

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CORNÉLIO PROCÓPIO  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
MESTRADO EM ENGENHARIA ELÉTRICA**

**DANIEL CORDEIRO JACINTO**

**CONTROLE DE POSIÇÃO UTILIZANDO ALGORITMO GENÉTICO COM  
MINIMIZAÇÃO DE ENTROPIA DO ERRO**

**DISSERTAÇÃO**

**CORNÉLIO PROCÓPIO**

**2018**

**DANIEL CORDEIRO JACINTO**

**CONTROLE DE POSIÇÃO UTILIZANDO ALGORITMO GENÉTICO COM  
MINIMIZAÇÃO DE ENTROPIA DO ERRO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Mestre em Engenharia Elétrica”.

Orientador: Prof. Dr. Paulo Rogério Scalassara

Coorientador: Prof. Dr. Cristiano Marcos Agulhari

**CORNÉLIO PROCÓPIO**

**2018**

---

### Dados Internacionais de Catalogação na Publicação

---

J121 Jacinto, Daniel Cordeiro

Controle de posição utilizando algoritmo genético com minimização de entropia do erro / Daniel Cordeiro Jacinto. – 2018.  
68 f. : il. color. ; 31 cm.

Orientador: Paulo Rogério Scalassara.  
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. Cornélio Procópio, 2018.  
Bibliografia: p. 58-60.

1. Entropia. 2. Erro. 3. Algoritmos genéticos. 4. Engenharia Elétrica – Dissertações. I. Scalassara, Paulo Rogério, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDD (22. ed.) 621.3

---

### Biblioteca da UTFPR - Câmpus Cornélio Procópio

Bibliotecários/Documentalistas responsáveis:  
Simone Fidêncio de Oliveira Guerra – CRB-9/1276  
Romeu Righetti de Araujo – CRB-9/1676

## TERMO DE APROVAÇÃO

Título da Dissertação Nº **045**:

**“CONTROLE DE POSIÇÃO UTILIZANDO ALGORITMO GENÉTICO COM MINIMIZAÇÃO DE ENTROPIA DO ERRO”.**

por

**Daniel Cordeiro Jacinto**

Orientador: **Prof. Dr. Paulo Rogério Scalassara**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM ENGENHARIA ELÉTRICA – Área de Concentração: Sistemas Eletrônicos Industriais, pelo Programa de Pós-Graduação em Engenharia Elétrica – PPGEE – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às \_\_\_\_\_ horas do dia **11 de maio de 2018**. O trabalho foi \_\_\_\_\_ pela Banca Examinadora, composta pelos professores:

Prof. Dr. Paulo Rogério Scalassara  
Presidente

Prof. Dr. Bruno Augusto Angélico  
POLI/USP

Prof. Dr. Wagner Endo  
UTFPR-CP

Prof. Dr. Cristiano Marcos Agulhari  
UTFPR-CP

Visto da coordenação:

\_\_\_\_\_  
Prof. Dr. Alessandro do Nascimento Vargas  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica  
UTFPR Câmpus Cornélio Procópio

“A Folha de Aprovação assinada encontra-se na Coordenação do Programa”.

Dedico este trabalho a Deus e às pessoas mais especiais em minha vida:

Meus pais, José e Suely por sempre estarem zelando por mim e me desejando o melhor com seu amor incondicional.

Minha namorada, Mariely por me acompanhar durante esta etapa e sempre me incentivar.

Meus tios, João e Sílvia que me deram forças durante a graduação e fazem parte de tudo isso.

A todos que participaram de algum modo nessa caminhada e me apoiaram.

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus, que sempre me deu forças e foi minha inspiração para que não desistisse de meus sonhos, pois acredito que se consegui chegar em algum lugar foi por que Ele me capacitou, e que se enfrentaria desafios Ele me ajudaria e daria forças para vencê-los, pois acredito que estava nos seus planos.

Ao meu orientador Paulo Rogério Scalassara e coorientador Cristiano Marcos Agulhari por me ajudarem nesse sonho e serem parte crucial da realização deste trabalho.

Aos demais professores do mestrado, por transmitirem seu conhecimento e me fazer crescer e amadurecer intelectualmente e a secretaria do curso por sempre estar a disposição para ajudar e tirar dúvidas a qualquer momento.

Agradeço aos meus pais e irmãos por estarem sempre ao meu lado e me incentivando, e compartilhando esses momentos.

À minha namorada por estar sempre do meu lado e não me deixando desanimar pelas barreiras durante essa trajetória, acreditando e torcendo para que tudo desse certo.

Aos demais familiares, em especial meus tios que me acompanharam, cuidaram e cederam seu lar durante toda a graduação.

Aos amigos feitos durante esta etapa e que ficarão para sempre presentes.

À CAPES e Fundação Araucária pela bolsa de estudos que foi parte importante para que pudesse me dedicar no mestrado.

À instituição UTFPR-CP.

"Porque Dele, e por Ele, e para Ele, são todas as coisas; glória, pois, a Ele eternamente. Amém."Romanos 11:36.

## RESUMO

JACINTO, Daniel Cordeiro. **Controle de Posição Utilizando Algoritmo Genético com Minimização de Entropia do Erro**. 2018. 68 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Este trabalho tem como proposta a síntese de controladores aplicando um Algoritmo Genético, cuja função objetivo é minimizar a entropia do erro. Recentes estudos demonstram que métodos utilizados em sistemas que utilizam o erro quadrático médio para estimativa de erros não apresentam desempenho satisfatório se tratando de sinais não-gaussianos e não-lineares, assim foi necessária a busca de novas alternativas para resolução de problemas mais complexos. O método de minimização de entropia do erro vem sendo utilizado em pesquisas e apresentando desempenho satisfatório nesta área. Os controladores utilizados são dados na forma de função de transferência e buscou-se pela sintonização dos parâmetros do algoritmo genético em busca de melhor performance para o controlador gerado. Para testes foram feitas simulações utilizando o software MATLAB e a validação foi realizada em uma planta torcional com MATLAB/Simulink. Também é apresentada uma comparação com o método do erro quadrático médio. Resultados satisfatórios foram encontrados para ambos os métodos, porém, notou-se maior tempo de execução para a minimização de entropia devido a maior complexidade de sua função, que utiliza técnicas de janelamento de Parzen para estimar a função densidade de probabilidade do erro.

**Palavras-chave:** Minimização de Entropia do Erro, Algoritmo Genético, Planta Torsional, Erro Quadrático Médio



## ABSTRACT

JACINTO, Daniel Cordeiro. **Position Control Using Genetic Algorithms with Minimization of Error Entropy**. 2018. 68 f. Master Thesis – Electrical Engineering Graduate Program, Federal University of Technology - Paraná. Cornélio Procópio, 2018.

This work proposes the synthesis of controllers applying a Genetic Algorithm, whose objective function is to minimize the entropy of the error. Recent studies demonstrate that methods used in systems that use the mean square error for error estimation do not present satisfactory performance when dealing with non-Gaussian and nonlinear signals, so it was necessary to search for new alternatives to solve more complex problems. The error entropy minimization method has been used in researches and presenting satisfactory performance in this area. The controllers used are data in the form of a transfer function and we searched for the tuning of the parameters of the genetic algorithm in search of better performance for the generated controller. For tests, simulations were performed using MATLAB software and the validation was performed in a torsion plant with MATLAB / Simulink. A comparison with the mean square error method is also presented. Satisfactory results were found for both methods, however, it was observed a longer execution time for the entropy minimization due to the greater complexity of its function, which uses Parzen's windowing techniques to estimate the probability density function of the error

**Keywords:** Minimization of Error Entropy, Genetic Algorithm, Torsion Plant, Mean Square Error

## LISTA DE FIGURAS

FIGURA 1 – Diagrama de blocos controlador . . . . .	17
FIGURA 2 – Resposta degrau . . . . .	18
FIGURA 3 – Diagrama da função de transferência . . . . .	19
FIGURA 4 – Planta torcional com 3 discos com pesos, modelo 205a. . . . .	28
FIGURA 5 – Diagrama esquemático do funcionamento do ECP. AS - Amplificador servo, MP - Módulo de potência . . . . .	30
FIGURA 6 – Esquema ilustrativo do sistema de controle ECP. . . . .	30
FIGURA 7 – Bloco do sistema no software Simulink. . . . .	31
FIGURA 8 – Bloco de reiniciar sistema no software Simulink. . . . .	31
FIGURA 9 – Configuração com 3 discos livres. . . . .	33
FIGURA 10 – Diagrama da planta com 3 graus de liberdade, 3 discos livres, modelo 205a. . . . .	33
FIGURA 11 – Diagrama da planta com 1 grau de liberdade. . . . .	36
FIGURA 12 – Configuração 1. . . . .	37
FIGURA 13 – Sistema de blocos em malha fechada em <i>Simulink</i> . . . . .	43
FIGURA 14 – Resposta degrau sistema com norma alta, simulado no <i>Matlab</i> . . . . .	46
FIGURA 15 – Resposta impulso sistema com norma alta, simulado no <i>Matlab</i> . . . . .	46
FIGURA 16 – Simulação linear de sinal senoidal, sistema com norma alta, simulado no <i>Matlab</i> . . . . .	47
FIGURA 17 – Resposta ao degrau, sigma 50, simulado no <i>Matlab</i> . . . . .	48
FIGURA 18 – Resposta impulso, sigma 50, simulado no <i>Matlab</i> . . . . .	48
FIGURA 19 – Resposta do sinal senoidal, sigma 50, simulado no <i>Matlab</i> . . . . .	49
FIGURA 20 – Resposta ao degrau, sigma 50, validação real do sistema. . . . .	50
FIGURA 21 – Resposta do sinal senoidal, sigma 50, validação real do sistema. . . . .	50
FIGURA 22 – Resposta ao degrau, utilizando EQM, simulado no <i>Matlab</i> . . . . .	51
FIGURA 23 – Resposta impulso, utilizando EQM, simulado no <i>Matlab</i> . . . . .	52
FIGURA 24 – Resposta do sinal senoidal, utilizando EQM, simulado no <i>Matlab</i> . . . . .	52
FIGURA 25 – Resposta ao degrau, utilizando EQM, validação real do sistema. . . . .	53
FIGURA 26 – Resposta do sinal senoidal, utilizando EQM, validação real do sistema. . . . .	54

## LISTA DE SIGLAS

AG	Algoritmo Genético
DA	Digital-Analógico
DSP	Processador Digital de Sinais
EQM	Erro Quadrático Médio
FA	Filtro Adaptativo
FT	Função de Transferência
IP	Potencial de Informação
ITL	Aprendizagem por Teoria da Informação
KLMS	<i>Kernel Least Mean Square</i>
MCC	Critério da máxima correntropia
MEE	Minimização da Entropia do Erro
MIMO	<i>Multiple Input, Multiple Output</i>
pdf	função densidade de probabilidade
PI	Proporcional Integral
RKHS	<i>Reproducing Kernel Hilbert Space</i>
RMS	Raiz do Valor Quadrático Médio
SIMO	<i>Single Input, Multiple Output</i>
SISO	<i>Single Input, Single Output</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	ESTADO DA ARTE	12
1.2	MOTIVAÇÃO	14
1.3	OBJETIVOS	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
2.1	CONCEITOS BÁSICOS SOBRE CONTROLE	16
2.1.1	Modelagem em Função de Transferência	18
2.1.2	Norma $H_2$	20
2.2	ALGORITMOS GENÉTICOS	20
2.2.1	Funcionamento do Algoritmo Genético	21
2.3	TEORIA DA INFORMAÇÃO	25
2.3.1	Minimização da Entropia do Erro	25
<b>3</b>	<b>PLANTA TORCIONAL</b>	<b>28</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>37</b>
4.1	MÉTODO PARA OBTENÇÃO DO CONTROLADOR	37
4.1.1	Geração de indivíduos	39
4.1.2	Avaliação da população	39
4.1.3	Seleção de indivíduos	40
4.1.4	Cruzamento de indivíduos	41
4.1.5	Mutação de indivíduos	42
4.1.6	Restrições de respostas	42
4.2	METODOLOGIA PARA A ANÁLISE DE RESULTADOS	43
4.2.1	Proposta para a validação dos resultados	43
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>45</b>
5.1	RESULTADOS E ANÁLISE DA SATURAÇÃO DO SISTEMA	45
5.2	RESULTADOS PARA NÚCLEO DE TAMANHO 50	46
5.3	RESPOSTA UTILIZANDO EQM COMO FUNÇÃO CUSTO	51
5.4	RESULTADOS PARA DIFERENTES VALORES DE NÚCLEO	55
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>57</b>
	<b>REFERÊNCIAS</b>	<b>58</b>
	<b>ANEXO A – ALGORITMO AG MEE</b>	<b>62</b>

## 1 INTRODUÇÃO

Um grande desafio na engenharia para se trabalhar com dados e informações é a extração e filtragem, pois frequentemente acaba-se absorvendo dados irrelevantes e assim impossibilitando obter-se um modelo estimado que tenha uma boa representação dos dados. Uma boa solução utilizada são os chamados sistemas adaptativos. Contudo, para avaliação desses sistemas faz-se necessário a utilização de métodos de cálculo de erros para se obter a diferença entre os dados reais e valores estimados. O cálculo do Erro Quadrático Médio (EQM) é um exemplo de ferramenta utilizada para essa determinada tarefa (PRINCIPE, 2010). Wiener utilizou-se da proposta do EQM para resolver problemas em filtragem adaptativa, baseando-se em uma abordagem estatística com o propósito de minimizar ruídos presentes nos dados (HAYKIN, 2002).

No entanto, muitas vezes a estatística de segunda ordem não é adequada para situações não-lineares e não gaussianas. Portanto, vários critérios não-EQM, que envolvem estatísticas de ordem superior, têm recebido um crescente interesse. Recentemente, a correntropia, como uma alternativa de EQM, assim como a Minimização da Entropia do Erro (MEE), têm sido utilizados com sucesso para essas situações e em domínios de aprendizagem de máquinas (ZHAO; CHEN; PRINCIPE, 2011).

Os Algoritmos Genéticos (AG) são algoritmos do tipo evolucionários, metaheurísticos, estocásticos, possuem alta probabilidade de encontrar uma solução ótima ou bem próxima dela. Surgiram da teoria de evolução Darwiniana da genética e seleção natural, em que há a diversidade entre indivíduos e que os mais aptos evoluem e perpetuam a espécie (HOLLAND, 1975). Mais sobre este tipo de algoritmo é discutido nos próximos capítulos.

O objetivo principal deste trabalho é a sintonia de controladores através de função de transferência (FT). A sintonia torna possível a obtenção de uma saída próxima à entrada desejada para o sistema. O método proposto é a aplicação de Algoritmos Genéticos, que implementam a busca dos valores ideais dos indivíduos (controladores) para a minimização ou maximização da função objetivo. Escolheu-se o critério de minimização de entropia do erro como função objetivo, que é uma adaptação da entropia quadrática de Rényi utilizando técnicas de janelamento de Parzen, propondo uma estimativa da função de densidade de probabilidade do erro.

### 1.1 ESTADO DA ARTE

Várias pesquisas têm sido feitas na área de aprendizagem supervisionada para substituição de critérios de estimativa de erros, entre artigos relacionados, alguns são citados a seguir.

Santamaría, Pokharel e Principe (2006) desenvolveram uma nova medida de correlação generalizada, denominada correntropia. Esta nova medida inclui informações das características básicas de interesse de processos da engenharia, a distribuição estatística de amplitudes e da estrutura temporal de um processo estocástico, visto que os métodos já utilizados, geralmente eram focados apenas em alguma dessas características. Também esclarecem como pode se utilizar um método de núcleo (kernel) para interpretar esta nova medida, esta grandeza é diretamente relacionada com a entropia quadrática de Rényi e a estimativa de dados através do método de Parzen, discutidos no Cap. 2.

Xu, Erdogmus e Principe (2005), usaram a técnica de Aprendizagem por Teoria da Informação (ITL), aplicada ao observador estendido de Luenberger. Propuseram utilizar a minimização de entropia do erro entre a saída medida e a saída estimada para atualizar os ganhos do observador em vez de pré-especificar os ganhos de observadores globalmente estáveis para sistemas dinâmicos não-lineares. Um algoritmo estocástico baseado em gradiente é demonstrado e apresenta desempenho do observador de entropia tanto para sistemas dinâmicos lineares e não-lineares, ressaltam ainda que essa abordagem leva a introdução para métodos de núcleos em estimativa de estados.

Liu, Pokharel e Príncipe (2007) desenvolvem um conceito para ajudar a entender sobre a aplicação correta de correntropia em processamento de sinais não-lineares e não-gaussianos. Exploram como a correntropia está relacionado à probabilidade de quão semelhante duas variáveis aleatórias são em um espaço comum controlado pela largura de banda do núcleo, e em como ela atua controlando a janela de observação onde a similaridade é avaliada. O Critério da máxima correntropia (MCC) é um método local para medida de semelhanças e apresenta bons resultados quando o dados possuem ruídos sem média zero, não-gaussianos e quando muitos desses ruídos são impulsivos (*outliers*).

Zhao, Chen e Principe (2011) desenvolveram um novo algoritmo adaptativo do núcleo, chamado *kernel maximum correntropy*, que combina as vantagens de um algoritmo chamado *Kernel Least Mean Square* (KLMS) que utiliza o critério de EQM com as vantagens do critério da máxima correntropia, que é utilizado como função custo para minimizar a diferença entre os dados desejados e a saída do filtro já que o MCC é um princípio de adaptação robusta para valores não-gaussianos. Também estudaram suas propriedades de convergência e auto-regularização usando a relação de conservação de energia. O desempenho superior do novo algoritmo foi demonstrado por experimentos de simulação no problema de duplicação de frequências ruidosas.

Já em Craciun et al. (2011), foram comparados os desempenhos de duas funções custo, o filtro analítico de Wiener que objetiva minimizar o EQM e um Filtro Adaptativo (FA) que adapta seus pesos minimizando a entropia do erro. Geralmente em métodos para decodificação dos movimentos de ações motoras através de contagem de pulsos neurais utilizando FA, muitas vezes

dependem de minimizar o EQM, no entanto, para a distribuição não gaussiana de erros, esta abordagem não é otimizada para o desempenho. Portanto, ao invés de utilizarem modelagem probabilística, propuseram uma abordagem alternativa não-paramétrica. Assim, propõe-se a utilização do MEE para extrair mais estruturas do sinal de entrada (contagem de pulsos neurais). Ainda abordaram uma solução para uma desvantagem da utilização do MEE como função custo, que é o aumento da sua complexidade computacional. Para isso, o mapearam para uma arquitetura paralela em hardware reconfigurável adaptado às necessidades computacionais do filtro e observaram uma considerável aceleração de tempo de execução por conta do projeto de hardware.

No artigo de Zhang et al. (2012), foi desenvolvido um filtro usando o critério de minimização de entropia do erro, utilizado no estudo de detecção de falhas em sistemas de controle em rede com atrasos aleatórios, perda de pacotes e ruídos. Utiliza-se o resíduo do filtro MEE para avaliar e detectar falhas em sistemas de controle na rede, é também demonstrado no artigo um sistema ilustrativo de controle em rede para exemplificar a eficácia da proposta abordada.

Em Wang et al. (2015), foi desenvolvido um novo tipo de algoritmo utilizando a correntropia. Neste artigo, foi proposto um novo algoritmo MCC com um degrau de tamanho variável (VSS), chamado de algoritmo VSS-MCC, que pode alcançar uma velocidade de convergência muito mais rápida, mantendo o desempenho de estabilidade semelhante. No novo algoritmo, o tamanho do degrau é atualizado com base em uma aproximação da curvatura da superfície de desempenho. Os resultados da simulação demonstraram uma performance superior do método em comparação com o algoritmo MCC original.

## 1.2 MOTIVAÇÃO

Com os avanços científicos em pesquisas na área de ITL, a utilização e exploração do conceito de entropia para diversas aplicações se torna muito interessante, visto que esta ferramenta tem uma grande capacidade de absorção de informações e se tornou ideal para aplicações em problemas que exigiam técnicas que utilizem estatística superior à segunda ordem. Encontrar e sintonizar controladores para modelos matemáticos pode ser uma tarefa não muito simples, um método para facilitar esse processo seria uma ferramenta realmente útil.

A intuição por trás do critério de entropia para aprendizagem supervisionada possui o seguinte conceito: em um conjunto de amostras de entrada e saída, estimado e desejado, busca-se extrair a maior quantidade de informações das estruturas dos dados, para minimizar o conteúdo da diferença de informação do sinal de erro entre as amostras. Para isso deve-se minimizar a entropia, quanto menor for o valor de entropia, menor será a incerteza, pois terá menor número de diferenças de informações sobre os elementos.

