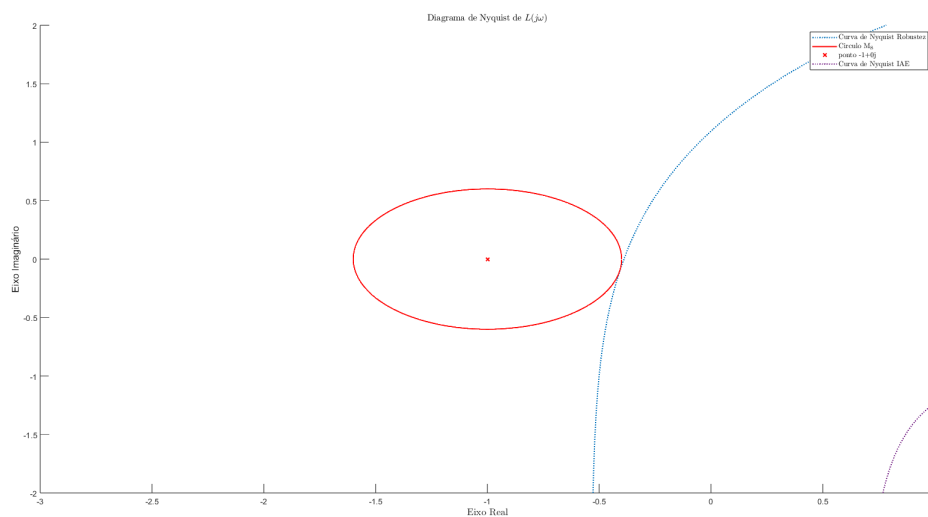


Figura 5.18: Comparação do diagrama de *Nyquist* do Exemplo 5.1.3

5.5 Resultados da Concorrência entre a Otimização

Na seção de resultados, iremos examinar os resultados conseguidos com a função de *fitness* definida na seção 4.6, que tenta considerar os dois parâmetros de busca. Nesse

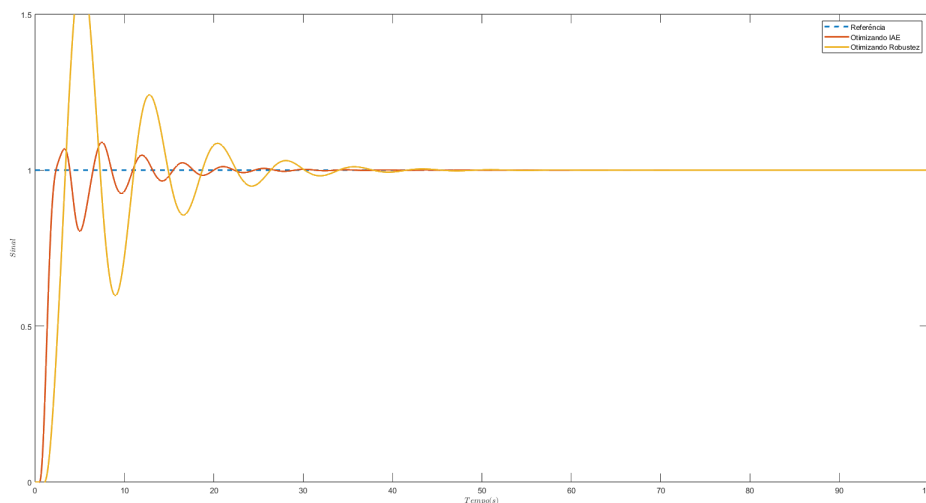


Figura 5.19: Comparação da simulação do Sistema do Exemplo 5.1.3

caso, estamos atribuindo a mesma importância à robustez e ao IAE , já que o valor do parâmetro α é 0,5.

Usando essa abordagem, é possível encontrar um equilíbrio que mantenha a robustez do sistema ao mesmo tempo que melhora o desempenho, reduzindo o tempo de acomodação. Os ganhos do controlador PID para esse método podem ser observados na Tabela 5.4.

Exemplo	k_p	k_i	k_d	Robustez	IAE	ff
1	2,9222	5,4647	3,6449	0,0073	0,9149	0,4611
2	0,5145	0,2578	0,4004	0,0041	7,3408	3,6725
3	0,4372	0,7548	1,8163	0,4856	2,4390	1,4623

Tabela 5.4: Tabela de Ganhos do PID com Método proposto - Comparação da Otimização de IAE com robustez

5.5.1 Resultado exemplo 1

A Figura 5.20 revela que, para o exemplo 5.1.1, o diagrama de Nyquist apresenta uma pequena distância da referência M_s , mantendo sinais de um sistema mais robusto, sem perder a prioridade da eficiência. Essa situação difere da Figura 5.8, onde o diagrama tende a se afastar ainda mais do raio de robustez estabelecido.

Analisando a Figura 5.21, percebe-se que a combinação dos parâmetros resulta em uma melhoria geral em relação à simulação em que a busca tentou otimizar apenas o índice IAE , como evidenciado na Figura 5.9, e também em comparação com a simulação do trabalho base DANTAS (2019).

