

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL  
FACULDADE DE ENGENHARIAS, ARQUITETURA E URBANISMO E GEOGRAFIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
MESTRADO EM COMPUTAÇÃO FLEXÍVEL

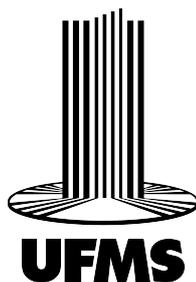


# **Desenvolvimento de um Controlador Preditivo Generalizado Otimizado com Algoritmo Genético para Rastreamento de Sinais Senoidais com Nível CC**

**Matheus Córdoba Caramalac**

Campo Grande - MS

15 de Julho de 2024



UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL  
FACULDADE DE ENGENHARIAS, ARQUITETURA E URBANISMO E GEOGRAFIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
MESTRADO EM COMPUTAÇÃO FLEXÍVEL



# **Desenvolvimento de um Controlador Preditivo Generalizado Otimizado com Algoritmo Genético para Rastreamento de Sinais Senoidais com Nível CC**

Matheus Córdoba Caramalac

Dissertação de mestrado apresentada como exigência para obtenção do título de Mestrado em Engenharia Elétrica na área de Computação Flexível da Universidade Federal de Mato Grosso do Sul – UFMS.

Orientador: Prof. Dr. Raymundo Cordero García

Campo Grande - MS

15 de Julho de 2024

Desenvolvimento de um Controlador Preditivo Generalizado  
Otimizado com Algoritmo Genético para Rastreamento de Sinais  
Senoidais com Nível CC

Dissertação de mestrado apresentada como exigência para obtenção do título de  
Mestrado em Engenharia Elétrica na área de Computação Flexível da Universidade  
Federal de Mato Grosso do Sul – UFMS.

**Banca Examinadora:**

---

Prof. Dr. Raymundo Cordero García

---

Prof. Dr. Luigi Galotto Junior

---

Prof. Walter Issamo Suemitsu

Campo Grande - MS

15 de Julho de 2024

# Agradecimentos

Em primeiro lugar, inicio esta série de agradecimentos a Deus, por todas as oportunidades que tem me dado desde tenra idade, ainda mais pelo fato de que, sem Ele, aqui não estaríamos todos nós, tampouco a oportunidade de cursar tal programa de mestrado seria possível, quanto mais realizável.

Em seguida, presto agradecimentos para minha família, que não tem deixado de me dar todo o suporte necessário para concluir mais esta fase da vida. Em especial devo agradecer a meu pai, Josiel Caramalac, e minha mãe, Cecília de Souza Córdoba Caramalac, que foram sempre meu porto seguro, minha base, meu apoio bem presente em todos os momentos, sempre me impulsionando a prosseguir mesmo em meio aos problemas. Estes são meus exemplos de vida e de perseverança e trabalho que desejo seguir sempre.

Sem ser remisso, também dispenso agradecimentos a todos meus amigos e colegas de trabalho e estudos, que me auxiliaram a crescer durante todo meu período de estudos. Com especial estima, cito aqui o nome do Thyago Vasconcelos Estrabis, pois nosso fatídico encontro após tantos anos distanciados foi o que permitiu o início de minha caminhada no Programa de Pós Graduação em Engenharia Elétrica (PPGEE) da UFMS

Dispenso especial agradecimento ao meu orientador, Dr. Raymundo Cordero García, por ser um incrível profissional assim como é uma incrível pessoa, devido sua paciência, dedicação e apoio em todos os momentos desta caminhada, sendo um grande exemplo do profissional da educação e do ensino que almejo me tornar se um dia esta possibilidade for concretizada.

Por fim, estendo meus agradecimentos a todas as pessoas que, de sua própria maneira, me auxiliaram e contribuíram para que este projeto pudesse ser concluído.

# Resumo

O Controle Preditivo baseado em Modelo (do inglês *Model Predictive Control* - abreviado para MPC) é uma das técnicas de controle mais avançadas que pode ser usada em diferentes aplicações industriais. Entre as técnicas que são englobadas por esta denominação de controle se encontra o Controle Preditivo Generalizado (do inglês, *Generalized Predictive Control* - GPC), cujo objetivo é obter um conjunto de leis de controle futuras que sejam capazes de reduzir o erro existente entre as saídas preditas e os valores futuros da referência da planta. Entretanto, a técnica de GPC tradicional não é capaz de ser utilizada para rastrear sinais senoidais, as quais são de grande importância em aplicações práticas. Ademais, algumas aplicações requerem que a referência senoidal possua um nível CC, ou que perturbações tipo degrau sejam rejeitadas. Assim sendo, este trabalho tem por objetivo o desenvolvimento e implementação experimental de um algoritmo GPC que seja capaz de rastrear referências que sejam uma combinação de sinais degrau e senoidal, além da utilização de algoritmos genéticos para a determinação dos parâmetros ótimos de implementação de uma planta em tempo discreto sujeita a ruídos brancos gaussianos. Resultados de simulação e experimentais comprovam que a técnica aqui proposta é capaz de rastrear adequadamente estes tipos de referências citadas, e também possui a capacidade de rejeitar perturbações senoidais e tipo degrau.

**Palavras-chaves:** controle por modelo preditivo, mpc, controle preditivo generalizado, gpc, rastreamento de referência senoidal, rejeição de perturbações.

# Abstract

Model Predictive Control (MPC) is one of the most advanced control techniques that can be used in different industrial applications. One of the techniques developed under this definition is called Generalized Predictive Control (GPC), whose objective is to obtain a set of future control laws that are capable of reducing the error between the predicted outputs and the future references of the plant. However, the traditional GPC technique is not capable of being used to trace and/or reject sinusoidal references, which have great importance in practical applications. Furthermore, some applications require that the sinusoidal reference also contain a DC level, or step disturbances to be rejected. Thus, this work is aimed at the development and experimental implementation of a GPC algorithm that is capable of tracking references composed from a combination of step and sinusoidal signals as well as the usage of genetic algorithms in order to determine the optimized implementation parameters of a GPC controller associated with a plant under the influence of white gaussian noise. Simulation and experimental results confirm that the technique proposed here is capable of adequately tracking these types of references, and also has the ability to reject sinusoidal and step disturbances.

**Keywords:** model predictive control, mpc, generalized predictive control, gpc, sinusoidal reference tracking, disturbance rejection.

# Lista de ilustrações

Figura 1	–	Estratégia de Funcionamento de um MPC . . . . .	16
Figura 2	–	Estrutura de Implementação de um MPC . . . . .	18
Figura 3	–	Representações do Plano- $\mathcal{Z}$ Complexo . . . . .	25
Figura 4	–	Exemplo de um problema de otimização unidimensional . . . . .	30
Figura 5	–	Representação de um conjunto viável de acordo com a Equação 2.23	30
Figura 6	–	Exemplo de pontos críticos de interesse em algoritmos de otimização em uma função com uma única variável . . . . .	31
Figura 7	–	Fluxo de Desenvolvimento de Otimização . . . . .	32
Figura 8	–	Identificação de um indivíduo, um cromossomo e um gene em base binária . . . . .	37
Figura 9	–	Exemplo de uma roleta de seleção de indivíduos em algoritmo genético	39
Figura 11	–	Fluxograma de operação de um algoritmo genético . . . . .	39
Figura 10	–	Exemplos de cruzamentos e mutações de algoritmos genéticos em números em base binária . . . . .	40
Figura 12	–	Diagrama de um bloco de um sistema GPC de malha fechada . . . . .	46
Figura 13	–	Diagrama de um circuito que atende o design de rastreamento e rejeição robustos . . . . .	47
Figura 14	–	Sinal de Entrada - Referência do tipo Senoidal . . . . .	55
Figura 15	–	Estrutura do Controlador GPC Proposto . . . . .	60
Figura 16	–	Sinal de Referência em tempo discreto utilizado nas simulações de controle em malha fechada no MATLAB . . . . .	64
Figura 17	–	Fluxo de aplicação do algoritmo genético no cálculo do indivíduo ótimo . . . . .	66
Figura 18	–	Configuração dos experimentos realizados . . . . .	68
Figura 19	–	Sinal de Referência a ser rastreado no Teste 1 . . . . .	68
Figura 20	–	Resultados experimentais Teste 1: Lei de Controle - Parâmetros $N_P = 20$ , $N_C = 1$ e $r_w = 4$ . . . . .	69
Figura 21	–	Resultados experimentais Teste 1: Sinal de Erro - Parâmetros $N_P = 20$ , $N_C = 1$ e $r_w = 4$ . . . . .	69
Figura 22	–	Resultados experimentais Teste 1: Lei de Controle - Parâmetros $N_P = 20$ , $N_C = 12$ e $r_w = 1$ . . . . .	70
Figura 23	–	Resultados experimentais Teste 1: Sinal de Erro - Parâmetros $N_P = 20$ , $N_C = 12$ e $r_w = 1$ . . . . .	70

Figura 24	–	Resultados experimentais Teste 1: Lei de Controle - Parâmetros $N_P = 25$ , $N_C = 4$ e $r_w = 32$ . . . . .	71
Figura 25	–	Resultados experimentais Teste 1: Sinal de Erro - Parâmetros $N_P = 25$ , $N_C = 4$ e $r_w = 32$ . . . . .	71
Figura 26	–	Resultados experimentais Teste 1: Efeito da alteração de $N_C$ na Lei de Controle . . . . .	72
Figura 27	–	Resultados experimentais Teste 1: Efeito da alteração de $r_w$ na Lei de controle . . . . .	72
Figura 28	–	Circuito em malha fechada para teste de robustez do controlador implementado na ferramenta <i>Simulink</i> . . . . .	74
Figura 29	–	Resultados simulados Teste 2: Comparação entre a saída da planta e do sinal de referência - Parâmetros $N_P = 16$ , $N_C = 2$ e $r_w = 6.76$ . . . . .	76
Figura 30	–	Resultados simulados Teste 2: Sinal de erro do sistema - Parâmetros $N_P = 16$ , $N_C = 2$ e $r_w = 6.76$ . . . . .	76
Figura 31	–	Resultados simulados Teste 2: Comparação entre a saída da planta sujeita a ruído branco gaussiano e do sinal de referência - Parâmetros $N_P = 16$ , $N_C = 2$ e $r_w = 6.76$ . . . . .	77
Figura 32	–	Resultados simulados Teste 2: Sinal de erro do sistema sujeito a ruído branco gaussiano - Parâmetros $N_P = 16$ , $N_C = 2$ e $r_w = 6.76$ . . . . .	78
Figura 33	–	Resultados experimentais do Teste 2: Lei de Controle - Parâmetros $N_P = 16$ , $N_C = 2$ e $r_w = 6.76$ . . . . .	78
Figura 34	–	Resultados experimentais do Teste 2: Sinal de Erro - Parâmetros $N_P = 16$ , $N_C = 2$ e $r_w = 6.76$ . . . . .	79

# Lista de tabelas

Tabela 1	–	Parâmetros dos sinais utilizados nos testes com o controlador proposto . . . . .	63
Tabela 2	–	Cálculo do EQM (em mV) com diferentes parâmetros - Teste 1 . . .	73
Tabela 3	–	Valor do Tempo de Acomodação do Sinal (em ms) com diferentes parâmetros - Teste 1 . . . . .	73
Tabela 4	–	Termos e pesos aplicados na função custo, o valor do custo obtido e os cromossomos do indivíduo ótimo em cada teste realizado . . .	75

# Lista de abreviaturas e siglas

AG	Algoritmo Genético
ARIMAX	Modelo Autorregressivo com Média Integrada Variável e Entrada Exógena
CARIMA	Modelo Controlado Autorregressivo e com Média Integrada Variável
CC	Corrente Contínua
HEV	Veículo Elétrico Híbrido
GPC	Controle Preditivo Generalizado
GVM	Variância Mínima Generalizada
LCL	Filtro com 2 indutores (L) e 1 Capacitor (C)
MAC	Controle Algorítmico baseado em Modelos
MHC	Controle por Horizonte Deslizante
MPC	Controle Preditivo baseado em Modelos
MPHC	Controle Preditivo Heurístico baseado em Modelos
PP	- <i>Pole Placement</i> - Alocação de Polos
PI	Programação de Inteiros
PL	Programação Linear
PNL	Programação Não-Linear
PMIL	Programação Mista de Inteiros Lineares (PMIL)
PMINL	Programação Mista de Inteiros Não-Lineares (PMINL)
RDC	Região de Convergência
RHC	Controle por Horizonte Retrocedente
VM	Variância Mínima

# Lista de símbolos

$\overleftrightarrow{z}$	Sinais equivalentes no domínio do tempo e no domínio $\mathcal{Z}$
$\oint$	Integral de linha de curva fechada
$\approx$	Aproximadamente
$A_m$	Matriz de estados da planta
$A$	Matriz de estados do modelo aumentado
$\hat{A}_{eq}$	Matriz dos coeficientes das restrições de igualdade da função custo
$\hat{A}_{des}$	Matriz dos coeficientes das restrições de desigualdade da função custo
$B_m$	Matriz de entradas da planta
$B$	Matriz de estados do modelo aumentado
$\hat{B}_{eq}$	Matriz dos resultados das restrições de igualdade da função custo
$\hat{B}_{des}$	Matriz dos resultados das restrições de desigualdade da função custo
$C_m$	Matriz de saídas da planta
$C$	Matriz de saídas do modelo aumentado
$D_m$	Matriz de transmissão direta da planta
$\det \cdot $	Operador de cálculo do determinante de uma matriz
$e(i)$	Erro de rastreamento em tempo discreto
$\epsilon(\cdot)$	Matriz coluna contendo valores dos erros
$E$	Matriz dos coeficientes da equação do erro futuro $e(i + 1)$
$f(t)$	Função no domínio do tempo
$F(s)$	Função no domínio da frequência de Laplace
$F(z)$	Função no domínio da frequência $\mathcal{Z}$
$J$	Função Custo do modelo de controle
$\hat{J}$	Função Custo do Algoritmo Genético
$\mathcal{F} \cdot $	Transformada direta de Fourier

$\mathcal{F}^{-1} \cdot $	Transformada inversa de Fourier
$F$	Matriz de índices das variáveis de estado do modelo aumentado
$\delta$	Sinal do tipo impulso, discreto no domínio do tempo
$\Delta$	Operador de diferença de primeira ordem
$\phi$	Matriz dos coeficientes do vetor de diferença das variáveis de controle
$\mathcal{L} \cdot $	Transformada direta de Laplace
$\mu$	Média
$N_C$	Horizonte de Controle
$N_P$	Horizonte de Predição
$r_w$	Parâmetro de ajuste da função custo em malha fechada
$\mathbb{R}$	Conjunto dos números Reais
$s$	Variável laplaciana
$\sigma$	Variância
$ \cdot ^T$	Indicador da operação de transposição de um vetor/matriz
$t_s$	Tempo de Amostragem
$u(i)$	Variável manipulada ou manipulável discreta no domínio do tempo
$u(t+k t)$	Valor da lei de controle $u$ predita no tempo $t+k$ com informações disponíveis em $t$
$y(i)$	Saída do sistema
$y(t+k t)$	Valor da saída $y$ predita no tempo $t+k$ com informações disponíveis em $t$
$w(t+k t)$	Valor da referência $w$ preditas no tempo $t+k$ com informações disponíveis em $t$
$\hat{x}_i$	$i$ -ésimo indivíduo de uma população do algoritmo genético
$x i$	Sinal arbitrário discreto no domínio do tempo
$x(i)$	Vetor das variáveis de estado do modelo aumentado, discretas no domínio do tempo

$x_m(i)$	Vetor das variáveis de estado da planta, discretas no domínio do tempo
$z$	Variável complexa do domínio $\mathcal{Z}$
$\mathcal{Z} \cdot $	Transformada $\mathcal{Z}$ direta
$\mathcal{Z}^{-1} \cdot $	Transformada $\mathcal{Z}$ inversa

# Sumário

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Motivação	20
1.2	Objetivos	21
1.2.1	Objetivo Geral	21
1.2.2	Objetivos Específicos	21
1.3	Organização do Trabalho (Resumo dos capítulos)	22
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>23</b>
2.1	Fundamentação Teórica	23
2.2	Transformada $\mathcal{Z}$	23
2.2.1	Propriedades da Transformada $\mathcal{Z}$	25
2.3	Discretização de Sistemas Contínuos no Tempo	27
2.4	Otimização	29
2.4.1	Restrições	29
2.4.2	Pontos Críticos	30
2.4.3	Fluxo do Processo de Otimização	31
2.4.4	Categorias de Otimização	33
2.5	Métodos Heurísticos de Otimização	34
2.5.1	<i>Simulated Annealing</i> (SA)	34
2.5.2	Busca Tabu ( <i>Taboo Search</i> - TS)	34
2.6	Algoritmo Genético	35
2.6.1	População	35
2.6.2	Indivíduos	36
2.6.3	Cromossomo	36
2.6.4	Gene	36
2.6.5	Aptidão	37
2.6.6	Seleção	37
2.6.7	Cruzamento	38
2.6.8	Mutação	38
2.6.9	Gerações	38
2.7	GPC Convencional	40
2.7.1	Modelo Aumentado	40
2.7.2	Predição	42
2.7.3	Otimização do Modelo Aumentado	44
2.7.4	Sistema em Malha Fechada	45

---

2.8	Princípio do Modelo Interno para Rastreamento e Rejeição Robustos de Sinais . . . . .	46
2.9	Estado da Arte . . . . .	49
<b>3</b>	<b>Controlador Proposto . . . . .</b>	<b>54</b>
3.1	Desenvolvimento de GPC para Rastreamento de Sinal Senoidal com Nível CC . . . . .	54
3.1.1	Preliminares . . . . .	54
3.1.2	Modelo Aumentado . . . . .	56
3.1.3	Lei de Controle Preditivo através de GPC . . . . .	58
3.1.4	Capacidade de Rastreamento do Controlador . . . . .	60
3.2	Determinação dos Parâmetros Ótimos do GPC através de Algoritmo Genético . . . . .	62
3.2.1	Escolha do Algoritmo de Otimização . . . . .	62
<b>4</b>	<b>Resultados . . . . .</b>	<b>67</b>
4.1	Definição da Planta de Testes . . . . .	67
4.2	Teste 1: Rastreamento de Referências Senoidais com <i>bias</i> . . . . .	67
4.3	Teste 2: Escolha dos Melhores Parâmetros via Algoritmo Genético . . . . .	74
	<b>Conclusão . . . . .</b>	<b>80</b>
	Sugestões para Trabalhos Futuros . . . . .	80
	<b>Referências . . . . .</b>	<b>81</b>
	<b>Apêndices . . . . .</b>	<b>85</b>
	<b>APÊNDICE A Códigos . . . . .</b>	<b>86</b>
A.1	Determinação da Função Custo no MATLAB . . . . .	86
A.2	Aplicação de Algoritmo Genético no MATLAB . . . . .	92

# 1 Introdução

O incremento da complexidade e dos níveis de desempenho de processos industriais motiva o desenvolvimento de controladores eletrônicos com melhor velocidade de resposta, maior precisão e robustez. Estes controladores requerem de técnicas de controle capazes de ter respostas mais robustas aos imprevistos e perturbações, assim como maior rapidez nestas respostas. Buscando satisfazer estas necessidades, várias técnicas de controle surgiram com o passar do tempo, como por exemplo as técnicas de controle adaptativo, controle robusto, integração com inteligência artificial, entre outras. Uma destas técnicas é a chamada de Controle Preditivo baseado em Modelos (do inglês *Model Predictive Control* - abreviado para MPC). De maneira mais acertada, o MPC seria menos uma técnica e mais um algoritmo de implementação de controles preditivos.

Segundo [García, Prett e Morari \(1989\)](#), o interesse neste algoritmo surgiu ao final da década de 1970, com destaque para as produções de [Richalet et al. \(1977\)](#) e [Cutler e Ramaker \(1979\)](#), que apresentaram os conceitos de Controle Preditivo Heurístico baseado em Modelo (do inglês *Model Predictive Heuristic Control* - MPHC, posteriormente chamado de *Model Algorithmic Control* - MAC, ou Controle Algorítmico baseado em Modelo), e a de Controle por Matriz Dinâmica (do inglês *Dynamic Matrix Control*), respectivamente. E, ainda que tais modelos tenham atuações distintas entre si, como bem pontuado por [Camacho e Bordons \(2007\)](#), estes métodos, assim como os outros citados anteriormente, possuem como estruturas básicas três itens:

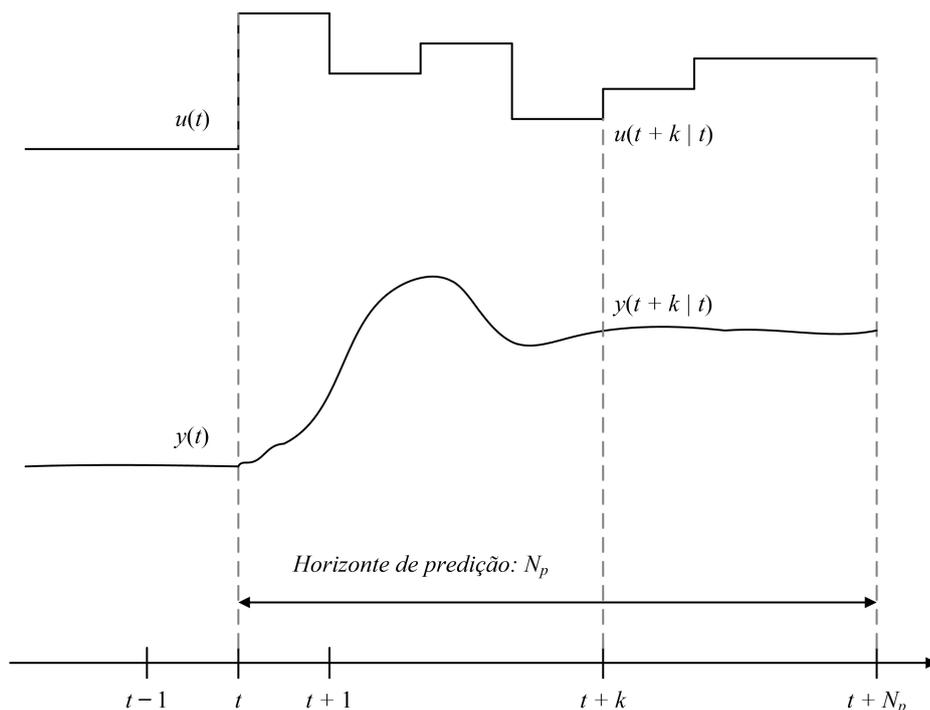
- Uso explícito de um modelo da planta utilizado para prever as saídas do processo em tempos futuros (o chamado “horizonte”);
- Cálculo de leis de controle com o objetivo de minimizar os erros de controle;
- Estratégia regressiva de forma que, a todo instante, o horizonte é demonstrado em direção ao futuro, graças a aplicação do primeiro sinal de controle que é calculada a cada passo da análise (o chamado “horizonte deslizante” ou ainda “horizonte retrocedente”).

Devido sua forma de aplicação e cálculo, os métodos de controle MPC possuem vantagens quando comparados a outros métodos, como apresentados por ([CAMACHO; BORDONS, 2007](#); [ORUKPE, 2012](#); [WANG, 2009](#); [QIN; BADGWELL, 2003](#); [RAKOVIC; LEVINE, 2019](#); [RAWLINGS; MAYNE; DIEHL, 2019](#); [KWON; HAN, 2005](#); [MAEDER; MORARI, 2010](#)):

- Capacidade de integração com outras técnicas de controle, como controle ótimo, estocástico, multivariável, etc.;
- Capacidade de implementação de restrições já na formulação do MPC;
- Aplicação para controle tanto de sistemas lineares quanto não lineares;
- Os modelos utilizados são compostos de conceitos bastante intuitivos mesmo para pessoas com pouco conhecimento da teoria de controle, o que permite ajustes relativamente fáceis de serem realizados;
- Fácil implementação em processos complexos, incluindo sistemas com grandes tempos de inatividade ou de atraso de transporte;
- Fácil aplicação em sistemas multivariáveis;
- Capacidade de implementação do controle com realimentação futura (*feed-forward*) de forma simples e capaz de compensar as perturbações capazes de serem medidas;
- A metodologia aplicada para sua execução é baseada em princípios básicos que permitem a expansão dos controladores e suas leis em casos necessários;

Todos os controladores que aplicam a metodologia MPC possuem como característica principal sua estratégia de funcionamento semelhante, que é representada através da [Figura 1](#).

Figura 1 – Estratégia de Funcionamento de um MPC



Fonte: ([CAMACHO; BORDONS, 2007](#))

A estratégia apresentada pode ser descrita em três passos:

1. As saídas futuras são previstas para os próximos  $N_p$  instantes de tempo  $t$ , denominado “horizonte de previsão”. Estas saídas são previstas utilizando o modelo do processo. As saídas previstas, apresentadas como  $y(t+k|t)$ <sup>1</sup> para  $k = 1, \dots, N$ , dependem de todos os valores conhecidos até o instante  $t$  (incluindo entradas e saídas passadas) assim como os sinais de controle futuros  $u(t+k|t), k = 0, \dots, N-1$ ;
2. O conjunto dos sinais de controle futuros é calculado através da otimização de um determinado critério, buscando manter o processo o mais próximo possível da trajetória de referência  $w(t+k|t), k = 0, \dots, N-1$  (que pode ser o próprio *set-point* ou uma aproximação dele). Este critério normalmente assume a forma de uma função quadrática dos erros entre o sinal de saída predito e a trajetória de referência predita. O esforço de controle é incluído na função objetivo na maior parte dos casos. Se o critério aplicado for uma função quadrática, o modelo for linear e ele não possuir restrições, é possível se obter uma solução explícita para estas leis de controle. Caso contrário, se faz necessária a utilização de um método iterativo de otimização para obtê-las.
3. O sinal de controle  $u(t|t)$ , ou seja, o sinal de controle no instante  $t$  calculado com as informações obtidas até o instante  $t$ , é enviado para o processo de controle. Entretanto, todos os sinais de controle  $u(t+k|t), k > 0$  são descartados, já que no próximo instante de amostragem o valor de  $y(t+1)$  será conhecido e medido, o passo 1 será realizado novamente e novos valores serão calculados. Logo, em teoria, o valor de  $u(t+1|t+1)$  obtido será, em princípio, distinto do valor calculado para  $u(t+1|t)$  devido as novas informações medidas, já que estas foram obtidas através da utilização do conceito de horizonte retrocedente (o que não significa, necessariamente, que os valores de  $u(t+1|t+1)$  e  $u(t+1|t)$  sejam distintos).

Seguindo todas as informações descritas, é possível apresentar uma estrutura de funcionamento básica para os controladores do tipo MPC, facilitando a compreensão de seu funcionamento. Tal estrutura é apresentada na [Figura 2](#) e esta pode ser aplicada tanto para modelos de processos quanto para modelos de perturbação. Segundo [Orukpe \(2012\)](#), os modelos de processo podem assumir os seguintes modelos:

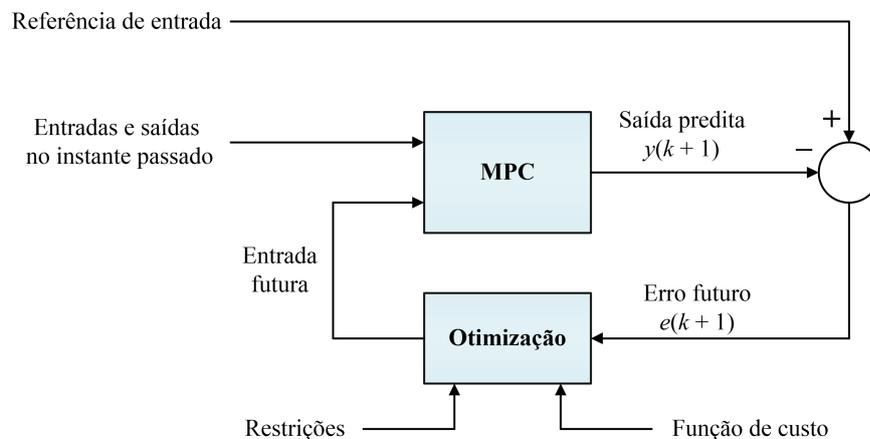
- Resposta ao Impulso - O mais usado na indústria de maneira geral;
- Resposta ao Degrau;

<sup>1</sup> esta representação da saída predita é denominada de “valor das saídas  $y$  previstas no tempo  $t+k$  baseada nas informações disponíveis no tempo  $t$ ”. Este tipo de representação de dados será aplicado durante toda extensão deste trabalho.

- Função de Transferência - Usado tanto pelas indústrias quanto pela comunidade acadêmica;
- Espaço de Estados - Principalmente utilizado pela comunidade acadêmica.

Já os modelos de perturbação normalmente utilizam modelos do tipo “CARIMA” (do inglês *Controlled AutoRegressive and Integrated Moving Average* - Modelo Controlado Autorregressivo e com Média Integrada Variável).

Figura 2 – Estrutura de Implementação de um MPC.



Fonte: (ESTRABIS, 2020)

Conforme demonstrado na Figura 2, nota-se que parte integrante da implementação de algoritmos de controle MPC é a otimização do sistema, onde as restrições à resposta da planta e do controlador são aplicadas, assim como a função de custo  $J$ .

