



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada
Mestrado Acadêmico

João Olegário de Oliveira de Souza

Metaheurísticas Aplicadas na Sintonia de Controladores PID:
Estudo de casos

São Leopoldo, 2013

JOÃO OLEGÁRIO DE OLIVEIRA DE SOUZA

Metaheurísticas Aplicadas na Sintonia de Controladores PID: Estudo de Casos

Dissertação de mestrado apresentada como requisito parcial para a obtenção do título de Mestre, pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Dr. João Francisco Valiati

São Leopoldo

2013

Ficha catalográfica

S729m Souza, João Olegário de Oliveira de
Metaheurísticas aplicadas na sintonia de controladores PID:
estudo de casos / por João Olegário de Oliveira de Souza. – 2013.
163 f. : il., 30cm.

Dissertação (mestrado) - Universidade do Vale do Rio dos
Sinos, Programa de Pós-Graduação em Computação Aplicada,
2013.

Orientação: Prof. Dr. João Francisco Valiati.

1. Controladores PID. 2. Algoritmos Genéticos. 3. *Simulated
Annealing*. 4. Otimização. 5. Inteligência artificial. I. Título.

CDU 004.421

Catálogo na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

*Ao meu pai, que mesmo longe
estará sempre presente na minha vida.*

AGRADECIMENTOS

Em especial, agradeço aos meus pais queridos, Euclides (in memorian) e Eloeci, pela vida, apoio e amor inesgotável.

Aos meus irmãos, exemplos de seriedade e honestidade, que sempre me motivaram e incentivaram em mais esta etapa de minha formação.

Ao professor orientador Dr. João Francisco Valiati pela grande dedicação e apoio, fundamentais para a execução desta dissertação.

Ao professor Adelmo Luis Cechin (in memorian) pelas primeiras caminhadas deste trabalho.

Aos demais professores do mestrado em Computação Aplicada pelas valiosas contribuições prestadas ao longo desses anos de curso.

À uma pessoa muito especial que entrou na minha vida este ano, minha Aline.

A Deus por ter me concedido força e perseverança para concluir esta dissertação em meio às diversas dificuldades enfrentadas.

*“Tantas vezes pensamos ter chegado,
Tantas vezes é preciso ir além...”
Fernando Pessoa.*

RESUMO

Os controladores do tipo Proporcional, Integral e Derivativo, comumente denominados de PID, são largamente utilizados no controle de processos industriais, tanto em sistemas monovariáveis quanto em sistemas multivariáveis. Hoje, cerca de 95% dos controladores utilizam este tipo de estrutura na indústria. O grande problema é que grande parte deles estão mal sintonizados, comprometendo em muitos casos o desempenho de malhas industriais. Neste trabalho é apresentada uma revisão geral sobre os algoritmos inspirados na natureza, *Simulated Annealing* e Algoritmos Genéticos (fundamentos, características, parâmetros, operadores) e sua aplicação ao problema da sintonia de controladores PID monovariáveis e multivariáveis. É estabelecida, através de estudo de casos, uma análise comparativa entre estas sintonizações com metaheurísticas e os métodos consagrados na literatura em aplicações industriais convencionais, utilizando como função de avaliação o índice Integral do Erro Absoluto ponderado pelo Tempo (ITAE). O trabalho também propõe o estudo de controladores PID através de Algoritmos Genéticos Multiobjetivos, que satisfaçam dois critérios de desempenho: *overshoot* e o índice de desempenho Integral do Erro Quadrático ponderado pelo Tempo (ITSE). Conforme demonstrado pelos resultados obtidos, pode-se afirmar que a metaheurística Algoritmos Genéticos é um método eficiente e confiável para a otimização de problemas de sintonia de controladores PID.

Palavras-Chave: Controladores PID. Algoritmos Genéticos. *Simulated Annealing*. Otimização. Inteligência Artificial.

ABSTRACT

The Proportional, Integral and Derivative controllers, commonly called PID controllers, are widely used in industrial process control, in both SISO and multivariable systems. Today about 95% of controllers use this type of structure in the industry. The big problem is that most of them are poorly tuned, in many cases compromising the performance of industrial loops. This work presents a general review on nature-inspired algorithms, Simulated Annealing and Genetic Algorithms (basement, characteristics, parameters, operators) and its application in the problem of tuning PID controllers in both single variable and multivariable systems. There will be through case studies, a comparative analysis of these metaheuristics with established methods in the literature in conventional industrial applications using as evaluation function the Integral of time multiplied by the Absolute Error (ITAE) index. The work also proposes the study of PID controllers using multiobjective genetic algorithms which meet two performance criteria: overshoot and the Integral Time Square Error (ITSE) index. The results obtained confirm that Genetic Algorithms are an effective and reliable method to optimize complex problems.

Keywords: *PID control. Genetic Algorithms. Simulated Annealing. Optimization, Artificial Intelligence.*

LISTA DE FIGURAS

Figura 1.1: Estrutura de realimentação.....	19
Figura 1.2: Sintonia de controladores PID através de metaheurísticas.	20
Figura 2.1: Diagrama em blocos de um controlador PID.....	26
Figura 2.2: Controlador PID adaptativo indireto.....	27
Figura 2.3: Controle monovariável.....	27
Figura 2.4: Controle de sistemas em cascata.....	28
Figura 2.5: Controle PID Multivariável.	28
Figura 2.6: Controle PID Multivariável descentralizado.	30
Figura 2.7: Sistema multivariável com controlador desacoplador.	31
Figura 2.8: Controle PID Multivariável desacoplado 2 x 2.	31
Figura 2.9: Método da Curva de reação.	33
Figura 2.10: Resposta do sistema - método curva de reação.....	33
Figura 2.11: Método da Sensibilidade Limite.	34
Figura 2.12: Resposta do sistema - método da Sensibilidade Limite.....	34
Figura 2.13: Diagrama em blocos de um sistema de controle multivariável- método <i>BLT</i>	37
Figura 2.14: Resposta de um sistema subamortecido com a aplicação de um degrau no <i>setpoint</i>	38
Figura 3.1: Representação de ótimos local e global em uma função bidimensional.....	41
Figura 3.2: Classes de técnicas de otimização.....	42
Figura 3.3: Conjunto de soluções factíveis e espaço objetivo factível de um problema de minimização com duas variáveis.....	43
Figura 3.4: Funções $f_1(x)$ e $f_2(x)$ de um problema de otimização multiobjetivo.....	44
Figura 3.5: Espaço objetivo do problema multiobjetivo.	44
Figura 3.6. Dominância de Pareto no espaço objetivo.	45
Figura 3.7: Exemplos de conjuntos Pareto-Ótimos.	45
Figura 3.8: Distribuição de soluções na Frente de Pareto.	46
Figura 4.1: Exemplo de estrutura de vizinhança.	49
Figura 4.2: Fluxograma de um método de busca baseado em solução única.....	49
Figura 4.3: Fluxograma de um método de busca por população.....	50
Figura 4.4: Fluxograma do algoritmo <i>Simulated Annealing</i>	55
Figura 4.5: Fluxograma de um AG clássico.	59
Figura 4.6: Método da Roleta.	61
Figura 4.7: Método do Torneio.....	61

Figura 4.8: Operador cruzamento.....	62
Figura 4.9: Operador mutação.....	63
Figura 5.1: Ambiente integrado de simulação.....	67
Figura 5.2: Ambiente de simulação PID.....	68
Figura 6.1: Sintonia PID SISO através de otimização mono-objetivo.....	73
Figura 6.2: Sintonia PID SISO através de otimização multiobjetivo.....	89
Figura 6.3: Diagrama em blocos do controle de sistema em cascata.....	98
Figura 6.4: Diagrama em blocos do controle de sistema multivariável.....	110

LISTA DE GRÁFICOS

Gráfico 6.1: Resposta do processo em malha fechada com diferentes índices de erro.	75
Gráfico 6.2: Convergência do Algoritmo Genético.....	83
Gráfico 6.3: Tamanho da população no AG com cruzamento aritmético.	83
Gráfico 6.4: Tamanho da população no AG com cruzamento heurístico.	84
Gráfico 6.5: Cruzamento aritmético com população = 100.	84
Gráfico 6.6: Cruzamento intermediário com população = 100.	85
Gráfico 6.7: Cruzamento heurístico com população = 100.....	85
Gráfico 6.8: Convergência do <i>Simulated Annealing</i>	87
Gráfico 6.9: Resposta do processo em malha fechada com controladores PID sintonizados por metaheurísticas.	88
Gráfico 6.10: Fronteira de Pareto com a metaheurística AG empregada.....	89
Gráfico 6.11: Resposta do processo com a melhor (a) e pior solução (b).....	90
Gráfico 6.12: Resposta do processo em malha fechada com diferentes abordagens.	91
Gráfico 6.13: Resposta do processo em malha fechada com controladores PID sintonizados pelos métodos ZN e metaheurísticas.	93
Gráfico 6.14: Seguimento da referência com os métodos ZN e metaheurísticas.....	94
Gráfico 6.15: Resposta do processo SISO em malha fechada com $G_1(s)$	95
Gráfico 6.16: Resposta do processo SISO em malha fechada com $G_2(s)$	96
Gráfico 6.17: Resposta do processo SISO em malha fechada com $G_3(s)$	96
Gráfico 6.18: Resposta do processo em malha fechada com controladores PID sintonizados pelas metaheurísticas com diferentes metodologias.....	104
Gráfico 6.19: Resposta do processo em cascata com controladores PID sintonizados pelos métodos Lee et al. (1998) e metaheurísticas.....	105
Gráfico 6.20: Resposta da malha interna com controladores PID sintonizados pelos métodos Lee et al. (1998) e metaheurísticas.	106
Gráfico 6.21: Resposta do processo em cascata para alterações na referência.	107
Gráfico 6.22: Resposta do processo para um degrau na carga L_2	108
Gráfico 6.23: Resposta do processo para um degrau na carga L_1	108
Gráfico 6.24: Resposta do processo (saída y_1) em malha fechada com controladores PID sintonizados pelos métodos BLT e metaheurísticas.	118
Gráfico 6.25: Resposta do processo (saída y_2) com controladores PID sintonizados pelos métodos BLT e metaheurísticas.	118
Gráfico 6.26: Resposta da saída y_1 com os métodos BLT e metaheurísticas.	119
Gráfico 6.27: Resposta da saída y_2 com os métodos BLT e metaheurísticas.	120

Gráfico 6.28: Respostas do processo (saídas y_1 e y_2) com controladores PID sintonizados pelos métodos BLT e metaheurísticas. 121

LISTA DE TABELAS

Tabela 2.1: Sintonia do controlador através do primeiro método de Ziegler-Nichols.	34
Tabela 2.2: Sintonia do controlador através do segundo método de Ziegler-Nichols.....	35
Tabela 2.3: Sintonia do controlador através do método de Tyreus-Luyben.....	35
Tabela 2.4: Regras de sintonia para modelos FOPDT e SOPDT.	36
Tabela 4.1: Relações entre os processos físico e de otimização.....	54
Tabela 5.1: Representação do cromossomo SISO.....	69
Tabela 5.2: Representação do cromossomo das estruturas em cascata e MIMO descentralizado.	70
Tabela 5.3: Representação do cromossomo MIMO desacoplado 2 x 2.....	70
Tabela 5.4: Representação da solução SISO no SA.	71
Tabela 6.1: Valores de custo e tempo de simulação dos AGs com diferentes FO.	74
Tabela 6.2: Parâmetros de sintonia com diferentes índices de erro.	74
Tabela 6.3: Desempenho do sistema sintonizado com diferentes índices de erro.....	75
Tabela 6.4: Relação de valores para as taxas de cruzamento e mutação e para o tamanho da população - processo SISO.....	76
Tabela 6.5: Valores de custo e desvio-padrão dos AGs mono-objetivos com <i>crossover</i> aritmético, mutações uniforme e adaptativa com população $N = 30$ e $N = 50$	77
Tabela 6.6: Valores de custo e tempo de simulação dos AGs mono-objetivos com <i>crossover</i> aritmético, mutações uniforme e adaptativa com população $N = 80$ e $N = 100$. 78	
Tabela 6.7: Valores de custo e tempo de simulação dos AGs mono-objetivos com <i>crossover</i> intermediário, mutações uniforme e adaptativa com população $N = 30$ e $N = 50$	79
Tabela 6.8: Valores de custo e tempo de simulação dos AGs mono-objetivos com <i>crossover</i> intermediário, mutações uniforme e adaptativa com população $N = 80$ e $N = 100$	80
Tabela 6.9: Valores de custo e tempo de simulação dos AGs mono-objetivos com <i>crossover</i> heurístico, mutações uniforme e adaptativa com população $N = 30$ e $N = 50$... 81	
Tabela 6.10: Valores de custo e tempo de simulação dos AGs mono-objetivos com <i>crossover</i> heurístico, mutações uniforme e adaptativa com população $N = 80$ e $N = 100$.. 82	
Tabela 6.11: Valores para temperatura inicial.....	86
Tabela 6.12: Valores de custo dos SAs mono-objetivos	87
Tabela 6.13: Valores de custo das metaheurísticas.	88
Tabela 6.14: Parâmetros de sintonia com metaheurísticas.....	88
Tabela 6.15: Desempenho do sistema sintonizado com metaheurísticas.....	89
Tabela 6.16: Parâmetros de sintonia com diferentes abordagens.....	91

Tabela 6.17: Desempenho do sistema sintonizado com diferentes abordagens.....	92
Tabela 6.18: Parâmetros de sintonia pelos métodos ZN e metaheurísticas.....	92
Tabela 6.19: Desempenho do sistema com os métodos ZN e metaheurísticas.	93
Tabela 6.20: Robustez do sistema com os métodos ZN e metaheurísticas.	94
Tabela 6.21: Desempenho do processo com alterações de parâmetros do sistema.....	97
Tabela 6.22: Relação de valores para as taxas de cruzamento e mutação e tamanho da população.....	99
Tabela 6.23: Valores de custo e desvio-padrão dos AGs com <i>crossover</i> aritmético e mutações uniforme e adaptativa - estrutura em cascata.	100
Tabela 6.24: Valores de custo e desvio-padrão dos AGs com <i>crossover</i> heurístico e mutações uniforme e adaptativa - estrutura em cascata.	101
Tabela 6.25: Valores de custo dos SAs - estrutura em cascata.....	102
Tabela 6.26: Valores de custo das metaheurísticas.	103
Tabela 6.27: Parâmetros de sintonia do sistema em cascata para diferentes metodologias.	103
Tabela 6.28: Desempenho dos sistemas em cascata com diferentes metodologias.....	104
Tabela 6.29: Parâmetros de sintonia pelos métodos Lee et al. (1998) e metaheurísticas..	105
Tabela 6.30: Desempenho dos sistemas em cascata sintonizados pelos métodos Lee et al. (1998) e metaheurísticas.....	106
Tabela 6.31: Relação de valores para a taxa de cruzamento - processo MIMO.....	110
Tabela 6.32: Média e desvio-padrão dos AGs com <i>crossover</i> aritmético e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 30$	111
Tabela 6.33: Média e desvio-padrão dos AGs com <i>crossover</i> aritmético e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 50$	112
Tabela 6.34: Média e desvio-padrão dos AGs com <i>crossover</i> aritmético e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 100$	113
Tabela 6.35: Média e desvio-padrão dos AGs com <i>crossover</i> heurístico e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 30$	114
Tabela 6.36: Média e desvio-padrão dos AGs com <i>crossover</i> heurístico e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 50$	115
Tabela 6.37: Média e desvio-padrão dos AGs com <i>crossover</i> heurístico e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 100$	116
Tabela 6.38: Parâmetros de sintonia pelos métodos BLT e metaheurísticas.....	117
Tabela 6.39: Desempenho dos sistemas MIMO 2 x 2 com controladores PID sintonizados pelos métodos BLT e metaheurísticas.....	119
Tabela 6.40: Robustez dos sistemas MIMO.....	122

LISTA DE ABREVIATURAS

ACO	<i>Ant Colony Optimization</i> (Otimização por Colônia de Formigas)
AG	Algoritmo Genético
AGM	Algoritmo Genético Multiobjetivo
BLT	<i>Biggest Log-Modulus Tuning</i>
EP	<i>Evolutionary Programming</i> (Programação Evolucionária)
FOPDT	<i>First-order plus dead time</i> (Primeira ordem com tempo morto)
GA	<i>Genetic Algorithms</i> (Algoritmos Genéticos)
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
GS	<i>Gain Schedule</i> (Escalonamento de Ganho)
FO	Função Objetivo
IAE	<i>Integral Absolute Error</i> (Integral do Erro Absoluto)
IFT	<i>Iterative Feedback Tuning</i>
IMC	<i>Internal Model Control</i>
ISE	<i>Integral Square Error</i> (Integral do Erro ao Quadrado)
ITAE	<i>Integral Time Absolute Error</i> (Integral do Erro Absoluto ponderado pelo Tempo)
ITSE	<i>Integral Time Square Error</i> (Integral do Erro Quadrático ponderado pelo Tempo)
LQR	<i>Linear Quadratic Regulator</i> (Regulador Linear Quadrático)
MATLAB	<i>MATrix LABoratory</i>
MIMO	<i>Multiple Input, Multiple Output</i> (Múltiplas Entradas, Múltiplas Saídas)
MOGA	<i>Multi-Objective Genetic Algorithms</i>
MOOP	<i>Multi-Objective Optimization Problem</i> (Problema de Otimização Multiobjetivo)
MRAC	<i>Model-Reference Adaptive Control</i> (Controle Adaptativo por Modelo de Referência)
MV	<i>Manipulated Variable</i> (Variável manipulada)
NPGA	<i>Niched Pareto Genetic Algorithm</i>
NSGA	<i>Non dominating Sorting Genetic Algorithm</i>
OS	<i>Overshoot</i> (Sobressalto)
PAES	<i>Pareto Archived. Evolution Strategy</i>
PESA	<i>Pareto Envelope based Selection</i>

PD	Proporcional, Derivativo
PI	Proporcional, Integral
PID	Proporcional, Integral, Derivativo
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)
PV	<i>Process Variable</i> (Variável do processo)
RBEMCE	<i>Reference Based Error with Minimum Control Effort</i> (Referência Baseada no Erro com Mínimo de Esforço de Controle)
RGA	<i>Relative Gain Array</i> (Matriz de Ganho Relativo)
RNA	Redes Neurais Artificiais
SA	<i>Simulated Annealing</i> (Recozimento Simulado)
SAT	<i>Sequential Autotuning</i>
SISO	<i>Single Input, Single Output</i> (Uma Entrada, Uma Saída)
SLC	<i>Sequential Loop Closing</i>
SOPDT	<i>Second-order plus dead time</i> (Segunda ordem com tempo morto)
SP	<i>Setpoint</i> (referência)
SPEA	<i>Strength Pareto Evolutionary Algorithm</i>
STR	<i>Self-Tuning Regulator</i> (Regulador Auto-Sintonizável)
TITO	<i>Two Input, Two Outputs</i> (Duas Entradas, Duas Saídas)
VEGA	<i>Vector-Evaluated Genetic Algorithm</i>
WPM	<i>Weighted Product Method</i> (Método do produto ponderado)
ZN	Ziegler-Nichols

LISTA DE SÍMBOLOS

NOTAÇÕES

E	Vetor erro
$E(s)$	Função de entrada do controlador no domínio Laplaciano
$e(t)$	Erro de uma variável de processo monitorada
\in	Relação de pertinência
$G(s)$	Função de transferência no domínio Laplaciano
$G_c(s)$	Função de transferência do controlador
K_{cr}	Ganho crítico de um sistema em malha aberta
K_p	Parâmetro do ganho proporcional dos controladores PD, PI e PID
K_i	Parâmetro do ganho integral dos controladores PI e PID
K_d	Parâmetro do ganho derivativo dos controladores PD e PID
L	<i>Delay time</i> (Atraso de transporte ou tempo morto)
max	Máximo
min	Mínimo
N	Tamanho da população - AG
N_k	Número de transições - SA
OS	<i>Overshoot</i> (Sobressalto)
P_{cr}	Período crítico de um sistema em oscilação sustentada
t	Tempo
T	Temperatura
T_0	Temperatura inicial - SA
T_d	Constante (de tempo) derivativa dos controladores PD e PID
T_i	Constante (de tempo) integrativa dos controladores PI e PID
t_R	<i>Rise time</i> (Tempo de subida)
t_S	<i>Settling time</i> (Tempo de acomodação)
$U(s)$	Função de saída do controlador no domínio Laplaciano
$u(t)$	Comportamento temporal da entrada do processo
X	Conjunto de soluções factíveis
$y(t)$	Comportamento temporal da saída do processo
$y_r(t)$	Comportamento temporal do <i>setpoint</i>
Y	Vetor de saída atual
$Y(s)$	Função de saída no domínio Laplaciano

Y_r	Vetor de saída desejado
Z	Espaço objetivo factível

LETRAS GREGAS

ξ	Constante de amortecimento
λ	Constante de tempo da malha fechada
τ	Constante de tempo
θ	Tempo morto
σ	Desvio padrão

SUMÁRIO

1 INTRODUÇÃO	19
1.1 Objetivos	21
1.2 Metodologia	21
1.3 Organização da Dissertação	22
2 CONTROLADORES PID	23
2.1 Breve história do controle automático	23
2.2 Ações de controle	24
2.2.1 Ação do controle proporcional	24
2.2.2 Ação do controle integral.....	24
2.2.3 Ação do controle derivativa	25
2.2.4 Estrutura do controlador PID.....	25
2.2.5 Controlador PID digital	26
2.2.6 Controlador PID adaptativo	26
2.3 Estruturas de controle	27
2.3.1 Controle de sistemas monovariáveis	27
2.3.2 Controle de sistemas em cascata	28
2.3.3 Sistemas Multivariáveis	28
2.4 Técnicas convencionais de sintonia	32
2.4.1 Ziegler-Nichols (ZN)	33
2.4.2 Método de sintonia através de séries de Maclaurin.....	35
2.4.3 Método BLT (<i>Biggest Log-Modulus Tuning</i>)	36
2.5 Critérios de desempenho	38
2.5.1 Integral Absoluto do Erro (IAE)	39
2.5.2 Integral do Erro Quadrático (ISE).....	39
2.5.3 Integral do Erro Absoluto ponderado pelo Tempo (ITAE).....	39
2.5.4 Integral do Erro Quadrático ponderado pelo Tempo (ITSE)	39
2.6 Robustez dos sistemas de controle.....	40
2.7 Comentários finais	40
3 OTIMIZAÇÃO	41
3.1 Definição e conceitos	41
3.2 Otimização Mono-objetivo.....	42
3.3 Otimização Multiobjetivo	43
3.3.1 Frentes de Pareto.....	43
3.3.2 Metas em Otimização Multiobjetivo	46
3.4 Considerações finais	46
4 METAHEURÍSTICAS	47
4.1 Definição e propriedades	47
4.2 Algumas principais metaheurísticas.....	48
4.3 Classificação das metaheurísticas.....	48
4.4 Diversificação e Intensificação.....	50
4.5 Estado da arte em controladores PID com metaheurísticas	51
4.6 <i>Simulated Annealing</i> (SA)	54
4.6.1 Vantagens e desvantagens do SA.....	56
4.6.2 Aspectos do método SA.....	56
4.6.3 Funções de Resfriamento	56
4.6.4 Geração de novas soluções (Função <i>Annealing</i>)	57
4.6.5 Temperatura inicial.....	57
4.6.6 <i>Reannealing</i>	58
4.7 Algoritmos Genéticos (AG)	58
4.7.1 Definições.....	58
4.7.2 Estrutura básica de um algoritmo genético.....	59
4.7.3 Vantagens e desvantagens do AG	59
4.7.4 Representação do AG	60
4.7.5 Função <i>fitness</i>	60

4.7.6 Operador de seleção	60
4.7.7 Elitismo	61
4.7.8 Operador de cruzamento (<i>crossover</i>)	62
4.7.9 Operador de mutação	63
4.7.10 Definição de parâmetros	64
4.7.11 Algoritmos Genéticos para Otimização Multiobjetivo	64
4.8 Considerações sobre as metaheurísticas aplicadas ao problema da sintonia	65
5 AMBIENTE DE SIMULAÇÃO	67
5.1 Configurações básicas das metaheurísticas adaptadas ao problema	68
5.1.1 Algoritmo Genético adaptado ao problema	69
5.1.2 <i>Simulated Annealing</i> adaptado ao problema	70
6 ESTUDO DE CASOS	73
6.1 Controle monovariável	73
6.1.1 Considerações iniciais para o sistema SISO	73
6.1.2 Experimentos dos algoritmos no controle de sistemas SISO	74
6.1.3 Desempenho das metaheurísticas no sistema monovariável	92
6.1.4 Robustez do sistema monovariável	94
6.1.5 Conclusões da sintonia dos controladores PID monovariáveis	97
6.2 Estrutura em cascata	97
6.2.1 Considerações iniciais para a estrutura em cascata	97
6.2.2 Experimentos dos algoritmos na estrutura em cascata	99
6.2.3 Desempenho das metaheurísticas no sistema em cascata	103
6.2.4 Distúrbios aplicados ao sistema em cascata	107
6.2.5 Conclusões da sintonia para estruturas em cascata	109
6.3 Controle multivariável	109
6.3.1 Considerações iniciais para o controle multivariável	109
6.3.2 Experimentos dos algoritmos no controle de sistemas MIMO	110
6.3.3 Desempenho das metaheurísticas no sistema MIMO 2 x 2	117
6.3.4 Robustez do sistema MIMO 2 x 2	120
6.3.5 Conclusões da sintonia dos controladores PID multivariáveis	122
6.4 Síntese dos Resultados	123
7 CONCLUSÕES	125
7.1 Trabalhos Futuros	126
8 REFERÊNCIAS BIBLIOGRÁFICAS	127
APÊNDICE A - <i>Scripts</i> para determinação dos parâmetros PID	137

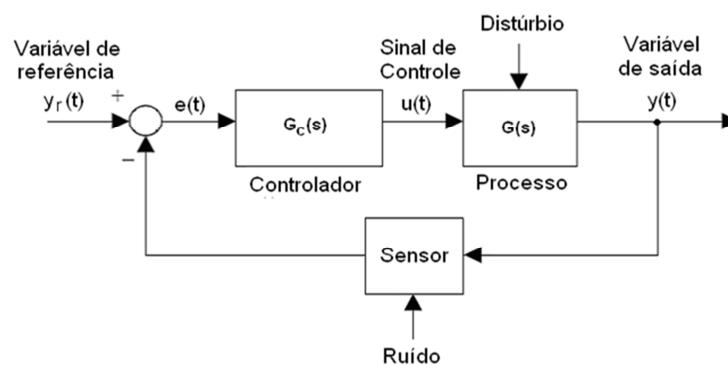
1 INTRODUÇÃO

Atualmente os sistemas de controle têm assumido um papel cada vez mais importante no desenvolvimento e avanço da civilização moderna e da tecnologia. Praticamente todos os aspectos do nosso dia-a-dia são afetados por algum tipo de sistema de controle (GOLNARAGHI; KUO, 2009).

Sistemas de controle são compostos de um processo a ser controlado que pode ser, por exemplo, um avião, um manipulador robótico ou um processo químico; um controlador responsável pela inteligência do sistema; um atuador que é capaz de converter um sinal elétrico recebido do controlador em outra grandeza física e um sensor que transforma a grandeza física a ser controlada em um sinal elétrico.

O conceito do sistema de controle automático é baseado na teoria da realimentação mostrado no diagrama em blocos da figura abaixo:

Figura 1.1: Estrutura de realimentação.



Fonte: Adaptado de Ogata (1997).

A variável $y_r(t)$ corresponde à referência do sistema (também chamada de *setpoint* ou SP). A saída do sistema, também chamada de variável do processo (PV - *Process Variable*), é representada por $y(t)$. A entrada do controlador $G_C(s)$ é dada pela variável $e(t)$ que corresponde ao erro que é a diferença entre valor desejado e o valor de saída. A grandeza $u(t)$ corresponde à variável de entrada do processo $G(s)$, ou seja, a variável manipulada (MV - *Manipulated Variable*). Basicamente, o funcionamento do controlador consiste em analisar o sinal de erro e enviar um comando para corrigir o sinal de entrada do processo.

Apesar dos processos industriais estarem se tornando cada vez mais complexos, os controladores Proporcional, Integral, Derivativo (PID) continuam sendo utilizados largamente em laços de controle industriais devido a sua estrutura simples e eficiente.

Um dos primeiros passos para o projeto de tais controladores é a sintonia, que consiste na determinação dos parâmetros K_p (ganho proporcional), K_i (ganho integral) e K_d (ganho derivativo) visando um desempenho satisfatório do sistema de controle em malha fechada. Estes parâmetros podem ser determinados em função do desempenho no domínio do tempo e/ou no domínio da frequência.

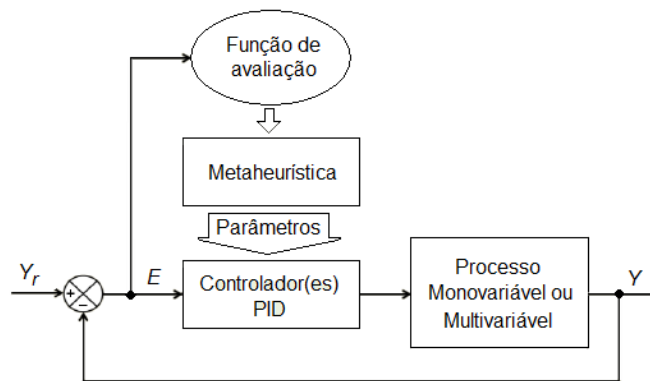
O grande problema dos controladores PID é que grande parte deles estão mal sintonizados, representando um alto custo para a indústria (CAMPESTRINI, 2006). Além disso, os parâmetros dos controladores são, muitas vezes, sintonizados de forma manual,

através de métodos de tentativa e erro, comprometendo em muitos casos o desempenho de malhas industriais (LUYBEN, 1990 e CARMO, 2006).

Åström e Hägglund (1995) destacam que apesar de sua importância e disseminação a mais de cinco décadas, existe grande potencial para melhoria do controlador PID. A habilidade de determinar a melhor configuração e estrutura do sistema de controle para um determinado problema tem motivado o desenvolvimento de estruturas de projeto baseadas em otimização para procurar por soluções com menor carga computacional (SKOGESTAD; POSTLETHWAITE, 1996).

Desta forma, a motivação deste trabalho é a busca por métodos de sintonia mais eficientes, onde as metaheurísticas, segundo Nagaraj e Vijayakumar (2011), têm demonstrado excelência nos resultados apresentados, baseados em critérios de desempenho dos processos no domínio do tempo e nos índices de desempenho. Luz (2012) ainda descreve que as metaheurísticas permitem obter boas soluções, em alguns casos, até mesmo a solução ótima. A figura 1.2 mostra o exemplo de um sistema de controle na forma de diagrama em blocos, onde a metaheurística deve ser capaz de realizar a sintonia do(s) controlador(es) PID tendo como ponto de partida um degrau unitário aplicado na sua referência.

Figura 1.2: Sintonia de controladores PID através de metaheurísticas.



Fonte: Elaborado pelo autor.

Encontra-se na literatura uma grande gama de métodos metaheurísticos para a sintonia de controladores PID. Neste trabalho foram utilizadas duas técnicas: Algoritmos Genéticos e *Simulated Annealing* que têm sido amplamente empregadas na resolução de problemas de otimização. Coello et al. (2005) descreve que os Algoritmos Genéticos têm potencial de procurar soluções em superfícies complexas da função objetivo, incorporando o conhecimento do projetista tanto na formulação como na solução de problemas. Justifica-se ainda o uso destes dois métodos a fim de verificar a influência dos seus parâmetros intrínsecos e, cujos trabalhos correlatados em alguns problemas do capítulo 4 não levam em consideração sua influência sobre a otimização dos parâmetros PID.

Embora existam diversas análises que apontam para uma impossibilidade de se determinar qual metaheurística é a melhor, entre elas o teorema “*No Free Lunch*”, é possível determinar quais as abordagens são mais indicadas para um determinado tipo de problema (MOEDINGER, 2005).

Pode-se afirmar que uma das contribuições deste trabalho é a utilização das metaheurísticas Algoritmos Genéticos e *Simulated Annealing* como uma ferramenta de otimização, capaz de resolver problemas reais na área de Engenharia de Controle e Automação, principalmente no aumento do desempenho de processos industriais.

1.1 Objetivos

O objetivo deste trabalho é investigar a utilização de metaheurísticas aplicadas à solução do problema da sintonia dos parâmetros de controladores PID, para diferentes estruturas de controle, propondo para cada estrutura diferentes abordagens para a função objetivo, com alternância de metodologias de sintonia e arquiteturas, com a intenção de verificar, em função de especificações de desempenho e robustez, a eficiência dos métodos empregados.

Os objetivos específicos a serem alcançados são:

- Revisar as estruturas de controladores PID;
- Estudar e aplicar as metaheurísticas Algoritmo Genético (AG) e *Simulated Annealing* (SA) na sintonia dos parâmetros de controladores PID monovariáveis e multivariáveis;
- Testar as diferentes configurações de parâmetros de trabalho para o Algoritmo Genético e o *Simulated Annealing*;
- Realizar simulações e análises comparativas entre os resultados obtidos com o uso de controladores sintonizados por técnicas inteligentes e os resultados obtidos através de técnicas de sintonia convencionais, em função de critérios no domínio do tempo, índice de desempenho, robustez e eliminação de distúrbios nas malhas.

1.2 Metodologia

A metodologia científica adotada consta das seguintes fases:

- Levantamento do estado da arte dos controladores PID sintonizados por metaheurísticas;
- Aplicação dos algoritmos na sintonia de sistemas monovariáveis, cascata e multivariáveis 2 X 2, testando variações do tamanho da população e dos tipos e taxas de cruzamento e mutação para o AG e alterações da temperatura inicial e das funções de resfriamento e geração de novas soluções para o SA.
- Aplicação dos algoritmos empregados em estudos de casos:
 - Estudo de caso 1: sistema monovariável sintonizado por Algoritmos Genéticos e *Simulated Annealing*, comparando seus resultados com o método Ziegler-Nichols;
 - Estudo de caso 2: Estrutura em cascata sintonizada por AG e SA comparando seus resultados com os resultados do método convencional proposto por Lee et al. (1998).
 - Estudo de caso 3: Controladores PID descentralizados e com desacopladores sintonizados por AG multiobjetivo controlando o sistema de coluna de destilação Wood and Berry. Seus resultados são comparados com o método *Biggest Log-Modulus Tuning* (BLT).

Para obter os resultados comparativos dos controladores foram utilizados os ambientes computacionais MATLAB® e Simulink® da Mathworks, especificamente as ferramentas de otimização global (*Global Optimization Toolbox*) com os métodos *Simulated Annealing Solver*, *Genetic Algorithm Solver* e *Multiobjective Genetic Algorithm Solver*.

1.3 Organização da Dissertação

A seguir é apresentada uma breve descrição dos capítulos que compõem esta dissertação. No capítulo 2 apresentam-se algumas considerações sobre sistemas de controle, conceitos e estruturas de controladores PID e métodos de ajuste clássicos.

O capítulo 3 aborda o assunto de otimização apresentando a formulação geral e os conceitos básicos do tema, introduzindo a otimização multiobjetivo.

O capítulo 4 retrata o estado da arte e exploram-se os fundamentos de cada uma das técnicas metaheurísticas que são empregadas nesta dissertação, no caso os Algoritmos Genéticos e o *Simulated Annealing*. São apresentadas também as vantagens e desvantagens da aplicação destas técnicas.

Já os capítulos 5 e 6 envolvem respectivamente a descrição do ambiente de simulação baseado nas ferramentas MATLAB® e Simulink® e os três estudos de caso com os métodos de sintonia empregados. Dessa forma, são projetados controladores PID (de forma *off-line*) com o objetivo de satisfazer para cada estrutura de controle certas especificações de desempenho. Após são apresentadas simulações das aplicações e discussão dos resultados obtidos.

O capítulo 7 compreende as conclusões, assim como sugestões de trabalhos futuros, baseadas nos resultados obtidos neste trabalho.

2 CONTROLADORES PID

Os controladores PID são largamente utilizados nas indústrias para aplicações de instrumentação, controladores de voo, indústria automobilística, entre outros. Cerca de 95% das malhas de controle de processos industriais são do tipo PID (ÅSTROM; HÄGGLUND, 1995).

A seguir, são apresentados os conceitos e características dos controladores PID, abordando as diversas estruturas de controle e os métodos convencionais de sintonia.

2.1 Breve história do controle automático

A partir do controle de velocidade para uma máquina a vapor desenvolvido por James Watt (1736-1819) em 1769, o estudo de sistemas automáticos de controle atraiu a atenção de importantes cientistas e engenheiros dos séculos XIX e XX. Um desses grandes nomes foi James Clerk Maxwell (1831-1879) que contribuiu com o trabalho *On Governors* em 1868, no estudo da estabilidade de sistemas dinâmicos. Maxwell investigou a estabilidade de sistemas dinâmicos em função de localizar as raízes de suas equações características. Isto o levou a especificar que um sistema era instável quando a parte real de suas raízes fosse positiva.

Edward Routh (1831-1907) abordou de forma matemática a estabilidade de sistemas de ordem superior através dos trabalhos de Augustin-Louis Cauchy (1789-1857) e Charles Sturm (1803-1855). Em 1895, Adolf Hurwitz (1859-1919) resolveu o problema de estabilidade de sistemas lineares, através de um conjunto de determinantes. Em 1922, Nicholas Minorsky (1885-1970) apresentou uma análise clara dos sistemas de controle de posição e formulou uma lei de controle que conhecemos hoje como controle PID. Outra importante contribuição foi de Harry Nyquist (1889-1976), em 1932 desenvolveu o método de analisar e de projetar os sistemas de controle sem requerer a manipulação de equações diferenciais (método resposta em frequência) Este trabalho teve continuidade através dos estudos realizados por Hendrik W. Bode (1905-1982) que em 1940 introduziu os conceitos de Margem de Ganho e Margem de Fase. Em 1934, Harold Locke Hazen (1901-1980) empreendeu o primeiro estudo teórico sobre servomecanismos (BENNET, 1996).

Outro marco da evolução da teoria e prática de controle automático ocorreu durante a Segunda Guerra Mundial, quando se tornou necessário o projeto e a construção de sistemas bélicos como radares, sistemas de piloto automático para aviões e outros sistemas militares baseados em controladores com realimentação. A complexidade e o desempenho desejados desses sistemas militares necessitaram uma extensão das técnicas de controle disponíveis e promoveu o interesse nos sistemas de controle e a pesquisa de novos métodos (DORF; BISHOP, 1998). Com exceção da técnica Lugar das Raízes, criada por Walter Evans (1920-1999) em 1948, o fim da segunda guerra culminou com a consolidação das técnicas de controle clássico. A busca por novos métodos impulsionou o chamado controle moderno. Entre os principais nomes citam-se as contribuições de Aleksandr Lyapunov (1857-1918) e Rudolf Kalman.

2.2 Ações de controle

Resumidamente, o controlador PID clássico é um algoritmo simples que possui três ações de controle (proporcional, integral e derivativa) e seus parâmetros devem ser corretamente sintonizados para um correto funcionamento do sistema, de maneira a obter um sistema robusto e com desempenho aceitável, conforme critério estabelecido. A seguir, serão descritas as principais características de cada ação de controle

2.2.1 Ação do controle proporcional

No controle proporcional (P), a ação de controle é proporcional ao erro. O efeito da ação proporcional pode ser verificado na equação abaixo:

$$y(t) = K_p e(t). \quad (2.1)$$

Apenas com a ação proporcional o desempenho do controlador é limitado. Com o aumento do ganho proporcional é obtida maior rapidez na resposta do sistema, aumento de sobressalto e redução do erro em regime permanente, porém essa diminuição jamais apresentará valor nulo.

A função de transferência da ação proporcional é dada por:

$$G_C(s) = \frac{U(s)}{E(s)} = K_p, \quad (2.2)$$

onde $U(s)$ e $E(s)$ correspondem respectivamente às funções de saída e entrada do controlador no domínio Laplaciano.

2.2.2 Ação do controle integral

Neste controlador, a saída é proporcional à integral do erro.

$$y(t) = K_p \frac{1}{T_i} \int_0^t e(t) dt. \quad (2.3)$$

Esta ação é utilizada para eliminar o erro em regime permanente. Quanto maior a relação K_p/T_i , maior será a tendência ao sistema apresentar oscilação e ocorrerá aumento do sobressalto. Em geral, só a ação integral não é utilizada, mas em combinação com a ação de controle P (GARCIA, 2001 apud LIMA, 2008) conforme equação (2.4):

$$y(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right]. \quad (2.4)$$

Aplicando a transformada de Laplace na equação acima, a função de transferência da ação proporcional-integral é dada por:

$$G_C(s) = K_p \left(1 + \frac{1}{T_i s} \right). \quad (2.5)$$

2.2.3 Ação do controle derivativo

O efeito da ação derivativa é proporcional à taxa de variação do erro, de acordo com a equação abaixo:

$$y(t) = K_p T_d \frac{de(t)}{dt}. \quad (2.6)$$

Seu objetivo é melhorar o comportamento transitório do sistema em malha fechada. Na prática, a ação derivativa não pode ser implementada isoladamente, ou seja, também é utilizada em combinação com a ação proporcional e sua equação é dada por:

$$y(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right). \quad (2.7)$$

A função de transferência da ação proporcional-derivativa é dada por:

$$G_C(s) = K_p (1 + T_d s). \quad (2.8)$$

2.2.4 Estrutura do controlador PID

Combinando as três ações de controle, o controlador PID é representado matematicamente como:

$$y(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right). \quad (2.9)$$

Aplicando a transformada de Laplace na equação acima, tem-se:

$$G_C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right). \quad (2.10)$$

Pode-se apresentar a equação da seguinte forma:

$$G_C(s) = K_p + \frac{K_i}{s} + K_d s, \quad (2.11)$$

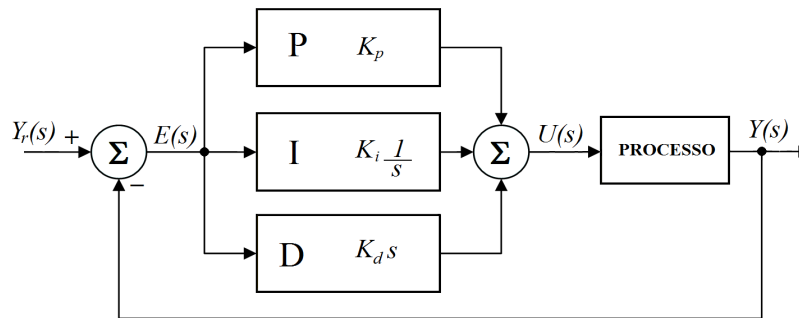
onde:

$$K_i = \frac{K_p}{T_i}, \quad (2.12)$$

$$K_d = K_p T_d. \quad (2.13)$$

O diagrama em blocos da figura 2.1 mostra o efeito de cada bloco do controlador na estrutura de controle.

Figura 2.1: Diagrama em blocos de um controlador PID.



Fonte: Adaptado de Ogata (1997).

Na figura 2.1, os termos K_p , K_i e K_d são os ganhos das parcelas P, I e D a serem determinados através das metaheurísticas empregadas.

2.2.5 Controlador PID digital

Durante os últimos anos, o campo de aplicação dos controles sofreu rápida expansão devido, principalmente, aos avanços dos semicondutores na área da eletrônica de potência bem como o processamento de sinais analógicos e digitais, culminando com a microeletrônica e os microprocessadores (LEONHARD, 2001). Hoje, praticamente todos os sistemas de controle que utilizam controladores PID são implementados em microprocessadores (ÅSTRÖM; HÄGGLUND, 1995).

2.2.6 Controlador PID adaptativo

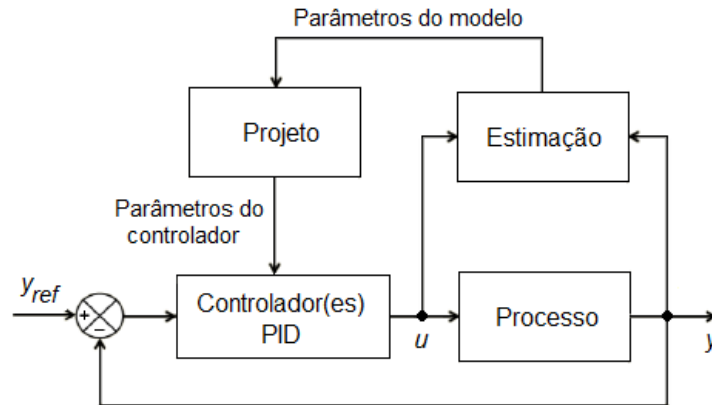
Diferentemente dos controladores convencionais (não adaptativos) em que os parâmetros do controlador são computados a partir dos parâmetros do processo, no controlador adaptativo realiza-se uma estimação dos parâmetros da planta através de algum estimador (*on-line*) de parâmetros que são utilizados no cálculo da lei de controle. Basicamente, um controlador adaptativo é um controlador com parâmetros ajustáveis e que apresenta um mecanismo para ajuste destes parâmetros (ÅSTRÖM; WITTENMARK, 1995).

Diferentes enfoques para o controle adaptativo são propostos na literatura entre eles: Escalonamento de Ganho (*Gain Schedule* - GS), Controle Adaptativo por Modelo de Referência (*Model-Reference Adaptive Control* - MRAC) e Regulador Auto-Sintonizável (*Self-Tuning Regulator* - STR) (ÅSTRÖM; WITTENMARK, 1995).

Existem duas abordagens de como o estimador de parâmetros é combinado com a lei de controle: direto ou indireto. No controle adaptativo direto, os parâmetros do controlador são ajustados diretamente pela otimização de um índice de desempenho baseado na saída do processo. O controle adaptativo indireto consiste basicamente de duas operações: construção do modelo via identificação e desenvolvimento do controlador usando o modelo identificado.

Na figura 2.2 apresenta-se em diagrama em blocos a estrutura básica de um controle PID adaptativo indireto.

Figura 2.2: Controlador PID adaptativo indireto.



Fonte: Adaptado de Dumont e Huzmezan (2002).

O controlador indireto é normalmente desenvolvido em tempo discreto, onde a cada instante de amostragem os parâmetros do processo são identificados com base nas entradas u e saídas y passadas e os parâmetros do controlador são determinados, gerando desta forma o sinal de controle u (ÅSTRÖM; WITTENMARK, 1995).

2.3 Estruturas de controle

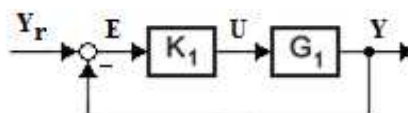
De acordo com as exigências do processo, são utilizadas diferentes estruturas de controle, entre elas (LUTZ; WENDT, 2000):

- Controle de sistema monovariável;
- Controle de sistema em cascata;
- Controle de sistema multivariável.

2.3.1 Controle de Sistemas Monovariáveis

Devido a sua simplicidade e a facilidade de implementação, o controle monovariável, também chamado controle de sistema SISO (*Single Input, Single Output*) é a estrutura empregada com maior frequência no controle de sistemas dinâmicos, embora não garanta o desempenho desejado na presença de perturbações (LUTZ; WENDT, 2000). A figura 2.3 mostra um exemplo de controle monovariável, onde K_1 é o controlador e G_1 a planta.

Figura 2.3: Controle monovariável.

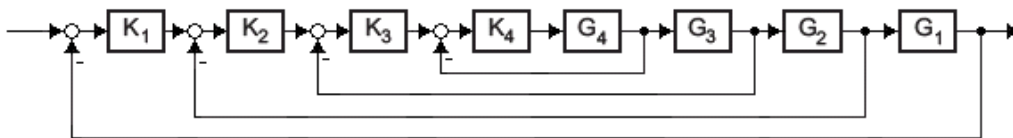


Fonte: Adaptado de Schumacher e Leonhard (2006).

2.3.2 Controle de sistemas em cascata

Um dos principais problemas que envolvem as estratégias de controle é a sensibilidade da variável controlada a perturbações. Para estruturas monovariáveis, a ação corretiva só começa depois que os efeitos aparecem na saída do sistema, o que leva a uma drástica redução do desempenho do controle. Para tornar a variável controlada menos sensível a estas perturbações, a utilização da estrutura em cascata em comparação aos controles de malha única é a estratégia que apresenta a melhor compensação (SCHUMACHER; LEONHARD, 2006). Sua malha de controle é constituída por malhas aninhadas (figura 2.4) e variáveis intermediárias que são utilizadas para responder mais rapidamente ao sinal de controle.

Figura 2.4: Controle de sistemas em cascata.



Fonte: Adaptado de Schumacher e Leonhard (2006).

É utilizada extensivamente em muitos processos químicos, tais como reatores químicos, colunas de destilação, aquecedores e trocadores de calor (LEE et al., 1998).