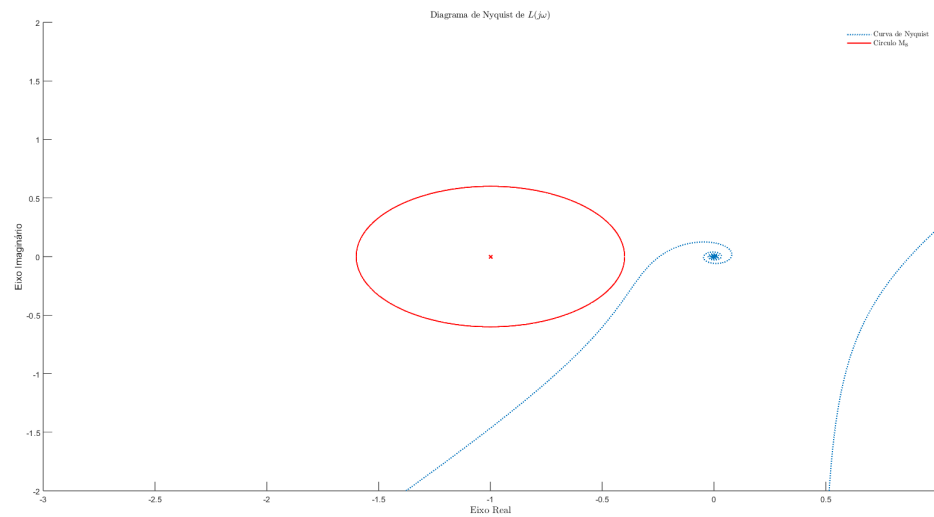


Figura 5.20: Diagrama de *Nyquist* com Método proposto do Exemplo 5.1.1

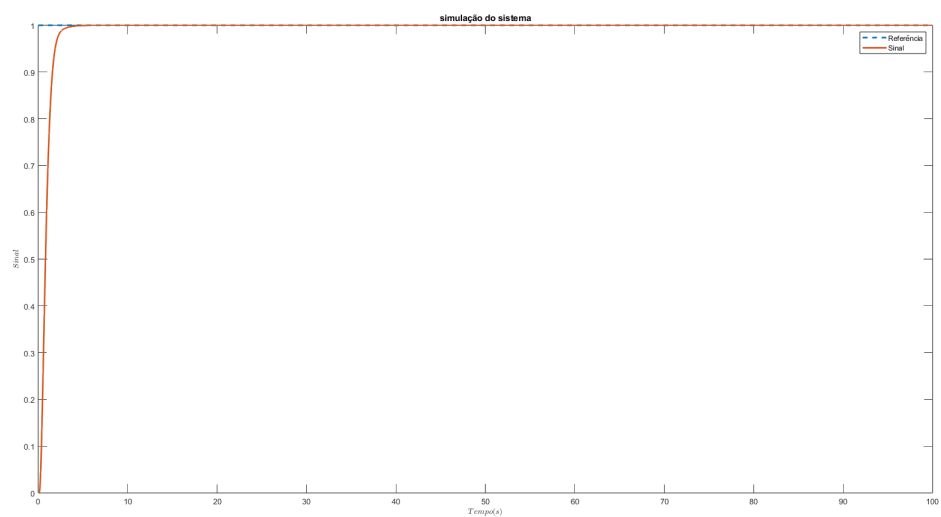


Figura 5.21: Simulação do Sistema com Método proposto do Exemplo 5.1.1

5.5.2 Resultado exemplo 2

No caso do exemplo 5.1.2, que apresenta maior complexidade, nota-se que o gráfico de Nyquist na Figura 5.22 está próximo à circunferência de robustez M_s , e os resultados da simulação mostram eficiência, conforme evidenciado na Figura 5.23. Esses resultados são superiores aos obtidos nas simulações em que é otimizado apenas um parâmetro na busca do algoritmo *GA*.