Um dos pontos de maior destaque para a aplicação de algoritmos de controle MPC é sua capacidade de lidar com restrições de projeto de maneira sistemática, conseguindo manter, até certo nível, as margens de estabilidade e performance da lei de controle de um sistema não restrito. Segundo Orukpe (2012), vários são os tipos de restrições que podem ser aplicados nas entradas, saídas ou estados do sistema, a saber:

- Restrições de Banda;
- Restrições de *overshoot* (ultrapassagem);
- Comportamentos monotônicos;
- Comportamento não-mínimo de fases;
- Não linearidades do atuador;
- Restrições de igualdade do estado terminal;

- Restrições do conjunto de terminais.

Estas restrições, quando aplicadas nas entradas do sistema, tendem a ser *físicas*, no sentido que normalmente representam limitações nos equipamentos associados ao processo (como, por exemplo, limite máximo de vazão em uma válvula ou de leitura em um sensor) e, por este motivo, normalmente não podem ser flexibilizadas. Já as restrições nas saídas possuem a tendência de serem determinadas por motivos de segurança e devem ser controladas *a priori*, pois as variáveis de saída são afetadas por processos dinâmicos.

Já a função custo  $J$  (também denominada por “função de aptidão”) tem como objetivo a verificação das divergências entre as saídas previstas e o sinal de referência, de maneira a penalizar resultados que sejam extremamente divergentes desta referência. Isto permite que o sistema busque cada vez mais se aproximar da saída desejada.

Além dos fatos já apresentados, deve-se também destacar que o amplo uso de algoritmos de controle da família MPC nas indústrias é baseado em mais fatores:

- Capacidade de manipular controle de multivariáveis com maior facilidade;
- Capacidade de considerar as restrições/limitações dos atuadores;
- Permite que as operações de controle sejam realizadas com valores mais próximos das restrições do sistema, diminuindo custos desnecessários para funcionamento e aumentando o lucro;
- Flexibilidade de implementação por ser capaz de lidar com processos que não se encontram em fase mínima ou não são estáveis;
- Facilidade em implementar ajustes necessários para o funcionamento do sistema;
- Robustez com mudanças estruturais do sistema.

Os itens listados previamente atestam a capacidade de aplicação de algoritmos de controle do tipo MPC em várias situações. Desta forma, devido a sua versatilidade, tais algoritmos podem ser divididos em duas classes principais:

a) MPC Linear

- Utiliza modelo linear do tipo  $\dot{x} = Ax + Bu$
- Possui função custo quadrática na forma  $J = x^T Qx + u^T Ru$
- Podem ser aplicadas restrições lineares do tipo  $Hx + Gu < 0$
- Sua programação é Quadrática

b) MPC Não Linear

- Utiliza modelo não linear do tipo  $\dot{x} = f(x, u)$ .
- A função custo pode não ser quadrática  $J = f(x, u)$ .
- Suas restrições não são lineares, ou seja,  $h(x, u) < 0$ .
- Sua programação não é linear.

## 1.1 Motivação

Uma das técnicas de controle de modelo preditivo bastante utilizada, devido ao fato de ser capaz de lidar tanto com plantas lineares quanto não lineares, é o Controle Preditivo Generalizado (do inglês *Generalized Predictive Control* - GPC). Em GPC, as respostas futuras da planta são previstas através de um modelo aumentado de predição. Tais saídas previstas, que compõem o chamado “horizonte de predição”, são usadas para definir uma função custo capaz de medir o desempenho do sistema frente ao rastreamento dos sinais desejados. Através do uso de técnicas de otimização, é possível calcular o conjunto de leis de controle futuras que devem ser utilizadas para a diminuição da função custo.

De maneira geral, utiliza-se como referência em sistemas GPC um valor constante no horizonte de predição nos casos onde a referência é do tipo degrau. Nos modelos aumentados para predição de resposta da planta, se encontra a presença de um integrador (o que diminui o erro em regime permanente para referências do tipo degrau) (CLARKE; MOHTADI; TUFFS, 1987). Outros sinais de referência, como por exemplo os do tipo polinomial (rampas, parábolas) também podem ser utilizados em diversas aplicações (CORDERO et al., 2022a; CORDERO et al., 2022b).

As características de funcionamento deste algoritmo de controle podem ser de grande valia para utilização conjunta em aplicações como controle ativo de vibrações, balanceamento ativo de rotores, mecânica, rolamentos magnéticos e outras funcionalidades que requerem o rastreamento e/ou rejeição de sinais periódicos (KAMALDAR; HOAGG, 2020; ESBROOK; TAN; KHALIL, 2013; HONG et al., 2010). Neste contexto, estas ações de rastreamento e rejeição de sinais senoidais com *bias* (ou seja, um sinal senoidal com um componente de CC) possuem várias aplicações importantes (KIM; MOON, 2020), como por exemplo a utilização em alguns atuadores eletro-hidráulicos, onde as perturbações do sistema podem ser modeladas como sinais senoidais com *bias* e propriamente rejeitadas (KIM et al., 2013). Outro exemplo é o sistema de controle implementado por Ji et al. (2021), que define o impulso vertical de uma aeronave como *bias* enquanto rejeita as perturbações, descritas como sinais senoidais. Temos também algumas máquinas com ímãs permanentes (MENG et al., 2021; DU et al., 2022) e máquinas de Vernier (LI et al., 2019; JIA et al., 2018; KONG et al., 2017) que necessitam de correntes senoidais com *bias* para que possam funcionar. A tensão de desacoplamento do capacitor no inversor apresetenquepor Xu, Chang e Shao (2020) é descrito por uma onda senoidal com

*bias* para ser capaz de compensar a potência oscilatória produzida pela rede onde se encontra. Além disso, o controle de conversores Boost (RIVERA; ORTEGA-CISNEROS; CHAVIRA, 2019) é realizado de forma a produzir uma tensão unipolar na saída do conversor. Todas estas aplicações, atualmente, se valem da utilização de controladores PI, proporcionais-ressonantes, funções de Lyapunov, entre outros, mas que são exemplos de aplicações que podem receber um controlador do tipo GPC de rastreamento e rejeição de sinais senoidais com *bias*.

Apesar de ter sido criado na década de 1980, grande parte do acervo literário produzido para este algoritmo está focado na demonstração do rastreamento de entradas do tipo degrau. Poucas publicações trazem formas de realizar o rastreamento em sinais do tipo senoidal, sendo as publicações de Wang e Rossiter (2008) e Cordero et al. (2022b) algumas das poucas demonstrações da possibilidade da aplicação deste algoritmo de maneira efetiva para este tipo de sinal. Logo, a motivação deste estudo é realizar modificações para que seja possível o rastreamento de sinais senoidais em sistemas SISO de maneira que sua formulação seja de fácil implementação em sistemas onde tal rastreamento seja necessário.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento e teste experimental de um algoritmo GPC para o rastreamento de referências que são uma combinação de sinais tipo degrau e senoidal. O sistema GPC proposto utiliza o operador de diferença de segunda ordem para obter um modelo aumentado com dois integradores embutidos e cuja saída é o erro de rastreamento. Desta maneira, as técnicas de otimização utilizadas no GPC convencional podem ser utilizadas para obter a lei de controle. Será aplicado um algoritmo genético para a sintonização do controlador preditivo.

### 1.2.2 Objetivos Específicos

1. Compreensão matemática do algoritmo GPC convencional.
2. Desenvolvimento do modelo aumentado de predição baseado no operador de diferença de segunda ordem.
3. Aplicação das técnicas de otimização para o cálculo da lei de controle.
4. Uso do algoritmo genético para sintonizar os parâmetros.
5. Implementação experimental e testes do algoritmo proposto.

### 1.3 Organização do Trabalho (Resumo dos capítulos)

- **Capítulo 2:** apresenta a fundamentação teórica para o desenvolvimento do controlador preditivo e os conceitos básicos de um algoritmo genético.
- **Capítulo 3:** apresenta-se uma nova estrutura de GPC que possibilita realizar o rastreamento de sinais senoidais e degrau.
- **Capítulo 4:** são apresentados os resultados dos ensaios realizados.
- **Capítulo 5:** encontram-se as conclusões sobre os resultados obtidos.

## 2 Revisão Bibliográfica

### 2.1 Fundamentação Teórica

Neste capítulo, serão apresentados conceitos teóricos importantes para o desenvolvimento do algoritmo proposto, incluindo a Transformada  $Z$ , a técnica de aplicação do GPC convencional, conceitos sobre otimização de problemas e do controlador de modelagem interna.

### 2.2 Transformada $\mathcal{Z}$

Sabe-se que a Transformada de Fourier é uma ferramenta de grande importância nas análises de sistemas lineares, permitindo a modificação do domínio de uma função do domínio do tempo para o domínio da frequência, de maneira a facilitar a obtenção de soluções simplificadas para inúmeras equações diferenciais e outros problemas a elas associadas. Generalizações da Transformada de Fourier geram as conhecidas Transformada de Laplace para sistemas no tempo contínuo e a Transformada  $\mathcal{Z}$  para sistemas discretos no tempo.

Deve-se destacar a importância da utilização da Transformada  $\mathcal{Z}$  baseada no fato que todas as medições e amostragens realizadas em sistemas digitais são representadas no tempo discreto, devido ao processo de amostragem utilizado por estes sistemas durante a aquisição de sinais. Sendo assim, a utilização de uma operação de transformação de domínio, permitindo que equações diferenciais de ordem elevadas possam ser até mesmo linearizadas - processo esse que facilita sua resolução - torna-se uma habilidade extremamente interessante e necessária para a análise e controle de sistemas digitais. Sua importância também reside no fato que a Transformada  $\mathcal{Z}$  pode ser aplicada na análise de sinais e sistemas em casos onde a Transformada de Fourier não existe (HAYKIN; VEEN, 2001).

Vários autores apresentam, até como forma de diferenciação, que sinais em tempos discretos são descritos com o uso de colchetes enquanto sinais em tempo contínuo são descritos com parênteses. Entretanto, considerando que este trabalho utilizará somente sinais em tempo contínuo, será convencionalizado para o restante deste trabalho que todos estes sinais serão apresentados através de parênteses, no formato  $x(i)$ .

Pelo fato de ambas as Transformadas de Laplace e  $\mathcal{Z}$  serem generalizações da Transformada de Fourier, muitas propriedades presentes em ambas são análogas entre si, fato que não impede o surgimento de diferenças entre tais propriedades, principalmente baseadas no fato de que sinais e sistemas contínuos e discretos possuem particularidades

exclusivas de cada um. Posta tais informações, temos que a Transformada  $\mathcal{Z}$  bilateral, segundo [Oppenheim, Willsky e Nawab \(1996\)](#), pode ser definida conforme (2.1):

$$X(z) \triangleq \sum_{n=-\infty}^{\infty} x(i)z^{-n} \quad (2.1)$$

onde temos que  $x(i)$  é um sinal arbitrário discreto no domínio do tempo e  $z$  é um número complexo exponencial, definido como  $z = re^{j\omega}$ .

Ainda que a Transformada  $\mathcal{Z}$  possa ser aplicada em casos onde a aplicação da Transformada de Fourier não é possível ou é de difícil aplicação, isto não significa que essa não possua condições de existência. Uma destas condições é a absoluta somabilidade de  $x(i)z^{-n}$ . Sabendo que  $|-x(i)z^{-n}| = |x(i)r^{-n}|$ , temos então que o somatório da equação (2.2) deve possuir um valor finito. A faixa de valores que a variável  $r$  pode assumir para satisfazer esta inequação é, então, denominada de “Região de Convergência” (RDC) ([HAYKIN; VEEN, 2001](#)).

$$\sum_{n=-\infty}^{\infty} |x(i)r^{-n}| < \infty \quad (2.2)$$

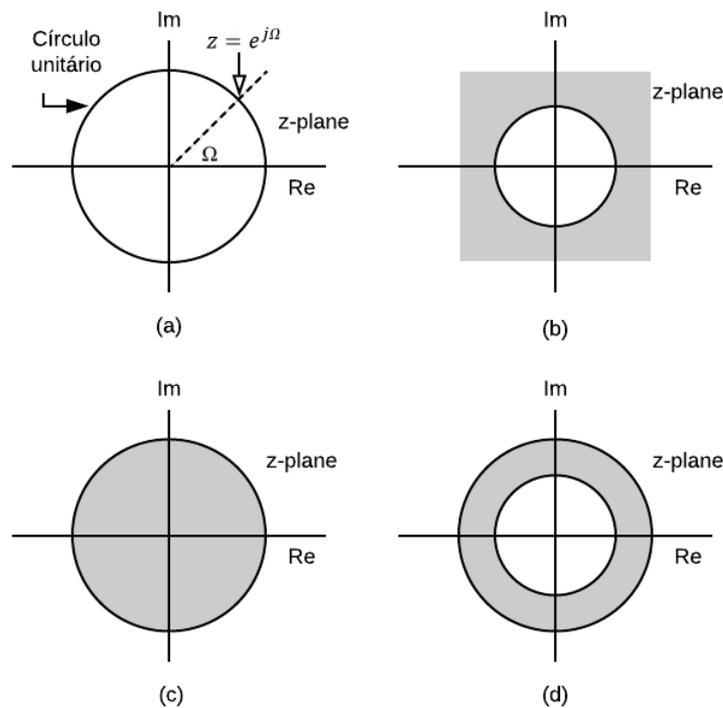
É importante destacar que existe a possibilidade de que uma Transformada  $\mathcal{Z}$  seja convertida para em uma Transformada de Fourier. Isso ocorre quando a magnitude da variável transformada  $z$  assume um valor unitário (por exemplo, quando  $z = e^{j\omega}$ ). Se apresentarmos esta situação em um Plano- $\mathcal{Z}$  complexo, encontra-se então um círculo de raio unitário, conforme apresentado na [Figura 3](#). É importante notar que, se a RDC da Transformada  $\mathcal{Z}$  incluir o círculo unitário, isto significa que a Transformada de Fourier também é convergente.

Segundo [Haykin e Veen \(2001\)](#), as RDCs possuem características próprias muito importantes que são ligadas as próprias características do sinal  $x(i)$  que originou tal transformada. Tais características são:

1. A RDC não contém nenhum polo da Transformada, já que a RDC é a região de valores de  $z$  para o qual a transformada converge - ou seja,  $X(z)$  deve possuir um valor finito para qualquer valor de  $z$  encontrado na RDC. Entretanto, se um valor  $p$  for um polo,  $|X(p)| = \infty$  e a transformada não converge. Logo, uma RDC não pode apresentar pólos.
2. Se um sinal possui uma duração finita, sua RDC é o Plano- $\mathcal{Z}$  inteiro, com a possibilidade de serem excluídos apenas os pontos  $z = 0$  e/ou  $z = \infty$ . Uma consequência desta característica é o fato que um sinal  $x(i) = c\delta(i)$  é o único tipo de sinal cuja RDC engloba o Plano- $\mathcal{Z}$  por completo.
3. Se um sinal possui uma duração infinita e for do tipo unilateral direito (ou seja, o sinal  $x[i]$  apenas possui valores quando  $i \geq 0$ ), tem-se que a RDC assume todos os valores finitos para o qual  $|z| > r_0$  (apresentado na [Figura 3 \(b\)](#))

4. Se um sinal possui uma duração infinita e for do tipo unilateral esquerdo (ou seja, o sinal  $x[i]$  apenas possui valores quando  $i < 0$ ), tem-se que a RDC assume todos os valores finitos para o qual  $|z| > r_0$  (apresentado na Figura 3 (c))
5. Se um sinal possui uma duração infinita e for do tipo bilateral (ou seja, o sinal  $x(i)$  possui valores para todos os valores de  $i$ ), tem-se que a RDC assume todos os valores finitos para o qual  $r_0 < |z| < r_1$  (apresentado na Figura 3 (d))

Figura 3 – Representações do Plano- $\mathcal{Z}$  Complexo. (a) Plano- $\mathcal{Z}$  Complexo, com a demonstração do círculo unitário onde a Transformada  $\mathcal{Z}$  é simplificada para uma Transformada de Fourier. (b) A área em cinza representa a RDC de um sinal infinito unilateral direito (c) A área em cinza representa a RDC de um sinal infinito unilateral esquerdo (d) A área em cinza representa a RDC de um sinal infinito bilateral.



Fonte: Adaptado de (OPPENHEIM; WILLSKY; NAWAB, 1996)

### 2.2.1 Propriedades da Transformada $\mathcal{Z}$

Também se faz importante demonstrar aqui algumas propriedades que a Transformada  $\mathcal{Z}$  possui. Assumindo que temos 2 Transformadas  $\mathcal{Z}$  conforme indicado:

$$x[i] \iff X(z), \quad RDC = R_x \quad (2.3)$$

$$y[i] \iff Y(z), \quad RDC = R_y \quad (2.4)$$

Tem-se que a primeira propriedade apresentada é a Linearidade. A Linearidade é uma propriedade que versa que a Transformada  $\mathcal{Z}$  da soma de sinais é igual a soma das Transformadas  $\mathcal{Z}$  dos sinais, conforme apresentado por Haykin e Veen (2001) em (2.5).

Tem-se que o RDC desta soma é dado por  $R_x \cap R_y$  já que a transformada da soma é válida somente quando ambas as transformadas convergem.

$$ax[i] + by[i] \xleftrightarrow{z} aX(z) + bY(z), \quad RDC_{min} = R_x \cap R_y \quad (2.5)$$

Outra propriedade importante é a da Inversão do Tempo. Isto significa em substituir  $z$  por  $z^{-1}$ . Logo, tem-se que

$$x[-i] \xleftrightarrow{z} X\left(\frac{1}{z}\right), \quad RDC = \frac{1}{R_x} \quad (2.6)$$

As Transformadas  $Z$  também possuem a propriedade do deslocamento no tempo, que versam que

$$x[i - i_0] \xleftrightarrow{z} z^{-i_0} X(z), \quad RDC = R_x \quad (2.7)$$

Neste caso, pode ser que o RDC deste deslocamento não inclua os valores de  $z = 0$  ou de  $|z| = \infty$ . Isso porque, ao multiplicar a transformada por  $z^{-i_0}$ , é introduzido um polo de ordem  $i_0$  em  $z = 0$  se  $i_0 > 0$ , o que impede a inclusão de 0 como parte do RDC, exceto se  $R_x$  possua zeros de ordem igual ou superior a  $i_0$  que possam cancelar os novos polos. Agora, se  $i_0 < 0$ , a multiplicação irá inserir  $i_0$  polos no infinito que, se não forem cancelados pelo RDC original, excluirão  $|z| = \infty$  do novo RDC deslocado.

Há também a propriedade de Multiplicação por Sequência Exponencial. Se admitirmos um número complexo  $\alpha$  seja multiplicado pelo sinal original, tem-se que

$$\alpha^n x[i] \xleftrightarrow{z} X\left(\frac{z}{\alpha}\right), \quad RDC = |\alpha| R_x \quad (2.8)$$

Existe também a propriedade de Convolução aplicada as Transformadas  $Z$ , onde tem-se que a convolução de sinais no domínio do tempo equivale à multiplicação de suas Transformadas  $Z$ .

$$x[i] * y[i] \xleftrightarrow{z} X(z)Y(z), \quad RDC_{min} = R_x \cap R_y \quad (2.9)$$

E por fim, tem-se a propriedade de Diferenciação no Domínio  $Z$ , onde a multiplicação de um sinal  $x[i]$  por  $n$  no domínio do tempo equivale a diferenciação em função de  $z$  e multiplicação por  $-z$  no domínio da frequência, sem alteração de RDC.

$$nx[i] \xleftrightarrow{z} -z \frac{d}{dz} X(z), \quad RDC = R_x \quad (2.10)$$

Por fim, destaca-se aqui o processo da Transformada  $Z$  inversa, já que muitas vezes se faz necessário que a solução encontrada seja retornada para o domínio do tempo para que decisões sejam tomadas ou atuações corretas possam ser escolhidas. Assim sendo, tem-se que

$$x[i] = \mathcal{Z}^{-1}\{X(z)\} = \frac{1}{2\pi} \oint X(z)z^{n-1}dz \quad (2.11)$$

## 2.3 Discretização de Sistemas Contínuos no Tempo

Muitas das plantas que necessitam ser controladas são definidas por modelos em tempo contínuo, seja por suas funções de transferência ou seus modelos em espaço de estado. Entretanto, como citado anteriormente, muitos algoritmos de controle trabalham com o tempo discreto, até mesmo pelos tipos de variáveis de controle e de observação que o sistema pode dispor para uso. Portanto, o processo de discretização dos sistemas se torna um passo necessário.

Existem várias técnicas para se obter uma função de transferência discreta  $G(z)$  a partir de uma planta  $G(s)$  definida no tempo contínuo. Uma destas técnicas é a aproximação da variável laplaciana  $s$  a uma função definida em  $z$ . Isto só é possível pois, como demonstrado anteriormente, ambas as Transformadas de Laplace e Transformada  $\mathcal{Z}$  são generalizações de uma mesma transformada, a anteriormente citada Transformada de Fourier. Se considerarmos um tempo de amostragem  $t_s$  para os sinais do sistema, [Haykin e Veen \(2001\)](#) demonstra que algumas das técnicas de discretização mais populares que utilizam este parâmetro são:

$$\text{Euler em atraso: } s \approx \frac{1 - z^{-1}}{t_s} \quad (2.12)$$

$$\text{Euler em avanço: } s \approx \frac{1 - z^{-1}}{t_s z^{-1}} \quad (2.13)$$

$$\text{Tustin: } s \approx \frac{2}{t_s} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (2.14)$$

Se considerarmos, então, um sinal de entrada  $X(z)$  de tempo discreto como a entrada desta planta discretizada, e um sinal de saída  $Y(z)$  como a saída desta planta dada a entrada  $X(z)$ , obtemos então a seguinte função de transferência:

$$G(z) = \frac{Y(z)}{X(z)} \quad (2.15)$$

que pode ser reescrita, com o auxílio da definição apresentada em (2.1), como uma razão de polinômios

$$G(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} \quad (2.16)$$

onde, se substituirmos (2.15) em (2.16), teremos

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} \\ Y(z)[1 + a_1 z^{-1} + \dots + a_N z^{-N}] = X(z)[b_0 + b_1 z^{-1} + \dots + b_M z^{-M}] \quad (2.17)$$

Sabendo que o resultado de (2.11) para um sinal do tipo  $z^{-m}X(z) = x(k-m)$ , podemos então obter como resultado da Transformada  $\mathcal{Z}$  inversa de (2.17) a seguinte equação de diferenças:

$$\begin{aligned} y(i) + a_1y(i-1) + \dots + a_Ny(k-N) &= b_0x(i) + b_1x(i-1) + \dots + b_Mx(k-M) \\ y(i) &= b_0x(i) + b_1x(i-1) + \dots + b_Mx(k-M) - [a_1y(i-1) + \dots + a_Ny(k-N)] \end{aligned} \quad (2.18)$$

A equação de diferenças apresentada em (2.18) permite que este sistema possa ser implementado em processadores digitais. Não só isso, mas pode-se também produzir diferentes modelos em espaço de estado, dependendo somente das variáveis de estado escolhidas - geralmente, até por questão de facilidade de implementação, são escolhidas grandezas físicas que podem ser mensuradas.

O sistema em tempo discreto modelado pela equação em diferenças em (2.15) pode ser modelado através do seguinte modelo em espaço de estados discreto:

$$x(i+1) = A_mx(i) + B_mu(i) \quad (2.19)$$

$$y(i) = C_mx(i) + Du(i) \quad (2.20)$$

onde

$$A_m = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{L-1} & -a_L \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}, B_m = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, C_m = [\beta_1 \ \beta_2 \ \dots \ \beta_L], D_m = b_0$$

onde, nestes casos, temos que

$$L = \max(M, N)$$

$$\beta_i = b_i - b_0a_i, \quad i = 1, 2, \dots, L$$

com  $M$  sendo o valor máximo da ordem dos zeros e  $N$  o valor máximo da ordem dos pólos da planta.

Sejam  $A_c$ ,  $B_c$  e  $C_c$  as matrizes de estado, entrada e saída de um modelo de espaço de estados em tempo contínuo. Segundo Chen (1999), as matrizes do modelo em tempo discreto podem ser obtidas da seguinte através dos seguintes cálculos:

$$A_m = e^{A_c t_s}, \quad B_m = \left( \int_0^{t_s} e^{A_c \tau} d\tau \right) B_c, \quad C = C_c, \quad (2.21)$$

Sendo  $t_s$  o tempo de amostragem. Este será o método usado neste trabalho para obter o modelo em espaço de estado em tempo discreto.

## 2.4 Otimização

De forma geral, o principal objetivo de desenvolvimento de um sistema é a capacidade de gerar resultados satisfatórios com a menor quantidade de recursos dispensados e no menor tempo possível para sua execução, garantindo o reinício deste ciclo. Entende-se que diversas formas e modelos são capazes de apresentar o resultado selecionado, porém muitas vezes com situações adversas que devem ser levadas em consideração quando da decisão de uma solução em detrimento de outra. Portanto, a comprovação e escolha da melhor solução para um dado problema, levando em conta diversos fatores e alternativas, é considerado como o conceito de otimização (SARKER; NEWTON, 2007).

Um problema de otimização unidimensional básico pode, então, ser descrito como apresentado em (2.22), onde uma função  $f(x)$  pode ter seu valor maximizado ou minimizado de acordo com elementos presentes em um conjunto  $\chi$ .

$$\text{Maximize/Minimize } f(\mathbf{x}) \text{ sujeito a } x \in \chi \quad (2.22)$$

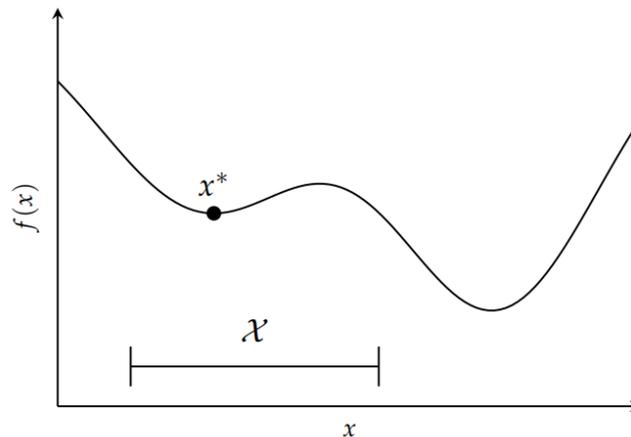
A função  $f(x)$  cujo valor máximo ou mínimo se deseja encontrar é chamada de “função objetivo” ou “função custo”. Já o conjunto de elementos  $\chi$  que podem ser inseridos na função e gerar resultados é chamado de “conjunto viável”. Por fim, o valor que satisfaz a condição de máximo ou mínimo dentro deste conjunto é chamada de “solução”, e é normalmente apresentado como  $x^*$ . Deve-se destacar, entretanto, que o fato de uma solução ser encontrada entre os termos de um conjunto viável não implica, necessariamente, que o valor ideal da função foi obtido (seja ele mínimo ou máximo), mas que o valor ideal do conjunto viável foi encontrado.

Como demonstrado na Figura 4, um conjunto viável pode gerar um resultado mínimo, onde a solução é a menor possível entre os componentes do grupo utilizado, mas há a possibilidade de existirem valores ainda melhores para satisfazer a otimização. Este fato demonstra como o conjunto viável utilizado é a base para a obtenção dos melhores resultados possíveis no processo.

### 2.4.1 Restrições

Muitos problemas de otimização contém uma ou mais “regras” que limitam quais são os conjuntos de soluções possíveis - as chamadas “restrições”. O conjunto de todas as regras aplicadas ao problema define o conjunto viável, que contém todos elementos cujos valores não violam nenhuma das restrições do problema. Isto pode ser observado na Figura 5, que demonstra o conjunto viável definido pelas restrições apresentadas em (2.23). Deve-se tomar cuidado, entretanto, com a definição das restrições aplicadas, já que tais definições podem transformar o problema de otimização em um problema sem solução.

