

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

Jhonat Heberson Avelino de Souza ¹

Prof. Dr. Carlos Eduardo Trabuco Dórea ²

Prof. Dr. José Mário Araújo ³

¹jhonatheberson@gmail.com

²cetdorea@dca.ufrn.br

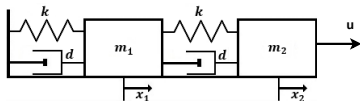
³prof.jomario@gmail.com

Departamento de Engenharia de Computação e Automação – DCA
Universidade Federal do Rio Grande do Norte – UFRN

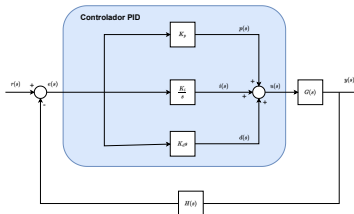
Sumário

- 1 Introdução
- 2 Fundamentação Teórica
- 3 Definição do Problema
- 4 Metodologia
- 5 Experimentos
- 6 Resultados
- 7 Conclusões
- 8 Referencias

Introdução - Contextualização



(a) Sistema massa e mola



(b) Controlador PID

Figura 1: Sistema de Segunda ordem, e Controlador PID

Introdução - Trabalhos anteriores do grupo

- Problema com realimentação de estado
- Alocação parcial de polos
- Monovariável
- Sistemas estáveis em malha aberta

Contribuímos com utilização do controlador Proporcional e Integrativo Derivativo (PID) com realimentação de saída para rastreamento e considerar sistemas estáveis em malha aberta, otimizando índice IAE e robustez de forma concorrente.

Introdução - Objetivos

- Estudo de sistemas de segunda ordem com atraso.
- Propor uma solução para um problema de controle para um sistema de segunda ordem com atraso
- Controlador Proporcional e Integrativo Derivativo (PID).
- Definir o problema de controle nos termos da resposta em frequência usando Receptância
- Critério de estabilidade de *Nyquist* para definir a função as regras de otimização.
- Otimização do controlador com base no índice de *IAE*, concorrente ao critério de robustez.
- Aprimorar algoritmo heurístico de otimização (GA) para encontrar os ganhos do controlador PID que atenda os critérios estabelecidos.

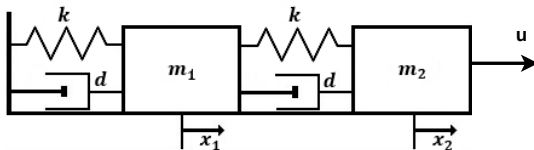


Figura 2: Sistema massa e mola

Equação do sistema

$$m_1 \ddot{x}_1(t) + d(2\dot{x}_1(t) - \dot{x}_2(t)) + k(2x_1(t) - x_2(t)) = 0 \quad (1)$$

$$m_2 \ddot{x}_2(t) + d(\dot{x}_2(t) - \dot{x}_1(t)) + k(x_2(t) - x_1(t)) = u \quad (2)$$

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} 2d & -d \\ -d & d \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (3)$$

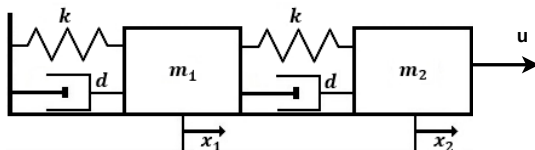


Figura 3: Sistema massa e mola

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{b}u(t) \quad (4)$$

Para um sistema com n graus de liberdade e m entradas, têm-se $\mathbf{M} \in \mathbb{R}^{n \times n}$, é uma matriz de massas, $\mathbf{C} \in \mathbb{R}^{n \times n}$ é uma matriz de amortecimento, $\mathbf{K} \in \mathbb{R}^{n \times n}$ é uma matriz de rigidez, $\mathbf{B} \in \mathbb{R}^{n \times m}$ é uma matriz de controle, $\mathbf{x} \in \mathbb{R}^n$ é o vetor de deslocamento e $\mathbf{u} \in \mathbb{R}^m$ é um vetor de entradas Single Input Single Output (SISO)

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{b}u(t) \quad (5)$$

Aplicando agora a transformada de Laplace a (5) obtém-se:

$$\mathbf{X}(s) = [\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K}]^{-1}\mathbf{b}\mathbf{U}(s) = \mathbf{H}(s)\mathbf{U}(s), \quad (6)$$

A saída do sistema definida como:

$$y(t) = \mathbf{l}\mathbf{x}(t) \quad (7)$$

em que $\mathbf{l} \in \mathbb{R}^{1 \times n}$ é uma matriz de composição de sensores.