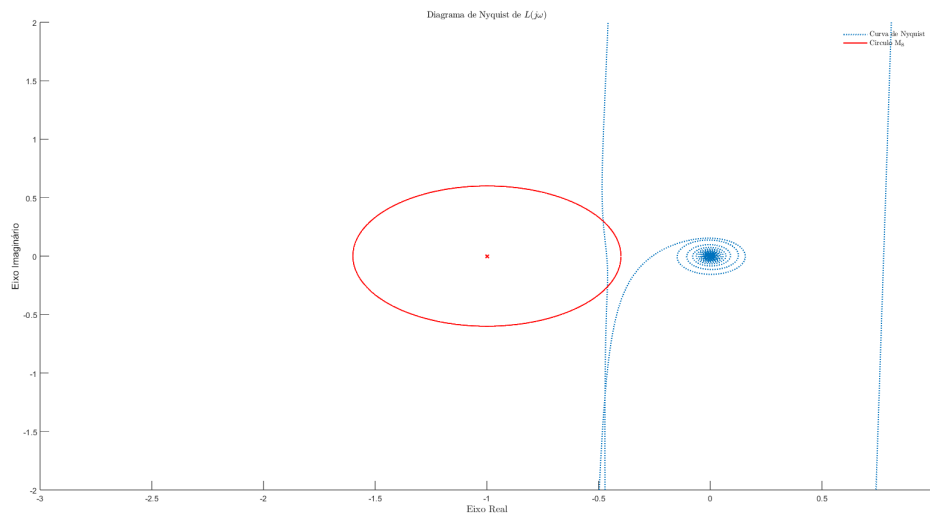


Figura 5.22: Diagrama de *Nyquist* com Método proposto do Exemplo 5.1.2

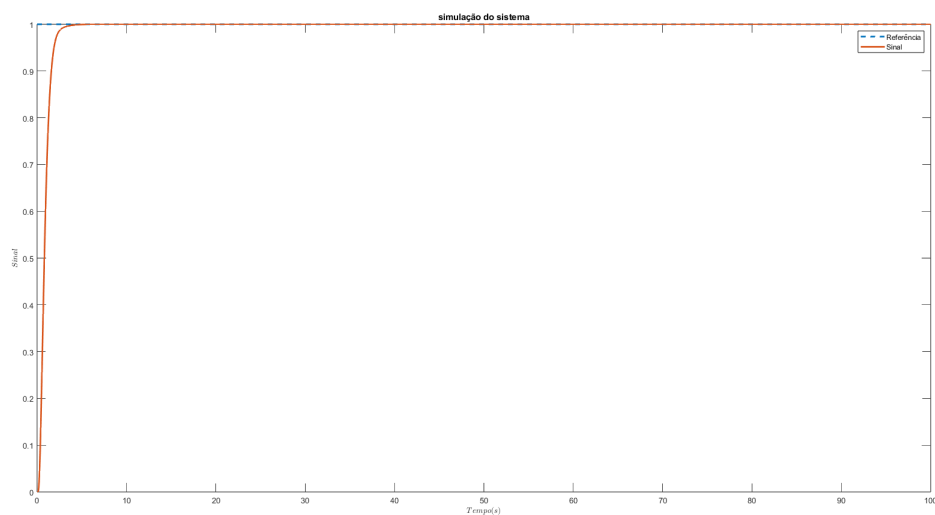


Figura 5.23: Simulação do Sistema com Método proposto do Exemplo 5.1.2

5.5.3 Resultado exemplo 3

Nesta variação do exemplo 5.1.2, introduzimos uma perturbação no sistema para simular um sistema não robusto. Na Figura 5.25, podemos observar que a resposta transitória do sistema é menor em comparação com as simulações realizadas nas seções anteriores, onde a otimização foi realizada em apenas um dos parâmetros. Na Figura 5.22, a curva de Nyquist não se afasta muito da circunferência, o que indica que o sistema de controle continua eficiente em termos de robustez e tempo de acomodação, que são importantes para bons resultados em sistemas de controle OGATA (2009).

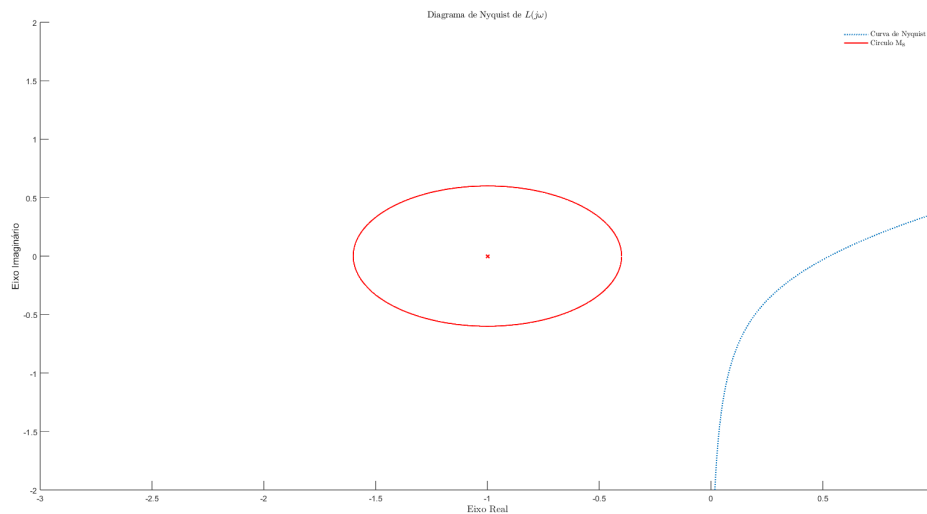


Figura 5.24: Diagrama de *Nyquist* com Método proposto do Exemplo 5.1.3

5.6 Resultados da perturbação no sistema

Nesta seção, simularemos uma perturbação do sistema, com objetivo de verificar quão os sistemas são robustos para os casos da otimização do *IAE* apresentado na seção 5.3, e para o método proposto neste trabalho.

Para realizar a simulação da perturbação, analisaremos o sistema do exemplo 5.1.2, com ganhos obtidos da simulação para este exemplo, executaremos a simulação do exemplo 5.1.3, o qual representa uma perturbação no sistema, devido à variação da Matriz *C*, de um exemplo para outro.

5.6.1 Perturbação para Otimização de *IAE*

Na Figura 5.26, o gráfico de *Nyquist*, não apresenta variação significativa se comparado com a Figura 5.12. Ao observar a Figura 5.27 que compara as duas simulações, uma sem a perturbação e outro com perturbação, podemos inferir que o sistema com a perturbação, tivemos um maior *Overshoot*, além de tempo de subida maior, como tempo