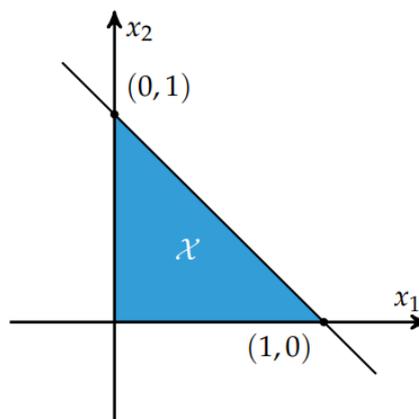
Figura 4 – Exemplo de um problema de otimização unidimensional



Fonte: (KOCHENDERFER; WHEELER, 2019)

$$\text{Minimize } f(x_1, x_2) \text{ sujeito a } x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 1 \quad (2.23)$$

Figura 5 – Representação de um conjunto viável de acordo com a Equação 2.23



Fonte: (KOCHENDERFER; WHEELER, 2019)

### 2.4.2 Pontos Críticos

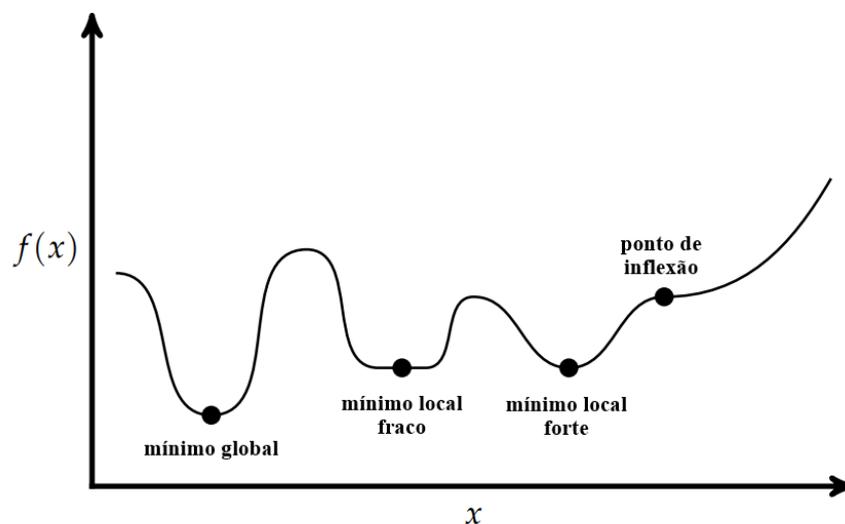
Quando se fala de processos de otimização, os pontos mais importantes para a análise são os pontos conhecidos como pontos críticos. Estes pontos são áreas de interesse pois são pontos pertencentes à função objetivo que possuem derivada igual a zero, o que normalmente pode indicar pontos de máximos ou mínimos da função, como demonstrado na Figura 6. Entretanto, deve-se destacar que é geralmente difícil comprovar que um ponto é o mínimo ou máximo global, de forma que o melhor que pode ser feito é verificar se tal ponto é um mínimo ou máximo local (KOCHENDERFER; WHEELER, 2019).

Um ponto é considerado como mínimo ou máximo local forte se é o único ponto que atinge este valor em sua vizinhança. Já um mínimo ou máximo local fraco não é o

único ponto que atinge este valor quando comparado aos pontos vizinhos. Por fim, deve-se ter em mente que o fato de possuir uma derivada igual a zero não significa que o ponto é, necessariamente, um ponto de máximo ou mínimo.

Os pontos chamados de “Pontos de Inflexão” são pontos na função em que a derivada é nula, porém os pontos vizinhos apresentam as declividades da reta tangente à função (ou seja, as derivadas da função) com o mesmo sinal. Com esta situação, nota-se que a derivada zero da função objetivo é uma condição necessária para mínimos e máximos locais, mas não é condição suficiente (KOCHENDERFER; WHEELER, 2019).

Figura 6 – Exemplo de pontos críticos de interesse em algoritmos de otimização em uma função com uma única variável

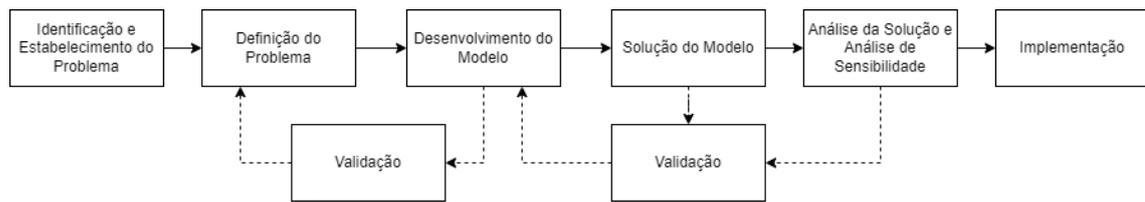


Fonte: Adaptado de (SARKER; NEWTON, 2007)

### 2.4.3 Fluxo do Processo de Otimização

Segundo Sarker e Newton (2007), o processo de otimização é um integrante imprescindível nas tomadas de decisão e design de processos, cuja premissa nada mais é do que a aplicação de tomadas de decisão frente a sistemas da vida real buscando obter o melhor resultado (também conhecido como resultado ótimo). Este é um processo de gerenciamento que pode se tornar bastante complexo, pois as tomadas de decisão podem utilizar diferentes métodos para um mesmo problema e, acima de tudo, o melhor resultado pode ser obtido através de métodos que não são inicialmente óbvios, ou ainda pode ser obtido por diferentes métodos. Portanto, de forma geral, pode-se entender que o processo de otimização de um sistema pode ser resumido conforme demonstrado na Figura 7.

Figura 7 – Fluxo de Desenvolvimento de Otimização



Fonte: Adaptado de (SARKER; NEWTON, 2007)

De forma simples, pode-se resumir cada um dos passos deste processo da seguinte forma:

- **Identificação e Estabelecimento do Problema:** Desenvolver um entendimento claro do problema, geralmente através de observações detalhadas dos sistemas analisados. Em cenários ideais, é possível utilizar diferentes métodos para atingir um determinado objetivo e existir dúvidas genuínas sobre qual método é o que produz o melhor resultado;
- **Definição do Problema:** Este é o processo de definir exatamente o problema a ser trabalhado, determinar o objetivo a ser alcançado, as medidas de efetividade utilizadas para comparar alternativas, as presunções, restrições e limitações dos sistemas, assim como verificar todas as informações possíveis sobre recursos e parâmetros envolvidos no problema;
- **Desenvolvimento do Modelo:** Este é o processo onde um modelo matemático do sistema é desenvolvido, buscando replicar da melhor maneira possível o comportamento real do sistema em questão.
- **Solução do Modelo:** Após o problema ser definido e um modelo matemático que o descreva ser desenvolvido, é possível encontrar uma solução através deste modelo descrito. As soluções iniciais geralmente iniciam uma sequência de operações e avaliações matemáticas com o objetivo de encontrar uma solução idealmente mais apta ao problema em questão;
- **Análise da Solução e Análise de Sensibilidade:** Esta fase tem como foco a observação de como os dados de entrada do sistema afetam a solução ótima do sistema, permitindo a verificação de quão robusta é uma solução
- **Validação:** Esta é a fase onde é possível realizar uma investigação do sistema como um todo com foco na melhoria da performance do sistema, sendo o momento onde alterações podem ser entendidas como necessárias, ocasionando alterações no modelo proposto ou até mesmo (ainda que não desejável), alterações na definição do problema em questão.

- **Implementação:** Fase final de uma operação, onde a solução encontrada após extensivos testes e análises é finalmente implementado. É importante que a performance do sistema seja analisada, garantindo que a implementação foi corretamente efetuada e que a performance não decaia com o tempo.

#### 2.4.4 Categorias de Otimização

Por fim, considerando que o processo de otimização é sempre associado a modelos que descrevem o funcionamento de um sistema, é possível criar certas categorias amplas que abarcam problemas similares, de forma que Diwekar (2008) define que os processos de otimização podem ser divididos de acordo com as funções objetivos, restrições e conjuntos viáveis da seguinte maneira:

- **Programação Linear (PL):** A função objetivo e as restrições são lineares, enquanto o os conjunto viável é composto de variáveis escalares e contínuas;
- **Programação Não-Linear (PNL):** A função objetivo e/ou restrições são não-lineares. O conjunto viável é composto de variáveis escalares e contínuas;
- **Programação de Inteiros (PI):** O conjunto viável é composto de variáveis escalares e inteiras;
- **Programação Mista de Inteiros Lineares (PMIL):** A função objetivo e as restrições são lineares. O conjunto viável é composto de variáveis escalares, algumas inteiras e outras, contínuas;
- **Programação Mista de Inteiros Não-Lineares (PMINL):** Se trata de um problema de PNL cujo conjunto viável é composto de variáveis inteiras assim como variáveis contínuas;
- **Otimização Discreta:** Problemas de otimização que possuem conjuntos viáveis compostos de variáveis discretas. Inclui processos do tipo PI, PMIL e PMINL;
- **Controle Ótimo:** As variáveis do conjunto viável são vetores;
- **Otimização Estocástica:** Processo de otimização cujas funções objetivo e/ou suas restrições possuem variáveis incertas (ou seja, sujeitas a randomicidade); e
- **Otimização Multiobjetiva:** Problemas que envolvem mais de uma função objetivo.

Tais categorizações são criadas como forma de compreender o quanto o processo de otimização é abrangente em relação aos tipos de situações que pode tratar.

## 2.5 Métodos Heurísticos de Otimização

Os métodos tradicionais de otimização, que buscam encontrar uma solução ótima exata para um sistema, dentro das restrições selecionadas para tal. Entretanto, sua aplicabilidade torna-se complexa em casos onde os problemas a serem otimizados não são bem definidos ou então precisam tratar de problemas complexos, de grandes escalas ou cujos espaços de soluções viáveis são muito grandes. Isto porque, na busca da melhor solução, acaba-se utilizando elevados recursos computacionais e, normalmente como uma consequência deste primeiro fato, torna-se um processo lento de ser concluído.

Os métodos heurísticos de otimização, em contrapartida, são técnicas práticas que buscam soluções viáveis para problemas complexos, sem garantir a solução ótima em caráter global, mas que são capazes de satisfazer os requisitos estipulados. São métodos de otimização cujas regras de implementação e cálculo são baseadas em situações empíricas, experiências anteriores e aproximações, permitindo decisões rápidas e eficazes, sendo que sua utilização é indicada principalmente em casos onde não é possível definir claramente os problemas a serem otimizados, ou ainda quando há uma vasta amplitude de soluções que precisam ser verificadas.

Não é o foco desta dissertação discorrer de forma extensiva sobre os diversos métodos heurísticos que são atualmente disponibilizados, mas compreende-se importante ao menos discorrer brevemente sobre eles, devido sua importância para o trabalho em questão.

### 2.5.1 *Simulated Annealing* (SA)

Método proposto inicialmente por [Kirkpatrick, Jr. e Vecchi \(1983\)](#), é um método que foi inspirado no processo metalúrgico conhecido como “recozimento” (*annealing* em inglês), onde um material é aquecido e em seguida lentamente resfriado, garantindo uma estrutura molecular mais organizada. Este método possui como vantagem a simplicidade de ser implementado e, mais importante, é capaz de escapar de mínimos locais, garantindo que seu resultado é muito próximo do ótimo global. Entretanto, tem a tendência de ser um processo de otimização mais lento exatamente devido a esta capacidade de verificar amplos conjuntos viáveis, além de ser extremamente sensível aos parâmetros iniciais que foram escolhidos para o processo de otimização.

### 2.5.2 Busca Tabu (*Taboo Search* - TS)

O algoritmo de busca tabu foi pela primeira vez apresentado por [Glover \(1989\)](#) e seu algoritmo é baseado em “evitar” analisar soluções que sejam muito próximas da solução atualmente analisada por um certo número de iterações, desta forma evitando ficar preso nos mínimos ou máximos locais. As soluções atuais são analisadas através da função

custo, já as soluções consideradas “próximas” são determinadas através de uma função chamada “função de vizinhança” que, como o próprio nome diz, é capaz de determinar soluções consideradas próximas e que devem ser evitadas. Estas soluções são adicionadas em uma lista, conhecida como lista tabu, que determina quais valores não podem ser utilizados pelo algoritmo por um certo número de iterações, até que a melhor solução encontrada é apresentada como resultado desta otimização. Este método possui como vantagens sua capacidade de tratar problemas que possuem grandes conjuntos viáveis, além de sua flexibilidade de implementação. Entretanto, é um método extremamente complexo de ser implementado, além de ser dependente de certo nível de ajuste fino nos parâmetros utilizados inicialmente.

## 2.6 Algoritmo Genético

Outro algoritmo de otimização heurístico, além dos dois citados anteriormente, se trata do chamado algoritmo genético. O algoritmo genético (abreviado como AG) é um algoritmo de otimização numérica inspirado nos conceitos biológicos de seleção natural e de genética (COLEY, 1999). Este algoritmo foi desenvolvido por John Holland e, apesar de ser uma excelente técnica de otimização para funções complexas, não se trata de um algoritmo criado especificamente para resolver qualquer tipo particular de problema (como problemas de classificação ou até mesmo de otimização, como citado), mas sim de uma simulação de alto nível de um sistema adaptativo biológico - a evolução - como lembrado por DeJong (1992). Ainda assim, é inegável que tal algoritmo possui elevado sucesso como técnica de otimização, fato este que permite sua aplicação em diferentes áreas de conhecimento, desde simulações de mercado (PRICE, 1997), design de aeronaves (DUNN, 1997) e até mesmo design molecular (TAYAL; DIWEKAR, 2001), entre outros campos de conhecimento.

Ao contrário de outros algoritmos utilizados para otimização, onde todo o processo inicia a partir de uma solução inicial e cujas verificações são realizadas uma solução por vez, o algoritmo genético inicia seu processo a partir do que é denominado “população”, que nada mais é do que um conjunto de soluções geradas de forma aleatória, e cada novo teste utiliza uma nova população de soluções geradas a partir das melhores soluções da população anterior - numa clara alusão ao conceito de “seleção natural” e de “sobrevivência do mais apto”. De fato, grande parte da terminologia utilizada para este algoritmo é originária do conceito biológico de genética, como demonstrado a seguir.

### 2.6.1 População

Como citado anteriormente, uma população nada mais é do que um conjunto de “indivíduos”, neste caso, um conjunto aleatório de soluções para a função objetivo que se

deseja otimizar. Após processos combinatórios realizados entre os indivíduos pertencentes a esta população, novos indivíduos são gerados, dando origem a uma nova população que será utilizada em um novo ciclo do processo de otimização. Um dos parâmetros-chave neste algoritmo é a determinação do tamanho da população a ser utilizada (DIWEKAR, 2008).

### 2.6.2 Indivíduos

Como demonstrado anteriormente, a definição de indivíduo em algoritmo genético se refere a uma possível solução para o problema em mãos - neste caso, um problema de otimização. Cada indivíduo é composto por um número de termos igual ao número de variáveis que se deseja otimizar. A [Figura 8](#) apresenta um indivíduo que possui 2 termos.

### 2.6.3 Cromossomo

No estudo da genética, um indivíduo possui diversas características em seu DNA, cada uma destas descrita em seu código genético através de seus cromossomos. Em algoritmos genéticos, a denominação de cromossomo é, então, aplicada à codificação de cada um dos termos que compõe um indivíduo pertencente à uma população. Isto se deve ao fato de que todos os indivíduos tem a possibilidade de “transmitir” suas características individuais (“genéticas”) para as próximas gerações, assim como cromossomos contém características genéticas que podem ser passadas para as futuras gerações. Geralmente, os valores iniciais dos cromossomos nos indivíduos da população inicial de um algoritmo genético são escolhidos de forma aleatória, tentando evitar ao máximo uma otimização que seja “viciada” logo de início. Deve-se destacar que, em um primeiro momento, os cromossomos eram transformados em formato binário para a execução do algoritmo, porém atualmente já é possível a utilização cromossomos com números em base-10 por exemplo. A apresentação em formato binário é utilizada como forma de mais fácil compreensão dos conceitos apresentados.

### 2.6.4 Gene

Um cromossomo é composto de vários genes. De maneira simplificada, poderia ser dito que cada gene é responsável pela transmissão de uma característica genética entre as várias características que um cromossomo pode transmitir (por exemplo, cor do cabelo, cor dos olhos, entre outras). No caso do algoritmo genético, pode-se dizer que cada gene pode ser um único algarismo ou um conjunto de algarismos de um cromossomo. A [Figura 8](#) apresenta uma ilustração de como cromossomos e genes se relacionam no caso de números binários.



pais de uma próxima população. Um exemplo de alocação de indivíduos em uma roleta é apresentado na [Figura 9](#).

### 2.6.7 Cruzamento

O processo de cruzamento (também conhecido como *crossover*) permite que os indivíduos selecionados como pais da próxima população sejam capazes de realizar “trocas de informações genéticas” de forma similar aos métodos utilizados pelos organismos vivos em suas reproduções sexuadas. No caso do algoritmo genético, esta troca é feita a partir de seleções aleatórias de parte dos genes dos pais que são trocadas entre si, gerando novos indivíduos. Existem diversos métodos utilizados para a geração de novos indivíduos através do cruzamento, sendo que alguns destes são apresentados na [Figura 10](#).

### 2.6.8 Mutação

O processo de mutação é outro operador de busca utilizado no algoritmo genético, geralmente executado após o *crossover*. Assim como sua contra-parte biológica, ocorre de forma aleatória através de alterações nos cromossomos (indivíduos) da nova população, garantindo uma variabilidade da população.

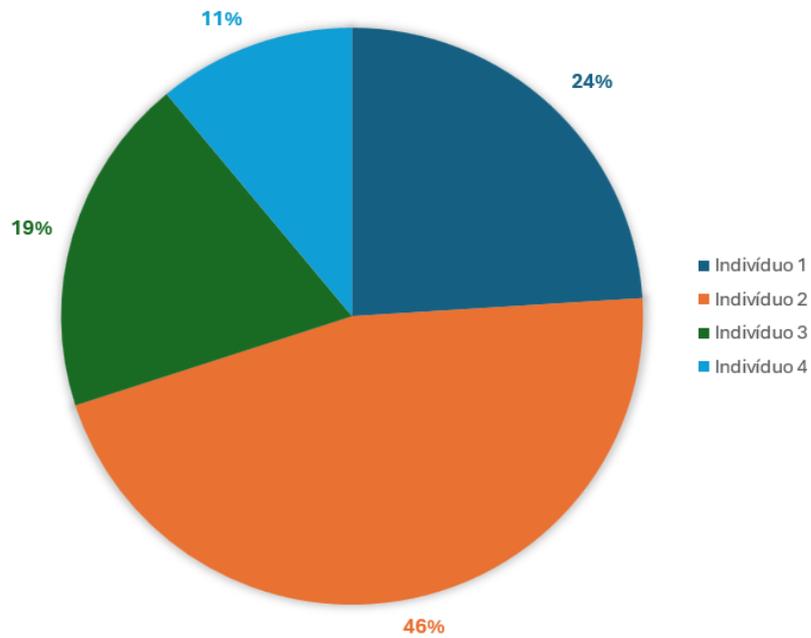
No caso do algoritmo genético, a mutação ocorre a partir da seleção aleatória de um gene de um indivíduo, seguida da alteração do valor deste gene dentro do alcance permitido. A principal vantagem deste processo, quando em níveis baixos, é a capacidade de prevenir que os indivíduos possuam sempre genes que não são alterados, evitando que uma característica seja indefinidamente idêntica durante todo o processo. Entretanto, nos casos onde taxas mais elevadas de mutação são aplicadas, corre-se o risco de transformar o processo de criação de novas populações em uma simples geração aleatória de valores de indivíduos, devido a elevada taxa de alteração das características dos indivíduos ([DIWEKAR, 2008](#)). Um exemplo de mutação em base binária é apresentado na [Figura 10](#).

A importância do processo de mutação na execução do algoritmo tem sua importância maior no fato de que é um dos métodos utilizados para tentar evitar que o processo seja preso em um mínimo ou máximo local, inserindo um nível de aleatoriedade nas populações e garantindo que o algoritmo possa verificar a aptidão de indivíduos que normalmente não seriam trazidos para a análise.

### 2.6.9 Gerações

Todos os processos anteriormente citados, ao serem executados em conjunto, definem uma geração do algoritmo genético. Gerações são importantes pois tendem a trazer novos valores para os indivíduos, gerando novos testes de aptidão e, quando corretamente configurados, se aproximando estatisticamente de um resultado ótimo, além da possibili-

Figura 9 – Exemplo de uma roleta de seleção de indivíduos em algoritmo genético.

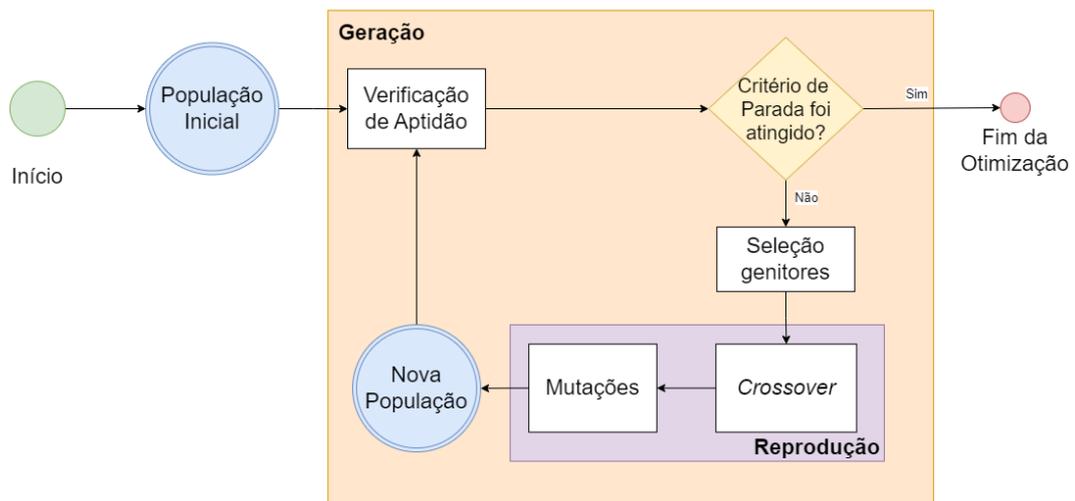


Fonte: Autor

dade de sua utilização como um dos vários critérios de parada que podem ser utilizados para encerrar a execução do algoritmo genético.

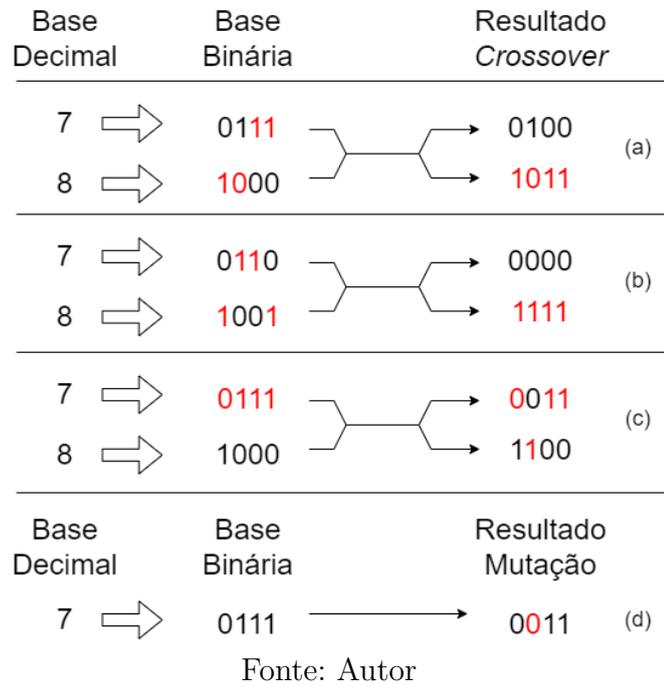
Por fim, a [Figura 11](#) apresenta um fluxo simplificado de execução de um algoritmo genético.

Figura 11 – Fluxograma de operação de um algoritmo genético



Fonte: Autor

Figura 10 – Exemplos de cruzamentos e mutações de algoritmos genéticos em números em base binária. Em (a) é apresentado o “Cruzamento com corte em um ponto” (b) demonstra o “Cruzamento com corte em dois pontos” e (c) demonstra o “Cruzamento de gene”. Por fim, em (d) é apresentado o processo de mutação.



## 2.7 GPC Convencional

### 2.7.1 Modelo Aumentado

Segundo Wang (2009), sistemas de Controle por Modelo Preditivo são desenvolvidos com base em um modelo matemático da planta que se deseja controlar. Além disso, ao se utilizar um modelo de espaço de estados, a informação atual necessária para previsões futuras pode ser representada pela variável de estado no tempo presente. Assim sendo, uma planta de ordem  $n$  com uma entrada e uma saída (do inglês *Single Input, Single Output - SISO*) pode ser definida, no tempo discreto, pelas seguintes equações:

$$x_m(i + 1) = A_m x_m(i) + B_m u(i) \tag{2.24}$$

$$y(i) = C_m x_m(i) + D_m u(i) \tag{2.25}$$

onde  $u(i)$  é a variável manipulada ou manipulável (também chamada de variável de entrada),  $y(i)$  é a saída do sistema, e  $x_m(i)$  é o vetor das variáveis de estado com uma dimensão assumida em  $n$ , enquanto  $A_m$  é a chamada Matriz de Transição de Estados,  $B_m$  é a Matriz de Entradas,  $C_m$  é a Matriz de Saídas e  $D_m$  é a Matriz de Transmissão Direta. Inicialmente considera-se que a planta não possui um integrador embutido. Entretanto, para que o erro em regime permanente desta planta seja zero para as entradas do tipo degrau, é necessário que um integrador seja adicionado a planta, assim como é necessário que este integrador seja adicionado para que o controlador GPC possa ser desenvolvido em seu modelo aumentado, por isso considera-se a matriz  $D_m$ .

Se considerarmos o princípio do horizonte retrocedente, onde a informação atual da planta é necessária para predição e controle, assume-se que a entrada  $u(i)$  não pode afetar a saída  $y(i)$  ao mesmo tempo, portanto  $D_m$  acaba se igualando a 0. Desta forma, o modelo de espaço de estados é simplificado e se apresenta como

$$x_m(i+1) = A_m x_m(i) + B_m u(i) \quad (2.26)$$

$$y(i) = C_m x_m(i) \quad (2.27)$$

Usando a equação (2.26) como referência, pode-se então inferir que o vetor das variáveis de estado no instante atual  $x_m(i)$  pode ser descrito como demonstrado abaixo.

$$x_m(i) = A_m x_m(i-1) + B_m u(i-1) \quad (2.28)$$

Ao se subtrair e (2.28) de (2.26), obtém-se a seguinte equação:

$$x_m(i+1) - x_m(i) = A_m x_m(i) + B_m u(i) - A_m x_m(i-1) - B_m u(i-1) \quad (2.29)$$

$$x_m(i+1) - x_m(i) = A_m (x_m(i) - x_m(i-1)) + B_m (u(i) - u(i-1))$$

Em (2.30), (2.31), e (2.32) são apresentadas, respectivamente, as definições para a diferença das variáveis de estado presente  $\Delta x_m(i)$ , de estado futuro  $\Delta x_m(i+1)$  e da variável de controle  $\Delta u(i)$ , onde  $\Delta$  é o operador de diferença de primeira ordem.

$$\Delta x_m(i) \triangleq x_m(i) - x_m(i-1) \quad (2.30)$$

$$\Delta x_m(i+1) \triangleq x_m(i+1) - x_m(i) \quad (2.31)$$

$$\Delta u(i) \triangleq u(i) - u(i-1) \quad (2.32)$$

Com estas 3 definições, é possível reescrever (2.29) como (2.33).

$$\Delta x_m(i+1) = A_m \Delta x_m(i) + B_m \Delta u(i) \quad (2.33)$$

No modelo aumentado descrito por Wang (2009), os sinais  $\Delta x_m(i)$  e  $y(i)$  são associados entre si através de um novo vetor de variáveis de estado, descrito em (2.34), onde  $T$  representa a transposta de uma matriz.

$$x(i) = [\Delta x_m(i)^T \quad y(i)]^T, \quad (2.34)$$

Para que seja possível determinar o modelo aumentado em espaço de estados utilizado no GPC, também é necessário encontrar o valor de  $y(i+1)$ . Utilizando a mesma base lógica utilizada para determinar (2.28), tem-se que este valor pode ser encontrado através de (2.35)

$$y(i+1) = C_m x_m(i+1) \quad (2.35)$$

Utilizando (2.35) e subtraindo  $y(i)$  deste valor, além de utilizar os cálculos para a obtenção de  $x_m(i+1)$  e de  $x_m(i)$ , tem-se que  $y(i+1)$  é encontrado através do cálculo a seguir.

$$\begin{aligned}
y(i+1) - y(i) &= C_m x_m(i+1) - C_m x_m(i) \\
y(i+1) &= C_m (x_m(i+1) - x_m(i)) + y(i) \\
y(i+1) &= C_m [(A_m x_m(i) + B_m u(i)) - (A_m x_m(i-1) + B_m u(i-1))] + y(i) \\
y(i+1) &= C_m [(A_m x_m(i) - A_m x_m(i-1)) + (B_m u(i) - B_m u(i-1))] + y(i) \\
y(i+1) &= C_m A_m \Delta x_m + C_m B_m \Delta u(i) + y(i)
\end{aligned} \tag{2.36}$$

Com tais informações, é possível descrever o modelo aumentado do controlador do tipo GPC convencional conforme demonstrado em (WANG, 2009), que se apresenta da seguinte forma:

$$\begin{aligned}
\begin{bmatrix} \Delta x_m(i+1) \\ y(i+1) \end{bmatrix} &= \overbrace{\begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix}}^A \overbrace{\begin{bmatrix} \Delta x_m(i) \\ y(i) \end{bmatrix}}^{x(i)} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^B \Delta u(i) \\
y(i) &= \overbrace{\begin{bmatrix} 0_m & 1 \end{bmatrix}}^C \overbrace{\begin{bmatrix} \Delta x_m(i) \\ y(i) \end{bmatrix}}^{x(i)}
\end{aligned} \tag{2.37}$$

Na equação apresentada em (2.37), as matrizes  $A$ ,  $B$  e  $C$  são consideradas as como variáveis do modelo aumentado, responsáveis por descrever o cito modelo. Já variável  $0_m$ , que aparece nas matrizes  $A$  e  $C$  da equação, é uma matriz linha de zeros, com comprimento  $n$ , de forma que  $0_m = \overbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}}^n$ .

