Universidade Federal de Ouro Preto
Escola de Minas
Colegiado do Curso de Engenharia de Controle e Automação - CECAU

SIMULATED ANNEALING APLICADO AO PROBLEMA DE SINTONIA DE PARÂMETROS DE CONTROLADORES PID

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Flávio Thimotio da Silva

Ouro Preto, 2005

Flávio Thimotio da Silva

SIMULATED ANNEALING APLICADO AO PROBLEMA DE SINTONIA DE PARÂMETROS DE CONTROLADORES PID

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção de grau como Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Marcone Jamilson Freitas Souza
Co-orientador: Prof. Dr. Ronilson Rocha

Ouro Preto

Escola de Minas - UFOP

Julho, 2005

Monografia defendida e aprovada, em 25 de Julho de 2005, pela comissão avaliadora constituída pelos professores:

_____

Orientador: Prof. Dr. Marcone Jamilson Freitas Souza

_____

Convidado: Prof. Dr. Paulo Marcos de Barros Monteiro

_____

Convidado: Prof. Dr. Henor Artur de Souza

Aos meus Pais, que nortearam minha vida e, abrindo mão de alguns de seus sonhos, tornou este meu sonho possível.

AGRADECIMENTOS

Agradeço aos meus pais pelo amor e carinho. Familiares e amigos que sempre contribuíram para a minha chegada até este ponto. Aos Camisas Azuis pelo exemplo de amizade. Á República Canaan pela escola de vida. A Escola de Minas, orgulho de ser EMOPIANO e a Universidade Federal de Ouro Preto. Ao ensino Público e Gratuito.

SUMÁRIO

RESUMO ..... **VII**

ABSTRACT ..... **VIII**

LISTA DE FIGURAS ..... **IX**

LISTA DE TABELAS ..... **X**

LISTA DE SIGLAS **11**

I. INTRODUÇÃO **1**

FIGURA 1.1 DIAGRAMA DE BLOCOS DE UM SISTEMA DE CONTROLE 2
1.1 ORIGEM DO TRABALHO 2
1.2 IMPORTÂNCIA DO TRABALHO 2
1.3 OBJETIVO 3
1.3.1 Objetivo Geral *4*
1.3.2 Objetivos Específicos *4*
1.4 LIMITAÇÕES DO TRABALHO 4
1.5 ESTRUTURA DO TRABALHO 5

II. CONTROLADORES INDUSTRIAIS **7**

2.1 INTERPRETAÇÃO DOS PARÂMETROS DO CONTROLADOR 12
2.2 CONSIDERAÇÕES DO CONTROLE PID 13
2.2.1 O Efeito Reset-Windup *13*
2.2.2 Filtragem do Termo Derivativo *14*
2.2.3 Ponderação da referência no termo proporcional *15*
2.3 CONTROLADOR PID DISCRETO 15

III. REVISÃO BIBLIOGRÁFICA **17**

3.1 MÉTODOS DE SINTONIA 17
3.2 ÍNDICES DE DESEMPENHO E ALGORITMOS 21
3.2.1 Integral do Erro Absoluto ponderada pelo Tempo (ITAE) *22*
3.2.2 Integral do Erro Quadrático (ISE) *23*
3.2.3 Integral Absoluta do Erro (IAE) *24*
3.2.4 Média do Erro Quadrático (MSE) *25*
3.2.5 Referência Baseado no Erro com Mínimo de Esforço de Controle (RBEMCE) *26*
3.2.6 Referência Baseada na Média do Erro Quadrático com Mínimo de Esforço de Controle (RBMSEMCE) *27*
3.3 MÉTODOS DE SINTONIA EMPÍRICOS 28
3.3.1 Método de Ziegler-Nichols em malha fechada *28*
3.4 META-HEURÍSTICAS 30
3.4.1 Algoritmos Baseados em Simulated Anneling *31*
3.4.2 Simulated Annealing proposto *32*

IV. METODOLOGIA **34**

4.1 O CONTROLADOR PID 34
4.2 REPRESENTAÇÃO DA SOLUÇÃO DO PROBLEMA 36
4.3 DETERMINAÇÃO DA SOLUÇÃO INICIAL 36
4.4 ESTRUTURA DE VIZINHANÇA 37
4.5 O ALGORITMO *SIMULATED ANNEALING* 37
4.6 AMBIENTE DE SIMULAÇÃO 39
4.7 FUNÇÃO DE TRANSFERÊNCIA 41
4.8 PARÂMETROS DE CONTROLE 43
4.9 DIAGRAMA DE BLOCOS DO SISTEMA 44