### 1.3 OBJETIVOS

Nesta seção apresentam-se os objetivos deste trabalho, começando pelo objetivo geral e seguido pelos objetivos específicos.

#### 1.3.1 Objetivo Geral

Desenvolver um Algoritmo Genético (AG) capaz de encontrar uma função de transferência de controle para a realimentação de um modelo de uma planta de sistema torcional, utilizando uma função custo que absorva o máximo de informações dos dados e minimize a entropia do erro, em seguida compará-la com uma função custo utilizando erro quadrático médio.

#### 1.3.2 Objetivos Específicos

Os objetivos específicos são organizados da seguinte maneira:

- Implementar um algoritmo genético para obtenção de controlador para a planta.
- Identificar parâmetros e valores para o algoritmo genético, como população, indivíduos, operações genéticas, função custo.
- Utilizar função de transferência do modelo da planta torcional para a busca de controladores.
- Fazer testes para encontrar melhores configurações do algoritmo para o modelo da planta.
- Obter e simular no Matlab ,controladores para a planta torcional.
- Testar e validar os controles encontrados para o modelo do sistema, utilizando o software *MATLAB/Simulink*

Este trabalho aplica testes para a planta torcional em um modelo de segunda ordem, apenas com efeito no controle de posição. No entanto, o método proposto pode ser aplicado a outras plantas e com ordens diferentes.

Este trabalho está organizado da seguinte maneira. No Capítulo 2, aborda-se a fundamentação teórica para todas as ferramentas utilizadas no desenvolvimento do trabalho. São apresentadas conceitos sobre controle, algoritmos genéticos, teoria da informação e métodos de núcleo. No Capítulo 3, é feito um estudo do equipamento de sistema torcional. Na sequência, no Capítulo 4, é apresentada toda a metodologia para a realização do trabalho. Os resultados obtidos são apresentados no Capítulo 5 e em seguida no último capítulo, serão apresentadas as considerações finais sobre o trabalho.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, abordam-se os tópicos principais sobre a teoria que abrange os temas e ferramentas a serem utilizadas no trabalho. Inicialmente serão apresentados os conceitos básicos de controle por realimentação na forma de função de transferência, seguido por um explicativo sobre os algoritmos genéticos, teoria da informação, entropia e métodos de núcleo.

### 2.1 CONCEITOS BÁSICOS SOBRE CONTROLE

Sistemas de controle são constituídos em subsistemas e processos (ou plantas) agrupados, sendo um dos intuitos, o de se obter o controle das saídas dos processos. Como exemplo, pode ser citado uma caldeira que com a entrada de fluxo de combustível produz calor como resultado. Os subsistemas neste caso são as válvulas de combustível e seus atuadores são usados para regular a temperatura, através do controle da saída de calor da caldeira. Outros subsistemas medem a temperatura de saída, se comportando como sensores, como os termostatos. Basicamente, sistemas de controle são definidos por fornecer uma resposta ou saída para um certo estímulo ou entrada (NISE, 2002).

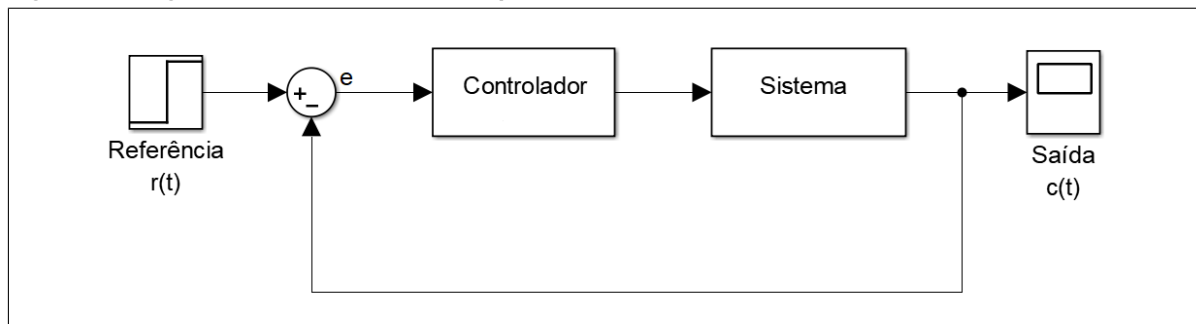
Plantas podem ser definidas como objetos físicos a serem controlados, e são compostas por componentes que trabalham de maneira integrada em um equipamento e que possuem determinadas operações como objetivo. Refere-se à finalidade de controle com realimentação quando se tende a diminuir a diferença entre a entrada de referência e a saída do sistema, mesmo que com presença de distúrbios, sua atuação será na diferença de sinais para aplicar a correção necessária (OGATA, 2012).

Em sistemas em malha fechada, também chamados de sistemas de controle com retroação, a diferença entre sinal de referência (entrada ou resposta desejada) e a saída atual (resposta real) é obtida através da realimentação. Um controlador é então projetado para minimizar esta diferença (erro), aplicando sinais de controle até que a saída do sistema alcance o valor de referência ou bem próximo dele. Um esquema deste sistema de controle pode ser visto pela representação em diagrama de blocos na Figura 1.

Os sistemas de malha fechada possuem vantagens em vista dos sistemas de malha aberta, como menor sensibilidade e capacidade de correção a perturbações (ou distúrbios) que tendem a afetar de maneira adversa a saída de um sistema. Com isso, pode-se ter uma economia com componentes menos precisos e baratos para se controlar determinado sistema, o que não seria possível utilizando sistemas de malha aberta (NISE, 2002; OGATA, 2012).

Em contraste, a estabilidade em sistemas com realimentação constitui um problema importante, visto que podem tender a corrigir os erros além do que se necessita, podendo

**Figura 1 – Diagrama de blocos de um exemplo de sistema de controle em malha fechada.**



**Fonte: Adaptado de (NISE, 2002).**

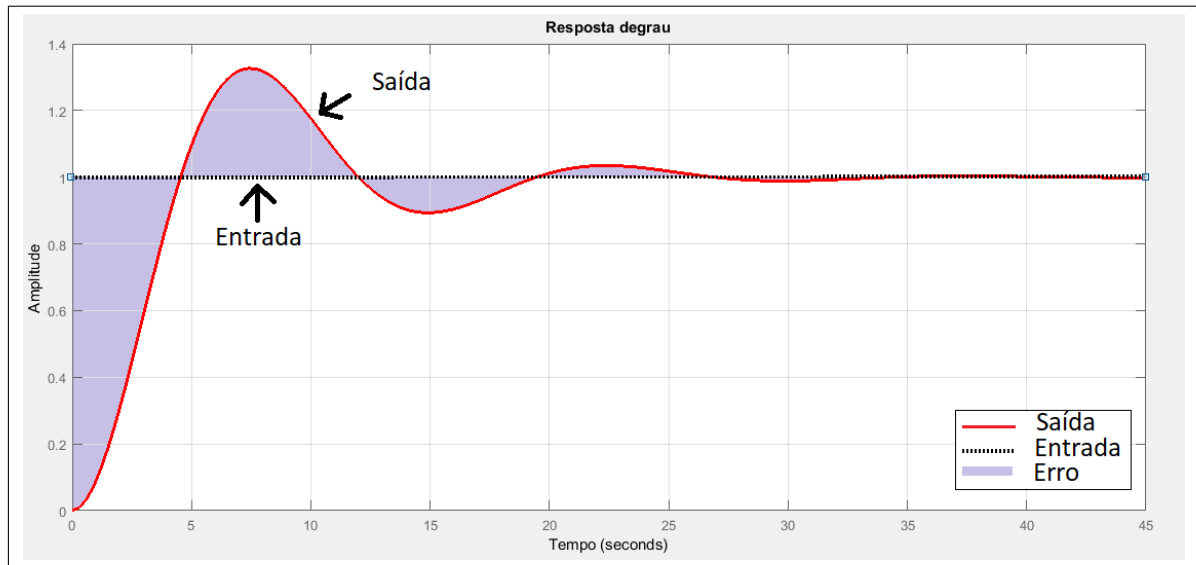
ocasionar oscilações de amplitude variável ou contante. Já em malha aberta, a estabilidade é um problema menos significativo, facilitando a construção do controle, quando isentos a presença de distúrbios (OGATA, 2012).

Nos sistemas com realimentação pode-se, muitas vezes através de um pequeno ajuste de ganho na malha ou também reprojetoando o controlador, obter melhor controle e maior flexibilidade para o erro de estado estacionário, definido como a diferença resultante após um determinado intervalo de tempo, e para a resposta transitória que consiste na porção inicial do sinal (NISE, 2002).

Um exemplo gráfico com os sinais de entrada (Referência) e saída real de um sistema em função do tempo pode ser visto na Figura 2, referente a uma entrada degrau unitário, uma das referências comumente usada para análise de controle. No exemplo, o sinal de saída apresenta erros até que se estabilize no valor desejado, por volta dos 35 segundos. O sinal neste período de estabilização é chamado de resposta transitória. A partir do momento em que o sinal estabiliza e se mantém com um comportamento constante, tem-se a chamada resposta de regime permanente ou estado estacionário. O conceito de estabilidade se dá quando o sistema tende a manter o valor desejado sem alterações imprevistas, mesmo com a ação de pequenas perturbações (NISE, 2002).

Dependendo do tipo de aplicação, é necessário que a saída satisfaça certos requisitos de desempenho. Em alguns casos necessita-se de uma rápida resposta que tenda a reduzir o erro para o menor possível no menor tempo possível; em outros casos, é necessário que o valor tenda gradativamente ao valor de referência em um tempo maior, com um comportamento mais suave, como em um elevador, em que não se deve aplicar a aceleração total de uma só vez ou causaria certo desconforto nas pessoas que o utilizam, e quando se chega ao andar de destino não é desejada muita oscilação até que se estabilize (OGATA, 2012).

Figura 2 – Saída de um sistema com erros em regime transitório.



Fonte: Adaptado de (NISE, 2002).

### 2.1.1 Modelagem em Função de Transferência

Para estabelecer uma definição viável de uma função de relação algébrica entre saída do sistema e sua entrada, que inclusive permite a análise do sistema total em sub-blocos, como mostrado na Figura 1, é utilizada a representação por função de transferência. Diferentemente da equação diferencial, esta representação pode ser separada em três partes distintas, sendo entrada, sistema e saída. Também pode-se combiná-las para representação total do sistema (NISE, 2002).

A equação diferencial de forma geral, que descreve um sistema linear e invariante no tempo, de ordem  $n$ , Eq. (1), sendo  $c(t)$  e  $r(t)$  a saída e entrada, respectivamente, e os elementos  $a_0 \cdots a_n, b_0 \cdots b_m$  coeficientes constantes e reais e  $m \leq n$  (NISE, 2002).

$$a_n \frac{d^n c(t)}{dt^n} + a_{n-1} \frac{d^{n-1} c(t)}{dt^{n-1}} + \dots + a_0 c(t) = b_m \frac{d^m r(t)}{dt^m} + b_{m-1} \frac{d^{m-1} r(t)}{dt^{m-1}} + \dots + b_0 r(t) \quad (1)$$

Após a aplicação da Transformada de Laplace nos dois lados da equação, obtém-se:

$$\begin{aligned} & a_n s^n C(s) + a_{n-1} s^{n-1} C(s) + \dots + a_0 C(s) + \text{termos de condição inicial envolvendo } c(t) \\ & = b_m s^m R(s) + b_{m-1} s^{m-1} R(s) + \dots + b_0 R(s) + \text{termos de condição inicial envolvendo } r(t) \end{aligned} \quad (2)$$

Considerando que as condições iniciais sejam nulas, tem-se:

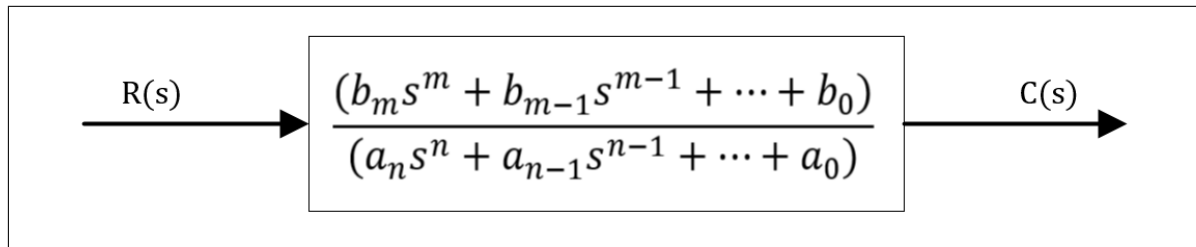
$$(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0)C(s) = (b_m s^m + b_{m-1} s^{m-1} + \dots + b_0)R(s) \quad (3)$$

Dividindo a transformada da saída  $C(s)$  pela da entrada  $R(s)$  resulta em

$$\frac{C(s)}{R(s)} = \frac{(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0)}{(b_m s^m + b_{m-1} s^{m-1} + \dots + b_0)} \quad (4)$$

Chamamos esta relação de Função de Transferência (FT) (NISE, 2002). A FT também pode ser representada em forma de um diagrama de blocos, como pode ser visto na Figura 3.

**Figura 3 – Diagrama de blocos de um exemplo de sistema de função de transferência.**



Fonte: (NISE, 2002)

Assim a relação de entrada e saída de um sistema igual ao diagrama de blocos da figura 1, pode ser definido como na Eq. (5), sendo  $G(s)$  a função de transferência do controlador e  $H(s)$  a da planta.

$$\frac{C(s)}{R(s)} = \frac{G(s)H(s)}{1 + G(s)H(s)} \quad (5)$$

Toda função de transferência pode ser decomposta em termos de polos e zeros. A análise dos polos e zeros pode dizer muito sobre a estabilidade do sistema. Um sistema de controle só é estável se todos os pólos de malha fechada pertencerem ao semiplano esquerdo do plano  $s$ . Isto é, a estabilidade é garantida se os pólos possuírem parte real negativa, alterações nestes podem alterar o desempenho do sistema. Porém este fato ainda não garante que as características da resposta transitória sejam satisfatórias. Se os polos complexos conjugados dominantes estiverem próximos do eixo  $j\omega$ , a resposta poderá ser muito lenta ou com oscilações excessivas (OGATA, 2012).

Pode-se também analisar as raízes da equação característica (polinômio do denominador) da malha fechada, o critério de Routh-Hurwitz possibilita estabelecer a quantidade de pólos que se situam no plano direito de  $s$ , sem a necessidade de fatoração do polinômio denominador. Ele nos diz se há a ocorrência ou não de raízes instáveis em uma equação polinomial, sem

a necessidade de resolvê-la. O comportamento da resposta transitória e a estabilidade em sistemas de malha fechada estão ligadas com as posições das raízes da equação característica (OGATA, 2012).

### 2.1.2 Norma $H_2$

A norma  $H_2$  de um sistema estável  $H$  é a Raiz do Valor Quadrático Médio (RMS) da resposta ao impulso do sistema. A norma  $H_2$  mede a covariância de estado estacionário (ou potência) da saída  $y = Hw$  às entradas unitárias de ruído branco  $w$  (BRUINSMA; STEINBUCH, 1990):

$$\|H\|_2^2 = \lim_{t \rightarrow \infty} E\{y(t)^T y(t)\}, \quad E\{w(t)w(\tau)^T\} = \delta(t - \tau)I \quad (6)$$

A norma  $H_2$  de um sistema de tempo contínuo com função de transferência  $H(s)$  é dado

$$\|H\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Traço}[H(j\omega)^H H(j\omega)] d\omega} \quad (7)$$

A realização de uma medida de energia de certos sinais de interesse, é uma das maneiras mais utilizadas para se atender especificações de desempenho, que é um dos principais objetivos a serem atingidos no desenvolvimento de um sistema de controle (TROFINO; COUTINHO; BARBOSA, 2005).

## 2.2 ALGORITMOS GENÉTICOS

Os algoritmos genéticos (AG's) são ferramentas para otimização, que buscam melhorar resultados para alguma circunstância. Na engenharia, busca-se, em geral, a minimização de recursos e esforços e a maximização de resultados para determinadas tarefas. Nos últimos anos, houve um aumento considerável nas propostas de novos métodos de otimização não tradicionais, ou chamados também de modernos, que fogem do conceito das técnicas de otimização habitual. Estas técnicas, podem ser bioinspiradas, baseadas em comportamentos e características de sistemas neurobiológicos, enxame de insetos, biologia e moléculas, por exemplo (RAO, 2009). Os algoritmos genéticos, otimização por enxame de partículas, otimização de colônia de formigas e recozimento simulado são exemplos deste tipo de otimização. A maioria destes métodos foram desenvolvidos apenas nos últimos anos e são cada vez mais utilizados para resolução de problemas complexos de engenharia (RAO, 2009).

Algoritmos genéticos são inspirados na genética natural e nos princípios de seleção natural. Filosoficamente, baseiam-se na teoria Darwiniana de evolução da espécie (HOLLAND,

1975). O AG é um método estocástico, apresentando alta probabilidade de encontrar um mínimo global e é naturalmente aplicável para solução de problemas de otimização (HOLLAND; GOLDBERG, 1989; RAO, 2009).

Vários aspectos diferenciam os AG's dos métodos tradicionais, dentre eles: a utilização de uma população de vários indivíduos e não um único, geralmente, atribui-se de duas a quatro vezes a quantidade de variáveis de projeto, para que se reduza a probabilidade de encontrar um mínimo local, visto que a banda de exploração é maior. Usa-se exclusivamente dos valores da função objetivo ou *fitness* no processo de busca, sendo tal função utilizada para determinar se foi encontrado um bom resultado ou não. Representação das variáveis de projeto como sequências binárias ou utilizando valores inteiros ou reais para corresponder aos cromossomos da natureza genética. Em cada geração, novos conjuntos de respostas são produzidos utilizando operações genéticas com gerações anteriores, assim gerando uma nova população.

### 2.2.1 Funcionamento do Algoritmo Genético

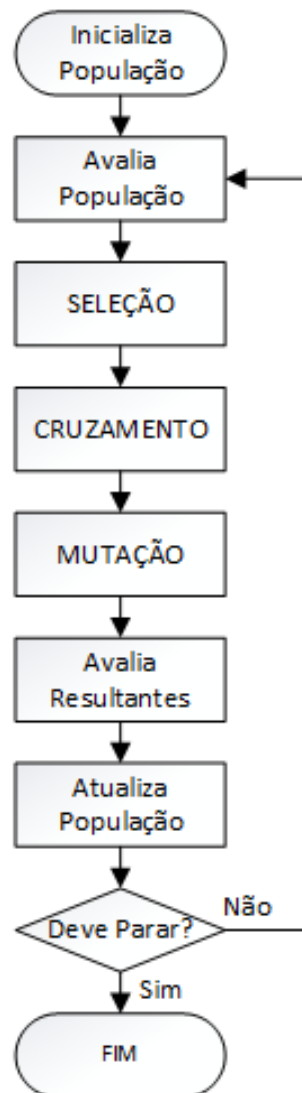
O Fluxograma 1 ilustra o funcionamento básico de um AG, descrito em detalhes a seguir.

Na primeira etapa, cria-se uma população aleatória de indivíduos que representem possíveis respostas do problema a ser otimizado. Essa aleatoriedade proporciona uma maior variedade entre os indivíduos, o que garante maior espaço de busca. A quantidade de indivíduos pode variar, mas se for um valor muito pequeno, é possível que o AG não encontre um resultado global. Por outro lado, uma quantidade muito grande de indivíduos tem mais chances de explorar as respostas e encontrar um bom resultado, mas pode requerer um custo computacional maior (HOLLAND; GOLDBERG, 1989).

Além do tamanho da população, é importante também determinar o método de inicialização. Segundo Holland e Goldberg (1989), Geyer-Schulz (1997), há várias formas de se inicializar a população inicial, sendo

- o método de inicialização aleatória uniforme, onde cada gene do indivíduo recebe, através de sorteios uniformemente aleatórios, valores possíveis de um conjunto;
- método de inicialização aleatória não uniforme, em que certos valores do conjunto têm uma probabilidade maior de serem armazenados no gene;
- na forma aleatória com "dope", há a inserção de indivíduos já otimizados entre a geração inicial, o que pode ocasionar um convergência prematura;
- inicialização parcialmente enumerativa, que insere indivíduos em uma faixa de valores fazendo com que uma determinada ordem já seja selecionada, determinando os esquemas

Fluxograma 1 – Fluxograma de funcionamento de um algoritmo genético tradicional.