## 2.7.2 Predição

Após a determinação do modelo matemático, o próximo passo a ser dado é calcular a saída prevista da planta, utilizando os sinais de controle futuros como as variáveis manipuláveis. Esta previsão é descrita dentro de uma janela de otimização. Considerando que, no instante de amostragem  $i > 0$ , o vetor das variáveis de estados do modelo aumentado  $x(i)$  é conhecido através de medições, a trajetória do controle futuro pode, então, ser descrita através dos valores em (2.38), onde  $N_C$  é um valor denominado “horizonte de controle”, responsável por definir o número de parâmetros utilizados para capturar a trajetória do controle futuro.

$$\Delta u(i), \Delta u(i+1), \dots, \Delta u(i+N_C-1) \tag{2.38}$$

Com  $x(i)$  em mãos, as variáveis de estado futuras são previstas para um número  $N_P$  de amostras, onde  $N_P$  é o comprimento da janela de otimização e  $N_C \leq N_P$ .

Assim como mostrado na introdução, as variáveis de estado futuras são apresentadas como demonstrado em (2.39) e tais denotações podem ser lidas como “variáveis de estado  $x$  calculada no intervalo  $i + 1, i + 2, \dots, i + N_P$  com as informações disponíveis no instante  $i$ ”. Para simplificar a notação nas demonstrações matemáticas, o termo  $|i$  será eliminado.

$$x(i + 1|i), x(i + 2|i), \dots, x(i + N_P|i) \quad (2.39)$$

De acordo com (WANG, 2009), ao levar todos os dados apresentados em consideração, é possível calcular as variáveis de estado futuras e as variáveis de saída previstas, respectivamente, conforme apresentado a seguir.

$$\begin{aligned} x(i + 1) &= Ax(i) + B\Delta u(i) \\ x(i + 2) &= A^2x(i) + AB\Delta u(i) + B\Delta u(i + 1) \\ &\vdots \\ x(i + N_P) &= A^{N_P}x(i) + A^{N_P-1}B\Delta u(i) + \\ &A^{N_P-2}B\Delta u(i + 1) + \dots + A^{N_P-N_C}B\Delta u(i + N_C - 1) \end{aligned} \quad (2.40)$$

$$\begin{aligned} y(i + 1) &= CAx(i) + CB\Delta u(i) \\ y(i + 2) &= CA^2x(i) + CAB\Delta u(i) + CB\Delta u(i + 1) \\ &\vdots \\ y(i + N_P) &= CA^{N_P}x(i) + CA^{N_P-1}B\Delta u(i) + \\ &CA^{N_P-2}B\Delta u(i + 1) + \dots + CA^{N_P-N_C}B\Delta u(i + N_C - 1) \end{aligned} \quad (2.41)$$

Ao observar os conjuntos de equações apresentados em (2.40) e (2.41), nota-se que todas as variáveis preditas são dependentes das variáveis de estado atual  $x(i)$  e da variação das variáveis de controle  $\Delta u(i + j)$ , onde  $j = 0, 1, \dots, N_C - 1$ . Utilizando tais equações, é possível definir, então,  $Y$  como o vetor de saídas preditas e  $\Delta U$  como o vetor de diferença das variáveis de controle, conforme apresentado abaixo.

$$Y = [y(i + 1) \quad y(i + 2) \quad \dots \quad y(i + N_P)]^T \quad (2.42)$$

$$\Delta U = [\Delta u(i) \quad \Delta u(i + 1) \quad \dots \quad \Delta u(i + N_C - 1)]^T \quad (2.43)$$

Os vetores apresentados em (2.42) e (2.43), então, podem ser relacionados através de uma equação matricial, como demonstrado a seguir:

$$Y = Fx(i) + \phi\Delta U \quad (2.44)$$

onde:

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_P} \end{bmatrix} \quad \text{e} \quad \phi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ CA^{N_P-1}B & CA^{N_P-2}B & \dots & CA^{N_P-N_C}B \end{bmatrix} \quad (2.45)$$

### 2.7.3 Otimização do Modelo Aumentado

Entende-se que o objetivo principal de um sistema de controle preditivo é fazer com que suas saídas previstas sejam as mais próximas possíveis de um sinal de referência  $r(i)$ , com tempo de amostragem  $t_s$ , onde se assume que esta referência permanece constante durante toda a janela de otimização. Este objetivo, então, é transformado em encontrar qual é o melhor vetor de parâmetro de controle  $\Delta U$  que permita que a função custo  $J$  determinada pela diferença entre o sinal de referência e a saída predita seja a menor possível.

Define-se que o vetor de referência tipo degrau da janela de otimização  $R_s$  é descrito como:

$$R_s^T = \overbrace{[1 \quad 1 \quad \dots \quad 1]}^{N_P} \quad (2.46)$$

Já a função custo  $J$ , segundo Wang (2009), possui a seguinte estrutura:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (2.47)$$

Nesta função custo, temos que o primeiro termo é associado ao objetivo de minimizar os erros entre a saída predita e o sinal de referência, enquanto o segundo termo demonstra a importância de  $\Delta U$  quando o objetivo da função custo é se tornar a menor possível. Já a variável  $\bar{R}$  que aparece no segundo termo da equação (2.47) é uma matriz diagonal que possui o valor:

$$\bar{R} = r_w I_{N_C}, \text{ para } r_w \geq 0 \quad (2.48)$$

Nota-se a presença de duas novas variáveis no cálculo de  $\bar{R}$ :  $I_{N_C}$  é uma matriz identidade de dimensão  $N_C$ , enquanto  $r_w$  é um parâmetro de ajuste da função custo para obtenção da performance desejada em malha fechada. Quando  $r_w = 0$ , a função custo ignora qualquer que seja a dimensão de  $\Delta U$  e o objetivo passa a ser obter o menor erro  $(R_s - Y)^T (R_s - Y)$  possível. No caso de  $r_w$  ter um alto valor numérico positivo,  $J$  apresenta a situação onde a magnitude de  $\Delta U$  deve ser considerada com bastante atenção durante o processo de redução do valor do erro  $(R_s - Y)^T (R_s - Y)$ .

Para encontrar, então, o valor ótimo de  $\Delta U$  que irá gerar o valor mínimo da função custo, pode-se aplicar o critério da primeira derivada nesta função. Substituindo os valores necessários e calculando a primeira derivada em função do próprio  $\Delta U$ , a função custo se iguala a:

$$\frac{\partial J}{\partial \Delta U} = 2\phi^T (R_s - Fx(i)) + 2(\phi^T \phi + \bar{R}) \Delta U \quad (2.49)$$

Segundo o próprio critério da primeira derivada, para que possamos obter o valor mínimo de  $J$ , o valor desta derivada deve ser igualado a zero. Assim sendo, temos que a

solução ótima para o sinal de controle é descrita por Wang (2009) como:

$$\Delta U = (\phi^T \phi + \bar{R})^{-1} \phi^T (R_S - Fx(i)) \quad (2.50)$$

desde que se considere que a matriz  $(\phi^T \phi + \bar{R})^{-1}$  exista. Esta matriz, na literatura de otimização, é chamada de matriz Hessiana. Assim sendo, podemos reescrever a equação (2.50) como:

$$\Delta U = (\phi^T \phi + \bar{R})^{-1} (\phi^T R_S - \phi^T Fx(i)) \quad (2.51)$$

onde  $R_s = [1 \quad 1 \quad \cdots \quad 1]_{1 \times N_P}^T$

#### 2.7.4 Sistema em Malha Fechada

Em um sistema de malha fechada, a expressão  $(\phi^T \phi + \bar{R})^{-1} \phi^T R_S$ , encontrada em (2.51), corresponde à mudança do sinal de referência, enquanto o termo  $(\phi^T \phi + \bar{R})^{-1} \phi^T Fx(i)$  corresponde ao estado do controle de realimentação dentro da estrutura do controle preditivo. Ambos os termos são dependentes dos parâmetros do sistema, portanto, em um sistema invariante no tempo, estes termos não sofrem nenhuma alteração. Pelo princípio de controle horizontal de recuo, somente o primeiro elemento de  $\Delta U$  no tempo  $i$ , é utilizado para calcular a lei de controle da planta. Portanto:

$$\begin{aligned} \Delta u(i) &= \overbrace{[1 \quad 0 \quad \cdots \quad 0]}^{N_c} (\phi^T \phi + \bar{R})^{-1} (\phi^T \bar{R}_S - \phi^T Fx(i)) \\ \Delta u(i) &= K_y r(i) - K_{mpc} x(i) \end{aligned} \quad (2.52)$$

onde  $K_y$  é o primeiro elemento de  $(\phi^T \phi + \bar{R})^{-1} (\phi^T \bar{R}_S)$  e  $K_{mpc}$  é a primeira linha de  $(\phi^T \phi + \bar{R})^{-1} (\phi^T Fx(i))$ .

A equação de  $\Delta u(i)$  se encontra num formato padrão do controle de estado linear retroalimentado invariante no tempo, enquanto o vetor do ganho do estado de realimentação é descrito por  $K_{mpc}$ . Sabendo que o modelo aumentado de controle para  $x(i+1)$  é descrito por (2.26), o sistema em malha fechada pode ser obtido ao substituir  $\Delta u(i)$  nesta equação aumentada. Ao trocarmos o índice  $i$  por  $k$ , encontramos a equação em malha fechada a seguir:

$$\begin{aligned} x(i+1) &= Ax(i) + B(K_y r(i) - K_{mpc} x(i)) \\ x(i+1) &= (A - BK_{mpc})x(i) + BK_y r(i) \end{aligned} \quad (2.53)$$

Por ser um sistema em malha fechada, é possível então encontrar os autovalores do sistema através da equação característica indicada e da equação apresentada em (2.54).

$$\det[\lambda I - (A - BK_{mpc})] = 0 \quad (2.54)$$

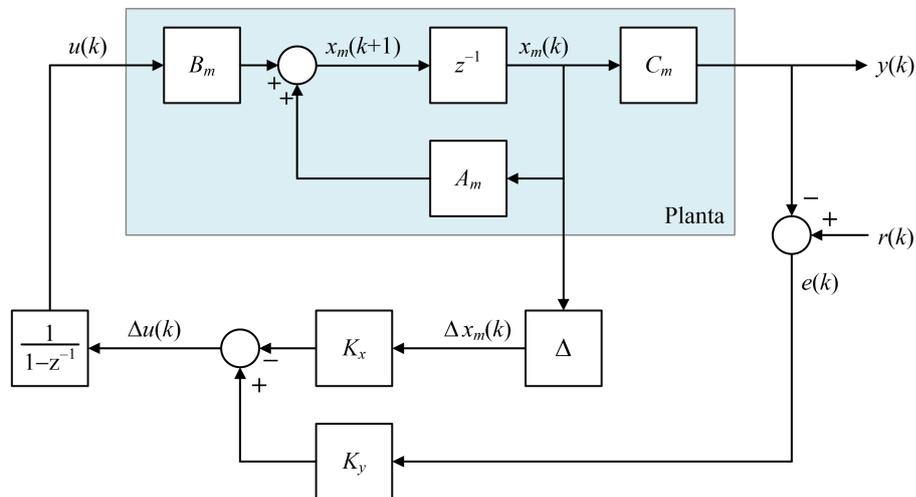
Por causa da estrutura especial das matrizes  $C$  e  $A$ , a última coluna de  $F$  é idêntica a  $\bar{R}_S$ , que é  $[1 \quad 1 \quad \cdots \quad 1]^T$ . Consequentemente, o ganho  $K_y$  será o último elemento de

$K_{mpc}$ . Considerando as definições de  $x(i), x(i + 1)$  e do ganho  $K_y$ , o vetor de ganho  $K_{mpc}$  pode ser rescrito da seguinte maneira:

$$K_{mpc} = [K_x \ K_y] \tag{2.55}$$

onde  $K_x$  corresponde ao vetor ganho de realimentação relacionado a  $\Delta x_m(i)$ . O ganho  $K_y$  será o ganho de realimentação relacionado a saída  $\Delta y(i)$ . Assim, o sistema GPC em malha fechada pode ser descrito segundo a Figura 12. Nesta figura, o bloco  $z^{-1}$  indica o operador de deslocamento de atraso,  $1 - z^{-1}$  é a função de transferência discreta equivalente ao operador  $\Delta$ , enquanto que  $\frac{1}{1-z^{-1}}$  corresponde a um integrador de tempo discreto.

Figura 12 – Diagrama de um bloco de um sistema GPC de malha fechada.



Fonte: (ESTRABIS, 2020)

## 2.8 Princípio do Modelo Interno para Rastreamento e Rejeição Robustos de Sinais

Segundo o Princípio do Modelo Interno, para que um controlador seja capaz de ignorar viéses (*bias*) ou perturbações em seu sistema, ele deve possuir dentro de si um modelo dos sinais e perturbações que se deseja anular. Entretanto, se considerarmos que a planta controlada pode sofrer variações nos valores de seus polos, como nos casos em que a carga no sistema é alterada, a função de transferência total pode se alterar de forma que o rastreamento ou a rejeição de sinais apresentem erros indesejáveis. Sendo assim, é necessário que o controlador seja calculado de forma a apresentar a robustez necessária para conseguir lidar com possíveis alterações de planta sem que o rastreamento ou rejeição dos sinais de entrada ou das perturbações sejam afetados.

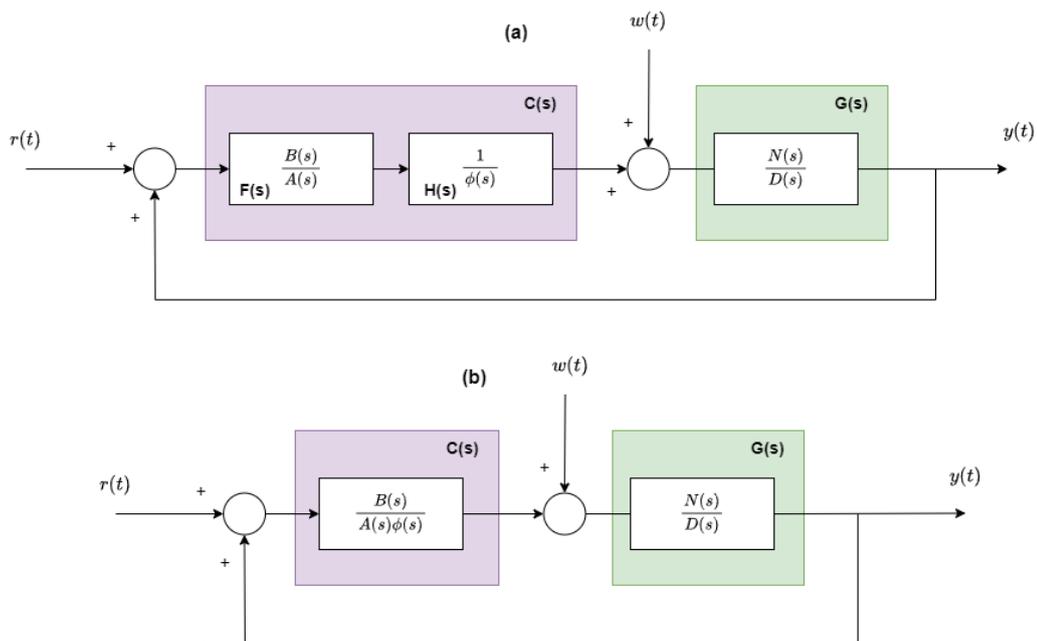
Considere uma planta como a apresentada na Figura 13, que possui função de transferência  $G(s)$ . Esta planta receberá um sinal de entrada de referência  $r(t)$  e estará

sujeita a um sinal de perturbação  $\omega(t)$ , ambos que não se aproximam de 0 quando o tempo  $t \rightarrow \infty$ . Desejamos projetar um controlador  $C(s)$  que seja capaz de rastrear e/ou rejeitar seja o sinal de referência, seja o sinal de perturbação dado. Para que a realização deste projeto seja possível, é necessário possuir em mãos algumas informações sobre os sinais que se deseja rastrear ou rejeitar. Assim sendo, pode-se considerar que as transformadas de Laplace da perturbação e da referência são dadas por:

$$R(s) = \mathcal{L}[r(t)] = \frac{N_r(s)}{D_r(s)} \quad \text{e} \quad W(s) = \mathcal{L}[\omega(t)] = \frac{N_\omega(s)}{D_\omega(s)} \quad (2.56)$$

onde os denominadores  $D_r(s)$  e  $D_\omega(s)$  são polinômios conhecidos, entretanto os numeradores  $N_r(s)$  e  $N_\omega(s)$  não são.

Figura 13 – Diagrama de um circuito que atende o design de rastreamento e rejeição robustos. (a) Apresentação de um sistema simplificado de controle em malha fechada de uma planta  $G(s)$ , com função de transferência unitária na realimentação. O controlador  $C(s)$  é composto por dois blocos de função de transferência em série. (b) Simplificação do controlador  $C(s)$  em um único bloco.



Fonte: Autor

Para projetar um controlador  $C(s)$  que seja capaz de rastrear ou rejeitar os sinais apresentados em (2.56), (CHEN, 1999) propõe que o controlador  $C(s)$  seja composto por duas funções de transferência em série,  $F(s)$  e  $H(s)$ , descritas como

$$F(s) = \frac{B(s)}{A(s)} \quad \text{e} \quad H(s) = \frac{1}{\phi(s)} \quad (2.57)$$

onde  $A(s)$  e  $B(s)$  são polinômios quaisquer, enquanto  $\phi(s)$  é o menor denominador comum calculado entre todos os polos instáveis de  $R(s)$  e  $W(s)$ . Estes polos são utilizados porque são capazes de afetar o resultado da saída  $y$  do sistema quando  $t \rightarrow \infty$ , portanto todas as raízes de  $\phi(s)$  possuem parte real positiva ou nula.

Considerando que a função de transferência da planta  $G(s) = \frac{N(s)}{D(s)}$  é estritamente própria (ou seja, que a ordem de  $D(s)$  é maior que a de  $N(s)$ ) e assumindo que  $N(s)$  e  $D(s)$  são coprimos (ou seja, que não possuem polos em comum), [Chen \(1999\)](#) afirma que, na hipótese que nenhuma das raízes de  $\phi(s)$  sejam zeros de  $G(s)$ , torna-se possível projetar um controlador  $C(s)$  capaz de rastrear  $R(s)$  e rejeitar  $W(s)$  de forma assintótica e robusta. Este controlador possui a forma

$$C(s) = \frac{B(s)}{A(s)\phi(s)}.$$

Ciente dos fatores acima elencados, é possível deduzir que  $D(s)\phi(s)$  e  $N(s)$  também são coprimos. Portanto, é possível desenvolver um controlador de forma que a equação polinomial

$$F(s) = A(s)D(s)\phi(s) + B(s)N(s) \quad (2.58)$$

possua qualquer raiz desejada e, portanto, permitirá que o controlador  $C(s)$  seja capaz de realizar os rastreamentos e rejeições desejadas.

Pode-se comprovar a capacidade de rejeição deste controlador ao se calcular a função de transferência entre  $W(s)$  e  $Y(s)$ :

$$\begin{aligned} G_{y\omega}(s) &= \frac{N(s)/D(s)}{1 + (B(s)/A(s)\phi(s))(N(s)/D(s))} \\ &= \frac{N(s)A(s)\phi(s)}{A(s)D(s)\phi(s) + B(s)N(s)} \\ G_{y\omega}(s) &= \frac{N(s)A(s)\phi(s)}{F(s)} \end{aligned} \quad (2.59)$$

Com a função de transferência obtida em (2.59), pode-se determinar que a saída  $Y_\omega(s)$  resultante da perturbação  $W(s)$  é igual a

$$Y_\omega(s) = G_{y\omega}(s)W(s) = \frac{N(s)A(s)\phi(s)}{F(s)} \frac{N_\omega(s)}{D_\omega(s)}. \quad (2.60)$$

Devido ao fato do polinômio  $\phi(s)$  ser capaz de cancelar todas as raízes instáveis de  $D_\omega(s)$ , todos os pólos de  $Y_\omega(s)$  possuem partes reais negativas. Isto significa que, no domínio do tempo,  $y_\omega(t) \rightarrow 0$  quando  $t \rightarrow \infty$ , o que comprova que  $\omega(t)$  é assintoticamente suprimido na saída.

De igual modo, é possível computar uma saída  $Y_r(s)$  gerada por uma referência  $R(s)$  como:

$$Y_r(s) = G_{yr}(s)R(s) = \frac{B(s)N(s)}{A(s)D(s)\phi(s) + B(s)N(s)} R(s)$$

Tendo um valor de saída  $Y_r(s)$ , pode-se calcular então o valor do erro do sistema:

$$E(s) = R(s) - Y_r(s) = (1 - G_{yr}(s))R(s)$$

$$E(s) = \frac{A(s)D(s)\phi(s)}{F(s)} \frac{N_r(s)}{D_r(s)} \quad (2.61)$$

Semelhantemente ao caso das perturbações, todos os polos instáveis de  $D_r(s)$  são cancelados por  $\phi(s)$ . Logo, conclui-se que  $r(t) - y_r(t) \rightarrow 0$  quando  $t \rightarrow \infty$ .

Pelo princípio da linearidade,  $y(t) = y_\omega(t) + y_r(t)$  é verdadeiro. Por consequência,  $r(t) - y(t) \rightarrow 0$  quando  $t \rightarrow \infty$ , o que comprova o rastreamento assintótico e a rejeição das perturbações.

Uma análise das equações (2.60) e (2.61) revela que, ainda que estas sofram variações em seus termos  $D(s)$ ,  $N(s)$ ,  $A(s)$  e  $B(s)$ , o sistema é capaz de realizar o rastreamento de referências e rejeição de perturbações, se o mesmo permanecer BIBO estável (do inglês *Bound Input, Bound Output*) e  $\phi(s)$  ainda for capaz de cancelar as raízes instáveis de  $D_r(s)$  e  $D_\omega(s)$ . Logo, ao encontrar um modelo  $\frac{1}{\phi(s)}$  que descreva a referência e a perturbação da planta e inserí-lo na malha fechada, é possível satisfazer o princípio do modelo interno. Aplicando métodos de design de alocação de polos para realizar as manipulações necessárias e garantir que os polos instáveis serão cancelados, torna-se possível desenvolver um design robusto para os controladores.

Por fim, cabe destacar que, ainda que as demonstrações de [Chen \(1999\)](#) sejam apresentadas em tempo contínuo utilizando o domínio da frequência em  $s$ , o mesmo conceito é facilmente aplicado para o tempo discreto.

## 2.9 Estado da Arte

O uso de algoritmos do tipo GPC em plantas de controle industriais e comerciais continua crescendo desde os primórdios de sua introdução. Entretanto, para compreender por quais motivos sua utilização é tão amplamente difundida e aplicada, torna-se necessário regressar no tempo, não muito distante do advento do próprio conceito de controle moderno, onde [Holkar e Waghmare \(2010\)](#) apresentam o fato de que, na década de 1960, quaisquer algoritmos de controle que não seguissem a “forma clássica” de um controlador PID seriam considerados como algoritmos avançados de controle. Controladores PID eram tão amplamente aplicados que eles representavam mais de 80% de todos os controladores retroalimentados automáticos em equipamentos utilizados em processos industriais.

Apesar de serem controladores de baixo custo e fácil implementação, os controladores PID possuem como revés o fato de não serem adequados para casos onde as plantas a serem controladas estão sujeitas a longos períodos de inatividade, conforme pontuado

por Lewis, Evans e Sandoz (1991). Como consequência, diversos algoritmos foram desenvolvidos ao longo dos anos, buscando trazer soluções mais adequadas e precisas para os mais diferentes casos e necessidades existentes, como por exemplo os algoritmos de Variância Mínima (VM), Variância Mínima Generalizada (GMV), Alocação de Polos (do inglês, *Pole Placement* - PP), entre outros.

Um destes algoritmos é conhecido como Controle Preditivo baseado em Modelo (MPC). Suas origens remontam ao trabalho publicado por Propoi (1963), que trouxe luz ao método conhecido como “horizonte deslizante” ou ainda “horizonte retrocedente”, cálculo que é a base de todos os algoritmos do tipo MPC. Inclusive, Orukpe (2012) apresenta que os algoritmos MPC podem ser conhecidos por dois outros nomes: Controle por Horizonte Deslizante (MHC, do inglês *Moving Horizon Control*) ou ainda Controle por Horizonte Retrocedente (RHC, do inglês *Receding Horizon Control*).

Entretanto, ainda que seus conceitos tenham sido apresentados em 1963, os algoritmos MPC permaneceram em relativa obscuridade até o final da década de 1970, onde as produções de Richalet et al. (1977) e Cutler e Ramaker (1979) apresentaram aplicações de sucesso de controladores baseados neste algoritmo, despertando o interesse de vários componentes de diferentes indústrias, especialmente da indústria petroquímica (pois Cutler e Ramaker eram engenheiros da empresa Shell e em seu trabalho apresentaram a aplicação deste algoritmo em processos complexos da extração de petróleo com resultados excelentes). Deve-se destacar que ambos os trabalhos apresentam algoritmos que incluem modelos dinâmicos explícitos das plantas controladas, que são utilizadas para prever o efeito de futuras ações das variáveis manipuladas na saída da planta (o que justifica o nome de “Controle Preditivo baseado em Modelo”).

Entretanto, aplicações de sucesso isoladas não são capazes de, por si só, gerar mudança de paradigmas em uma indústria consolidada. O que fez com que estes algoritmos de controle ganhassem tração e sucesso em meio à indústria é o fato de que os sistemas apresentados demonstraram vantagens que outros algoritmos não são capazes de replicar com tamanha eficiência. Considerando o exposto por Camacho e Bordons (2007) e Orukpe (2012), bem como por outros autores, os principais motivos pelo qual algoritmos do tipo MPC se tornaram tão difundidos na indústria remonta a fatores como a capacidade de tratar problemas multivariáveis com facilidade, levar em conta as limitações dos atuadores quando do desenvolvimento do controlador, conseguir tratar de alterações estruturais de maneira autônoma, ter capacidade de tratar processos instáveis ou que não se encontram em sua fase mínima, além de ser de um método de fácil sintonização de parâmetros, entre outras características que tornam este algoritmo desejável.

Mesmo com vantagens em relação a vários tipos de algoritmos, nenhum dos algoritmos que foram desenvolvidos sob a definição de controle adaptativo ou de auto-sintonização (a saber, a capacidade de estimar os parâmetros de controle de forma direta)

tinham a capacidade de serem usados como um "algoritmo de uso geral" para os mais diferentes casos reais de processos de controle. Desbravando este objetivo, [Clarke, Mohtadi e Tuffs \(1987\)](#) publicaram a definição do algoritmo de Controle Preditivo Generalizado (GPC), que surgiu como um método capaz de ser aplicado em casos onde as plantas possuem fases não-mínimas, onde a planta é instável em malha aberta ou com polos incorretamente atenuados, plantas com tempo inativo variável ou desconhecido ou até mesmo em casos onde a ordem da planta é desconhecida.

A flexibilidade de uso deste algoritmo é originária de uma sequência de fatores. O primeiro deles reside no fato de utilizar modelos de planta chamados CARIMA (do inglês, *Controlled Auto-Regressive Integrated Moving Average*, que também pode ser denominado de ARIMAX - *Auto-Regressive Integrated Moving Average with Exogenous Input*). Como pontuando corretamente por [Ganesh et al. \(2011\)](#), um modelo CARIMA possui como vantagens a capacidade de rastrear pontos futuros variantes e constantes, ruídos do sistema possuem características de drift estocástico e produzem respostas sem *offsets*, além destes modelos serem capazes de minimizar funções custo multiestágios definidas com um horizonte de predição, além de incluir as perturbações na sua modelagem de forma a deduzir a estrutura correta do controlador.

Outro grande fator que contribui para tal flexibilidade é a consequência natural de existir um integrador no projeto do controlador, baseado nas suposições feitas sobre o modelo básico da planta a ser controlada. Tal integrador é responsável pela capacidade do controlador ser capaz de alcançar um comportamento livre de *offsets* em malhas fechadas nos casos de perturbações de cargas com formato de valores aleatórios em instantes também aleatórios nos casos considerados determinísticos ou quando há movimentos Brownianos em sistemas estocásticos. Outros modelos de desenvolvimento de controladores são dependentes das adições de integradores de acordo com a necessidade do projetista, tornando o processo de desenvolvimento muito mais dependente da própria planta a ser controlada.

Destaca-se também a versatilidade deste tipo de algoritmo de controle devido a capacidade de lidar não somente com *offsets* ou plantas cujo projetista desconhece grande parte de seus dados, por ser hábil em tratar casos onde se faz necessário a utilização de alimentação direta ou ainda nos casos de plantas multivariáveis, conforme novamente apontado por [Holkar e Wagmare \(2010\)](#).

