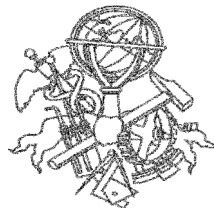


APLICAÇÃO DE ALGORITMOS GENÉTICOS PARA SINTONIA DE CONTROLADORES

Leonardo Silveira Paiva



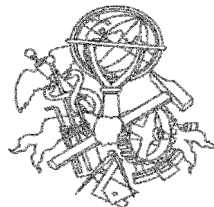
Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Sistemas de Planeamento Industrial
Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto
2010

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Leonardo Silveira Paiva, N° 1080032, 1080032@isep.ipp.pt

Orientação científica: Isabel Maria de Sousa de Jesus, isj@isep.ipp.pt

Co-orientação científica: Ramiro de Sousa Barbosa, rsb@isep.ipp.pt



Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Sistemas de Planeamento Industrial
Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2010

Aos meus queridos pais.

Agradecimentos

Em primeiro lugar agradeço a Deus por ter proporcionado todos os meios necessários para a minha formação, e que apesar dos problemas causados pela distância, sempre me deu forças e coragem para continuar superando todas as adversidades que ocorreram ao longo desta caminhada.

Um agradecimento especial a minha família, que mesmo de longe, esteve sempre presente em ligações e orações. Aos meus pais, Geraldo e Eulina, a quem digo que nenhum diploma é suficiente para demonstrar o orgulho e a admiração que tenho ao saber que ambos não precisaram de um percurso académico para ensinar aos filhos aquilo que se deve saber para crescermos como pessoas.

Ao meu irmão e a sua querida família, que em todos os momentos sempre me deram o apoio necessário para finalizar este curso. Um grande obrigado a todas as comemorações que foram feitas em meu nome, mas sem a minha presença.

Aos meus orientadores, Isabel Maria de Sousa de Jesus e Ramiro de Sousa Barbosa, agradeço a total dedicação e disponibilidade que ambos sempre apresentaram ao longo desta jornada. Por todas as explicações e revisões que foram realizadas de forma a valorizar e enriquecer o meu trabalho, e pela confiança que ambos depositaram em mim, eu vos digo um “Muito Obrigado”.

Gostaria de deixar um agradecimento a todos os amigos de curso, que ao longo destes anos foram inúmeras as vezes que esta amizade substituiu a saudade de casa. Um agradecimento especial a Jorge, Steve, Joana, Álvaro e Ricardo, por todas as vitórias que foram conquistadas depois de longas horas de estudos, e que apesar de intermináveis, eram sempre bons momentos que eu compartilhava aos vossos lados.

Para finalizar, eu dedico este trabalho a uma pessoa que possibilitou que este sonho fosse realizado, a quem eu compartilhei os bons e os maus momentos e que sempre esteve presente ao meu lado como parte da minha família. A uma pessoa a quem eu respeito e admiro a cada dia, ao Manuel Cardoso meus eternos e sinceros agradecimentos.

Resumo

Este trabalho visa propor um método de sintonia, através da aplicação de Algoritmos Genéticos, para controladores do tipo proporcional, integral e derivativo (PID). Os Algoritmos Genéticos (AGs) são uma técnica de busca estocástica, onde a procura do óptimo global é baseada no processo de evolução natural. Vários são os casos de sucesso da aplicação dos AGs em problemas de alto grau de complexidade e que apresentam regiões de grande dimensão para a busca do ponto óptimo.

Como ferramenta de programação, foi utilizado o *software* MatLab/Simulink, que possibilita ao utilizador a criação de um ambiente de modelação interactiva e a visualização dos resultados obtidos para os sistemas analisados, bem como uma biblioteca de optimização para Algoritmos Genéticos.

A técnica de sintonia proposta é aplicada inicialmente a um sistema genérico, de forma a validar a estrutura do algoritmo e tirar as conclusões necessárias para posteriormente ser aplicada a outros sistemas, com diferentes funções de transferência e diferentes complexidades dos mesmos. Os resultados das simulações serão então analisados e comparados com os métodos de sintonia clássica, e métodos de optimização existentes na literatura, como o método ITSE de Zhuang e Atherton e as rotinas de optimização existentes em MatLab.

Para finalizar o trabalho realizado, será apresentado um ambiente computacional designado por “Interface LIR”, que de modo interactiva permite a simulação dos sistemas estudados no âmbito desta tese de mestrado, bem como a configuração de novos sistemas.

Palavras-Chave

Algoritmos Genéticos, Controlo PID, Computação Evolutiva, Sintonia.

Abstract

This paper proposes a tuning method through the application of Genetic Algorithms (GAs), for the proportional, integral and derivative (PID) controller. The GAs are a stochastic search technique, based on the process of natural evolution of species. The GAs have been successfully applied in problems of high complexity and wider regions of search space.

As a programming tool, it was used the MatLab / Simulink, which enables the users to create an interactive environment for modeling and visualization of the results obtained for the being systems analyzed, as well the use of the Toolbox of genetic algorithms of MatLab.

The proposed tuning technique is initially applied to a generic system to validate the algorithm. The simulation results are then analyzed and compared with the classical methods of tuning, the ISTE optimization method of Zhuang and Atherton and the optimization functions available in Matlab.

Finally it is presented a GUI Interface designated by “Interface LIR”, that allows the simulation and analysis of the systems studied in this dissertation, as well the development of new applications.

Keywords

Genetic Algorithms, PID Control, Evolutionary Computation, Tuning.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XI
ACRÓNIMOS	XIII
1. INTRODUÇÃO	1
1.1. MOTIVAÇÃO E OBJECTIVOS	2
1.2. ORGANIZAÇÃO DO RELATÓRIO	3
2. O CONTROLADOR PID	5
2.1. INTRODUÇÃO	6
2.2. ACÇÕES DE CONTROLO.....	7
2.3. TIPOS DE CONTROLADORES	10
2.4. MÉTODOS DE SINTONIA DE MALHAS	13
2.5. CONCLUSÃO	17
3. ALGORITMOS GENÉTICOS	19
3.1. INTRODUÇÃO	19
3.2. COMPARAÇÃO ENTRE OS AGS E AS TÉCNICAS TRADICIONAIS DE OPTIMIZAÇÃO	21
3.3. FUNCIONAMENTO DOS ALGORITMOS GENÉTICOS	23
3.4. OPERADORES GENÉTICOS	25
3.5. APLICAÇÕES DOS ALGORITMOS GENÉTICOS	29
3.6. CONCLUSÕES E PERSPECTIVAS FUTURAS SOBRE OS AGS	32
4. SINTONIA DO CONTROLADOR PID	33
4.1. INTRODUÇÃO	34
4.2. SINTONIA DE ZIEGLER-NICHOLS PARA DETERMINAR OS GANHOS DO CONTROLADOR.....	35
4.3. ANÁLISE DA RESPOSTA DO SISTEMA	37
4.4. FERRAMENTA DE DESENVOLVIMENTO – MATLAB – PARA O AJUSTE DO CONTROLADOR PID	38
4.5. ESTRUTURA DO AG DESENVOLVIDO.....	39
4.6. AVALIAÇÃO DOS PARÂMETROS DO AG	43
4.7. AVALIAÇÃO DOS ÍNDICES DE DESEMPENHO	46
4.8. SINTONIA COM AS FUNÇÕES DE OPTIMIZAÇÃO DO MATLAB – <i>FMINCON</i> E <i>PATTERNSEARCH</i>	55

4.9.	ESQUEMA HÍBRIDO COM ALGORITMOS GENÉTICOS	63
4.1.	CONCLUSÃO	68
5.	APLICAÇÕES DOS ALGORITMOS GENÉTICOS HÍBRIDOS	69
5.1.	INTRODUÇÃO.....	69
5.2.	O MÉTODO DE ZHUANG E ATHERTON.....	70
5.3.	APLICAÇÃO DA METODOLOGIA PROPOSTA: EXEMPLOS ILUSTRATIVOS.....	75
5.4.	COMENTÁRIOS	87
5.5.	INTERFACE GRÁFICA PARA SINTONIA DE CONTROLADORES COM ALGORITMOS GENÉTICOS	88
6.	CONCLUSÕES E PERSPECTIVAS DE TRABALHOS FUTUROS	103
6.1.	ANÁLISE DOS RESULTADOS OBTIDOS.....	103
6.2.	CONTRIBUIÇÕES DO TRABALHO	106
6.3.	PERSPECTIVAS PARA TRABALHOS FUTUROS	106
	REFERÊNCIAS DOCUMENTAIS.....	109
	ANEXO A. SINTONIA DE CONTROLADORES PID PARA PROCESSOS DE PRIMEIRA ORDEM COM TEMPO DE ATRASO. CÁLCULO DOS PARÂMETROS POR ZHUANG E ATHERTON...113	

Índice de Figuras

Figura 1	Diagrama de blocos de um sistema de controlo com o Controlador PID.....	11
Figura 2	Controlador PID com filtro na saída [Salamanca, 2007]	13
Figura 3	Controlador PI+D com filtro na saída da parte derivativa [Salamanca, 2007].....	13
Figura 4	Método de sintonia de Ziegler-Nichols em malha aberta [Ribeiro, 2001]	15
Figura 5	Comparação da procura dos algoritmos em análise [L. Soares, 1997].....	21
Figura 6	Estrutura de funcionamento de um AG tradicional.....	23
Figura 7	Estrutura de um cromossoma [Morelli, 2006].....	24
Figura 8	Operadores de cruzamento 1 ponto	28
Figura 9	Operadores de cruzamento 2 pontos	28
Figura 10	Operação de Mutação.....	29
Figura 11	Diagrama de bloco do sistema de controlo [Ogata, 2003]	34
Figura 12	Resposta do sistema a uma entrada em degrau unitário	37
Figura 13	Diagrama de blocos para o sistema de controlo implementado em Simulink.....	39
Figura 14	Ajuste do controlador PID através do AG.....	39
Figura 15	Desempenho do AG em função do aumento do tamanho da população.....	48
Figura 16	Desempenho do controlador PID por Ziegler-Nichols (linha contínua), <i>versus</i> Desempenho do controlador PID para cada índice de desempenho com AG (linha tracejada)	51
Figura 17	População <i>versus</i> Geração para o índice de desempenho ITAE	52
Figura 18	Ponto óptimo da FO dado pelo índice ITAE.....	53
Figura 19	Resposta do sistema para o ponto óptimo dado pelo índice ITAE.....	54
Figura 20	Resposta do Sistema a uma entrada em degrau unitário para AG (linha contínua) e <i>fmincon</i> (linha tracejada).....	60
Figura 21	Resposta do Sistema a uma entrada em degrau unitário para AG (linha contínua) e <i>patternsearch</i> (linha tracejada).....	62
Figura 22	Diagrama esquemático do AGH.....	65
Figura 23	Variação do erro ao longo do aumento da população para AG, AGH1 e AGH2.....	67
Figura 24	Sistema de controlo PID com realimentação.....	71
Figura 25	Diagrama de blocos para o exemplo 1	75
Figura 26	Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH2 com índice de desempenho ITAE, para o exemplo 1	77
Figura 27	Diagrama de blocos para o exemplo 2	78
Figura 28	Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH2 com índice de desempenho IAE, para o exemplo 2.....	80
Figura 29	Diagrama de blocos para o exemplo 3	81

Figura 30	Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH2 com índice de desempenho IAE, para o exemplo 3.....	83
Figura 31	Sistema de Difusão de calor [Jesus, 2007]	84
Figura 32	Diagrama de blocos para o exemplo 4.....	84
Figura 33	Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH1 com índice de desempenho ITAE, para o exemplo 4	87
Figura 34	Menu principal da interface gráfica.....	90
Figura 35	Menu dos módulos de simulação e código M.File dos índices de desempenho.....	91
Figura 36	Módulo de aplicação da interface gráfica referente ao uso de algoritmos genéticos para a sintonia de controladores	92
Figura 37	Menu de opções para os módulos da interface gráfica.....	93
Figura 38	Exemplo de funcionamento da interface	94
Figura 39	Opções gráficas para visualizar o desempenho do AG	94
Figura 40	Módulo de configuração para um novo sistema.....	96
Figura 41	Módulo da interface referente ao critério de otimização de Zhuang e Atherton	97
Figura 42	Módulo desenvolvido para a simulação do primeiro sistema.....	98
Figura 43	Módulo desenvolvido para a simulação do segundo sistema	98
Figura 44	Módulo desenvolvido para a simulação do terceiro sistema	99
Figura 45	Módulo desenvolvido para a simulação do quarto sistema	100
Figura 46	Módulo desenvolvido para o método de otimização de Z-A.....	101
Figura 47	Controlador PID com acção derivativa no sinal de saída.....	115

Índice de Tabelas

Tabela 1	Sintonia PID pelo método de Ziegler-Nichols em malha aberta.....	15
Tabela 2	Sintonia PID pelo método de Ziegler-Nichols em malha fechada	16
Tabela 3	Método de Sintonia de Cohen e Coon [Silva, 2010].....	17
Tabela 4	Exemplos de genótipos e fenótipos correspondentes em alguns tipos de problema	24
Tabela 5	Sintonia PID pelo método de Ziegler-Nichols em malha fechada	36
Tabela 6	Parâmetros do controlador PID	36
Tabela 7	Ganho do controlador PID	36
Tabela 8	Desempenho do AG em função do aumento do tamanho da população.....	49
Tabela 9	Valores de inicialização da população do AG.....	50
Tabela 10	Parâmetros e especificações das respostas do sistema	50
Tabela 11	Valor do erro para o índice ITAE.....	53
Tabela 12	Parâmetros e especificações das respostas do sistema para Z-N, AG e <i>fmincon</i>	59
Tabela 13	Parâmetros e especificações das respostas do sistema para Z-N, AG e <i>patternsearch</i>	61
Tabela 14	Tabela de comparação dos resultados obtidos com AG, AGH1 e AGH2.....	66
Tabela 15	Parâmetros para variação do valor de referência de um controlador PI.....	72
Tabela 16	Parâmetros para variação do valor de referência de um controlador PID	72
Tabela 17	Sintonia PID com acção derivativa no sinal de saída.....	73
Tabela 18	Sintonia PI para a presença de perturbação.....	74
Tabela 19	Sintonia PID para a presença de perturbação	74
Tabela 20	Parâmetros de inicialização da população do AG para o exemplo 1.....	76
Tabela 21	Ganhos do controlador PID e especificações temporais para o exemplo 1.....	76
Tabela 22	Parâmetros de inicialização da população do AG para o exemplo 2.....	79
Tabela 23	Ganhos do controlador PID e especificações temporais para o exemplo 2.....	79
Tabela 24	Parâmetros de inicialização da população do AG para o exemplo 3.....	81
Tabela 25	Ganhos do controlador PID e especificações temporais para o exemplo 3.....	82
Tabela 26	Parâmetros de inicialização da população do AG para o exemplo 4.....	85
Tabela 27	Ganhos do controlador PID e especificações temporais para o exemplo 4.....	86

Acrónimos

- AG – Algoritmo Genético
- AGH – Algoritmo Genético Híbrido
- AM – Algoritmo Memético
- CE – Computação Evolutiva
- FOPDT – *First-Order Plus Dead Time*
- GPS – *Generalized Pattern Search*
- IAE – *Integral of Absolute Error*
- ID – Índice de Desempenho
- ISE – *Integral of the Square Error*
- IT²SE – *Integral Squared Time-Squared Weighted Error*
- ITAE – *Integral of Time Absolute Error*
- ITSE – *Integral of Time-Weighted Squared Error*
- MSE – *Mean of the Squared Error*
- PID – Proporcional, Integral, Derivativo
- PSO – *Particle Swarm Optimizacion*
- PSQ – Programação Quadrática Sequencial

1. INTRODUÇÃO

Controlar um sistema, é um assunto que tem raízes em praticamente todas as áreas do conhecimento, e com efeito, em todas as especialidades da Engenharia. Na especialidade dos Sistemas de Planeamento Industrial, o uso de sistemas de controlo dá-se em diversos níveis, podendo citar os Sistemas Integrados de Produção, Controlo de Fábrica, Células Flexíveis de Manufacturas, entre muitos outros.

Na maioria dos sistemas industriais, a melhor forma de controlar uma malha é através do uso dos controladores PID, ou suas derivações. Na literatura são várias as técnicas propostas para sintonizar controladores, como sejam os métodos de Ziegler-Nichols, Cohen-Coon [Ogata, 2003], bem como os métodos baseados na análise do erro, que serão citados e exemplificados posteriormente.

Neste trabalho será utilizado um método de optimização, diferente das técnicas mencionadas anteriormente, onde será avaliada a utilização da computação evolutiva, nomeadamente os Algoritmos Genéticos (AGs), para estabelecer a procura dos valores óptimos para os ganhos do controlador PID. A procura destes valores é realizada através da manipulação dos operadores genéticos, e a solução encontrada será avaliada em função dos índices de desempenho descritos nas funções de aptidão, que têm como objectivo minimizar o valor do erro que corresponde à diferença entre o valor medido e o valor de referência, aplicado na entrada do sistema.

O resultado da programação do AG é feito através da devolução dos três parâmetros do controlador PID, cujo objectivo é minimizar o índice de desempenho escolhido. Também será possível fazer uma análise gráfica desta sintonia, de forma a perceber se os parâmetros retornados pelo AG correspondem a um desempenho aceitável para o sistema em análise.

A ferramenta escolhida para a programação e simulação deste trabalho é o MatLab/Simulink. O MatLab (*Matrix Laboratory*) é uma aplicação desenvolvida pela MathWorks. Com esta ferramenta, é possível projectar um ambiente de desenvolvimento interactivo e visualmente amigável para o utilizador, baseado em diagrama de blocos resultantes da modelação de um sistema, bem como uma vasta biblioteca de funções matemáticas e algoritmos bastante difundidos na comunidade científica.

Para a realização deste trabalho, foi de grande utilidade o uso da *toolbox* de optimização para algoritmos genéticos no ambiente MatLab, e a possibilidade de conjugar a programação do AG com o ambiente Simulink para a sintonia dos controladores PID.

1.1. MOTIVAÇÃO E OBJECTIVOS

A motivação para estudar o tema proposto, Aplicação de Algoritmos Genéticos para a Sintonia de Controladores, deveu-se ao facto de aprofundar o conhecimento a um método de busca que se encontra emergente no panorama académico, a Computação Evolutiva (CE).

A grande diversidade dos algoritmos que compõem a CE e a vasta aplicabilidade que estes possuem, estendem-se ao domínio dos controladores, sendo nestes casos descritos como métodos de optimização alternativos, face aos métodos clássicos de sintonia.

Para os sistemas analisados neste trabalho, pretende-se avaliar um conjunto de estruturas de programação do AG, validando aspectos como robustez e facilidade de implementação computacional. A avaliação do desempenho de todos os sistemas vai incidir sobre a minimização do erro dado por diferentes índices de desempenho.

Como objectivos específicos podemos citar:

- Revisão da literatura sobre os métodos mais utilizados para resolver o problema da sintonia dos controladores.

- Estudar os diferentes Índices de Desempenho, baseados na análise do erro.
- Estudar as funções de optimização do MatLab para a resolução de problemas com restrições.
- Conjuguar as funções de optimização do MatLab com os AGs na procura do óptimo global.
- Estudar a linguagem de programação e o ambiente de desenvolvimento MatLab/Simulink.
- Desenvolver uma ferramenta onde se conjugue os diagramas de blocos elaborado em Simulink com as funções de aptidão.
- Realização de testes, para um número variado de sistemas, utilizando os AGs para determinar os ganhos do controlador e comparar os resultados obtidos com outras técnicas de sintonia existentes na literatura.
- Desenvolver um ambiente computacional, com uma interface amigável para a utilização de Algoritmos Genéticos na Sintonia de Controladores de forma a possibilitar ao utilizador realizar experiencias com os sistemas estudados neste trabalho, bem como a possibilidade de criação de novos sistemas.

1.2. ORGANIZAÇÃO DO RELATÓRIO

Esta dissertação está estruturada em seis capítulos e um anexo.

No primeiro capítulo é feita uma introdução ao assunto que será abordado neste documento, além da motivação e os objectivos que levarão à realização deste trabalho. Para finalizar este capítulo, faz-se uma descrição detalhada dos capítulos restantes e ao anexo que compõe esta tese de mestrado.

No capítulo 2, O Controlador PID, é apresentada uma introdução e os principais conceitos associados a este tipo de controlador, não pormenorizando as diversas estruturas e tipos de controladores existentes nos dias actuais. O objectivo de tal capítulo é enquadrar o leitor sobre o que representa a sintonia do Controlador PID associada ao desempenho de um sistema.

No capítulo 3, Algoritmos Genéticos, é feita uma revisão bibliográfica da literatura acerca deste assunto, especificando as principais terminologias, o método de optimização e as vantagens dos Algoritmos Genéticos para os sistemas a serem tratados.

No capítulo 4, Sintonia do Controlador PID, comparam-se os resultados obtidos pelos AGs em relação aos métodos clássicos de optimização para um sistema de terceira ordem, em malha fechada. Neste capítulo é pormenorizada a estrutura do AG, detalhando o que cada parâmetro afecta na sintonia do sistema e propondo uma estrutura que servirá de base para a sintonia de sistemas mais complexos que serão abordados no capítulo seguinte. Neste mesmo capítulo é apresentado o conceito dos Índices de Desempenho (ID), direccionados para a minimização do erro do sistema, bem como os parâmetros que serão analisados para determinar a qualidade da solução final.

No capítulo 5, Aplicação dos Algoritmos Genéticos, é utilizado um conjunto de sistemas que diferem pela sua função de transferência e os quais possuem comportamentos conhecidos, já que foram objecto de estudos de outros trabalhos. A escolha dos sistemas a serem utilizados nesse capítulo teve como critério de selecção o nível de dificuldade que estes apresentam, e a possibilidade de comparar os resultados obtidos com os AGs face a outros métodos de optimização. Para finalizar este mesmo capítulo é apresentado o ambiente computacional, desenvolvido em MatLab/Simulink, para a simulação de controladores através de AGs.

Por último, no capítulo 6, Conclusões e Perspectivas de Trabalhos Futuros, são tecidas as principais conclusões acerca deste estudo e também apresentadas algumas perspectivas para trabalhos futuros.

Para finalizar este trabalho, é apresentado no Anexo A o código gerado em MatLab, para calcular os ganhos do controlador PID com base no método de optimização de Zhuang e Atherton.

2. O CONTROLADOR PID

Inicialmente a humanidade não dispunha de meios para extrair energia a partir da matéria simples, era necessário o trabalho humano com a ajuda de animais domésticos para conseguir essa energia. Com a criação da máquina a vapor, por volta do século XVIII, foi possível transformar energia em trabalho, e nesse momento a humanidade teve a condição de obter energia advinda do trabalho mental, e não somente do trabalho físico.

Devido à baixa solicitação inicial de energia, era fácil administrá-la, mas com o passar dos tempos e o aumento das necessidades a humanidade viu-se obrigada a desenvolver métodos e equipamentos que pudessem substituir o homem nesta nova tarefa, e evitar expô-lo constantemente ao perigo devido a falta de segurança. Neste momento surge o controlo automático.

O controlador que possui três tipos de acções de controlo, nomeadamente do tipo Proporcional (P), Integral (I) e Derivativo (D) é vulgarmente designado de controlador PID. Este capítulo tem por objectivo expor os principais conceitos associados a esse tipo de controlo, de forma a permitir que o leitor tenha um conhecimento dos benefícios adquiridos por um sistema de controlo quando sintonizados por um PID.

2.1. INTRODUÇÃO

Em 1942 com o desenvolvimento do método de sintonia de Ziegler-Nichols, a utilização dos controladores PID torna-se mais popular, continuando ainda hoje a ser o principal tipo de controlo em ambiente industrial.

Existe actualmente uma grande variedade de equipamentos que podem ser utilizados em diversas malhas de controlo, e nos mais variados tipos de processos. São várias as formas de implementar um controlador PID, e na sua maioria, são controladores usados em processos com realimentação. Particularmente nos níveis mais baixos da estrutura hierárquica dos sistemas de controlo, nenhum controlador compete com a simplicidade e funcionalidade do controlador PID.

No estudo das malhas de controlo de um processo industrial realizado em 2004, na Companhia de minério de ferro, Vale do Rio Doce localizada no estado de Minas Gerais/Brasil, engenheiros e técnicos descreveram que o baixo desempenho destas malhas se deviam em parte à má sintonia dos controladores [CVRD, 2006]. Neste sentido vale a pena salientar que várias pesquisas e desenvolvimentos da tecnologia PID estão a focalizar os seus esforços na procura de novos métodos que permitam melhorar o desempenho dos controladores e sua sintonia.

Segundo [Salamanca, 2007], os parâmetros do controlador PID são usualmente otimizados para que o sistema em malha fechada cumpra os seguintes objectivos:

- Estabilidade robusta face a incertezas paramétricas e estruturais, usualmente medida no domínio das frequências.
- Resposta transitória, incluindo tempo de subida, sobreelongação e tempo de estabelecimento.
- Eliminação de perturbações na carga.
- Atenuação de ruído.

Com estes objectivos, os métodos de sintonia dos controladores podem ser agrupados de acordo com a sua natureza e uso em: Métodos Heurísticos, Métodos de resposta em

frequência, Métodos Analíticos e Métodos de Optimização Numérica. Mais informação sobre estes métodos podem ser consultadas em [Ribeiro, 2001; Salamanca, 2007].

De seguida são apresentadas as acções de controlo e os principais tipos de controladores usualmente aplicados na indústria, bem como os seus métodos de sintonia.

2.2. ACÇÕES DE CONTROLO

Pode-se entender um controlador como um dispositivo capaz de realizar certas operações matemáticas de forma a comparar o sinal de entrada da planta com a referência estipulada, determinar a diferença existente entre estes sinais, designada como o erro, e a partir deste determinar o sinal de controlo a ser aplicado ao sistema por forma a satisfazer um determinado objectivo. Estas operações matemáticas constituem o que se chama de acções de controlo, e podem ser identificadas como:

- Acção Liga-Desliga (*on-off*)
- Acção Proporcional
- Acção Integral
- Acção Derivativa

A compreensão destas 4 acções de controlo é fundamental para o entendimento da sintonia dos controladores PID. Como pressuposto deste trabalho, o texto que se segue é puramente teórico, proporcionando ao leitor uma sensibilidade básica sobre o controlador PID. Foi omissa toda a análise matemática relacionada com as acções de controlo, e consequentemente as configurações do controlador PID, visto que o assunto se estenderia de forma a prejudicar o objectivo deste trabalho. Para mais informações, poder-se-á no entanto consultar as seguintes referências bibliográficas: [Dutton, 1997; Ribeiro, 2001; Ogata, 2003; Salamanca, 2007].

2.2.1. ACÇÃO DE CONTROLO *ON-OFF* (LIGA - DESLIGA)

Existem várias referências sobre a designação da acção de controlo liga-desliga, podendo também ser citada como duas posições, *on-off*, tudo ou nada, 0-1, controlo radical, *bang-bang*.

A acção liga-desliga é discreta e não contínua. A saída do controlador só assume um de dois valores possíveis: ou 0 ou 100%. Consequentemente, a válvula de controlo só pode assumir duas posições: ou totalmente fechada (0%) ou totalmente aberta (100%). Não há posição intermediária e não há meio-termo, por isso é chamado de controlo radical. Entretanto, este tipo de acção possui limitações no que diz respeito ao comportamento dinâmico e em regime permanente do sistema em malha fechada. As suas aplicações restringem-se a sistemas onde não é necessária precisão nem um bom desempenho dinâmico. Como exemplos de aplicação deste tipo de controlo temos: termóstato do frigorífico e o controlo de nível de água utilizando bóia.

2.2.2. ACÇÃO PROPORCIONAL

Partindo do pressuposto que se tem o prévio conhecimento das características de um controlo que executa a acção liga-desliga, sabe-se que quando a variável controlada se desvia do valor ajustado, o elemento final de controlo realiza um movimento brusco de *on* (liga) para *off* (desliga), provocando uma oscilação no resultado de controlo. Para evitar tal tipo de movimento foi desenvolvido um tipo de acção no qual a acção correctiva produzida por este mecanismo é proporcional ao valor do desvio. Tal acção denominou-se de acção proporcional.

A acção proporcional é a acção correctiva do controlador que é proporcional ao valor do desvio entre a medição e o ponto de ajuste. É uma acção de controlo contínua, analógica e uniforme, sendo a saída do controlador proporcional à amplitude do erro [Ribeiro, 2001].

No controlo proporcional a válvula de controlo pode assumir qualquer valor intermédio entre 0 e 100% de abertura.

O controlador com a acção proporcional isolada é aplicado nos processos com pequena variação da carga e em processos onde pode haver pequenos desvios da medição em relação ao ponto de ajuste. De acordo com [Silva, 2010] “*o controlador proporcional é aplicado no controlo do processo onde a estabilidade é mais importante que a igualdade da medição com o ponto de ajuste*”. O nível é a variável que é tipicamente controlada apenas com a acção proporcional.

2.2.3. ACÇÃO INTEGRAL

Segundo Senai [Senai, 1999], com a acção de controlo proporcional, consegue-se eliminar as oscilações provocadas pelo controlador liga-desliga. Entretanto, ocorre o inconveniente da não eliminação do erro do sistema em regime permanente, facto que pode ser resolvido utilizando a acção de controlo integral.

A acção integral vai actuar no processo ao longo do tempo enquanto existir diferença entre o valor desejado e o valor medido. Assim, o sinal de correcção é integrado no tempo e por isto, enquanto a acção proporcional actua de forma instantânea quando acontece um distúrbio, a acção integral vai actuar de forma lenta até eliminar por completo o erro, sendo por isso o objectivo da acção integral o de eliminar o desvio permanente deixado pela acção proporcional. Devido a esta característica, a acção integral é chamada de acção *reset* ou de reajuste, pois elimina o erro residual, reajustando o valor da medição para um valor igual ao ponto de ajuste [Ribeiro, 2001].

Quando a acção integral, é associada à acção proporcional, a primeira começa a actuar logo após a segunda, ou seja, a acção integral está atrasada em relação à acção proporcional. A acção integral vai repetindo a acção proporcional continuamente até que a medição fique igual ao ponto de ajuste.

2.2.4. ACÇÃO DERIVATIVA

De acordo com Griffin [Griffin, 2003], “*a acção derivativa é a acção correctiva proporcional à derivada em relação ao tempo do erro entre a medição e o ponto de ajuste*”. A acção derivativa detecta a variação do erro entre a medição e o ponto de ajuste e fornece uma saída proporcional a esta taxa de variação.

A acção derivativa não se antecipa ao aparecimento do erro. Quando aparece um erro e ele é detectado pelo controlador, a acção derivativa sente a sua velocidade de variação e produz uma componente correctiva proporcional a esta variação [Ogata, 2003]. A acção derivativa, na realidade, atrasa a resposta proporcional do controlador, fazendo-o dar uma saída exagerada para variações bruscas da medição. A taxa da acção derivativa é proporcional à velocidade de variação do sinal medido.

Erroneamente, acha-se que todas as variáveis lentas necessitam da acção derivativa. A lentidão da variável é uma condição necessária mas não é suficiente para justificar a

aplicação da acção derivativa. É também necessário que haja variação rápida da carga do processo para exigir a aplicação da acção derivativa. O processo lento que sofre variação lenta da carga não necessita da acção derivativa [Silva, 2010].

2.3. TIPOS DE CONTROLADORES

Apesar de se ter a disponibilidade das três acções básicas, dependendo da aplicação não será necessária a utilização de uma ou mais destas acções. Basicamente existem 4 configurações possíveis de controladores a partir de uma estrutura PID, e estas configurações serão detalhadas a seguir.

2.3.1. CONTROLADOR P

O controlador proporcional ajusta o ganho do sistema, K . Se o processo tem um ganho K_p e o controlador tem ganho K_c , o ganho do sistema é:

$$K = K_c K_p \quad (1)$$

Se o sistema é instável para o ganho K_p , deve-se diminuir o ganho do controlador K_c de forma a diminuir o ganho K do sistema. Se a instabilidade não é o problema, pode usar-se o controlador proporcional para aumentar o seu ganho diminuindo o erro em regime permanente (Dutton, 1997). A equação de transferência que descreve o controlador proporcional é dada por:

$$G_c(s) = K_p \quad (2)$$

2.3.2. CONTROLADOR PI

O controlador PI, também chamado de compensador em atraso, estabiliza o processo permitindo que a variável controlada seja igual ao ponto de ajuste [Astrom, 1995]. Como este controlador possui acção integral, pode ocorrer saturação na saída do controlador. A sua resposta é relativamente lenta devido à acção da componente integral. Geralmente o controlador PI é o mais usado no controlo de processos.

Para o controlador PI a equação de transferência é dada por:

$$G_c(s) = K_p \left(1 + \frac{1}{sT_i} \right) \quad (3)$$

2.3.3. CONTROLADOR PD

O controlador PD, também chamado de compensador em avanço, permite a alteração da resposta transitória do sistema. O uso do controlador PD pode alterar um sistema sub-amortecido para um sistema criticamente amortecido ou sobre amortecido. Este tipo de controlador estabiliza rapidamente o processo, porém, normalmente ocorre um desvio em regime permanente.