Para uma dada referência $r(t)$, o erro de rastreamento é definido por:

$$e(t) = r(t) - y(t) \quad (8)$$

$$u(t) = k_p e(t - \tau) + k_i \int_0^t e(t - \tau) d\tau + k_d \frac{de(t - \tau)}{dt} \quad (9)$$

Fundamentação Teórica - Matriz de receptância

Aplicando a transformada de Laplace em (7), (8) e (9) obtém-se:

$$U(s) = -\left(k_p + \frac{k_i}{s} + k_d s\right)e^{-\tau s} \mathbf{I} \mathbf{X}(s) + V(s), \quad (10)$$

em que $q(s) = \left(k_p + \frac{k_i}{s} + k_d s\right)$ e $V(s) = q(s)R(s)$.

Da substituição de (10) em (5) resulta:

$$[\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + e^{-\tau s} q(s) \mathbf{b} \mathbf{l}] \mathbf{X}(s) = \mathbf{b} V(s). \quad (11)$$

Logo,

$$\mathbf{X}(s) = [\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + e^{-\tau s} q(s) \mathbf{b} \mathbf{l}]^{-1} \mathbf{b} V(s) = \hat{\mathbf{H}}(s) \mathbf{V}(s) \quad (12)$$

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \quad (13)$$

Aplicando a fórmula de *Sherman-Morrison* na inversa presente em (12), obtém-se a matriz de receptância de malha fechada do sistema, dada por:

$$\hat{\mathbf{H}}(s) = \mathbf{H}(s) - \frac{(k_p + \frac{k_i}{s} + k_d s)\mathbf{H}(s)\mathbf{b}\mathbf{I}\mathbf{H}(s)e^{-\tau s}}{1 + (k_p + \frac{k_i}{s} + k_d s)\mathbf{I}\mathbf{H}(s)\mathbf{b}e^{-\tau s}} \quad (14)$$

na qual $\hat{\mathbf{H}}(s) = (\mathbf{M}s^2 + \mathbf{C}s + \mathbf{K} + q(s)\mathbf{b}\mathbf{I})^{-1}$ 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, que, na prática, pode ser medida pela resposta em frequência $\mathbf{H}(j\omega)$.

A equação característica de (14) é definida como:

$$1 + (k_p + \frac{k_i}{s} + k_d s)\mathbf{I}\mathbf{H}(s)\mathbf{b}e^{-\tau s} = 0 \quad (15)$$

Resposta em frequência

- A matriz de receptância nos fornece a resposta em frequência $H(j\omega)$ do sistema
- O ganho de malha da equação característica em malha fechada é descrito na Equação 16.

$$L(s) = (k_p + \frac{k_i}{s} + k_d s) \mathbf{H}(s) B e^{-s\tau} \quad (16)$$

Definição do Problema - Estabilidade

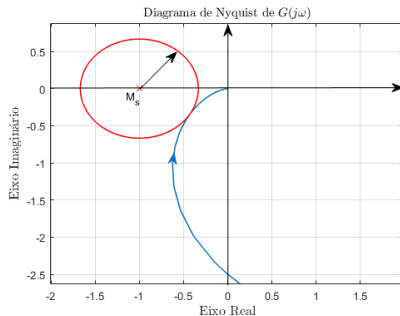


Figura 4: Diagrama de Nyquist Exemplo de Circunferência. M_s

Restrições de estabilidade

- Podemos afirmar pela teórica do critério de estabilidade de Nyquist que circunferência M_s representa quanto robusto é o sistema
- M_s é menor distância entre o ponto $(-1, 0)$ e a curva de Nyquist

Definição do Problema - Instabilidade

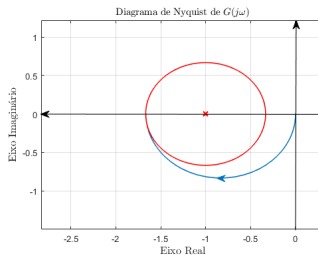


Figura 5: Diagrama de Nyquist de um Sistema Genérico para Exemplo de um Caso de Instabilidade.