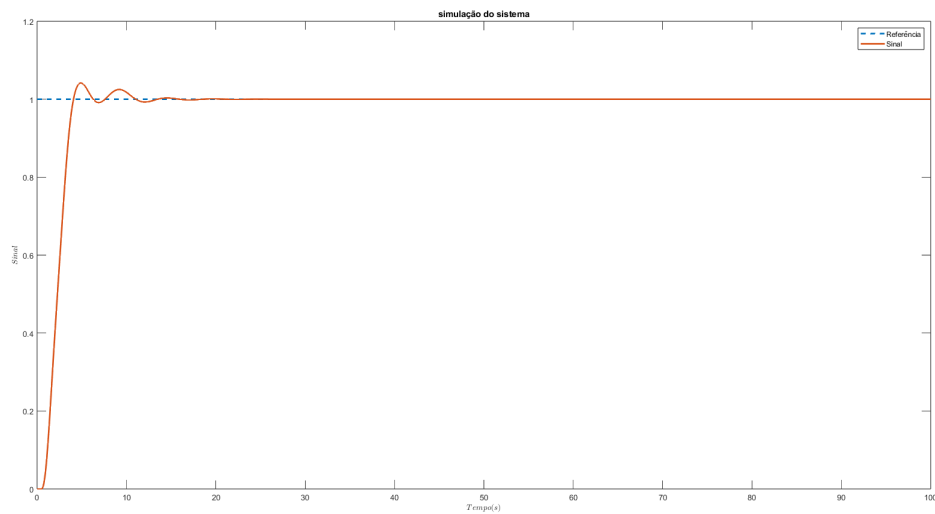


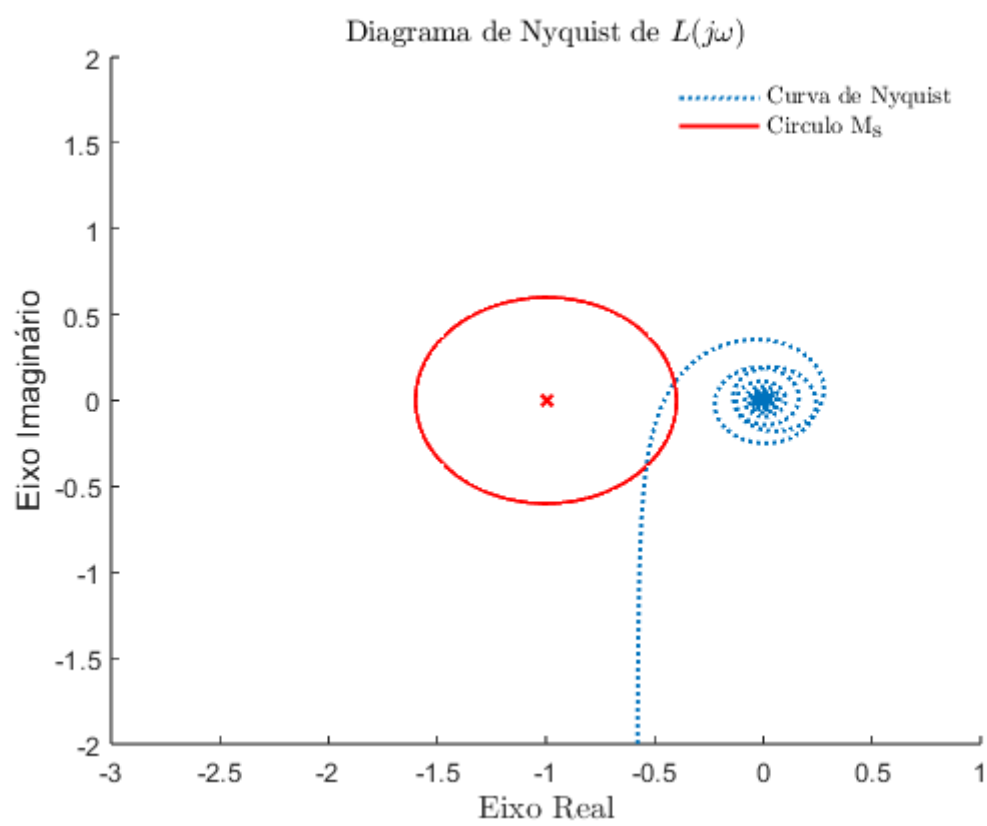
Figura 5.25: Simulação do Sistema com Método proposto do Exemplo 5.1.3

de primeiro pico, e oscilação maior do sistema em regime transitório. Mas percebe-se que o sistema permanece estável em malha fechada, evidenciado pelo Gráfico 5.28, os qual todos os polos estão no SPD.

5.6.2 Perturbação para Concorrência entre a Otimização

Para o método proposto neste trabalho, observamos um comportamento semelhante, conforme a Figura 5.29, apresenta, com uma acentuação dos resultados apresentados na seção 5.6.1, principalmente para o *Overshoot*, como apresentado na Figura 5.30.

O sistema de segunda ordem, com atraso, permanece estável em malha fechada conforme a Figura 5.31, confirma. Isso evidencia que a robustez é atenuada, para esses métodos, mais sem perder a estabilidade e robustez, além, percebemos que a perda da robustez é significativamente pequena, em relação aos ganhos de desempenho, adquiridos com introdução da busca com concorrência.