Como ele não possui a ação integral, não há perigo de saturação da saída do controlador, além disso, com a ação derivativa, a sua resposta é relativamente mais rápida. A equação de transferência do controlador PD é dada por:

$$G_c(s) = K_p(1 + sT_d) \quad (4)$$

2.3.4. CONTROLADOR PID

O controle proporcional associado ao integral e ao derivativo é o tipo de controle mais sofisticado, utilizado em sistemas de malha fechada. A componente proporcional aumenta a velocidade de resposta, a integral elimina o desvio em regime permanente, enquanto a derivativa fornece ao sistema uma ação de antecipação, para evitar que o desvio se torne ainda maior [Senai, 2009]. Pode-se dizer que este tipo de controlador resulta num compromisso entre as vantagens e desvantagens de um PI e as vantagens de um PD.

Os três tipos de controle, quando combinados podem ser representados pela seguinte função de transferência:

$$G_c(s) = K \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (5)$$

Esta função pode ser ilustrada pelo diagrama de blocos apresentado na Figura 1.

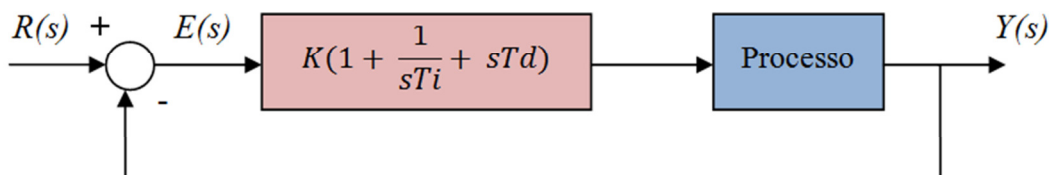


Figura 1 Diagrama de blocos de um sistema de controle com o Controlador PID

Basicamente a função do controlador PID é actuar sobre a variável a ser manipulada através de uma combinação adequada das três acções de controlo.

A maior parte dos controladores industriais utilizam a configuração PID, facto que faz deste controlador objecto de estudo para inúmeros investigadores da área. Particularmente, quando o modelo matemático do sistema não é conhecido e portanto, os métodos analíticos não podem ser utilizados, os controladores PID revelam-se como a ferramenta mais útil para esta situação.

2.3.5. CONTROLADOR PID COM FILTRO NA SAÍDA

No controlo de processos industriais, onde há a presença de dispositivos electrónicos, o ruído é um fenómeno que ocasiona problemas significativos no desempenho do controlador, visto que interfere nos sinais transmitidos ou processados. Se o ruído é de origem electromagnética, a solução pode estar na blindagem dos cabos onde são transmitidos os sinais, o que engloba um alto custo. Uma solução adequada consiste em incluir no sistema de controlo, filtros digitais sintonizados adequadamente, os quais minimizam a magnitude das componentes de ruído, sem afectar a dinâmica do processo.

Um filtro projectado para minimizar a amplitude do ruído, pode definir-se como um dispositivo electrónico, que permite a passagem das componentes de baixa frequência de um sinal e reduz a amplitude das componentes de alta-frequência (frequência maior que a frequência de corte) [Kuo, 2003].

Os filtros digitais mais usuais em sistemas de controlo são do tipo: Filtro de Primeira Ordem, Filtro de Segunda Ordem, Filtro *Noise-Spike*, Filtro Média-Móvel. Para mais informações sobre estes filtros digitais consultar [Salamanca, 2007]

Na configuração de malha PID com filtro na saída, o sinal da variável de processo é filtrado antes de ser comparado com o sinal de referência, atenuando-se as variações de alta-frequência causadas pelo ruído. Assim a derivada do erro é menor e o esforço de controlo diminui.

A Figura 2 apresenta a configuração de uma malha com filtro na saída, onde $y_r(k)$ corresponde à referência, $u(k)$ a variável de controlo, $y(k)$ a variável do processo, $d_{in}(k)$ e $d_{out}(k)$ as perturbações de entrada e saída do processo, respectivamente e $\eta(k)$ o ruído na variável de processo.

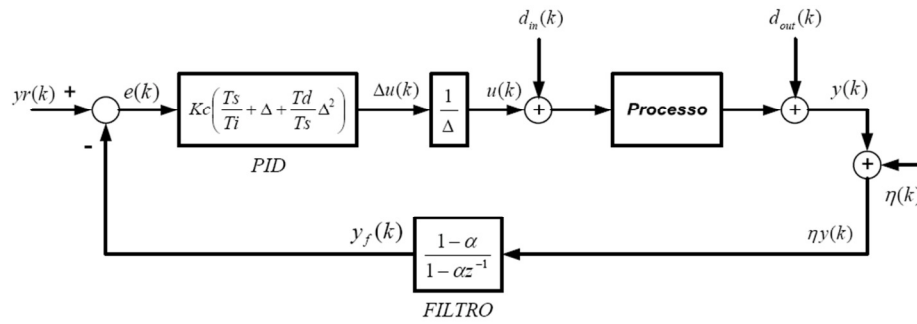


Figura 2 Controlador PID com filtro na saída [Salamanca, 2007]

2.3.6. CONTROLADOR PI+D COM FILTRO NA SAÍDA DA PARTE DERIVATIVA

Segundo Ribeiro [Ribeiro, 2001], o controlador PI+D com filtro na saída da parte derivativa, além de ter a vantagem de que a derivada só actua na variável do processo, tem outra vantagem que é a do sinal derivado ser atenuado pelo filtro, limitando o esforço do sinal de controlo, ocasionado pelas perturbações e pelo ruído.

O sistema de controlo em destaque é apresentado na Figura 3.

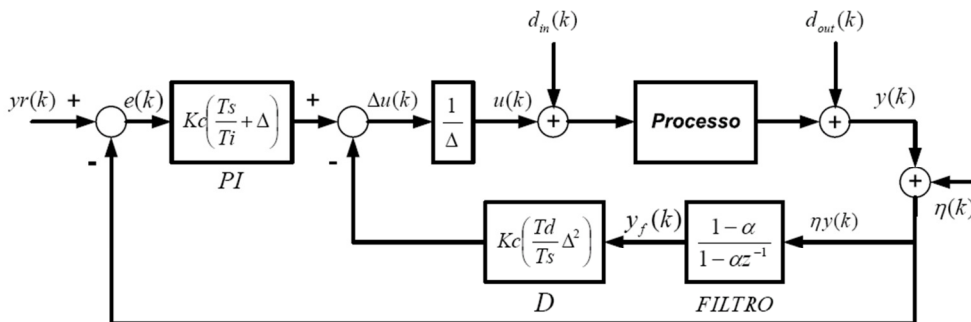


Figura 3 Controlador PI+D com filtro na saída da parte derivativa [Salamanca, 2007]

Existem outras configurações de malha PID com filtros, que não serão abordadas nesta dissertação, entretanto, se o assunto focar o interesse do leitor, o mesmo pode recorrer a [Astrom, 1995; kuo, 2003], onde encontram informações sobre tais configurações, assim como a análise matemática das estruturas apresentadas nas Figuras 2 e 3.

2.4. MÉTODOS DE SINTONIA DE MALHAS

Os principais métodos utilizados para sintonizar as malhas de controlo do tipo realimentado são descritos nas próximas subsecções.

2.4.1. MÉTODO DE APROXIMAÇÕES SUCESSIVAS OU TENTATIVA E ERRO

O método de tentativa e erro consiste em observar os efeitos provocados na variável de processo, a partir das modificações efectuadas nas acções de controlo. A grande desvantagem deste procedimento é que em algumas situações ele torna-se demorado, por causa dos períodos de oscilação excessivamente longos da planta.

Para a utilização deste método se faz necessário um prévio conhecimento sobre o sistema que se deseja controlar, e neste caso a experiência do projectista é fundamental para minimizar os efeitos negativos causados por este tipo de sintonia, que na maioria dos casos demanda elevados gastos com um má qualidade do produto final.

Em muitos casos, pode ser necessário fazer uma grande alteração na planta para obter o grau de operação desejado. É conveniente que o engenheiro de controlo possa utilizar procedimentos que lhe forneçam informação sobre a estabilidade do sistema antes da construção da planta real, eliminando alterações custosas no projecto, que serão necessárias depois de a planta ter sido construída.

2.4.2. MÉTODO DE SINTONIA DE ZIEGLER E NICHOLS EM MALHA ABERTA

De acordo com Ribeiro [Ribeiro, 2001], o método em destaque resume-se nos seguintes passos:

- Determinação dos valores K_c , T_i e T_d , a partir das características da resposta transitória da planta do sistema.
- Com este método pretende-se obter no máximo 25% de sobreelongação.
- O método de sintonia de Ziegler-Nichols em malha aberta só pode ser aplicado a plantas que não envolvam nem integradores, nem pólos complexos conjugados.
- Caso as condições anteriores se confirmem, então a curva da resposta a um degrau assemelhar-se-á a uma curva em forma de S, tal como é apresentada na Figura 4. Caso a curva não tenha esta forma, então este método de sintonia não pode ser aplicado.
- As curvas de resposta ao degrau podem ser obtidas experimentalmente.

Estas curvas são caracterizadas por um tempo de atraso L e uma constante de tempo T , tal como se pode observar na Figura 4.

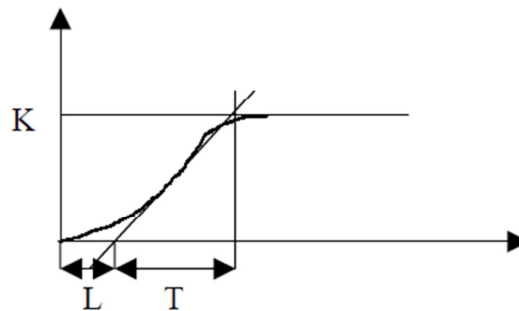


Figura 4 Método de sintonia de Ziegler-Nichols em malha aberta [Ribeiro, 2001]

Se a resposta do sistema satisfaz esta condição, o modelo do processo pode ser aproximadamente determinado pela função de transferência, dada por.

$$G(s) = \frac{K e^{-sL}}{Ts + 1} \quad (6)$$

em que L representa o tempo de atraso, T a constante de tempo e K o ganho da função de transferência. Tais valores podem se obtidos directamente da curva de resposta como indicado na Figura 4.

A Tabela 1 apresenta as regras de sintonia de vários tipos de controladores PID pelo método de Ziegler-Nichols:

Tabela 1 Sintonia PID pelo método de Ziegler-Nichols em malha aberta

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \left(\frac{T}{L}\right)$	$\frac{L}{0,3}$	0
PID	$1,2 \left(\frac{T}{L}\right)$	$2L$	$0,5L$

2.4.3. MÉTODO DE SINTONIA DE ZIEGLER-NICHOLS DE MALHA FECHADA

Este método baseia-se na observação da resposta do processo e do conhecimento da estrutura do controlador. É um dos métodos que permite o cálculo das acções de controlo

sem a necessidade de se conhecerem os parâmetros do processo. Este método é indicado para processos estáveis, mas não se adapta muito bem a processos com tempo de atraso elevado.

O método consiste em colocar a malha de controle em oscilação. O período das oscilações P_{cr} e o ganho crítico do controlador K_{cr} , que ocasiona as oscilações, permitem os cálculos das ações a serem fixadas no controlador. O cálculo depende da estrutura do controlador utilizado e do modo de regulação escolhido (P, PI e PID).

De acordo com Senai [Senai, 1999], este método pode ser resumido em:

- Determinar K_{cr} crítico, considerando-se apenas o ganho proporcional, isto é: fazendo T_d igual a zero e T_i igual a infinito.
- Determinar a frequência de oscilação.
- Determinar o período crítico.

A Tabela 2 apresenta as regras de sintonia pelo método de Ziegler-Nichols em malha fechada para os controladores P, PI e PID.

Tabela 2 Sintonia PID pelo método de Ziegler-Nichols em malha fechada

Tipo de Controlador	K_p	T_i	T_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$\frac{1}{2}P_{cr}$	$0,125P_{cr}$

2.4.4. MÉTODO HEURÍSTICO DE COHEN E COON

Assim como o método de sintonia de Ziegler-Nichols o modelo da planta para o método de Cohen e Coon é considerado como uma função de transferência de primeira ordem com tempo de atraso, dado pela equação (6).

O método proposto por Cohen e Coon, para os controladores do tipo P, PI, PD e PID resumem-se na Tabela 3:

Tabela 3 Método de Sintonia de Cohen e Coon [Silva, 2010]

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{1}{K} \left(\frac{T}{L} + \frac{1}{3} \right)$	0	0
PI	$\frac{1}{K} \left(\frac{0,9T}{L} + \frac{1}{12} \right)$	$3L \left(\frac{10 + \frac{L}{T}}{9 + \frac{20L}{T}} \right)$	0
PD	$\frac{1}{K} \left(\frac{1,25T}{L} + \frac{1}{6} \right)$	0	$2L \left(\frac{3 - \frac{L}{T}}{22 + \frac{2L}{T}} \right)$
PID	$\frac{1}{K} \left(\frac{4T}{3L} + \frac{1}{4} \right)$	$L \left(\frac{32 + \frac{6L}{T}}{13 + \frac{8L}{T}} \right)$	$\frac{4L}{11 + \frac{2L}{T}}$

A vantagem deste método deve-se ao facto que, determinada a curva de reacção, os parâmetros do controlador podem ser ajustados imediatamente, sendo de grande utilidade para processos que possuem um elevado tempo de estabelecimento.

Cabe ressaltar, que tais métodos não são exactos, e que na maior parte das vezes, conduzem às primeiras aproximações para os ganhos dos controladores, necessitando posteriormente de serem ajustados as especificações de cada malha de controlo.

Inúmeras derivações surgiram a partir destes métodos de sintonia, entre as quais: Método de sintonia Takahashi, Método de sintonia Lambda, Método de sintonia Haalman ou Método da Alocação de pólos. Caso seja do interesse do leitor a análise dos métodos citados, consultar as seguintes referências bibliográficas [Lourenço, 1997; Ogata, 2003].

2.5. CONCLUSÃO

Este capítulo fez uma revisão sobre a sintonia de controladores do tipo PID. Reconhece-se a importância deste assunto para engenheiros envolvidos no controlo processos industriais. O controlo de sistemas é um assunto que possui vertentes em praticamente todos os campos de conhecimento e, em todas as áreas da Engenharia. São vários os documentos literários que abordam este tema, tendo as referências citadas neste trabalho a intenção de proporcionar ao leitor uma sensibilidade sobre o assunto e direccioná-lo a um estudo mais aprofundando, caso seja de interesse para o mesmo.

Foram abordados alguns métodos de sintonia tradicionais, como por exemplo os métodos que se baseiam no estudo das oscilações permanentes do sistema em malha fechada e os métodos baseados na curva de resposta do processo em malha aberta. Entretanto, inúmeros outros merecem destaque, como os métodos fundamentados no desempenho óptimo do sistema em malha fechada, os quais serão alvo de aplicação nos capítulos seguintes.

3. ALGORITMOS GENÉTICOS

Este capítulo destina-se a apresentar os Algoritmos Genéticos (AGs) e alguns conceitos relacionados com o tema, de forma a construir uma base teórica para posterior discussão das técnicas aplicadas na resolução de problemas. Serão também apresentados os conceitos de algoritmos evolutivos e técnicas de optimização, assim como alguns exemplos da vasta gama de aplicabilidade dos AGs. Para finalizar serão apresentadas outras técnicas de optimização, que podem ser conjugadas com os AGs, e que ajudam na busca da melhor solução.

3.1. INTRODUÇÃO

Há muito tempo que a natureza tem sido objecto de inspiração para as leis que regem o nosso dia-a-dia e que foram formuladas com base na observação do meio onde vivemos, como exemplo temos as leis da termodinâmica e as leis de Newton. Como exemplos de objectos inspirados na natureza podemos citar os aviões (pássaros), sonares (morcegos), coletes à prova de balas (teias de aranha), velcro (plantas).

Na mesma filosofia de criação, surgem os Algoritmos Genéticos como uma área de pesquisa que se insere no ramo das técnicas de Inteligência Artificial (IA), cuja base para a resolução dos problemas fundamenta-se nas teorias desenvolvidas por Charles Darwin a respeito da evolução natural das espécies. Existem actualmente inúmeras áreas de pesquisa

que fazem parte da computação natural, e além dos algoritmos genéticos também podem ser citadas Lógica Nebulosa, Inteligência de Enxame e as Redes Neurais Artificiais.

O conceito dos AGs foi introduzido por John Holland, e pelos seus colegas da Universidade de Michigan nos Estados Unidos entre os anos de 1960 e 1970 [Mitchell, 1999], e hoje a sua popularidade vem ganhando espaço com inúmeros trabalhos e aplicações, principalmente em projectos de optimização desenvolvidos ao longo dos últimos anos.

Os métodos de optimização são geralmente classificados em programação Linear e não-Linear, sendo este último por sua vez subdividido em três grupos: Numerativos, Determinísticos e Estocásticos.

Os métodos numerativos consistem em garantir a busca do óptimo global realizando uma busca completa em todo o espaço de soluções admissíveis, e como tal tem a sua eficiência prejudicada em casos que apresentam grandes regiões de pesquisa.

Nos métodos determinísticos, a busca do ponto óptimo é local, ou seja, a busca ocorre nas proximidades do ponto, utilizando derivadas da função. A sua aplicação é mais recomendada em problemas com funções contínuas, convexas e unimodais [C.Lucas, 2002].

Por último os métodos estocásticos utilizam regras de probabilidade para realizar a procura no espaço de soluções e neste método, encontram-se os AGs. Inicialmente o AG não possui conhecimento sobre a solução do problema, e por meio de operadores como cruzamento e mutação, aplicados sobre uma população, é formado então subpopulações que se distribuem não só em torno da solução global, como também em outras regiões do problema. A Figura 5 apresenta o método de busca dos AGs em comparação com outros métodos estocásticos, nomeadamente *Simulated Annealing* (SA) e busca Tabu (Tabu).

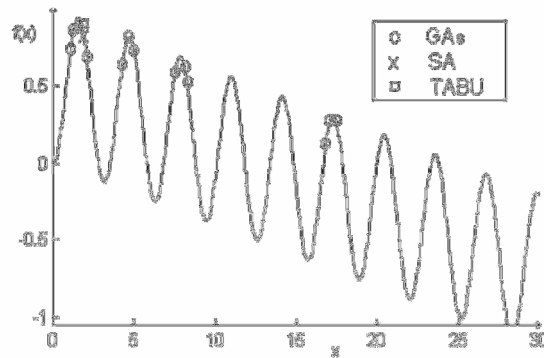


Figura 5 Comparação da procura dos algoritmos em análise [L. Soares, 1997]

Ao comparar os três conjuntos de algoritmos de otimização, podemos avaliar o desempenho de acordo com aquilo que é desejado. Ou seja, se o tempo computacional não for um factor preponderante, os métodos numéricos são a melhor escolha, já que procuram encontrar a solução global. Entretanto, se o problema aceitar como resposta qualquer solução compreendida dentro do espaço de soluções admissíveis, associada a uma rápida convergência, então os métodos determinísticos são os mais adequados. E finalmente sendo o problema com um nível de complexidade mais elevado e necessitando de uma solução global em um tempo computacional razoável, as técnicas estocásticas são as mais indicadas.

3.2. COMPARAÇÃO ENTRE OS AGS E AS TÉCNICAS TRADICIONAIS DE OPTIMIZAÇÃO

Segundo Mitchell [Mitchell, 1999], ao contrário das técnicas tradicionais de otimização que manipulam a solução do problema por meio de um único candidato, as técnicas de computação evolutiva operam sobre uma população de candidatos em paralelo e realizam uma busca em diferentes áreas do espaço de soluções.

Os AGs diferem das técnicas tradicionais em quatro aspectos [Reis, 2009]:

- “ *Os AG’s trabalham com uma codificação do conjunto de parâmetros, não com os próprios parâmetros do problema;*
- *Os AG’s realizam a busca através da evolução de uma população de pontos (filosofia concorrencial ou distribuída), e não baseados apenas num único ponto;*

- *Os AG's utilizam apenas a informação dada pela função objectivo, não a das suas derivadas ou qualquer outro conhecimento auxiliar;*
- *Os AG's usam regras de transição probabilísticas, não usam regras determinísticas ”.*

Em [Solteiro Pires,1998] destacam-se as seguintes vantagens:

- *“As restrições do problema são facilmente tratadas, inserindo-as na palavra de codificação;*
- *Os AGs são uma técnica capaz de resolver problemas multinodais, não diferenciáveis, não contínuos, não polinomiais, multidimensionais e redundantes;*
- *Os AGs estruturados providenciam uma ferramenta para otimizar a topologia ou a estrutura paralela com os parâmetros da solução, para um problema particular;*
- *Os AGs executam uma pesquisa multidireccional, mantendo uma população de soluções potenciais, encorajando a formação de informações e troca entre essas direcções ”.*

Apesar de os AGs serem aplicados a uma série de problemas que, até então, não eram facilmente resolvidos com técnicas tradicionais de optimização, eles possuem certas limitações que de acordo com Reis [Reis, 2009] podem ser destacadas em:

- Dificuldades em definir correctamente a representação para o problema;
- Pode ocorrer convergência prematura;
- O problema da escolha dos diversos parâmetros: tamanho da população, taxa de cruzamento, taxa de mutação, método de selecção;
- Não é bom para identificar óptimos locais;
- Não existe uma condição de fim eficaz;
- Requerem grande número de avaliações da função objectivo.

3.3. FUNCIONAMENTO DOS ALGORITMOS GENÉTICOS

Inspirado na maneira como o Darwinismo explica o processo de evolução das espécies, [Holland, 1975], no seu livro “*Adaptation in Natural and Artificial Systems*” decompôs o funcionamento dos AGs, como uma abstracção da evolução biológica cujas etapas se compreendem em inicialização, avaliação, selecção, cruzamento, mutação, actualização e finalização, conforme mostra a Figura 6.

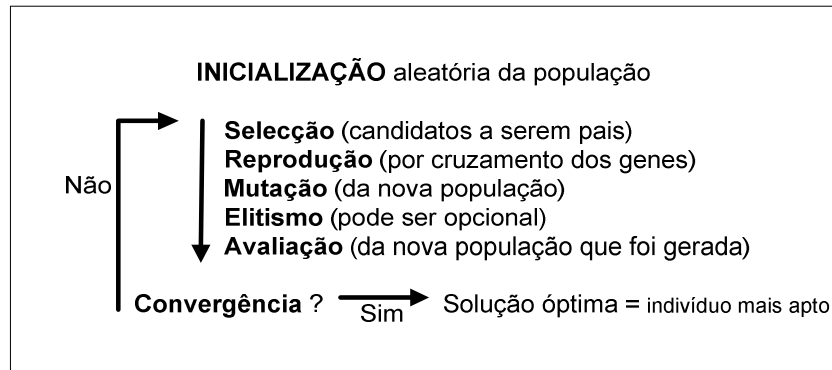


Figura 6 Estrutura de funcionamento de um AG tradicional

Basicamente, o que um AG faz é criar uma população de possíveis respostas para o problema a ser tratado para depois, submete-la ao processo de evolução, constituído pelas seguintes etapas:

1. **Avaliação:** avalia-se a aptidão das soluções (indivíduos da população). É feita uma análise para que se estabeleça quão bem elas respondem ao problema proposto.
2. **Seleção:** indivíduos são seleccionados para a reprodução. A probabilidade de uma dada solução i ser seleccionada é proporcional à sua aptidão.
3. **Cruzamento:** os pais são cruzados, com uma determinada probabilidade, para gerar os descendentes. Se nenhum cruzamento for realizado, a prole é a cópia exacta dos pais.
4. **Mutaçao:** características dos indivíduos resultantes do processo de reprodução são alteradas, acrescentando assim variedade à população.
5. **Actualização:** os indivíduos criados nesta geração são inseridos na população.

6. **Finalização:** verifica se as condições de encerramento da evolução foram atingidas, retornando para a etapa de avaliação em caso negativo e encerrando a execução em caso positivo.

3.3.1. REPRESENTAÇÃO GENÉTICA

Antes da definição dos operadores e da lógica da programação é necessário identificar a forma como serão apresentadas as possíveis soluções do problema. Segundo Davis [Davis,1992] é através da correcta codificação e manipulação dos indivíduos presentes na população dos AGs que é possível encontrar respostas para o problema. Geralmente este indivíduos são representados por vectores com valores binários, mas também podem ser utilizado números reais ou qualquer outro tipo representação.

Como sinónimo para tais indivíduos, em AG também pode ser utilizados os termos genoma ou cromossoma. Fazendo uma analogia aos sistemas naturais, um ou mais cromossomas juntam-se para formar uma carga genética com todas as características do indivíduo. Para este conjunto de cargas genéticas dá-se o nome de genótipo. O organismo formado pela interacção de toda a carga genética denomina-se de fenótipo. A Tabela 4 apresenta alguns exemplos de genótipos e fenótipos e sua correlação com alguns tipos de problema.

Tabela 4 Exemplos de genótipos e fenótipos correspondentes em alguns tipos de problema

Genótipo	Fenótipo	Problema
10100011011010	10468	Optimização Numérica
FACGBED	Iniciar na cidade F, depois passar pelas cidades A, C, G, B, E e terminar na cidade D	Caixeiro-viajante

A Figura 7 apresenta uma metáfora simplificada da estrutura de um cromossoma.

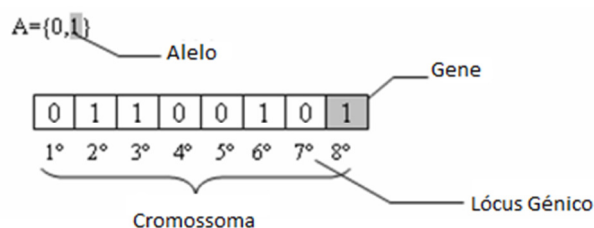


Figura 7 Estrutura de um cromossoma [Morelli, 2006]

3.3.2. GERAÇÃO DA POPULAÇÃO

A população inicial pode ser obtida aleatoriamente ou através de heurísticas relacionadas com as características do problema. Conforme acontece na natureza, o processo de evolução deve-se à dinâmica das populações que propagam boas características às gerações posteriores enquanto novas características são testadas.

Num AG é necessário gerar uma população inicial que cubra a maior área possível do espaço de busca, uma vez que uma população pequena pode não representar adequadamente este espaço e fazer com que o algoritmo se direcione para uma solução prematura que esteja distante da solução óptima. Por outro lado, populações grandes podem necessitar de muito tempo de processamento tornando o tempo de espera por uma solução demasiado longo.

As características presentes da população dos AGs são:

- **Geração:** diz respeito ao número de vezes pelas quais a população passou pelo processo de selecção, reprodução, mutação e actualização;
- **Média de adaptação:** representa o quão próximo a média de adaptação da geração actual está das suas anteriores;
- **Diversidade:** mede o grau de variação entre os genótipos presentes na população. A diversidade é fundamental para a amplitude da busca, a sua diminuição está fortemente vinculada ao fenómeno de convergência prematura;
- **Elite:** é composta pelos indivíduos mais adaptados da população. Uma técnica muito usada nos AGs é a do elitismo, onde os melhores indivíduos são sempre mantidos nas novas gerações.

3.4. OPERADORES GENÉTICOS

Os operadores genéticos fornecem o mecanismo de pesquisa básica do AG, que são usados para criar novas soluções baseadas em soluções existentes na população. Os operadores genéticos são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores. Segue abaixo uma descrição dos operadores genéticos envolvidos na estrutura de um AG tradicional.

3.4.1. INICIALIZAÇÃO

A inicialização básica de um AG resume-se na geração de uma população inicial, sobre a qual serão aplicadas as acções dos passos subseqüentes do processo. Tal operador consiste na criação de uma população que pode ser gerada de forma aleatória ou até mesmo de forma determinística, na qual os indivíduos são gerados de acordo com uma função heurística [Solteiro Pires, 1998].

Caso a inicialização da população não represente adequadamente o espaço de busca, esta poderá convergir prematuramente, isso significa que num curto espaço de tempo a população possuirá indivíduos muito semelhantes, ou seja, com pouca diversidade genética. Após a criação da população inicial, é necessário avaliar todos os cromossomas gerados, o que é efectuado com o próximo operador genético a ser descrito.

3.4.2. AVALIAÇÃO

Nesta etapa cada indivíduo da população é avaliado segundo uma função objectivo, para que seja determinada a qualidade da solução gerada por este indivíduo. A função objectivo é a única ferramenta que o AG dispõe para resolver um problema de optimização, não havendo necessidade de informações adicionais como a continuidade e a existência de derivadas da função objectivo do problema. Actualmente, várias formas de avaliação são utilizadas: em casos de optimização de funções matemáticas, o próprio valor de retorno destas tende a ser escolhido, e em problemas com muitas restrições, as mais comuns são as funções baseadas em penalidades [C. Lucas, 2002].

3.4.3. SELECÇÃO

O objectivo desta etapa é simular o processo de selecção natural, onde apenas os mais fortes sobrevivem e possibilitam a perpetuação das boas características da espécie. Assim como acontece nos processos naturais, a selecção é um método cumulativo, pois os benefícios gerados por este processo são mantidos na próxima geração.

Nesta etapa, fazendo uso do grau de adequação de cada indivíduo, é efectuado um sorteio onde os mais aptos possuem maior probabilidade de se reproduzirem. Este grau é calculado a partir da avaliação de cada indivíduo, e determina aquele que está mais apto à reprodução em relação à população a que pertence.

Uma selecção muito forte levará a que indivíduos sub-óptimos altamente aptos tomem conta da população, reduzindo a diversidade necessária para a evolução e o progresso. Uma selecção muito fraca resulta numa evolução muito lenta [Reis, 2009].

Dentre os métodos que existem para formar a selecção, os principais são:

- Selecção por *Ranking (rank selection)*: Os indivíduos da população são ordenados de acordo com o seu valor de adequação sendo-lhes atribuída uma probabilidade de escolha conforme a posição que ocupam;
- Selecção por Roleta (*roulette wheel selection*): Os pais são seleccionados de acordo com a sua adequação. Quanto melhores os cromossomas, mais hipóteses terão de serem seleccionados [AG_toolbox, 2009]. Compara-se a uma roleta onde são colocados todos os cromossomas da população. A secção da roleta é proporcional ao valor da adequação de cada cromossoma: quanto maior for esse valor, mais larga será a secção;
- Selecção por Torneio (*tournament selection*): Ao contrário da selecção por roleta, a selecção por torneio fornece uma pressão selectiva, aplicando uma competição por torneio entre os N indivíduos da população. O melhor indivíduo do torneio é o indivíduo com a função de avaliação mais elevada (é o vencedor do torneio). Os outros competidores são novamente colocados no grupo reprodutor (*matting pool*). O torneio repete-se até serem seleccionados todos os reprodutores [Reis, 2009].
- Selecção Uniforme: Neste tipo de selecção todos os indivíduos possuem a mesma probabilidade de serem seleccionados [AG_toolbox, 2009]. Obviamente, esta forma de selecção possui uma probabilidade muito remota de causar uma melhoria da população sobre a qual actua.

Para mais detalhes sobre os métodos de selecção acima citados, e outros métodos de selecção que não foram mencionados nesta secção consultar [Goldberg,1989; Mitchell, 1999].

3.4.4. REPRODUÇÃO

Nos AGs, a reprodução assemelha-se ao mecanismo de reprodução biológica, em que criaturas da mesma espécie acasalam e dão origem a uma nova criatura. O cruzamento é o operador responsável pela recombinação das características dos pais durante a reprodução,

permitindo que as próximas gerações herdem essas características. É considerado o operador genético predominante, por isso é aplicado com uma probabilidade dada pela taxa de cruzamento, que deve ser maior que a taxa de mutação [Reis, 2009].

Abaixo serão apresentados dois operadores de cruzamento mais usuais nos AGs.

Um ponto: um ponto de cruzamento é escolhido aleatoriamente e a partir deste ponto as informações genéticas dos pais são trocadas, conforme pode ser observado na Figura 8.

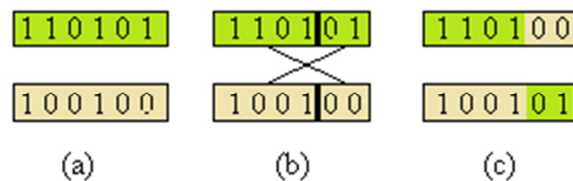


Figura 8 Operadores de cruzamento 1 ponto

Dois pontos: De acordo com a Figura 9, são escolhidos dois pontos de cruzamento (aleatoriamente) e as informações genéticas dos pais são trocadas.



Figura 9 Operadores de cruzamento 2 pontos

Para mais informações sobre o método de reprodução dos AGs podem ser consultadas as seguintes referências bibliográficas [Goldberg, 1989; Davis, 1992].

3.4.5. MUTAÇÃO

O operador de mutação é necessário para a manutenção e introdução de variedade genética na população. Com este operador altera-se um ou mais genes da estrutura para um alelo escolhido aleatoriamente [Morelli, 2006].

Algumas das formas de mutação mais usuais são a mudança de um gene escolhido aleatoriamente, troca de genes entre *locus genéticos* e inserção do valor de um gene num outro *locus genético* aleatório.

A Figura10 representa o processo de mutação em um cromossoma binário.

