

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**METAHEURÍSTICAS POPULACIONAIS: ESTUDO COMPARATIVO NA
SINTONIA DE PARÂMETROS DE CONTROLADORES CLÁSSICOS**

JUAN FERREIRA VIDAL

DM 28/2016

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil

2016

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JUAN FERREIRA VIDAL

**METAHEURÍSTICAS POPULACIONAIS: ESTUDO COMPARATIVO NA
SINTONIA DE PARÂMETROS DE CONTROLADORES CLÁSSICOS**

DM 28 /2016

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil
2016

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JUAN FERREIRA VIDAL

**METAHEURÍSTICAS POPULACIONAIS: ESTUDO COMPARATIVO NA
SINTONIA DE PARÂMETROS DE CONTROLADORES CLÁSSICOS**

Dissertação submetida à Banca Examinadora do Programa de Pós- Graduação em Engenharia Elétrica da UFPA para a obtenção do Grau de Mestre em Engenharia Elétrica na Área de Computação Aplicada.

UFPA/ITEC/PPGEE

Campus Universitário do Guamá

Belém – Pará – Brasil

2016

Dados Internacionais de Catalogação - na - Publicação (CIP) Sistema de
Bibliotecas da UFPA

Vidal, Juan Ferreira, 1990-

Metaheurísticas populacionais : estudo comparativo na sintonia de
parâmetros de controladores clássicos / Juan Ferreira Vidal.- 2016.

Orientadora : Adriana Rosa Garcez Castro

Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de
Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém,
2016.

1. Inteligência artificial. 2. Controladores programáveis. 3. Algoritmos
genéticos. I. Título.

CDD 23. ed. 006.3

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**METAHEURÍSTICAS POPULACIONAIS: ESTUDO COMPARATIVO NA
SINTONIA DE PARÂMETROS DE CONTROLADORES CLÁSSICOS**

Este trabalho foi julgado em ___/___/___ de 2016, adequado para obtenção do Grau de Mestre em Engenharia Elétrica, e aprovado na sua forma final pela banca examinadora que atribuiu o conceito _____.

Prof^a. Dr^a. Adriana Rosa Garcez Castro
ORIENTADORA
MEMBRO DA BANCA
EXAMINADORA

Prof. Dr. Carlos Tavares da Costa Junior
MEMBRO DA BANCA
EXAMINADORA

Prof. Dr. Orlando Fonseca Silva
MEMBRO DA BANCA
EXAMINADORA

DEDICATÓRIA

Dedico este trabalho a minha Mãe Elen Vidal que é uma das minhas maiores incentivadoras nesta vida. E foi a pessoa que me deu todas as condições necessárias para que eu pudesse me dedicar da melhor maneira possível para a conclusão deste Mestrado.

Mãe muito obrigado, esta vitória é em grande parte devido a Senhora!

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus por iluminar meu caminho e por sempre estar presente em minha vida me dando força, coragem, perseverança e por ter permitido a conclusão de mais esta etapa.

Agradeço a toda minha família em especial a minha mãe Elen Vidal por sempre me apoiar nesta minha caminhada e por ser meu exemplo e minha base nesta vida, e por quem tenho todo respeito, admiração, carinho e amor. Ao meu avô Antônio Ferreira e a minha avó Maria Elizabeth, por quem tenho um amor incondicional, por todo o amor e carinho que sempre tiveram comigo. Ao meu irmão Alan Vidal e a minha irmã Deice Vidal por sempre estarem me apoiando e incentivando no decorrer da construção deste trabalho.

Agradeço a minha orientadora, professora Dr^a Adriana Castro, por todo o conhecimento passado, pelas revisões feitas que foram essenciais para o desenvolvimento deste trabalho. Agradeço pela oportunidade e confiança dadas a mim para o desenvolvimento desta dissertação.

Agradeço a todos os meus amigos do Programa de Pós Graduação em Engenharia Elétrica (PPGEE) pela amizade e ajuda em muitos momentos. Em especial as minhas amigas Evelin Cardoso, Aline Ohashi e aos amigos Alan Marciel e André Carvalho obrigado pela companhia e por todo o apoio nesta caminhada.

Agradeço a todos os meus amigos em especial a Nadjala Barbosa, Thaisse Dias Wendson Campos e Alana Lima pela amizade, apoio e por entenderem minha ausência em muitos momentos por conta da construção deste trabalho.

Agradeço as minhas amigas do Curso de Engenharia da Computação Elô, Juliana e Ivone por todos os momentos de distração que fizeram estes períodos ser mais fáceis de serem percorridos.

Agradeço a todos que de maneira direta ou indireta me ajudaram a conseguir concluir esta dissertação e contribuíram para que eu pudesse concluir mais esta etapa. Mesmo não podendo citar o nome de todos, quero deixar meu mais sincero obrigado.

Agradeço ao PPGEE pela oportunidade de aprendizagem e crescimento que foram dadas a mim. Agradeço ao Cnpq pelo apoio financeiro, que possibilitou que eu pudesse me dedicar para realizar mais este objetivo.

“Não é o mais forte que sobrevive, nem o mais inteligente. Quem sobrevive é o mais disposto à mudança.”

(Charles Darwin)

SUMÁRIO

SUMÁRIO	IX
LISTA DE ILUSTRAÇÕES	XI
LISTA DE TABELAS	XIII
RESUMO	XIV
ABSTRACT	XV
1 INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.2 ESTRUTURA DO TRABALHO	3
2 METAHEURÍSTICAS POPULACIONAIS	4
2.1 INTRODUÇÃO	4
2.2 ALGORITMO GENÉTICO (AG)	6
2.2.1 <i>Histórico dos Algoritmos Genéticos</i>	6
2.2.2 <i>Características dos Algoritmos Genéticos</i>	8
2.2.3 <i>Seleção dos Pais</i>	11
2.2.4 <i>Operador de Crossover</i>	14
2.2.5 <i>Mutação</i>	17
2.2.6 <i>Elitismo</i>	18
2.3 PARTICLE SWARM OPTIMIZATION (PSO)	18
2.3.1 <i>Inteligência de Enxame</i>	18
2.3.2 <i>Características do PSO padrão</i>	20
2.3.3 <i>Considerações sobre o PSO</i>	22
2.4 BACTERIAL FORAGING OPTIMIZATION (BFO)	23
2.4.1 <i>Comportamento de forrageio bacteriano</i>	23
2.4.2 <i>Características do algoritmo BFO</i>	24
2.5 ESTADO DA ARTE DE METAHEURÍSTICAS POPULACIONAIS APLICADAS À SISTEMAS DE CONTROLE	26
3 CONTROLADORES PID	30
3.1 INTRODUÇÃO	30
3.2 AÇÕES DE CONTROLE PID	32
3.3 SINTONIA DE CONTROLADORES PID	34
3.3.1 <i>Métodos Tradicionais para ajuste dos Parâmetros do Controlador PID</i>	36
3.3.2 <i>Índices de Desempenhos</i>	38
4 SINTONIA DOS PARÂMETROS DE CONTROLADORES PID USANDO METAHEURÍSTICA POPULACIONAL	40
4.1 INTRODUÇÃO	40
4.2 METAHEURÍSTICA POPULACIONAL PARA A SINTONIA DOS PARÂMETROS DE UM CONTROLADOR PID	41
4.2.1 <i>Representação dos indivíduos da população</i>	41
4.2.2 <i>Estrutura de avaliação dos indivíduos da população</i>	42
4.3 AG NO MODELO DE ILHAS (AGMI) PARA A SINTONIA DO PID	43
4.4 APLICAÇÃO DE METAHEURÍSTICA POPULACIONAL PARA A SINTONIA DOS PARÂMETROS DO PID	44
4.4.1 <i>Considerações Iniciais</i>	44
4.4.2 <i>Sistemas de primeira ordem</i>	45
4.4.3 <i>Sistema de Segunda Ordem</i>	60
4.4.4 <i>Sistema de Ordem superior</i>	67

4.4.5	<i>Considerações sobre resultados alcançados.....</i>	70
5	CONCLUSÃO	72
5.1	TRABALHOS FUTUROS	72
6	REFERÊNCIAS BIBLIOGRÁFICAS	74

LISTA DE ILUSTRAÇÕES

Figura 2.1 -Fluxograma de um AG básico	11
Figura 2.2 -Roleta para os indivíduos da Tabela 2.1(LIDEN, 2006)	13
Figura 2.3 -Exemplo de um cromossomo binário e seus pontos de corte(LIDEN, 2006).	14
Figura 2.4 -Crossover de um ponto(LIDEN, 2006).	15
Figura 2.5 -Crossover de dois pontos(LIDEN, 2006).	15
Figura 2.6 -Crossover Uniforme(LIDEN, 2006).	16
Figura 2.7 -Ação do operador de mutação	18
Figura 2.8 -Fluxograma do PSO padrão.....	20
Figura 2.9 -Pseudo código do BFO (TEODORO et al, 2010).....	26
Figura 3.1 -Diagrama de blocos de sistema de controle em malha fechada (OGATA, 2003).....	35
Figura 4.1 -Estrutura usada para a sintonia do PID com metaheurística.....	42
Figura 4.2 -Resposta do sistema do caso 1 para uma entrada degrau unitário.....	46
Figura 4.3 -Resposta do sistema com o uso dos controladores sintonizados para o caso 1	48
Figura 4.4 -Sinais de controle gerados com o uso dos controladores para o caso 1	49
Figura 4.5 -Sinal de saída do sistema	50
Figura 4.6 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AG para o caso 2	51
Figura 4.7 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AGMI para o caso 2	52
Figura 4.8 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo BFO para o caso 2	52
Figura 4.9 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo PSO para o caso 2.....	53
Figura 4.10 -Sinais de controle gerados com o uso dos controladores para o caso 2	55
Figura 4.11 -Resposta do sistema do caso 3 para uma entrada do tipo degrau unitário. 56	
Figura 4.12 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AG para o caso 3.....	57
Figura 4.13 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AGMI para o caso 3.....	58
Figura 4.14 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo BFO para o caso 3.....	58
Figura 4.15 -Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo PSO para o caso 3	59
Figura 4.16 -Sinais de controle gerados com o uso dos controladores para o caso 3	60
Figura 4.17 -Resposta do sistema do caso 4 para uma entrada do tipo degrau unitário. 61	
Figura 4.18 -Comparação da resposta limitante de projeto e a resposta com o uso dos controladores sintonizados pelos algoritmos para o caso 4.....	62

Figura 4.19 -Sinais de controle gerados com o uso dos controladores para o caso 4	63
Figura 4.20 -Resposta do sistema do caso 5 para uma entrada do tipo degrau unitário.	64
Figura 4.21 -Comparação da resposta limitante de projeto e a resposta com o uso dos controladores sintonizados pelos algoritmos para o caso 5	66
Figura 4.22 -Sinais de controle gerados com o uso dos controladores para o caso 5	67
Figura 4.23 -Comparação da resposta limitante de projeto e a resposta com o uso dos controladores sintonizados pelos algoritmos para o caso 6	69
Figura 4.24 -Sinais de controle gerados com o uso dos controladores para o caso 6	70

LISTA DE TABELAS

Tabela 2.1 -Exemplo da separação da roleta em uma população(LIDEN, 2006)	12
Tabela 4.1 -Parâmetros do AG	45
Tabela 4.2 -Parâmetros do BFO	45
Tabela 4.3 -Parâmetros do AGMI	45
Tabela 4.4 -Parâmetros do PSO.....	45
Tabela 4.5 -Especificações de projeto e valores obtidos pelos algoritmos para o caso 147	
Tabela 4.6 -Parâmetros do PID sintonizado por cada algoritmo para o caso 1	47
Tabela 4.7 -Valores dos índices de desempenho para caso 1	48
Tabela 4.8 -Especificação de projeto e valores alcançados para o caso 2.....	50
Tabela 4.9 -Parâmetros sintonizados pelos algoritmos para o caso 2.....	53
Tabela 4.10 -Valores dos índices de desempenho para o caso 2.....	54
Tabela 4.11 -Especificação de projeto e valores alcançados para o caso 3.....	56
Tabela 4.12 -Valores dos parâmetros sintonizados pelos algoritmos para o caso 3.....	57
Tabela 4.13 -Valores dos índices de desempenho para caso 3	59
Tabela 4.14 -Especificação de projeto e valores alcançados para o caso 4.....	61
Tabela 4.15 -Valores dos parâmetros sintonizados pelos algoritmos para o caso 4.....	62
Tabela 4.16 -Valores dos índices de desempenho para caso 4.....	63
Tabela 4.17 -Especificação de projeto e valores alcançados para o caso 5	65
Tabela 4.18 -Valores dos parâmetros sintonizados pelos algoritmos para o caso 5.....	65
Tabela 4.19 -Valores dos índices de desempenho para o caso 5.....	66
Tabela 4.20 -Especificação de projeto e valores alcançados para o caso 6.....	68
Tabela 4.21 -Valores dos parâmetros sintonizados pelos algoritmos para o caso 6.....	68
Tabela 4.22 -Valores dos índices de desempenho para o caso 6.....	69

RESUMO

As metaheurísticas populacionais são técnicas pertencentes ao campo da Inteligência Computacional baseadas em modelos naturais e surgiram como alternativas para resolver problemas de otimização, onde as técnicas tradicionais não podem ser aplicadas, ou ainda onde não se dispõe de um modelo de solução para o problema, fazendo com que a solução seja encontrada por intermédio de meios empíricos. Diante da capacidade de oferecer soluções aceitáveis, em um tempo hábil, para muitos dos problemas complexos encontrados, as metaheurísticas populacionais vêm sendo aplicadas com êxito diferentes problemas de sistemas de controle encontrados na literatura. Este trabalho apresenta, de um modo geral, como as metaheurísticas vêm sendo aplicadas na solução de problemas de controle e realiza um estudo comparativo de desempenho entre quatro algoritmos bioinspirados na sintonia dos parâmetros de um controlador Proporcional-Integral-Derivativo (PID). Foram utilizados os seguintes algoritmos: Algoritmo Genético (AG), Algoritmo genético no Modelo de Ilhas (AGMI), *Bacterial Foraging Optimization* (BFO) e o *Particle Swarm Optimization* (PSO). Os resultados demonstram que os algoritmos apresentam um ótimo desempenho para a sintonia do PID, produzindo resposta que atendem as exigências de projetos. Foram utilizados diferentes sistemas com características distintas para avaliar os algoritmos. Considerando os resultados obtidos, o PSO se mostrou como o melhor algoritmo entre os quatro usados, produzindo resposta em um tempo mais rápido e apresentando menor desvio padrão nos ensaios realizados.

PALAVRAS-CHAVES: Algoritmos Genéticos, Controlador PID, *Particle Swarm Optimization*, *Bacterial Foraging Optimization*.

ABSTRACT

Population metaheuristics are techniques belonging to the field of Computational Intelligence and are based on natural models, have emerged as alternatives to solve optimization problems where the traditional techniques cannot be applied, or even where a solution model for the problem is not available with which the solution is found through empirical means. Given these capabilities to provide acceptable solutions in a timely manner for most of the complex problems encountered, metaheuristics has been applied successfully in most of the control system problems found in the literature. This work presents in general how the metaheuristics are being applied in the solution of control problems and performs a comparative study of performance among four algorithms bioinspirados in the tuning of the PID parameters. The following algorithms were used: Genetic Algorithm (AG), Genetic Algorithm in the Islands Model (AGMI), Bacterial Foraging Optimization (BFO) and Particle Swarm Optimization (PSO). The results demonstrate that the algorithms present an excellent performance in the tuning of the PID producing response that met the project requirements. Different systems with different characteristics were used to evaluate the algorithms. The PSO was shown as the best algorithm among the four used, producing response in a faster time and presented lower deviated standard in the trials.

KEYWORDS: Genetic Algorithms, PID Controller, Particle Swarm Optimization, Bacterial Foraging Optimization.

1 INTRODUÇÃO

Os problemas de otimização são encontrados frequentemente em diferentes contextos em nosso dia-a-dia, e por este motivo são cada vez mais encontradas novas técnicas para solucionar estes tipos de problemas. Uma vez que os problemas do mundo real apresentam-se como problemas complexos e que as vezes não podem ser solucionados pelas técnicas tradicionais, surgiu a necessidade da criação de técnicas capazes de oferecer soluções adequadas em um tempo aceitável. As metaheurísticas populacionais se baseiam em modelos encontrados na natureza, onde as espécies encontram formas de se adaptar e desta forma maximizar suas chances de sobrevivência em seu ambiente. Entre os conceitos baseados para a construção dos algoritmos tem-se a teoria da seleção natural proposta por Darwin, os modelos físicos e ainda a relação social de alguns grupos de animais e entre outros modelos (YANG, 2008; GLOVER & KOCHENBERG, 2008).

Em muitos sistemas de controle, existe uma grande preocupação na busca pelo desenvolvimento de controladores capazes de se adaptar as constantes mudanças encontradas nos ambientes onde eles estão inseridos e isso se mostra como um fator primordial para a qualidade no processo de controle. E ainda, observa-se uma tendência cada vez maior no aumento da complexidade dos sistemas a serem controlados, exigindo desta forma controladores cada vez mais robustos (CHEN, 1995; DORF & BISHOP, 2001; NISE, 2002; OGATA, 2003). Uma vez que a muitos dos problemas de controle encontrados na literatura podem ser transformados em um problema de otimização, as metaheurísticas populacionais vêm sendo aplicadas com êxitos em muitos desses problemas (TARIQUE et al, 2013; VIDAL et al, 2015; CHANG & SHIH, 2010; RAJINIKANTH & LATHA, 2012; LATHA et al, 2013).

Na literatura estão disponíveis diferentes tipos de controladores, e entre estes o controlador Proporcional-Integral-Derivativo (PID) ainda se destaca como o mais utilizado em muitas malhas de controle industriais. As técnicas disponíveis na literatura para o ajuste dos parâmetros dos controladores PID visam garantir o desempenho satisfatório para os sistemas encontrados, e quando se conhece as características dinâmicas do processo a ser controlado, se dispõe de diferentes técnicas analíticas para o projeto de controladores PID. No entanto, muitas vezes se desconhece as características desses sistemas e então a saída é utilizar meios empíricos para realizar a

sintonia. Os métodos heurísticos propostos por Ziegler-Nichols são um dos mais utilizados nestes casos e oferecem bons valores de parâmetros, mas muitas vezes, impõem características específicas para os seu uso na sintonia dos parâmetros. E uma vez que seus ganhos são escolhidos, seu funcionamento se torna satisfatório apenas se não ocorrerem mudanças significativas no processo em que o controlador está inserido. Na ocorrência de mudanças no comportamento do processo, caso não havendo alterações nos parâmetros o controlador sintonizado previamente pode não funcionar corretamente e até deixar o sistema em malha fechada instável (NISE, 2002; OGATA, 2003).

Na literatura, são propostas diferentes metodologias para utilizar metaheurísticas populacionais para realizar a sintonia de controladores PI e PID modificados para aplicação no controle dos mais variados tipos de sistemas (PERREIRA, 2008; RAJINIKANTH & LATHA, 2010; CHIOU et al, 2012; IBRAHIM et al, 2013; GUPTA & GUPTA, 2014; KANG et al, 2014; CHAIB et al, 2015; PATEL & KUMAR, 2015; SAYAED et al, 2015; SINGH et al, 2015) .

1.1 Objetivos

Diante da importância do desenvolvimento de técnicas eficientes no projeto de controladores robustos e na crescente utilização de diferentes técnicas de Inteligência Computacional nos problemas de sistemas de controle, este trabalho tem como objetivo apresentar, de um modo geral, como as metaheurísticas vêm sendo aplicadas nos problemas de sistemas de controle, destacando as principais características dos algoritmos usados, para isso é realizado um estudo comparativo entre o desempenho de quatro metaheurísticas populacionais (AG, AGMI, PSO e o BFO) na sintonia dos parâmetros de controladores PID, usando uma abordagem baseada na diminuição do erro entre as respostas dos indivíduos das populações e a resposta criada a partir de exigências de projetos passadas como entrada para os algoritmos.

A metodologia proposta será aplicada a diferentes sistemas e serão avaliadas as características das respostas geradas em relação às características esperadas no projeto. Serão avaliados também os índices de desempenhos *Integral of the Absolute Error* (IAE), *Integral of the Square Absolute Error* (ISE), *Integral of Time multiplie by Absolute Error* (ITAE) e *Integral of Time multiplie by Square Error* (ITSE) gerados

pelo uso dos controladores projetados com cada algoritmo, além do tempo de processamento de cada um.

1.2 Estrutura do trabalho

Este trabalho está dividido em 5 Capítulos, estando os próximos organizados da seguinte maneira: O Capítulo 2 apresenta a fundamentação teórica sobre as metaheurísticas populacionais usada no trabalho, bem como o estado da arte das metaheurísticas populacionais aplicadas em sistemas de controle. O Capítulo 3 apresenta a teoria de sistemas de controle relacionada com os controladores PID, destacando as principais características desse controlador e apresentando, resumidamente, três formas de sintonia usadas na literatura. O Capítulo 4 apresenta a metodologia usada neste trabalho para realizar a sintonia dos parâmetros de um controlador PID, e contém também os resultados e a análise dos ensaios realizados para avaliar as metaheurísticas nos diferentes tipos de sistemas. O Capítulo 5 contém as conclusões do trabalho.

2 METAHEURISTICAS POPULACIONAIS

2.1 Introdução

O processo de otimização pode ser encontrado em quase toda atividade humana, desde tarefas mais simples, realizadas diariamente, como planejar o caminho mais rápido de casa para o trabalho, até tarefas mais complexas, como o projeto de uma usina hidrelétrica. Otimização está em toda parte, de empresas para projetos de engenharia, de estudos acadêmicos e científicos a aplicações práticas. Sendo assim, fica evidente a importância do estudo e o desenvolvimento de técnicas e ferramentas que visem atender esta necessidade de otimização, não importando a aplicação. Seja qual for o problema do mundo real é quase sempre possível a formalização do problema de otimização em uma forma genérica. Em termos matemáticos, o problema de otimização com objetivos explícitos pode em geral, ser expresso como um problema de otimização não-linear com restrições, usando (YANG, 2008):

Maximizar/minimizar

$$\vec{X} \in \mathfrak{R}^n f(\vec{X}), \vec{X} = (x_1, x_2, \dots, x_n)^T \in \mathfrak{R}^n \quad (2.1)$$

Sujeito a

$$\phi_j(\vec{X}) = 0, (j = 1, 2, \dots, M) \quad (2.2)$$

$$\psi_k(\vec{X}) \geq 0, (k = 1, 2, \dots, N) \quad (2.3)$$

Diante da possibilidade de representação em uma forma matemática dos problemas, surgiram os métodos de otimização tradicionais ou determinísticos, que utilizam o cálculo diferencial, a programação linear ou informações sobre o gradiente da função a ser otimizada. Estes métodos apresentam bons desempenhos nas soluções de problemas, desde que a função seja diferenciável em todos os pontos e, em alguns casos, exigindo que a mesma seja convexa. Quando estas características não estão presentes, ou o problema encontrado não apresenta uma forma matemática bem definida, a eficiência no uso destas técnicas determinísticas diminui. Como a maioria dos problemas do mundo real apresentam funções que podem ter descontinuidades em alguns pontos e apresentam muitas vezes vários pontos de ótimos locais, aumentando a

complexidade do problema, foi necessária a criação de outras técnicas para serem aplicadas nesse tipo de problema. Assim, surgiram os métodos de otimizações modernos, que têm relação com estratégias de solução baseadas em heurísticas, e implementam, muitas vezes, o uso de estruturas de compartilhamento de informações entre vizinhanças. A heurística é uma estratégia que visa produzir soluções aceitáveis para problemas complexos em um tempo razoável. Muitas vezes, a complexidade dos problemas torna impossível a busca por cada possível solução do espaço de busca, e o objetivo então passa a ser encontrar boas soluções viáveis em uma escala de tempo aceitável, mesmo que não exista garantia de que as melhores soluções podem ser encontradas (RAO, 2009).

Esta necessidade de resolver problemas, cuja dificuldade de se encontrar uma solução aceitável é refletida na quantidade de tempo e memória computacional usada no processo, levaram os pesquisadores nas últimas décadas a desenvolverem meta-heurísticas guiadas por funções objetivos, que implementam uma forma de interação entre processos de melhoria local e um alto nível de estratégia que visam garantir ao algoritmo a capacidade de escapar de pontos de ótimos locais, realizando deste modo uma pesquisa robusta sobre o espaço das soluções do problema. Estes métodos definem critérios para a transição de uma solução para a outra, ou para construir ou destruir soluções em processos construtivos e destrutivos (GLOVER & KOCHENBERG, 2008).

