



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE
COMUNICAÇÃO E AUTOMAÇÃO**

EMMANUEL RAMON MARQUES DANTAS

**DESENVOLVIMENTO DE UMA *TOOLBOX* PARA
APLICAÇÃO DE INTELIGÊNCIA COMPUTACIONAL
EM SISTEMAS DE CONTROLE CLÁSSICO**

**MOSSORÓ – RN
2013**

EMMANUEL RAMON MARQUES DANTAS

**DESENVOLVIMENTO DE UMA *TOOLBOX* PARA
APLICAÇÃO DE INTELIGÊNCIA COMPUTACIONAL
EM SISTEMAS DE CONTROLE CLÁSSICO**

Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação – PPGSCA, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

Orientador: Prof. Dr. José Patrocínio da Silva –
UFRN

Co-orientador: Prof. Dr. Augusto Carlos Pavão –
UFERSA

**MOSSORÓ – RN
2013**

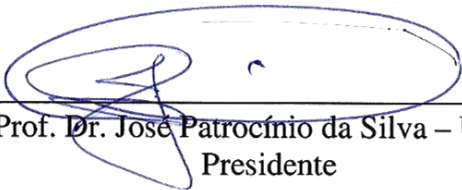
EMMANUEL RAMON MARQUES DANTAS

**DESENVOLVIMENTO DE UMA *TOOLBOX* PARA
APLICAÇÃO DE INTELIGÊNCIA COMPUTACIONAL
EM SISTEMAS DE CONTROLE CLÁSSICO**

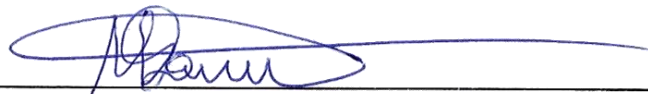
Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação – PPGSCA, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

APROVADA EM: 27 / 12 / 2013.

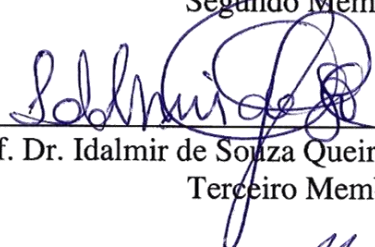
BANCA EXAMINADORA




Prof. Dr. José Patrocínio da Silva – UFRN
Presidente



Prof. Dr. Moisés Dantas dos Santos – UFPB
Segundo Membro



Prof. Dr. Idalmir de Souza Queiroz Júnior – UFERSA
Terceiro Membro



Prof. Dr. Elmer Rolando Llanos Villarreal – UFERSA
Quarto Membro

Mossoró, 27 de dezembro de 2013.

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Central Orlando Teixeira (BCOT)
Setor de Informação e Referência

D192d Dantas, Emmanuel Ramon Marques.

Desenvolvimento de uma toolbox para aplicação de inteligência computacional em sistemas de controle clássico. / Emmanuel Ramon Marques Dantas. -- Mossoró, 2013.

73f.: il.

Orientador: Prof. Dr. José do Patrocínio da Silva

Co-orientador: Prof. Dr. Augusto Carlos Pavão

Dissertação (Mestrado em Sistemas de Comunicação e Automação. Área de concentração em Sistemas de Comunicação e Automação) – Universidade Federal Rural do Semi-Árido. Pró-Reitoria de Pós-Graduação.

1. Inteligência computacional. 2. Algoritmo genético. 3. Sistemas de controle. 4. Interface gráfica I. Título.

RN/UFERSA/BCOT

CDD: 006.3

Bibliotecária: Vanessa Christiane Alves de Souza Borba
CRB-15/452

DEDICATÓRIA

Este trabalho é dedicado à minha mãe,
Zélia Marques.

AGRADECIMENTOS

À Deus, meu maior professor, pelo dom da vida e pela força e perseverança que me concedeu, sempre iluminando o meu caminho.

Ao Prof. D. Sc. José Patrocínio da Silva pela orientação, por sua disponibilidade incondicional, pela confiança em mim depositada, pelos ensinamentos e pela motivação.

À minha mãe, Zélia Marques, e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida, em especial àqueles que me acolheram durante as atividades finais do Mestrado em Natal: tio Jorge, com sua dedicação e aspiração pelo trabalho de excelência, e a tia Maria José, fazendo-me lembrar vovó Dalva com seu infinito cuidado e carinho. Além do meu primo Marcos pelos momentos de descontração e incentivo, todos estes foram fundamentais para concluir mais esta etapa.

À Carla Reis, pelo apoio absoluto, pela compreensão na ausência, pelo carinho, afeto e cumplicidade.

Aos professores e amigos do PPGSCA pela paciência na orientação e incentivo que tornaram possível a conclusão desta dissertação. Em especial à Profa. Dra. Danielle Simone da Silva Casillo e ao Prof. Dr. Idalmir de Souza Queiroz Júnior pelas sugestões, recomendações e críticas construtivas no exame de qualificação, que permitiram melhorar a qualidade do trabalho.

Aos amigos e colegas, pelo incentivo e apoio constantes.

Às instituições UFERSA e UFRN pelas estruturas disponibilizadas para realização das atividades do mestrado e à CAPES pelo suporte financeiro.

EPÍGRAFE

“A fé é o fundamento da esperança, é uma certeza a respeito do que não se vê. Foi ela que fez a glória dos nossos antepassados.”
(Hebreus 11, 1-2)

RESUMO

Uma nova ferramenta baseada no uso da inteligência computacional aplicada em sistemas de controle é apresentada. Esse tipo de aplicação tem despertado o interesse de pesquisadores por apresentar vantagens em relação aos outros métodos de ajustes para controladores, principalmente quando o sistema apresenta uma dinâmica complexa de ser ajustada pelos métodos tradicionais. A ferramenta computacional proposta – GACT (*Genetic Algorithm Control Tool*) foi desenvolvida para funcionar como uma *Toolbox* do software *MATLAB*[®] para aplicação em sistemas de controle e fundamenta-se na aplicação conjunta de Algoritmos Genéticos (GA – do inglês *Genetic Algorithm*) com a teoria de controle clássico. O GACT baseia-se no avanço dos sistemas operacionais que deixaram de ser totalmente manuseados por linhas de comando e passaram a ter uma interface gráfica interativa. Ou seja, a referida *Toolbox* possibilita a implementação de um sistema de controle inteligente de maneira mais simplificada e interativa. A interface gráfica foi concebida através do software GUIDE, integrado ao *MATLAB*[®] de maneira a relacionar-se com os códigos fontes e os diagramas de blocos do sistema a ser controlado no *SIMULINK*[®].

Palavras-Chave: Inteligência Computacional, Algoritmo Genético, Sistemas de Controle, Interface Gráfica.

ABSTRACT

A new tool based on the use of computational intelligence applied in control systems is presented. This type of application has attracted the interest of researchers due the advantages in relation to others methods for controlling settings, especially when the system has a complex dynamics to be adjusted by traditional methods. The proposed computational tool – GACT (Genetic Algorithm Control Tool) was developed to work as a Toolbox of MATLAB[®] software for use in control systems and is based on the joint application of Genetic Algorithms (GA) and the classical control theory. The GACT based on the advancement of the operating systems that are no longer fully handled by command lines, and now have an interactive graphical user interface. That is, the referred Toolbox allows the implementation of an intelligent control system in a way more simplified and interactive. The graphical user interface (GUI) was designed using the software GUIDE, integrated with MATLAB[®] in order to connect with the source code and block diagrams of the system to be controlled at the SIMULINK[®].

Keywords: Computational Intelligence, Genetic Algorithm, Control Systems, Graphical User Interface.

SUMÁRIO

LISTA DE TABELAS	xii
LISTA DE FIGURAS	xiii
LISTA DE SIGLAS, SÍMBOLOS E ABREVIATURAS	xv
CAPÍTULO 1 - INTRODUÇÃO	18
1.1 MOTIVAÇÃO E RELEVÂNCIA	19
1.2 OBJETIVOS	20
1.3 METODOLOGIA	21
1.4 ORGANIZAÇÃO DO TRABALHO	21
CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA	23
2.1 INTELIGÊNCIA COMPUTACIONAL	23
2.1.1 Algoritmos Genéticos	25
2.1.1.1 População inicial.....	29
2.1.1.2 Decodificação	29
2.1.1.3 Avaliação	31
2.1.1.4 Seleção	32
2.1.1.5 Cruzamento.....	36
2.1.1.6 Mutação	37
2.1.1.7 Elitismo.....	37
2.2 SISTEMAS DE CONTROLE.....	38
2.2.1 Projeto de controladores	41
2.2.2 Controlador PID	41
2.3 FERRAMENTAS COMPUTACIONAIS: MATLAB® E SIMULINK®.....	46
CAPÍTULO 3 - FERRAMENTA COMPUTACIONAL DESENVOLVIDA	49
3.1 SISTEMA DO SIMULINK	53
3.2 CONTROLADOR	53

3.3 MÉTODO DE AJUSTE.....	54
3.4 PARÂMETROS DO ALGORITMO GENÉTICO	55
3.5 PARÂMETROS DO CONTROLADOR.....	56
3.6 GANHOS (AJUSTE MANUAL)	57
3.7 GRÁFICO DA RESPOSTA DO SISTEMA CONTROLADO.....	57
3.8 VALORES DOS GANHOS AJUSTADOS.....	58
3.9 INFORMAÇÕES	59
CAPÍTULO 4 - RESULTADOS E DISCUSSÕES	60
4.1 CONTROLE DE NÍVEL EM SISTEMA DE TANQUES ACOPLADOS.....	60
4.2 CONTROLE DE TENSÃO DE EXCITAÇÃO EM GERADOR SÍNCRONO ...	64
CAPÍTULO 5 - CONCLUSÕES E TRABALHOS FUTUROS	68
REFERÊNCIAS BIBLIOGRÁFICAS	70

LISTA DE TABELAS

Tabela 1 - Analogia do vocabulário do GA.....	27
Tabela 2 - Valores inteiro e decodificado de cada indivíduo (cromossomo binário).....	30
Tabela 3 - Valores de <i>fitness</i> e probabilidade de seleção de cada indivíduo (cromossomo binário) para seleção do tipo roleta	33
Tabela 4 - Novos valores de <i>fitness</i> e probabilidade de seleção de cada indivíduo (cromossomo binário) para seleção do tipo classificatória	34
Tabela 5 - Parâmetros do GACT para controle em sistema de tanques acoplados	61
Tabela 6 - Comparação das respostas ao degrau unitário.....	64
Tabela 7 - Parâmetros do GACT para controle no sistema do gerador síncrono	65
Tabela 8 - Comparação das respostas ao degrau unitário.....	67

LISTA DE FIGURAS

Figura 1 - Temas da Inteligência Computacional relacionados no diagrama de <i>Venn</i> ...	24
Figura 2 - Classificação dos métodos de otimização.....	26
Figura 3 - Fluxograma do GA da ferramenta computacional proposta.....	28
Figura 4 - Representação da população binária.....	29
Figura 5 - Exemplo dos valores posicionais para cálculo do valor de X_b	30
Figura 6 - Representação da roleta.....	33
Figura 7 - Situação da “nova roleta” depois da Classificação.....	35
Figura 8 - Cruzamento com (a) um ponto de corte e (b) dois pontos de corte.....	36
Figura 9 - Mutação do Indivíduo #02 (a) em apenas um bit e (b) no cromossomo completo.....	37
Figura 10 - Sistema de controle de (a) malha aberta e (b) malha fechada.....	39
Figura 11 - Exemplo de (a) diagrama de blocos e (b) gráfico da resposta ao degrau através do software MATLAB® SIMULINK®.....	40
Figura 12 - Representação do controlador PID em um sistema de realimentação unitária.....	42
Figura 13 - Representação no domínio da frequência do sistema de controle da Figura 11.....	42
Figura 14 - Exemplo de sistema de controle utilizado para análise da ação de cada ganho do controlador PID.....	43
Figura 15 - Gráfico da análise da ação do ganho proporcional, com $K_P=1$, $K_P=5$ e $K_P=50$	43
Figura 16 - Gráfico da análise da ação do ganho derivativo, para K_P fixo e com $K_D=0$, $K_D=5$ e $K_D=50$	44
Figura 17 - Gráfico da análise da ação do ganho integral, para K_P e K_D fixos e com $K_I=0$, $K_I=5$ e $K_I=50$	44
Figura 18 - Gráfico da análise da ação do ganho integral, para K_P e K_D fixos e com $K_I=0$, $K_I=5$ e $K_I=50$, no período entre 9 e 10 segundos.....	45
Figura 19 - Gráfico da análise da ação do ganho integral, para K_P e K_D fixos e com $K_I=0$, $K_I=5$ e $K_I=50$, no período entre 0 e 0,4 segundo.....	45
Figura 20 - Janela do software MATLAB® GUIDE.....	47

Figura 21 - Janela resultante da compilação.....	48
Figura 22 - Implementação do GA desenvolvido sem interface gráfica interativa.....	50
Figura 23 - Implantação do GA desenvolvido com interface gráfica interativa proposta	50
Figura 24 - Comunicação/relação entre as duas aplicações, MATLAB® e SIMULINK®, usado no método proposto	51
Figura 25 - GACT com as 09 (nove) funcionalidades destacadas	52
Figura 26 - Box "Sistema do Simulink"	53
Figura 27 - Box "Controlador"	54
Figura 28 - Box "Método de Ajuste"	54
Figura 29 - Box "Parâmetros do Algoritmo Genético"	55
Figura 30 - Janela informativa sobre o parâmetro "Taxa de Mutação"	56
Figura 31 - Box "Parâmetros do Controlador"	56
Figura 32 - Box "Ganhos (ajuste manual)"	57
Figura 33 - Box "Gráfico da Resposta do Sistema Controlado"	58
Figura 34 - Box "Valores dos Ganhos Ajustados"	58
Figura 35 - Botões de informações.....	59
Figura 36 - Janela informativa “Sobre” o programa.....	59
Figura 37 - (a) Sistema de tanques de acoplamento desenvolvido por Quanser Inc. (b) Representação esquemática do sistema.	60
Figura 38 - Modelo utilizado no MATLAB® SIMULINK® proposto por Quanser Inc. (2008).....	62
Figura 39 - Comparação entre controlador PI sintonizado pelo GA e controlador PI clássico.....	62
Figura 40 - Resposta ao degrau do controle PI clássico.....	63
Figura 41 - Resposta ao degrau do controle PI ajustado com o GACT	63
Figura 42 - Modelo no SIMULINK® do sistema AVR, juntamente com o controlador PID.....	65
Figura 43 - Gráfico comparativo da resposta ao degrau unitário do sistema controlado por ZN e GACT	66

LISTA DE SIGLAS, SÍMBOLOS E ABREVIATURAS

<i>corte</i>	Local do último bit antes do ponto de corte
DNA	Acido Desoxirribonucleico, do inglês <i>DeoxyriboNucleic Acid</i> .
$E(s)$	Transformada de laplace do sinal de erro
$e(t)$	Sinal do erro, diferença entre os sinais de saída e de referência
f_i	Valor da <i>fitness</i> (aptidão) do indivíduo i
f_i'	Valor da <i>fitness</i> (aptidão) do indivíduo i usando método da seleção classificatória
$G(s)$	Função de transferência de um bloco ou sistema
GA	Algoritmo Genético, do inglês <i>Genetic Algorithm</i>
GACT	<i>Genetic Algorithm Control Tool</i>
GUI	Interface Gráfica do Usuário, do inglês <i>Graphical User Interface</i>
i	Índice do indivíduo
IA	Inteligência Artificial
IAE	Integral do Erro Absoluto, do inglês <i>Integral of Absolute Error</i>
IC	Inteligência Computacional
IDS	Índice de Desempenho do Sistema
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISE	Integral do Erro ao Quadrado, do inglês <i>Integral of Squared Error</i>
ITAE	Integral do Erro Absoluto ponderado no Tempo, do inglês <i>Integral of Time multiply Absolute Error</i>
ITSE	Integral do Erro ao Quadrado vezes o Tempo, do inglês <i>Integral of Time Multiply Squared Error</i>
k	Número mínimo de bits para determinada precisão na decodificação

K_D	Ganho Derivativo
K_I	Ganho Integral
K_P	Ganho Proporcional
MIMO	Entradas e Saídas Múltiplas, do inglês <i>Multiple Input, Multiple Output</i>
M_p	Máximo Sobressinal
n	Número de bits do cromossomo (<i>tam_crom</i>)
N	Número total de indivíduos na população (<i>tam_pop</i>)
p	Quantidade de casas decimais de precisão do indivíduo decodificado
p.u.	Por unidade
PI	Proporcional e Integral
p_i	Probabilidade do indivíduo i ser selecionado
p_i'	Probabilidade do indivíduo i ser selecionado usando método da seleção classificatória
PID	Proporcional, Integral e Derivativo
PPGSCA	Programa de Pós-Graduação em Sistemas de Comunicação e Automação
$R(s)$	Transformada de laplace do sinal de referência do sistema de controle
$r(t)$	Sinal de referência do sistema de controle
s	Variável complexa da transformada de Laplace
SISO	Entradas e Saídas Simples, do inglês <i>Single Input, Single Output</i>
<i>tam_crom</i>	Tamanho do cromossomo (n)
<i>tam_pop</i>	Tamanho da população, quantidade de indivíduos (N)
t_s	Tempo de Acomodação
$U(s)$	Transformada de laplace do sinal de controle
$u(t)$	Sinal de controle
UFERSA	Universidade Federal Rural do Semi-Árido

X_b	Valor decimal que representa o número binário do cromossomo
X_{max}	Valor máximo do intervalo (domínio) de busca do GA
X_{min}	Valor mínimo do intervalo (domínio) de busca do GA
X_R	Valor decodificado do cromossomo binário
$Y(s)$	Transformada de Laplace do sinal de saída do sistema de controle
$y(t)$	Sinal de saída do sistema de controle
ZN	Método de ajuste de Ziegler e Nichols
$\theta_i(s)$	Transformada de Laplace da variável de entrada de um bloco ou sistema
$\theta_o(s)$	Transformada de Laplace da variável de saída de um bloco ou sistema

Capítulo 1 - Introdução

O advento dos computadores e dos sistemas de controle resultou em grandes avanços no campo da ciência, a partir da explosão tecnológica do século XX. Assim, nos dias atuais, os sistemas de controle automático e computadores se fazem presentes durante todo o cotidiano humano. Estes sistemas e computadores agem como promovedores do progresso e do desenvolvimento, impulsionando a sociedade no século atual (D’AZZO e HOUPIS, 2003).

Desenvolvimentos tecnológicos nestas duas áreas tornaram possíveis diversas aplicações e impulsionam pesquisas em sistemas de controle e automação. Os sistemas de controle são norteados, notadamente, por duas linhas teóricas com abordagens diferentes: Controle Clássico com abordagem no domínio da frequência, contendo os controladores PID e compensadores Avanço-Atraso, entre outros; e o Controle Moderno com abordagem no domínio do tempo, contendo os controladores modelados no Espaço de Estados (TIANFIELD, 2005).

Nos últimos anos, os avanços em aplicações que envolvem controle automático, tem buscado formas para obtenção de processos de otimização do desempenho de sistemas dinâmicos (LO, WONG e RAD, 2006).

Tal otimização pode ser alcançada através das técnicas de Inteligência Computacional (IC). Sumathi e Paneerselvam (2010), definem que a IC estuda problemas para os quais não existem algoritmos eficientes, quer seja devido a impossibilidade de formulá-los ou devido à complexidade dos mesmos. Assim, a definição ampla é dada por: “IC é um ramo da ciência da computação que estuda problemas para os quais não existem algoritmos computacionais eficientes”.