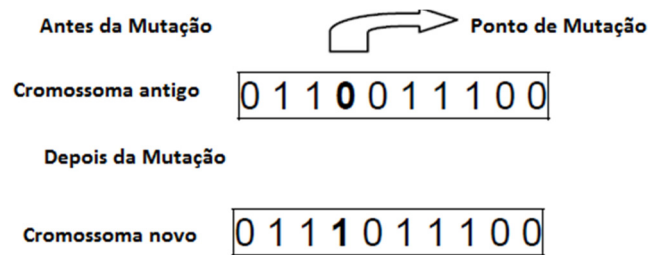


Figura 10 Operação de Mutação

3.4.6. ACTUALIZAÇÃO

Na actualização a população antiga é substituída por uma nova população, formada pelo cruzamento dos indivíduos seleccionados da população anterior. As formas mais conhecidas de actualização são $(x + y)$ e (x, y) , também chamadas de estratégia soma e estratégia vírgula [Bittencourt, 2009]. Na estratégia soma os indivíduos da população anterior convivem com a população formada pelos seus filhos. Essa estratégia também pode ser chamada de elitismo e geralmente uma percentagem muito pequena é seleccionada para a próxima geração, pois corre-se o risco de uma convergência prematura do AG. Na estratégia vírgula, a população anterior não convive com a próxima população, perdendo-se as boas soluções encontradas. Na forma mais tradicional a população mantém um tamanho fixo e os indivíduos são criados em igual número aos seus antecessores.

3.4.7. FINALIZAÇÃO

O operador de finalização é o responsável por determinar se a execução do AG (evolução da população) será concluída ou não. Tal acção é realizada a partir da execução de testes baseados numa condição de paragem pré-estabelecida. Os critérios para a paragem podem ser vários, desde o número de gerações criadas até o grau de convergência da actual população. Em suma, pode-se dizer que a finalização não envolve o uso de nenhum operador genético: ela simplesmente é composta por um teste que dá fim ao processo de evolução, caso o AG tenha chegado a algum ponto pré-estabelecido de paragem.

3.5. APLICAÇÕES DOS ALGORITMOS GENÉTICOS

Por se tratar de uma técnica bastante difundida e geralmente oferecer bons resultados, devido principalmente à rápida evolução dos computadores, que tornarão as aplicações dos

AG mais viáveis, estes são usados numa vasta gama de problemas. A seguir, são apresentados alguns casos de sucesso da utilização dos AGs.

A. **Música:** Problemas computacionais difíceis de serem resolvidos, como por exemplo a transcrição automática de música, que se resume em transcrever uma música para uma pauta, segundo um determinado acorde, tempos, rimas, etc. Novas abordagens para este problema têm sido apresentados com a utilização de AG. Em 1999 na CEC 99 – IEEE – *International Conference on Evolutionary Computation* foi apresentada uma técnica, utilizando Algoritmos Genéticos, para a avaliação de músicas (sequências de acordes). A avaliação é feita com base em factores relevantes à audição de músicas, dividindo a população em grupos de quatro vozes (soprano, contralto, tenor e baixo) ou coros, e a avaliação é feita seguindo critérios como melodia, harmonia e oitavas. A composição destes três critérios dá-nos a aptidão (*fitness*) definida pela função de selecção, a qual retorna o melhor indivíduo ou melhor coro. O sistema criado foi denominado “*Vox Populi*”. Mais informações disponíveis em:

(<http://www.lem.ep.usp.br/Pef411/%7ECristiano%20Oliveira/CristianoOliveira.htm>)

B. **Reconhecimento de faces humanas através de Algoritmo Genético:**“(…) *Na emergente busca por sistemas automáticos que realizem o reconhecimento de faces humanas, pesquisadores têm agrupado diferentes técnicas para obterem resultados que possam ser cada vez mais robustos e ágeis. Essa preocupação não se restringe aos meios académicos, também existe interesse legal e comercial em utilizar sistemas que reconheçam faces humanas como por exemplo, que locais em que seja difícil a identificação de pessoas pelos seres humanos*” [A. Barberá, 2009].

Este trabalho teve como objectivo desenvolver um sistema de reconhecimento automático de faces por intermédio das suas proporções, aplicando uma técnica, em que, por meio de um Algoritmo Genético, que tem como entrada o número de segmentos áureos encontrados na face, é possível determinar as proporções existentes e, assim, encontrar a face desejada. Os resultados demonstram que a Razão Áurea, utilizada desde a antiguidade por pintores e escultores, está presente nas proporções das faces humanas, a qual fornece um meio de medida biométrica que permite a identificação de um indivíduo, mesmo com expressões faciais.

- C. **Engenharia de Controlo:** Assim como nas áreas de aplicação já apresentadas, os AGs destinados a área de Engenharia de Controlo tem tido grande desenvolvimento. Em projectos de sistemas de controlo, questões como o desempenho, estabilidade do sistema, tratamento de sinais sujeitos a determinadas restrições, eliminação de ruídos ou qualquer outro tipo de perturbações são factores levados em conta para o correcto desempenho do processo. Contudo cada uma destas questões depende fortemente da estrutura do controlador e seus parâmetros. Entretanto tais processos podem não ser definidos com uma formulação matemática pelo que se utilizam aproximações ao sistema real, cujo desempenho final pode não reflectir o melhor desempenho para o controlador. Como exemplos, cita-se a aplicação dos AGs em Controlo Multibjectivo, Controlo PID e Controlo Robusto.
- D. **Robótica:** Vários estudos demonstram um conjunto de aplicações dos AGs nos sistemas robóticos, e neste trabalho serve como citação a dissertação apresentada por [Solteiro Pires, 1998] onde o autor apresenta os resultados da investigação na área do planeamento de trajectórias para manipuladores robóticos planares, de dois, três e quatro graus de liberdade. “ *O estudo descreve e analisa técnicas, baseadas em AGs, para o planeamento de trajectórias, utilizando a cinemática no espaço operacional. O algoritmo ainda otimiza os deslocamentos das juntas do robô e minimiza a ondulação residual da cinemática diferencial das juntas*” [Solteiro Pires, 1998].
- E. **Outras Aplicações:** De ressaltar que são inúmeros os casos de sucesso da aplicação dos AGs, tanto em problemas considerados difíceis de serem otimizados, ou em casos em que os métodos tradicionais de optimização falham. Um dos principais sucessos dos AGs e de toda a computação evolutiva, deve-se ao facto de que inúmeros problemas, antes resolvidos por meio de técnicas tradicionais de optimização, possuem agora diferentes abordagens no que se refere a busca da melhor solução, quando utilizado os AGs. Como exemplos podem ser citados:
- Alocação de tarefas;
 - Configuração de sistemas complexos;
 - Selecção de rotas;

- Problemas de otimização e de aprendizagem de máquina;

Inúmeras aplicações dos Algoritmos Genéticos podem ser encontradas em [Reis, 2009; Morelli, 2006; Mitchell, 1999].

3.6. CONCLUSÕES E PERSPECTIVAS FUTURAS SOBRE OS AGS

No decorrer do capítulo foi apresentado o conceito de Algoritmos Genéticos como mecanismos de procura estocástica, direccionado para obtenção das melhores soluções parciais, baseado nos processos da genética e da selecção natural. Neste sentido, as possíveis soluções são representadas por indivíduos e a função de aptidão representa um ambiente de sobrevivência, e através de ajustes dos parâmetros do próprio AG são geradas novas soluções parciais direccionadas à obtenção de um óptimo global do problema tratado.

A partir do AG proposto por Holland, várias inovações ocorreram no decorrer dos anos, entre elas os vários tipos de cruzamento, selecção, técnicas de convergência e também outras técnicas para a codificação dos AGs (técnicas determinísticas).Tão importante como estas inovações é a própria evolução do Algoritmo Genético no sentido de suprimir as falhas, a partir das suas derivações.

Devido à sua vasta aplicabilidade, as perspectivas futuras dos AGs são as mais promissoras. Actualmente os investigadores tentam integrar vários métodos combinando as suas vantagens, tanto quanto possível. Como metas existem esquemas de como um AG poderá controlar os parâmetros de outro AG, embora não se tenha chegado a uma implementação prática.

4. SINTONIA DO CONTROLADOR PID

Este capítulo destina-se a apresentar os resultados obtidos para um sistema em malha fechada, ao qual foram aplicadas técnicas de sintonia para o controlador PID. A metodologia utilizada baseia-se em comparar os resultados obtidos pelo AG com as técnicas tradicionais de optimização, nomeadamente as rotinas disponibilizadas pelo MatLab, e outros métodos óptimos de sintonia existentes na literatura.

Neste mesmo capítulo será avaliado o código implementado para o AG, dando ênfase a cada parâmetro de configuração necessário para o bom desempenho do sistema. Da mesma forma serão analisadas as formulações matemáticas dos índices de desempenho, bem como o código criado para a avaliação do AG, o qual consiste na minimização do erro entre a referência e a saída do sistema.

Para complementar o trabalho, o AG será conjugado com outra função de optimização com o objectivo de se obterem melhores resultados na procura da solução global. Neste sentido, será utilizado um Algoritmo Genético Híbrido (AGH), formado pelo AG com as rotinas de optimização do MatLab.

4.1. INTRODUÇÃO

Para dar início ao desenvolvimento deste projecto, foi escolhido o sistema de controlo apresentado na Figura 11, no qual é utilizado um controlador PID para o controlo do sistema. Embora se disponha de muitos métodos analíticos para determinar os parâmetros do controlador PID, para o presente estudo será aplicada a regra de sintonia de Ziegler-Nichols em malha fechada, para determinar os valores dos ganhos do controlador.

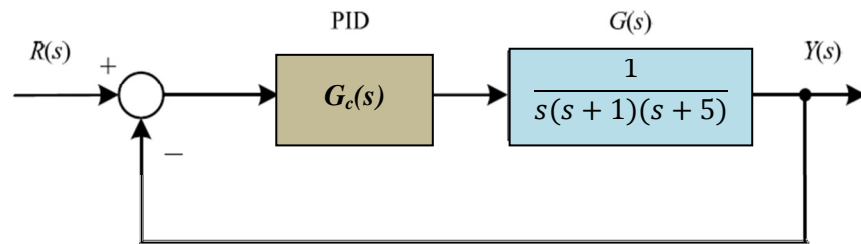


Figura 11 Diagrama de bloco do sistema de controlo [Ogata, 2003]

Este sistema representa uma mais-valia para analisar a configuração inicial do AG, visto que o mesmo já foi objecto de estudo na literatura [Ogata, 2003], em que o autor utilizou a configuração de Ziegler-Nichols como ponto de partida para uma posterior sintonia fina do sistema.

A metodologia deste trabalho assemelha-se à utilizada por Ogata [Ogata, 2003], em que o desempenho do sistema obtido com o método de Ziegler-Nichols será utilizado como ponto de partida para a sintonia do sistema com AG.

4.1.1. O CONTROLADOR PID UTILIZADO

A estrutura típica de um controlador PID no domínio dos tempos é dada pela seguinte expressão:

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right) \quad (7)$$

onde $u(t)$ representa o sinal de controlo aplicado na entrada do processo e $e(t)$ é o sinal de erro ao qual é aplicado as acções do controlador de forma a minimizá-lo. O ganho proporcional, a constante de tempo integral e a constante de tempo derivativa é dada respectivamente por K , T_i e T_d .

A função de transferência de $G_c(s)$ é representada por:

$$G_c(s) = \frac{U(s)}{E(s)} = K \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (8)$$

Esta equação do controlador PID nem sempre é a mais indicada para o controlo de processos industriais. Em aplicações práticas, utiliza-se a acção derivativa acoplada a um filtro de primeira ordem para minimizar os efeitos causados pela presença de ruídos e variação brusca da entrada de referência, desta forma a equação prática do controlador PID é dada pela seguinte expressão:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d s}{N}} \right) \quad (9)$$

Pela análise da equação (9), verifica-se que a estrutura do bloco PID possui a parte derivativa associada a um filtro de primeira ordem, com constante de tempo equivalente a $\frac{T_d}{N}$. De acordo com Z-A [Zhuang e Atherton, 1993] quando é utilizada a constante do filtro N com um valor maior ou igual a 10 ($N \geq 10$), os resultados obtidos por meio das equações (8) e (9), são muito semelhantes. Desta forma a estrutura do controlador utilizada para a realização deste trabalho é dada pela equação (9) com o valor da constante do filtro $N = 100$. Este é o valor definido por omissão no bloco PID do ambiente Simulink, e seguindo o conselho de Z-A foi mantido o valor de $N = 100$ no intuito de obter resultados semelhantes, quer usando a expressão (8) ou a expressão (9).

4.2. SINTONIA DE ZIEGLER-NICHOLS PARA DETERMINAR OS GANHOS DO CONTROLADOR

Para o sistema apresentado na Figura 11, o controlador PID possui a função de transferência, dada pela equação (9):

Como o processo a controlar possui um integrador, será utilizado o segundo método das regras de sintonia de Ziegler-Nichols em malha fechada (ver Tabela 2).

Ajustando-se $T_i = \infty$ e $T_d = 0$, obtém-se a seguinte função de transferência em malha fechada:

$$\frac{C(s)}{R(s)} = \frac{K_p}{s(s+1)(s+5) + K_p} \quad (10)$$

O valor de K_p que torna o sistema marginalmente estável, ou seja, de modo a ocorrerem oscilações constantes na saída do sistema, pode ser obtido através do critério de estabilidade de Routh Hurwitz [Ogata, 2003].

Aplicando-se o método, obtém-se o valor do ganho crítico $K_{cr} = 30$ e o período de oscilação $P_{cr} = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{5}} = 2,8099$ s.

Com base na Tabela 5, determinam-se os valores de K_p , T_i e T_d apresentados na Tabela 6.

Tabela 5 Sintonia PID pelo método de Ziegler-Nichols em malha fechada

Controlador	K_p	T_i	T_d
PID	$0.6(K_{cr})$	$\frac{P_{cr}}{2}$	$\frac{P_{cr}}{8}$

Tabela 6 Parâmetros do controlador PID

Controlador	K_p	T_i	T_d
PID	18	1,405	0,3512

Usando as relações: $K_i = \frac{K_p}{T_i}$ e $K_d = K_p T_d$ chega-se aos valores dos ganhos do PID da Tabela 7.

Tabela 7 Ganho do controlador PID

Controlador	K_p	K_i	K_d
PID	18	12,81	6,3216

A resposta do sistema, a uma entrada em degrau unitário, encontra-se representada na Figura 12.

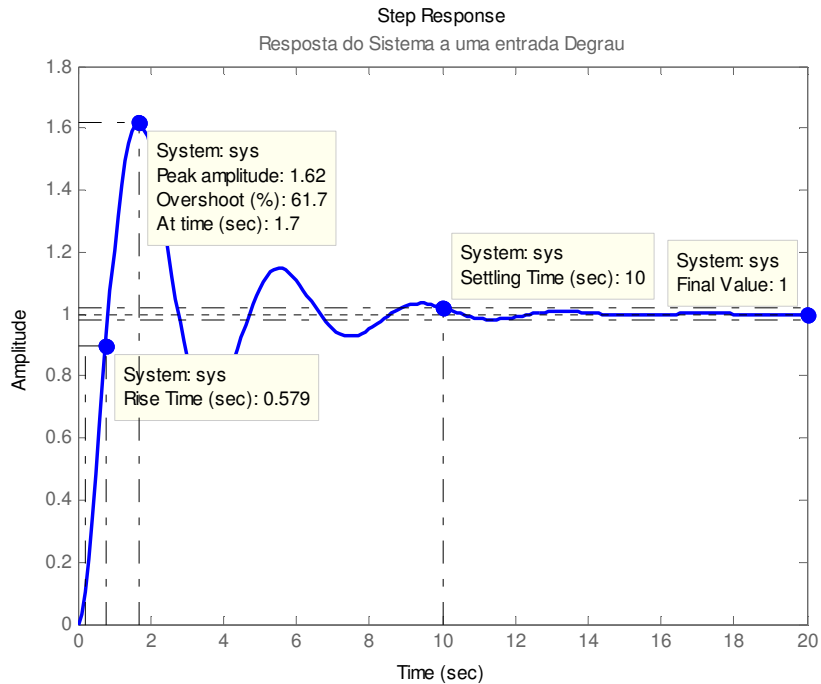


Figura 12 Resposta do sistema a uma entrada em degrau unitário

4.3. ANÁLISE DA RESPOSTA DO SISTEMA

A análise do diagrama apresentado na Figura 12 será feita pelos seguintes parâmetros:

- Tempo de subida, t_r
- Sobreelongação Máxima, $M_p\%$
- Tempo de estabelecimento, t_s

O tempo de subida t_r , é o tempo que o sistema demora a ir de 10% a 90% do valor final.

No caso da Figura 12, $t_r \cong 0.579s$.

A sobreelongação máxima do sistema $M_p\%$, é a diferença entre o valor máximo e o valor no qual o sistema se estabiliza. No caso da Figura 12, $M_p \cong 61,7\%$.

Finalmente o tempo de estabelecimento t_s , é o tempo que o sistema demora até se manter em $\pm 2\%$ do valor final. No caso da Figura 12, $t_s \cong 10s$.

A partir desta análise, pode-se dizer que o sistema apresenta uma sintonia distante daquela que seria aceitável para o controlo de um processo, onde sobreelongação muito elevada na

sua saída não é normalmente desejado. Em várias situações industriais, os parâmetros apresentados por Ziegler-Nichols servem como ponto de partida para melhorar o desempenho do sistema.

Partindo deste pressuposto, as experiências que se seguem utilizam os valores dos parâmetros K_p , K_i e K_d , obtidos através do método de Ziegler-Nichols, como ponto de partida para obtenção de uma melhor sintonia do controlador. Para isso, foi utilizado o AG e algumas rotinas de optimização do MatLab, assim como outros métodos de optimização existentes na literatura.

4.4. FERRAMENTA DE DESENVOLVIMENTO – MATLAB – PARA O AJUSTE DO CONTROLADOR PID

Como ferramenta de desenvolvimento, para este trabalho, utilizou-se o MATLAB (Versão 2009b) e as funções que este disponibiliza para a implementação de Algoritmos Genéticos.

Existem inúmeras aplicações especializadas que proporcionam um ambiente com linguagem específica para configuração do AG. No MatLab, a implementação do AG é feita através da escolha dos parâmetros genéticos, e caso seja necessário, é possível estender as capacidades do AG ao escrever o próprio código num ficheiro *M-file*, e conjugá-lo com outras *toolboxes* de optimização. Estas facilidades de implementação, juntamente com outros recursos que o MatLab possui, como as funções de optimização que servirão de comparação para o modelo apresentado, foram os factores que levaram à escolha desta ferramenta.

Para simular o sistema, o MatLab apresenta um grande vantagem, pois ao utilizar o Simulink, ferramenta integrada no MatLab que permite a resolução de problemas de análise e simulação de sistemas de controlo, foi possível desenvolver o diagrama de blocos representativo do sistema em estudo e conjugá-lo na função objectivo utilizada pelo AG. Esta representação, em diagrama de blocos, pode ser visualizada na Figura 13.

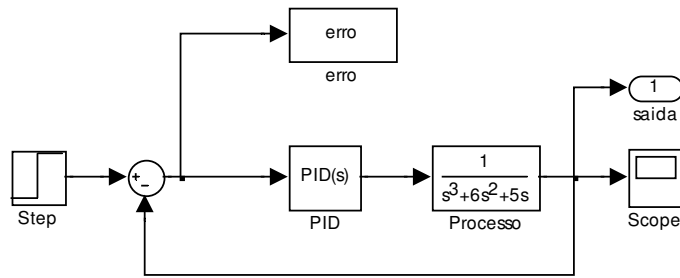


Figura 13 Diagrama de blocos para o sistema de controlo implementado em Simulink

Na função de avaliação do AG é executado o comando “sim” que simula um ficheiro do Simulink que contém o diagrama de blocos do sistema considerado. Por cada chamada desta função, é feita uma simulação com os parâmetros do controlador PID definidos na altura. Esta função recebe como parâmetros o nome do ficheiro Simulink e o tempo de simulação, e retorna o vector com os valores do tempo, do erro e da saída do sistema. No final da função objectivo é retornado o valor final do erro, de modo a ser minimizado pelo AG.

Figura 14 apresenta um esquema da metodologia que foi adoptada.

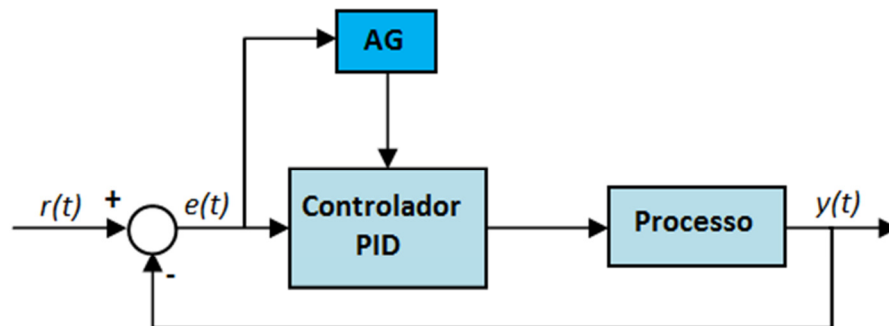


Figura 14 Ajuste do controlador PID através do AG

4.5. ESTRUTURA DO AG DESENVOLVIDO

O algoritmo distribui-se por três ficheiros: programa principal (“*ga_pid*”), função objectivo (FO) (“*objective_function_ID*”), em que *ID* corresponde ao índice de desempenho a ser minimizado e função de visualização da resposta do sistema (“*plot_function*”).

4.5.1. PROGRAMA PRINCIPAL

Este é o ficheiro principal do algoritmo. É a partir deste que são chamadas todas as outras funções adjacentes ao programa. O código principal correspondente a este programa é apresentado a seguir.

```
clear all
close all
global Kp Ki Kd Y T
%ObjectiveFunction = @objective_function_IAE;
%ObjectiveFunction = @objective_function_ITAE;
%ObjectiveFunction = @objective_function_ISE;
%ObjectiveFunction = @objective_function_ITSE;
%ObjectiveFunction = @objective_function_MSE;
nvars = 3;
```

O programa começa com a definição da função objetivo, que neste caso, visa minimizar o erro entre o sinal de entrada e o sinal de saída do sistema, e logo de seguida a indicação do número de variáveis que vão ser encontradas pelo algoritmo.

É guardada na variável “*options*” as definições que se quer para o AG, em substituição das opções já definidas por omissão no Matlab. O código apresentado abaixo corresponde aos parâmetros de iniciação do AG como tamanho da população, número de gerações, entre outros. Para saber quais os valores definidos por omissão para a *toolbox* de optimização dos AGs, basta consultar a função *gaoptimset* via o comando *help ga* no ambiente de trabalho do MatLab.

```
options =
    PopInitRange: [2x1 double]
    PopulationSize: 20
    EliteCount: 2
    Generations: 100
    InitialPopulation: [18 12.4100 6.3216]
    CreationFcn: @gacreationlinearfeasible
    SelectionFcn: @selectiontournament
    CrossoverFcn: @crossoversinglepoint
    PlotFcns: {[1x1 function_handle] [1x1
function_handle]}
```

4.5.2. FUNÇÃO OBJECTIVO E ÍNDICES DE DESEMPENHO

É neste ficheiro que se encontra a função a minimizar chamada pelo AG. Várias FO foram escritas com base no critério de desempenho do erro. Em cada FO apenas a secção do código que define o critério de desempenho de erro pretendido para a ser implementado, é alterada.

Para otimizar o desempenho de um sistema de controlo PID, os ganhos do controlador são ajustados para minimizar um determinado índice de desempenho. O índice de desempenho é calculado durante um intervalo de tempo, t , normalmente na região de $0 \leq t \leq t_s$, onde t_s é o tempo de estabelecimento do sistema. Os índices de desempenho (ID) utilizados foram o IAE, ITAE, ISE, ITSE e MSE, a seguir descritos.

- **Integral do Erro Absoluto (IAE)**

Este índice é equivalente à soma das áreas acima e abaixo do valor de referência. O critério IAE considera o módulo do erro, sendo muito utilizado em simulações de controlo de processos. Matematicamente o IAE é definido como:

$$I_{IAE} = \int_0^{\infty} |e(t)| dt \quad (11)$$

O algoritmo de avaliação é implementado, conforme apresentado em seguida.

```
%Integral do Erro Absoluto (IAE)
time=0:0.1:20;
[T X Y]=sim('dp_pid',time);
error = abs(erro);

%IAE (integral of absolute error)
IAE=sum(error);
```

- **Integral do Erro Absoluto Ponderado pelo Tempo (ITAE)**

Basicamente este índice é a extensão do índice de desempenho anterior, onde se acrescentou o vector dos tempos. O índice ITAE penaliza erros que se mantêm ao longo do tempo.

Matematicamente o ITAE é definido por:

$$I_{ITAE} = \int_0^{\infty} t|e(t)|dt \quad (12)$$

O algoritmo de avaliação do índice de desempenho para o ITAE é implementado, conforme apresentado a seguir.

```

% Integral do Erro Absoluto ponderada pelo Tempo
(ITAE)

time=0:0.1:20;
[T X Y]=sim('dp_pid',time);
error=abs(error).*T;

% ITAE (integral of time-weighted absolute error)
ITAE=sum(error);

```

- **Integral do Erro Quadrático (ISE)**

Este ID apresenta pouca selectividade, sendo que as variações nos parâmetros do processo correspondem a pequenas variações no índice ISE. Sistemas que são configurados, tendo como critério a minimização deste ID tendem a apresentar uma rápida resposta, mas com baixa estabilidade, apresentando oscilações.

Matematicamente o ISE é definido da seguinte forma:

$$I_{ISE} = \int_0^{\infty} [e(t)]^2 dt \quad (13)$$

O algoritmo de avaliação é implementado, conforme apresentado a seguir.

```

%Integral do Erro Quadrático (ISE)

time=0:0.1:20;
[T X Y]=sim('dp_pid',time);
error = error;
error = error.^2;

% ISE (integral of squared error)
ISE = sum(error);

```

- **Integral do Erro Quadrático Ponderado pelo Tempo (ITSE)**

O ITSE é um índice de desempenho que pondera os erros ao quadrado, penalizando os erros que persistem ao longo do tempo, permitindo assim avaliar a influência que o tempo tem sobre o valor do erro produzido pelo sistema.

Matematicamente o ITSE é assim definido por:

$$I_{ITSE} = \int_0^{\infty} t[e(t)]^2 dt \quad (14)$$

O código referente a esta implementação em MatLab é apresentado a seguir.

```

% Integral do Erro Quadrático ponderado pelo
Tempo (ITSE)

time=0:0.1:20;
[T X Y]=sim('pid_simulink',time);

error = erro;
error =(error.^2).*T;

% ITSE (integral of time-weighted squared error)
ITSE = (sum(error));

```

- **Média do Erro Quadrático (MSE)**

O critério MSE reflecte todas as variações e desvios em relação à referência. Este critério tem apresentado as melhores respostas quando utilizado como função de avaliação pura ou híbrida para meta-heurística, tais como Algoritmos Genéticos e Colónia de Formigas [GRIFFIN, 2003].

Matematicamente o MSE é definido como:

$$I_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N e^2(i) \quad (15)$$

em que N se refere ao número de amostras a ser utilizado pelo PID.

O algoritmo de avaliação é implementado conforme apresentado a seguir.

```

% Média do Erro Quadrático (MSE)

time=0:0.1:20;
N=length(time);
[T X Y]=sim('dp_pid',time);
error = erro;
error = error.^2;

%MSE (Mean of the Square of the Error)
MSE=sum(error)/N;

```

4.6. AVALIAÇÃO DOS PARÂMETROS DO AG

Anteriormente foi apresentado o código de inicialização do AG. Nesta secção pretende-se descrever a metodologia utilizada para a configuração deste código com base nas características do sistema apresentado.

4.6.1. INICIALIZAÇÃO DA POPULAÇÃO DO AG

Para iniciar a população do AG é necessário configurar um conjunto de parâmetros que irão quantificar e qualificar a classe dos indivíduos que vão representar o conjunto de soluções para o problema.

Para gerar a população do AG é necessário definir o seu tamanho, os seus limites e a função de geração. De seguida apresenta-se o código em MatLab para definição de tais parâmetros.

```
% Opções para Inicialização da População do AG

PopulationSize = 20;
PopInitRange = [0; 300];
lb = [0; 0; 0]
ub = [200; 100; 300]
CreationFcn = @gacreationlinearfeasible;
```

A descrição do código apresentado é feita a seguir.

- **PopulationSize** – O primeiro passo para gerar a população do AG é definir qual o tamanho da população. Através deste comando é definido quantos indivíduos estarão presentes em cada geração.
- **PopIniRange** – Tem por finalidade especificar os limites inferiores e superiores para o vector que corresponde à geração da população inicial. No caso particular deste exemplo, como se trata de otimizar os ganhos de um controlador PID, é possível ainda criar uma matriz para limitar o vector de cada parâmetro do controlador definida como *Lower bound* (lb) e *Upper bound* (ub).
- **CreationFcn** – Para o presente trabalho utilizou-se a função de criação (gacreationlinearfeasible). Esta é a função destinada a problemas com restrições a serem tratadas, de modo a criar uma população inicial aleatória que tende a ser bem dispersa por todo o espaço de soluções, satisfazendo todos os limites e restrições presentes no problema.

Existem outras opções para gerar a população inicial do AG que não foram utilizadas neste trabalho. Para mais informações consultar [Guide MatLab, 2009].

4.6.2. CONFIGURAÇÃO DOS RESTANTES PARÂMETROS DO AG

Para todas as simulações que serão realizadas ao longo deste trabalho, os parâmetros aqui citados permaneceram fixos na configuração do AG. As mudanças que se julgarem necessárias para o melhor desempenho do sistema a ser estudado ocorrerão nas configurações da população inicial do AG, descritas anteriormente. Repete-se o código em MatLab para as configurações descritas nesta subsecção.

```
options =  
    EliteCount: 2  
    Generations: 100  
    StallGenLimit: "Nº de Gerações"  
    InitialPopulation: [18 12.4100 6.3216]  
    CreationFcn: @gacreationlinearfeasible  
    SelectionFcn: @selectiontournament  
    CrossoverFcn: @crossoversinglepoint  
    PlotFcns: {[1x1 function_handle] [1x1  
function_handle]}
```

- ***Elitecount*** – Especifica o número de melhores indivíduos que passam para a geração seguinte, sem sofrerem alterações. No âmbito deste trabalho, a regra utilizada foi a de seleccionar os melhores 10% de indivíduos de uma geração para passar para a geração seguinte. Desta forma garante-se que as melhores soluções de uma geração não se perdem.
- ***Generations*** – Especifica o número máximo de iterações feitas pelo AG ao longo de cada simulação.
- ***StallGenLimit*** – Este código é constituído por um número positivo e inteiro que especifica o critério de paragem do AG caso não ocorra nenhuma melhoria na função objectivo durante o número de iterações estipulado por este código.
- ***InitialPopulation*** – Com este código é possível criar a população do AG, de forma global ou parcial. No caso específico deste trabalho é inserido apenas um único indivíduo na população inicial, que corresponde aos valores dos ganhos do controlador PID obtidos através do método de Ziegler-Nichols, sendo os demais indivíduos criados aleatoriamente.
- ***CrossoverFcn e SelectionFcn***- A partir dos trabalhos realizados nesta área, e dos melhores resultados obtidos, adoptou-se como função de cruzamento um único ponto e como função de selecção o torneio.

4.6.3. EXECUÇÃO DO AG

O AG é executado através do código a seguir apresentado:

```
% Desempenho do AG
[X, FVAL, EXITFLAG, OUTPUT, POPULATION, SCORES] =
ga(ObjectiveFunction, nvars, ...
[], [], [], [], lb, ub, [], options)
```

Como resposta o AG irá retornar cinco variáveis:

- X – Os melhores valores encontrados para os ganhos K_p , K_i e K_d , que geram o menor valor para a função objectivo.
- FVAL – O valor da função objectivo, neste caso, o menor valor para o ID escolhido.
- EXITFLAG – Descreve a condição de paragem do AG. A resposta é dada por uma numeração limitada entre -5 a 5, cujo significado pode ser consultado em [Guide MatLab, 2009].
- OUTPUT – Retorna a estrutura de saída com informações referentes ao número de iterações efectuadas, critério de paragem, total de funções que foram avaliadas e uma mensagem final sobre a convergência do AG.
- POPULATION – Retorna todos os indivíduos que correspondem à população final do AG.
- SCORES – Retorna o valor que cada indivíduo da população final representa na função objectivo.

4.7. AVALIAÇÃO DOS ÍNDICES DE DESEMPENHO

Várias simulações foram realizadas para avaliar qual dos cinco critérios de desempenho apresentados produzia melhores resultados quando implementados no AG. Para cada ID, foi criada uma função objectivo descrita como: *objective_function_IAE.m*, *objective_function_ITAE.m*, *objective_function_ISE.m*, *objective_function_ITSE.m* e *objective_function_MSE.m*.

Segundo dados do MatLab [Guide MatLab, 2009], para se ter melhores resultados com os AGs, é necessário testar diferentes opções de configuração até se chegar a um resultado

satisfatório. E um dos mais importantes factores a serem avaliados é a diversidade populacional.

Se, a distância média entre os indivíduos for alta, gera-se uma maior diversidade populacional. Caso contrário, se a distância média entre os indivíduos for baixa, gera uma menor diversidade populacional. Assim sendo, ao especificar os limites superiores e inferiores para o vector que corresponde à criação da população é necessário especificar o número de indivíduos a serem utilizados para “cobrir” da melhor maneira a área compreendida por estes limites. Segundo o mesmo documento [*Guide MatLab*, 2009] uma diversidade muito alta, ou muito baixa pode fazer com que o AG não tenha um bom desempenho.