Restrições de instabilidade

- Garantir que a curva de Nyquist não contenha a circunferência a englobando
- Enlaçar o ponto $(1, 0)$, se evidencia um caso de instabilidade

Definição do Problema - Funções de otimização

$$\begin{aligned} \min_{k_p, k_i, k_d} \quad & h(k_p, k_i, k_d) = \left(\min_{\omega_i} |L(j\omega) + 1| - M_s^{-1} \right)^2 \\ \text{s.a.} \quad & L(j\omega_i) = -(k_p + \frac{k_i}{j\omega_i} + k_d j\omega_i) / \mathbf{H}(j\omega_i) B e^{-j\omega_i \tau} \\ & \text{Re} \{L(j\omega_i)\} \geq -1 + M_s^{-1} \quad \forall \omega_i / \text{Im} \{L(j\omega_i)\} = 0 \end{aligned} \quad (17)$$

Definição do Problema - Funções de otimização

$$\min_{k_p, k_i, k_d} \int_0^t |e(t)| dt \quad (18)$$

$$\begin{aligned} s.a. \quad & h(k_p, k_i, k_d) = \left(\min_{\omega_i} |L(j\omega) + 1| - M_s^{-1} \right)^2 \leq \epsilon \\ & Z - N = 0 \end{aligned} \quad (19)$$

Função	Dimensão	PSO [3]	PSO	GA [3]	GA	GA [5]
Esfera	30	1.0454E+05 ±7.1998E+04	2.241E+03 ±7.030E+02	6.4415E+03 ±1.6876E+03	6.7564E+01 ±4.4760E+01	2.0532E+02 ± 4.6377E+01
Rosenbrock	2	7.0289E+08 ±4.8937E+08	7.0261E+08 ±3.666E+08	1.2493E+07 ±8.6725E+06	9.7689E+00 ±2.5590E+03	1.2280E+02 ±1.9833E+02
Rastrigin	30	5.4130E+02 ±1.5969E+01	1.7430E+02 ±2.506E+01	5.5900E+01 ±1.4294E+01	1.7650E+01 ±3.2255E+00	6.9160E+01 ± 9.5182E+00

Figura 6: Média e desvio-padrão do fitness de 20 execuções do algoritmo GA comparado ao PSO, nas condições da ref [3]. Resultados do algoritmo deste trabalho em negrito

Busca dos ganhos do controlador

- Uso de meta-heurística para encontrar os ganhos
- Algoritmo Genético
- Resolve de forma genérica problemas de otimização
- Geralmente aplicadas a problemas para os quais não se conhece algoritmo eficiente

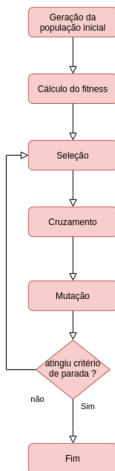


Figura 7: Fluxograma do algoritmo genético

Algoritmo 1: Função de Busca

```
Entrada: M, C, K, B, l,  $\tau$ ,  $\alpha$ ,  $M_f$  e simulink
Saída: gain = [Kp Ki Kd];
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         avaliacao = []
11         se execucao > 0 então
12             populacao(2:n,:) = random(n-1);
13         fim
14         para i = 1 até populacao=n faça
15             L ← LFunction(M, C, K, B, l,  $\tau$ ,  $\alpha$ , populacao(i, 1:3));
16             robustez ← RobustezFunction(L(s),  $M_f$ );
17             restrictions ← RestrictionsFunction(L(s), 0);
18             IAE ← IaeFunction(populacao, simulink);
19             ff ← FitnessFunction(IAE, robustez, alfa);
20             avaliacao(i,1:end) = [ff, restrictions];
21             i = i + 1;
22         fim
23         populacao = [populacao avaliacao];
24         populacao ← AssortmentFunction(populacao);
25         populacao ← CrossOverFunction(populacao(1:n,1:end-2));
26         L ← LFunction(M, C, K, B, l,  $\tau$ ,  $\alpha$ , populacao(1, 1:3));
27         robustez ← RobustezFunction(L(s),  $M_f$ );
28         restrictions ← RestrictionsFunction(L(s), 0);
29         IAE ← IaeFunction(populacao, simulink);
30         ff ← 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     execucao = execucao + 1;
41 até (objetivo ≠ "fim");
```