A descrição do algoritmo GPC proposto por [Clarke, Mohtadi e Tuffs \(1987\)](#) foi apresentada somente em formato polinomial e, desde então, diversas publicações tentaram derivar um controlador que replicasse as mesmas características e propriedades do algoritmo utilizando a descrição através do espaço de estados. Nesta busca, [Ordys e Clarke \(1993\)](#) demonstraram uma solução bastante elegante para esta derivação, apresentando as relações entre os parâmetros do espaço de estados e as entradas e saídas do sistema

polinomial, além de apresentar um método para obtenção da resposta à uma entrada do tipo degrau, assim como demonstração da equivalência entre os métodos discutidos. Não somente isto, mas a utilização do algoritmo em espaço de estados trás uma série de vantagens quando comparado com sua descrição em formato polinomial, como por exemplo a capacidade de observar as propriedades do controlador nos estados transitórios, a fácil implementação de casos multidimensionais e, com algumas restrições, até mesmo a capacidade de sua implementação em sistemas não lineares.

Não se deve acreditar, porém, que a utilização de algoritmos de controle do tipo GPC ou MPC não apresenta uma série de desafios para sua implementação. Um destes desafios, como discutido por [Grimble \(1992\)](#), é o fato de que sistemas baseados em GPC que não se encontram em fase mínima são instáveis quando seus parâmetros de controle são pequenos, além do fato que os próprios parâmetros de estabilidade são difíceis de estabelecer de formas que não sejam empíricas. Ademais, [Bordons e Camacho \(1998\)](#) são bastante enfáticos em listar que outro desafio de implementação de algoritmos preditivos reside na própria obtenção dos parâmetros destes controladores em implementações industriais em tempo real quando se há a necessidade de realizar novos cálculos a cada período de amostragem devido as alterações que os parâmetros são sujeitos, pois muitos componentes utilizados possuíam capacidades computacionais limitadas para efetuar o processamento necessário, ou ainda pelo fato que o tempo de cálculo dos referidos parâmetros pode exceder o tempo de amostragem necessário para o correto funcionamento do processo. Por este motivo, seu trabalho desenvolveu um método para aproximação dos parâmetros do GPC nos casos onde o poder computacional é reduzido, mantendo a robustez de estabilidade ao ser consideravelmente insensível às incertezas. Esta adaptação, entretanto, era limitada a processos industriais que podem ser modelados como um sistema de primeira ordem com tempo inativo. O que [Kennel, Linder e Linke \(2001\)](#) destacaram, acertadamente, foi que a falta de poder de processamento matemático dos equipamentos não seria um problema perene, ao se levar em consideração a evolução tecnológica e industrial que ocorre de forma contínua. Não somente isto, mas que a obtenção de melhores resultados de controle também podem justificar um maior custo operacional, custo este que as indústrias, de maneira geral, podem e estão dispostas a arcar.

Quanto a seleção de parâmetros ótimos para o desenvolvimento do controlador GPC, [Martínez, Senent e Blasco \(1996\)](#) trouxeram um estudo comparativo entre os resultados em malha fechada gerados por controladores desenvolvidos através de processo de otimização clássica e os obtidos através da otimização via algoritmos genéticos, comprovando que a utilização deste algoritmo estocástico é capaz de produzir resultados extremamente similares e efetivos quando comparado com os resultados de um GPC tradicional. Tal fato demonstra que a implementação destes algoritmos pode ser recomendada para o desenvolvimento dos controladores, garantindo um desenvolvimento mais ágil e com maiores chances de obter os melhores parâmetros possíveis através do processamento ma-

temático efetuado por máquinas. Não somente isto, mas o uso do algoritmo genético permite uma série de possibilidades para a otimização do controlador, incluindo, mas não se limitando, a aplicação de novas funções custo que permitem o uso de mais restrições e até mesmo não linearidades.

Por causa de todas as vantagens anteriormente elencadas, não é surpreendente que o uso e estudo das técnicas de controle preditivo baseado em modelo tenha se expandido consideravelmente. Entretanto, ainda é de causar espanto o fato de que, mesmo com sua ampla acessibilidade e capacidade de aplicação, grande parte das publicações sobre o assunto focam sua atenção na regulação ou rastreamento de pontos constantes. Soma-se a este fato o grande vácuo acadêmico sobre a capacidade que estes modelos de algoritmos possuem no rastreamento de sinais de características periódicas, como sinais senoidais, por exemplo. Um dos primeiros materiais que se destaca sobre o assunto é o trabalho realizado por Wang e Rossiter (2008), demonstrando a aptidão que controladores do tipo GPC possuem para o rastreamento deste tipo de sinal, sendo possível alcançar erro zero em estado estacionário devido ao Princípio do Modelo Interno que é aplicado no desenvolvimento do controlador.

O trabalho de Lima, Montagner e Maccari (2017) apresenta o design de um controlador do tipo GPC aplicado nos inversores conectados ao *grid* elétrico com filtros do tipo LCL, de forma que este controlador foi capaz de demonstrar a capacidade de rastrear sinais de correntes elétricas senoidais enquanto rejeitava tensões de perturbação de terceira, quinta e sétima harmônicas introduzidas no sistema pelo próprio *grid*. Tal rejeição foi possível através da introdução de quatro dinâmicas ressonantes em um dos polinômios do modelo de controle, sendo que este modelo também apresenta respostas transitórias rápidas, excelentes para o objetivo proposto.

Já o trabalho de Cordero et al. (2022b), no qual este autor também tomou parte em sua produção, apresenta uma nova aplicação de um sistema GPC onde o modelo aumentado da planta imbui os pólos da Transformada  $\mathcal{Z}$  de um sinal senoidal com *bias*, sendo que o trabalho foi capaz de rastrear e rejeitar assintoticamente os sinais em questão de maneira experimental enquanto mantém todas as vantagens de um sistema GPC, também citando a possibilidade de implementação de técnicas de otimização para a obtenção da lei de controle e seleção do indivíduo ótimo.

## 3 Controlador Proposto

### 3.1 Desenvolvimento de GPC para Rastreamento de Sinal Senoidal com Nível CC

Neste capítulo, será apresentado todo o desenvolvimento matemático e lógico do sistema GPC proposto, baseado no desenvolvimento de um novo modelo aumentado para a predição da resposta do sistema. A saída do modelo aumentado proposto é o erro de rastreamento do sinal senoidal em vez da saída da planta.

#### 3.1.1 Preliminares

Inicia-se este desenvolvimento através da seleção de um sistema em que o GPC possa se basear. O sistema selecionado é o previamente descrito em (2.26) e (2.27), pois apresenta um modelo discreto de ordem  $n$  de uma planta SISO, onde  $u(i)$  é a entrada da planta,  $y(i)$  é a saída do sistema e  $x_m(i)$  é o vetor de estados da planta.

Tendo em mente que o objetivo deste sistema é a capacidade de rastreamento de sinais senoidais, o sinal senoidal de referência deste sistema pode ser descrito por (3.1), onde  $t$  é o tempo,  $\omega$  é a frequência em rad/s,  $\rho$  é a amplitude e  $b$  é o *bias* (viés) do sinal. A Figura 14 exemplifica o comportamento de um sinal de referência senoidal com nível CC diferente de zero.

$$r(t) = b + \rho \sin(\omega t) \quad (3.1)$$

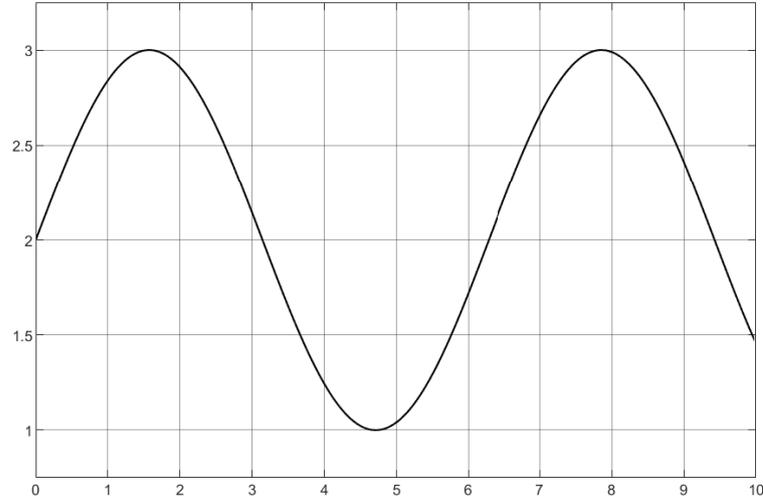
Assume-se que o sinal de referência  $r(t)$  descrito previamente e as variáveis de estado da planta são amostrados com um período de amostragem  $t_s$ . Considerando que a relação entre o tempo contínuo e discreto pode ser definida como  $t = t_s i$ , onde  $i$  representa um instante qualquer da medição, torna-se possível descrever o sinal de referência no tempo discreto  $r(i)$  conforme (3.2).

$$r(i) = b + \rho \sin(\Omega i), \quad \Omega = \omega t_s. \quad (3.2)$$

O conceito de controladores por modelos preditivos dispõe que as saídas e referências futuras de uma planta podem ser obtidas com base nas informações disponíveis nos momentos atual e prévios. Com base neste conceito, (3.3) apresenta uma generalização de como o sinal de referência no instante futuro  $r(i+1)$  pode ser descrito, onde  $\alpha$ ,  $\beta$ ,  $\delta$ , ... são valores reais quaisquer.

$$r(i+1) = \alpha r(i) + \beta r(i-1) + \delta r(i-2) + \dots \quad (3.3)$$

Figura 14 – Sinal de Entrada - Referência do tipo senoidal com um nível cc. Neste exemplo, temos um sinal com *bias*  $b = 2$ , amplitude  $\rho = 1$  e  $\omega = 1$ , fazendo com que o sinal de referência seja descrito por  $r(t) = 2 + 1 \sin(1t)$



Fonte: Autor

Sabe-se que o sinal de referência é um sinal senoidal. Portanto, entende-se por necessário trazer à memória algumas identidades trigonométricas importantes para a realização dos cálculos necessários para o desenvolvimento deste controlador, apresentadas a seguir.

$$\sin(\alpha) - \sin(\beta) = 2 \sin\left(\frac{\alpha - \beta}{2}\right) \cos\left(\frac{\alpha + \beta}{2}\right) \quad (3.4)$$

$$\sin(3\psi) = (2 \cos(2\psi) + 1) \sin(\psi) \quad (3.5)$$

Considerando as identidades apresentadas em (3.4) e (3.5), é possível calcular as diferenças entre valores do sinal de referência em diferentes instantes, cujos cálculos são demonstrados abaixo.

$$\begin{aligned} r(i) - r(i-1) &= b + \rho \sin(\Omega i) - [b + \rho \sin(\Omega(i-1))] \\ &= \rho[\sin(\Omega i) - \sin(\Omega(i-1))] \\ &= 2\rho \sin(0.5\Omega) \cos(\Omega i - 0.5\Omega) \end{aligned} \quad (3.6)$$

$$\begin{aligned} r(i+1) - r(i-2) &= b + \rho \sin(\Omega(i+1)) - [b + \rho \sin(\Omega(i-2))] \\ &= \rho[\sin(\Omega(i+1)) - \sin(\Omega(i-2))] \\ &= 2\rho \sin[3(0.5\Omega)] \cos(\Omega i - 0.5\Omega) \\ &= (2 \cos(\Omega) + 1)[2\rho \sin(0.5\Omega) \cos(\Omega i - 0.5\Omega)] \end{aligned} \quad (3.7)$$

Ao substituir (3.6) em (3.7), é possível expressar  $r(i+1)$  da seguinte maneira:

$$r(i+1) = \gamma r(i) - \gamma r(i-1) + r(i-2), \quad \gamma = 2 \cos(\Omega) + 1 \quad (3.8)$$

### 3.1.2 Modelo Aumentado

Conforme já explicitado anteriormente, as equações apresentadas em (2.26) e (2.27) descrevem um sistema SISO de  $n$ -ésima ordem, utilizado como a base deste desenvolvimento. Pode-se, então, obter a função de transferência da planta no tempo discreto,  $G(z)$ , se valendo das matrizes  $A_m$ ,  $B_m$  e  $C_m$  da equação do espaço de estados, definida em (3.9), onde  $I_n$  é uma matriz identidade de ordem  $n$ .

$$G(z) = C_m(zI_n - A_m)^{-1}B_m \quad (3.9)$$

Resta, por fim, determinar qual o valor do erro de rastreamento em tempo discreto, que é descrito por (3.10).

$$e(i) = r(i) - y(i) = r(i) - C_mx_m(i) \quad (3.10)$$

Se o erro no instante atual é dependente apenas do sinal de referência e das variáveis de estado no mesmo instante, pode-se inferir então, que o sinal de erro em um instante futuro é exclusivamente dependente dos sinais de referência e dos valores das variáveis de estado neste mesmo instante. Considerando um instante futuro  $i + 1$ , é possível substituir (3.8) em (3.10), conforme apresentado em (3.11).

$$\begin{aligned} e(i) = r(i) - C_mx_m(i) \quad \therefore \quad e(i + 1) = r(i + 1) - C_mx_m(i + 1) \\ e(i + 1) = \underbrace{\gamma r(i) - \gamma r(i - 1) + r(i - 2)}_{r(i+1)} - C_mx_m(i + 1) \end{aligned} \quad (3.11)$$

Ainda utilizando (3.10) como base, é possível inferir, também, que o sinal de referência em um instante anterior  $i - j$  pode ser descrito como  $r(i - j) = e(i - j) + C_mx(i - j)$ ,  $j = 0, 1, 2, \dots$ , o que permite reescrever (3.11) como (3.12).

$$\begin{aligned} e(i + 1) = \gamma \underbrace{[e(i) + C_mx_m(i)]}_{r(i)} - \gamma \underbrace{[e(i - 1) + C_mx_m(i - 1)]}_{r(i-1)} \\ + \underbrace{[e(i - 2) + C_mx_m(i - 2)]}_{r(i-2)} - C_mx_m(i + 1) \\ e(i + 1) = \gamma e(i) - \gamma e(i - 1) + e(i - 2) \\ + C_m[x_m(i - 2) - \gamma x_m(i - 1) + \gamma x_m(i) - x_m(i + 1)] \end{aligned} \quad (3.12)$$

Se duas variáveis  $q(i)$  e  $v(i)$  forem definidas como, respectivamente, funções das variáveis de estados e dos sinais de entrada no instante atual e em anteriores, conforme apresentado em (3.13) e (3.14), é possível utilizar (2.26) para encontrar valores futuros, presentes e passados das variáveis de estado, assim como o valor futuro de  $q(i)$ , conforme

(3.15) e (3.16).

$$q(i) = x_m(i-3) - \gamma x_m(i-2) + \gamma x_m(i-1) - x_m(i) \quad (3.13)$$

$$v(i) = u(i-3) - \gamma u(i-2) + \gamma u(i-1) - u(i) \quad (3.14)$$

$$x_m(i+1-k) = A_m x_m(i-k) + B_m u(i-k), \quad k = 0, 1, 2, \dots \quad (3.15)$$

$$q(i+1) = x_m(i-2) - \gamma x_m(i-1) + \gamma x_m(i) - x_m(i+1). \quad (3.16)$$

Aplicando as equações (3.15) e (3.16) em conjunto com as definições de  $q(i)$  e  $v(i)$ , pode-se expressar  $q(i+1)$  como:

$$\begin{aligned} q(i+1) &= [A_m x_m(i-3) + B_m u(i-3)] - \gamma [A_m x_m(i-2) + B_m u(i-2)] \\ &\quad + \gamma [A_m x_m(i-1) + B_m u(i-1)] - [A_m x_m(i) + B_m u(i)] \\ q(i+1) &= A_m [x_m(i-3) - \gamma x_m(i-2) + \gamma x_m(i-1) - x_m(i)] \\ &\quad + B_m [u(i-3) - \gamma u(i-2) + \gamma u(i-1) - u(i)] \\ q(i+1) &= A_m q(i) + B_m v(i) \end{aligned} \quad (3.17)$$

e, com esta definição, é possível atualizar a equação apresentada em (3.12) para obtenção do valor do erro futuro conforme demonstrado abaixo.

$$\begin{aligned} e(i+1) &= \gamma e(i) - \gamma e(i-1) + e(i-2) + C_m q(i+1) \\ e(i+1) &= \gamma e(i) - \gamma e(i-1) + e(i-2) + C_m A_m q(i) + C_m B_m v(i) \end{aligned} \quad (3.18)$$

A equação apresentada em (3.18) é então utilizada para deduzir a seguinte equação matricial, onde as matrizes  $\varepsilon(i)$  e  $\varepsilon(i+1)$  são matrizes do tipo coluna contendo os valores de erros nos instantes passados, presente e futuro;  $E$  é a matriz contendo os índices da equação (3.18); por fim,  $\sigma$  é o vetor que indica que somente a equação do erro no instante futuro  $e(i+1)$  possui o termo  $C_m A_m q(i) + C_m B_m v(i)$ .

$$\underbrace{\begin{bmatrix} e(i-1) \\ e(i) \\ e(i+1) \end{bmatrix}}_{\varepsilon(i+1)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -\gamma & \gamma \end{bmatrix}}_E \underbrace{\begin{bmatrix} e(i-2) \\ e(i-1) \\ e(i) \end{bmatrix}}_{\varepsilon(i)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\sigma} [C_m A_m q(i) + C_m B_m v(i)] \quad (3.19)$$

Ao levar em conta todos os dados apresentados até o momento, torna-se bastante simples compreender que tanto  $q(i)$  quanto  $\varepsilon(i)$  podem atuar como as variáveis de estado do modelo aumentado. Com esta informação e posicionando ambos vetores dentro do vetor de variáveis de estado do modelo aumentado, pode-se obter a relação em (3.20), que permite encontrar a definição do modelo aumentado proposto, apresentado em (3.21), onde  $v(i)$  é a entrada do modelo,  $e(i)$  é a saída,  $x(i)$  é o vetor de estados desse modelo e  $T$  é a indicação do processo de transposição de uma matriz.

$$[0_{1 \times n} \quad \sigma^T] \begin{bmatrix} q(i) \\ \varepsilon(i) \end{bmatrix} = [0_{1 \times n} \quad 0 \quad 0 \quad 1] \begin{bmatrix} q(i) \\ e(i-2) \\ e(i-1) \\ e(i) \end{bmatrix} = e(i) \quad (3.20)$$

$$\underbrace{\begin{bmatrix} q(i+1) \\ \varepsilon(i+1) \end{bmatrix}}_{x(i+1)} = \underbrace{\begin{bmatrix} A_m & 0_{n \times 3} \\ \sigma C_m A_m & E \end{bmatrix}}_A \underbrace{\begin{bmatrix} q(i) \\ \varepsilon(i) \end{bmatrix}}_{x(i)} + \underbrace{\begin{bmatrix} B_m \\ \sigma C_m B_m \end{bmatrix}}_B v(i) \quad (3.21)$$

$$e(i) = \underbrace{[0_{1 \times n} \quad \sigma^T]}_C x(i)$$

Assim como demonstrado em (2.26) e (2.27),  $A_m$ ,  $B_m$  e  $C_m$  são as matrizes do modelo da planta a ser controlada e estas mesmas matrizes são utilizadas como submatrizes no modelo aumentado apresentado em (3.21). As matrizes do modelo aumentado seguem a mesma nomenclatura apresentada em (2.37) e são utilizadas com o objetivo de simplificar a representação dos cálculos do modelo aumentado em questão. As ordens destas três matrizes são:

$$A \in \mathbb{R}^{(n+3) \times (n+3)} \quad B \in \mathbb{R}^{(n+3) \times 1} \quad C \in \mathbb{R}^{1 \times (n+3)}.$$

Por fim, deve-se destacar que, para o desenvolvimento deste modelo, assume-se que o valor da frequência  $\omega$  é conhecida, pois esta é integral para o cálculo do valor de  $\gamma$ . Entretanto, segundo Bobtsov e Pyrkin (2012) e Verma et al. (2021), estimadores podem ser utilizados para que o valor de  $\omega$  seja obtido sem maiores perdas.

### 3.1.3 Lei de Controle Preditivo através de GPC

A teoria de GPC permite a predição de saídas futuras no formato apresentado em (3.21), como por exemplo o rastreamento dos erros de controle. Considerando todo o conteúdo apresentado até o momento em conjunto com as equações (2.42) e (2.43), já foi demonstrado a possibilidade de associar todas as saídas previstas de um modelo em um vetor  $Y$ , assim como todos os seus sinais de controle em um vetor  $U$ . Considerando, então, o que foi apresentado em (3.21), onde as saídas planejadas do modelo aumentado são os erros de controle futuros e que as entradas de controle, neste caso, são equivalentes as entradas previstas do modelo, é possível reorganizar os termos dos vetores  $Y$  e  $U$  conforme (3.22) e (3.23), onde  $N_P$  e  $N_C$  permanece como o horizonte de predição e o horizonte de controle, respectivamente, e, assim como antes,  $N_C \leq N_P$ .

$$Y = [e(i+1) \quad e(i+2) \quad \dots \quad e(i+N_P)]^T \in \mathbb{R}^{N_P \times 1} \quad (3.22)$$

$$U = [v(i) \quad v(i+1) \quad \dots \quad v(i+N_C-1)]^T \in \mathbb{R}^{N_C \times 1} \quad (3.23)$$

Assim como demonstrado em (2.44), o vetor de saídas  $Y$  pode ser calculado em função do vetor de variáveis de estado e de variáveis do controle do sistema. Este conceito também se aplica ao modelo aumentado salvo as devidas alterações e que é apresentado em (3.24) e (3.25), considerando que as matrizes  $A$ ,  $B$  e  $C$  já foram definidas em (3.21).

$$Y = Fx(i) + \phi U \quad (3.24)$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_P-1} \\ CA^{N_P} \end{bmatrix} \quad \text{e} \quad \phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_P-1}B & CA^{N_P-2}B & \dots & \dots & CA^{N_P-N_C}B \end{bmatrix} \quad (3.25)$$

Como demonstrado previamente, a aplicação de uma função custo  $J$  em controladores do tipo GPC permite a obtenção de um valor ótimo para  $U$  através do processo de otimização. Tal processo foi previamente demonstrado em (2.47) e que, para a aplicação no modelo aumentado, pode ser descrito através de (3.26), onde  $r_w$  e  $I_{N_C}$  permanecem como, respectivamente, o termo de ajuste da função em malha fechada e uma matriz identidade de ordem  $N_C$ .

$$J = U^T \bar{R} U + Y^T Y, \quad \bar{R} = r_w I_{N_C} \quad (3.26)$$

Para a obtenção do valor ótimo da função de custo, aplica-se o teorema da primeira derivada, conforme demonstrado anteriormente. Desta forma, o valor ótimo para  $U$  é apresentado a seguir.

$$U = -(\phi^T \phi + \bar{R})^{-1} \phi^T F x(i) \quad (3.27)$$

Segundo o que é disposto na técnica do horizonte de recuo, apenas o primeiro elemento do vetor  $U$  é utilizado para estimar a lei de controle, enquanto os outros termos do vetor são descartados e novos cálculos são realizados para obtenção dos próximos valores de controle. Assim sendo, temos que o valor da lei de controle atual utilizada pode ser obtida através de (3.28). Destaca-se que todos os termos identificados como  $K_{bs}$  atuam como um ganho aplicado às variáveis de estado da planta aumentada, o que permite uma implementação simplificada deste sistema.

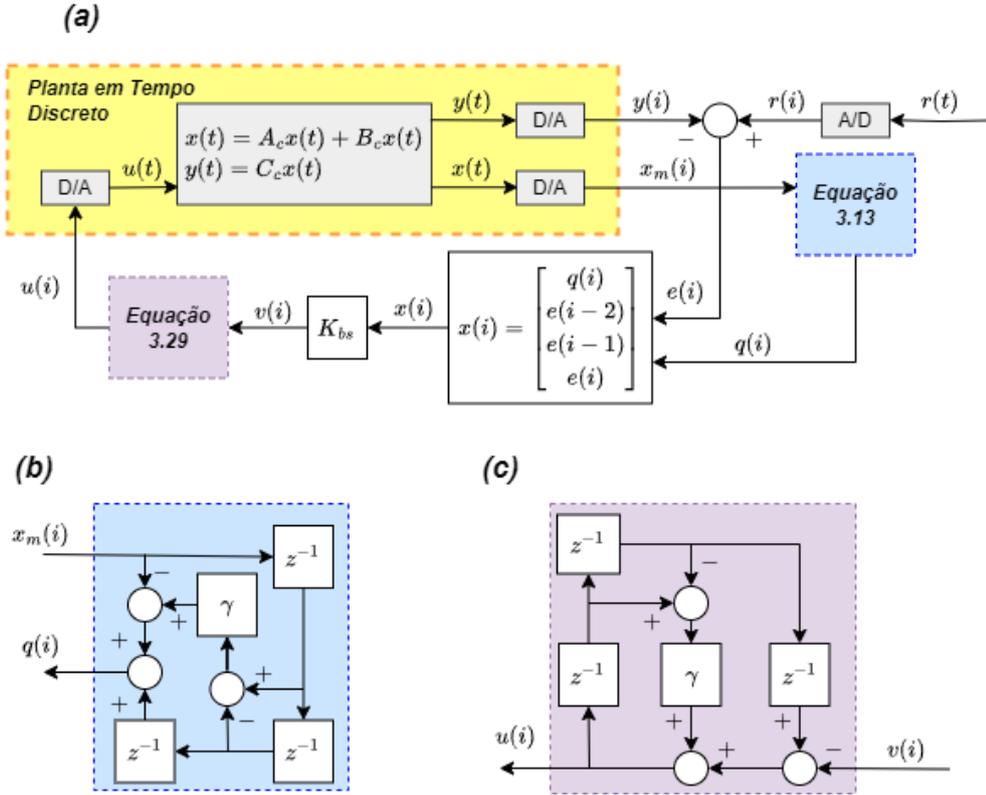
$$v(i) = \underbrace{-[1 \quad 0_{1 \times (N_C-1)}]}_{K_{bs}} (\phi^T \phi + \bar{R})^{-1} \phi^T F x(i) \quad (3.28)$$

A aplicação do novo cálculo da lei de controle atual simplifica a implementação da entrada do sistema, de forma que utilizar esta nova definição de  $v(i)$  permite que reescrevamos a equação de  $u(i)$  como (3.29).

$$u(i) = u(i-3) + \gamma[u(i-1) - u(i-2)] - v(i) \quad (3.29)$$

Todas as definições apresentadas permitem a apresentação de um controlador baseado em controle preditivo generalizado como o demonstrado na Figura 15.

Figura 15 – Estrutura do Controlador GPC Proposto. (a) apresenta a implementação do controlador, onde  $A_c$ ,  $B_c$  e  $C_c$  são as matrizes do modelo em espaço de estado da planta em tempo contínuo. (b) apresenta a implementação da função (3.13) e (c) apresenta a implementação da função (3.29). Em ambos os casos, o bloco  $z^{-1}$  indica o operador de deslocamento em atraso e o bloco  $\gamma$  é o ganho dado por  $\gamma = 2 \cos \Omega + 1$



Fonte: Autor

### 3.1.4 Capacidade de Rastreamento do Controlador

Usando como base a álgebra apresentada em Cordero et al. (2022a), é possível estimar a função de transferência  $G_a(z)$  do modelo aumentado descrito em (3.21), conforme demonstrado a seguir.

$$\begin{aligned}
 G_a(z) &= \underbrace{\begin{bmatrix} 0_{1 \times n} & \sigma^T \end{bmatrix}}_C \underbrace{\begin{bmatrix} zI_n - A_m & 0_{n \times 3} \\ -\sigma C_m A_m & zI_3 - E \end{bmatrix}^{-1}}_{zI_{n+3} - A} \underbrace{\begin{bmatrix} B_m \\ \sigma C_m B_m \end{bmatrix}}_B \\
 &= \begin{bmatrix} 0_{1 \times n} & \sigma^T \end{bmatrix} \begin{bmatrix} (zI_n - A_m)^{-1} & 0_{n \times 3} \\ (zI_3 - E)^{-1} \sigma C_m A_m (zI_n - A_m)^{-1} & (zI_3 - E)^{-1} \end{bmatrix} \\
 &= \sigma^T (zI_3 - E)^{-1} \sigma C_m [A_m (zI_n - A_m)^{-1} + I_n] B_m \\
 &= \sigma^T (zI_3 - E)^{-1} \sigma C_m [A_m (zI_n - A_m)^{-1} + (zI_n - A_m)(zI_n - A_m)^{-1}] B_m \\
 G_a(z) &= z \left( \sigma^T (zI_3 - E)^{-1} \sigma \right) \left( C_m (zI_n - A_m)^{-1} B_m \right) \quad (3.30)
 \end{aligned}$$

Nesta função de transferência, destaca-se a expressão  $zI_3 - E$ . Sabe-se que

$$zI_3 - E = \begin{bmatrix} z & -1 & 0 \\ 0 & z & -1 \\ -1 & \gamma & z - \gamma \end{bmatrix} \quad \text{e que} \quad (zI_3 - E)^{-1} = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix}$$

portanto, é possível demonstrar que

$$\sigma^T (zI_3 - E)^{-1} \sigma = [0 \quad 0 \quad 1] \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = l_{33}$$

Utilizando o conceito de Matriz Adjunta, é possível encontrar um valor para  $l_{33}$ , de forma que:

$$l_{33} = \frac{z^2}{\det(zI_3 - E)} \quad \therefore \quad l_{33} = \frac{z^2}{z^2(z - \gamma) + z\gamma - 1} \quad (3.31)$$

Substituindo os valores encontrados em (3.9) e (3.31) em (3.30), encontra-se que

$$\begin{aligned} G_a(z) &= z \frac{z^2}{z^2[z - (2 \cos(\Omega) + 1)] + z(2 \cos(\Omega) + 1) - 1} G(z) \\ G_a(z) &= \frac{z^3}{(z - 1)(z^2 - 2 \cos(\Omega)z + 1)} G(z) \end{aligned} \quad (3.32)$$

A Transformada  $\mathcal{Z}$  de (3.14) pode ser reorganizada de forma a descrever a função de transferência do modelo aumentado do controlador. Isso é possível pelo fato de que esta função possui termos que descrevem a entrada da planta  $u(i)$  assim como termos que descrevem da ação de controle de GPC  $v(i)$ . Por se tratar da função de transferência do modelo, isto significa que tal transformada deve se igualar à função encontrada em (3.32), conforme demonstrado a seguir.

$$\frac{\mathcal{Z}\{u(i)\}}{\mathcal{Z}\{v(i)\}} = \frac{z^3}{1 - \gamma z + \gamma z^2 - z^3} = \frac{-z^3}{(z - 1)(z^2 - 2 \cos(\Omega)z + 1)} G(z) \quad (3.33)$$

Se for considerada a hipótese de que o numerador da função  $G(z)$  não possui raízes em comum com o denominador  $(z - 1)(z^2 - 2 \cos(\Omega)z + 1)$ , isto significa que ambas as funções de transferência demonstradas na igualdade (3.33) possuem pólos idênticos. Ademais, o fato de que as transformadas- $\mathcal{Z}$  de sinais do tipo degrau unitário e  $\sin(\Omega i)$  podem ser descritas, respectivamente como:

$$\begin{aligned} \mathcal{Z}\{u[n]\} &= \frac{z}{z - 1} \\ \mathcal{Z}\{\sin(\Omega i)\} &= \frac{\sin(\Omega)z}{z^2 - 2 \cos(\Omega)z + 1} \end{aligned}$$

permite verificar que tanto o modelo de predição quanto a lei de controle apresentados em (3.33) possuem internamente os polos de modelos de sinais senoidais e do tipo degrau. Assim sendo, pelo princípio do modelo interno, o controlador é capaz de rastrear ou rejeitar ambos os tipos de sinais, permitindo o tratamento de sinais de entrada ou de perturbações do tipo senoidal com *bias* (viés).