Os organismos biológicos resolvem estes tipos de problemas todos os dias: extraíndo os significados a partir da percepção, compreendendo linguagens e resolvendo problemas de visão computacionais mal definidas, graças à adaptação evolutiva do cérebro no meio ambiente (SUMATHI e PANEERSELVAM, 2010).

1.1 Motivação e Relevância

É prudente que os sistemas de controle apresentem uma metodologia bem formulada, baseada nas teorias de controle clássico e moderno, apresentando diversas rotinas computacionais para consecução dos ajustes dos controladores (OGATA, 2010). No entanto, diversos trabalhos na literatura demonstram a superioridade da IC no desenvolvimento de soluções mais satisfatórias para obtenção do controle de sistemas dinâmicos (SUMATHI e PANEERSELVAM, 2010).

Dentre os possíveis controladores da literatura, o mais largamente utilizado na indústria é o controlador clássico PID (Proporcional, Integral, Derivativo). De acordo com Ramos *et al* (2005), de 70 a 90% das malhas de controle de processos industriais são do tipo PID e segundo Normey-Rico (2001) os sistemas de controle mais complexos tem o PID como elemento básico.

O êxito da operação de todos estes sistemas depende fundamentalmente da modelagem adequada e do correto projeto do controlador, além de simulações e testes (OGATA, 2010). Uma ferramenta computacional já consolidada para essa modelagem, simulação e análise é o MATLAB[®]/SIMULINK[®], bastante popular no ambiente acadêmico devido sua grande flexibilidade, precisão e facilidade de uso (FREITAS e FRANÇA, 2005).

Com isso, o MATLAB[®]/SIMULINK[®] se apresenta como ferramenta efetiva para o auxílio no projeto de sistemas de controle, desde um simples sistema massa-mola, até uma complexa planta industrial (MATHWORKS[®], 2013).

O MATLAB[®] apresenta diversas extensões (*Toolboxes*) para uma variedade de aplicações, dentre estas existem uma *Toolbox* para implementação de problemas em Redes Neurais Artificiais (*Neural Network Toolbox*), para Lógica Fuzzy (*Fuzzy Logic Toolbox*), outra para ajuste de controlador PID (*PIDtool*) e outra para otimização (*Optimization Tool*), onde é possível obter soluções utilizando GA.

A motivação do trabalho baseia-se nos seguintes pontos:

- Um sistema computacional é implementado para agir, durante o controle da planta em questão, tal qual o faria um especialista humano;
- As técnicas tradicionais de controle podem ter dificuldade e/ou impossibilidade de atingir níveis desejados de eficiência;

- Tal dificuldade e/ou impossibilidade é manifestada em sistemas cuja dinâmica é extremamente complexa, e/ou não completamente formalizada;
- Existe uma ampla literatura apontando benefícios de sistemas inteligentes;
- Os controladores clássicos PID estão presentes em mais de 70% das malhas de controle de processos industriais.
- A complexidade no desenvolvimento de um código para obtenção de um sistema inteligente específico para determinada aplicação inviabiliza o experimento de alguns usuários não totalmente familiarizados com o assunto.

Neste contexto, a ferramenta computacional desenvolvida neste trabalho foi concebida para atuar como uma *Toolbox* do *software* MATLAB[®]. Intitulada GACT (*Genetic Algorithm Control Tool*), essa ferramenta funciona para aplicação em sistemas de controle fundamentada na aplicação conjunta dos GAs com a teoria de controle clássico. O GACT baseia-se no avanço dos sistemas operacionais que deixaram de ser totalmente manuseados por linhas de comando (requerendo um maior nível técnico dos seus usuários), e passaram a ter uma interface gráfica interativa (simplificando a forma de manuseio).

1.2 Objetivos

O objetivo geral deste trabalho é fazer uso da Inteligência Computacional, por meio da implantação de Algoritmos Genético, em um programa com interface gráfica, possibilitando ao usuário desta *Toolbox* aplicar as técnicas do GA para projetar o controlador de um sistema modelado no SIMULINK[®].

Em adição, para cumprir o objetivo geral, foram elencados os seguintes objetivos específicos:

- Entender os sistemas computacionais dentro de uma perspectiva da Inteligência Computacional aplicada a sistemas de controle;
- Familiarizar-se com as metodologias e técnicas de desenvolvimento de sistemas inteligentes, mais precisamente através de GAs.
- Familiarizar-se e estabelecer comunicação entre as aplicações MATLAB[®] SIMULINK[®] e GUIDE.

- Aplicar os GAs e analisar a eficiência computacional fazendo as respectivas comparações com as técnicas tradicionais de controle.
- Comprovar a possibilidade de aplicação da técnica e da ferramenta proposta comparando os parâmetros obtidos com a literatura recente e das técnicas tradicionais de controle.

1.3 Metodologia

Para alcançar os objetivos específicos desse trabalho, algumas atividades foram realizadas durante o tempo de estudo, como:

- Pesquisar, de modo geral, sobre Inteligência Computacional embasando-se no uso de Algoritmos Genéticos para otimização de parâmetros em diversos problemas particulares, como por exemplo, busca de menor caminho entre dois pontos com restrições;
- Pesquisar, também de modo geral, os sistemas de controle constatando como são tratados os parâmetros envolvidos nesta problemática e como são desenvolvidas as tradicionais ferramentas computacionais utilizadas;
- Realizar um estudo sobre as aplicações MATLAB®, SIMULINK® e GUIDE afim de verificar como as mesmas enviam e recebem informações entre si dentro da perspectiva da implementação da *Toolbox* proposta.
- Elaborar um Algoritmo Genético para solução da problemática de sistemas de controle clássicos – ajuste dos valores dos ganhos do controlador que otimize a resposta (rápida e sem elevada superação de sinal) do sistema de controle.
- Comparar as respostas obtidas através do GA da *Toolbox* desenvolvida com as da literatura recente e das técnicas tradicionais de controle.

1.4 Organização do trabalho

A partir da metodologia adotada os trabalhos desenvolvidos são apresentados neste documento estruturado da seguinte maneira:

No Capítulo 2 é feita a revisão da literatura, expondo-se os conceitos relacionados à Inteligência Computacional e Sistemas de Controle, além de apresentar a origem do trabalho, discutir e comentar as informações já publicadas sobre o tema da dissertação assim como as ferramentas computacionais utilizadas nesses estudos.

No Capítulo 3 a ferramenta computacional (*Toolbox*) desenvolvida neste trabalho é apresentada com a descrição de cada uma de suas funcionalidades.

No Capítulo 4 apresentam-se os resultados e alguns comparativos entre as respostas de dois sistemas de controle diferentes (controle de nível de coluna d'água em uma planta de tanques acoplados e controle de tensão de excitação de um gerador síncrono).

No Capítulos 5 são apresentadas as conclusões baseadas nos resultados obtidos no Capítulo 4, verificando-se a eficiência da *Toolbox* desenvolvida na perspectiva de obter o ajuste dos ganhos que otimiza as respostas dos sistemas avaliados. Também são levantadas algumas considerações e elencadas sugestões para trabalhos futuros.

Capítulo 2 - Revisão Bibliográfica

Neste capítulo serão apresentados os conceitos e definições que fundamentam este trabalho, demonstrando a potencialidade e a importância da aplicação de técnicas de inteligência computacional para controle de sistemas de dinâmicos.

2.1 Inteligência Computacional

O estudo científico dos processos mentais e do comportamento do ser humano – Psicologia – define inteligência, em linhas gerais, como a capacidade de aprendizado, raciocínio, abstração de conceitos, reconhecimento, planejamento e compreensão de ideias e linguagens.

Em um ser humano este comportamento é responsável pela capacidade de pensar e raciocinar, aprender e compreender a partir da experiência, valer-se da razão para solucionar problemas, adquirir e aplicar conhecimentos, lidar com situações complexas ou desconcertantes, reagir pronta e eficazmente a situações novas, reconhecer a importância relativa de elementos de uma situação e ainda manipular informações ambíguas, incompletas ou errôneas (SUMATHI e PANEERSELVAM, 2010)

De acordo com sua origem etimológica, a palavra inteligência consiste da união de dois vocábulos latinos: “*inter*” (entre) e “*elligere*” (escolher). Sugere-se, assim, que ser inteligente implica ser capaz de saber efetuar escolhas.

Utilizando-se destas características que o distingue dos animais irracionais, o homem ousou na tentativa de conceber sistemas (máquinas) inteligentes capazes de adaptar-se a novas situações, raciocinar, entender relações entre fatos, descobrir significados, reconhecer a verdade e aprender com base em suas experiências.

De acordo com Bittencourt (2001) a Inteligência Artificial (IA) nasceu oficialmente em 1956 num workshop de verão na Faculdade de Dartmouth, nos Estados Unidos.

O termo foi empregado inicialmente por Marvin Minski e John McCarthy. Naquela ocasião, um grupo de dez cientistas de renome se reuniu para pesquisar teoria de autômatos e estudar como se processa a inteligência (BARRETO, 2001).

De acordo com Sumathi e Paneerselvam (2010), a Inteligência Computacional (IC) é o sucessor de inteligência artificial. IC baseia-se em algoritmos heurísticos, como em sistemas *fuzzy*, redes neurais e computação evolutiva. Além disso, a inteligência computacional também abrange técnicas que usam inteligência coletiva, fractais e teoria do caos, sistemas imunológicos artificiais, *wavelets*, etc. IC é uma combinação de aprendizado, adaptação e evolução usada para aplicações inteligentes e inovadoras. IC está intimamente associada com a computação suave de uma combinação de redes neurais artificiais, lógica *fuzzy* e algoritmos genéticos, sistemas conexionistas, como a inteligência artificial e cibernética.

A maioria dos sistemas inteligentes é comparada com analogia de sistemas biológicos. Os sistemas inteligentes realizam operações semelhantes aos seres humanos, como tarefas de controle, reconhecimento de padrões, ou a tomada de decisões (SUMATHI e PANEERSELVAM, 2010).

Ainda segundo Sumathi e Paneerselvam (2010), IC estuda problemas para os quais não existem algoritmos eficientes, quer seja devido à impossibilidade de formulá-los ou devido serem demasiadamente complexos e, assim, ineficazes para aplicações reais.

A *IEEE Computational Intelligence Society* define seus temas de interesse como redes neurais artificiais, sistemas nebulosos (*fuzzy*) e computação evolutiva, incluindo inteligência coletiva (*swarm intelligence*).

A Figura 1 ilustra como estes temas podem estar relacionados de forma a conceber-se sistemas híbridos.

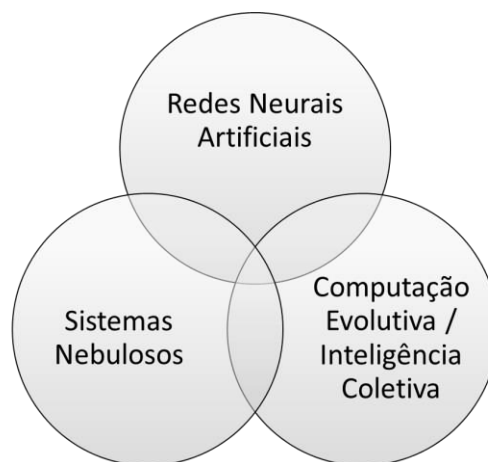


Figura 1 - Temas da Inteligência Computacional relacionados no diagrama de Venn

Fonte: Autoria própria, 2013.

As redes neurais artificiais estão associadas à capacidade de aprendizado, os sistemas nebulosos estão associados à capacidade de representação e a computação evolutiva juntamente a inteligência coletiva está associada à capacidade de otimização (IEEE COMPUTATIONAL INTELLIGENCE SOCIETY, 2013).

Na computação evolutiva, destacam-se os Algoritmos Genéticos – GA (do inglês *Genetic Algorithm*), uma técnica de programação que imita a evolução biológica como estratégia para resolver problemas.

2.1.1 Algoritmos Genéticos

De acordo com a qualidade requerida de soluções, existem duas grandes famílias de algoritmos de busca para problemas de otimização probabilística: Métodos exatos e Métodos aproximados (TALBI, 2002).

Esta primeira grande família tem como objetivo encontrar a melhor solução para o problema em questão e provar a sua otimização. A segunda tem como objetivo encontrar uma boa qualidade ou uma solução quase ótima para o problema abordado em um tempo razoável, explorando uma parte selecionada do espaço de soluções em que as soluções de boa qualidade são esperados (MEHDI, 2011).

Ao contrário dos métodos exatos, não há nenhuma garantia de que as soluções encontradas nos métodos aproximados sejam as melhores. Esta família de métodos de pesquisa também é composta de diferentes tipos de algoritmos classificados de acordo com sua estratégia de pesquisa: algoritmos de aproximação e algoritmos heurísticos. Os algoritmos de busca mais populares da classe de métodos heurísticos são as meta-heurísticas. Meta-heurísticas são métodos de otimização de uso geral que podem ser aplicadas a qualquer tipo de problemas, pois não contém nenhum conhecimento específico do problema em sua linha de projeto (BLUM e ROLI, 2003).

A Figura 2 apresenta como estão classificados esses algoritmos de otimização de acordo com Mehdi (2011). Os algoritmos genéticos pertence ao grupo dos algoritmos meta-heurísticos baseados em população (em destaque), e portanto tratam-se de métodos aproximados para solução de problemas.

É válido ressaltar que essa subjetividade, ou falta de precisão dos métodos heurísticos é tratada como uma particularidade comparada à inteligência humana, e não como uma deficiência ou ineficácia na solução do problemas.

Bueno (2009) destaca que muitas vezes, no cotidiano, diversos problemas são resolvidos sem total conhecimento e precisão sobre os mesmos. Alguns exemplos citados são: não preocupação com o tamanho exato do veículo e da vaga de estacionamento ao estacionar os mesmos; não conhecimento das propriedades do soluto e solvente ao adoçar uma bebida; não preocupação com as leis da física para auxiliar ou comprometer a ação ao tentar mover ou levantar um objeto. Nestas e em diversas outras situações, em detrimento de soluções comprovadamente ótimas e precisas, a melhor solução imediata – “heurística” – é encontrada e adotada, apresentando eficiência e relevante qualidade.

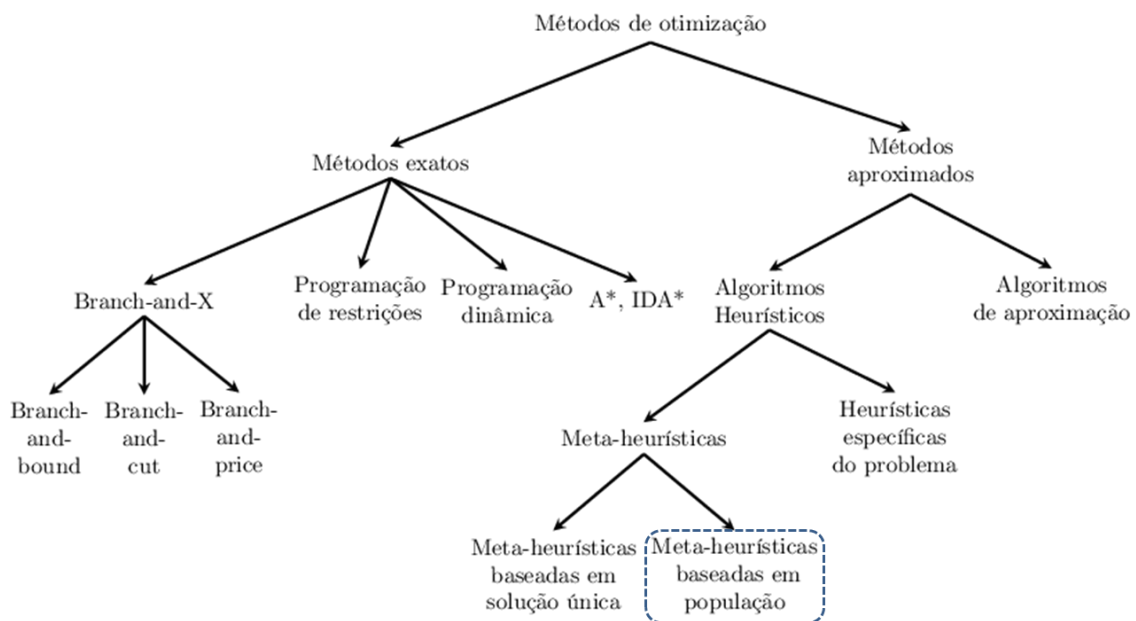


Figura 2 - Classificação dos métodos de otimização

Fonte: Adaptado de Talbi, 2002 *apud* Mehdi, 2013.

O Algoritmo Genético é uma importante técnica heurística baseada na teoria evolutiva de Charles Darwin, para controle e otimização de vários problemas em engenharia, por exemplo, os problemas que envolvem sistema de controle. Este método foi desenvolvido por John Holland, da Universidade de Michigan (HOLLAND, 1975) e, de acordo com Sumathi e Paneerselvam (2010), envolve uma população inicial de soluções possíveis para um problema que valendo-se de operadores genéticos e gerações sucessivas, produz indivíduos otimizados, devido o processo evolutivo proposto na Teoria de Evolução de Darwin.

Segundo esta teoria, a combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo indivíduo muito melhor adaptado às características de seu meio

ambiente ao combinar características possivelmente positivas de cada um dos reprodutores (GABRIEL e DELBEM, 2008).

Por ser um método que espelha-se na Genética Natural de Charles Darwin, prevalecendo os indivíduos (soluções) mais adaptados ao ambiente (função *fitness*), o GA herdou alguns termos da biologia. A Tabela 1 lista a analogia do vocabulário desses termos.

Tabela 1 - Analogia do vocabulário do GA

Genética Natural	Algoritmo Genético
Cromossomos	<i>Strings</i>
Genótipo (conjunto de cromossomos de um indivíduo)	Estruturas (conjuntos de <i>Strings</i>)
Fenótipo (corpo formado pela interação de carga genética para o meio ambiente)	Descodificar a estrutura de um conjunto de parâmetros que compõem uma solução particular
Genes (elementos que compõem os cromossomos são compostos de sequências de DNA)	Parte de uma sequência codificando um determinado parâmetro
Alelos (proteína que compreende o DNA)	Cada um dos bits que constituem a estrutura de cadeia
Ambiente	Função objetivo ou aptidão (<i>Fitness</i>)
Indivíduos	Solução

Fonte: Adaptado de Pacheco (1999)

Os princípios básicos do GA proposto na ferramenta computacional desenvolvida neste trabalho são mostrados na Figura 3, os detalhes dessas várias etapas envolvidas na aplicação de uma GA será descrita nos tópicos a seguir.