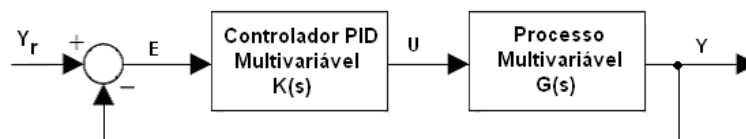
Conforme Åström e Hägglund (1995), Lutz e Wendt (2000) e Schumacher e Leonhard (2006) as principais características da estrutura em cascata são:

- Estrutura transparente, empregando todas as variáveis de estado mensuráveis;
- Projeto passo a passo, começando com o laço de controle mais interno, resolvendo desse modo o problema da estabilidade nas etapas inferiores;
- O valor máximo de cada variável de realimentação pode ser limitado pelo ajuste da referência pertinente;
- Para a escolha das variáveis intermediárias alguns critérios são levados em consideração como: deve existir uma relação bem definida entre a variável de controle e as variáveis intermediárias; a malha interna deve ser mais rápida que a malha externa; deve ser possível ter um ganho na malha interna.

2.3.3 Sistemas Multivariáveis

Além da utilização dos controladores PID em sistemas monovariáveis, tem-se observado nos últimos anos o aumento do interesse da indústria na utilização de técnicas de controle multivariável ou controle de sistemas MIMO (*Multiple Input, Multiple Output*) (TORRES, 2002). A estrutura do controlador PID multivariável é considerada na figura abaixo:

Figura 2.5: Controle PID Multivariável.



Fonte: Adaptado de Johnson e Moradi (2005).

Pode-se representar um sistema linear multivariável através de uma matriz de transferência como mostrado abaixo:

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s), \quad (2.14)$$

onde $\mathbf{Y}(s)$ e $\mathbf{U}(s)$ são vetores com n componentes representando respectivamente a Transformada de Laplace das saídas e entradas do processo e $\mathbf{G}(s)$ é a matriz de transferência do processo. Assim, a descrição entrada-saída de um processo $n \times n$ é dada por:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ \vdots \\ y_n(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) & g_{13}(s) & \cdots & g_{1n}(s) \\ g_{21}(s) & g_{22}(s) & g_{23}(s) & \cdots & g_{2n}(s) \\ \vdots & \vdots & & \ddots & \\ g_{n1}(s) & g_{n2}(s) & g_{n3}(s) & \cdots & g_{nm}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_n(s) \end{bmatrix}, \quad (2.15)$$

onde cada elemento $g_{mn}(s)$ é a função de transferência que relaciona a entrada 'n' com a saída 'n'.

Assume-se um controlador PID multivariável $n \times n$ possuindo a seguinte representação:

$$\mathbf{K}(s) = \begin{bmatrix} K_{11}(s) & \cdots & K_{1n}(s) \\ \vdots & \ddots & \vdots \\ K_{n1}(s) & \cdots & K_{nn}(s) \end{bmatrix}. \quad (2.16)$$

A forma do controlador $K_{ij}(s)$, para $i, j = \{1, 2, \dots, n\}$ é dado por:

$$K_{ij}(s) = K_{p_{ij}} \left(I + \frac{I}{T_{i_{ij}} s} + T_{d_{ij}} s \right). \quad (2.17)$$

O vetor de saída desejado é representado por:

$$\mathbf{Y}_r = [y_{r1}, y_{r2}, \dots, y_{rn}]^T. \quad (2.18)$$

O vetor de saída atual é representado por:

$$\mathbf{Y} = [y_1, y_2, \dots, y_n]^T. \quad (2.19)$$

O vetor erro é descrito como o vetor desejado menos o vetor de saída atual, dado por:

$$\mathbf{E} = \mathbf{Y}_r - \mathbf{Y} = [y_{r1} - y_1, y_{r2} - y_2, \dots, y_{rn} - y_n]^T, \quad (2.20)$$

podendo ser representado ainda da seguinte forma:

$$\mathbf{E} = [e_1, e_2, \dots, e_n]^T. \quad (2.21)$$

Um dos pontos de maior preocupação é a interação entre as malhas, ou seja, uma variável controlada sofre efeitos de várias outras variáveis manipuladas, dificultando a sintonia dos controladores. Bristol (1966) desenvolveu uma metodologia, a Matriz de Ganho Relativo (RGA - *Relative Gain Array*), para medir o grau de interação entre as variáveis do processo e recomendar os melhores pares de variáveis controladas e manipuladas. Esta análise leva em consideração apenas o comportamento das variáveis no estado estacionário.

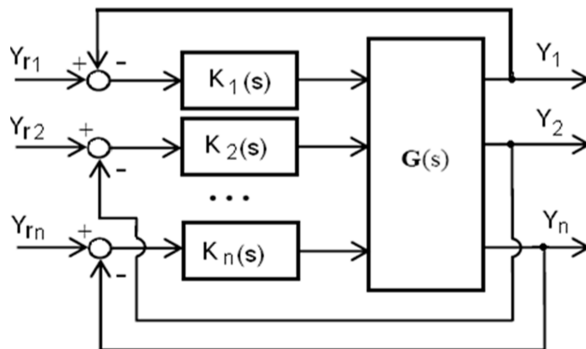
Existem diversos métodos para o projeto de controladores para sistemas multivariáveis. Neste trabalho são apresentadas duas formas de projetos: os controladores PID multivariáveis descentralizados e os controladores PID multivariáveis com desacopladores. Estas estruturas são as mais comuns em aplicações de controle na indústria (CAMPESTRINI, 2006 e HU et al., 2011) o que conduz ao interesse do trabalho proposto.

2.3.3.1 Controladores PID Multivariáveis descentralizados

Um dos esquemas de controle mais comuns e que são frequentemente encontrados em indústrias químicas e de processo são os controladores PID descentralizados (também conhecidos como controladores PID multi-malhas).

Possuem uma estrutura simples e desempenho satisfatório. A figura 2.6 apresenta o diagrama em blocos de um controle PID descentralizado.

Figura 2.6: Controle PID Multivariável descentralizado.



Fonte: Adaptado de Vu e Lee (2008).

São utilizados múltiplos controladores, um para cada variável a ser controlada, gerando desta forma, várias malhas de controle independentes, o que é mais fácil de implementar na prática (LUYBEN, 1990).

A matriz de controle é mostrada abaixo:

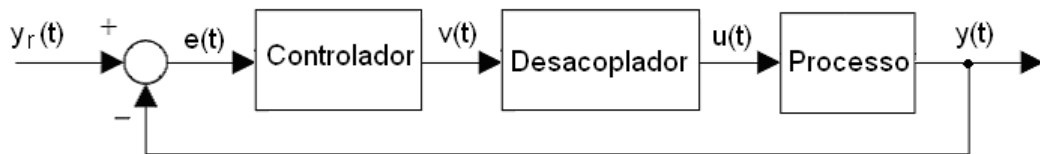
$$\mathbf{K}(s) = \begin{bmatrix} K_1(s) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & K_n(s) \end{bmatrix}. \quad (2.22)$$

Em alguns casos, devido às interações entre as malhas, sua sintonia pode ser difícil, o que pode limitar o desempenho do sistema (ALBERTOS; SALA, 2004).

2.3.3.2 Controladores PID Multivariáveis com desacopladores

Os desacopladores têm como objetivo eliminar ou reduzir a influência entre as variáveis em um sistema MIMO. O seu uso pode trazer benefícios interessantes, pois sem as interações entre as malhas, permite utilizar técnicas de controle monovariável que geralmente são mais simples. A figura 2.7 apresenta o diagrama em blocos de um sistema com controlador desacoplador.

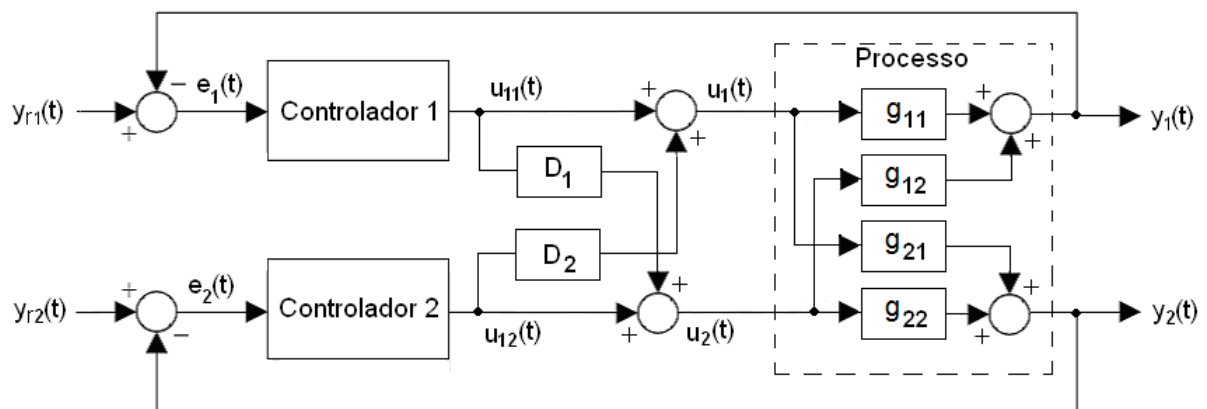
Figura 2.7: Sistema multivariável com controlador desacoplador.



Fonte: Adaptado de Campestrini (2006).

Há diversas formas de se projetar desacopladores. O projeto generalizado de desacopladores pode ser visto em Ogunnaike e Ray (1994). A figura 2.8 apresenta um exemplo de um sistema de controle desacoplado para um processo 2 X 2 (2 entradas e 2 saídas).

Figura 2.8: Controle PID Multivariável desacoplado 2 x 2.



Fonte: Adaptado de Seborg et al. (1989).

São utilizados quatro controladores, dois convencionais (controladores 1 e 2) e dois desacopladores (D_1 e D_2). O controlador 1 manipula a variável $y_1(t)$ e o controlador 2 manipula a variável $y_2(t)$.

As duas variáveis controláveis $y_1(t)$ e $y_2(t)$ e as duas variáveis manipuladas $u_1(t)$ e $u_2(t)$ estão relacionadas por quatro funções de transferência, g_{11} , g_{12} , e assim por diante. A matriz de transferência do processo é mostrada abaixo:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \quad (2.23)$$

Com os desacopladores, as ações de controle são mostradas na equação abaixo:

$$u_1(s) = u_{11}(s) + D_2(s)u_{12}(s), \quad (2.24)$$

$$u_2(s) = u_{12}(s) + D_1(s)u_{11}(s). \quad (2.25)$$

Substituindo (2.24) e (2.25) em (2.23), obtêm-se:

$$y_1(s) = [g_{11}(s) + D_1(s)g_{12}(s)]u_{11}(s) + [D_2(s)g_{11}(s) + g_{12}(s)]u_{12}(s), \quad (2.26)$$

$$y_2(s) = [g_{21}(s) + D_1(s)g_{22}(s)]u_{11}(s) + [D_2(s)g_{21}(s) + g_{22}(s)]u_{12}(s). \quad (2.27)$$

O desacoplador $D_1(s)$ deve ser projetado para cancelar, ou atenuar, o efeito da interação entre $u_1(s)$ e $y_2(s)$ representada pela função de transferência $g_{21}(s)$, e o desacoplador $D_2(s)$ deve ser projetado para cancelar, ou atenuar, o efeito da interação entre $u_2(s)$ e $y_1(s)$ representada pela função de transferência $g_{12}(s)$. Desta forma, obtêm-se:

$$D_2(s)g_{11}(s) + g_{12}(s) = 0, \quad (2.28)$$

$$g_{21}(s) + D_1(s)g_{22}(s) = 0. \quad (2.29)$$

As equações que permitem projetar o desacoplador ideal para um sistema TITO (*Two Inputs, Two Outputs*) são mostradas abaixo (SEBORG et al., 1989):

$$D_1(s) = -\frac{g_{21}(s)}{g_{22}(s)}, \quad (2.30)$$

$$D_2(s) = -\frac{g_{12}(s)}{g_{11}(s)}. \quad (2.31)$$

O desacoplamento pode ser do tipo dinâmico, que elimina todas as interações entre as malhas, parcial que elimina apenas as interações mais fortes entre as malhas ou estático que elimina apenas as interações em regime permanente.

Em algumas situações, o uso de desacopladores estáticos pode proporcionar um melhor controle do que o desacoplador dinâmico, pois não depende fortemente da exatidão do modelo. É um projeto menos ambicioso do que o desacoplador dinâmico, de modo que este desacoplador irá eliminar ou atenuar as interações em estado estacionário entre as malhas (SEBORG et al., 1989).

As equações de projeto para os desacopladores estáticos podem ser obtidas a partir das equações (2.30) e (2.31) fazendo-se $s = 0$ resultando nas equações (2.32) e (2.33) que são os ganhos em regime permanente (SEBORG et al., 1989):

$$D_1 = -\frac{K_{p21}}{K_{p22}}, \quad (2.32)$$

$$D_2 = -\frac{K_{p12}}{K_{p11}}, \quad (2.33)$$

onde K_{p11} , K_{p12} , K_{p21} e K_{p22} são respectivamente os ganhos regime permanente das funções de transferência g_{11} , g_{12} , g_{11} e g_{12} .

2.4 Técnicas convencionais de sintonia

Há inúmeros métodos de sintonia de controladores PID com grande aceitação em processos industriais, como: Ziegler e Nichols (1942), Cohen e Coon (1953) e Chien, Hrones e Reswick (1952), método da Integral do Erro (LÓPEZ et al., 1967), método *Biggest Log-Modulus Tuning* (LUYBEN, 1986). Embora o desenvolvimento de novas técnicas tenha se ampliado e ganhado aceitação na indústria, os métodos tradicionais ainda constituem marcos referenciais (CARMO, 2006).

Os métodos utilizados para fins de comparação com as técnicas metaheurísticas são o método de Ziegler e Nichols (1942) para sistemas monovariáveis, o método de Lee et al. (1998) para estruturas em cascata e o método BLT (LUYBEN, 1986) para sistemas multivariáveis. A seguir são expostos estes três métodos.

2.4.1 Ziegler-Nichols (ZN)

Ziegler e Nichols propuseram em 1942 dois métodos para determinar os parâmetros de controladores PID, a partir de um processo SISO. Estes métodos ainda são largamente utilizados na sua forma original ou com algumas modificações.

O primeiro método, conhecido como método da curva de reação, a resposta do sistema em malha aberta a uma entrada degrau unitário (figura 2.9).

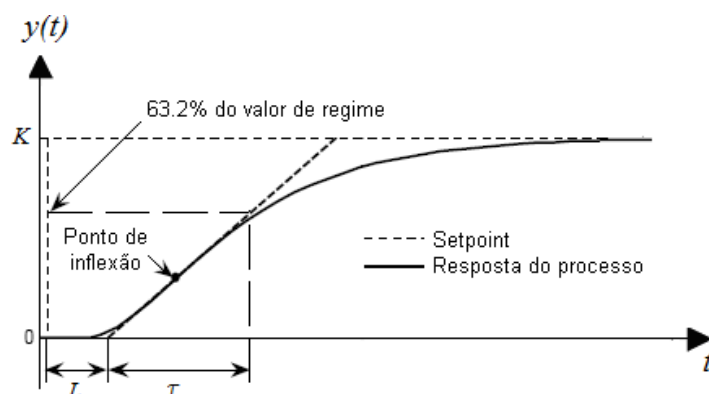
Figura 2.9: Método da Curva de reação.



Fonte: Adaptado de Ogata (1997).

Se o sistema produzir uma resposta como mostrado pela figura abaixo (em forma de S), o tempo morto (ou *delay time*) L e a constante de tempo τ podem ser determinados traçando-se uma reta tangente à curva no ponto de inflexão e determinando-se as interseções com o eixo dos tempos e com a reta $y(t) = K$. A figura 2.10 apresenta um exemplo da curva apresentada.

Figura 2.10: Resposta do sistema - método curva de reação.



Fonte: Adaptado de Ogata (1997).

A tabela 2.1 apresenta a sintonia dos controladores a partir dos valores L e τ .

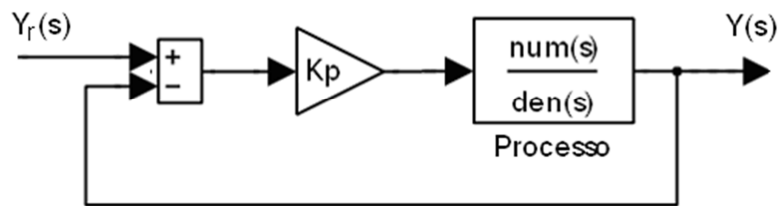
Tabela 2.1: Sintonia do controlador através do primeiro método de Ziegler-Nichols.

Controlador	K_p	T_i	T_d
P	$\frac{\tau}{L}$	-	-
PI	$0,9 \frac{\tau}{L}$	$\frac{L}{0,3}$	-
PID	$1,2 \frac{\tau}{L}$	$2L$	$0,5L$

Fonte: Adaptado de Ogata (1997).

O segundo método proposto por Ziegler e Nichols, também conhecido como método da sensibilidade limite, ou método da oscilação crítica, é baseado na utilização de um controlador proporcional ($T_i = \infty$ e $T_d = 0$) em malha fechada (figura 2.11), variando-se seu ganho até que o sistema apresente uma resposta oscilatória com amplitude constante (figura 2.12).

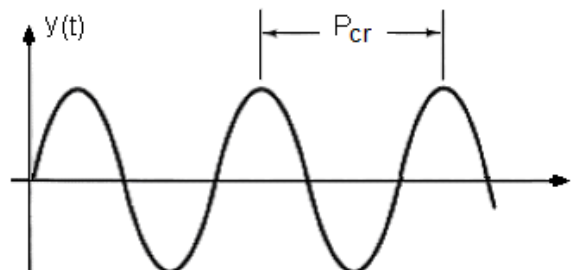
Figura 2.11: Método da Sensibilidade Limite.



Fonte: Adaptado de Ogata (1997).

Este ganho é conhecido como ganho crítico (K_{cr}) e o período de oscilação é o período crítico (P_{cr}).

Figura 2.12: Resposta do sistema - método da Sensibilidade Limite.



Fonte: Adaptado de Ogata (1997).

A tabela 2.2 apresenta a sintonia dos controladores a partir dos valores K_{cr} e P_{cr} .

Tabela 2.2: Sintonia do controlador através do segundo método de Ziegler-Nichols.

Controlador	K_p	T_i	T_d
P	$0,5 K_{cr}$	-	-
PI	$0,45 K_{cr}$	$P_{cr}/1,2$	-
PID	$0,6 K_{cr}$	$K_{cr}/2$	$K_{cr}/8$

Fonte: Adaptado de Ogata (1997).

Existem outras fórmulas baseadas nas fórmulas originais de Ziegler-Nichols, entre elas as de Tyreus-Luyben (LUYBEN, 1986), Através do ganho K_{cr} e do período P_{cr} críticos pode-se determinar os parâmetros do controlador (tabela 2.3). Tem como principais características um desempenho mais lento do sistema e *overshoot* menor.

Tabela 2.3: Sintonia do controlador através do método de Tyreus-Luyben.

Controlador	K_p	T_i	T_d
PI	$K_{cr}/3,2$	$2,2 P_{cr}$	-
PID	$K_{cr}/2,2$	$2,2 P_{cr}$	$K_{cr}/6,3$

Fonte: Adaptado de Luyben (1986).

2.4.2 Método de sintonia através de séries de Maclaurin

Lee et al. (1998) propõem um método para a sintonia de controladores PID para controle de sistemas em cascata. Esta metodologia permite sintonizar os parâmetros dos controladores PID, tanto da malha interna quanto da malha externa, simultaneamente. A regra de ajuste é baseada no modelo de processo e na resposta da malha fechada, ou seja, consiste em encontrar o controlador ideal que é obtido através da resposta em malha fechada e após a aproximação do controlador PID ideal através de séries de Maclaurin. São apresentadas simulações utilizando como modelos sistemas de primeira ordem com tempo morto (FOPDT) e sistemas de segunda ordem com tempo morto (SOPDT), que são os modelos aproximados mais comumente utilizados para processos químicos (LEE et al., 1998). Suas respectivas funções de transferências são apresentadas abaixo:

$$G = \frac{Ke^{-\theta s}}{\tau s + 1}, \quad (2.34)$$

$$G = \frac{Ke^{-\theta s}}{(\tau^2 s^2 + 2\xi\tau s + 1)}, \quad (2.35)$$

onde ξ é a constante de amortecimento, K é o ganho estacionário do processo, τ é a constante de tempo do modelo do processo, θ é o tempo morto.

A tabela 2.4 apresenta a sintonia dos controladores utilizando o método de Lee et al. (1998).

Na metodologia proposta, os parâmetros λ_1 e λ_2 são utilizados para ajustar a velocidade de resposta em malha fechada, e são escolhidos de modo a proporcionar um bom desempenho e robustez.

Tabela 2.4: Regras de sintonia para modelos FOPDT e SOPDT.

Processo	Modelo do processo	K_p	T_i	T_d
FOPDT (malha interna)	$G_2 = \frac{K_2 e^{-\theta_2 s}}{\tau_2 s + 1}$	$\frac{T_i}{k_2(\lambda_2 + \theta_2)}$	$\tau_2 + \frac{\theta_2^2}{2(\lambda_2 + \theta_2)}$	$\frac{\theta_2^2}{6(\lambda_2 + \theta_2)} \left[3 - \frac{\theta_2}{T_i} \right]$
FOPDT (malha externa)	$G_1 = \frac{K_1 e^{-\theta_1 s}}{\tau_1 s + 1}$	$\frac{T_i}{k_1(\lambda_1 + \theta_1 + \theta_2)}$	$\tau_1 + \lambda_2 + \frac{(\theta_1 + \theta_2)^2}{2(\lambda_1 + \theta_1 + \theta_2)}$	$\frac{\lambda_2 T_i - \frac{(\theta_1 + \theta_2)^3}{6(\lambda_1 + \theta_1 + \theta_2)}}{T_i} + \frac{(\theta_1 + \theta_2)^2}{2(\lambda_1 + \theta_1 + \theta_2)}$
SOPDT (malha interna)	$G_2 = \frac{K_2 e^{-\theta_2 s}}{(\tau^2 s^2 + 2\xi\tau s + 1)}$	$\frac{T_i}{k_2(\lambda_2 + \theta_2)}$	$2\tau\xi + \frac{\theta_2^2}{2(\lambda_2 + \theta_2)}$	$\frac{\tau^2 - \frac{\theta_2^3}{6(\lambda_2 + \theta_2)}}{T_i} + \frac{\theta_2^2}{2(\lambda_2 + \theta_2)}$
SOPDT (malha externa)	$G_1 = \frac{K_1 e^{-\theta_1 s}}{(\tau^2 s^2 + 2\xi\tau s + 1)}$	$\frac{T_i}{k_1(\lambda_1 + \theta_1 + \theta_2)}$	$2\tau\xi + \lambda_2 + \frac{(\theta_1 + \theta_2)^2}{2(\lambda_1 + \theta_1 + \theta_2)}$	$\frac{\tau^2 + 2\tau\xi\lambda_2 - \frac{(\theta_1 + \theta_2)^3}{6(\lambda_1 + \theta_1 + \theta_2)}}{T_i} + \frac{(\theta_1 + \theta_2)^2}{2(\lambda_1 + \theta_1 + \theta_2)}$

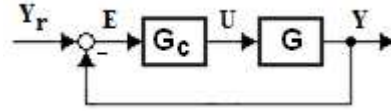
Fonte: Adaptado de Lee et al. (1998).

2.4.3 Método BLT (*Biggest Log-Modulus Tuning*)

O método BLT proposto por Luyben (1986) é uma analogia direta ao critério de estabilidade de Nyquist para sistemas SISO. O critério de Nyquist multivariável fornece uma resposta do tipo SIM/NÃO à questão de estabilidade do sistema. Ele estabelece uma condição necessária e suficiente de estabilidade (CRUZ, 1996).

Segundo o diagrama em blocos do sistema de controle apresentado na figura 2.13:

Figura 2.13: Diagrama em blocos de um sistema de controle multivariável - método BLT



Fonte: Elaborado pelo autor.

onde G é a matriz de transferência do processo e G_c é a matriz diagonal de transferência do controlador, o primeiro passo do método BLT consiste em calcular, de forma descentralizada, os parâmetros dos controladores através do método de Ziegler-Nichols, ou seja, o ganho K_{cr} e o período P_{cr} de cada elemento diagonal da matriz função de transferência $g_{ij}(s)$ são calculados na forma clássica SISO. Após, é especificado um fator F (típico variar de 2 a 5) que é aplicado a todas as malhas (LUYBEN, 1986).

Os parâmetros $K_{p j, BLT}$ (ganho) e $T_{i j, BLT}$ (constante do integrador) de cada controlador são calculados utilizando o fator F como um parâmetro de dessintonização conforme as seguintes equações:

$$K_{p j, BLT} = \frac{K_{p j, ZN}}{F}, \quad (2.36)$$

$$T_{i j, BLT} = F T_{i j, ZN}, \quad (2.37)$$

onde $j \in \{1, 2, \dots, n\}$, $K_{p j, ZN}$ e $T_{i j, ZN}$ são os parâmetros do controlador PI determinados pelo método de Ziegler e Nichols.

Em seguida, as interações são levadas em consideração pela dissonância de cada controlador e é traçado o gráfico $W(i\omega)$ de Nyquist segundo a equação:

$$W(i\omega) = -I + \det(I + GG_c), \quad (2.38)$$

para frequências $0 \leq \omega \leq \infty$,

Após, é calculada a função $L(i\omega)$:

$$L(i\omega) = 20 \log \left| \frac{W(i\omega)}{1 + W(i\omega)} \right|. \quad (2.39)$$

O valor de F é variado até que o pico $L(i\omega)$ seja igual a $2N$ (LUYBEN, 1986), isto é:

$$L(i\omega)^{max} = \max_{\omega} L(i\omega) = 2N, \quad (2.40)$$

onde N é a dimensão do sistema multivariável e $L(i\omega)^{max}$ é chamado de “*Biggest Log-Modulus*”.

A atratividade deste método é devido à sua simplicidade de implementação e compreensão. No entanto, uma desvantagem é que as configurações do controlador são realizadas de forma mais conservadora, pois quanto maior o valor de F , mais estável será o sistema, mas também será mais lento (VU; LEE, 2008), dando um compromisso razoável entre a estabilidade e o desempenho em sistemas multivariáveis (LUYBEN, 1986).

Outro método de projeto é o *Sequential Loop Closing* (SLC) introduzido por Mayne (1973) e mais tarde estudado por Hovd e Skogestad (1994). Neste método, os controladores são sintonizados sequencialmente, onde as malhas são ajustadas individualmente, fechando uma após a outra de modo que a interação causada pelo fechamento do anterior é contabilizada durante o ajuste da malha corrente.

Loh et al. (1993) apresentaram um método de auto-ajuste para controlares PID multi-malhas usando uma combinação do método SLC e auto-sintonia por relé, onde o sistema de controle multi-malhas também é sintonizado sequencialmente malha a malha. Ele requer muito pouco conhecimento do processo e tem como vantagem a garantia da estabilidade em malha fechada. Este método é conhecido como SAT (*Sequential Autotuning*).

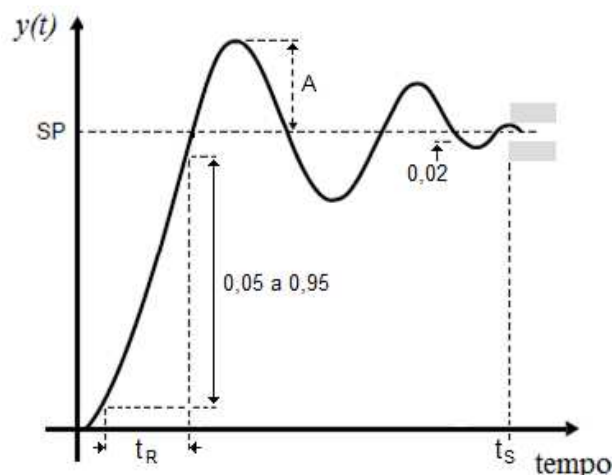
2.5 Critérios de desempenho

Segundo Lutz e Wendt (2000), Schumacher e Leonhard (2006) e Carvalho (2000), critérios de desempenho dos sistemas de controle são estabelecidos para avaliar e comparar a capacidade dos diferentes tipos de controladores de manter a variável controlada próximo ao *setpoint*, mesmo na presença de perturbações externas.

Estas especificações de desempenho estão relacionadas às respostas transitórias e em regime permanente da variável de saída do sistema. A figura 2.14 apresenta um exemplo de resposta de um sistema subamortecido com a aplicação de um degrau no *setpoint*. Pode-se destacar:

- **Tempo de subida (t_R):** tempo necessário para a resposta passar de 5% a 95% ou 0% a 100% do seu valor final (valor de *setpoint* - SP) ;
- **Tempo de acomodação (t_s):** tempo necessário para que a diferença entre a resposta e a referência seja menor que uma faixa pré-determinada (2% a 5%);
- **Sobressalto ou overshoot (OS):** valor máximo de pico que a resposta atinge em relação à referência. Pode ser dado em função da razão A/SP (figura 2.14). Não há um valor limite absoluto, depende de cada objetivo do laço de controle; em alguns sistemas não se deseja nenhum sobressalto, mas uma limitação comumente empregada é $OS < 20\%$ (BAZANELLA; GOMES DA SILVA, 2005).

Figura 2.14: Resposta de um sistema subamortecido com a aplicação de um degrau no *setpoint*.



A avaliação do desempenho de sistemas de controle em malha fechada também pode ser feita através de índices calculados por integração de uma função do erro ao longo do tempo quando perturbações na carga e no *setpoint* da variável controlada são aplicadas. A seguir serão expostos alguns índices de erro.

2.5.1 Integral Absoluto do Erro (IAE)

Conforme Ogata (1997) é o índice de mais fácil aplicação. É matematicamente definido como:

$$IAE = \int_0^{t_N} |e(t)| dt, \quad (2.41)$$

onde $e(t)$ é o erro dado pela diferença entre a referência (*setpoint*) e o valor da saída do processo e t_N é um tempo escolhido de modo que a integral tenda a um valor estacionário.

2.5.2 Integral do Erro Quadrático (ISE)

O índice ISE fornece grande peso aos erros maiores e pequeno peso aos erros menores (OGATA, 1997 e DORF; BISHOP, 1998). É definido matematicamente como:

$$ISE = \int_0^{t_N} e^2(t) dt. \quad (2.42)$$

2.5.3 Integral do Erro Absoluto ponderado pelo Tempo (ITAE)

O índice ITAE pondera um grande erro inicial com pequeno peso enquanto que os erros gerados na resposta transitória têm maiores pesos (OGATA, 1997 e DORF; BISHOP, 1998). Matematicamente o ITAE é assim definido:

$$ITAE = \int_0^{t_N} t |e(t)| dt. \quad (2.43)$$

2.5.4 Integral do Erro Quadrático ponderado pelo Tempo (ITSE)

O índice ITSE possibilita também uma penalização mais severa para os erros de estado estacionário gerados pela resposta ao degrau. É descrito pela equação (2.44):

$$ITSE = \int_0^{t_N} t e^2(t) dt. \quad (2.44)$$

2.5.5 Esforço de controle

Esta medida representa o esforço sobre os elementos atuadores (DORF; BISHOP, 1998). É definida matematicamente pela equação (2.45):

$$Esforço = \int_0^{t_N} u(t) dt. \quad (2.45)$$

2.6 Robustez dos sistemas de controle

Pode-se ainda avaliar a qualidade da resposta do sistema através da robustez, que é a capacidade de assegurar o desempenho desejado na presença de variações dos parâmetros do sistema. Entre os critérios, pode-se destacar:

- **Margem de Ganho (MG):** medida de quanto o sistema suportaria para uma mudança de ganho estático. O valor limite recomendado é $MG > 1,7$ (TRIERWEILER; POSSER, 2002).
- **Margem de Fase (MF):** margem para a fase cair abaixo de 180° . O limite recomendado é $MF > 30^\circ$ (BOLTON, 1995).

Em resumo, pode-se dizer que a boa sintonia dos controladores PID é sempre um compromisso entre a estabilidade/robustez e a velocidade de resposta/desempenho da malha de controle (LUYBEN, 1986).

2.7 Comentários finais

Neste capítulo foram descritas as características dos controladores PID, técnicas de sintonia e algumas de suas estruturas (monovariável, cascata, multivariável com desacopladores e multivariável descentralizado). Entretanto, os métodos convencionais de sintonia não apresentam bom desempenho quando as plantas apresentam alta ordem, atrasos de transporte elevados, características não-lineares, dentre outras (MAZZINI, 2003).

Neste sentido, a teoria de controle automático está cada vez mais integrada à área da computação. Novos algoritmos e recursos computacionais têm sido utilizados a fim de realizar os mais diversos tipos de controle (DORF; BISHOP, 1998 apud MOROMISATO et al., 2007). Muitas soluções propostas têm acrescentado algum tipo de inteligência artificial no projeto de controladores PID (MELOTTI et al., 2008). Uma das possibilidades é o emprego de métodos metaheurísticos para determinação dos parâmetros dos controladores PID. Nos próximos capítulos, será realizada uma revisão sobre otimização e sobre as metaheurísticas utilizadas para a sintonia dos controladores monovariáveis e multivariáveis neste trabalho.

3 OTIMIZAÇÃO

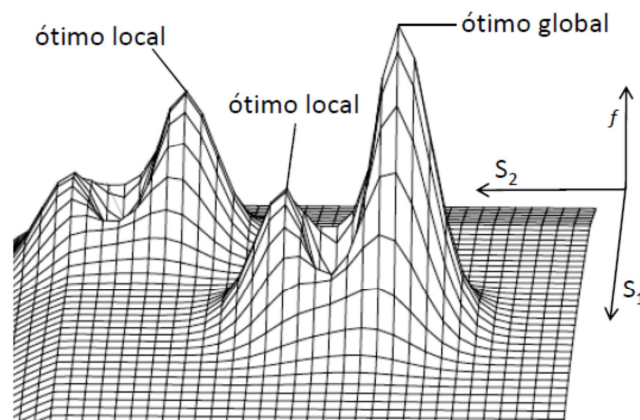
3.1 Definição e conceitos

Segundo Souza (2005), otimizar um sistema é projetá-lo com mais eficiência e menor custo, onde o objetivo é encontrar os parâmetros do sistema em estudo, de modo que o desempenho do sistema seja próximo do valor ótimo, baseado em um critério prévio de busca.

Castro (2001) descreve algumas entidades importantes referentes aos problemas de otimização:

- Variáveis de otimização: são parâmetros que se alteram durante o processo de otimização. Elas podem ser classificadas em contínuas ou discretas (binárias ou inteiras);
- Restrições: são funções de igualdade ou desigualdade que descrevem e caracterizam as situações indesejáveis de projeto;
- Espaço de busca: é o conjunto, espaço ou região que compreende as possíveis ou viáveis soluções do problema a ser otimizado, devendo ser caracterizado pelas funções de restrições;
- Função objetivo (FO) ou de avaliação: é a função que se quer minimizar ou maximizar. Ela pode ser classificada de duas formas: unidimensional, quando possui uma única variável e multidimensional quando possui mais de uma variável;
- Ponto ótimo: é o ponto pertencente ao espaço de busca e que é determinado pelas variáveis de projeto que extremizam a função objetivo;
- Valor ótimo: é o valor da função objetivo no ponto ótimo;
- Solução ótima: é o par formado pelo ponto ótimo e valor ótimo. Pode ser do tipo local, quando qualquer movimento a ser feito piore o valor atual da função objetivo ou global, que corresponde ao maior valor da função objetivo (no caso de uma maximização), entre todos os ótimos locais existentes no espaço de busca. A figura 3.1 ilustra um problema de maximização com uma função objetivo f definida sobre um espaço bidimensional $S = (S_1; S_2)$.

Figura 3.1: Representação de ótimos local e global em uma função bidimensional.



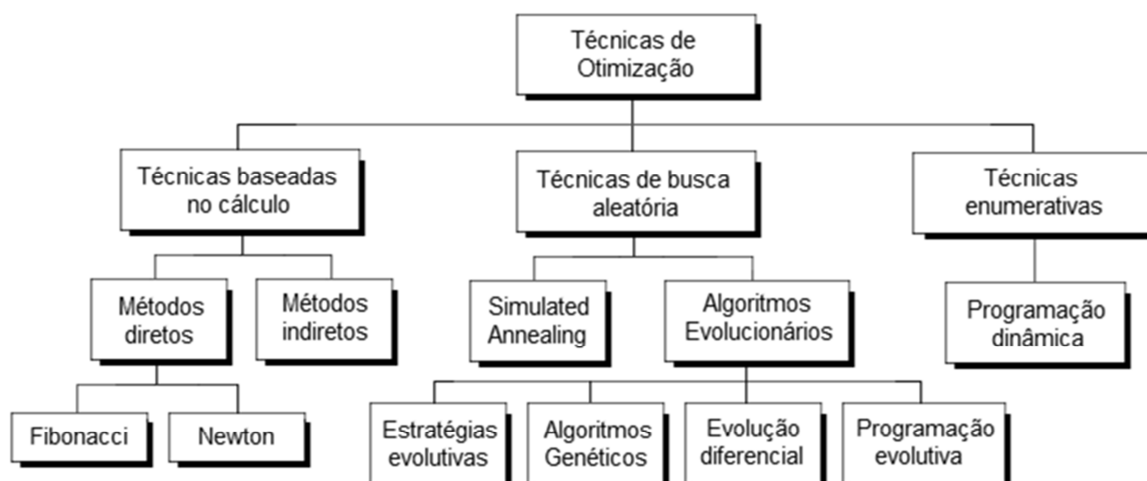
Fonte: Adaptado de Weise (2009).

Segundo Ribeiro Filho et al. (1994), os métodos de otimização podem ser classificados em três classes:

- Baseados em cálculos: utilizam um conjunto de condições necessárias e suficientes que devem ser satisfeitas pelas soluções de um problema de otimização;
- Enumerativos: procuram o ponto ótimo pesquisando sequencialmente cada ponto do espaço de busca (finito e discreto) (BARCELLOS, 2000);
- Dirigidos por pesquisa aleatória: são baseados em técnicas enumerativas, mas usam informações adicionais para dirigir a pesquisa. Podem-se citar os algoritmos evolucionários e o *Simulated Annealing*.

A figura 3.2 apresenta os grupos com as referidas técnicas de otimização.

Figura 3.2: Classes de técnicas de otimização.



Fonte: Adaptado de Ribeiro Filho et al. (1994).

3.2 Otimização Mono-objetivo

Um problema de otimização mono-objetivo apresenta apenas uma função objetivo. O algoritmo consiste basicamente em analisar soluções simultâneas ou sequenciais na região factível em busca da solução ótima. Esta função objetivo nos sistemas em estudo pode ser, por exemplo, o menor tempo de acomodação do processo.

Pode-se representar um problema típico de otimização mono-objetivo da seguinte forma (COELLO et al., 2007):

$$\min \text{ (ou max) } f(x) \quad (3.1)$$

$$\text{sujeito a } g_i(x) \leq b, \quad i = 1, \dots, m \quad (3.2)$$

$$h_j(x) = c, \quad j = 1, \dots, p \quad (3.3)$$

onde $f(x)$ é a função objetivo a ser minimizada (ou maximizada), $g_i(x)$ são as restrições de desigualdade, $h_j(x)$ são as restrições de igualdade e x é o vetor de variáveis de decisão.

3.3 Otimização Multiobjetivo

Um problema de otimização multiobjetivo (*Multi-Objective Optimization Problem*, MOOP), também chamado de otimização multicriterial, trabalha com mais de uma função objetivo e pode ser definido por (ABRAHAN et al., 2005 e DEB, 2001):

$$\min (\text{ou max}) z = \{f_1(x), f_2(x), \dots, f_n(x)\} \quad (3.4)$$

$$\text{sujeito a } g_i(x) \leq b, i = 1, \dots, m \quad , \quad (3.5)$$

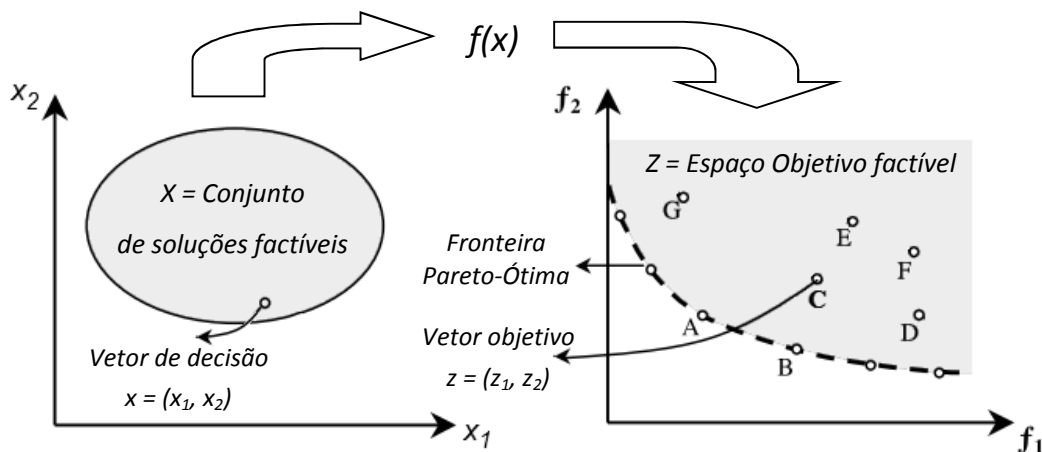
$$h_j(x) = c, j = 1, \dots, p \quad (3.6)$$

onde $g_i(x)$ são as restrições de desigualdade, $h_j(x)$ são as restrições de igualdade, x é o vetor de variáveis de decisão e z é o conjunto que contém os n objetivos ($f_1(x), f_2(x), \dots, f_n(x)$) a serem minimizados (ou maximizados), conhecido como espaço objetivo.

Desta forma, em um MOOP, para cada solução x no espaço de decisão, existe um $f(x)$ no espaço objetivo. Esta é uma diferença fundamental em relação à otimização mono-objetivo, cujo espaço de objetivos é unidimensional (TICONA, 2003).

A figura 3.3 apresenta o espaço de decisão e o espaço objetivo de um problema de minimização com duas variáveis.

Figura 3.3: Conjunto de soluções factíveis e espaço objetivo factível de um problema de minimização com duas variáveis.



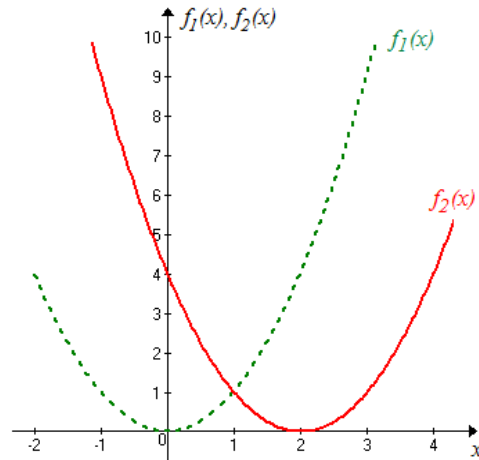
Fonte: Adaptado de Hashimoto (2004).

3.3.1 Frentes de Pareto

Num problema multiobjetivo, em geral, não existe uma única solução que seja ótima com relação a todos os objetivos, e sim um conjunto de soluções ótimas (ou não-dominadas) quando todos os objetivos são simultaneamente considerados (HASHIMOTO, 2004).

Um exemplo de conjunto ótimo de Pareto pode ser visto na figura 3.4. Trata-se de um simples problema de otimização multiobjetivo proposto por Schaffer (1984) apud Castro (2001). As funções $f_1(x) = x^2$ e $f_2(x) = (x - 2)^2$ necessitam ser simultaneamente minimizadas.

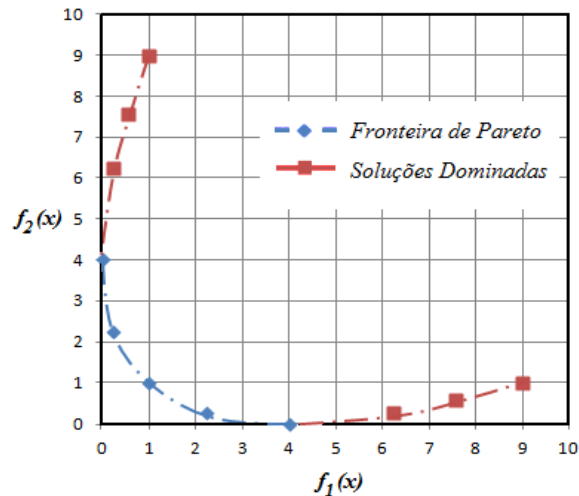
Figura 3.4: Funções $f_1(x)$ e $f_2(x)$ de um problema de otimização multiobjetivo.



Fonte: Adaptado de Schaffer (1984) apud Castro (2001).

As funções objetivo $f_1(x)$ e $f_2(x)$ têm seus mínimos em $x=0$ e $x=2$, respectivamente. No intervalo $[0, 2]$ qualquer solução é igualmente ideal, ou seja, tem-se um conjunto de soluções ideais. O espaço objetivo deste problema de minimização com duas variáveis é apresentado na figura 3.5.

Figura 3.5: Espaço objetivo do problema multiobjetivo.



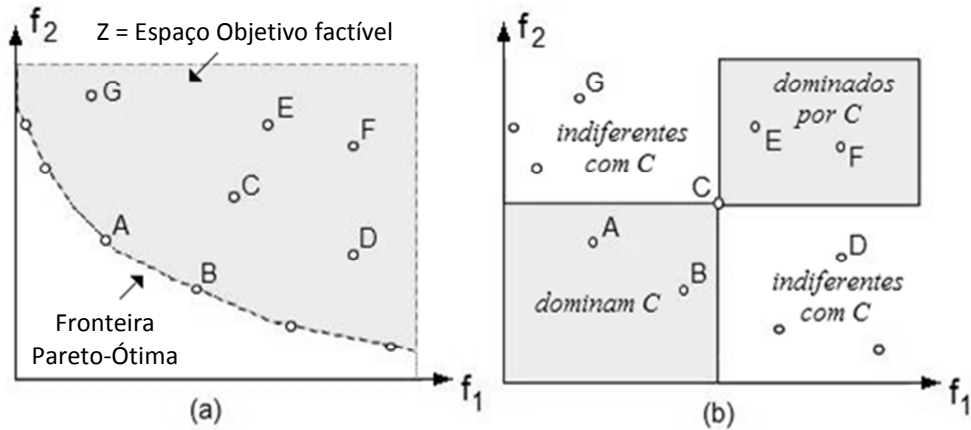
Fonte: Adaptado de Schaffer (1984) apud Castro (2001).

Este conjunto também é denominado de conjunto de Pareto, frente (fronteira) de Pareto, ou fronteira Pareto-Ótima. As funções são avaliadas através da Dominância de Pareto (DEB, 2001). Dados dois vetores de decisão $x, y \in X$, existem três possibilidades para seus correspondentes vetores objetivos:

- o vetor objetivo $z_1 = f(x)$ domina o vetor objetivo $z_2 = f(y)$ e o vetor de decisão x domina o vetor de decisão y se $f(x) \leq f(y)$;
- o vetor objetivo $z_2 = f(y)$ é dominado pelo vetor objetivo $z_1 = f(x)$ e o vetor de decisão y é dominado pelo vetor de decisão x se $f(y) \leq f(x)$;
- $f(x)$ e $f(y)$ são indiferentes e o vetor de decisão x é indiferente com y e vice-versa se z_1 não domina z_2 nem z_2 domina z_1 .

A figura 3.6(a) mostra a fronteira Pareto-Ótima e sua vizinhança de um problema de minimização e 3.6(b) apresenta os vetores objetivos, onde os pontos A e B (pertencentes ao retângulo esquerdo) dominam o ponto C, os pontos E e F (retângulo superior direito) são dominados por C e os pontos D e G são indiferentes com C.