Figura 8: Pseudo código do algoritmo de busca

Algoritmo 3: Função L

Entrada: M ; C ; K ; B ; I ; 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    $L(i) = (K_p + K_i/\beta + 0.0001) + \beta \times K_d) \times I \times (M \times \beta^2 + C \times \beta + K)^{-1} \times B \times e^{-\tau \times \beta};$   
7    $i = i + 1;$   
8 fim
```

Figura 9: Pseudo código da função L

Nyquist infinito

- Perturbação na ação integrativa
- Evita valores infinitos para cálculo do diagrama de *Nyquist*

Metodologia - Algoritmo verificando a variabilidade genética

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
```

Figura 10: Pseudo código da função de variabilidade

Verificação de variabilidade genética

- Análise de melhoria da população com base na interação do algoritmo

Metodologia - Métodos de seleção

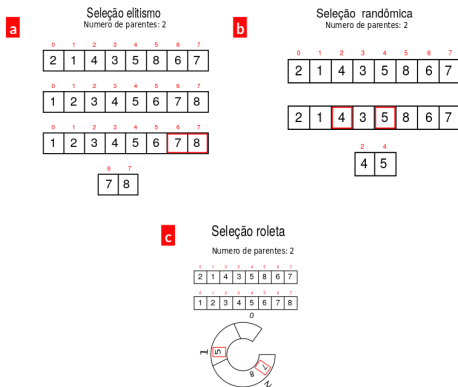


Figura 11: Ilustração dos métodos implementados de seleção. (a) Seleção elitismo, ordenando de forma crescente e escolhendo os indivíduos com maior fitness. (b) Seleção Randômica, que seleciona os pais aleatoriamente entre a população. (c) Seleção roleta, a qual ordena e rearranja para que simula uma roleta.

Metodologia - Métodos de cruzamento

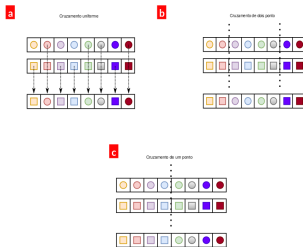


Figura 12: Algoritmos de cruzamento (crossover). (a) Cruzamento uniforme, em que o novo cromossomo (abaixo) é formado selecionando genes aleatórios de cada um dos pais. (b) Cruzamento de dois pontos, em que o novo cromossomo é formado pelas extremidades de um dos pais e a parte central do outro. Os pontos de corte (tracejados) são sorteados aleatoriamente para cada indivíduo. (c) Cruzamento de um ponto, em que o novo indivíduo é gerado com o início do cromossomo de um dos pais e o final do outro. O ponto de corte (linha tracejada) é decidido aleatoriamente para cada indivíduo.

Experimentos - Exemplo 1

Na Figura 3, 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 ram2009state.

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; POSTLETHWAITE, 2007).

Experimentos - Exemplo 2

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & -0,9999 \\ -0,9999 & 1 \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$$

Considerando o método apresentado em ram2011partial, vamos utilizar como exemplo prático uma matriz de entradas $\mathbf{B}' = \mathbf{I}' = [0 \ 1]$ e uma constante de atraso $\tau = 5$.

Novamente a busca pela solução do problema descrito pela equação (17) é realizada para o círculo $M_s = 1,6667$.

Experimentos - Exemplo 3

Neste caso específico, foi feita uma pequena modificação do exemplo ??, 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 $\tau = 0,5$.

Experimentos - Exemplo 4

Este exemplo, adaptado de shapiro2005stability, aborda um caso instável em malha aberta, cujas matrizes são dadas como:

$$\mathbf{M} = \begin{bmatrix} 10 & 0 \\ 0 & 11 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 4 & 1 \\ 1 & 5 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 8 & 4 \\ -4 & 9 \end{bmatrix}$$

com $\mathbf{B}' = [-1 \ 1]$ e $\mathbf{I} = [1 \ 0]$ além de um atraso $\tau = 0.5$. Os polos de malha aberta estão resumidos na tabela X.

s_1	$-0.4312 + j0.8953$
s_2	$-0.4312 - j0.8953$
s_3	$0.0039 + j0.9001$
s_4	$0.0039 - j0.9001$

Tabela 1: Polos de malha aberta Exemplo 4.