Fonte: Adaptado de Holland (1975).

possíveis.

A população inicial é então criada e submetida ao processo de evolução, composto pelas seguintes partes:

- **Avaliação:** Avaliar os indivíduos em relação à função custo, também chamada de função objetivo. Nesta etapa, todos os elementos são testados de acordo com o critério a ser otimizado. Aqui deve ser desenvolvido um sistema ideal de verificação para o problema.

Esta tarefa pode ser complexa e em algumas situações é normal encontrar a utilização de funções não determinísticas que utilizam apenas uma parte dos dados dos indivíduos e não sua totalidade. Esta operação é feita várias vezes durante o processo de evolução, o que pode ser computacionalmente custoso.

• **Seleção:** Seleciona as melhores soluções para que possam sofrer as evoluções como cruzamento e mutação, ocorrendo nesta etapa a perpetuação das melhores características. Segundo Dawkins (1996) a seleção é um processo dirigido e cumulativo, uma vez que o indivíduo só sobrevive e se reproduz se ele possuir capacidade para isso. O efeito cumulativo se dá pelo fato de sempre se manter as características boas de uma geração para outra, resultando em elementos complexos e mais evoluídos. A probabilidade de seleção é proporcional à sua qualidade de avaliação, isto é, quanto melhores suas características forem, mais chances terá de ser escolhido.

Há vários métodos para se utilizar na seleção, alguns deles são:

- seleção por giro de roleta, em que calcula-se o total de todos os valores de avaliação, sorteia-se um valor pertencente ao intervalo de 0 ao total, e o indivíduo que estiver com o valor na faixa sorteada é selecionado (DAVIS, 1990).
- seleção por ranking, em que cada indivíduo recebe uma probabilidade de seleção de acordo com seu valor avaliação (DAVIS, 1990; HOLLAND; GOLDBERG, 1989).
- há também a seleção uniforme, que não é muito vantajosa na maioria das utilizações, pois todos os indivíduos apresentam a mesma probabilidade de serem escolhidos, assim tornando mais difícil uma melhora de resultados da população (GEYER-SCHULZ, 1997; HOLLAND; GOLDBERG, 1989; WALL, 2008).
- seleção por torneio, em que são separados grupos sucessivos e os melhores indivíduos desses grupos são escolhidos (GEYER-SCHULZ, 1997; HOLLAND; GOLDBERG, 1989; WALL, 2008)

• **Cruzamento:** *crossover* ou reprodução, recombina-se as características das soluções selecionadas (pais) para gerar outros indivíduos (filhos). Há diversas variações de cruzamento, algumas delas estão apresentadas a seguir, como pode ser visto também em Geyer-Schulz (1997).

Cruzamentos podem ser feitos através de escolha ou sorteio de "pontos" de corte, isto é, os pontos são os genes dos indivíduos.

- o cruzamento de um único ponto faz com que seja sorteado um número entre a quantidade de total de genes dos pais, que são sorteados. O primeiro filho recebe os genes do primeiro pai até o ponto escolhido, e o restante do outro pai. O segundo filho acaba recebendo o inverso do primeiro.



- cruzamento multipontos, em que é sorteado um número de pontos e, a partir deles, é feito igualmente ao sistema de um ponto só que dessa vez intercalam-se os pais a herdarem os genes a cada vez que se troca-se o ponto.
- cruzamento segmentado é parecido com o sistema multipontos mas ao invés de se manter os mesmos pontos sempre, alteram-se os pontos de troca toda vez que é executado.
- no cruzamento uniforme, sorteia-se um por um a cada gene a ser inserido nos filhos, de qual pai herdará, gerando uma certa aleatoriedade.
- no sistema de cruzamento por combinação parcial são sorteados dois pontos, os genes entre esses pontos são recebidos pelos filhos, um filho recebe os genes do primeiro pai e o outro do segundo, o restante dos genes faltantes são preenchidos de forma mais adequada para cada um.

• **Mutação:** faz com que algumas características dos indivíduos resultantes sejam alteradas para gerar mais variedade genética. Com certa porcentagem de acontecimento, cada filho pode sofrer uma alteração em seus genes. Entre as possíveis abordagens de mutação, podem-se citar (GEYER-SCHULZ, 1997; HOLLAND; GOLDBERG, 1989; WALL, 2008):

- mutação por troca ou *swap mutation*, onde é aleatoriamente escolhido um par ou mais de elementos para fazerem a troca de posição entre si.
- mutação *creep*, onde é escolhido um elemento aleatório que pode sofrer a alteração somando-se ou subtraindo-se um determinado valor dele.
- mutação *flip*, em que determinam-se elementos possíveis como substituição para os elementos dos filhos sorteados.

• **Atualização** da população, faz com que os novos indivíduos criados e que sofreram alterações substituam a população inicial.

Geralmente, mantém-se a quantidade inicial de indivíduos, mas esta não é a única opção. Podem haver a inserção de um número menor de indivíduos, também há a opção de trocarem todos os pais pelos filhos gerados ou a escolha dos que possuem melhor aptidão.

• **Finalização:** Finaliza-se se todas as condições de término estiverem alcançadas, caso contrário, continua-se a rotina retornando para a etapa de avaliação. O critério de parada pode ser por limite de iterações máximas ou uma faixa de convergência aceitável.

## 2.3 TEORIA DA INFORMAÇÃO

A área de teoria da informação avançou, paralelamente, às pesquisas na área de sistemas adaptativos. O conceito de entropia de informação foi proposto como uma medida de incerteza por Shannon que o transformou em uma medida de quantidade de informação. Shannon, em busca de uma solução para segurança em comunicação, formalizou conceitos de medida de informação, de decodificação e capacidade de transferência de informação sobre um canal (SHANNON; WEAVER, 1949).

O matemático Húngaro Alfred Rényi contribuiu bastante para o avanço das pesquisas nesta área, ele apresentou uma família paramétrica de entropia, estendendo o conceito e generalizando matematicamente a entropia de Shannon (RÉNYI, 1961).

Nessa área, Príncipe (2010) foi um dos primeiros a utilizar o termo ITL, onde se tem como principal questão, como estimar os dois mais importantes descritores estatísticos sugeridos pela teoria da informação, a entropia e divergências diretas das amostras (REGO, 2014).

A ITL não tem a necessidade de conhecer a distribuição dos dados inicialmente, somente da disposição dos mesmos, ela também independe da arquitetura de aprendizagem de máquina (PRINCIPE; XU; FISHER, 2000).

Métodos tradicionais de algoritmos de adaptação que utilizam o erro quadrático médio não são muito viáveis em questões de eficiência se considerarmos sinais não-gaussianos e momentos além de segunda ordem de distribuição de erro. Para isso, em Erdogmus (2002) desenvolveu-se uma função custo alternativa para adaptação chamada de critério de MEE, que busca a eliminação das incertezas do sinal de erro. A partir da entropia de Rényi, foi desenvolvido um estimador não paramétrico que usa as informações diretamente contidas nos dados obtidos de experimentos. Este método, juntamente com outras otimizações, possibilitou a utilização da entropia de Rényi em outros tipos de problemas como a extração de características, separação cega de fontes, entre outras (REGO, 2014).

### 2.3.1 Minimização da Entropia do Erro

A entropia quadrática de Rényi, utiliza-se de técnicas de janelamento de Parzen para estimar a função densidade de probabilidade (pdf) do erro. A entropia quadrática de Rényi  $H_2$ , de uma variável aleatória  $X$  com pdf  $f_x(x)$ , demonstrada em Liu, Pokharel e Príncipe (2007), também utilizada em Xu, Erdogmus e Príncipe (2005), Craciun et al. (2011), Zhang et al. (2012), é definida na Eq. (8)

$$H_2(X) = -\log \int f_X^2(x) dx \quad (8)$$

$H_2$  utiliza de forma implícita a distância euclidiana do ponto  $f(x)$ , em um espaço vetorial

chamado "simplex", para a origem do espaço. O "simplex" consiste em todas as distribuições de probabilidade possíveis para uma variável aleatória multidimensional,  $N$  (PRINCIPE, 2010).

A estimativa de Parzen da pdf, dado um conjunto de dados independentes de distribuição idêntica  $\{x_i\}_{i=1}^N$  extraídos da distribuição, é Eq. (9)

$$\hat{f}_{X;\sigma}(X) = \frac{1}{N} \sum_{i=1}^N \kappa_{\sigma}(x - x_i) \quad (9)$$

Sendo o *kernel* Gaussiano  $\kappa_{\sigma}(x - x_i)$ ,  $N$  o número de pontos dos dados e  $\sigma$  a largura do *kernel* ou parâmetro de suavização, Eq. (10)

$$\kappa_{\sigma}(x - x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - x_i)^2}{2\sigma^2}\right) \quad (10)$$

O tamanho do *kernel* é um parâmetro livre e deve ser escolhido pelo usuário aplicando conceitos de estimativa de densidade, como a regra de *Silverman* (SILVERMAN, 1986) que apresenta sugestões de escolha desse parâmetro ou escolha por máxima verossimilhança.

Um estudo experimental foi feito e verificado que o tamanho do kernel afeta muito menos a performance dos algoritmos ITL do que a estimativa de densidade (ERDOGMUS; PRINCIPE, 2006).

Segundo Liu, Pokharel e Príncipe (2007), o Potencial de Informação (IP), Eq. (11), representa a média quadrática dos dados projetados, isto é, o primeiro momento estatístico dos dados de um espaço de Hilbert reproduzido por núcleo chamado de *Reproducing Kernel Hilbert Space* (RKHS).

$$IP(X) = \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N \kappa_{\sqrt{2}\sigma}(x_j - x_i) \quad (11)$$

Assim, gerando uma estimativa da entropia quadrática não paramétrica diretamente das amostras sem a necessidade de resolver integrais, utilizando núcleos gaussianos, Eq. (12), sendo o logaritmo uma função sempre crescente. A maximização da entropia quadrática refere-se a minimização do argumento do logaritmo ou vice-versa, denominado potencial de informação. Ele pode ser denotado como uma norma quadrática do vetor média dos dados no espaço de características de Mercer (REGO, 2014).

$$H_2(X) = -\log IP(X) \quad (12)$$

O *IP* se assemelha a estimadores convencionais de média e variância trabalhando com amostras diretamente, sem depender da *pdf*, porém, este estimador utiliza-se da largura do núcleo, um parâmetro livre que deve ser escolhido (REGO, 2014). Assim, quando se estima o

Potencial de Informação, os valores de entropia que se resultam, dependem do valor de núcleo escolhido (PRINCIPE, 2010).

Apesar de existirem muitas maneiras para se estimar a entropia de uma variável aleatória, quando as amostras não fazem parte de uma família paramétrica de *pdf's* conhecidas, utiliza-se, na maioria das vezes, uma estimativa não paramétrica. Ou estima-se a *pdf* para então calcular a entropia (ERDOGMUS, 2002).

Este capítulo apresentou alguns conceitos relacionados sobre controle, algoritmos genéticos e teoria da informação, assuntos que serão utilizados na metodologia proposta, assim como a planta torcional, que será detalhada no próximo capítulo.

### 3 PLANTA TORCIONAL

A planta torcional, Modelo 205 da *Educational Control Products* é um equipamento de estudo de controle, que possui várias formas de se explorar e utilizar técnicas de identificação, controles e validações. Como pode ser visto na Figura 4, ele é composto de várias partes que podem ser alteradas, gerando configurações distintas. Ele possui três discos ligados por dois eixos torcionais flexíveis, sendo o disco inferior movido por um servo-motor através de correia. Seu sistema se assemelha à dinâmica de perfuratrizes e outras situações típicas encontradas no controle de sistemas industriais que apresentam flexibilidade.

**Figura 4 – Planta torcional com 3 discos com pesos, modelo 205a.**



Fonte: (PARKS, 1999).

Na Figura 4, os itens enumerados se referem a:

1. Sensores de posição em cada disco (codificador óptico de alta resolução), que medem o deslocamento angular;
2. Discos de fácil remoção (Planta pode conter 1, 2 ou 3 discos), ou podem ser travados;
3. Cabo de torção (Aço temperado,  $k$  - constante de mola);
4. Ajustadores de  $J$  - momento de inércia - (Pesos móveis e removíveis);
5. Suporte de rolamentos (Rolamentos de precisão em cada disco);
6. Atuador (Servo motor DC de alto  $T$  - torque -, sem escova);
7. O motor movimenta o disco inferior através de correia de distribuição rígida com redução de velocidade 3:1.

O sistema ECP trabalha em certas unidades e fatores de conversão, sendo:

**Counts:** Unidade utilizada pelos codificadores, corresponde ao número de ranhuras do disco ótico. Uma volta completa corresponde a 16.000 *counts*,  $\frac{2\pi}{16.000}$  radianos e  $\frac{360}{16.000}$  graus.

**DACcounts:** utilizada internamente pelo DSP na entrada do conversor Digital-Analógico (DA) . Possui o valor de 32 *counts* ou  $\frac{10}{32.768}$  Volts (PALHARES et al., 2002).

A possibilidade de alteração de seus parâmetros físicos, como os valores de inércia e constantes de mola, o tornam ideal para várias pesquisas. No Quadro 1, podem ser observadas algumas configurações da planta.

**Quadro 1 – Configurações da planta ECP modelo 205.**

Configurações: 6 tipos na forma padrão, 9 com 3º disco opcional, 18 com acessório secundário
Dinâmica: Ajustável a 2ª, 4ª e 6ª ordem (3 opções de disco), Sistemas de tipos 0 e 2
I/O: <i>Single Input, Single Output</i> (SISO), <i>Single Input, Multiple Output</i> (SIMO), <i>Multiple Input, Multiple Output</i> (MIMO) (com acessório de acionamento secundário)
Polos e Zeros: Ajustáveis 0.8 - 7 Hz
Taxa de Ajuste da Inércia: 10:1
Taxa de ajuste de mola: 2:1 (certas configurações)
Feedback: Codificador de alta resolução (16.000 count/ciclo)
Atuador: Motor sem escova de alto torque, 2,0 N.m
Tamanho: 30x30x96 cm. (12x12x36 pol.)
Características de segurança: Proteção contra sobrecarga do amplificador. No <i>firmware</i> : proteção de deslocamento relativo (mola), proteção contra sobrecarga de velocidade, proteção térmica

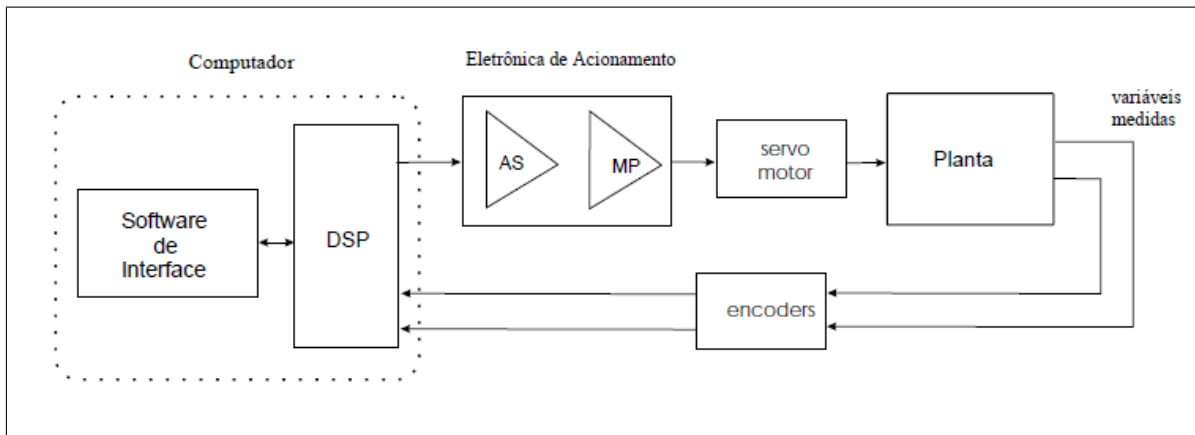
Fonte: (PARKS, 1999).

Este equipamento conta com software de controle que se comunica com o Processador Digital de Sinais (DSP) para controles em tempo real. Neste software, podem-se implementar vários tipos de técnicas de controle, gravar dados, criar gráficos, entre outras aplicações. Na sua interface, podem ser observados os valores dos decodificadores do equipamento na medida em que se alteram.

O esquema de funcionamento da planta pode ser observado através do diagrama da Figura 5.

O amplificador servo e o módulo de potência fornecem a potência necessária para o motor, aplicando a tensão ordenada (na saída do conversor DA) no motor instantaneamente.

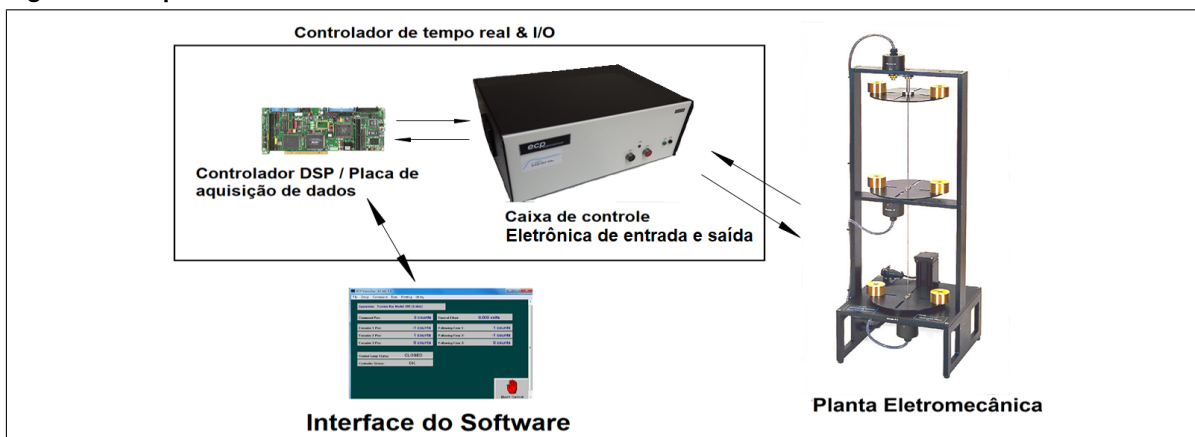
Figura 5 – Diagrama esquemático do funcionamento do ECP. AS - Amplificador servo, MP - Módulo de potência



Fonte: (PALHARES et al., 2002).

A Figura 6 ilustra o fluxo de informações.

Figura 6 – Esquema ilustrativo do sistema de controle ECP.

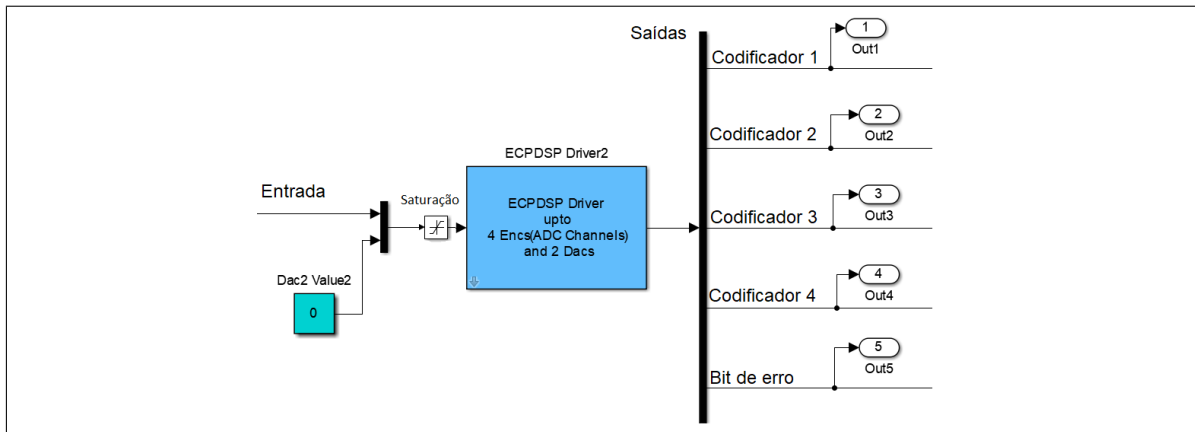


Fonte: Adaptado de (PARKS, 1999).