A maioria das técnicas encontradas nos procedimentos usados pelas meta-heurísticas são baseados em modelos encontrados na natureza. Entre estes modelos tem-se os que utilizam conhecimento baseado em população, onde cada indivíduo que a compõem são uma representação de uma possível solução para o problema, e a partir da avaliação individual é utilizado um procedimento de troca de informações positivas para que a população possa convergir para soluções próximas do ponto ótimo. Nestes processos são utilizados operadores que apresentam componentes probabilísticas que garantem aos algoritmos a capacidade de “fuga” de armadilhas criadas pelos pontos de ótimos locais fazendo com que todos os pontos do espaço de busca tenham uma chance de ser visitado no processo de pesquisa. Os Algoritmos Genéticos (AG) pertencem à classe dos algoritmos evolutivos que se baseiam na teoria da seleção natural proposta por Darwin. Os AGs implementam operadores que possibilitam a troca de material genético entre um indivíduo solução e os demais, fazendo com que os mais aptos possam gerar novos descendentes que tendem a mesclar as boas características de seus progenitores. A ideia é fazer com que uma população inicial possa evoluir para boas

soluções através da competição de seus indivíduos (GLOVER & KOCHENBERG, 2008).

Outro paradigma de metaheurística populacional, porém baseado na cooperação entre seus indivíduos é a inteligência de enxame, que está baseada na forma pela qual os seres vivos de diferentes espécies colaboram entre si para realizarem tarefas tais como caça de presas, para a defesa contra predadores ou a busca por recursos como água e comida. Um bom exemplo dessa cooperação entre seres de uma mesma espécie pode ser observado na interação entre os grupos sociais de alguns insetos tais como abelhas e colônias de formigas onde se tem uma hierarquia organizacional que permite ao coletivo apresentar características de inteligência, através da troca eficiente de informações sobre o ambiente entre os diferentes indivíduos. Um dos principais algoritmos de inteligência de enxame é conhecido como PSO (*Particle Swarm Optimization*) introduzido por Kennedy e Eberhart (1995), este algoritmo é baseado no movimento de grupos de animais, tais como bandos de aves e cardumes de peixe. As observações de como esses animais se movimentam em conjunto de forma sincronizada e harmoniosa, levou a tentativa de criação de modelos computacionais que descrevessem o princípio por trás dessa comunicação (CLERC, 2006; SUN et al, 2012; XIANG & GAO, 2014).

2.2 Algoritmo Genético (AG)

2.2.1 Histórico dos Algoritmos Genéticos

Os Algoritmos evolutivos se baseiam na teoria da seleção natural proposta por Darwin em seu livro conhecido atualmente como “A origem das espécies”, utilizando os seus conceitos e aplicando modelos computacionais que visam através da emulação do processo de evolução biológico a tentativa de aplicação na solução de problemas de otimização. Segundo a teoria da seleção natural, todos os indivíduos dentro de um ecossistema competem entre si por recursos limitados como comida e água. E os indivíduos (animais, vegetais, e insetos etc.) de uma mesma espécie que não obtêm êxito tendem a ter uma prole menor e esta descendência reduzida faz com que a probabilidade de ter suas características passadas ao longo de sucessivas gerações seja menor, ou seja, somente as características que tornam os indivíduos mais aptos para a sobrevivência no seu ecossistema têm uma alta probabilidade de estar presente nas gerações futuras (LIDEN, 2006).

Quando Darwin afirmou que a evolução e a seleção natural faziam com que as espécies fossem se adaptando naturalmente ao meio ambiente, ele não sabia quais eram os mecanismos básicos através dos quais esta adaptação acontecia, pois o processo de transmissão de informação genética era ainda desconhecido na época. Neste mesmo período um padre chamado Gregor Mendel, investigava a herança de características, ou traços, em seus experimentos com ervilhas. Ao examinar os híbridos que eram gerados a partir de tipos diferentes de plantas ele obteve algumas noções sobre as interações entre as características herdadas. Ele observou que certas características se sobressaíam em híbridos gerados, independentemente de qual planta havia doado o pólen, por exemplo, cruzando plantas altas com plantas baixas, todos os híbridos gerados eram altos. Mendel declarou que a característica, ou gene, como ficaria conhecido mais tarde, para plantas altas era dominante em relação à característica para plantas baixas. Embora as experiências de Mendel tenham estabelecido as bases para o estudo da genética, foram somente 30 anos depois de sua morte que Walter Sutton (1877-1916) descobriu que os genes eram parte dos cromossomos no núcleo (MITCHELL, 1998).

No entanto, a teoria de Darwin enfatizou o papel da contínua variação dentro de cada espécie. Em contraste, diferenças distintas entre espécies não são incomuns na natureza, isto é, existente variação descontínua. Hugo de Varis (1848-1895) observou que, em uma população de cultivo de plantas, notavelmente diferentes variações ocasionalmente aparecem. Para explicar essa variação descontínua, de Varis desenvolveu a teoria da mutação. Superficialmente, a nova ciência da genética da época parecia apoiar a teoria da mutação da evolução contra o darwinismo ortodoxo. Com uma melhor compreensão da estrutura dos genes, os geneticistas vieram a perceber o quão sutil os efeitos da mutação poderiam ser. Se uma característica é determinada por um único gene, a mutação pode ter um efeito dramático, mas se um conjunto de genes combina para controlar essa característica, a mutação em um deles pode ter apenas um efeito negligenciável. Está claro, portanto, que não há distinção entre a mutação e a teoria darwiniana da evolução, já que elas se sobrepõem. O princípio da seleção natural, no entanto, permanecem sólidos (MITCHELL, 1998).

A história dos Algoritmos Genéticos tem início no começo da década de 1950 e meados de 1960 quando muitos pesquisadores começaram a estudar de forma independente sistemas evolucionários com a ideia de que esses sistemas poderiam ser usados como ferramentas para a otimização em problemas de engenharia. A ideia em todos estes sistemas era evoluir uma população de candidatos à solução de um

determinado problema, usando operadores inspirados na genética básica e na seleção natural, descrita anteriormente (MITCHELL, 1998; LIDEN, 2006).

Foi somente na década de 1970 quando o professor norte-americano John Holland e os seus estudantes da universidade de Michigan tentaram desenvolver um algoritmo que incorporasse os conceitos da evolução biológica e fosse capaz de ser usado para resolver diversos problemas, que foi criado o Algoritmo Genético. Através do estudo da evolução das espécies foi proposto um modelo heurístico computacional que quando implementado poderia oferecer boas soluções para problemas extremamente difíceis que eram irresolúveis computacionalmente até aquela época (MITCHELL, 1998; ROTHLAUF, 2006).

Em 1975, Holland publicou seu livro “*Adaptation in Natural and Artificial Systems*” onde fez um estudo dos processos evolutivos, e apresentou o algoritmo genético como uma metáfora para os processos de evolução, de uma forma que fosse possível estudar sua adaptação e evolução simulando-os em computadores. Desde então os AGs começaram a se expandir em toda a comunidade científica, gerando uma série de aplicações nas mais diversas áreas. Hoje em dia muitos pesquisadores da área de algoritmos evolucionários têm buscado a interdisciplinaridade como forma de inspiração em outras áreas de tal forma a absorver seus conceitos e aplicá-los aos algoritmos genéticos para torná-los mais eficientes (LIDEN, 2006).

2.2.2 Características dos Algoritmos Genéticos

Os Algoritmos Genéticos pertencem a classe dos Algoritmos Evolutivos (EA), e são métodos de busca estocásticas e de otimização baseados em uma metáfora para o processo de evolução biológico, usando técnicas baseadas em conceitos tais como hereditariedade, mutação, seleção natural e recombinação genética (*crossover*) (GOLDBERG, 1989; MICHALEWICZ, 1992; HAUPT & HAUPT, 2004; LIDEN, 2006).

Para que um problema possa ser resolvido utilizando um AG ele precisa ser representado no contexto dos algoritmos genéticos. Como os AGs são altamente inspirados na genética básica e na teoria da evolução das espécies, há uma analogia muito forte entre os termos da biologia e os termos usados computacionalmente. Destacam-se (LIDEN, 2006):

- **Genes:** Na natureza a unidade básica de codificação de características é o gene. De acordo com sua localização e valor, eles codificam uma determinada característica. No AG os caracteres que formam o indivíduo solução e os caracterizam são chamados de genes. Na codificação, no formato de cadeia binária, cada bit é um gene.
- **Cromossomo:** São estruturas que agrupam vários genes e são responsáveis por formar as características genéticas de um indivíduo de uma dada espécie. No contexto do AG os termos cromossomo e indivíduos são usados para representar uma possível solução do problema.
- **Fenótipo:** Na natureza é o conjunto de características que definem um indivíduo. Na linguagem do AG é o conjunto de parâmetros que um determinado cromossomo representa.
- **Locus:** É a posição ocupada por um gene em um cromossomo.
- **População:** É formada pelo conjunto de indivíduos da mesma espécie que habitam a mesma área no mesmo espaço de tempo. No AG a população é formada por cromossomos (indivíduos-solução).

Os AGs partem de uma população inicial, que é formada por um conjunto de cromossomos criados aleatoriamente. Os cromossomos representam uma solução-candidata para o problema. Na literatura são encontradas diferentes formas de representação dos parâmetros, ficando a melhor e mais indicada representação de acordo com o tipo de problema a ser resolvido, sendo que entre as disponíveis tem-se a representação real e a codificação binária (GOLDBERG, 1989; MICHALEWICZ, 1992; LIDEN, 2006). A população inicial de um AG deve ser criada aleatoriamente de forma a ser inserida o maior número possível de diversidade genética, fazendo com que o AG faça uma busca por uma maior área no início do espaço solução. O tamanho da população de um AG é um parâmetro muito importante para o desempenho do mesmo. Ele deve ser grande o suficiente para gerar diversidade nos materiais genéticos entre cada indivíduo e, ao mesmo tempo, não ser grande demais a ponto de deixar o programa lento.

Depois que a população é criada, cada indivíduo que a compõe é avaliado através de uma função custo. Esta função é a maneira pela qual os AGs avaliam cada candidato à solução do problema em questão. Portanto ela determina diretamente o desempenho do AG. Sendo assim, é necessário que seja escolhida de forma cuidadosa e nela deve ser embutido todo o conhecimento que se possui sobre o problema, tanto suas

restrições quanto seus objetivos e qualidades esperadas. Esta função calcula um valor numérico que reflete o quão bons os parâmetros são para solução ótima (MICHALEWICZ, 1992; HAUPT & HAUPT, 2004; LIDEN, 2006).

Depois que todos os indivíduos são avaliados é necessário realizar a seleção dos pais que irão gerar os filhos que irão compor a população na próxima geração. O método usado para selecionar os pais deve simular o mecanismo da seleção natural que atua sobre as espécies biológicas, em que os pais mais aptos à sobrevivência no ambiente geram mais filhos, ao mesmo tempo em que os menos aptos também podem gerar descendentes, porém em menor proporção. Em outras palavras deve ser garantido aos indivíduos do AG com o maior valor de avaliação, uma probabilidade maior em relação aos indivíduos menos aptos, porém sem desprezar completamente os indivíduos com avaliação extremamente baixas. Entre os métodos de seleção mais usados na literatura estão a roleta e a seleção por torneio (GOLDBERG, 1989; MITCHELL, 1998; ROTHLAUF, 2006).

A principal característica da evolução das espécies emulada pelo AG é a recombinação dos materiais genéticos entre os indivíduos, para gerarem descendentes que sejam cada vez mais aptos à sobrevivência no ambiente. Estas características são realizadas através do uso dos operadores genéticos que são: o operador de crossover e o operador de mutação.

Quando os pais são selecionados, sobre eles é aplicado o operador de *crossover*, que consiste na mistura dos genes para formar novos indivíduos (filhos). A forma como ocorre essa mistura do material genético dos pais vai depender do tipo de representação usado no AG e o tipo de problema a ser resolvido. Geralmente o processo de recombinação genética consiste em particionar o cromossomo dos pais em um ou mais pontos e formarem cada um dos filhos usando pedaços de ambos.

O operador de mutação é usado para inserir novos genes em alguma posição dos cromossomos dos filhos que foram gerados. O objetivo é fazer com que novas características possam aparecer e deste modo se aumente a diversidade genética na população. Este operador tem associado um parâmetro chamado taxa de mutação, que consiste na probabilidade do operador atuar e mudar um determinado gene no cromossomo, já que, na natureza a mutação não acontece de forma frequente e sim com uma probabilidade pequena. Esta é uma das características importantes emuladas pelo AG. Através dela é possível que o AG evite convergências prematuras evitando assim, ficar preso em máximos ou mínimos locais.

Uma vez que sejam gerados todos os filhos de acordo com o processo de seleção, recombinação e mutação descritas anteriormente, a nova população formada pelos filhos irá substituir a população antiga e é realizada a avaliação dos novos indivíduos. Então pode ser verificado se algum dos indivíduos desta população atende os requisitos impostos no critério de parada, se sim ele será então retornado como a solução encontrada pelo AG. Porém no caso de que nenhum critério de parada tenha sido satisfeito, o algoritmo irá para a seleção dos pais para formarem os novos filhos que irão substituir a população. Este processo se repete ao longo de um número máximo de gerações imposta como critério de parada.

Resumidamente um AG básico pode ser definido pelos seguintes passos: Criação da população inicial, avaliação dos indivíduos, seleção dos pais, *crossover* e mutação (GOLDBERG, 1989; MITCHELL, 1998; ROTHLAUF, 2006). A Figura 2.1 apresenta o fluxograma dos passos do algoritmo genético.

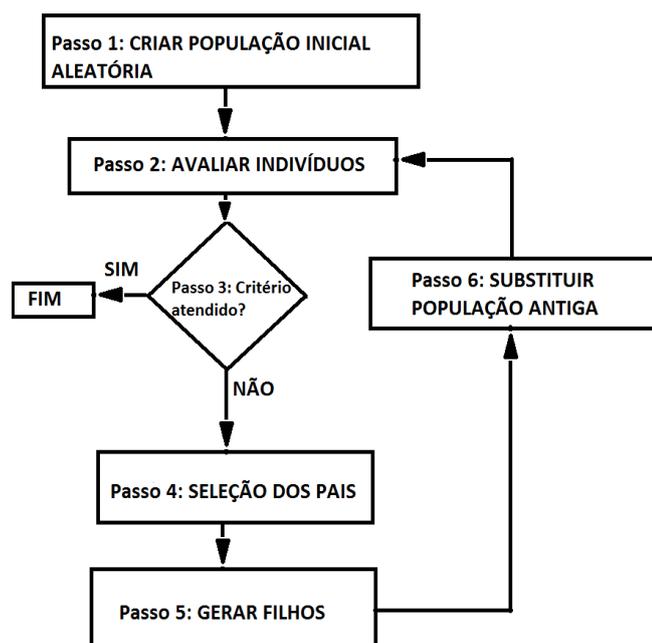


Figura 2.1-Fluxograma de um AG básico

2.2.3 Seleção dos Pais

O método de seleção dos pais no AG tenta emular o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os indivíduos mais aptos à sobrevivência no ambiente geram mais filhos, ao mesmo tempo em que os pais menos

aptos também podem gerar descendentes. Para seleção dos pais, privilegiam-se os indivíduos com função de avaliação alta, porém sem desprezar completamente os indivíduos com função de avaliação extremamente baixa (LIDEN, 2006; HAUPT & HAUPT, 2004).

O fato de ser importante garantir que indivíduos com baixa avaliação também se reproduzam é que, se somente os indivíduos com avaliação alta se reproduzirem, a população tenderá a ser composta de indivíduos cada vez mais semelhantes e faltará diversidade para que a evolução prossiga de forma satisfatória, sendo que a este efeito chama-se de convergência genética prematura. Isso pode fazer com que o Algoritmo Genético fique preso em pontos de ótimos locais. Na literatura encontram-se disponíveis diferentes métodos para realizar a etapa de seleção em um AG, onde destacam-se:

a) *Seleção por Roleta*

Após a avaliação de todos os indivíduos, é calculada a avaliação relativa de cada cromossomo em relação à avaliação total da população (soma de todas as avaliações dos indivíduos da população), sendo este processo denominado de função de adequabilidade. Em seguida é criada uma “roleta” (virtual) onde cada cromossomo receberá uma parte proporcional a sua avaliação relativa.

A Tabela 2.1 apresenta 4 indivíduos que formam uma população qualquer e suas avaliações e a Figura 2.2 apresenta a sua roleta para a seleção.

Tabela 2.1-Exemplo da separação da roleta em uma população (LIDEN, 2006)

Indivíduo	Avaliação	Parte da roleta(%)	Parte da roleta (%)
1	1	1.61	5.8
2	9	14.51	52.2
3	16	25.81	92.9
4	36	58.07	209.1
Total	62	100.00	360.0

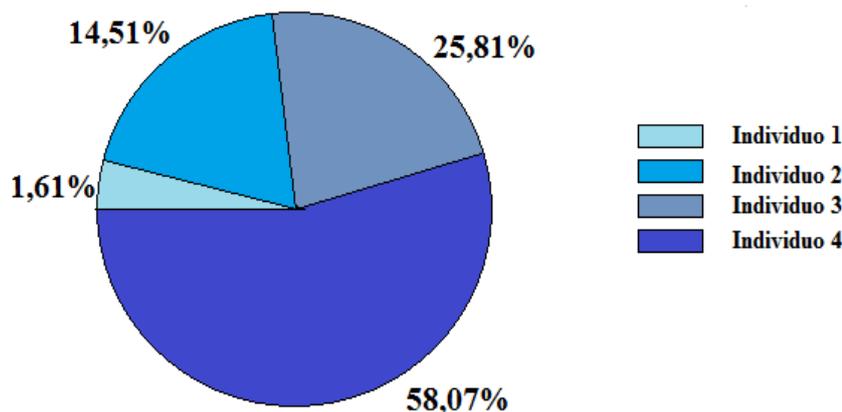


Figura 2.2-Roleta para os indivíduos da Tabela 2.1 (LIDEN, 2006)

Para simular o ato de rodar a roleta escolhe-se um número aleatório entre 0 e 100 (representando a porcentagem que corresponde a cada indivíduo), ou entre 0 e 360 (representando uma posição do círculo), ou ainda entre 0 e a soma total das avaliações (representando um pedaço do somatório). Sendo assim, os indivíduos com uma avaliação maior tem uma probabilidade proporcional de ser escolhido conseguindo repassar seus materiais genéticos para as próximas gerações.

b) Seleção por Torneio

Este método consiste em selecionar uma série de indivíduos da população atual e fazer com que eles entrem em uma competição direta pelo direito de ser o pai, usando como arma a sua avaliação. Neste método, existe um parâmetro denominado “tamanho do torneio (k)” que define quantos indivíduos são selecionados aleatoriamente dentro da população para competir. Uma vez escolhidos os competidores, aquele dentre eles que possuir a melhor avaliação é selecionado para a aplicação do operador genético. O valor do parâmetro k deve estar entre 2 e um valor N, o qual deve ser menor que o tamanho da população, por razões óbvias, pois se ele for do tamanho da população o indivíduo com maior avaliação sempre será escolhido causando uma convergência genética prematura na população. Como os indivíduos são selecionados para participar do torneio de forma aleatória, não existe nenhum favorecimento para os melhores indivíduos, como no caso da roleta. A única vantagem que os melhores indivíduos da população têm é que, se selecionados eles vencem o torneio (LIDEN, 2006).

2.2.4 Operador de *Crossover*

Uma das principais características da evolução das espécies emulada pelo AG é a recombinação dos materiais genéticos entre indivíduos, para gerar descendentes cada vez mais aptos a sobrevivência no ambiente. Através da hereditariedade os pais podem propagar seus genes para as próximas gerações e ao longo de sucessivas gerações a população no AG tenderá a ser composta por indivíduos com características cada vez mais promissoras. O operador de *crossover* entra em execução logo após a seleção dos pais e se encarrega de formar os filhos combinando os materiais genéticos dos indivíduos escolhidos. De acordo com o tipo de representação dos cromossomos adotada no AG, se dispõem de diferentes tipos de operadores de *crossover*.

a) *Operador de crossover para representação binária*

No AG de representação binária cada possível solução do problema é representada por um cromossomo, que consiste em uma sequência de bits, onde cada variável existente no domínio do problema é composta por um pedaço deste vetor binário. O operador de *crossover* age dividindo o cromossomo dos pais em um ou mais pontos de corte para que os filhos sejam formados a partir da combinação desses pedaços. Um ponto de corte consiste em uma posição entre dois genes (bits) de um cromossomo. Cada indivíduo formado por n genes contém $n - 1$ pontos de corte, e este ponto será o ponto de separação entre os materiais genéticos de cada pai. A Figura 2.3 apresenta um indivíduo formado por cinco genes, e as setas apontam para os seus quatro pontos de corte (LIDEN, 2006).

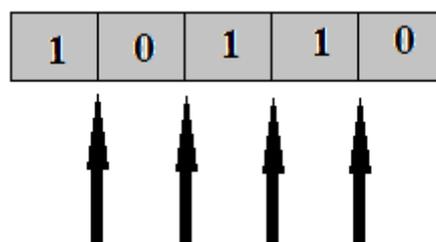


Figura 2.3-Exemplo de um cromossomo binário e seus pontos de corte (LIDEN, 2006).

Crossover de um ponto: Depois de sorteado o ponto de corte, os pais são separados em dois pedaços: um a esquerda do ponto de corte e outro a direita. O passo seguinte é fazer a concatenação entre as partes do primeiro pai e as partes do segundo pai para formar os filhos. A Figura 2.4 mostra o processo do *crossover* de um ponto.

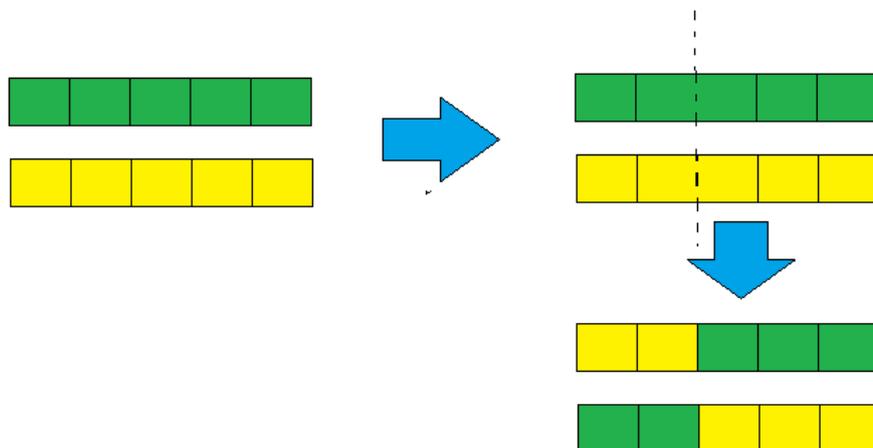


Figura 2.4-Crossover de um ponto (LIDEN, 2006).

Crossover de dois pontos: Diferentemente do operador de um ponto, neste serão escolhidos dois pontos de corte e depois será feita a concatenação das partes dos pais para formar os filhos. O primeiro filho será formado pelas partes do primeiro pai fora dos pontos de cortes e pela parte do segundo pai dentro dos pontos de cortes e o segundo filho pelas partes restantes. A Figura 2.5 apresenta o funcionamento do operador de *crossover* de dois pontos.



Figura 2.5-Crossover de dois pontos (LIDEN, 2006).

Crossover Uniforme: Este operador não utiliza pontos de cortes. Para cada posição dos genes dos filhos é sorteado um número: um ou zero. Se o número sorteado for igual a um, o primeiro filho recebe o gene corrente da posição do primeiro pai e o segundo filho o gene do segundo pai. De forma análoga, se o número for igual a zero, então o primeiro filho recebe o gene do segundo pai e o segundo filho recebe o gene do primeiro pai. A Figura 2.6 apresenta o funcionamento do *crossover* uniforme.