Figura 5.26: Diagrama de *Nyquist* com perturbação no sistema

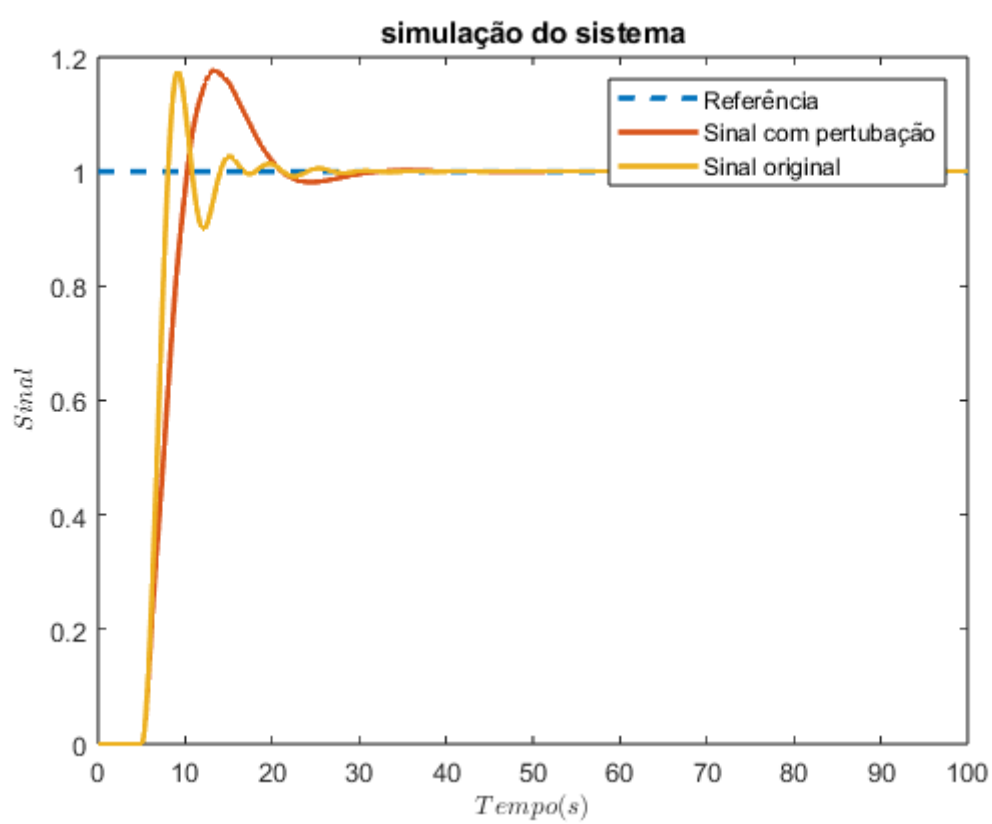


Figura 5.27: Simulação