Para a escolha da dimensão da população, os testes iniciais foram baseados em trabalhos de outros autores e na opinião de Reeves [Reeves, 2009]. Segundo este autor, um dos parâmetros que mais influência o desempenho do AG é o tamanho da população. Tendo como referência outros trabalhos já efectuados [Goldberg, 1989; Davis, 1989; Mitchell, 1996; Reeves 1993], a faixa de valores recomendados para o tamanho da população é de 30 a 200 indivíduos. No MatLab o valor definido por omissão para o tamanho da população é de 20 indivíduos.

No caso específico deste trabalho, foram utilizados valores intermédios aos sugeridos pelos autores mencionados anteriormente. Como se trata de cinco diferentes FOs para o mesmo sistema, não se pode afirmar que um determinado número de indivíduos seja suficiente para satisfazer da mesma maneira todas as funções que foram criadas. Desta forma, a metodologia que se segue para determinar o número de indivíduos para cada ID, baseia-se em fazer a população variar dentro de um intervalo devidamente especificado, na esperança de que a partir de um determinado ponto, ou um determinado número de indivíduos da população, o resultado da FO permaneça fixo.

Segundo Goldberg [Goldberg, 1989], apesar de utilizarmos valores bem definidos para os parâmetros dos AGs, como tamanho da população, taxa de cruzamento e mutação, pode ocorrer o problema de convergência prematura, em que a presença de bons indivíduos é predominante, porém estes não são óptimos, o que pode fazer a solução ficar limitada a óptimos locais.

O código desenvolvido para este fim, permite que o AG avalie a FO diversas vezes, variando o número de indivíduos para cada simulação. Para este sistema fez-se a população variar de um limite inferior de 20 indivíduos a um limite superior de 200 indivíduos com incrementos de 20. No final, além de se visualizar os ganhos do PID e o valor da FO para cada simulação é possível criar um gráfico que apresentada o desempenho do AG ao longo de todo o processo. Este gráfico é apresentado na Figura 15, para o sistema de controlo da Figura 11.

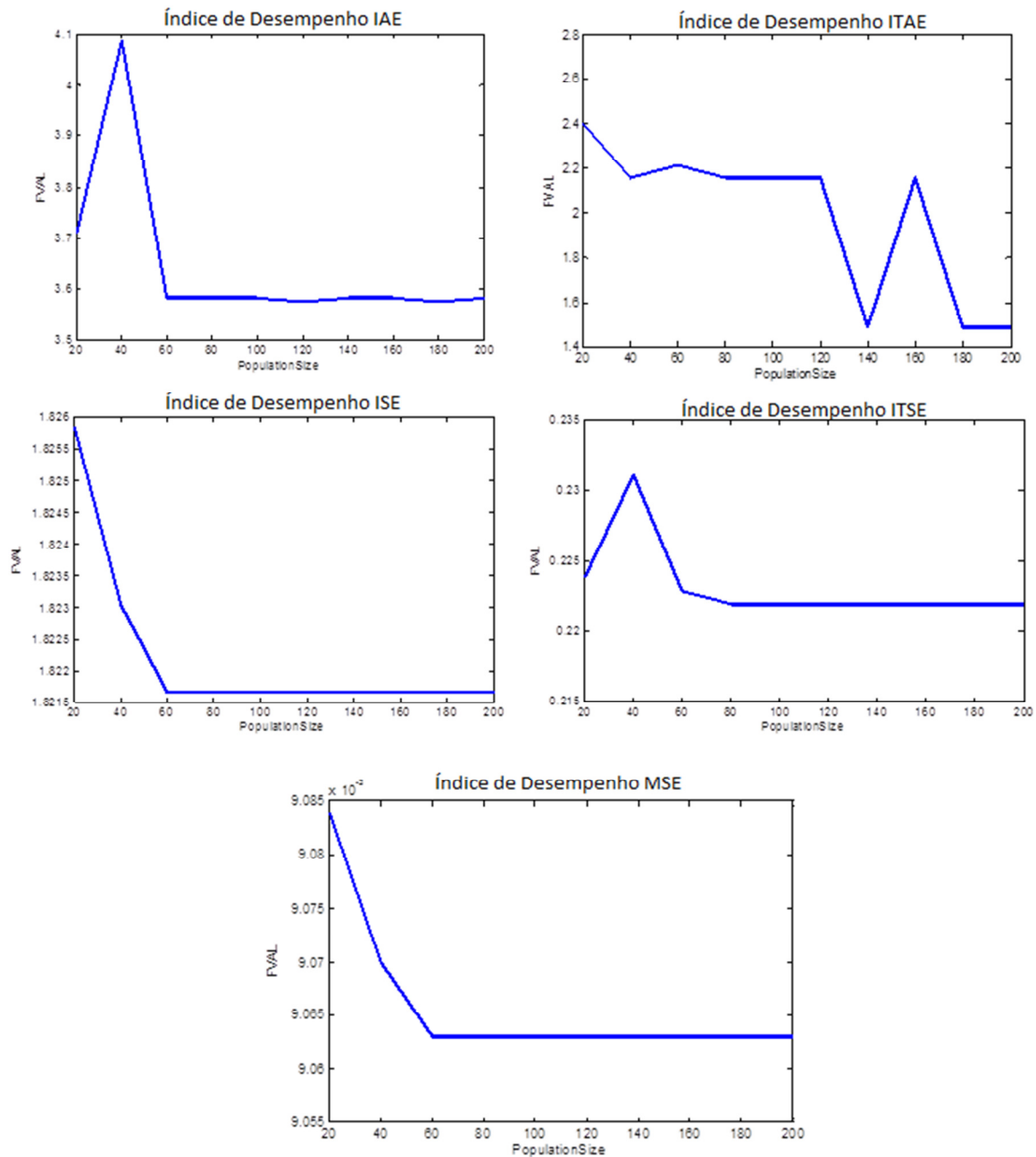


Figura 15 Desempenho do AG em função do aumento do tamanho da população

O valor do erro para cada ID apresentado na Figura 15 pode ser analisado na Tabela 8.

Tabela 8 Desempenho do AG em função do aumento do tamanho da população

	Valor do Erro				
	IAE	ITAE	ISE	ITSE	MSE
200	3,5825	1,4884	1,8217	0,2220	0,00906
180	3,5761	1,4884	1,8217	0,2220	0,00906
160	3,5825	2,1574	1,8217	0,2220	0,00906
140	3,5826	1,4883	1,8217	0,2220	0,00906
120	3,5760	2,1575	1,8217	0,2220	0,00906
100	3,5830	2,1575	1,8217	0,2220	0,00906
80	3,5825	2,1576	1,8217	0,2220	0,00906
60	3,5828	2,2139	1,8217	0,2229	0,00906
40	4,0865	2,1575	1,8231	0,2311	0,00908
20	3,7096	2,4021	1,8259	0,2239	0,00909

Pelos resultados apresentados na Tabela 8, pode-se concluir que para todos os índices de desempenho, excepto os índices IAE e ITAE, o valor da FO começa a estabilizar a partir de um determinado tamanho populacional, Assim os melhores resultados para os índices {IAE; ITAE; ISE; ITSE; MSE}, ocorreram para um tamanho populacional de {120; 140; 60; 80; 60}, respectivamente. Continuando a análise da Tabela 8, observamos que para o ID MSE, independentemente do número de indivíduos utilizados, o valor de erro permanece praticamente o mesmo ao longo de todo o processo.

Com vista a satisfazer o melhor resultado para todos os índices descritos, será então utilizado um valor de 140 indivíduos, visto que este valor engloba o melhor resultado de todos os IDs e ainda está compreendido entre os valores de referência dos autores citados anteriormente.

A Tabela 9 apresenta os parâmetros de inicialização da população do AG, com ênfase nos melhores resultados obtidos para cada ID.

Tabela 9 Valores de inicialização da população do AG

Parâmetros de inicialização do AG	
Tamanho da População	140
Número de Gerações	100
População Inicial	[18; 12,81; 6,3216]
Limite da População Inicial	[0; 300]
Limite dos valores de $[K_p; K_i; K_d]$	lb = [0; 0; 0]
	ub = [200; 100; 300]

Na Tabela10, registou-se os valores obtidos para cada parâmetro, em função do ID atribuído, de forma a possibilitar uma visão global do desempenho do sistema.

Tabela 10 Parâmetros e especificações das respostas do sistema

	Ganhos do Controlador			Desempenho da Malha				
	K_p	K_i	K_d	t_r	t_s	MP%	Erro	
Ziegler-Nichols	18	12,81	6,3216	0,579	10	61,7		
AG	IAE	55,9427	0,0296	102,7247	0,123	1,96	47,0	3,5825
	ITAE	80,0872	0,0047	108,2577	0,117	1,92	51,2	1,4883
	ISE	18,2520	0,6818	122,5410	0,112	3,04	46,6	1,8217
	ITSE	29,9945	0,1190	58,8304	0,181	1,89	30,8	0,2220
	MSE	16,5390	1,0525	123,2786	0,112	3,08	46,6	0,00906

Para os ganhos encontrados por Ziegler-Nichols, o erro do sistema diferencia-se para cada ID, logo para $\{K_p; K_i; K_d\} = \{18; 12,81; 6,3216\}$ resulta num erro para $\{IAE; ITAE; ISE; ITSE; MSE\} = \{20,0224; 52,4392; 9,0674; 11,8543; 0,0456\}$. Sendo assim, para o sistema em estudo, todos os índices de desempenho utilizados para sintonizar o controlador com o AG apresentaram melhores resultados quando comparados com o critério de Ziegler-Nichols, como aliás seria de esperar.

A Figura 16 ilustra a resposta do sistema a uma entrada em degrau obtida pelo método de Ziegler-Nichols *versus* a resposta do sistema para a mesma entrada obtida para cada ID.

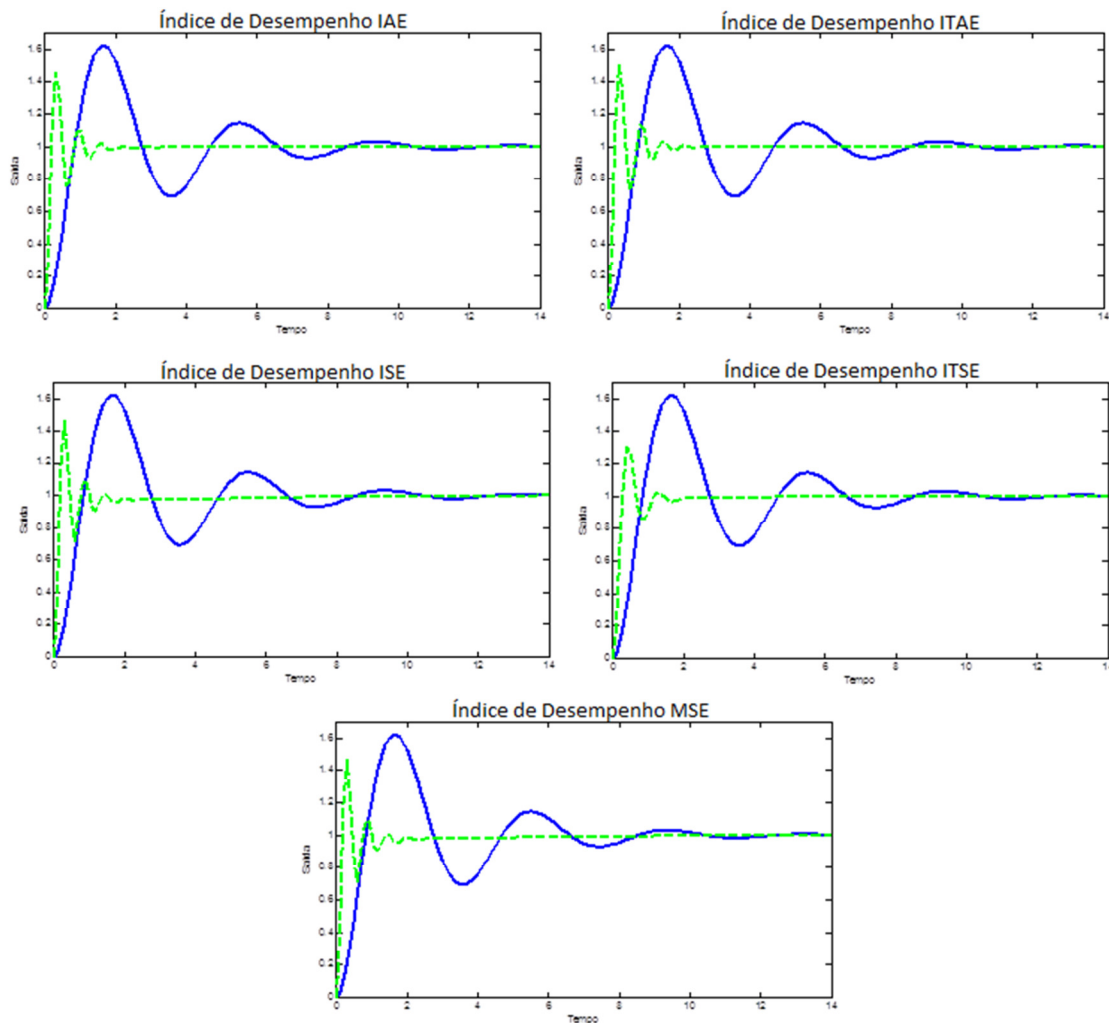


Figura 16 Desempenho do controlador PID por Ziegler-Nichols (linha contínua), *versus* Desempenho do controlador PID para cada índice de desempenho com AG (linha tracejada)

4.7.1. POPULAÇÃO *VERSUS* GERAÇÃO

O princípio apresentado na secção anterior consistia em variar o tamanho da população de forma a obter o melhor desempenho para o sistema apresentado. Seguindo esta mesma linha de raciocínio, esta subsecção pretende avaliar a influência que o número de gerações tem sobre o desempenho do sistema.

Para realização desta simulação, escolheu-se o índice de desempenho ITAE, visto que de entre todos os IDs estudados para o presente sistema, o AG teve maior dificuldade em encontrar a melhor solução, e ainda assim o melhor resultado encontrado evidência um elevado valor de sobreelongação, perante os demais.

Desta forma a Figura 17 apresenta os resultados obtidos para seis simulações, cuja metodologia consistia em fixar o número de gerações em um intervalo de 100 a 200, com incremento de 20 gerações. Espera-se com isto identificar um ponto que represente um melhor desempenho, para o sistema apresentado nesta secção, nomeadamente um par de valores (população; geração) que estabeleça um bom compromisso de funcionamento para o AG.

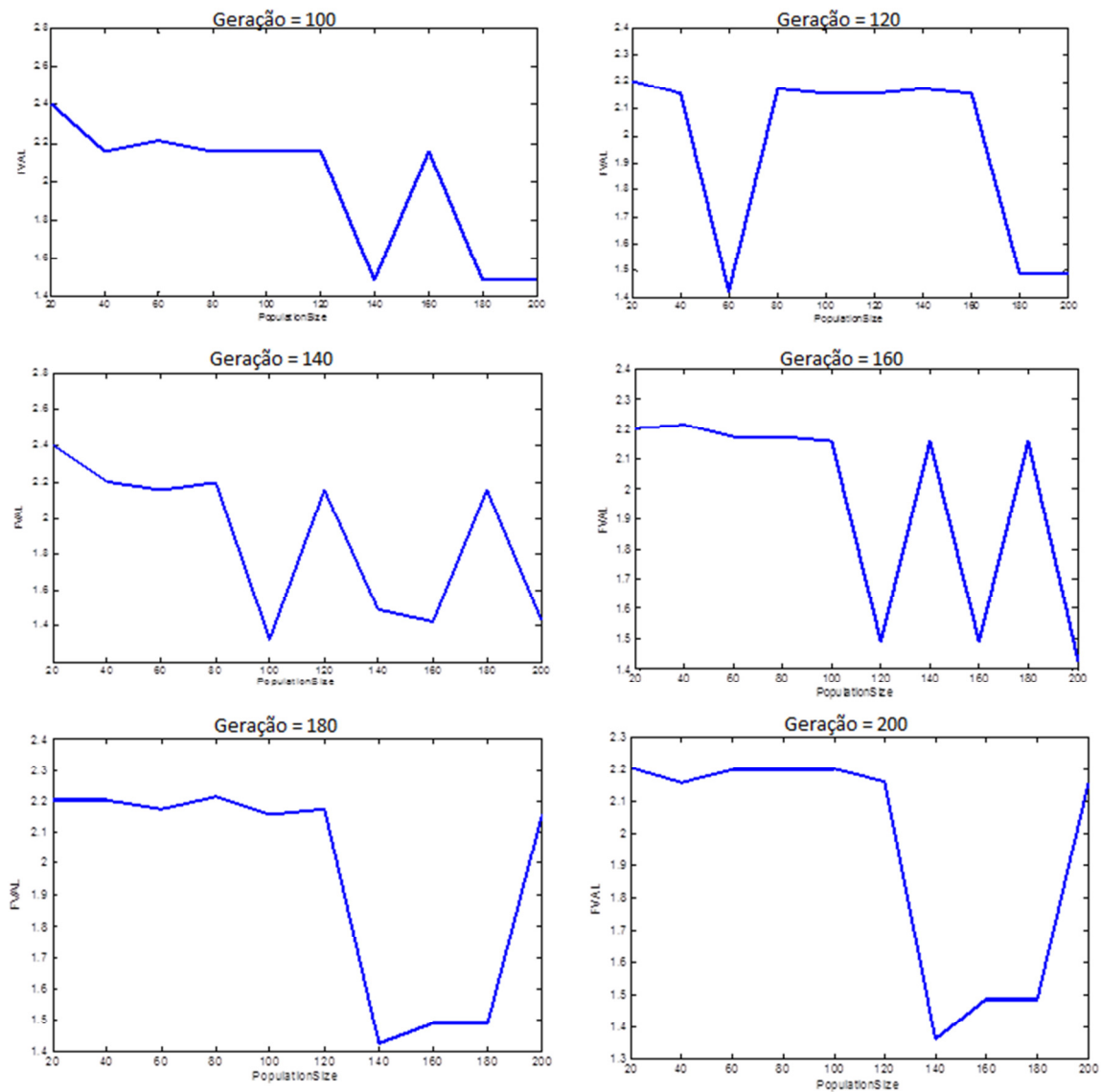


Figura 17 População versus Geração para o índice de desempenho ITAE

A Tabela 11 apresenta os resultados obtidos para as simulações que foram realizadas.

Tabela 11 Valor do erro para o índice ITAE

Valor do Erro para o ID ITAE

Tamanho da População	200	1,4884	1,4884	1,4271	1,4250	2,1572	2,1571
	180	1,4884	1,4882	2,1572	2,1574	1,4887	1,4883
	160	2,1574	2,1572	1,4251	1,4883	1,4882	1,4882
	140	1,4883	2,1748	1,4883	2,1577	1,4248	1,3620
	120	2,1575	2,1574	2,1574	1,4883	2,1748	2,1594
	100	2,1575	2,1575	1,2622	2,1574	2,1575	2,2020
	80	2,1576	2,1748	2,1996	2,1748	2,2136	2,2020
	60	2,2139	1,4253	2,1594	2,1747	2,1749	2,1987
	40	2,1575	2,1574	2,2028	2,2138	2,2027	2,1574
	20	2,4021	2,2034	2,4021	2,2034	2,2034	2,2034
		100	120	140	160	180	200
		Número de Gerações					

Reunindo todos os dados da Tabela 11 é possível fazer uma representação em 3D usando a função *surf* (combinação *surf-contour*) do MatLab. Pela análise da Figura 18 conclui-se que para o conjunto de simulações realizadas o ponto óptimo da FO dada pelo índice ITAE corresponde a um número de gerações igual a 140 e o tamanho da população igual a 100, revelando um valor de FO igual a 1,262.

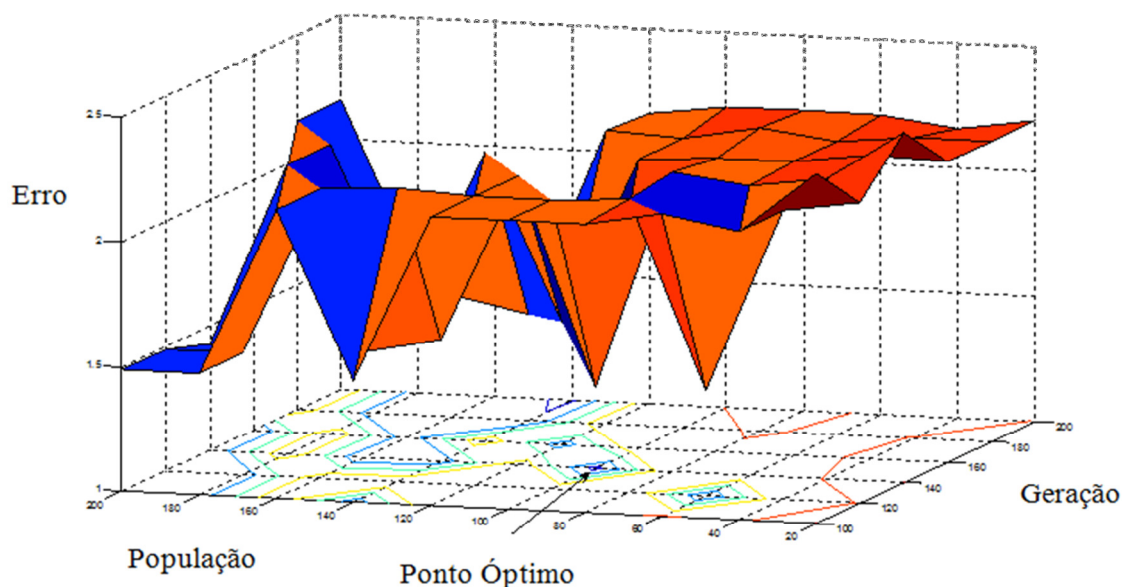


Figura 18 Ponto óptimo da FO dado pelo índice ITAE

A Figura 19 apresenta o desempenho do sistema a uma entrada em degrau unitário, dado pelos ganhos do controlador $\{K_p; K_i; K_d\} = \{41,8658; 0,0002; 45,7076\}$ referentes ao ponto óptimo da Figura 18.

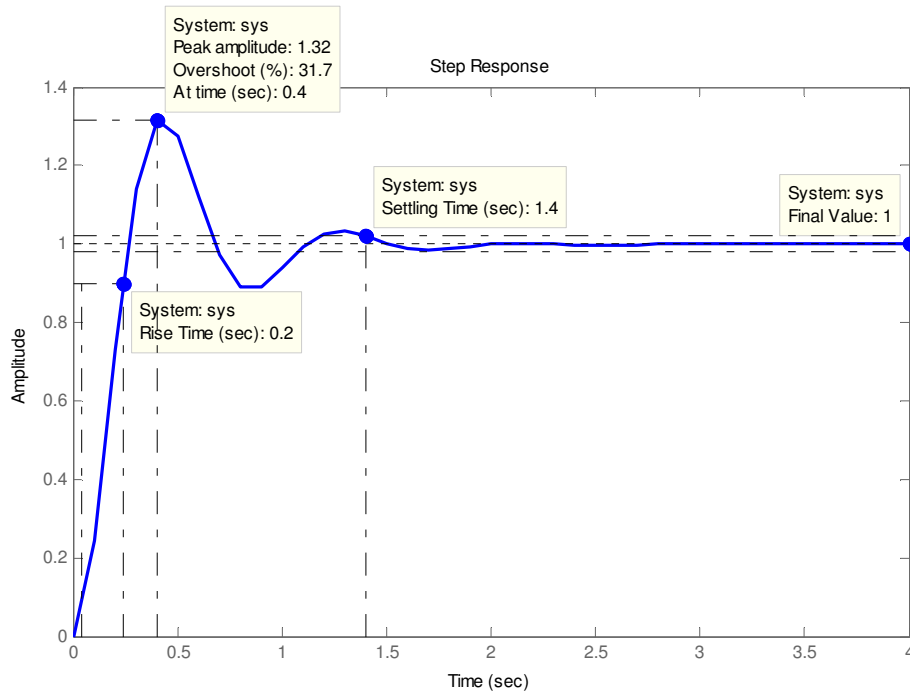


Figura 19 Resposta do sistema para o ponto óptimo dado pelo índice ITAE

O resultado encontrado evidêcia um melhor desempenho para o sistema em termos de sobreelongação, tempo de estabelecimento e tempo de subida. Esta particularidade ocorreu apenas com o índice ITAE, visto que a mesma simulação foi usada para os outros índices mas o melhor resultado continuava a ser o apresentado na Tabela 10.

Para concluir o que foi apresentado nesta subsecção, pode-se dizer que apesar das inúmeras vantagens que existem em utilizar os AGs como método de pesquisa para os mais diversos tipos de aplicações, é extremamente difícil determinar correctamente qual a melhor configuração para o AG.

Não existem métodos empíricos capazes de determinar com exactidão quais os melhores valores para o tamanho da população, o número de gerações, a taxa de elitismo e assim por diante. Segundo o manual do MatLab [*Guide MatLab*, 2009], quando não se conhece correctamente o espaço de soluções do problema a ser tratado, estes parâmetros são determinados por meio de tentativa e erro.

Na subsecção que se segue, será apresentado as funções de optimização disponibilizadas pelo MatLab de forma a comparar o desempenho do sistema em estudo, quando sintonizado através dos AGs e quando sintonizados através de tais funções de optimização.

4.8. SINTONIA COM AS FUNÇÕES DE OPTIMIZAÇÃO DO MATLAB – *FMINCON* E *PATTERNSEARCH*

As funções que serão abordadas nesta secção fazem parte da *toolbox* de Optimização do MatLab. Nesta ferramenta existe um conjunto de funções pré-programadas, entre as quais as funções de optimização contidas numa livraria (*toolbox*) apropriada. Na *toolbox* de optimização existem cerca de 20 rotinas, cada uma dedicada a resolver problemas específicos. No caso deste trabalho, as rotinas que se aplicam ao problema são denominadas *fmincon* e *patternsearch*, ambas específicas para problemas que possuem restrições a serem tratadas.

No início da elaboração desta dissertação, outras funções de optimização presentes no MatLab foram testadas, nomeadamente as funções *fminsearch* e *fminunc*. Entretanto, à medida que o estudo se desenvolvia, a necessidade de estipular os limites inferiores e superiores para os ganhos K_p , K_i e K_d do controlador era de extrema importância para o bom desempenho do mesmo. Como as funções *fminsearch* e *fminunc* não permitem a utilização de restrições, e conseqüentemente não permitem a introdução dos limites inferiores e superiores para os ganhos do PID, as mesmas deram lugar a funções que eram mais adequadas à resolução deste problema, as funções *fmincon* e *patternsearch*.

Um outro factor que deve ser mencionado e que levou à utilização destas funções deve-se ao facto de que na próxima secção será apresentado o esquema de hibridação com o AG. Para o MatLab, que é a ferramenta utilizada neste trabalho, as funções que permitem ser conjugadas com o AG numa estrutura híbrida são: *fminsearch* e *fminunc*, ambas para problemas sem restrições, *patternsearch* para problemas com ou sem restrições e *fmincon* para problemas com restrições.

4.8.1. DESCRIÇÃO DAS FUNÇÕES *FMINCON* E *PATTERNSEARCH*

A função *fmincon*, está presente na *toolbox* de optimização do MatLab e é específica para problemas com restrições lineares ou não lineares de várias variáveis. Esta função utiliza o método de Programação Quadrática Sequencial (PSQ). O método PSQ tem-se mostrado de

grande eficiência para problemas de programação não lineares em relação ao sucesso das soluções encontradas. A cada iteração o problema a ser tratado é dividido em subproblemas, de forma a facilitar a busca do ponto ótimo, e estes por sua vez são resolvidos por programação quadrática. Mais informações sobre este método de procura podem ser consultadas nos documentos de ajuda do MatLab [*Guide MatLab*, 2009].

O *fmincon* possui a seguinte estrutura [MatLab, 2009]:

$$\text{Min } f(x) \text{ sujeito a } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases} \quad (16)$$

Onde x , representa o vector que define os valores iniciais das variáveis independentes, e $f(x)$ a função a minimizar; b e beq , são vectores que representam, respectivamente, os termos independentes das restrições lineares de desigualdade e igualdade; lb e ub são vectores que representam os limites inferiores e superiores das variáveis independentes, respectivamente; A e Aeq são matrizes que definem as funções das restrições lineares de desigualdade e igualdade, respectivamente; $c(x)$ e $ceq(x)$ são funções que tem como retorno vectores e definem respectivamente as restrições não lineares de desigualdade e igualdade.

A segunda função a ser descrita, *patternsearch*, está presente na *toolbox* de optimização de algoritmos genéticos e de procura directa. *Patternsearch* é um método de procura directa em padrões, que têm como objectivo determinar o ponto ótimo de uma função não linear utilizando como técnica de optimização a comparação de um conjunto de valores dados pelos pontos de uma certa região no espaço de pesquisa. Assim como os AGs, este método não utiliza derivadas da função objectivo e tende a ser um método de busca exploratório que tem como referência o comportamento da FO.

Tratando-se de um método de exploração do espaço de soluções, a função *fmincon* utiliza o método conhecido como “*hill climbing*”, ou subida da encosta, que tem como princípio avaliar os pontos adjacentes do espaço de procura, movendo-se na direcção que minimiza o resultado da função objectivo. O algoritmo *patternsearch* trabalha utilizando um conjunto de pontos padrão que, dependendo do comportamento da FO, pode ser expandido ou reduzido dentro de um determinado padrão estabelecido, por omissão, no MatLab. O

critério de paragem para esta função ocorre quando um determinado padrão mínimo é atingido.

A função *patternsearch* além de encontrar o mínimo de uma função não linear utilizando o método de procura padrão (*Pattern Search*), também utiliza um algoritmo de busca directa generalizado (*Generalized Pattern Search - GPS*). Caso o assunto desperte o interesse do leitor está disponível nos anexos da mesma bibliografia anteriormente referida [Guide MatLab, 2009] um vasto conjunto de referências sobre a função *patternsearch*.

Tais métodos de optimização procuram a solução a partir de uma semente. A semente é uma estimativa da solução e, quanto mais próxima estiver da solução, maior sucesso terá a optimização. Dependendo do tipo de problema, a dedução das sementes pode ser mais simples ou mais complexa. Neste último caso, é quase impossível estimar sementes boas; geralmente utiliza-se algum tipo de conhecimento prévio sobre o problema. De qualquer forma, é importante, ao menos, evitar utilizar sementes com valores absurdos. Neste sentido serão utilizados os valores dos parâmetros K_p , K_i e K_d encontrados pelo método de Ziegler-Nichols, o que garante que as sementes possuem valores aceitáveis para iniciar as simulações.

4.8.2. ESTRUTURA DAS FUNÇÕES *FMINCON* E *PATTERNSEARCH*

O algoritmo para estas funções é muito semelhante ao usado para a sintonia do controlador PID com algoritmos genéticos. Neste caso, o mesmo distribui-se por 2 ficheiros: função objectivo ("*objective_function_ID.m*") e a função de optimização ("*fmincon.m*") ou ("*patternsearch.m*"), que são usadas separadamente para encontrar o valor óptimo da função.

Para a função objectivo, o valor de *ID* refere-se ao índice de desempenho que se deseja simular. Tal função e tais índices serão omissos neste momento, visto que os mesmos foram descritos anteriormente.

O programa correspondente à função objectivo é o mesmo que foi utilizado pelos AGs, onde várias funções objectivo foram escritas com base no critério de desempenho do erro. Cada função objectivo é fundamentalmente a mesma, excepto para a secção do código que define o critério de desempenho de erro específico a ser implementado.

A sintaxe utilizada para as duas funções (*fmincon* e *patternsearch*) é:

$[x, fval, exitflag, output] = fmincon('nomearquivo', x0, opções)$

$[x, fval, exitflag, output] = patternsearch('nomearquivo', x0, opções)$, onde:

- x – representa a solução.
- $fval$ – o valor da FO calculada na solução x .
- $exitflag$ – indica a condição de saída. Se for > 0 , indica que a função convergiu para a solução “ x ”. Se for $= 0$, significa que o número máximo de iterações foi atingido. Se for < 0 , significa que a função não convergiu para a solução.
- $output$ – a estrutura com informações sobre a otimização.
- ‘nomearquivo’ – é o nome do arquivo que contém a FO.
- $x0$ – a semente para a otimização.
- opções – o vector de opções do algoritmo.

A diferença entre ambos os códigos ocorre apenas quando se faz a chamada da função de otimização: *fmincon* ou *patternsearch*.

4.8.3. AVALIAÇÃO DAS FUNÇÕES DE OPTIMIZAÇÃO

Nesta subsecção será possível analisar o desempenho do sistema quando se utilizam as funções de otimização do MatLab. Como já foi referido, para ambas as funções de otimização será utilizada como “semente” os valores referentes aos ganhos do controlador PID obtidos pelo método de Ziegler-Nichols.

Para as funções de otimização, o ponto dado como semente corresponde a um local no espaço de soluções e a busca pelo ponto óptimo ocorre a partir deste ponto inicial. Esta talvez seja a principal desvantagem que as funções do MatLab possuem em relação aos AGs, que se resume na total dependência do ponto de partida para a procura da melhor solução. Enquanto os AGs trabalham com um conjunto de pontos em simultâneo, e exploram maiores regiões no espaço de solução, as funções de otimização do MatLab iniciam a pesquisa na região correspondente ao ponto dado como semente, e a melhoria da

solução ocorre a partir desta região. Sabe-se que quanto mais afastado do óptimo global forem as sementes, pior será o desempenho destas funções.