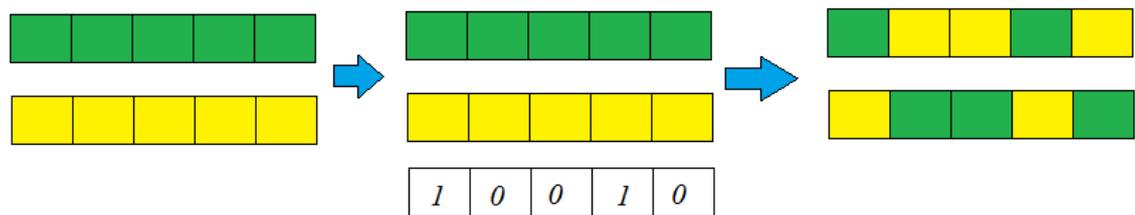


Figura 2.6-Crossover Uniforme (LIDEN, 2006).

b) *Operador de crossover em representação Real*

Em problemas cujo domínio das variáveis é um valor contínuo ou onde o número de variáveis do problema é muito grande, a representação por números binários no AG pode não ser a mais indicada. Seja pela necessidade de uma maior precisão na representação ou o número elevado de variáveis a ser representada, o tamanho dos cromossomos que seriam usados nestes casos teriam que ser grandes o que poderia resultar em um maior consumo de recursos computacionais para a implementação do algoritmo. Como alternativa se dispõe do AG com representação Real, onde cada cromossomo é formado por um vetor de números reais do tamanho do número de variáveis do problema. Neste caso dada uma função objetivo $f(x_1, x_2, \dots, x_n)$ um cromossomo “i” da população seria representado por:

$$C_i = [p_1, p_2, \dots, p_n] \quad (2.1)$$

Neste caso cada variável x_n seria representada por um parâmetro p_n no cromossomo. Assim como na representação binária o objetivo da utilização do operador de *crossover* é realizar a combinação das informações genéticas contidas em cada um dos pais para formar os filhos, e sendo assim na literatura são propostos diferentes operadores, como por exemplo:

Crossover média: Dado dois cromossomos P_1 e P_2 representando respectivamente o pai 1 e o pai 2, o filho (F) será gerado a partir da médias entre os parâmetros deles usando:

$$F = \frac{(P_1 + P_2)}{2} \quad (2.2)$$

Crossover Média Geométrica: Onde cada gene F_i do filho F é gerado de acordo com a Equação (2.3). Este operador tende a levar os parâmetros para o centro do intervalo permitido causando uma perda de diversidade.

$$F_i = \sqrt{P_{1i}P_{2i}} \quad (2.3)$$

Crossover Aritmético: São gerados dois filhos a partir de:

$$F_1 = \beta \cdot P_1 + (1 - \beta) \cdot P_2 \quad (2.4)$$

$$F_2 = \beta \cdot P_2 + (1 - \beta) \cdot P_1 \quad (2.5)$$

Onde β é um número aleatório gerado entre 0 e 1.

2.2.5 Mutação

A ação do operador de *crossover* causa uma pressão para a convergência da população no AG. Quando os indivíduos estão próximos da região do ponto ótimo essa convergência é desejada, porém como geralmente as funções a serem otimizadas podem possuir vários pontos de ótimos locais e neste caso a ação do operador de *crossover* em certas ocasiões pode levar a convergência para estes pontos, fazendo com que a população fique presa, sendo necessário evitar esta situação. Para fazer com que a população do AG continue a possuir diversidade e possa realizar a busca em pontos que não estavam presentes na população inicial é usado o operador de mutação. Este operador age depois que o operador de *crossover* gera os filhos, e consiste em escolher uma posição do cromossomo dos filhos e inserir um novo gene, isso faz com que novas informações sejam inseridas na população. A Figura 2.8 exemplifica esta ação. No caso da representação binária, a ação da mutação consiste em inverter o bit da posição escolhida, no caso do bit ser 1 ele é alterado para 0 e de forma análoga se ele for 0 é colocado 1 no lugar. Na representação real a mutação substitui um determinado gene por um número real aleatório dentro do limite do parâmetro, e em alguns casos pode ser gerada a mutação a partir de uma distribuição especificada centrada em zero. Assim como na natureza a mutação não ocorre sempre e no AG é especificado um parâmetro chamado taxa de mutação, ele controla quando o operador vai atuar.

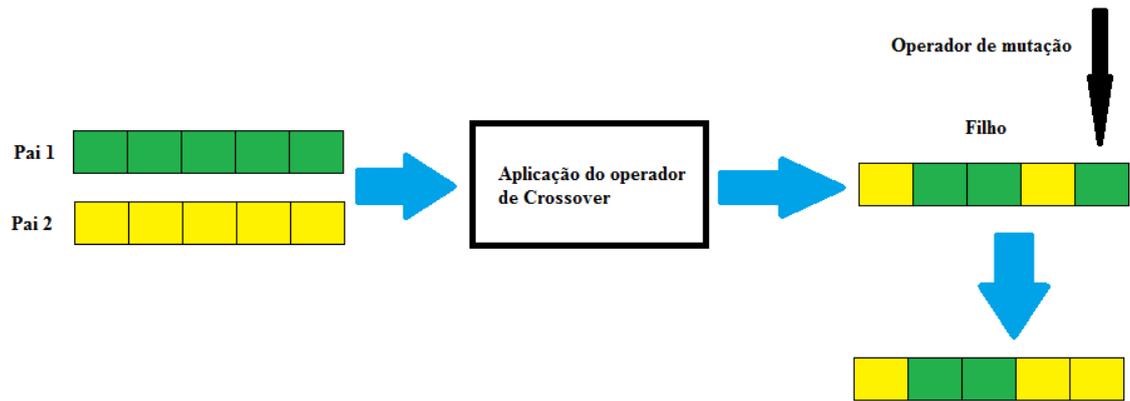


Figura 2.7-Ação do operador de mutação

2.2.6 Elitismo

A ação do elitismo visa garantir que o melhor indivíduo encontrado na próxima geração tenha sempre avaliação igual ou superior ao melhor indivíduo da geração passada, já que ao ser executado os operadores de *crossover* e mutação tem uma chance de os filhos gerados terem avaliação inferior ou que o melhor indivíduo não seja selecionado para gerar descendentes. O elitismo faz com que uma copia exata do melhor indivíduo seja passada para a próxima geração (LIDEN, 2006; MICHALEWICZ, 1992).

2.3 Particle Swarm Optimization (PSO)

2.3.1 Inteligência de Enxame

Para conseguir sobreviver em seu habitat cada indivíduo deve ser capaz de interagir, mudar e adaptar-se aos desafios encontrados diariamente em seu caminho, seja no processo de busca de comida, água ou de parceiros reprodutivos. Como é afirmado pela teoria da seleção natural, frequentemente as espécies estão competindo pelo direito de sobrevivência em seu ambiente. Para alguns indivíduos de determinadas espécies a tarefa de sobrevivência por conta própria poderia ser, um tanto quanto difícil, sejam pelas dificuldades de busca dos recursos necessários para a sobrevivência ou pela ameaça de predadores. Na tentativa de melhorar suas chances algumas espécies apresentam um comportamento coletivo e uma organização hierárquica que facilita a sobrevivência.

Então, quando estes grupos de seres vivos resolvem os problemas na natureza, essa capacidade costuma ser atribuída à “inteligência” do grupo. Na natureza é possível encontrar vários exemplos deste tipo de comportamento, por exemplo, muitas colônias sociais de insetos tais como formigas e abelhas, entre outras. Muitos destes grupos de animais apresentam propriedades interessantes e vêm servindo de modelos para a ciência da computação, para a criação de modelos computacionais para resolver diversos problemas (PEREIRA, 2008).

A inteligência de enxame (IE) foi uma terminologia introduzida inicialmente para descrever sistemas multi-robóticos. Foi originalmente usada para representar o comportamento coletivo de sistemas descentralizados e auto-organizados, que são naturais ou artificiais. Atualmente ela refere-se geralmente a uma classe de meta-heurísticas de otimização moderna. Um sistema de IE típico consiste de uma população de agentes simples que interagem localmente uns com os outros e com o ambiente onde vivem. O agente individualmente não mostra inteligência e segue regras bem simples, sem estruturas de controle centralizadas que digam como este deve se comportar. As interações entre tais agentes que compõem o sistema são locais e ainda apresentam certo grau de aleatoriedade. No entanto, os padrões globais funcionais coerentes ou comportamento global “inteligente” que é desconhecido para os agentes individualmente surgem devido a tais interações locais.

O algoritmo de otimização por enxames de partículas, pertence a categoria de algoritmos de Inteligência de enxames e é uma metaheurística baseada em população. O PSO foi originalmente proposto por Kennedy e Eberhart em 1995. Ele foi motivado pela observação do comportamento social de alguns grupos de animais, como bando de pássaros, cardumes de peixes e rebanhos de animais. Estes grupos representam exemplos interessantes de sistemas naturais, onde o comportamento agregado é encontrado, produzindo impressionantes movimentos sincronizados livres de colisões. Nestes sistemas o comportamento de cada membro do grupo é baseado em respostas simples e inerentes, embora o movimento do ponto de vista macroscópico seja relativamente complexo. Por exemplo, o voo de um bando de aves pode ser simulado com uma precisão relativa, simplesmente mantendo uma distância entre cada ave e seus vizinhos imediatos. Esta distância pode depender do seu tamanho e do comportamento desejado.

2.3.2 Características do PSO padrão

Assim como a maioria das metaheurísticas populacionais, o PSO parte de uma população inicial, ou enxame, criado de forma pseudoaleatória, que é formada por um número N de partículas. Cada partícula está associada a dois vetores, um com sua posição e outro com sua velocidade, respectivamente Equações (2.6) e (2.7). Estes vetores são inicializados de forma aleatória e a cada iteração do algoritmo tem seus valores atualizados, para que a partículas possam se movimentar pelo espaço de busca.

$$X_i(t) = \{x_{i1}(t), x_{i2}(t), \dots, x_{in}(t)\} \quad (2.6)$$

$$V_i(t) = \{v_{i1}(t), v_{i2}(t), \dots, v_{in}(t)\} \quad (2.7)$$

A posição de cada partícula é atualizada de acordo com sua velocidade atual, o conhecimento adquirido pela partícula e o conhecimento adquirido pelo bando. Neste algoritmo, os ajustes da velocidade e da posição são efetuados por equações algébricas simples. A Figura 2.8 apresenta um fluxograma com os passos que compõem o PSO padrão.

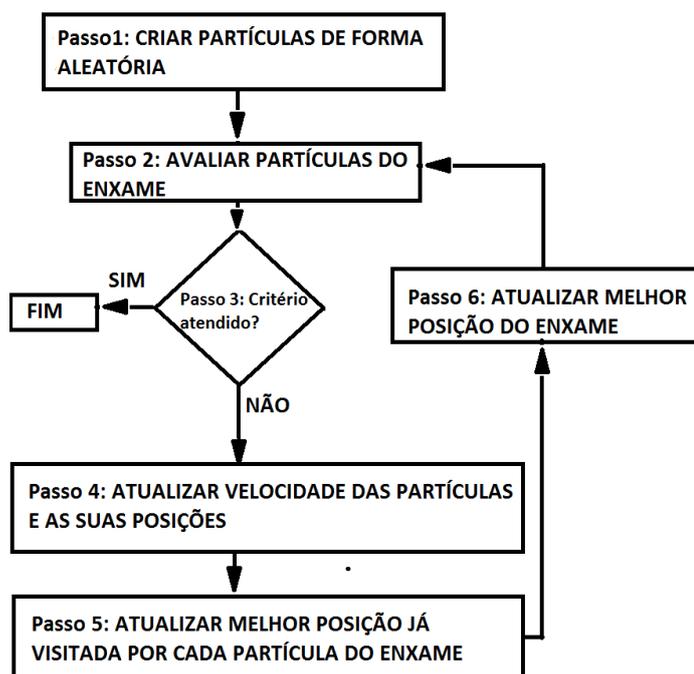


Figura 2.8-Fluxograma do PSO padrão

O desempenho individual de cada partícula é calculado mediante a aplicação de uma função custo, definida previamente. Uma vez que todas as partículas tenham sido

avaliadas, são armazenados os melhores resultados individualmente de cada partícula (P_{best}) e o melhor resultado obtido pelo grupo (G_{best}).

As partículas integram dois tipos de aquisição de conhecimento para se movimentarem no espaço de busca. O primeiro é através de sua experiência e o outro é através da partilha social de informação com os outros membros do enxame. O primeiro caso é conhecido como fator cognitivo e o último como fator social. O comportamento de cada partícula no espaço de busca é modelado pelo cálculo da velocidade de cada partícula e usando esta nova velocidade para o cálculo da próxima posição, utilizando:

$$V_i(t + 1) = wv_i(t) + r_1c_1(P_{best}(t) - X_i(t)) + r_2c_2(G_{best}(t) - X_i(t)) \quad (2.8)$$

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad (2.9)$$

Onde

w é o fator inercial;

r_1 e r_2 são números aleatórios gerados entre 0 e 1;

c_1 e c_2 são duas constantes que representam respectivamente os fatores cognitivos e sociais;

A cada iteração, depois que a velocidade e a posição do enxame foram atualizadas, é feito a atualização dos P_{best} de cada partícula, usando a seguinte expressão:

$$P_i(t + 1) = \begin{cases} X_i(t + 1), & \text{se } f(X_i(t + 1)) \geq f(P_i(t)) \\ P_i(t), & \text{caso contrário} \end{cases} \quad (2.10)$$

Em seguida, a melhor posição global (G_{best}) do bando é atualizada mediante os novos valores de P_{best} . Estas etapas se repetem até que um dos critérios de parada seja atendido.

Para evitar que a cada atualização das velocidades de cada partícula ocorra extrapolações, é limitado um valor de velocidade máxima v_{max} e um valor de velocidade mínima v_{min} . De tal forma que:

$$V_i(t + 1) = \begin{cases} v_{max}, & \text{se } V_i(t + 1) > v_{max} \\ v_{min}, & V_i(t + 1) < v_{min} \end{cases} \quad (2.11)$$

De uma maneira análoga, a posição das partículas também é limitada, para garantir que as soluções encontradas se mantenham em uma região factível. Após a atualização da posição é verificado se houve extrapolações da seguinte forma:

$$X_i(t + 1) = \begin{cases} x_{max}, & \text{se } X_i(t + 1) > x_{max} \\ x_{min}, & X_i(t + 1) < x_{min} \end{cases} \quad (2.12)$$

2.3.3 Considerações sobre o PSO

A parametrização do PSO pode determinar como será direcionada a busca das partículas pelo espaço de soluções do problema. Por exemplo, um valor elevado de fator de inércia (w) pode orientar a busca para uma pesquisa global, enquanto que um valor baixo pode orientá-la para uma busca mais local. Por esta razão algumas variações do PSO são propostas na literatura. Em (MOHAMED et al, 2010) é utilizado uma variação do algoritmo conhecida como *Inertial Weight Approach* PSO (IWAPSO), onde é usado um peso inercial decrescente em função do número de iterações. Fazendo com que no início do algoritmo as partículas tendam a fazer uma busca mais global e ao final das iterações uma busca mais local seja feita para melhorar a posição em torno da sua vizinhança.

No PSO as partículas são influenciadas pelo sucesso de outras as quais estão ligadas. No modelo padrão a partícula que exerce a maior influência no enxame é aquela com a melhor posição já visitada entre todas as outras. Esta é a topologia conhecida como estrela ou gráfico totalmente ligado. No entanto, existem outras topologias propostas na literatura, que definem diferentes estruturas sociais para realizarem a influência como a melhor partícula, que não são necessariamente partículas que estão próximas uma das outras.

Nos últimos anos diferentes variações do PSO vêm sendo propostas a partir da análise do desempenho do modelo padrão visando melhorar a eficiência do algoritmo. Janson e Middendorf propuseram o *Hierarchical* PSO (H-PSO), onde as partículas são organizadas em uma hierarquia dinâmica que define as vizinhanças, fazendo com que as partículas se movimentem subindo ou descendo a estrutura.

2.4 Bacterial Foraging Optimization (BFO)

2.4.1 Comportamento de forrageio bacteriano

Os animais estão frequentemente em busca de alimentos para a sua sobrevivência, e ao longo de sua vida desenvolvem mecanismos para maximizar suas chances de busca. Alguns animais possuem a estratégia de mover-se ao redor de seu próprio habitat à procura de suas presas ou de nutrientes. Em quanto outros preferem esperar que as presas venham seu encontro, e para isso utilizam “armadilhas” para as capturarem. Quanto maior for área disponível para forragear, mais alimento está disponível, porém custará mais energia para percorrê-la e é mais perigoso.

Esta característica de forrageio não é restrita somente aos animais, ela também pode ser observada no comportamento de seres pertencentes a outros reinos, como é o caso de algumas bactérias. A bactéria *Escherichia Coli*, por exemplo, tem um sistema de controle que habilita ela a procurar por comida e ao mesmo tempo evitar substâncias nocivas, que podem ser prejudiciais a sua saúde. A busca eficiente por comida e de um ambiente seguro é importante para a manutenção das espécies. A *E. Coli* em um ambiente promissor pode crescer, e em seguida reproduzir-se, dividindo-se ao meio gerando duas filhas com materiais genéticos idênticos (MAHESHWARI, 2012; BERG & BROWN, 1972).

A *E. Coli* se movimenta no seu ambiente através do nado, usando para isso uma estrutura chamada de flagelo. Estes mecanismos atuam como uma hélice giratória em contraste com um cílio, que atua como leme. Elas nadam ao rodarem os seus flagelos que emergem de posições aleatórias em sua superfície. Quando os flagelos rodam no sentido anti-horário, eles formam um feixe coerente que transmite uma força propulsiva. O resultado é um nado suave, quase em linha reta. Por outro lado, quando os flagelos rodam no sentido horário, o feixe se dispersa, porque os flagelos helicoidais podem se encaixar uns com os outros. Assim, a bactéria faz rodopios transitoriamente e, então, nada em uma direção diferente quando a rotação anti-horária é restaurada (BERG & BROWN, 1972; XING & GAO, 2014).

As bactérias usam o nado para manter-se em direções promissoras, no entanto ao se depararem com ambientes nocivos ou com ambiente improdutivos, elas tendem a executar “rodopios” a fim de encontrar outras direções que as levem para ambientes

melhores. Como dito em um ambiente promissor a E. Coli irá se reproduzir e ocorrerá o aumento da população, no entanto ao estar presente em um ambiente pobre de nutrientes ela tenderá a se movimentar para outro ambiente mais produtivo, se ela falhar em localizá-lo ela não poderá se reproduzir e eventualmente irá morrer (BERG, 2000; XING & GAO, 2014).

2.4.2 Características do algoritmo BFO

O BFO foi originalmente proposto por (PASSINO, 2002). Este algoritmo pertence a classe dos algoritmos de inteligência de enxame, e é baseado no comportamento da bactéria E. Coli, onde sua estratégia de forrageio é simulada para solucionar problemas de otimização. O BFO consiste basicamente de quatro mecanismos principais: Quimiotaxia, *swarming*, reprodução e o evento de eliminação-dispersão. Tais mecanismos podem ser caracterizados como:

Quimiotaxia: Como explicado anteriormente, o movimento da E. Coli pode ser executado através de duas maneiras diferentes: o nado, que significa movimentar-se na mesma direção, e o *tumbling*, ou “rodopio”, que consiste em um movimento em uma direção aleatória. O movimento da *i-ésima* bactéria pode ser calculado usando:

$$\theta_i(j + 1, k, l) = \theta_i(j, k, l) + C(i)\phi(j) \quad (2.13)$$

$$\begin{cases} \theta_i(j + 1, k, l) > \theta_i(j, k, l), \text{ nado } \phi(j) = \phi(j - 1) \\ \theta_i(j + 1, k, l) < \theta_i(j, k, l), \text{ tumbling } \phi(j) \in [0, 2\pi] \end{cases} \quad (2.14)$$

Onde $\theta_i(j + 1, k, l)$ representa a localização da *i-ésima* bactéria e j, k e l são as iterações respectivamente de quimiotaxia, reprodução e eliminação-dispersão. $C(i)$ é o tamanho da caminhada da *i-ésima* bactéria. E $\phi(j)$ é o ângulo de direção do movimento na *j-ésima* interação da quimiotaxia.

Swarming: Diante de algumas situações, as bactérias liberam atrativos para sinalizar as outras bactérias para elas se aproximarem e nadarem juntas, enquanto que elas também liberam repelentes para sinalizarem para outras ficarem a uma distância mínima da mesma. A célula de sinalização foi representada em (PASSINO, 2002) usando a seguinte equação:

$$\begin{aligned}
J_{cc}(\theta, P(j, k, l)) &= \sum_{i=1}^S J_{cc}^i(\theta, \theta_i(j, k, l)) \\
J_{cc}(\theta, P(j, k, l)) &= \sum_{i=1}^S \left[-d_{attract} \exp \left(-w_{attract} \sum_{m=1}^P (\theta_m - \theta_m^i)^2 \right) \right] \\
&+ \sum_{i=1}^S \left[-h_{Repellant} \exp \left(-w_{Repellant} \sum_{m=1}^P (\theta_m - \theta_m^i)^2 \right) \right] \quad (2.15)
\end{aligned}$$

Onde

$J_{cc}(\theta, P(j, k, l))$ é o valor da função objetivo a ser adicionado ao valor atual da função objetivo; S é o total de bactérias; P é o número de variáveis a ser otimizada que estão presentes em cada bactéria da colônia; $\theta = [\theta_1, \theta_2, \dots, \theta_P]^T$ denota um ponto no espaço P -dimensional do espaço de busca; $d_{attract}$ é a profundidade do atrativo liberado; $w_{attract}$ é o valor da largura do sinal de atração; $h_{Repellant}$ é a altura do efeito de repelente e $w_{Repellant}$ é o valor a largura o repelente.

Reprodução: Depois de N_c (Número máximo de etapas de quimiotaxia) iterações da quimiotaxia, os passos da reprodução devem ser executados. Os valores dos *fitness*, ou aptidões, da colônia são armazenados em forma crescente. Então as bactérias menos saudáveis, com *fitness* baixos, eventualmente morrem e as bactérias restantes que são saudáveis, com os valores maiores de *fitness*, irão se dividir em duas bactérias idênticas e serão colocadas na mesma posição.

Dispersão-eliminação: Com o propósito de evitar pontos de ótimos locais, o processo de dispersão e eliminação é implementado depois de N_r (número máximo de etapa de reprodução) interações de reprodução. De acordo com uma probabilidade (P_{ed}), uma bactéria é escolhida para ser dispersa e movida para outra posição no ambiente, o número máximo de etapas de Dispersão-Eliminação é definido pelo parâmetro N_{ed} .

Os passos para implementar o BFO podem ser sintetizados como no pseudocódigo apresentado na Figura 2.3 (PASSINO, 2002; BOUSSAID et al, 2013; EL-ABD, 2012; TANG and WU, 2009).

Algoritmo BFO

```

1: Inicialize os parâmetros:  $n, S, Nc, Ns, Nre, Ned, Ped, Sr, C^i, \theta^i (i = 1, 2, \dots, S)$  ( $\theta^i$  re-
   apresenta a posição da  $i$ -ésima bactéria)
2:  $l \leftarrow 0$ 
3: repeat
4:    $k \leftarrow 0$ 
5:   repeat
6:      $j \leftarrow 0$ 
7:     repeat {etapa de quimiotaxia}
8:       for all  $i$  do
9:          $J_{last} \leftarrow J(i, j, k, l)$  {fitness da  $i$ -ésima bactéria}
10:        Gere um vetor unitário aleatório:  $\Delta^i$ 
11:         $\theta^i(j+1, k, l) \leftarrow \theta^i(j, k, l) + \Delta^i C^i$  {nova posição da bactéria}
12:        Calcule  $J(i, j+1, k, l)$  com  $\theta^i(j+1, k, l)$  {novo fitness da bactéria}
13:         $m \leftarrow 0$ 
14:        while  $m < Ns$  e  $J(i, j+1, k, l) < J_{last}$  do {swim}
15:           $m \leftarrow m + 1$ 
16:          Mova a bactéria mais um passo, de comprimento  $C^i$ , na direção  $\Delta^i$ 
17:          Calcule  $J(i, j+1, k, l)$  com o novo  $\theta^i(j+1, k, l)$ 
18:        end while
19:      end for
20:       $j \leftarrow j + 1$ 
21:    until  $j \geq Nc$ 
22:    for all  $i$  do {calcula a "saúde" das bactérias}
23:       $J_{health}^i \leftarrow \sum_{j=0}^{Nc} J(i, j, k, l)$ 
24:    end for
25:    Substitua as  $Sr$  bactérias com os maiores  $J_{health}$  por cópias das  $Sr$  bactérias com os
    menores  $J_{health}$  {reprodução}
26:     $k \leftarrow k + 1$ 
27:  until  $k \geq Nre$ 
28:  Com probabilidade  $Ped$ , atribua uma posição aleatória para cada bactéria {dispersão}
29:   $l \leftarrow l + 1$ 
30: until  $l \geq Ned$ 

```

Figura 2.9-Pseudo código do BFO (TEODORO et al, 2010)

2.5 Estado da arte de metaheurísticas populacionais aplicadas à Sistemas de Controle

Sistemas de controle são partes integrantes da sociedade moderna e podem ser vistos em vários exemplos ao nosso redor. A cada momento os sistemas se tornam mais complexos e a exigência por desempenhos cada vez mais ótimos aumenta. A busca por novas técnicas que possibilitem processos mais robustos e capazes de se adaptar as mudanças frequentes que ocorrem em seus ambientes vêm estimulando muitos pesquisadores ao estudo de métodos que atendam estes altos níveis de qualidade. Entre as técnicas usadas para a otimização em sistemas de controle, as metaheurísticas populacionais vêm sendo aplicadas em muitos trabalhos na literatura (SANTOS et al,

2011; TARIQUE et al, 2013; VIDAL et al, 2015; CHANG & SHIH, 2010; RAJINIKANTH & LATHA, 2012; LATHA et al, 2013). Estas técnicas oferecem desempenhos adequados para maioria dos problemas complexos encontrados, principalmente para aqueles em que para se chegar a uma solução aceitável se faz uso de meios empíricos.