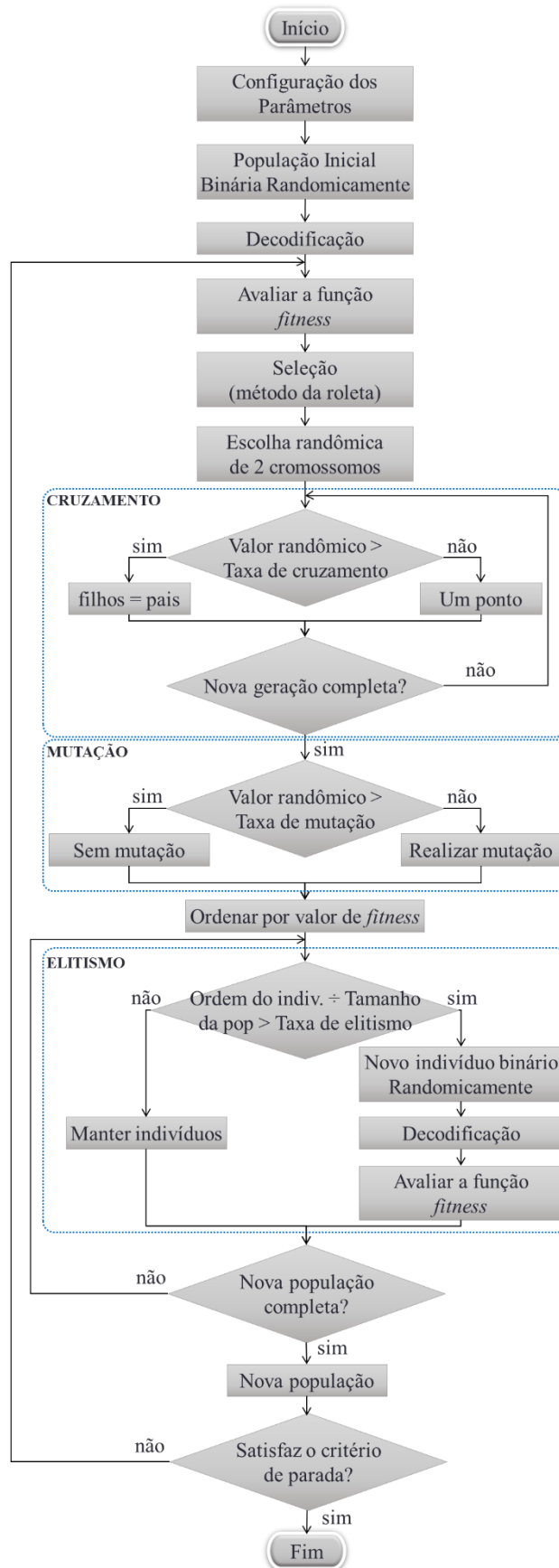


Figura 3 - Fluxograma do GA da ferramenta computacional proposta.

Fonte: Adaptado de Anh, Ahn e Nam, 2011.

2.1.1.1 População inicial

No GA, os indivíduos que representam as soluções para um problema específico podem ser representados por meio de cadeias binárias que consistem de zeros e uns, modelados segundo a maneira de cadeias de DNA de seres vivos orgânicos (BITTENCOURT, 2001).

O GA é principiado com a criação de uma população inicial de ‘*tam_pop*’ indivíduos, cada um representado por uma *string* contendo uma sequência aleatória de ‘*tam_crom*’ zeros e uns, conforme Figura 4.

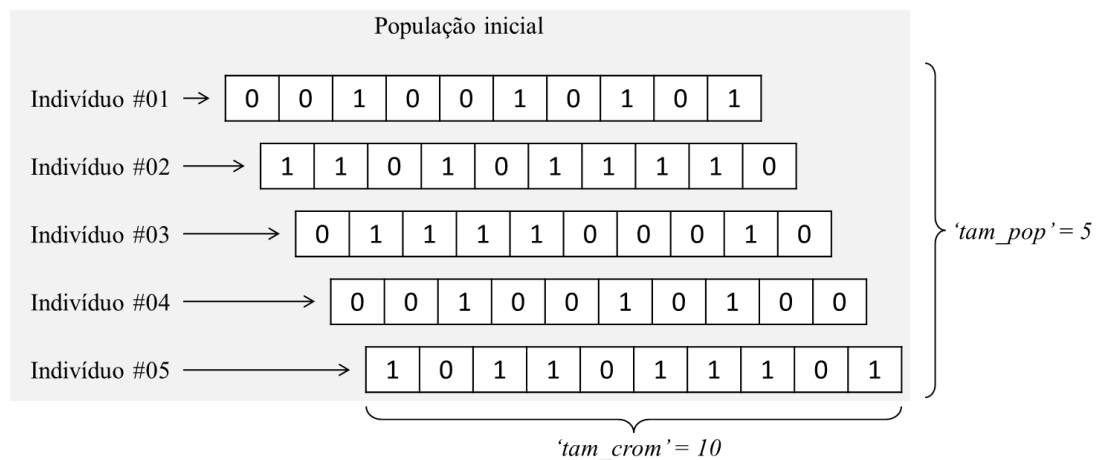


Figura 4 - Representação da população binária

Fonte: Autoria própria, 2013.

A representação das possíveis soluções para o espaço de busca de um problema define a estrutura do cromossomo a ser manipulada pelo algoritmo. Na Figura 4 cada um dos 05 (cinco) indivíduos da população inicial são representados por uma *string* de 10 (dez) dígitos binários (*bits*) de comprimento.

2.1.1.2 Decodificação

A *string* anteriormente citada recebe o nome de cromossomo que é decodificado a partir do domínio da variável real representada, conforme a equação (1).

$$X_R = X_b \cdot \frac{|X_{\min} - X_{\max}|}{2^n - 1} + X_{\min} \quad (1)$$

Onde X_R pertence ao intervalo $[X_{\min}, X_{\max}]$, X_b é o número inteiro que corresponde ao binário e n é o número de bits do cromossomo (PACHECO, 1999).

O valor de X_b é obtido através da conversão de binário para decimal. Tocci e Widmer (2011) definem o sistema de numeração binário como um sistema posicional em

que cada dígito binário (*bit*) possui um certo peso de acordo com a posição relativa ao *bit* da direita da *string*: o menos significativo (de menor peso ou LSB – do inglês *least significant bit*). Tomando o Indivíduo #01 como exemplo, temos a seguinte relação:

$$0010010101_2 = 0 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

$$0010010101_2 = 0 + 0 + 128 + 0 + 0 + 16 + 0 + 4 + 0 + 1 = 149_{10}$$

Aplicando os pesos ilustrados na Figura 5 em cada um dos bits, o valor decimal inteiro que o Indivíduo #01 representa é 149.

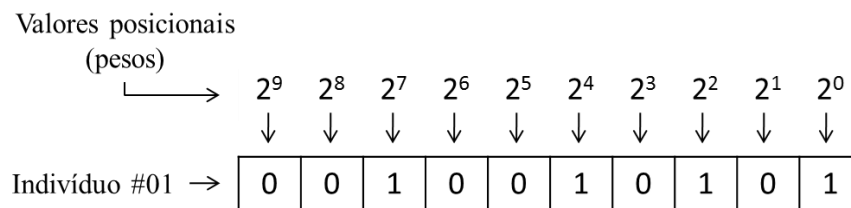


Figura 5 - Exemplo dos valores posicionais para cálculo do valor de X_b .

Fonte: Autoria própria, 2013.

Para um domínio da variável real representada no intervalo, de por exemplo, $X_{min}=0$ e $X_{max}=10$, o valor de X_R que representa o Indivíduo #01 da Figura 4 é calculado a partir da equação (1) como segue:

$$X_R = 149 \cdot \frac{|0-10|}{2^{10}-1} + 0$$

$$X_R = 149 \cdot \frac{10}{1023}$$

$$X_R = 1,4565$$

Desta forma cada um dos indivíduos podem ser igualmente decodificados e seus valores são expressos na Tabela 2.

Tabela 2 - Valores inteiro e decodificado de cada indivíduo (cromossomo binário)

Indivíduo	Cromossomo binário	Valor inteiro (X_b)	Valor decodificado (X_R)
#01	0010010101	149	1,4565
#02	1101011110	862	8,4261
#03	0111100010	482	4,7116
#04	0010010100	148	1,4467
#05	1011011101	733	7,1652

Fonte: Autoria própria, 2013.

No GA da ferramenta computacional proposta, X_{min} e X_{max} representam, respectivamente, os valores mínimos e máximos de cada um dos ganhos (K_P , K_I e K_D) do controlador clássico – PID.

Da tabela, é importante ressaltar a abreviada diferença entre os indivíduos #01 e #03. O valor inteiro que representa o Indivíduo #1 é 149 e o valor inteiro que representa o Indivíduo #03 é 148. Tais valores resultam numa diferença menor que 0,01 entre os valores codificados ($1,4565 - 1,4467 = 0,0098$), implicando numa precisão de apenas 02 (duas) casas decimais.

Este pequeno valor, também conhecido como resolução ou tamanho do degrau (por ser a diferença do valor convertido entre dois valores de X_b consecutivos, neste caso 148 e 149) implica na precisão desejada do valor decodificado.

O número mínimo de *bits* k para uma determinada precisão de p casas decimais pode ser obtido a partir da equação (2).

$$k \geq \log_2 \left((X_{max} - X_{min}) \cdot 10^p \right) \quad (2)$$

O próximo valor inteiro de k deve ser tomado como o tamanho do cromossomo ‘*tam_crom*’, de forma que o GA desta ferramenta computacional opere com p casas decimais de precisão (PACHECO, 1999). Por exemplo, para uma precisão de 3 casas decimais, tem-se:

$$k \geq \log_2 \left((10 - 0) \cdot 10^3 \right)$$

$$k \geq \log_2 \left(10^4 \right)$$

$$k \geq 13,2877$$

$$n = 14$$

Ou seja, seria necessário um cromossomo com 14 *bits* para que, no intervalo escolhido [0,10], os valores decodificados apresentem precisão de 3 casas decimais. Após decodificados, os indivíduos são submetidos ao processo de avaliação, como indicado no fluxograma da Figura 3.

2.1.1.3 Avaliação

A avaliação é feita através de uma função que melhor represente o problema e tem como objetivo fornecer uma medida de aptidão de cada indivíduo da população atual, que vai dirigir o processo de busca (SIVANANDAM e DEEPA, 2010).

Funções de avaliação ou aptidão são específicas para cada problema. A saber, em sistemas de controle, almeja-se obter uma resposta do sistema próximo da referência

estabelecida, além de minimizar o máximo sobressinal e o tempo de acomodação da curva de resposta.

Desta forma, em sistemas de controle, a função aptidão (ou no inglês *fitness*) pode atribuir diversas configurações a fim de minimizar os seguintes índices de desempenho:

- Integral do Erro ao Quadrado (ISE – do inglês *Integral of Squared Error*)
- Integral do Erro Absoluto (IAE – do inglês *Integral of Absolute Error*)
- Integral do Erro ao Quadrado vezes o Tempo (ITSE – do inglês *Integral of Time Multiply Squared Error*)
- Integral do Erro Absoluto ponderado no Tempo (ITAE – do inglês *Integral of Time multiply Absolute Error*)

Sendo:

$$ISE = \int e^2(t)dt \quad (3)$$

$$IAE = \int |e(t)|dt \quad (4)$$

$$ITSE = \int t \cdot e^2(t)dt \quad (5)$$

$$ITAE = \int t \cdot |e(t)|dt \quad (6)$$

Onde $e(t)$ é o sinal do erro, diferença entre o sinal de saída e o sinal de referência.

Além disto, em sistemas de controle requer-se atingir a referência rapidamente e sem superação excessiva, podendo-se, então, utilizar a *fitness* a fim de minimizar os valores de máximo sobressinal e tempo de acomodação.

O máximo sobressinal (Mp), é o valor máximo de pico da curva de resposta medido a partir da referência, e o tempo de acomodação (t_s) é o tempo necessário para que a curva de resposta atinja valores em uma faixa (comumente de 2% ou 5%) em torno do valor desejado (OGATA, 2010).

A avaliação pode ser tratada de forma análoga ao meio ambiente onde os seres vivos são postos a provar sua existência e perpetuar suas respectivas espécies. No GA, ela determina a probabilidade de cada solução (indivíduo) ser selecionada para reproduzir e perpetuar suas informações para as geração (iterações) futuras.

2.1.1.4 Seleção

O processo de seleção do algoritmo genético seleciona indivíduos para reprodução de acordo com a aptidão. Indivíduos mais aptos (com maior *fitness*) têm maior

probabilidade de serem selecionados para a reprodução (SIVANANDAM e DEEPA, 2010). Existem vários métodos de seleção nos GAs, dentre outros, destacam-se os seguintes:

- Seleção por roleta proporcional (*Roulette wheel selection*):

Método mais tradicional, fundamentada em uma busca linear através de uma “roleta” com as fatias da roda ponderados na proporção de valores de aptidão relativa de cada indivíduo (PACHECO, 1999). De tal modo que, se f_i é o valor da fitness do indivíduo i na população corrente, a probabilidade do indivíduo ser selecionado p_i é dada por (7):

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \tag{7}$$

Onde N é o número de indivíduos na população . A Figura 6 representa a roleta para uma fitness $f(x) = x^2$ (quadrado do valor decodificado), conforme dados da Tabela 3.

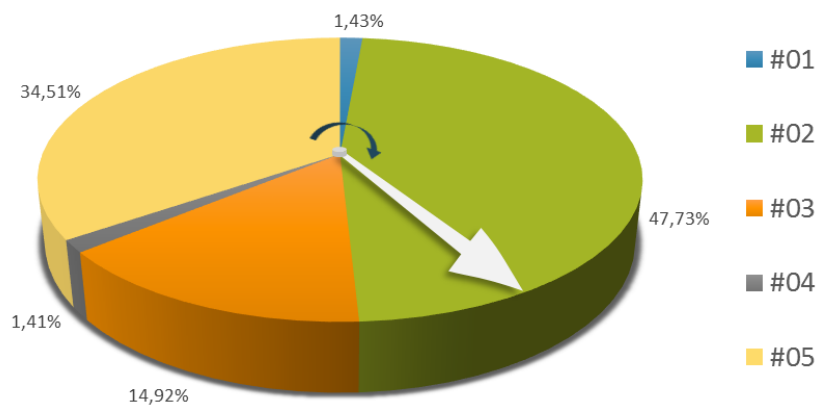


Figura 6 - Representação da roleta

Fonte: Autoria própria, 2013.

Tabela 3 - Valores de *fitness* e probabilidade de seleção de cada indivíduo (cromossomo binário) para seleção do tipo roleta

Indivíduo	Cromossomo binário	Valor decodificado (X_R)	Valor da fitness ($f_i = X_R^2$)	Valor da probabilidade (p_i)
#01	0010010101	1,4565	2,12	1,43 %
#02	1101011110	8,4261	70,99	47,73 %
#03	0111100010	4,7116	22,19	14,92 %
#04	0010010100	1,4467	2,09	1,41 %
#05	1011011101	7,1652	51,34	34,51 %
			148,75	100,00 %

Fonte: Autoria própria, 2013.

Desta forma, o indivíduo com maior probabilidade de ser selecionado é o #02, conforme representado na Figura 6 por uma “flecha” que gira sobre a roleta contendo fatias ponderadas de acordo com a equação (7). A probabilidade da flecha parar sobre o Indivíduo #02 é 47,73%, por exemplo.

- Seleção randômica (*Random selection*):

Neste tipo de seleção os indivíduos são selecionados aleatoriamente, desprezando o valor da *fitness* dos mesmo, tornando o GA mais rápido, porém na maioria das vezes ineficaz (SIVANANDAM e DEEPA, 2010).

- Seleção classificatória (*Rank selection*):

Este método é utilizado quando há grandes diferenças entre os valores de *fitness*. Por exemplo, se a melhor probabilidade p_i entre os indivíduos é 90% da soma de todas as adequações, então haverá soluções com chances muito baixas de serem selecionados.

O método consiste em classificar cada indivíduo de acordo com o valor da *fitness* e determinar um novo valor para essa adaptação como uma pontuação progressiva do pior ao melhor adaptado (SIVANANDAM e DEEPA, 2010).

Por exemplo, o pior terá *fitness* igual a 1, o segundo pior 2 e assim por diante, de forma que o melhor terá *fitness* igual a N (número de indivíduos na população). A Tabela 4 apresenta os novos valores de *fitness* f_i' para cada indivíduo ranqueados conforme método da seleção classificatória.

Tabela 4 - Novos valores de *fitness* e probabilidade de seleção de cada indivíduo (cromossomo binário) para seleção do tipo classificatória

Indivíduo	Cromossomo binário	Valor da <i>fitness</i> ($f_i = X_R^2$)	Novo valor da <i>fitness</i> (f_i')	Novo valor da probabilidade (p_i')
#01	0010010101	2,12 %	2	13,33 %
#02	1101011110	70,99 %	5	33,33 %
#03	0111100010	22,19 %	3	20,00 %
#04	0010010100	2,09 %	1	6,67 %
#05	1011011101	51,34 %	4	26,67 %
			15	100,00 %

Fonte: Autoria própria, 2013.

A partir desses novos valores de probabilidade p_i' é possível montar uma nova distribuição de probabilidade e verificar a diferença entre esta e a definida pelo método da roleta. A Figura 7 apresenta como a “nova roleta” estaria distribuída utilizando o método da seleção classificatória.

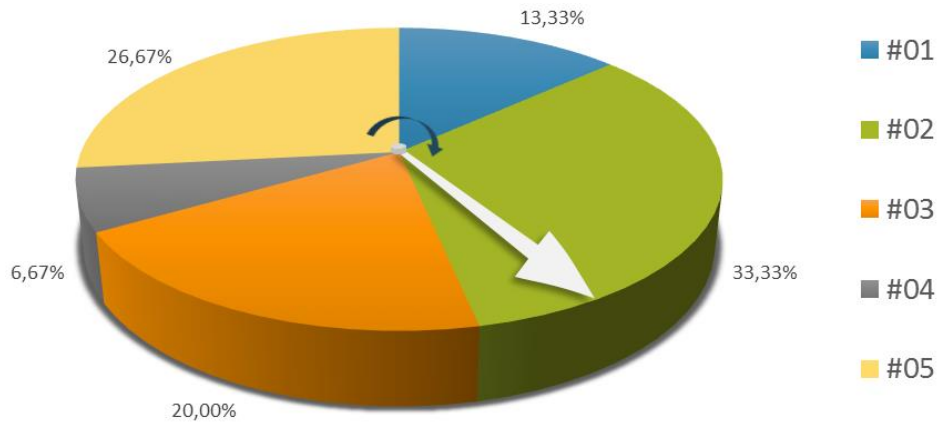


Figura 7 - Situação da “nova roleta” depois da Classificação

Fonte: Autoria própria, 2013.

Após a classificação os indivíduos apresentam valores de probabilidade mais próximos, diminuindo a superioridade do Indivíduo #03 e aumentando a probabilidade do Indivíduo #04 em quase 5 vezes, homogeneizando as probabilidades de seleção. Entretanto, este método pode resultar em menor convergência, porque os melhores indivíduos não se distinguem muito dos demais.

- Seleção por torneio (ou campeonato):

Neste tipo de seleção os indivíduos são escolhidos, dois a dois (ou em pequenos grupos), aleatoriamente, e os valores de *fitness* são comparados. O indivíduo com maior *fitness* entre os dois (ou entre o grupo) vence o “torneio” e é assumido como cromossomo “pai” que irá gerar novos indivíduos nos passos seguintes (SIVANANDAM e DEEPA, 2010).

Após o processo de seleção, por qualquer um dos métodos supracitados, os indivíduos escolhidos são enviados para o cruzamento e ao operador de mutação genética para obter nova geração. A seleção é uma etapa importante do GA, pois é responsável por garantir a sobrevivência dos melhores indivíduos (HUANG, 2011).

2.1.1.5 Cruzamento

O cruzamento é responsável por fornecer um novo arranjo para os indivíduos da população e é aplicado após a fase de seleção.

De acordo com o fluxograma da Figura 3, o cruzamento será ou não realizado dependendo do valor de sua taxa. Quando o valor sorteado aleatoriamente for inferior à taxa probabilística de cruzamento indica que os cromossomos dos pais (sequência de zeros e uns antiga) são divididos em duas partes. O filho (nova sequência de zeros e uns) recebe a combinação das partes dos dois pais (ANH, AHN e NAM, 2011).

Os valores mais usuais na literatura ficam em torno de 70 a 90%.