Component CommunityInstaller.AutoStartAction failed: O acesso à chave do Registro

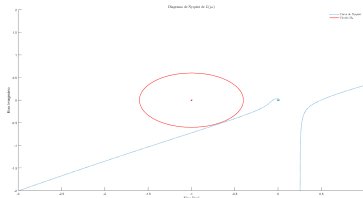
Resultados - Resultados da Otimização de Robustez

Os valores dos vetores de ganho descobertos utilizando o algoritmo genético (método proposto) com atraso estão apresentados na Tabela 2.

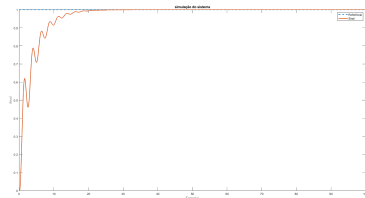
Exemplo	kp	ki	kd	robustez
1	1,4160	1,5225	0,8718	0,1536
2	0,1774	0,2229	0,4433	0,0001
3	-0,2400	1,0990	1,6015	0,3148

Tabela 2: Tabela de Ganhos do PID - Otimização de Robustez

Resultados - Resultados da Otimização de Robustez



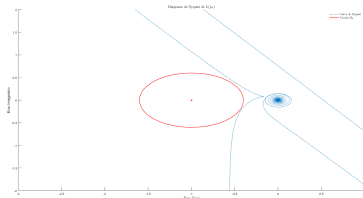
(a) Diagrama de Nyquist Exemplo 1



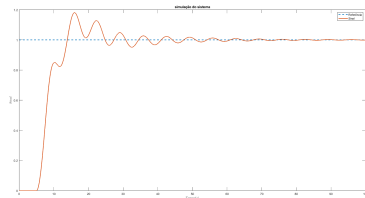
(b) Simulação do Sistema do Exemplo 1

Figura 13: Resultado exemplo 1

Resultados - Resultados da Otimização de Robustez



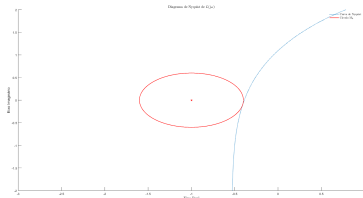
(a) Diagrama de Nyquist Exemplo 2



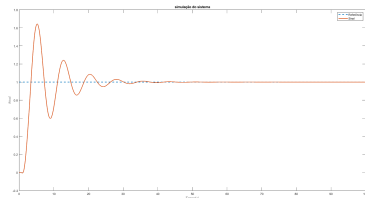
(b) Simulação do Sistema do Exemplo 2

Figura 14: Resultado exemplo 2

Resultados - Resultados da Otimização de Robustez



(a) Diagrama de Nyquist Exemplo 3



(b) Simulação do Sistema do Exemplo 3

Figura 15: Resultado exemplo 3

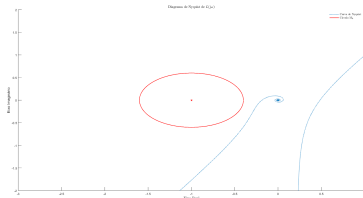
Resultados - Resultados da Otimização de IAE

Os resultados da otimização com o índice Integral do Erro Absoluto são apresentados neste trecho, em comparação aos resultados teóricos discutidos na página 29. 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 3 fornece os valores dos ganhos obtidos pelo algoritmo genético.

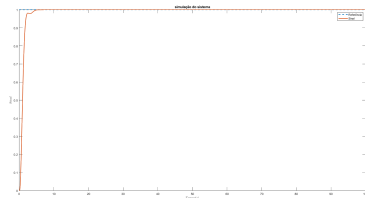
Exemplo	kp	ki	kd	IAE
1	1,5467	4,7074	2,8071	1,0622
2	0,5146	0,2539	0,4016	7,3368
3	1,7283	1,0606	1,9955	2,2072