A utilização destas metaheurísticas para o auxílio na tarefa de parametrização de diferentes tipos de controladores mostraram-se eficientes, agregando robustez e confiabilidade ao processo.

Em (PEREIRA, 2008) é realizado um estudo e a aplicação do PSO na otimização de controladores PID. É usada uma versão multiobjetivo do PSO, conhecida como MOPSO. O algoritmo é testado em um conjunto de funções padrões, seguindo-se da sintonia dos parâmetros do PID e a comparação com os métodos clássicos encontrados na literatura. São considerados dois objetivos de projetos para a otimização: a otimização da resposta a um sinal do tipo degrau aplicada como referência na entrada e a otimização da resposta em função de um sinal de perturbação degrau aplicada na saída.

Em (CHIOU et al, 2012) o PSO é usado como um método para o projeto de um controlador PID fuzzy ótimo para sistema de suspensão de automóveis. Já em (SINGH et al, 2015) é utilizado o *Bat Algorithm* (BA) para sintonizar os parâmetros de um controlador PID para realizar o controle de posição de um servo motor. A performance do sistema é avaliada em relação ao tempo de subida, tempo de acomodação, sobre sinal máximo e ao erro em regime estacionário. É usado o índice de desempenho ITSE (*Integral of Time multiplied by Square Error*) como medida a ser otimizada.

No trabalho de (GUPTA & GUPTA, 2014) é utilizado o algoritmo híbrido com *Bacterial Foraging Optimization* e o *Particle Swarm Optimization* para realizarem a sintonia dos parâmetros ótimos do controlador PID. É feita uma comparação de desempenho deste algoritmo com o uso os algoritmos BFO e PSO individualmente.

Outra utilização interessante de sintonia de controladores PID pode ser vista no trabalho de (RAJINIKANTH & LATHA, 2010) onde é utilizado o *Particle Swarm Optimization* para realizar a identificação e a parametrização de um controlador de um reator bioquímico instável. O PSO é usado para sintonizar os parâmetros de um controlador PID. E para validar o desempenho do controlador projetado é realizada uma comparação com o controle feito através do IMC (*Internal Model Controller*).

O trabalho de (IBRAHIM et al, 2014) usa os algoritmos PSO e BFO para determinar os parâmetros ótimos de um controlador PID para o controle da velocidade de motores CC. As técnicas propostas neste trabalho visam a otimização das características da resposta a uma excitação do tipo degrau.

Outras variações do PID também podem ser parametrizadas com o uso de metaheurísticas, por exemplo, em (CHAIB et al, 2015) é proposto um estabilizador de sistemas de energia, que é baseado na hibridização de um PID de ordem fracionária ($PI^\lambda D^\mu$). O problema de otimização do sistema é baseado em índices de desempenhos, tais como IAE (*Integral of Absolute Error*), ISE (*Integral of Squared Error*), ITAE (*Integral of Time multiplied by Absolute Error*) e o ITSE (*Integral of Time multiplied by Squared Error*). Para a otimização do problema é usado uma metaheurística baseada no comportamento de eco localização de morcegos, conhecido como *Bat Algorithm* (BA). Em (PATEL & KUMAR, 2015) é implementado um controlador chamado *Artificial Neuro Fuzzy Logic* PID (ANFLC) para realizar o controle de sistemas não lineares. É utilizado uma variação híbrida entre o BFO e o PSO chamada de BF-PSO para otimizar os parâmetros do sistema. Já em (KANG et al, 2014) é apresentado o uso o PSO para a inicializar os pesos de um controlador PID neural adaptativo, de tal forma a aumentar a velocidade no treinamento e ao mesmo tempo evitar possíveis convergências prematuras para ótimos locais.

Em (BERTACHI, 2014) apresenta-se a aplicação de métodos de otimização dos parâmetros de um controlador não linear de estrutura variável de variância mínima generalizada. Para realizar a otimização é usado duas metaheurísticas populacionais: o *Particle Swarm Optimization* (PSO) e o *Ant Colony Optimization* (ACO). Estes algoritmos são aplicados para se obter os valores otimizados dos parâmetros.

Em (MANDAL et al, 2015) é feito uma aplicação do BFO para realizar o projeto ótimo em tempo real de um controlador Fuzzy que é usado para controle de sistemas hidráulicos.

(SAYAED et al, 2015) propõem uma versão híbrida do PSO que utiliza um operador de mutação baseado em funções Gaussianas e de Cauchy para melhorar o desempenho do PSO padrão. O algoritmo chamado de *Hybrid Jump* PSO (HJPSO) é aplicado na sintonia dos ganhos de um controlador PI para o controle da temperatura em uma caldeira.

As metaheurísticas não são usadas somente na parametrização de controladores mas também são usadas em tarefas como modelagem e identificação de sistemas

dinâmicos (PARMAR et al, 2007; SATAKSHI et al, 2005; STEFOPOULOS et al, 2005; PILLAY et al, 1997; NANGSUE et al, 1999). Como por exemplo, em (Ferreira et al, 2011) é apresentada uma metodologia utilizando AGs para realizar a redução de modelos dinâmicos. A convergência é obtida com a minimização do erro entre a resposta do modelo original e do modelo reduzido, sendo proposto manter as características dinâmicas do sistema original. No trabalho de (GAO et al, 2009) é proposto um *framework* para a identificação de sistemas de potência baseado no uso de um Algoritmo Genético. O AG usado possui uma função objetivo logarítmica e um espaço de busca adaptável. O *framework* é aplicável à maioria dos sistemas de potências, necessitando somente de alguns ajustes no modelo de identificação usado. O sistema é avaliado na identificação dos parâmetros de uma turbina.

3 CONTROLADORES PID

3.1 Introdução

O controle automático tem influenciado nas últimas décadas para o avanço constante da engenharia e da ciência. Desde a primeira revolução industrial o homem moderno passou a conviver de uma maneira mais frequente com sistemas automáticos, e isso contribuiu para o aumento das produções industriais e o bem estar da sociedade. Atualmente, existe um número variado de sistemas que fazem o uso de controle automático, desde produções em sistemas de pequeno porte até sistemas de grandes dimensões. Por exemplo, a produção de papel na indústria de celulose possui várias etapas que são executadas por sistemas automáticos, assim como a indústria aeroespacial e a automobilística fazem uso desse tipo de sistema. Considerando que os avanços na teoria e na prática do controle automático propiciam meios para se atingir desempenhos ótimos de sistemas dinâmicos além de melhorias de produtividade, os estudos na área de sistemas de controle tornam-se cada vez mais importante, seja para o desenvolvimento de novas técnicas ou para melhoria das já existentes (OGATA, 2003).

Uma etapa importante para ser possível a realização do controle de um sistema é o estudo sobre as características do mesmo. Esta etapa é conhecida como modelagem, e consiste na obtenção de um modelo que descreva o comportamento de um sistema, de tal forma que seja possível estabelecer uma relação de causa e efeito entre as variáveis de entrada e saída. Quando se tem conhecimento da natureza do processo e as leis físicas que o regem, a modelagem é feita por uma técnica conhecida na literatura como modelagem em caixa branca, ou ainda modelagem conceitual (AGUIRRE, 2007). Uma vez de posse desse modelo do sistema é possível verificar se o desempenho dele está de acordo com o esperado. Quando a resposta do sistema não atende os requisitos necessários é preciso realizar o controle, e neste caso se dispõe de diferentes técnicas analíticas na literatura para o projeto de controladores.

No entanto, nem sempre é viável modelar um sistema partindo do equacionamento dos fenômenos envolvidos, seja pela falta de conhecimento sobre as leis que envolvem o sistema ou pela presença de não-linearidades. A solução então passa a ser usar a identificação de sistemas, que é uma área da modelagem matemática que estuda técnica alternativa à modelagem caixa branca. Uma das características dessas técnicas é que pouco ou nenhum conhecimento prévio do sistema é necessário e,

consequentemente tais métodos são também referenciados como modelagem (ou identificação) caixa preta ou modelagem empírica (AGUIRRE, 2007).

Os controladores PID são controladores clássicos baseados em três ações de controle: Proporcional, Integral e Derivativa, e apresentam bom desempenho em uma grande faixa de controle de diferentes sistemas. A partir da sintonia de seus parâmetros é possível obter respostas com boas características no transitório e de estado estacionário da maioria dos sistemas dinâmicos. Devido sua simplicidade conceitual, e o número pequeno de parâmetros, estes controladores estão presentes em grande parte dos sistemas automáticos. É importante destacar que grande parte dos controladores encontrados atualmente nas malhas industriais opera com o uso de PID ou PID modificados.

Tendo em vista que a maioria dos controladores PID são ajustados no local de uso, têm sido propostos na literatura diferentes tipos de técnicas de sintonia de seus parâmetros. A utilização dessas regras de sintonia tem tornado possível o ajuste suave e preciso dos parâmetros do controlador PID. Quando se desconhece a dinâmica dos sistemas com precisão, um dos métodos mais utilizados da literatura são os métodos propostos por Ziegler-Nichols. Estas regras são heurísticas que se propõe a determinar os valores dos ganhos dos controladores clássicos com base nas características das respostas transitórias de cada planta a ser controlada. Uma das vantagens destes métodos é que são aplicáveis à maioria dos sistemas encontrados na indústria e são relativamente simples de ser implementados. Porém um dos problemas destes métodos é que uma vez que seus ganhos são escolhidos, seu funcionamento ocorrerá de maneira satisfatória apenas se não ocorrerem mudanças significativas no processo em que o controlador está inserido, caso ocorra uma grande variação no comportamento da planta, o controlador sintonizado previamente pode não funcionar corretamente e até tornar o sistema em malha fechada instável (CHEN, 1995; DORF & BISHOP, 2001; NISE, 2002; OGATA, 2003).

Diante destas limitações surge a necessidade de outras técnicas para realizar a sintonia dos parâmetros dos controladores. Entre as diferentes técnicas propostas na literatura destacam-se atualmente os métodos baseados em Inteligência Computacional (IC). Estes métodos agregam a capacidade de adaptabilidade aos sistemas e flexibilizam o uso para sistemas com diferentes características dinâmicas. Entre os algoritmos de IC as metaheurísticas populacionais vêm se mostrando eficientes na tarefa de sintonia de diferentes tipos de controladores.

3.2 Ações de controle PID

Um controlador automático compara o valor real da grandeza de saída do processo com a grandeza de referência (valor desejado), determina o desvio e produz um sinal de controle que tem o propósito de reduzir o desvio para zero ou a um valor pequeno. A maneira pela qual o controlador produz este sinal é conhecida como ação de controle. Como mencionado anteriormente, o controlador PID é formado por três ações de controle baseadas no ganho proporcional, integral e derivativo do erro (OGATA,2003; NISE,2002).

Ação de controle proporcional (P): Para um controlador com ação de controle proporcional, a relação entre o sinal do controlador $u(t)$ e o sinal de erro $e(t)$ é dada por:

$$u(t) = K_p e(t) \quad (3.1)$$

Com função de transferência dada por:

$$C(s) = \frac{U(s)}{E(s)} = K_p \quad (3.2)$$

Sendo K_p a constante proporcional.

Esta ação de controle sobre um processo, cuja função de transferência não tenha um integrador, causa um erro em regime estacionário, ou erro residual, na resposta a uma excitação do tipo degrau (OGATA, 2003).

Ação de controle integral (I): No controle I a ação de controle $u(t)$ é proporcional à integral do sinal de erro $e(t)$ de acordo com :

$$u(t) = \frac{1}{T_i} \int e(t) dt \quad (3.3)$$

Com função de transferência dada por:

$$C(s) = \frac{U(s)}{E(s)} = \frac{1}{T_i s} \quad (3.4)$$

A constante T_i é chamada de tempo integral ou *reset-time*.

O controle integral atua melhorando a precisão do sistema em regime permanente, permitindo zerar o erro em regime estacionário, posicionando um polo em malha aberta na origem, aumentando, assim, o tipo do sistema em uma unidade, no entanto, esta ação tende a tornar a resposta mais lenta e pode até mesmo instabilizar o sistema dependendo de sua ordem.

Ação de controle Proporcional-Integral (PI): A ação de controle $u(t)$ de um controlador PI é proporcional ao erro $e(t)$ e à integral do erro:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt \right] \quad (3.5)$$

Com função de transferência dada por:

$$C(s) = \frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} \right] \quad (3.6)$$

A ação de controle Integral somada à ação Proporcional elimina o erro residual, aumentando em uma unidade o tipo do sistema, conseqüentemente o que pode tornar o sistema de controle instável para valores grandes de K_p , se as raízes da equação característica possuírem parte real positiva. A ação de controle proporcional tende a estabilizar o sistema, enquanto que a ação de controle integral tende a eliminar o erro em regime estacionário (CHEN, 1995; OGATA, 2003).

Ação de controle Proporcional-Derivativa (PD): Neste caso o sinal de controle atuante em um sistema é proporcional ao erro e à derivada do erro. A ação de controle é dada por:

$$u(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} \right] \quad (3.7)$$

E a função de transferência dada por:

$$C(s) = \frac{U(s)}{E(s)} = K_p [1 + T_d s] \quad (3.8)$$

Onde T_d é a constante de tempo derivativo.

Ação de controle derivativa, quando adicionada a um controlador proporcional, propicia um meio de obter um controlador com alta sensibilidade. Uma vantagem em se usar a ação de controle derivativa é que ela responde à taxa de variação do erro atuante e pode produzir uma correção significativa antes do valor do erro atuante tornar-se demasiadamente grande. Embora o controle derivativo não afete diretamente o erro em regime estacionário, ele introduz amortecimento no sistema e, portanto, permite o uso de um valor maior do ganho K_p , o que resulta em uma melhoria na precisão em regime estacionário (OGATA, 2003; NISE, 2002).

Ação de controle Proporcional-Integral-Derivativa (PID): A combinação das ações de controle Proporcional, Integral e Derivativa formam o controle PID que possui as vantagens de cada uma das três ações de controle individuais, atuando no sinal de erro do sistema. É bastante utilizado em indústrias, pois normalmente apresentam um bom desempenho no controle de sistemas de primeira e segunda ordem ou superior. Sua ação de controle e função de transferência são dada por :

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (3.9)$$

$$C(s) = \frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} + T_d s \right] \quad (3.10)$$

3.3 Sintonia de Controladores PID

A Figura 3.1 apresenta um exemplo de um sistema em malha fechada com o uso de controlador PID para o controle de um processo. Quando se tem um modelo matemático do processo é possível aplicar várias técnicas visando à determinação dos

parâmetros do controlador que atendam às especificações de regime transitório e estacionário. Nos casos, no entanto, em que o processo seja tão complicado que seu modelo matemático não possa ser obtido com facilidade, então a abordagem analítica para se projetar um controlador PID deixa de ser viável. Nestes casos deve-se fazer uso das técnicas experimentais de sintonia dos parâmetros dos controladores PID (OGATA, 2003).

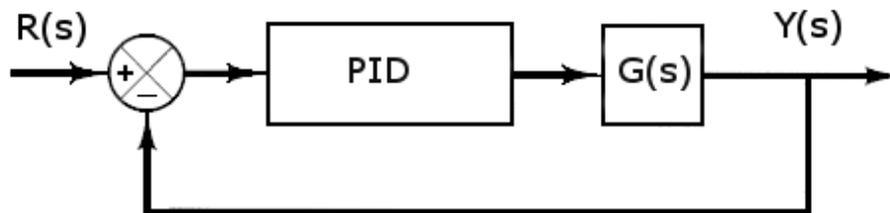


Figura 3.1-Diagrama de blocos de sistema de controle em malha fechada (OGATA, 2003)

A realização de um projeto de controle visa satisfazer algumas especificações impostas sobre as características de regime transitório e estacionário da resposta do sistema que está sendo controlado. Geralmente, estas especificações recaem sobre os valores do tempo de subida (T_r), tempo de acomodação (T_s), valor máximo de ultrapassagem (M_p) e um valor máximo de erro estacionário aceitável.

De acordo com as especificações desejadas no projeto, como o valor de M_p , T_s e T_r , é possível gerar uma resposta a partir de um modelo com as características desejadas. Este modelo é conhecido como função de transferência desejada ($G_d(s)$), e sua ordem é determinada de acordo com as características das especificações de projeto.

Considerando uma aproximação para um sistema de segunda ordem, quando a resposta desejada apresentar um valor de $M_p > 0$, tem-se um sistema subamortecido e o valor de M_p pode ser utilizado para calcular o valor do coeficiente de amortecimento desejado (ξ) através de:

$$\xi = \frac{-\ln(M_p)}{\sqrt{\pi^2 + \ln^2(M_p)}} \quad (3.11)$$

O valor da frequência natural não amortecida (W_n) pode ser calculado usando o valor de ξ obtido pela Equação (3.11) e o tempo de acomodação especificado de acordo com:

$$W_n = \frac{4}{\xi T_s} \quad (3.12)$$

Quando a especificação de projeto tiver um valor de $M_p = 0$, tem-se uma resposta desejada com características de sistema de primeira ordem, o valor da constante de tempo da função de transferência em malha fechada desejada (T) pode ser obtido usando o valor do tempo de subida desejado (T_r) a partir de:

$$T = \frac{T_r}{2.2} \quad (3.13)$$

Deste modo, para ambos os casos, 1ª e 2ª ordem, a função de transferência em malha fechada desejada fica determinada e são representadas respectivamente por:

$$G_d(s) = \frac{K}{Ts + 1} \quad (3.14)$$

$$G_d(s) = \frac{KW_n^2}{s^2 + 2\xi W_n s + W_n^2} \quad (3.15)$$

Com a função de transferência desejada $G_d(s)$ determinada é então possível gerar a resposta $Y_d(s)$ que atende as especificações estabelecidas no projeto para que ela possa ser usada como referência na sintonia do PID.

3.3.1 Métodos Tradicionais para ajuste dos Parâmetros do Controlador PID

Os métodos heurísticos propostos por Ziegler-Nichols se propõem a alcançar respostas com características transitórias e de regime estacionário aceitáveis para a entrada de um sinal do tipo degrau, através da sintonia dos parâmetros do controlador com base na resposta experimental do sistema. Existem dois métodos designados como métodos de Ziegler-Nichols. No primeiro método se obtém experimentalmente a resposta do processo a controlar a uma excitação em degrau unitário. Quando o processo a controlar não envolve integradores nem pólos dominantes complexos-conjugados, a curva de resposta ao degrau unitário pode se assemelhar a uma curva em “S”. Então neste caso o método pode ser aplicado e através da caracterização da curva em “S” em duas constantes, tempo de retardo (L) e constante de tempo (τ), é feita a parametrização de K_p , T_i e T_d em função de L e τ .

No segundo método de Ziegler-Nichols é determinado o valor de K_p que resulta em estabilidade marginal quando se utiliza unicamente a ação de controle proporcional com o sistema em malha fechada. Para isso é feita a variação do ganho K_p de 0 a um valor crítico de ganho (K_{cr}) para o qual o sinal de saída apresenta oscilações. Se o sinal de saída não apresentar oscilações, para quaisquer que forem os valores de K_p , então o método não se aplica. Com o ganho (K_{cr}) determinado, calcula-se o período de oscilação crítico (P_{cr}) correspondente e então a sintonia dos parâmetros é feita usando fórmulas em função destes parâmetros.

Outra forma de se realizar o projeto de controladores PID é fazendo uso do método do Lugar Geométrico das Raízes (LGR), que consiste de um método gráfico para determinar a localização de todos os pólos a malha fechada a partir do conhecimento da localização dos pólos e zeros de malha aberta à medida que o valor de um parâmetro (usualmente o ganho) é variado de zero ao infinito. O LGR conduz a uma indicação clara dos efeitos do ajuste dos parâmetros. Através do LGR é possível perceber se o sistema pode alcançar o desempenho desejado e para qual valor de ganho isso ocorre. Além disso, pode acontecer de que o sistema não possa atingir o desempenho desejado unicamente pelo ajuste do valor do ganho ou ainda que para determinados valores o sistema possa ficar instável. Para estes casos os lugares das raízes devem ser modificados inserindo-se um controlador. De posse do conhecimento dos efeitos da adição de pólos e zeros sobre o LGR, pode-se determinar de forma adequada a localização dos pólos e zeros do controlador que produzirão o condicionamento da forma do lugar das raízes como desejado. Em essência, no projeto pelo LGR os lugares geométricos das raízes dos sistemas são submetidos a uma mudança de forma através do uso de um controlador de modo que um par de pólos a malha fechada dominantes possa ser alocado na posição desejada.

A abordagem em resposta em frequência permite uma maneira fácil de obter a função de transferência de forma experimental de componentes complicados e esta abordagem possui a vantagem de se projetar sistemas de modo que os efeitos de ruídos indesejáveis sejam desprezíveis e que a análise e o projeto possa ser estendido a determinadas classes de sistemas não-lineares. Neste método submete-se o sistema a um sinal de entrada senoidal e então varia-se a frequência deste sinal ao longo de uma faixa de interesse e estuda-se a resposta resultante. Nesta abordagem de projeto do sistema o desempenho em regime transitório é especificado de forma indireta. Isto é, o

desempenho da resposta transitória é expresso em termos de margem de fase, margem de ganho, magnitude do pico de ressonância (que fornecem uma estimativa do amortecimento do sistema), da frequência de cruzamento de ganho, da frequência de ressonância, da banda passante (que fornecessem uma estimativa razoável da velocidade da resposta transitória) e das constantes de erro estático (que fornecem a exatidão do sistema em regime permanente).

3.3.2 Índices de Desempenhos

A teoria de controle moderna supõe que o engenheiro de sistemas possa especificar quantitativamente o desempenho requerido do sistema. Por conseguinte, um índice de desempenho pode ser calculado e usado para se medir o desempenho do sistema. A medida quantitativa do desempenho de um sistema é necessária para a operação de sistemas de controle e para o projeto ótimo de sistemas (DORF & BISHOP, 2001).

Um índice de desempenho é uma medida quantitativa do desempenho de um sistema e é escolhido de modo que a ênfase seja dada as especificações de sistemas importantes. Desde modo um sistema é considerado um sistema de controle ótimo quando os parâmetros do sistema são ajustados de modo que o índice alcance um valor extremo, comumente um valor mínimo (DORF & BISHOP, 2001; D'AZZO & HOUPS, 1975).

a) *Índice ISE (Integral of Square Error)*

Este índice de desempenho é definido pela Equação (3.16) e calcula a integral do quadro do erro em um tempo T (DORF & BISHOP, 2001).

$$ISE = \int_0^T e^2(t)dt \quad (3.16)$$

b) *Índice IAE (Integral of Absolute Error)*

O índice IAE é representado pela Equação (3.17), e pelo fato de usar o valor absoluto do erro, a integral aumenta para erros positivos ou negativos. Isto acarreta em

um bom sistema sub amortecido quando este índice é minimizado (D'AZZO & HOUPS, 1975).

$$IAE = \int_0^T |e(t)| dt \quad (3.17)$$

c) *Índice ITAE (Integral of Time multiplied by Absolute Error)*

O índice ITAE é calculado usando a Equação (3.18). Este índice é usado para reduzir a contribuição de grandes erros iniciais no valor da integral de desempenho, bem como enfatizar erros que acontecem mais tarde na resposta (D'AZZO & HOUPS, 1975).

$$ITAE = \int_0^T t|e(t)| dt \quad (3.18)$$

d) *Índice ITSE (Integral of Time multiplied by Square Error)*

Este índice é calculado usando a Equação (3.19) e também conduz a um sistema com bom desempenho.

$$ITSE = \int_0^T t e^2(t) dt \quad (3.19)$$

4 SINTONIA DOS PARÂMETROS DE CONTROLADORES PID USANDO METAHEURÍSTICA POPULACIONAL

4.1 Introdução

Como apresentado no estado da arte de metaheurísticas aplicadas aos sistemas de controle, muitos trabalhos vêm sendo desenvolvidos na área de sintonia de controladores PID, esse avanço nas pesquisas desta área se deve principalmente à capacidade das metaheurísticas em percorrer espaços de buscas extensos de uma forma robusta, evitando pontos de ótimos locais e gerando soluções aceitáveis em tempos hábeis nos mais diversos tipos de problemas.