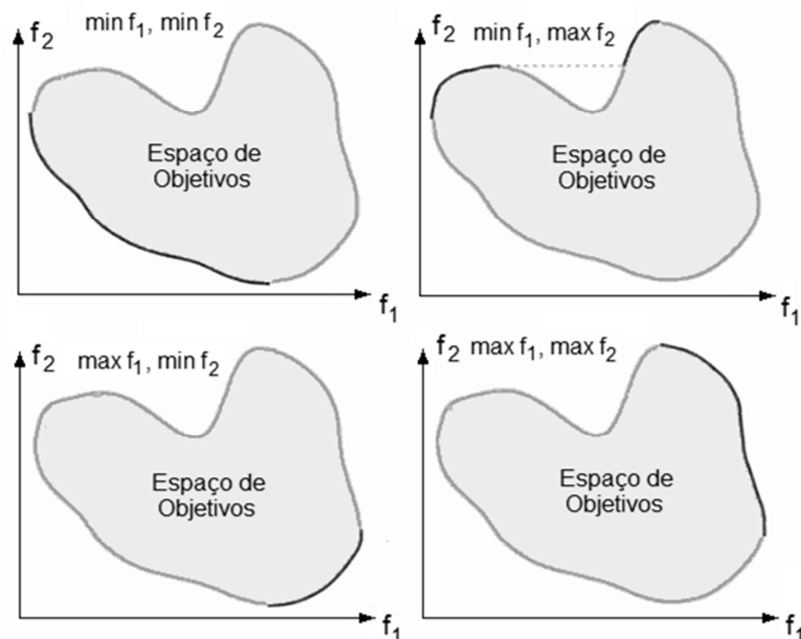
Figura 3.6. Dominância de Pareto no espaço objetivo.



Fonte: Adaptado de Arroyo (2002).

As frentes de Pareto podem se diferenciar quanto à convexidade, onde as funções mais comuns são côncavas ou convexas e quanto à continuidade, podendo ser formado por uma região contínua ou pela união de regiões descontínuas. A figura 3.7 apresenta alguns exemplos de conjuntos Pareto-Ótimos de problemas de otimização com duas funções f_1 e f_2 em combinações de maximização e minimização (na ordem esquerda-direita e cima-baixo, minimizar/minimizar, minimizar/maximizar, maximizar/minimizar e maximizar/maximizar.). A fronteira de Pareto, ilustrada em cada figura por uma linha em negrito, pode ser formada por regiões contínuas ou pela união de regiões descontínuas.

Figura 3.7: Exemplos de conjuntos Pareto-Ótimos.



Fonte: Adaptado de Deb (2001).

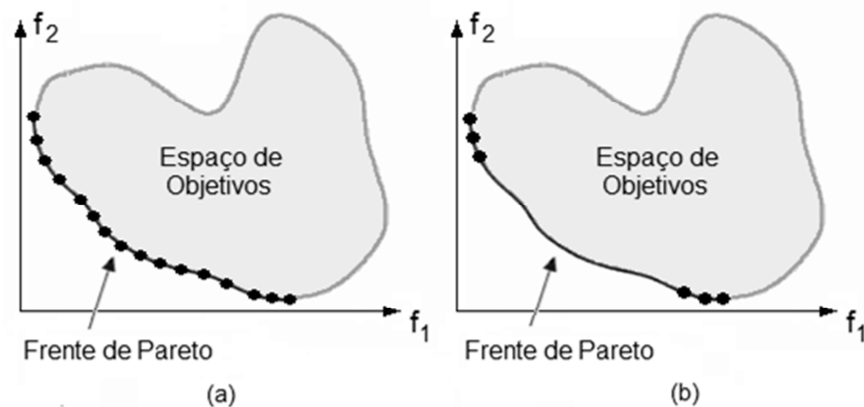
3.3.2 Metas em Otimização Multiobjetivo

Segundo Deb (2001), o processo de otimização multiobjetivo procura encontrar um conjunto de soluções:

- O mais próximo possível da Frente de Pareto;
- Com maior diversidade possível dentro das possíveis soluções.

A primeira meta a ser atingida é comum para qualquer problema de otimização. As soluções muito distantes da Frente de Pareto não são desejáveis. Já a segunda, é uma meta específica para problemas de otimização multiobjetivo. Uma boa diversidade estabelece um forte compromisso entre os objetivos desejados. A figura 3.8(a) apresenta um exemplo com uma boa diversidade e a figura 3.8(b) um exemplo com fraca diversidade.

Figura 3.8: Distribuição de soluções na Frente de Pareto.



Fonte: Adaptado de Deb (2001).

Deb (2001) apresenta duas abordagens para a solução de problemas de otimização multiobjetivo. A primeira abordagem é o método de agregação de funções que consiste em transformar os n objetivos em um único objetivo por meio de uma soma ponderada dos objetivos originais, através da atribuição de pesos para cada objetivo, como pode ser visto na equação (3.7):

$$\min(\text{ou max}) z = \sum_{i=1}^k W_i f_i(x), \quad (3.7)$$

onde z é a função objetivo, k é o número total de objetivos, f_i é cada função objetivo original e W_i é o peso atribuído ao objetivo i .

A segunda abordagem é através da aplicação da otimização de Pareto. O decisor escolhe a solução a partir dos resultados do conjunto Pareto-Ótimo.

3.4 Considerações finais

No contexto das otimizações mono-objetivo e multiobjetivo, as metaheurísticas são os métodos escolhidos para a sintonia dos controladores PID e são sucintamente descritas no capítulo seguinte.

4 METAHEURÍSTICAS

Muitos problemas de otimização combinatória são classificados como NP-Difíceis (*Non-deterministic polynomial-time hard*), ou seja, são problemas que não podem ser deterministicamente resolvidos em tempo computacional polinomial.

Neste contexto, os algoritmos para a resolução dos problemas podem ser classificados como exatos e aproximativos. Nemhauser e Wolsey (1999) descrevem que os métodos exatos (ou completos) garantem encontrar a solução ótima para um problema de otimização combinatória, mas necessitam, geralmente, de um tempo computacional exponencial. Já nos algoritmos aproximativos, onde se encaixam as metaheurísticas, sacrifica-se a garantia de encontrar a solução ótima, na busca por boas soluções em um tempo computacional viável (BLUM; ROLI, 2003).

Neste capítulo, são apresentadas as definições e propriedades das metaheurísticas, o estado da arte no contexto do trabalho e as revisões sobre as origens e características de cada metaheurística utilizada.

4.1 Definição e propriedades

Há muitas definições de algoritmos e métodos metaheurísticos. A palavra metaheurística tem sua origem de duas palavras gregas: *'heuriskein'* que significa encontrar, achar novas estratégias para resolver problemas e *'meta'* que significa metodologia de nível superior (BLUM; ROLI, 2003).

Em sua definição original, são métodos de soluções que orquestram uma interação entre os procedimentos de melhoria local e estratégias de alto nível para criar um processo capaz de escapar de ótimos locais e efetuar uma busca robusta em uma região viável (GLOVER; KOCHENBERGER, 2003).

Algumas propriedades fundamentais que caracterizam as metaheurísticas são descritas abaixo (BLUM; ROLI, 2003):

- Estratégias que “guiam” o processo de busca;
- Tem como objetivo explorar eficientemente o espaço de busca a fim de encontrar soluções ótimas ou próximas do ótimo;
- Variam desde procedimentos de busca local simples até processos de aprendizado complexos;
- São aproximativas e não-determinísticas;
- Podem incorporar mecanismos para evitar que fiquem presas em um ótimo local do espaço de busca;
- Não são específicas para um único tipo de problema;
- Podem utilizar a experiência obtida durante a busca (por meio de uma memória) para guiar a busca.

4.2 Algumas principais metaheurísticas

Dentre as metaheurísticas mais conhecidas e utilizadas, pode-se destacar:

- Otimização por colônia de formigas (Ant Colony Optimization - ACO): simula a inteligência de enxames baseada no comportamento natural de uma colônia de formigas, que se movimentam guiadas pelo feromônio depositado por outras formigas (DORIGO; STÜTZLE, 2004);
- Otimização por enxame de partículas (*Particle Swarm Optimization* - PSO): simula a inteligência de enxames baseada no comportamento de bandos (ou cardumes), que se movimentam na busca por alimento seguindo o pássaro (ou peixe) mais próximo do objetivo e, evitando a colisão entre os membros (KENNEDY; EBERHART, 1995);
- Busca Tabu: utiliza uma lista de movimentos proibidos que impede o retorno às últimas T soluções já geradas anteriormente e desta forma conduz a busca em direção ao ótimo (GLOVER; LAGUNA, 1997);
- *Simulated Annealing*: baseado em princípios da termodinâmica, especificamente na forma de como líquidos congelam e cristalizam ou metais resfriam. O material é inicialmente aquecido e então resfriado lentamente com o objetivo de produzir uma estrutura cristalina (KIRKPATRICK et al., 1983);
- Algoritmos Genéticos: inspirados nos conceitos provenientes da teoria da evolução natural de Darwin, ou seja, da sobrevivência dos mais aptos. Utilizam recombinações genéticas para efetuar a evolução das populações (GOLDBERG, 1989).

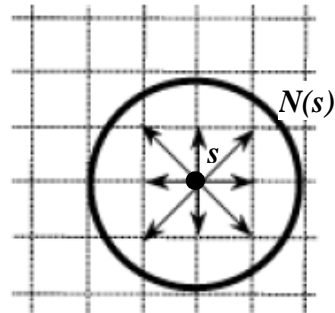
4.3 Classificação das metaheurísticas

Conforme Talbi (2009) e Blum e Roli (2003), muitos critérios podem ser usados para classificar as metaheurísticas. Um desses critérios é quanto a sua origem, ou seja, inspiradas ou não em processos naturais. Desta forma, pode-se diferenciar os algoritmos inspirados na natureza como os Algoritmos Evolucionários (inspirados na biologia), a Otimização por Nuvem de Partículas (Ciências Sociais) e o *Simulated Annealing* (física) daqueles não inspirados na natureza, como a metaheurística Busca Tabu.

Ainda de acordo com Talbi (2009), tem-se outro critério de classificação que é o número de soluções usadas ao mesmo tempo: solução única e população.

A primeira representa um grupo de metaheurísticas que manipula e transforma uma única solução durante seu processo de busca, gerando outra solução promissora em sua vizinhança. Um conceito fundamental destes métodos é como passar de uma solução factível para outra. Para isso, define-se $N(s)$ como sendo a vizinhança da solução s , que é o conjunto de soluções para as quais é possível obter uma solução $s' \in N(s)$, com apenas uma operação, denominada movimento sobre s . A figura 4.1 apresenta uma vizinhança.

Figura 4.1: Exemplo de estrutura de vizinhança.

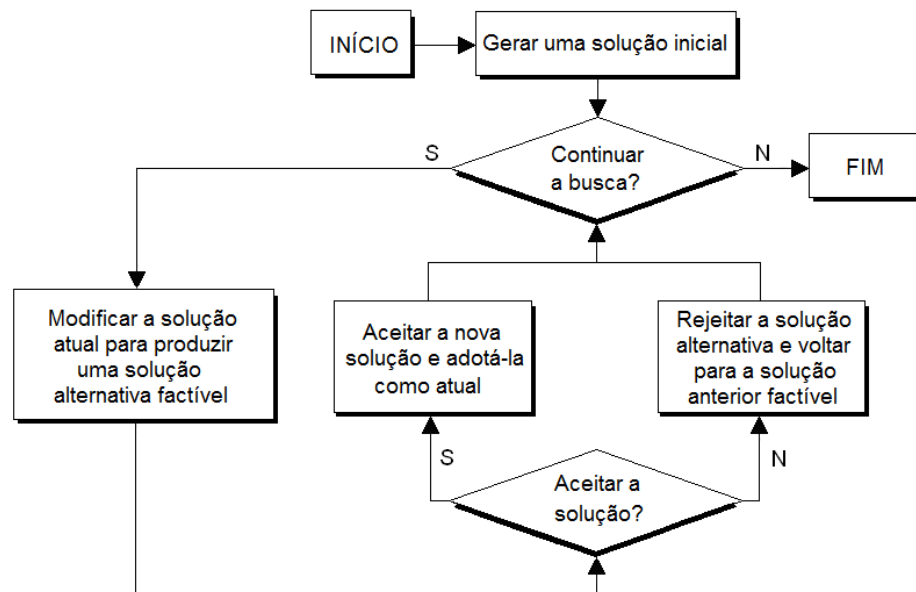


Fonte: Adaptado de Correia et al. (2000).

Em geral, esses métodos buscam entre as possíveis soluções da vizinhança alguma solução que tenha um melhor valor de FO do que a solução atual, e repetem este processo até que não existam melhores soluções no conjunto ou algum critério de parada seja satisfeito (FALCONE, 2004).

A figura 4.2 apresenta o fluxograma de um algoritmo baseado em uma solução única. Como exemplos deste grupo têm-se as metaheurísticas *Simulated Annealing*, *Busca Tabu* e *Iterated Local Search*.

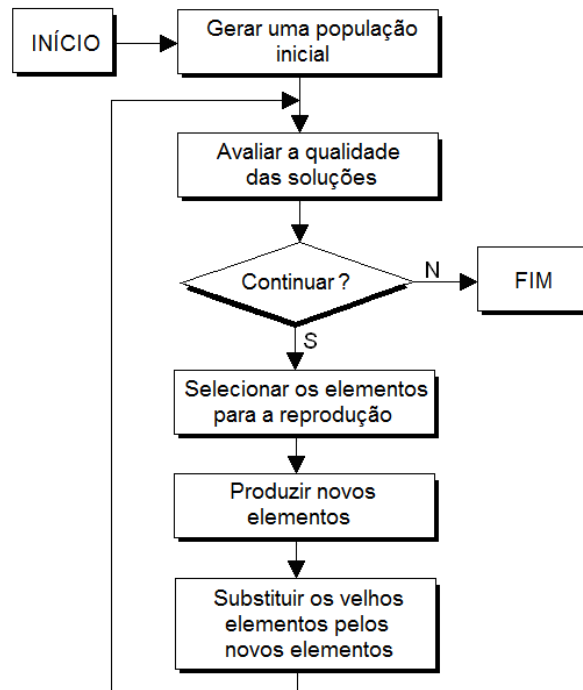
Figura 4.2: Fluxograma de um método de busca baseado em solução única.



Fonte: Adaptado de Murray (1999).

No segundo grupo de metaheurísticas baseadas em um conjunto de possíveis soluções (população), as interações entre seus elementos tentam produzir soluções ainda melhores. A figura 4.3 apresenta o fluxograma de um algoritmo de busca por população. Como exemplos deste grupo, têm-se os Algoritmos Evolucionários e a Otimização por Nuvem de Partículas.

Figura 4.3: Fluxograma de um método de busca por população.



Fonte: Adaptado de Ashlock (2005).

4.4 Diversificação e Intensificação

Para obterem melhores resultados, as metaheurísticas precisam ser adaptadas ao problema em que estão sendo aplicadas, seja com a alteração da função objetivo empregada, seja com relação aos parâmetros utilizados. Uma significativa melhoria no desempenho das metaheurísticas pode ser alcançada com a implementação das estratégias de diversificação e intensificação.

O termo investigar se refere à capacidade da metaheurística de saltar, a passos largos, de uma região para outra no espaço de busca (BLUM; ROLI, 2003). Esta estratégia também é conhecida como diversificação e direciona a busca por regiões pouco ou ainda não exploradas, com o objetivo de varrer todo o espaço de busca do problema. Uma das formas é a geração de soluções com atributos diferentes dos já encontrados nas melhores soluções obtidas. Já o termo exploração reflete a capacidade em explorar de forma mais intensa uma mesma região dentro do espaço de busca. A exploração, comumente chamada de intensificação, tem por objetivo concentrar a busca em regiões promissoras do espaço de busca (BLUM; ROLI, 2003). Uma das estratégias é a modificação das regras para encorajar combinações ou características historicamente boas (GLOVER; LAGUNA, 1997).

Em resumo, o que existe são dois processos de busca, a investigação e a exploração, e a metaheurística terá êxito, em um problema de otimização, se proporcionar um equilíbrio dinâmico entre estas duas habilidades da busca (BLUM; ROLI, 2003).

A seguir serão apresentados o estado da arte de controladores PID com metaheurísticas e mais detalhes dos dois algoritmos aplicados neste trabalho para o problema da sintonia de controladores.

4.5 Estado da arte em controladores PID com metaheurísticas

A sintonia ótima dos parâmetros dos controladores PID constitui ainda hoje um desafio em nível de engenharia, onde uma adequada estratégia de controle dos processos pode resultar em um aumento da vida útil dos atuadores, na melhoria da velocidade de resposta do sistema e no aumento da eficiência energética, ocasionando vantagens de ordem econômica e ambiental. Desta forma, a busca por melhores métodos de sintonia tem justificado o uso de metaheurísticas para suprir certas deficiências dos métodos convencionais. Vários autores propuseram diferentes metaheurísticas para ajustes dos parâmetros de controladores PID, descritas nos próximos parágrafos.

Nagaraj e Vijayakumar (2011) abordaram quatro metaheurísticas (Algoritmos Genéticos, Programação Evolutiva, Otimização por Enxame de Partículas e Otimização por Colônia de Formigas) para otimizar o projeto de um controlador PID para um motor de corrente contínua. Foi realizada uma comparação entre os resultados obtidos pelos métodos heurísticos e por um método convencional (Ziegler-Nichols), utilizando critérios de desempenho no domínio do tempo (*overshoot*, índice de erro ISE e tempos de acomodação e subida). Cada metaheurística utilizou como função custo o índice ISE. Os resultados da simulação demonstraram que todos os métodos de sintonia com metaheurísticas apresentaram desempenhos superiores (*overshoot*, tempos de subida e acomodação) em relação ao método convencional abordado. Dentre as metaheurísticas, o método PSO apresentou o melhor desempenho em todos os critérios do domínio do tempo e menor função custo.

Ganapathy e Jerome (2010) utilizaram o algoritmo Busca Tabu direto e um método de busca local para determinar os parâmetros de um controlador PID para uma classe de sistemas de controle que apresentam como características atrasos de transporte no processo. Os resultados das simulações foram comparados com os métodos de Tyreus e Luyben (1992), Visioli (2001), Chidambaram e Sree (2003), Marlin (1995) e Rotstein e Lewin (1991). A metaheurística utilizou como função objetivo o índice ISE. O método proposto apresentou melhor desempenho em termos de índice de erro, mas resultados inferiores em relação à robustez quando os parâmetros de atraso da planta foram alterados.

Iruthayarajan e Baskar (2009) compararam o desempenho de projetos de controladores PID multivariáveis descentralizados obtidos pelos métodos Algoritmo Genético codificado com números reais (*Real Coded Genetic Algorithm - RGA*), Otimização por Enxame de partículas modificado (*Modified Particle Swarm Optimization - MPSO*), Evolução Diferencial (*Differential Evolution - DE*) e Estratégia Evolutiva com Adaptação da Matriz de Covariância (*Covariance Matrix Adaptation Evolution Strategy - CMAES*). Foi utilizada como planta do processo a coluna de destilação Wood and Berry que tem uma forte interação entre as malhas, e atrasos de tempo diferentes. Os resultados também foram comparados com outros métodos mais tradicionais como BLT e Algoritmos Genéticos com multicruzamento. Os resultados mostraram o melhor desempenho dos algoritmos CMAES e MPSO em termos de menor *overshoot* e tempo de acomodação e um ligeiro aumento do tempo de subida.

Wang et al. (2009) apresentaram um algoritmo híbrido chamado de Recozimento Simulado Genético (*Genetic Simulated Annealing Algorithm - GSAA*) para otimizar os parâmetros de um controlador PID. Este algoritmo proposto combinou os méritos do Algoritmo Genético (AG) com o algoritmo *Simulated Annealing* (SA). O método consistiu basicamente em criar uma população inicial aleatoriamente, e em seguida produzir novos indivíduos através de um operador de seleção, um operador de cruzamento e um operador de mutação. Na sequência, o operador *Simulated Annealing* era aplicado em cada indivíduo da população, obtendo desta forma os indivíduos da geração seguinte. Esta iteração era repetida

até atingir a condição de parada. Como função de avaliação foi utilizada uma soma ponderada de quatro objetivos: *overshoot*, tempo de subida, esforço de controle e índice Valor Absoluto do Erro (IAE). Para a função de resfriamento do algoritmo SA foi utilizado o *Fast Simulated Annealing* e para o AG foi inserida a característica adaptativa no cruzamento e na mutação. Os resultados da simulação demonstraram que, comparados com o Algoritmo Genético e o *Simulated Annealing*, o método híbrido apresentou o melhor desempenho.

Hung et al. (2008) propuseram um algoritmo *Simulated Annealing* multiobjetivo e sua aplicação no desenvolvimento de controladores PID. Três objetivos de minimização foram considerados na formulação do problema de controle ótimo: atenuação de distúrbios, robustez e rastreamento preciso do *setpoint*. O mecanismo de geração multiobjetivo utiliza frentes de Pareto baseadas em uma função de pontuação sem o uso de heurísticas e considerando os objetivos individual e global. Seus desempenhos foram comparados com os algoritmos *Strength Pareto Evolutionary Algorithm 2* (SPEA2) e *Simulated Annealing* Ortogonal (HO et al., 2004) e com os métodos de Chen e Cheng (1998) e Kitsios et al. (2001), utilizando para fins de simulação um sistema de controle longitudinal de aeronave supermanobrável. A metaheurística proposta apresentou os melhores resultados juntamente com os métodos *Strength Pareto Evolutionary Algorithm 2* (SPEA2) e de Chen e Cheng (1998).

Chang (2007) propôs uma alteração no operador cruzamento do Algoritmo Genético para determinar os parâmetros do controlador PID para processos multivariáveis. Esta alteração consistia em mudar o número de cromossomos do operador *crossover* de dois para três, gerando desta forma, três descendentes. Simulações foram realizadas para demonstrar a eficácia do algoritmo proposto e seus resultados foram comparados com os obtidos pelos métodos BLT e algoritmo genético tradicional. Foi utilizada como planta do processo a coluna de destilação Wood and Berry e como função de avaliação o índice IAE. O AG proposto apresentou as melhores respostas de saída no domínio do tempo e menor índice IAE.

Nasri et al. (2007) utilizaram o método PSO para a determinação dos parâmetros do controlador PID para controle de velocidade de um motor DC *brushless* linear. Como função objetivo foi utilizada a abordagem da agregação de funções com quatro critérios de desempenho no domínio do tempo: *overshoot*, tempo de subida, tempo de acomodação e erro estacionário em regime permanente. Os resultados das simulações foram comparados com os obtidos pelos métodos Algoritmos Genéticos e Regulador Linear Quadrático (LQR - *Linear Quadratic Regulator*). O método proposto foi o mais eficiente nos quatro critérios de desempenho utilizados na função objetivo.

Hai-Bin et al. (2006) apresentaram uma aplicação do algoritmo de Otimização de Colônia de Formigas para otimizar os parâmetros de um controlador PID não linear. A metaheurística utilizou como função custo o índice ISE. Para validar a metodologia, foi utilizado um sistema de controle de voo que tem como características alta precisão, resposta de rastreamento rápida e alta confiabilidade. As simulações realizadas apresentaram bons resultados em relação ao seguimento da referência e satisfatórios em relação ao *overshoot*.

Sadasivarao e Chidambaram (2006) apresentaram um algoritmo genético para o ajuste de controladores PID para controle de sistemas em cascata. Neste trabalho foi proposta uma metodologia para selecionar a região de busca utilizando o método de sintonia de Ziegler e Nichols com o objetivo de eliminar a região de instabilidade do domínio de busca de forma a garantir a robustez. As malhas (interna e externa) eram sintonizadas de forma simultânea e utilizavam como função objetivo a soma dos erros de cada malha através do índice IAE, quando degraus unitários eram aplicados em cada malha de carga do sistema. Dois exemplos diferentes com duas diferentes estratégias de controle (P/PI e PID/PID) foram estudados e seus desempenhos comparados com o método de Lee et al. (1998) e com o algoritmo de Luus

e Jaakola (1973), utilizando o índice de desempenho IAE. Os resultados mostraram a superioridade da metaheurística sobre os outros métodos. A robustez para todos os métodos foi verificada através de uma perturbação de +20% adicionada nos atrasos de transporte das malhas interna e externa e na carga da malha interna. Os três métodos deram respostas estáveis, com o AG apresentando os melhores resultados.

Popov et al. (2005) propuseram um algoritmo genético multiobjetivo para melhorar o desempenho de uma planta de neutralização química, conhecida por ser altamente não-linear e com diferentes atrasos de transporte. Primeiro dados experimentais da planta funcionando em malha fechada foram recolhidos. O conjunto de dados foi então usado para treinar uma rede neural, através de um *toolbox* do MATLAB® para a identificação de sistemas dinâmicos e desta forma, um modelo satisfatório da planta foi obtido. Uma comparação dos resultados do modelo com um conjunto de dados de validação cruzada foi realizado. Após, dois métodos foram utilizados para encontrar os parâmetros de um controlador PID digital: agregação e Pareto-Ótimo. Para o AG multiobjetivo foi utilizado o algoritmo SPEA2. Os métodos levaram em consideração dois objetivos: o erro entre o sinal de referência e a saída (através do índice ISE) e o mínimo esforço de controle. Na análise realizada, as soluções do método da agregação de funções convergiam para um dos lados de cada objetivo, apresentando desta forma uma lacuna em algumas regiões da superfície de Pareto. Isto foi devido à elevada sensibilidade da função objetivo agregada. Já o AG multiobjetivo apresentou uma boa distribuição sobre a superfície de Pareto. Três soluções foram escolhidas, onde dois pontos correspondiam tanto aos resultados da otimização com agregação quanto da otimização de Pareto. A experiência comprovou que a solução apresentada pelo AG multiobjetivo apresentou superioridade em termos de eficiência.

Swiech (2005) abordou o problema de sintonizar controladores PI/PID em processos multivariáveis como um problema de otimização multiobjetivo, que foi resolvido através de um Algoritmo Genético modificado. Os critérios ITSE e a variância mínima aplicados ao sinal de controle e à saída do processo foram combinados em uma função de avaliação para cada malha. Para a validação da metodologia, foram utilizados os controladores PID e PD-nebuloso, aplicados sob arquitetura de controle descentralizado em três diferentes processos multivariáveis, coluna de destilação Wood and Berry, coluna de destilação de Isopropanol e processo de Craqueamento Catalítico em Leito Fluidizado. Os resultados mostraram a aplicabilidade e eficiência da metodologia proposta.

Varol e Bingul (2004) utilizaram o algoritmo de Otimização por Colônia de Formigas para determinar os parâmetros do controlador PID baseados nos índices IAE, ISE e RBEMCE (Referência Baseada no Erro com Mínimo de Esforço de Controle). Seus resultados foram comparados com os métodos ZN (Ziegler-Nichols), IMC (*Internal Model Control*) e IFT (*Iterative Feedback Tuning*) através dos critérios no domínio do tempo (*overshoot*, tempo de acomodação e tempo de subida) e robustez com três alterações no sistema (aumento do ganho estacionário, mudança de um dos polos do processo e adição de um atraso de transporte na planta) avaliando individualmente cada efeito. A metaheurística apresentou o melhor desempenho juntamente com o método IFT. No teste de robustez, a metaheurística foi o método que apresentou, em cada caso, os melhores resultados quanto ao tempo de acomodação e *overshoot*.

Observou-se nos trabalhos anteriores que os mesmos não levaram em consideração a influência dos parâmetros intrínsecos das metaheurísticas e a maior parte dos trabalhos citados utilizou os índices de desempenho baseados no erro (IAE, ISE, ITAE ou ITSE) como função custo, mas não foram realizados testes de comparação. Apenas o trabalho de Varol e Bingul (2004) realizou este experimento quanto ao índice de erro.

Desta forma, o primeiro ponto levantado nesta dissertação é com relação à escolha da função custo, onde testes são realizados considerando três índices baseados no erro: IAE, ISE e ITAE. O segundo ponto deste trabalho investiga as influências do tamanho da população e dos operadores cruzamento e mutação nos Algoritmos Genéticos e as influências da temperatura inicial e das funções de resfriamento e geração de novas soluções no *Simulated Annealing*. O trabalho também propõe, para a escolha da solução ótima dos parâmetros PID, a utilização de um procedimento de Hu et al. (2011) para o decisor da otimização multiobjetivo, levando em consideração a robustez do sistema de controle com o objetivo de obter melhores desempenhos do processo frente aos trabalhos apresentados na literatura.

4.6 *Simulated Annealing* (SA)

O *Simulated Annealing* (SA) ou Recozimento Simulado é uma metaheurística proposta por Kirkpatrick et al. (1983) e que tem sido amplamente utilizada para resolver problemas de otimização combinatória. Seu objetivo é a minimização de uma função objetivo, aceitando movimentos de piora como forma de escapar de ótimos locais.

É uma técnica baseada nos princípios da termodinâmica, análoga ao processo de recozimento empregado na fabricação de cerâmicas, cristais e vidros. Este método também é conhecido na literatura como *Monte Carlo Annealing*, *Probabilistic Hill-Climbing*, *Stochastic Relaxation* e *Probabilistic Exchange Algorithm* (ZBIGNIEW; FOGEL, 2007).

A ideia do SA veio de um artigo publicado por Metropolis et al. (1953). Este algoritmo simula o material como um sistema de partículas, onde a temperatura de um metal é inicialmente elevada e em seguida lentamente baixada até alcançar um estado mínimo de energia (KIRKPATRICK et al., 1983). O objetivo deste processo é a obtenção de uma estrutura cristalina (sem imperfeições) proporcionando propriedades mais elevadas de dureza e resistência mecânica.

As principais relações entre o processo físico e o processo de otimização estão listadas na tabela 4.1 (YAMAMOTO, 2004):

Tabela 4.1: Relações entre os processos físico e de otimização.

Processo físico	Processo de otimização
Energia	Função Objetivo
Temperatura	Parâmetro de controle
Estados de transição	Possíveis soluções
Estrutura cristalina	Ótimo global

Fonte: Adaptado de Yamamoto (2004).

A ideia básica do algoritmo é gerar uma solução S_j na vizinhança de S_i . Se o valor da função de avaliação $f(S_j)$ for menor que o da solução corrente $f(S_i)$, com $\Delta E = f(S_i) - f(S_j)$, ela é automaticamente aceita. Caso contrário, a solução gerada é aceita ou rejeitada de acordo com uma probabilidade dada pela equação:

$$P = e^{(-\Delta E/T)}, \quad (4.1)$$

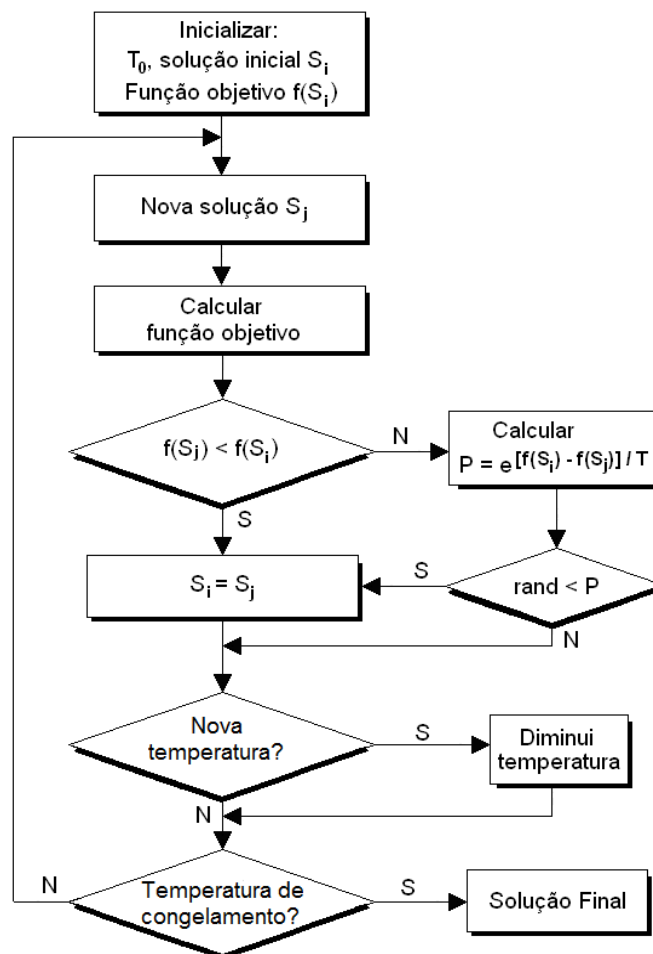
onde ΔE é a variação de energia e T é a temperatura.

Percebe-se pela equação (4.1) que a probabilidade de aceitação é controlada por dois fatores: pela diferença entre os valores de energia (função objetivo) e pela temperatura. Desta forma, quanto maior a temperatura, maior a probabilidade de aceitação da solução gerada. Para uma temperatura fixa, quanto maior a diferença de energia, menor a probabilidade de aceitar os movimentos que pioram a solução. O processo inicia-se em geral com T assumindo um valor alto, o que permite uma exploração do espaço de busca em toda a sua extensão.

Existe um processo de resfriamento nesta metaheurística e, à medida que a temperatura diminui, a probabilidade de aceitação diminui até que T seja tão pequena que nenhuma solução possa ser aceita.

A figura 4.4 apresenta o fluxograma de um algoritmo SA.

Figura 4.4: Fluxograma do algoritmo *Simulated Annealing*.



Fonte: Adaptado de Yamamoto (2004).

De acordo com Blum e Roli (2003), a regra de resfriamento pode variar durante a busca, com o objetivo de ajustar o balanço entre diversificação e intensificação. Por exemplo, no início da busca, a temperatura T pode ser constante ou diminuir linearmente, favorecendo assim a investigação do espaço de busca do problema. À medida que a temperatura diminui ocorre o inverso, a intensificação passa a ser maior do que a diversificação.

O final do processo ocorre quando a temperatura se aproxima de zero (temperatura de congelamento) e/ou mais nenhuma solução de piora é aceita, obtendo desta forma um ótimo local. Outra forma é fixar um número máximo de iterações.

4.6.1 Vantagens e desvantagens do SA

Tan et al. (2001) e Busseti (2001) apud Gomes (2003) descrevem algumas vantagens desta metaheurística como:

- Tem habilidade e flexibilidade para se aproximar da solução ótima;
- É relativamente fácil de codificar, mesmo para problemas complexos.

Porém, conforme Gomes (2003), o algoritmo possui algumas desvantagens, como:

- A princípio é necessário um processo lento de resfriamento e isso resulta em tempos elevados de processamento;
- Apesar de convergir para a solução ótima, a velocidade de redução da temperatura exigida implica em visitar um número exponencial de soluções.

4.6.2 Aspectos do método SA

O método *Simulated Annealing* consiste de três componentes fundamentais (INGBER, 1989):

- Uma densidade de probabilidade $g(x)$ de D parâmetros $x = \{x^i; i = 1, D\}$ usada na geração de novas soluções;
- Uma densidade de probabilidade $h(x)$ usada na aceitação/rejeição de novas soluções;
- Uma função temperatura (de Resfriamento) $T(t)$, para determinar como a temperatura deve ser alterada.

Alguns fatores influenciam diretamente o processo e desta forma precisam ser definidos ou determinados. Entre eles:

- A temperatura inicial T_0 ;
- O número de transições N_k , a ser realizada em cada temperatura T ;
- O critério de parada para terminar o algoritmo.

4.6.3 Funções de Resfriamento

Há várias formas de fazer o resfriamento. Uma das possibilidades é através da função de temperatura proporcional ou método geométrico, dada pela equação:

$$T_{k+1} = \alpha T_k, \quad (4.2)$$

onde α é uma constante e T_k é a temperatura da k -ésima iteração.

Este método fornece decréscimos menores em T à medida que a temperatura se aproxima de zero. Valores típicos para α usados na prática estão entre 0,8 e 0,99 (KIRKPATRICK et al., 1983).

Um segundo método é o Boltzman *Annealing* com sua função de resfriamento dada pela equação (4.3):

$$T_{k+1} = \frac{T_0}{\ln N_k}. \quad (4.3)$$

Uma vez que o método anterior é muito lento para uso na prática, outros métodos foram propostos por pesquisadores como Szu e Hartly (1987) como o método *Fast Simulated Annealing*, dado pela equação de resfriamento:

$$T_{k+1} = \frac{T_0}{N_k}. \quad (4.4)$$

4.6.4 Geração de novas soluções (Função *Annealing*)

O algoritmo empregado altera a solução corrente S_i gerando uma nova solução S_j . Após ele escolhe a distância da nova solução a partir da solução corrente por uma distribuição de probabilidade, com uma escala que depende da temperatura atual (ROEVA, 2011). Esta distribuição de probabilidade (densidade de probabilidade) é chamada de Função *Annealing*. Para gerar novas soluções em cada transição, pode-se utilizar o método Boltzman *Annealing*, onde a densidade de probabilidade $g(x)$ é dada pela equação abaixo (INGBER, 1989):

$$g(x) = (2\pi T)^{-D/2} \exp[-\Delta x^2 / (2T)], \quad (4.5)$$

onde $\Delta x = x - x_0$ é o desvio entre x (solução gerada) e x_0 (solução corrente).

Outra forma é a distribuição do método *Fast Simulated Annealing* descrita abaixo (INGBER, 1989),

$$g(x) = \frac{T}{(\Delta x^2 + T^2)^{(D+1)/2}}, \quad (4.6)$$

Esta distribuição, em relação à distribuição de Boltzmann, permite um acesso mais rápido para testar um mínimo local na busca pelo mínimo global (INGBER, 1989).

4.6.5 Temperatura inicial

Para o SA, foi estudada também a influência da temperatura inicial. Uma determinação de T_0 é descrita por Reeves (1993) na equação abaixo:

$$T_0 = \frac{\mu}{-\ln \phi} f(x^0), \quad (4.7)$$

onde assume-se que ϕ % dos movimentos contrários, que são μ % piores que a solução inicial $f(x^0)$ serão aceitas na temperatura inicial T_0 . Comumente, utiliza-se um valor de $\phi = 0,85$, ou seja, 85% dos testes de aceitação são aceitos nesta temperatura inicial.

4.6.6 Reannealing

Esta é uma técnica utilizada na metaheurística, que tem como finalidade escapar de um possível ótimo local. Consiste em aplicar novamente o algoritmo na melhor solução já obtida como solução inicial, para que possa atingir outras regiões e tentar melhorar ainda mais a solução.

4.7 Algoritmos Genéticos (AG)

O Algoritmo Genético é um método de busca global estocástico baseado nos conceitos da seleção natural e da genética. O princípio da seleção natural foi inspirado na teoria de Darwin onde os indivíduos mais aptos, com relação a um determinado ambiente, têm maiores chances de sobrevivência e reprodução, enquanto indivíduos menos aptos tendem a desaparecer durante o processo evolutivo. Porém, a teoria de Darwin não explicava como era realizada a transmissão das características dos pais para os filhos. Apenas por volta de 1900, o trabalho de Gregor Mendel, desenvolvido em 1865, sobre os princípios básicos da herança genética, foi redescoberto e estudado por vários cientistas, influenciando grande parte dos trabalhos relacionados à evolução.

Este método emprega um processo adaptativo e paralelo em busca da solução ótima em problemas complexos. O processo é considerado adaptativo, pois as soluções atuais influenciam a busca por futuras soluções e é considerado paralelo em decorrência natural do fato de que diversas soluções são consideradas a cada momento (DAVIS, 1991).

Conforme Reeves apud Glover e Kochenberger (2003), o termo Algoritmo Genético foi usado pela primeira vez em 1975 pelo engenheiro eletricitista John Holland em seu livro "*Adaptation in Natural and Artificial Systems*". Holland foi professor da Universidade do Michigan nos EUA, e vários de seus alunos deram contribuições importantes, como Kennedy De Jong em sua tese de doutorado "*An Analysis of the Behavior of a Class of Genetic Adaptive Systems*", sendo o primeiro a fornecer um tratamento completo das capacidades de otimização dos AGs. Outro aluno de Holland, David Goldberg deu a estrutura básica do AG em seu popular livro "*Genetic Algorithms in Search, Optimization and Machine Learning*".

4.7.1 Definições

As principais definições relacionadas com os algoritmos genéticos e de relevância para este trabalho são (CASTRO, 2001 e GUERRA, 2008):

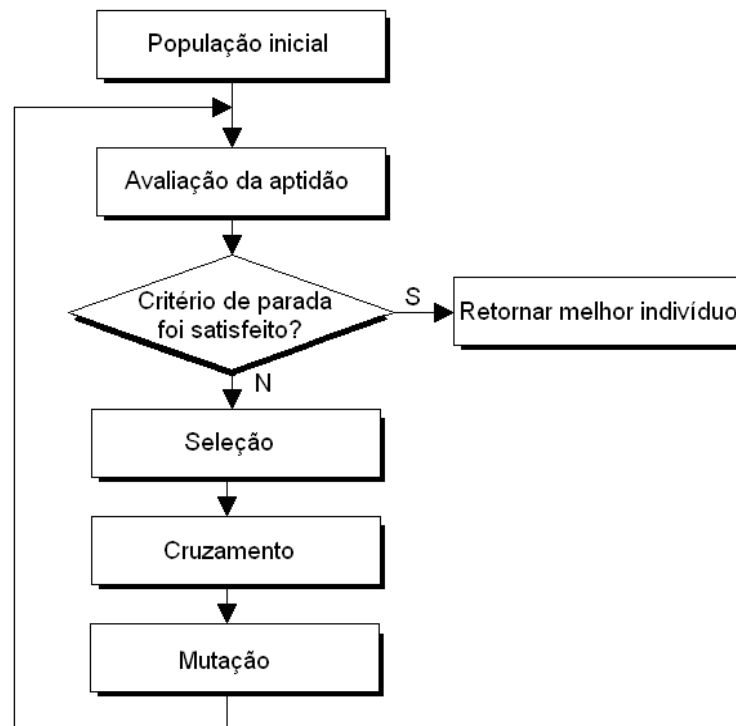
- **Indivíduo ou cromossomo:** potencial solução do algoritmo genético;
- **Gene:** unidade básica do cromossomo, ou seja, sua menor porção;
- **População:** conjunto de todas as potenciais soluções;
- **Pais e filhos:** em cada geração o AG cria populações. Uma delas é chamada de população de pais a qual é utilizada para criar novos indivíduos na próxima geração chamada de população de filhos.
- **Operações genéticas:** operações que o Algoritmo Genético realiza sobre cada um dos cromossomos;
- **Geração:** iteração completa que o algoritmo genético executa.

4.7.2 Estrutura básica de um algoritmo genético

A ideia básica do algoritmo genético é uma população composta por um grupo de indivíduos. Cada indivíduo representa uma das possíveis soluções. Esta população é avaliada segundo algum critério que mede a qualidade do indivíduo (função *fitness*). As melhores soluções têm maior probabilidade de sobreviver e através de operadores genéticos, novas soluções são geradas.

Os principais operadores genéticos são: seleção, cruzamento e mutação. Outros operadores têm sido propostos na literatura entre eles o mais utilizado é o Elitismo. Repetem-se estas operações até atingir algum critério de parada (encontrar uma solução aceitável e/ou atingir um número limite de gerações e/ou o algoritmo não consiga mais mostrar evolução). A figura 4.5 apresenta o fluxograma de um AG clássico.

Figura 4.5: Fluxograma de um AG clássico.



Fonte: Elaborado pelo autor.

4.7.3 Vantagens e desvantagens do AG

Castro (2001) e Guerra (2008) apontam algumas vantagens do AG:

- Não requerem conhecimentos ou informações dos gradientes da superfície definida pela função objetivo;
- São mais resistentes a se prenderem a ótimos locais;
- São de fácil implementação e proporcionam maior flexibilidade no tratamento do problema a ser resolvido.

E como desvantagens citam:

- Requerem um grande número de avaliações de funções de aptidão;
- A escolha dos operadores e valores para as configurações pode complicar a resolução do problema tratado.

4.7.4 Representação do AG

Existem basicamente quatro formas de realizar a representação da população dos algoritmos genéticos computacionalmente: binária, real e inteira (CASTRO, 2001 e MICHALEWICZ, 1996).

Na representação binária, os elementos do cromossomo pertencem ao conjunto $\{0,1\}$, sendo representados por uma cadeia de *bits*. Na representação real, cada cromossomo é um vetor pertencente a \mathfrak{R}^n . Man et al. (1996) destacam o crescente interesse dos pesquisadores pela representação direta das variáveis através de números reais em problemas contínuos.

4.7.5 Função *fitness*

A função de *fitness* é uma importante ligação entre o algoritmo genético e o sistema a ser otimizado. É responsável por avaliar cada cromossomo estabelecendo uma medida quantitativa, ou seja, seu grau de aptidão ao ambiente.

4.7.6 Operador de seleção

A seleção depende da avaliação quantificada de cada solução candidata que é obtida a partir da função custo. Ela determina quais soluções serão escolhidas, de acordo com o princípio de sobrevivência do mais forte, para a operação de cruzamento. Os mecanismos de seleção mais utilizados são o da Roleta e o Torneio.

O método da Roleta (*roulette wheel*) é o mecanismo original de seleção proposto por Holland. Este método consiste em calcular o *fitness* relativo de cada um dos indivíduos. Assim, quanto maior seu valor, maior é a chance de escolha deste indivíduo se reproduzir.

O cálculo do *fitness* relativo é determinado pela equação abaixo:

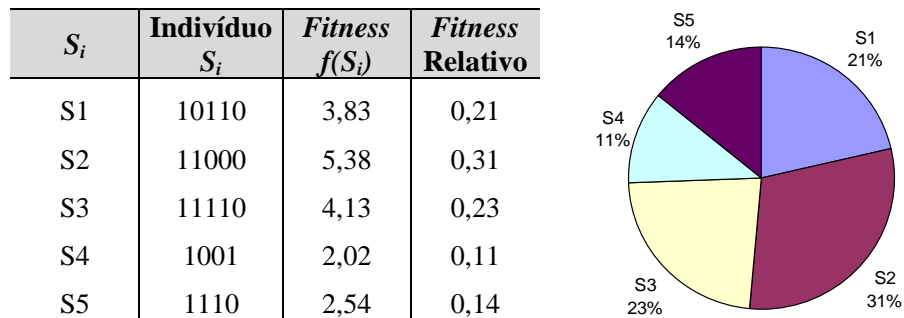
$$p_i = \frac{f(S_i)}{\sum_{k=1}^N f(S_k)}, \quad (4.8)$$

onde N é o tamanho da população, p_i a probabilidade de seleção e $f(S_i)$ é o *fitness* do indivíduo S_i da população.

Realiza-se o lançamento um determinado número de vezes da roleta, dependendo do tamanho da população, e são escolhidos para a próxima geração aqueles indivíduos por ela sorteados.

A figura 4.6 apresenta um exemplo do método da Roleta.

Figura 4.6: Método da Roleta.



Fonte: Elaborado pelo autor.

Na seleção por Torneio, gera-se um subconjunto aleatório da população. O indivíduo selecionado dependerá de uma probabilidade $Psel$ definida previamente. O indivíduo com o maior nível de aptidão no subconjunto é selecionado para sofrer a ação dos operadores genéticos. A figura 4.7 mostra o pseudocódigo do método do Torneio com dois indivíduos.

Figura 4.7: Método do Torneio.

```

Início
Psel = 0,75
Repita N vezes
  Escolha 2 indivíduos da população aleatoriamente
  r = valor aleatório de [0, 1]
  Se r < Psel
    o melhor indivíduo é escolhido
  senão
    o outro indivíduo é escolhido
Fim se
Fim Repita
Fim
  
```

Fonte: Adaptado de Mitchell (1999).

4.7.7 Elitismo

Introduzido por Kenneth De Jong em 1975, o elitismo é uma técnica onde são especificados quantos indivíduos da geração atual são garantidos para sobreviver na próxima geração, reduzindo desta forma o efeito aleatório do processo seletivo. Consiste basicamente em realizar o processo de seleção em duas etapas:

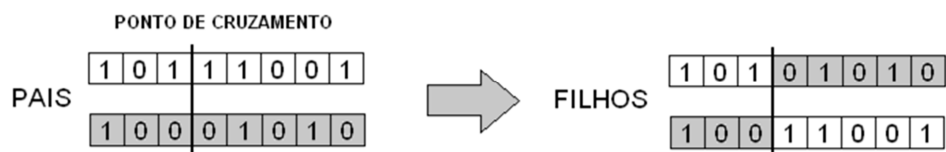
- Seleciona-se uma elite de ' e ' membros entre os melhores da população atual, os quais são incorporados diretamente à próxima população;
- Os demais elementos da população são obtidos a partir dos $(n - e)$ elementos restantes da população atual de tamanho n .

4.7.8 Operador de cruzamento (*crossover*)

Combina as características de dois pais para criar novas soluções. O operador permite uma melhora na espécie, em termos da evolução de novas soluções que estão mais aptas do que as anteriores. Mas nem todos os indivíduos são necessariamente gerados a partir do cruzamento (uma taxa de cruzamento é escolhida).

Existem diferentes tipos de operadores de cruzamento, entre eles o cruzamento de ponto único, de dois pontos e de multipontos (MAN et al., 1996). O operador de cruzamento de ponto único é mostrado na figura 4.8:

Figura 4.8: Operador cruzamento.



Fonte: Elaborado pelo autor.

Os operadores de cruzamento convencionais podem ser aplicados com a codificação real, mas possuem deficiências, pois eles apenas trocam informações entre os pais não criando novos valores (SANTA CATARINA, 2009). Alguns operadores para codificação real são: aritmético, intermediário e heurístico.