A Tabela 12 apresenta os valores para cada parâmetro especificado e o valor de erro em função do ID atribuído. Nesta mesma tabela, de forma a comparar a resposta do sistema com estes dois métodos de optimização, reportou-se novamente os valores obtidos pelo AG para todos os ID da Tabela 10, excepto para o ID ITAE, que reflecte o estudo presente na Figura 19, em virtude de este traduzir o melhor desempenho para este ID.

Tabela 12 Parâmetros e especificações das respostas do sistema para Z-N, AG e *fmincon*

		Ganhos do Controlador			Desempenho da Malha			
		K_p	K_i	K_d	t_r	t_s	Mp%	Erro
Ziegler-Nichols		18	12,81	6,3216	0,579	10	61,7	
AG	IAE	55,9427	0,0296	102,7247	0,123	1,96	47,0	3,5825
	ITAE	41,8658	0,0002	45,7076	0,200	1,45	31,7	1,2622
	ISE	18,2520	0,6818	122,5410	0,112	3,04	46,6	1,8217
	ITSE	29,9945	0,1190	58,8304	0,181	1,89	30,8	0,2220
	MSE	16,5390	1,0525	123,2786	0,112	3,08	46,6	0,00906
<i>fmincon</i>	IAE	36,5201	0,0031	51,2930	0,196	1,48	30,6	3,6291
	ITAE	41,9313	0	45,6998	0,208	1,62	31,2	1,2618
	ISE	27,7518	0,0700	119,8186	0,113	2,43	47,1	1,8312
	ITSE	27,2368	0,1076	56,3510	0,187	1,96	29,1	0,2229
	MSE	21,7956	0	44,9778	0,221	2,25	22,9	0,0105

A Figura 20 ilustra a resposta do sistema em estudo nesta secção a uma entrada em degrau unitário ao aplicar os ganhos do controlador dados por AG e a função de optimização do MatLab *fmincon*.

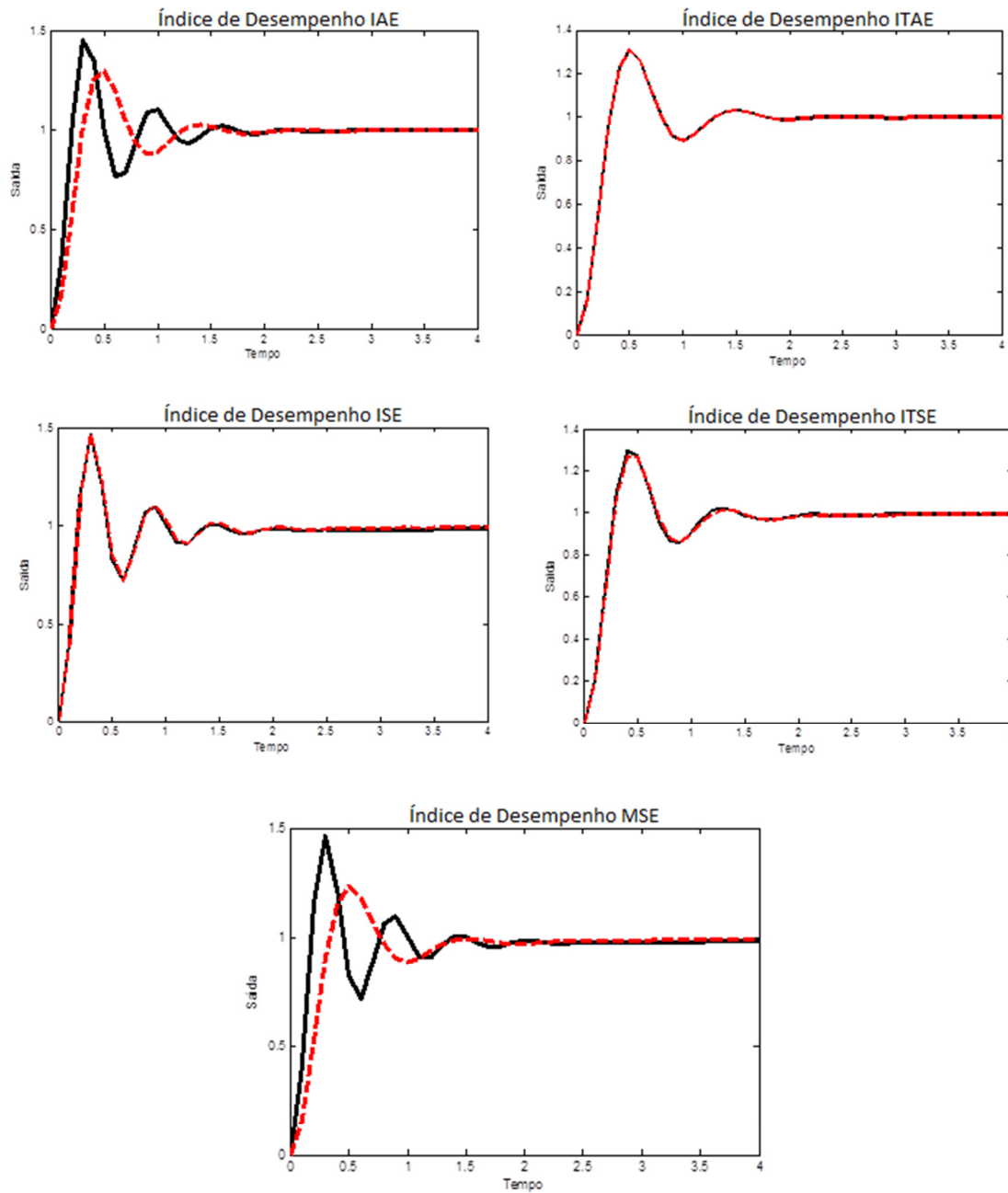


Figura 20 Resposta do Sistema a uma entrada em degrau unitário para AG (linha contínua) e *fmincon* (linha tracejada)

Seguindo a mesma analogia apresentada para a função *fmincon*, segue na Tabela 13, os resultados obtidos utilizando a função de otimização do MatLab *patternsearch*, em comparação com os resultados obtidos com o AG e método de Ziegler-Nichols.

Tabela 13 Parâmetros e especificações das respostas do sistema para Z-N, AG e *patternsearch*

		Ganhos do Controlador			Desempenho da Malha			
		K_p	K_i	K_d	t_r	t_s	Mp%	Erro
Ziegler-Nichols		18	12,81	6,3216	0,579	10	61,7	
AG	IAE	55,9427	0,0296	102,7247	0,123	1,96	47,0	3,5825
	ITAE	41,8658	0,0002	45,7076	0,200	1,45	31,7	1,2622
	ISE	18,2520	0,6818	122,5410	0,112	3,04	46,6	1,8217
	ITSE	29,9945	0,1190	58,8304	0,181	1,89	30,8	0,2220
	MSE	16,5390	1,0525	123,2786	0,112	3,08	46,6	0,00906
<i>patternsearch</i>	IAE	57,7642	0,0229	102,8128	0,123	1,95	47,3	3,5829
	ITAE	118,8770	78,8146	84,1966	0,134	1,96	55,4	2,1560
	ISE	16,5625	1,0561	123,2747	0,112	3,08	46,6	1,8217
	ITSE	29,9521	0,1064	58,7786	0,181	1,89	30,8	0,2220
	MSE	16,5625	1,0561	123,2747	0,112	3,08	46,6	0,00906

A Figura 21 ilustra a resposta do sistema, em estudo nesta secção, a uma entrada em degrau unitário ao aplicar os ganhos do controlador dados pelo AG e a função de optimização do MatLab *patternsearch*.

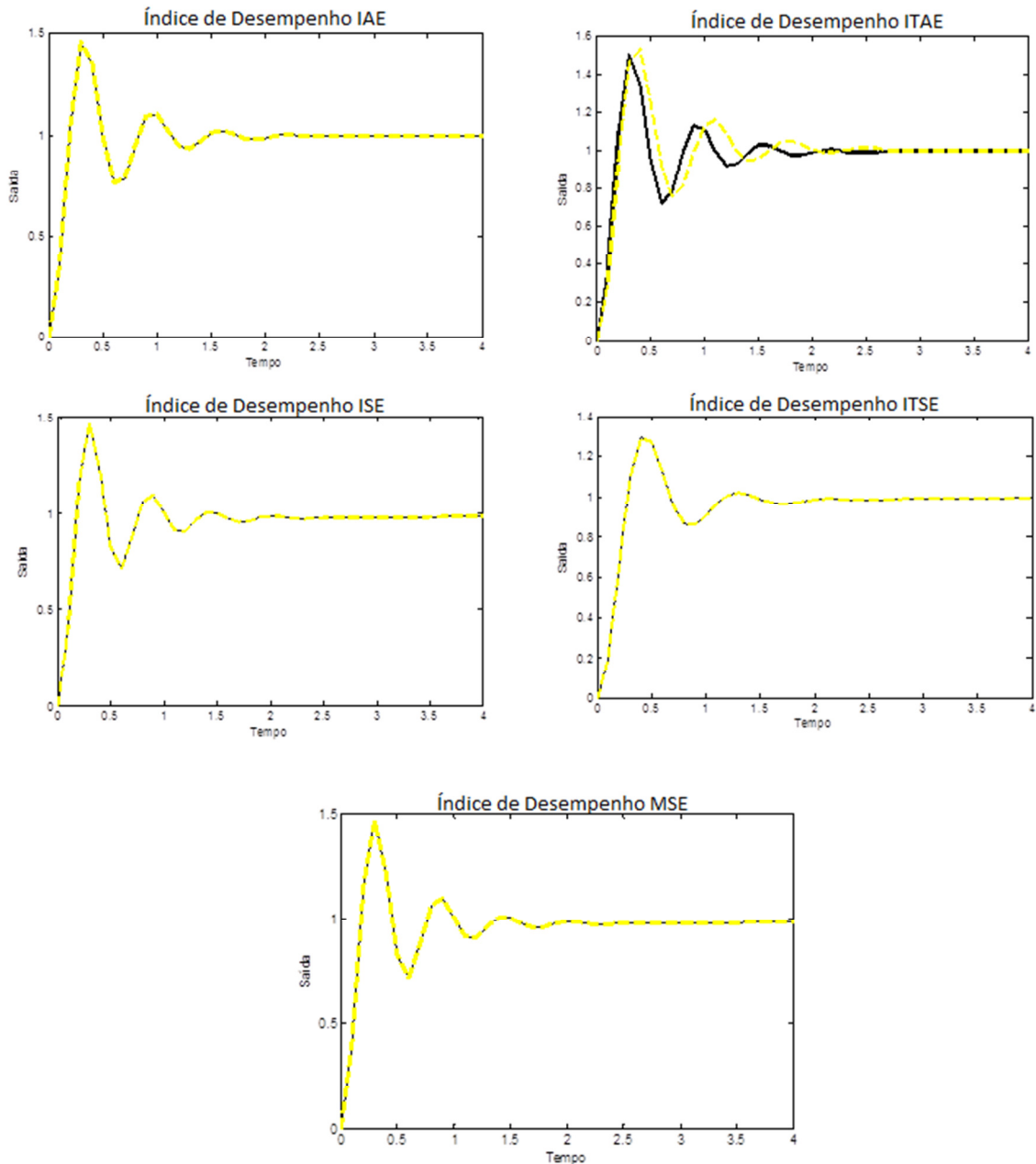


Figura 21 Resposta do Sistema a uma entrada em degrau unitário para AG (linha contínua) e *patternsearch* (linha tracejada)

Os resultados apresentados evidenciam que as funções de otimização do MatLab, apesar de obterem resultados ora inferiores ora iguais aos AGs, são ferramentas de otimização que podem ser utilizadas como uma solução inicial para a sintonia dos controladores PID, e que posteriormente esta solução pode ser melhorada com o uso de outras técnicas de otimização, como por exemplo os AGs.

Para o sistema em questão a função de otimização *patternsearch* foi a que mais se aproximou dos resultados obtidos com os AGs, sendo que, dos cinco índices de

desempenho analisados, três deles apresentaram a mesmo comportamento para ambos os algoritmos de otimização (ISE, ITSE e MSE).

A partir de tais resultados teve-se a sensibilidade de conjugar as potencialidades do AG com as técnicas de busca local das funções de otimização. Para tal pressuposto, surgiu o esquema de hibridação que será detalhado na secção seguinte.

4.9. ESQUEMA HÍBRIDO COM ALGORITMOS GENÉTICOS

O Algoritmo Genético Híbrido (AGH), também conhecido como Algoritmo Memético (AM), é um esquema desenvolvido com o intuito de conjugar as potencialidades do genético com o da procura local. Sabe-se que os AGs são eficazes no que respeita à procura global, mas apresentam deficiência quando se trata de refinar as soluções encontradas. Isto acontece porque, na fase final das interacções, os indivíduos da população compartilham praticamente o mesmo código genético, e qualquer melhoria que venha a acontecer nesta fase se deve ao operador de mutação, que além de actuar com uma baixa probabilidade age de forma aleatória.

Para inúmeros problemas, o desempenho do AG não se mostra satisfatório, visto que já existem métodos que fornecem uma solução aproximada num tempo computacional consideravelmente mais reduzido. Estas limitações do AG levam diversos investigadores a proporem modificações na sua forma tradicional, conjugando técnicas de procura local, ou ferramentas de outras metáheurísticas tais como: *Simulated Annealing* e *Tabu Search*, ou a desenvolverem versões paralelas para melhorar o desempenho dos AGs.

Para melhorar o desempenho e obter respostas satisfatórias nas situações em que os AGs simples não são particularmente bons, vários métodos de hibridação vêm sendo propostos. Segundo Davis [Davis, 1992] uma das formas mais utilizadas pelos AGHs é incorporar uma optimização local como um complemento a ser aplicado no operador de selecção. Assim, a optimização local é aplicada a cada filho gerado na população, e o operador de selecção somente será executado após a procura local ser concluída.

Vários são os motivos que levam ao estudo e desenvolvimento do AGH, entre eles a grande variedade de aplicações que obtiveram sucesso ao utilizarem versões híbridas desse algoritmo para resolver determinados tipos de problemas. Como exemplo destas aplicações cita-se, o bom resultado obtido em [Davis, 1992; Mitchell, 1999 e Ochi, 2009], ao utilizar

métodos de busca local em algoritmos genéticos, de forma a melhorar o resultado por meio do refinamento da solução. O AG pode atingir a região perto de um ponto ótimo, mas pode necessitar de muitas avaliações da função para atingir esta convergência [Mitchel, 1999].

Dado que os AGs são resistentes a ficarem presos em ótimos locais e falham no que diz respeito a refinar as soluções encontradas, seguiu-se o conselho de Davis [Davis, 1992-p.57], “(...) *Hibridizar sempre que possível*”. Segundo Dias [Dias, 2005], o sucesso dos algoritmos meméticos pode ser explicado como sendo uma consequência directa das diferentes abordagens que este assunto incorpora.

Segue na Figura 22, um diagrama esquemático do funcionamento do AGH proposto neste trabalho.

Nesta mesma figura observa-se que a solução encontrada pelo AG no que diz respeito aos ganhos do controlador PID, será então o ponto de partida, ou a semente a ser utilizada pela função de optimização do MatLab, de forma a proceder a um refinamento da solução fornecendo como saída o melhor resultado encontrado pelo conjunto dos dois sistemas.

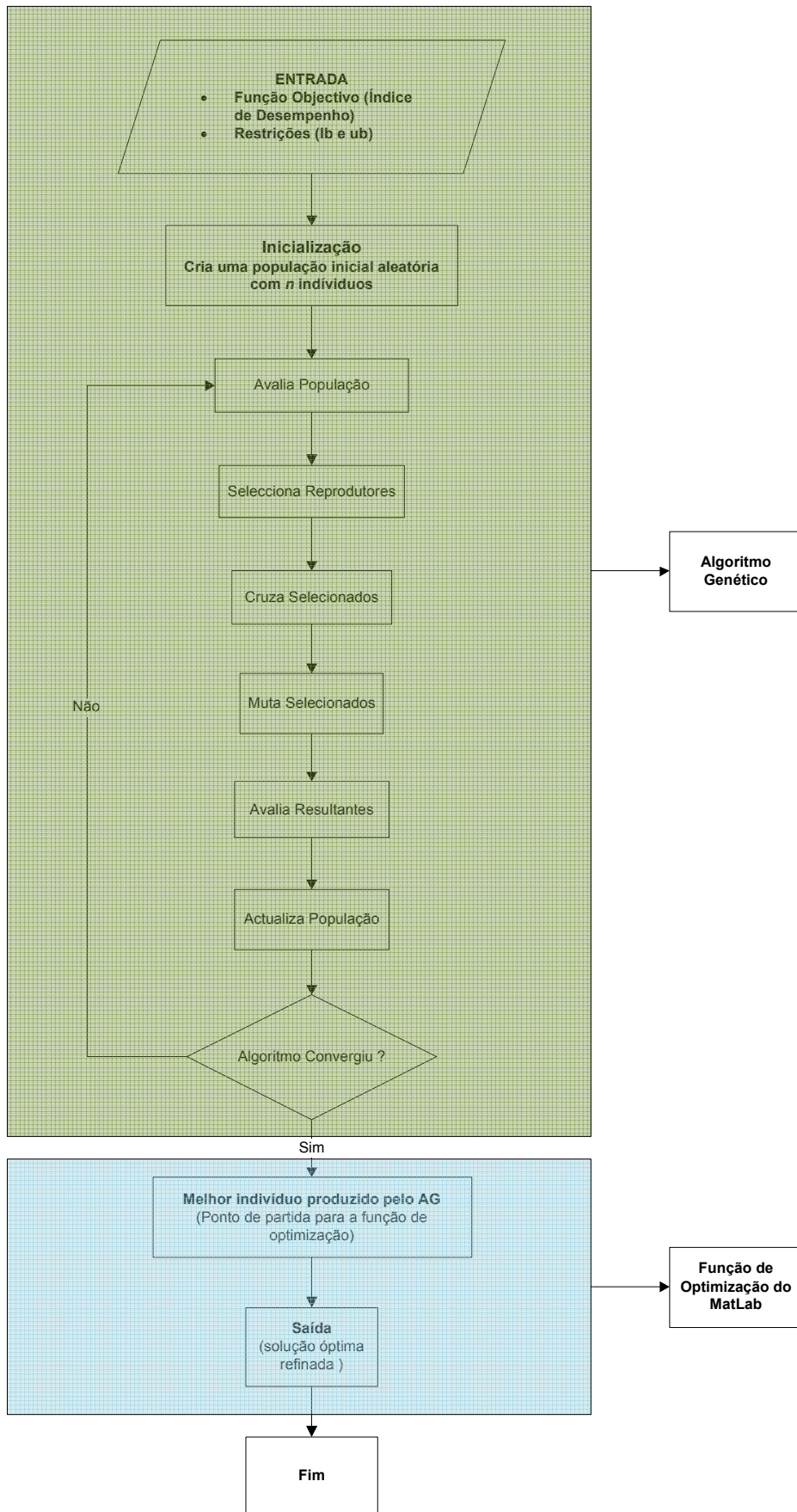


Figura 22 Diagrama esquemático do AGH

4.9.1. RESULTADOS DO MODELO HÍBRIDO

A metodologia que se apresenta de seguida, visa avaliar o desempenho do modelo híbrido comparando o resultado obtidos com as simulações que foram efectuadas na secção 4.7, quando foi utilizado o AG sem a conjugação de nenhuma função de optimização, com os resultados do AGH formado pela conjugação do AG com a função *fmincon* (AGH1) e do AG com a função *patternsearch* (AGH2).

De forma a validar os resultados obtidos, reportou-se novamente os valores de erro apresentados pelo AG sem funções de optimização do MatLab (Tabela 8), e juntou-se a estes os valores de erros obtidos com os modelos híbridos, gerando a Tabela 14 para os resultados do AG com o AGH1 e AGH2.

Tabela 14 Tabela de comparação dos resultados obtidos com AG, AGH1 e AGH2

		Valor de Erro para (AG - AGH1 - AGH2)														
População		IAE			ITAE			ISE			ITSE			MSE		
		200	3,5825	3,5825	3,5825	1,4884	1,4884	1,4878	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091
180	3,5761	3,5761	3,5767	1,4884	1,4884	1,4878	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091	0,0091	
160	3,5825	3,5825	3,5825	2,1574	2,1558	2,1558	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091	0,0091	
140	3,5826	3,5826	3,5825	1,4883	1,4883	1,4878	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091	0,0091	
120	3,5767	3,5768	3,5767	2,1575	2,1558	2,1558	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091	0,0091	
100	3,5830	3,5830	3,5830	2,1575	2,1558	2,1558	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091	0,0091	
80	3,5825	3,5825	3,5825	2,1576	2,1561	2,1561	1,8217	1,8217	1,8217	0,2220	0,2220	0,2220	0,0091	0,0091	0,0091	
60	3,5828	3,5828	3,5828	2,2139	2,2118	2,2118	1,8217	1,8217	1,8217	0,2229	0,2220	0,2220	0,0091	0,0091	0,0091	
40	4,0865	4,0865	4,0864	2,1575	2,1558	2,1558	1,8231	1,8222	1,8217	0,2311	0,2311	0,2220	0,0091	0,0091	0,0091	
20	3,7096	3,7078	3,5767	2,4021	2,4021	2,1561	1,8259	1,8258	1,8217	0,2239	0,2239	0,2220	0,0091	0,0091	0,0091	

Pela análise da Figura 23 é possível perceber a evolução do erro obtido para cada índice de desempenho face ao aumento da população. Nesta mesma figura, será possível observar, que uma vez mais, o ID MSE não difere com o aumento da população

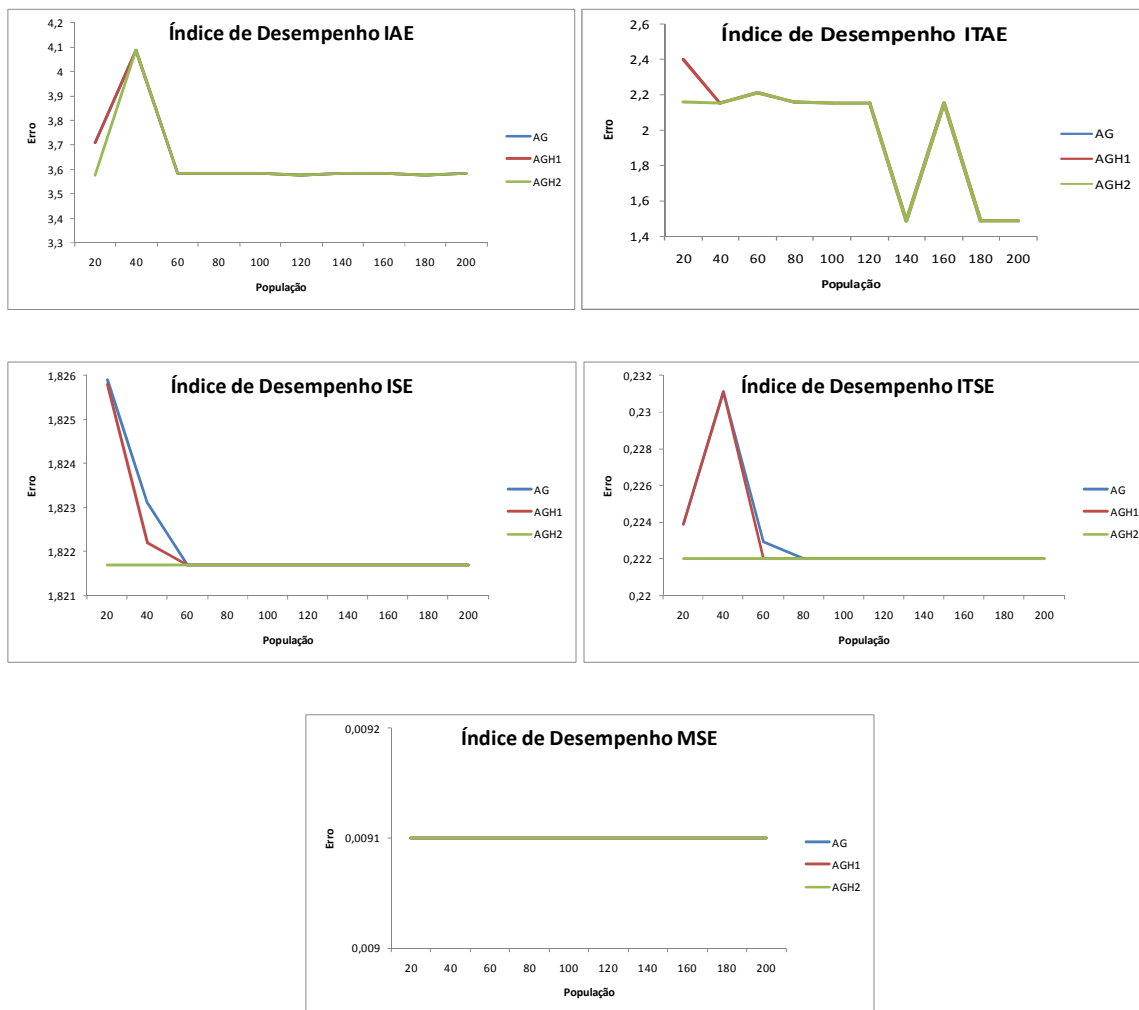


Figura 23 Variação do erro ao longo do aumento da população para AG, AGH1 e AGH2

Pelos resultados obtidos, pode-se observar que para os índices ISE e ITSE o sistema estabiliza mais rapidamente quando se utiliza o AG conjugado com uma de função otimização de procura local, nomeadamente a função *patternsearch*. Para os índices IAE e ITAE, verifica-se que o valor de erro diminui para os instantes iniciais quando novamente é utilizado a função *patternsearch* conjugada com o AG. Para o índice MSE observa-se, que uma vez mais, o valor de erro não difere com o aumento da população para nenhum dos algoritmos utilizados neste exemplo.

A partir do ponto onde o sistema estabiliza, não ocorre nenhuma mudança ao acrescentar as funções de otimização do MatLab. Este facto indica que o AG chegou a um ponto possivelmente óptimo para o desempenho do sistema. Este esquema híbrido será de grande utilidade no capítulo seguinte, onde serão estudados sistemas considerados mais difíceis de serem otimizados, devido às características físicas que dificultam a sua sintonia.

4.1. CONCLUSÃO

Este capítulo apresentou uma análise pormenorizada de várias situações de simulação para o sistema em destaque. O mesmo serviu como base teórica para as aplicações que se seguem, de forma a avaliar a robustez do AG implementado, quando utilizado em outros sistemas considerados mais difíceis de serem controlados.

Neste momento, pode-se constatar os grandes benefícios de utilizar o AG da *toolbox* de otimização do MatLab, validando a facilidade de implementação e a conjugação com outras rotinas de otimização, nomeadamente funções de otimização do MatLab e o ambiente Simulink.

5. APLICAÇÕES DOS ALGORITMOS GENÉTICOS HÍBRIDOS

Este capítulo destina-se a utilizar uma série de exemplos ilustrativos encontrados na literatura sobre a sintonia de controladores PID, de forma a validar a robustez do AG face a outros métodos de optimização já existentes quando aplicados a sistemas mais complexos. Tal comparação é de grande valor no âmbito desta tese, visto que o AG proposto será avaliado em sistemas conhecidos e considerados difíceis de serem controlados devido a factores como o atraso na resposta do sistema e constantes de tempo elevadas.

O objectivo proposto é melhorar a resposta dinâmica dos sistemas que serão apresentados, utilizando os AGs para determinar os parâmetros do controlador PID com base na minimização do erro dado pelos índices de desempenho IAE, ITAE, ISE, ITSE e MSE. Para as simulações que se seguem serão então utilizadas as duas estruturas híbridas descritas no capítulo anterior, designadas como AGH1 e AGH2.

5.1. INTRODUÇÃO

Determinar os parâmetros de um controlador PID, para que o sistema apresente um desempenho considerado óptimo não é tarefa fácil. De facto para a determinação dos parâmetros do PID que melhor se ajustam aos objectivos pretendidos para um sistema, é necessário analisar o tipo de sistema em estudo e o método de sintonia a ser aplicado.

Ao longo deste trabalho foram apresentadas técnicas de sintonia de controladores como as desenvolvidas por Ziegler-Nichols e por Cohen e Coon. Estas técnicas são baseadas em modelos matemáticos de baixa ordem, nomeadamente um modelo com um ganho K , uma constante de tempo T e um atraso τ na resposta do sistema ao degrau unitário, representado pela seguinte função de transferência:

$$G(s) = \frac{Ke^{-s\tau}}{Ts + 1} \quad (17)$$

Sabe-se que as técnicas de Ziegler-Nichols são as mais difundidas quando se trata da sintonia de controladores industriais. Entretanto verifica-se que os controladores sintonizados por este método tendem normalmente a não apresentar resultados satisfatórios, no que diz respeito à resposta a um degrau unitário, apresentando sobrelevações elevadas e grandes tempos de estabelecimento.

Neste sentido, outras técnicas de sintonia foram desenvolvidas ao longo dos últimos anos de modo a melhorar o desempenho do PID, como por exemplo os métodos baseados na minimização do integral do erro, os métodos baseados no lugar geométrico de raízes [Astron, 1995], ou métodos baseados em técnicas de optimização [Zhuang e Atherton, 1993].

A técnica de sintonia de Zhuang e Atherton (Z-A) tem-se mostrado bastante eficiente no que respeita aos sistemas de controlo em malha fechada. Este método de sintonia fará parte das comparações que serão feitas para os sistemas apresentados nesta secção, merecendo uma introdução sobre o assunto.

5.2. O MÉTODO DE ZHUANG E ATHERTON

Zhuang e Atherton [Zhuang e Atherton, 1993] propuseram um conjunto de fórmulas para a sintonia do controlador PID, baseado na minimização de uma integral. Segundo estes autores os critérios de desempenho podem ser expressos da seguinte forma:

$$J_n(\theta) = \int_0^{\infty} \{t^n e(\theta, t)\}^2 dt \quad (18)$$

Onde, θ representa os parâmetros do controlador PID que podem ser determinados através da minimização de $J_n(\theta)$ e $e(\theta, t)$ representa o sinal de erro. O valor de n é compreendido entre 0 e 2 e tende a determinar qual o critério de desempenho a ser utilizado. Para $n = 0$,

utiliza-se o índice ISE; para $n = 1$, utiliza-se o índice ITSE e finalmente para $n = 2$ utiliza-se o IT^2SE (*Integral Squared Time-Squared Weighted Error*).

Para o sistema apresentado na Figura 24, Z-A propuseram duas estratégias de controlo. Uma para a variação do valor de entrada $r(t)$ e outra para a variação do sinal de perturbação $q(t)$.

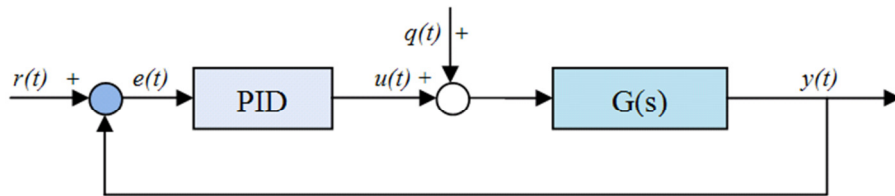


Figura 24 Sistema de controlo PID com realimentação

5.2.1. SINTONIA PARA A VARIAÇÃO DO SINAL DE ENTRADA

Se o sistema puder ser representado por uma função de transferência de primeira ordem, com tempo de atraso (FOPDT), os parâmetros do controlador PI podem ser determinados pelas seguintes equações [Zhuang e Atherton, 1993]:

$$K_p = \frac{a_1}{K} (\tau/T)^{b_1} \quad (19)$$

$$T_i = \frac{\tau}{a_2 + b_2(\tau/T)} \quad (20)$$

Os valores para os coeficientes (a_j e b_j) para $j = 1, 2$ respectivamente, podem ser obtidos pela Tabela 15.

Tabela 15 Parâmetros para variação do valor de referência de um controlador PI

τ/T	[0,1 – 1]			[1,1 – 2]		
Critério	ISE	ISTE	IST ² E	ISE	ISTE	IST ² E
a_1	0,980	0,712	0,569	1,072	0,786	0,628
b_1	-0,892	-0,921	-0,951	-0,560	-0,559	-0,583
a_2	0,690	0,968	1,023	0,648	0,883	1,007
b_2	-0,155	-0,247	-0,179	-0,114	-0,158	-0,167

Para um controlador do tipo PID os ganhos podem ser determinados por (19), (20) e pela equação (21).

$$T_d = a_3 T (\tau/T)^{b_3} \quad (21)$$

Os valores para os coeficientes (a_j e b_j) para $j = 1, 2$ e 3 respectivamente, podem ser obtidos pela Tabela 16.