Neste capítulo, será apresentada, de uma maneira geral, como as metaheurísticas populacionais podem ser utilizadas para o processo de sintonia dos controladores PID. Em seguida, as seguintes metaheurísticas populacionais serão utilizadas para a sintonia de controladores PID: Algoritmo Genético (AG), Particle Swarm Optimization (PSO), Bacterial Foraging Optimization (BFO) e o Algoritmo Genético no Modelo de Ilhas (AGMI).

O objetivo é realizar um estudo comparativo de desempenho entre as 4 metaheurísticas populacionais para a sintonia de parâmetros de controladores PID em casos de sistemas de primeira ordem, sistemas de segunda e em sistemas de ordem superior, onde serão avaliadas as características das respostas geradas em relação às características esperadas no projeto. Serão avaliados também os índices de desempenhos: ITAE (*Integral of Time multiply by Absolute Error*), ISE (*Integral of the Square Error*), IAE (*Integral of the Absolute Error*) e ITSE (*Integral of Time multiply by the Square Error*) gerados pelo uso dos controladores projetados com cada algoritmo, além do tempo de processamento de cada algoritmo no processo de busca pelos parâmetros do controlador.

4.2 Metaheurística populacional para a sintonia dos parâmetros de um controlador PID

A sintonia do controlador PID é basicamente um problema de busca em um espaço tridimensional, onde cada ponto do espaço corresponde a diferentes escolhas de parâmetros do controlador. Através da escolha de pontos diferentes podem-se produzir, por exemplo, diferentes respostas para uma entrada do tipo degrau. Um controlador PID então pode ser determinado movendo-se neste espaço de busca fazendo uso de alguma heurística (DORF, 2001).

As metaheurísticas populacionais podem então ser utilizadas para realizar a busca pelos melhores pontos no espaço de soluções, de forma a ser minimizado o erro entre a resposta dos sistemas gerada pelo uso de cada um dos controladores representados pelos indivíduos da população e a resposta desejada de projeto. O objetivo principal é obter uma resposta que atenda aos requisitos de projeto, ou no caso de requisitos conflitantes, se obterem respostas aceitáveis.

4.2.1 Representação dos indivíduos da população

Para ser realizada a sintonia dos parâmetros do controlador PID usando uma metaheurística populacional, cada indivíduo que compõe a população deve representar uma combinação de valores do ganho proporcional, e dos tempos integral e derivativo, ou seja, um ponto no espaço de busca. Neste caso, cada indivíduo n da população é representado por um vetor que contém os parâmetros do controlador PID, de acordo com:

$$\Theta_n = [K_p \ T_i \ T_d] \quad (4.1)$$

Cada resposta gerada com o uso do controlador representado por cada indivíduo da população deve ser avaliada para que seja usada uma heurística para guiar a população em direção às regiões onde estão contidas as melhores soluções, ou seja, aquelas que atendem aos requisitos de projeto. A forma como cada metaheurística usa as informações dos melhores indivíduos para “guiar” a população para as regiões promissoras é implementada de uma forma particular através de analogias a processos encontrados na natureza, como a teoria da evolução das espécies e a seleção natural ou a partir da emulação das interações de grupos específicos de seres vivos.

4.2.2 Estrutura de avaliação dos indivíduos da população

A Figura 4.1 apresenta a estrutura em diagrama de blocos que pode ser usada para avaliar a qualidade da resposta gerada pelos parâmetros do PID representados por cada indivíduo. A estrutura é formada pelo bloco que contém a função de transferência de malha fechada desejada $G_d(s)$ e o bloco de avaliação. O bloco de avaliação é composto pelo sistema em malha fechada com o controlador PID, $C_n(s)$, a planta $G(s)$ mais o bloco representativo do algoritmo de sintonia utilizado (metaheurística populacional). A função de transferência de malha fechada desejada $G_d(s)$ é formada a partir das especificações projeto, quais sejam: M_p , T_s e T_r .

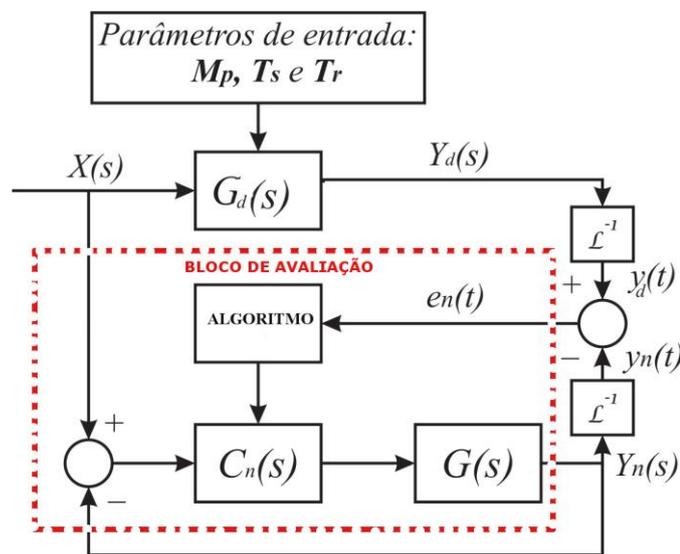


Figura 4.1-Estrutura usada para a sintonia do PID com metaheurística

Um sinal $X(s)$ é aplicado à função em malha fechada desejada gerando o sinal $Y_d(s)$. O mesmo sinal de entrada é aplicado ao bloco de avaliação, sendo que os parâmetros do controlador $C_n(s)$ serão determinados pelo indivíduo Θ_n . Para cada Θ_n será gerada a resposta $Y_n(s)$, com $n=1,2,\dots,N$, onde N é o tamanho da população. Cada sinal $Y_n(s)$ será então convertido para o domínio do tempo juntamente com o sinal $Y_d(s)$. O erro entre a resposta do indivíduo n da população e a resposta limitante de projeto será então computado conforme:

$$e_n(k) = y_n(k) - y_d(k) \quad (4.2)$$

com $k = 1, 2, \dots, m$, sendo o número de elementos do vetor e_n .

Em seguida cada vetor de erros e_n de cada indivíduo da população será repassado ao algoritmo de sintonia para que possa ser usado no cálculo da avaliação de cada indivíduo. A partir do vetor de erro calculado pela Equação (4.2), será calculado o erro médio quadrático para cada indivíduo conforme:

$$\bar{e}_n = \frac{\sum_{k=1}^m [e_n(k)]^2}{m} \quad (4.3)$$

Por fim será possível avaliar cada indivíduo usando a Função Objetivo:

$$F_n = \frac{1}{0,001 + \bar{e}_n} \quad (4.4)$$

Como pode ser observado da função objetivo (avaliação), os indivíduos que contiverem os menores valores de erros serão os com maior valor de avaliação, e assim os mais próximos da resposta limitante de projeto.

4.3 AG no Modelo de Ilhas (AGMI) para a sintonia do PID

Em um controlador PID as ações de controle proporcional, integral e derivativa são combinadas para que possam ser produzidas respostas com valores sem erro de regime estacionário, a partir da ação integral, e para que possa ser gerada uma resposta com uma melhor característica no transitório e estável é usada a ação de controle derivativa (OGATA, 2003). Sendo assim, além do AG tradicional será utilizado também neste trabalho, para a sintonia dos parâmetros do controlador PID o algoritmo genético no modelo de ilhas (AGMI).

Um AG que possua uma população com um tamanho de N indivíduos e execute por G gerações irá manipular um total de $N \times G$ indivíduos ao longo de sua excursão. Em casos onde o custo para se testar a qualidade de um único indivíduo seja um processo demorado, o uso de uma população muito grande torna-se indesejável. Uma variante na literatura conhecida como μ AG utiliza um tamanho reduzido de indivíduos em sua população, e para evitar convergências prematuras que podem ocorrer devido a

falta de diversidade genética é empregada uma estratégia frequente de reinicialização da população no decorrer de poucas gerações. Além disso, é usada alguma heurística para inicializar a população próxima das regiões onde se tem mais chance de se encontrar os melhores indivíduos (ZALZALA&FLEMING, 1997; KRISHNAKUMAR, 1989). Considerando então a importância do desenvolvimento de AGs baseados em heurísticas para inicialização das populações próximas das regiões onde se tem mais chance de ser encontrado os melhores indivíduos, em (VIDAL et al, 2015) propõe-se o AGMI baseado na utilização de três Algoritmos Genéticos, denominados de AG1, AG2 e AG3, para realizar a busca dos parâmetros do controlador PID. O AG1 evolui uma população aleatória ao longo de gerações calculando o erro entre as respostas dos indivíduos e a resposta desejada de projeto somente no período transitório. Enquanto que o AG2 evolui uma população criada aleatoriamente comparando o erro somente das respostas em regime permanente. Assim os indivíduos do AG1 são levados para próximo da região que contém os melhores valores dos ganhos que diminuem o erro no transitório, ou seja, com uma maior influência para o ganho derivativo. De forma análoga o AG2 faz o mesmo para o regime estacionário, com maior influência para os ganhos proporcional e integral. Os dois primeiros AGs possuem uma população pequena e um número de geração baixo a fim de que não ocorra a completa convergência dos indivíduos. O AG3 recebe as duas populações anteriores e evolui as mesmas ao longo de sua geração calculando o erro em todo o tempo de resposta. Assim o AG3 começa a busca com um bom ponto de partida, uma vez que irá iniciar a busca em uma região com boas informações genéticas dos parâmetros.

4.4 Aplicação de Metaheurística populacional para a sintonia dos parâmetros do PID

4.4.1 Considerações Iniciais

Neste tópico serão utilizadas as metaheurísticas populacionais para realizar a sintonia dos parâmetros de controladores PID para diferentes sistemas. Os sistemas usados foram simulados e apresentam diferentes características, afim de que se possa analisar o desempenho dos algoritmos para os diferentes requisitos de projeto. Para a realização das sintonias para todos os sistemas foram realizados 20 ensaios para cada algoritmo utilizado.

A Tabela 4.1 apresenta os valores dos parâmetros usados no AG. A Tabela 4.2 apresenta os valores utilizados no BFO. Os valores dos parâmetros utilizado no AGMI estão apresentados na Tabela 4.3. A Tabela 4.4 contém os valores dos parâmetros usados no PSO. Os algoritmos foram parametrizados com os valores encontrados na literatura e através de forma heurística.

Tabela 4.1-Parâmetros do AG

População	Taxa de mutação	Geração	Seleção	Elitismo
100	10%	100	Torneio	sim

Tabela 4.2-Parâmetros do BFO

Colônia	Nc	Nr	Ned	Ped
40	50	4	4	2%

Tabela 4.3-Parâmetros do AGMI

	População	Taxa de mutação	Geração
AG1	20	10%	20
AG2	20	10%	20
AG3	40	10%	30

Tabela 4.4-Parâmetros do PSO

Enxame	W	C1 e C2	Interação máxima
40	0.4	2	100

4.4.2 Sistemas de primeira ordem

Para avaliar o desempenho dos algoritmos na sintonia de controladores aplicados em sistemas de primeira ordem foram utilizados três sistemas diferentes com

características específicas de regime transitório e permanente, e para cada um deles foram utilizadas especificações diferentes de projetos.

a) **Caso 1: Sistema rápido com erro de regime permanente**

Para este caso foi utilizado o sistema representado pela função de transferência mostrada na Equação 4.5. A Figura 4.2 representa o sinal de resposta deste sistema a uma entrada do tipo degrau unitário.

$$G(s) = \frac{2,99}{0,5s + 1} \quad (4.5)$$

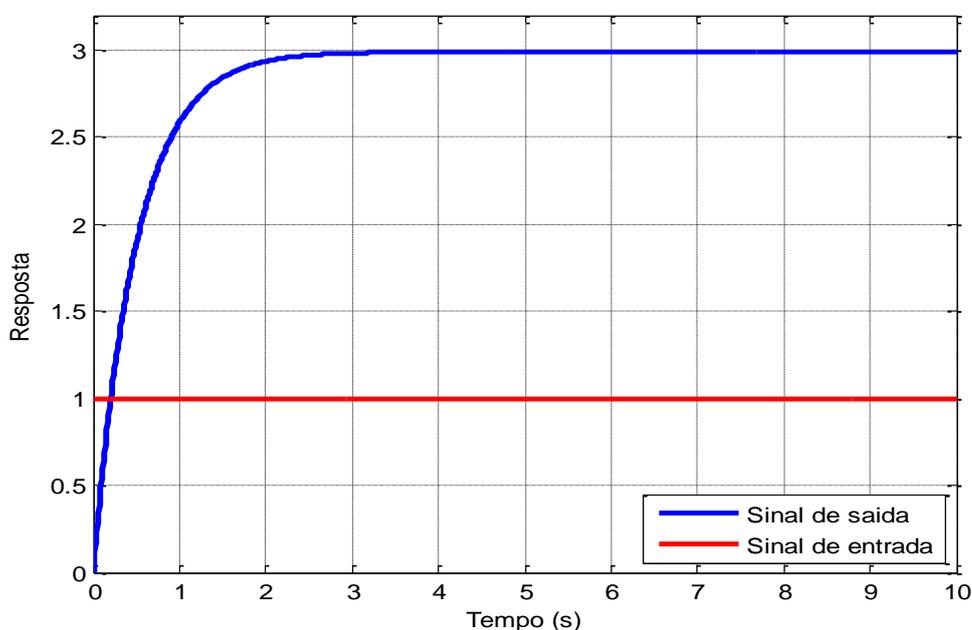


Figura 4.2-Resposta do sistema do caso 1 para uma entrada degrau unitário

O sistema apresenta uma resposta com boa característica em regime transitório, respondendo de maneira rápida, porém apresentando erro em regime permanente. Para este sistema é proposta a sintonia de um controlador PID que mantenha a constante de tempo do sistema e diminua o erro de estado estacionário. As exigências de projetos passadas para os algoritmos estão apresentadas na Tabela 4.5. A Tabela 4.5 também contém os valores alcançados para o sistema com o uso de cada controlador sintonizado pelos algoritmos AG, AGMI, BFO e o PSO.

A Tabela 4.6 apresenta os valores de cada parâmetro do PID sintonizado pelos algoritmos, onde é possível verificar também o comportamento médio obtido nos ensaios realizados com o uso de cada algoritmo. A Figura 4.3 apresenta as respostas do sistema em malha fechada com o uso de cada PID sintonizado pelos algoritmos apresentados na Tabela 4.6.

Tabela 4.5-Especificações de projeto e valores obtidos pelos algoritmos para o caso 1

Parâmetros	AG	AGMI	BFO	PSO	Projeto
M_p	0%	0%	0%	0%	0%
M_p médio	0%	0%	0%	0%	-
M_p Desvio	0%	0%	0%	0%	-
T_r (s)	1,10	1,10	1,00	1,097	1,11
T_r médio	1,10	1,09	0,98	1,097	-
T_r Desvio	0,01	0,01	0,02	0,00	-
T_s (s)	1,98	1,98	1,81	1,97	2,00
T_s médio	1,99	1,97	1,76	1,97	-
T_s Desvio	0,05	0,02	0,04	0,00	-
Tempo médio de busca (s)	114,18	83,9	153,77	69,98	-

Tabela 4.6-Parâmetros do PID sintonizado por cada algoritmo para o caso 1

Parâmetros	AG	AGMI	BFO	PSO
K_p	0,3522	0,3395	0,4691	0,3314
K_p médio	0,3463	0,3687	0,4514	0,3314
K_p Desvio	0,0114	0,0998	0,2487	0,0000
T_i	0,5165	0,5071	0,5640	0,5000
T_i médio	0,5137	0,5202	0,5168	0,5000
T_i Desvio	0,0103	0,0620	0,1591	0,0000
T_d	0,0002	0,0005	0,0000	0,0000
T_d médio	0,0039	0,0003	0,0002	0,0000
T_d Desvio	0,0043	0,0002	0,0004	0,0000

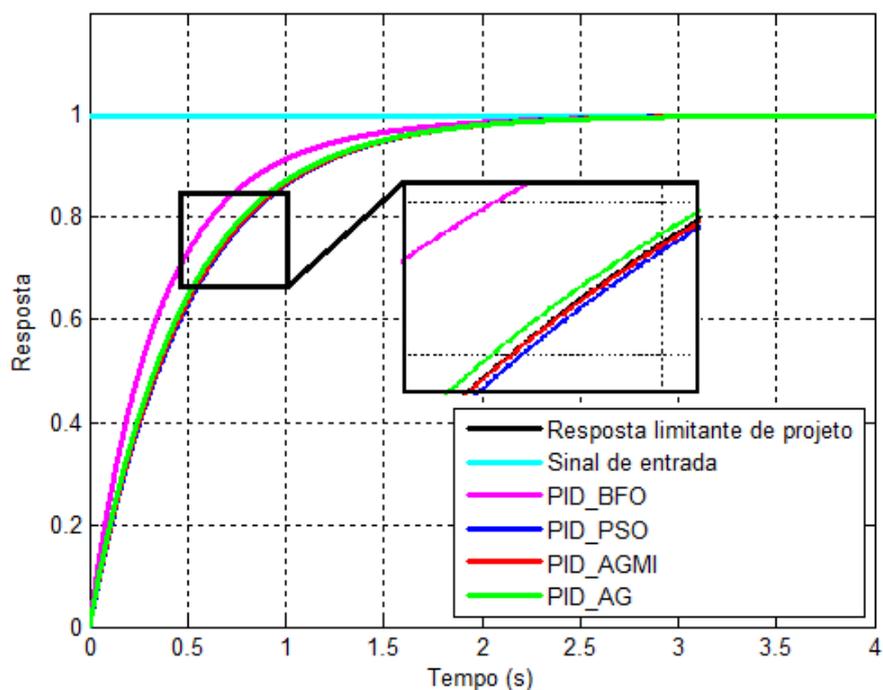


Figura 4.3-Resposta do sistema com o uso dos controladores sintonizados para o caso 1

Com observado na Figura 4.3 e nos valores da Tabela 4.5, todos os controladores sintonizados atendem as especificações estabelecidas no projeto, ou seja, produziram respostas com valores abaixo da resposta limitante de projeto. Sendo que as respostas para o AGMI, AG e o PSO ficaram mais próxima da resposta limitante e a resposta produzida com o uso do PID sintonizado pelo BFO apresentou resposta mais rápida.

A Tabela 4.7 apresenta os índices de desempenho obtidos com o uso de cada controlador onde pode ser observado que todos algoritmos apresentam ótimos valores, ou seja, apresentam índices com valores pequenos, sendo que o BFO apresenta melhor valor entre todos os algoritmos.

Tabela 4.7-Valores dos índices de desempenho para caso 1

Índices	AG	AGMI	BFO	PSO
IAE	0,4902	0,4993	0,4017	0,5044
ISE	0,2412	0,2480	0,1877	0,2523
ITAE	0,2472	0,2520	0,1854	0,2538
ITSE	0,0597	0,0619	0,0377	0,0637

A Figura 4.4 mostra os respectivos sinais de controle gerados com o uso de cada controlador todos menores que um o que pode ser considerado baixo. Como pode ser observado pela Tabela 4.5 o PSO possui o menor tempo de busca em relação aos outros

algoritmos, além de apresentar um desvio padrão igual a zero mostrando-se mais robusto entre os quatro algoritmos.

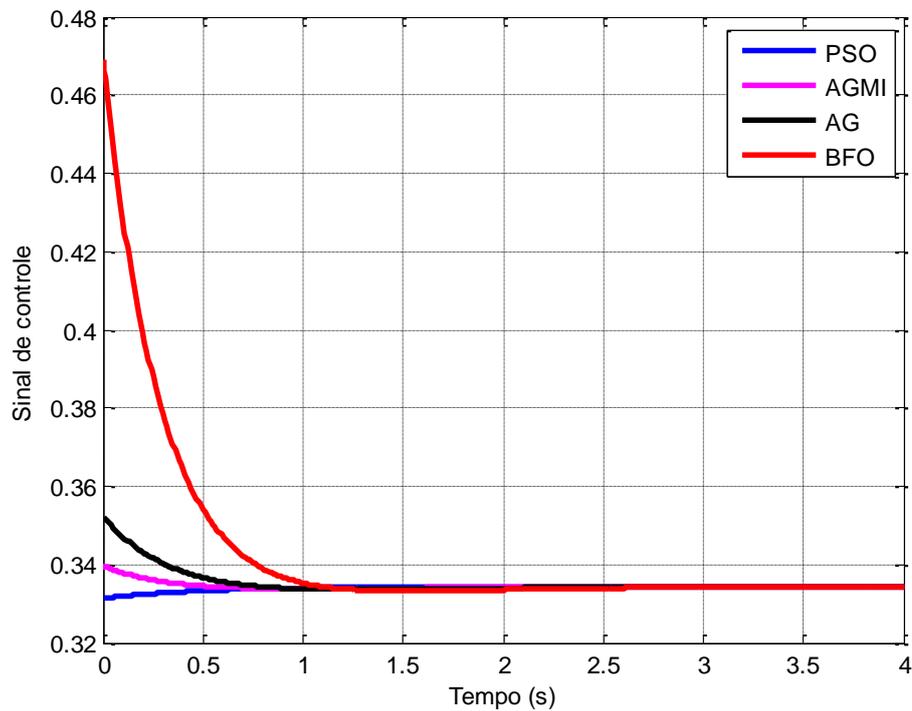


Figura 4.4-Sinais de controle gerados com o uso dos controladores para o caso 1

b) *Caso 2: Sistema lento com erro de regime permanente*

A Equação 4.6 representa a função de transferência do sistema de primeira ordem utilizado para este caso:

$$G(s) = \frac{0,62}{8,5s + 1} \quad (4.6)$$

A Figura 4.5 apresenta a resposta do sistema a uma entrada do tipo degrau unitário.

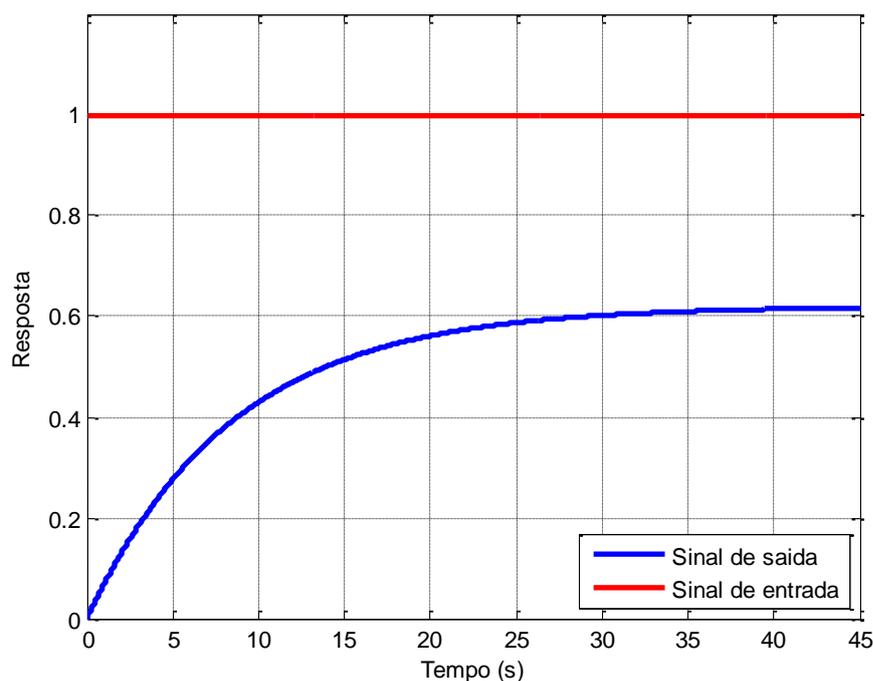


Figura 4.5-Sinal de saída do sistema

Este sistema responde de forma lenta a excitação do sinal de entrada e apresenta erro em regime estacionário e desta forma o PID sintonizado pelos algoritmos deverá melhorar as características da resposta transitória e de regime permanente. Os valores das especificações do projeto são mostrados na Tabela 4.8 juntamente com os resultados obtidos com os algoritmos.