No *crossover* aritmético, os dois filhos (h_1 e h_2) são gerados a partir da combinação linear de dois pais (p_1 e p_2) da seguinte forma (MICHALEWICZ et al., 1994):

$$h_1 = \beta p_1 + (1 - \beta) p_2, \quad (4.9)$$

$$h_2 = \beta p_2 + (1 - \beta) p_1, \quad (4.10)$$

onde β é um número aleatório escolhido de uma distribuição uniforme no intervalo $[0,1]$.

O *crossover* intermediário cria filhos através dos pais considerando um valor de α (escolhido aleatoriamente) para cada filho (h), conforme equação (4.11) (MICHALEWICZ et al., 1994).

$$h = p_1 + \alpha(p_2 - p_1), \quad (4.11)$$

onde α é um número aleatório no intervalo $[0, 1]$.

O *crossover* heurístico leva em consideração o *fitness* do melhor indivíduo entre os dois pais com a finalidade de tentar gerar herdeiros mais semelhantes a esse pai de melhor aptidão. O filho (h) é gerado a partir de dois pais (p_1 e p_2) da seguinte forma:

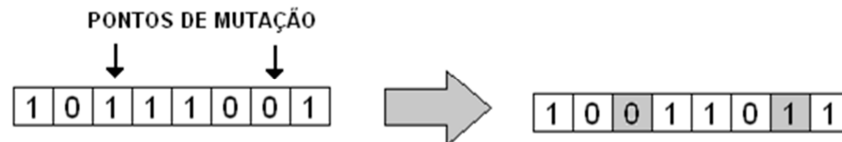
$$h = p_1 + \beta(p_1 - p_2), \quad (4.12)$$

onde p_1 é melhor que p_2 e β é um número aleatório no intervalo $[0, 1]$. Caso o valor de h seja inviável, outro valor de β é gerado e outro filho é criado.

4.7.9 Operador de mutação

O operador de mutação é o operador responsável pelo aumento da diversidade da população através da alteração aleatória de um ou mais genes de um cromossomo. Na figura 4.9 pode-se ver um exemplo de funcionamento do operador de mutação para o caso de codificação binária.

Figura 4.9: Operador mutação.



Fonte: Elaborado pelo autor.

Conforme Michalewicz e Schoenauer (1996) e Michalewicz et al. (1994) para codificação em números reais, existem outras formas de operadores de mutação como uniforme, gaussiana, *Adaptive Feasible* e não uniforme.

A mutação uniforme consiste em selecionar aleatoriamente genes do cromossomo para a mutação, onde cada gene tem uma taxa de probabilidade de ser mutada. Após, o algoritmo substitui cada gene selecionado por um número aleatório segundo uma distribuição uniforme, entre os limites mínimo e máximo permitidos para o gene (MICHALEWICZ; SCHOENAUER, 1996).

A mutação gaussiana substitui todos os genes de um cromossomo $x = [x_1 \dots x_n]$ por $x' = x + N(0, \sigma)$, onde $N(0, \sigma)$ é uma distribuição normal com média zero e desvio padrão σ . O valor de σ pode ser diminuído à medida que se aumenta o número de gerações (MICHALEWICZ; SCHOENAUER, 1996).

A mutação *Adaptive Feasible* gera aleatoriamente vetores de direção que são adaptáveis no que diz respeito à última geração bem sucedida ou não. A região viável é limitada pelas restrições de desigualdade. Um tamanho de passo é escolhido ao longo de cada direção para que as restrições sejam satisfeitas. Caso o indivíduo mutado esteja localizado numa região inviável, o operador ajusta automaticamente o tamanho do passo para um valor menor e gera um outro indivíduo mutante ao longo do vetor de direção (PAIS et al., 2010).

A mutação não-uniforme substitui o gene selecionado por um número extraído de uma distribuição não-uniforme. O procedimento deste operador é mostrado a seguir:

- Calcular o valor de y_k da seguinte forma:

$$y_k = \begin{cases} \max(k) - x_k, & \text{se bit sorteado} = 0 \\ x_k - \min(k), & \text{se bit sorteado} = 1 \end{cases} \quad (4.13)$$

onde $k \in \{1, 2, \dots, n\}$, $\max(k)$ e $\min(k)$ são os valores máximo e mínimo dos limites do parâmetro x_k , respectivamente.

- Calcular o valor da função probabilística $\Delta(g, y_k)$ através da equação:

$$\Delta(g, y_k) = y_k \left[1 - r \left(1 - \frac{g}{G} \right)^b \right], \quad (4.14)$$

onde g é o número da geração corrente (atual), G é o número máximo de gerações, b é um parâmetro do sistema que determina o grau de não-uniformidade e r um número aleatório de $[0, 1]$.

A função $\Delta(g, y_k)$ retorna um valor no intervalo $[0, y_k]$. O novo gene x'_k será determinado através da equação abaixo (MICHALEWICZ et al., 1994):

$$x'_k = \begin{cases} x_k + \Delta(g, y_k), & \text{se bit sorteado} = 0 \\ x_k - \Delta(g, y_k), & \text{se bit sorteado} = 1 \end{cases}. \quad (4.15)$$

4.7.10 Definição de parâmetros

O comportamento dos AGs é influenciado pelos operadores genéticos, bem como pelos parâmetros genéticos utilizados. Segundo Castro (2001), podem ser listados como:

- **Tamanho da população (N):** o tamanho da população pode afetar o desempenho dos algoritmos genéticos. Populações muito pequenas podem provocar um sério risco de não obter cobertura do espaço de busca do problema. Uma grande população fornece uma cobertura adequada, mas são necessários mais recursos computacionais para a resolução do problema, caso contrário o algoritmo trabalhará por um período de tempo muito maior;
- **Número de gerações:** varia de acordo com a complexidade do problema e deve ser determinado experimentalmente;
- **Taxa de cruzamento:** este parâmetro determina a taxa ou probabilidade com que ocorrerá o cruzamento entre os indivíduos selecionados na população. Valores comuns são de 50% a 95%;
- **Taxa de mutação:** este parâmetro determina a taxa ou probabilidade de uma mutação nos indivíduos da população. A probabilidade de mutação geralmente é baixa, pois uma alta taxa degeneraria o algoritmo em uma busca aleatória. Valores comuns são de 0,1% a 10%.
- **Critério de parada:** utiliza-se, geralmente, o número máximo de gerações estabelecido, o tempo limite de processamento, o critério de estagnação, ou seja, quando não se observa melhoria da aptidão da população por um determinado número de gerações ou uma combinação destes critérios.

4.7.11 Algoritmos Genéticos para Otimização Multiobjetivo

Os Algoritmos Genéticos Multiobjetivos (AGMs) são formados pela união entre uma estratégia de otimização multiobjetivo e os AGs. De modo geral, estes algoritmos são adaptados para que continuem a utilizar seus três operadores genéticos básicos: seleção, cruzamento e mutação (ZINI, 2009).

A primeira implementação de um AGM foi realizada por Schaffer em 1985, denominada VEGA (*Vector-Evaluated Genetic Algorithm*). Este algoritmo avalia cada objetivo separadamente e tem como desvantagem não apresentar uma boa diversidade nas soluções da Fronteira de Pareto (TICONA, 2003).

Durante quase uma década, nenhum estudo relevante foi apresentado sobre o assunto, com exceção de Goldberg (1989), que desenvolveu um procedimento para ordenação de soluções baseado no conceito de dominância, publicado em seu livro *Genetic algorithms in search, optimization, and machine learning*.

A partir do algoritmo de Goldberg, vários pesquisadores passaram a desenvolver uma série de técnicas baseadas em Algoritmos Genéticos Multiobjetivos, entre eles, destacam-se:

- MOGA - *Multiple Objective Genetic Algorithm* (FONSECA; FLEMING, 1993);
- NPGA - *Niched-Pareto Genetic Algorithm* (HORN et al., 1994);
- NSGA - *Non-Dominated Sorting Genetic Algorithm* (SRINIVAS; DEB, 1995);
- SPEA - *Strength Pareto Evolutionary Algorithm* (ZITZLER; THIELE, 1998);
- PESA - *Pareto Envelope-Base Selection Algorithms* (CORNE et al., 2000);
- NSGA-II (DEB et al., 2002);
- PAES - *Pareto-Archived Evolutionary Strategy* (KNOWLES; CORNE, 1999).

Uma revisão mais completa sobre Algoritmos Genéticos Multiobjetivos pode ser encontrada em Coello et al. (2007) e em Deb (2001).

4.8 Considerações sobre as metaheurísticas aplicadas ao problema da sintonia

A sintonia dos controladores PID para diferentes estruturas de controle é feita diretamente pela busca dos melhores valores de K_p , K_i e K_d através da aplicação das metaheurísticas propostas, em substituição aos métodos clássicos como Ziegler e Nichols (1942) para sistemas SISO, Lee et al. (1998) para estruturas em cascata e *Biggest Log-Modulus Tuning* (LUYBEN, 1986) para sistemas MIMO.

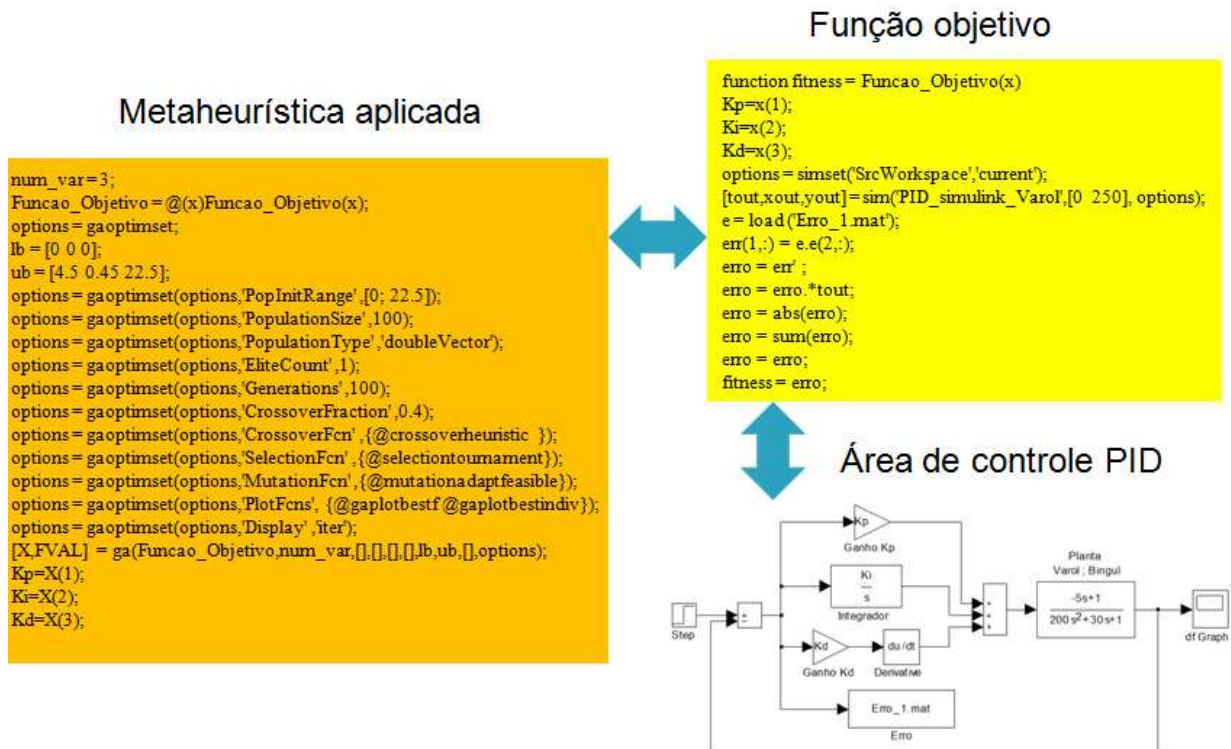
Neste capítulo foram apresentadas as metaheurísticas *Simulated Annealing* e Algoritmos Genéticos, enfatizando suas características que permitem empregá-las na sintonia de controladores PID. Além disso, relatou-se o Algoritmo Genético Multiobjetivo que também será utilizado para a otimização dos parâmetros de controle.

Os dois próximos capítulos apresentam respectivamente o ambiente de simulação implementando através do MATLAB® e do Simulink® e os testes realizados para verificar a influência dos parâmetros intrínsecos das metaheurísticas empregadas e as análises de desempenho das estruturas de controle sintonizadas por outras técnicas inteligentes e técnicas de sintonia convencionais.

5 AMBIENTE DE SIMULAÇÃO

O ambiente integrado de simulação (figura 5.1) foi dividido em três partes: metaheurística aplicada, função objetivo e área de controle PID.

Figura 5.1: Ambiente integrado de simulação.

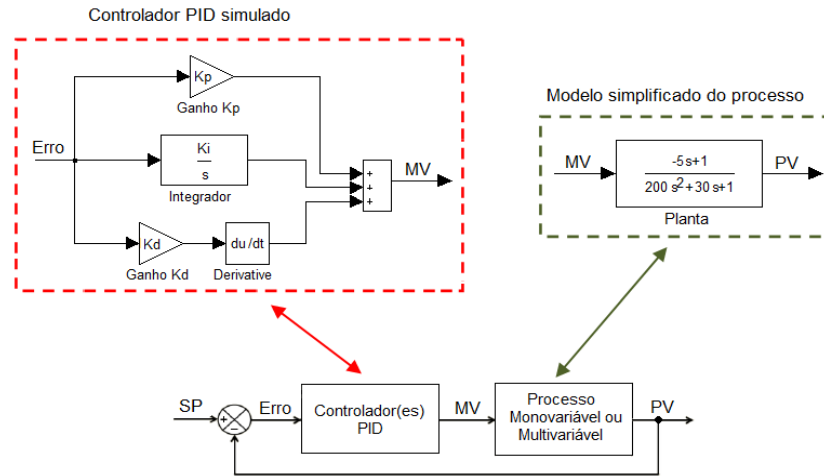


Fonte: Elaborado pelo autor.

A primeira seção corresponde ao tipo de metaheurística utilizada e ambiente de configuração dos seus parâmetros. A segunda seção é chamada pela metaheurística para realizar o cálculo da função objetivo aplicada na simulação dos controladores PID. Por fim, a terceira parte é a interface de simulação do controle de sistema dinâmico, composto pela planta, fonte degrau (*step*), controlador(es) PID com os três parâmetros a serem determinados e bloco gerador de dados (no caso uma tabela de erros ao longo do tempo).

Ambos os componentes (controladores e planta) são simulados por modelos computacionais que representam os comportamentos de interesse dos sistemas envolvidos. A sintonia dos controladores PID deste trabalho é realizada de forma *off-line* para cada estudo de caso (vide figura 5.2).

Figura 5.2: Ambiente de simulação PID.



Fonte: Elaborado pelo autor.

A figura 5.2 mostra um exemplo de sistema SISO representado pelo diagrama em blocos de controle formado por um controlador PID a ser projetado e uma planta a ser controlada. Ambos componentes estão escritos no domínio de Laplace.

5.1 Configurações básicas das metaheurísticas adaptadas ao problema

Matematicamente, o problema da sintonia de controladores PID pode ser definido por:

$$\begin{aligned}
 \min f(K_p, K_i, K_d) &= erro \\
 \text{sujeito a } 0 &\leq K_p \leq 10 \\
 &0 \leq K_i \leq 10 \quad , \\
 &0 \leq K_d \leq 10
 \end{aligned}
 \tag{5.1}$$

onde $f(K_p, K_i, K_d)$ é a função objetivo a ser minimizada sujeita às restrições dadas pelos limites inferior e superior de cada parâmetro do controlador PID. O número de parâmetros será dado de acordo com o problema a ser abordado (monovariável, cascata ou multivariável).

Os indivíduos da população e da solução única presentes respectivamente nos algoritmos AG e SA são compostos por variáveis reais, representando cada uma delas, os parâmetros a serem estimados. O modelo apresentado na equação (5.1) trata da estrutura de monovariável com três parâmetros (K_p , K_i e K_d).

O ambiente MATLAB® da Mathworks foi escolhido para a realização deste trabalho devido às capacidades gráficas e aos *toolboxes* com ferramentas disponíveis para controle e otimização global (*Global Optimization Toolbox*), aplicando as metaheurísticas através dos métodos *Simulated Annealing Solver*, *Genetic Algorithm Solver* e *Multiobjective Genetic Algorithm Solver*. O equipamento utilizado para simulação, testes e verificação de resultados foi um computador Intel Celeron de 2,13 GHz, 2 GB de RAM, com sistema operacional Windows Vista.

Os *scripts* do MATLAB® e ambientes de simulação do Simulink®, implementados para cada estrutura de controle, são apresentados no Apêndice A.

5.1.1 Algoritmo Genético adaptado ao problema

O trabalho utilizou as seguintes opções de configuração do AG para a estimação dos parâmetros dos controladores PID: representação das variáveis através de números reais; seleção por Torneio. Como critérios de parada foram utilizados o número máximo de gerações (100) e o número de gerações onde o valor ótimo fica estabilizado (*stall generation*), assumindo-se valor igual a 50.

5.1.1.1 Implementação do AG para sintonia PID

Resumidamente, o Algoritmo Genético implementado para a sintonia de controladores PID funciona da seguinte forma:

- O algoritmo cria uma população inicial aleatória, dentro dos limites especificados de K_p , K_i e K_d ;
- Estes valores são aplicados no modelo de controle em malha fechada estudado, e a partir de um degrau unitário aplicado em sua entrada, uma tabela de erros ao longo do tempo é gerada;
- O algoritmo utiliza esta tabela e outros critérios para avaliar a aptidão de cada indivíduo da população;
- Através do processo de seleção escolhido, indivíduos são selecionados e os operadores de cruzamento e mutação são aplicados gerando uma nova população;
- O algoritmo para quando um determinado critério de parada pré-estabelecido é encontrado.

5.1.1.2 Representação do AG no problema

Para o controle de sistemas SISO, o cromossomo é formado por três genes que correspondem aos três parâmetros a serem ajustados: K_p , K_i e K_d . A tabela 5.1 ilustra um exemplo desta representação.

Tabela 5.1: Representação do cromossomo SISO.

K_p	K_i	K_d
3,1415	0,0123	1,2189
1º gene	2º gene	3º gene

Fonte: Elaborado pelo autor.

Para um sistema multivariável, do tipo MIMO 2 x 2 descentralizado e estrutura em cascata com duas malhas, o cromossomo terá seis parâmetros, como mostrado no exemplo da tabela abaixo:

Tabela 5.2: Representação do cromossomo das estruturas em cascata e MIMO descentralizado.

K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
3,1415	0,0123	1,2189	5,7568	0,3457	2,0245
1° gene	2° gene	3° gene	4° gene	5° gene	6° gene

Fonte: Elaborado pelo autor.

Para um sistema MIMO 2 x 2 com desacopladores, a tabela 5.3 ilustra um exemplo de valores para o vetor cromossomo formado pelos oito parâmetros.

Tabela 5.3: Representação do cromossomo MIMO desacoplado 2 x 2.

K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}	D_1	D_2
3,1415	0,0123	1,2189	5,7568	0,3457	2,0245	- 1,0214	- 0,8633
1° gene	2° gene	3° gene	4° gene	5° gene	6° gene	7° gene	8° gene

Fonte: Elaborado pelo autor.

5.1.1.3 Método de seleção utilizado no problema

Na seleção por Roleta, os indivíduos destacados podem introduzir um favoritismo no início da procura que pode causar uma convergência prematura e uma perda da diversidade (TALBI, 2009). Desta forma, optou-se pela utilização da seleção por Torneio.

5.1.1.4 Elitismo utilizado no problema

Zitzler et al. (2000) e Van Veldhuizen e Lamont (2000) descrevem que o elitismo pode melhorar o desempenho dos AGs e impedir que soluções eficientes encontradas durante a procura sejam perdidas. Como não será realizada uma análise sobre a influência do elitismo, o fator escolhido foi igual a 1 (GOLDBERG, 1989), ou seja, será mantido na próxima população apenas o melhor indivíduo.

5.1.2 *Simulated Annealing* adaptado ao problema

As opções de configuração do SA para a estimação dos parâmetros dos controladores PID no trabalho proposto são: representação das variáveis através de números reais; número máximo de iterações igual a 3000 como critério de parada.

5.1.2.1 Implementação do SA

Resumidamente, o algoritmo *Simulated Annealing* implementado funciona da seguinte forma:

- O algoritmo cria uma solução inicial aleatória, dentro dos limites especificados de K_p , K_i e K_d ;
- Uma solução vizinha é criada também dentro dos limites especificados;
- Estes valores são aplicados ao modelo de controle em malha fechada estudado, e a partir de um degrau unitário aplicado em sua entrada uma tabela de erros ao longo do tempo é gerada. O algoritmo utiliza esta tabela e outros critérios para avaliar cada solução;
- Se o valor da função de avaliação da solução vizinha for menor que o da solução corrente, ela é aceita. Caso contrário, a solução gerada é aceita ou rejeitada de acordo com uma probabilidade;
- Após um número de iterações, a temperatura é gradativamente diminuída por uma função de resfriamento;
- O algoritmo para quando um determinado critério de parada pré-estabelecido é encontrado (temperatura chega a um valor próximo de zero ou número máximo de iterações).

5.1.2.2 Representação do SA no problema

Uma possível representação para uma solução SISO é apresentada na tabela 5.4, onde cada variável representa um dos parâmetros a serem estimados.

Tabela 5.4: Representação da solução SISO no SA.

K_p	K_i	K_d
3,1415	0,0123	1,2189
1ª variável	2ª variável	3ª variável

Fonte: Elaborado pelo autor.

As demais representações dos parâmetros dos controladores (cascata e MIMO) são similares às utilizadas nos Algoritmos Genéticos.

No próximo capítulo são apresentados estudos de casos. Cada estudo apresentou uma das três estruturas revisadas neste trabalho. No primeiro estudo de caso foi utilizada a estrutura monovariável, onde duas otimizações e diferentes índices de erro foram comparados. Para o segundo estudo de caso a estrutura em cascata foi escolhida e duas metodologias de sintonia (sequencial e simultânea), através das metaheurísticas AG e SA, foram comparadas. O terceiro estudo de caso apresentou o controle multivariável e duas arquiteturas (descentralizada e com desacopladores) foram avaliadas.

Além disso, em cada estudo é realizada uma análise comparativa entre as metaheurísticas empregadas neste trabalho, os métodos tradicionais de sintonia e as metaheurísticas propostas por autores citados no estado da arte.

6 ESTUDO DE CASOS

Este capítulo apresenta estudos de casos para as diferentes estruturas de controle, aos quais serão aplicados os controladores PID com otimização inspirada na natureza através de duas metaheurísticas: Algoritmo Genético e *Simulated Annealing*.

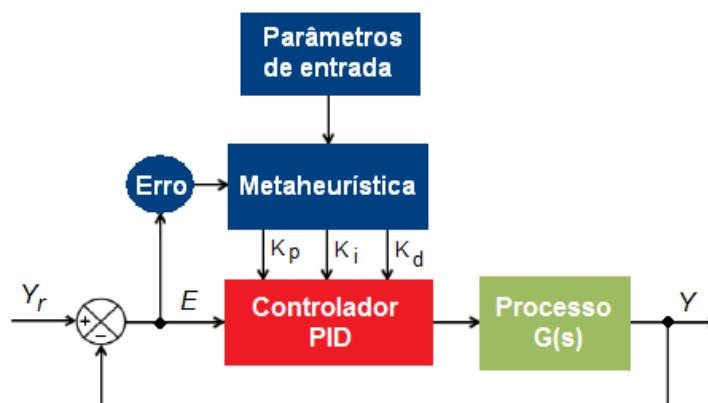
6.1 Controle monovariável

6.1.1 Considerações iniciais para o sistema SISO

Com relação à sintonia de controladores PID para sistema monovariável, esta dissertação utilizou o trabalho de Varol e Bingul (2004) como referência para o primeiro estudo de caso, avaliando o uso das abordagens mono-objetivo e multiobjetivo na função objetivo das metaheurísticas. Foi utilizada a mesma planta, faixa dos ganhos do controlador PID, testes de desempenho e robustez dos sistemas. Porém, existem duas importantes diferenças. Primeiramente, a função objetivo utilizada; enquanto Varol e Bingul (2004) utilizaram o erro com mínimo esforço de controle, este trabalho testou os índices IAE, ISE e ITAE, onde este último índice é o critério de desempenho que melhor concilia propriedades de desempenho e robustez (BARBOSA, 2007). Testes foram realizados para verificar seu desempenho. A segunda diferença é a utilização dos conceitos Margem de Ganho e Margem de Fase para avaliar a robustez dos sistemas de controle.

A figura 6.1 apresenta o diagrama em blocos do sistema de controle de processos SISO formado pelo processo, controlador PID e método de ajuste dos parâmetros de controle através de metaheurística.

Figura 6.1: Sintonia PID SISO através de otimização mono-objetivo.



Fonte: Elaborado pelo autor.

Para o estudo da sintonia de controladores PID para sistema monovariáveis foi utilizada a função de transferência do estudo realizado por Varol e Bingul (2004) dada por:

$$G(s) = \frac{(1-5s)}{(1+10s)(1+20s)}. \quad (6.1)$$

As análises comparativas levaram em consideração durante a validação apenas o índice de desempenho e nas comparações entre as abordagens de otimização (mono-objetivo e multiobjetivo) e os métodos de sintonia (convencional e inteligentes) os resultados das respostas do processo através de desempenhos no domínio do tempo (*overshoot*, tempo de subida e tempo de acomodação), índice de desempenho e robustez dos sistemas.

Os parâmetros de simulação utilizados para o Algoritmo Genético e *Simulated Annealing* estão definidos abaixo:

- Tempo de execução do processo para simulação: 250 s;
- Período de amostragem para determinação do índice de desempenho: 0,5 s;
- *Setpoint*: degrau unitário na referência a partir do tempo igual a zero;
- Faixa dos ganhos do controlador PID: $0 \leq K_p \leq 4,5$; $0 \leq K_i \leq 0,45$; $0 \leq K_d \leq 22,5$.

6.1.2 Experimentos dos algoritmos no controle de sistemas SISO

A primeira análise avaliou o desempenho de cada método de sintonia. Inicialmente foram comparados os dois tipos de otimização para a função objetivo das metaheurísticas empregadas: mono-objetivo e multiobjetivo.

Testes preliminares foram realizados para a definição da função objetivo; os índices utilizados foram o IAE, o ISE e o ITAE. Cada função foi executada trinta vezes no AG. Ao término das execuções foi calculada a média da função custo e o desvio padrão, e apresentados na tabela 6.1 juntamente com o tempo médio de simulação.

Tabela 6.1: Valores de custo e tempo de simulação dos AGs com diferentes FO.

Método de sintonia	Função custo (média)	Desvio padrão	Tempo médio de simulação (s)
AG IAE	24,93	1,23	120,1
AG ISE	19,51	0,65	120,6
AG ITAE	148,23	4,59	121,0

Fonte: Elaborado pelo autor.

Cabe salientar que para avaliar cada índice, foi determinado o seu conjunto de parâmetros PID e utilizados critérios de desempenho para fins de comparação. A tabela 6.2 apresenta os melhores resultados dos conjuntos de parâmetros do controlador PID para cada função custo testada.

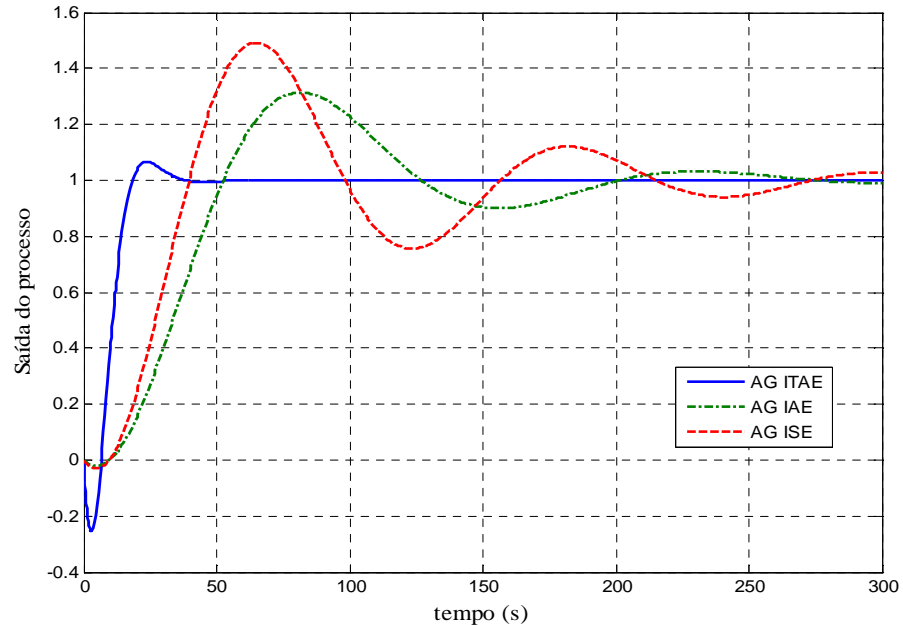
Tabela 6.2: Parâmetros de sintonia com diferentes índices de erro.

Método de sintonia	Parâmetros PID		
	K_p	K_i	K_d
AG IAE	0,3120	0,0449	0,0011
AG ISE	0,4951	0,0687	0,000786
AG ITAE	4,0258	0,0844	19,5183

Fonte: Elaborado pelo autor.

O gráfico 6.1 mostra a simulação da resposta do processo em malha fechada para cada controlador PID sintonizado.

Gráfico 6.1: Resposta do processo em malha fechada com diferentes índices de erro.



Fonte: Elaborado pelo autor.

Cada solução encontrada pela metaheurística foi avaliada pelos critérios no domínio do tempo (*overshoot*, tempo de subida, tempo de acomodação) conforme tabela 6.3.

Tabela 6.3: Desempenho do sistema sintonizado com diferentes índices de erro.

Método de sintonia	<i>Overshoot</i> (%)	Tempo de subida (s)	Tempo de acomodação (s)
AG IAE	31,40	31,0	252,0
AG ISE	49,19	22,0	264,0
AG ITAE	6,32	8,5	33,0

Fonte: Elaborado pelo autor.

Os índices IAE e ISE apresentaram altos valores de *overshoot* e tempos de subida e acomodação. Desta forma o índice escolhido para a função objetivo do AG e do SA foi o ITAE.

Um segundo teste preliminar foi realizado para estudar as influências dos operadores de cruzamento e mutação e do tamanho da população. Foram avaliados três tipos de operadores de cruzamento: Heurístico, Intermediário e Aritmético e três operadores de mutação foram testados: Gaussiana, Uniforme e *Adaptive Feasible* (implementação no MATLAB® semelhante à mutação *Creep*). Cada algoritmo AG foi executado trinta vezes utilizando a abordagem mono-objetivo.

A tabela 6.4 apresenta respectivamente os valores propostos para a taxa de cruzamento, taxa de mutação e tamanho da população.

Tabela 6.4: Relação de valores para as taxas de cruzamento e mutação e para o tamanho da população - processo SISO.

Taxa de cruzamento (%)	Taxa de mutação (%)	Tamanho da população
20, 40, 60, 80 e 90	1, 2, 5, 10 e 15	30, 50, 80 e 100

Fonte: Elaborado pelo autor.

Os valores das taxas de cruzamento e mutação da tabela 6.4 foram combinados entre si juntamente com os valores do tamanho da população e tipos de operadores de cruzamento e mutação gerando 360 conjuntos de parâmetros para o estudo de caso.

Ao término das execuções foram calculados e exibidos o valor médio da função custo, o desvio padrão e o número de gerações de cada simulação. Nesta análise, a melhor sintonia será aquela que apresentar o menor índice de erro calculado. As tabelas 6.5 e 6.6 apresentam os resultados do AG mono-objetivo com *crossover* aritmético, mutações uniforme e *Adaptive Feasible* com populações iguais a 30, 50, 80 e 100 respectivamente. As tabelas 6.7 e 6.8 apresentam os resultados do AG mono-objetivo com *crossover* intermediário, mutações uniforme e *Adaptive Feasible* com população N igual a 30 e 50 e N igual a 80 e 100 respectivamente. As tabelas 6.9 e 6.10 apresentam os resultados do AG mono-objetivo com *crossover* heurístico, mutações uniforme e *Adaptive Feasible* com população N igual a 30 e 50 e N igual a 80 e 100 respectivamente.

As colunas mostram o tamanho da população (primeira coluna) e a variação da taxa de mutação (terceira coluna) e as linhas apresentam a variação da taxa de cruzamento. Os resultados apresentados com base nos parâmetros mostrados são o valor médio obtido para a função objetivo, o desvio padrão em torno desta média e a geração onde ocorreu a estagnação do algoritmo. Os valores destacados mostram as melhores médias para cada tipo de cruzamento.

Tabela 6.5: Valores de custo e desvio-padrão dos AGs mono-objetivos com *crossover* aritmético, mutações uniforme e adaptativa com população $N = 30$ e $N = 50$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
30	20	164,03	151,61	162,32	156,19	156,02	153,28	Média
		10,22	6,24	9,76	9,87	9,97	9,35	Desvio
		18	25	22	31	29	43	Geração
	40	162,90	152,38	159,76	156,20	154,50	150,99	Média
		7,92	5,8	7,98	8,86	7,44	8,57	Desvio
50	20	17	18	27	27	25	42	Geração
		162,41	153,07	162,51	158,40	146,31	156,33	Média
	9,29	5,77	8,07	9,14	10,3	8,22	Desvio	
	9	19	22	27	23	37	Geração	
	60	160,45	153,20	157,46	156,60	157,39	157,68	Média
7,57		5,23	9,22	7,79	9,04	9,96	Desvio	
7		15	13	26	23	37	Geração	
30	90	159,99	151,79	159,92	159,56	159,70	157,01	Média
		7,77	5,61	7,71	6,92	9,72	8,62	Desvio
		9	11	14	12	15	31	Geração
	40	157,49	157,90	155,23	152,47	150,75	145,45	Média
		8,82	9,54	7,63	8,58	6,60	4,98	Desvio
21	27	26	27	32	46	Geração		
50	40	157,12	155,83	156,26	151,63	151,40	150,61	Média
		9,12	8,39	8,01	5,72	6,15	9,58	Desvio
	11	16	15	24	23	37	Geração	
	60	157,52	157,68	155,79	153,95	156,35	150,74	Média
		6,53	7,74	7,21	7,26	8,13	6,46	Desvio
9	15	20	23	24	35	Geração		
80	156,82	155,43	157,09	152,84	155,02	151,63	Média	
	6,58	5,88	7,33	6,30	8,10	5,81	Desvio	
	7	17	16	19	18	34	Geração	
90	155,27	158,39	156,18	154,79	154,11	152,81	Média	
	7,91	8,86	6,48	7,54	7,87	6,21	Desvio	
	7	7	8	10	19	30	Geração	

Fonte: Elaborado pelo autor.

Tabela 6.6: Valores de custo e desvio-padrão dos AGs mono-objetivos com *crossover* aritmético, mutações uniforme e adaptativa com população $N = 80$ e $N = 100$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
80	20	152,20	151,46	150,55	148,24	146,49	143,19	Média
		7,64	5,71	5,50	4,92	4,29	1,88	Desvio
		26	23	26	29	28	40	Geração
	40	154,53	154,00	150,92	149,89	147,96	145,70	Média
		7,60	6,56	5,82	5,59	3,62	3,89	Desvio
100	60	154,22	153,40	150,28	149,24	151,69	147,58	Média
		5,91	5,33	5,79	4,97	3,96	5,65	Desvio
	13	11	20	22	21	36	Geração	
	80	153,55	152,41	152,09	152,43	151,92	150,21	Média
		5,28	5,28	5,91	5,64	5,47	4,86	Desvio
8	12	19	16	19	28	Geração		
80	90	153,39	153,24	153,18	152,24	150,98	153,00	Média
		5,90	6,30	7,21	4,56	6,58	7,41	Desvio
		8	15	13	17	15	31	Geração
	20	151,57	151,61	148,38	150,46	146,75	142,30	Média
		5,74	6,24	4,93	7,98	3,83	1,86	Desvio
24	23	26	26	26	38	Geração		
100	40	150,48	152,38	150,63	149,75	147,58	144,67	Média
		4,73	5,8	6,09	5,99	4,85	4,42	Desvio
		18	19	22	26	25	38	Geração
	60	154,92	153,07	152,02	151,52	147,67	148,19	Média
		5,65	5,77	6,13	6,45	4,08	4,76	Desvio
15	19	20	24	23	32	Geração		
80	154,14	153,20	152,29	151,20	151,62	149,32	Média	
	5,24	5,23	5,12	5,18	5,66	4,86	Desvio	
	10	16	16	15	22	31	Geração	
90	153,16	151,79	150,28	150,83	151,96	149,88	Média	
	4,61	5,61	4,72	5,10	4,94	4,90	Desvio	
8	10	15	16	19	29	Geração		

Fonte: Elaborado pelo autor.

Tabela 6.7: Valores de custo e desvio-padrão dos AGs mono-objetivos com *crossover* intermediário, mutações uniforme e adaptativa com população $N = 30$ e $N = 50$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
30	20	165,58	161,47	158,74	158,36	153,68	151,29	Média
		9,90	8,79	9,96	10,70	8,71	9,47	Desvio
		18	25	29	31	29	42	Geração
	40	158,00	158,19	160,04	155,54	155,69	153,24	Média
		8,87	7,49	9,77	10,01	10,46	9,37	Desvio
		13	21	17	22	27	39	Geração
	60	158,63	158,41	159,68	155,46	158,03	154,79	Média
		7,73	8,06	9,91	8,63	8,38	8,62	Desvio
		9	11	19	16	18	35	Geração
	80	161,35	160,16	157,17	157,09	159,07	157,64	Média
8,62		8,00	5,86	7,44	7,78	7,76	Desvio	
11		10	10	15	15	32	Geração	
90	158,20	157,92	158,09	156,52	154,75	154,54	Média	
	8,77	7,77	7,06	6,15	6,76	7,51	Desvio	
	8	10	11	9	12	22	Geração	
50	20	153,98	153,55	154,90	153,00	150,56	146,23	Média
		8,08	8,53	9,43	8,09	7,48	6,36	Desvio
		20	18	30	24	25	38	Geração
	40	155,53	154,38	150,08	149,79	150,23	147,67	Média
		6,22	7,84	7,24	5,23	7,04	6,42	Desvio
		14	19	16	21	23	35	Geração
	60	155,26	152,63	155,39	153,80	151,26	150,12	Média
		7,81	6,43	7,69	6,32	6,34	6,81	Desvio
		11	12	13	18	23	34	Geração
	80	154,52	155,22	153,49	153,94	154,43	152,60	Média
8,12		7,59	6,15	5,90	6,48	6,48	Desvio	
9		13	10	16	13	29	Geração	
90	154,48	153,00	152,09	153,43	151,68	153,31	Média	
	5,41	4,86	5,82	7,09	5,59	6,54	Desvio	
	11	9	13	13	12	27	Geração	

Fonte: Elaborado pelo autor.

Tabela 6.8: Valores de custo e desvio-padrão dos AGs mono-objetivos com *crossover* intermediário, mutações uniforme e adaptativa com população $N = 80$ e $N = 100$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
80	20	150,78	150,35	150,95	148,18	146,20	143,65	Média
		5,85	6,15	7,24	5,68	3,50	3,07	Desvio
		15	17	24	27	27	40	Geração
	40	153,32	152,05	150,42	149,09	148,33	147,42	Média
		6,56	5,77	7,00	5,78	5,34	5,48	Desvio
		15	14	17	25	28	40	Geração
80	60	152,11	151,47	151,62	149,97	148,04	148,15	Média
		5,66	6,04	6,03	5,50	5,25	5,81	Desvio
		11	14	13	16	18	34	Geração
	80	151,37	152,19	149,97	151,46	149,50	149,82	Média
		4,92	6,25	5,58	5,89	5,20	5,52	Desvio
		11	12	12	15	19	28	Geração
100	20	151,91	152,04	150,28	151,77	151,08	150,26	Média
		6,20	5,59	4,85	5,73	5,62	4,99	Desvio
		10	11	12	12	15	27	Geração
	20	150,00	149,81	148,36	146,58	146,33	142,47	Média
		6,60	5,42	5,02	4,19	4,54	2,62	Desvio
18		20	21	24	26	39	Geração	
40		151,50	151,07	149,31	147,21	148,82	144,53	Média
		5,08	5,50	6,06	4,75	4,46	2,99	Desvio
	12	13	18	16	23	31	Geração	
60	150,86	150,18	149,58	150,38	148,43	147,40	Média	
	5,01	5,19	5,39	6,21	4,38	4,09	Desvio	
	11	13	13	15	20	34	Geração	
	80	148,29	149,50	149,93	149,34	149,58	148,90	Média
		4,32	4,38	5,19	6,40	4,11	3,91	Desvio
12		11	13	14	14	29	Geração	
90	150,44	150,81	150,93	148,78	150,23	150,37	Média	
	4,31	4,06	4,07	5,04	5,19	6,04	Desvio	
	11	11	12	13	15	23	Geração	

Fonte: Elaborado pelo autor.

Tabela 6.9: Valores de custo e desvio-padrão dos AGs mono-objetivos com *crossover* heurístico, mutações uniforme e adaptativa com população $N = 30$ e $N = 50$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
30	20	160,68	161,81	158,09	156,98	153,94	150,21	Média
		10,59	12,15	11,74	11,98	11,62	10,68	Desvio
		23	28	31	29	34	42	Geração
	40	160,85	161,21	159,33	154,72	154,84	148,01	Média
		14,31	10,45	12,52	11,69	11,35	8,50	Desvio
50	60	162,60	157,03	154,97	155,56	153,19	153,11	Média
		13,89	11,06	10,44	10,60	10,02	9,62	Desvio
	80	19	18	25	26	31	41	Geração
		163,70	160,07	157,83	159,94	155,09	150,84	Média
		14,36	12,41	10,41	12,32	10,94	10,32	Desvio
30	90	12	17	17	22	27	38	Geração
		158,29	159,81	158,69	158,32	156,42	157,39	Média
	20	10,27	13,47	12,16	12,40	10,80	10,60	Desvio
		9	16	15	18	24	31	Geração
		152,62	154,80	149,98	153,10	146,74	144,23	Média
50	40	8,61	8,96	7,88	10,07	3,90	5,85	Desvio
		23	24	32	31	34	41	Geração
	60	152,99	152,16	148,87	148,85	148,12	143,53	Média
		8,34	8,44	7,75	6,57	7,65	3,58	Desvio
		20	20	33	23	28	32	Geração
80	154,00	150,37	148,90	148,58	151,16	145,78	Média	
	9,76	8,90	8,21	6,65	9,26	5,94	Desvio	
	17	19	19	32	22	36	Geração	
50	90	150,43	149,39	152,36	150,47	148,02	147,74	Média
		7,52	7,27	10,39	9,28	7,60	6,92	Desvio
	20	11	17	16	25	21	35	Geração
		151,77	151,02	152,22	150,67	149,49	146,40	Média
		9,77	10,37	9,43	8,35	7,46	5,47	Desvio
40	11	13	17	18	23	29	Geração	

Fonte: Elaborado pelo autor.

Tabela 6.10: Valores de custo e desvio-padrão dos AGs mono-objetivos com *crossover* heurístico, mutações uniforme e adaptativa com população $N = 80$ e $N = 100$.

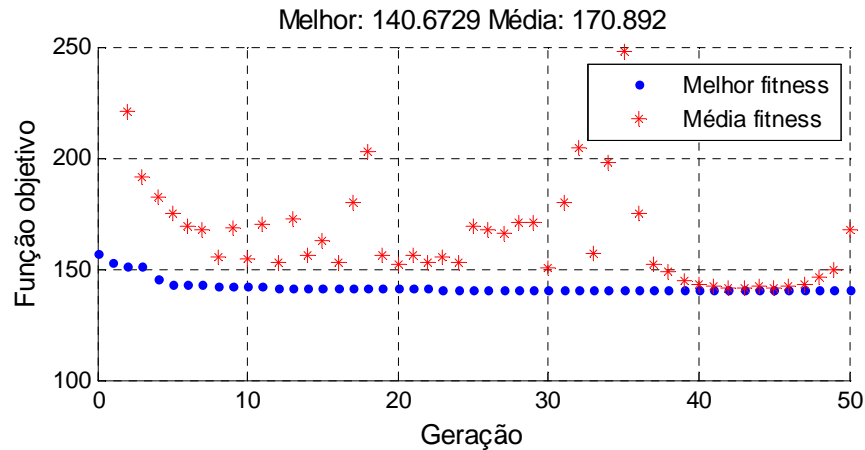
N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
80	20	150,38	150,13	147,28	148,22	144,35	141,79	Média
		9,18	7,48	7,31	8,96	3,54	3,05	Desvio
		24	25	32	30	34	37	Geração
	40	150,46	147,64	146,72	144,94	144,09	142,66	Média
		6,68	5,70	5,95	3,47	3,31	3,38	Desvio
		21	22	23	25	26	34	Geração
	60	147,20	146,48	145,84	144,84	145,33	142,58	Média
		4,53	5,03	6,32	3,96	5,26	3,00	Desvio
		14	17	22	14	24	23	Geração
	80	146,77	145,43	144,69	146,23	144,73	144,55	Média
6,71		3,75	3,15	6,23	6,01	6,09	Desvio	
15		13	20	17	18	27	Geração	
90	148,25	146,85	144,95	148,58	144,90	142,87	Média	
	7,21	5,25	4,34	7,45	3,89	3,17	Desvio	
	10	13	19	17	17	30	Geração	
100	20	146,92	146,67	144,99	143,92	143,38	141,10	Média
		6,74	6,34	3,89	5,92	2,77	0,83	Desvio
		20	26	33	31	32	37	Geração
	40	146,20	145,15	144,67	143,70	143,37	141,20	Média
		4,54	3,34	3,69	1,94	2,83	1,46	Desvio
		21	21	23	25	27	27	Geração
	60	145,74	147,11	147,13	143,41	142,36	141,91	Média
		4,76	5,82	8,52	3,43	1,34	1,51	Desvio
		17	23	22	23	26	23	Geração
	80	145,07	144,84	146,31	144,89	143,76	142,76	Média
3,76		5,26	7,00	3,73	2,77	4,06	Desvio	
16		15	18	20	17	26	Geração	
90	145,65	144,32	145,60	144,02	143,43	143,78	Média	
	4,56	3,95	5,74	3,93	2,81	4,68	Desvio	
	13	13	17	18	17	23	Geração	

Fonte: Elaborado pelo autor.

Observou-se nas simulações realizadas que a mutação gaussiana viola as restrições dos ganhos do controlador PID e desta forma seus valores não foram inseridos nas tabelas.

O gráfico 6.2 ilustra um exemplo da convergência da metaheurística AG empregada para o problema da sintonia de controladores PID em processos SISO.

Gráfico 6.2: Convergência do Algoritmo Genético.

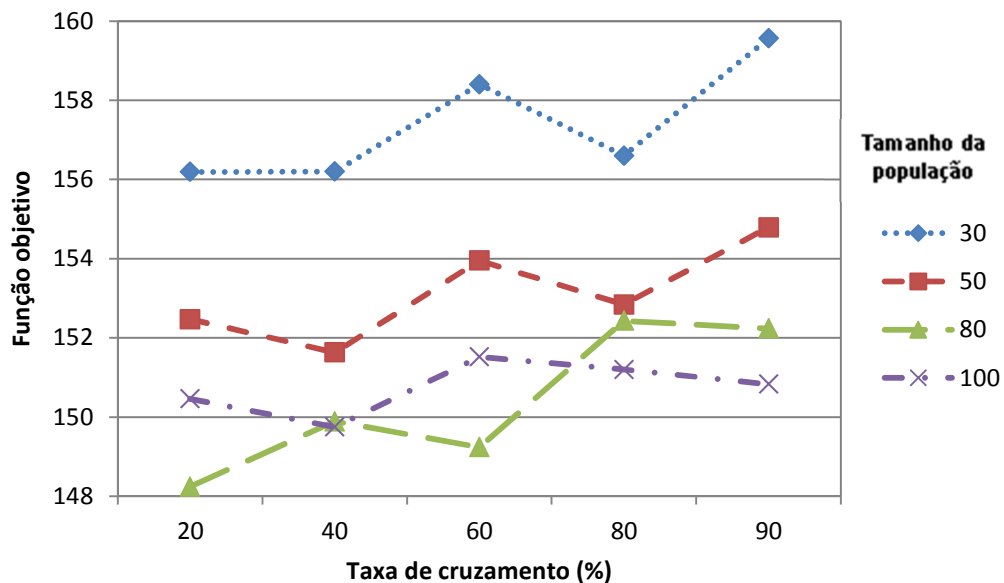


Fonte: Elaborado pelo autor.

Observam-se melhorias significativas nas primeiras gerações (5 a 10) e pequenas variações entre as gerações 20 e 40, conforme tabelas anteriores.