do Sistema com perturbação no sistema

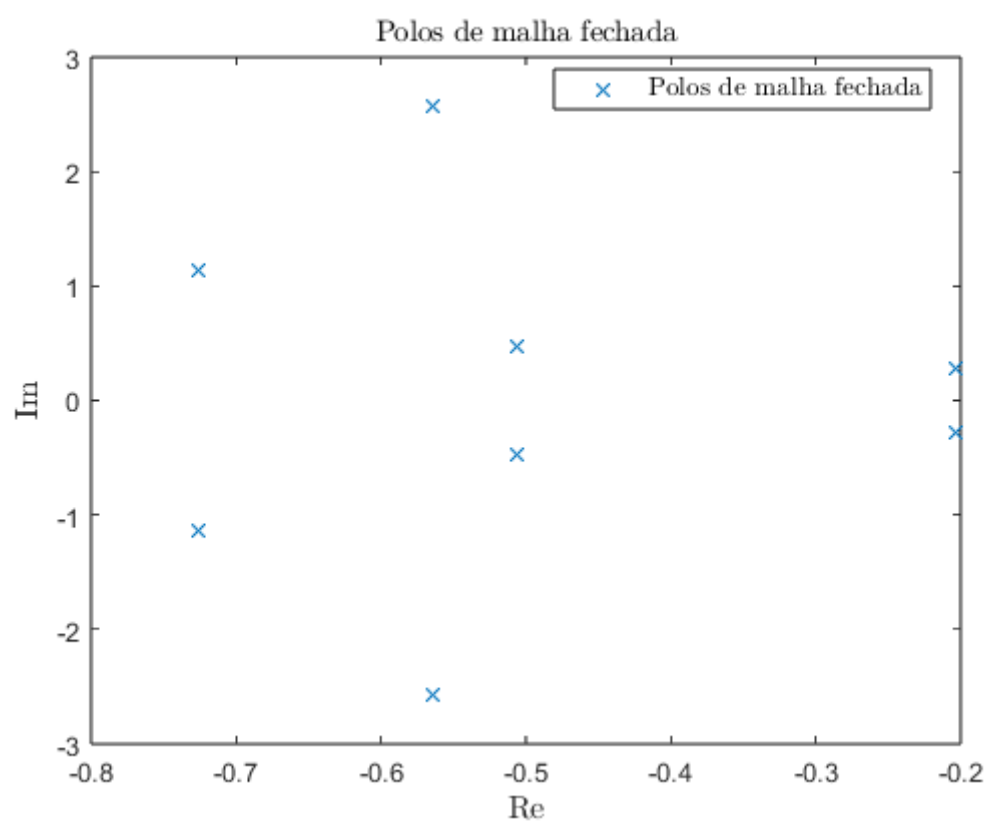
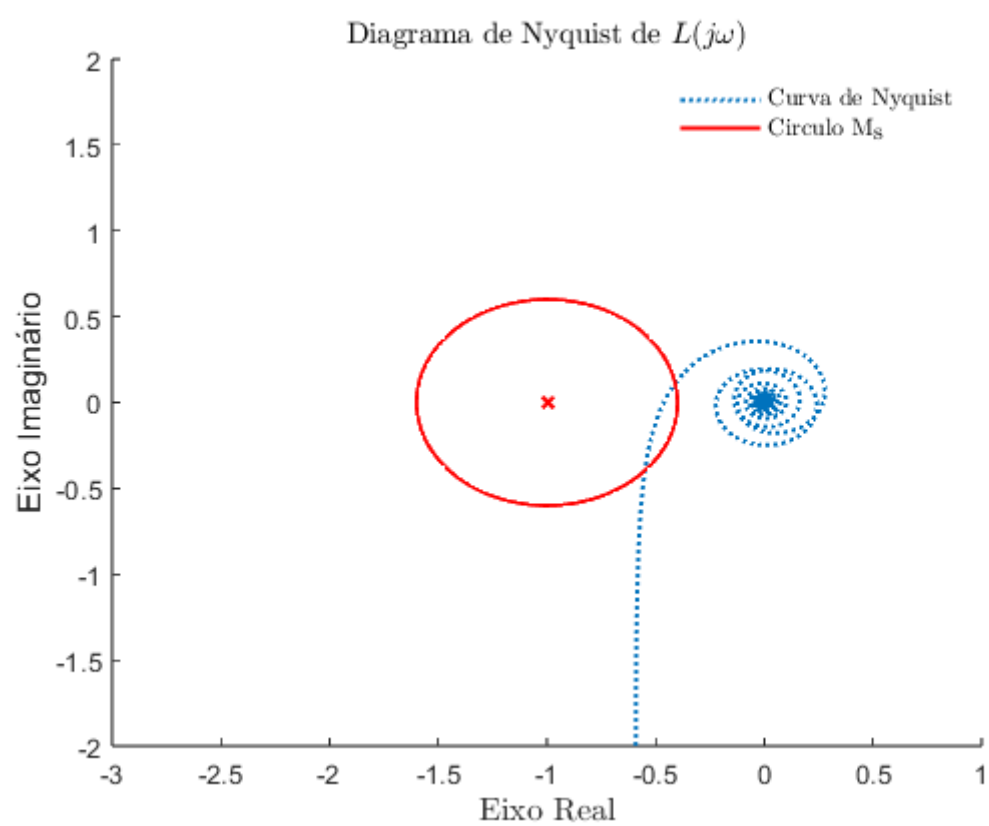