A Figura 8 ilustra como pode acontecer o cruzamento de 02 (dois) pais selecionados (indivíduos #02 e #05) de forma a gerar descendentes (filhos).

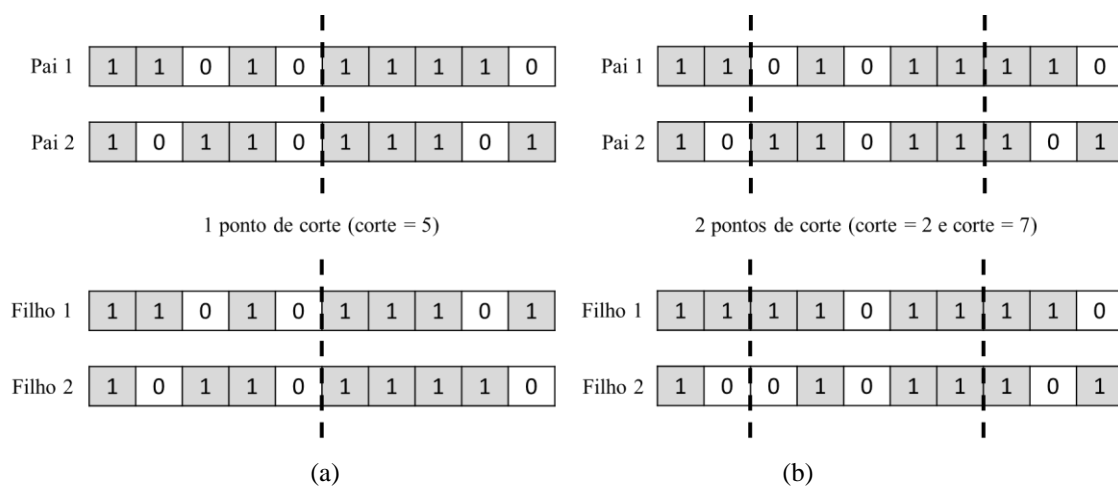


Figura 8 - Cruzamento com (a) um ponto de corte e (b) dois pontos de corte

Fonte: Autoria própria, 2013.

Conforme Figura 8, o cruzamento pode acontecer em apenas um ponto ou pode ter múltiplos pontos de corte. E ainda, as posições destes pontos podem ser definidas previamente ou aleatoriamente.

Para apenas um ponto de corte e com a posição definida de forma aleatória, é sorteado um valor entre 0 e *tam_crom* (o tamanho do cromossomo). Quando o valor 'corte' for igual a 1, apenas o primeiro bit de cada pai é mantido e os demais são trocados entre eles: do segundo ao último bit do 'Pai 1' une-se ao primeiro bit do 'Pai 2' e vice-versa. Quando o valor 'corte' for igual ao tamanho do cromossomo, as *strings* dos pais são mantidas e os filhos são idênticos aos pais.

2.1.1.6 Mutação

Após o cruzamento, pode-se também fazer a mutação dos cromossomos de maneira a alterar completamente as características, inserindo diversidade à população. Geralmente, na mutação cada '0' do cromossomo é substituído por '1' e vice-versa. Também é possível alterar os valores de parte do cromossomo de modo aleatório ou definido previamente (ANH, AHN e NAM, 2011).

De acordo com Huang (2011), esta etapa é responsável por fazer pequenas alterações para os bits do cromossomo devido mudanças aleatórias. Este operador genético é necessário porque insere diversidade à população e reduz as possibilidades de criar uma população de indivíduos iguais, em outras palavras, evita mínimos locais.

Os valores mais usuais na literatura ficam em torno de 1 a 2%. A Figura 9 ilustra como pode acontecer a mutação dos cromossomos.

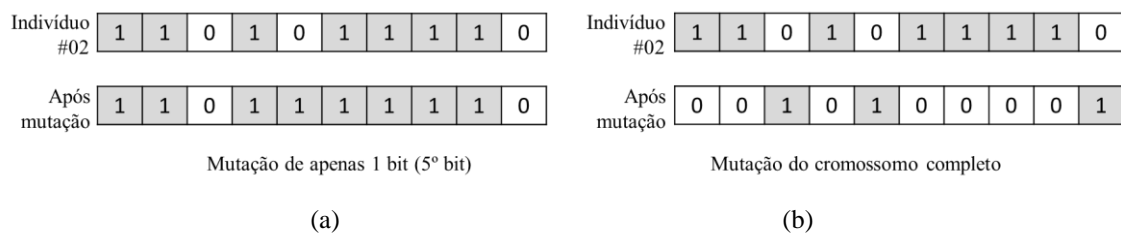


Figura 9 - Mutação do Indivíduo #02 (a) em apenas um bit e (b) no cromossomo completo

Fonte: Autoria própria, 2013.

Conforme Figura 9, a mutação pode acontecer em 01 (um) ou mais bits (no 5º bit, por exemplo) ou pode acontecer no cromossomo completo.

2.1.1.7 Elitismo

Nesta etapa, um grupo com os melhores indivíduos (elite) são selecionados de acordo com a taxa percentual de elitismo, levando-os para a próxima geração juntamente aos novos indivíduos que complementam a população (SIVANANDAM e DEEPA, 2010). Por exemplo, uma taxa de elitismo de 75% em uma população com 100 indivíduos indica que serão selecionados os 75 melhores e levados para a geração seguinte juntamente a outros 25 criados randomicamente.

Através de mecanismos naturais de busca do GA é possível encontrar uma solução para um problema específico de otimização. Então, é muito importante a correta definição de parâmetros como: função de aptidão (*fitness*); população; mecanismo de seleção dos pais; operadores de variação e reprodução; as taxas de recombinação e mutação. Assim,

é possível ter um GA suficientemente capaz de desenvolver um adequado mecanismo de busca, como citado por Sumathi e Paneerselvam (2010) e Peng et al. (2001).

2.2 Sistemas de Controle

Ogata (2010) afirma que o controle automático é essencial em qualquer campo da engenharia e da ciência. O controle automático é um componente importante e intrínseco em sistemas de veículos espaciais, sistemas robóticos, modernos sistemas de manufatura e quaisquer operações industriais que envolvam o controle de temperatura, pressão, umidade, viscosidade, vazão etc.

Conforme Tianfield (2005), cronologicamente, e de acordo com suas respectivas abordagens, os sistemas de controle são divididos em 02 (dois) grandes grupos:

- Controle Clássico: abordagem no domínio da frequência, aplicado a sistemas com uma entrada e uma saída – SISO (do inglês *Single Input, Single Output*), realizando a análise do sistema através de Diagramas de Bode, Critério de Nyquist, Critério de Routh-Hurwitz, Lugar das Raízes (Evans) e Margens de Ganho e Fase. Compõe os Controladores PID e Compensação Avanço-Atraso.
- Controle Moderno: abordagem no domínio do tempo, podendo ser aplicado a sistemas com mais de uma entrada e mais de uma saída – MIMO (do inglês *Multiple Input, Multiple Output*), realizando a análise do sistema através de Modelo por Espaço de Estados, Controlabilidade, Observabilidade, Processos Estocásticos.

Bolton (1995) compara sistema a uma caixa preta com uma entrada e uma saída, onde não se conhece o interior desta caixa, mas somente a relação entre a saída e a entrada. Desta forma, um sistema pode ser chamado ‘sistema de controle’ quando sua saída é controlada para assumir um valor particular ou seguir uma determinada entrada.

Já Maya e Leonardi (2010) definem sistema como um conjunto de elementos que se relacionam e que de alguma forma interagem uns com os outros. Em sua definição genérica, sistema de controle é um sistema que tenta impor às variáveis de saída um determinado comportamento, cuja referência está de alguma forma contida nas variáveis de entrada.

Ogata (2010) divide os sistemas de controle em 02 (dois) grupos distintos, quanto a característica da planta: sistema de controle de malha aberta e sistema de controle de malha fechada.

No sistema de controle de malha aberta o sinal da saída (ou resposta) não exerce nenhuma ação de controle, ou seja, não influencia sobre o sinal de entrada, conforme ilustrado na Figura 10a. Já no sistema de controle de malha fechada é observada a presença de um sinal de realimentação, que pode ser o próprio sinal de saída (ou uma função do sinal de saída e suas derivadas e/ou integrais). A diferença do sinal de entrada com tal sinal de realimentação produz o sinal do erro atuante, conforme ilustrado na Figura 10b (OGATA, 2010).

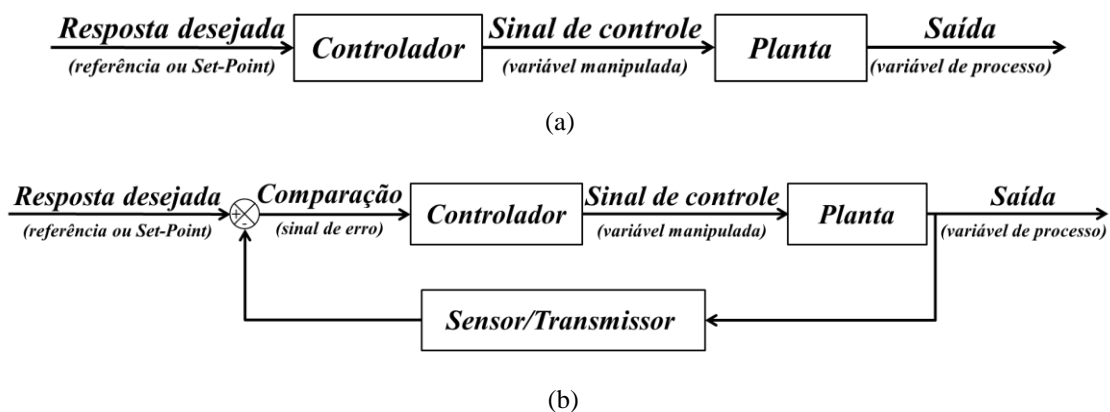


Figura 10 - Sistema de controle de (a) malha aberta e (b) malha fechada

Fonte: Adaptado de OGATA, 2010.

A Figura 10 apresenta os dois tipos de sistema de controle representados na forma de diagrama de blocos. Ogata (2010) apresenta os diagramas de blocos como uma representação gráfica das funções desempenhadas por cada componente de um sistema e do fluxo de sinais entre estes. Maya e Leonardi (2010) ratificam sobre a utilidade da representação de sistemas por meio de diagramas de blocos funcionais (descritivos ou qualitativos). Segundo eles, tais diagramas são uma extensão ou detalhamento da representação gráfica de um sistema por um retângulo com as variáveis de entrada e de saída indicadas por meio de setas.

Tal bloco funcional é definido por Ogata (2010) como um símbolo da operação matemática que é aplicada ao sinal de entrada do bloco que produz o sinal da saída. Nestes blocos, normalmente são incluídos as suas respectivas funções de transferência.

Bolton (1995) define a função de transferência de um sistema linear que descreve o comportamento dinâmico como a razão da transformada de Laplace da variável de saída

$\theta_o(s)$ pela transformada de Laplace da variável de entrada $\theta_i(s)$, supondo todas as condições iniciais nulas, conforme (8).

$$G(s) = \frac{\theta_o(s)}{\theta_i(s)} \quad (8)$$

A Figura 11a apresenta um diagrama de blocos de um sistema de malha aberta construído no software MATLAB® SIMULINK® contendo um sinal de entrada degrau aplicado a um bloco funcional que representa a função transferência ' $1/(s+1)$ ' e o bloco 'scope', responsável por mostrar o gráfico do sinal da saída (resposta ao degrau), mostrado na Figura 11b.

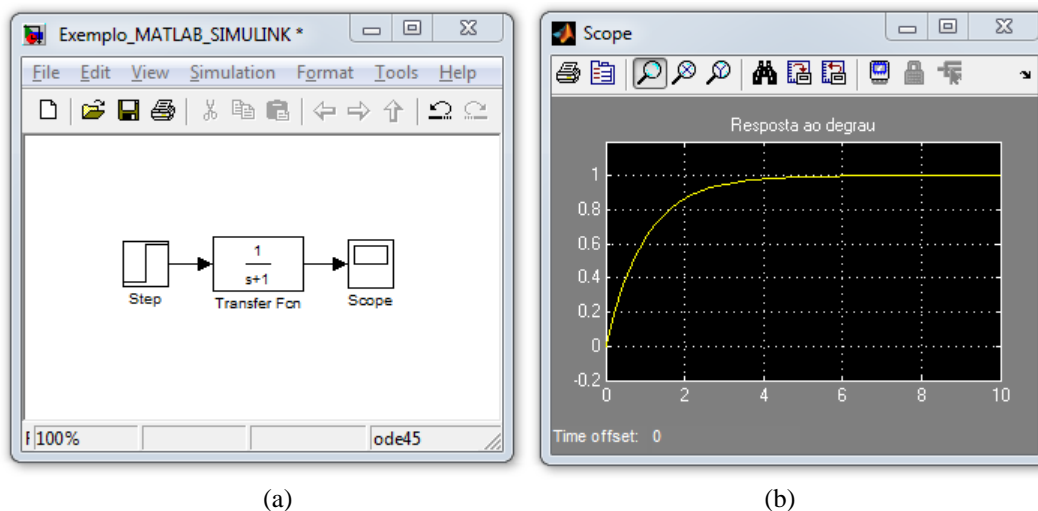


Figura 11 - Exemplo de (a) diagrama de blocos e (b) gráfico da resposta ao degrau através do software MATLAB® SIMULINK®.

Fonte: Adaptado de Matlab, 2011.

O software citado apresenta-se como uma ferramenta computacional importante para a modelagem, simulação e análise de sistemas dinâmicos.

É possível observar na Figura 11 que o sistema mesmo sem a presença de um controlador atinge a referência após aproximadamente 6 segundos. Isto acontece devido as características da função de transferência da planta. Em outros casos, para plantas diferentes, o sistema pode apresentar uma resposta que não convirja para a referência tão rapidamente, necessitando-se, então, de um controlador.

Para implantação deste controlador, após justificada sua necessidade, é indispensável que este seja devidamente projetado para atender as especificações desejadas. Em linhas gerais, deseja-se que o sistema atinja a referência rapidamente e sem superação excessiva.

2.2.1 Projeto de controladores

O projeto dos controladores pode ser concebido de várias formas diferente. Da literatura é possível citar os seguintes métodos: Tentativa e erro (ajuste manual), Métodos de sintonia de Ziegler-Nichols, Cohen e Coon, Método da Integral do Erro, Lugar das raízes, Alocação de polos, Domínio da Frequência, Espaço de Estados, Lógica *Fuzzy*, Algoritmos Genéticos, entre outros.

De acordo com a configuração do sistema e utilizando-se uma das abordagens de sistemas de controle (Clássico ou Moderno) é possível conceber um controlador para a planta em questão. Contudo, é razoável utilizar ferramentas computacionais para obter êxito no projeto de controladores.

2.2.2 Controlador PID

É prudente que o controlador serve para corrigir o sinal do erro – diferença entre a referência estabelecida e o sinal de saída da planta. O controlador Proporcional Integral Derivativo (PID) convencional, conforme citado no tópico 1.1, apresenta elevada aplicação industrial.

Tal controlador, ilustrado na Figura 12, baseia-se na ação de três ganhos: na ação *P* a correção acontece de forma proporcional ao erro, de modo que o sinal de correção deve crescer na proporção do erro entre o valor real e o desejado; na ação *I* a correção é proporcional ao produto erro \times (vezes) tempo, representando uma ação mais intensa à erros pequenos, porém duráveis por longo período de tempo; na ação *D* a correção acontece proporcionalmente à taxa de variação do erro, de tal sorte que se o erro variar muito rápido, esta taxa de variação deve ser reduzida para evitar oscilações (BAZANELLA e SILVA, 2005).

A Figura 12 ilustra um controlador PID aplicado em um sistema de controle por realimentação unitária.

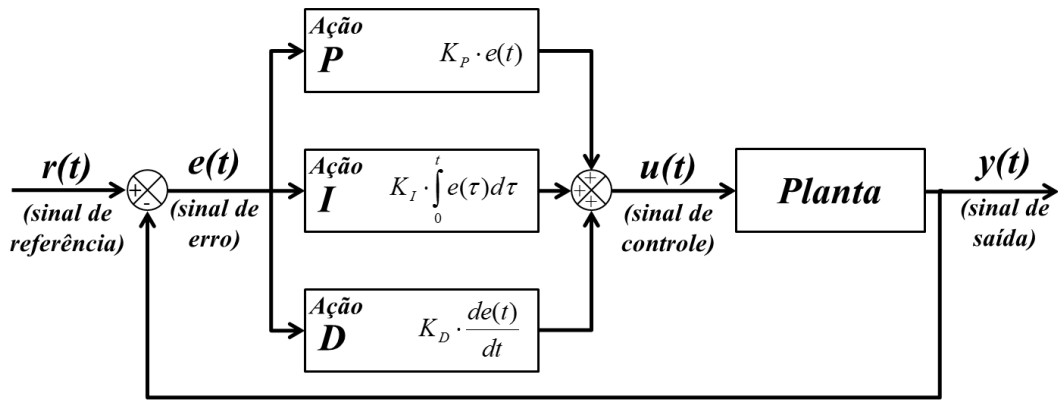


Figura 12 - Representação do controlador PID em um sistema de realimentação unitária

Fonte: Autoria própria, 2013.

Conforme Figura 12, o sinal de controle compreende o somatório das 03 (três) ações sobre o erro. Dessa forma, este sinal $u(t)$ é dado pela equação (9):

$$u(t) = K_p \cdot e(t) + K_I \cdot \int_0^t e(\tau) d\tau + K_D \cdot \frac{d}{dt} e(t) \quad (9)$$

Aplicando a transformada de Laplace supondo todas as condições iniciais nulas, esse somatório, no domínio da frequência, fica conforme (10):

$$U(s) = K_p \cdot E(s) + K_I \cdot E(s) \cdot \frac{1}{s} + K_D \cdot E(s) \cdot s \quad (10)$$

De modo que a função de transferência do controlador – relação entre a variável de saída e a de entrada pode ser definido pela equação (11):

$$\frac{U(s)}{E(s)} = K_p + K_I \cdot \frac{1}{s} + K_D \cdot s \quad (11)$$

Ou ainda, por (12) – resolvendo o mínimo múltiplo comum e rearranjando os termos:

$$\frac{U(s)}{E(s)} = \frac{K_D \cdot s^2 + K_p \cdot s + K_I}{s} \quad (12)$$

De tal forma que o sistema de controle da Figura 12 pode ser representado no domínio da frequência pelo diagrama de bloco da Figura 13.

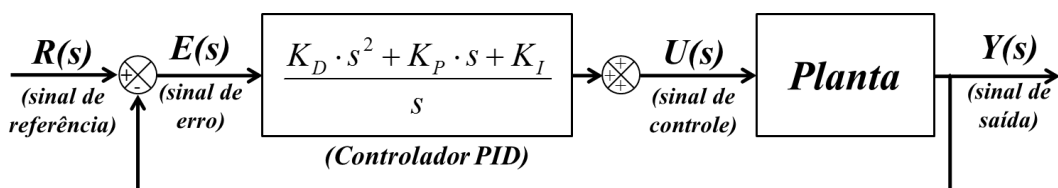


Figura 13 - Representação no domínio da frequência do sistema de controle da Figura 12.

Fonte: Autoria própria, 2013.

Nos gráficos das próximas 05 (cinco) figuras é realizada uma análise da ação dos três ganhos do controlador PID para a planta com a função de transferência conforme ilustrado na Figura 14.