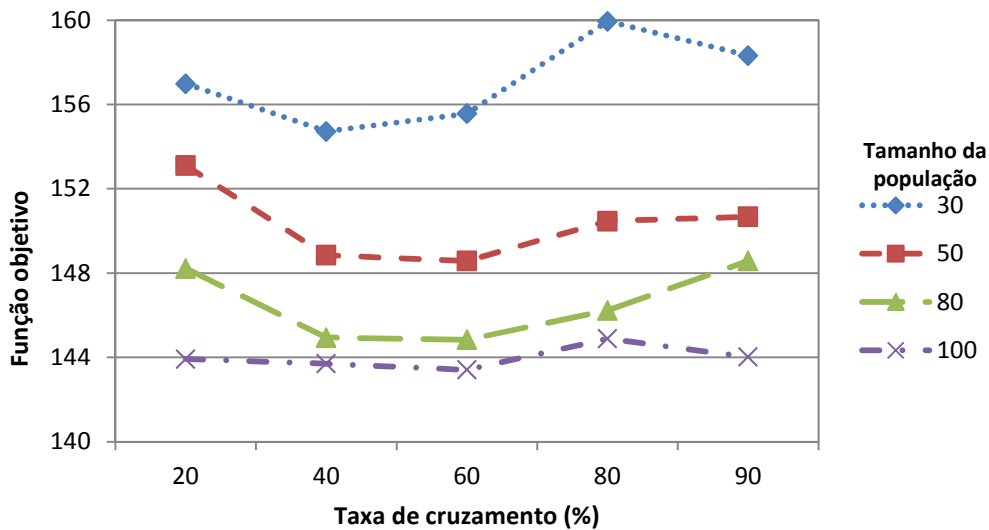
Os dois próximos gráficos (6.3 e 6.4) mostram a função custo em função da taxa de cruzamento para diferentes tamanhos da população. O primeiro gráfico com cruzamento aritmético e o segundo com cruzamento heurístico e ambos com taxa de mutação de 10%.

Gráfico 6.3: Tamanho da população no AG com cruzamento aritmético.



Fonte: Elaborado pelo autor.

Gráfico 6.4: Tamanho da população no AG com cruzamento heurístico.

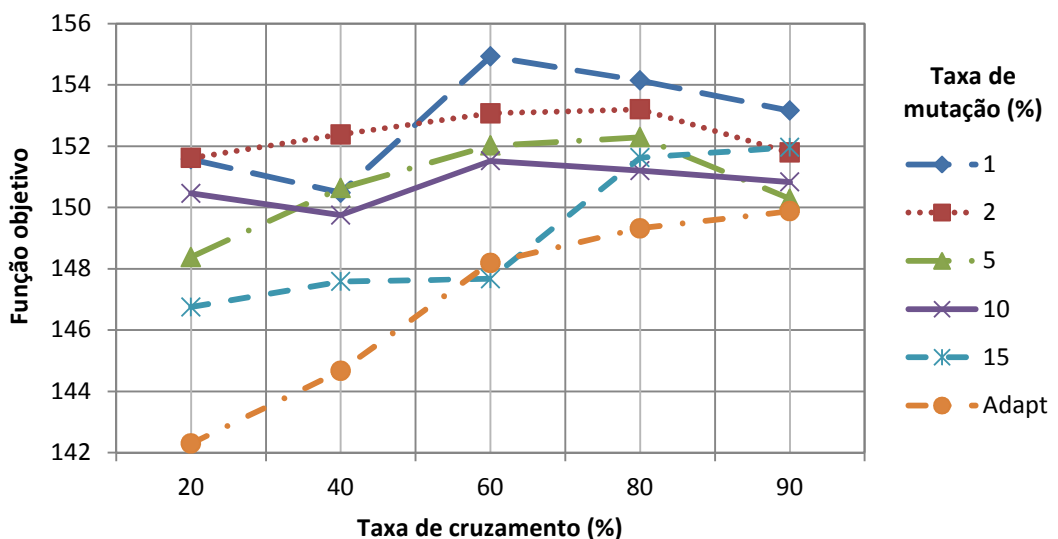


Fonte: Elaborado pelo autor.

Os resultados apresentados nos gráficos acima apontam que as configurações com maior população apresentam os melhores valores quanto à função custo. O aumento da população aumenta sua diversidade e reduz a probabilidade de convergência prematura de um AG para um ponto ótimo local. Em contrapartida, este aumento também produz um aumento significativo no tempo computacional. Observou-se ainda que as diferenças das funções custo entre as populações 80 e 100 foram menos significativas, ocorrendo até uma instabilidade entre os valores no cruzamento aritmético. Ainda assim, foi escolhida a configuração com população igual a 100 para o AG mono-objetivo e para o AG multiobjetivo.

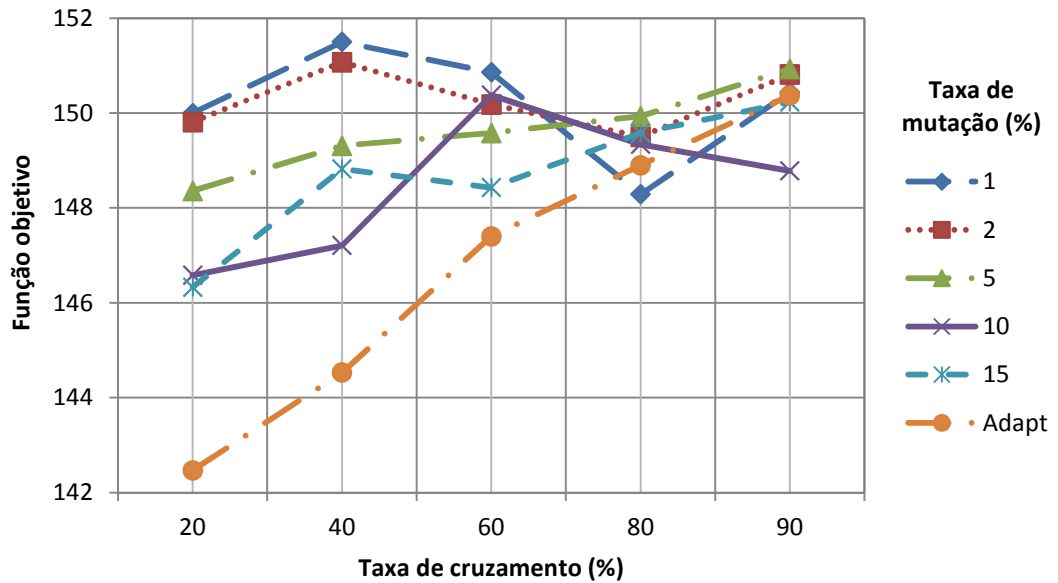
Com base nos dados das tabelas 6.5 a 6.10, os gráficos 6.5, 6.6 e 6.7 apresentam os comportamentos das médias das funções objetivos para cada tipo de cruzamento. No eixo vertical mostra-se o valor da função objetivo e no eixo horizontal as taxas de cruzamento. As linhas mostram os diferentes valores analisados para a taxa de mutação.

Gráfico 6.5: Cruzamento aritmético com população = 100.



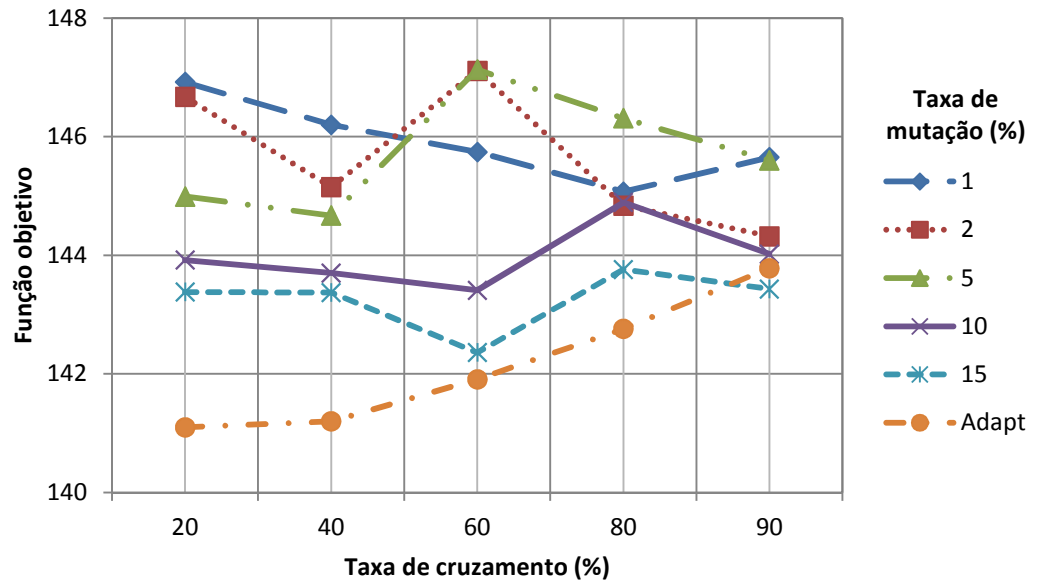
Fonte: Elaborado pelo autor.

Gráfico 6.6: Cruzamento intermediário com população = 100.



Fonte: Elaborado pelo autor.

Gráfico 6.7: Cruzamento heurístico com população = 100.



Fonte: Elaborado pelo autor.

Verifica-se que os melhores resultados são obtidos, na maior parte, com altas taxas de mutação e baixas taxas de cruzamento. Estes comportamentos se devem à complexidade do problema, fazendo com que o aumento da probabilidade de cruzamento se reflita no aumento da possibilidade da quebra de boas soluções e o aumento da taxa de mutação na diversificação da população pela busca de melhores soluções.

Observa-se ainda nas tabelas anteriores que o melhor resultado e melhor dos “piores” resultados quanto à média da função custo são obtidos com o cruzamento heurístico. Dentre as avaliações realizadas, o cruzamento heurístico com taxa de 0,2 em combinação com a

mutação *Adaptive Feasible* apresentou os melhores resultados quanto à função custo. Esta mutação tem a capacidade de realizar a manutenção adequada da diversidade da população, direcionando a busca para regiões ainda não exploradas e ao mesmo tempo, quando necessário, consegue manter a característica de diversificação. Desta forma é a configuração escolhida para a metaheurística AG na determinação dos parâmetros PID.

O terceiro experimento foi realizado para determinar a temperatura inicial e as funções de resfriamento e geração de novas soluções do *Simulated Annealing*. As funções de resfriamento testadas foram: exponencial, logarítmica e linear (*Fast Cauchy*). As funções testadas para a geração de novas soluções foram: *Boltzman Annealing*, descrita na equação (4.5) e *Fast Annealing*, descrita na equação (4.6).

Três temperaturas iniciais são propostas para análise de sua influência no algoritmo. Os valores são apresentados na tabela 6.11.

Tabela 6.11: Valores para temperatura inicial.

Temperatura inicial
50
100
200

Fonte: Elaborado pelo autor.

Os valores da tabela anterior são combinados com as funções de resfriamento e geração de novas soluções gerando 18 conjuntos de parâmetros para o estudo de caso.

Para a determinação da solução inicial do método SA foi utilizado um gerador aleatório de solução. Cada algoritmo foi executado trinta vezes utilizando a abordagem mono-objetivo. Ao término das execuções foram calculados e exibidos o valor médio da função custo e o desvio padrão.

Nesta análise, a melhor sintonia será aquela que apresentar o menor índice de erro calculado. A tabela 6.12 apresenta os resultados do SA mono-objetivo para diferentes funções de resfriamento e geração de novas soluções (função *Annealing*).

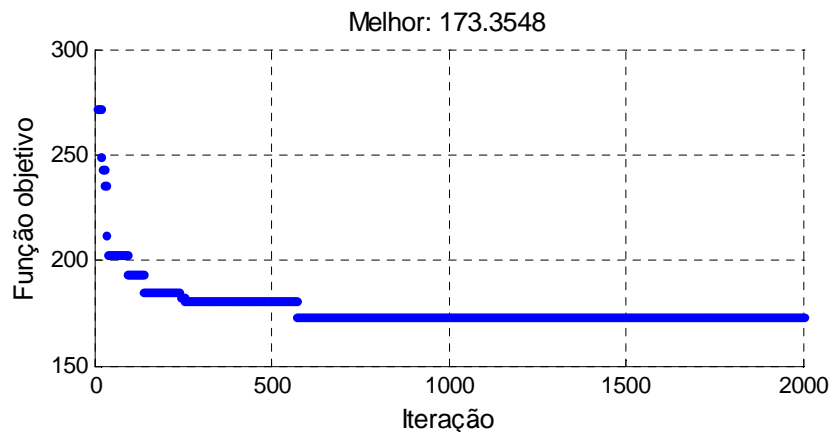
As colunas mostram a temperatura inicial (primeira coluna) e a função de resfriamento (terceira coluna) e as linhas apresentam a função *annealing*. Os resultados apresentados são o valor médio obtido para a função objetivo e o desvio padrão em torno desta média, com base nos parâmetros mostrados.

Tabela 6.12: Valores de custo dos SAs mono-objetivos

Temp inicial	Função <i>Annealing</i>	Função de resfriamento			Parâmetros de avaliação
		Exponencial	Logarítmica	Linear	
50	<i>Fast Annealing</i>	170,50	168,37	162,56	Média
		11,50	10,18	12,31	Desvio
	<i>Boltzman Annealing</i>	170,26	162,52	170,95	Média
		12,12	12,18	10,44	Desvio
100	<i>Fast Annealing</i>	168,27	166,96	160,88	Média
		11,28	11,03	14,12	Desvio
	<i>Boltzman Annealing</i>	169,53	161,41	163,17	Média
		12,55	11,56	13,04	Desvio
200	<i>Fast Annealing</i>	168,59	163,66	161,28	Média
		12,02	8,23	14,15	Desvio
	<i>Boltzman Annealing</i>	166,58	160,68	161,47	Média
		13,80	9,89	13,45	Desvio

Fonte: Elaborado pelo autor.

O gráfico 6.8 ilustra a convergência da metaheurística SA empregada para o problema da sintonia do controlador PID.

Gráfico 6.8: Convergência do *Simulated Annealing*.

Fonte: Elaborado pelo autor.

Dentre as avaliações realizadas, a função de resfriamento logarítmica, com temperatura inicial 200 em combinação com a função de geração de novas soluções *Boltzman Annealing*, apresentou os melhores resultados quanto à função custo e desta forma é a configuração escolhida para a metaheurística SA na determinação dos parâmetros PID.

O quarto teste preliminar avaliou o desempenho das duas metaheurísticas empregadas. A tabela 6.13 apresenta a média da função custo e o desvio padrão juntamente com o melhor e o pior valor absoluto.

Tabela 6.13: Valores de custo das metaheurísticas.

Método de sintonia	Função custo (média)	Desvio padrão	Função custo (melhor)	Função custo (pior)
AG	141,10	0,83	139,39	144,22
SA	160,68	9,89	145,84	179,14

Fonte: Elaborado pelo autor.

O conjunto de parâmetros do controlador PID determinado por para cada metaheurística está relacionado aos resultados da média da função custo, ou seja, foi escolhido o resultado entre as trinta execuções de cada metaheurística que apresentou o valor da função custo mais próximo da média das execuções. Os valores são apresentados na tabela 6.14.

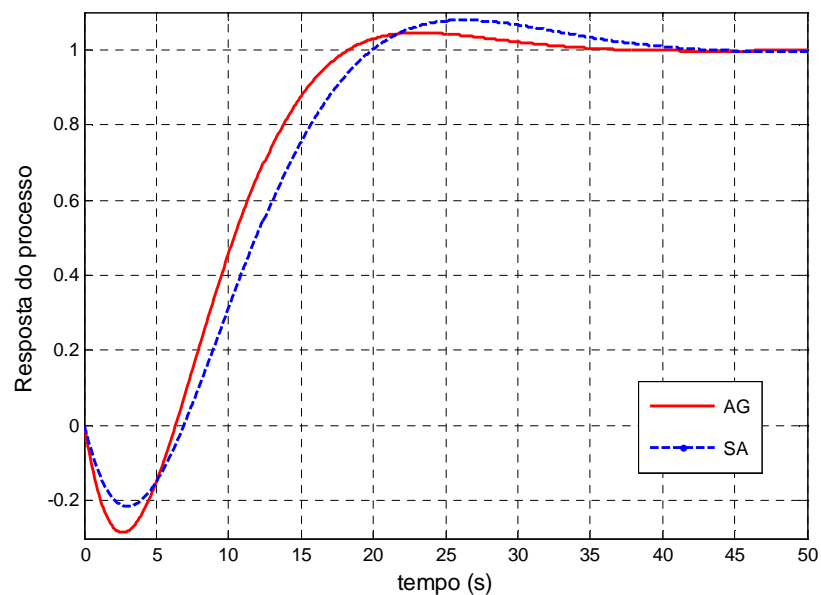
Tabela 6.14: Parâmetros de sintonia com metaheurísticas.

Método de sintonia	Parâmetros PID		
	K_p	K_i	K_d
AG	4,3299	0,0856	21,9300
SA	3,6079	0,0807	17,1420

Fonte: Elaborado pelo autor.

O gráfico 6.9 mostra a simulação da resposta do processo em malha fechada para cada controlador PID sintonizado.

Gráfico 6.9: Resposta do processo em malha fechada com controladores PID sintonizados por metaheurísticas.



Fonte: Elaborado pelo autor.

Cada solução encontrada pelas metaheurísticas foi avaliada pelos critérios no domínio do tempo (*overshoot*, tempo de subida, tempo de acomodação) e através do índice de desempenho IAE conforme tabela 6.15.

Tabela 6.15: Desempenho do sistema sintonizado com metaheurísticas.

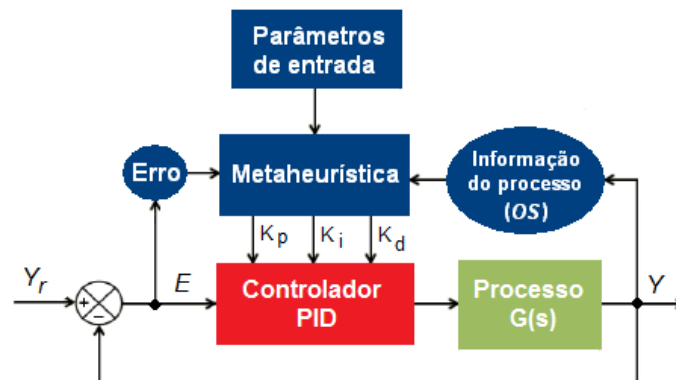
Método de sintonia	Overshoot (%)	Tempo de subida (s)	Tempo de acomodação (s)	Índice de erro IAE
AG	4,56	8,32	28,89	25,54
SA	7,87	9,46	36,49	29,08

Fonte: Elaborado pelo autor.

Percebe-se na tabela 6.15 que sintonia empregando a metaheurística Algoritmos Genéticos apresenta resultados superiores em todos os critérios no domínio do tempo.

A próxima análise foi feita com relação às duas formas de otimização: mono-objetivo e multiobjetivo. Para a otimização mono-objetivo foram utilizados os experimentos anteriores com AG. Para a otimização multiobjetivo foi utilizada a abordagem com frentes de Pareto. Como função objetivo, levou-se em consideração o *overshoot* (objetivo 1) e o índice ITAE (objetivo 2). A figura 6.2 apresenta o diagrama em blocos do sistema.

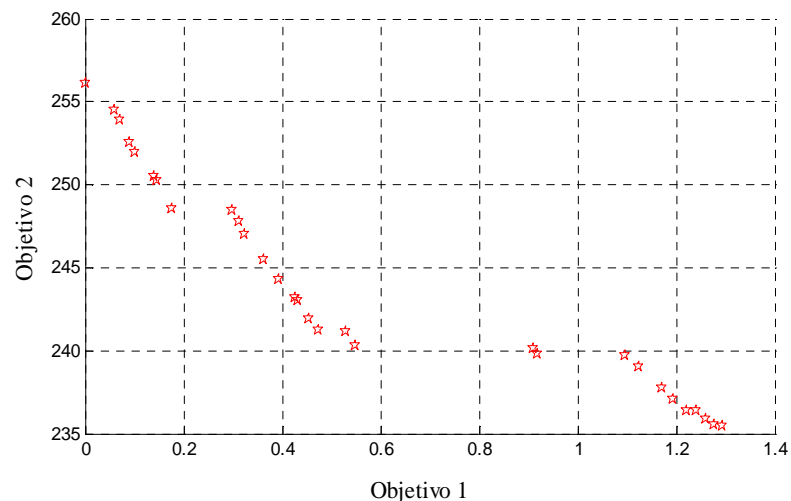
Figura 6.2: Sintonia PID SISO através de otimização multiobjetivo.



Fonte: Elaborado pelo autor.

No gráfico 6.10 é apresentada a fronteira encontrada com o AG empregado.

Gráfico 6.10: Fronteira de Pareto com a metaheurística AG empregada.



Fonte: Elaborado pelo autor.

Como tomador de decisão para a escolha da solução ótima dos parâmetros, foi utilizado um procedimento que classifica as soluções quanto à robustez, indicando as soluções mais comprometidas com este quesito. O procedimento para determinação da solução está descrito abaixo (HU et al., 2011):

- Determinar o intervalo dos parâmetros do objeto pesquisado, e eleger um modelo de amostragem aleatória;
- Repetir a experiência de simulação N vezes, e obter uma coleção bidimensional de desempenho incluindo o tempo de acomodação e *overshoot*.
- Comparar os efeitos de cada solução e a dispersão dos índices de desempenho dos resultados da experiência Monte Carlo. Quanto menor o intervalo do desempenho do sistema de controle, mais forte é a robustez do sistema de controle.

Dado o modelo da planta:

$$G_{ij}(s) = (K_{ij}/T_{ij}s + 1)e^{-\theta_{ij}s}, i, j = 1, 2, \quad (6.2)$$

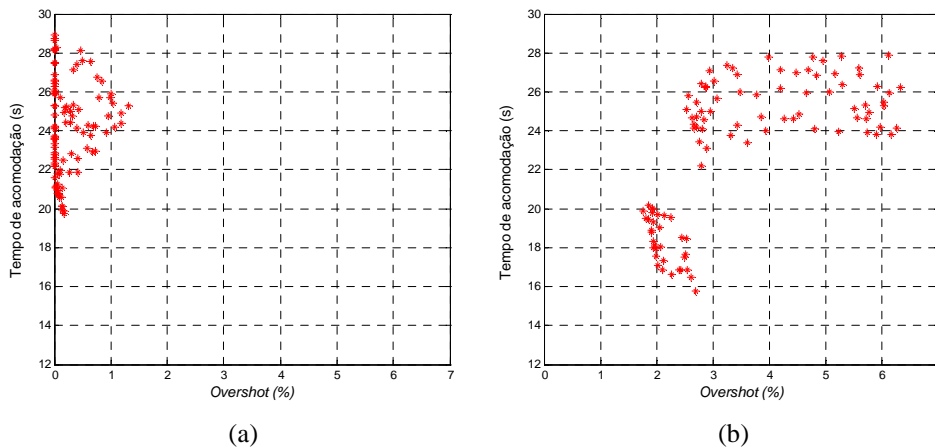
e assumindo que o intervalo de perturbação dos parâmetros (K_{ij} , T_{ij} , θ_{ij}) do modelo é de $\pm 10\%$ e o número de experimentos estocásticos H é igual a 30, foi utilizada uma adaptação do método do produto ponderado (WPM - *Weighted Product Method*), dado pela equação abaixo (POHEKAR; RAMACHANDRAN, 2004):

$$WPM(A_K / A_L) = \prod_{j=1}^Q (a_{Kj} / a_{Lj})^{w_j}, \quad (6.3)$$

onde a_{ij} é o valor do critério j ao utilizar a i -ésima alternativa, w_j é o peso (importância) do critério j e Q é o número de critérios de desempenho utilizados.

Este índice gerado a partir do método WPM compara as soluções A_K e A_L . Se $WPM(A_K/A_L)$ for menor que um, a solução A_K é melhor que a solução A_L . Logo, a melhor solução é aquela cujo valor WPM é menor ou igual em relação a todas as outras soluções (POHEKAR; RAMACHANDRAN, 2004). Foram comparadas apenas as soluções com *overshoot* abaixo de 20%. O gráfico 6.11 apresenta a coleção bidimensional de desempenho (tempo de acomodação e *overshoot*) da melhor e pior solução.

Gráfico 6.11: Resposta do processo com a melhor (a) e pior solução (b).



Os valores da melhor e da pior solução estão associados à combinação dos parâmetros PID (K_p , K_i , K_d). Na melhor solução, observam-se os menores valores e o menor intervalo de *overshoot* enquanto que a pior solução apresenta menores valores de tempo de acomodação, mas um maior intervalo deste valor evidenciando uma menor robustez do sistema de controle, de acordo com o método empregado.

A metaheurística e abordagens utilizadas são apresentadas como:

- **AG Mono:** Algoritmo Genético mono-objetivo;
- **AG Pareto:** Algoritmo Genético multiobjetivo com frentes de Pareto.

A tabela 6.16 apresenta os parâmetros obtidos com cada abordagem.

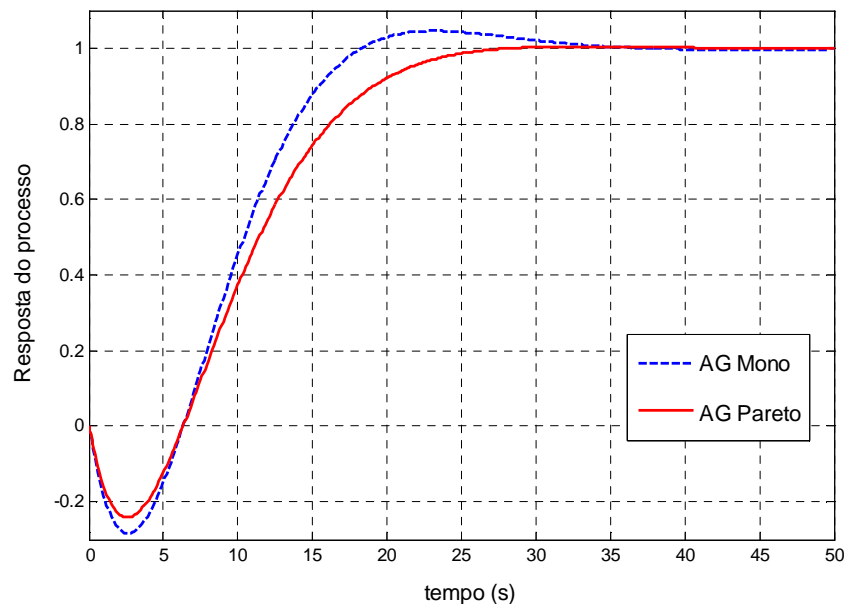
Tabela 6.16: Parâmetros de sintonia com diferentes abordagens.

Método de sintonia	Parâmetros PID		
	K_p	K_i	K_d
AG Mono	4,3299	0,0856	21,9300
AG Pareto	3,7767	0,0745	22,2167

Fonte: Elaborado pelo autor.

O gráfico 6.12 apresenta a resposta do processo em malha fechada para cada controlador PID sintonizado.

Gráfico 6.12: Resposta do processo em malha fechada com diferentes abordagens.



Fonte: Elaborado pelo autor.

Cada abordagem foi avaliada novamente pelos critérios no domínio do tempo (*overshoot*, tempo de subida, tempo de acomodação) e através do índice de desempenho IAE conforme tabela 6.17.

Tabela 6.17: Desempenho do sistema sintonizado com diferentes abordagens.

Método de sintonia	<i>Overshoot</i> (%)	Tempo de subida (s)	Tempo de acomodação (s)	Índice de erro IAE
AG Mono	4,56	8,32	28,89	25,54
AG Pareto	0	11,81	23,62	27,39

Fonte: Elaborado pelo autor.

Observa-se na tabela 6.16 que a otimização multiobjetivo (Pareto) apresentou uma redução nos valores dos parâmetros K_p e K_i em comparação aos valores da abordagem mono-objetivo. Isto ocasionou uma redução no seu valor de *overshoot*, mostrado na tabela 6.17. Além disso, percebeu-se que esta otimização apresentou também menor valor de tempo de acomodação se comparado com a mono-objetivo.

Desta forma, devido ao valor de *overshoot* ligeiramente menor, foi escolhida a otimização multiobjetivo com frentes de Pareto para fins de comparação com o Algoritmo Colônia de Formigas (ACO), proposto por Varol e Bingul (2004), e o método convencional de sintonia de Ziegler e Nichols.

6.1.3 Desempenho das metaheurísticas no sistema monovariável

Para este estudo os objetivos a serem atingidos pelo sistema de controle são:

- Erro nulo de regime permanente;
- Desempenhos no domínio do tempo superiores ao método Ziegler-Nichols;
- Resultados similares aos obtidos no trabalho de Varol e Bingul (2004).

A tabela 6.18 apresenta os parâmetros obtidos com cada método de sintonia.

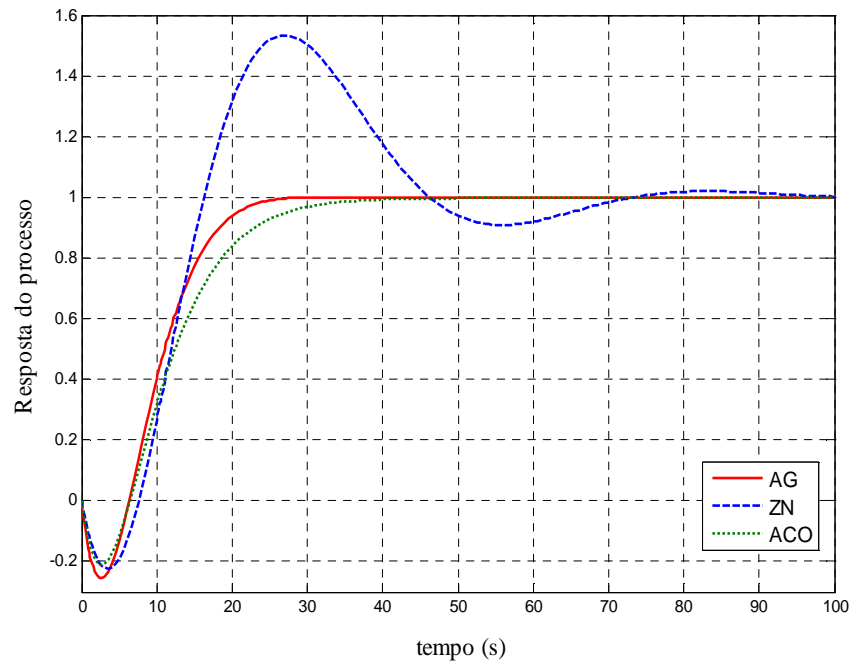
Tabela 6.18: Parâmetros de sintonia pelos métodos ZN e metaheurísticas.

Método de sintonia	Parâmetros PID		
	K_p	K_i	K_d
AG	3,7767	0,0745	22,2167
ZN	3,5290	0,2100	14,8230
ACO	3,3358	0,0661	21,7854

Fonte: Elaborado pelo autor.

O gráfico 6.13 mostra a simulação da resposta do processo em malha fechada para cada controlador PID sintonizado.

Gráfico 6.13: Resposta do processo em malha fechada com controladores PID sintonizados pelos métodos ZN e metaheurísticas.



Fonte: Elaborado pelo autor.

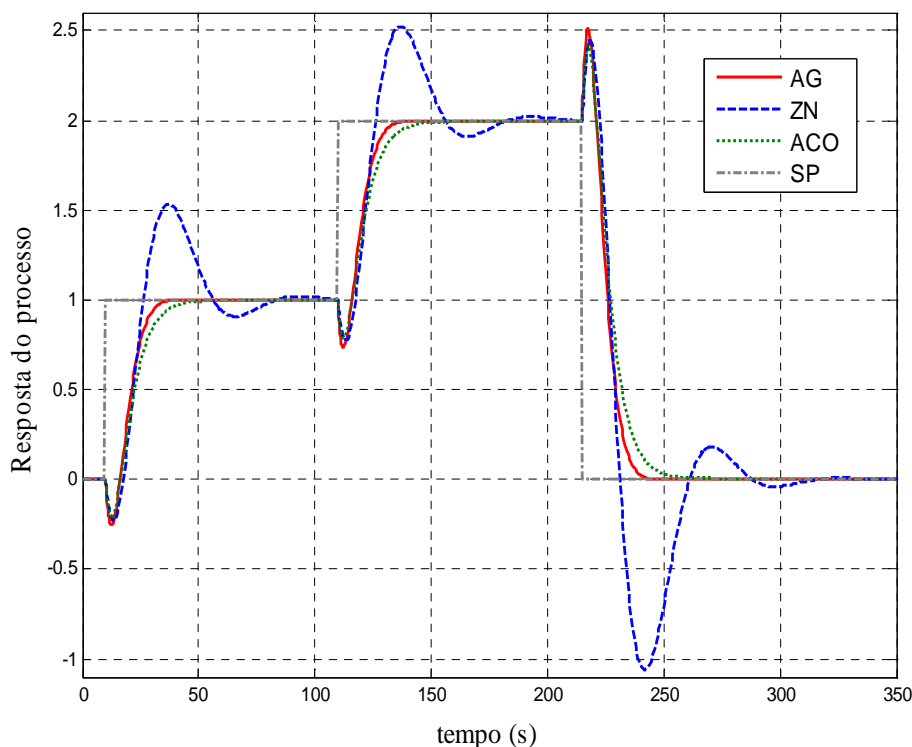
Cada solução encontrada pelas metaheurísticas foi avaliada novamente pelos critérios no domínio do tempo (*overshoot*, tempo de subida, tempo de acomodação) e através do índice de desempenho IAE conforme tabela 6.19.

Tabela 6.19: Desempenho do sistema com os métodos ZN e metaheurísticas.

Método de sintonia	<i>Overshoot</i> (%)	Tempo de subida (s)	Tempo de acomodação (s)	Índice de erro IAE
AG	0	11,81	23,62	27,39
ZN	53,3	7,0	83,50	49,34
ACO	0	14,97	31,70	30,72

Fonte: Elaborado pelo autor.

O gráfico 6.14 apresenta as respostas da variável controlada para diferentes pontos de operação (*setpoint* - SP) na forma de degraus ascendentes e descendentes.

Gráfico 6.14: Seguimento da referência com os métodos ZN e metaheurísticas.

Fonte: Elaborado pelo autor.

Com relação ao resultado das simulações, pode-se afirmar que a metaheurística empregada apresentou tempos de subida e acomodação e índice de desempenho melhores que o algoritmo de Varol e Bingul (2004) e um seguimento da referência dentro do esperado. Observa-se ainda a relação entre o tempo de subida e o *overshoot* em cada método de sintonia, onde, por exemplo, o método de Ziegler-Nichols apresenta o menor t_R , mas o maior *overshoot* e a metaheurística ACO o inverso. Esta relação é explicada por Bazanella e Gomes da Silva (2005), onde em muitos casos, à medida que se reduz o valor de t_R a partir da sintonia de controladores PID, pode-se obter um aumento do *overshoot*.

6.1.4 Robustez do sistema monovariável

A segunda análise avaliou a robustez do sistema com cada método de sintonia. Neste caso analisaram-se os critérios Margem de Ganho e Margem de Fase (tabela 6.20).

Tabela 6.20: Robustez do sistema com os métodos ZN e metaheurísticas.

Método de sintonia	Margem de Ganho	Margem de Fase (°)
ZN	8,63	31,7
AG	5,11	50,5
ACO	5,28	57,5

Fonte: Elaborado pelo autor.

Os resultados apontam também a qualidade das respostas da metaheurística quanto à robustez, pois o método apresentou Margem de Ganho e Margem de Fase adequados aos limites ($MG > 1,7$ e $MF > 30^\circ$).

Simulações também foram realizadas para avaliar a robustez dos métodos de sintonia propostos conforme Varol e Bingul (2004). Foram três alterações na planta avaliando individualmente cada efeito da resposta do sistema.

Primeiramente a função de transferência do processo foi alterada conforme a equação (6.4), incrementando em 50% o ganho em regime estacionário:

$$G_1(s) = \frac{1,5(1-5s)}{(1+10s)(1+20s)}. \quad (6.4)$$

A segunda alteração da planta é a mudança de um dos polos do processo, como mostrado na equação (6.5).

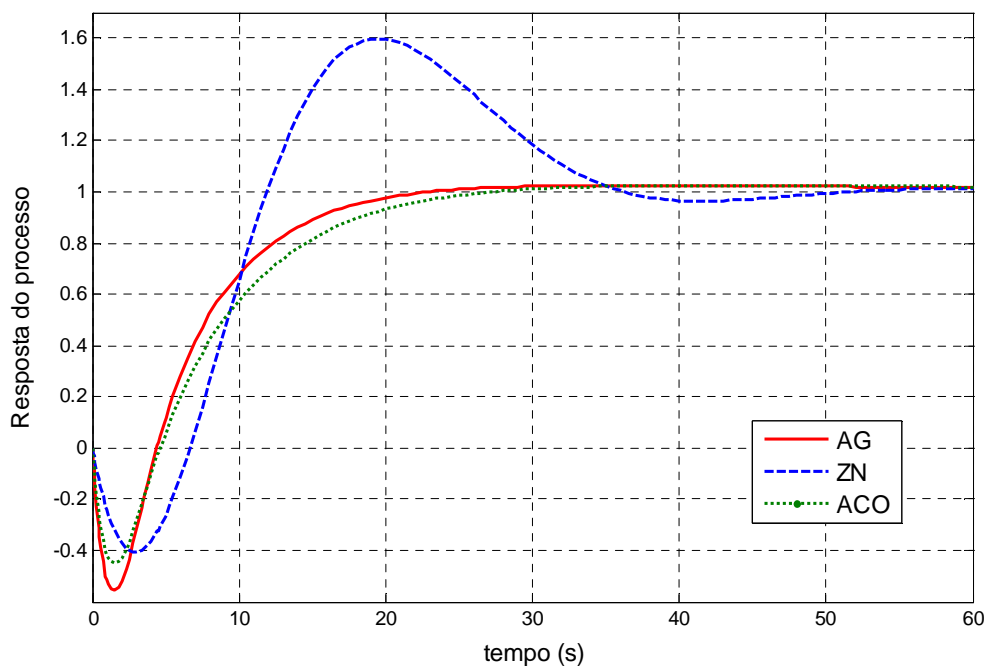
$$G_2(s) = \frac{(1-5s)}{(1+10s)(1+25s)}. \quad (6.5)$$

A terceira e última mudança do sistema para avaliação do desempenho é a inserção de um atraso temporal na planta. A equação do processo é apresentada abaixo.

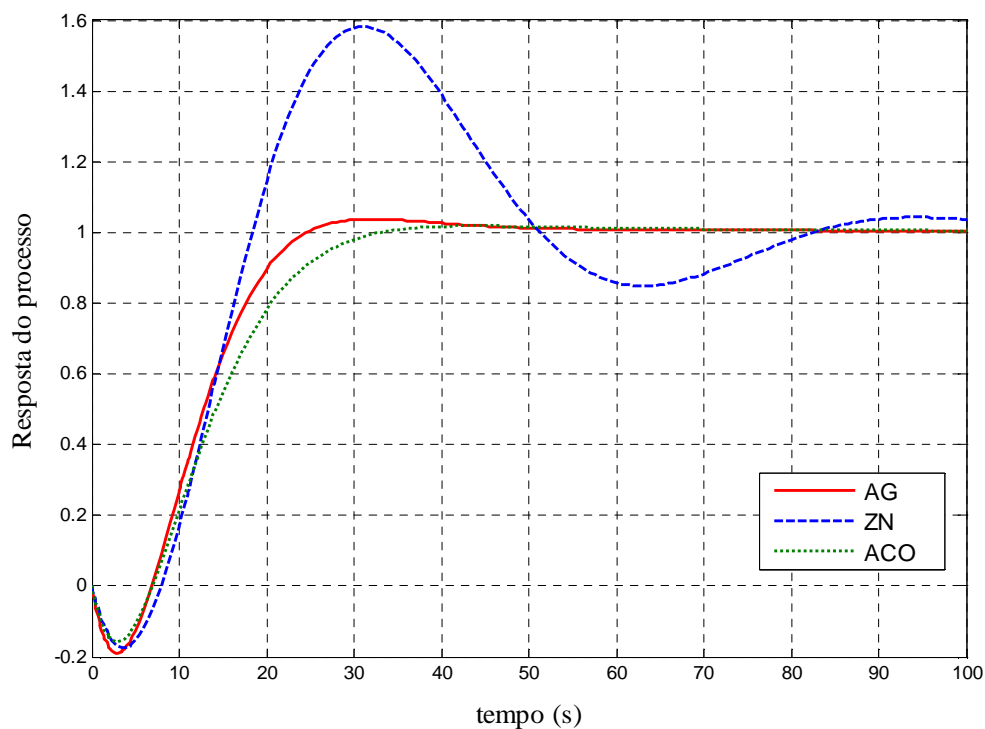
$$G_3(s) = \frac{(1-5s)}{(1+10s)(1+25s)} e^{-1,5s}. \quad (6.6)$$

Os gráficos 6.15, 6.16 e 6.17 apresentam a resposta do processo em malha fechada para cada alteração realizada no sistema.

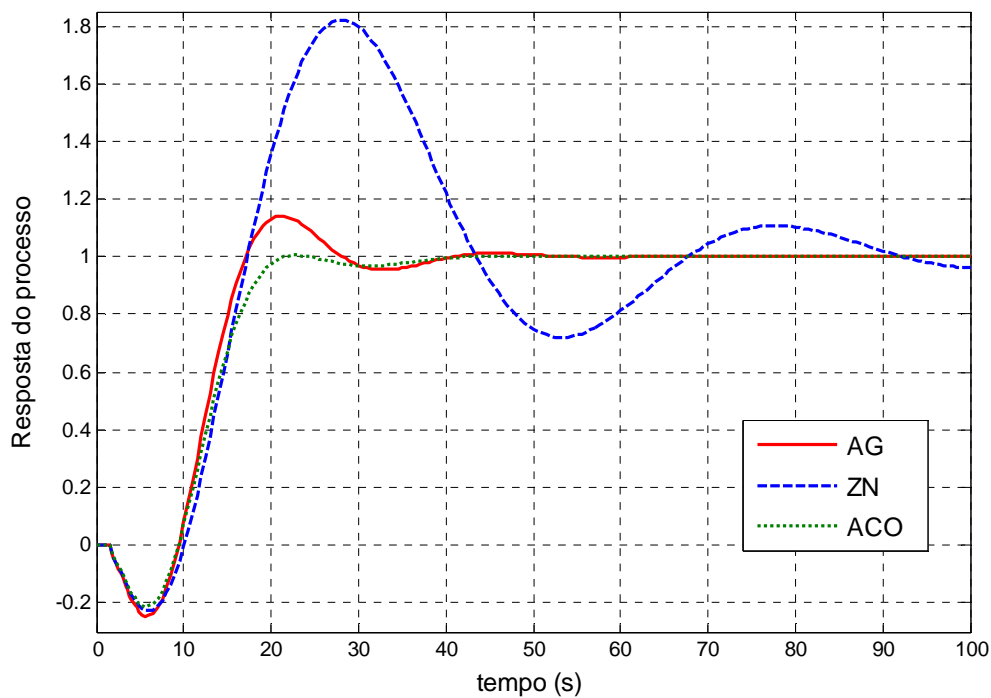
Gráfico 6.15: Resposta do processo SISO em malha fechada com $G_1(s)$.



Fonte: Elaborado pelo autor.

Gráfico 6.16: Resposta do processo SISO em malha fechada com $G_2(s)$.

Fonte: Elaborado pelo autor.

Gráfico 6.17: Resposta do processo SISO em malha fechada com $G_3(s)$.

Fonte: Elaborado pelo autor.

A tabela 6.21 apresenta o desempenho do processo com os métodos ZN e metaheurísticas.

Tabela 6.21: Desempenho do processo com alterações de parâmetros do sistema.

	Método de sintonia	Overshoot (%)	Tempo de subida (s)	Tempo de acomodação (s)	Índice de erro IAE
$G_1(s)$	AG	2,52	10,44	19,27	24,10
	ACO	2,52	12,92	23,15	26,98
	ZN	60,21	4,08	44,98	39,59
$G_2(s)$	AG	3,80	12,02	41,30	31,38
	ACO	1,61	15,91	29,83	34,33
	ZN	57,86	8,04	104,09	60,06
$G_3(s)$	AG	14,0	5,79	37,04	31,66
	ACO	0,52	7,74	34,58	30,89
	ZN	83,15	5,62	108,14	70,57

Fonte: Elaborado pelo autor.

Pode-se observar que as sintonias com metaheurísticas mantiveram praticamente o tempo de subida dentro do esperado e tempo de acomodação e índice de desempenho melhores que o método de Ziegler-Nichols. Quanto ao critério *overshoot*, em todas as situações, o algoritmo ACO apresentou os menores valores e a metaheurística empregada valores abaixo de 20%.

6.1.5 Conclusões da sintonia dos controladores PID monovariáveis

Este estudo de caso utilizou as metaheurísticas AG e SA na sintonia de controladores PID aplicados em sistemas monovariáveis. Observou-se nas simulações apresentadas que a escolha dos parâmetros intrínsecos dos algoritmos influencia diretamente no desempenho dos mesmos.

Nas comparações realizadas entre as metaheurísticas, o AG apresentou resultados satisfatórios em relação a todos os critérios de desempenho. As respostas apresentadas foram comparadas às obtidas pelos métodos de Ziegler e Nichols e Varol e Bingul e demonstraram melhora em relação a algumas características como diminuição do tempo de acomodação e índice de desempenho IAE.

6.2 Estrutura em cascata

6.2.1 Considerações iniciais para a estrutura em cascata

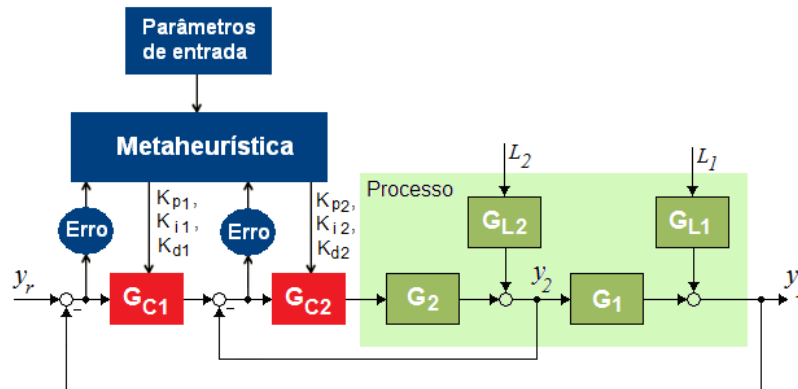
O segundo estudo de caso das metaheurísticas utilizou o controle de sistemas em cascata, extensivamente utilizado em indústrias de processos químicos. Com relação à sintonia de seus controladores PID, no artigo de Sadasivarao e Chidambaram (2006), os autores utilizaram como metodologia a determinação de forma simultânea dos parâmetros dos controladores das malhas interna e externa e a aplicação de degraus unitários nas malhas de carga interna e externa.

Neste caso foi utilizada uma das plantas estudadas por Sadasivarao e Chidambaram (2006) e, além da metodologia anterior, é proposta a aplicação de uma estratégia de otimização sequencial (começando do laço de controle mais interno até a malha externa) para determinar os parâmetros de controle de cada malha, esperando com isso menores valores de *overshoot* em cada malha. Outra diferença é com relação à aplicação de degraus unitários; no trabalho de Sadasivarao e Chidambaram (2006), os degraus eram aplicados em cada malha de carga do sistema. Na metodologia sequencial será aplicado o degrau unitário em cada referência das malhas interna e externa e na metodologia simultânea apenas na referência da malha externa, com o objetivo de obter menores valores de *overshoot* e manter a capacidade de compensar os distúrbios aplicados. Como é levado em consideração apenas o *overshoot* em cada metodologia de sintonia, não foram utilizadas frentes de Pareto na otimização dos parâmetros PID.

A avaliação do emprego das metaheurísticas AG e SA, nas duas metodologias de sintonia para a determinação dos parâmetros PID do controle de sistema em cascata, é efetuada pela comparação dos resultados advindos de cada uma delas e, posteriormente, pela comparação destes com os resultados obtidos pelo método convencional de Lee et al. (1998) e pelo trabalho proposto por Sadasivarao e Chidambaram (2006).

A figura 6.3 apresenta o diagrama em blocos de um sistema em cascata formado por duas malhas, onde L_1 e L_2 são as cargas, G_1 e G_2 são as funções de transferência do processo, G_{L1} e G_{L2} são as funções de transferência das cargas e G_{C1} e G_{C2} são as funções de transferência dos controladores e método de ajuste dos parâmetros de controle através de metaheurística.

Figura 6.3: Diagrama em blocos do controle de sistema em cascata.