As posições angulares ou realimentação incremental das posições são fornecidas pelos *encoders*, essas informações são diretamente encaminhadas através da caixa de acionamento (ou caixa de controle) para o DSP. A placa DSP converte os sinais de pulsos em uma base de tempo fixa para valores numéricos como a velocidade e posição. Através dos conversores digitais-analógicos ela atua sobre a planta eletromecânica e também recebe informações de segurança do sistema por meio de sensores (PALHARES et al., 2002).

Além do software fornecido pelo equipamento, ele possui também comunicação com o *software Simulink*, onde há um bloco (Figura 7) representando o sistema ECP que faz a comunicação com a placa do DSP. Deve ser feita a escolha de qual utilizar, *Simulink* ou software supervisor fornecido, sendo somente possível uma única comunicação por vez.

**Figura 7 – Bloco do sistema no software Simulink.**

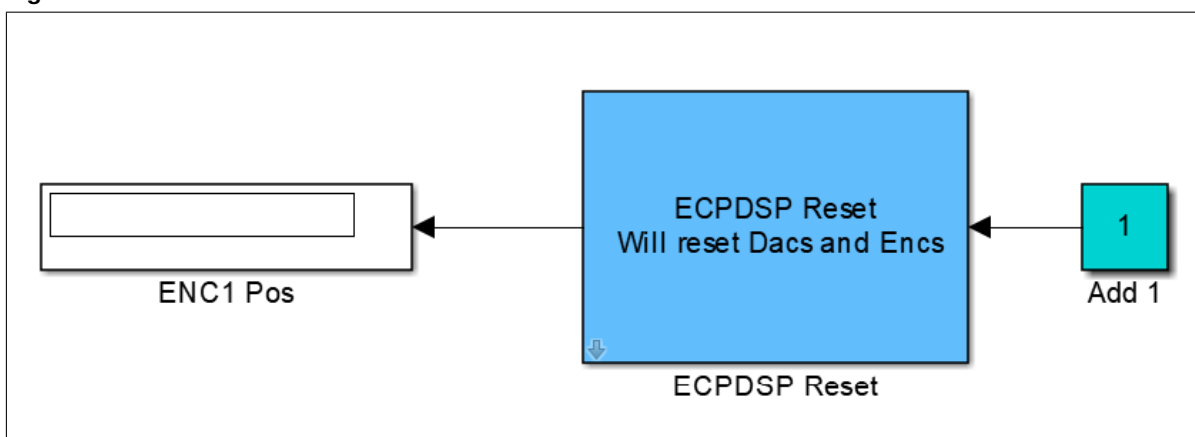


Fonte: Adaptado do software Simulink.

Na entrada deve se também ser inserido um bloco de saturação entre -32.000 a 32.000 proposto pelo próprio modelo padrão fornecido pelo ECP. As saídas dos codificadores 1 a 4 demonstram as posições angulares sendo de 1 a 3 para os 3 discos, em que o primeiro se refere ao disco inferior, e o 4<sup>o</sup> quando se utiliza o aparato de pêndulo invertido (PARKS, 1999), que não é considerado neste trabalho.

É disponibilizado também um modelo do *Simulink* (Figura 8) para "resetar" os codificadores, que é recomendável ser executado antes de cada ensaio. A saída "ENC1 Pos", demonstrada na figura, informa se há (valor 0) ou não (valor 1) erros no processo, que podem ocorrer, por exemplo, se ensaios anteriores foram interrompidos pelo sistema de proteção.

**Figura 8 – Bloco de reiniciar sistema no software Simulink.**



Fonte: Adaptado do software Simulink.

Os valores convertidos pela placa DSP podem ser considerados como disponíveis num contínuo ao longo do tempo pois a conversão dos pulsos em sinal digital e de sinal digital para analógico ocorrem em frequência suficientemente elevada, significando que não há perda de



informações relevantes quando se adota valores intermediários interpolados entre duas amostras subsequentes. Isto faz com que seja possível a representação de sistemas mecânicos (evitando a representação amostrada) em tempo contínuo e os modelos para conversões DA e conversões de pulsos são atribuídos por simples ganhos algébricos (PALHARES et al., 2002).

O motor DC sem escovas, também conhecido como motor síncrono de imã permanente, é capaz de fornecer altos valores de torque através de um pequeno rotor. O campo é comutado eletronicamente e um controlador Proporcional Integral (PI) é responsável pelo seu acionamento, o que faz com que sua resposta dinâmica seja predominantemente instantânea do ponto de vista da planta mecânica, assim sua dinâmica pode ser desconsiderada (PARKS, 1999).

No supervisorio fornecido, as configurações de algoritmo de controle possibilitam a utilização de várias estruturas e parâmetros para controle em tempo real, como PID, PI com realimentação de velocidade, caminho dinâmico direto, pré-filtro dinâmico/caminho de retorno, realimentação de estados, entre outros. Pode-se também importar controladores de outros softwares como *Matlab* e *Matrix X*. Outra função presente no *software* do sistema, é que ele pode ser configurado para várias opções de entrada, como os sinais: impulso, degrau, rampa, parábola, cúbico, senoidal, *chirp* ou um sinal definido pelo usuário. Todas as entradas podem ser especificadas como unidirecional ou bidirecional, sendo, bidirecional quando se faz o movimentos em sentido horário e anti-horário em relação ao eixo. Geralmente, se escolhe bidirecional quando o sistema está configurado em modo de corpo rígido (um disco girando livremente) e operando em malha aberta (PARKS, 1999).

O eixo é acionado por um motor DC ligado através de uma correia rígida (flexibilidade de tensão negligenciável) e sistema de polia com uma proporção de redução de velocidade de 3:1. Um codificador localizado na base do eixo mede o deslocamento angular,  $\theta_1$  do primeiro disco. O segundo disco é conectado ao seu codificador por uma correia/polia rígida com uma relação de velocidade de 1:1. Há um codificador em cada posição dos discos (PARKS, 1999).

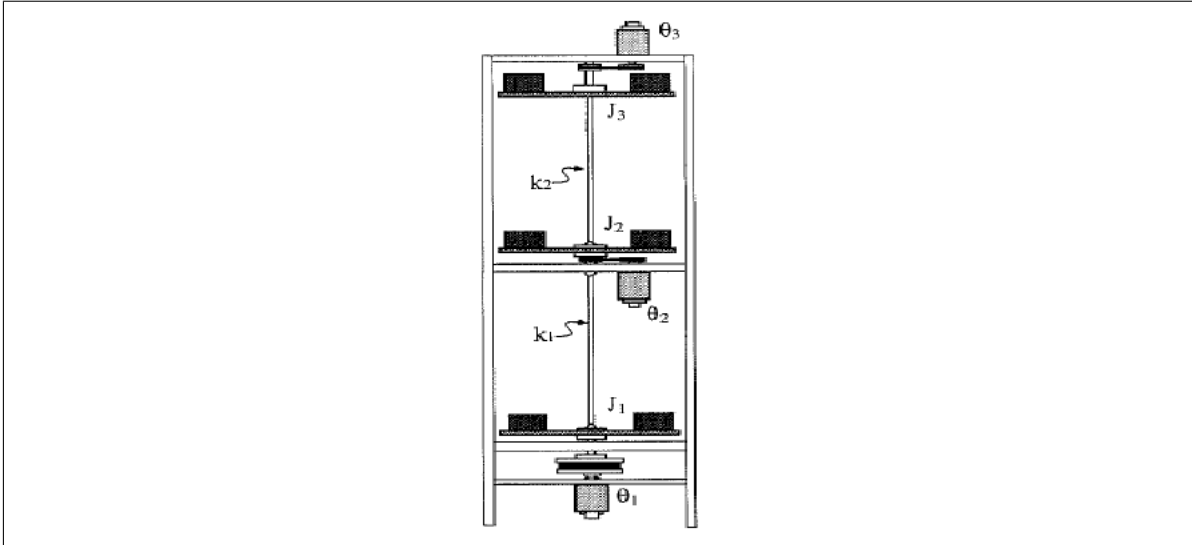
São apresentadas características da planta utilizando 3 discos com massas em todos os discos, mas as definições são válidas para outras configurações da planta. Logo após são definidas as equações para o modelo com um único disco utilizando massas, que é a configuração utilizada nesse trabalho.

As Figuras 9 e 10 mostram os parâmetros de cada item e o diagrama da planta com 3 discos, pode-se também mudar para 1 ou 2 caso desejado, alterando então seu grau de liberdade, também é possível alterar os valores de inércia colocando e ajustando os pesos nos discos.

Sendo:

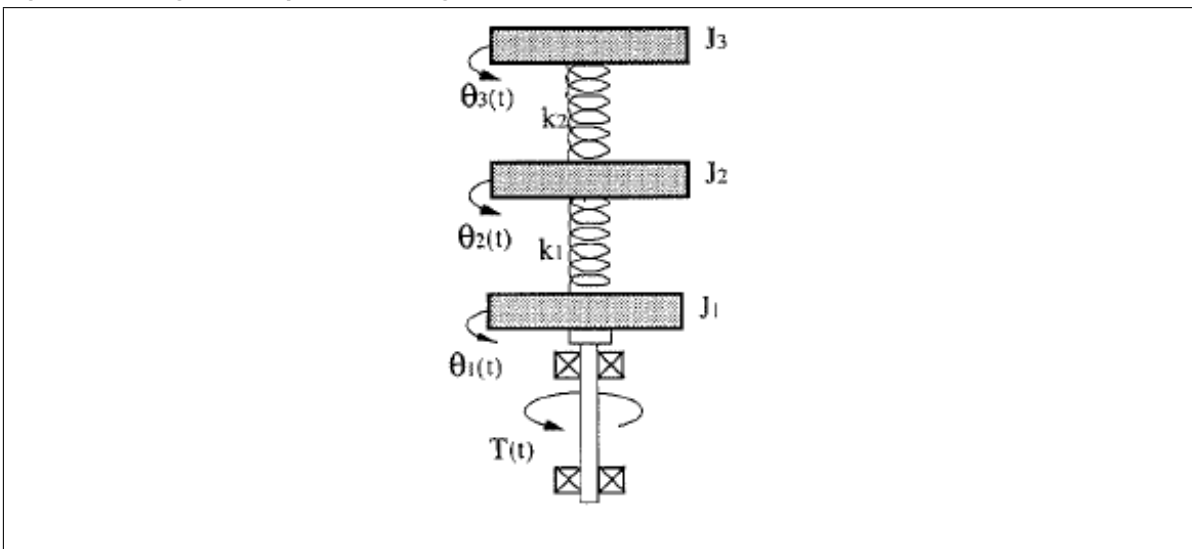
- $\theta_1, \theta_2, \theta_3$ : deslocamentos angulares dos discos 1, 2 e 3, respectivamente;
- $T(t)$ : torque aplicado ao disco 1;
- $J_1, J_2, J_3$ : momentos de inércia dos discos;

Figura 9 – Configuração com 3 discos livres.



Fonte: (PARKS, 1999).

Figura 10 – Diagrama da planta com 3 graus de liberdade, 3 discos livres, modelo 205a.



Fonte: (PARKS, 1999).

- $c_1, c_2, c_3$ : coeficientes de atrito viscoso;
- $k_1, k_2$ : constantes de mola;

As equações de movimento no domínio do tempo para o modelo de 3 graus de liberdade para os discos 1,2 e 3 são (PARKS, 1999):

$$J_1 \ddot{\theta}_1 + c_1 \dot{\theta}_1 + k_1 \theta_1 - k_1 \theta_2 = T(t) \quad (13)$$

$$J_2 \ddot{\theta}_2 + c_2 \dot{\theta}_2 + (k_1 + k_2) \theta_2 - k_1 \theta_1 - k_2 \theta_3 = 0 \quad (14)$$

$$J_3\ddot{\theta}_3 + c_3\dot{\theta}_3 + k_2\theta_3 - k_2\theta_2 = 0 \quad (15)$$

As Equações (13)-(15) também podem ser expressas em uma relação de espaços de estados:

$$\begin{aligned} \dot{x} &= Ax + BT(t) \\ y &= Cx \end{aligned} \quad (16)$$

sendo

$$x = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1}{J_1} & \frac{-c_1}{J_1} & \frac{k_1}{J_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{J_2} & 0 & \frac{-(k_1+k_2)}{J_2} & \frac{-c_2}{J_2} & \frac{k_2}{J_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_2}{J_3} & 0 & \frac{-k_2}{J_3} & \frac{-c_3}{J_3} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{J_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_6 \end{bmatrix}$$

e  $C_i = 1 (i = 1, 2, 3, 4)$  quando  $x_i$  é uma saída, sendo igual a 0 caso contrário.

Encontram-se, então, com as transformadas de Laplace das Eq. (13) a (15) as funções de transferência:

$$\frac{\theta_1(s)}{T(s)} = \frac{N_1(s)}{D(s)}$$

$$\frac{\theta_2(s)}{T(s)} = \frac{N_2(s)}{D(s)}$$

$$\frac{\theta_3(s)}{T(s)} = \frac{N_3(s)}{D(s)}$$

Sendo

$$N_1(s) = J_2 J_3 s^4 + [J_2 c_3 + J_3 c_2] s^3 + [J_2 k_2 + c_2 c_3 + J_3 k_1 + J_3 k_2] s^2 + [c_2 k_2 + c_3 k_1] + c_3 k_2] s + k_1 k_2 \quad (17)$$

$$N_2(s) = k_1 [J_3 s^2 + c_3 s + k_2] \quad (18)$$

$$N_3(s) = k_1 k_2 \quad (19)$$

$$D(s) = J_1 J_2 J_3 s^6 + [J_1 J_2 c_3 + J_1 J_3 c_2 + J_2 J_3 c_1] s^5 + [J_1 (J_2 k_2 + J_3 k_1 + J_3 k_2 + c_2 c_3) + J_2 (J_3 k_1 + c_1 c_3) + J_3 c_1 c_2] s^4 + [J_1 (c_2 k_2 + c_3 k_1 + c_3 k_2) + J_2 (c_1 k_2 + c_3 k_1) + J_3 (c_1 k_1 + c_1 k_2 + c_2 k_1) + c_1 c_2 c_3] s^3 + [(J_1 J_2 J_3) k_1 k_2 + c_1 (c_2 k_2 + c_3 k_1 + c_3 k_2) + c_2 c_3 k_1] s^2 + [(c_1 + c_1 + c_1) k_1 k_2] s \quad (20)$$

As quais também podem ser expressas como:

$$\frac{\theta_1(s)}{T(s)} = \frac{K_1 (s^2 + 2\zeta_{z1}\omega_{z1}s + \omega_{z1}^2)(s^2 + 2\zeta_{z2}\omega_{z2}s + \omega_{z2}^2)}{s(s + c^*)(s^2 + 2\zeta_{p2}\omega_{p2}s + \omega_{p2}^2)(s^2 + 2\zeta_{p3}\omega_{p3}s + \omega_{p3}^2)} \quad (21)$$

$$\frac{\theta_2(s)}{T(s)} = \frac{K_2 (s^2 + 2\zeta_z\omega_zs + \omega_z^2)}{s(s + c^*)(s^2 + 2\zeta_{p2}\omega_{p2}s + \omega_{p2}^2)(s^2 + 2\zeta_{p3}\omega_{p3}s + \omega_{p3}^2)} \quad (22)$$

$$\frac{\theta_3(s)}{T(s)} = \frac{K_3}{s(s + c^*)(s^2 + 2\zeta_{p2}\omega_{p2}s + \omega_{p2}^2)(s^2 + 2\zeta_{p3}\omega_{p3}s + \omega_{p3}^2)} \quad (23)$$

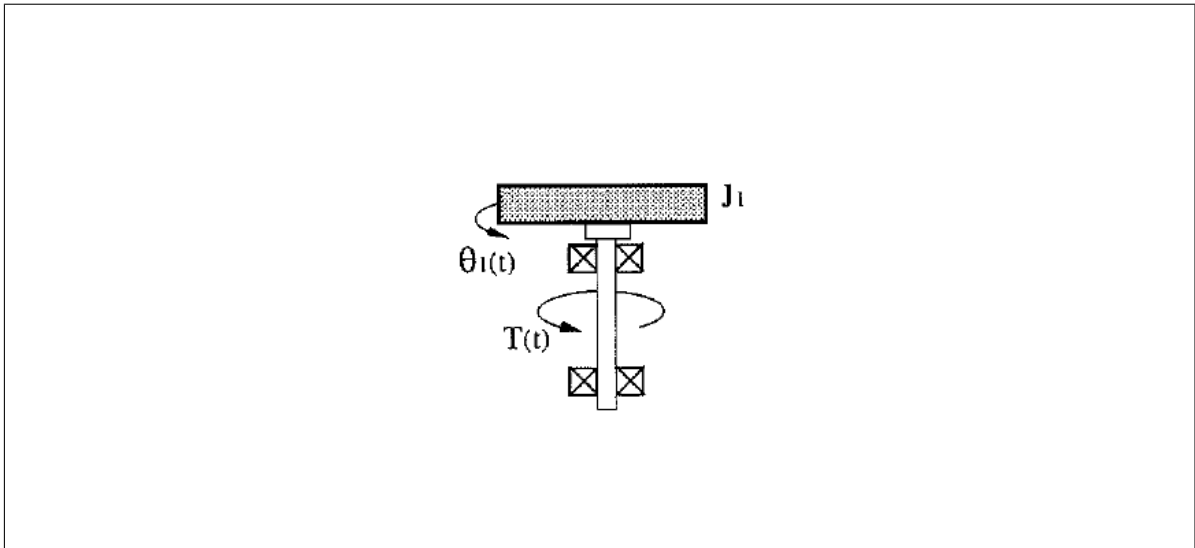
onde  $\omega_i$  e  $\zeta_i$  são as respectivas frequências naturais e taxas de amortecimento do  $i$ -ésimo disco, e os ganhos ordenados são nominalmente iguais a  $K_1 = 1/J_1$ ,  $K_2 = k_1/J_1 J_2$  e  $K_3 = k_1 k_2/J_1 J_2 J_2$ .

A Figura 11, demonstra o diagrama para o sistema com apenas o primeiro disco.

A equação dinâmica de movimento para modelo de 1 grau de liberdade, tomando  $k_1 = 0$ , é

$$J_1 \ddot{\theta}_1 + c_1 \dot{\theta}_1 = T(t) \quad (24)$$

Figura 11 – Diagrama da planta com 1 grau de liberdade.



Fonte: (PARKS, 1999).

Possuindo a função de transferência,

$$\frac{\theta_1(s)}{T(s)} = \frac{1}{J_1 s^2 + c_1 s} \quad (25)$$

Em espaços de estados é expressa como

$$x = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ \frac{-k_1}{J_1} & \frac{-c_1}{J_1} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{J_1} \end{bmatrix}, C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}$$

## 4 METODOLOGIA

Neste capítulo, é abordada a metodologia de elaboração desta pesquisa, como os materiais utilizados e os métodos de desenvolvimento.

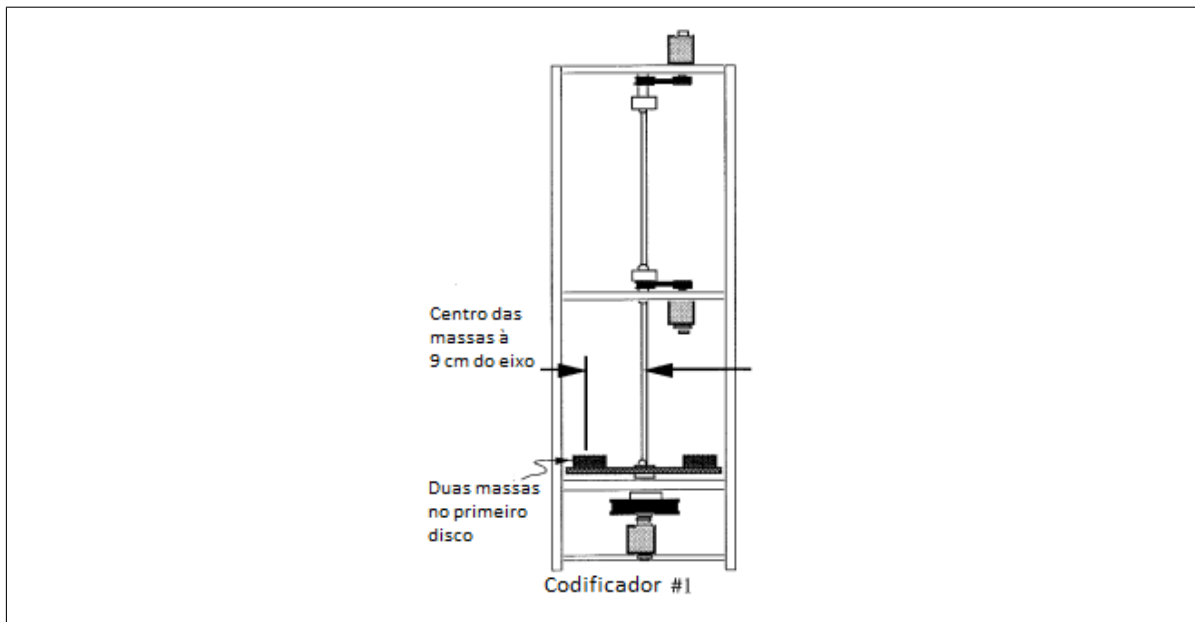
O sistema torcional possui várias configurações possíveis, dependendo dos discos e pesos utilizados, conforme seção anterior.