Figura 5.28: polos do Sistema com perturbação no sistema

Figura 5.29: Diagrama de *Nyquist* com perturbação no sistema

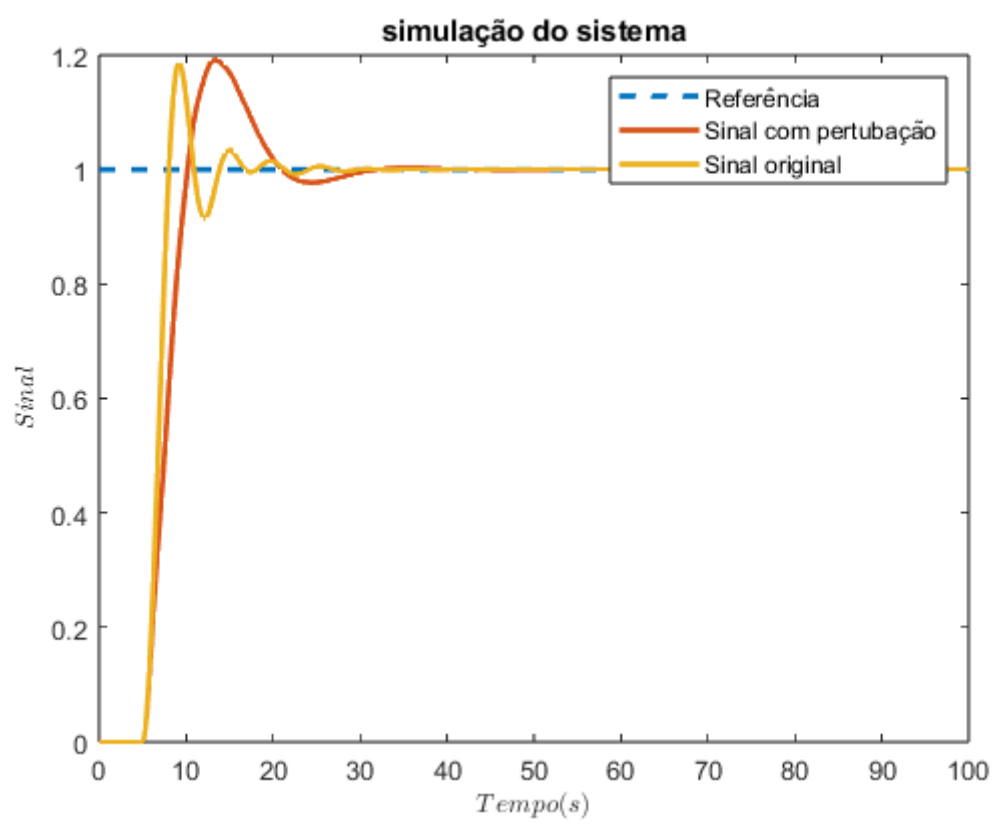


Figura 5.30: Simulação do Sistema com perturbação no sistema

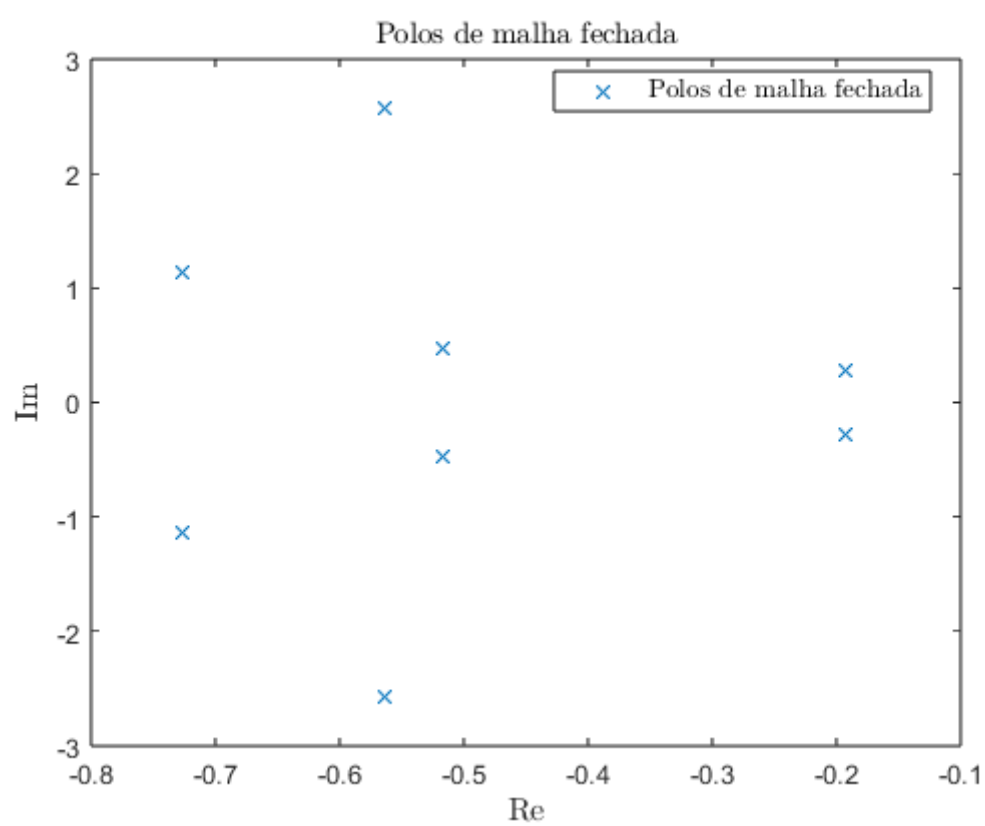


Figura 5.31: polos do Sistema com perturbação no sistema

Capítulo 6

Conclusão

Neste trabalho, foram propostas soluções para resolver o problema de controle de sistemas de segunda ordem com atraso por meio do uso de um Controlador PID. Uma alternativa interessante para lidar com sistemas mais complexos é a utilização da representação experimental dos sistemas de segunda ordem com atraso por meio de uma matriz de receptância. Para obter os vetores de ganho do controlador PID, o problema de controle foi expresso como um problema de otimização matemática. O objetivo foi encontrar ganhos que atendam de forma satisfatória tanto ao critério de projeto estabelecido quanto à robustez do sistema e ao índice de *IAE*, resultando em uma solução ótima que obtenha um sistema minimamente robusto e eficiente em termos de tempo de acomodação.

Além disso, constatou-se que a robustez e a otimização do índice do *IAE* apresentam comportamentos que competem entre si. Ou seja, não é possível otimizar a Integral do Erro Absoluto sem prejudicar a robustez do sistema. Por isso, foi necessário desenvolver um método que visasse melhorar ambos os aspectos de forma equilibrada.

Os resultados apresentados no capítulo 5 mostram que o método que busca otimizar ambos os parâmetros apresenta resultados mais satisfatórios do que a busca de algoritmos com apenas um parâmetro. Além disso, foi possível obter resultados superiores em relação a trabalhos anteriores deste mesmo grupo, como os estudos realizados por Dantas DANTAS (2019), e NUNES (2022), com melhor tempo de acomodação e *Overshoot*.

Existem outras propostas de interesse acadêmico que podem ser abordadas em trabalhos futuros, como a aplicação deste método em sistemas de controle instáveis para avaliar seu desempenho, bem como a aplicação em sistemas multivariáveis com atraso. Além disso, o uso de outras meta-heurísticas, como o Algoritmo de Enxame de Partículas (PSO), pode ser explorado para buscar resultados diferentes e possivelmente melhorias no controle desses sistemas.