Fonte: Elaborado pelo autor.

Para análise dos métodos, foram utilizados como funções de transferência do processo, os modelos apresentados em Krishnaswamy (1990) apud Lee et al. (1998), dados por:

$$G_1(s) = \frac{2e^{-2s}}{(20s + 1)}, \quad (6.7)$$

$$G_2(s) = \frac{e^{-10s}}{(100s + 1)}, \quad (6.8)$$

$$G_{L1}(s) = \frac{e^{-10s}}{(10s + 1)}, \quad (6.9)$$

$$G_{L2}(s) = 1. \quad (6.10)$$

Como comentado anteriormente, foram empregadas duas metodologias de sintonia dos controladores com metaheurísticas. A primeira metodologia utiliza uma função de agregação dos erros das duas malhas (interna e externa) para otimizar os parâmetros dos controladores de forma simultânea. A segunda metodologia realiza uma sintonia sequencial dos controladores, ou seja, primeiro sintoniza o controlador da malha interna, utilizando o próprio erro da malha como função custo, e após sintoniza o controlador da outra malha, utilizando também o erro da malha externa como função objetivo.

As análises comparativas levaram em consideração, durante a validação, apenas o índice de desempenho e nas comparações entre as metodologias de sintonia (sequencial e simultânea) e os métodos convencional e inteligente, os resultados das respostas do processo através de desempenhos no domínio do tempo (*overshoot*, tempo de subida e tempo de acomodação), índice de desempenho e teste de rejeição de distúrbios.

Os parâmetros de simulação utilizados para o Algoritmo Genético e *Simulated Annealing* estão definidos abaixo:

- Tempo de execução do processo para simulação: 250 s;
- Período de amostragem para determinação do índice de desempenho: 0,5 s;
- *Setpoint*: degrau unitário na referência a partir do tempo igual a zero;
- Faixa dos ganhos dos controladores:
 - Controlador PID interno: $0 \leq K_{p1} \leq 10,0$; $0 \leq K_{i1} \leq 1,0$; $0 \leq K_{d1} \leq 10,0$;
 - Controlador PID externo: $0 \leq K_{p2} \leq 10,0$; $0 \leq K_{i2} \leq 0,1$; $0 \leq K_{d2} \leq 40,0$.
- Índices de desempenho utilizados nas funções objetivo: ITAE.

6.2.2 Experimentos dos algoritmos na estrutura em cascata

Primeiramente serão testadas as diferentes configurações de parâmetros de trabalho para o Algoritmo Genético (variações do tamanho da população e dos tipos e taxas de cruzamento e mutação) e o *Simulated Annealing* (alterações da temperatura inicial e das funções de resfriamento e geração de novas soluções).

O primeiro experimento determinou o tamanho da população do Algoritmo Genético, tipos de cruzamento e mutação e seus respectivos parâmetros. Foram avaliados dois tipos de operadores de cruzamento: Heurístico e Aritmético e dois operadores de mutação: Uniforme e *Adaptive Feasible* e os seguintes valores para as taxas de cruzamento e mutação e tamanho da população conforme tabela 6.22.

Tabela 6.22: Relação de valores para as taxas de cruzamento e mutação e tamanho da população.

Taxa de cruzamento (%)	Taxa de mutação (%)	Tamanho da população
20, 40, 60, 80 e 90	1, 2, 5, 10 e 15	30, 50 e 100

Fonte: Elaborado pelo autor.

Foram gerados 180 conjuntos de parâmetros para o estudo de caso. Cada algoritmo AG foi executado trinta vezes utilizando a abordagem agregação de funções.

Ao término das trinta execuções foram calculados e exibidos o valor médio da função custo e o seu desvio padrão para cada conjunto de parâmetros. As tabelas 6.23 e 6.24 apresentam respectivamente os resultados do AG com *crossover* aritmético e heurístico. Os valores destacados mostram as melhores médias para cada tipo de cruzamento.

Tabela 6.23: Valores de custo e desvio-padrão dos AGs com *crossover* aritmético e mutações uniforme e adaptativa - estrutura em cascata.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
30	20	379,63	282,50	232,55	214,59	212,09	206,16	Média
		239,13	113,98	60,44	44,67	29,96	21,29	Desvio
	40	324,64	275,05	267,97	238,68	229,37	235,25	Média
		90,57	87,44	82,99	80,38	55,02	67,95	Desvio
	60	347,55	385,96	263,16	255,12	225,28	249,40	Média
		137,60	224,93	74,60	81,73	32,88	75,79	Desvio
	80	644,22	483,07	309,85	273,06	261,58	256,99	Média
		894,21	325,57	214,07	80,04	116,11	81,42	Desvio
	90	672,60	460,43	415,00	371,80	290,34	254,42	Média
		766,31	294,41	259,95	287,72	94,39	52,75	Desvio
50	20	300,91	250,87	208,08	206,32	195,70	205,11	Média
		199,24	87,12	25,36	31,17	14,88	28,60	Desvio
	40	310,57	248,77	214,97	205,41	207,68	214,81	Média
		206,51	58,48	24,84	19,12	22,93	39,84	Desvio
	60	298,30	286,57	228,73	218,80	211,19	235,64	Média
		64,31	169,33	32,22	34,33	33,48	46,64	Desvio
	80	373,20	305,43	260,68	262,71	213,84	229,50	Média
		165,03	108,90	58,37	97,81	22,30	42,48	Desvio
	90	390,71	401,61	311,92	246,72	246,06	235,07	Média
		193,67	284,77	158,02	81,96	42,59	43,78	Desvio
100	20	226,60	224,62	198,29	194,94	191,46	190,99	Média
		38,27	39,48	15,14	14,58	7,59	19,29	Desvio
	40	225,88	216,79	201,95	195,33	194,37	195,06	Média
		32,67	25,19	21,28	10,33	12,21	16,03	Desvio
	60	241,59	221,05	207,82	202,38	196,77	205,31	Média
		32,52	22,97	18,53	20,35	13,51	38,31	Desvio
	80	256,61	236,30	213,58	211,82	205,81	212,01	Média
		52,14	44,54	22,10	19,47	19,51	38,14	Desvio
	90	283,37	266,58	222,19	215,75	209,51	218,52	Média
		76,29	50,91	26,44	20,78	20,57	34,59	Desvio

Fonte: Elaborado pelo autor.

Tabela 6.24: Valores de custo e desvio-padrão dos AGs com *crossover* heurístico e mutações uniforme e adaptativa - estrutura em cascata.

N	Taxa de cruzamento (%)	Taxa de mutação (%)						Parâmetros de avaliação
		1	2	5	10	15	<i>Adaptive Feasible</i>	
30	20	505,66	240,37	250,23	229,12	197,73	218,36	Média
		353,13	51,97	77,57	43,52	21,57	44,59	Desvio
	40	405,87	316,25	235,10	222,04	207,14	262,98	Média
		278,94	144,79	59,08	45,40	33,84	59,06	Desvio
	60	415,47	301,34	249,40	224,62	205,18	265,44	Média
218,40		135,24	59,86	27,58	22,70	61,36	Desvio	
80	539,26	350,88	289,29	268,58	257,77	315,97	Média	
	480,23	145,25	98,33	81,27	68,00	143,75	Desvio	
90	458,37	413,	334,53	303,03	279,35	311,97	Média	
	234,94	156,24	117,63	87,24	74,33	98,60	Desvio	
50	20	283,53	253,38	212,79	208,68	197,25	189,86	Média
		86,87	88,93	32,49	27,32	19,65	16,66	Desvio
	40	271,20	239,18	228,07	209,51	202,83	252,01	Média
		102,55	72,02	49,11	28,86	22,19	80,71	Desvio
	60	327,24	245,27	245,27	227,06	206,85	238,42	Média
189,71		42,28	42,28	124,17	20,67	55,41	Desvio	
80	336,67	278,91	232,40	230,72	221,54	249,67	Média	
	240,04	77,90	57,81	47,29	34,30	54,51	Desvio	
90	326,56	354,79	286,82	263,24	245,66	289,02	Média	
	108,01	206,25	109,93	68,61	90,96	177,77	Desvio	
100	20	226,41	211,92	201,89	187,42	182,69	187,57	Média
		30,95	28,24	28,47	13,62	11,06	15,92	Desvio
	40	220,97	214,72	203,35	197,55	187,95	208,18	Média
		38,36	37,66	20,34	19,59	12,55	54,92	Desvio
	60	231,28	229,85	214,94	200,66	192,14	207,61	Média
35,28		42,18	39,63	15,70	15,62	37,08	Desvio	
80	236,63	230,62	213,77	207,90	204,40	206,79	Média	
	29,55	46,11	35,61	35,24	21,73	33,97	Desvio	
90	255,04	241,19	237,63	205,86	221,78	225,41	Média	
	59,97	47,28	46,53	23,96	39,09	53,94	Desvio	

Fonte: Elaborado pelo autor.

Nesta análise, para fins de comparação, a melhor sintonia será aquela que apresentar o menor índice de erro. Verifica-se novamente que os melhores resultados são obtidos, na maior parte, com altas taxas de mutação e baixas taxas de cruzamento. Outra constatação é novamente o aumento do número de indivíduos da população que se refletiu nos melhores resultados quanto à função objetivo. Observa-se ainda nas tabelas anteriores que o melhor resultado e melhor dos “piores” resultados quanto à função custo são obtidos com o cruzamento heurístico. Dentre as avaliações realizadas, o cruzamento heurístico com taxa de 0,2 em combinação com a mutação uniforme com taxa de 0,15 e população igual a 100 apresentou os melhores resultados quanto à função custo e desta forma é a configuração escolhida para a metaheurística AG na determinação dos parâmetros PID.

O segundo experimento determinou a temperatura inicial e as funções de resfriamento e geração de novas soluções do *Simulated Annealing*. Cada algoritmo foi executado trinta vezes e ao término das execuções foram calculados e exibidos o valor médio da função custo e o desvio padrão. Os resultados obtidos são apresentados na tabela 6.25 e o valor destacado mostra a melhor média.

Tabela 6.25: Valores de custo dos SAs - estrutura em cascata

Temp inicial	Função <i>Annealing</i>	Função de resfriamento			Parâmetros de avaliação
		Exponencial	Logarítmica	Linear	
50	<i>Fast Annealing</i>	246,08	216,79	240,15	Média
		112,63	29,18	72,62	Desvio
	<i>Boltzman Annealing</i>	305,94	224,42	629,76	Média
		91,99	33,67	805,95	Desvio
100	<i>Fast Annealing</i>	234,62	227,11	219,46	Média
		76,76	31,42	35,28	Desvio
	<i>Boltzman Annealing</i>	305,12	222,27	461,29	Média
		160,94	30,93	652,00	Desvio
200	<i>Fast Annealing</i>	237,07	215,77	218,13	Média
		64,40	16,02	36,48	Desvio
	<i>Boltzman Annealing</i>	273,01	222,28	230,51	Média
		88,20	23,68	55,22	Desvio

Fonte: Elaborado pelo autor.

Dentre as avaliações realizadas, a função de resfriamento logarítmica com temperatura inicial 200 em combinação com a função de geração de novas soluções *Fast Annealing* apresentou os melhores resultados quanto à função custo e desta forma é a configuração escolhida para a metaheurística SA na determinação dos parâmetros PID.

6.2.3 Desempenho das metaheurísticas no sistema em cascata

Para este estudo, os objetivos a serem atingidos pelo sistema de controle com estrutura em cascata sintonizado por AG e SA foram:

- Erro nulo de regime permanente;
- Rejeitar distúrbios aplicados nas malhas de carga interna e externa do sistema;
- Desempenhos no domínio do tempo superiores aos métodos propostos por Lee et al. (1998) e Sadasivarao e Chidambaram (2006).

A primeira comparação realizada foi entre duas metodologias abordadas para as metaheurísticas do trabalho: a sintonia simultânea e a sintonia sequencial. Cada algoritmo foi executado trinta vezes e escolhidos os conjuntos de parâmetros do controlador PID que apresentaram valores próximos à média da função custo. As tabelas 6.26 e 6.27 apresentam respectivamente os valores custo (média, desvio padrão juntamente com o melhor e o pior valor absoluto) e os resultados dos parâmetros determinados mais próximos da média da FO. As duas metodologias são apresentadas respectivamente como Sim e Seq.

Tabela 6.26: Valores de custo das metaheurísticas.

Método de sintonia	Função custo (média)	Desvio padrão	Função custo (melhor)	Função custo (pior)
AG Sim	141,10	0,83	139,39	144,22
SA Sim	215,77	16,02	166,92	524,32
AG Seq	330,45	181,59	122,97	690,67
SA Seq	369,85	233,46	118,65	821,61

Fonte: Elaborado pelo autor.

Tabela 6.27: Parâmetros de sintonia do sistema em cascata para diferentes metodologias.

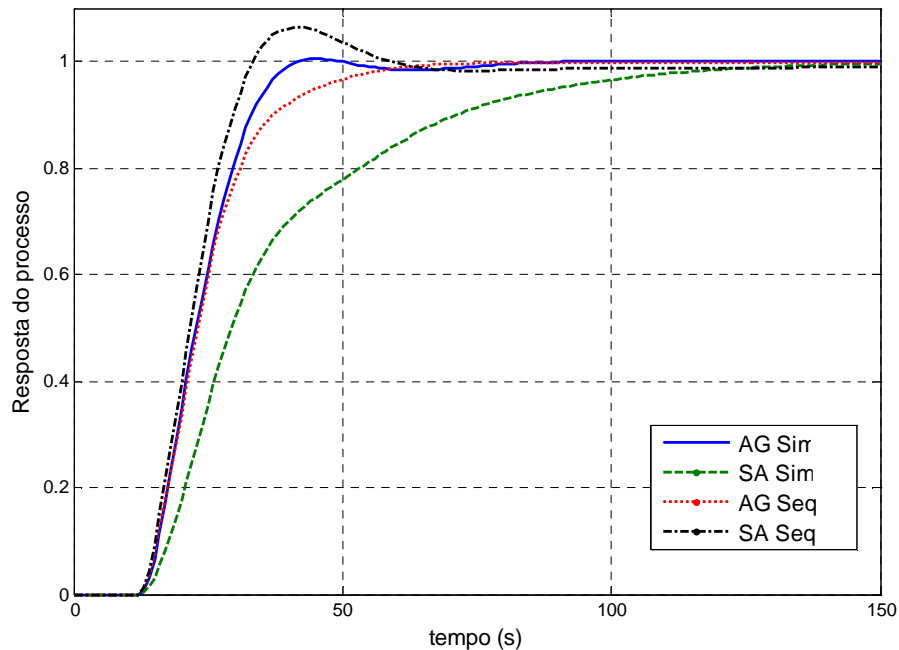
Método de sintonia	Parâmetros PID					
	K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
AG Sim	3,0596	0,4088	0,6292	4,5189	0,0417	7,7058
SA Sim	1,8854	0,1141	0,7775	3,7159	0,0265	39,39
AG Seq	3,3393	0,1359	2,6004	5,4114	0,0389	32,5662
SA Seq	3,5489	0,1420	2,7913	6,2658	0,0421	24,6593

Fonte: Elaborado pelo autor.

A primeira constatação dos experimentos foi com relação ao tempo de simulação. A metodologia simultânea apresentou valor três vezes menor com relação à metodologia sequencial.

O gráfico 6.18 mostra a simulação da resposta do processo em malha fechada para cada metodologia utilizada.

Gráfico 6.18: Resposta do processo em malha fechada com controladores PID sintonizados pelas metaheurísticas com diferentes metodologias.



Fonte: Elaborado pelo autor.

Observou-se uma piora nos resultados da média da função objetivo da metodologia sequencial. Isto foi devido à sintonia dos parâmetros da malha interna, pois se constatou que quando os valores da função objetivo da malha interna estão muito próximos do valor ideal (menor valor possível) tornando as respostas da saída mais rápidas, não há garantia de obtenção dos menores valores de *overshoot* comprometendo a sintonia da malha externa.

Cada solução encontrada pelas metaheurísticas foi avaliada pelos critérios de *overshoot*, tempo de acomodação e índice de desempenho IAE, conforme tabela 6.28.

Tabela 6.28: Desempenho dos sistemas em cascata com diferentes metodologias.

Método de sintonia	<i>Overshoot</i> (%)	Tempo de subida (s)	Tempo de acomodação (s)	Índice de erro IAE
AG Sim	0,55	17,34	38,33	24,55
SA Sim	0	53,41	113,25	38,52
AG Seq	0	21,59	55,35	25,89
SA Seq	6,54	14,51	53,54	24,62

Fonte: Elaborado pelo autor.

Pode-se observar pela superposição das curvas dos gráficos anteriores e pela tabela 6.28 que as metaheurísticas com diferentes metodologias apresentaram desempenhos semelhantes quanto ao *overshoot*, onde o método simultâneo com AG apresentou os menores valores de tempo de acomodação e índice de erro e o método sequencial com SA o menor tempo de subida e bons valores quanto ao tempo de acomodação e índice de erro IAE. Por isso, estas abordagens foram as escolhidas para fins de comparação com os valores do método BLT de Lee et al. (1998) e com os valores do Algoritmo Genético (AG SC), proposto por Sadasivarao e Chidambaram (2006). Os parâmetros são mostrados na tabela 6.29.

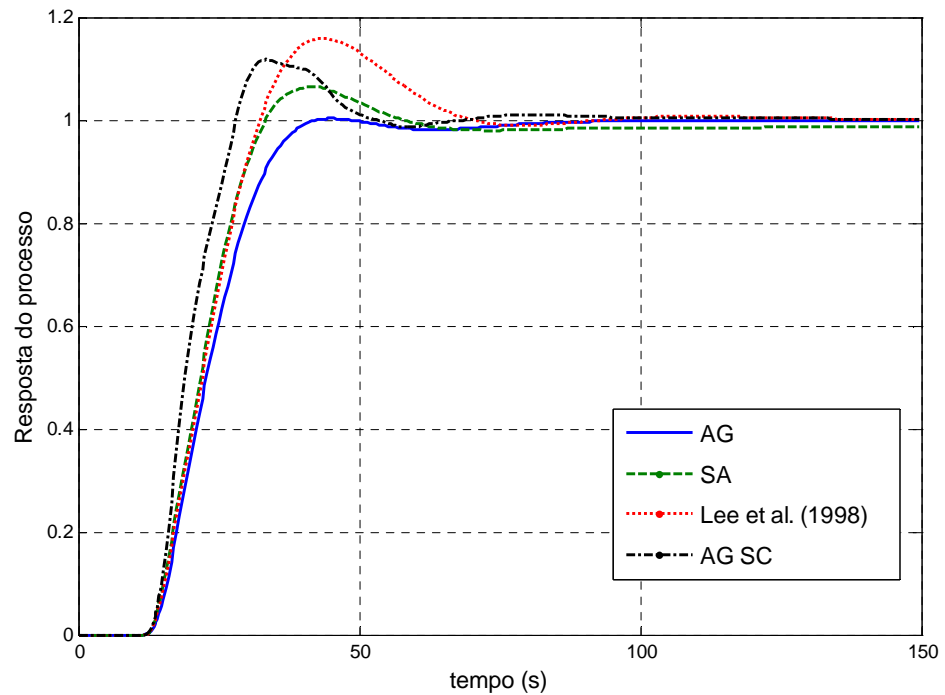
Tabela 6.29: Parâmetros de sintonia pelos métodos Lee et al. (1998) e metaheurísticas.

Método de sintonia	Parâmetros PID					
	K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
AG	3,0596	0,4088	0,6292	4,5189	0,0417	7,7058
SA	3,5489	0,1420	2,7913	6,2658	0,0421	24,6593
Lee et al. (1998)	3,4440	0,1667	2,2217	5,8300	0,0555	27,984
AG SC	3,9089	0,7850	0,1406	5,9728	0,0590	9,2746

Fonte: Elaborado pelo autor.

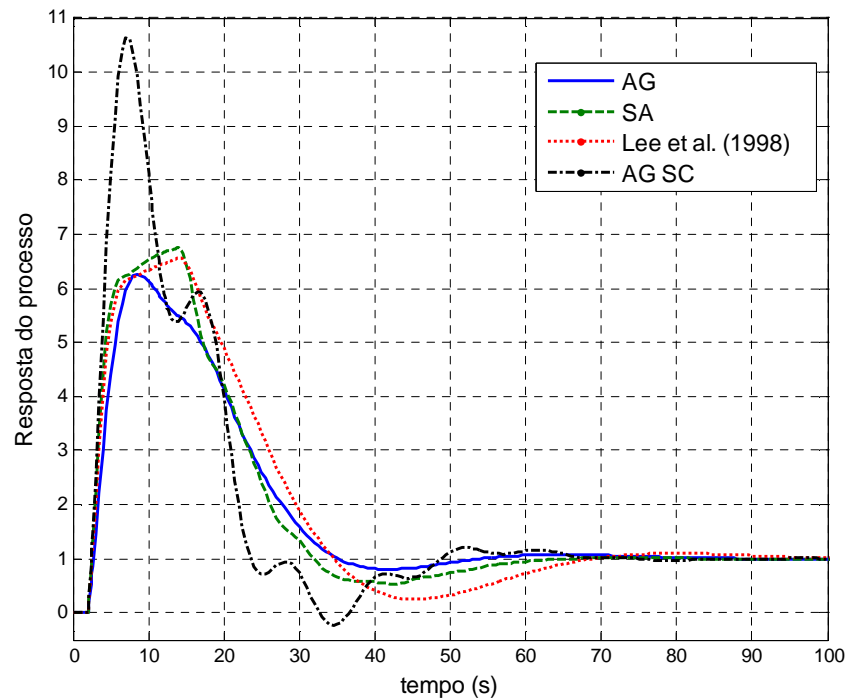
As respostas das malhas externa e interna a um degrau unitário aplicado na referência dos controladores PID podem ser observadas nos gráficos 6.19 e 6.20 respectivamente.

Gráfico 6.19: Resposta do processo em cascata com controladores PID sintonizados pelos métodos Lee et al. (1998) e metaheurísticas.



Fonte: Elaborado pelo autor.

Gráfico 6.20: Resposta da malha interna com controladores PID sintonizados pelos métodos Lee et al. (1998) e metaheurísticas.



Fonte: Elaborado pelo autor.

A tabela 6.30 mostra os índices de desempenho produzidos pelos controladores PID sintonizados pelos diferentes métodos.

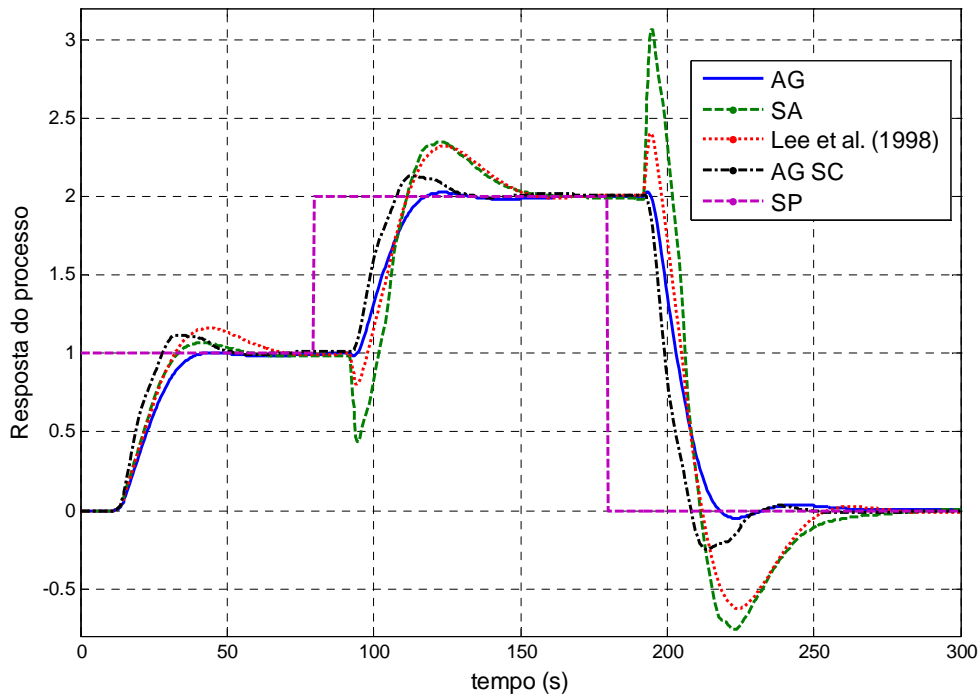
Tabela 6.30: Desempenho dos sistemas em cascata sintonizados pelos métodos Lee et al. (1998) e metaheurísticas.

Método de sintonia	<i>Overshoot</i> (%)	Tempo de subida (s)	Tempo de acomodação (s)	Índice de erro IAE
AG	0,55	17,34	38,33	24,55
SA	6,54	14,51	53,54	24,62
Lee et al. (1998)	16,09	14,50	65,50	26,52
AG SC	11,96	11,50	49,00	22,56

Fonte: Elaborado pelo autor.

O gráfico 6.21 apresenta as respostas da variável controlada para diferentes valores de referência.

Gráfico 6.21: Resposta do processo em cascata para alterações na referência.

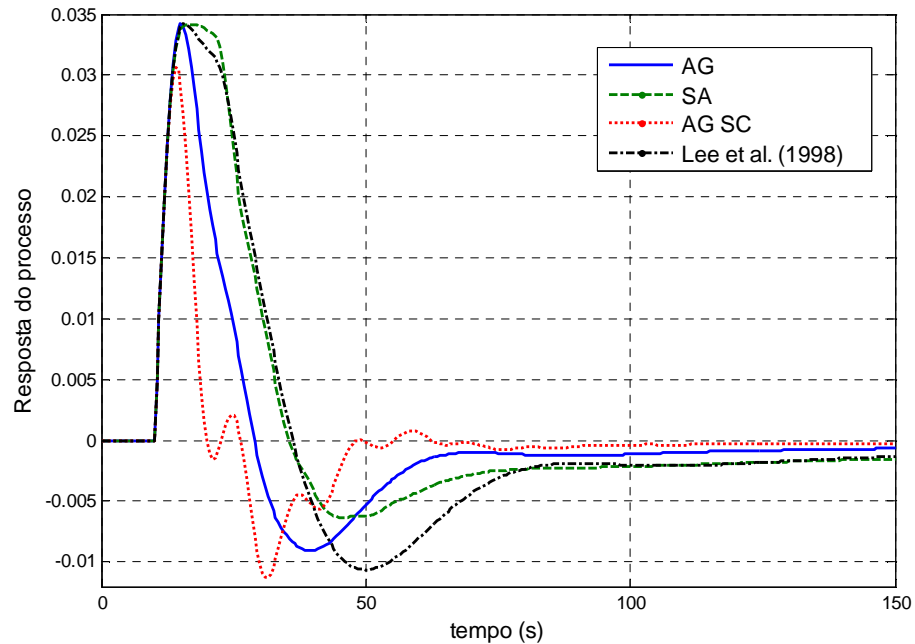


Fonte: Elaborado pelo autor.

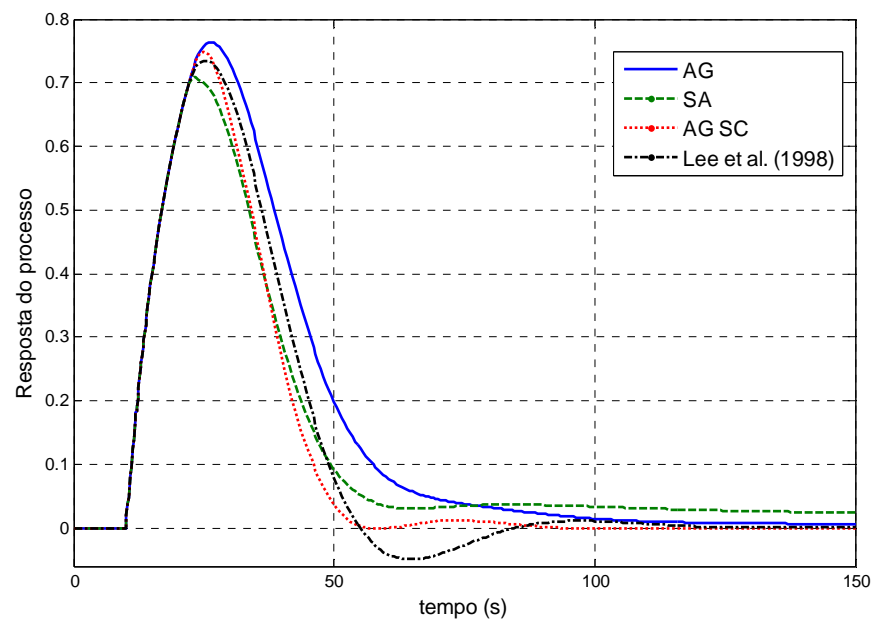
É possível perceber nos gráficos 6.19 e 6.20 que as metaheurísticas empregadas apresentam erro de estado estacionário nulo nas saídas das malhas externa e interna. Para a malha externa, a metaheurística AG empregada apresentou os menores valores de *overshoot* e tempo de acomodação comparado aos demais métodos. Na malha interna, observa-se ainda que as duas metaheurísticas também apresentaram os menores valores de *overshoot*, onde o SA apresentou *overshoot* de 6,7 vezes o valor de referência, o AG empregado em torno de 6,2 vezes enquanto que o método convencional e o AG de Sadasivarao e Chidambaram (2006) apresentaram respectivamente valores entre 6,5 e 10 vezes a referência.

6.2.4 Distúrbios aplicados ao sistema em cascata

Nos testes de rejeição de distúrbios realizados com os controladores, o sinal de referência foi mantido em zero e os distúrbios foram gerados na forma de degraus unitários com valores iguais a 1,0 e adicionados individualmente nas malhas de carga (interna e externa) do processo. Os resultados das respostas nas malhas são apresentados nos gráficos 6.22 e 6.23.

Gráfico 6.22: Resposta do processo para um degrau na carga L_2 .

Fonte: Elaborado pelo autor.

Gráfico 6.23: Resposta do processo para um degrau na carga L_1 .

Fonte: Elaborado pelo autor.

Para o distúrbio aplicado em L_2 , as metaheurísticas AG e SA apresentaram *overshoot* respectivamente de 11,73% e 11,40% acima do menor valor (AG SC). Para o degrau em L_1 , o AG apresentou *overshoot* de 7,82% em relação ao valor de SA, que foi o menor valor. Contudo, percebe-se pelos gráficos anteriores que todos controladores possuem a capacidade de compensar os distúrbios aplicados, demonstrando que os métodos são robustos a este tipo de perturbação. Aliado aos resultados de desempenho obtidos anteriormente, pode-se afirmar que as metaheurísticas empregadas apresentaram um compromisso entre o desempenho e a eliminação de distúrbios no sistema.

6.2.5 Conclusões da sintonia para estruturas em cascata

Simulações preliminares no controle de sistemas dinâmicos com estrutura em cascata foram realizadas para a determinação dos melhores parâmetros das metaheurísticas empregadas. As comparações entre as duas metodologias abordadas evidenciaram que a forma simultânea (que consiste na otimização simultânea de todos os parâmetros dos controladores PID) com AG obteve de forma geral melhores respostas do processo em relação à abordagem sequencial. Outro ponto é o tempo de simulação, onde a metodologia simultânea apresenta os menores valores (cerca de três vezes menor).

Ainda de acordo com o estudo realizado, em comparação aos trabalhos de Lee et al. (1998) e Sadasivarao e Chidambaram (2006), esta abordagem pode ser considerada a mais eficiente em termos de critérios de desempenho, sobretudo no tempo de acomodação e *overshoot* e mostrou bons resultados quanto à robustez na compensação de distúrbios.

6.3 Controle Multivariável

6.3.1 Considerações iniciais para o controle multivariável

Para o terceiro estudo de caso desta dissertação foi utilizada a mesma planta e testes de desempenho dos sistemas apresentados no trabalho de Iruthayarajan e Baskar (2009). Também foi utilizada a mesma estrutura descentralizada dos trabalhos de Swiech (2005), Chang (2007) e Iruthayarajan e Baskar (2009) e adicionada a arquitetura desacoplada para fins de comparação. Nesta arquitetura apresentada, além de otimizar os três parâmetros de cada controlador PID, também foram determinados os ganhos estáticos. Em se tratando da função objetivo, Chang (2007) e Iruthayarajan e Baskar (2009) utilizaram uma função de avaliação baseada diretamente no erro de cada malha (índice IAE) e Swiech (2005) uma função de agregação que levou em consideração o índice ITSE e a variância mínima aplicados ao sinal de controle e à saída. Para este trabalho foi utilizada frente de Pareto levando em consideração o índice ITSE e *overshoot*, das saídas y_1 e y_2 com o objetivo de obter menores valores de *overshoot* e tempo de acomodação.

Para avaliação da metaheurística aplicada no controle de processos MIMO foi utilizada a coluna de destilação Wood and Berry. É um modelo multivariável com duas entradas e duas saídas de difícil controle, pois possui significativo atraso de transporte e forte acoplamento entre as variáveis (CAMPESTRINI, 2006). As funções de transferência que descrevem o comportamento da coluna são dadas por:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12,8e^{-s}}{16,7s+1} & \frac{-18,9e^{-3s}}{21s+1} \\ \frac{6,6e^{-7s}}{10,9s+1} & \frac{-19,4e^{-3s}}{14,4s+1} \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix}, \quad (6.11)$$

onde $y_1(s)$ é a porcentagem mássica de metanol no topo, $y_2(s)$ é a porcentagem mássica de metanol no fundo, $x_1(s)$ é a vazão mássica de refluxo e $x_2(s)$ é a vazão mássica de vapor. Os pontos de operação $y_1(s)$ e $y_2(s)$ são respectivamente 96% e 0,5%.

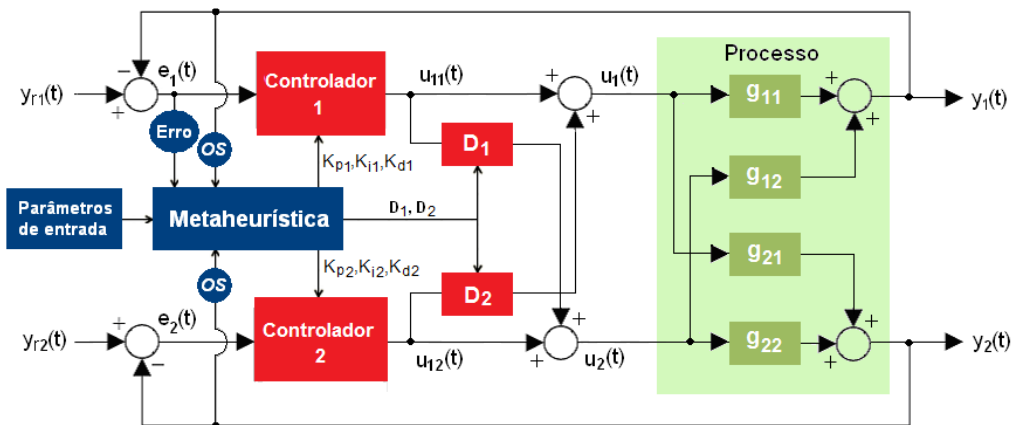
São utilizadas duas arquiteturas, a descentralizada e a estrutura com desacopladores. Os parâmetros de cada controlador foram determinados de forma simultânea através da metaheurística Algoritmo Genético com otimização multiobjetivo.

Os parâmetros de simulação utilizados para o AG estão definidos abaixo:

- Tempo de execução do processo para simulação: 300 s;
- Período de amostragem para determinação do índice de desempenho: 1 s;
- *Setpoint*: degrau de 0,96 na referência $y_1(s)$ a partir do tempo igual a zero e degrau de 0,05 na referência $y_2(s)$ a partir do tempo igual a 150 s;
- Faixa dos ganhos dos controladores PID descentralizados e com desacopladores: - 0,2 a 0,2 para todos os parâmetros.

A figura 6.4 apresenta o diagrama em blocos do sistema. Para a otimização foram utilizadas frentes de Pareto levando-se em consideração os *overshoots* (OS) das saídas y_1 (objetivo 1) e y_2 (objetivo 2) e o índice ITSE da saída y_1 (objetivo 3).

Figura 6.4: Diagrama em blocos do sistema de controle de sistema multivariável.



Fonte: Elaborado pelo autor.

6.3.2 Experimentos dos algoritmos no controle de sistemas MIMO

O primeiro experimento do controle de processos MIMO determinou o tamanho da população do Algoritmo Genético, tipos de cruzamento e mutação e seus respectivos parâmetros. Foram avaliados dois tipos de operadores de cruzamento: Heurístico e Aritmético e dois operadores de mutação: Uniforme e *Adaptive Feasible*. A tabela 6.31 apresenta os valores propostos para a taxa de cruzamento, taxa de mutação e tamanho da população.

Tabela 6.31: Relação de valores para a taxa de cruzamento - processo MIMO.

Taxa de cruzamento (%)	Taxa de mutação (%)	Tamanho da população
20, 40, 60 e 90	2, 5, 10 e 15	30, 50 e 100

Fonte: Elaborado pelo autor.

Estes valores serão combinados entre si juntamente com os tipos de operadores de cruzamento e mutação gerando 40 conjuntos de parâmetros para o estudo de caso.

Cada algoritmo AG foi executado trinta vezes utilizando frentes de Pareto. Como critério de tomada de decisão da solução foi utilizado o critério de menor OS para cada saída. Ao término das execuções foram calculados e exibidos o valor médio da função custo, o desvio padrão e o número de gerações de cada simulação.

As tabelas 6.32, 6.33 e 6.34 apresentam os resultados do AG multiobjetivo com *crossover* aritmético com populações iguais a 30, 50 e 100 respectivamente e as tabelas 6.35, 6.36 e 6.37 os resultados com *crossover* heurístico com população N igual a 30, 50 e 100 respectivamente. As colunas mostram o tamanho da população (primeira coluna), a variação da taxa de mutação (terceira coluna) e os parâmetros de avaliação (última coluna) onde $OS1$ é o *overshoot* da saída y_1 , $OS2$ o *overshoot* da saída y_2 e ITSE o índice de erro da saída y_1 . Na segunda coluna, as linhas apresentam a variação da taxa de cruzamento. Os resultados apresentados são o valor médio obtido para cada função objetivo e o desvio padrão em torno desta média, com base nos parâmetros mostrados. Nesta análise, para fins de comparação, a melhor sintonia será aquela que apresentar o menor índice de erro. Os valores destacados mostram as melhores médias para cada tipo de cruzamento.

Tabela 6.32: Média e desvio-padrão dos AGs com *crossover* aritmético e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 30$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)					Parâmetros de avaliação		
		2	5	10	15	<i>Adaptive Feasible</i>			
30	20	0	0	0	0	0	Média	$OS1$	
		0	0	0	0	0	Desvio		
		0	0	0	0	0	Média	$OS2$	
		0	0	0	0	0	Desvio		
		35,78	23,46	20,94	19,17	23,03	Média	ITSE	
		28,88	7,85	4,09	1,44	10,41	Desvio		
		40	0	0	0	0	0	Média	$OS1$
			0	0	0	0	0	Desvio	
			0	0	0	0	0	Média	$OS2$
			0	0	0	0	0	Desvio	
	28,42		26,04	20,38	21,14	20,65	Média	ITSE	
	12,76		8,81	3,05	7,55	3,69	Desvio		
	60	0	0	0	0	0	Média	$OS1$	
		0	0	0	0	0	Desvio		
		0	0	0	0	0	Média	$OS2$	
		0	0	0	0	0	Desvio		
		42,05	29,06	25,13	23,75	29,99	Média	ITSE	
		29,42	15,59	10,78	13,50	28,51	Desvio		
		90	0	0	0	0	0	Média	$OS1$
			0	0	0	0	0	Desvio	
0			0	0	0	0	Média	$OS2$	
0			0	0	0	0	Desvio		
61,19	37,77		42,40	36,89	48,51	Média	ITSE		
34,40	22,81		32,02	23,89	37,45	Desvio			

Fonte: Elaborado pelo autor.

Tabela 6.33: Média e desvio-padrão dos AGs com *crossover* aritmético e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 50$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)					Parâmetros de avaliação	
		2	5	10	15	<i>Adaptive Feasible</i>		
50	20	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		22,34	20,26	18,62	18,46	19,49	Média	ITSE
		3,95	2,67	1,37	0,69	4,85	Desvio	
	40	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		22,20	19,63	19,15	18,90	20,47	Média	ITSE
		6,62	1,89	2,50	1,03	4,21	Desvio	
	60	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		24,09	21,44	20,54	20,30	23,45	Média	ITSE
		5,08	4,73	4,01	4,33	5,66	Desvio	
	90	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		45,19	29,22	31,74	25,43	38,11	Média	ITSE
		31,65	13,72	23,44	10,03	51,31	Desvio	

Fonte: Elaborado pelo autor.

Tabela 6.34: Média e desvio-padrão dos AGs com *crossover* aritmético e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 100$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)					Parâmetros de avaliação	
		2	5	10	15	<i>Adaptive Feasible</i>		
100	20	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		21,84	18,65	18,20	18,02	17,61	Média	ITSE
		6,58	1,34	0,83	0,77	0,82	Desvio	
	40	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		22,51	18,97	18,44	18,11	18,44	Média	ITSE
		6,55	1,09	0,78	0,65	1,38	Desvio	
	60	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		22,82	21,61	18,58	18,62	18,72	Média	ITSE
		7,38	9,23	0,98	0,83	1,46	Desvio	
	90	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		40,62	26,88	31,16	20,99	23,87	Média	ITSE
		57,09	11,28	51,42	2,19	6,50	Desvio	

Fonte: Elaborado pelo autor.

Tabela 6.35: Média e desvio-padrão dos AGs com *crossover* heurístico e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 30$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)					Parâmetros de avaliação	
		2	5	10	15	<i>Adaptive Feasible</i>		
30	20	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		34,70	22,41	20,35	19,44	20,60	Média	ITSE
		24,92	4,67	2,23	1,37	4,26	Desvio	
	40	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		28,44	24,00	19,66	19,52	20,98	Média	ITSE
		12,79	14,02	1,84	1,63	4,99	Desvio	
	60	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		36,63	26,12	21,58	20,37	22,08	Média	ITSE
		22,59	10,74	3,61	1,77	4,08	Desvio	
	90	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		47,51	45,30	35,89	26,19	33,08	Média	ITSE
		38,45	45,71	25,43	6,67	16,62	Desvio	

Fonte: Elaborado pelo autor.

Tabela 6.36: Média e desvio-padrão dos AGs com *crossover* heurístico e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 50$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)					Parâmetros de avaliação	
		2	5	10	15	<i>Adaptive Feasible</i>		
50	20	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		25,83	20,51	18,57	18,81	19,49	Média	ITSE
		8,44	2,54	1,00	1,84	4,55	Desvio	
	40	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		25,19	19,95	18,77	19,04	20,97	Média	ITSE
		13,79	2,98	1,34	2,17	5,18	Desvio	
	60	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		25,06	22,86	19,60	19,79	20,28	Média	ITSE
		6,59	8,45	2,10	1,99	3,74	Desvio	
	90	0	0	0	0	0	Média	$OS1$
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	$OS2$
		0	0	0	0	0	Desvio	
		35,32	25,09	23,46	24,73	27,02	Média	ITSE
		15,79	5,34	6,79	7,39	10,28	Desvio	

Fonte: Elaborado pelo autor.

Tabela 6.37: Média e desvio-padrão dos AGs com *crossover* heurístico e mutações uniforme e adaptativa - controle de sistema MIMO com população $N = 100$.

N	Taxa de cruzamento (%)	Taxa de mutação (%)					Parâmetros de avaliação	
		2	5	10	15	<i>Adaptive Feasible</i>		
100	20	0	0	0	0	0	Média	OS1
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	OS2
		0	0	0	0	0	Desvio	
		19,89	18,84	18,38	18,17	17,87	Média	ITSE
		1,45	1,35	0,90	0,90	1,37	Desvio	
	40	0	0	0	0	0	Média	OS1
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	OS2
		0	0	0	0	0	Desvio	
		20,10	18,64	18,17	17,85	17,75	Média	ITSE
		2,72	1,12	0,77	0,43	1,09	Desvio	
	60	0	0	0	0	0	Média	OS1
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	OS2
		0	0	0	0	0	Desvio	
		22,28	19,55	18,71	18,48	18,94	Média	ITSE
		2,84	2,06	1,58	1,05	3,42	Desvio	
	90	0	0	0	0	0	Média	OS1
		0	0	0	0	0	Desvio	
		0	0	0	0	0	Média	OS2
		0	0	0	0	0	Desvio	
		26,94	23,85	21,93	20,97	21,28	Média	ITSE
		8,88	5,99	4,40	6,53	5,39	Desvio	

Fonte: Elaborado pelo autor.

Pode-se verificar nas tabelas anteriores que o aumento da probabilidade de cruzamento não promove melhorias na função custo. Também se observa que o aumento da probabilidade de mutação provoca melhorias nos resultados. Estes comportamentos se devem, novamente, à complexidade do problema, fazendo com que o aumento da taxa de cruzamento se reflita no aumento da possibilidade da quebra de boas soluções e o aumento da taxa de mutação na diversificação da população pela busca de melhores soluções.

Dentre as avaliações realizadas, o cruzamento aritmético com taxa de 0,2 em combinação com a mutação *Adaptive Feasible* apresentou os melhores resultados quanto à função custo e desta forma é a configuração escolhida para a metaheurística AG na determinação dos parâmetros PID.

6.3.3 Desempenho das metaheurísticas no sistema MIMO 2 x 2

A primeira comparação avaliou os critérios *overshoot* (OS), tempo de subida (t_R), tempo de acomodação (t_S) e índice de desempenho IAE das duas arquiteturas empregadas para as metaheurísticas do trabalho, do método BLT (*Biggest Log-Modulus*), proposto por Luyben (1986) e do Algoritmo Genético proposto por Swiech (2005). As metaheurísticas e arquiteturas utilizadas são apresentadas como:

- **AG Descen:** Controladores PID descentralizados sintonizados por Algoritmos Genéticos;
- **AG Desaco:** Controladores PID com desacopladores sintonizados por Algoritmos Genéticos;
- **BLT:** Controladores PI descentralizados sintonizados pelo método *Biggest Log-Modulus Tuning* proposto por Luyben (1986);
- **AG SW:** Controladores PID descentralizados sintonizados por Algoritmos Genéticos propostos por Swiech (2005).

A tabela 6.38 apresenta os resultados dos parâmetros determinados.

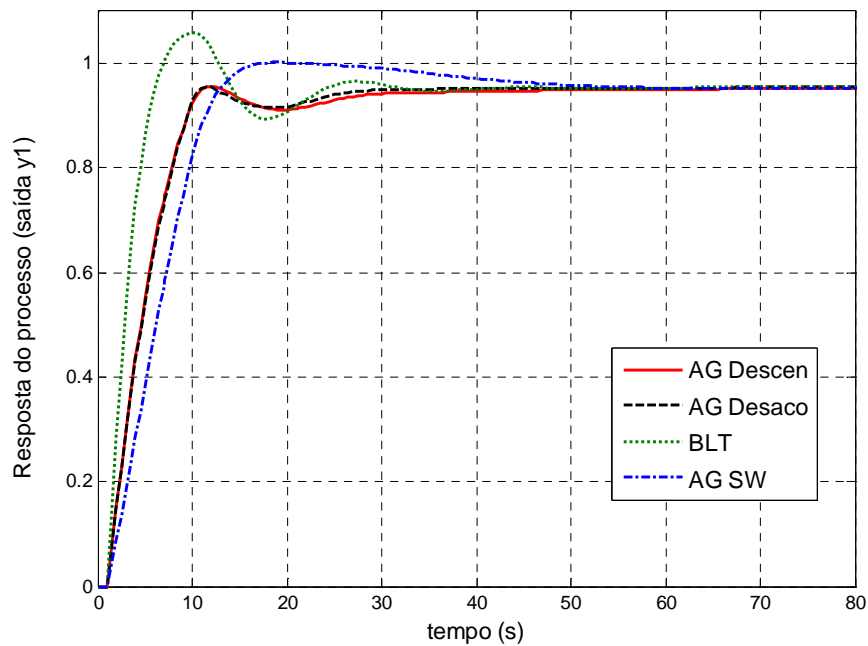
Tabela 6.38: Parâmetros de sintonia pelos métodos BLT e metaheurísticas.

Método de sintonia	Parâmetros PID							
	K_{p1}	K_{i1}	K_{d1}	D_1	K_{p2}	K_{i2}	K_{d2}	D_2
AG Descen	0,2000	0,0251	- 0,0063	0	- 0,0972	- 0,0047	- 0,1165	0
AG Desaco	0,1964	0,0261	- 0,0396	0,0442	-0,1140	-0,0053	-0,1780	0,0477
BLT	0,3750	0,0452	0	0	-0,075	-0,0032	0	0
AG SW	0,11238	0,02431	0,00639	0	-0,12129	-0,00158	-0,13549	0

Fonte: Elaborado pelo autor.