Tabela 16 Parâmetros para variação do valor de referência de um controlador PID

τ/T	[0,1 – 1]			[1,1 – 2]		
Critério	ISE	ISTE	IST ² E	ISE	ISTE	IST ² E
a_1	1,048	1,042	0,968	1,154	1,142	1,061
b_1	-0,897	-0,897	-0,904	-0,567	0,579	-0,583
a_2	1,195	0,987	0,977	1,047	0,919	0,892
b_2	-0,368	-0,238	-0,253	-0,220	-0,172	-0,165
a_3	0,489	0,385	0,316	0,490	0,384	0,315
b_3	0,888	0,906	0,892	0,708	0,839	0,832

Em alguns casos é incluído um filtro na acção derivativa do sinal de saída, o que corresponde a um sinal de controlo $u(t)$ dado pela seguinte equação:

$$U(s) = K_p \left(1 + \frac{1}{T_i s} \right) E(s) - \frac{s T_d}{1 + s T_d / N} y(s) \quad (22)$$

Para este tipo de controlador os valores para os coeficientes (a_j e b_j) para $j = 1, 2$ e 3 respectivamente, podem ser obtidos pela Tabela 17.

Tabela 17 Sintonia PID com acção derivativa no sinal de saída

τ/T	[0,1 – 1]			[1,1 – 2]		
Critério	ISE	ISTE	IST ² E	ISE	ISTE	IST ² E
a_1	1,260	1,053	0,942	1,295	1,120	1,001
b_1	-0,887	-0,930	-0,933	-0,619	-0,625	-0,624
a_2	0,701	0,736	0,770	0,661	0,720	0,754
b_2	-0,147	-0,126	-0,130	-0,110	-0,114	-0,116
a_3	0,375	0,349	0,308	0,378	0,350	0,308
b_3	0,886	0,907	0,897	0,756	0,811	0,813

5.2.2. SINTONIA PARA A VARIAÇÃO DO SINAL DE PERTURBAÇÃO

Apesar de os controladores serem projectados para a sintonia de um processo devido à variação do sinal de entrada, existem casos em que podem ocorrer perturbações que são prejudiciais ao desempenho do sistema. Desta forma Z-A também dispuseram um conjunto de equações para minimizar o efeito destas perturbações.

Para o controlador do tipo PI, os parâmetros podem ser calculados por:

$$K_p = \frac{a_1}{T} (\tau/T)^{b_1} \quad (23)$$

$$T_i = \frac{T}{a_2} (\tau/T)^{b_2} \quad (24)$$

Para o cálculo dos parâmetros do controlador do tipo PID, acrescenta-se a equação (25), em complemento às equações (23) e (24).

$$T_d = a_3 T (\tau/T)^{b_3} \quad (25)$$

Os valores para os coeficientes (a_j e b_j) para $j = 1, 2$ e 3 respectivamente, podem ser obtidos pelas Tabelas 18 e 19.

Tabela 18 Sintonia PI para a presença de perturbação

τ/T	0,1 - 1			1,1 - 2		
Critério	ISE	ISTE	IST ² E	ISE	ISTE	IST ² E
a_1	1,279	1,015	1,021	1,346	1,065	1,076
b_1	-0,945	-0,957	-0,953	-0,675	-0,673	-0,648
a_2	0,535	0,667	0,629	0,552	0,687	0,650
b_2	0,586	0,552	0,546	0,438	0,427	0,442

Tabela 19 Sintonia PID para a presença de perturbação

τ/T	0,1 - 1			1,1 - 2		
Critério	ISE	ISTE	IST ² E	ISE	ISTE	IST ² E
a_1	1,473	1,468	1,531	1,524	1,515	1,592
b_1	-0,970	-0,970	-0,960	-0,735	-0,730	-0,705
a_2	1,115	0,942	0,971	1,130	0,957	0,957
b_2	0,753	0,725	0,746	0,641	0,598	0,597
a_3	0,550	0,443	0,413	0,552	0,444	0,414
b_3	0,948	0,939	0,933	0,851	0,847	0,850

Apesar das tabelas apresentadas nesta subsecção fornecerem uma base simples para calcular os parâmetros de um controlador PID com base no critério de optimização de Z-A, encontra-se disponível no Anexo A desta dissertação, o código desenvolvido em MatLab para o cálculo destes parâmetros.

5.3. APLICAÇÃO DA METODOLOGIA PROPOSTA: EXEMPLOS ILUSTRATIVOS

Nesta secção será aplicado o método de sintonia de controladores PID com AG, baseado na minimização do erro dado pelos índices de desempenho IAE, ITAE, ISE, ITSE e MSE. Face aos resultados obtidos na secção anterior, o AG que será utilizado para os exemplos desta secção será conjugado com as funções de optimização do MatLab, e para efeito de validação os resultados desses modelos híbridos serão comparados com o método clássico de sintonia de Ziegler-Nichols (Z-N) e o método de optimização de Z-A.

Para cada sistema será apresentado uma tabela que descreve os parâmetros de inicialização da população do AGH. Os outros parâmetros como tipo de selecção, função de geração, função de cruzamento e critério de paragem permaneceram os mesmos parâmetros descritos na subsecção 4.6.2.

Seguindo a mesma linha de raciocínio, será introduzido na população do AGH um indivíduo que corresponde aos valores de K_p , K_i e K_d obtidos pelo método de Z-N. Os três primeiros exemplos que serão apresentados foram retirados do artigo que se intitula “Automatic tuning of optimum PID controllers” [Zhuang e Atherton, 1993].

- **EXEMPLO 1: SISTEMA DE 5ª ORDEM**

O primeiro exemplo é formado por um sistema de quinta ordem, cuja função de transferência é dada por :

$$G(s) = \frac{1}{(s + 1)^5} \quad (26)$$

A estrutura deste primeiro exemplo, elaborada em Simulink, é apresentada na Figura 25.

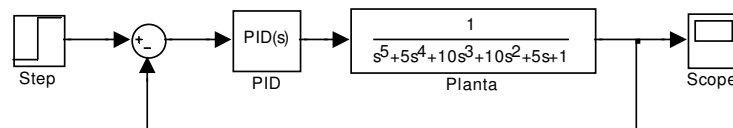


Figura 25 Diagrama de blocos para o exemplo 1

Os parâmetros de inicialização da população do AG encontram-se descritos na Tabela 20. Esta tabela descreve uma população de 140 indivíduos, sendo que um destes indivíduos representa os ganhos do controlador dados pela heurística de Z-N. Para o limite dos valores

da população inicial, teve-se em conta um grande número de simulações que foram realizadas para este exemplo, aliada ao facto de se estabelecer um intervalo que fosse suficientemente grande para a pesquisa do AG no espaço de soluções, mas que não sobrecarrega-se o algoritmo. Sendo assim, os limites para a população inicial estão compreendidos entre 0 e 50.

Tabela 20 Parâmetros de inicialização da população do AG para o exemplo 1

Parâmetros de inicialização do AG	
Tamanho da População	140
Número de Gerações	100
População Inicial	[1,607; 0,358; 1,803]
Limite da População Inicial	[0; 50]
Limite dos valores de [K_p ; K_i ; K_d]	lb = [0; 0; 0]
	ub = [50; 30; 50]

A Tabela 21 especifica os ganhos do controlador PID para os métodos que servem de base para avaliar os resultados encontrados, ou seja, Z-N, Z-A, AGH1 e AGH2, bem como os valores para o tempo de subida t_r , o tempo de estabelecimento t_s e o valor de sobreelongação MP%.

Tabela 21 Ganhos do controlador PID e especificações temporais para o exemplo 1

	Método	K_p	K_i	K_d	t_r (s)	t_s (s)	Mp%
	Z-N	1,607	0,358	1,803	2,47	14,40	13,10
	Z-A	1,618	0,3072	1,8852	2,49	18,70	8,30
AGH1	IAE	1,4631	0,4675	2,5850	2,30	14,20	10,10
	ITAE	1,2439	0,3580	1,6627	2,93	7,61	4,98
	ISE	1,5220	0,6835	4,5985	1,71	24,70	20,40
	ITSE	1,5508	0,5064	3,1233	2,07	13,60	12,10
	MSE	1,5234	0,6833	4,5970	1,69	24,80	20,50
	AGH2	IAE	1,4871	0,4830	2,6605	2,30	14,20
ITAE		1,2331	0,3536	1,6370	2,97	7,59	4,60
ISE		1,5216	0,6836	4,5993	1,71	24,70	20,40
ITSE		1,5560	0,5075	3,1215	2,07	13,60	12,10
MSE		1,5259	0,6871	4,6155	1,70	24,70	20,60

Para o primeiro método apresentado na Tabela 21, Z-N, o valor obtido para o tempo de subida assemelha-se ao encontrado pelo método de optimização de Z-A. Quando se trata do tempo de estabelecimento, observa-se que o sistema estabiliza mais rapidamente quando se utiliza o método de Z-N, embora quando se trata do valor de sobreelongação

este método apresenta um desempenho bem inferior, quando comparado com o método de Z-A.

Os resultados encontrados pelos AGH1 e AGH2 são bem semelhantes entre si, e para o exemplo em questão, o melhor desempenho para o sistema ocorreu para o índice ITAE. Para o sistema do exemplo 1 o AGH proposto, quando se utiliza o índice de desempenho ITAE, apresenta uma considerável melhoria no que diz respeito ao tempo de subida, ao tempo de estabelecimento e ao valor máximo de sobreelongação. Dentre os métodos de sintonia que foram utilizados pode-se dizer que os resultados obtidos levam a validar o AGH2 como o mais indicado para o sistema em questão.

A Figura 26 apresenta a resposta do sistema a uma entrada do tipo degrau unitário para os métodos de sintonia de Z-N, método óptimo de Z-A e AGH2, cujo índice que se encontra especificado é o ITAE face aos resultados obtidos.

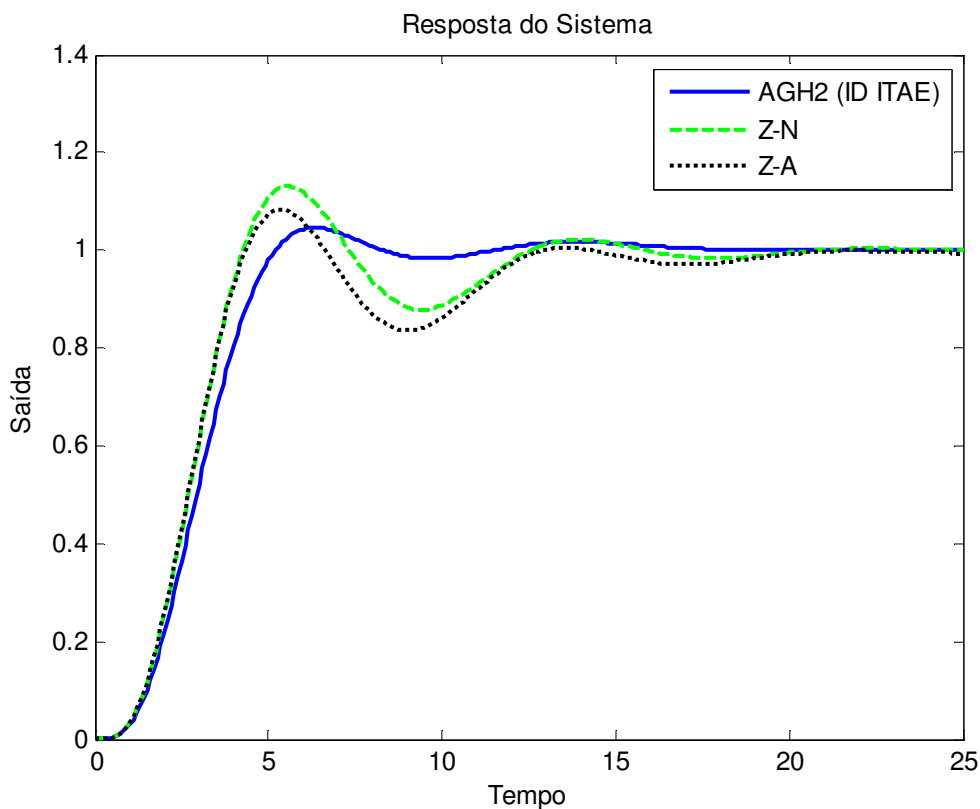


Figura 26 Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH2 com índice de desempenho ITAE, para o exemplo 1

- **EXEMPLO 2: SISTEMA DE 2ª ORDEM COM TEMPO DE ATRASO**

O segundo exemplo é formado por um sistema de segunda ordem com tempo de atraso dado pela seguinte função de transferência

$$G(s) = \frac{1,2e^{-10s}}{(5s + 1)(2,5s + 1)} \quad (27)$$

Neste sistema é introduzida uma perturbação em degrau de amplitude $p(t - t_0) = 0,5$ no instante $t_0 = 100$ s. O diagrama de blocos deste sistema, desenvolvido em Simulink, está apresentado na Figura 27.

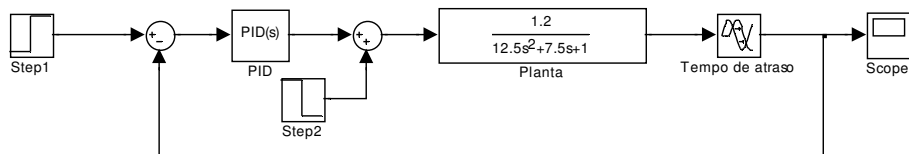


Figura 27 Diagrama de blocos para o exemplo 2

Para este sistema, os parâmetros de inicialização da população do AG encontram-se descritos na Tabela 22. Os valores referentes ao tamanho da população e o número de gerações foram definidos face a um grande conjunto de simulações que foram realizadas de forma a obter o melhor desempenho para o sistema face à conjugação destes dois parâmetros e tendo em conta todos os IDs que foram estudados. Desta forma estipulou-se um valor de 160 indivíduos para o tamanho da população e um número de gerações igual a 80.

Novamente foi introduzido na população inicial um indivíduo correspondente aos ganhos do controlador obtidos por Z-N e os limites dos valores da população inicial foram definidos tendo em conta a necessidade de se ter uma região que fosse suficientemente adequada para a pesquisa do AG.

Tabela 22 Parâmetros de inicialização da população do AG para o exemplo 2

Parâmetros de inicialização do AG	
Tamanho da População	160
Número de Gerações	80
População Inicial	[0,771; 0,04727; 3,14337]
Limite da População Inicial	[0; 50]
Limite dos valores de [K_p ; K_i ; K_d]	lb = [0; 0; 0]
	ub = [10; 5; 50]

A Tabela 23 especifica os ganhos do controlador PID para os três métodos, bem como os valores de t_r , t_s e o valor de MP%.

Tabela 23 Ganhos do controlador PID e especificações temporais para o exemplo 2

	Método	K_p	K_i	K_d	t_r (s)	t_s (s)	Mp%
	Z-N	0,7710	0,0473	3,1434	6,50	91,3	10,60
	Z-A	0,6550	0,0632	2,6878	7,69	55,2	7,99
AGH1	IAE	0,6515	0,0660	2,6705	7,62	55,3	9,58
	ITAE	0,6591	0,0656	2,5230	7,69	55,1	10,70
	ISE	0,6824	0,0805	4,0935	5,18	84,6	23,20
	ITSE	0,7426	0,0863	4,2708	4,49	86,3	32,40
	MSE	0,7031	0,0754	3,6889	5,67	61,7	21,00
AGH2	IAE	0,6482	0,0651	2,5932	7,77	55,2	8,94
	ITAE	0,6862	0,0695	2,8625	6,94	55,4	14,80
	ISE	0,6808	0,0810	4,1473	5,10	84,7	23,50
	ITSE	0,7425	0,0863	4,2724	4,49	86,3	32,30
	MSE	0,6812	0,0810	4,1486	5,10	84,7	23,60

A partir dos resultados obtidos, pode-se dizer que o desempenho do sistema sintonizado por Z-N apesar de apresentar um bom resultado para o tempo de subida apresenta sobreelongações mais elevadas (neste caso, subelongações) que podem ser extremamente prejudiciais ao sistema. Em relação ao tempo de estabelecimento, esta heurística, apresentou o maior valor de t_s , perante as demais. Para o sistema em análise, o valor de t_s obtido por Z-N é igual a 91,3 s para uma entrada em degrau unitário.

Para o critério de Z-A, o sistema apresentou um bom compromisso para o desempenho do sistema proposto, com sobreelongações admissíveis e um bom resultado para o tempo de estabelecimento. Estes resultados verificam-se tanto para as variações na entrada de referência, como para variações na entrada de perturbação.

Novamente os resultados obtidos com o AGH1 e AGH2 são bem semelhantes para todos os parâmetros analisados na Tabela 23, podendo dizer-se que o melhor compromisso em termos de t_r , t_s e $MP\%$, ocorreu para o AGH2 utilizando o índice de desempenho IAE.

A Figura 28 apresenta a resposta do sistema a uma entrada do tipo degrau unitário para os métodos de sintonia de Z-N, método óptimo de Z-A e AGH2, cujo índice que se encontra especificado é o IAE, face aos melhores resultados obtidos. Neste mesmo gráfico é evidente o efeito da perturbação no sistema no instante $t = 100$ s.

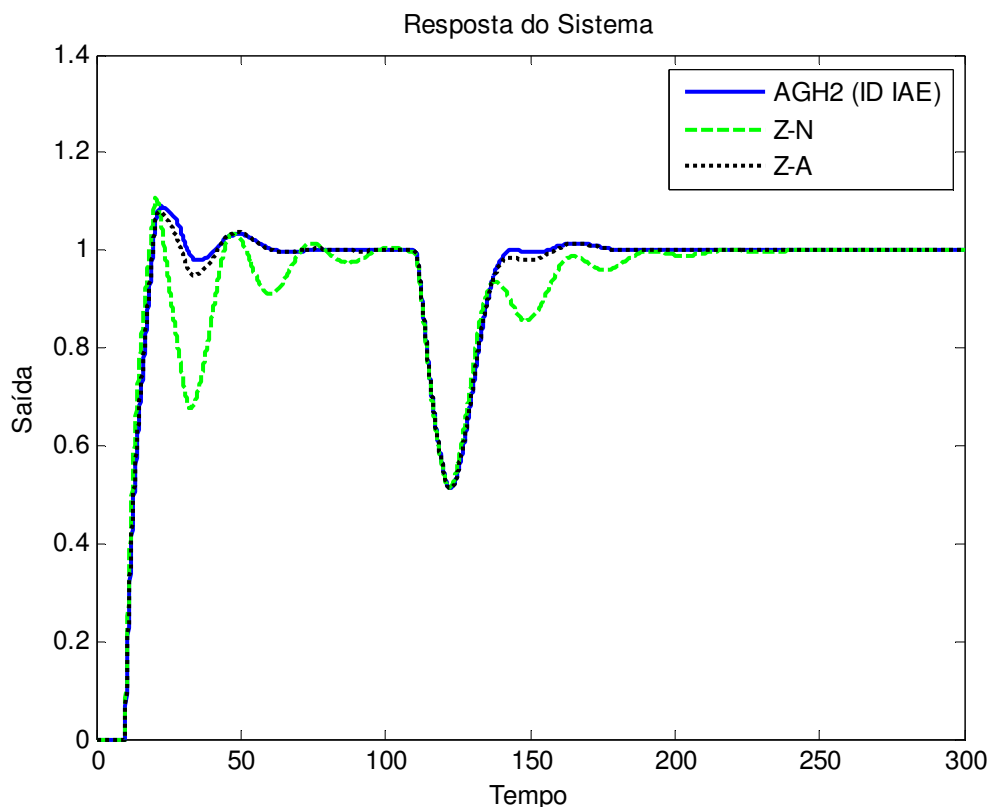


Figura 28 Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH2 com índice de desempenho IAE, para o exemplo 2

- **EXEMPLO 3: SISTEMA DE 2ª ORDEM COM A PRESENÇA DE UM ZERO NA FUNÇÃO DE TRANSFERÊNCIA**

Para o exemplo 3 tem-se um sistema de segunda ordem com tempo de atraso igual a 0,1 s. A função de transferência para este processo é dada por :

$$G(s) = \frac{2(0,5s + 1)e^{-0,1s}}{(s + 1)(4s + 1)} \quad (28)$$

Este sistema além de possuir um tempo de atraso, possui um zero na função de transferência localizado no semi-plano esquerdo. Segundo Ogata [Ogata, 2003] os zeros da função de transferência não influenciam a estabilidade do sistema que fica determinada apenas pelos seus pólos. Entretanto, para sistemas de segunda ordem, como é o caso deste exemplo, a presença do zero na função de transferência acarreta um alto valor de sobreelongação, facto que é constatado quando observamos a resposta do sistema obtida com o método de Z-N.

O diagrama de blocos deste sistema, feito em Simulink, é apresentado na Figura 29.

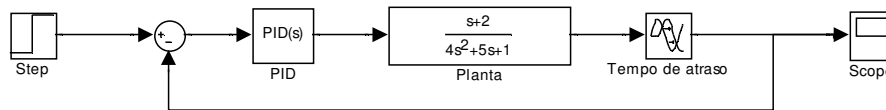


Figura 29 Diagrama de blocos para o exemplo 3

Para o exemplo 3, a Tabela 24 descreve os parâmetros de inicialização da população do AG.

Tabela 24 Parâmetros de inicialização da população do AG para o exemplo 3

Parâmetros de inicialização do AG	
Tamanho da População	200
Número de Gerações	140
População Inicial	[36,304; 175,891; 1,8878]
Limite da População Inicial	[0; 300]
Limite dos valores de [K_p ; K_i ; K_d]	lb = [0; 0; 0]
	ub = [200; 300; 100]

Observa-se que para este sistema, os limites inferiores e superiores especificados para os ganhos do controlador são maiores quando comparados com os exemplos anteriores. A escolha destes limites é feita tendo por base os valores de K_p , K_i e K_d encontrados pelo método de Ziegler-Nichols, e de forma a garantir que os ganhos encontrados pelo AG estejam dentro de valores aceitáveis para o desempenho do sistema. Estes limites definem a região do espaço a ser pesquisada pelo AG, e quanto maior esta região mais lenta se torna a convergência do AG.

A Tabela 25 especifica os ganhos do controlador PID para os três métodos, bem como os valores de t_r , t_s e o valor de MP%.

Tabela 25 Ganhos do controlador PID e especificações temporais para o exemplo 3.

	Método	K_p	K_i	K_d	t_r (s)	t_s (s)	Mp%
	Z-N	36,3040	175,8914	1,8878	0,0394	1,270	72,3
	Z-A	30,8580	4,0860	1,3916	0,0649	0,994	20,5
AGH1	IAE	28,4752	5,4190	1,5055	0,0666	1,100	16,6
	ITAE	42,1101	175,6884	1,7380	0,0378	0,867	88,5
	ISE	26,1852	2,4252	1,8836	0,0592	1,050	18,3
	ITSE	27,2974	3,9166	1,6772	0,0638	1,080	17,3
	MSE	42,6490	55,4883	1,7413	0,0391	1,190	68
	AGH2	IAE	28,4879	5,4261	1,5059	0,0665	1,100
ITAE		41,0237	95,7770	1,4034	0,0463	0,720	68,7
ISE		26,1846	2,4260	1,8832	0,0592	1,050	18,3
ITSE		27,2951	3,9153	1,6776	0,0638	1,080	17,3
MSE		26,1841	2,4262	1,8836	0,0592	1,050	18,3

A partir dos resultados obtidos, verifica-se o baixo desempenho do sistema quando sintonizado pelo critério de Z-N, devido a um valor elevado de sobreelongação e tempo de estabelecimento. Para os resultados obtidos com o método de Z-A, observa-se novamente um bom compromisso em termos de t_r , t_s e Mp%.

Nesta mesma tabela observa-se a maior discrepância de resultados entre o AGH1 e AGH2, quando utilizamos o índice MSE, revelando que para este índice o AGH2 obteve melhores resultados. Para os outros índices ambos os algoritmos obtiveram resultados semelhantes.

A Figura 30 apresenta a resposta do sistema a uma entrada do tipo degrau unitário para os métodos de sintonia de Z-N, método óptimo de Z-A e AGH2, cujo índice que se encontra especificado é o IAE, face aos melhores resultados em termos de tempo de estabelecimento e sobreelongação.

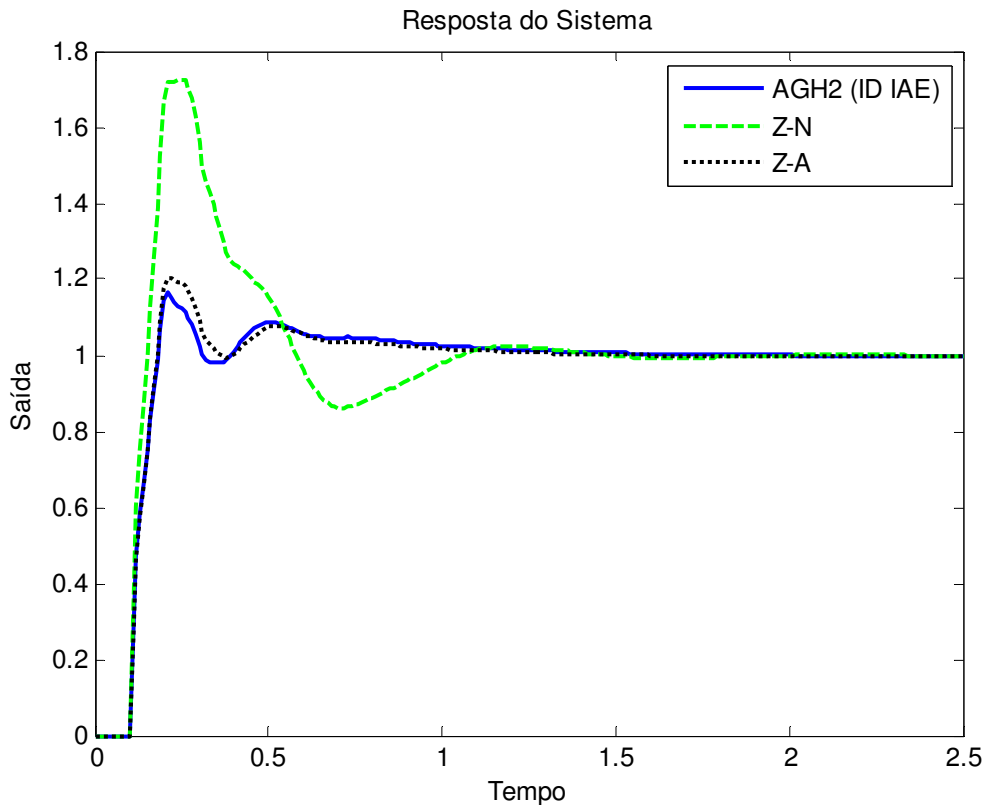


Figura 30 Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH2 com índice de desempenho IAE, para o exemplo 3

- **EXEMPLO 4: SISTEMA DE DIFUSÃO DE CALOR DE 1ª ORDEM COM ELEVADO TEMPO DE ATRASO.**

Para o exemplo 4, utilizou-se a função de transferência de um sistema de difusão de calor, fruto do estudo efectuado por Jesus [Jesus, 2007]. Este modelo representa uma superfície plana, perfeitamente isolada e de 3 m de comprimento, onde se aplica uma temperatura constante em $x = 0$ e se analisa a difusão de calor ao longo da coordenada horizontal x , correspondente ao comprimento, conforme pode ser observado no esboço presente na Figura 31.

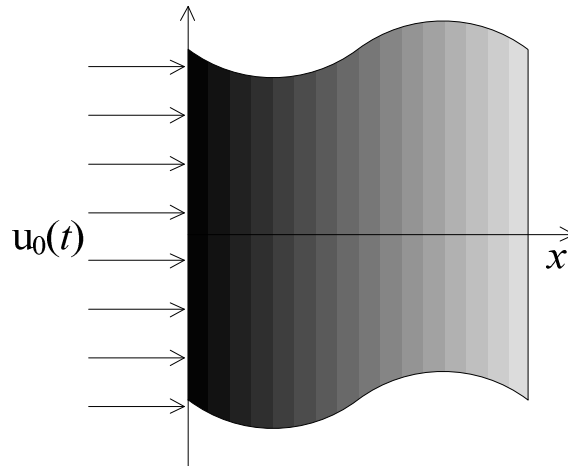


Figura 31 Sistema de Difusão de calor [Jesus, 2007]

A função de transferência descrita para o sistema de difusão de calor, a ser otimizada, é dada pela equação [Jesus, 2007]:

$$G(s) = \frac{0,52e^{-28s}}{162s + 1} \quad (29)$$

O diagrama de blocos para este sistema, elaborado em Simulink, é apresentado na Figura 32.

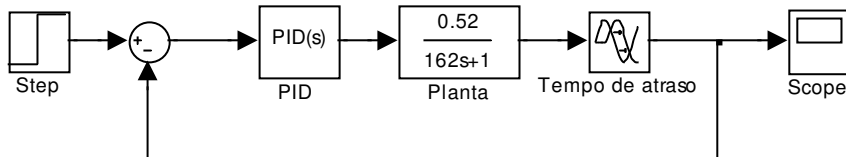


Figura 32 Diagrama de blocos para o exemplo 4

No trabalho realizado por Jesus [Jesus, 2007], foram obtidos os seguintes valores para os parâmetros K_p , T_i e T_d sintonizados através do método de Z-N em malha aberta (ZNMA) $\{K_p; T_i; T_d\} = \{13,46; 56,0; 14,0\}$. Utilizando a relação $K_i = \frac{K_p}{T_i}$ e $K_d = K_p(T_d)$, os ganhos do controlador $\{K_p; K_i; K_d\} = \{13,46; 0,24; 188,44\}$.

Para se obter os valores dos ganhos do PID, pelo método de otimização de Z-A, esta disponibilizado no Anexo A, um código em MatLab chamado “*optPID*”, onde é possível determinar os parâmetros de um controlador PI ou PID com base no referido critério de

otimização de Z-A. Neste mesmo anexo encontram-se os cálculos efectuados para a obtenção dos ganhos do controlador para o exemplo em questão.

Utilizando os códigos apresentados no Anexo A, tem-se que $\{K_p; T_i; T_d\} = \{10,3614; 226,8202; 11,5049\}$.

Utilizando novamente a relação $K_i = \frac{K_p}{T_i}$ e $K_d = K_p(T_d)$, tem-se que $\{K_p; K_i; K_d\} = \{10,3614; 0,0456; 119,2068\}$.

Os parâmetros que foram utilizados para a configuração da população inicial do AG são apresentados na Tabela 26.

Tabela 26 Parâmetros de inicialização da população do AG para o exemplo 4

Parâmetros de inicialização do AG	
Tamanho da População	200
Número de Gerações	140
População Inicial	[13,86; 0,24; 188,44]
Limite da População Inicial	[0; 400]
Limite dos valores de $[K_p; K_i; K_d]$	lb = [0; 0; 0]
	ub = [100; 30; 400]

Tal como para os exemplos anteriores, os limites inferiores e superiores para a população foram especificados tendo em consideração os ganhos do controlador obtidos pelo método de Z-N. Estes limites representam a região do espaço de busca do AG, e por ser demasiado grande observa-se que o AG apresenta um elevado tempo computacional para explorar este espaço de pesquisa.

Os ganhos do controlador PID, bem como os valores de t_r , t_s e $MP\%$, foram agrupados na Tabela 27, de forma a avaliar o desempenho do sistema exposto para o quarto exemplo.

Tabela 27 Ganhos do controlador PID e especificações temporais para o exemplo 4

Método	K_p	K_i	K_d	t_r	t_s	Mp%	
Z-N	13,4600	0,2400	188,4400	7,1	252	90,20	
Z-A	10,3614	0,0456	119,2068	16,9	218	20,00	
AGH1	IAE	7,8813	0,0454	137,4000	20,2	161	7,42
	ITAE	8,8366	0,0511	92,0448	22,7	109	3,53
	ISE	9,5713	0,0671	153,6155	14,5	211	27,40
	ITSE	9,4262	0,0637	182,4041	11,7	259	34,50
	MSE	9,5640	0,0671	153,5996	14,5	211	27,70
AGH2	IAE	9,8750	0,0566	120,5312	17,4	116	20,70
	ITAE	8,8328	0,0510	92,9459	22,6	109	3,64
	ISE	9,5736	0,0671	154,4030	14,4	211	28,40
	ITSE	9,6250	0,0586	132,1953	16,7	136	21,40
	MSE	9,5662	0,0671	153,5947	14,5	211	27,30

O melhor compromisso que se obteve, ocorreu ao utilizar o AGH1, com o índice de desempenho ITAE, cujos resultados revelam um baixo valor de sobreelongação e o menor tempo de estabelecimento perante aos demais índices. Para o método de otimização de Z-A, o sistema apresenta um comportamento inferior quando comparamos com os melhores resultados obtidos pelos AGHs, revelando um alto valor para o tempo de estabelecimento e sobreelongação.

A Figura 33 apresenta a resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH1, cujo índice de desempenho que se encontra especificado é o ITAE, face ao melhor compromisso dos resultados obtidos para os parâmetros descritos na Tabela 27.