A configuração utilizada neste trabalho é mostrada na Figura 12, utilizando apenas o disco inferior com duas massas de 500 gramas distribuídas a 9cm do eixo, cuja FT é demonstrada na Eq. (26).

$$\frac{\theta_c(s)}{U_c(s)} = \frac{1000,5765}{s^2 + 0,8934s} \quad (26)$$

A função do sistema foi identificada conforme instruções do manual (PARKS, 1999), considerando entrada e saída em *counts*. O objetivo é encontrar uma função de transferência que represente um controlador para o sistema para o controle de posição do disco inferior.

Figura 12 – Configuração 1.



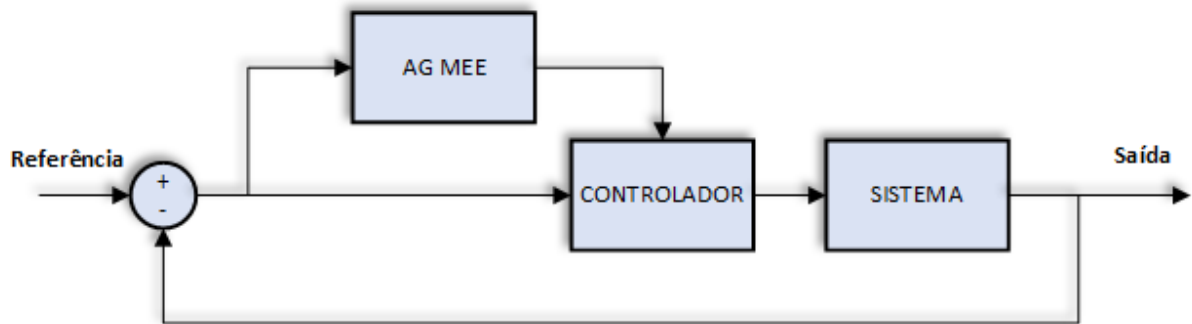
Fonte: Adaptado de (PARKS, 1999).

### 4.1 MÉTODO PARA OBTENÇÃO DO CONTROLADOR

O método de busca para controladores na planta torcional consiste em utilizar um algoritmo genético com função custo sobre o sinal de erro e algumas restrições sobre os controladores encontrados. O Fluxograma 2 representa a estrutura do sistema de busca, no AG

MEE é feito a busca de controladores, que passam a serem testados com sistema em malha fechada.

**Fluxograma 2 – Fluxograma da estrutura do método de busca do controlador.**

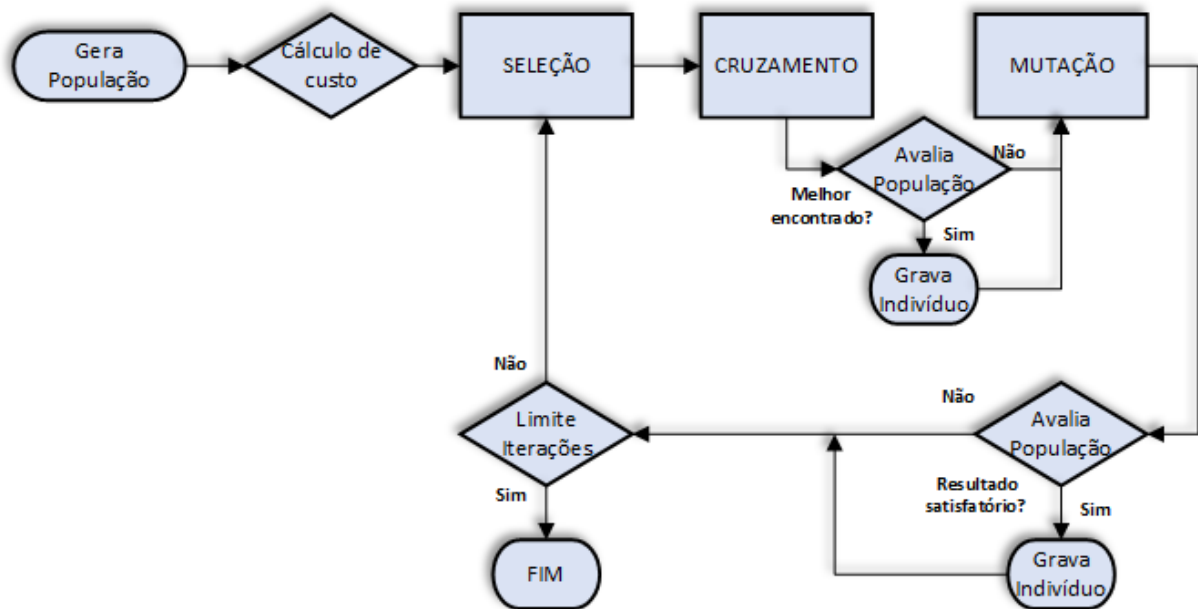


Fonte: Autoria Própria

O algoritmo foi criado para encontrar um controlador para qualquer função de transferência e, posteriormente, ajustado conforme necessidade para este trabalho.

O algoritmo foi elaborado no software MATLAB e obedece o Fluxograma 3.

**Fluxograma 3 – Fluxograma do algoritmo genético elaborado.**



Fonte: Autoria Própria

O cálculo do MEE entra como função objetivo na etapa de cálculo de custo e nas etapas de avaliação da população, depois do cruzamento e da mutação. O melhor indivíduo é salvo como referência para a busca dos próximos, sendo aceito apenas melhores que ele.

#### 4.1.1 Geração de indivíduos

Os indivíduos do algoritmo são gerados pelo método de inicialização aleatória uniforme, visto no Capítulo 2, de acordo com a maior ordem do polinômio do modelo a ser controlado. Para modelos de segunda ordem, são gerados seis valores para cada indivíduo, sendo três para o numerador do controlador e três para o denominador, resultando em um controlador de segunda ordem também. A estrutura do controlador é dada pela Eq. (27), sendo os termos  $a_2, a_1, a_0$  e  $b_2, b_1, b_0$  encontrados pelo AG.

$$G(s) = \frac{b_2s^2 + b_1s + b_0}{a_2s^2 + a_1s + a_0} \quad (27)$$

---

#### Algoritmo 1 – Pseudocódigo: Geração de indivíduos

---

- 1: Obter a ordem do sistema;
  - 2: Determinar o tamanho da população;
  - 3: Iniciar vetor de indivíduos com valores aleatórios;
  - 4:  $\text{indiv} \leftarrow A * \text{rand}(\text{população}, \text{ordem} * 2 + 2);$
- 

No Algoritmo 1, pode-se observar que  $A$  controla a banda de aleatoriedade dos indivíduos, quanto maior for esta variável maior são os valores iniciais possíveis para cada elemento do indivíduo. A função *rand* representa um gerador de números aleatórios no intervalo  $[0,1]$ , seguindo uma distribuição uniforme. O tamanho da população inicial considerado neste trabalho é de 50 indivíduos, para que haja uma boa variedade de elementos.

#### 4.1.2 Avaliação da população

Na etapa de avaliação é elaborada a função custo, também conhecida como função objetivo ou *fitness*. Esta é a equação matemática que avaliará os resultados obtidos pelo algoritmo e para decidir se o indivíduo é adequado. Os indivíduos gerados são analisados, posteriormente, para seguirem à etapa de seleção.

Para esta avaliação, é desejado minimizar a entropia do erro entre um sinal de referência (senoide) e a saída real do sistema realimentado com o controlador encontrado, com a finalidade de pesquisa sobre o funcionamento e eficiência da função MEE para esta tarefa.

O pseudocódigo da função utilizada, apresentada na seção 2.3.1, está demonstrado no Algoritmo 2 e Algoritmo 3.



**Algoritmo 2 – Pseudocódigo: Função custo**

- 
- 1: Definir indivíduos (compensadores) em formato de função de transferência;
  - 2: Implementar os compensadores em malha fechada;
  - 3: Obter a saída a uma entrada de referência;
  - 4: Obter o vetor de erros;
  - 5: Calcular a MEE;
  - 6:  $\text{Fit} \leftarrow \text{renyi2}(\text{abs}(e), \sigma)$ ;
- 

No código, a variável *Fit* recebe os valores de entropia encontrados para cada controlador após seu cálculo ser feito pela função, sendo os parâmetros necessários para essa função: o valor do tamanho do *kernel*  $\sigma$  e valores absolutos do vetor de erros  $\text{abs}(e)$ , obtido pela diferença entre o sinal de referência e o sinal encontrado (saída). Para obtenção desses sinais, aplica-se uma função degrau ou senoidal, por exemplo. O sinal de referência utilizado foi um sinal senoidal de 3 segundos discretizados em 300 amostras.

A função chamada no Algoritmo 2 calcula entropia do erro, e seu pseudocódigo é apresentado no Algoritmo 3.

**Algoritmo 3 – Pseudocódigo: MEE**

- 
- 1: **função**  $H_2 = \text{renyi2}(e, \sigma)$
  - 2: Calcular o kernel gaussiano;
  - 3:  $\kappa_\sigma(e - e_i) \leftarrow \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(e - e_i)^2}{2\sigma^2}\right)$ ;
  - 4: Calcular o Potencial de informação;
  - 5:  $IP(X) \leftarrow \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N \kappa_{\sqrt{2}\sigma}(e_j - e_i)$ ;
  - 6: Obter o valor da entropia;
  - 7:  $H_2(X) \leftarrow -\log IP(X)$ ;
- 

Para um melhor desempenho da função objetivo, foi adicionado ao sinal de referência um pequeno ruído gaussiano com variância de 0.01. Como visto nos artigos relacionados sobre o MEE, ele possui melhor desempenho se tratando de sinais gaussianos e impulsivos.

**4.1.3 Seleção de indivíduos**

Após serem avaliados, passam por uma etapa de seleção, onde os indivíduos que obtiveram menores valores de entropia recebem uma maior probabilidade de serem escolhidos para posteriormente sofrerem alterações genéticas necessárias, como o cruzamento e mutação, sendo que cada indivíduo pode ser escolhido mais de uma vez. O método utilizado para esta etapa foi o de seleção por ranking, visto na Seção 2.2. O Algoritmo 4 apresenta o método para a seleção utilizado neste trabalho.

**Algoritmo 4 – Pseudocódigo: Seleção de indivíduos mais aptos**


---

```

1: Estabelecer faixa de probabilidades, sendo a menor entropia com maior probabilidade;
2: Aplicar seleção por ranking na faixa de probabilidades;
3: para i=1 até tamanho_população faça
4:   sorteio de número aleatório na faixa de probabilidades;
5:   encontrar indivíduo sorteado;
6:   nova população recebe indivíduo sorteado;
7: fim para
8: Trocar população pelos selecionados;
9:   população  $\leftarrow$  nova população;

```

---

## 4.1.4 Cruzamento de indivíduos

No cruzamento, faz-se uma repetição no valor da metade da quantidade total da população. A cada iteração são escolhidos aleatoriamente dois elementos da população que são chamados de pais. Com esses dois pais geram-se dois filhos. Cada filho recebe uma porcentagem aleatória dos valores dos genes do primeiro pai e o restante da porcentagem ele recebe do outro pai. O segundo filho gerado nessa iteração recebe o inverso do filho 1, isto é, a porcentagem dos genes que o filho 1 recebeu do pai 2 é a mesma porcentagem dos genes que o filho 2 receberá do pai 1 e o restante do pai 2.

O pseudocódigo de cruzamento pode ser observado no Algoritmo 5.

**Algoritmo 5 – Pseudocódigo: Cruzamento de indivíduos**


---

```

1: para i=1 até tamanho_população/2 faça
2:   sortear aleatoriamente entre a população, uniformemente;
3:   pai1  $\leftarrow$  primeiro sorteado;
4:   pai2  $\leftarrow$  segundo sorteado;
5:   alpha  $\leftarrow$  porcentagem aleatória;
6:   filho1  $\leftarrow$  alpha*pai1+(1-alpha)*pai2;
7:   filho2  $\leftarrow$  (1-alpha)*pai1+alpha*pai2;
8:   novapop  $\leftarrow$  [novapop; filho1; filho2];
9: fim para
10:  população  $\leftarrow$  novapop;

```

---

Os filhos gerados são adicionados à nova população (variável *novapop*). Logo eles são todos testados novamente e os piores resultados são eliminados. O melhor indivíduo é então gravado e seu valor se torna o mínimo encontrado até então. A nova população passa para a etapa de mutação.

#### 4.1.5 Mutação de indivíduos

A mutação ocorre com uma probabilidade pré-definida, chamada aqui de "pm" (porcentagem de mutação), definida em 25%, onde cada cromossomo dos indivíduos pode ou não sofrer alteração de seus valores. Faz-se com que o cromossomo (monômios do controlador) seja multiplicado por um valor aleatório, no caso  $(1 + A * randn)$ , sendo  $A$  um valor para a margem de distribuição da aleatoriedade do  $randn$ . Outras operações podem ser utilizadas para efetuarem a mutação. Logo depois os indivíduos são testados e novamente avaliados. O código de mutação é apresentado no Algoritmo 6.

---

#### Algoritmo 6 – Pseudocódigo: mutação de indivíduos

---

```

1: pm  $\leftarrow$  define porcentagem de mutação;
2: Percorrer todos elementos de todos os indivíduos;
3: para i=1 até tamanho_população faça
4:   para j= 1 até último cromossomo faça
5:     se pm>rand então
6:       indiv(i,j)  $\leftarrow$  indiv(i,j)*(1+A*randn);
7:     fim se
8:   fim para
9:   Avaliar indivíduo modificado;
10: fim para

```

---

#### 4.1.6 Restrições de respostas

Entre as etapas que avaliam as respostas, foi necessária a adição de restrições para as funções encontradas, restringindo os possíveis controladores. Foi necessário estabelecer uma faixa de norma  $H_2$  da malha fechada para que as funções encontradas sejam aceitas, visto que as funções obtidas somente pela minimização da entropia encontravam ótimas respostas, porém, à medida que se minimizava a entropia do erro, as normas se tornavam cada vez mais elevadas ou infinitas, o que desestabilizava o sistema. Se a função em malha fechada encontrada possuir um valor elevado de norma, sua resposta ao impulso apresentará um tempo de resposta muito rápido e o valor de pico elevado, então o sistema torcional não é capaz de responder a este controle devido às suas limitações de tempo de resposta e esforço, assim desabilitando o sistema por segurança. Um exemplo desse problema é analisado nos resultados no Capítulo 5.

Também foi necessária a verificação das raízes dos denominadores da função de transferência dos controladores, aceitando-se somente funções com raízes com partes reais negativas.

Sempre que são testados os indivíduos, se algum gera valores infinitos ou sem valores ("NaN"), eles são substituídos por outros até que todos sejam válidos. Se nenhum elemento

obtiver o valor satisfatório, a nova população retorna novamente para a etapa de seleção e percorre todas as etapas mais uma vez, isso ocorre até que se encontre um valor aceitável ou o número máximo de iterações seja atingido.

Caso não se saiba o valor ideal e o melhor possível para parada, utiliza-se um método de atualização de precisões, onde a cada final de iteração ou quando se acha um indivíduo adequado, atualiza-se a precisão salvando o valor de menor entropia, onde o AG então só aceita salvar outro resultado se ele for um valor melhor que o achado anteriormente.

## 4.2 METODOLOGIA PARA A ANÁLISE DE RESULTADOS

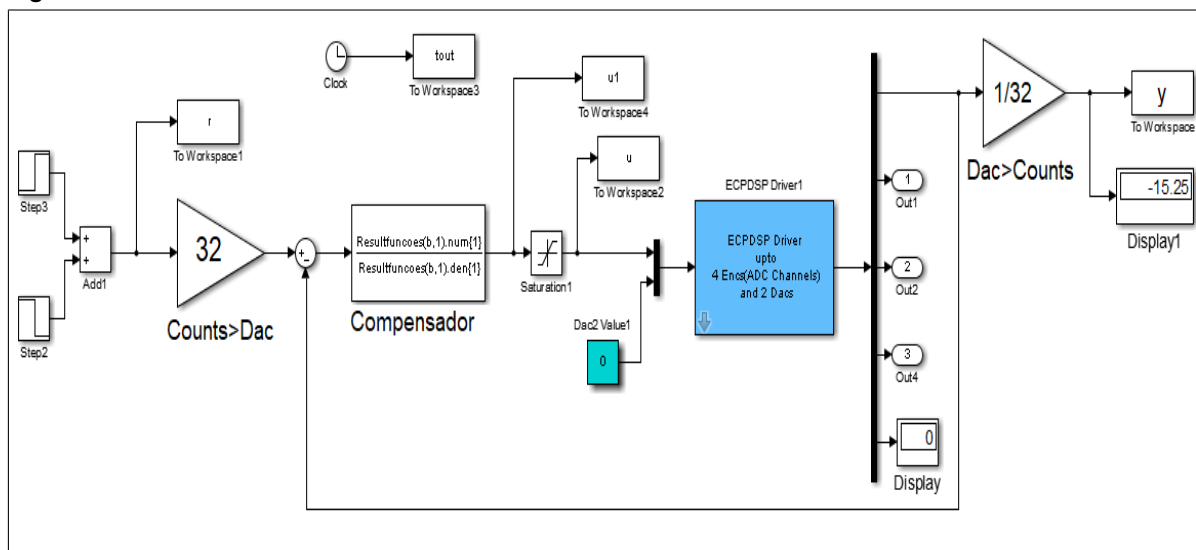
Após a obtenção de resultados, é verificado se o controle está funcionando, através de simulação no MATLAB, utilizando o comando de entrada degrau ou com a função *lsim* para sinal senoidal, assim analisando-se os gráficos gerados.

### 4.2.1 Proposta para a validação dos resultados

Para a proposta de validação no sistema físico, foi utilizado o *software Simulink*, utilizando-se do bloco do sistema torcional fornecido pela *ECP Control Products*. A configuração deve ser feita conforme manual de instruções para que haja comunicação entre o *Simulink* e o DSP instalado no computador.

O sistema elaborado pode ser observado na Figura 13.

**Figura 13 – Sistema de blocos em malha fechada em *Simulink*.**



**Fonte: Autoria Própria.**

Diferentemente do *software ECP control executive* que reconhece 360 graus como 16.000 counts, o sistema no *Simulink* utiliza 512.000 como 360 graus. Assim, para se utilizar o

valor em *counts* no *Simulink*, precisou-se adicionar o ganho no valor de 32 após a entrada do sistema e de 1/32 antes da saída, que é o fator de conversão utilizado na entrada do conversor DA, explicado no Capítulo 3.

Para a aquisição de dados e apresentação de gráficos, os dados do modelo são enviados para o *workspace* do Matlab após o término da execução do movimento desejado.

Uma comparação entre os métodos utilizando MEE e EQM é realizada nos resultados, como método de custo. Para a minimização do EQM é utilizada a integral aproximada do sinal do erro pelo método trapezoidal (função *trapz* no *MATLAB*), tendendo a diminuir o valor da integral. No algoritmo com EQM também são utilizadas as restrições aplicadas no AG, como a garantia de raízes negativas para o denominador do controlador e a norma entre os valores 1 e 8, restringindo assim as respostas encontradas para que não saturem quando aplicadas na prática no sistema torcional.

Neste capítulo, apresentou-se a elaboração e métodos utilizados para a construção do trabalho proposto, no capítulo seguinte serão demonstrados os resultados obtidos através dos testes realizados com simulações e posteriormente testados para ver seu desempenho na prática.

## 5 RESULTADOS E DISCUSSÕES

Neste capítulo, são analisados e discutidos os resultados, tanto em simulações quanto em testes práticos com a planta torcional.