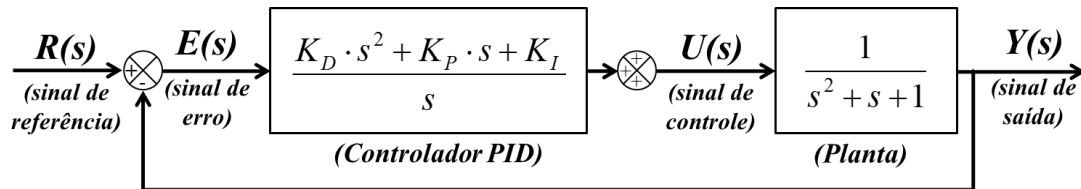


Figura 14 - Exemplo de sistema de controle utilizado para análise da ação de cada ganho do controlador PID.

Fonte: Autoria própria, 2013.

Primeiramente verifica-se a ação proporcional isoladamente. Na Figura 15 constata-se que a medida que o ganho proporcional aumenta de $K_P=1$ até $K_P=50$, mantendo-se K_I e K_D nulos, o valor da elevação máxima de sobressinal (*overshoot*) aumenta expressivamente, chegando a um sobressinal de quase 80%, e em contrapartida o tempo em que o sinal de resposta atinge a referência pela primeira vez (tempo de subida) diminui de maneira significativa.

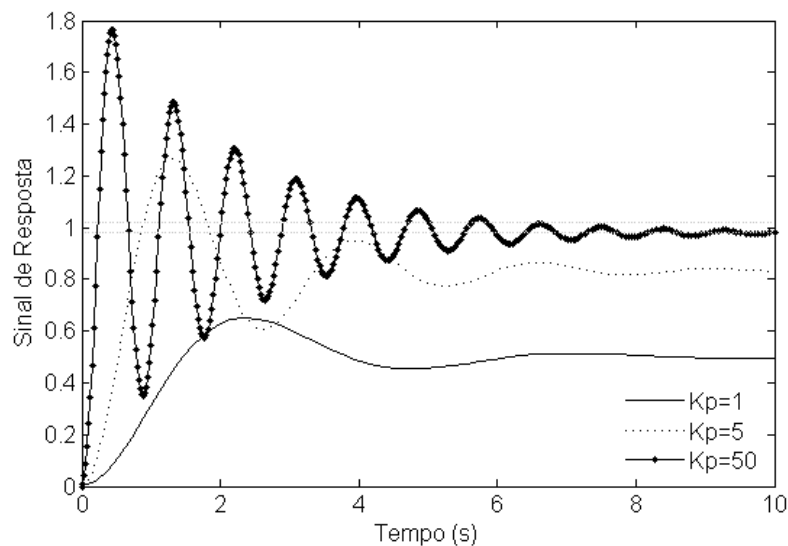


Figura 15 - Gráfico da análise da ação do ganho proporcional, com $K_P=1$, $K_P=5$ e $K_P=50$.

Fonte: Autoria própria, 2013.

Optando-se pelo valor que mais se aproximou da referência, $K_P=50$, variar-se-á no gráfico da Figura 16 os valores de K_D , mantendo-se K_I nulo e K_P neste valor constante ($K_P=50$).

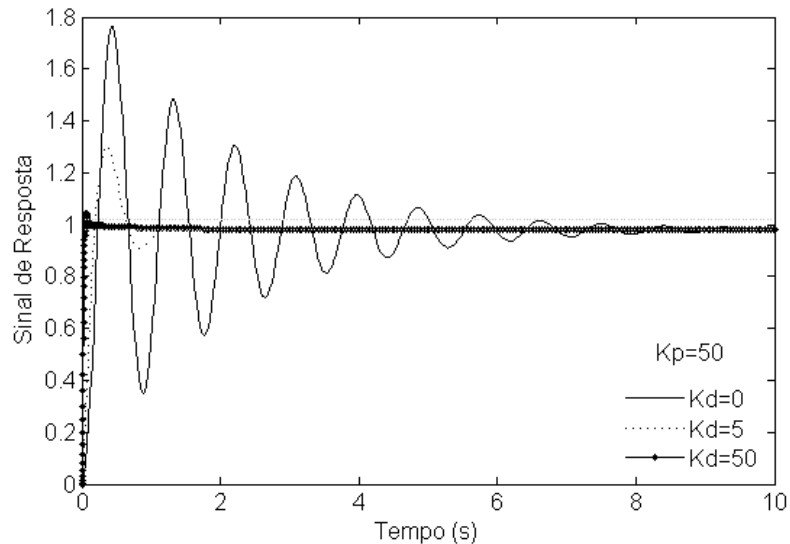


Figura 16 - Gráfico da análise da ação do ganho derivativo, para K_P fixo e com $K_D=0$, $K_D=5$ e $K_D=50$.

Fonte: Autoria própria, 2013.

A medida que ganho derivativo aumenta de $K_D=0$ até $K_D=50$ o valor do sobressinal é minimizado, e o sinal de resposta acomoda-se mais rapidamente entre $\pm 2\%$ do sinal de referência, porém com um pequeno valor de erro em regime permanente – diferença entre o sinal de saída e o de referência para um tempo relativamente alto.

Novamente, escolhendo o valor que mais se aproximou da referência, $K_P=50$ e $K_D=50$, variar-se-á no gráfico da Figura 16 os valores de K_I , mantendo-se K_P e K_D constantes ($K_P=50$ e $K_D=50$).

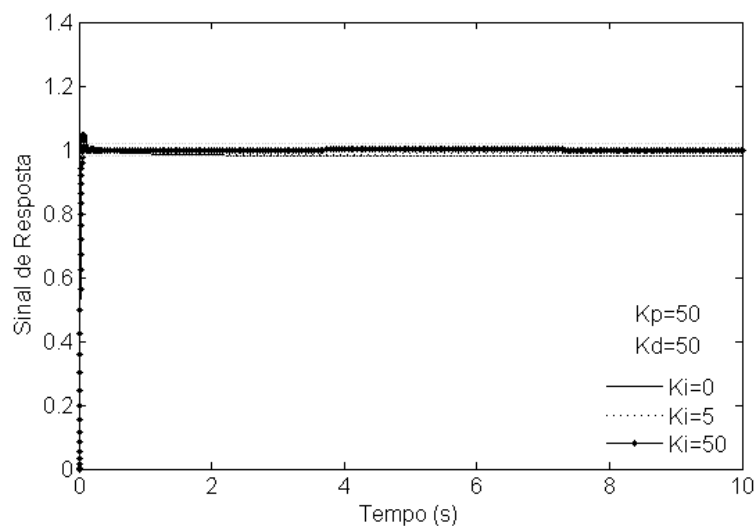


Figura 17 - Gráfico da análise da ação do ganho integral, para K_P e K_D fixos e com $K_I=0$, $K_I=5$ e $K_I=50$.

Fonte: Autoria própria, 2013.

A medida que o ganho integrativo aumenta de $K_I=0$ até $K_I=50$ o erro de regime permanente é minimizado. A Figura 18 apresenta a visualização do último segundo de

duração da simulação (entre 9 e 10 segundos) da resposta acima, confirmando que para $K_I=50$ o erro de regime permanente é nulo.

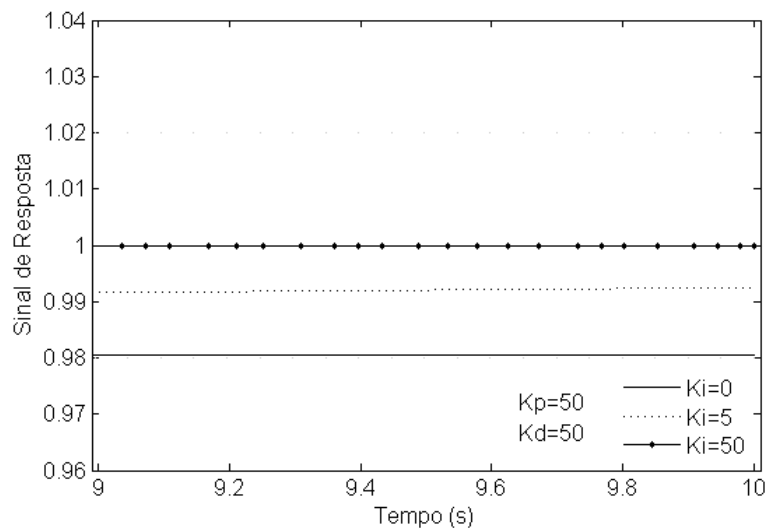


Figura 18 - Gráfico da análise da ação do ganho integral, para K_P e K_D fixos e com $K_I=0$, $K_I=5$ e $K_I=50$, no período entre 9 e 10 segundos.

Fonte: Autoria própria, 2013.

A Figura 19 apresenta o valor do máximo sobressinal obtido, mostrando o resultado da ação dos ganhos durante o período entre 0 e 0,4 segundo.

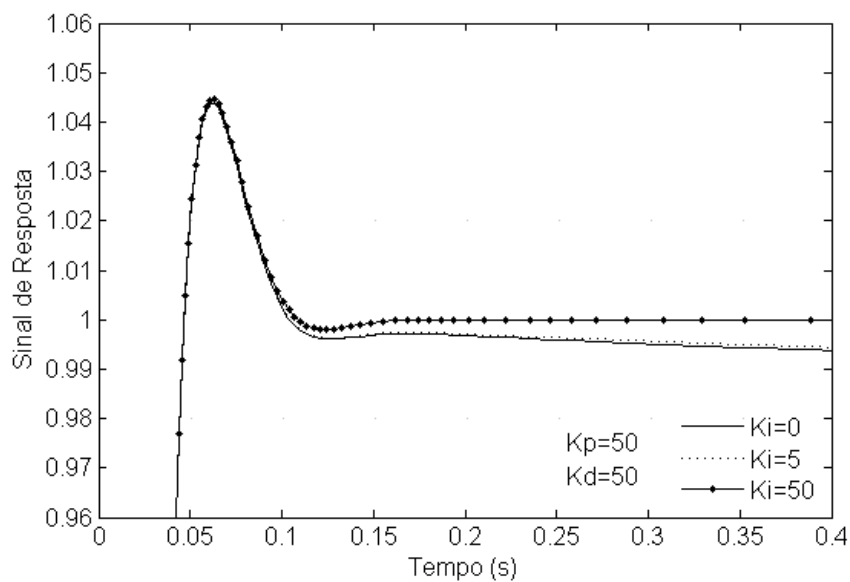


Figura 19 - Gráfico da análise da ação do ganho integral, para K_P e K_D fixos e com $K_I=0$, $K_I=5$ e $K_I=50$, no período entre 0 e 0,4 segundo.

Fonte: Autoria própria, 2013.

O valor do máximo sobressinal, neste exemplo, usando $K_P=50$, $K_D=50$ e $K_I=50$ é aproximadamente 4,45%.

Desta forma, conclui-se que a elevação excessiva do sinal, o erro de regime permanente e os tempos de subida e acomodação estão relacionados com os valores dos ganhos PID. A ação P diminui o tempo de subida, a ação D reduz o máximo sobressinal e o tempo de acomodação e a ação I tende a eliminar o erro de regime permanente.

A singularidade existente nesta relação motiva a necessidade da utilização de ferramentas computacionais para o ajuste mais adequado dos ganhos.

2.3 Ferramentas computacionais: MATLAB® e SIMULINK®

Em seu portal corporativo, *MathWorks*® (2013) define MATLAB® como um ambiente interativo para computação numérica, visualização e programação. Com o *software* MATLAB® é possível analisar os dados, desenvolver algoritmos e criar modelos e aplicações. A linguagem, ferramentas, e as funções matemáticas embutidas permite explorar múltiplas abordagens e chegar a uma solução mais rápida do que com planilhas ou linguagens de programação tradicionais, como C/C++ ou Java®. Além disso, é possível utilizar o MATLAB® para uma variedade de aplicações, incluindo processamento de sinais e comunicação, processamento de imagem e vídeo, sistemas de controle, teste e medição, finanças computacionais e biologia computacional (OGATA, 2008). Ainda segundo a empresa, mais de um milhão de engenheiros e cientistas na indústria e na academia usam o *software* MATLAB®.

Integrado ao MATLAB®, existe um ambiente de diagrama de blocos para a simulação de sistemas – o SIMULINK® – permitindo incorporar algoritmos MATLAB® em modelos e exportar os resultados de simulação para MATLAB® para análise posterior. O SIMULINK® suporta o projeto em nível de sistema, simulação, geração automática de código, teste contínuo e verificação de sistemas embarcados. SIMULINK® fornece ainda um editor gráfico, bibliotecas de blocos personalizáveis e solucionadores de modelagem para simulação de sistemas dinâmicos.

Ao contrário do MATLAB®, que utiliza linha de comando, o SIMULINK® utiliza uma interface gráfica, representando o sistema por diagramas de blocos, no qual cada bloco representa sua respectiva função de transferência. É válido ressaltar que apesar do SIMULINK® ser uma aplicação específica, este não trabalha independentemente do MATLAB®.

Uma terceira ferramenta computacional relevante para o presente trabalho é o GUIDE, ambiente de desenvolvimento de interface gráfica do usuário no MATLAB®, equipado com um conjunto de ferramentas para a criação de interfaces gráficas de usuário (GUIs, sigla do inglês *Graphical User Interfaces*). Estas ferramentas simplificam o processo de programação e arranjo das GUIs.

A GUI é uma representação gráfica de uma ou mais janelas que contêm controles, chamados de componentes, que permitem ao usuário realizar tarefas interativas. O usuário do GUI não precisa criar um script ou digitar comandos na linha de comando para realizar as tarefas. Ao contrário dos programas de codificação para realizar tarefas, o usuário de uma GUI não precisa entender os detalhes de como as tarefas são executadas.

Componentes GUI pode incluir menus, barras de ferramentas, botões, caixas de listagem e deslizantes, entre outros. As GUIs criadas usando ferramentas MATLAB® também pode realizar qualquer tipo de cálculo, ler e escrever arquivos de dados, comunicar com outros GUIs, e exibir dados como tabelas e/ou gráficos (MATHWORKS®, 2013).

A Figura 20 apresenta a janela do GUIDE contendo um exemplo da criação de uma GUI.

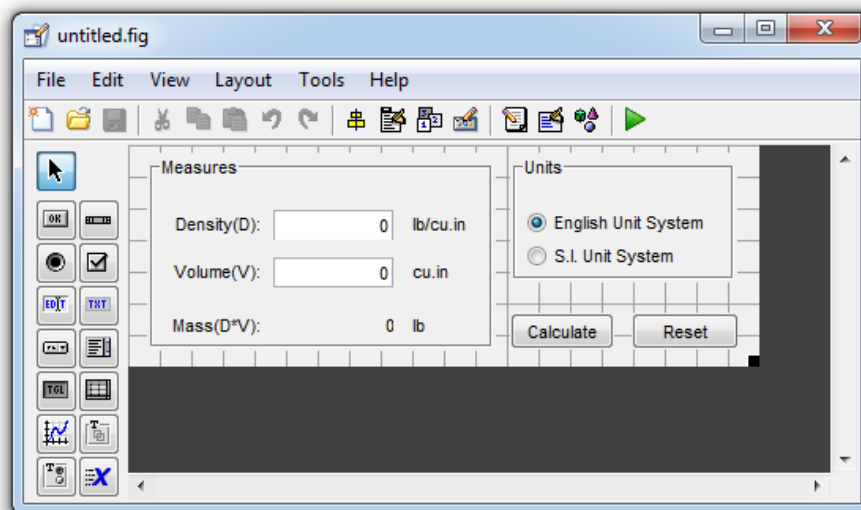


Figura 20 - Janela do software MATLAB® GUIDE

Fonte: Adaptado de Matlab, 2011.

Do lado esquerdo há 14 (catorze) opções de componentes que podem ser inseridos e editados na janela deste *software*. Além da escolha das características destes componentes é possível estabelecer uma programação para que determinada ação aconteça ao clicar ou editar o respectivo componente. Tal programação é realizada no

MATLAB[®] e o resultado da compilação deste código criado é a janela mostrada na Figura 21.

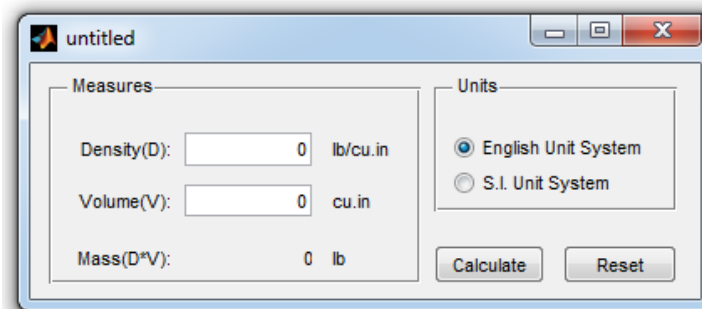


Figura 21 - Janela resultante da compilação

Fonte: Adaptado de Matlab, 2011.

A partir dessas aplicações, no capítulo seguinte serão apresentadas as funcionalidades da ferramenta computacional desenvolvida. A versão do software MATLAB[®] utilizado foi a R2011b, cuja licença pertence ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação – PPGSCA – da Universidade Federal Rural do Semi-Árido – UFERSA.

Capítulo 3 - Ferramenta Computacional Desenvolvida

A proposta central deste trabalho versa sobre o desenvolvimento de uma ferramenta computacional a partir das três outras anteriormente citadas: MATLAB[®], SIMULINK[®] e o GUIDE.

O MATLAB[®] apresenta diversas extensões (*Toolboxes*) para uma variedade de aplicações, dentre estas existem uma *Toolbox* para implementação de problemas em Redes Neurais Artificiais (*Neural Network Toolbox*), para Lógica Fuzzy (*Fuzzy Logic Toolbox*), outra para ajuste de controlador PID (*PIDtool*) e outra para otimização (*Optimization Tool*), onde é possível obter soluções utilizando GA.

A ferramenta computacional desenvolvida neste trabalho – GACT (*Genetic Algorithm Control Tool*) – permite ao usuário, por meio de uma janela de interface gráfica interativa, criar um novo sistema no SIMULINK[®] (ou utilizar um já existente) e conceber o controle deste sistema através das técnicas de Inteligência Computacional, com o subsídio de um Algoritmo Genético também de elaboração própria.

O GACT foi desenvolvido para funcionar como uma *Toolbox* do *software* MATLAB[®] para aplicação em sistemas de controle e fundamenta-se na aplicação conjunta dos GAs com a teoria de controle clássico. O GACT baseia-se no avanço dos sistemas operacionais que deixaram de ser totalmente manuseados por linhas de comando (requerendo um maior nível técnico dos seus usuários), e passaram a ter uma interface gráfica interativa (simplificando a forma de manuseio).

A criação de tal *Toolbox* é justificada pela necessidade da aplicação das técnicas de IC por usuários com baixa familiarização com o assunto, podendo estes comparar os resultados obtidos com outros métodos de ajustes de controlador.

As Figuras 22 e 23 ilustram as diferenças entre o GA implementado em um programa com e sem interface gráfica interativa, respectivamente.

Desta forma, o GACT possibilita a implementação de um sistema de controle ajustado por inteligência computacional de maneira mais simplificada e interativa. A interface gráfica foi concebida através do software GUIDE, integrado ao MATLAB[®] de maneira a relacionar-se com os códigos fontes e os diagramas de blocos do sistema a ser controlado no SIMULINK[®].