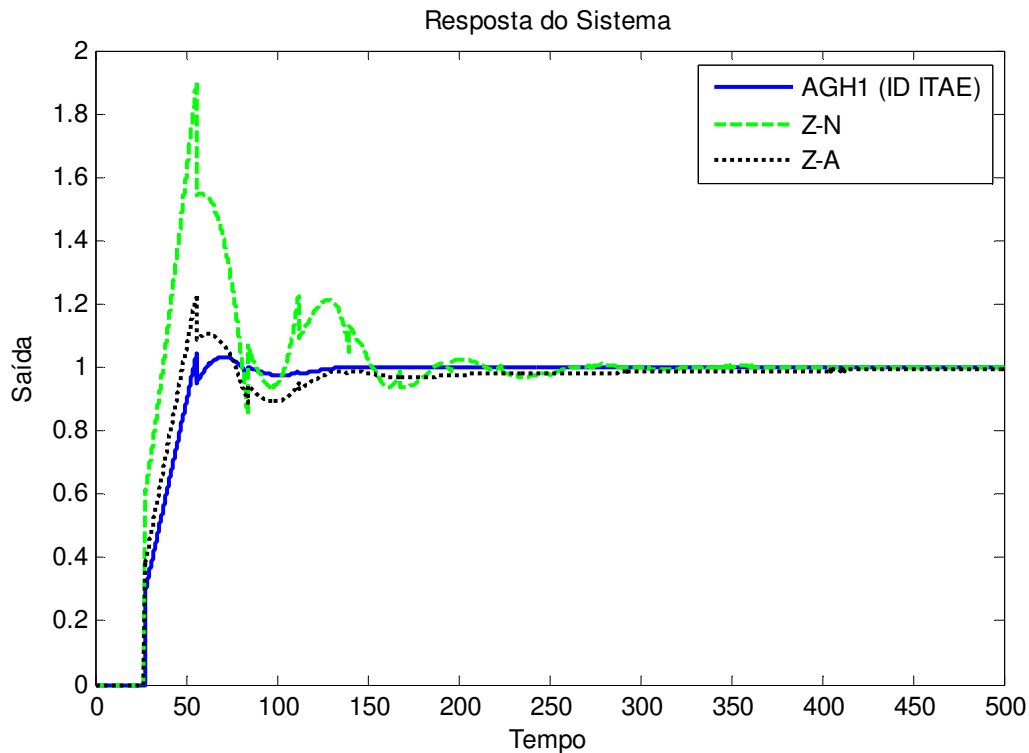


Figura 33 Resposta do sistema a uma entrada em degrau unitário para os métodos de Z-N, Z-A e AGH1 com índice de desempenho ITAE, para o exemplo 4

5.4. COMENTÁRIOS

No capítulo anterior, exemplificou-se a eficácia do AG quando comparado com os métodos clássicos de controlo e a facilidade de conjugá-lo com outras funções de optimização de forma a melhorar o seu desempenho.

Neste capítulo, teve-se a intenção de avaliar a robustez do AG, para sistemas que apresentassem características comportamentais que dificultassem a sua sintonia. Tais sistemas representaram um grande contributo para este trabalho, visto que os mesmos foram objecto de estudos por outros autores, e na posse dos resultados obtidos pode-se fazer uma comparação entre o desempenho do controlador utilizando o AG e outros métodos de optimização.

Como era de esperar, todos os métodos que foram descritos neste trabalho, apresentaram melhores resultados quando comparados com os resultados obtidos utilizando o método de sintonia de Z-N. Entretanto esta comparação não é necessariamente justa, visto que a parametrização destes métodos requerem um prévio conhecimento do modelo do processo, enquanto a heurística de Z-N não necessita deste prévio conhecimento. A justificativa para

se utilizar o método de Z-N foi aproveitar a facilidade que se tem em obter resultados iniciais através desta heurística, podendo depois utiliza-los como ponto de partida para melhorar o desempenho dos sistemas que estejam a ser analisados.

Assim, o estudo realizado leva-nos a enfatizar que não é possível descartar os métodos tradicionais de sintonia, como por exemplo as heurísticas de Z-N. Entretanto o responsável pelo controlo pode utilizar os AGs como alternativa para a sintonia de processos em que os resultados obtidos com os métodos tradicionais não são satisfatórios.

Outro facto que pode ser comprovado através da realização deste trabalho é que os AGs são eficientes na procura global, mas apresentam deficiências na convergência da solução final, o que leva a conjugar este método com outras heurísticas, formando então as estruturas híbridas. Esta metodologia, apesar de obter bons resultados e de existir inúmeras referências bibliográficas sobre o assunto, apresenta inúmeras questões que se mantêm em aberto e as quais tem sido motivo de grandes estudos no meio académico.

Não existe um método de sintonia que seja superior a todos os outros, e que se adapte perfeitamente a todos os tipos de problemas, por outro lado, afirmar que os AGs ou a computação evolutiva é a solução para todos os tipos de problemas, é uma insensatez. O que se pode afirmar, é que os AGs são uma ferramenta de optimização muito satisfatória para a maioria de problemas deste tipo, e para todas as solicitações de um sistema, nomeadamente no que concerne à resposta transitória do sistema.

5.5. INTERFACE GRÁFICA PARA SINTONIA DE CONTROLADORES COM ALGORITMOS GENÉTICOS

O objectivo desta secção é descrever o ambiente computacional que foi desenvolvido utilizando a ferramenta GUI (*Graphic User Interface*) em MatLab, para a sintonia de controladores com AGs. Esta interface, designada de LIR, tem como objectivo utilizar os sistemas que foram objecto de estudo desta dissertação, bem como a possibilidade de criar novos sistemas, de forma a facilitar ao utilizador a compreensão dos AGs, e consequentemente a sua aplicação na área do controlo.

A interface desenvolvida permite ao utilizador configurar uma série de parâmetros dos AGs, bem como ter acesso ao código desenvolvido para as funções objectivo e ao ambiente Simulink. Como complemento a esta interface criou-se um módulo, onde é

possível determinar os parâmetros do controlador PID utilizando o método de Zhuang e Atherton, dado que este método de otimização foi uma mais-valia para o trabalho desenvolvido.

Assim, espera-se que esta interface possa contribuir para o ensino, e de uma forma fácil e intuitiva despertar um maior interesse daqueles que venham a aprofundar os conhecimentos não só sobre os AGs, mas também sobre a sintonia de controladores.

5.5.1. INTRODUÇÃO

As interfaces computacionais para o ensino de engenharia vêm sendo cada vez mais aplicadas como uma forma amigável de aprendizagem. Este recurso é capaz de substituir vários equipamentos electrónicos e trazer grandes vantagens com mais versatilidade para captar o interesse do aluno. O desenvolvimento de um ambiente iterativo de ensino com um propósito específico foi o que levou ao desenvolvimento desta interface no âmbito desta dissertação.

Apesar do MatLab fornecer um conjunto de funções para a utilização de AGs, esta aplicação foi desenvolvida para evitar o seu uso, e assim fornecer ao utilizador uma aplicação mais intuitiva e didáctica das funções que foram utilizadas no desenvolvimento deste trabalho.

As aplicações desta interface, referentes à sintonia dos controladores com AGs, são divididas em cinco partes principais:

1. Parametização do AG.
2. Identificação do algoritmo a ser utilizado, descrito como AG, AGH1 e AGH2, fazendo referência aos algoritmos descritos no capítulo anterior.
3. Sintonia do Controlador PID com base na minimização dos índices de desempenho IAE, ITAE, ISE, ITSE e MSE.
4. Visualização dos resultados, nomeadamente os valores de K_p , K_i e K_d e o valor do erro que corresponde ao critério a ser minimizado.
5. Visualização gráfica da resposta temporal do sistema.

5.5.2. VISÃO GERAL DO AMBIENTE

O programa é dividido em sete módulos, sendo que o primeiro módulo é a janela principal do ambiente, onde o utilizador tem acesso a um conjunto de informações que o auxiliam na utilização da interface, bem como um menu, onde o mesmo pode direccionar a aplicação aos módulos seguintes.

A Figura 34 apresenta o primeiro módulo da interface desenvolvida. Designado como “LIR” este módulo é a janela que dá acesso às demais aplicações, bem como a um manual de utilização que foi escrito de forma a facilitar o uso da mesma.

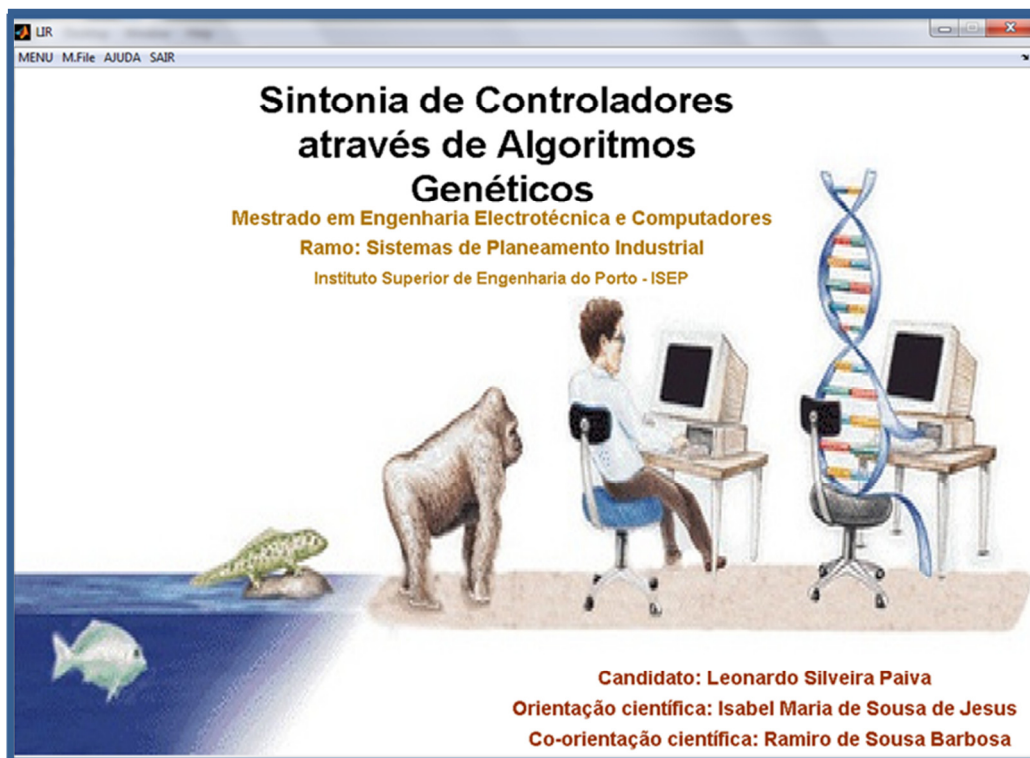


Figura 34 Menu principal da interface gráfica

Conforme pode ser visualizado na Figura 35, por meio do primeiro módulo, o utilizador pode ter acesso às outras janelas da interface, onde lhe é possível realizar as simulações que pretender. Com o intuito de possibilitar a visualização dos códigos criados para as funções de aptidão, é possível aceder a estes códigos através do menu “M.File”, que dá acesso aos códigos desenvolvidos para os cinco índices de desempenho.



Figura 35 Menu dos módulos de simulação e código M.File dos índices de desempenho

Os módulos denominados “Primeiro_Sistema”, “Segundo_Sistema”, “Terceiro_Sistema” e “Quarto_Sistema”, correspondem aos exemplos que foram utilizados na secção anterior. Estes módulos foram criados com a intenção de permitir ao utilizador realizar simulações para os sistemas estudados neste trabalho, com configurações diferentes das que foram utilizadas, e desta forma avaliar a influência que a modificação de um parâmetro, ou um conjunto de parâmetros podem ocasionar no desempenho do AG.

A janela denominada “Novo_Sistema” corresponde a um módulo pré-programado, que possibilita a simulação de um novo sistema com base na criação de uma estrutura em blocos a ser configurada no ambiente Simulink. Este módulo foi desenvolvido com o objectivo de expandir o uso desta interface a outras aplicações, com diferentes tipos de sistemas para além daqueles que foram utilizados neste trabalho.

Para complementar a aplicabilidade deste ambiente computacional foi criado um módulo denominado “Zhuang e Atherton”, onde é possível determinar os parâmetros do controlador PID, com base no critério de optimização de Z-A. A seguir serão detalhados todos os módulos da interface, dando destaque ao modo de utilização e à visualização dos resultados obtidos.

5.5.3. FUNCIONALIDADE DA INTERFACE GRÁFICA

Pela análise da Figura 36, é feita a descrição do funcionamento da interface referente aos módulos que correspondem à sintonia do controlador PID através do uso de algoritmos genéticos.

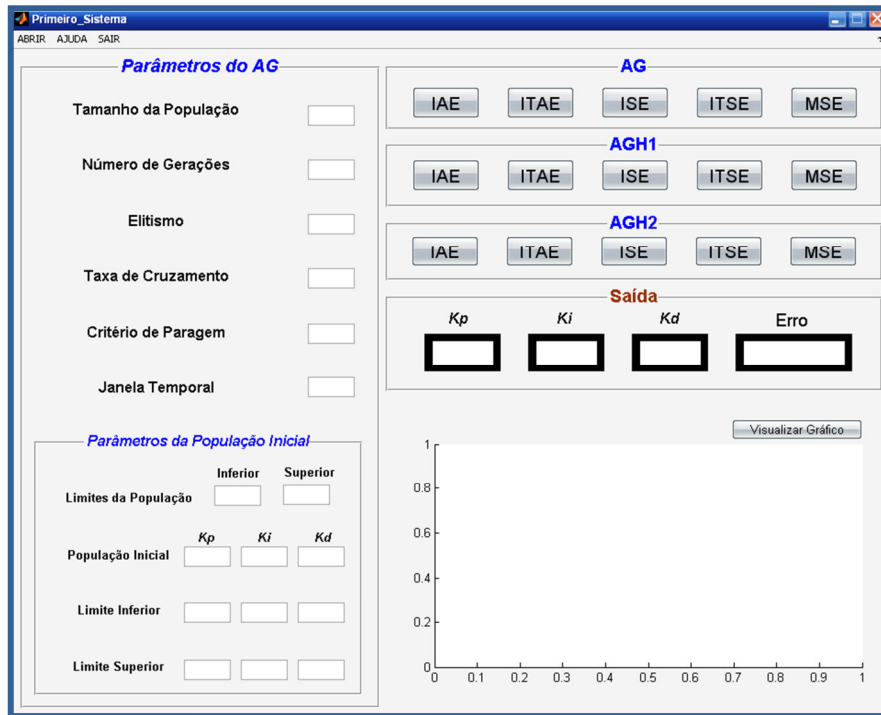


Figura 36 Módulo de aplicação da interface gráfica referente ao uso de algoritmos genéticos para a sintonia de controladores

Estes módulos dividem-se entre entrada de dados, escolha do algoritmo para simular o sistema, e saída dos resultados. Os parâmetros para configuração do AG serão omissos neste momento visto que os mesmos foram detalhados no capítulo anterior. Pelo mesmo motivo será omissa a descrição dos algoritmos descritos como AG, AGH1 e AGH2.

Além destas aplicações, foi criado um menu de opções onde o utilizador pode ter acesso ao ambiente Simulink para visualizar o diagrama de blocos criado para o sistema em questão, bem como o código criado para a função *plot*. Esta opção permite ao utilizador incluir novos sistemas no ambiente Simulink, para posterior visualização da sua resposta em comparação com a resposta do sistema obtida com os AGs.

Entretanto, caso o utilizador desta interface não esteja ciente do significado destes parâmetros, e de como eles devem ser configurados, foi criado um menu de ajuda que

fornece informação sobre o sistema em questão, bem como a forma que o AG deve ser configurado. Tais opções podem ser visualizadas na Figura 37.

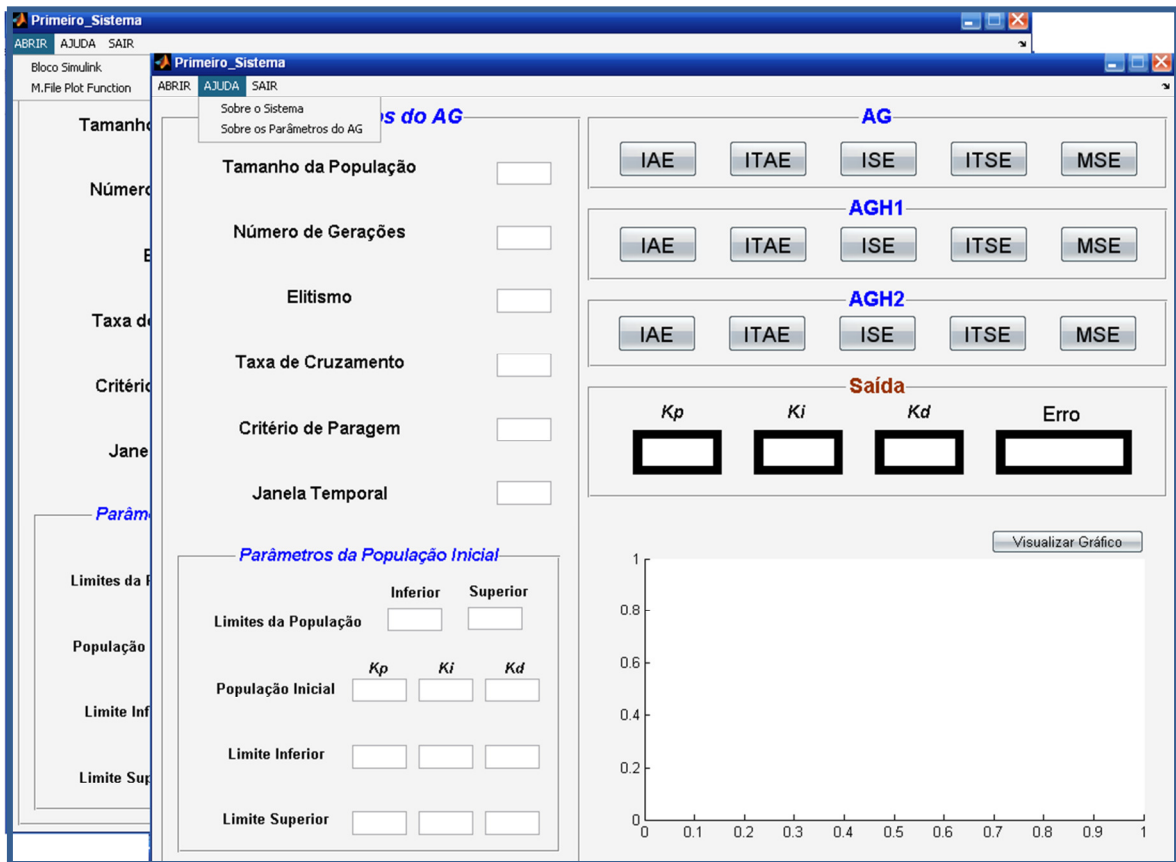


Figura 37 Menu de opções para os módulos da interface gráfica

Para a entrada de dados, todos os parâmetros que estão citados para a configuração do AG devem ser preenchidos para o correcto funcionamento da interface. Caso algum destes parâmetros tenha ficado em branco, ou então não tenha sido preenchido correctamente, é emitida uma mensagem de erro no *prompt* do MatLab de forma a alertar o utilizador para a correcção do problema que tenha causado a falha.

A Figura 38 apresenta um exemplo de utilização da interface, onde se pode visualizar os ganhos do controlador PID dado pelo índice IAE do AG. A resposta do sistema pode ser visualizada logo após o AG ter executado a simulação. Em conjunto com esta resposta é também visualizada as respostas obtidas pelas aplicações dos métodos de Z-N e Z-A.

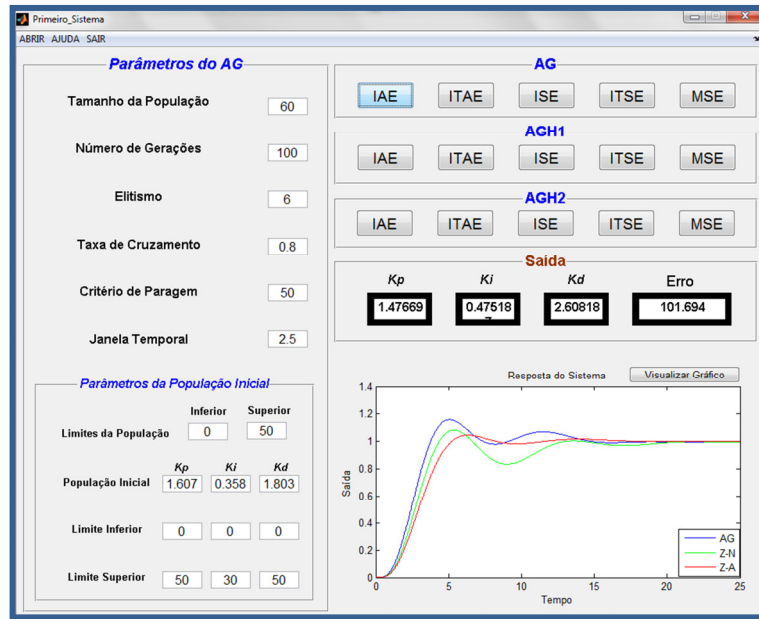


Figura 38 Exemplo de funcionamento da interface

Ao inicializar-se a simulação, é gerado um conjunto de gráficos iterativos, que correspondem a um conjunto de opções disponíveis em MatLab para visualizar o desempenho do AG. Estes gráficos são apresentados na Figura 39.

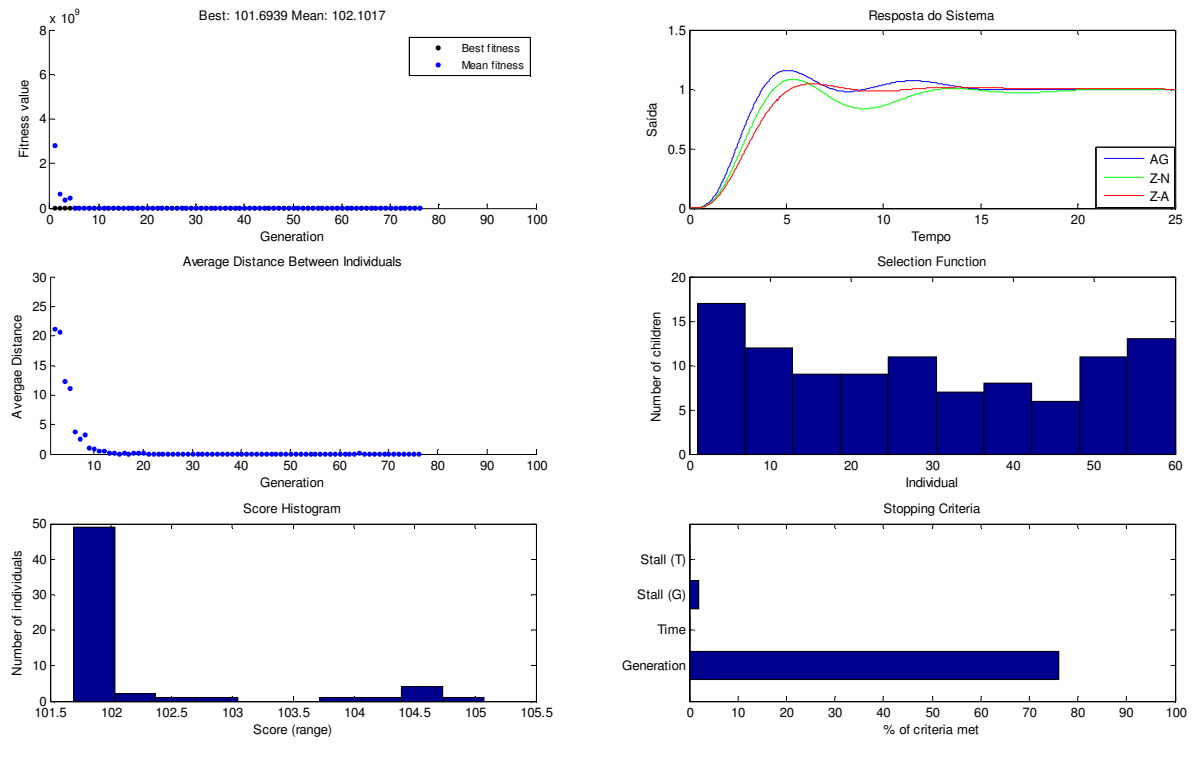


Figura 39 Opções gráficas para visualizar o desempenho do AG

Os gráficos apresentados na Figura 39 referem-se a:

- **gplotbestf**: Apresenta tanto a média como a melhor *fitness* da população gerada pelo AG em cada geração.
- **Resposta do Sistema**: Apresenta a resposta do sistema para os ganhos do controlador encontrados pelo AG em comparação com a resposta do mesmo sistema para os ganhos obtidos por Z-N e Z-A.
- **gplotdistance**: Apresenta a distância média dos indivíduos de cada geração.
- **gplotselection**: Apresenta em forma de um histograma a parcela de pais que estão a contribuir para cada geração. Este gráfico fornece-nos uma melhor percepção da diversidade da população que foi gerada pelo AG. Ou seja, se apenas poucos dos pais contribuíram para o desenvolvimento das novas gerações, significa que grande parte da população está a partilhar o mesmo código genético. Caso isto aconteça, pode ser necessário alterar alguns parâmetros do AG para obter uma maior diversidade.
- **gplotscorediversity**: Apresenta em forma de histograma a pontuação de todos os indivíduos que foram gerados em cada geração.
- **gplotstopping**: Apresenta em termos de percentagem, os critérios de paragem do AG.

Para o módulo correspondente a “Novo_Sistema”, basta que o utilizador aceda ao bloco pré-programado em ambiente Simulink, e especifique a função de transferência que descreve o processo que deseja simular. Caso seja necessário podem também ser incluídas outras especificações no sistema, como por exemplo tempo de atraso, saturação ou algum tipo de perturbação. Entretanto é necessário que neste módulo o utilizador tenha o mínimo de conhecimento sobre o ambiente Simulink, para que seja possível representar correctamente em diagrama de blocos o sistema que deseja simular com o AG.

A Figura 40 apresenta uma visão geral deste módulo de utilização que se intitula “Novo_Sistema”.

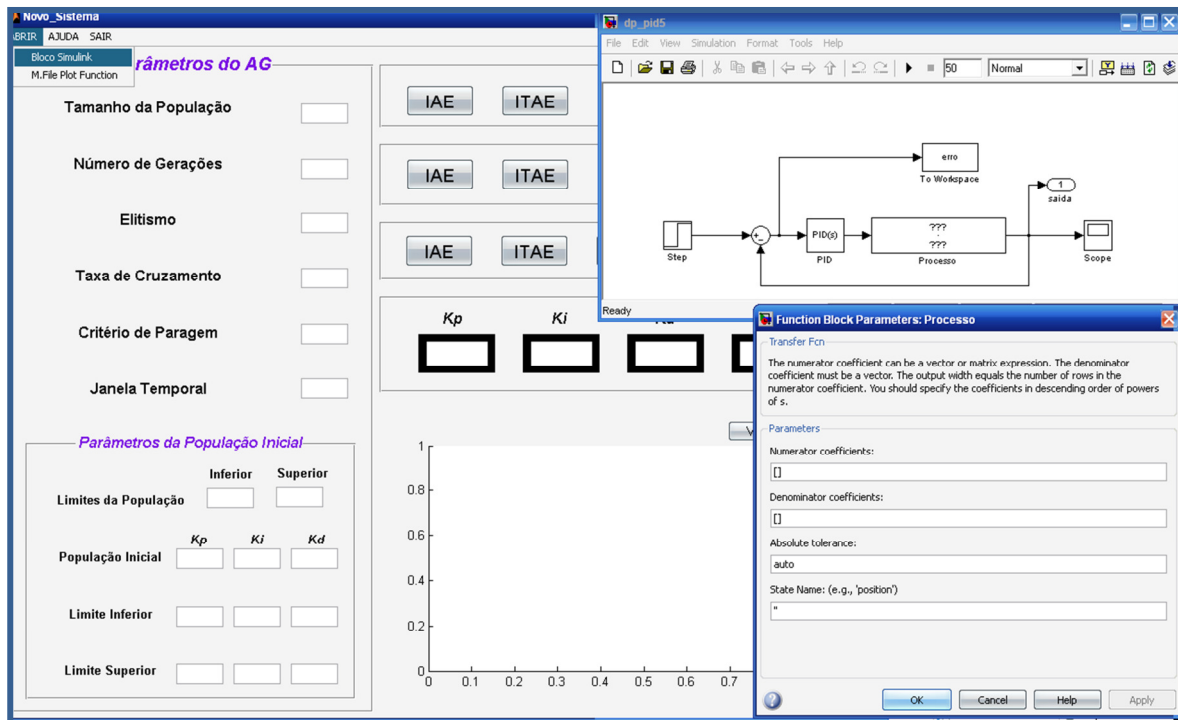


Figura 40 Módulo de configuração para um novo sistema

Para finalizar os aplicativos da interface desenvolvida, foi criado um módulo denominado de Zhuang e Atherton, visto que este critério de otimização foi um bom “ponto de partida” para validação dos algoritmos estudados na secção anterior. Na posse de uma função de transferência de um processo de primeira ordem com tempo de atraso, basta que o utilizador especifique os parâmetros desta função de transferência, o tipo de controlador a ser utilizado e o tipo de sintonia.

Como se trata de um assunto que já foi objecto de estudo na secção anterior, o mesmo será omissso neste momento. Entretanto para o utilizador da interface, foi criado um menu de ajuda que poderá auxiliá-lo a compreender o critério de otimização de Z-A, e como utilizar correctamente o módulo da interface destinado a este propósito. O módulo da interface correspondente ao método de optimização de Z-A pode ter visualizado na Figura 41.

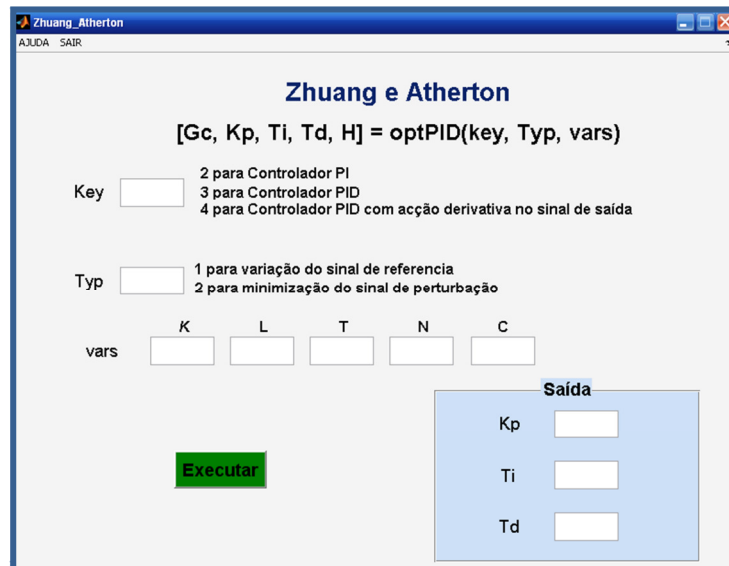


Figura 41 Módulo da interface referente ao critério de optimização de Zhuang e Atherton

5.5.4. EXEMPLOS ILUSTRATIVOS DE UTILIZAÇÃO DA INTERFACE LIR

Neste momento será apresentado um conjunto de simulações que foram efectuadas para demonstrar a funcionalidade da interface gráfica desenvolvida. Foram feitas simulações com diferentes configurações para o AG, onde os valores adoptados assim como a visualização dos resultados e o gráfico que representa a resposta do sistema para os parâmetros obtidos através dos AGs, Z-N e Z-A, podem ser visualizados nas imagens que se seguem.

A Figura 42 apresenta uma simulação realizada para o primeiro sistema, onde se utilizou o AGH2, com o intuito de obter o melhor desempenho para o controlador com base na minimização do ID ITAE.

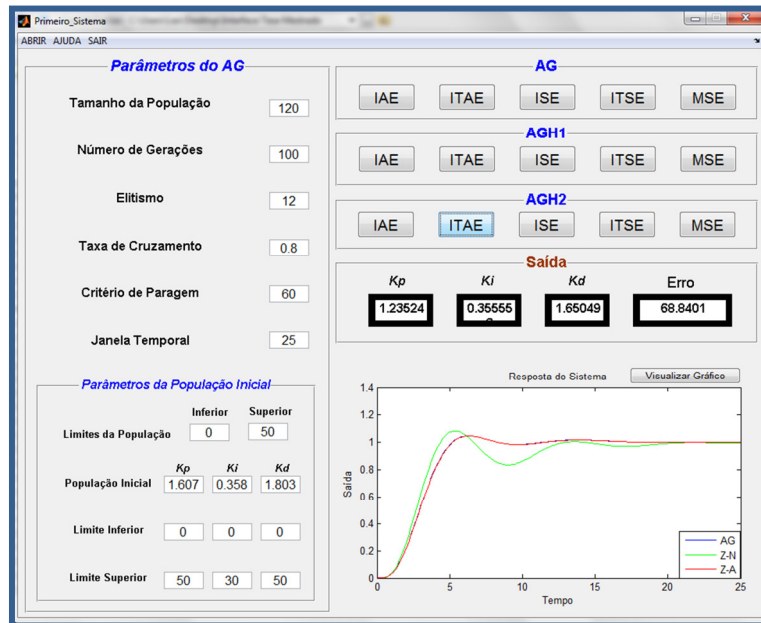


Figura 42 Módulo desenvolvido para a simulação do primeiro sistema

A Figura 43 apresenta uma simulação para o sistema citado na secção anterior como exemplo 2, onde foi utilizado o AGH2 para sintonizar o controlador PID através da minimização do ID IAE. Observa-se também que, apesar das janelas serem idênticas para todos os módulos referentes à sintonia do controlador PID com AG, adoptou-se cores diferentes como forma de diferenciação de um módulo perante aos demais.