### 5.1 RESULTADOS E ANÁLISE DA SATURAÇÃO DO SISTEMA

É apresentado nesta subseção um dos problemas encontrados durante os testes realizados. Algumas funções encontradas pelo AG possuíam ótimas e rápidas respostas simuladas, porém quando aplicado o resultado na validação, o sistema apresentava um ruído e se desabilitava devido à saturação. Após algumas análises, percebeu-se que sistemas com normas  $H_2$  menores da malha fechada, porém mais lentas, conseguiam ser reproduzidas e validadas, então notou-se que o sistema teria uma limitação de esforço de controle e velocidade. Para a solução foi necessária a adição de uma restrição, citada no capítulo anterior, como uma faixa aceitável para a norma  $H_2$  de 1 a 8, obtidas experimentalmente. Define-se maior do que 1, pois, caso seja muito pequeno, o sistema não demonstra reação à referência dada ou apresenta uma resposta muito lenta.

Um exemplo deste problema pode ser visto a seguir. O controlador em questão é apresentado na Eq. (28).

$$G(s) = \frac{0.2445s^2 + 0.2913s + 0.4236}{0.001414s^2 + 1.156s + 0.4464} \quad (28)$$

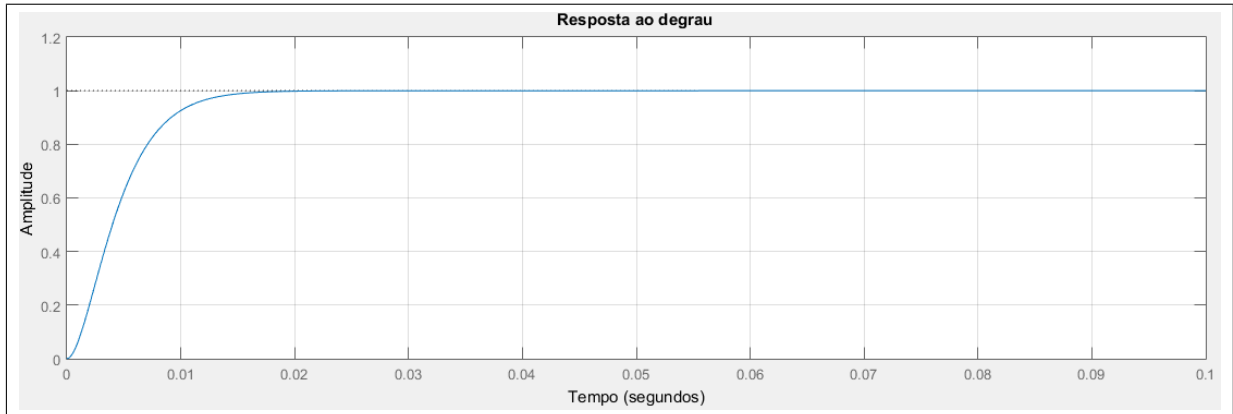
Aplicando-se a realimentação, modelo utilizado na Figura 1, obtêm-se a função em malha fechada, Eq. (29). Obtendo-se as raízes de seu denominador, notou-se valores elevados e seu valor de norma foi de 10.2864.

$$Hmf(s) = \frac{244.6s^2 + 291.5s + 423.8}{0.001414s^4 + 1.157s^3 + 246.1s^2 + 291.9s + 423.8} \quad (29)$$

A Figura 14 mostra a entrada degrau para o sistema em malha fechada, o sistema atingiu a referência em 0,024 segundos e com pequeno erro de regime de 0,28%.

Outra maneira que se pode perceber o quão grande seria o esforço necessário para o sistema se estabilizar, é pela entrada ao impulso (Figura 15), nota-se que ao receber o impulso, seu valor se eleva para a amplitude de 155 em apenas 0,0023 segundos. Após análises de outras respostas, notou-se que, quando aplicado impulso nos sistemas e sua resposta possuir a combinação de um valor de pico elevado (aproximadamente maior que 90) e uma velocidade de reação menor que 0,005 segundos (aproximadamente), faz com que seja necessário muito

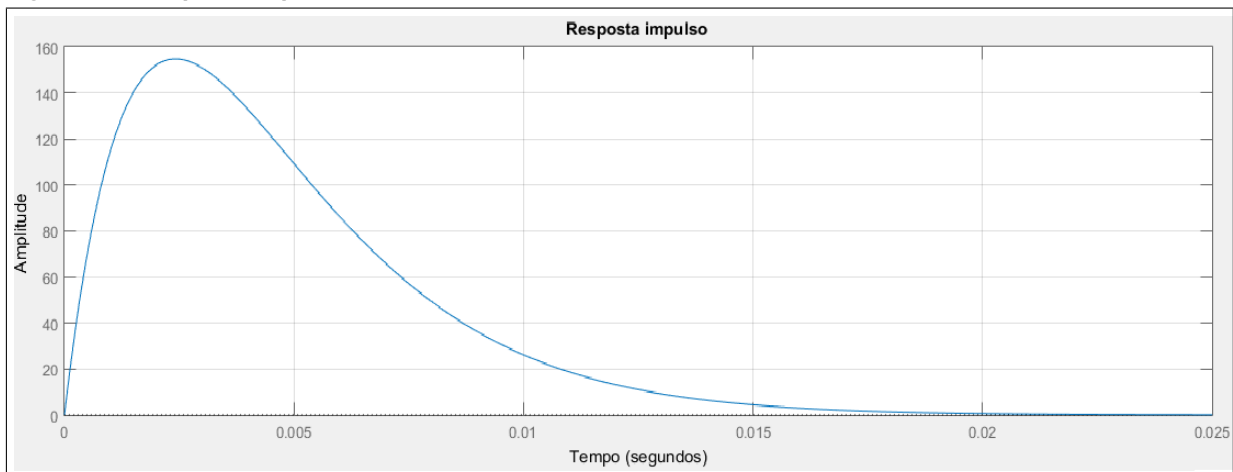
**Figura 14 – Resposta degrau sistema com norma alta, simulado no *Matlab*.**



**Fonte: Autoria Própria.**

esforço do sistema para correção, por isso quando o sistema é testado na prática, o equipamento apresenta um ruído, trepidações e desabilita o controle.

**Figura 15 – Resposta impulso sistema com norma alta, simulado no *Matlab*.**



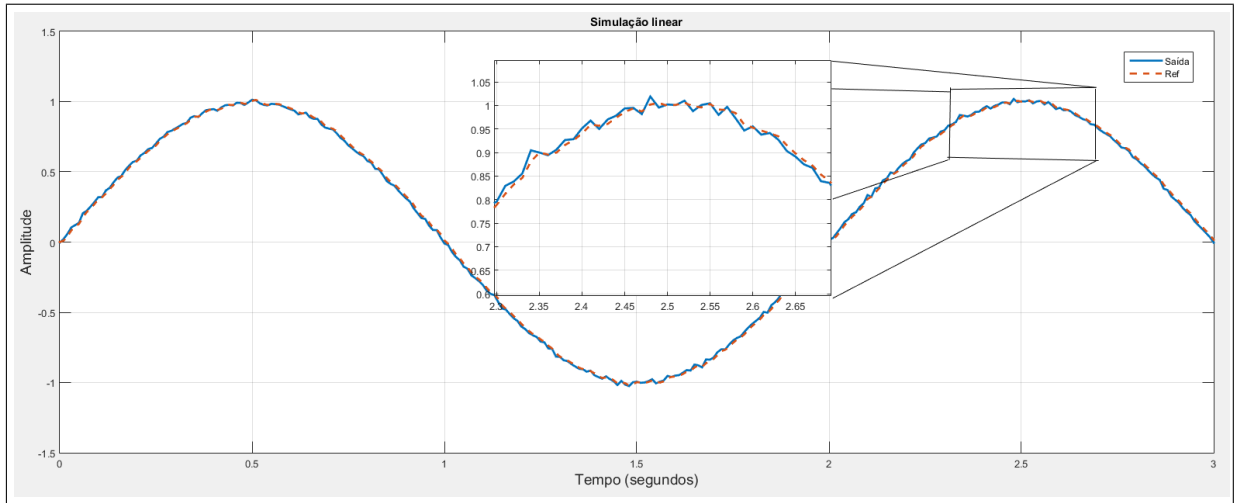
**Fonte: Autoria Própria.**

Na simulação linear do sinal senoidal (Figura 16), com aplicação de sinal ruidoso, pode-se ver a aproximação da saída do sistema com a referência, apresentando uma taxa média de erro absoluto de 0,74% (este seria um bom resultado prático caso a planta torcional não saturasse quando aplicado o controle). Na figura, o sinal contínuo representa a saída do sistema e o sinal seccionado a referência dada.

## 5.2 RESULTADOS PARA NÚCLEO DE TAMANHO 50

Foram testados diferentes valores para o tamanho do núcleo ( $\sigma$ ), a fim de se buscar melhores resultados. Encontrou-se várias respostas válidas para diferentes valores de núcleo,

Figura 16 – Simulação linear de sinal senoidal, sistema com norma alta, simulado no *Matlab*.



Fonte: Autoria Própria.

que são apresentadas a seguir. Um dos melhores resultados foi obtido com o valor de  $\sigma$  igual a 50, obtendo um valor de entropia mínima de 7,46960427372813 cujo controlador encontrado é o da Equação (30).

$$G(s) = \frac{0.1302s^2 + 2.563s + 0.6982}{0.007902s^2 + 1.888s + 0.1017} \quad (30)$$

A função em malha fechada com o sistema, Eq. (31).

$$Hmf(s) = \frac{130.3s^2 + 2564s + 698.6}{0.007902s^4 + 1.895s^3 + 132.1s^2 + 2564s + 698.6} \quad (31)$$

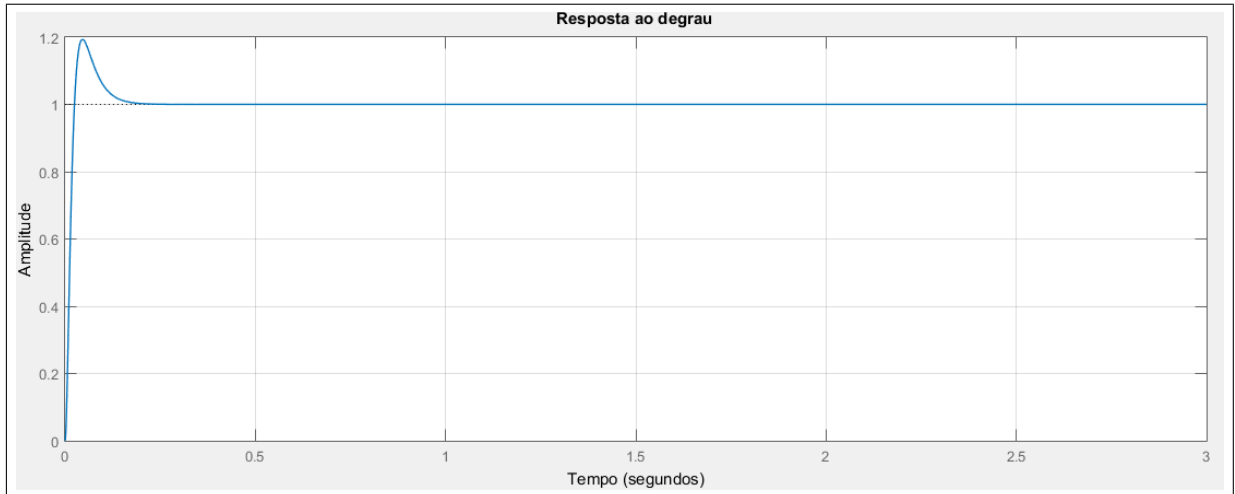
O valor de norma para esta função foi de 6.8916.

A entrada ao degrau desse sistema pode ser observada na Figura 17.

Observa-se na simulação da entrada ao degrau o máximo sobressinal não ultrapassou 20% do valor de referência, chegando ao valor de 19,32% e um tempo de assentamento de 0,135 segundos.



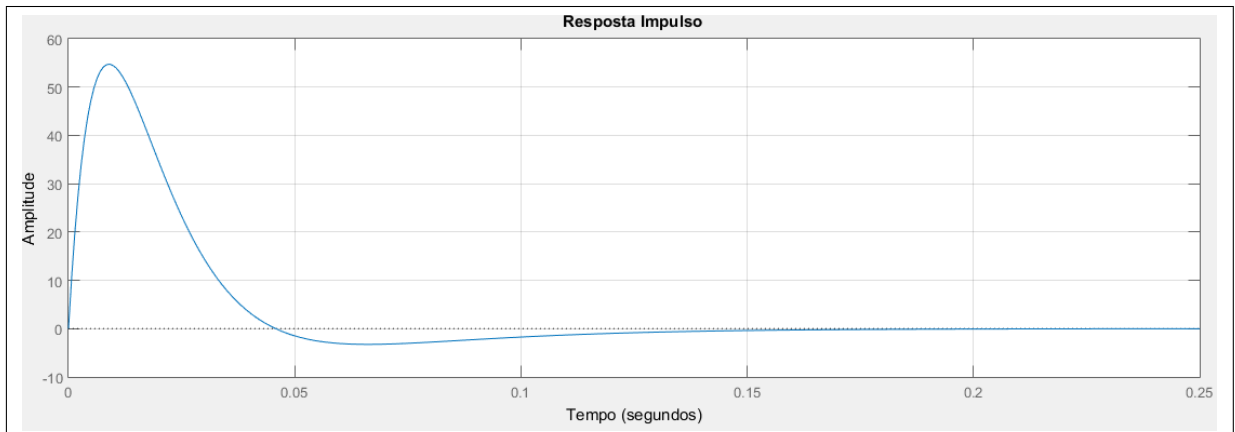
**Figura 17 – Resposta ao degrau, sigma 50, simulado no *Matlab*.**



**Fonte: Autoria Própria.**

A resposta impulso (Figura 18) desse modelo não atingiu um valor de pico tão alto quanto o apresentado anteriormente, alcançou a amplitude de 54.73, em um tempo de 0,009 segundos, o que não causou saturação quando aplicada na prática no sistema torcional.

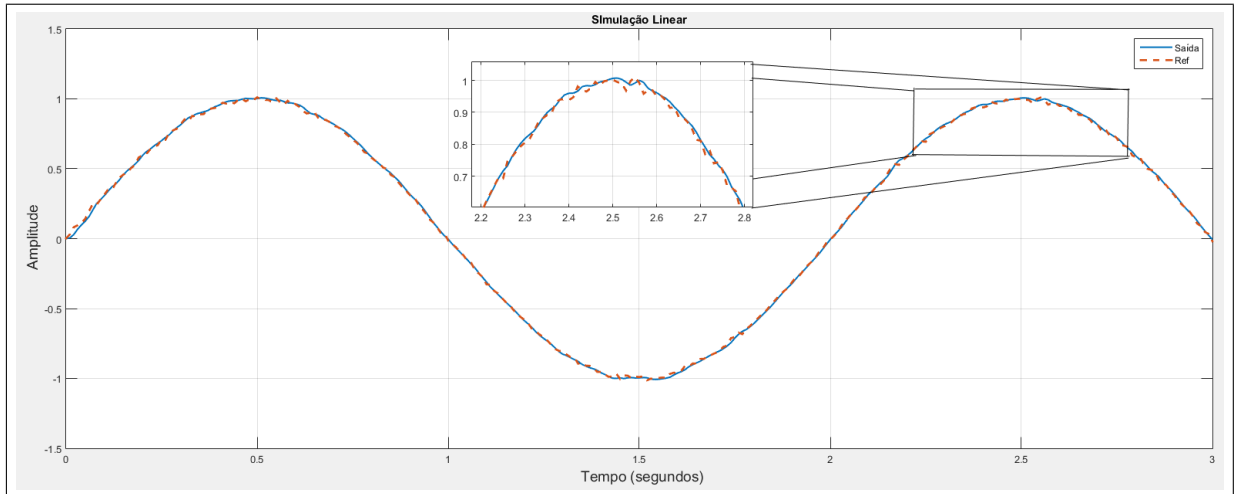
**Figura 18 – Resposta impulso, sigma 50, simulado no *Matlab*.**



**Fonte: Autoria Própria.**

Simulando e comparando a resposta com a referência em um sinal senoidal no *Matlab*, obtêm-se o resultado mostrado na Figura 19.

**Figura 19 – Resposta do sinal senoidal, sigma 50, simulado no *Matlab*.**



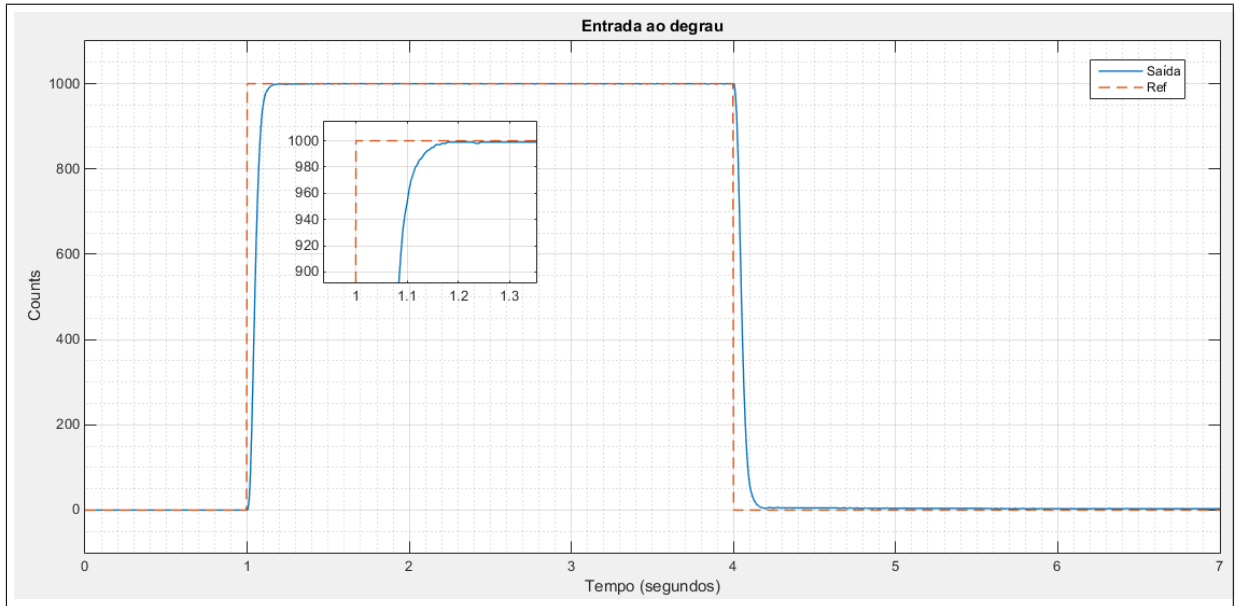
**Fonte: Autoria Própria.**

Pelo gráfico ampliado, nota-se que o erro teve uma margem um pouco maior, tendo como taxa média de erro absoluto 0,99% de amplitude, um pouco maior que a do controlador demonstrado anteriormente. Entretanto, este é compatível às limitações do sistema torcional sem saturar o equipamento.

Observando o resultado satisfatório obtido, foi implementada a função de controle diretamente na planta, através de realimentação explicada no Capítulo 4, utilizando o modelo elaborado no *Simulink*.

Aplicando um sinal degrau de 1000 *counts* em 1 segundo e de -1000 em 4 segundos obteve-se o comportamento ilustrado na Figura 20. Nota-se que o controle apresentou um rápido tempo de subida, onde por volta de 0,2 segundos estabilizou-se em seu valor de referência, também não apresentou visualmente sobressinal, sendo de 0,41%, erro médio em regime permanente de 0,2591% e tempo de subida de 0,16 segundos.