## 3.2 Determinação dos Parâmetros Ótimos do GPC através de Algoritmo Genético

Assim como comentado previamente, o desenvolvimento de um GPC é dependente da definição de três parâmetros que definem sua capacidade de predição: o horizonte de predição  $N_P$ , o horizonte de controle  $N_C$  e o termo de ajuste da função  $r_w$ . Suas definições determinam o quanto as soluções podem ser matematicamente custosas mas também definem a quantidade de predições futuras utilizadas para a determinação dos novos sinais de controle, assim como o quanto tais sinais de controle tem a capacidade de afetar as saídas do sistema.

Um dos grandes desafios que surgem ao se utilizar um sistema de controle do tipo GPC reside na determinação destes três parâmetros com o objetivo de alcançar um sistema de controle robusto e ágil. Tal determinação normalmente é relegada para métodos de tentativa e erro, onde é necessário modelar o sistema e colocá-lo à prova para verificar suas respostas. Surge então como objetivo desenvolver uma forma de encontrar os valores ótimos para estes parâmetros, garantindo que o conjunto de valores mais adequados possa ser selecionado sem necessitar de extensos cálculos por parte do desenvolvedor. Nesta dissertação, um algoritmo genético (AG) será usado para definir os valores ótimos dos parâmetros do controlador preditivo proposto.

### 3.2.1 Escolha do Algoritmo de Otimização

O  $i$ -ésimo indivíduo  $\hat{x}_i$  possui, como seus cromossomos, os parâmetros do GPC, de acordo com (3.34). Os cromossomos destes indivíduos possuem restrições quanto aos valores que podem assumir, que são descritas de (3.35) à (3.38).

$$\hat{x}_i = [N_p \quad N_c \quad r_w]^T \quad (3.34)$$

$$0 \leq r_w \quad (3.35)$$

$$N_c \leq N_p \quad (3.36)$$

$$N_p \leq 100 \quad (3.37)$$

$$2 \leq N_c \quad (3.38)$$

Para obter os valores ótimos desejados, será utilizada a ferramenta *ga* do *software* MATLAB, ferramenta esta existente para implementação de algoritmos genéticos. Tal ferramenta pode receber variados argumentos, sendo um deles as definições de restrições dos valores dos cromossomos. Tais restrições podem adotar dois formatos distintos, através de equações de igualdades e equações de desigualdade, conforme apresentados em (3.39)

e (3.40).

$$\hat{A}_{eq}\hat{x} = \hat{b}_{eq} \quad (3.39)$$

$$\hat{A}_{des}\hat{x} \leq \hat{b}_{des} \quad (3.40)$$

A determinação do indivíduo ótimo nesta dissertação utiliza somente restrições do tipo desigualdade. Portanto, as restrições indicadas de (3.35) até (3.38) podem ser expressadas no formato de (3.40), onde  $\hat{A}_{des}$  e  $\hat{b}_{des}$  são descritos em (3.41).

$$\hat{A}_{des} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \quad \hat{b}_{des} = \begin{bmatrix} 0 \\ 0 \\ 100 \\ -2 \end{bmatrix} \quad (3.41)$$

Por fim, o algoritmo genético, por se tratar de um processo de otimização, necessita de uma função custo  $\hat{J}$  para determinar quão desejáveis são os resultados obtidos com cada indivíduo de uma população. Como o nome já preconiza, quanto menor o valor obtido no cálculo da função custo, melhor é considerado o resultado. Portanto, é importante descrever alguns fatores que foram levados em consideração durante a definição da função custo utilizada no algoritmo genético.

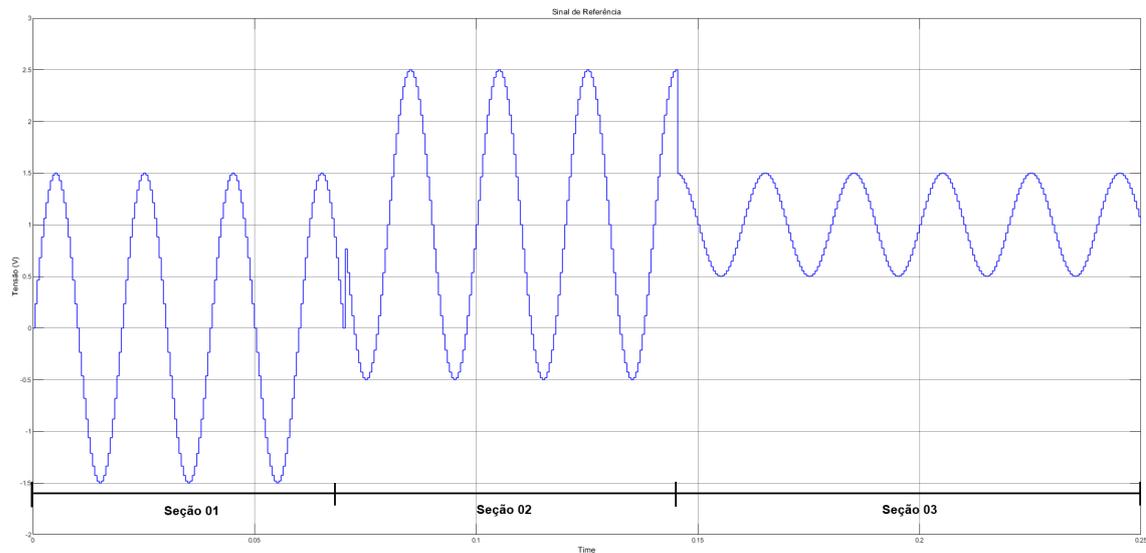
O primeiro fator reside na necessidade deste controlador ser robusto. Deve-se ter em mente que sistemas reais serão sempre suscetíveis à diferentes tipos de interferências em suas implementações, incluindo até mesmo a presença de ruídos nas próprias variáveis de estado produzidas pela planta. Assim sendo, entende-se salutar simular sistemas em malha fechada que sejam sujeitos a ruídos. Neste trabalho, foi definida a utilização de um ruído branco gaussiano aplicado diretamente às variáveis de estado, com média  $\mu = 0$  e desvio padrão  $\sigma = 0.05$ .

O segundo fator de importância reside na própria natureza variável dos sinais de referência e de possíveis interferências aplicadas em um sistema. Não é incomum que as amplitudes e valores de *bias* de um sinal recebido por uma planta sejam alterados durante seu funcionamento. Por este motivo, foi selecionado um sinal de referência composto de três seções, cada uma regida por um sinal senoidal com amplitude e bias distintos, conforme os valores dispostos na Tabela 1 e cujo comportamento é apresentado na Figura 16.

Tabela 1 – Parâmetros dos sinais utilizados nos testes com o controlador proposto

Seção	<i>bias</i> (b)	Amplitude ( $\rho$ )	Frequência ( $\omega$ )
1	0V	1.5V	$100\pi \frac{rad}{s}$ (50Hz)
2	1V	1.5V	$100\pi \frac{rad}{s}$ (50Hz)
3	1V	0.5V	$100\pi \frac{rad}{s}$ (50Hz)

Figura 16 – Sinal de Referência em tempo discreto utilizado nas simulações de controle em malha fechada no MATLAB



Fonte: Autor

Tendo em vista o tipo de sinal de referência proposto assim como os objetivos almejados de robustez, agilidade no rastreamento de sinais senoidais e possibilidade de implementação em sistemas reais, é possível definir quais são as variáveis obtidas através do sistema que serão utilizadas para o cálculo do custo de cada teste.

A primeira variável a ser considerada neste caso é o pico da lei de controle aplicada na planta, mais especificamente o pico da lei de controle quando as variáveis de estado são sujeitas a um nível de ruído branco,  $u_{Pr}$ . Tal escolha é realizada por dois principais motivos: o primeiro é pela observação do comportamento do controlador quando este encontra sinais distintos dos considerados ideais, afetados diretamente por ruídos que podem comprometer o correto funcionamento do sistema. O segundo motivo, não menos importante, reside no fato de que um sistema de controle pode apresentar extrema agilidade em rastrear sinais, porém o preço pago por tal agilidade normalmente se apresenta em picos de sinal extremamente elevados, não raramente apresentando ordens de grandeza muito maiores que os sinais de referência.

Ainda que seja possível obter, sem maiores complicações, sinais com valores de pico extremamente elevados, quando da aplicação de tais controladores em casos reais e concretos, a possibilidade de que o sistema de controle cause danos irreparáveis às plantas as quais está associado devido aos picos da lei de controle é elevada. Portanto, ao utilizar esta variável no cálculo do custo, busca-se encontrar um controlador que seja capaz de produzir uma lei de controle que não coloque em risco a integridade da planta, mesmo quando sujeita à ruídos.

Já a segunda variável selecionada para o cálculo da função custo é o erro quadrático médio do sistema com variáveis de estado com ruído,  $e_{qmr}$ . O erro quadrático médio é

uma medida estatística amplamente utilizada para a verificação da precisão de modelos, verificando o quão precisa é a previsão de um modelo frente aos dados desejados. É obtido através do cálculo da média do quadrado das diferenças entre os diversos pontos do sinal desejado subtraído do sinal obtido. Possui, como uma de suas características principais, a penalização do resultados frente a erros mais elevados, já que eleva ao quadrado esta diferença. Este comportamento é bastante desejado nesta análise pois busca-se obter controladores que, mesmo afetados por ruídos, sejam capazes de prever com elevada precisão quais os sinais futuros desejados. Quanto menor for o erro encontrado entre o sinal predito e o sinal de referência, mais este controlador será desejado.

Por último, leva-se também em grande consideração o tempo necessário para que o controlador seja capaz de levar o sistema a um ponto onde o erro seja o mais próximo senão igual a zero, afinal um controlador que é capaz de prever sinais com elevada precisão porém necessita de longos períodos para atingir tal patamar não é mais desejável do que um sistema que pode apresentar um certo grau de erro, porém atinge uma faixa aceitável de divergência em poucos instantes. Para tanto, é utilizado o valor do chamado tempo de acomodação. Normalmente, este tempo é calculado através da resposta do sistema a uma entrada do tipo degrau.

Entretanto, pela natureza oscilatória do sinal senoidal a ser aplicado neste controlador, entende-se mais interessante que o cálculo do tempo de assentamento não seja realizado frente a resposta a uma entrada do tipo degrau, mas sim do tempo necessário para que o sinal do erro atinja uma amplitude igual ou inferior à 2% da amplitude do sinal, tempo esse que será denominado como tempo de assentamento do sinal de erro  $t_{ae}$ . Como o sinal de referência utilizado é composto por três seções, cada uma descrita por um sinal senoidal distinto, o valor de  $t_{ae}$  é calculado através da média dos tempos de acomodação do sinal do erro para cada sinal senoidal.

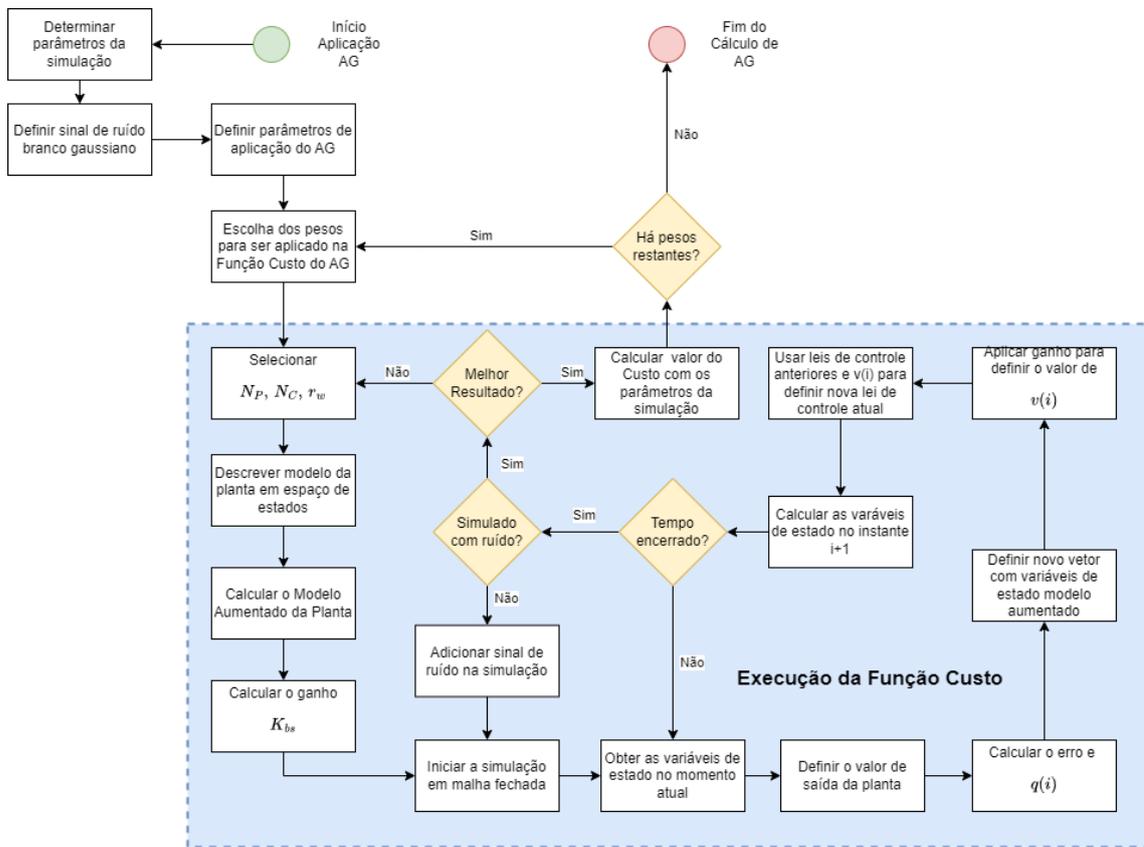
Com as variáveis escolhidas apresentadas, tem-se que a função desenvolvida nesta dissertação para a estimativa dos parâmetros ótimos do controlador é dada pela equação linear (3.42). Destaca-se que a variável  $u_{Pr}$  é dividida por 1000 para garantir que se encontre na mesma ordem de grandeza que as outras variáveis, impedindo que pequenas variações no valor de  $u_{Pr}$  gerem uma grande alteração no custo da função. Além disto, cada uma das variáveis é multiplicada por um dos parâmetros  $w_1$ ,  $w_2$  ou  $w_3$ , que agem como pesos aplicados a cada um dos termos com o objetivo de dosar sua participação no cálculo do custo. Isto só é possível pelo fato de que  $w_1 + w_2 + w_3 = 1$  e que todos estes três pesos se encontram no intervalo  $[0, 1]$ .

$$\hat{J} = w_1 \frac{u_{Pr}}{1000} + w_2 e_{qmr} + w_3 t_{ae} \quad (3.42)$$

O desenvolvimento de (3.42) para determinação dos resultados do AG foi inspirado em [Monteiro et al. \(2020\)](#), onde algoritmos genéticos foram utilizados para a determinação

de um indivíduo ótimo cujos cromossomos são utilizados para determinar os parâmetros de um controlador e cujos termos da função custo são multiplicados por pesos. Os valores dos pesos aplicados na função custo, serão apresentados no [Capítulo 4](#), junto à demonstração dos resultados obtidos com a utilização do AG. Já o fluxo da aplicação do algoritmo genético é demonstrado na [Figura 17](#), onde o quadro em destaque representa os passos executados na simulação da planta em malha fechada com e sem ruído, e a utilização dos parâmetros destes sinais para a escolha do melhor indivíduo em cada teste realizado com pesos distintos..

Figura 17 – Fluxo de aplicação do algoritmo genético no cálculo do indivíduo ótimo.



Fonte: Autor

## 4 Resultados

Após a apresentação da teoria utilizada para a obtenção do controlador desejado, é necessário demonstrar que o controlador proposto possui a performance e a capacidade de atuar em uma planta no tempo discreto. Ao ser comprovado de forma empírica a capacidade de rastreamento e rejeição de sinais senoidais, foi utilizado um algoritmo genético para a determinação de parâmetros ótimos de desenvolvimento do controlador. Estes testes são, então, apresentados em três casos distintos para melhor apreciação dos resultados, conforme a lista abaixo.

- Teste 1: Rastreamento de sinal de referência senoidal com *bias* e ausência de perturbação; e
- Teste 2: Escolha dos Melhores Parâmetros via Algoritmo Genético.

### 4.1 Definição da Planta de Testes

Em todos os testes elencados, foi utilizada uma planta cuja função de transferência é descrita em (4.1). Esta planta, em seu processo de discretização, utilizou  $t_s = 0.5$  ms como seu tempo de amostragem, de forma que, quando esta planta é apresentada na forma de espaço de estados, suas matrizes são descritas como (4.2), as quais foram obtidas usando o método de discretização em (2.21):

$$G(s) = \frac{1.418 \times 10^6 s + 3.637 \times 10^8}{s^3 + 2179s^2 + 2.273 \times 10^6 s + 7.274 \times 10^8} \quad (4.1)$$

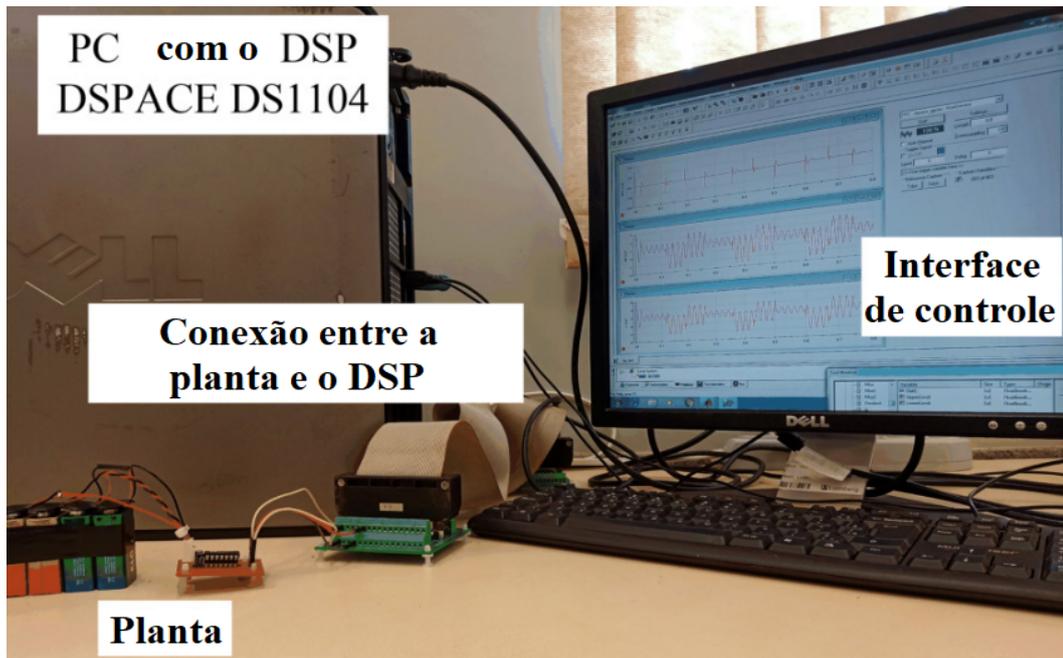
$$A = \begin{bmatrix} 0.413 & 0.454 & 0 \\ -0.240 & 0.788 & 0 \\ -0.437 & 0.422 & 0.774 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1331 \\ 0.4528 \\ 0.1273 \end{bmatrix} \quad \text{e} \quad C = [0 \quad 0 \quad 1] \quad (4.2)$$

A implementação deste sistema foi realizada com o auxílio do *software* MATLAB e a integração da planta com seu controlador foi realizada através da placa de controle DSP dSPACE DS1104, conforme demonstrado na [Figura 18](#).

### 4.2 Teste 1: Rastreamento de Referências Senoidais com *bias*

Para a execução deste teste, foi criado e inserido na planta um sinal composto de três seções distintas, cada uma destas sendo descrita por um sinal senoidal distinto sem a presença de sinais de interferência. Este sinal é apresentado na [Figura 19](#) e sua aplicação

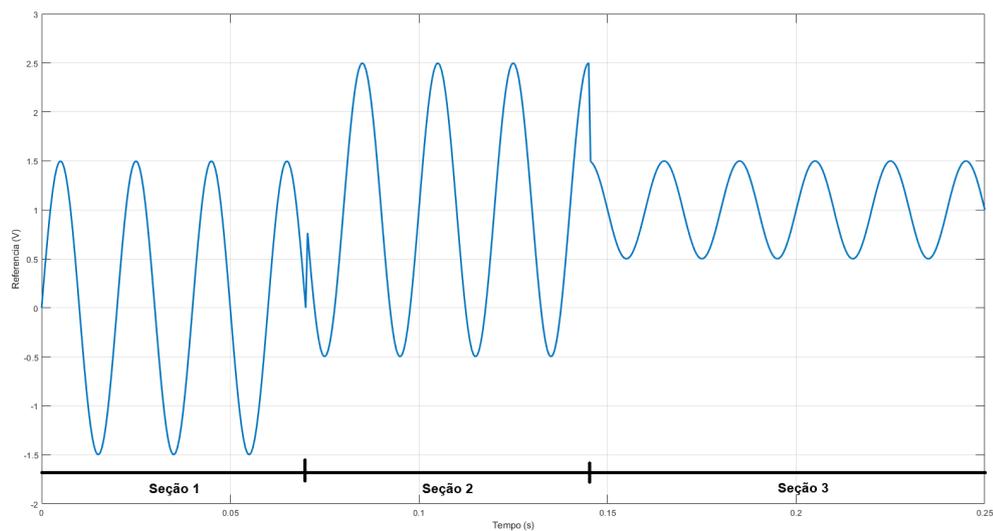
Figura 18 – Configuração dos experimentos realizados.



Fonte: Autor

tem o objetivo de demonstrar a capacidade de rastreamento do sistema de controle ao sinal desejado. Destaca-se que somente o valor da frequência da senoide foi mantida constante nas três seções. A [Tabela 1](#) apresentada no capítulo anterior demonstrou as características de cada um dos setores do sinal de referência.

Figura 19 – Sinal de Referência a ser rastreado no Teste 1.

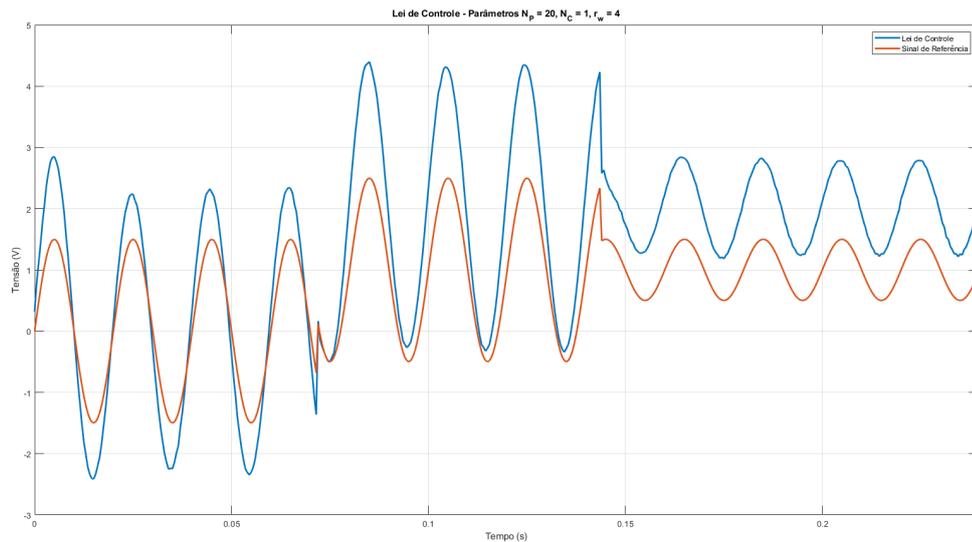


Fonte: Autor

Para a realização dos testes e comparação dos resultados, foram escolhidos através de testes empíricos os valores do horizonte de predição  $N_P$ , horizonte de controle  $N_C$  e o ajuste da função custo  $r_w$ . Conforme discutido previamente neste trabalho, no

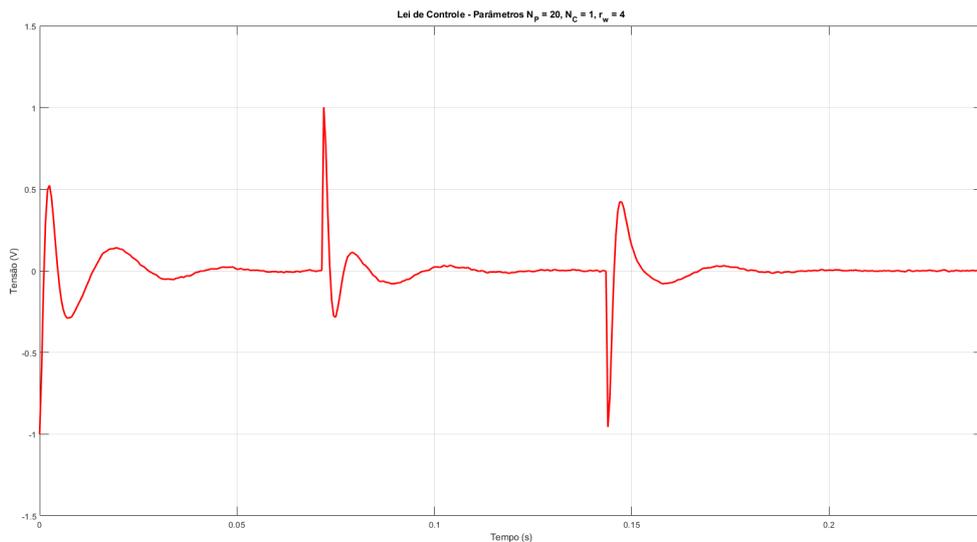
desenvolvimento de controladores do tipo GPC estes valores normalmente são escolhidos pelo próprio projetista. Portanto, testes com diferentes combinações destas três variáveis foram realizados, demonstrando como estes parâmetros podem afetar o desempenho de uma planta sem ruído. Desta forma, as Figuras 20 a 25 apresentam os resultados obtidos mediante as variações dos parâmetros supracitados.