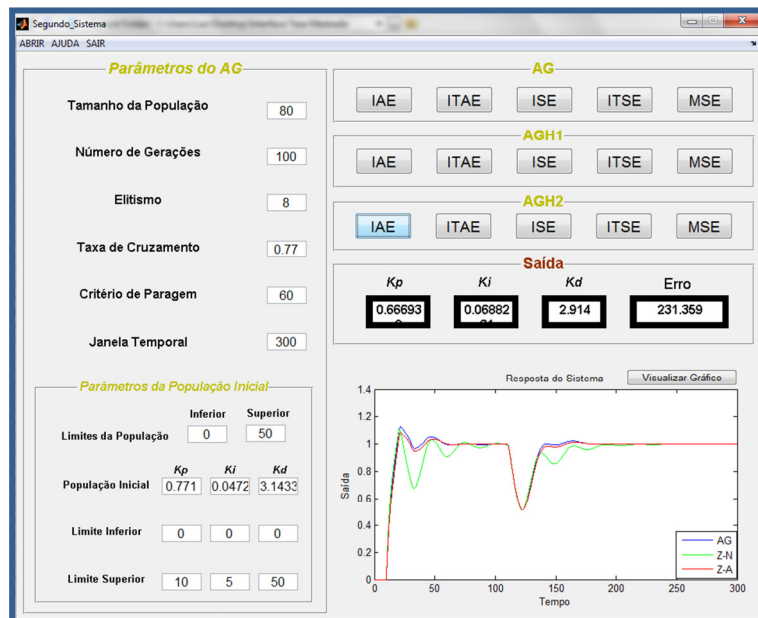


Figura 43 Módulo desenvolvido para a simulação do segundo sistema

A Figura 44 apresenta o módulo desenvolvido para a sintonia do controlador PID destinado ao terceiro sistema. Nesta simulação adoptou-se o AGH2 como método de optimização, através da minimização do ID IAE.

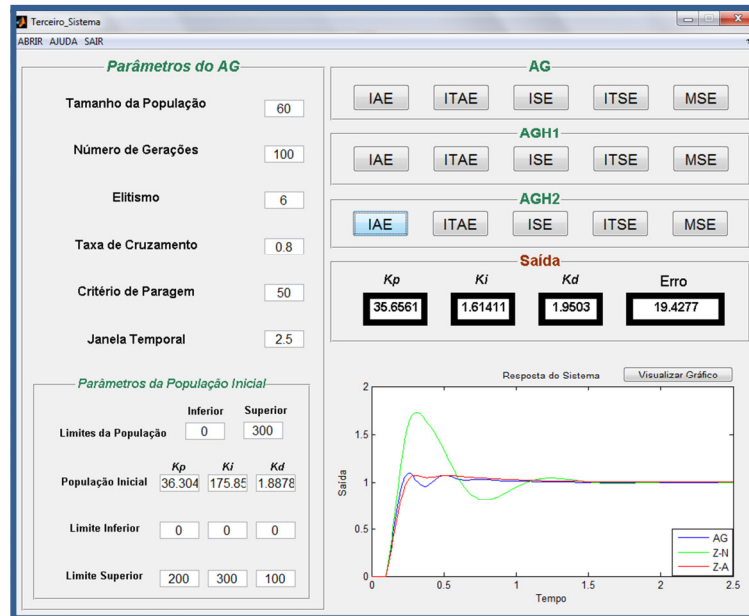


Figura 44 Módulo desenvolvido para a simulação do terceiro sistema

A Figura 45 apresenta o módulo da interface gráfica criado para simular a sintonia do controlador PID com AG para o quarto sistema, que neste caso refere-se ao sistema de difusão de calor apresentado na secção anterior. Nesta simulação utilizou-se o AG para a sintonia do controlador, através da minimização do ID MSE.

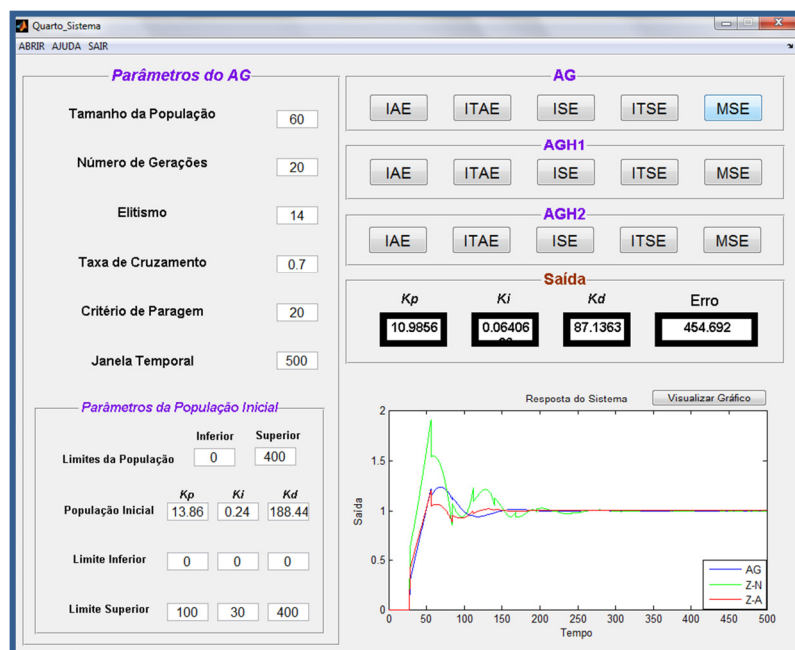


Figura 45 Módulo desenvolvido para a simulação do quarto sistema

Finalmente a Figura 46 apresenta o módulo da interface destinado a obter os parâmetros do controlador PID através do método de optimização de Zhuang e Atherton. Como exemplo ilustrativo para este módulo utilizou-se a equação (29) que faz referência à função de transferência do sistema de difusão de calor, que foi objecto de estudo para o quarto exemplo da secção anterior.

No Anexo A desta tese de mestrado é apresentado o algoritmo desenvolvido em MatLab para determinar os parâmetros do controlador PID, pelo método de optimização de Z-A. Como forma de aumentar a aplicabilidade da interface desenvolvida, criou-se um módulo destinado ao cálculo dos parâmetros do controlador PID através do método de Z-A. Para esta aplicação é necessário introduzir os dados referentes a uma função de transferência do tipo FOPDT, como é apresentado na Figura 46.

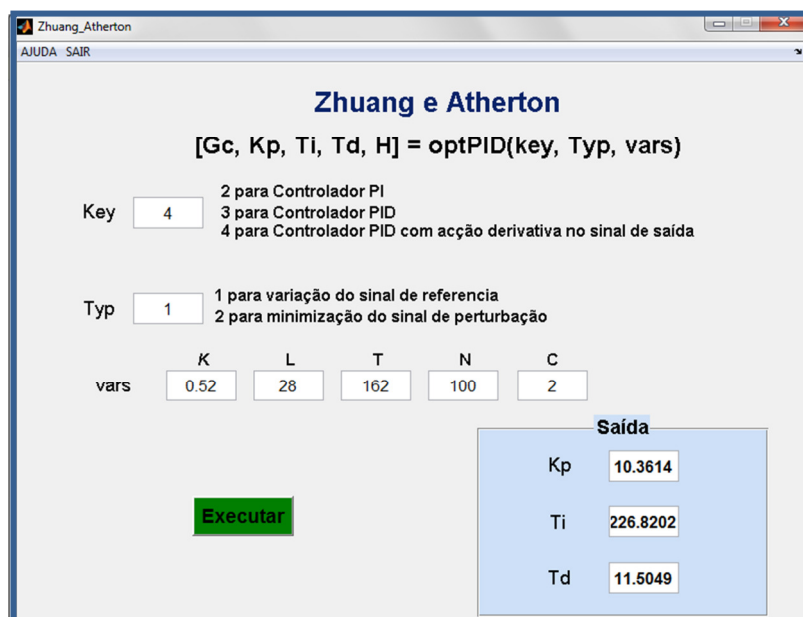


Figura 46 Módulo desenvolvido para o método de optimização de Z-A

5.5.5. CONCLUSÃO

Em linhas gerais pode-se concluir que a interface gráfica criada no âmbito desta tese de mestrado se mostrou de grande utilidade para as finalidades que foram propostas. Numa primeira abordagem teve-se a intenção de criar um ambiente computacional que pudesse atrair a atenção de possíveis interessados neste assunto e complementar os conhecimentos que venham a ser adquiridos sobre os AGs e a sintonia dos controladores.

Sem a necessidade de se ter um conhecimento avançado sobre as funcionalidades da ferramenta MatLab, assim como do ambiente Simulink ou da *toolbox* de optimização para AGs, o utilizador apenas necessita de fornecer os valores para os parâmetros do AG e escolher o índice a ser optimizado. Neste sentido, esta interface mostra-se de grande utilidade, graças a um conjunto de informações que se encontram disponíveis para auxiliar o utilizador a fazer uma correcta entrada de dados, e escolher o critério de optimização a ser minimizado.

Esta interface também pode ser de grande utilidade para usuários que já tenham algum conhecimento sobre o assunto, sendo que estes podem aceder aos códigos correspondentes aos IDs, assim como a possibilidade de criar novos sistemas para além daqueles que foram utilizados neste trabalho.

Apesar deste ambiente computacional ter sido criado para um propósito específico, que era simular os sistemas que foram objecto de estudos no âmbito desta dissertação, é sempre possível que o mesmo possa ser modificado de forma a atender a outros interesses, ou até mesmo sofrer melhorias na sua concepção. A interface desenvolvida serviu não somente como agente factível dos conhecimentos adquiridos ao longo das pesquisas efectuadas para a realização deste trabalho, como também servirá de elemento enriquecedor a ser explorado em aulas relacionadas com os sistemas de controlo, ou aplicações dos AGs e até mesmo para futuras dissertações que venham a ser desenvolvidas nesta área de pesquisa.

6. CONCLUSÕES E PERSPECTIVAS DE TRABALHOS FUTUROS

Neste capítulo são apresentados as principais conclusões do trabalho, suas contribuições em relação ao tema que foi abordado e linhas de raciocínio para futuros projectos que possam vir a ser desenvolvidos nesta área. Na secção 6.1 é apresentado uma discussão sobre os resultados alcançados, a metodologia proposta e as ferramentas de desenvolvimento. A secção 6.2 aponta as contribuições desta tese para as linhas de pesquisas em engenharia de controlo e sobre os algoritmos genéticos. Na secção 6.3 apresenta sugestões para trabalhos futuros bem como uma visão pessoal do autor sobre o trabalho que foi desenvolvido.

6.1. ANÁLISE DOS RESULTADOS OBTIDOS

Face aos resultados obtidos nos capítulos 4 e 5, pode-se dizer que o AG desenvolvido neste trabalho mostrou ser uma valiosa ferramenta no que respeita à sintonia de controladores PID. Em relação aos métodos clássicos de optimização, estes são importantes para nos fornecerem um ponto inicial, pelo qual podemos nos guiar para estipularmos os valores

para os parâmetros do AG. Entretanto, pode ser observado que a correcta parametrização do AG envolve muitos passos através do método de tentativa e erro, e sem a garantia de que a resposta encontrada apresenta o melhor resultado para o problema.

A metodologia proposta para este trabalho, bem como a revisão da literatura sobre o assunto, demonstrou que os AGs possuem grande eficácia e aplicabilidade na área do controlo. Cabe ainda ressaltar que os resultados obtidos foram comparados com outros métodos de optimização, chegando a resultados ora idênticos, ora superiores às técnicas de optimização existentes.

Foram também usadas outras funções de optimização para a sintonia do controlador PID, nomeadamente as funções do MatLab (*fmincon* e *pattersearch*) e o método de optimização de Zhuang e Atherton. Quanto às funções do MatLab, pode-se dizer que são boas ferramentas, obtendo-se boas respostas com um razoável tempo computacional. Entretanto, essas mesmas funções são muito dependentes do ponto dado como “semente”, visto que a resposta para o sistema era tanto melhor quanto mais próximo este ponto estivesse da melhor solução.

Utilizar como modelo de comparação o método de sintonia de Z-A, foi de grande valor para o trabalho desenvolvido, visto que esta metodologia é uma referência quando se trata sobre o estudo de sintonia de controladores, e cujos resultados encontrados demonstram valores aceitáveis para parâmetros da resposta do sistema no domínio dos tempos.

No que respeita ao tempo computacional, não se pode dizer que os AGs são a melhor alternativa. Face aos valores atribuídos para o número de gerações e tamanho da população, o tempo computacional pode ser longo, e este pode ser um factor prejudicial ao desempenho da malha de controlo do processo.

Neste trabalho, o critério de sintonia utilizado foi baseado na minimização do erro e vários foram os índices de desempenho utilizados para este propósito. A utilização de vários ID em detrimento de utilizarmos apenas um ID como base para os cálculos foi de grande importância para este trabalho, visto que o melhor compromisso para os sistemas que foram especificados, não era obtido com o mesmo ID. Segundo Ogata [Ogata, 2003], o melhor índice a ser utilizado é o ITAE. Entretanto, neste trabalho, foram obtidos melhores resultados com outros índices de desempenho a saber IAE e MSE.

A utilização das funções de optimização do MatLab, em conjunto com o AG, formando então a estrutura híbrida que foi utilizada em todos os sistemas do capítulo 5, demonstrou ser de grande utilidade. Apesar destas funções de optimização obterem bons resultados para o desempenho dos sistemas que foram analisados, estas são muito dependentes no ponto dado como semente, já que se este ponto não for bem especificado poderá a solução convergir para resultados de baixa qualidade. Neste sentido, utilizar os AGs como a primeira heurística de procura global, garante uma “varredura” mais completa sobre o espaço de soluções, cuja solução final encontrada pelo AG é um bom ponto de partida para uma função de busca local, garantindo que o resultado final seja o mais favorável.

O ambiente de simulação utilizado, MatLab, também se mostrou uma importante ferramenta para os propósitos deste trabalho, devido à facilidade de implementação do AG, à disponibilidade de outras funções de optimização para a sintonia dos controladores, e o ambiente Simulink, onde foi possível criar digramas de blocos e utiliza-los na função objectivo. Outra grande utilidade da ferramenta MatLab, deu-se com o desenvolvimento da interface gráfica para os propósitos desta dissertação.

Em termos pessoais, este projecto possibilitou que eu pudesse aumentar os meus conhecimentos sobre a engenharia de controlo e os algoritmos genéticos. Vários métodos óptimos de sintonia foram objectos de estudo para a elaboração deste trabalho, que até então eram por mim desconhecidos. A nível de licenciatura ficamos muito confortáveis em utilizar as técnicas de Z-N, mas a partir deste trabalho posso afirmar que outras técnicas de sintonia fazem parte dos meus conhecimentos académicos.

Juntamente com o conhecimento sobre a engenharia de controlo, surgiram então os algoritmos genéticos e com ela toda a parte da computação evolutiva que despertou em mim uma verdadeira admiração sobre esta metodologia. Trata-se de um assunto que surgiu há já algum tempo e que vai continuar a ser objecto de estudo face às inúmeras vantagens apresentadas. Neste sentido, ao realizar este trabalho foi-me proporcionado um breve, mas valioso conhecimento, sobre um assunto actual e ao mesmo tempo promissor.

6.2. CONTRIBUIÇÕES DO TRABALHO

As contribuições que surgiram deste trabalho podem ser enumeradas em:

- Estudo de vários índices de desempenho com base na minimização do sinal de erro, para sintonia de controladores.
- Validação dos resultados do AG, em comparação com outras técnicas de sintonia em detrimento da comparação apenas com as técnicas clássicas de controlo, como as de Ziegler-Nichols e Cohen-Coon.
- Implementação de uma estrutura híbrida, com a conjugação de heurísticas para o refinamento das soluções do AG, utilizando o método “hill climbing” (função *fmincon*) e conjunto de pontos padrão (função *patternsearch*).
- Estudo comparativo da complexidade de convergência dos AGs, em função do tamanho da população face ao uso de vários índices de desempenho.
- Disponibilidade de uma interface gráfica para o ensino/aprendizagem da aplicabilidade dos AGs na área do controlo.

6.3. PERSPECTIVAS PARA TRABALHOS FUTUROS

Após a realização deste trabalho, pode-se então discutir algumas linhas de pesquisas para trabalhos futuros, de entre as quais se enumeram:

- Utilizar um conjunto de restrições em simultâneo com a optimização de um critério de desempenho, para sistemas em que o efeito das perturbações ou variação da referência de entrada representam situações que limitam o desempenho da malha de controlo.
- Utilização de outras técnicas evolutivas, como por exemplo o uso do PSO (*Particle Swarm Optimizacion*) em comparação com os resultados obtidos pelos algoritmos genéticos em sistemas que representam processos industriais de grande complexidade.
- Aplicação dos algoritmos genéticos para a sintonia de controladores de sistemas multiobjectivo utilizando como referência o valor da sobreelongação ou o tempo de estabelecimento. Este estudo pode ser feito utilizando as funções objectivo que foram

desenvolvidas neste trabalho acrescentando restrições referentes ao desempenho destes parâmetros.

- Estudo da viabilidade de se aplicar a sintonia de controladores com algoritmos genéticos em processos *on-line*, tendo como factor preponderante o tempo computacional.
- Utilização de outras heurísticas de procura local, na estrutura híbrida, num ambiente de programação diferente do MatLab, dado que este limita as funções de optimização a serem conjugadas com os AGs.
- Estudar a viabilidade de outros parâmetros na configuração do AG, como por exemplo o uso de funções de penalidades. Ou até mesmo, o uso de regras determinísticas para a obtenção dos valores como taxa de cruzamento, tamanho da população, taxa de mutação e número de gerações.
- Desenvolver novas ferramentas de controlo na interface criada, de forma a permitir que esta tenha uma maior aplicabilidade em ambientes reais.

Referências Documentais

A. BARBERÁ, Walison — *Recognition of Human Faces through Genetic Algorithm applied in Reasons Áurea*. <http://www.fafibe.br/epeq/artigos/18-> . (Dezembro, 2009).

ASTROM, K. J.; T. HAGGLUND — *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, 1995.

BARBOSA, Ramiro — *SISCA: Sistema de Controlo Avançado* – Apontamentos da disciplina Sistema de Controlo Avançado ministradas pelo Eng^o. Ramiro Barbosa no Instituto Superior de Engenharia do Porto, primeiro semestre 2010.

BITTENCOURT, Guilherme — *Inteligência Computacional*. <http://www.das.ufsc.br/gia/softcomp/node11.html>. (Dezembro, 2009).

C.LUCAS, Diogo — *Algoritmos Genéticos: Uma Introdução* – Apóstila elaborada sob orientação do Prof. Luís Otávio Álvares para a disciplina de Inteligência Artificial na Universidade Federal do Rio Grande do Sul/Brasil 2002.

CAO, J. ; WU, H — *Teaching genetic algorithm using MatLab*. *Internacional journal engineering education*, Manchester, 1999.

COELLO, C. A., VAN VELDHUIZEN, D. A., LAMONT, G. A. — *Evolutionary Algorithms for Solving Multiobjective Problems*, Kluwer Academic Publishers. IEEE Swarm Intelligence Symposium, IEEE Service Center, Indianapolis, USA, 2003.

CVRD — *Relatório de estágio da Companhia Mineradora Vale do Rio Doce, MG/Brasil* 2006.

DARWIN, C. — *On the origin of species*. <http://www.hn.psu.edu/faculty/jmanis/darwin.htm>. (Novembro, 2009).

DAVIS, Lawrence — *Handbook of genetic algorithms*. Van Nostrand Reinhold, N.Y. 1992.

De MOURA OLIVEIRA, B. — *Modern Heuristics Review for PID Control Systems Optimization: A Teaching Experiment*. IEEE International Conference on Control and Automation, Budapeste, Hungria, 2005.

DIAS, Joana; CAPTIVO, M. Eugênia; CLIMACO, João — *A Hybrid Algorithm for Dynamic Location Problems* - Universidade de Coimbra e Universidade de Lisboa, 2005.

DUTTON, Ken; THOMPSON, Steve; BARRACLOUGH, Bill — *The Art of Control Engineering*. Prentice Hall International, 1997.

EIBEN, A. E., HINTERDING, R., MICHALEWICZ, Z. — *Parameter Control in Evolutionary Algorithms*, IEEE Transactions on Evolutionary Computation, 1999.

FONSECA, C. M., FLEMING, P. J. — *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization*, Genetic Algorithms: Proceedings of the Fifth International Conference, San Mateo, CA, 1993.

GOLDBERG, D. — *Genetic algorithm in search, optimization, and machine learning*, Addison - Wesley Publishing Company, Reading, Massachusetts, 1989.

GRIFFIN, Ian — *On-Line PID Controller Tuning Using Genetic Algorithms*, M.Eng in Electronic Systems oriented by Ms Jennifer Bruton, 2003.

HOLLAND, J. — *Adaptation in Natural and Artificial Systems*, Ann Arbor: Univ. of Michigan Press, 1975.

ISHIBUCHI, H., YOSHIDA, T., MURATA, T. — *Balance Between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling*, IEEE Transactions on Evolutionary Computation, 2003.

JESUS, Isabel S. — *Sistemas Dinâmicos de Ordem Fraccionária* – Dissertação de Doutorado em Engenharia Electrotécnica orientada pelo Prof. Doutor José António Tenreiro Machado, e apresentada na Universidade de Trás-os-Montes e Alto Douro na especialidade de Engenharia Electrotécnica em 2007.

KUO, Benjamin — *Automatic Control Systems*. Univ. of Illinois, Urbana-Champaign, 2003.

L. SOARES, Gustavo — *Algoritmos Genéticos: Estudo, Novas Técnicas e Aplicações* – Dissertação de Pós-Graduação em Engenharia Eléctrica – Centro de Pesquisa em

Engenharia Eléctrica orientada pelo Prof. João António de Vasconcelos e apresentada na Universidade Federal de Minas Gerais / Brasil, na especialidade de Engenharia Eléctrica em 1997.

LOURENÇO, João — *Sintonia de Controladores PID*. Escola Superior de Tecnologia, 1997.

MATHWORKS — *Building GUIs with MATLAB*, ver 5.0, 1997.

MatLab — *Genetic algorithm and direct search toolbox user's guide*, © COPYRIGHT 2004–2009 by The MathWorks, Inc, 2009.

MICHALEWICZ, Z.; HINTERDING, R.; MICHALEWICZ, M. — *Evolutionary Algorithms*, Chapter 2 in *Fuzzy Evolutionary Computation*, W. Pedrycz, Kluwer Academic, 1997.

MITCHELL, Melanie — *An Introduction to Genetic Algorithms*. A Bradford Book the MIT Press, London England, 1999.

MORELLI, Márcio — *Análise do uso de Algoritmos Genéticos na Otimização do Planejamento Mestre da produção*. Dissertação de Mestrado em Engenharia de Produção e Sistemas orientada pelo Prof. Guilherme Ernani Vieira e apresentada na Pontifícia Universidade Católica do Paraná/Brasil, na especialidade de Engenharia de Produção e Sistemas em 2006.

OCHI, Luiz — *Algoritmos Genéticos: Origem e Evolução*.
<http://www.sbmac.org.br/bol/bol-2/artigos/satoru/satoru.html#geneticos>. (Dezembro, 2009).

OGATA, Katsuhiko. — *Engenharia de Controle Moderno*. 4^o edição. São Paulo: Editora Prentice Hall, 2003.

REEVES, C. R. — *Modern Heuristic Techniques for Combinatorial Problems*. Blakwell Scientific Publications, 1993.

REEVES, C. R. — *Using Genetic Algorithms with Small Populations*. Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, 1993.

REIS, Cecília — *ALGEN: Algoritmos Genéticos* – Apontamentos da disciplina Algoritmos Genéticos ministradas pela Eng^a. Cecília Reis no Instituto Superior de Engenharia do Porto, primeiro semestre 2009.

RENSBURG, P. J. V., SHAW, I. S. & WHY J. D. V. *Adaptive PID – Control using a Genetic Algorithm*. IEEE Conference on Knowledge-Based Intelligent Electronic System, Australia, 1998.

RIBEIRO, Marco Antônio — *Controle de Processos, Teoria e Aplicações*. Tek Treinamento e Consultoria. Salvador, BA/Brasil, 2001.

SALAMANCA, Henry — *Análise, Sintonia e Experimentação de Controladores PID via Projectos GMV e GPC*. Dissertação de Mestrado em Engenharia Eléctrica orientada pelo Prof. Antonio Augusto Rodrigues Coelho e apresentada na Universidade Federal de Santa Catarina/Brasil, na especialidade de Engenharia Eléctrica em 2007.

SENAI/CST — *Fundamentos de Controlo de Processo*. Manual de Instrumentação elaborado para o Programa de Certificação de Pessoal e Manutenção coordenado por Evandro de Figueiredo Neto e Robson Santos Cardoso. ES/Brasil, 1999.

SIAM — “*Society for Industrial and Applied Mathematics*”. <http://www.siam.org>. (Agosto, 2010).

SILVA, Joao Manoel — *O Controlador PID*, <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node24.html>. (Janeiro, 2010).

SOLTEIRO PIRES, Eduardo — *Algoritmos Genéticos: Aplicação a Robótica*. Dissertação de Mestrado em Engenharia Electrotécnica e Computadores orientada pelo Eng. José António Tenreiro Machado e apresentada na Faculdade de Engenharia da Universidade do Porto, na especialidade de Sistemas, 1998.

TANOMARU, J. — *Motivação, fundamentos e aplicações de algoritmos genéticos*. Anais do II Congresso Brasileiro de Redes Neurais, 1995.

ZHUANG, M.; D.P.Atherton — *Automatic tuning of optimum PID controllers*. IEE Proceedings-D, 1993.

Anexo A. Sintonia de controladores PID para processos de primeira ordem com tempo de atraso. Cálculo dos parâmetros por Zhuang e Atherton.

O código presente neste anexo faz parte de um conjunto de funções que foram desenvolvidas em MatLab para o cálculo dos parâmetros de um controlador com base em vários critérios de sintonia. O código abaixo, foi utilizado para determinar os ganhos do controlador com base no critério de otimização de Zhuang e Atherton, para o exemplo 4 do capítulo 5. Mais informações sobre este, e sobre outros códigos desenvolvidos em MatLab para determinação dos ganhos do controlador PID, consultar: SIAM - “*Society for Industrial and Applied Mathematics*”

```
%% Função optPID %%  
  
function [Gc,Kp,Ti,Td,H]=optPID(key,typ,vars)  
k=vars(1); L=vars(2); T=vars(3); N=vars(4);  
Td=[];  
if length(vars)==5, iC=vars(5);  
switch key  
case 2  
A=[0.980,0.712,0.569,1.072,0.786,0.628;  
0.892,0.921,0.951,0.560,0.559,0.583;  
0.690,0.968,1.023,0.648,0.883,1.007;  
0.155,0.247,0.179,0.114,0.158,0.167];  
case 3  
A=[1.048,1.042,0.968,1.154,1.142,1.061;  
0.897,0.897,0.904,0.567,0.579,0.583;  
1.195,0.987,0.977,1.047,0.919,0.892;  
0.368,0.238,0.253,0.220,0.172,0.165;  
0.489,0.385,0.316,0.490,0.384,0.315;  
0.888,0.906,0.892,0.708,0.839,0.832];  
case 4  
A=[1.260,1.053,0.942,1.295,1.120,1.001;  
0.887,0.930,0.933,0.619,0.625,0.624;  
0.701,0.736,0.770,0.661,0.720,0.754;  
0.147,0.126,0.130,0.110,0.114,0.116;  
0.375,0.349,0.308,0.378,0.350,0.308;  
0.886,0.907,0.897,0.756,0.811,0.813];  
end  
ii=0; if (L/T>1) ii=3; end; tt=L/T;  
a1=A(1,ii+iC); b1=-A(2,ii+iC);  
a2=A(3,ii+iC); b2=-A(4,ii+iC); Kp=a1/k*tt^b1;  
Ti=T/(a2+b2*tt);
```

```

if key==3|| key==4
a3=A(5,ii+iC); b3=A(6,ii+iC); Td=a3*T*tt^b3;
end
else
Kc=vars(5); Tc=vars(6); k=vars(7);
switch key
case 2, Kp=0.361*Kc;Ti=0.083*(1.935*k+1)*Tc;
case 3, Kp=0.509*Kc; Td=0.125*Tc;
Ti=0.051*(3.302*k+1)*Tc;
case 4, Kp=(4.437*k-1.587)/(8.024*k-1.435)*Kc;
Ti=0.037*(5.89*k+1)*Tc; Td=0.112*Tc;
end
end
[Gc,H]=writePID(Kp,Ti,Td,N,key);

```

```

%% Função writePID %%

function [Gc,H]=writePID(Kp,Ti,Td,N,key)
switch key
case 1, Gc=Kp;
case 2, Gc=tf(Kp*[Ti,1],[Ti,0]); H=1;
case 3, nn=[Kp*Ti*Td*(N+1)/N,Kp*(Ti+Td/N),Kp];
dd=Ti*[Td/N,1,0]; Gc=tf(nn,dd); H=1;
case 4, d0=sqrt(Ti*(Ti-4*Td)); Ti0=Ti;
Kp=0.5*(Ti+d0)*Kp/Ti;
Ti=0.5*(Ti+d0); Td=Ti0-Ti;
Gc=tf(Kp*[Ti,1],[Ti,0]);
nH=[(1+Kp/N)*Ti*Td, Kp*(Ti+Td/N), Kp];
H=tf(nH,Kp*conv([Ti,1],[Td/N,1]));
case 5, Gc=tf(Kp*[Td*(N+1)/N,1],[Td/N,1]); H=1;
end

```

A síntese usada para chamar a função `optPID` no ambiente de trabalho do MatLab é:

```
[Gc,Kp,Ti,Td,H]=optPID(key,typ,vars)
```

Onde:

`key` = 2,3 ou 4 para o controlador PI, PID ou PID com filtro na acção derivativa, respectivamente.

`typ` = 1 ou 2, para mudança do sinal de referencia, ou minimização do sinal de perturbação, respectivamente.

`vars` = [K, L, T, N, C], onde C, representa qual critério que será utilizado. Para C = 1, 2, ou 3, tem-se ISE, ISTE ou IST^2E respectivamente. Como retorno tem-se a função de transferência G_c , que corresponde a equação do controlador PID, com a presença de um filtro de 1ª ordem, representada pela equação (9), descrita na subsecção 4.1.1, e os parâmetros do controlador K_p , T_i e T_d .

De salientar que, quando seleccionado key = 4, o retorno da função de transferência G_c é dado pela função de transferência do controlador PI, dada pela equação (3), apresentada na subsecção 2.3.2.

Desta forma a variável H, refere-se a função de transferência. Equivalente a uma malha de realimentação, de um controlador PID com acção derivativa no sinal de saída, dada pela seguinte equação [SIAM, 2010].

$$H(s) = \frac{\left(1 + \frac{K_p}{N}\right) T_i T_d s^2 + K_p \left(T_i + \frac{T_d}{N}\right) + K_p}{K_p (T_i s + 1) \left(\frac{T_d s}{N} + 1\right)} \quad (30)$$

O diagrama de blocos do sistema quando seleccionado (key = 4) é apresentado na Figura 47.

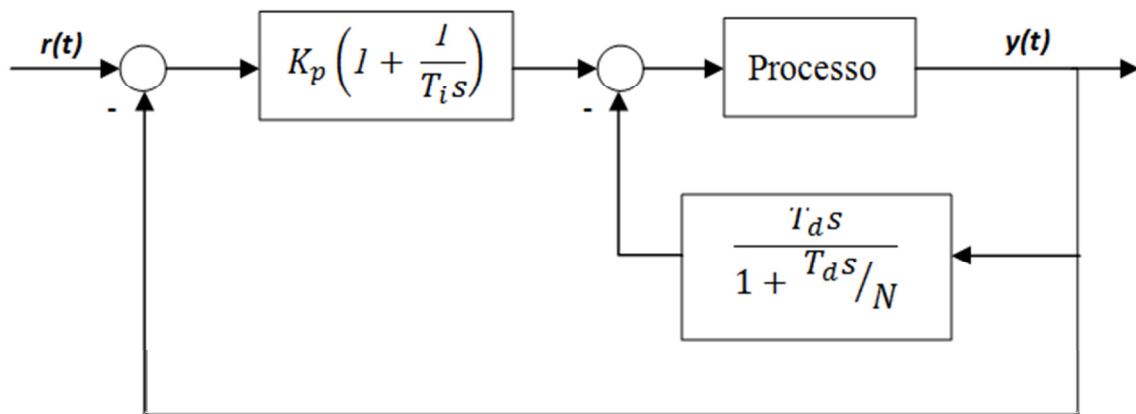


Figura 47 Controlador PID com acção derivativa no sinal de saída

- **Calculo dos parâmetros do controlador para o exemplo 4.**

Seja a função de transferência do exemplo 4, dada por:

$$G(s) = \frac{0,52e^{-28s}}{162s + 1}$$

Sendo então o controlador do tipo PID com filtro na acção derivativa, temos que key = 4 e typ = 1, visto que o cálculo a ser realizado é para a mudança do sinal de referência.

Logo: `optPID(key,typ,vars) = optPID(4,1,[0.52,28,162,100,2])`

```
>> [Gc,Kp,Ti,Td,H]=optPID(4,1,[0.52,28,162,100,2])  %% Prompt do MatLab
```

Transfer function:

$$2105 s + 9.806$$

$$214.7 s$$

$$Kp = 10.3614$$

$$Ti = 226.8202$$

$$Td = 11.5049$$

Transfer function:

$$2865 s^2 + 2106 s + 9.806$$

$$255.9 s^2 + 2106 s + 9.806$$