**Figura 20 – Resposta ao degrau, sigma 50, validação real do sistema.**

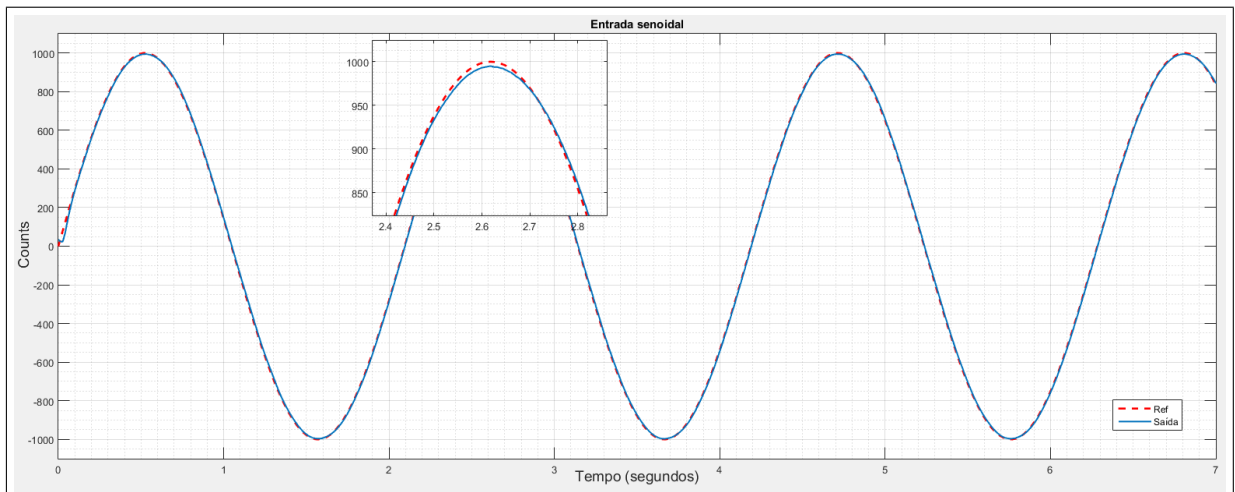


**Fonte: Autoria Própria.**

Os valores calculados para este gráfico são de 0,412% de sobressinal, 0,2591% de erro médio em regime permanente, calculado durante o último segundo do primeiro degrau e tempo de subida de 0,16 segundos.

A resposta senoidal também foi aplicada ao sistema de forma prática e seu resultado pode ser observado na Figura 21.

**Figura 21 – Resposta do sinal senoidal, sigma 50, validação real do sistema.**



**Fonte: Autoria Própria.**

Sem a aplicação da ferramenta de zoom não se pode notar muitos desvios do sinal de saída para a referência do sistema, sendo a taxa de erro médio de 0,623% .

### 5.3 RESPOSTA UTILIZANDO EQM COMO FUNÇÃO CUSTO

Em comparação feita utilizando o algoritmo com função custo aplicando o erro quadrático médio encontrou-se um resultado bem próximo ao encontrado pelo algoritmo utilizando MEE.

A função do controlador encontrado pode ser analisada na Equação (32), abaixo:

$$G(s) = \frac{0.01923s^2 + 0.3225s + 0.4588}{0.001001s^2 + 0.2574s + 0.2535} \quad (32)$$

A função em malha fechada com o sistema fica, Eq. (33).

$$Hmf(s) = \frac{19.24s^2 + 322.7s + 459.1}{0.001001s^4 + 0.2582s^3 + 19.73s^2 + 322.9s + 459.1} \quad (33)$$

A Figura 22 apresenta a resposta ao degrau da função encontrada utilizando EQM como função custo.

**Figura 22 – Resposta ao degrau, utilizando EQM, simulado no *Matlab*.**

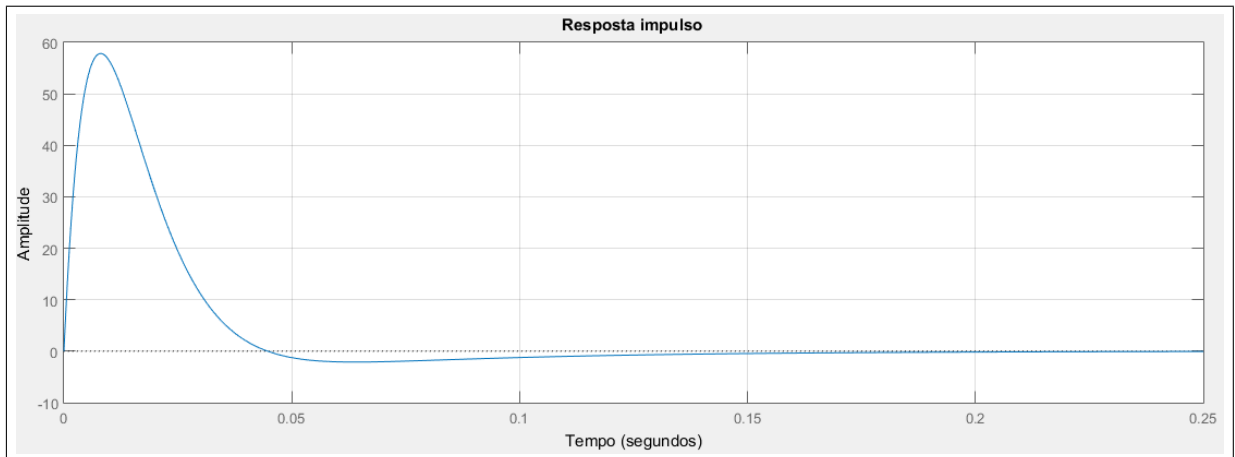


**Fonte: Autoria Própria.**

Notou-se que nesta houve o sobressinal de 15,23%, um pouco menor que a do outro método, mas com um assentamento ligeiramente mais lento de 0,16 segundos.

Aplicando a resposta ao impulso no sistema obtêm-se o gráfico da Figura 23. O valor de pico e velocidade da resposta impulso se aproxima da encontrada pelo algoritmo MEE, sendo os valores de 57,83 de amplitude máxima em um tempo de 0,008 segundos.

**Figura 23 – Resposta impulso, utilizando EQM, simulado no *Matlab*.**

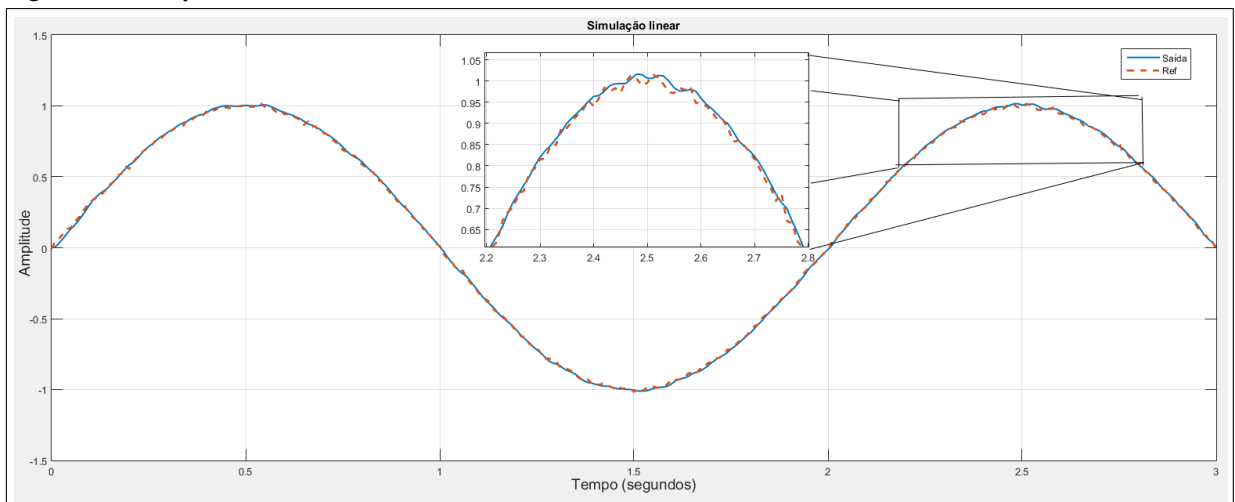


**Fonte: Autoria Própria.**

O valor de norma para esta função foi de 6.9074, bem próximo ao encontrado utilizando MEE.

Para análise da simulação linear do sinal senoidal, a Figura 24 foi gerada, e também semelhantemente a resposta utilizando MEE, o sinal de saída ficou bem próximo a referência dada, com taxa média de erro absoluto de 0.97%.

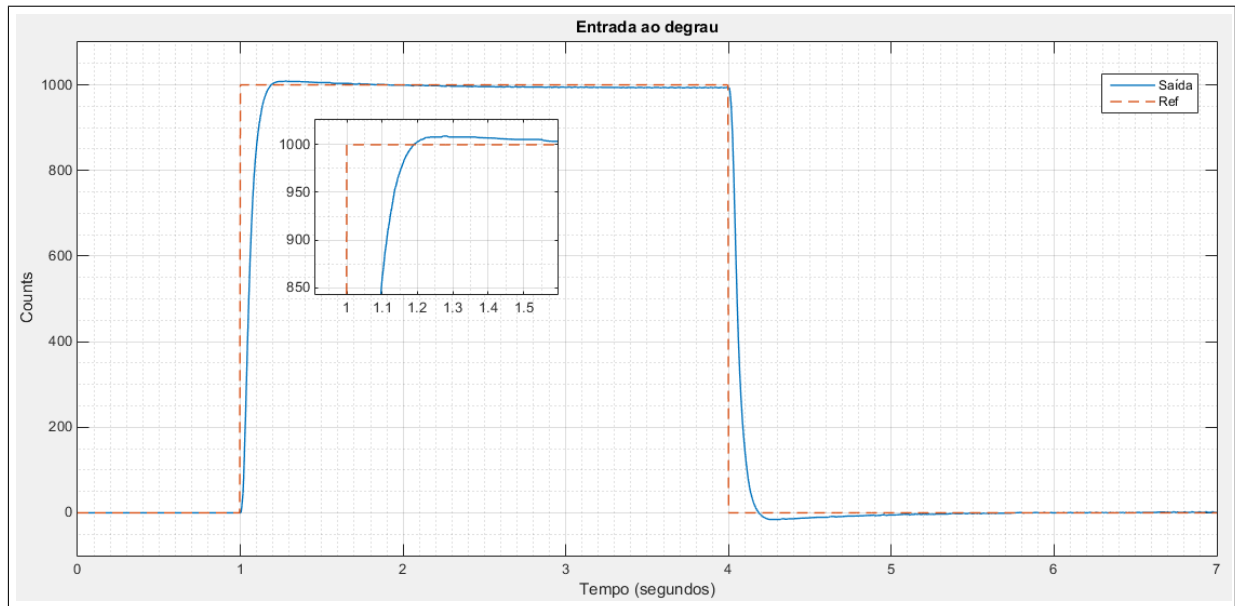
**Figura 24 – Resposta do sinal senoidal, utilizando EQM, simulado no *Matlab*.**



**Fonte: Autoria Própria.**

Na aplicação prática deste modelo, utilizando a mesma referência usada para o modelo anterior, através do *software Simulink*, obteve-se a resposta ao degrau, Figura 25.

**Figura 25 – Resposta ao degrau, utilizando EQM, validação real do sistema.**



**Fonte: Autoria Própria.**

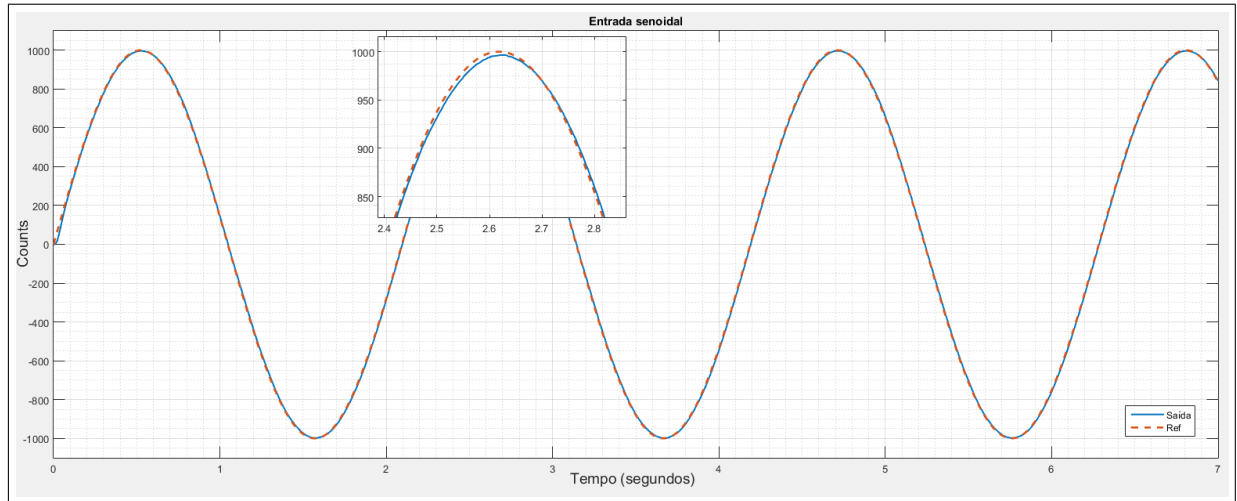
Nota-se um desempenho inferior ao do método MEE, pois ao contrário do outro modelo, este apresenta um sobressinal visível e um assentamento mais lento, porém não se pode dizer que o algoritmo seja pior, acredita-se que poderia obter uma resposta igual ou melhor que a encontrada pelo MEE, caso o algoritmo chegasse a valores parecidos para um controlador criado, para fim de dúvida, os dois controladores serão testados para ambas funções custo. Os dados calculados para esta entrada ao degrau podem ser analisados na Tabela 1, em comparação com o método utilizando MEE.

**Tabela 1 – Comparação entre MEE e EQM para resposta degrau**

	<b>Sobressinal</b>	<b>Erro médio em regime permanente</b>	<b>Tempo de subida</b>
EQM	0,946 %	0,5813 %	0,188 s
MEE	0,412 %	0,2591 %	0,16 s

A resposta aplicando um sinal senoidal, igualmente ao modelo anterior, pode ser vista na Figura 26, a seguir:

**Figura 26 – Resposta do sinal senoidal, utilizando EQM, validação real do sistema.**



**Fonte: Autoria Própria.**

A resposta senoidal não aparenta muita diferença visualmente do modelo encontrado utilizando MEE, pois está seguindo o sinal de referência sem alterações visíveis caso não seja aplicado a ferramenta de zoom, a taxa de erro médio foi de 0,6370%, bem próxima a encontrada com o MEE que foi de 0,623%.

Calculou-se a taxa média do erro total, entre a referência aplicada e o movimento real realizado, dos dois modelos aplicados na validação, tanto para a entrada ao degrau, quanto para senoidal. A Tabela 2 apresenta os valores encontrados.

**Tabela 2 – Erro médio total**

Erro médio (%)		
MEE	Degrau	1,7604
	Senoidal	0,623
EQM	Degrau	2,1229
	Senoidal	0,637

Na tabela, o erro na aplicação senoidal não obteve diferenças grandes, como era esperado, já que os gráficos estão bem parecidos visualmente. A diferença da entrada ao degrau se estendeu um pouco mais no caso do EQM.

Para fim de comparação, foi feito o uso das duas funções objetivo para os dois controladores encontrados, utilizando o mesmo sinal de referência. Os resultados da minimização da entropia do erro e da minimização do erro quadrático médio estão demonstrados na Tabela 3.

**Tabela 3 – Comparação entre valores de entropia quadrática e EQM**

	<b>MEE</b>	<b>EQM</b>
Controlador encontrado com MEE	7.4696042 <b>69715</b>	0.029 <b>39830</b>
Controlador encontrado com EQM	7.4696042 <b>70136</b>	0.0294 <b>2871</b>

De fato, para o sinal de referência usado, com a quantidade de 300 amostras no sinal, ambas as funções objetivo foram capazes de distinguir o melhor controlador encontrado, onde o que utilizou MEE obteve valores menores que o EQM para ambos os métodos.

O tempo de execução das funções custo MEE e EQM podem ser vistos na Tabela 4.

**Tabela 4 – Tempo médio de execução das funções custo**

<b>Tempo médio de execução</b>	<b>Segundos</b>
EQM	$3.2106 \times 10^{-05}$
MEE	$8.9828 \times 10^{-04}$

Cada função custo foi executada 100 vezes com o mesmo vetor de erro para ambas e com o valor de sigma 50 para a MEE, após foi encontrada o valor médio de cada um. Nota-se a vantagem do EQM em relação ao MEE, esta vantagem tende a aumentar com a maior discretização do sinal de referência. O hardware utilizado foi um notebook com processador I5-7200U 2.5 ~ 3.1 Ghz, 8 gb de memória ram DDR4, placa gráfica GeForce 940MX 2Gb VRAM dedicadas.

#### 5.4 RESULTADOS PARA DIFERENTES VALORES DE NÚCLEO

Para validação, fez-se comparações de diferentes valores de  $\sigma$  na função de minimização de entropia do erro. Foram usados valores diversos, porém apenas alguns convergiram com resultados satisfatórios e que puderam ser aplicados na prática na planta. Dentre os valores testados estão: 0,001; 0,01; 0,1 até 1,15 ao passo de 0,05 e de 5 à 60 ao passo de 5.

A Tabela 5 apresenta os resultados da simulação que convergiram e algumas características destes.

Percebe-se que quanto menores os valores de sigma, menor é a entropia. Porém a menor entropia só significa uma melhor resposta se tratando da utilização do mesmo valor de sigma, assim o melhor resultado tende a ter menor entropia do erro.

O tempo de subida, devido ao sinal de referência estar discretizado de 0 a 3 segundos com amostras a cada 0,01 segundo, fez com que não houvesse muitas variações, uma discretização com mais amostras solucionaria este problema, porém a execução da função objetivo iria ter custo computacional maior.

Analisando os resultados simulados escolheria-se o resultado de sigma 5 por ter menor sobressinal e menor média de erro em regime permanente, porém os resultados práticos



**Tabela 5 – Resultados simulados para diferentes valores de núcleo para resposta ao degrau**

<b>Sigma</b>	<b>Entropia</b>	<b>Norma</b>	<b>Máx. Sobressinal</b>	<b>Média erro reg. perm.</b>	<b>Tempo subida</b>
0.5	0,825948	6,02931	18,08%	0,001988%	0,04s
0.7	1,311271	7,32669	11,38%	0,001704%	0,03s
0.8	1,503881	7,83368	17,84%	0,002329%	0,02s
1	1,828608	7,88446	18,14%	0,000022%	0,02s
5	4,147678	7,55562	9,55%	0,000978%	0,02s
20	6,147684	7,53778	14,96%	0,010842%	0,02s
40	7,147676	7,94470	34,73%	0,002853%	0,03s
45	7,317603	7,50135	25,41%	0,001700%	0,03s
50	7,469604	6,89160	18,99%	0,005231%	0,03s

demonstram outra proposta.

A Tabela 6 apresenta os resultados que convergiram na prática e os detalhes para a resposta ao degrau.

**Tabela 6 – Resultados práticos para diferentes valores de núcleo para resposta ao degrau**

<b>Sigma</b>	<b>Máx. Sobressinal</b>	<b>Média erro reg. perm.</b>	<b>Tempo subida</b>
0.5	7,1531%	0,4597%	0,120s
0.7	3,2500%	1,1611%	0,388s
0.8	7,6969%	0,1020%	0,236s
1	13,6969%	0,1694%	0,192s
5	-2,0500%	2,0532%	1,620s
20	4,0500%	2,6054%	0,148s
40	0,9469%	0,2127%	0,156s
45	-1,0500%	1,0540%	1,484s
50	0,4125%	0,2591%	0,160s

Conforme demonstrado anteriormente o melhor resultado encontrado na prática, foi para o sigma de valor 50, devido a possuir o menor sobressinal, um baixo erro de regime permanente e um rápido tempo de subida. Os outros resultados também apresentaram desempenho satisfatório, porém os resultados de valores de sigma 5 e 45 não alcançaram o valor de referência ficando na média de -2 e -1 por cento da referência, respectivamente.

Como citado no capítulo 2, no artigo de Erdogmus e Principe (2006) um estudo experimental foi elaborado e verificado que o tamanho do kernel afeta muito menos a performance dos algoritmos ITL do que a estimativa de densidade.

## 6 CONSIDERAÇÕES FINAIS

Notou-se com a elaboração deste trabalho, que a função custo MEE se provou tão eficiente quanto utilizando EQM para sistemas de segunda ordem. Percebe-se também que, apenas utilizando a função custo, o algoritmo não apresentaria bons resultados sem a utilização das restrições tanto para MEE quanto para EQM.

O MEE apresentou resultados melhores quanto ao controlador encontrado, entretanto, analisando a Tabela 3, o algoritmo com EQM poderia resultar um controlador tão bom quanto, caso ocorresse de receber os valores aleatórios parecidos com o do MEE. Os dois métodos são capazes de buscar e distinguir controladores eficientes para esta configuração da planta torcional.

O tempo de execução para a função custo EQM se demonstrou menor, esta diferença tende a aumentar cada vez, tornando o MEE mais lento, com a maior discretização do sinal de referência.

Conclui-se que o MEE demonstrou capacidade de localizar um controlador eficiente para a planta torcional na sua configuração de segunda ordem, com pouca porcentagem de erro, como pôde ser visto no capítulo anterior através das respostas obtidas nos gráficos que foram adquiridos no teste prático.