Tabela 3: Tabela de Ganhos do PID - Otimização de IAE

Resultados - Resultados da Otimização de IAE



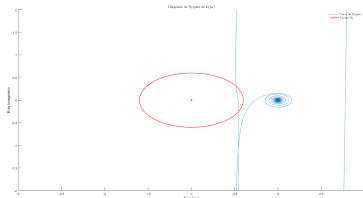
(a) Diagrama de Nyquist Exemplo 1



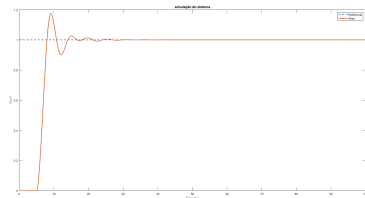
(b) Simulação do Sistema do Exemplo 1

Figura 16: Resultado exemplo 1

Resultados - Resultados da Otimização de IAE



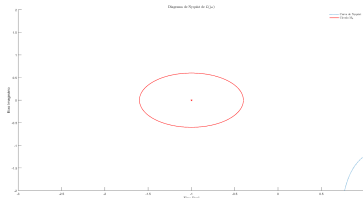
(a) Diagrama de Nyquist Exemplo 2



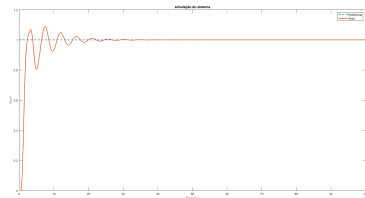
(b) Simulação do Sistema do Exemplo 2

Figura 17: Resultado exemplo 2

Resultados - Resultados da Otimização de IAE



(a) Diagrama de Nyquist Exemplo 3



(b) Simulação do Sistema do Exemplo 3

Figura 18: Resultado exemplo 3

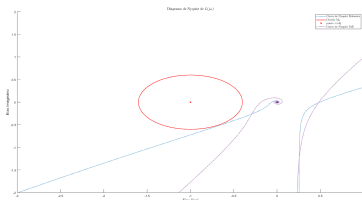
Resultados - Resultados da Comparação entre as Otimização

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

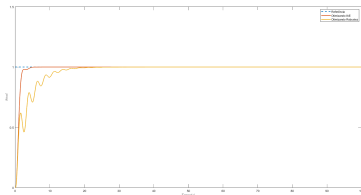
Tabela 4: Tabela de Variança dos Ganhos do PID - Comparação da Otimização de *IAE* com robustez

Após analisar a Tabela 4, é perceptível que no exemplo 2, a variação nos ganhos é baixa, o que era esperado devido ao comportamento semelhante no diagrama de *Nyquist*. No entanto, para os exemplos 1 e 3, há uma variação maior, especialmente no ganho integral e proporcional.

Resultados - Resultados da Comparação entre as Otimização



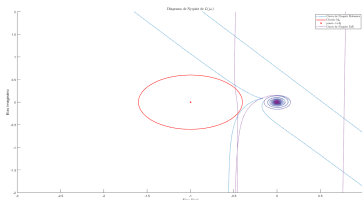
(a) Comparação do diagrama de *Nyquist* do Exemplo 1



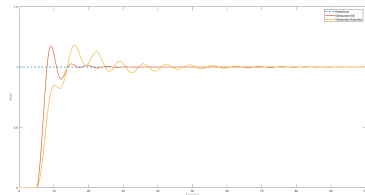
(b) Comparação da simulação do Sistema do Exemplo 1

Figura 19: Resultado exemplo 1

Resultados - Resultados da Comparação entre as Otimização



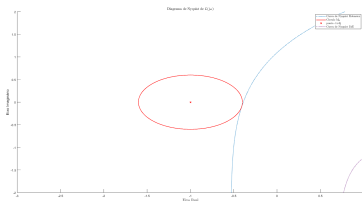
(a) Comparação do diagrama de *Nyquist* do Exemplo 2



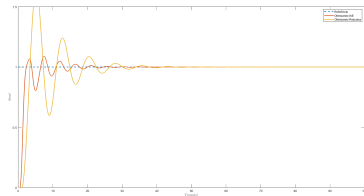
(b) Comparação da simulação do Sistema do Exemplo 2

Figura 20: Resultado exemplo 2

Resultados - Resultados da Comparação entre as Otimização



(a) Comparação do diagrama de *Nyquist* do Exemplo 3



(b) Comparação da simulação do Sistema do Exemplo 3

Figura 21: Resultado exemplo 3

Resultados - Resultados da Concorrência entre a Otimização

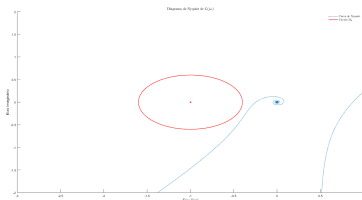
Na seção de resultados, iremos examinar os resultados conseguidos com a função de *fitness* de concorrência, que tenta considerar os dois parâmetros de busca. Nesse caso, estamos atribuindo a mesma importância à robustez e ao *IAE*, já que o valor do parâmetro alfa é 0,5.