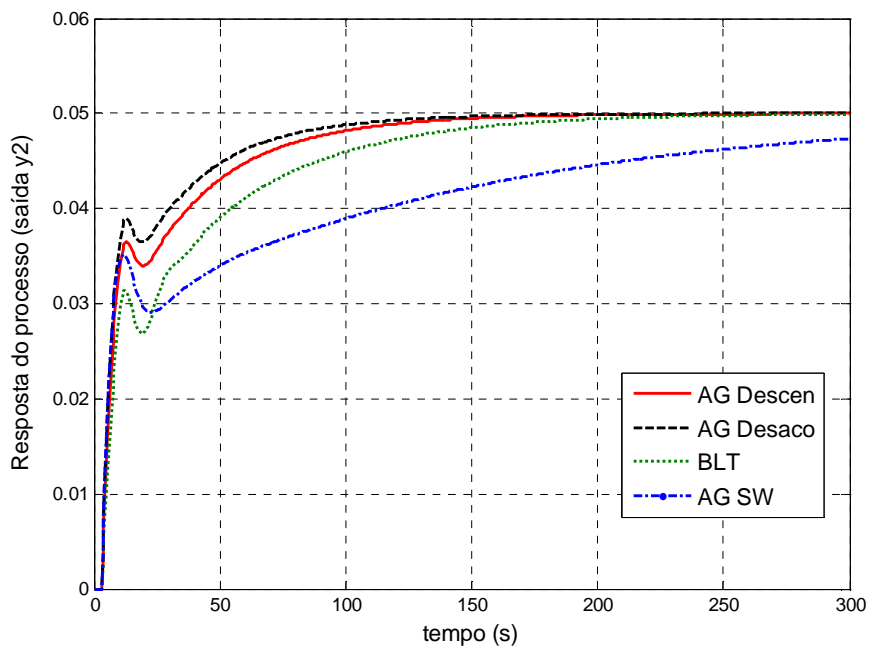
Os gráficos 6.24 e 6.25 mostram a simulação das respostas das saídas y_1 e y_2 em malha fechada para cada arquitetura empregada. Para as saídas y_1 e y_2 de graus de 0,96 e 0,05 no tempo igual a zero são aplicados respectivamente nas referências x_1 e x_2 dos controladores PID.

Gráfico 6.24: Resposta do processo (saída y_1) em malha fechada com controladores PID sintonizados pelos métodos BLT e metaheurísticas.



Fonte: Elaborado pelo autor.

Gráfico 6.25: Resposta do processo (saída y_2) com controladores PID sintonizados pelos métodos BLT e metaheurísticas.



Fonte: Elaborado pelo autor.

Nos dois gráficos anteriores, observa-se que o sistema apresentou nas duas arquiteturas sintonizadas por AG Multiobjetivo, valores de erro de estado estacionário nulo para as saídas y_1 e y_2 .

E na tabela 6.39 é apresentado o desempenho de cada método de sintonia. Observa-se na tabela que para a saída y_1 , as sintonias efetuadas pelas metaheurísticas empregadas apresentaram *overshoot* igual a zero e os menores tempos de acomodação e subida se comparados ao método AG de Swiech enquanto que o método BLT apresentou um valor de *overshoot* considerável. Verifica-se ainda que para y_2 as estruturas com o AG empregado apresentaram os menores valores em todos os critérios de desempenho.

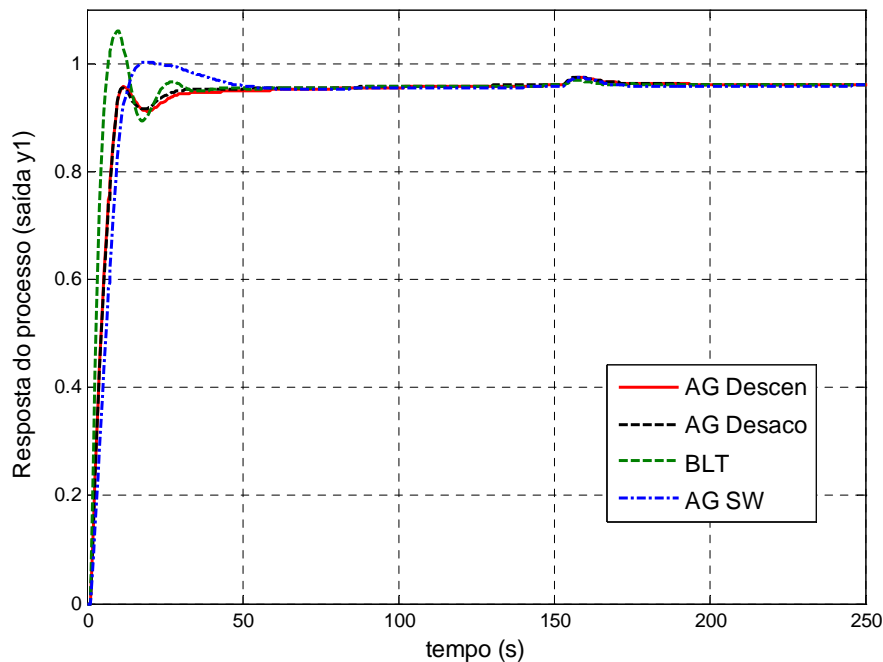
Tabela 6.39: Desempenho dos sistemas MIMO 2 x 2 com controladores PID sintonizados pelos métodos BLT e metaheurísticas.

Método de sintonia	x_1y_1				x_2y_2			
	OS (%)	t_R (s)	t_S (s)	Índice IAE	OS (%)	t_R (s)	t_S (s)	Índice IAE
AG Descen	0	7,17	28,69	6,49	0	57,69	123,52	1,12
AG Desaco	0	7,23	25,19	6,19	0	47,50	106,17	0,93
BLT	10,4	3,68	22,90	4,815	0	85,05	171,41	1,64
AG SW	4,38	8,60	36,03	7,847	0	206,82	433,13	2,94

Fonte: Elaborado pelo autor.

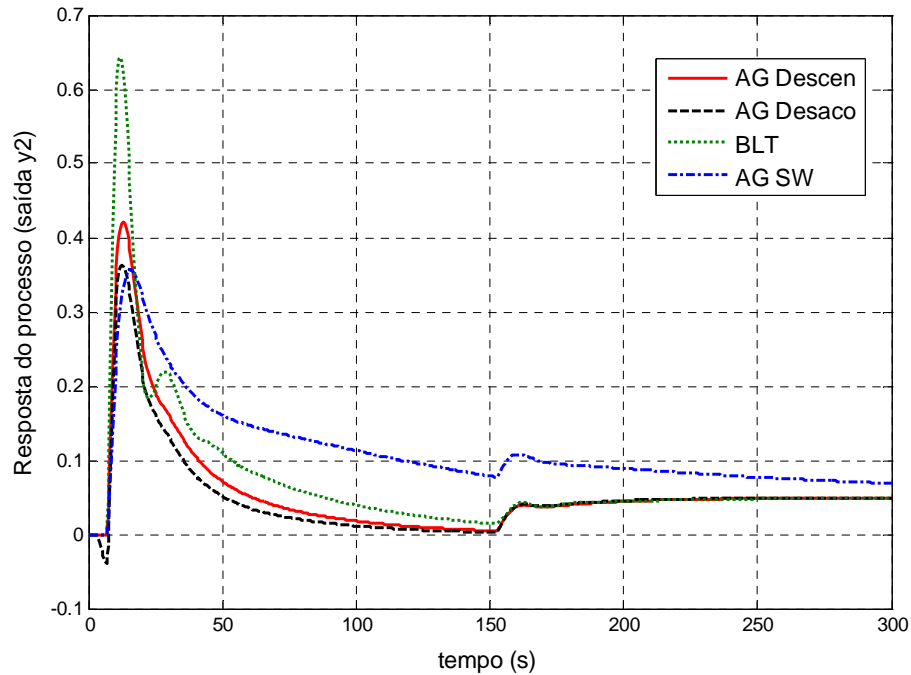
Para analisar o desempenho de cada método na eliminação dos distúrbios provocados pelas interações das malhas, um degrau de 0,96 é aplicado na entrada de referência x_1 no tempo igual a zero e um degrau de 0,05 é aplicado na entrada de referência x_2 no tempo igual a 150 s. Os gráficos 6.26 e 6.27 mostram as respostas das saídas y_1 e y_2 .

Gráfico 6.26: Resposta da saída y_1 com os métodos BLT e metaheurísticas.



Fonte: Elaborado pelo autor.

Gráfico 6.27: Resposta da saída y_2 com os métodos BLT e metaheurísticas.



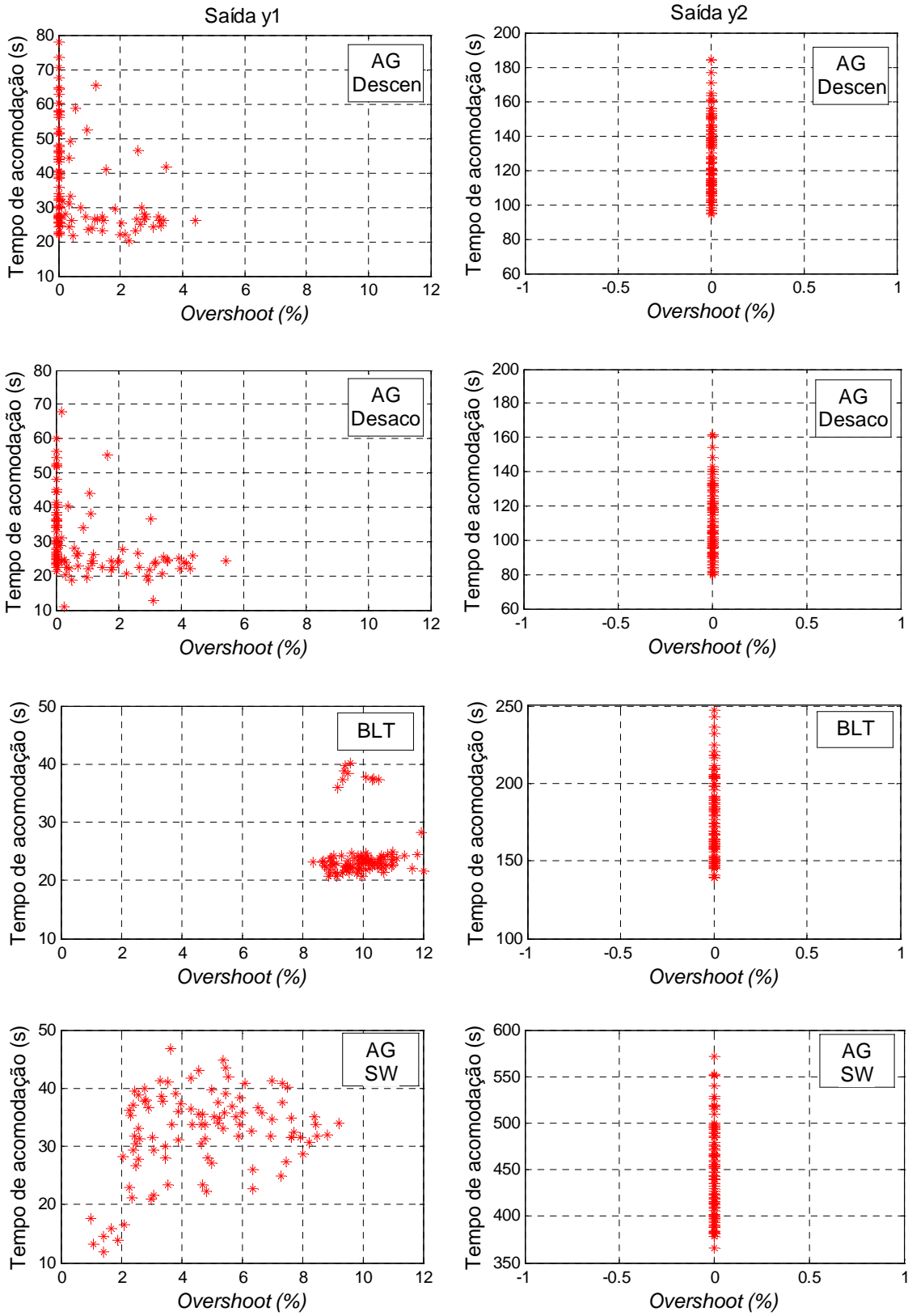
Fonte: Elaborado pelo autor.

Com relação aos resultados obtidos, as metodologias e arquiteturas apresentaram resultados similares quanto à eliminação dos distúrbios da saída y_1 . Para a saída y_2 observa-se no gráfico 6.27 que as metaheurísticas empregadas eliminam o distúrbio da interação das malhas, mais rápido que o AG SW e o método BLT. Em relação ao valor de *overshoot*, a arquitetura com desacopladores juntamente com o AG de Swiech apresentaram os menores valores enquanto que a arquitetura descentralizada e o método BLT apresentaram respectivamente valores de 13,5% e 75,6 % acima dos menores sobressinais.

6.3.4 Robustez do sistema MIMO 2 x 2

A segunda análise testou a robustez dos sistemas de controle através do método Monte Carlo já descrito anteriormente em Hu et al. (2011). O intervalo de perturbação dos parâmetros (K_{ij} , T_{ij} , θ_{ij}) do modelo da planta foi de $\pm 10\%$ e o número de experimentos estocásticos H foi de 100. O gráfico 6.28 mostra a robustez do desempenho do sistema no caso de existência de incerteza na planta controlada, avaliando os métodos quanto ao *overshoot* e tempo de acomodação.

Gráfico 6.28: Respostas do processo (saídas y_1 e y_2) com controladores PID sintonizados pelos métodos BLT e metaheurísticas.



Fonte: Elaborado pelo autor.

A tabela 6.40 apresenta os resultados numéricos do gráfico anterior.

Tabela 6.40: Robustez dos sistemas MIMO.

	Método de sintonia	OS (%)		t_s (s)	
		Faixa	Média	Faixa	Média
x_1y_1	AG Descen	0 - 4,40	0,73	20,36 - 77,97	37,40
	AG Desaco	0 - 5,46	1,03	10,77 - 67,90	30,16
	BLT	8,35 - 12,04	9,96	20,73 - 40,17	24,46
	AG SW	1,00 - 9,20	4,71	11,92 - 46,87	32,40
x_2y_2	AG Descen	0	0	94,13 - 184,11	128,31
	AG Desaco	0	0	79,79 - 161,46	110,66
	BLT	0	0	138,96 - 246,82	177,70
	AG SW	0	0	366,44 - 571,79	444,83

Fonte: Elaborado pelo autor.

Pode-se observar na tabela anterior que as sintonias com a metaheurística empregada obtiveram para a saída y_1 uma dispersão grande quanto ao tempo de acomodação e para a saída y_2 os menores valores. Quanto ao critério *overshoot*, as metaheurísticas deste trabalho obtiveram os menores valores para as duas saídas.

6.3.5 Conclusões da sintonia dos controladores multivariáveis

O estudo de caso três utilizou otimização multiobjetivo através da metaheurística AG para a determinação dos parâmetros de controladores PID em um sistema MIMO. A avaliação do emprego da metaheurística nas duas arquiteturas de sintonia foi efetuada pela comparação dos resultados obtidos por cada uma delas e, posteriormente, pela comparação destes com os resultados obtidos pelo método convencional BLT e pelo trabalho proposto por Swiech (2005).

A metodologia adotada para a determinação de uma solução a partir do conjunto ótimo de Pareto se mostrou muito eficiente. As simulações realizadas mostraram em relação aos critérios de desempenho bons resultados para a saída y_1 e os melhores resultados para a saída y_2 . Em relação à eliminação de distúrbios devido às interações das malhas e a robustez no caso de existência de incertezas na planta controlada, novamente para a saída y_2 , as técnicas empregadas neste trabalho apresentaram os melhores resultados.

6.4 Síntese dos Resultados

A avaliação experimental contemplou inicialmente o estudo de Algoritmos Genéticos e *Simulated Annealing* mono-objetivos que otimizassem os parâmetros de controladores em processos SISO. Foram avaliados no AG o tamanho da população, tipos e taxas dos operadores de cruzamento e mutação e no SA, a temperatura inicial e as funções de resfriamento e geração de novas soluções. Além disso, também foi avaliado o uso de diferentes índices de erro para a função objetivo: IAE, ISE, ITAE e ITSE. Pode-se afirmar que a eficiência das metaheurísticas está associada diretamente aos parâmetros e funções utilizados.

Os experimentos mostraram que no Algoritmo Genético, os melhores resultados da função custo foram obtidos com baixas taxas de uso do operador cruzamento e altas taxas de mutação, gerando ainda os menores desvios padrões. No SA, os melhores valores são obtidos quando utilizados como função de resfriamento a função logarítmica, embora o ajuste dos parâmetros não tenha melhorado de forma expressiva o desempenho do sistema. Definido os melhores parâmetros dos algoritmos, passou-se à simulação dos controladores através dos programas MATLAB® e Simulink® da Mathworks e em seguida estendendo para a abordagem multiobjetivo. A metodologia adotada no projeto foi eficaz na determinação dos parâmetros dos controladores PID.

No segundo estudo, para estruturas em cascata, foram apresentadas duas metodologias de sintonia. De acordo com os experimentos analisados, a abordagem 1 com AG (que consiste na otimização simultânea dos controladores) pôde ser considerada a mais eficiente em termos de melhor *overshoot* e tempo de acomodação.

Ao comparar os resultados dos dois primeiros estudos de casos, verificou-se que o algoritmo *Simulated Annealing* não obteve soluções de tão boa qualidade quanto à função custo, se comparado ao Algoritmo Genético. Uma possibilidade de melhoria e como continuidade do trabalho seria a utilização de uma heurística, por exemplo, o algoritmo GRASP - *Greedy Randomized Adaptive Search Procedure* (RESENDE; RIBEIRO, 2003) para determinar uma solução inicial para o SA, desta forma transformando o algoritmo em híbrido. Uma possível vantagem seria a redução do tempo de simulação, onde o SA em comparação ao AG é cerca de três vezes menor.

O estudo de caso 3 (controle multivariável) apresentou duas arquiteturas de controle e o uso de Algoritmos Genéticos Multiobjetivos para a determinação dos ganhos dos controladores. Após os ajustes dos parâmetros da metaheurística, as arquiteturas foram comparadas com uma metodologia convencional e outra metaheurística citada no estado da arte. Os valores das metaheurísticas empregadas apresentaram os melhores resultados na precisão das respostas das duas saídas e bons resultados quanto ao tempo de acomodação e na eliminação dos distúrbios provocados pelas interações das malhas.

7 CONCLUSÕES

A busca por melhores índices de desempenho nos processos industriais tais como redução do desgaste dos atuadores, melhoria da produtividade, maior segurança e minimização da intervenção humana passam por dois pontos: a aplicação de uma estratégia correta no controle de processos e o ajuste dos parâmetros dos controladores. O presente trabalho apresentou o uso de metaheurísticas, no caso os Algoritmos Genéticos e o *Simulated Annealing*, na determinação de parâmetros de controladores PID (K_p , K_i e K_d) para diferentes metodologias de sintonia e arquiteturas de controle com a intenção de verificar, em função de especificações de desempenho e robustez, a eficiência dos métodos empregados.

Neste trabalho, explorou-se a utilização, análise e comparação da metodologia empregada com diversos métodos de sintonia correntes na literatura e com trabalhos apresentados no estado da arte, onde cada um deles foi executado várias vezes e tiveram calculadas suas métricas de desempenho.

A metodologia empregada foi experimentar diferentes combinações dos parâmetros intrínsecos para cada metaheurística, após utilizar diferentes estruturas e arquiteturas de controle e formas de sintonia, na determinação dos parâmetros PID. Uma vantagem deste tipo de sintonização é a mudança do enfoque adotado nas malhas de controle, ou seja, pode-se alterar o objetivo a fim de obter um melhor desempenho no domínio do tempo, ou um menor esforço de controle ou um desempenho razoável na saída em conjunto com um menor esforço de controle.

Os estudos foram aplicados em três casos, nas estruturas mais comuns utilizadas em processos industriais: SISO, cascata e MIMO. No primeiro e terceiro estudo de caso, a abordagem foi estendida para a otimização multiobjetivo, compreendendo a obtenção do conjunto ótimo de Pareto. No primeiro estudo de caso foi utilizado como função objetivo, na abordagem com Frentes de Pareto, o erro (ITAE) da malha e o *overshoot* da saída e como critério de tomada de decisão o procedimento de Hu et al. (2011). No segundo estudo de caso foi utilizada uma função de agregação levando em consideração os erros (ITAE) de cada malha. O terceiro estudo de caso utilizou na função multiobjetivo os *overshoots* das variáveis do processo e o erro (ITSE) de uma das saídas e, como procedimento de escolha da solução os menores valores de *overshoot* das saídas.

Uma contribuição deste trabalho é sua utilização como uma ferramenta de otimização em problemas na área de Engenharia de Controle e Automação, comparando seus resultados aos procedimentos comumente utilizados na literatura de controle de sistemas dinâmicos. A dissertação é relevante por apresentar várias estruturas e arquiteturas e sintonizar seus parâmetros através das metaheurísticas, que podem ter sua função custo flexibilizada de forma a atender certos requisitos de controle. Outra contribuição é a utilização de um procedimento de Hu et al. (2011) para o decisor da otimização multiobjetivo, levando em consideração a robustez do sistema de controle.

Pode-se concluir, em relação às técnicas de otimização empregadas, que quando implementadas e aplicadas corretamente, produzem resultados satisfatórios nos diversos problemas abordados. Os experimentos mostraram que o Algoritmo Genético em relação ao *Simulated Annealing* apresentou maior eficiência na solução de parâmetros, gerando funções custo de menor valor e obtendo melhores resultados de desempenho dos sistemas dinâmicos. Ainda na comparação com os métodos tradicionais e outros correlatados no estado da arte, as

sintonias empregadas neste trabalho apresentaram resultados superiores, principalmente no domínio do tempo.

7.1 Trabalhos Futuros

Além da inserção de uma heurística para o SA, citada na síntese do capítulo anterior, sugere-se ainda como possíveis extensões desse trabalho:

- Uso de Redes Neurais Artificiais (RNA) para levantar o modelo do comportamento dinâmico dos processos, e fornecer previsões para os controladores;
- Realização de outros testes com o objetivo de melhorar os algoritmos empregados, por exemplo, mudança do tipo de seleção, outros operadores de cruzamento e mutação no AG e outras funções de resfriamento para o SA;
- Utilização das metaheurísticas em outras estratégias de controle como na otimização das funções de peso do controle H_{infinito} para incertezas descritas na forma coprima (FC H_{infinito}) e filtros do controle H_{infinito} pelas funções de sensibilidade mista (SM H_{infinito}) (COSTA et al., 2010);
- Avaliação de outras metaheurísticas tais como Busca Tabu, Colônia de Formigas, Otimização por Enxame de Partículas e métodos híbridos, para a solução do problema de sintonia. Realizar ainda o ajuste correto dos parâmetros intrínsecos dos algoritmos através de testes e simulações;
- Utilização de controle fuzzy sintonizado finamente por alguma metaheurística citada no item anterior.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- ABRAHAN, A.; JAIN, L. C.; GOLDBERG, R. **Evolutionary multiobjective optimization: theoretical Advances and applications**. Springer. London, 2004.
- ALBERTOS, P. P.; SALAS, A. **Multivariable control systems: an engineering approach**. Springer. London, 2004.
- ARROYO, J. C. **Heurísticas e metaheurísticas para otimização combinatória multiobjetivo**. 2002. 256p. Tese (Doutorado). Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas, Campinas, SP, 2002.
- ASHLOCK, D. **Evolutionary Computation for Modeling and Optimization**. Springer, 2005.
- ÅSTRÖM, K. J.; HÄGGLUND, T. **PID controllers: Theory, Design and Tuning**. 2. ed. North Carolina: Instrument Society of America, 1995.
- ÅSTRÖM, K. J., WITTENMARK, B. **Adaptive control**. 2. ed. Addison-Wesley, 1995.
- BARBOSA, J. A. A. **Determinação dos parâmetros do controlador preditivo generalizado com função de pesos variante no tempo através de algoritmos genéticos**. 2007. 118p. Dissertação (Mestrado). Universidade Federal do Rio Grande do Norte, Natal, RN, 2007.
- BARCELLOS, J. C. H. **Algoritmos genéticos adaptativos: um estudo comparativo**. 2000. 132 p. Dissertação (Mestrado). Escola Politécnica. Universidade Estadual de São Paulo, São Paulo, SP, 2000.
- BAZANELLA, A.S.; GOMES DA SILVA, J. **Sistemas de Controle: princípios e métodos de projeto**. Editora UFRGS, Porto Alegre, 2005.
- BENNET, S. **A Brief History of Automatic Control**. *Control Systems*. IEEE, vol. 16, no. 3, pp. 17 - 25, 1996.
- BLUM, C.; ROLI, A. **Metaheuristics in combinatorial optimization: Overview and Conceptual Comparison**. *ACM Computing Surveys*, vol. 35, no. 3, pp. 268-308. 2003.
- BOLTON, W. **Engenharia de Controle Moderno**. Makron Books do Brasil. São Paulo, 1995.
- BRISTOL, E. **On a new measure of interaction for multivariable process control**. *IEEE Trans. Aut. Contr.* 11, pp. 133-134, 1966.
- CAMPESTRINI, L. **Sintonia de controladores PID descentralizados baseada no método do ponto crítico**. 2006. 97 p. Dissertação (Mestrado). Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2006.

- CARMO, M. J. **Ambiente educacional multifuncional integrado para sintonia e avaliação do desempenho de malhas industriais de controle.** 2006. Dissertação (Mestrado). Universidade Federal de Juiz de Fora, Juiz de Fora, MG, 2006.
- CARVALHO, J. L. M. **Sistemas de controle automático.** LTC. Rio de Janeiro, RJ, 2000.
- CASTRO, R. E. **Otimização de estruturas, com multiobjetivos via algoritmo genético de Pareto.** 2001. 202 p. Tese (Doutorado). Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2001.
- CHANG, W. D. **A multi-crossover genetic approach to multivariable PID controllers tuning.** Expert Systems with Applications, 33, pp. 620-626, 2007.
- CHEN, B.-S.; CHENG, Y.-M. **A structure-specified H_{∞} optimal control design for practical applications: A genetic approach.** IEEE Trans. Control Syst. Tech., vol. 6, no. 6, pp. 707-718, 1998.
- CHIDAMBARAM, M.; SREE, R.P. **A simple method of tuning PID controllers for integrator/dead time processes.** Comput. Chem. Eng., 27, pp. 211-215, 2003.
- CHIEN, K. L.; HRONES, A.; RESWICK, J. B. **On the automatic control of generalized passive systems.** Trans. ASME, vol. 74, pp. 175-185, 1952.
- COELLO, C. A. C.; PULIDO, G. T.; MONTES, E. M. **Current and future research trends in evolutionary multiobjective optimization.** Advanced Information and Knowledge Processing - Industrial Applications to Academic Speculations, pp. 213-231, 2005.
- COELLO, C.; VELDHUIZEN, D.; LAMONT, G. **Evolutionary Algorithms for Solving Multi-Objective Problems.** 2. ed. Springer-Verlag, New York, NY, 2007.
- COHEN, H.; COON, G. A. **Theoretical consideration of retarded control.** Trans. Asme, vol. 75, pp. 827-834, 1953.
- CORNE, D.; KNOWLES, J.; OATES, M. **The Pareto Envelope-Based Selection Algorithm for Multi-Objective Optimization.** In Proceedings of The Sixth International Conference on Parallel Problem Solving from Nature VI (PPSN-VI), pp. 839-848, 2000.
- CORREIA, M. H.; OLIVEIRA, J. F.; FERREIRA, J. S. **Cylinder packing by simulated annealing.** Pesquisa Operacional, vol. 20, no. 2, p. 269-286, 2000.
- COSTA, M. H.; SILVA, V. V. R.; CANDÉ, S. **Seleção da Estratégia de Controle através da Otimização On-Line de Parâmetros para um Sistema Não Linear.** Computational Intelligence Techniques for Optimization and Data Modeling, no. 94, vol. XXIX, pp. 9193-9203, 2010.
- CRUZ, J. J. **Controle Robusto Multivariável: o método LQG/LTR.** Universidade de São Paulo, São Paulo, 1996.
- DAVIS, L. **Handbook of Genetic Algorithms.** Van Nostrand Reinhold. New York, 1991.

- DE JONG, K. A. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems.** PhD thesis, University of Michigan, Ann Arbor, MI, Dissertation Abstracts International 36 (10), 1975.
- DEB, K. **Multi-objective optimization using evolutionary algorithms.** John Wiley & Sons. New York, 2001.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. **A fast and elitist multiobjective genetic algorithm: NSGA II.** IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.
- DORF, R. C.; BISHOP, R. H. **Modern control systems.** 8. ed. California: Addison-Wesley, 1998.
- DORIGO, M.; STÜTZLE, T. **Ant Colony optimization.** Cambridge: MIT Press, 2004.
- DUMONT, G. A.; HUZMEZAN, M. **Concepts, Methods and Techniques in Adaptive Control.** American Control Conference, 2002.
- FALCONE, M. A. G. **Estudo comparativo entre Algoritmos Genéticos e Evolução Diferencial para otimização de um modelo de cadeia de suprimento simplificada.** 2004. 81 p. Dissertação (Mestrado). Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2004.
- FONSECA, C.M. ; FLEMING, P.J. **Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization.** Proceedings of the Fifth International Conference on Genetic Algorithms, (Forrest, Ed.), pp. 416-423, 1993.
- GANAPATHY, K.; JEROME, J. **Control of dead time systems using improved tabu search.** International Journal of Soft Computing 5, pp. 177-184, 2010.
- GUERRA, C. **Otimização paramétrica de estruturas treliçadas por Algoritmos Genéticos.** 2008. 111 p. Dissertação (Mestrado). Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2008.
- GLOVER, F.; KOCHENBERGER, G. A. **Handbook of metaheuristics.** Kluwer Academic. Boston, 2003.
- GLOVER, F.; LAGUNA, M. **Tabu Search.** Kluwer Academic Publishers, 1997.
- GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning.** Addison-Wesley, Massachusetts, 1989.
- GOLNARAGHI, F.; KUO, B. C. **Automatic Control Systems.** 9. ed. John Wiley & Sons, 2009.
- GOMES, H. A. S. **Utilização da metaheurística Simulated Annealing no problema de alocação de pessoal em empresas de transporte coletivo por ônibus.** 2003. 137 p. Dissertação (Mestrado). Universidade Federal do Ceará, Fortaleza, CE, 2003.

HAI-BIN, D.; DAO-BO, W.; XIU-FEN, Y. **Novel Approach to Nonlinear PID Parameter Optimization Using Ant Colony Optimization Algorithm**. Journal of Bionic Engineering vol. 3 issue 2, pp. 73-78, 2006.

HASHIMOTO, K. **Técnicas de otimização combinatória multiobjetivo aplicadas na estimação do desempenho elétrico de redes de distribuição**. 2004. 113 p. Tese (Doutorado). Escola Politécnica. Universidade de São Paulo, São Paulo, SP, 2004.

HO, S.-J.; HO, S.-Y.; SHU, L.-S. **OSA: Orthogonal simulated annealing algorithm and its application to designing mixed H₂/H_∞ optimal controllers**. IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 34, no. 5, pp. 588-600, 2004.

HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. **A Niche Pareto Genetic Algorithm for Multiobjective Optimization**. First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 82-87, 1994.

HOVD, M.; SKOGESTAD, S. **Sequential Design of Decentralized Controllers**. Automatica, 30, pp. 1601-1607, 1994.

HU, Z.; LI, D.; WANG, J.; XUE, F. **Analytical Design of PID Decoupling Control for TITO Processes with Time Delays**. Journal of Computers vol. 6, no. 6, pp. 1064-1070, 2011.

HUNG, M.; SHU, L.; HO, S.; HWANG, S.; HO, S. **A Novel Intelligent Multiobjective Simulated Annealing Algorithm for Designing Robust PID Controllers**. IEEE Transactions on systems, man, and cybernetics - part a: systems and humans, vol.38, no. 2, 2008.

INGBER, L. **Very fast simulated re-annealing**. J. Math. and Comput. Modelling, vol.12, no. 8, pp. 967-973, 1989.

IRUTHAYARAJAN, M. W.; BASKAR S. **Evolutionary algorithms based design of multivariable PID controller**. Expert Systems with Applications, vol.36, no. 9, pp. 9159-9167, 2009.

JOHNSON, M. A.; MORADI, M. H. **PID Control - New Identification and Design Methods**. Springer, Berlin-Heidelberg, 2005.

JULSTROM, B.A. **Comparing darwinian, baldwinian, and lamarckian search in a genetic algorithm for the 4-cycle problem**. Genetic and Evolutionary Computation Conference, pp. 134-138, 1999.

KENNEDY, J.; EBERHART R.C. **Particle swarm optimization**. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Austrália, pp. 1942-1948, 1995.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI M. P. **Optimization by Simulated Annealing**. Science, vol. 220, no. 4598, pp. 671-680, 1983.

KITSIOS, I.; PIMENIDES, T.; GROUMPOS, P. **A genetic algorithm for designing H_∞ structured specified controllers**. In Proc. IEEE Int. Conf. Control Appl., Mexico, pp. 1196-1201, 2001.

KNOWLES, J. D.; CORNE, D. W. **The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation**. Congress on Evolutionary Computation, pp. 98-105, Washington, 1999.

LEE, Y.; PARK, S.; LEE, M. **PID controller tuning to obtain desired closed loop responses for cascade control systems**. Ind. Eng. Chem. Res., 37, pp. 1859-1865, 1998.

LEONHARD, W. **Control of electrical drives**. 3. ed. Springer. Berlin, 2001.

LIMA, F. R. G. **Análise comparativa de estratégias de controle modernas aplicadas no controle de velocidade de um servomotor**. 2008. 193 p. Dissertação (mestrado). Instituto Mauá de Tecnologia, São Caetano do Sul, SP, 2008.

LOH, A. P.; HANG, C. C.; QUEK, C. K.; VASNANI, V. U. **Autotuning of multi-loop proportional - integral controllers using relay feedback**. In. Eng. Chem. Res., vol. 32, pp. 1102-1107, 1993.

LÓPEZ, A. M.; MILLER, J. A.; SMITH, C. L.; MURRIL, P. W. **Tuning controllers with error-integral criteria**. In: Instrumentation Technology, 1967.

LUTZ, H.; WENDT, W. **Taschenbuch der Regelungstechnik**. 3. ed. Harri Deutsch. Frankfurt Am Main, 2000.

LUUS, R.; JAAKOLA, T.H.I. **Optimization by direct search and systematic reduction of the size of search region**. AIChE Journal, vol. 19, no. 4, pp. 760-766, 1973.

LUYBEN, W. L. **Simple method for tuning SISO controllers in multivariable systems**. Ind. Eng. Chem. Process Des. Dev., vol.25, pp. 654-660, 1986.

LUYBEN, W. L. **Process Modeling, Simulation and Control for Chemical Engineers**. 2. ed. McGraw Hill, New York, 1990.

LUZ, E. F. P. **Metaheurísticas paralelas na solução de problemas inversos**. 2012. 131 p. Tese (Doutorado), Instituto Nacional de Pesquisas Espaciais. São José dos Campos, SP, 2012.

MAN, K. F.; TANG, K. S.; KWONG, S. **Genetic Algorithms: Concepts and Applications**. IEEE Transactions on Industrial Electronics, vol. 43, no. 5, pp. 519-534, 1996.

MARLIN, T.E. **Process Control: Designing Processes and Control systems for Dynamic Performance**. McGraw Hill, New York, 1995.

MAYNE, D.Q. **The design of linear multivariable systems**. Automatica 9, pp. 201-207, 1973.

MAZZINI, H. M. **Estudo sobre a compensação de processos integradores com atraso e a proposição de uma nova abordagem**. 2003. 132 p. Tese (Doutorado), Universidade Federal de Uberlândia, Uberlândia, MG, 2003.

MELOTTI, G.; BAETA, B. S.; MAZZINI, H. M.; MAGELA NETO, O. **Sintonia de um controle PID para o sistema mola-amortecedor da suspensão de um veículo usando**

Otimização Multiobjetivo. Anais: XVII Congresso Brasileiro de Automática, Juiz de Fora, MG, 2008.

METROPOLIS, N.; ROSENBLUTH, A.W.; ROSENBLUTH, M.N.; TELLER, A.H.; TELLER, E. **Equation of state calculations by fast computing machines**, J. Chem. Phys. 21 (6), 1087-1092, 1953.

MICHALEWICZ, Z. **Genetic algorithms + data structures = evolution programs**. 3. ed. Springer-Verlag, New York, 1996.

MICHALEWICZ, Z.; LOGAN, T.; SWAMINATHAN, S. **Evolutionary operators for continuous convex parameter spaces**. In Proceedings of the 3rd Annual Conference on Evolutionary Programming, pp. 84-97. World Scientific, 1994.

MICHALEWICZ, Z.; SCHOENAUER, M. **Evolutionary Algorithms for Constrained Parameter Optimization Problems**, *Evolutionary Computation*. Vol. 4, no. 1, pp. 1-32, 1996.

MITCHELL, M. **An introduction to genetic algorithms**. Mit Press. Cambridge, 1997.

MOEDINGER, L. H. **Algoritmos Evolutivos e Inteligência Coletiva aplicados a problemas de otimização não-linear com restrições: fundamentos e estudo comparativo**. 2005. 77 p. Dissertação (Mestrado). Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2005.

MOROMISATO, G. D. Y.; MELOTTI, G.; GARCIA, F. A.; MAZZINI, H. M.; NEPOMUCENO, E. G. **A utilização de um software livre no ensino de sistemas de controle**. International Conference on Engineering and Computer Education. 1: pp. 597-601, 2007.

MURRAY, A.T. **Spatial restriction in harvest scheduling**. Forest Science, USA, v. 45, no. 1, pp. 45-52, 1999.

NAGARAJ, B.; VIJAYAKUMAR, P. **A Comparative Study of PID Controller Tuning using GA, EP, PSO and ACO**. Journal of Automation, Mobile Robotics & Intelligent Systems, vol. 5, no. 2, 2011.

NASRI, M.; NEZAMABADI, H.; MAGHFOORI, M. **A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor**. Proceeding Of World Academy of Science, Engineering and Technology, no. 26, pp. 211-215, 2007.

NEMHAUSER, G. L.; WOLSEY, L.A. **Integer and combinatorial optimization**. New York: Willey & Sons, 1999.

OGATA, K. **Engenharia de controle moderno**. 4. ed. Prentice-Hall. New Jersey, 1997.

OGUNNAIKE, B.; RAY, W. **Process Dynamics, Modeling, and Control**. Oxford University Press, New York, 1994.

PAIS, M. S.; FERREIRA, J. C.; TEIXEIRA, M. B.; YAMANAKA, K.; CARRIJO, G. A. **Cost optimization of a localized irrigation system using genetic algorithms.** Proceedings of the 11th International Conference on Intelligent Data Engineering and Automated Learning, pp. 29-36, 2010.

POPOV, A.; FARAG, A.; WERNER, H. **Tuning of a PID controller using a multi-objective optimization technique applied to a neutralization Plant.** European Control Conference, Seville, Spain, 2005.

POHEKAR, S. D.; RAMACHANDRAN, M. **Application of Multi-criteria Decision Making to Sustainable Energy Planning - A Review.** Renewable and Sustainable Energy Reviews, vol. 8, pp. 365-381, 2004.

REEVES, C. R. **Modern Heuristic Techniques for Combinatorial Problems.** Blackwell, 1993.

RESENDE, M.G.C.; RIBEIRO, C.C. **Greedy randomized adaptive search procedures.** In F. Glover and G. Kochenberger, editors, Handbook of Metaheuristics, pp. 219-249, Kluwer Academic Publishers, 2003.

RIBEIRO FILHO, J.L., TRELEAVEN, P.C., ALIPPI, C. **Genetic algorithm programming environments.** IEEE, computer, vol. 27, pp. 28-43, 1994.

ROEVA, O. **A Comparison of Simulated Annealing and Genetic Algorithm Approaches for Cultivation Model Identification.** 8th IMACS Seminar on Monte Carlo Methods, Borovets, Bulgaria, 2011.

ROTSTEIN, G.E.; LEWIN, D.R. **Simple PI and PID tuning for open-loop unstable systems.** Ind. Eng. Chem. Res., 30, pp. 1864-1875, 1991.

SADASIVARAO, M. V.; CHIDAMBARAM, M. **PID Controller tuning of cascade control systems using genetic algorithm.** Journal Indian Institute of Science, no. 86, pp. 343-354, 2006.

SANTA CATARINA, A. **SAHGA - Um algoritmo genético híbrido com representação explícita de relacionamentos espaciais para análise de dados geoespaciais.** 2009. 112p. Tese (Doutorado). Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, 2009.

SCHAFFER, J. **Multiple objective optimization with vector evaluated genetic algorithms. Genetic Algorithms and their Applications.** First International Conference on Genetic Algorithms, v. I, pp. 93-100, 1985.

SCHUMACHER, W.; LEONHARD, W. **Grundlagen der Regelungstechnik.** Technische Universität Braunschweig, 2006.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. **Process Dynamics and Control,** John Wiley & Sons, 1989.

SKOGESTAD, S.; POSTLETHWAITE, I. **Multivariable feedback control analysis and design.** Chichester, UK: John Wiley & Sons, 1996.

- SOUZA, D. A. **Otimização pelo método dos Algoritmos Genéticos e dimensionamento de estruturas tubulares metálicas espaciais com barras cruzadas para coberturas**. 2005. 125 p. Tese (Doutorado). Universidade Federal de Uberlândia, Uberlândia, MG, 2005.
- SRINIVAS, N.; DEB, K. **Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms**. *Evolutionary Computation*, vol. 2, pp. 221-248, 1995.
- SWIECH, M.C.S. **Algoritmos Genéticos para a sintonia simultânea de múltiplos controladores em processos de refino**. 2005. 127 p. Dissertação (Mestrado). Universidade Tecnológica Federal do Paraná, Curitiba, PR, 2005.
- SZU, H.; HARTLEY, R. **Fast Simulated Annealing**. *Phys. Lett. A*, vol. 122, pp. 157-162, 1987.
- TALBI, E.G. **Metaheuristics: From Design to Implementation**, John Wiley & Sons, Inc., New Jersey. 2009.
- TAN, K.C.; LEE, L. H.; ZHU, Q. L.; OU, K. **Heuristic methods for vehicle routing problem with time windows**. *Artificial Intelligence in Engineering*, vol. 15, pp. 281-295, 2001.
- TICONA, W. G. C. **Aplicação de algoritmos genéticos Multiobjetivo para alinhamento de sequências biológicas**. 2003. 129 p. Dissertação (Mestrado). Instituto de Ciências Matemáticas e de Computação (ICMC). Universidade de São Paulo, São Paulo, SP, 2003.
- TORRES, B. S. **Sintonia de controladores PID em um sistema multi-malha**. 2002. 192 p. Dissertação (Mestrado). Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2002.
- TRIERWEILER, J. O. ; POSSER, M. S. **PID-Toolbox: Uma Ferramenta Para o Ensino e Ajuste de Controladores PIDs**. Apostila, Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2002.
- TYREUS, B.D.; LUYBEN, W.L. **Tuning PI controllers for integrator/dead time processes**. *Ind. Eng. Chem. Res.*, 31, pp. 2625-2628, 1992.
- VAN VELDHUIZEN D.A.; LAMONT, G.B. **Multiobjective evolutionary algorithms: Analysing the state-of-the-art**. *Evolutionary Computation* 8, no. 2, pp. 125-147, 2000.
- VAROL, H. A.; BINGUL, Z. **A New PID Tuning Technique Using Ant Algorithm**. *American Control Conference*, Boston, USA, pp.2154-2159, 2004.
- VISIOLI, A. **Optimal tuning of PID controllers for integral and unstable processes**. *IEEE Proc. Control Theory Appl.*, 148, pp. 180-184, 2001.
- VU, T. N. L.; LEE, M. **Multi-loop PI/PID Controller Design Based on Direct Synthesis for Multivariable Systems**. *World Congress on Engineering and Computer Science (WCECS)*, San Francisco, USA, 2008.
- WANG, J.; JIN, G.; WANG, Y.; CHEN, X. **Genetic Simulated Annealing Algorithm Used for PID Parameters Optimization**. *International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1, pp. 397-401, 2009.

WEISE, T. **Global optimization algorithms: theory and application**. Kassel: University of Kassel, 2009. Disponível em: www.it-weise.de/projects/book.pdf. Acesso em: 4 de agosto de 2012.

WOOD, R. K.; BERRY, M. W. **Terminal composition of a binary distillation column**. Chemical. Engineering Sciences, vol. 28, pp. 1707-1717, 1973.

YAMAMOTO, L. **Uso do Simulated Annealing e Algoritmo Genético no problema da reconfiguração de uma rede de distribuição de energia elétrica**. 2004. 92 p. Dissertação (Mestrado). Universidade Federal do Paraná, Curitiba, PR, 2004.

ZBIGNIEW, M.; FOGEL, D.B. **How to Solve it: Modern Heuristics**. Springer-Verlag, 2000.

ZIEGLER, J. B.; NICHOLS, N. B. **Optimum settings for automatic controls**. ASME Transactions, vol. 64, pp. 759-768, 1942.

ZINI, E. O. C. **Algoritmo Genético especializado na resolução de problemas com variáveis contínuas e altamente restritos**. 2009. 149 p. Dissertação (Mestrado). Universidade Estadual Paulista, Ilha Solteira, SP, 2009.

ZITZLER, E.; DEB, K.; THIELE, L. **Comparison of multiobjective evolutionary algorithms: Empirical results**. Evolutionary Computation 8, pp. 173-195, 2000.

ZITZLER, E.; THIELE, L. **An evolutionary algorithm for multiobjective optimization: The strength pareto approach**. Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, 1998.

APÊNDICE A - *Scripts* para determinação dos parâmetros PID

Os anexos A.1 a A.30 apresentam os códigos-fonte e ambientes de simulação implementados para cada estrutura de controle e metodologia empregada na otimização dos parâmetros PID.