Tabela 4.8-Especificação de projeto e valores alcançados para o caso 2

Parâmetros	AG	AGMI	BFO	PSO	Projeto
M_p	0%	0%	0%	0%	0%
M_p médio	0%	0%	0%	0%	-
M_p Desvio	0,000	0,000	0,000	0,000	-
T_r (s)	0,827	0,828	0,833	0,814	0,833
T_r médio	0,812	0,810	0,832	0,814	-
T_r Desvio	0,014	0,019	0,003	0,000	-
T_s (s)	1,490	1,491	0,150	1,466	1,5
T_s médio	1,462	1,459	1,498	1,466	-
T_s Desvio	0,024	0,035	0,005	0,000	-
Tempo médio de busca(s)	55,3	37,22	113,86	25,79	

Estas especificações impostas como valores de projetos são usadas para gerar a resposta limitante que é usada por cada algoritmo como referência para sintonizar o PID. As Figuras 4.6, 4.7, 4.8 e 4.9 apresentam respectivamente a comparação das respostas geradas pelo PID sintonizados pelo AG, AGMI, BFO e PSO em relação a resposta limitante de projeto.

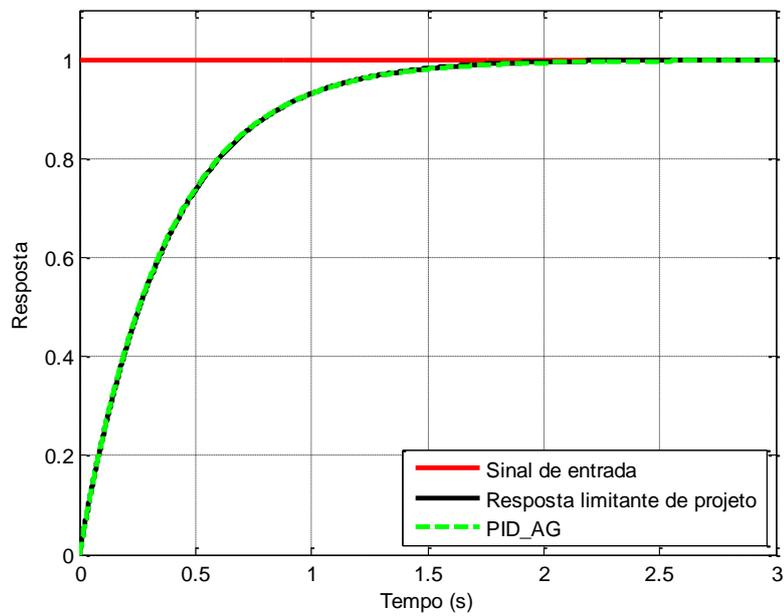


Figura 4.6- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AG para o caso 2

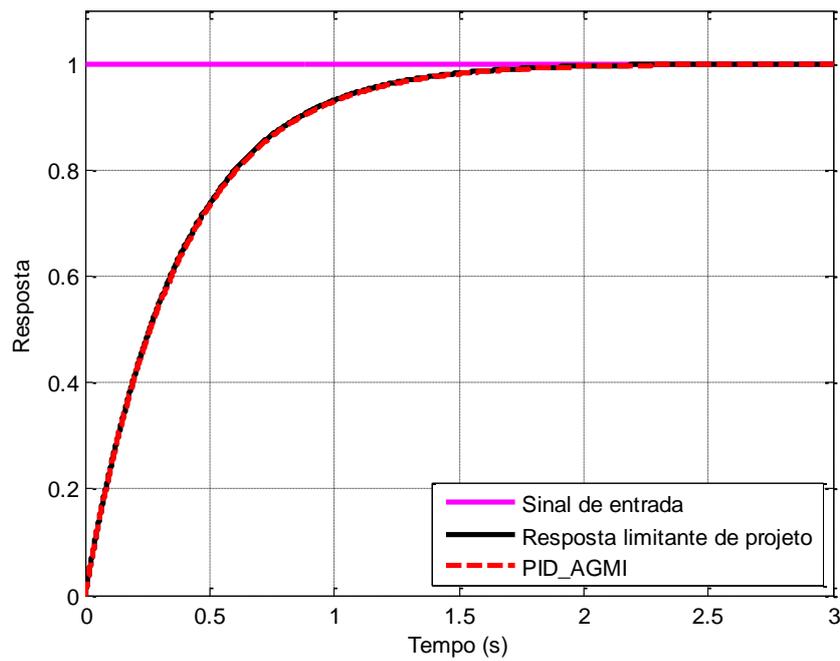


Figura 4.7- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AGMI para o caso 2

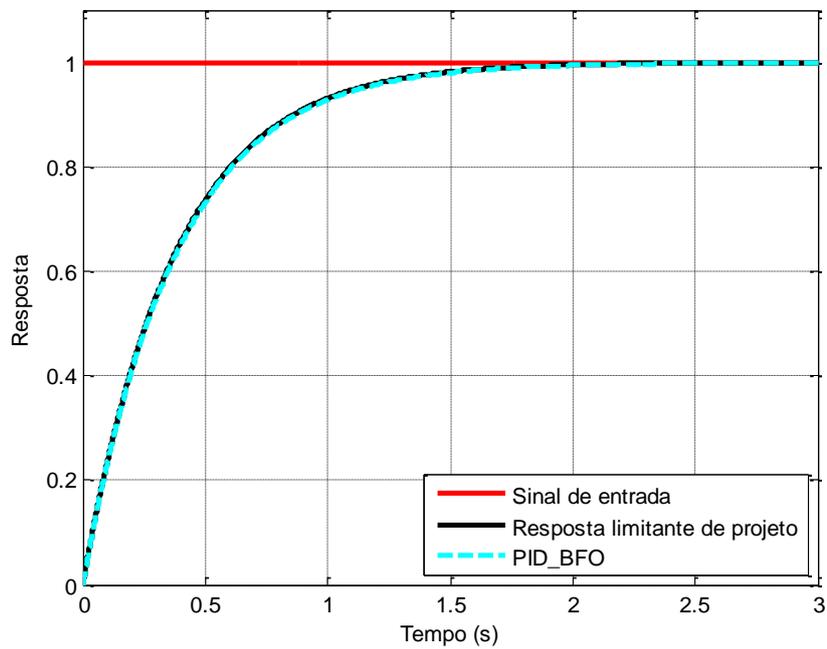


Figura 4.8- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo BFO para o caso 2

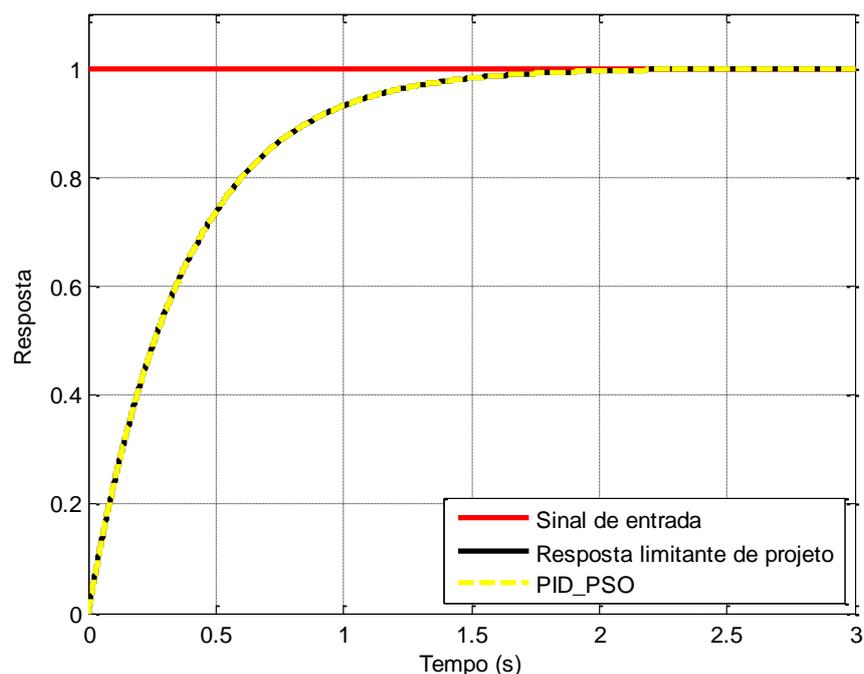


Figura 4.9- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo PSO para o caso 2

A Tabela 4.9 apresenta os valores sintonizados pelos algoritmos e na Tabela 4.10 são mostrados os valores dos índices de desempenho obtidos. A Figura 4.10 representa os sinais de controle gerados com cada um dos PID sintonizados.

Tabela 4.9- Parâmetros sintonizados pelos algoritmos para o caso 2

Parâmetros	AG	AGMI	BFO	PSO
K_p	33,199	32,569	32,489	32,883
K_p médio	33,744	33,109	32,651	32,883
K_p Desvio	0,5623	0,742	0,140	0,000
T_i	8,981	8,621	8,702	8,500
T_i médio	8,999	8,514	8,706	8,500
T_i Desvio	0,0421	0,089	0,050	0,000
T_d	0,000	0,000	0,000	0,000
T_d médio	0,000	0,000	0,000	0,000
T_d Desvio	0,000	0,000	0,000	0,000

Tabela 4.10-Valores dos índices de desempenho para o caso 2

Índices	AG	AGMI	BFO	PSO
IAE	0,3756	0,3793	0,3811	0,3745
ISE	0,1860	0,1892	0,1469	0,1873
ITAE	0,1455	0,1448	0,1706	0,1399
ITSE	0,0347	0,0359	0,0361	0,0351

Para este caso todos os algoritmos conseguiram sintonizar o PID de forma a produzir uma resposta muito próxima da resposta limitante de projeto, atendendo deste modo os requisitos de projeto, sendo que o BFO se aproximou mais como pode ser visto na Tabela 4.8. Como pode ser observados nas Tabelas 4.8 e 4.9 o PSO obteve uma boa convergência nas simulações feitas gerando um desvio padrão igual a zero e obtendo ainda o melhor tempo de busca em relação aos outros algoritmos. Os índices de desempenho gerados por cada algoritmo (ver Tabela 4.10) mostram valores baixos o que indica que os PIDs sintonizados pelos algoritmos produzem boas respostas a uma entrada degrau, e o PSO foi o que produziu o melhor valor em relação aos outros algoritmos. A Figura 4.10 apresenta os sinais de controle obtidos com o uso de cada um dos PIDs sintonizados pelos algoritmos, como pode ser visto os sinais de controle para este caso apresenta um valor inicial bastante elevado, o que pode não ser aceitável para uma aplicação real.

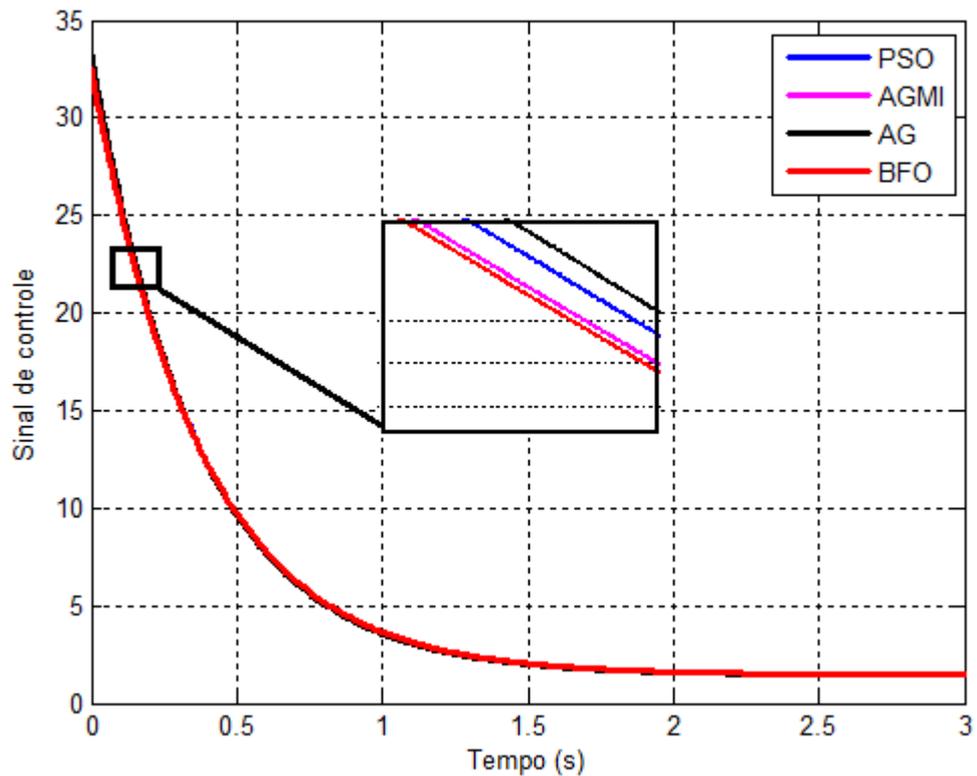


Figura 4.10-Sinais de controle gerados com o uso dos controladores para o caso 2

c) ***Caso 3: Sistema lento sem erro de regime permanente***

Para este caso a função do sistema usada está representada na Equação 4.7 e o seu sinal de saída para um sinal de entrada do tipo degrau é apresentado na Figura 4.11. Este sistema possui uma resposta que consegue seguir um sinal de entrada do tipo degrau unitário, porém apresenta uma característica de resposta lenta como pode ser observado na Figura 4.11. Desse modo, para este sistema, propõe-se a sintonia de um controlador PID que torne o sistema mais rápido e mantenha o sistema sem erro de regime permanente.

$$G(s) = \frac{1}{7s + 1} \quad (4.7)$$

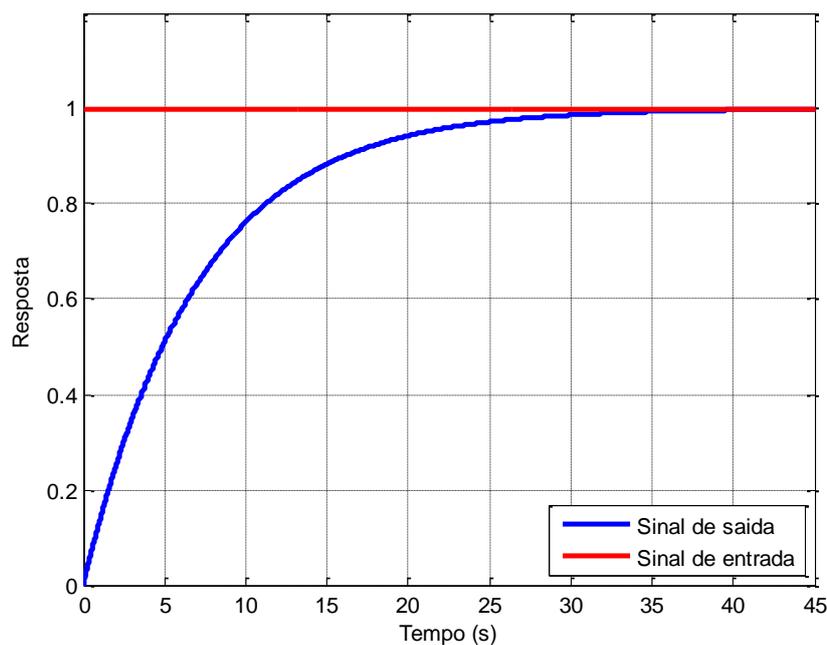


Figura 4.11-Resposta do sistema do caso 3 para uma entrada do tipo degrau unitário

A Tabela 4.11 apresenta os valores das especificações de projeto e os valores alcançados com o uso de cada PID sintonizado com o uso dos algoritmos. A Tabela 4.12 contém os valores dos parâmetros sintonizados com o uso do AG, AGMI, BFO e PSO. Nas Figuras 4.12, 4.13, 4.14 e 4.15 são apresentadas respectivamente a comparação da resposta do controlador sintonizado com o AG, AGMI, BFO e PSO em relação à resposta desejada de projeto. Todos os algoritmos conseguiram atender as especificações impostas no projeto, gerando respostas bem próximas da resposta limitante de projeto. O tempo de busca obtido pelo PSO se mostrou mais rápido em relação ao tempo médio gasto pelo os outros três algoritmos.

Tabela 4.11-Especificação de projeto e valores alcançados para o caso 3

Parâmetros	AG	AGMI	BFO	PSO	Projeto
M_p	0%	0%	0%	0%	0%
M_p médio	0%	0%	0%	0%	-
M_p Desvio	0,000	0,000	0,000	0,000	-
T_r (s)	1,659	1,653	1,661	1,629	1,67
T_r médio	1,643	1,637	1,650	1,629	-
T_r Desvio	0,018	0,012	0,034	0,000	-
T_s (s)	2,984	2,976	2,990	2,932	3
T_s médio	2,957	2,947	2,970	2,932	-

T_s Desvio	0,032	0,021	0,019	0,000	-
Tempo médio simulação(s)	55,14	33,87	87,42	23,34	-

Tabela 4.12-Valores dos parâmetros sintonizados pelos algoritmos para o caso 3

Parâmetros	AG	AGMI	BFO	PSO
K_p	9,551	9,406	9,511	9,343
K_p médio	9,408	9,383	9,319	9,343
K_p Desvio	0,105	0,027	0,241	0,000
T_i	7,283	7,134	7,253	7,000
T_i médio	7,099	7,031	7,067	7,000
T_i Desvio	0,134	0,078	0,149	0,000
T_d	0,000	0,003	0,000	0,000
T_d médio	0,001	0,005	0,000	0,000
T_d Desvio	0,002	0,002	0,000	0,000

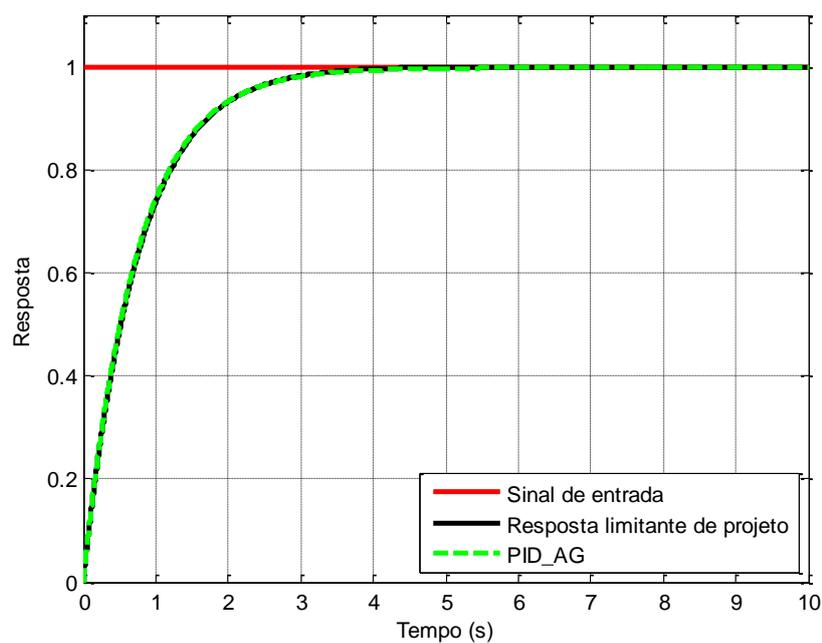


Figura 4.12-Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AG para o caso 3

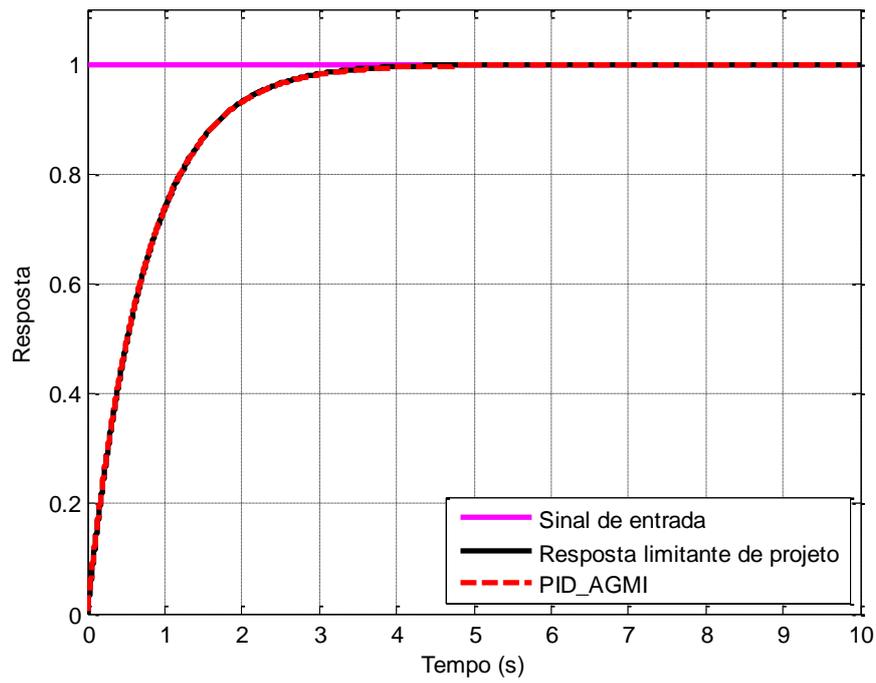


Figura 4.13- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo AGMI para o caso 3

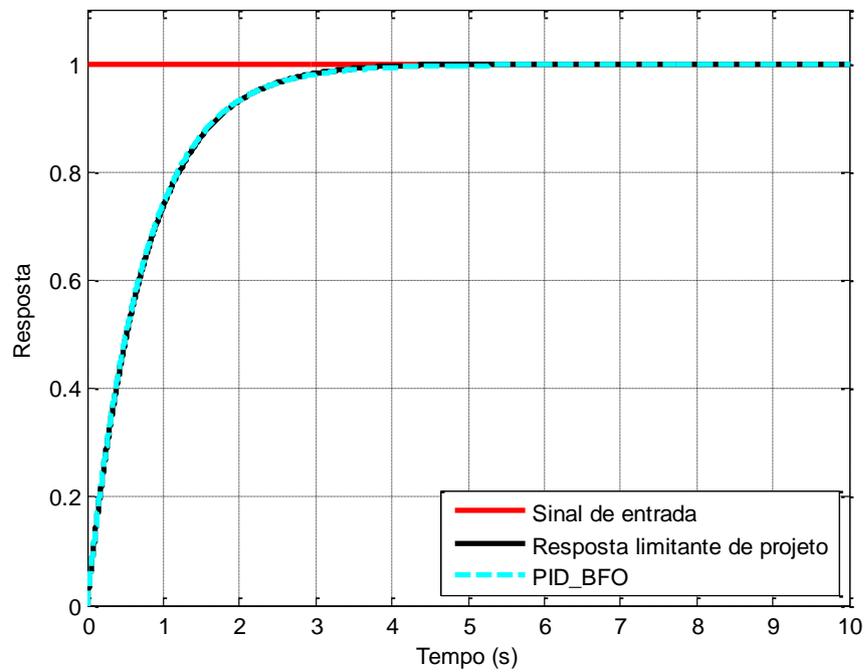


Figura 4.14- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo BFO para o caso 3

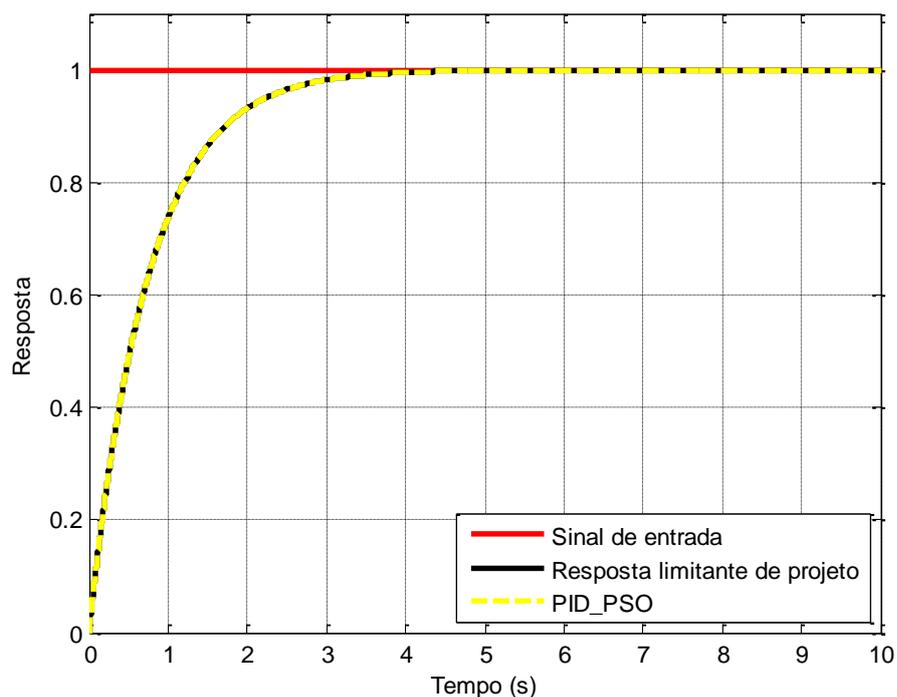


Figura 4.15- Comparação da resposta limitante de projeto e a resposta com o uso do PID sintonizado pelo PSO para o caso 3

A Tabela 4.13 contém os valores dos índices de desempenho gerados com o uso dos controladores e a Figura 4.16 apresenta a comparação entre os sinais de controle gerados por cada PID sintonizado, onde também pode ser observado que os sinais estão muito próximos e apresentam um valor inicial alto.