Figura 20 – Resultados experimentais Teste 1: Lei de Controle -  $N_P = 20$ ,  $N_C = 1$  e  $r_w = 4$



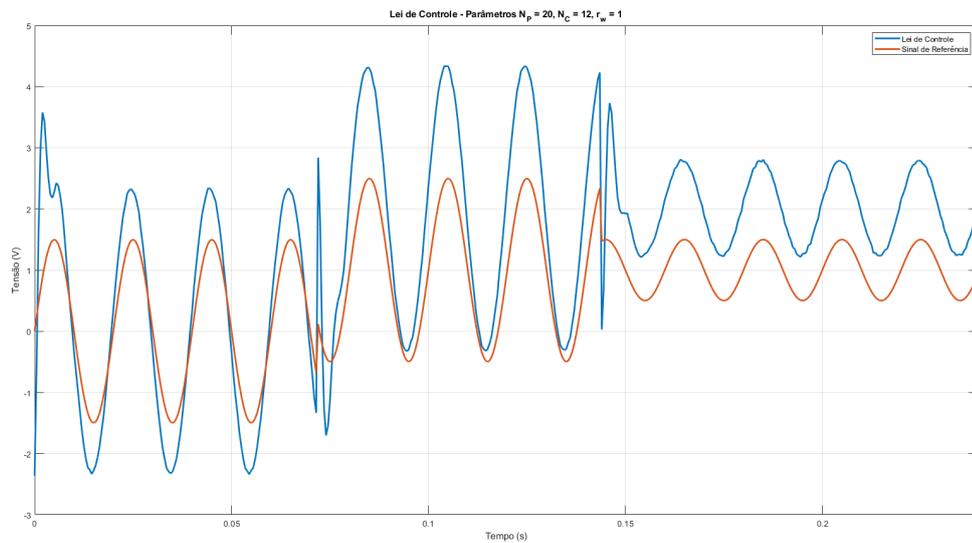
Fonte: Autor

Figura 21 – Resultados experimentais Teste 1: Sinal de Erro - Parâmetros  $N_P = 20$ ,  $N_C = 1$  e  $r_w = 4$

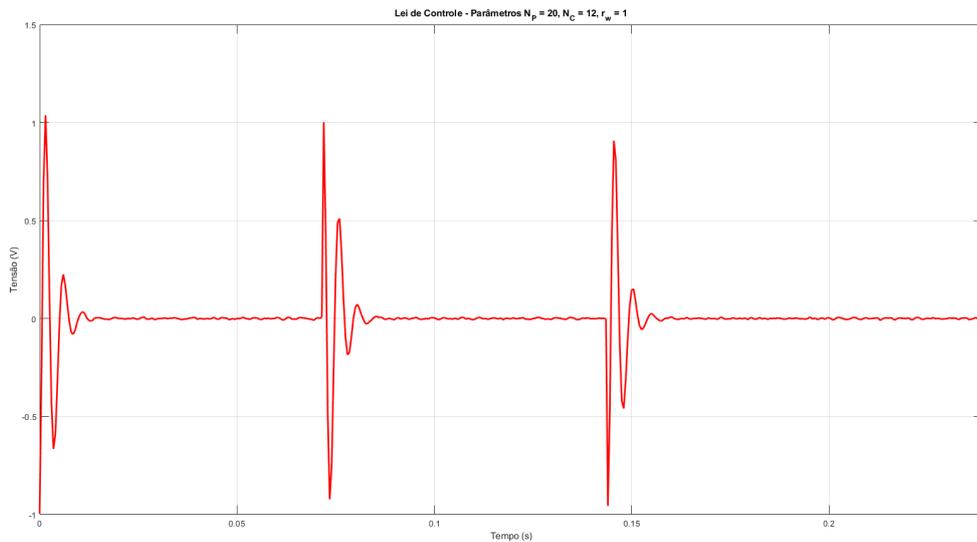


Fonte: Autor

Conforme discutido previamente, os controladores do tipo GPC são completamente dependentes dos parâmetros selecionados para seu desenvolvimento. Também é impor-

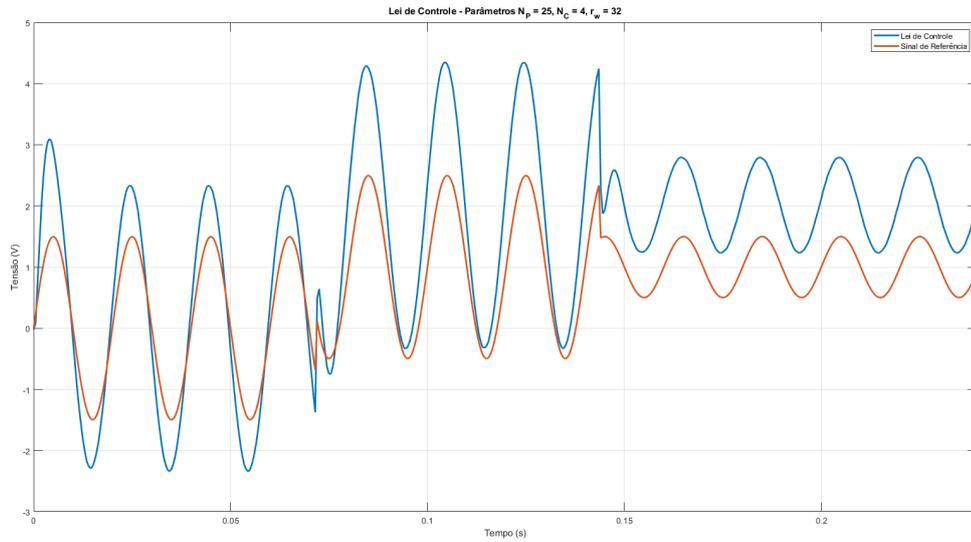
Figura 22 – Resultados experimentais Teste 1: Lei de Controle -  $N_P = 20$ ,  $N_C = 12$  e  $r_w = 1$ 

Fonte: Autor

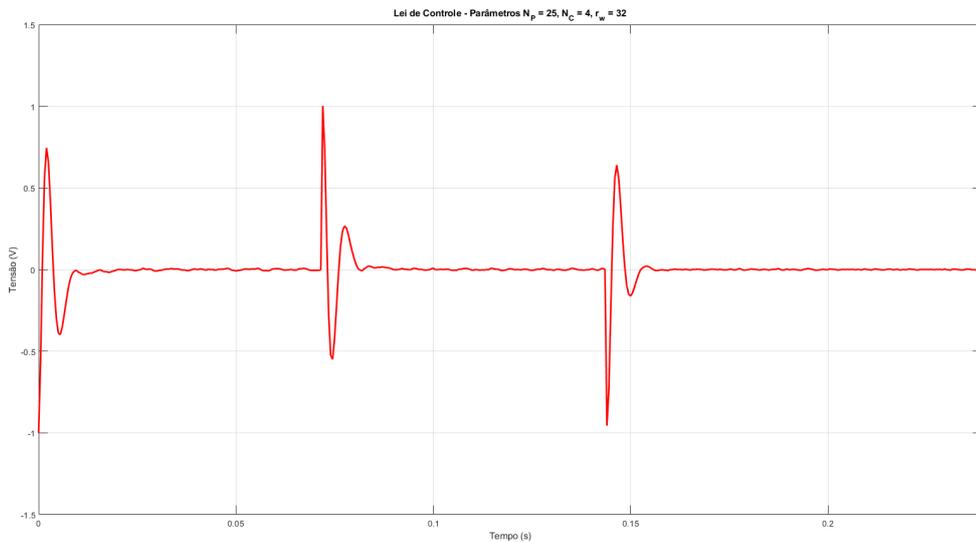
Figura 23 – Resultados experimentais Teste 1: Sinal de Erro - Parâmetros  $N_P = 20$ ,  $N_C = 12$  e  $r_w = 1$ 

Fonte: Autor

tante destacar que não existe necessariamente um relacionamento linear entre a variação dos parâmetros e o comportamento apresentado pelas respostas obtidas. Mesmo assim, compreende-se importante demonstrar quais são os possíveis efeitos de uma variação dos parâmetros, considerando que os outros dois parâmetros se mantêm inalterados. Assim sendo, as Figuras 26 e 27 fazem uma representação gráfica das variações dos termos  $N_C$  e  $r_w$ , respectivamente. Por fim, as tabelas 2 e 3 apresentam, respectivamente, os valores do erro quadrático médio  $e_{qm}$  (também conhecido como *Mean Square Error - MSE*) e do

Figura 24 – Resultados experimentais Teste 1: Lei de Controle -  $N_P = 25$ ,  $N_C = 4$  e  $r_w = 32$ 

Fonte: Autor

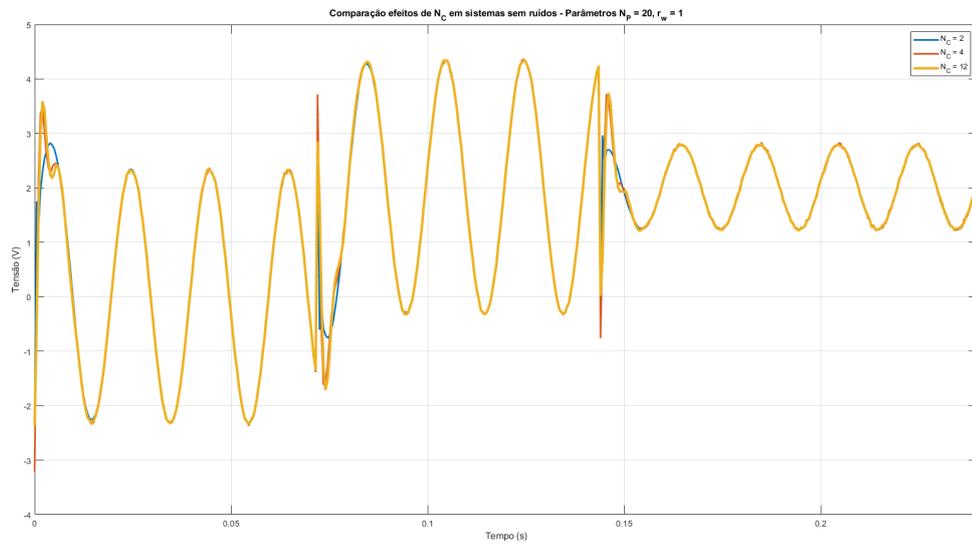
Figura 25 – Resultados experimentais Teste 1: Sinal de Erro - Parâmetros  $N_P = 25$ ,  $N_C = 4$  e  $r_w = 32$ 

Fonte: Autor

tempo de acomodação do erro da planta de controle  $t_{ae}$  de todos os sinais apresentados, além de incluir alguns resultados apresentados no trabalho de [Cordero et al. \(2022b\)](#), que utilizou a mesma planta para produção de seus resultados.

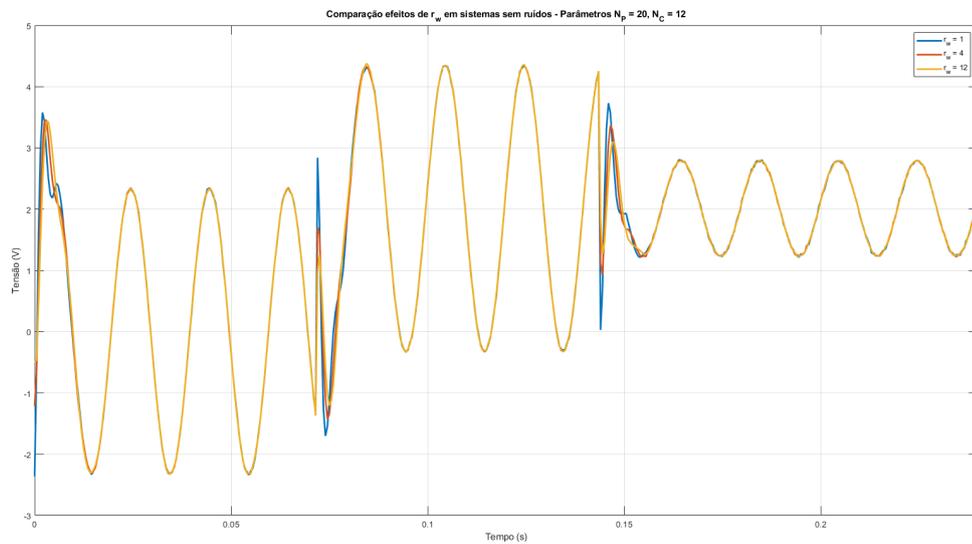
Observando os gráficos obtidos e as informações contidas nas tabelas, percebe-se que o  $e_{qm}$  em cada um dos setores, mesmo para diferentes valores de parâmetros, permanece na ordem de grandeza  $10^{-3}$  (ou seja, milivolts - mV), indicando um excelente nível de precisão. Não somente isto, mas o valor do tempo de acomodação demonstra que

Figura 26 – Resultados experimentais Teste 1: Efeito da alteração de  $N_C$  na Lei de controle - Parâmetros  $N_P = 20$  e  $r_w = 1$



Fonte: Autor

Figura 27 – Resultados experimentais Teste 1: Efeito da alteração de  $r_w$  na Lei de controle - Parâmetros  $N_P = 20$  e  $N_C = 12$



Fonte: Autor

não só o controle é realizado de forma precisa, mas também com agilidade, já que sua ordem de grandeza é de  $10^{-3}$  segundos.

Considerando o que foi previamente discutido na [subseção 2.7.3](#), um controlador baseado em algoritmo GPC é desenvolvido com base no Princípio do Modelo Interno, o que tem por consequência a capacidade do rastreamento e rejeição de sinais utilizados

Tabela 2 – Cálculo do EQM (em mV) com diferentes parâmetros - Teste 1. Os dados obtidos com parâmetros  $r_w < 1$  foram retirados de [Cordero et al. \(2022b\)](#)

$N_P$	$N_C$	$r_w$	$e_{qm}$ Geral	$e_{qm}$ Seção 1	$e_{qm}$ Seção 2	$e_{qm}$ Seção 3
20	1	0.3	130.8	152.1	123.3	118.6
20	1	0.07	130.9	152.3	122.9	118.7
20	1	4	17.1	23.2	15.1	13.9
20	2	1	11.2	14.4	11.6	8.5
20	4	1	21.6	27.8	22.8	16.0
20	12	1	24.0	30.6	25.5	18.0
20	12	4	25.7	32.9	26.9	19.4
20	12	12	26.9	34.6	27.8	20.7
25	4	32	19.0	24.4	19.1	14.9

Tabela 3 – Valor do Tempo de Acomodação do Sinal (em ms) com diferentes parâmetros - Teste 1. Os dados obtidos com parâmetros  $r_w < 1$  foram retirados de [Cordero et al. \(2022b\)](#)

$N_P$	$N_C$	$r_w$	$t_{ae}$ Seção 1	$t_{ae}$ Seção 2	$t_{ae}$ Seção 3
20	1	0.3	49.2	106.4	177.2
20	1	0.07	49.2	107.8	176.4
20	1	4	49.5	37.5	34.0
20	2	1	18.0	13.0	7.5
20	4	1	9.0	9.5	9.5
20	12	1	12.0	12.0	12.0
20	12	4	13.5	16.0	13.5
20	12	12	15.0	15.0	13.0
25	4	32	14.0	12.5	11.0

para seu cálculo. Logo, a consequência natural para controladores deste tipo é que seu sinal de erro se aproxime de zero. O conjunto dos parâmetros destes controladores, são capazes de determinar a velocidade da correção do sinal de saída do controlador ou ainda inserir instabilidade no sistema, sendo necessária atenção quanto a sua escolha.

Tendo tais fatos em mente, ao se observar os dados apresentados nas tabelas 2 e 3, percebe-se que valores muito baixos de  $r_w$  tem como consequência tempo maior para o controlador exercer sua capacidade de rastrear o sinal de referência, fato que ocasiona um maior valor de erro quadrático médio entre os sinais, demonstrando que, ainda que seja possível rastrear o sinal, tal rastreamento não é executado de forma ágil. Considerando que o termo  $r_w$  determina a intensidade dos efeitos da lei de controle sobre o cálculo do erro, de forma que uma baixa interação desta lei ocasiona maior lentidão na estabilização do sistema.

Este controlador também é capaz de rejeitar perturbações senoidais com nível de CC, conforme apresentado no trabalho de [Cordero et al. \(2022b\)](#) e que este autor tomou parte.



pesos aplicados em cada turno de testes, assim como os valores dos termos utilizados para o cálculo da função custo  $\hat{J}$ , o valor da referida função, assim como  $N_P$ ,  $N_C$  e  $r_w$ , integrantes do indivíduo ótimo de cada otimização.

A tabela apresenta uma linha em destaque, que demonstra o menor valor da função custo obtido após os testes realizados através do algoritmo genético. Este valor da função custo foi obtido através do uso dos parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$ , e dos pesos  $w_1 = 0.8$ ,  $w_2 = 0.1$  e  $w_3 = 0.1$ .

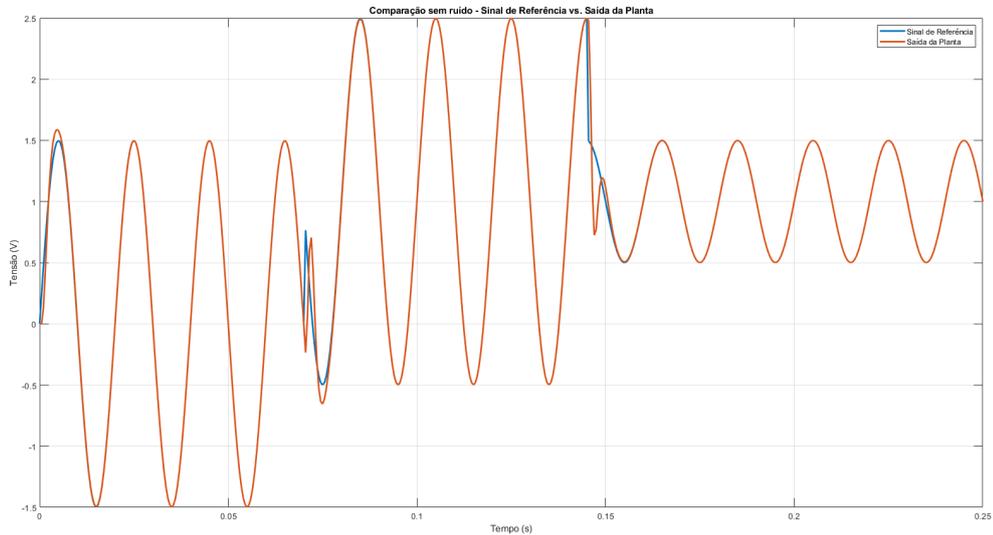
Tabela 4 – Termos e pesos aplicados na função custo, o valor obtido e os cromossomos do indivíduo ótimo em cada teste realizado. Em destaque se encontra o caso de teste em que foi obtido o menor valor de custo

Pesos			$\frac{u_{Pr}}{1000}(mV)$	$e_{qmr}(mV)$	$t_{ae}(ms)$	$\hat{J}(10^{-3})$	$N_P$	$N_C$	$r_w$
$w_1$	$w_2$	$w_3$							
0.8	0.1	0.1	4.72	10.88	79.83	12.85	16	2	6.76
0.1	0.8	0.1	5.17	9.35	79.67	15.97	16	2	0
0.1	0.1	0.8	5.83	13.17	77.67	64.03	24	2	8.33
0.7	0.1	0.2	4.81	10.38	79.67	20.34	16	2	4.60
0.7	0.2	0.1	4.98	9.64	79.67	13.38	16	2	1.32
0.1	0.7	0.2	5.17	9.35	79.67	23	16	2	0
0.2	0.7	0.1	5.17	9.35	79.67	15.55	16	2	0
0.1	0.2	0.7	5.83	13.17	77.67	57.58	24	2	8.33
0.2	0.1	0.7	5.83	13.17	77.67	56.85	24	2	8.33
0.6	0.2	0.2	5.04	9.46	79.67	20.85	16	2	0.49
0.2	0.6	0.2	5.17	9.35	79.67	22.58	16	2	0
0.2	0.2	0.6	5.36	10.88	78.5	50.34	21	2	4.44
0.5	0.2	0.3	5.04	9.46	79.67	28.31	16	2	0.49
0.5	0.3	0.2	5.04	9.46	79.67	21.29	16	2	0.49
0.2	0.5	0.3	5.17	9.35	79.67	29.61	16	2	0
0.3	0.5	0.2	5.17	9.35	79.67	22.16	16	2	0
0.2	0.3	0.5	5.36	10.88	78.5	43.59	21	2	4.44
0.3	0.2	0.5	5.36	10.88	78.5	43.04	21	2	4.44
0.4	0.2	0.4	5.36	10.88	78.5	35.72	21	2	4.44
0.4	0.3	0.3	5.04	9.46	79.67	28.75	16	2	0.49
0.4	0.4	0.2	5.04	9.46	79.67	21.73	16	2	0.49
0.3	0.4	0.3	5.12	9.39	79.67	29.19	16	2	0.19
0.2	0.4	0.4	5.04	9.46	79.67	48.83	16	2	0.49
0.3	0.3	0.4	5.04	9.46	79.67	36.22	16	2	0.49

Com o melhor resultado obtido pelo algoritmo genético em mãos, parte-se para produção de resultados através de simulações no *software* MATLAB usando um controlador que possui os parâmetros em questão. Inicialmente, foi realizada uma simulação da planta previamente apresentada em malha fechada, considerando a hipótese onde o sistema como um todo não possui incidência de ruídos. Tal situação produziu a [Figura 29](#), onde é possível verificar a agilidade do sistema em aproximar rapidamente a saída da planta com o sinal de referência utilizado. Tal agilidade é traduzida em sinais de erros

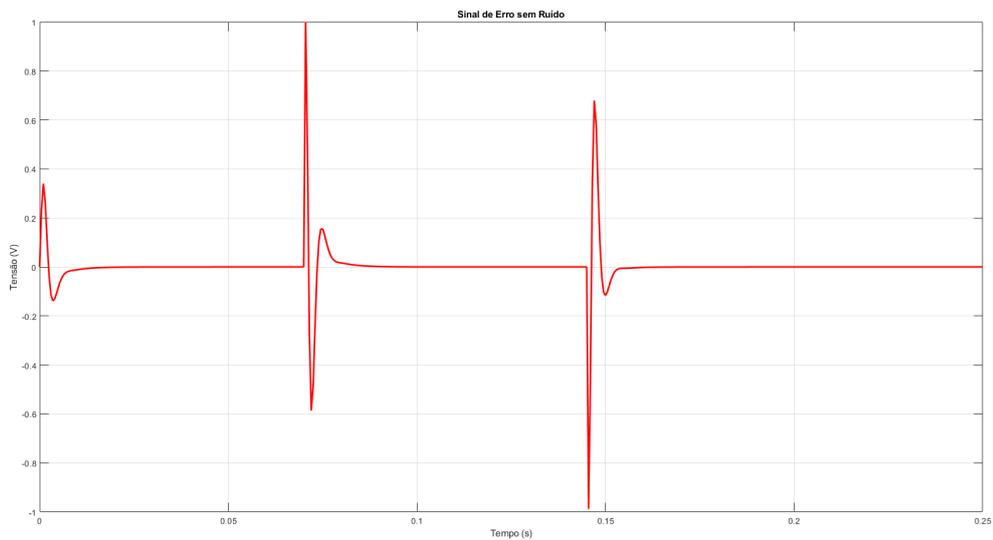
que também, em mesma velocidade, se aproximam do valor zero, conforme apresentado na Figura 30 indicando rastreamento completo do sinal.

Figura 29 – Resultados simulados Teste 2: Comparação entre a saída da planta e do sinal de referência - Parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$



Fonte: Autor

Figura 30 – Resultados simulados Teste 2: Sinal de erro do sistema - Parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$

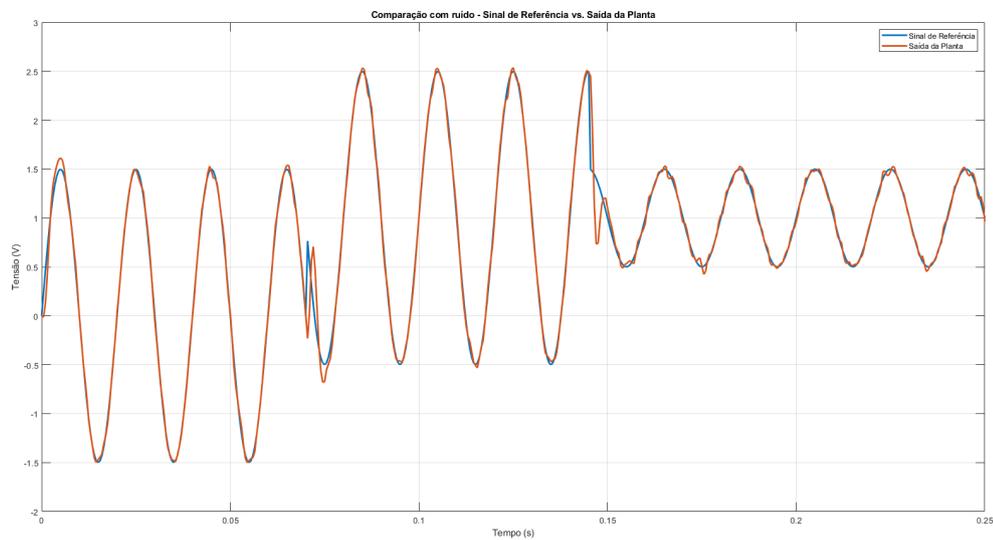


Fonte: Autor

Foi realizada também simulação no caso onde a planta é sujeita a um ruído branco gaussiano com as características já previamente descritas nesta dissertação. Tal teste tem como objetivo demonstrar a robustez do controlador proposto. Assim sendo, é apresentada

a Figura 31, onde são comparados o sinal de referência e a saída da planta sujeita ao ruído citado. Nota-se que, ainda que exista uma pequena oscilação na saída devido a ação do ruído que é aplicado diretamente nas variáveis de estado, isso não impede do sistema corretamente rastrear o sinal de referência fornecido. Isto se traduz, também, em um sinal de erro que, ainda que não seja igualado a zero quando atinge a “zona de tolerância” para ser considerado uma resposta estacionária, possui valores inferiores a variância do próprio ruído branco gaussiano injetado neste sistema, mostrando a robustez deste controlador frente às perturbações. Tal sinal de erro é apresentado na Figura 32.

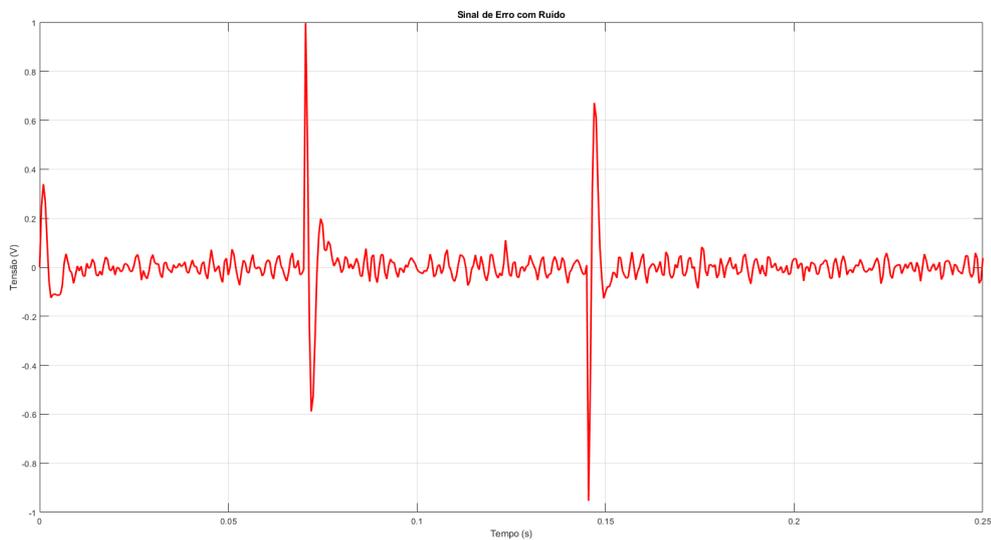
Figura 31 – Resultados simulados Teste 2: Comparação entre a saída da planta sujeita a ruído branco gaussiano e do sinal de referência - Parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$



Fonte: Autor

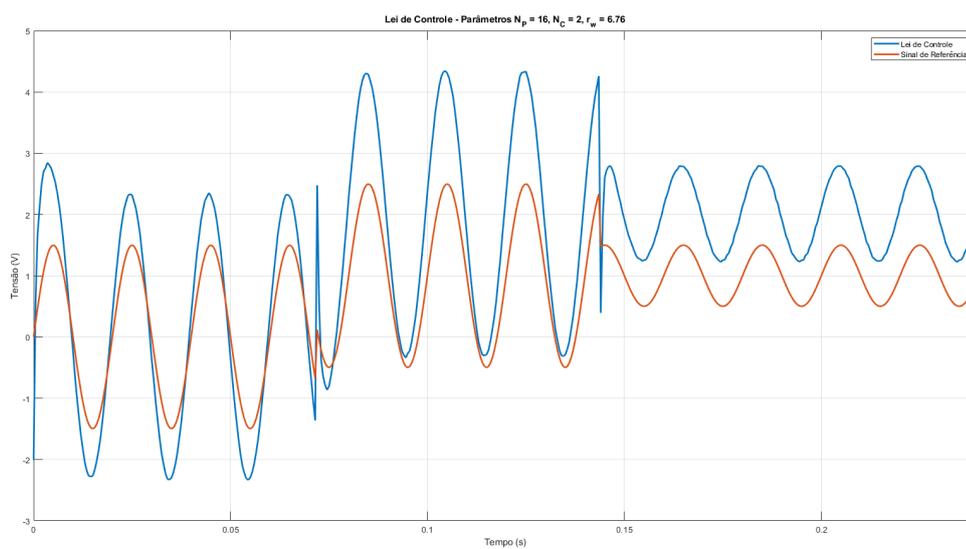
Por fim, foram utilizados os mesmos parâmetros encontrados pelo algoritmo genético e inseridos no controlador que foi utilizado no teste anterior, observando quais os resultados e comportamentos do controlador fora de um ambiente simulado, conforme apresentado nas figuras 33 e 34, demonstrando que a robustez e qualidade da resposta pode ser obtida não somente em ambientes simulados, mas também em casos concretos.