Referências Bibliográficas

- ABDELAZIZ, THS e M VALÁŠEK (2004), ‘Pole-placement for siso linear systems by state-derivative feedback’, *IEE Proceedings-Control Theory and Applications* **151**(4), 377–385.
- ARAÚJO, José Mário (2018a), ‘Discussion on ‘state feedback control with time delay’’, *Mechanical Systems and Signal Processing* **98**, 368–370.
- ARAÚJO, José Mário (2018b), ‘Partial eigenvalue assignment in linear time-invariant systems using state-derivative feedback and a left eigenvectors parametrization’, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* p. 0959651818811010.
- ARAÚJO, José Mário e Tito Luís Maia SANTOS (2018), ‘Control of a class of second-order linear vibrating systems with time-delay: Smith predictor approach’, *Mechanical Systems and Signal Processing* **108**, 173–187.
- Astrom, Karl J (1995), ‘Pid controllers: theory, design, and tuning’, *The International Society of Measurement and Control* .
- BALAS, Mark (1982), ‘Trends in large space structure control theory: fondest hopes, wildest dreams’, *IEEE Transactions on Automatic Control* **27**(3), 522–535.
- DANTAS, N. J. B. (2019), Projeto de controladores para sistemas de segunda ordem com atraso via resposta em frequência., Dissertação (mestrado) — mestrado em engenharia mecânica, Universidade Federal do Rio Grande do Norte, Natal.
- DANTAS, Nelson J.B., Carlos E.T. Dórea e José M. Araújo (2020), ‘Design of rank-one modification feedback controllers for second-order systems with time delay using frequency response methods’, *Mechanical Systems and Signal Processing* **137**, 106404. Special issue on control of second-order vibrating systems with time delay.
URL: <https://www.sciencedirect.com/science/article/pii/S0888327019306259>
- DATTA, Biswa (2004), ‘Numerical methods for linear control systems’, **1**.
- EUZÉBIO, Thiago AM e Péricles R BARROS (2015), ‘Iterative procedure for tuning decentralized pid controllers’, *IFAC-PapersOnLine* **48**(8), 1180–1185.
- EUZÉBIO, Thiago Antonio Melo e Péricles Rezende BARROS (2013), ‘Optimal integral gain for smooth pi control’, *IFAC Proceedings Volumes* **46**(11), 529–533.

- FRANKLIN, Gene F, J David Powell e Abbas Emami-Naeini (2013), *Sistemas de controle para engenharia*, Bookman Editora.
- GOLDBERG, David E e John H Holland (1988), ‘Genetic algorithms and machine learning’, *Machine learning* **3**(2), 95–99.
- GOLUB, Gene H e Charles F VAN LOAN (2012), *Matrix computations*, Vol. 3, JHU Press.
- GONTIJO, Danielle S, José M Araújo, Tito LM SANTOS e Fernando O Souza (2022), ‘Proportional-integral-derivative-acceleration robust controllers for vibrating systems’, *Journal of Vibration and Control* p. 10775463211060898.
- MARTINS, José KEC, Fábio MU Araújo e Carlos ET Dórea (2020), Um método baseado em otimização para sintonia de controladores pi para sistemas sujeitos a restrições, em ‘Congresso Brasileiro de Automática-CBA’.
- MATHWORKS. (2019), ‘Genetic algorithm - matlab & simulink’.
URL: <https://www.mathworks.com/help/gads/genetic-algorithm.html>
- MOTTERSHEAD, John E e Yitshak M RAM (2007), ‘Receptance method in active vibration control’, *AIAA journal* **45**(3), 562–567.
- Nunes, Leonardo A, Nelson JB Dantas, Carlos ET Dórea e José M Araújo (2021), Controle por realimentação derivativa de estados de sistemas dinâmicos de segunda ordem com atraso, em ‘Simpósio Brasileiro de Automação Inteligente-SBAI’.
- NUNES, LEONARDO Araújo (2022), Controle por realimentação derivativa para sistemas dinâmicos de segunda ordem com atraso, Dissertação (mestrado) — mestrado em engenharia mecânica, Universidade Federal do Rio Grande do Norte, Natal.
- OGATA, Katsuhiko (2009), *Modern control engineering*, Prentice Hall Upper Saddle River, NJ.
- RAM, YM, Akshay Singh e John E MOTTERSHEAD (2009), ‘State feedback control with time delay’, *Mechanical Systems and Signal Processing* **23**(6), 1940–1945.
- RAM, YM, JE Mottershead e M Ghandchi TEHRANI (2011), ‘Partial pole placement with time delay in structures using the receptance and the system matrices’, *Linear Algebra and its Applications* **434**(7), 1689–1696.
- Rego, Everton JC, Carlos ET Dórea e Andre L Maitelli (2017), ‘Ressintonia automática de controladores pi embarcados em clp, baseada em estimativa de robustez’, *XIII Simpósio Brasileiro de Automação Inteligente* pp. 1733–1738.
- SANTOS, Tito LM, José Mário Araújo e Taniel S Franklin (2018), ‘Receptance-based stability criterion for second-order linear systems with time-varying delay’, *Mechanical Systems and Signal Processing* **110**, 428–441.

SKOGESTAD, Sigurd e Ian POSTLETHWAITE (2007), *Multivariable feedback control: analysis and design*, Vol. 2, Wiley New York.

SOUZA, J. H. A.; CARVALHO, E. J. F. (2020), ‘Implementação e avaliação de algoritmos genéticos e de enxame no ajuste de campos de força em dinâmica molecular’, *XXXI Congresso de Iniciação Científica e Tecnológica da UFRN - eCICT 2020* pp. 1604–1605.

TEHRANI, Maryam Ghandchi, Robin NR Elliott e John E Mottershead (2010), ‘Partial pole placement in structures by the method of receptances: theory and experiments’, *Journal of Sound and Vibration* **329**(24), 5017–5035.

VAN LOAN, Charles F e Gene H GOLUB (1983), *Matrix computations*, Johns Hopkins University Press.

VANDERVELDE, Wallace E (1986), ‘Control of large flexible space structures’.