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.

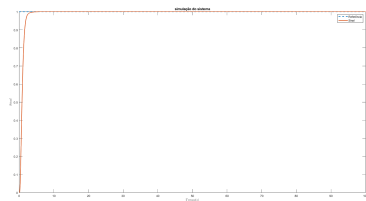
Exemplo	kp	ki	kd	Robustez	IAE	ff
1	2.9222	5,4647	3,6449	0,0073	0,9149	0,4611
2	0,5145	0,2578	0,4004	0,0041	7,3408	3,6725
3	0,4372	0,7548	1,8163	0,4856	2,4390	1,4623

Tabela 5: Tabela de Ganhos do PID com Método proposto - Comparação da Otimização de *IAE* com robustez

Resultados - Resultados da Concorrência entre a Otimização



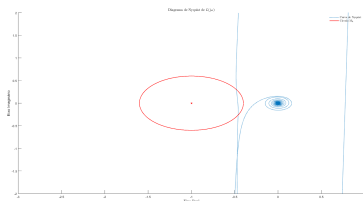
(a) Diagrama de *Nyquist* com Método proposto do Exemplo 1



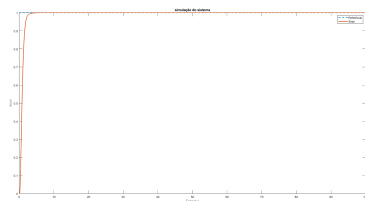
(b) Simulação do Sistema com Método proposto do Exemplo 1

Figura 22: Resultado exemplo 1

Resultados - Resultados da Concorrência entre a Otimização



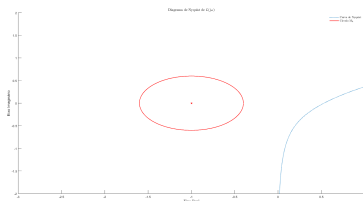
(a) Diagrama de *Nyquist* com Método proposto do Exemplo 2



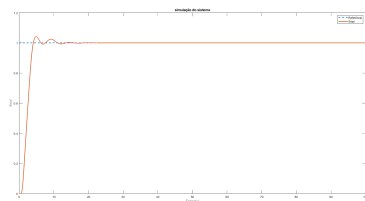
(b) Simulação do Sistema com Método proposto do Exemplo 2

Figura 23: Resultado exemplo 2

Resultados - Resultados da Concorrência entre a Otimização



(a) Diagrama de *Nyquist* com Método proposto do Exemplo 3



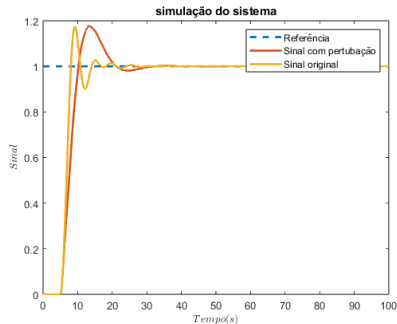
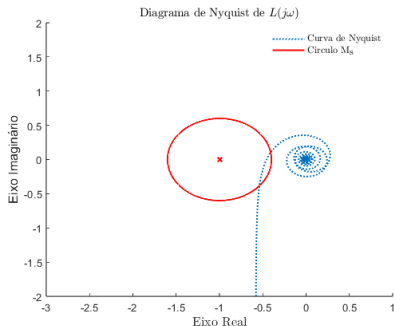
(b) Simulação do Sistema com Método proposto do Exemplo 3

Figura 24: Resultado exemplo 3

Resultados - 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 página 33, e para o método proposto neste trabalho. Para realizar a simulação da perturbação, analisaremos o sistema do exemplo 2, com ganhos obtidos da simulação para este exemplo, executaremos a simulação do exemplo 3, o qual representa uma perturbação no sistema, devido à variação da Matriz **C**, de um exemplo para outro.