Tabela 4.13- Valores dos índices de desempenho para caso 3

Índices	AG	AGMI	BFO	PSO
IAE	0,7553	0,7542	0,7543	0,7492
ISE	0,3679	0,3722	0,3693	0,3746
ITAE	0,6358	0,6036	0,6304	0,5613
ITSE	0,1360	0,1393	0,1373	0,1403

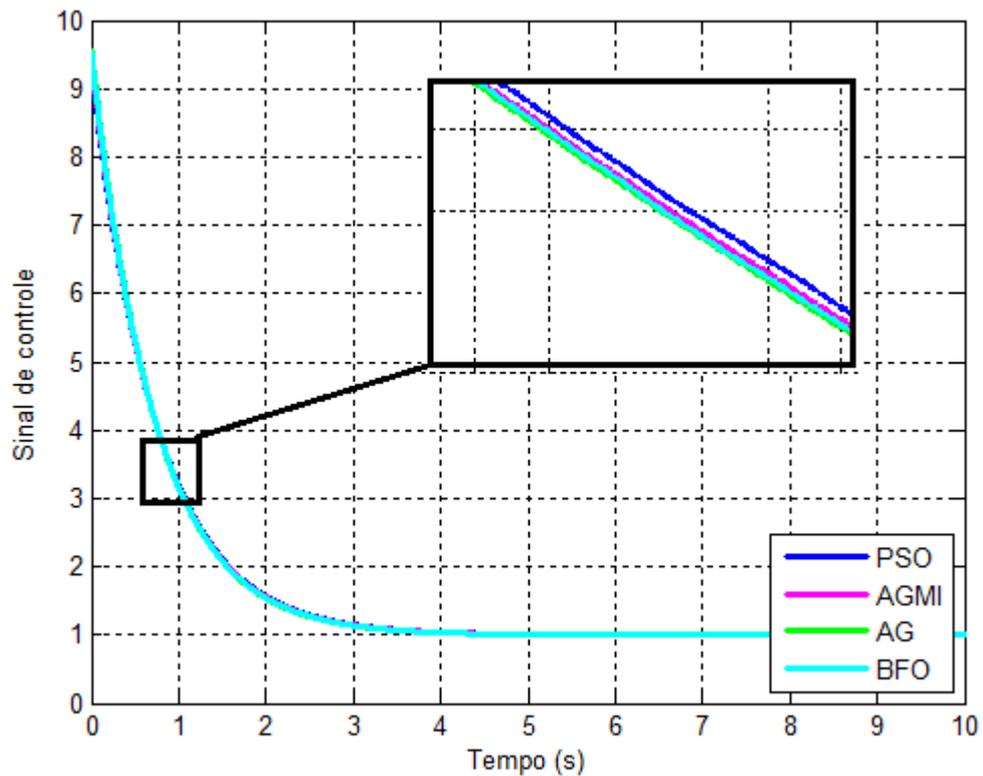


Figura 4.16-Sinais de controle gerados com o uso dos controladores para o caso 3

4.4.3 Sistema de Segunda Ordem

Para avaliar o desempenho dos algoritmos na sintonia de controladores PID para o caso de sistemas de segunda ordem foram utilizados dois sistemas de segunda ordem: um sistema superamortecido e um sistema sub amortecido.

a) *Caso 4: Sistema superamortecido*

A função de transferência usada para este caso é apresentada na Equação 4.8, trata-se de um sistema de segunda ordem superamortecido e sua resposta a um degrau unitário é mostrada na Figura 4.17.

$$G(s) = \frac{10}{s^2 + 6s + 5} \quad (4.8)$$

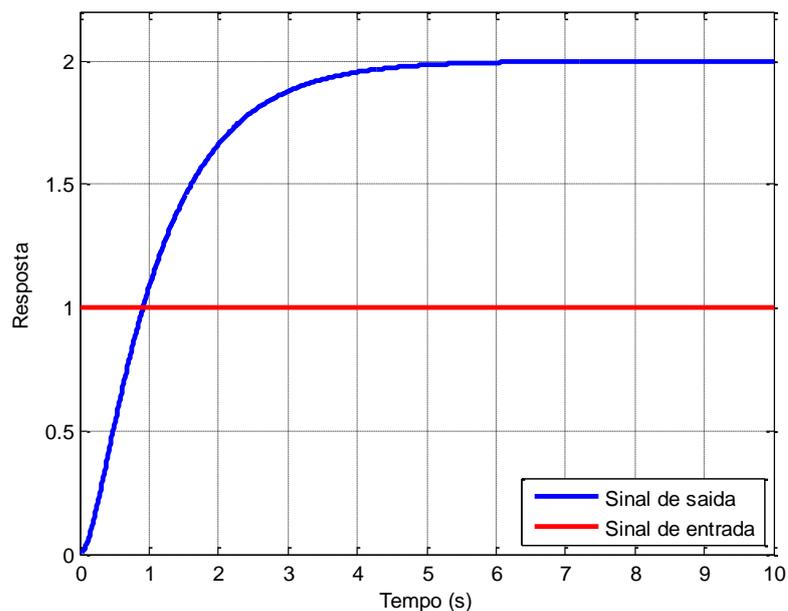


Figura 4.17-Resposta do sistema do caso 4 para uma entrada do tipo degrau unitário

A Tabela 4.14 contém as especificações de projetos para este caso e os valores obtidos com o uso de cada controlador sintonizado pelos algoritmos. A Tabela 4.15 mostra os valores sintonizados dos parâmetros do PID pelo AG, AGMI, BFO e PSO. Na Figura 4.18 é realizada a comparação entre a resposta com o uso destes controladores e a resposta de projeto.

Tabela 4.14-Especificação de projeto e valores alcançados para o caso 4

Parâmetros	AG	AGMI	BFO	PSO	Projeto
M_p	11,48%	11,48%	11,5%	11,45%	11,5%
M_p médio	11,46%	11,47%	11%	11,22%	-
M_p Desvio	0,0001	0,0002	0,0044	0,0043	-
T_r (s)	0,6950	0,7744	0,7389	0,7028	0,833
T_r médio	0,6951	0,7178	0,7778	0,7013	-
T_r Desvio	0,0042	0,0405	0,0554	0,0038	-
T_s (s)	1,2510	1,3940	1,3300	1,2650	1,50
T_s médio	1,2513	1,2920	1,4000	1,2622	-
T_s Desvio	0,0076	0,0730	0,0997	0,0069	-
Tempo médio Simulação (s)	56,91	34,79	62,89	26,13	-

Tabela 4.15-Valores dos parâmetros sintonizados pelos algoritmos para o caso 4

Parâmetros	AG	AGMI	BFO	PSO
K_p	2,1165	1,9514	1,9498	2,0459
K_p médio	2,1253	2,0995	1,9516	2,0409
K_p Desvio	0,0173	0,1131	0,3200	0,0274
T_i	1,0403	0,9358	1,0028	1,0485
T_i médio	1,0371	0,9987	0,9468	1,0575
T_i Desvio	0,0053	0,0449	0,0414	0,0110
T_d	0,0051	0,0116	0,0000	0,0000
T_d médio	0,0062	0,0103	0,0106	0,0000
T_d Desvio	0,0009	0,0012	0,0212	0,0000

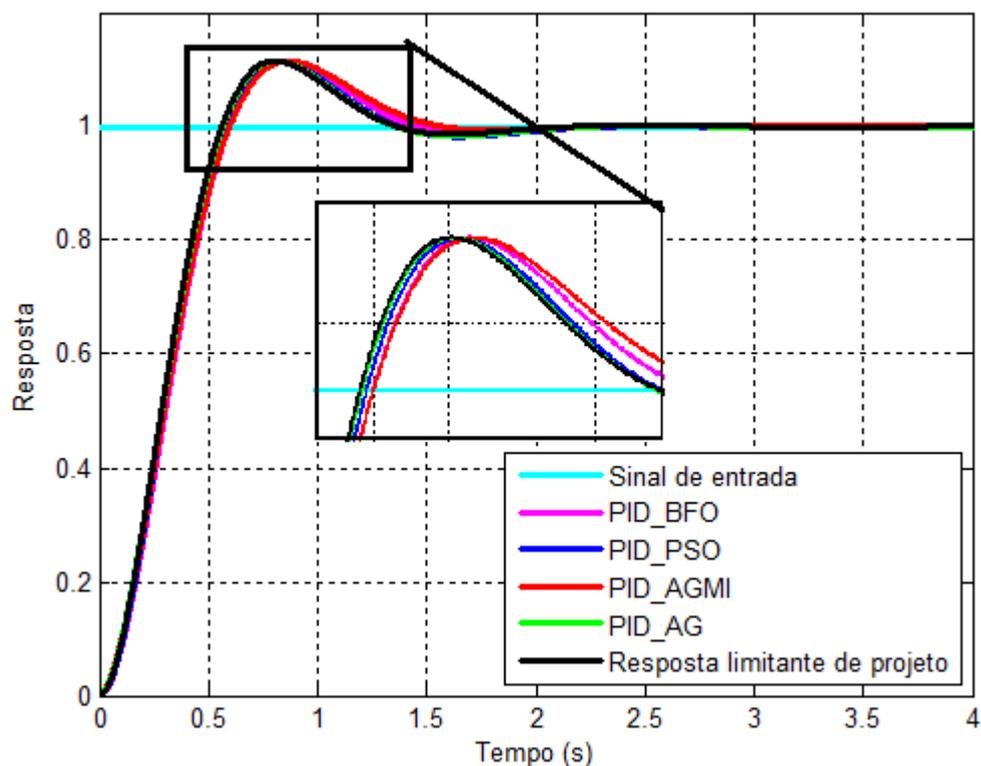


Figura 4.18-Comparação da resposta limitante de projeto e a resposta com o uso dos controladores sintonizados pelos algoritmos para o caso 4

Para este caso todos os algoritmos atenderam as exigências de projeto como pode ser observado através da Tabela 4.14, sendo que os algoritmos que mais se aproximaram do tempo de acomodação (T_s) e do tempo de subida (T_r) de projeto foram

o AGMI e o BFO. O PSO e o AGMI foram os que obtiveram os tempos médios de simulações mais rápidos. A Figura 4.19 apresenta os sinais de controle gerados e na Tabela 4.16 os valores dos índices de desempenho obtidos com o uso dos controladores, onde pode ser visto que para este caso o AGMI produziu os melhores valores de índices, mas seguido com valores próximos pelos outros algoritmos.

Tabela 4.16-Valores dos índices de desempenho para caso 4

Índices	AG	AGMI	BFO	PSO
IAE	0,3567	0,3754	0,3713	0,3619
ISE	0,2172	0,2270	0,2282	0,2220
ITAE	0,1192	0,1330	0,1254	0,1215
ITSE	0,0332	0,0337	0,0364	0,0343

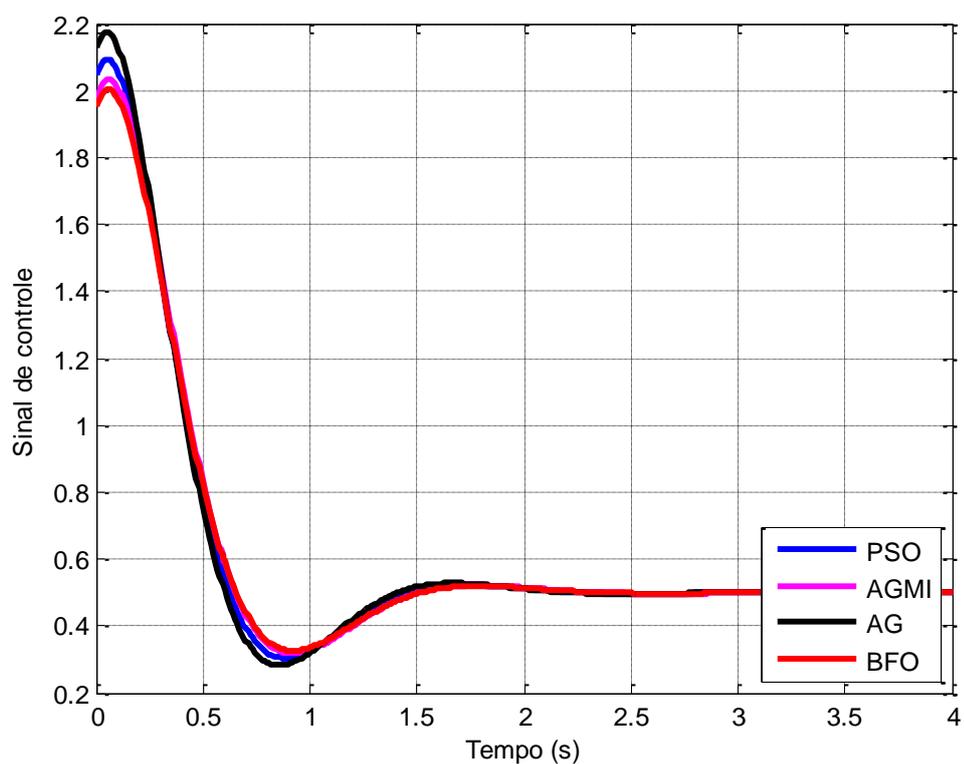


Figura 4.19-Sinais de controle gerados com o uso dos controladores para o caso 4

b) Caso 5: Sistema sub amortecido

Para este caso foi utilizado o sistema representado pela função de transferência da Equação 4.9. Trata-se de um sistema sub amortecido com uma resposta a uma excitação do tipo degrau que apresenta uma característica oscilatória no período transitório e um erro de regime estacionário (ver Figura 4.20). Para este sistema é proposta a sintonia de um controlador PID que melhore a resposta transitória e elimine o erro de regime permanente de sua resposta.

$$G(s) = \frac{0,294}{s^2 + 0,14s + 0,49} \quad (4.9)$$

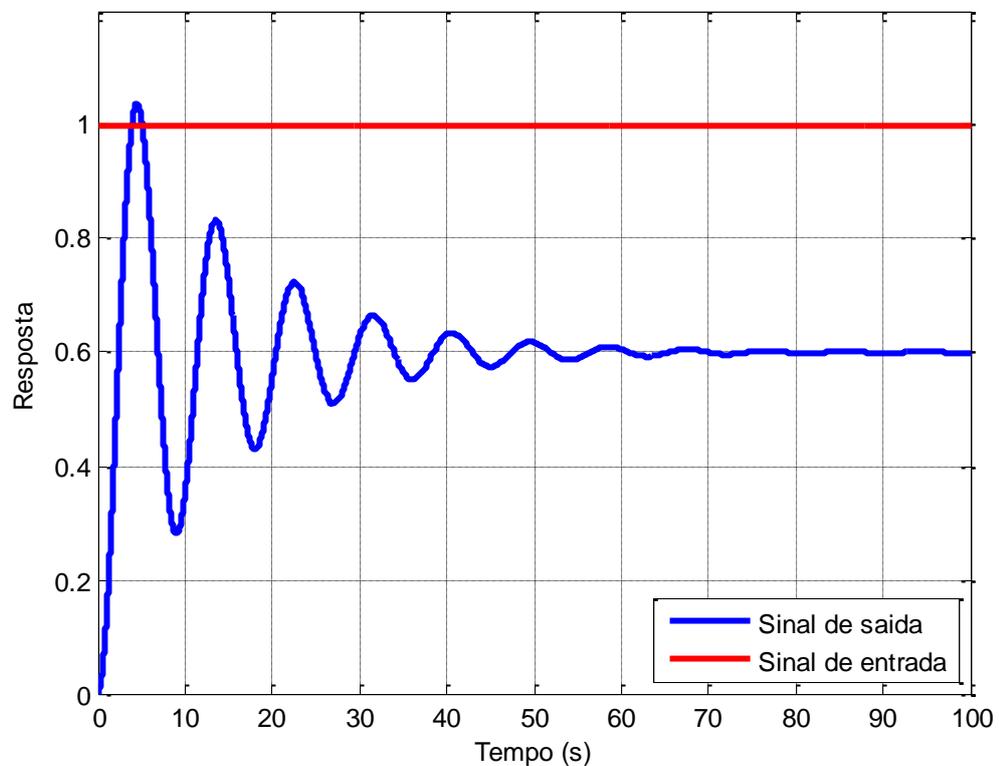


Figura 4.20-Resposta do sistema do caso 5 para uma entrada do tipo degrau unitário

A Tabela 4.17 apresenta os valores das especificações de projeto para este caso assim como os valores obtidos pelo uso dos controladores sintonizados pelo AG, AGMI, BFO e PSO. Para este caso o PSO produziu um valor médio de M_p mais próximo em relação à resposta de projeto e o BFO conseguiu um valor médio de T_s e T_r mais próximo das especificações de projeto. O melhor valor de M_p foi obtido com o BFO e os melhores T_s e T_r foram pelo AGMI.

Tabela 4.17-Especificação de projeto e valores alcançados para o caso 5

Parâmetros	AG	AGMI	BFO	PSO	Projeto
M_p	12,91%	14,39%	10,77%	14,29%	15%
M_p médio	12,31%	12,90%	10,43%	13,95%	-
M_p Desvio	0,0119	0,0172	0,0028	0,0039	-
T_r (s)	0,6256	0,5400	0,667	0,570	0,83
T_r médio	0,6055	0,5372	0,6695	0,5919	-
T_r Desvio	0,0403	0,0109	0,0191	0,0253	-
T_s (s)	1,126	0,9720	1,2000	1,026	1,5
T_s médio	1,089	0,9670	1,2050	1,0655	-
T_s Desvio	0,0725	0,0196	0,0106	0,0456	-
Tempo médio simulação (s)	165,61	129,14	179,86	108,57	-

A Tabela 4.18 contém os valores dos parâmetros sintonizados por cada algoritmo e na Figura 4.21 é feita a comparação das respostas do sistema com o uso de cada um dos controladores e a resposta limitante de projeto.

Tabela 4.18-Valores dos parâmetros sintonizados pelos algoritmos para o caso 5

Parâmetros	AG	AGMI	BFO	PSO
K_p	63,1334	77,6885	48,7144	66,7695
K_p médio	66,3482	78,1710	48,5543	63,9626
K_p Desvio	6,4296	1,6931	0,5137	3,2411
T_i	2,1735	5,1281	2,7201	5,4302
T_i médio	2,7995	5,5401	2,6262	4,1683
T_i Desvio	1,2520	1,2980	0,3016	1,4571
T_d	0,4671	0,3782	0,5793	0,4070
T_d médio	0,4697	0,4121	0,5945	0,4266
T_d Desvio	0,0051	0,0392	0,0167	0,0126

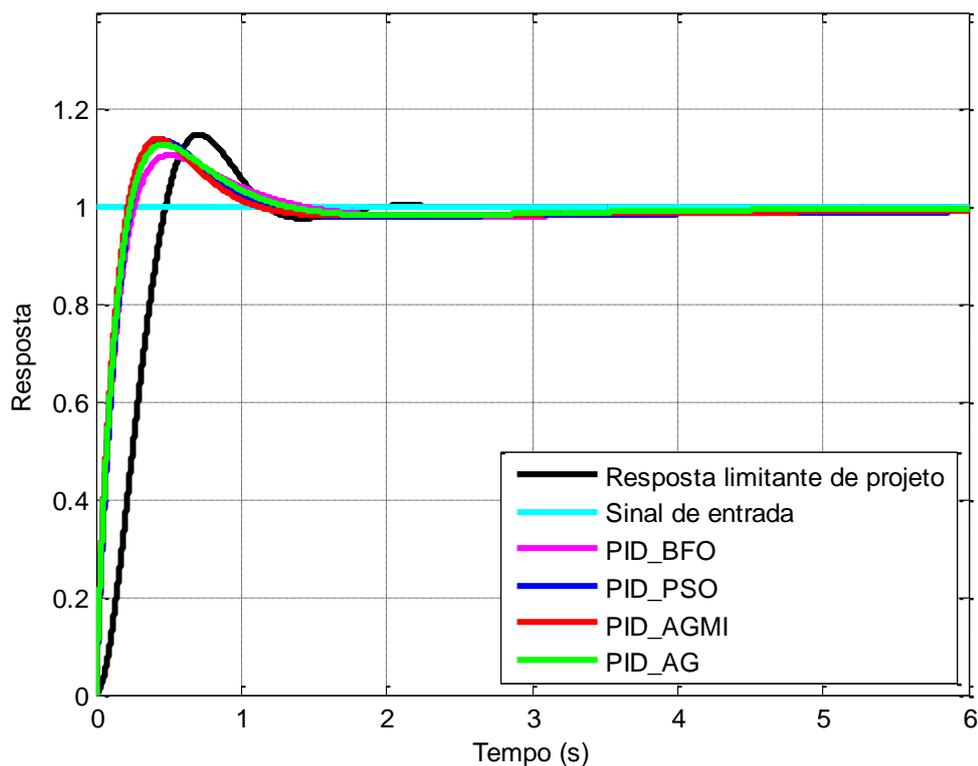


Figura 4.21- Comparação da resposta limitante de projeto e a resposta com o uso dos controladores sintonizados pelos algoritmos para o caso 5

A Tabela 4.19 contém os valores dos índices de desempenho e nela pode ser visto que os valores alcançados por todos os algoritmos foram bons, sendo que o menor valor e por consequência o melhor foi obtido pelo AG.

Tabela 4.19- Valores dos índices de desempenho para o caso 5

Índices	AG	AGMI	BFO	PSO
IAE	0,2015	0,2095	0,2167	0,2290
ISE	0,0658	0,0619	0,0644	0,0664
ITAE	0,1736	0,2301	0,2400	0,2724
ITSE	0,0074	0,0078	0,0083	0,0095

A Figura 4.22 apresenta os sinais de controle de cada um dos PID que foram sintonizados pelos algoritmos. Para este caso os valores alcançados com o uso dos controladores estão excessivamente altos e não seriam aceitáveis em uma aplicação real, estes valores altos deve-se ao fato da função objetivo enfatizar a minimização do erro

entre as resposta dos indivíduos e a resposta de projeto, sem incluir, por exemplo, a minimização do esforço de controle.