Figura 32 – Resultados simulados Teste 2: Sinal de erro do sistema sujeito a ruído branco gaussiano - Parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$



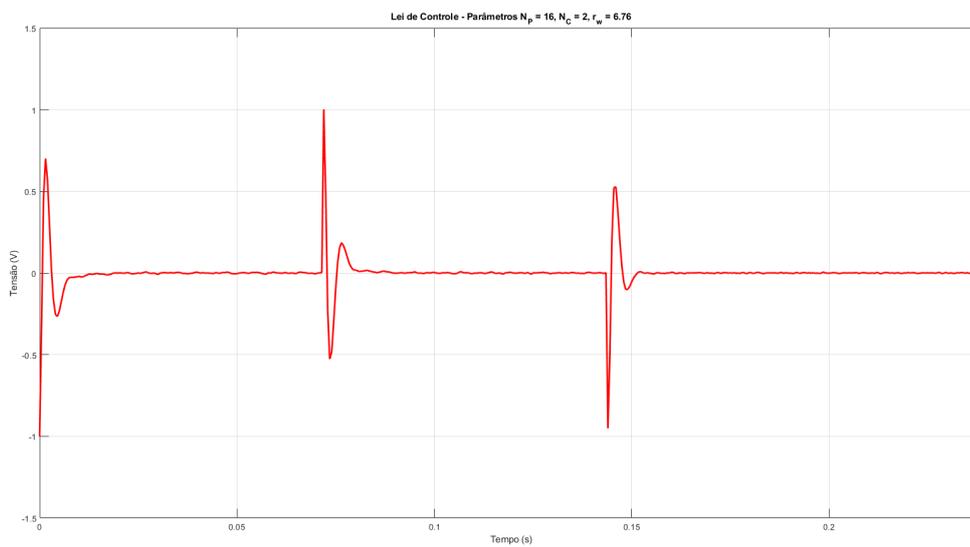
Fonte: Autor

Figura 33 – Resultados experimentais do Teste 2: Lei de Controle - Parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$



Fonte: Autor

Figura 34 – Resultados experimentais do Teste 2: Sinal de Erro - Parâmetros  $N_P = 16$ ,  $N_C = 2$  e  $r_w = 6.76$



Fonte: Autor

# Conclusão

Esta dissertação apresenta uma adaptação de um algoritmo GPC convencional para rastreamento de sinais de referência e de perturbação do tipo senoidal. Tal sinal é amplamente utilizado para descrever diferentes tipos de informações em aplicações de controle na indústria, incluindo mas não se limitando a controle de aeronaves, atuadores eletro-hidráulicos, máquinas de Vernier, entre outras. A principal vantagem da aplicação deste algoritmo de controle reside em sua facilidade de implementação, além do fato que tanto o seu modelo de predição quanto sua lei de controle são definidos em modelos de tempo discreto, o que dispensa todo o processo de discretização dos controladores em tempo real.

Durante este trabalho, foi discutido como a resposta deste tipo de controlador é dependente dos termos  $N_P$ ,  $N_C$  e  $r_w$ , motivo pelo qual foi aplicada a técnica de otimização heurística conhecida como algoritmo genético para a obtenção dos parâmetros ótimos para o modelo de controlador proposto quando uma planta discreta é sujeita ou imune à ruídos atuantes em suas variáveis de estados, buscando os menores valores possíveis quanto ao equilíbrio entre o pico da lei de controle aplicada e do erro quadrático médio existente entre o sinal de referência e a saída da planta, ambos estes sob influência de ruídos, e o tempo de assentamento do sinal de erro quando o sistema é imune ao ruído, buscando encontrar um sistema que permita rápido rastreamento sem deixar de lado a precisão e a integridade dos equipamentos associados, mesmo quando há variação de amplitude e *bias* no sinal rastreado.

## Sugestões para Trabalhos Futuros

Os resultados obtidos neste trabalho demonstram o quão promissora tal implementação se apresenta. Entretanto, o escopo dos testes tratou apenas de alterações de amplitude dos sinais, não sendo observados os casos onde a característica do sinal rastreado que se altera é a frequência. Portanto, é material para trabalhos futuros a aplicação das técnicas aqui apresentadas frente a sinais com frequências variantes.

Além disto, este trabalho considerou apenas o caso de plantas do tipo SISO, de forma que outro trabalho futuro que se origina deste é a adaptação do algoritmo proposto para sua aplicação em plantas MIMO (Múltiplas Entradas e Múltiplas Saídas, do inglês *Multiple Inputs, Multiple Outputs*).

# Referências

- BOBTSOV, A. A.; PYRKIN, A. A. Cancellation of unknown multiharmonic disturbance for nonlinear plant with input delay. *International Journal of Adaptive Control and Signal Processing*, Wiley Online Library, v. 26, n. 4, p. 302–315, 2012.
- BORDONS, C.; CAMACHO, E. F. A generalized predictive controller for a wide class of industrial processes. *IEEE Transactions on Control Systems Technology*, v. 6, n. 3, p. 372–387, 1998.
- CAMACHO, E. F.; BORDONS, C. *Model Predictive Control*. 2. ed. [S.l.]: Springer, London, 2007. 2-10 p.
- CHEN, C. *Linear System Theory and Design*. 3rd. ed. [S.l.]: Oxford University Press, 1999. 277-281 p.
- CLARKE, D. W.; MOHTADI, C.; TUFFS, P. Generalized predictive control algorithm\* part, i. the basic. *Automatica*, v. 23, n. 2, p. 137–148, 1987.
- COLEY, D. A. *An Introduction to Genetic Algorithms for Scientists and Engineers*. [S.l.]: World Scientific, 1999. ISBN 981-02-3602-6.
- CORDERO, R. et al. Development of a resonant generalized predictive controller for sinusoidal reference tracking. *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 69, n. 3, p. 1218–1222, 2022.
- CORDERO, R. et al. Tracking and rejection of biased sinusoidal signals using generalized predictive controller. *Energies*, v. 15, n. 15, 2022. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/15/15/5664>.
- CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control™ - a computer control algorithm. In: *The 86th National Meeting of the American Institute of Chemical Engineers*. [S.l.: s.n.], 1979.
- DEJONG, K. A. Genetic algorithms are not function optimizers. In: . [s.n.], 1992. Disponível em: <https://api.semanticscholar.org/CorpusID:7840248>.
- DIWEKAR, U. *Introduction to Applied Optimization*. [S.l.]: Springer Science+Business Media, 2008. ISSN 1931-6828. ISBN 978-0-387-76634-8.
- DU, Y. et al. Partitioned stator hybrid excited machine with dc-biased sinusoidal current. *IEEE Transactions on Industrial Electronics*, v. 69, n. 1, p. 236–248, 2022.
- DUNN, S. Modified genetic algorithm for the identification of aircraft structures. *Journal of aircraft*, American Institute of Aeronautics and Astronautics, v. 34, n. 2, p. 251–253, 1997.
- ESBROOK, A.; TAN, X.; KHALIL, H. K. An indirect adaptive servocompensator for signals of unknown frequencies with application to nanopositioning. *Automatica*, v. 49, n. 7, p. 2006–2016, 2013. ISSN 0005-1098. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0005109813001854>.

- ESTRABIS, T. V. *Desenvolvimento e Implementação de um Controlador Preditivo Generalizado (GPC) para Rastreamento de Referência Rampa*. Dissertação (Mestrado) — Universidade Federal de Mato Grosso do Sul, 2020.
- GANESH, U. L. et al. Review on models for generalized predictive controller. In: ACADEMY & INDUSTRY RESEARCH COLLABORATION CENTER - AIRCC. *First International Conference On Computer Science, Engineering And Applications - CCSSEA 2011*. [S.l.], 2011. p. 418–424.
- GARCÍA, C. E.; PRETT, D. M.; MORARI, M. Model predictive control: Theory and practice - a survey. *Automatica*, v. 25, n. 3, p. 335–348, 1989.
- GLOVER, F. Tabu search—part i. *ORSA Journal on computing, Informs*, v. 1, n. 3, p. 190–206, 1989.
- GRIMBLE, M. Generalized predictive optimal control: An introduction to the advantages and limitations. *International Journal of Systems Science - IJSySc*, v. 23, p. 85–98, 01 1992.
- HAYKIN, S.; VEEN, B. V. *Sinais e Sistemas*. [S.l.]: Bookman, 2001.
- HOLKAR, K. S.; WAGHMARE, L. M. An overview of model predictive control. *International Journal of Control and Automation*, v. 3, n. 4, p. 47–64, December 2010.
- HONG, F. et al. Adaptive disturbance rejection in the presence of uncertain resonance mode in hard disk drives. *Transactions of the Institute of Measurement and Control*, v. 32, n. 2, p. 99–119, 2010. Disponível em: <https://doi.org/10.1177/0142331209339845>.
- JI, W. et al. Pid controller design based on bpn for helicopter vibration attenuation. In: IEEE. *2021 40th Chinese Control Conference (CCC)*. [S.l.], 2021. p. 2045–2050.
- JIA, S. et al. Hybrid excitation stator pm vernier machines with novel dc-biased sinusoidal armature current. *IEEE Transactions on Industry Applications*, v. 54, n. 2, p. 1339–1348, 2018.
- KAMALDAR, M.; HOAGG, J. B. Adaptive harmonic control for rejection of sinusoidal disturbances acting on an unknown system. *IEEE Transactions on Control Systems Technology*, v. 28, n. 2, p. 277–290, 2020.
- KENNEL, R.; LINDER, A.; LINKE, M. Generalized predictive control (gpc)-ready for use in drive applications? In: *2001 IEEE 32nd Annual Power Electronics Specialists Conference (IEEE Cat. No.01CH37230)*. [S.l.: s.n.], 2001. v. 4, p. 1839–1844.
- KIM, W.; MOON, J. Adaptive nonlinear output tracking control with rejection of unmatched biased sinusoidal disturbances for nonlinear systems. *IEEE Access*, v. 8, p. 216210–216218, 2020.
- KIM, W. et al. Disturbance-observer-based position tracking controller in the presence of biased sinusoidal disturbance for electrohydraulic actuators. *IEEE Transactions on Control Systems Technology*, v. 21, n. 6, p. 2290–2298, 2013.
- KIRKPATRICK, S.; JR., C. D. G.; VECCHI, M. P. Optimization by simulated annealing. *science*, American association for the advancement of science, v. 220, n. 4598, p. 671–680, 1983.

- KOCHENDERFER, M. J.; WHEELER, T. A. *Algorithms for Optimization*. [S.l.]: The MIT Press, 2019. ISBN 978-0-262-03942-0.
- KONG, W. et al. Drive for dc-biased sinusoidal current vernier reluctance motors with reduced power electronics devices. In: IEEE. *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*. [S.l.], 2017. p. 1–6.
- KWON, W. H.; HAN, S. *Receding Horizon Control: Model Predictive Control for State Models*. 1. ed. [S.l.]: Springer, London, 2005.
- LEWIS, D. G.; EVANS, C. L.; SANDOZ, D. J. Application of predictive control techniques to a distillation column. *Journal of Process Control*, v. 1, p. 207–216, 1991.
- LI, A. et al. Three-phase four-leg drive for dc-biased sinusoidal current vernier reluctance machine. *IEEE Transactions on Industry Applications*, IEEE, v. 55, n. 3, p. 2758–2769, 2019.
- LIMA, D. M.; MONTAGNER, V. F.; MACCARI, L. A. Generalized predictive control with harmonic rejection applied to a grid-connected inverter with lcl filter. In: *2017 Brazilian Power Electronics Conference (COBEP)*. [S.l.: s.n.], 2017. p. 1–6.
- MAEDER, U.; MORARI, M. Offset-free reference tracking with model predictive control. *Automatica*, v. 46, n. 9, p. 1469–1476, 2010.
- MARTÍNEZ, M.; SENENT, J.; BLASCO, X. A comparative study of classical vs genetic algorithm optimization applied in gpc controller. *IFAC Proceedings Volumes*, v. 29, n. 1, p. 1769–1774, 1996. ISSN 1474-6670. 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1474667017579252>.
- MENG, Y. et al. A new hybrid-excited doubly salient dual-pm machine with dc-biased sinusoidal current. *IEEE Transactions on Applied Superconductivity*, v. 31, n. 8, p. 1–5, 2021.
- MONTEIRO, F. A. et al. Tuning of a type-iii software-based resolver-to-digital converter through genetic algorithm. In: IEEE. *2020 IEEE International Conference on Industrial Technology (ICIT)*. [S.l.], 2020. p. 576–581.
- OPPENHEIM, A. V.; WILLISKY, A. S.; NAWAB, S. H. *Signals & Systems*. 2nd. ed. [S.l.]: Prentice Hall, 1996.
- ORDYS, A. W.; CLARKE, D. W. A state-space description for gpc controllers. *International Journal of Systems Science*, Taylor & Francis, v. 24, n. 9, p. 1727–1744, 1993.
- ORUKPE, P. E. Model predictive control fundamentals. *Nigerian Journal of Technology (NIJOTECH)*, v. 31, n. 1, p. 139–148, July 2012.
- PRICE, T. C. Using co-evolutionary programming to simulate strategic behaviour in markets. *Journal of evolutionary economics*, Springer, v. 7, p. 219–254, 1997.
- PROPOI, A. Use of lp methods for synthesizing sampled-data automatic systems. *Autumn Remote Control*, v. 24, 1963.

- QIN, S. J.; BADGWELL, T. A. A survey of industrial model predictive control technology. *Transactions of ASME, Journal of Basic Engineering*, v. 11, p. 736–746, 2003.
- RAKOVIC, S.; LEVINE, W. S. *Handbook of Model Predictive Control*. 1. ed. [S.l.]: Springer, London, 2019.
- RAWLINGS, J. B.; MAYNE, D. Q.; DIEHL, M. M. *Model Predictive Control: Theory, Computation, and Design*. 2. ed. [S.l.]: LLC, California, 2019.
- RICHALET, J. et al. Model algorithmic control of industrial processes. *IFAC Proceedings Volumes*, Elsevier, v. 10, n. 16, p. 103–120, 1977.
- RIVERA, J.; ORTEGA-CISNEROS, S.; CHAVIRA, F. Sliding mode output regulation for a boost power converter. *Energies*, v. 12, n. 5, 2019. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/12/5/879>.
- SARKER, R. A.; NEWTON, C. S. *Optimization Modeling: A Practical Approach*. 1. ed. [S.l.]: CRC Press, Boca Raton, 2007.
- TAYAL, M. C.; DIWEKAR, U. M. Novel sampling approach to optimal molecular design under uncertainty. *AIChE journal*, Wiley Online Library, v. 47, n. 3, p. 609–628, 2001.
- VERMA, A. K. et al. A robust half-cycle single-phase prefiltered open-loop frequency estimator for fast tracking of amplitude and phase. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 71, p. 1–12, 2021.
- WANG, L. *Model Predictive Control System Design and Implementation Using Matlab*. 1. ed. [S.l.]: Springer, London, 2009.
- WANG, L.; ROSSITER, J. A. Disturbance rejection and set-point tracking of sinusoidal signals using generalized predictive control. In: *2008 47th IEEE Conference on Decision and Control*. [S.l.: s.n.], 2008. p. 4079–4084.
- XU, S.; CHANG, L.; SHAO, R. Single-phase voltage source inverter with voltage boosting and power decoupling capabilities. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, v. 8, n. 3, p. 2977–2988, 2020.

# Apêndices

# APÊNDICE A – Códigos

## A.1 Determinação da Função Custo no MATLAB

O código apresentado nesta seção demonstra como a função custo utilizada no algoritmo genético aplicado nesta dissertação foi definido, seus cálculos internos e o resultado que é devolvido. Devido a natureza da aplicação de AG do MATLAB, a função custo deve ser definida como uma nova função pelo usuário para que possa ser aplicada corretamente.

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %      Funcao Custo do Algoritmo Genetico
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 function J = custoGa(individuo , w1, w2, w3, ruido)
5
6 % Definicao dos termos Np Nc e rw
7 Np = individuo(1);
8 Nc = individuo(2);
9 rw = individuo(3);
10
11 % Variaveis da simulacao
12 ts = 5e-4; % tempo de amostragem
13 f0 = 50;
14 Omega = 2*pi*f0*ts;
15 gamma = 1 + 2*cos(Omega);
16
17 %Definicao sinal composto de 3 senoides
18 tFirst = 0:ts:0.07; % intervalo de tempo primeiro sinal
19 lFirst = length(tFirst); % quantidade de amostras primeiro sinal
20 sinFirst = 1.5*sin(2*pi*f0*tFirst); % 0V bias , 1.5V amplitude
21 tSec = 0.0705:ts:0.145; % intervalo de tempo segundo sinal
22 lSec = length(tSec); % quantidade de amostras segundo sinal
23 sinSec = 1 + 1.5*sin(2*pi*f0*tSec); % 1V bias , 1.5V amplitude
24 tLast = 0.1455:ts:0.25; % intervalo de tempo terceiro sinal
25 lLast = length(tLast); % quantidade de amostras terceiro sinal
26 sinLast = 1 + 0.5*sin(2*pi*f0*tLast); % 1V bias , 0.5V amplitude

```

```

27
28 t = [tFirst , tSec , tLast]; % vetor contendo o tempo da simulacao
29 lTime = length(t); %quantidade de amostras total
30 r = [sinFirst , sinSec , sinLast]; %sinal de referencia completo
31
32 % Modelo discreto da Planta
33 Ad =[0.4134 0.4536 0; -0.2404 0.7876 0; -0.4366 0.4224 0.7738];
34 Bd =[0.1331; 0.4528; 0.1273];
35 Cd=[0 0 1];
36 n = length(Ad); % ordem da planta
37
38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 %                               Calculo de Kbs
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41 E=[0 1 0; 0 0 1; 1 -gamma gamma];
42 sigma = [0;0;1];
43
44 % Matrizes do Modelo Aumentado de Predicao
45 A=[Ad zeros(n,3); sigma*Cd*Ad E];
46 B=[Bd; sigma*Cd*Bd];
47 C=[zeros(1,3) sigma'];
48
49 % —— Matrizes do GPC ——
50
51 % Criacao da Matriz F
52 nau = length(A);
53 F = zeros(Np, nau);
54 for i=1:Np
55     F(i,:) = C*(A^i);
56 end
57
58 % Criacao Matriz Phi
59 Phi = zeros(Np, Nc);
60 for col = 1:Nc
61     if col == 1
62         for linha = 1:Np
63             Phi(linha , col) = C*(A^(linha-1))*B;
64         end
65     else

```

```
66         Phi(col:Np, col) = Phi(1:Np-(col-1), 1);
67     end
68 end
69
70 PTP = Phi'*Phi; %Calculo de Phi transposta * Phi
71
72 % Matriz de selecao do primeiro elemento de U
73 if (Nc == 1)
74     vetor10 = [1];
75 else
76     vetor10 = [1 zeros(1, Nc-1)];
77 end
78
79 Rbarra = rw*eye(Nc);
80 Hessiana=inv(PTP + Rbarra); %Calculo Matriz Hessiana
81
82 % Calculo do Ganho da Planta
83 Kbs = -1*vetor10*Hessiana*(Phi')*F;
84
85 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
86 %   Simulacao Controlador Malha Fechada
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88
89 % vetor para armazenar sinais quando nao ha introducao de ruido
90 erro_sem_ruido = zeros(1,lTime);
91
92 % vetor para armazenar sinais quando ha introducao de ruido
93 erro_com_ruido = zeros(1,lTime);
94 u_com_ruido = zeros(1,lTime);
95
96 %Calculo da saida da planta e feito 2 vezes: na 1a vez a nao ha
97   ruido ,
98 %na 2a vez com ruido sendo adicionado nas variaveis de estado
99 for loop=1:2
100     x = zeros(n,lTime+1); %armazenar vetor de estados da planta
101     y = zeros(1,lTime); %armazenar saida da planta
102     erro =zeros(1,lTime); %armazenar erro
103     u =zeros(1,lTime); %armazenar lei de controle
104     xk_1=zeros(n,1); %x(k-1)
```

```

104     xk_2=zeros(n,1); %x(k-2)
105     xk_3=zeros(n,1); %x(k-3)
106     uk_1=0; %u(k-1)
107     uk_2=0; %u(k-2)
108     uk_3=0; %u(k-3)
109     ek_1=0; %e(k-1)
110     ek_2=0; %e(k-2)
111
112     % Loop por todos os tempos de amostragem
113     for k=1:lTime
114         if loop==1 %sinal sem ruado
115             xk = x(:,k); % estado da planta;
116         else % sinal com ruído
117             xk = x(:,k) + ruído(:, k); %estado da planta com
                ruído;
118         end
119         rk = r(k);
120         yk = Cd*x(:,k); %saida
121         ek = rk-yk; %erro
122         qk = xk_3+(-gamma*xk_2)+(gamma*xk_1)-xk;
123         xbarra= [qk;ek_2;ek_1;ek];
124         vk=Kbs*xbarra;
125         uk= uk_3 + gamma*(uk_1-uk_2)-vk;
126
127         % Modelo da planta : novo vetor de estado
128         x(:,k+1) = (Ad*x(:,k))+(Bd*uk);
129         % Armazenando valores
130         erro(k) = ek;
131         u(k) = uk;
132         y(k) = yk;
133         % Atualizacao de variaveis passadas
134         xk_3 = xk_2;
135         xk_2 = xk_1;
136         xk_1 = xk;
137         uk_3 = uk_2;
138         uk_2 = uk_1;
139         uk_1 = uk;
140         ek_2 = ek_1;
141         ek_1 = ek;

```

```
142     end
143
144     if loop==1
145         erro_sem_ruido = erro;
146     else
147         erro_com_ruido = erro;
148         u_com_ruido = u;
149     end
150
151 end
152
153 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
154 %           Calculo do Valor do Custo
155 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
156
157 eqm_com_ruido = mean(erro_com_ruido.^2); %erro quadratico medio
158           com ruido
159 pico_u_com_ruido = max(abs(u_com_ruido))/1000; %pico de lei de
160           controle
161
162 % — Tempo de Assentamento SEM ruido —
163
164 lSemRuido = length(erro_sem_ruido); % Numero de medicoes sem
165           ruido
166 k = lSemRuido; % contador baseado no numero de medicoes
167 ta = [0 0 0]; % vetor contendo tempos de acomodacao dos sinais
168           sem erro
169 minMed = 0; % Variavel contendo numero minimo de medicoes em
170           cada sinal
171 minErro = 0; % variavel contendo 2% do sinal de referencia
172
173 for signal=3:-1:1 % loop nos 3 sinais
174     achado = 0;
175
176     % define criterios de parada para todos as iteracoes
177     if signal == 3
178         minMed = lFirst + lSec;
179         minErro = 0.02*max(abs(sinLast));
180     elseif signal == 2
```

```
176         minMed = lFirst;
177         minErro = 0.02*max(abs(sinSec));
178     else
179         minMed = 2;
180         minErro = 0.02*max(abs(sinFirst));
181     end
182
183     while(achado == 0)
184         if abs(erro_sem_ruido(k))> minErro
185             achado = 1;
186             if signal == 1
187                 ta(signal) = ts*k;
188             else
189                 ta(signal) = ts*(k - minMed);
190             end
191         elseif (k < minMed)
192             achado = 1;
193             % se o sinal nao for estavel o suficiente , e
194             % inserido um alto
195             % valor para o tempo de assentamento , garantindo que
196             % esta
197             % resposta tera um altissimo valor de custo e
198             % provavelmente
199             % sera descartada
200             ta(signal) = 1000;
201         else
202             k = k - 1;
203         end
204     end
205
206     tAssentamento = mean(ta); %calculo media de tempo de
207     % assentamento
208
209     J = (w1*pico_u_com_ruido) + (w2*eqm_com_ruido) + (w3*
210     tAssentamento);
211
212 end
```

## A.2 Aplicação de Algoritmo Genético no MATLAB

Conforme discutido nesta dissertação, a aplicação do algoritmo genético se deu através da utilização do *software* MATLAB, que possui funções específicas para a execução de tais cálculos. Segue, então, o código utilizado para a aplicação deste algoritmo para encontrar o indivíduo ótimo.

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %                               Algoritmo Genetico
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5 % E necessario declarar uma matriz chamada paraMatrix com 24
6 % linhas e 10 colunas , que contem todos os pesos dos 24 testes
7 % e tambem ira abrigar as informacoes de cada um dos testes .
8
9 % criacao do sinal de ruido branco a ser aplicado na variavel de
   estado
10 tempo = 0:ts:0.25;
11 ruido = 0.05*randn(3, length(tempo)+1);
12
13 % Definicao dos argumentos do AG
14
15 A_des = [0 0 -1; -1 1 0; 1 0 0; 0 -1 0]; %Matriz de
   desigualdades A_des;
16 b_des = [0 0 100 -2]'; %Matriz de desigualdades b_des;
17 ub = [100, 99, 10]; % Definicao dos valores maximos aceitos para
   cada
18
   % cromossomo do individuo
19 intcon = [1 2]; % determina quais cromossomos do individuo sao
   numeros
20
   % obrigatoriamente inteiros
21
22 %Declaracao das opcoes de funcionamento do AG
23 options = optimoptions("ga", "CreationFcn", "gacreationuniformint
   ", ...
24     "MutationFcn", "mutationpower", "Display", "iter", ...
25     "SelectionFcn", "selectionroulette", ...
26     "PlotFcn", ["gaplotdistance", "gaplotselection", "
   gaplotstopping"], ...
27     "gaplotbestf", "gaplotbestindiv"]);

```

```
28
29 % — "CreationFcn", "gacreationuniformint" —
30 % Funcao de Criacao da Populacao inicial do AG
31 % "gacreationuniform" e a funcao padrao para AG quando o
    problema possui
32 % restricoes inteiras , aplicando um limite artificial para os
    componentes
33 % que nao sao limitados , criando individuos de maneira uniforme
    dentro
34 % dos limites e garantindo a aplicacao das restricoes inteiras .
35
36 % — "CrossoverFcn" —
37 % Define como o AG combina dois individuos selecionados como
    pais
38 % no caso de existir restricoes lineares , a funcao padrao
    selecionada e
39 % "crossoverintermediate"
40
41 % — "MutationFcn", "mutationpower" —
42 % Define o comportamento das possiveis mutacoes nos individuos
43 % "mutationpower" e a funcao de mutacao padrao para AG quando o
    problema
44 % envolve limites inteiros
45
46 % — "SelectionFcn", "selectionroulette" —
47 % Determina a funcao de selecao dos individuos para serem pais
48 % "selectionroulette" determina a selecao de individuos atraves
    do
49 % metodo de roleta .
50
51 % — PlotFnc —
52 % Determina quais graficos sao gerados com os dados do AG.
53 % "gaplotdistance" – distancia media entre individuos a cada
    geracao
54 % "gaplotselection" – histograma dos pais selecionados
55 % "gaplotstopping" – indicacao dos niveis de cada criterio de
    parada
56 % "gaplotbestf" – indicacao do melhor valor de resultado e do
    resultado
```

```
57 % medio
58 % "gplotbestindiv" – indicacao do individuos com a melhor valor
    de custo
59 % em cada geracao
60
61 % paraMatrix e a matriz contendo os pesos de cada um dos testes
62 for line=1:height(paraMatrix)
63     % indicacao dos pesos dos testes
64     w1 = paraMatrix(line , 1);
65     w2 = paraMatrix(line , 2);
66     w3 = paraMatrix(line , 3);
67
68     %declaracao da funcao custo sem indicar qual e o individuo x
69     planta = @(x)custoGa(x, w1, w2, w3, ruido);
70
71     % Declaracao do Algoritmo Genetico
72     [best_ind , cost] = ga(planta , 3, A_des, b_des, [], [], [],
        ub, [], ...
73         intcon , options)
74     param = newInfoPlant(best_ind , ruido)
75
76     %insercao dos parametros na matriz
77     paraMatrix(line , 4) = param(1);
78     paraMatrix(line , 5) = param(2);
79     paraMatrix(line , 6) = param(3);
80     paraMatrix(line , 7) = cost;
81     paraMatrix(line , 8) = best_ind(1);
82     paraMatrix(line , 9) = best_ind(2);
83     paraMatrix(line , 10) = best_ind(3);
84
85 end
86
87 % transformacao da matriz de parametros em uma tabela para mais
    facil visualizacao dos dados
88 gaResults = array2table(paraMatrix , 'VariableNames' , {'w1' , 'w2' ,
    'w3' , 'up' , 'eqm' , 'ta' , 'J' , 'Np' , 'Nc' , 'rw' })
```