A.1: Código-fonte (MATLAB) do AG para controle PID de processos SISO

```
% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística AG aplicada no controle monovariável
% *****

num_var = 3;
Funcao_Objetivo = @(x)Funcao_Objetivo(x);
options = gaoptimset;
lb = [0 0 0];
ub = [4.5 0.45 22.5];
options = gaoptimset(options,'PopInitRange',[0; 22.5]);
options = gaoptimset(options,'PopulationSize',100);
options = gaoptimset(options,'PopulationType','doubleVector');
options = gaoptimset(options,'EliteCount',1);
options = gaoptimset(options,'Generations',100);
options = gaoptimset(options,'CrossoverFraction',0.4);
options = gaoptimset(options,'CrossoverFcn',{ @crossoverheuristic });
options = gaoptimset(options,'SelectionFcn',{ @selectiontournament });
options = gaoptimset(options,'MutationFcn',{ @mutationadaptfeasible });
options = gaoptimset(options,'PlotFcns',{ @gaplotbestf @gaplotbestindiv });
options = gaoptimset(options,'Display','iter');

[X,FVAL] = ga(Funcao_Objetivo,num_var,[],[],[],[],lb,ub,[],options);

Kp=X(1);
Ki=X(2);
Kd=X(3);

Kp
Ki
Kd
```

A.2: Código-fonte (MATLAB) do SA para controle de processos SISO

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística SA aplicada no controle monovariável
% *****

Funcao_Objetivo = @(x)Funcao_Objetivo(x);
options = saoptimset;
lb = [0 0 0];
ub = [4.5 0.45 22.5];
a = 4.5 * rand(1);
b = 0.45 * rand(1);
c = 22.5 * rand(1);
x0 = [a b c];
options = saoptimset(options,'InitialTemperature', 100);
options = saoptimset(options,'MaxIter', 3000);
options = saoptimset(options,'AnnealingFcn', @annealingboltz );
options = saoptimset(options,'TemperatureFcn',@temperatureboltz);
options = saoptimset(options,'PlotFcns', { @splotbestf,@splotbestx });
options = saoptimset(options,'Display' , 'iter');

[X,FVAL] = simulannealbnd(Funcao_Objetivo, x0, lb, ub, options);

Kp=X(1);
Ki=X(2);
Kd=X(3);

Kp
Ki
Kd

```

A.3: Código-fonte (MATLAB) da Função objetivo (mono-objetivo) - controle monovariável

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo - controle monovariável
% *****

function fitness = Funcao_Objetivo(x)
Kp=x(1);
Ki=x(2);
Kd=x(3);
options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_simulink_Varol',[0 250], options);
e = load ('Erro_1.mat');
err(1,:) = e.e(2,:);
erro = err' ;
erro = erro.*tout;
erro = abs(erro);
erro = sum(erro);
erro = erro;
fitness = erro;

```

A.4: Código-fonte (MATLAB) do AG Multiobjetivo para controle monovariável

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística AG Multiobjetivo aplicada ao controle PID de processo monovariável
% *****

clc
clear

numberOfVariables = 3;
fitnessFunction = @(x)Funcao_Objetivo(x);

options = gaoptimset;
lb = [0 0 0];
ub = [4.5 0.5 22.5];

options = gaoptimset(options,'PopInitRange',[0; 22.5]);
options = gaoptimset(options,'PopulationSize',100);
options = gaoptimset(options,'PopulationType','doubleVector');
options = gaoptimset(options,'EliteCount',1);
options = gaoptimset(options,'Generations',100);
options = gaoptimset(options,'CrossoverFraction',0.4);
options = gaoptimset(options,'ParetoFraction',0.3);
options = gaoptimset(options,'CrossoverFcn',{@crossoverheuristic});
options = gaoptimset(options,'SelectionFcn',{@selectiontournament});
options = gaoptimset(options,'MutationFcn',{@mutationadaptfeasible});
options = gaoptimset(options,'PlotFcns',{@gaplotpareto,@gaplotscorediversity});
options = gaoptimset(options,'Display','iter');

[X,FVAL] = gamultiobj(fitnessFunction,numberOfVariables,[],[],[],[],lb,ub,options);

```

A.5: Código-fonte (MATLAB) da Função objetivo (Pareto) - sistema monovariável

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo - Frentes de Pareto na sintonia de controlador PID de processo monovariável
% *****

function f = Funcao_Objetivo(x)
f=[];

xSize = size(x,1);
numObj = 2;
f = zeros(xSize, numObj);

Kp=x(1);
Ki=x(2);
Kd=x(3);

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_Pareto',[0 150], options);

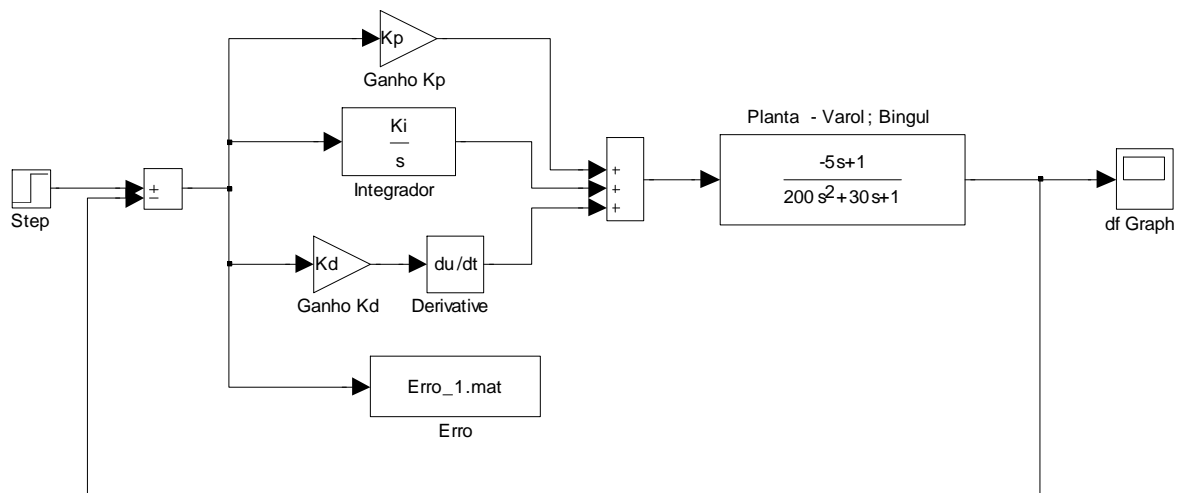
e = load ('Erro_1.mat');
err(1,:) = e.e(2,:);
error = err';
error = error.*tout;
error = abs(error);
error = sum(error);

s = stepinfo(Pareto_MIMO.signals.values(1:300,1),Pareto_MIMO.time(1:300,1),1);
b = s.Overshoot;
error2 = b;      % Overshoot

f(:,2) = error;
f(:,1) = error2;

```

A.6: Ambiente de simulação (*Simulink*) do controlador PID - processo SISO



A.7: Código-fonte (MATLAB) para plotagem dos gráficos

```
% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Plotagem - processo SISO
% *****

plot(SISO_IAE_ISE_ITAE.time(1:775,1),SISO_IAE_ISE_ITAE.signals.values(1:775,1:3));
```

A.8: Código-fonte (MATLAB) para determinação dos critérios de desempenho no domínio do tempo - processo SISO

```
% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Desempenho no domínio do tempo - processo SISO
% *****

clear
clc

Kp = 3.6685;
Ki = 0.0767;
Kd = 19.9614;
options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_simulink_Varol',[0 100], options);
AG = stepinfo(yout,tout,1);
AG
```


A.9: Código-fonte (MATLAB) para determinação dos índices de desempenho e robustez - processo SISO

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Índice de desempenho e robustez - processo SISO
% *****

clc
clear
disp('ERRO - IAE');

e1 = load ('Erro_AG.mat');
time(1,:)=e1.e(1,:);
err_AG(1,:) = e1.e(2,:);
error_AG = err_AG' ;
error_AG = abs(error_AG);
error_AG = sum(error_AG);
error_IAE_AG = error_AG;
error_IAE_AG

e4 = load ('Erro_ZN.mat');
err_ZN(1,:) = e4.e(2,:);
erro_ZN = err_ZN' ;
erro_ZN = abs(erro_ZN);
erro_ZN = sum(erro_ZN);
erro_IAE_ZN=erro_ZN;
erro_IAE_ZN

e17 = load ('Erro_ACO.mat');
err_ACO(1,:) = e17.e(2,:);
erro_ACO = err_ACO' ;
erro_ACO = abs(erro_ACO);
erro_ACO = sum(erro_ACO);
erro_IAE_ACO = erro_ACO;
erro_IAE_ACO

disp(' ');
disp('ROBUSTEZ - ALTERAÇÃO 1');

e5 = load ('Rob_1_AG.mat');
err_AG(1,:) = e5.e(2,:);
erro_AG = err_AG' ;
erro_AG = abs(erro_AG);
erro_AG = sum(erro_AG);
erro_Rob_1_AG = erro_AG;
erro_Rob_1_AG

e7 = load ('Rob_1_ZN.mat');
err_ZN(1,:) = e7.e(2,:);
erro_ZN = err_ZN' ;
erro_ZN = abs(erro_ZN);
erro_ZN = sum(erro_ZN);
erro_Rob_1_ZN = erro_ZN;
erro_Rob_1_ZN

e16 = load ('Rob_1_ACO.mat');
err_ACO(1,:) = e16.e(2,:);
erro_ACO = err_ACO' ;

```

```

erro_ACO = abs(erro_ACO);
erro_ACO = sum(erro_ACO);
erro_Rob_1_ACO = erro_ACO;
erro_Rob_1_ACO

disp(' ');
disp('ROBUSTEZ - ALTERAÇÃO 2');

e8 = load ('Rob_2_AG.mat');
err_AG(1,:) = e8.e(2,:);
erro_AG = err_AG' ;
erro_AG = abs(erro_AG);
erro_AG = sum(erro_AG);
erro_Rob_2_AG = erro_AG;
erro_Rob_2_AG

e10 = load ('Rob_2_ZN.mat');
err_ZN(1,:) = e10.e(2,:);
erro_ZN = err_ZN' ;
erro_ZN = abs(erro_ZN);
erro_ZN = sum(erro_ZN);
erro_Rob_2_ZN = erro_ZN;
erro_Rob_2_ZN

e15 = load ('Rob_2_ACO.mat');
err_ACO(1,:) = e15.e(2,:);
erro_ACO = err_ACO' ;
erro_ACO = abs(erro_ACO);
erro_ACO = sum(erro_ACO);
erro_Rob_2_ACO = erro_ACO;
erro_Rob_2_ACO

disp(' ');
disp('ROBUSTEZ - ALTERAÇÃO 3');

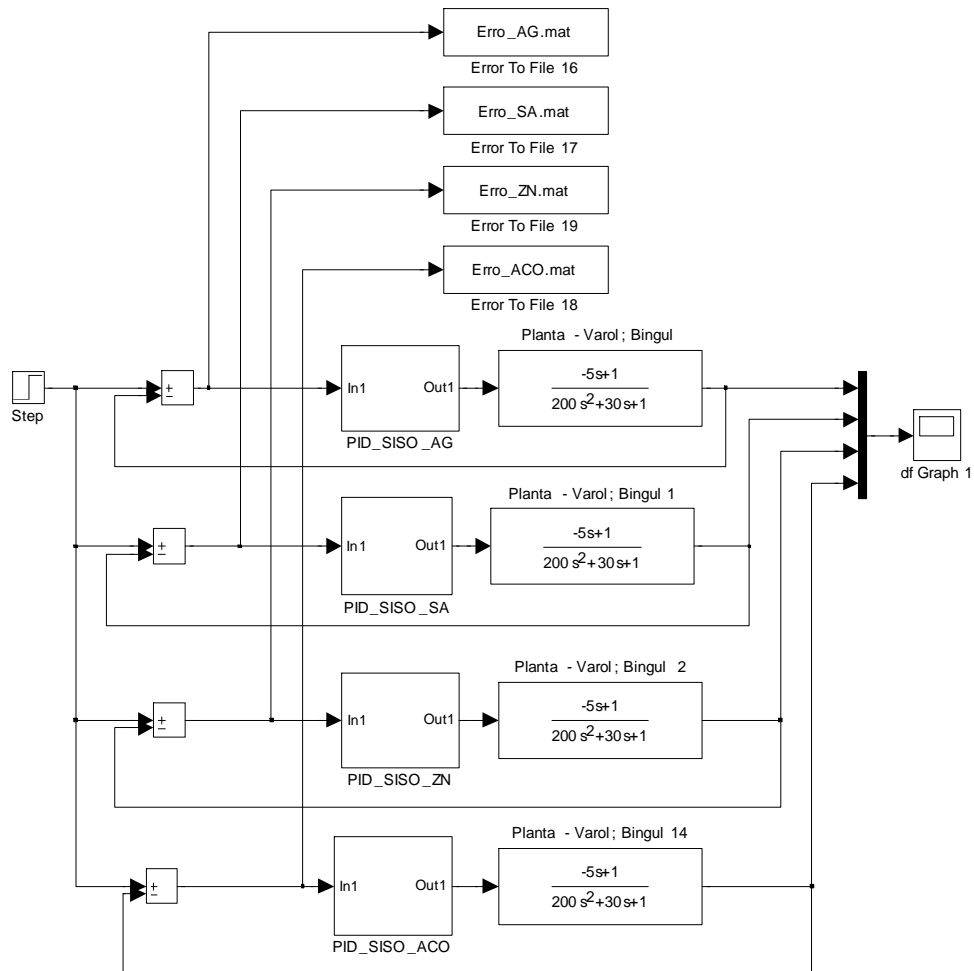
e11 = load ('Rob_3_AG.mat');
err_AG(1,:) = e11.e(2,:);
erro_AG = err_AG' ;
erro_AG = abs(erro_AG);
erro_AG = sum(erro_AG);
erro_Rob_3_AG = erro_AG;
erro_Rob_3_AG

e13 = load ('Rob_3_ZN.mat');
err_ZN(1,:) = e13.e(2,:);
erro_ZN = err_ZN' ;
erro_ZN = abs(erro_ZN);
erro_ZN = sum(erro_ZN);
erro_Rob_3_ZN = erro_ZN;
erro_Rob_3_ZN

e14 = load ('Rob_3_ACO.mat');
err_ACO(1,:) = e14.e(2,:);
erro_ACO = err_ACO' ;
erro_ACO = abs(erro_ACO);
erro_ACO = sum(erro_ACO);
erro_Rob_3_ACO = erro_ACO;
erro_Rob_3_ACO

```

A.10: Ambiente de simulação (*Simulink*) dos controladores PID para determinação dos índices de erro - processo SISO



A.11: Código-fonte (MATLAB) para determinação dos critérios Margem de Ganho e Margem de Fase - processo SISO

```
% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Cálculo de robustez do sistema (MG e RF)
% *****

Kp = 3.3358;
Ki = 0.0661;
Kd = 21.7854;
num=conv([-5 1], [Kd Kp Ki]);
den=conv([200 30 1],[1 0]);
[mod, fase, w]=bode(num, den);
margin(mod, fase, w);
```

A.12: Código-fonte (MATLAB) do AG para controladores com estrutura em cascata

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística AG aplicada à estrutura em cascata - metodologia simultânea
% *****

clc
clear
num_var = 6;
Funcao_Objetivo = @(x)Funcao_Objetivo(x);
options = gaoptimset;
lb = [0 0 0 0 0 0];
ub = [10 1 10 10 0.1 40];
options = gaoptimset(options,'PopulationType','doubleVector');
options = gaoptimset(options,'PopInitRange',[0; 40]);
options = gaoptimset(options,'PopulationSize',100);
options = gaoptimset(options,'Generations',100);
options = gaoptimset(options,'EliteCount',1);
options = gaoptimset(options,'CrossoverFraction',0.4);
options = gaoptimset(options,'CrossoverFcn',{ @crossoverheuristic });
options = gaoptimset(options,'SelectionFcn',{ @selectiontournament });
options = gaoptimset(options,'MutationFcn',{ @mutationadaptfeasible});
options = gaoptimset(options,'PlotFcns',{ @gaplotbestf @gaplotbestindiv});
options = gaoptimset(options,'Display','iter');

[X,FVAL] = ga(Funcao_Objetivo,num_var,[],[],[],[],lb,ub,[],options);

Kp1=X(1);
Ki1=X(2);
Kd1=X(3);
Kp2=X(4);
Ki2=X(5);
Kd2=X(6);

Kp1
Ki1
Kd1
Kp2
Ki2
Kd2

```

A.13: Código-fonte (MATLAB) do SA para determinação dos parâmetros dos controladores com estrutura em cascata

```

%*****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística SA aplicada à estrutura em cascata - metodologia simultânea
%*****

clc
clear
Funcao_Objetivo = @(x)Funcao_Objetivo(x);
options = saoptimset;
lb = [0 0 0 0 0 0];
ub = [10 1 10 10 0.1 40];
a = 10 * rand(1);
b = 1 * rand(1);
c = 10 * rand(1);
d = 10 * rand(1);
e = 0.1 * rand(1);
f = 40 * rand(1);
x0 = [a b c d e f];
options = saoptimset(options,'InitialTemperature', 100);
options = saoptimset(options,'MaxIter', 3000);
options = saoptimset(options,'AnnealingFcn', @annealingfast);
options = saoptimset(options,'TemperatureFcn',@temperaturefast);
options = saoptimset(options,'PlotFcns', { @splotbestf, @splotbestx });
% options = saoptimset(options,'Display', 'iter');
options = saoptimset(options,'TolFun', 1e-6);

[X,FVAL] = simulannealbnd(Funcao_Objetivo, x0, lb, ub, options);

Kp1=X(1);
Ki1=X(2);
Kd1=X(3);
Kp2=X(4);
Ki2=X(5);
Kd2=X(6);

Kp1
Ki1
Kd1
Kp2
Ki2
Kd2

```

A.14: Código-fonte (MATLAB) da Função objetivo - estrutura em cascata com metodologia simultânea

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo para estrutura em cascata - metodologia simultânea
% *****

function fitness = Funcao_Objetivo(x)

Kp1=x(1);
Ki1=x(2);
Kd1=x(3);
Kp2=x(4);
Ki2=x(5);
Kd2=x(6);
options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_simulink_Lee_par',[0 250], options);

e = load ('Erro_1.mat');
err(1,:) = e.e(2,:);
erro1 = err' ;

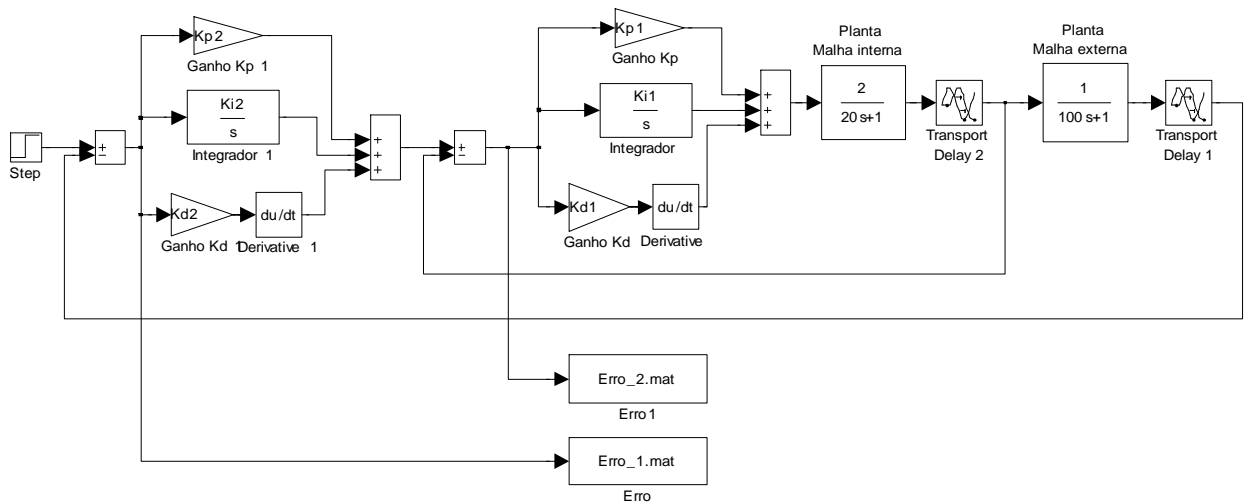
e2 = load ('Erro_2.mat');
err2(1,:) = e2.e(2,:);
erro2 = err2' ;

erro1 = erro1.*tout;
erro1 = abs(erro1);
erro1 = sum(erro1);

erro2 = erro2.*tout;
erro2 = abs(erro2);
erro2 = sum(erro2);
erro_t = erro1 + erro2;
fitness = erro_t;

```

A.15: Ambiente de simulação (*Simulink*) do controlador PID - estrutura em cascata



A.16: Código-fonte (MATLAB) do AG para determinação dos parâmetros dos controladores com estrutura em cascata - metodologia sequencial

```

%*****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística AG aplicada à estrutura em cascata - metodologia sequencial
%*****

clc
clear

num_var = 3;
Funcao_Objetivo_n = @(x)Funcao_Objetivo_P1(x);
options = gaoptimset;
lb = [0 0 0];
ub = [10 1 10];
options = gaoptimset(options,'PopulationType','doubleVector');
options = gaoptimset(options,'PopInitRange',[0; 10]);
options = gaoptimset(options,'PopulationSize',100);
options = gaoptimset(options,'Generations',100);
options = gaoptimset(options,'EliteCount',1);
options = gaoptimset(options,'CrossoverFraction',0.4);
options = gaoptimset(options,'CrossoverFcn',{@crossoverheuristic});
options = gaoptimset(options,'SelectionFcn',{@selectiontournament});
options = gaoptimset(options,'MutationFcn',{@mutationadaptfeasible});
options = gaoptimset(options,'PlotFcns',{@gaplotbestf @gaplotbestindiv});
options = gaoptimset(options,'Display','iter');
[X,FVAL] = ga(Funcao_Objetivo_n,num_var,[],[],[],lb,ub,[],options);
Kp1=X(1);
Ki1=X(2);
Kd1=X(3);

Funcao_Objetivo_p = @(y)Funcao_Objetivo(y, Kp1, Ki1, Kd1);
options = gaoptimset;
lb = [0 0 0];
ub = [10 1 40];
options = gaoptimset(options,'PopulationType','doubleVector');
options = gaoptimset(options,'PopInitRange',[0; 40]);
options = gaoptimset(options,'PopulationSize',100);
options = gaoptimset(options,'Generations',100);
options = gaoptimset(options,'EliteCount',1);
options = gaoptimset(options,'CrossoverFraction',0.4);
options = gaoptimset(options,'CrossoverFcn',{@crossoverheuristic});
options = gaoptimset(options,'SelectionFcn',{@selectiontournament});
options = gaoptimset(options,'MutationFcn',{@mutationadaptfeasible});
options = gaoptimset(options,'PlotFcns',{@gaplotbestf @gaplotbestindiv});
options = gaoptimset(options,'Display','iter');
[Y,FVAL] = ga(Funcao_Objetivo_p,num_var,[],[],[],lb,ub,[],options);

Kp2=Y(1);
Ki2=Y(2);
Kd2=Y(3);

Kp1
Ki1
Kd1
Kp2
Ki2
Kd2

```

A.17: Código-fonte (MATLAB) do SA para determinação dos parâmetros dos controladores com estrutura em cascata - metodologia sequencial

```

%*****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística SA aplicada à estrutura em cascata - metodologia sequencial
%*****

clc
clear
num_var = 3;
Funcao_Objetivo_n = @(x)Funcao_Objetivo_P1(x);

options = saoptimset;
lb = [0 0 0];
ub = [10 1 10];
a = 10 * rand(1);
b = 1 * rand(1);
c = 10 * rand(1);
x0 = [a b c];
options = saoptimset(options,'InitialTemperature', 100);
options = saoptimset(options,'MaxIter', 3000);
options = saoptimset(options,'AnnealingFcn', @annealingfast);
options = saoptimset(options,'TemperatureFcn',@temperatureboltz);
options = saoptimset(options,'PlotFcns', { @saplotbestf, @saplotbestx });
options = saoptimset(options,'TolFun', 1e-6);

[X,FVAL] = simulannealbnd(Funcao_Objetivo_n, x0, lb, ub, options);

Kp1=X(1);
Ki1=X(2);
Kd1=X(3);

Funcao_Objetivo_p = @(y)Funcao_Objetivo(a, y, Kp1, Ki1, Kd1);
options = saoptimset;
lb = [0 0 0];
ub = [10 0.1 40];
d = 10 * rand(1);
e = 0.1 * rand(1);
f = 40 * rand(1);
x0 = [d e f];
options = saoptimset(options,'InitialTemperature', 100);
options = saoptimset(options,'MaxIter', 3000);
options = saoptimset(options,'AnnealingFcn', @annealingfast);
options = saoptimset(options,'TemperatureFcn',@temperatureboltz);
options = saoptimset(options,'PlotFcns', { @saplotbestf, @saplotbestx });
options = saoptimset(options,'TolFun', 1e-6);

[Y,FVAL] = simulannealbnd(Funcao_Objetivo_p, x0, lb, ub, options);

Kp2=Y(1);
Ki2=Y(2);
Kd2=Y(3);

Kp1
Ki1
Kd1
Kp2
Ki2
Kd2

```


A.18: Código-fonte (MATLAB) da Função objetivo da malha interna - estrutura em cascata com metodologia sequencial

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo para estrutura em cascata (malha interna) - metodologia sequencial
% *****

function fitness = Funcao_Objetivo_P1(x)
Kp1=x(1);
Ki1=x(2);
Kd1=x(3);

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_simulink_Lee_P1',[0 100], options);
e2 = load ('Erro_2.mat');
err2(1,:) = e2.e(2,:);
erro2 = err2' ;
erro2 = erro2.*tout;
erro2 = abs(erro2);
erro2 = sum(erro2);
fitness = erro2;

```

A.19: Código-fonte (MATLAB) da Função objetivo da malha externa - estrutura em cascata com metodologia sequencial

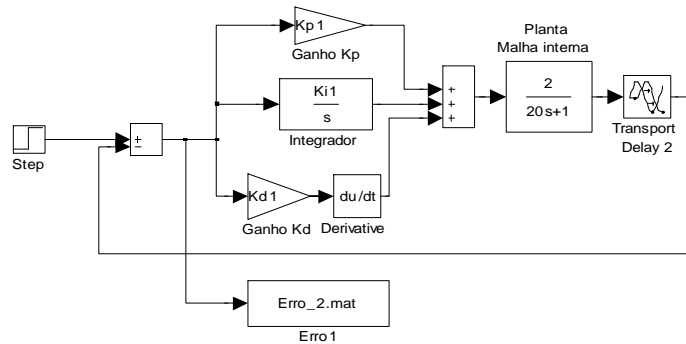
```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo para estrutura em cascata (malha externa) - metodologia sequencial
% *****

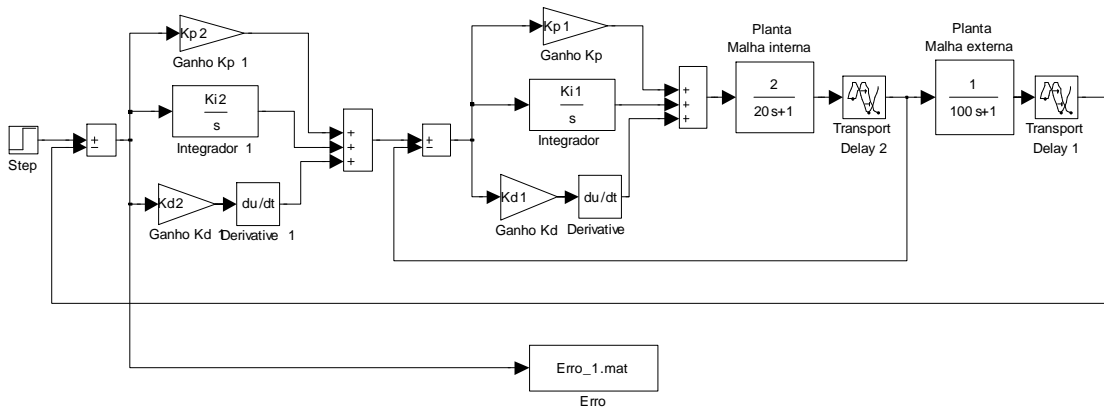
function fitness = Funcao_Objetivo(a, y, Kp1, Ki1, Kd1)
Kp1 = Kp1;
Ki1 = Ki1;
Kd1 = Kd1;
Kp2=y(1);
Ki2=y(2);
Kd2=y(3);
options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_simulink_Lee_par',[0 250], options);
e = load ('Erro_1.mat');
err(1,:) = e.e(2,:);
error = err' ;
error = error.*tout;
error = abs(error);
error = sum(error);
erro_t = error + a;
fitness = erro_t;

```

A.20: Ambiente de simulação (*Simulink*) do controlador PID malha interna - estrutura em cascata com metodologia sequencial



A.21: Ambiente de simulação (*Simulink*) do controlador PID malha externa- estrutura em cascata com metodologia sequencial



A.22: Código-fonte (MATLAB) para determinação dos critérios de desempenho no domínio do tempo - estrutura em cascata

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Desempenho no domínio do tempo - estrutura em cascata
% *****

AG_Sim=stepinfo(CASC_Sim_Seq.signals.values(1:757,1),CASC_Sim_Seq.time(1:757,1),1);
SA_Sim=stepinfo(CASC_Sim_Seq.signals.values(1:757,2),CASC_Sim_Seq.time(1:757,1),1);
AG_Seq=stepinfo(CASC_Sim_Seq.signals.values(1:757,3),CASC_Sim_Seq.time(1:757,1),1);
SA_Seq=stepinfo(CASC_Sim_Seq.signals.values(1:757,4),CASC_Sim_Seq.time(1:757,1),1);
    
```

A.23: Código-fonte (MATLAB) do AG Multiobjetivo - processo MIMO 2 x 2 com controladores PID descentralizados

```

%*****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística AG Multiobjetivo aplicada ao controle PID descentralizado
%*****

clc
clear
nvars = 6;
fitnessFunction = @(x)Funcao_Objetivo(x);

options = gaoptimset(options,'PopInitRange',[-0.2; 0.2]);
options = gaoptimset(options,'PopulationSize',100);
options = gaoptimset(options,'PopulationType','doubleVector');
options = gaoptimset(options,'EliteCount',1);
options = gaoptimset(options,'Generations',100);
options = gaoptimset(options,'CrossoverFraction',0.4);
options = gaoptimset(options,'ParetoFraction',0.3);
options = gaoptimset(options,'CrossoverFcn',{ @crossoverheuristic });
options = gaoptimset(options,'SelectionFcn',{ @selectiontournament });
options = gaoptimset(options,'MutationFcn',{ @mutationadaptfeasible });
options = gaoptimset(options,'PlotFcns',{ @gaplotpareto,@gaplotscorediversity});
options = gaoptimset(options,'Display','iter');

[X,FVAL] = gamultiobj(fitnessFunction,nvars,[],[],[],[],lb,ub,options);

FVAL

X

```

A.24: Código-fonte (MATLAB) da Função objetivo - processo MIMO 2 x 2 com controladores PID descentralizados

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo - controle PID descentralizado
% *****

function f = Funcao_Objetivo(x)
f=[];

xSize = size(x,1);
numObj = 3;
f = zeros(xSize, numObj);

Kp1=x(1);
Ki1=x(2);
Kd1=x(3);
Kp2=x(4);
Ki2=x(5);
Kd2=x(6);

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_MIMO_Pareto',[0 300], options);
e = load ('Erro_1.mat');
err(1,:) = e.e(2,:);
erro1 = err';

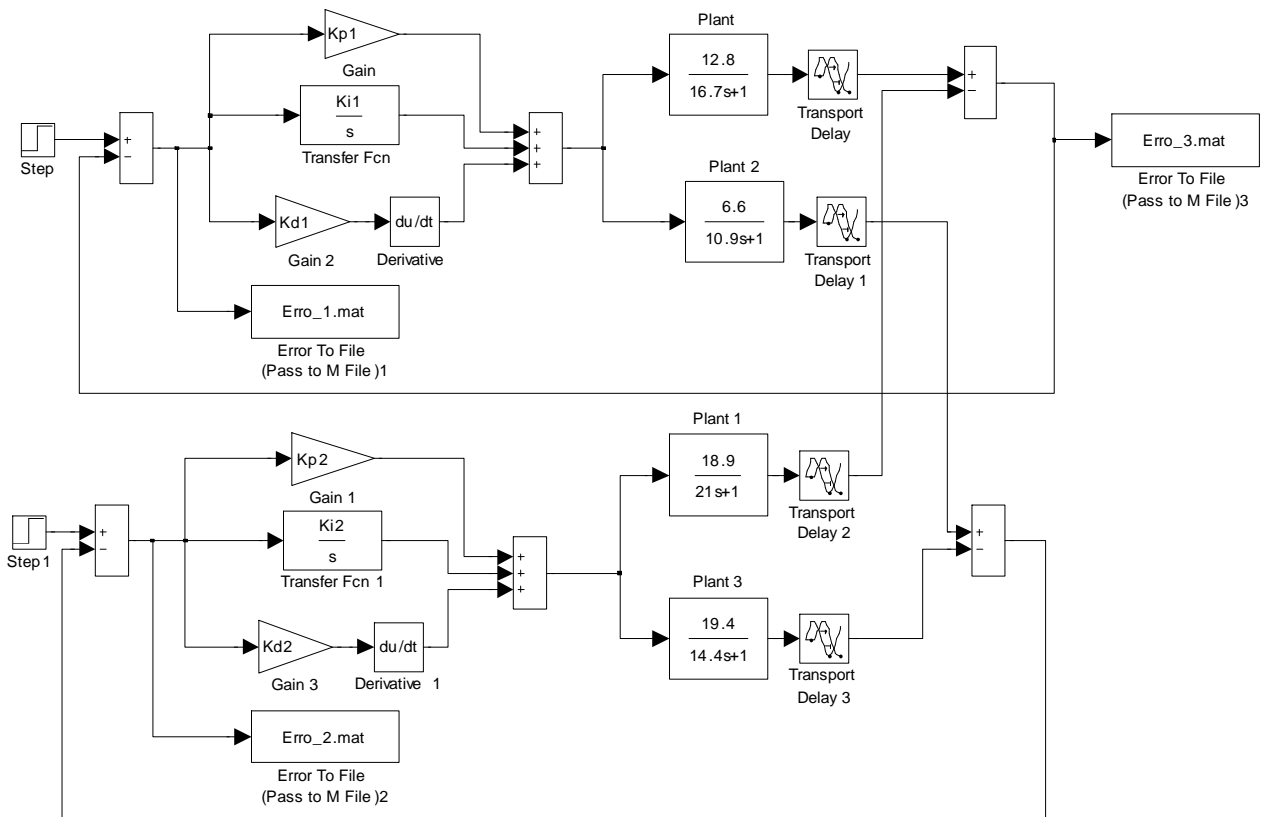
erro1 = erro1.^2;
erro1 = erro1.*tout;
erro1 = abs(erro1);
erro1 = sum(erro1);

r = stepinfo(Pareto_MIMO.signals.values(1:200,1),Pareto_MIMO.time(1:200,1),0.96);
o1 = r.Overshoot;
erro2 = o1;

s = stepinfo(Pareto_MIMO.signals.values(200:600,2),Pareto_MIMO.time(200:600,1),0.05);
o2 = s.Overshoot;
erro3 = o2;

f(:,1) = erro1;
f(:,2) = erro2;
f(:,3) = erro3;

```

A.25: Ambiente de simulação (*Simulink*) do controlador PID MIMO descentralizados

A.26: Código-fonte (MATLAB) do AG Multiobjetivo para controle de processo MIMO 2 x 2 com controladores PID com desacopladores

```

%*****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Metaheurística AG Multiobjetivo aplicada controle PID com desacopladores
%*****

clear

nvars = 8;
fitnessFunction = @(x)Funcao_Objetivo(x);

options = gaoptimset;
lb = [-0.2 -0.2 -0.2 -0.2 -0.2 -0.2 -0.2 -0.2];
ub = [0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2];

options = gaoptimset(options,'PopInitRange' ,[-0.2; 0.2]);
options = gaoptimset(options,'PopulationSize' ,100);
options = gaoptimset(options,'PopulationType' , 'doubleVector');
options = gaoptimset(options,'EliteCount' ,1);
options = gaoptimset(options,'Generations' ,100);
options = gaoptimset(options,'CrossoverFraction' ,0.4);
options = gaoptimset(options,'ParetoFraction' ,0.3);
options = gaoptimset(options,'CrossoverFcn' ,{ @crossoverheuristic });
options = gaoptimset(options,'SelectionFcn' ,{ @selectiontournament });
options = gaoptimset(options,'MutationFcn' ,{ @mutationadaptfeasible });
options = gaoptimset(options,'PlotFcns' , { @gaplotpareto,@gaplotscorediversity});
options = gaoptimset(options,'Display' , 'iter');

[X,FVAL] = gamultiobj(fitnessFunction,nvars,[],[],[],[],lb,ub,options);

FVAL
X

```

A.27: Código-fonte (MATLAB) da Função objetivo para controle de processo MIMO 2 x 2 com controladores PID com desacopladores

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Função objetivo - controle PID com desacopladores
% *****

function f = Funcao_Objetivo(x)
f=[];

xSize = size(x,1); % Population size
numObj = 3; % Number of objectives
f = zeros(xSize, numObj);

Kp1=x(1);
Ki1=x(2);
Kd1=x(3);
Kp2=x(4);
Ki2=x(5);
Kd2=x(6);
K1=x(7);
K2=x(8);

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('PID_MIMO_Pareto',[0 300], options);

e = load ('Erro_1.mat');
err(1,:) = e.e(2,:);
erro1 = err' ;

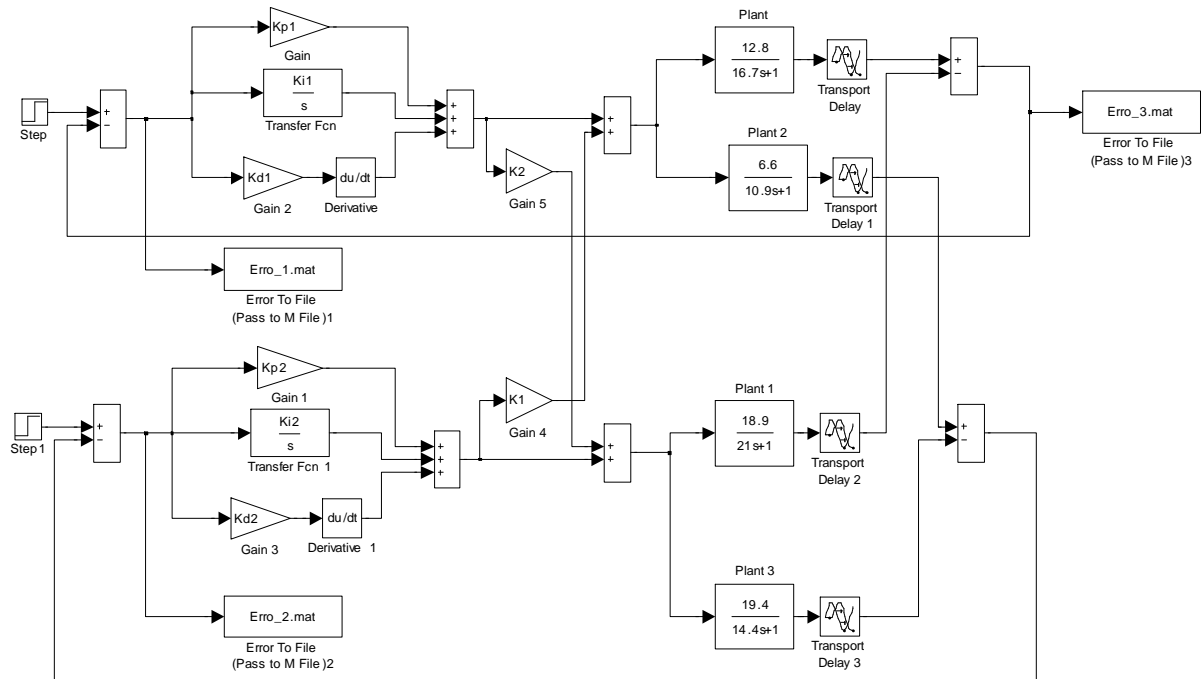
erro1 = erro1.^2;
erro1 = erro1.*tout;
erro1 = abs(erro1);
erro1 = sum(erro1);

r = stepinfo(Pareto_MIMO.signals.values(1:200,1),Pareto_MIMO.time(1:200,1),0.96);
o1 = r.Overshoot;
erro2 = o1;

s = stepinfo(Pareto_MIMO.signals.values(200:600,2),Pareto_MIMO.time(200:600,1),0.05);
o2 = s.Overshoot;
erro3 = o2;

f(:,1) = erro1;
f(:,2) = erro2;
f(:,3) = erro3;

```

A.28: Ambiente de simulação (*Simulink*) do controlador PID MIMO com desacopladores

A.29: Código-fonte (MATLAB) para determinação dos critérios de desempenho no domínio do tempo - processo MIMO

```

% *****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Desempenho - processo MIMO
% *****

clear
clc

AG=stepinfo(MIMO_Desc_Desac.signals.values(1:600,1),MIMO_Desc_Desac.time(1:600,1),0.96);
AG

```


A.30: Código-fonte (MATLAB) do teste de robustez PID descentralizado e com desacopladores sintonizados por AG multiobjetivo e método BLT

```

%*****
% Universidade do Vale do Rio dos Sinos (UNISINOS)
% Mestrado em Computação Aplicada
% Metaheurísticas aplicadas na sintonia de controladores PID: Estudo de casos
% Teste de robustez - processo MIMO
%*****

clear
clc

media_over_1 = 0;
media_over_2 = 0;
media_over_3 = 0;
media_over_4 = 0;

media_acom_1 = 0;
media_acom_2 = 0;
media_acom_3 = 0;
media_acom_4 = 0;

media_acom_1_Y2 = 0;
media_acom_2_Y2 = 0;
media_acom_3_Y2 = 0;
media_acom_4_Y2 = 0;

maior_acom_1 = 0;
maior_acom_2 = 0;
maior_acom_3 = 0;
maior_acom_4 = 0;

menor_acom_1 = 1000;
menor_acom_2 = 1000;
menor_acom_3 = 1000;
menor_acom_4 = 1000;

maior_over_1 = 0;
maior_over_2 = 0;
maior_over_3 = 0;
maior_over_4 = 0;

menor_over_1 = 100;
menor_over_2 = 100;
menor_over_3 = 100;
menor_over_4 = 100;

maior_acom_1_Y2 = 0;
menor_acom_1_Y2 = 1000;
maior_over_1_Y2 = 0;
menor_over_1_Y2 = 1000;

maior_acom_2_Y2 = 0;
menor_acom_2_Y2 = 1000;
maior_over_2_Y2 = 0;
menor_over_2_Y2 = 1000;

maior_acom_3_Y2 = 0;
menor_acom_3_Y2 = 1000;
maior_over_3_Y2 = 0;
menor_over_3_Y2 = 1000;

```

```

maior_acom_4_Y2 = 0;
menor_acom_4_Y2 = 1000;
maior_over_4_Y2 = 0;
menor_over_4_Y2 = 1000;

nsamples = 100;
valor = 2000;

for i=1:nsamples
K1 = 0.9 + 0.2*rand(1);
K2 = 6.3 + 1.4*rand(1);
K3 = 2.7 + 0.6*rand(1);
K4 = 2.7 + 0.6*rand(1);
K5 = 0.9 + 0.2*rand(1);
K6 = 0.9 + 0.2*rand(1);
K7 = 0.9 + 0.2*rand(1);
K8 = 0.9 + 0.2*rand(1);
K9 = 15 + 3.3*rand(1);
K10 = 13 + 3*rand(1);
K11 = 10 + 2*rand(1);
K12 = 19 + 4*rand(1);

Kp1 = 0.2;
Ki1 = 0.0251;
Kd1 = -0.0063;
Kp2 = -0.0972;
Ki2 = -0.0047;
Kd2 = -0.1165;
D1 = 0;
D2 = 0;

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('Wood_Berry',[0 1000], options);

figure(1)
subplot(2,2,1)
xlabel('Overshoot (%'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC.signals.values(1:valor,1),MIMO_MC.time(1:valor,1),0.96);
a_1 = s.SettlingTime;
b_1 = s.Overshoot;
media_over_1 = media_over_1 + b_1;
media_acom_1 = media_acom_1 + a_1;
plot(b_1, a_1,'r*'); grid on
hold on;
if (a_1 > maior_acom_1) maior_acom_1 = a_1;
end
if (a_1 < menor_acom_1) menor_acom_1 = a_1;
end
if (b_1 > maior_over_1) maior_over_1 = b_1;
end
if (b_1 < menor_over_1) menor_over_1 = b_1;
end

figure(1)
subplot(2,2,2)
xlabel('Overshoot (%'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC_Y2.signals.values(1:valor,1),MIMO_MC_Y2.time(1:valor,1),0.05);
a_2 = s.SettlingTime;
b_2 = s.Overshoot;

```

```

media_acom_1_Y2 = media_acom_1_Y2 + a_2;
plot(b_2, a_2, 'r*'); grid on
hold on;
if (a_2 > maior_acom_1_Y2) maior_acom_1_Y2 = a_2;
end
if (a_2 < menor_acom_1_Y2) menor_acom_1_Y2 = a_2;
end

Kp1=0.1964;
Ki1=0.0261;
Kd1=-0.0396;
Kp2=-0.1140;
Ki2=-0.0053;
Kd2=-0.1780;
D1=0.0442;
D2=0.0477;

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('Wood_Berry',[0 1000], options);

figure(1)
subplot(2,2,3)
xlabel('Overshoot (%)'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC.signals.values(1:valor,1),MIMO_MC.time(1:valor,1),0.96);
a_3 = s.SettlingTime;
b_3 = s.Overshoot;
media_over_2 = media_over_2 + b_3;
media_acom_2 = media_acom_2 + a_3;
plot(b_3, a_3, 'r*'); grid on
hold on;
if (a_3 > maior_acom_2) maior_acom_2 = a_3;
end
if (a_3 < menor_acom_2) menor_acom_2 = a_3;
end
if (b_3 > maior_over_2) maior_over_2 = b_3;
end
if (b_3 < menor_over_2) menor_over_2 = b_3;
end

figure(1)
subplot(2,2,4)
xlabel('Overshoot (%)'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC_Y2.signals.values(1:valor,1),MIMO_MC_Y2.time(1:valor,1),0.05);
a_4 = s.SettlingTime;
b_4 = s.Overshoot;
media_acom_2_Y2 = media_acom_2_Y2 + a_4;
plot(b_4, a_4, 'r*'); grid on
hold on;
if (a_4 > maior_acom_2_Y2) maior_acom_2_Y2 = a_4;
end
if (a_4 < menor_acom_2_Y2) menor_acom_2_Y2 = a_4;
end

Kp1 = 0.3750;
Ki1 = 0.0452;
Kd1 = 0;
Kp2 = -0.075;
Ki2 = -0.0032;
Kd2 = 0;

```

```

D1 = 0;
D2 = 0;

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('Wood_Berry',[0 1000], options);

figure(2)
subplot(2,2,1)
xlabel ('Overshoot (%)'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC.signals.values(1:valor,1),MIMO_MC.time(1:valor,1),0.96);
a_5 = s.SettlingTime;
b_5 = s.Overshoot;
media_over_3 = media_over_3 + b_5;
media_acom_3 = media_acom_3 + a_5;
plot(b_5, a_5,'r*'); grid on
hold on;
if (a_5 > maior_acom_3) maior_acom_3 = a_5;
end
if (a_5 < menor_acom_3) menor_acom_3 = a_5;
end
if (b_5 > maior_over_3) maior_over_3 = b_5;
end
if (b_5 < menor_over_3) menor_over_3 = b_5;
end

figure(2)
subplot(2,2,2)
xlabel ('Overshoot (%)'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC_Y2.signals.values(1:valor,1),MIMO_MC_Y2.time(1:valor,1), 0.05);
a_6 = s.SettlingTime;
b_6 = s.Overshoot;
media_acom_3_Y2 = media_acom_3_Y2 + a_6;
plot(b_6, a_6,'r*'); grid on
hold on;
if (a_6 > maior_acom_3_Y2) maior_acom_3_Y2 = a_6;
end
if (a_6 < menor_acom_3_Y2) menor_acom_3_Y2 = a_6;
end

Kp1 = 0.11238;
Ki1 = 0.02431;
Kd1 = 0.00639;
Kp2 = -0.12129;
Ki2 = -0.001580;
Kd2 = -0.13549;
D1 = 0;
D2 = 0;

options = simset('SrcWorkspace','current');
[tout,xout,yout] = sim('Wood_Berry',[0 1000], options);

figure(2)
subplot(2,2,3)
xlabel ('Overshoot (%)'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC.signals.values(1:valor,1),MIMO_MC.time(1:valor,1),0.96);
a_7 = s.SettlingTime;
b_7 = s.Overshoot;
media_over_4 = media_over_4 + b_7;
media_acom_4 = media_acom_4 + a_7;

```

```

plot(b_7, a_7,'r*'); grid on
hold on;
if (a_7 > maior_acom_4) maior_acom_4 = a_7;
end
if (a_7 < menor_acom_4) menor_acom_4 = a_7;
end
if (b_7 > maior_over_4) maior_over_4 = b_7;
end
if (b_7 < menor_over_4) menor_over_4 = b_7;
end

figure(2)
subplot(2,2,4)
xlabel('Overshoot (%)'); ylabel('Tempo de acomodação (s)');
s = stepinfo(MIMO_MC_Y2.signals.values(1:valor,1),MIMO_MC_Y2.time(1:valor,1), 0.05);
a_8 = s.SettlingTime;
b_8 = s.Overshoot;
media_acom_4_Y2 = media_acom_4_Y2 + a_8;
plot(b_8, a_8,'r*'); grid on
hold on;
if (a_8 > maior_acom_4_Y2) maior_acom_4_Y2 = a_8;
end
if (a_8 < menor_acom_4_Y2) menor_acom_4_Y2 = a_8;
end
end

maior_acom_1
maior_acom_2
maior_acom_3
maior_acom_4

menor_acom_1
menor_acom_2
menor_acom_3
menor_acom_4

maior_over_1
maior_over_2
maior_over_3
maior_over_4

menor_over_1
menor_over_2
menor_over_3
menor_over_4

maior_acom_1_Y2
maior_acom_2_Y2
maior_acom_3_Y2
maior_acom_4_Y2

menor_acom_1_Y2
menor_acom_2_Y2
menor_acom_3_Y2
menor_acom_4_Y2

media_over_1 = media_over_1/100;
media_over_2 = media_over_2/100;
media_over_3 = media_over_3/100;
media_over_4 = media_over_4/100;

```

media_acom_1 = media_acom_1/100;
media_acom_2 = media_acom_2/100;
media_acom_3 = media_acom_3/100;
media_acom_4 = media_acom_4/100;

media_acom_1_Y2 = media_acom_1_Y2/100;
media_acom_2_Y2 = media_acom_2_Y2/100;
media_acom_3_Y2 = media_acom_3_Y2/100;
media_acom_4_Y2 = media_acom_4_Y2/100;

media_over_1
media_over_2
media_over_3
media_over_4

media_acom_1
media_acom_2
media_acom_3
media_acom_4

media_acom_1_Y2
media_acom_2_Y2
media_acom_3_Y2
media_acom_4_Y2