Como propostas futuras, é esperado o estudo de respostas com sinais com maiores discretizações, porém necessita-se de um poder computacional maior. É interessante o uso de algoritmos mais otimizados para o MEE e também o experimento do algoritmo MCC. Outra proposta é utilizar e comparar outras configurações da planta torcional que possuem sistemas de maior ordem, assim como outros sistemas de plantas diferentes.

## REFERÊNCIAS

- BRUINSMA, N.A.; STEINBUCH, M. A fast algorithm to compute the hinf norm of a transfer function matrix. **Systems and Control Letters**, v. 14, n. 4, p. 287 – 293, 1990. ISSN 0167-6911. Citado na página 20.
- CRACIUN, Stefan et al. An information-theoretic approach to motor action decoding with a reconfigurable parallel architecture. **2011 Annual International Conference of the IEEE, (EMBC)**, Boston, MA, USA, p. 4621–4624, 2011. Citado 2 vezes nas páginas 13 e 25.
- DAVIS, Lawrence. **Handbook of Genetic Algorithms**. Van Nostrand Reinhold, New York: CNR Comp. Library, 1990. Citado na página 23.
- DAWKINS, Richard. **A escalada do monte improvável: uma defesa da teoria da evolução**. São Paulo, SP: Editora Companhia das Letras, 1996. Citado na página 23.
- ERDOGMUS, Deniz. **Information theoretic learning: Renyi's entropy and its applications to adaptive system training**. 2002. Tese (Doutorado) — University of Florida Gainesville, Florida, 2002. Citado 2 vezes nas páginas 25 e 27.
- ERDOGMUS, D.; PRINCIPE, J. C. From linear adaptive filtering to nonlinear information processing - the design and analysis of information processing systems. **IEEE Signal Processing Magazine**, v. 23, n. 6, p. 14–33, Nov 2006. ISSN 1053-5888. Citado 2 vezes nas páginas 26 e 56.
- GEYER-SCHULZ, Andreas. **Fuzzy rule-based expert systems and genetic machine learning**. Universidade de Michigan: Physica Verlag, 1997. v. 3. Citado 3 vezes nas páginas 21, 23 e 24.
- HAYKIN, S.S. **Adaptive Filter Theory**. Englewood Cliffs, NJ: Prentice Hall, 2002. (Prentice-Hall information and system sciences series). ISBN 9780130901262. Citado na página 12.
- HOLLAND, JH; GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. Boston, MA, USA: Addison-Wesley Publishing Company, 1989. ISBN 9780201157673. Citado 3 vezes nas páginas 21, 23 e 24.
- HOLLAND, John H. Adaptation in natural and artificial systems: An introductory analysis with application to biology, control, and artificial intelligence. **University of Michigan Press**, Ann Arbor, MI, 1975. Citado 3 vezes nas páginas 12, 21 e 22.
- LIU, Weifeng; POKHAREL, Puskal P; PRÍNCIPE, José C. Correntropy: Properties and applications in non-gaussian signal processing. **IEEE Transactions on Signal Processing**, v. 55, n. 11, p. 5286–5298, 2007. Citado 3 vezes nas páginas 13, 25 e 26.
- NISE, Norman S. **Engenharia de sistemas de controle**. 3. ed. Rio de Janeiro, RJ: LTC, 2002. ISBN 9788521613015. Citado 4 vezes nas páginas 16, 17, 18 e 19.
- OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Pearson, 2012. Citado 4 vezes nas páginas 16, 17, 19 e 20.
- PALHARES, Alvaro G. B. et al. **Notas de aula: EA 617 - Introdução à Simulação Analógica, Exp. 5 - Identificação de Parâmetros dos Equipamentos ECP: Resposta Temporal**. 2002. Disponível em: <[http://www.dt.fee.unicamp.br/~controle/ea617/pdf/exp5\\_1.pdf](http://www.dt.fee.unicamp.br/~controle/ea617/pdf/exp5_1.pdf)>. Acesso em: 18 de dezembro de 2017. Citado 3 vezes nas páginas 29, 30 e 32.

PARKS, Thomas R. **Manual For Model 205/205a Torsional Control System**: Instructor's edition. 1 Buckskin Court, Bell Canyon, CA, 1999. 189 p. Citado 8 vezes nas páginas 28, 29, 30, 31, 32, 33, 36 e 37.

PRINCIPE, Jose C. **Information theoretic learning: Renyi's entropy and kernel perspectives**. New York: Springer Science & Business Media, 2010. Citado 4 vezes nas páginas 12, 25, 26 e 27.

PRINCIPE, Jose C; XU, Dongxin; FISHER, John. Information theoretic learning. **Unsupervised adaptive filtering**, New York: Wiley, v. 1, p. 265–319, 2000. Citado na página 25.

RAO, Singiresu S. **Engineering optimization: theory and practice**. Roboken, New Jersey: John Wiley & Sons, 2009. Citado 2 vezes nas páginas 20 e 21.

REGO, Joilson B. A. **Extensões multidimensionais para correntropia e suas aplicações em estimativas robustas**. 2014. Tese (Doutorado) — Federal University of Rio Grande do Norte, Brazil, 2014. Citado 2 vezes nas páginas 25 e 26.

RÉNYI, Alfréd. On measures of entropy and information. In: **Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics**. Berkeley, California: University of California Press, 1961. Citado na página 25.

SANTAMARÍA, Ignacio; POKHAREL, Puskal P; PRINCIPE, José Carlos. Generalized correlation function: definition, properties, and application to blind equalization. **IEEE Transactions on Signal Processing**, v. 54, n. 6, p. 2187–2197, 2006. Citado na página 13.

SHANNON, Claude E; WEAVER, Warren. **The mathematical theory of communication**. Urbana: University of Illinois press, 1949. Citado na página 25.

SILVERMAN, Bernard W. **Density estimation for statistics and data analysis**. School of Mathematics University of Bath, UK: CRC press, 1986. v. 26. Citado na página 26.

TROFINO, Alexandre; COUTINHO, Daniel F.; BARBOSA, Karina A. Improved h2 and hinf conditions for robust analysis and control synthesis of linear systems. **Sba: Controle and Automação Sociedade Brasileira de Automatica**, scielo, v. 16, p. 427 – 434, 12 2005. ISSN 0103-1759. Citado na página 20.

WALL, Matthew. Galib: A c++ library of genetic algorithm components, 1999. **Mechanical Engineering Department, MIT**, 2008. Citado 2 vezes nas páginas 23 e 24.

WANG, Ren et al. A variable step-size adaptive algorithm under maximum correntropy criterion. In: **2015 International Joint Conference on Neural Networks (IJCNN)**. Killarney, Ireland: IEEE, 2015. p. 1–5. ISSN 2161-4393. Citado na página 14.

XU, Jian-Wu; ERDOGMUS, Deniz; PRINCIPE, Jose C. Minimum error entropy luenberger observer. In: IEEE. **Proceedings of the 2005 American Control Conference**. Portland, OR, USA, 2005. p. 1923–1928. ISSN 0743-1619. Citado 2 vezes nas páginas 13 e 25.

ZHANG, Jianhua et al. Minimum error entropy filter for fault detection of networked control systems. **Entropy**, Molecular Diversity Preservation International, v. 14, n. 3, p. 505–516, 2012. Citado 2 vezes nas páginas 14 e 25.

ZHAO, Songlin; CHEN, Badong; PRINCIPE, Jose C. Kernel adaptive filtering with maximum correntropy criterion. In: **The 2011 International Joint Conference on Neural Networks (IJCNN)**. San Jose, CA, USA: IEEE, 2011. p. 2012–2017. Citado 2 vezes nas páginas 12 e 13.

## **Anexos**

## ANEXO A – ALGORITMO AG MEE

### Algoritmo 1 – Código algoritmo genético criado no Matlab

```

1  clc; clear all; close all;
2
3  H = tf([1000.57653929],[1 0.8934 0]); %SISTEMA IDENTIFICADO
4
5  format long;
6  t=3; %tempo de simulação do sinal.
7  Ts = 0:0.01:t; %discretização
8  ref=gensig('sin',2,3,0.01); %geração de sinal de referência
9  ref=ref+0.01*randn(size(ref)); %Ruído
10 A=1; %aleatoriedade dos indivíduos
11 ordem = 2; %ordem do sistema
12 popini = 50; %população inicial
13 sigma = 50; %tamanho do kernel
14 precmax = 20; % Faixa de entropia.
15 precmin = -10;
16 normmax = 8; %Norma H2
17 normmin = 1;
18 pm = 0.25; %porcentagem de mutação
19 maxepoca = 100; %iterações
20
21 Result(200,1)=0; %inicialização de matrizes
22 Resultfuncoes(200,:)=tf(1,1);
23 G(popini,:)=tf(1,1);
24 Hmfg(popini,:)=tf(1,1);
25 xx=1;
26 Exx=zeros(1,2);
27
28 K=2; %quantidade de resets
29
30 for cont=1:K;
31 tic
32 randn('state',sum(A*randn*clock)); %reseta o gerador aleatório
33 indiv = A*rand(popini,ordem*2+2); %gera indivíduos
34 epoca=0;
35 E=zeros(popini,1);
36
37 % Funcao fitness

```

```

38     for i=1:size(indiv,1)
39         [E(i,:),G(i,:),Hmfg(i,)] = fitness_n(indiv,H,Ts,ref,sigma,i);
40     end
41     i=1; j=1;
42     while i ≠ (length(E(:,1))+1) %substituindo os NaN e inf
43         B=isnan(E(i,:)); C=isinf(E(i,:));
44         if B==1 || C==1
45             indiv(i,:)=A*rand(1,ordem*2+2); %substitui por outro
46             [E(i,:),G(i,:),Hmfg(i,)] = ...
47                 ↪ fitness_n(indiv,H,Ts,ref,sigma,i);
48         else i=i+1;
49         end
50     end
51     Ex(:,:)=50; %coloca-se um valor fora das precisoes para que se ...
52     ↪ passe nos "if's"
53     for i=1:size(indiv,1)
54         if ( (E(i,:)≠0) && (E(i,:) < precmax) && (E(i,:) > precmin) )
55             Ex(j,:)= E(i,:); %joga em novas variaveis pra ver a ...
56             ↪ minima depois
57             Ex(j,2)= i;
58             j=j+1;
59         end
60     end
61     j=1; Exx(:,:)=50;
62     for i=1:size(Ex,1)
63         [n]=(Ex(i,2));
64         if ((max(real(roots(G(n,:)).den{1}))) < -0.002) && ...
65             ↪ (norm(Hmfg(n,:)) < normmax) && (norm(Hmfg(n,:)) > ...
66             ↪ normmin) %verifica polos parta real negativa
67             Exx(j,:)= Ex(i,:);
68             j=j+1;
69         end
70     end
71     if ( (min(Exx(:,1)) < precmax) && ((min(Exx(:,1))) > precmin) )
72         [m,h] = min(Exx(:,1))
73         [k]=(Exx(h,2));
74         G(k,:)
75         epoca=epoca+1; disp(['Epoca= ' num2str(epoca)]); disp(' ');
76         Result(xx,:)= min(Exx(:,1));
77         Resultfuncoes(xx,:)=G(k,:);
78         salvar = ['Results sig ', num2str(sigma), '.mat'];
79         save (salvar, 'Result', 'Resultfuncoes', 'H', 'Ts', 'A', ...
80             ↪ 'popini', 'sigma', 'precmax', 'pm', 'maxepoca', 'cont', ...

```



```

        ↪ 'ref', 'epoca')
75     xx=xx+1;
76     precmax=min(Exx(:,1));
77     end
78
79     while epoca<maxepoca
80         % Seleção
81         disp('== Seleção ==')
82         fit=1./E;      %Para inverter, o menor número ser o melhor
83         aux = cumsum(fit); %Soma cumulativa do numero + anteriores
84         prob = aux/sum(fit); %Estabelece o intervalo de probabilidade.
85         for i=1:size(indiv,1) %Roleta no mesmo número de indivíduos.
86             x = rand(1,1); %Valor aleatório de 0 a 1.
87             P = find(prob>x); %Procura qual indivíduo possui o valor ...
                ↪ sorteado.
88             nindiv(i,:)=indiv(P(1,1),:); %indivíduos selecionados.
89         end
90         indiv=nindiv; %Substitui a população pela nova.
91
92         % Crossover
93         disp('== Crossover ==')
94         for i=1:(size(indiv,1)/2) %Seleção de cruzamento
95             x1=randi([1 size(indiv,1)],1,1); %escolha aleatória de pais
96             pai1=indiv(x1,:);
97             x2=randi([1 size(indiv,1)],1,1);
98             pai2=indiv(x2,:);
99             alpha=rand; %gerando filhos
100            filho1=alpha*pai1+(1-alpha)*pai2; %equação utilizada para ...
                ↪ cruzar os genes
101            filho2=(1-alpha)*pai1+alpha*pai2;
102            indivaux=nindiv;
103            nindiv=[indivaux; filho1; filho2]; %adicionando filhos ...
                ↪ criados a população
104        end
105        indiv=nindiv;
106        for i=1:size(indiv,1) %testando novamente os indivíduos
107            [E(i,:),G(i,:),Hmfg(i,:)] = fitness_n(indiv,H,Ts,ref,sigma,i);
108        end
109        %Deixando apenas os melhores indivíduos
110        for i = 1:popini %exclui os com entropia de maior valor
111            [m,n]=max(E);
112            G(n,:) = [];
113            nindiv(n,:) = [];

```

```

114         E(n,:) = [];
115         Hmfg(n,:)=[];
116     end
117     indiv=nindiv;
118     i=1; j=1;
119     while i ≠ (length(E(:,1))+1) %substituindo os NaN e inf
120         B=isnan(E(i,:)); C=isinf(E(i,:));
121         if B==1 || C==1
122             indiv(i,:)=A*rand(1,ordem*2+2); %substitui por outro
123             [E(i,:),G(i,:),Hmfg(i,:)] = fitness_n(indiv,H,Ts,ref,sigma,i);
124         else i=i+1;
125         end
126     end
127     Ex(:,:)=50;
128     for i=1:size(indiv,1)
129         if ( (E(i,:)≠0) && (E(i,:) < precmax) && (E(i,:) > precmin) )
130             Ex(j,:)= E(i,:); %joga em novas variaveis pra ver a minima ...
131             ↪ depois
132             Ex(j,2)= i;
133             j=j+1;
134         end
135     end
136     j=1; Exx(:,:)=50;
137     for i=1:size(Ex,1)
138         [n]=(Ex(i,2));
139         if (((max(real(roots(G(n,:).den{1})))) < -0.002) && ...
140             ↪ (norm(Hmfg(n,:)) < normmax) && (norm(Hmfg(n,:)) > ...
141             ↪ normmin)) %verifica polos parta real negativa
142             Exx(j,:)= Ex(i,:);
143             j=j+1;
144         end
145     end
146     if ( (min(Exx(:,1)) < precmax) && ((min(Exx(:,1))) > ...
147         ↪ precmin) )
148         [m,h] = min(Exx(:,1))
149         [k]=(Exx(h,2));
150         G(k,:);
151         epoca=epoca+1; disp(['Epoca= ' num2str(epoca)]); disp(' ');
152         Result(xx,:)=min(Exx(:,1));
153         Resultfuncoes(xx,:)=G(k,:);
154         salvar = ['Results sig ', num2str(sigma), '.mat'];
155         save (salvar, 'Result', 'Resultfuncoes', 'H', 'Ts', 'A', ...
156             ↪ 'popini', 'sigma', 'precmax', 'pm', 'maxepoca', 'cont', ...

```

```

        ↪ 'ref', 'epoca')
152     precmax=min(Exx(:,1));
153     xx=xx+1;
154 end
155
156 % Mutacao
157 disp('== Mutaçao ==')
158 i=1;
159 for i = 1:size(indiv,1) %percorre todos os elementos de todos os ...
    ↪ individuos
160     for j = 1:length(indiv(i,:))
161         if pm>rand %sofre alteraçao se o numero sorteado for ...
            ↪ menor que a porcentagem escolhida
162             indiv(i,j) = indiv(i,j)*(1 + A*randn(1,1));
163         end
164     end
165     [E(i,:),G(i,:),Hmfg(i,)] = fitness_n(indiv,H,Ts,ref,sigma,i);
166 end
167 i=1; j=1;
168 while i ≠ (length(E(:,1))+1) %substituindo os NaN e inf
169     B=isnan(E(i,:)); C=isinf(E(i,:));
170     if B==1 || C==1
171         indiv(i,:)=A*rand(1,ordem*2+2); %substitui por outro
172         [E(i,:),G(i,:),Hmfg(i,)] = ...
            ↪ fitness_n(indiv,H,Ts,ref,sigma,i);
173     else i=i+1;
174     end
175 end
176 Ex(:,:)=50;
177 for i=1:size(indiv,1)
178     if ( (E(i,:)≠0) && (E(i,:) < precmax) && (E(i,:) > precmin) )
179         Ex(j,:)= E(i,:); %joga em novas variaveis pra ver a min depois
180         Ex(j,2)= i;
181         j=j+1;
182     end
183 end
184 j=1; Exx(:,:)=50;
185 for i=1:size(Ex,1)
186     [n]=(Ex(i,2));
187     if ((max(real(roots(G(n,:)).den{1})))) < -0.002) && ...
        ↪ (norm(Hmfg(n,:)) < normmax) && (norm(Hmfg(n,:)) > ...
        ↪ normmin) %verifica polos parta real negativa
188         Exx(j,:)= Ex(i,:);

```

```

189         j=j+1;
190     end
191 end
192     if ( (min(Exx(:,1)) < precmax) && ((min(Exx(:,1))) > ...
193         ⇨ precmin) )
194         [m,h] = min(Exx(:,1))
195         [k]=(Exx(h,2));
196         G(k,:);
197         epoca=epoca+1; disp(['Epoca= ' num2str(epoca)]); disp(' ');
198         Result(xx,:)=min(Exx(:,1));
199         Resultfuncoes(xx,:)=G(k,:);
200         salvar = ['Results sig ', num2str(sigma), '.mat'];
201         save (salvar, 'Result', 'Resultfuncoes', 'H', 'Ts', 'A', ...
202             ⇨ 'popini', 'sigma', 'precmax', 'pm', 'maxepoca', 'cont', ...
203             ⇨ 'ref', 'epoca')
204         precmax=min(Exx(:,1));
205         xx=xx+1;
206     end
207     epoca=epoca+1; disp(['Epoca= ' num2str(epoca)]); disp(' ');
208     disp(['Minimo atual=' num2str(min(Exx))]);
209     Exx=zeros(1,2);
210     disp(' ');
211 end
212 toc
213 cont
214 end
215
216 %% ----- Função Fitness -----
217 function [E, G, Hmfg] = fitness_n(indiv,H,Ts,ref,sigma,i)
218
219     G=tf([indiv(i,1:length(indiv(i,:)))/2],[indiv(i,((length(indiv(i,:)))/2)+1) :
220
221     Hmfg = feedback(G*H,1);
222     [y]=lsim(Hmfg,ref,Ts);
223     e = ref - y;
224     E = renyi2(abs(e),sigma);
225
226 %% ----- Função MEE -----
227 % Renyi's quadratic entropy estimation using Parzen windowing
228 % and Gaussian kernels
229 %
230 % H2 = renyi2(x,s)
231 %

```

```

229 % H2 - Renyi's quadratic entropy (bits)
230 % x - signal
231 % s - Gaussian kernel size
232 %
233 % From
234 % Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe.
235 % Correntropy: Properties and Applications in Non-Gaussian Signal ...
    ↪ Processing.
236 % IEEE Transactions on Signal Processing, v. 55, n. 11, nov. 2007, ...
    ↪ 5286-5298
237 %
238 % PRS, 07/jun/2017
239 %
240 function H2 = renyi2(x,s)
241
242 if nargin ≠ 2
243     error('Wrong number of parameters')
244 end
245 x = x(:);
246 N = length(x);
247 X = repmat(x',N,1)-repmat(x,1,N);
248 IP = kernel_s2(X,sqrt(2)*s);
249 IP = sum(IP(:))/(N^2);
250 H2 = -log2(IP);
251 clear x N X IP
252
253 %% ----- Função Kernel -----
254 % Gaussian Kernel estimation
255 %
256 % PRS, fev/08/2016
257 % modified in 07/jun/2017 -> x can be a matrix
258 %
259 function k = kernel_s2(x,s)
260
261 k = (1/(sqrt(2*pi)*s)) * exp((-x.^2)/(2*s^2));

```