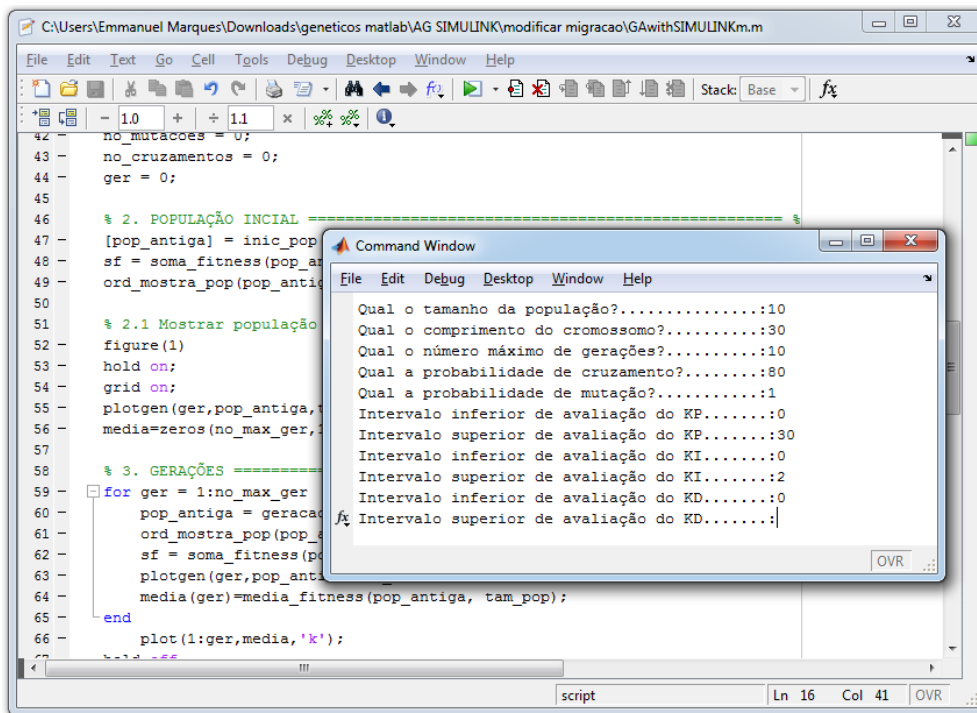


Figura 22 - Implantação do GA desenvolvido sem interface gráfica interativa

Fonte: Autoria própria, 2013.

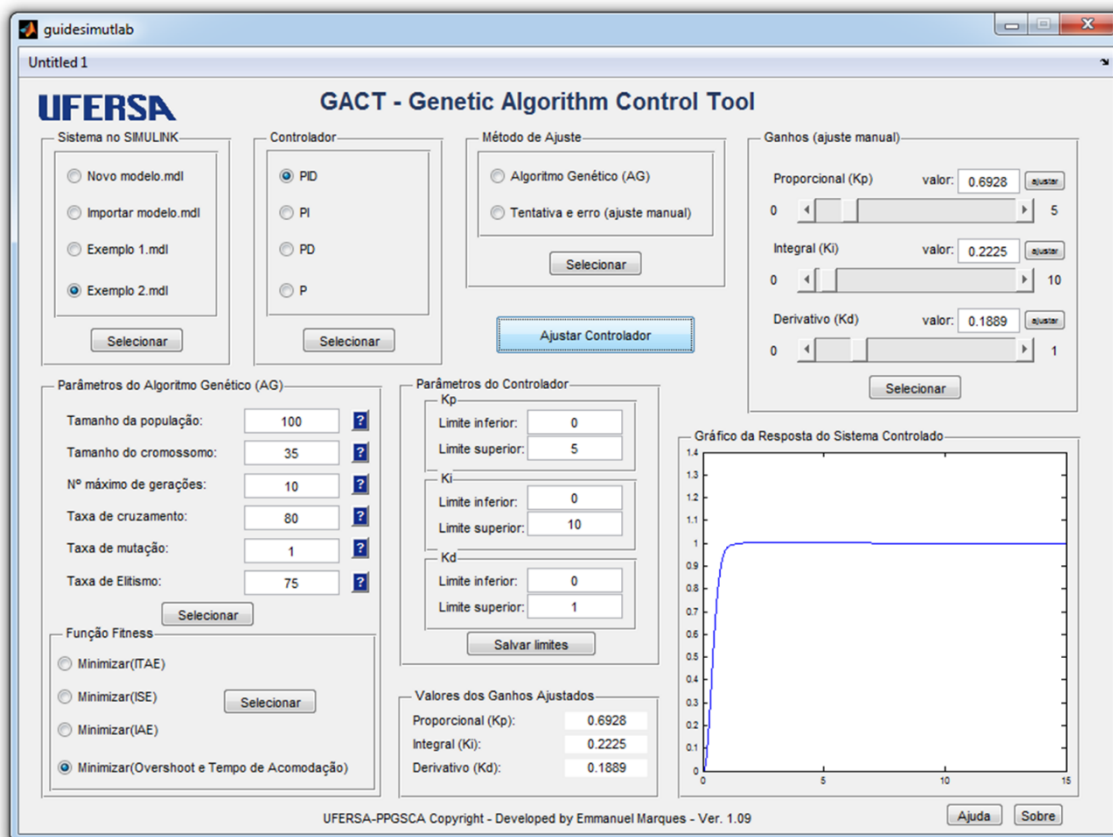


Figura 23 - Implantação do GA desenvolvido com interface gráfica interativa proposta

Fonte: Autoria própria, 2013.

Além do GUIDE, o GACT utilizar-se-á da relação entre o MATLAB® e o SIMULINK®. Por trás da interface gráfica existe um programa principal (*main_program.m*), várias subrotinas (*functions.m*), um modelo no SIMULINK® (*model.mdl*). As rotinas do GA escritas no MATLAB® interagem com o modelo no SIMULINK® conforme ilustrado na Figura 24.

Com isso, é possível estabelecer uma comunicação entre todas essas aplicações, fazendo com que o SIMULINK® envie e receba variáveis da *workspace* – espaço de trabalho do MATLAB®, onde se armazenam as variáveis (CHAPMAN, 2010).

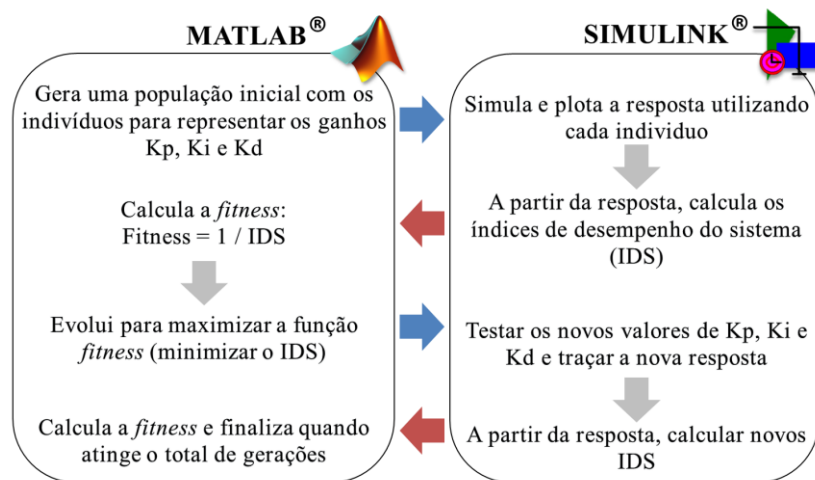


Figura 24 - Comunicação/relação entre as duas aplicações, MATLAB® e SIMULINK®, usado no método proposto

Fonte: Autoria própria, 2013.

A partir do GACT é possível estabelecer os valores dos ganhos proporcional, integral e derivativo que otimiza a resposta do sistema em questão, após a seleção do arquivo contendo o modelo no SIMULINK® de tal sistema, o tipo do controlador, e os respectivos parâmetros do GA e do controlador.

Os parâmetros do controlador se referem aos limites máximos e mínimos de cada um dos ganhos (K_P , K_I e K_D).

Os parâmetros do GA são: Tamanho da população, Tamanho do cromossomo, Número máximo de gerações, Taxa de cruzamento, Taxa de Mutação e Taxa de Elitismo. Cada um desses valores, conforme apresentados inicialmente apresentam um significado específico e influenciam de forma particular nos resultados.

Ao lado de cada um dos campos dos parâmetros do GA há ícones informativos que trazem sugestões de valores conforme literatura, além da explicação do que cada valor influencia na busca da solução.

A função *fitness* também pode ser escolhida pelo usuário, de forma a adaptar-se ao problema em questão. Para um problema simples onde deseja-se minimizar o máximo sobressinal (*overshoot*) e o tempo de acomodação da saída do sistema para a resposta à uma entrada degrau, é ponderável o uso da última função sugerida “*Minimizar (Overshoot e Tempo de Acomodação)*”.

Quando o problema buscar uma referência diferente do degrau, sem *set point* fixo, como por exemplo, uma onda quadrada ou senoidal, a melhor escolha é uma das 03 (três) primeiras funções. Essas funções *fitness* são definidas pelos seguintes índices de desempenho, conforme citadas no tópico 2.1.1.3:

- Integral do Erro ao Quadrado (ISE – do inglês *Integral of Squared Error*)
- Integral do Erro Absoluto (IAE – do inglês *Integral of Absolute Error*)
- Integral do Erro Absoluto ponderado no Tempo (ITAE – do inglês *Integral of Time multiply Absolute Error*)

Serão apresentadas, nos tópicos a seguir, as 09 (nove) funcionalidade de cada caixa (*box*) e botões destacados na Figura 25.

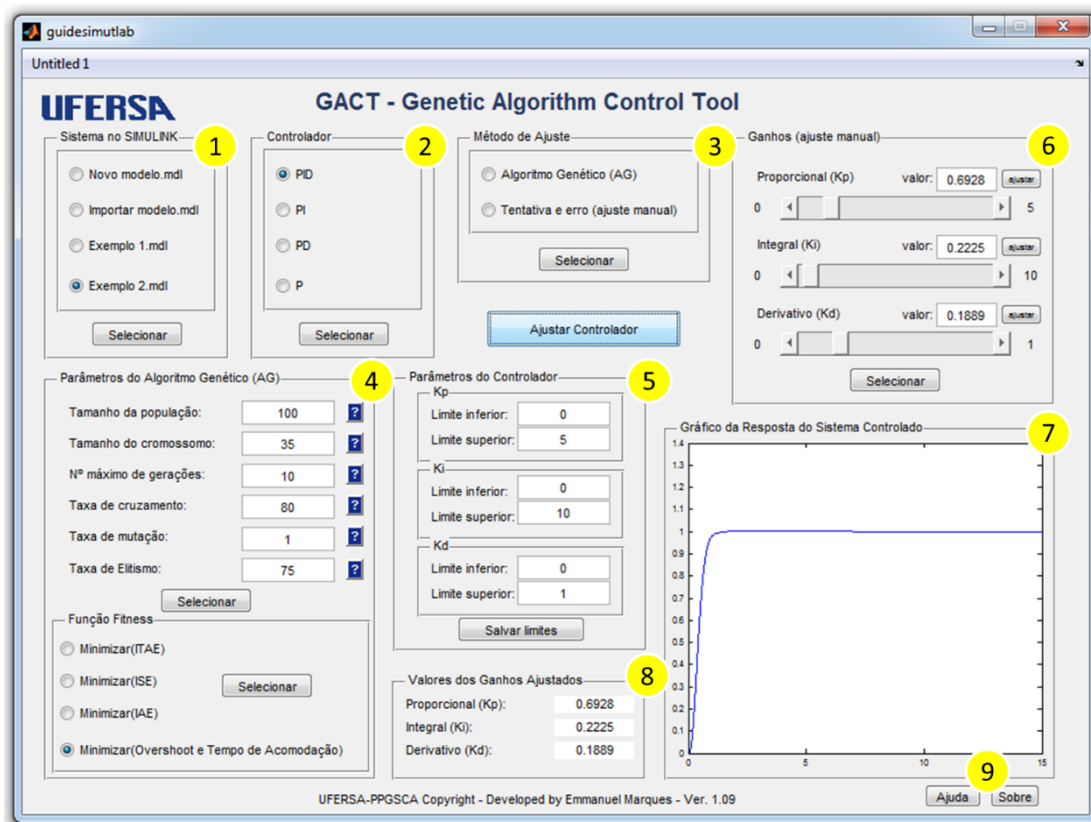


Figura 25 - GACT com as 09 (nove) funcionalidades destacadas

Fonte: Autoria própria, 2013.

3.1 Sistema do Simulink

Nesta caixa, em destaque na Figura 26, o usuário pode escolher entre utilizar um novo modelo do SIMULINK®, importar um modelo já criado ou ainda fazer uso de um dos dois exemplos inseridos nesta Toolbox. O “Exemplo 1” consiste no controle de nível de um sistema de tanques acoplados. O “Exemplo 2” consiste no controle da tensão de excitação de um gerador síncrono.

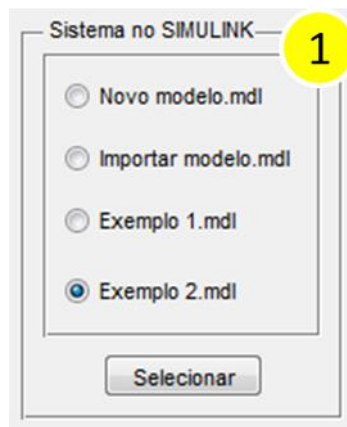


Figura 26 - Box "Sistema do Simulink"

Fonte: Autoria própria, 2013.

Após a escolher entre uma das quatro opções o usuário deve clicar no botão “Selecionar” para confirmar sua escolha e caso tenha indicado a primeira opção um novo modelo do SIMULINK® será aberto; caso tenha indicado a segunda uma janela abrirá informando-o que é necessário importar o modelo desejado renomeado como “model.mdl” para o diretório que contém o GACT; caso uma das duas últimas opções sejam escolhidas, o GACT estará pronto para ajustar os ganhos dos controladores para os sistemas anteriormente mencionados (tanque acoplado e gerador síncrono).

3.2 Controlador

Nesta caixa, em destaque na Figura 27, o usuário define que tipo de controlador deseja ajustar. As opções são:

- Controlador Proporcional, Integral e Derivativo (PID);
- Controlador Proporcional e Integral (PI);
- Controlador Proporcional e Derivativo (PD);
- Controlador Proporcional (P).

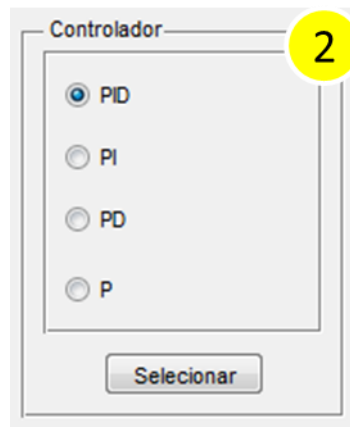


Figura 27 - Box "Controlador"

Fonte: Autoria própria, 2013.

Após a escolher entre uma das quatro opções o usuário deve clicar no botão “Selecionar” para confirmar sua escolha.

3.3 Método de Ajuste

Nesta outra caixa, em destaque na Figura 28, o usuário define que tipo de método de ajuste deseja utilizar: Algoritmo Genético ou Ajuste Manual. Caso escolha a segunda opção o usuário do GACT pode usar as barras de rolagem do box apresentado no tópico 3.6.

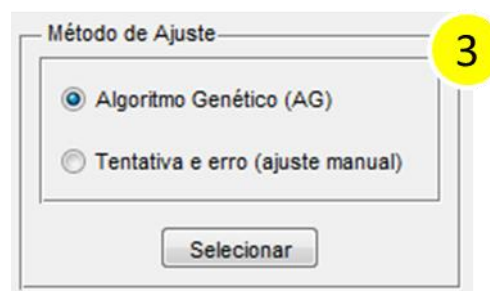


Figura 28 - Box "Método de Ajuste"

Fonte: Autoria própria, 2013.

Após selecionar a primeira opção é preciso definir os parâmetros do GA e os valores máximos e mínimos de cada ganho.

3.4 Parâmetros do Algoritmo Genético

Nesta caixa, em destaque na Figura 29, o usuário define os parâmetros do GA e ao lado de cada um desses parâmetros foram inseridos botões ajuda “?” para auxiliar no devido preenchimento afim de levar o algoritmo à convergir para um resultado satisfatório.

Parâmetros do Algoritmo Genético (AG)

Tamanho da população: 100 ?

Tamanho do cromossomo: 35 ?

Nº máximo de gerações: 10 ?

Probabilidade de cruzamento: 80 ?

Probabilidade de Mutação: 1 ?

Taxa de Elitismo: 75 ?

Selecionar

Função Fitness

Minimizar(ITAE)

Minimizar(ISE)

Minimizar(IAE)

Minimizar(Overshoot e Tempo de Acomodação)

Selecionar

Figura 29 - Box "Parâmetros do Algoritmo Genético"

Fonte: Autoria própria, 2013.

Um exemplo de janela informativa é mostrado na Figura 30, apresentado um texto de ajuda sobre o parâmetro “Taxa de Mutação”. Cada janela dessa apresenta ainda uma sugestão de valor conforme valores mais usados na literatura.

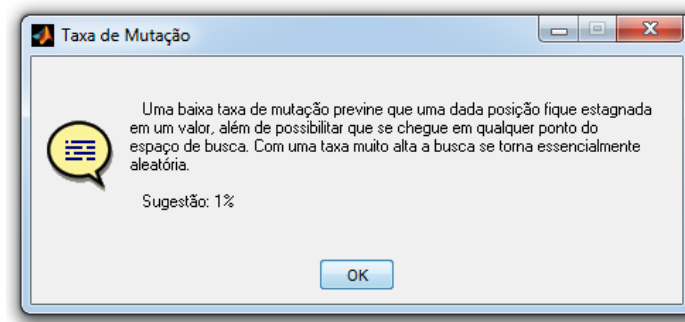


Figura 30 - Janela informativa sobre o parâmetro "Taxa de Mutação"

Fonte: Autoria própria, 2013.

Cada janela desta apresenta ainda uma sugestão de valor conforme valores mais usados na literatura.

3.5 Parâmetros do Controlador

Nesta outra caixa, em destaque na Figura 31, o usuário define os parâmetros do controlador. Estes parâmetros consistem nos limites inferior e superior de cada ganho (K_P , K_I e K_D).

A caixa de diálogo "Parâmetros do Controlador" contém três seções para configurar os ganhos. Cada seção possui dois campos de entrada: "Limite inferior" e "Limite superior".

- Para K_p : Limite inferior = 0, Limite superior = 5.
- Para K_i : Limite inferior = 0, Limite superior = 10.
- Para K_d : Limite inferior = 0, Limite superior = 1.

Um botão "Salvar limites" está localizado na base da caixa. Um círculo amarelo com o número "5" está sobreposto no canto superior direito da caixa.

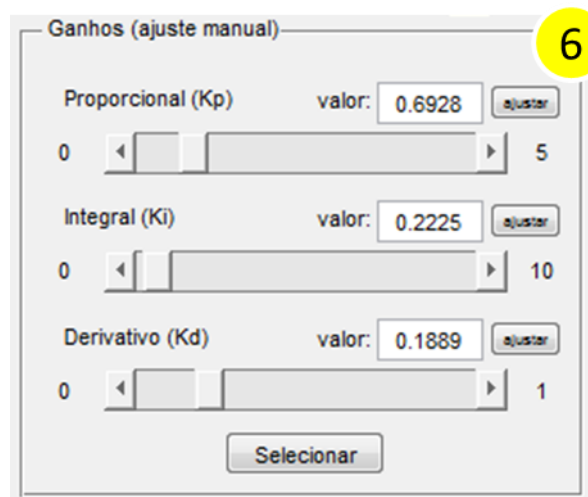
Figura 31 - Box "Parâmetros do Controlador"

Fonte: Autoria própria, 2013.