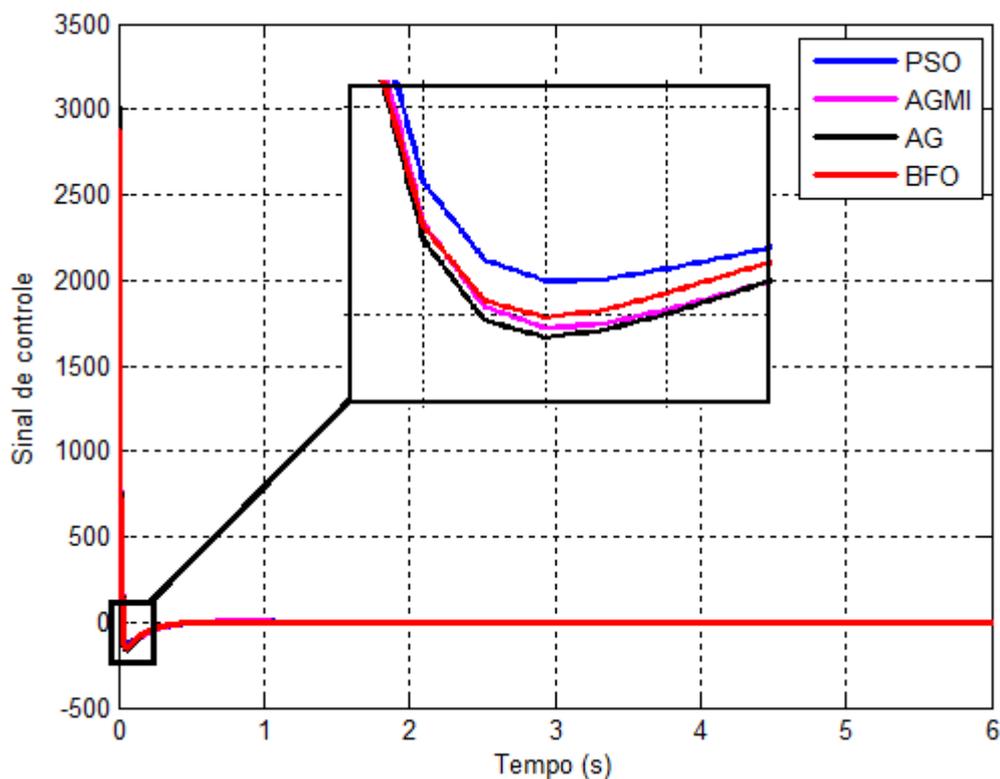


Figura 4.22-Sinais de controle gerados com o uso dos controladores para o caso 5

4.4.4 Sistema de Ordem superior

a) *Caso 6: Sistema de terceira ordem*

Para o caso 6 foi utilizado o sistema de terceira ordem, representado pela função de transferência da Equação 4.10. Os valores obtidos com o uso dos controladores sintonizados para este caso podem ser visto na Tabela 4.20 junto com os valores limitantes de projeto.

$$G(s) = \frac{1}{s^3 + 21s^2 + 20s} \quad (4.10)$$

Tabela 4.20-Especificação de projeto e valores alcançados para o caso 6

Parâmetros	AG	AGMI	BFO	PSO	Projeto
M_p	11,86%	10,2%	11,29%	12,01%	15%
M_p médio	11,46%	8,88%	11,71%	12,96%	-
M_p Desvio	0,0131	0,0242	0,0189	0,0099	-
T_r (s)	2,0978	1,9778	2,0050	2,0139	2,22
T_r médio	2,1583	1,9758	2,0217	2,0532	-
T_r Desvio	0,0664	0,0182	0,0333	0,0409	-
T_s (s)	3,7760	3,5600	3,6090	3,6250	4
T_s médio	3,8850	3,5565	3,6390	3,6957	-
T_s Desvio	0,1196	0,0328	0,0598	0,0737	-
Tempo médio de busca (s)	136,78	114,87	153,12	81,56	

O PSO obteve o valor médio de M_p mais próximo do valor especificado no projeto e o AG obteve os valores médios de T_s e T_r mais próximo da resposta limitante, obtendo. O AGMI obteve os melhores valores de M_p , T_s e T_r alcançados entre todos os algoritmos.

Na Tabela 4.21 são mostrados os valores dos parâmetros sintonizados por cada algoritmo e na Figura 4.23 podem ser vistas as respectivas respostas geradas com o uso de cada algoritmo.

Tabela 4.21-Valores dos parâmetros sintonizados pelos algoritmos para o caso 6

Parâmetros	AG	AGMI	BFO	PSO
K_p	43,4981	47,6570	47,1351	46,4164
K_p médio	41,9779	47,5394	46,4795	44,7923
K_p Desvio	2,4392	1,2261	0,7731	1,5231
T_i	45,9874	49,6442	42,7356	44,8235
T_i médio	44,5049	49,1836	42,3269	45,1977
T_i Desvio	3,9101	0,9857	0,5304	1,3117
T_d	0,4032	0,4590	0,4292	0,4082
T_d médio	0,4095	0,5038	0,4190	0,3803
T_d Desvio	0,0425	0,0813	0,0479	0,0264
Tempo médio simulação (s)				

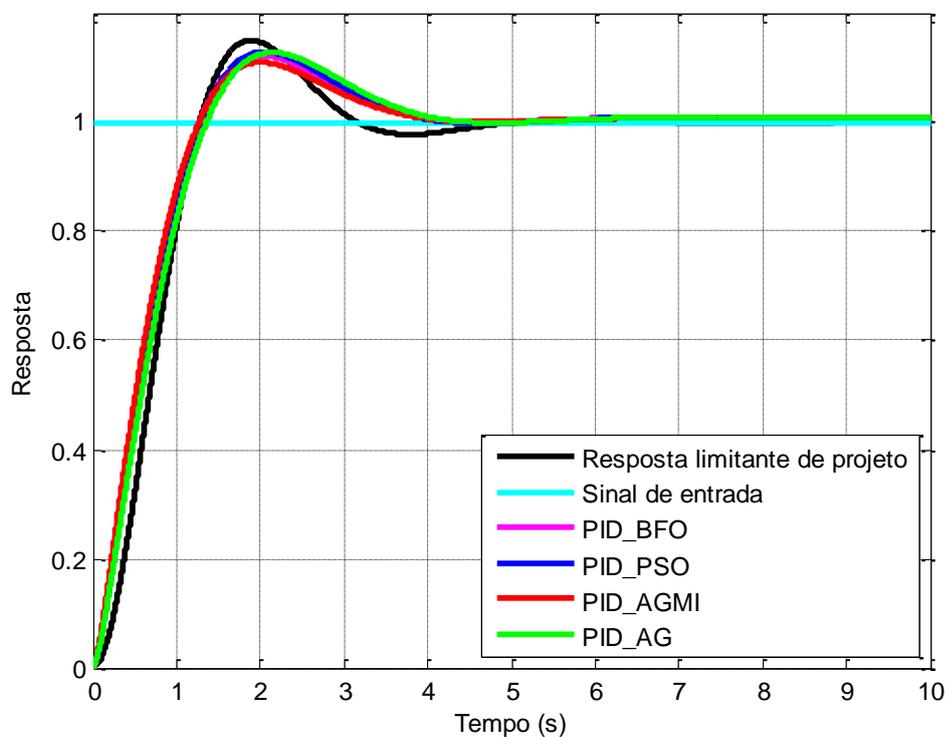


Figura 4.23- Comparação da resposta limitante de projeto e a resposta com o uso dos controladores sintonizados pelos algoritmos para o caso 6

A Tabela 4.22 contém os valores dos índices de cada controlador onde pode ser verificado que o melhor valor foi obtido pelo AG, seguido bem próximo pelos outros algoritmos. Na Figura 4.24 são mostrados os sinais de controle, e para este caso pode ser observado também um valor excessivo de ação de controle.

Tabela 4.22- Valores dos índices de desempenho para o caso 6

Índices	AG	AGMI	BFO	PSO
IAE	0,8516	0,7532	0,7920	0,8147
ISE	0,4304	0,3777	0,3946	0,4087
ITAE	1,0410	0,8646	0,9556	0,9779
ITSE	0,1665	0,1260	0,1409	0,1552

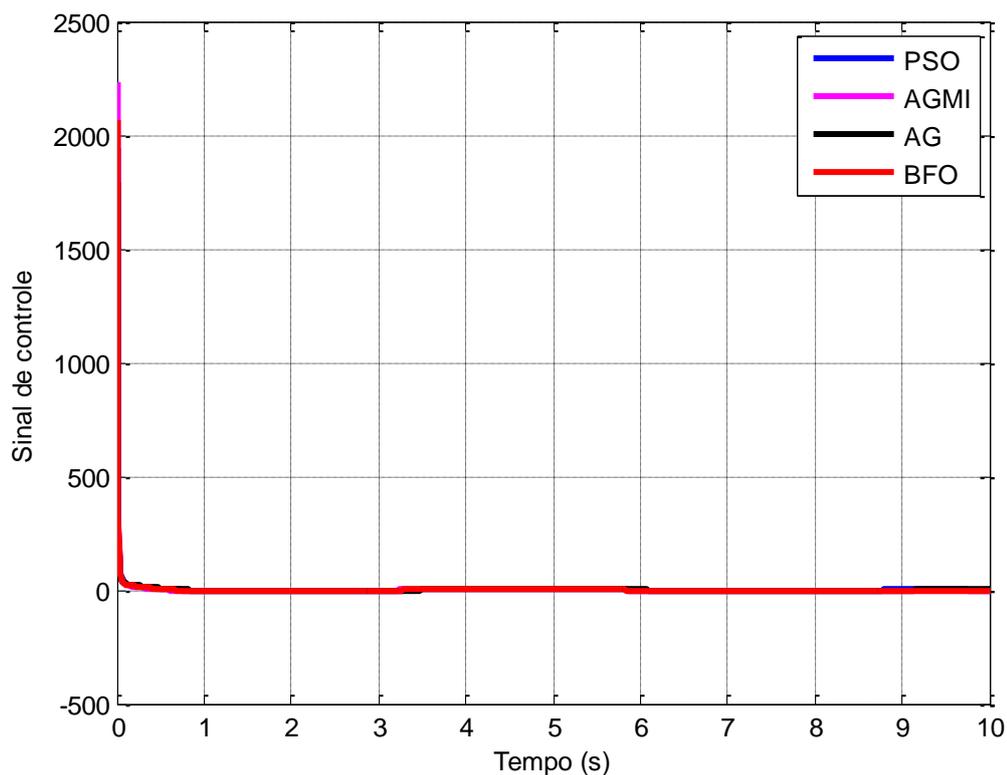


Figura 4.24-Sinais de controle gerados com o uso dos controladores para o caso 6

4.4.5 Considerações sobre resultados alcançados

Foram utilizados diferentes sistemas com características distintas a fim de avaliar o desempenho das metaheurísticas em realizar a sintonia dos parâmetros de controladores PID para atender certas especificações de projeto e, como pôde ser visto pelos resultados encontrados, todas as metaheurísticas tiveram um bom desempenho, produzindo respostas com valores de sobressinal máximo, tempo de subida e de acomodação próximos dos valores limite de projeto. Mesmo os índices de desempenho (ITAE ITSE, IAE e ISE) não sendo usados diretamente no processo de minimização da função custo, os resultados mostram que os algoritmos conseguiram produzir ótimos valores destes índices.

Pelo fato dos algoritmos utilizarem somente como referência o erro entre a resposta limitante de projeto e a resposta gerada pelo uso dos controladores representados por seus indivíduos, não é imposta nenhuma restrição sobre a característica da função de transferência do sistema a ser controlado como, por exemplo,

é imposto pelo método de Ziegler-Nichols, no entanto isso não garante que toda e qualquer exigência de projeto pode ser alcançada, já que isso exige que não sejam passadas especificações de projeto conflitantes e ainda, que exista a parametrização correspondente do PID, para este caso no intervalo de busca especificado.

Pelos resultados obtidos pode ser observado que o PSO possui os tempos médios de busca mais rápidos em relação aos outros algoritmos e isso pode ser atribuído ao número pequeno de etapas que o algoritmo usa, além do fato de serem usadas somente equações algébricas simples na atualização das partículas, enquanto que os outros algoritmos utilizam processos relativamente mais complexos. O PSO também se mostrou um algoritmo mais robusto para a maioria dos casos usados neste trabalho obtendo na maioria destes um valor de desvio padrão igual a zero. De um modo geral apesar de todos os algoritmos produzirem respostas que atenderam as exigências de projeto pode ser considerado que o PSO foi o que obteve o melhor desempenho entre os quatro algoritmos usados neste trabalho.

5 CONCLUSÃO

Neste trabalho foram apresentadas as principais características das metaheurísticas populacionais que tornam estas técnicas eficientes na maioria dos problemas de otimização e foi destacada a capacidade delas em resolver diferentes problemas encontrados na literatura de sistema de controle. Foi apresentado também o estado da arte destas metaheurísticas aplicadas em problemas de sistemas de controle.

Foi realizada a comparação de quatro metaheurísticas populacionais na sintonia dos parâmetros de um controlador PID que visava atender certas especificações de projeto que eram impostas aos algoritmos. A metodologia aqui apresentada é baseada na criação de uma resposta de referência a partir das especificações de projeto, sendo que esta resposta funciona como uma resposta limitante de projeto e o algoritmo deve minimizar o erro entre ela e as respostas de seus indivíduos, a fim de produzir respostas mais próximas dessa resposta de projeto, e por consequência respostas que atendam as exigências de projetos.

Foram utilizados diferentes sistemas com características específicas de regime transitório e de regime permanente para avaliar os algoritmos usados. E para avaliar a qualidade das respostas produzidas pelos controladores foram utilizados os índices de desempenho encontrados na literatura. Através dos resultados encontrados foi possível verificar que estas metaheurísticas possuem um bom desempenho e podem ser usadas para realizar a sintonia de controladores. Observou-se também que de um modo geral o algoritmo que obteve o melhor desempenho para o problema de sintonia foi o *Particle Swarm Optimization* Seguido pelo Algoritmo Genético no Modelo de Ilhas (AGMI).

5.1 Trabalhos Futuros

Em alguns casos mostrados neste trabalho o sinal de controle gerado pelos controladores ficou em um valor inicial bastante elevado e isso se deve ao fator de a função objetivo usada pelas metaheurísticas levar em consideração somente a minimização entre o erro das respostas dos indivíduos e a resposta de projeto, uma sugestão de trabalho futuro seria o estudo para a inclusão da ação de controle no processo de minimização. Desde modo sintonizar os parâmetros do PID para produzir respostas que atendam os critérios de parada e que gerem ações de controle com valores mínimos.

Outra sugestão de trabalho futuro seria a construção de uma interface gráfica contendo os quatros algoritmos, possibilitando a parametrização e utilização deles de uma forma mais fácil. Esta Interface poderia permitir que fossem selecionados os requisitos de projetos assim como o sistema a ser controlado. Esta abordagem poderia ser usada para aplicações de ensino e estudo das metaheurísticas aplicadas em sistemas de controle.

A aplicação desta metodologia apresentada neste trabalho pode ser usada para a parametrização de outros tipos de controladores encontrados na literatura, como por exemplo, o controlador RST. Outra possibilidade de trabalho futuro seria o estudo comparativo da sintonia dos parâmetros de controladores usando as metaheurísticas populacionais e os métodos clássicos disponíveis na literatura de sistemas de controle.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE, L. A. **“Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais”**. Editora UFMG. 3ª Ed, 2007.

BERG, H. C. and BROWN, D. A. **“Chemotaxis in Escherichia coli analyzed by three-dimensional tracking”**. Nature, 1972.

BERG, H. C., **“Motile Behavior of Bacterial”**. Physics Today, 2000.

BERTACHI, A. H., **“Otimização de Parâmetros via Metaheurísticas Populacionais e Validação de um Controlador de Estrutura Variável”**. Dissertação, Universidade Tecnológica Federal do Paraná, 2014.

BOUSSAID, I.; LEPAGNOT, I., and SIARRY, P., **“A survey on optimization metaheuristics”**. Information Sciences, 237, 82-117, 2013.

CHAIB, L.; CHOUCHA, A. and ARIF, S., **“Optimal Design and Tuning of Novel fractional Order PID Power System Stabilizer Using a New Metaheuristic Bat Algorithm”**. Aims Shams Engineering Journal, 2015.

CHANG, W. D. and SHIH, S. P., **“PID Controller Design of Nonlinear Systems using an improved Particle Swarm Optimization Approach”**. Commun Nonlinear SciNumerSimulat, journal, 2010.

CHEN, C. **“Analog and Digital Control System Design: Transfer-Function, State Space, and Algebraic Methods”**. Oxford University Press, USA, 1995.

CHIOU, J. S.; TSAI, S.H. and LIU, M. T., **“A PSO-Based Adaptive Fuzzy PID-Controllers”**. Simulation Modelling Practice and Theory Journal, 2012.

CLERC, M. **“Particle Swarm Optimization”**. ISTE Ltd, Great Britain, 2006.

DORF, R.C. e BISHOP, R. H. **“Sistemas de Controle Modernos”**. 8ª ed. Rio de Janeiro: LTC, 2001.

EL-ABD, M., **“Performance assessment of foraging algorithms vs. evolutionary algorithms”**. Information Sciences. 182, 243-263, 2012.

FERREIRA, A. O.; SILVA, O. F. e BARREIROS, J. J., **“Uso de Algoritmo Genético para Redução de Ordem de Modelos Matemáticos”**. X SBAI-Simpósio Brasileiro de Automação Inteligente, 2011.

GAO, L.; DAI Y. and XIA, J., **“A new framework for power system identification based on an improved genetic algorithm”**. Industrial Electronics and Applications, 4th IEEE Conference on. Xi'an, 2009.

GLOVER, F. and Kochenber, G.A. **“Handbook of Metaheuristics”**. Kluwer Academic Publishing, 2008.

GOLDBERG, D.E. **“Genetic Algorithms in Search, Optimization and Machine Learning”**. AddisonWesley, 1989.

GOLDBERG, D. E. **“Real-Coded Genetic Algorithms, Virtual Alphabets, and Blocking”**. Dept. of General Engineering, IlliGAL Report 90001, Illinois, EUA, 1990.

GUPTA, S. and GUPTA, R., **“PSO and PSO-BFO Based Tuning of PID Controller: A Comparative evaluation”**. International Journal of Emerging Technology and Advanced Engineering, 2014.

HAUPT, R. L. and HAUPT, S. E. **“Practical Genetic Algorithm”**. 2ª ed. John Wiley & Sons, Inc. New Jersey, 2004.

HELDEN, J. V., TOUSSAINT, A., and THIEFFRY, D., **“Bacterial Molecular Networks: Methods and Protocols”**. New York: Springer, 2012.

IBRAHIM, H. E. A.; HASSAN, F. N. and SHOMER, A. O., “**Optimal PID Control of a Brushless DC Motor Using PSO and BF Techniques**”. *Aim Shams Engineering Journal*, 2013.

JANSON, J. and MIDDENDORF, M., “**A hierarchical particle swarm optimizer for noisy and dynamic environments**”. *Genetic Programming and Evolvable Machines*, 7(4): 329-354, 2006.

KANG, J.; MENG, W.; ABRAHAM, A. and LIU, H. “**An adaptative PID Neural Network for Complex Nonlinear System Control**”. *Neurocomputing Journal*, 2014.

KENNEDY, J. and EBERHART, R. “**Particle Swarm Optimization**”. *Proceedings of IEEE of international Conference on Neural Networks*, 1995.

KRISHNAKUMAR, K., “**Micro-genetic algorithms for stationary and non-stationary function optimization**”. *Proceedings of SPIE Intelligent Control and Adaptive Systems Conference*, 1989.

LATHA, K. RAJINIKATH, V. and SUREKHA, P. M., “**PSO Based PID Controller Design for a Class of Stable and Unstable Systems**”. *Hindawi Publishing Corporation*, 2013.

LINDEN, R. “**Algoritmos Genéticos. Uma importante ferramenta da Inteligência Computacional**”. 2ª ed. Rio de Janeiro: Brasport, 2006.

MAHESHWARI, D. K., “**Bacteria in agrobiolgy: Plant probiotics**”. Berlin: Springer, 2012.

MANDAL, P.; SARKAR, B. K., SAHA, R. and CHATTERJEE, A., “**Real-Time Fuzzy-Feedforward Controller Design by Bacterial Foraging Optimization for an Electrohydraulic System**”. *Engineering Applications of Artificial Intelligence*, 2015.

MICHALEWICZ, Z. “**Algorithms +Data Structures = Evolutions Programs**”. New York, Springer-Verlag, 1992.

MOHAMED, K.; RAO, K. S. R.; HASAN, K. **“Application of Particle Swarm Optimization and its Variants to Interline Power Flow Controllers and Optimal Power Flow”**. International Conference on Intelligent and Advanced Systems (ICIAS), 2010.

MITCHELL, M. **“An Introduction to Genetic Algorithms”**. MIT Press, 1998.

NANGSUE, P.; PILLAY, P. and CONRY, S. E., **“Evolutionary algorithms for induction motor parameter determination”**. IEEE Trans. Energy Conversion, 1999.

NISE, N. S. **“Engenharia de sistemas de controle”**. 3^a ed. Rio de Janeiro: LTC, 2002.

OGATA, K. **“Engenharia de controle moderno”**. 4^aed. Rio de Janeiro: Prentice/ Hall do Brasil, 2003.

PARMAR, G.; PRASAD, R. and MUKHERJEE, S., **“Order Reduction of Linear Dynamic Systems using Stability Equation Method and Genetic Algorithm”**. International Journal of Computer, Information and Systems Science, 2007.

PARSOPOULOS, K. E. and VRAHATIS, M. N. **“Particle Swarm Optimization and Intelligence Advances and Applications”**. IGI Global, USA, 2010.

PASSINO, K. M. **“Biomimicry of bacterial foraging for distributed optimization and control”**. IEEE Control System Management, 22, 52–67, 2002.

PATEL, R. and KUMAR, V., **“Artificial Neuro Fuzzy Logic PID Controller Based on BF-PSO Algorithm”**. 11th International Multi-Conference on Information Processing, 2015.

PEREIRA, J. J., **“Sintonia do Controlador PID, Com Algoritmo de Otimização por Grupo de Partículas”**. Dissertação, Universidade de Trás-os-Montes e Alto Dourado, 2008.

PILLAY, P.; NOLAN, R. and HAQUE, T., "**Application of genetic algorithms to motor parameter determination for transient torque calculations**". IEEE Trans. Industry Applications, 1997.

RAJINIKANTH, V. and LATHA, K., "**Identification and Control of Unstable Biochemical Reactor**". International Journal of Chemical Engineering and Application, 2010.

RAJINIKANTH, V. and LATHA, K., "**Internal Model Control-Proportional Integral Derivative Controller Tuning for First Order plus Time Delayed Unstable System Using Bacterial Foraging Algorithm**". Scientific Research and Essays, 2012.

RAO S. S. "**Engineering Optimization: Theory and Practice**". Wiley, 2009.

ROTHLAUF, F. "**Representations for Genetic and Evolutionary Algorithms**". Springer-Verlag Berlin Heidelberg, 2006.

SANTOS F. G.; MAZZINI, H. A., "**Métodos de Sintonia de controladores PI/PID Aplicados a um Sistema de Nível**". X SBAI-Simpósio Brasileiro de Automação Inteligente, 2011.

SATAKSHI; MUKHERJEE, S. and MITTAL, R. C., "**Order Reduction of Linear Discrete Systems Using Genetic Algorithm, Applied Mathematical Modeling**". Elsevier, 2005.

SAYED, M.; GHARGHORY, S. M. and KAMAL, H. A., "**Gain Tuning PI Controllers for Boiler Turbine Unit Using a New Hybrid Jump PSO**". Journal of Electrical System and Information Technology, 2015.

SINGH, K.; VASANT, P.; ELAMVAZUTHI, I. and KANNAN, R., "**PID Tuning of Servo Motor using Bat Algorithm**". 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, 2015.

STEFOPOULOS,G. K.; GEORGILAKIS,P. S. ;HATZIARGYRIOU, N. D. and MELIOPOULOS,A. P. S.,"**A Genetic Algorithm Solution to the Governor-Turbine Dynamic Model Identification in Multi-Machine Power Systems**". Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, 2005.

SUN, J.; LAI, C. and WU, X. "**Particle Swarm Optimization Classical and Quantum Perspectives**". Taylor& Francis GroupLLC, 2012.

TANG, W. J., and WU, Q. H., "**Biologically inspired Optimization: A review**". Transactions of the Institute of Measurement and Control, 31, 495-515, 2009.

TARIQUE, A. and GABBAR, H. A., "**Particle Swarm Optimization (PSO) Based Turbine Control**". Intelligent Control and Automation Journal, 2013.

TEODORO, F. R.; PARPINELLI, R. S. e LOPES, H. S., "**Otimização por Colônia de Bactéria Aplicada a problemas de Engenharia**". Associação Argentina de Mecânica Computacional, 2010.

VIDAL, J. F.; SILVA, O. F.; FREITAS, V. S.; MEDEIROS, R. L. P. e CASTRO, A. R. G., "**Estratégia de Identificação de Sistemas e Controle PID Via Algoritmo Genético em modelo de Ilhas**". XII SBAI- Simpósio Brasileiro de Automação Inteligente, 2015.

XING, B. and GAO, W. "**Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms**". Springer International Publishing, Switzerland, 2014.

YANG, X. "**Introduction to Mathematical Optimization from Linear Programming to Metaheuristics**". Cambridge International Science Publishing, 2008.

ZALZALA, A. M. S. and FLEMING, P. J. "**Genetic Algorithms in Engineering Systems**". Institution of Electrical Engineers, United Kingdom, 1997.