Resultados - Resultados da perturbação no sistema



(a) Diagrama de *Nyquist* com perturbação no sistema (b) Simulação do Sistema com perturbação no sistema

Figura 25: Resultado otimizando índice *IAE*

Resultados - Resultados da perturbação no sistema

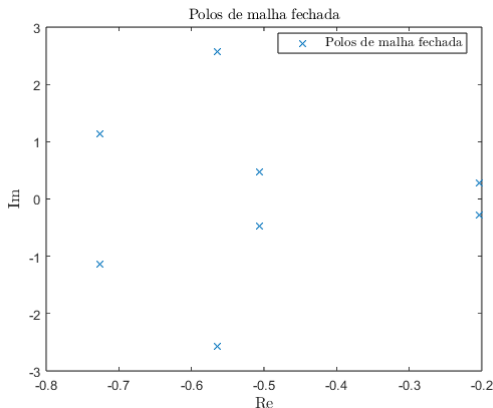
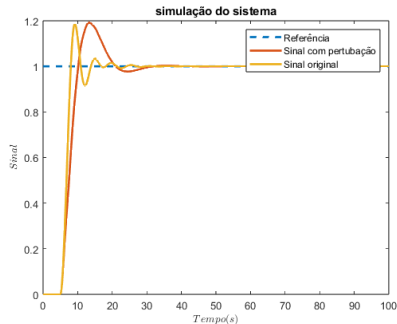
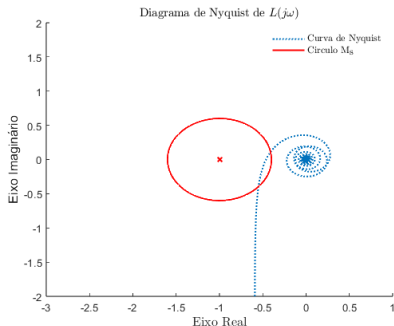


Figura 26: polos do Sistema com perturbação no sistema otimizando índice IAE

Resultados - Resultados da perturbação no sistema



(a) Diagrama de *Nyquist* com perturbação no sistema
(b) Simulação do Sistema com perturbação no sistema

Figura 27: Resultado otimizando concorrência entre *IAE* e robustez

Resultados - Resultados da perturbação no sistema

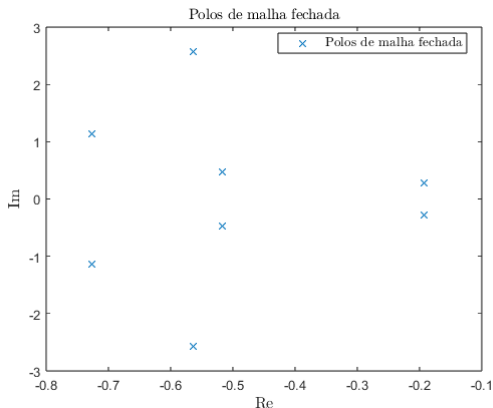



Figura 28: Polos do sistema com perturbação no sistema otimizando concorrência entre *IAE* e robustez

- PID com realimentação de saída para rastreamento.
- Estudo da eficiência para essa técnica utilizando controlador PID.
- Avaliação do critério de estabilidade de *Nyquist*.
- Avaliação do robustez do sistema como parametro de busca.
- Avaliação do índice *IAE* como parametro de busca.
- Estudo da concorrência entre a robustez do sistema e índice *IAE*.
- Avaliação da função *Fitness* desenvolvida que considera concorrência entre os parâmetros de busca.
- Trabalhos futuros: Verificar eficiência de outro algoritmo heurístico (PSO).

 SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable feedback control: analysis and design*. [S.l.]: Wiley New York, 2007. v. 2.

- [1] Ogata, Katsuhiko (2009), Modern control engineering, Prentice Hall Upper Saddle River, NJ.
- [2] DANTAS, N. J. B. (n.d.), Projeto de controladores para sistemas de segunda ordem com atraso via resposta em frequência., Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.
- [3] Mohd Nadhir Ab Wahab, Samia Nefti-Meziani, Adham Atyabi, A Comprehensive Review of Swarm Optimization Algorithms, PLOS ONE, 10, e0122827, 2015.
- [4] Solgi, R. M., geneticalgorithm, v 1.0.1, <https://github.com/rmsolgi/geneticalgorithm> Acessado em 24/08/2020.