Estes parâmetros escolhidos pelo usuário é o domínio de busca do GA. O limite inferior e superior representam X_{min} e X_{max} na equação (1), conforme tópico 2.1.1.2.

3.6 Ganhos (ajuste manual)

Esta outra caixa, em destaque na Figura 32, permite ao usuário fazer o ajuste manual dos parâmetros do controlador. Esse ajuste pode ser realizado através de 03 (três) barras de rolagem horizontal (*sliders*) ou editando o valor nas caixas de texto e clicando em ajustar.



A caixa de diálogo "Ganhos (ajuste manual)" apresenta três controles deslizantes para ajustar os ganhos do controlador. Cada controle deslizante possui uma barra horizontal com setas de navegação e um campo de texto para o valor atual. Abaixo dos controles deslizantes, há um botão "Selecionar".

Parâmetro	Valor	Domínio
Proporcional (Kp)	0.6928	0 a 5
Integral (Ki)	0.2225	0 a 10
Derivativo (Kd)	0.1889	0 a 1

Figura 32 - Box "Ganhos (ajuste manual)"

Fonte: Autoria própria, 2013.

Cada um desses *sliders* definem um valor percentual compreendido entre os valores do domínio definido no tópico 3.5.

3.7 Gráfico da Resposta do Sistema Controlado

Esta caixa, em destaque na Figura 33, apresenta o gráfico da resposta do sistema controlado a partir dos parâmetros ajustados pelo GACT.

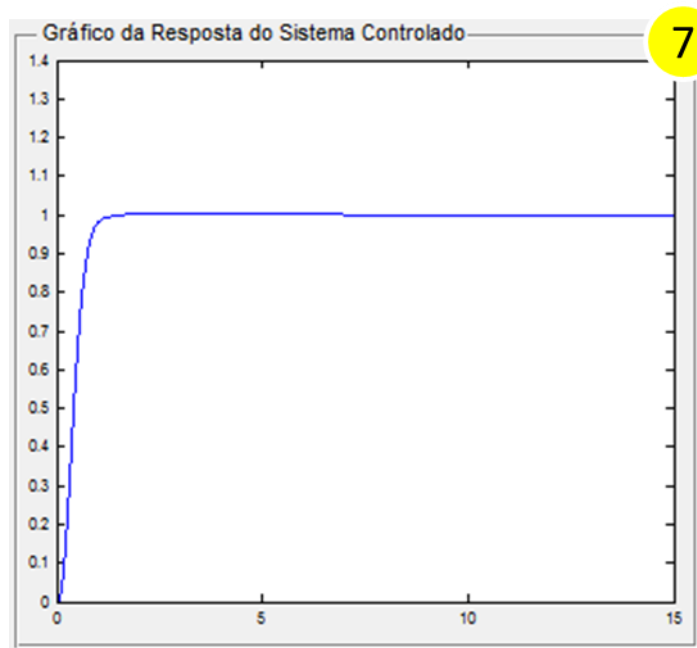


Figura 33 - Box "Gráfico da Resposta do Sistema Controlado"

Fonte: Autoria própria, 2013.

Neste exemplo, o gráfico exibe a resposta ao degrau unitário apresentando pequena elevação de sobressinal e com tempo de acomodação de aproximadamente 1 segundo.

3.8 Valores dos Ganhos Ajustados

Nesta outra caixa, em destaque na Figura 34, os valores dos ganhos ajustados pelo GACT são apresentados.



Valores dos Ganhos Ajustados

Proporcional (Kp):	0.6928
Integral (Ki):	0.2225
Derivativo (Kd):	0.1889

Figura 34 - Box "Valores dos Ganhos Ajustados"

Fonte: Autoria própria, 2013.

Esses valores ajustados também são representados nos *sliders* do tópico 3.6.

3.9 Informações

Estes botões, em destaque na Figura 35, auxiliam o usuário na utilização do programa e apresentam informações sobre a *Toolbox*.



Figura 35 - Botões de informações

Fonte: Autoria própria, 2013.

Ao clicar no botão “sobre” o GACT apresenta a seguinte janela mostrada na Figura 36.

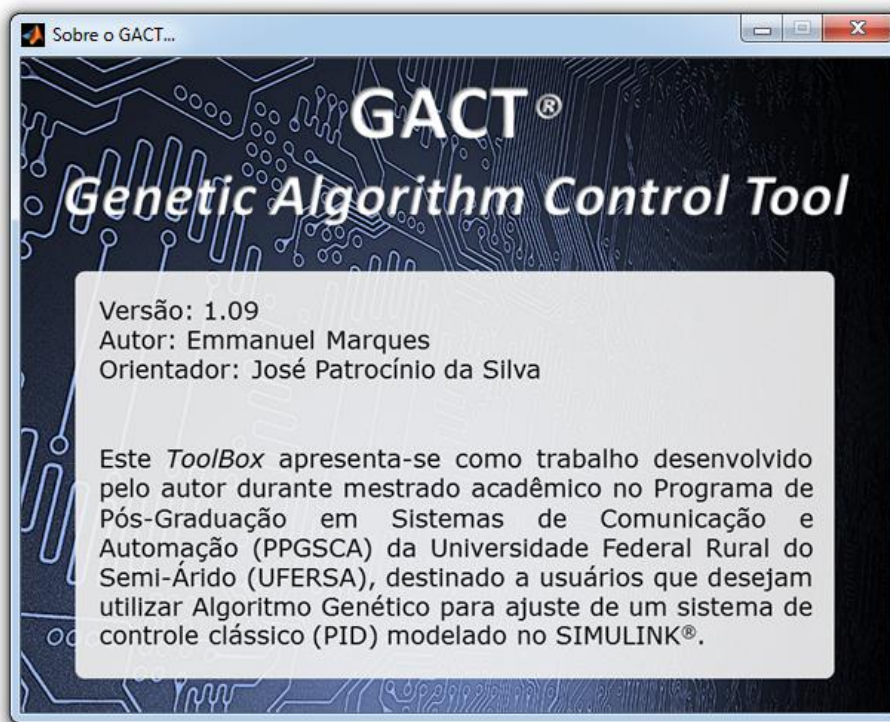


Figura 36 - Janela informativa “Sobre” o programa

Fonte: Autoria própria, 2013.

Todas as análises e resultados apresentados no capítulo a seguir foram obtidos através do GACT desenvolvido.

Capítulo 4 - Resultados e Discussões

A fim de corroborar a aplicação desenvolvida foram ajustado dois controladores, um para cada sistema anteriormente mencionadas no seção 3.1, a partir do GACT e a partir das técnicas tradicionais de controle. Os resultados e discussões de cada um desses dois sistemas são tratados nos tópicos a seguir.

4.1 Controle de Nível em Sistema de Tanques Acoplados

Neste primeiro sistema, o GACT realizou a busca dos ganhos do controlador PI (K_P e K_I) que otimiza características da resposta (tempo de acomodação e máximo sobressinal) de um sistema de tanques acoplados.

O modelo utilizado neste exemplo baseou-se no sistema de tanques acoplados desenvolvido por Quanser Inc. (2008), que consiste em dois tanques, como mostrado na Figura 37. Um motor-bomba de corrente contínua (*pump*) eleva o líquido do reservatório inferior de água (*water basin*) para duas conexões hidráulicas chamadas *OUT1* e *OUT2*. Um tubo liga a conexão *OUT1* de modo que o líquido passa para dentro do tanque 1.

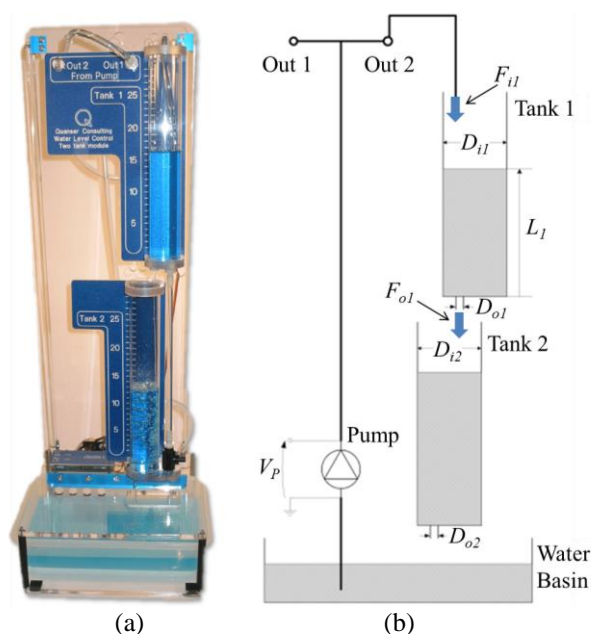


Figura 37 - (a) Sistema de tanques de acoplamento desenvolvido por Quanser Inc. (b) Representação esquemática do sistema.

Fonte: Adaptado de Quanser Inc., 2008.

Este líquido, a partir do primeiro tanque (parte superior da planta) flui por gravidade para dentro do segundo tanque (*tank 2*), passando através de um furo, cujo diâmetro pode ser variado através de uma simples mudança das peças de saída. A partir do tanque 2, também por ação da gravidade, o líquido flui de volta para o reservatório inferior de água através de um orifício com as mesmas características do orifício do tanque 1.

Para as simulações, foi utilizado o modelo do SIMULINK® mostrado na Figura 38. Para testar o máximo sobressinal e o tempo de acomodação, este modelo usa um ensaio contínuo de três pontos de ajuste: 15, 17 e 13 centímetros do nível de água no tanque 1.

Os parâmetros do Algoritmo Genético definidos no GACT estão apresentados na Tabela 5.

Tabela 5 - Parâmetros do GACT para controle em sistema de tanques acoplados

Parâmetro	Valor / Opção
Tamanho da população	100
Tamanho do cromossomo	35
Número máximo de gerações	10
Taxa de cruzamento	80 %
Taxa de mutação	1 %
Taxa de elitismo	75 %
Função <i>Fitness</i>	Minimizar (<i>ITAE</i>)

Fonte: Autoria própria, 2013.

A Figura 38 ilustra o diagrama de blocos do sistemas simulado com o controlador PI que foi otimizado de forma a obter o menor valor do índice ITAE.

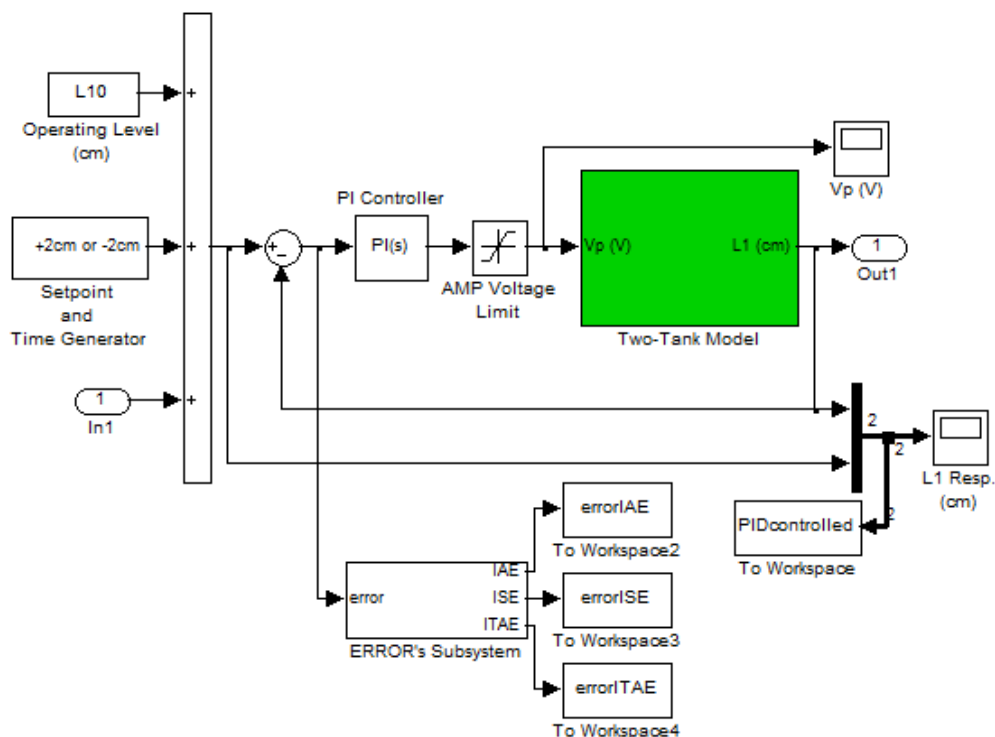


Figura 38 - Modelo utilizado no MATLAB® SIMULINK® proposto por Quanser Inc. (2008)

Fonte: Autoria própria, 2013.

Na Figura 39 os resultados obtidos do sistema controlado com um controlador PI ajustado através da teoria clássica de controle é comparado com o ajustado através do GA desenvolvido, nos 3 níveis de referência 15 cm, 17 cm e 13 cm.

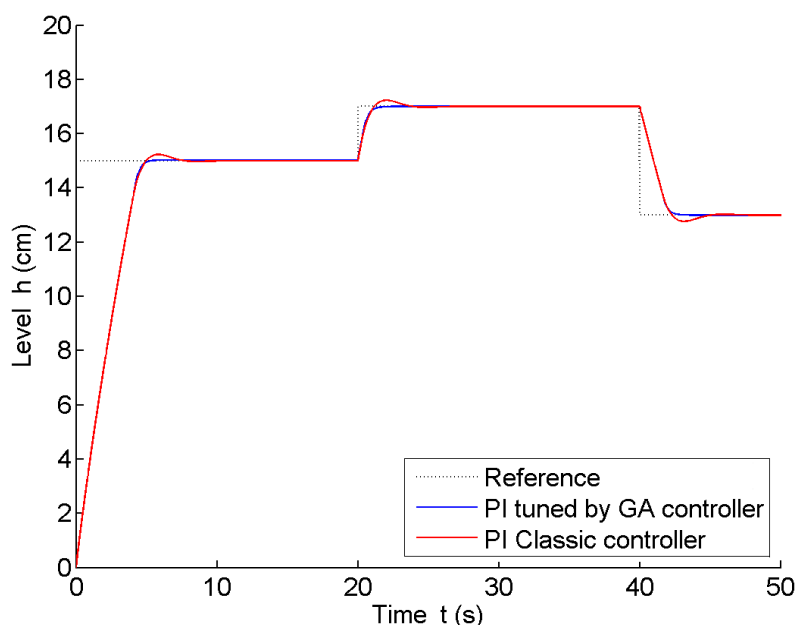


Figura 39 - Comparação entre controlador PI sintonizado pelo GA e controlador PI clássico

Fonte: Autoria própria, 2013.

Também foi analisada a resposta ao degrau de 15 cm para ambos os controladores, a fim de calcular o máximo sobressinal (*Max Overshoot*) e os valores de tempo de acomodação (*Settling Time*) e de pico (*Peak Time*). Os resultados destas simulações são mostrado nas Figuras 40 e 41.

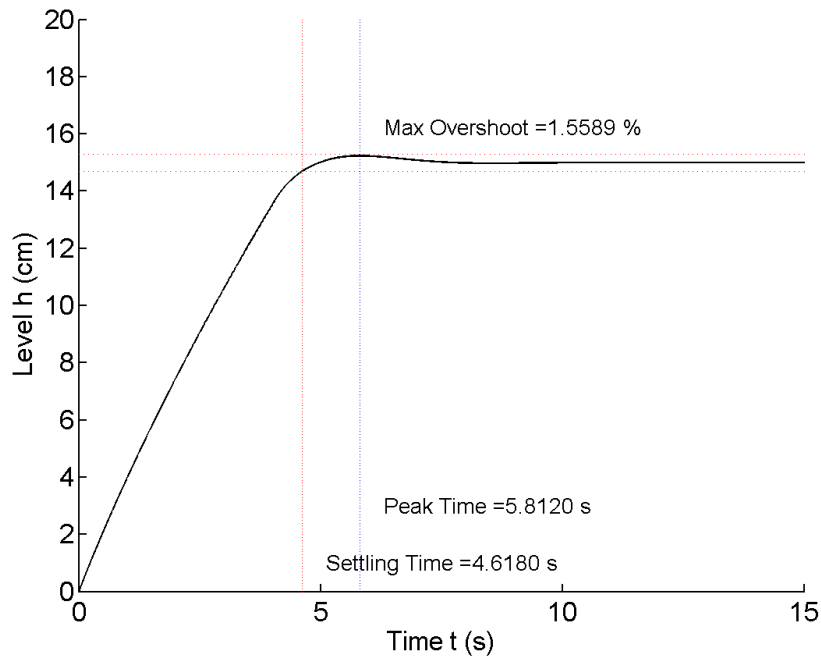


Figura 40 - Resposta ao degrau do controle PI clássico.

Fonte: Autoria própria, 2013.

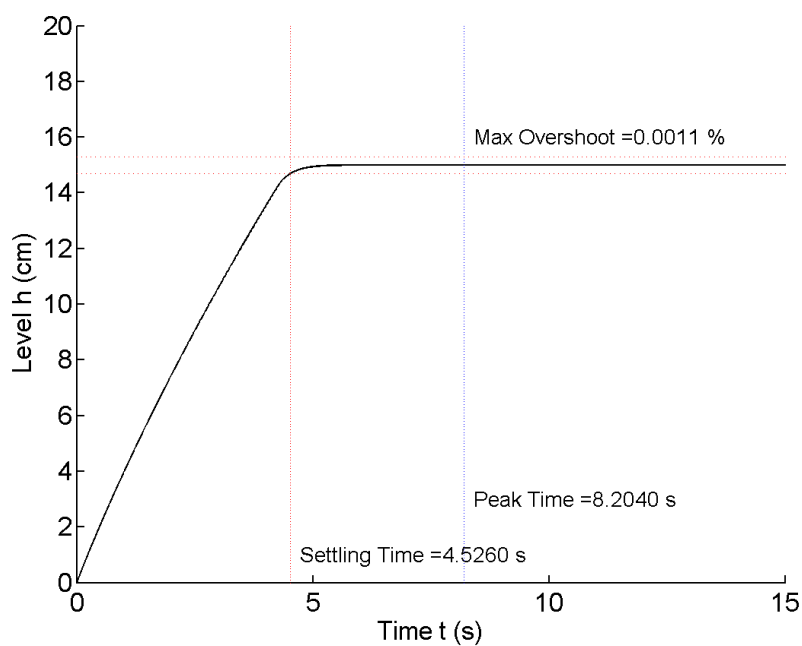


Figura 41 - Resposta ao degrau do controle PI ajustado com o GACT

Fonte: Autoria própria, 2013.

Os resultados obtidos pelo GACT são comparados e apresentados na Tabela 6 com os obtidos pelo controle clássico proposto por Quanser Inc. (2008).

Tabela 6 - Comparação das respostas ao degrau unitário

Sintonia	Máximo sobressinal	Tempo de acomodação
Controle clássico	1,5589 %	4.6180 s
GACT	0,0011 %	4.5260 s

Fonte: Autoria própria, 2013.

Assim, foi possível notar uma redução considerável do máximo sobressinal, e do valor do tempo de acomodação. O máximo sobressinal diminuiu de 1,5589% a 0,0011% e o tempo de acomodação diminuiu de 4,6180 para 4,5260 segundos.

4.2 Controle de Tensão de Excitação em Gerador Síncrono

A literatura recente tem mostrado uma grande quantidade de aplicações de GAs no problema do ajuste do controlador de diversos equipamentos relacionados com sistemas elétricos. A principal característica de tais aplicações é ajuste do controlador para controle de valores de frequência, tensão e corrente. Esses problemas são particularmente interessantes para GAs não só pela sua complexidade, mas também por serem dinâmicos (XIAOFANG *et al.*, 2010; KUMAR, GUPTA e SINGH, 2006; RAMYA, SELVI e NIVETHITHA, 2013; HASANIEN, 2013).

Neste segundo sistema, o GACT realizou a busca dos ganhos do controlador PID (K_P , K_I e K_D) que otimiza características da resposta (tempo de acomodação e máximo sobressinal) de um sistema elétrico de potência utilizado por Devaraj e Selvabala (2009).

Neste trabalho um regulador de tensão automático (AVR – do inglês *Automatic Voltage Regulator*) é utilizado para controlar a tensão de excitação de um gerador síncrono, tal sistema foi modelado no *SIMULINK*[®], conforme representado na Figura 42.

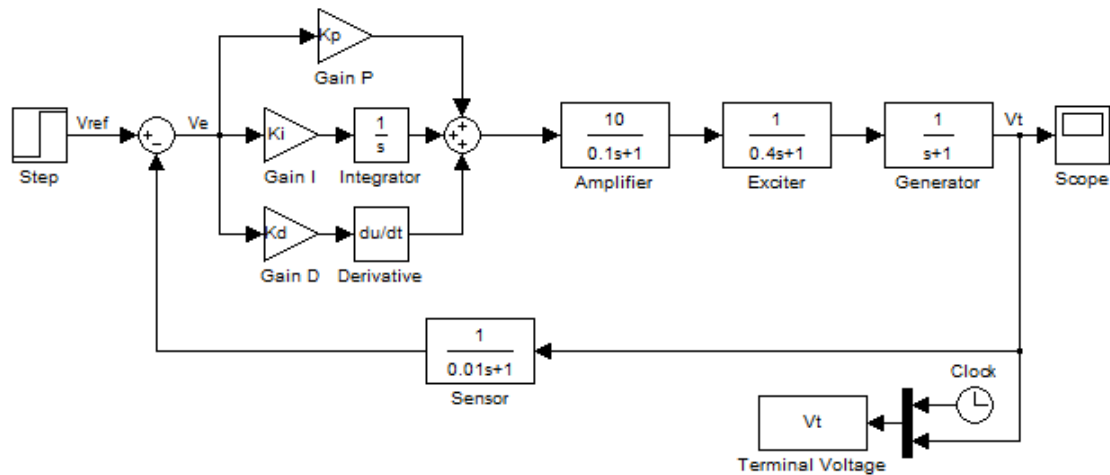


Figura 42 - Modelo no SIMULINK® do sistema AVR, juntamente com o controlador PID

Fonte: Autoria própria, 2013.

Os parâmetros do Algoritmo Genético definidos no GACT estão apresentados na Tabela 7.

Tabela 7 - Parâmetros do GACT para controle no sistema do gerador síncrono

Parâmetro	Valor / Opção
Tamanho da população	100
Tamanho do cromossomo	35
Número máximo de gerações	10
Taxa de cruzamento	80 %
Taxa de mutação	1 %
Taxa de elitismo	75 %
Função <i>Fitness</i>	Minimizar (<i>Overshoot</i> e Tempo de Acomodação)

Fonte: Autoria própria, 2013.

Os parâmetros do controlador foram baseados no ajuste PID pelo método de Ziegler-Nichols (ZN). A função de transferência em malha fechada do sistema de controle de tensão de excitação do gerador síncrono, em função dos ganhos proporcional, integral e derivativo (K_P , K_I e K_D) é:

$$\frac{V_t(s)}{V_{ref}(s)} = \frac{0,1K_D s^3 + (10K_D + 0,1K_P)s^2 + (10K_P + 0,1K_I)s + 10K_I}{0,0004s^5 + 0,0454s^4 + 0,555s^3 + (10K_D + 1,51)s^2 + (10K_P + 1)s + 10K_I}$$

Utilizando o método de ZN para este sistema obtêm-se os seguintes ganhos dos controladores: $K_P=1,155$; $K_I=2,1$; $K_D=0,1588$. A partir destes ganhos, escolhe-se os seguintes domínios de controlador (de zero à aproximadamente 5 vezes o valor do ganho obtido pelo ZN): $0,0000 \leq K_P \leq 5,0000$; $0,0000 \leq K_I \leq 10,0000$ E $0,0000 \leq K_D \leq 1,0000$.

Após selecionar o botão “Ajustar Controlador” o GACT executa o código fonte contendo o GA proposto retornando os seguintes valores de ganhos otimizados $K_P=0,6928$; $K_I=0,2225$; $K_D=0,1889$.

A resposta ao degrau unitário para o sistema controlado utilizando os ganhos ajustados por ZN e pelo GACT são mostrados na Figura 43.

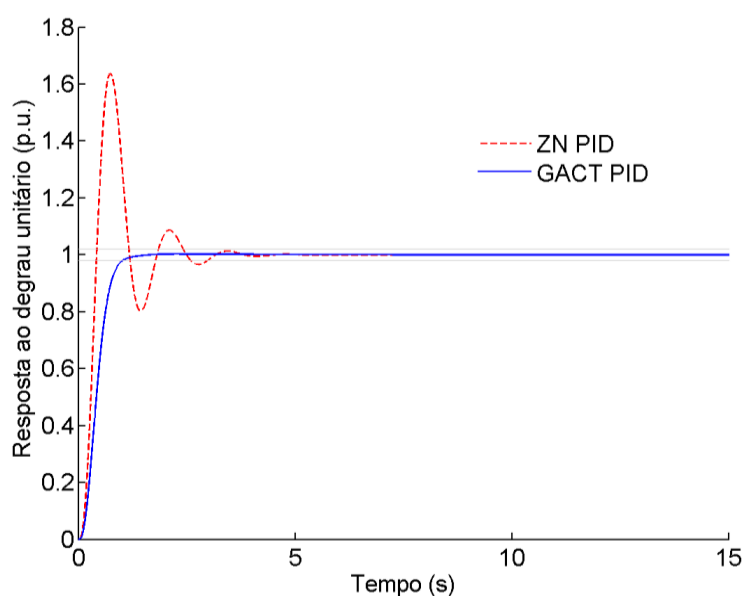


Figura 43 - Gráfico comparativo da resposta ao degrau unitário do sistema controlado por ZN e GACT

Fonte: Autoria própria, 2013.

Respostas à entrada degrau unitário do sistema controlado pelo PID ajustado através do GACT e através do método de Zigler-Nichols (ZN).

Do gráfico acima são obtidos os valores de máximo sobressinal e tempo de acomodação para cada PID. No PID ajustado pelo método de ZN o máximo sobressinal é de 63,7% e o tempo de acomodação (quando o sinal da resposta acomoda-se entre 0,98 e 1,02) é de 2,98 segundos. Já no PID ajustado pelo GACT o máximo sobressinal é de 0,3% e o tempo de acomodação é de 1,01 segundo.

Os resultados obtidos pelo GACT também são comparados como os obtidos por Devaraj e Selvabala (2009) e apresentados na Tabela 8.

Tabela 8 - Comparação das respostas ao degrau unitário

Sintonia	Máximo sobressinal	Tempo de acomodação
Ziegler-Nichols	63,7 %	2,98 s
GA (Devaraj e Selvabala)	0,4%	1,26 s
GACT	0,3%	1,01 s

Fonte: Autoria própria, 2013.

Como pode ser observado na Tabela 8, de forma geral, o GACT proposto neste trabalho se sobressaiu frente aos resultados do tradicional método de ZN e do GA desenvolvido por Devaraj e Selvabala (2009), apresentando valores inferiores de tempo de acomodação e máximo sobressinal.

O controle de processos é uma tarefa crucial para o perfeito funcionamento de uma planta industrial. O controlador sintonizado pelo GA é essencial a fim de reduzir ou eliminar as intervenções do operador, como citado por Batzias, Nikolaou e Kakos (2003) e Wang (2002), tais como superações (*overshoot*) excessivas, que pode ocorrer nas ações de controle durante as condições de arranque (*startup*), dinâmicas e transitórias.

Com esses resultados foi possível comprovar a eficácia não tão somente do GA desenvolvido como também da interação entre o MATLAB[®] e o SIMULINK[®] para a referida aplicação.

Desta forma, em linhas gerais, os objetivos deste trabalho foram atingidos a partir do desenvolvimento de um GA capaz de ajustar os ganhos do controlador PID de um sistema genérico representado por um modelo do SIMULINK[®] que pode ser importado pelo usuário da ferramenta computacional proposta: o GACT.

Capítulo 5 - Conclusões e Trabalhos Futuros

Nesse trabalho, a Inteligência Computacional foi considerada, através da implementação de um Algoritmo Genético, de autoria própria, para desenvolvimento de um programa com interface gráfica, agregada ao MATLAB[®], denominado GACT – *Genetic Algorithm Control Tool*. Tal aplicativo possibilita ao seu usuário aplicar as técnicas do GA para projetar um controlador a partir de um sistema modelado no SIMULINK[®].

Em adição, para fundamentar o modelo, foram estudadas várias referências bibliográficas sobre os sistemas computacionais dentro de uma perspectiva da Inteligência Computacional aplicada em sistemas de controle, com base na teoria de controle moderno, com a finalidade de familiarizar-se com as metodologias e técnicas de desenvolvimento de sistemas inteligentes, mais precisamente através de GAs. Neste aspecto, foi estabelecida uma interação entre as aplicações MATLAB[®], SIMULINK[®] e GUIDE de forma a permitir o uso de GAs em conjunto com a teoria de controle e estabelecer uma comparação envolvendo a técnica aqui proposta, com as técnicas tradicionais de controle. Tais comparações comprovaram a possibilidade de aplicação da técnica e da ferramenta proposta pois os parâmetros obtidos através do GACT apresentaram-se satisfatórios quando comparados com a literatura recente e as técnicas tradicionais de controle.

Para o primeiro sistema, analisou-se os dados obtidos nas simulações dos controladores PI clássico e PI sintonizado pelo GACT. Este último ofereceu um tempo de acomodação 1,99% mais rápido. Além de um desempenho superior com relação ao *overshoot*, reduzindo-o para aproximadamente zero.

A partir do exposto, fica evidente que o controlador PI ajustado pelo GACT pode ser utilizado para controlar sistemas de tanques acoplados, especialmente se for considerada a operação próxima das condições de fronteira, porque uma ultrapassagem nas regiões superiores ou inferiores podem representar um risco da operação do processo. Neste caso, se uma ultrapassagem ocorre durante a operação de um dos tanques, com o nível mais elevado, um transbordamento pode ocorrer na mudança do ponto de funcionamento. E, no caso de operação com o nível mais baixo, há a possibilidade de descontinuidade no abastecimento de água na saída tanques.

Para o segundo sistema, analisando os dados obtidos nas simulações do PID sintonizado por ZN e o PID sintonizado através do GACT, observou-se que este último obteve um tempo de acomodação três vezes mais rápido, além de uma melhora expressiva em relação à máxima superação de sinal, apresentando valor quase nulo (0,3%).

A partir do exposto, fica evidente que o controlador PID sintonizado através da ferramenta computacional desenvolvida pode ser usado para controlar a tensão de excitação de um gerador síncrono, especialmente devido ao valor quase nulo de máximo sobressinal, representando a diminuição dos riscos provocados pela superação excessiva da tensão do gerador.

Neste caso, se um máximo sobressinal ocorre durante a operação do sistema, pode ocorrer uma sobre tensão causada pelo elevado valor da resposta (1,637 por unidade – p.u.), provocando danos ao gerador e às cargas a ele conectadas.

Para trabalhos futuros sugere-se o desenvolvimento de um bloco no SIMULINK® que implemente o ajuste de um ou mais controladores através de Algoritmo Genético. De modo que este bloco tenha como parâmetro de entrada o sinal de saída do sistema a ser controlado, e como parâmetro de saída os ganhos do controlador que deverão ser ajustados de maneira a produzir um sinal de saída o mais próximo possível do desejado (referência).

Tal aplicação pode também empregar uma técnica híbrida de inteligência computacional, agregando-se lógica *fuzzy* ao sistema de controle.

Referências Bibliográficas

- ANH, H. P. H.; AHN, K. K.; NAM, N. T. Modeling Identification of the Nonlinear Robot Arm System Using MISO NARX Fuzzy Model and Genetic Algorithm. **Robot Arms**, Rijeka, 09 Jun 2011.
- BARRETO, J. M. **Inteligência artificial no limiar do Século XXI – abordagem híbrida**: simbólica, conexionista e evolucionária. 3a. ed. Florianópolis: Editora da UFSC, 2001.
- BATZIAS, F. A.; NIKOLAOU, N. P.; KAKOS, A. S. **Computer aided optimisation of an agro-industrial complex consisting of processes and inventories by means of a custom-developed hybrid genetic algorithm**. Industrial Technology, 2003 IEEE International Conference on. : IEEE. 2003. p. 60-65.
- BAZANELLA, A. S.; SILVA, J. M. G. **Sistemas de Controle**: Princípios e Métodos de Projeto. 1. ed. Porto Alegre: Editora da UFRGS, 2005.
- BITTENCOURT, G. **Inteligência Artificial - Ferramentas e Teorias**. 2a. ed. Florianópolis: Editora da UFSC, 2001.
- BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. **ACM Comput. Surv.**, New York, v. 35, n. 3, p. 268-308, Set 2003. ISSN 0360-0300.
- BOLTON, W. **Engenharia de Controle**. Tradução de Valcere Vieira Rocha e Silva. 1a. ed. São Paulo: Makron Books, 1995.
- BUENO, F. **Métodos Heurísticos - Teoria e Implementações**. IFSC. Araranguá. 2009.
- CHAPMAN, S. J. **Programação em Matlab para Engenheiros**. Tradução de Flávio Soares Correa da Silva. 2a. ed. São Paulo: Cengage Learning, 2010. ISBN 978-85-221-0789-6.
- D'AZZO, J. J.; HOUPIS, C. H. **Linear Control System Analysis and Design with Matlab**. 5a. ed. New York: Marcel Dekker, Inc., 2003. ISBN 0-8247-4038-6.
- DEVARAJ, D.; SELVABALA, B. Real-coded genetic algorithm and fuzzy logic approach for real-time tuning of proportional-integral - derivative controller in automatic voltage regulator system. **Generation, Transmission Distribution, IET**, Stevenage, 3, n. 7, 2009. 641-649.

- FREITAS, W.; FRANÇA, A. M. Modelagem e análise dinâmica de dispositivos DSTATCOM usando o SimPowerSystems para Matlab/Simulink. **Sba: Controle & Automação - Sociedade Brasileira de Automática**, Campinas, v. 16, p. 187 - 199, Jun 2005. ISSN 0103-1759.
- GABRIEL, P. H. R.; DELBEM, A. C. B. **Fundamentos de Algoritmos Evolutivos. Notas Didáticas do ICMC-USP**. São Carlos/SP: ICMC-USP, v. 75, 2008. 35 p. ISBN 0103-2585.
- HASANIEN, H. M. Design Optimization of PID Controller in Automatic Voltage Regulator System Using Taguchi Combined Genetic Algorithm Method. **Systems Journal, IEEE**, 7, n. 4, Dez 2013. 825 - 831.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor: Univ. Michigan Press, 1975.
- HUANG, H.-C. **Intelligent motion controller design for four-wheeled omnidirectional mobile robots using hybrid GA-PSO algorithm**. Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on. : . 2011. p. 2267-2272.
- IEEE COMPUTATIONAL INTELLIGENCE SOCIETY. IEEE Computational Intelligence Society, 2013. Disponível em: <<http://cis.ieee.org/>>. Acesso em: 2013 Set 17.
- KUMAR, R.; GUPTA, R. A.; SINGH, B. **Intelligent Tuned PID Controllers for PMSM Drive - A Critical Analysis**. Industrial Technology, 2006. ICIT 2006. IEEE International Conference on. India: IEEE. 2006. p. 2055-2060.
- LO, C. H.; WONG, Y. K.; RAD, A. B. Intelligent system for process supervision and fault diagnosis in dynamic physical systems. **Industrial Electronics, IEEE Transactions on**, Hong Kong, v. 53, p. 581-592, 2006. ISSN 0278-0046.
- MATHWORKS®. MATLAB Documentation Center. **MathWorks - MATLAB and Simulink for Technical Computing**, 2013. Disponível em: <<http://www.mathworks.com/help/documentation-center.html>>. Acesso em: 23 Julho 2013.
- MATLAB. **Matlab version R2011b**. Natick: MathWorks, Inc., 2011. Disponível em: <<http://www.mathworks.com>>.
- MAYA, P. A.; LEONARDI, F. **Controle essencial**. 1a. ed. São Paulo: Pearson Prentice Hall, 2010.

- MEHDI, M. **Parallel Hybrid Optimization Methods for Permutation Based Problems**. Universidade de Luxemburgo. Luxemburgo. 2011.
- NORMEY-RICO, J. E. **Projeto de controladores PID industriais**. Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina. Santa Catarina. 2001.
- OGATA, K. **Matlab for Control Engineers**. Upper Saddle River: Pearson Prentice Hall, 2008. ISBN 0-13-615077-2.
- OGATA, K. **Engenharia de controle moderno**. Tradução de Heloísa Coimbra de Souza. 5a. ed. São Paulo: Pearson Prentice Hall, 2010.
- PACHECO, M. A. C. **Algoritmos Genéticos: Princípios e Aplicações**. Rio de Janeiro: Edição do autor, 1999. Disponível em: <www.ica.ele.puc-rio.br>. ICA: Laboratório de Inteligência Computacional Aplicada.
- PENG, X.; VADAKKEPAT, P.; LEE, T.-H. **DNA coded GA for the rule base optimization of a fuzzy logic controller**. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on. : . 2001. p. 1191-1196.
- QUANSER INC. **Coupled-Tank Control Laboratory – Instructor Manual**. Markham, Ontario, Canadá: Quanser Inc., 2008.
- RAMOS, M. A. P. et al. **Genetic Rules to Tune Proportional + Derivative Controllers for Integrative Processes with Time Delays**. Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Proceedings. 15th International Conference on. Puebla, Mexico: IEEE. 2005. p. 143-148.
- RAMYA, R.; SELVI, K.; NIVETHITHA, S. S. **Optimization of synchronous generator excitation controller parameters**. Power, Energy and Control (ICPEC), 2013 International Conference on. Sri Rangalatchum Dindigul: IEEE. 2013. p. 585-590.
- SIVANANDAM, S. N.; DEEPA, S. N. **Introduction to Genetic Algorithms**. Heidelberg: Springer, 2010. ISBN 978-3-540-73189-4.
- SUMATHI, S.; PANEERSELVAM, S. **Computational intelligence paradigms: Theory & Applications using MATLAB**. Boca Raton: CRC Press, 2010.
- TALBI, E. -G. A Taxonomy of Hybrid Metaheuristics. **Journal of Heuristics**, 8, n. 5, 01 set. 2002. 541-564.
- TIANFIELD, H. Some Reflections on Intelligent Control. **Artificial Intelligence Review**, v. 23, n. 1, p. 57-91, 2005. ISSN 0269-2821. Disponível em: <<http://dx.doi.org/10.1007/s10462-004-4102-6>>.

- TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas Digitais - Princípios e Aplicações**. 11. ed. São Paulo: Prentice Hall, 2011. ISBN 9788576059226.
- WANG, P.-J. **PI-tuning methods based on GA**. Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on. : IEEE. 2002. p. 544-547.
- XIAOFANG, W. et al. **The application of GA-PID control algorithm to DC-DC converter**. Control Conference (CCC), 2010 29th Chinese. Beijing: IEEE. 2010. p. 3492-3496.