V. RESULTADOS OBTIDOS **45**

5.1 APRESENTAÇÃO DOS RESULTADOS 45
5.1.1 Simulação do Sistema 01 45
FIGURA 1.2 VALOR 45
FIGURA 1.3 1,0 45
5.1.2 Simulação do Sistema 2 47
FIGURA 1.4 VALOR 48
5.1.3 Simulação do Sistema 3 50
FIGURA 1.5 VALOR 50
FIGURA 1.6 1,0 50
5.2 ANÁLISE DOS RESULTADOS 53
VI. CONCLUSÕES E TRABALHOS FUTUROS 56
VII. REFERÊNCIAS BIBLIOGRÁFICAS 57
GLOSSÁRIO 60

RESUMO

Um ramo importante da área de Controle e Automação de processos é a sintonia dos parâmetros dos controladores PID. O presente trabalho tem como o objetivo o desenvolvimento de um algoritmo computacional para a sintonia destes. Os métodos clássicos de sintonia têm se mostrado ineficiente quanto ao desempenho de malhas de ordem superior e, conjuntamente a este cenário, é que está inserido o Problema de Sintonia de Parâmetros de Controladores (PSPC). A resolução deste problema define uma boa sintonia dos parâmetros livres do controlador, ou sintonia sub-ótima, a ser implementada no processo, de forma a atender aos índices de desempenho estabelecidos. O algoritmo de busca utilizado é baseado na meta-heurística *Simulated Annealing*, que faz analogia ao mecanismo de recozimento de metais, sendo utilizada em otimização combinatória para realizar buscas por soluções factíveis, de menor energia, sendo a energia mínima a solução ótima. São apresentados resultados da simulação para três funções de transferência não-lineares e um novo índice de desempenho é proposto.

Palavras-Chave: Sintonia, otimização, Controladores PID, Índice de Desempenho, *Simulated Annealing*.

ABSTRACT

Keywords*:*

LISTA DE FIGURAS

LISTA DE TABELAS

Tabela 1. Algoritmo ITAE 22
Tabela 2. Algoritmo ISE 23
Tabela 3. Algoritmo IAE 24
Tabela 4. Algoritmo MSE 25
Tabela 5. Algoritmo RBEMCE 26
Tabela 6. Algoritmo RBMSEMCE 27
Tabela 7. Método de ZIEGLER-NICHOLS para obter os parâmetros do controlador com resposta em “S” 29
Tabela 8. Método de ZIEGLER-NICHOLS para obter os parâmetros do controlador em malha fechada 30
Tabela 9. Algoritmo Controlador Proporcional 34
Tabela 10. Algoritmo Controlador Integral 35
Tabela 11. Algoritmo Controlador Derivativo 35
Tabela 12. Pseudo-Código para o procedimento SA 38
Tabela 13. Parâmetros de Simulação 43
Tabela 14. Configuração básica do Sistema 1 45
Tabela 15. Resultados da Simulação do Sistema 1 46
Tabela 16. Configuração básica de simulação do Sistema 2 48
Tabela 17. Resultados da Simulação Básica do Sistema 2 48
Tabela 18. Configuração básica de simulação do Sistema 3 50
Tabela 19. Resultados da Simulação Básica do Sistema 3 51

LISTA DE SIGLAS

PID	Proporcional - Integral - Derivativo
PSPC	Problema de Sintonia de Parâmetros de Controladores
K_p	Ganho Proporcional
K_i	Ganho Integral
K_d	Ganho Derivativo
AG	Algoritmos Genéticos
BT	Busca Tabu
SA	*Simulated Annealing*
CLP	Controlador Lógico Programável – idem PLC
PLC	*Programmable Logic Controller* – idem CLP
ISE	*Integral of the Square of the Error*
MSE	*Mean of the Square of the Error*
ITAE	*Integral of Time multiplied by Absolute Error*
IAE	*Integral of Absolute Magnitude of the Error*
RBEMCE	*Reference Based Error with Minimum Control Effort*
RBMSEMCE	*Reference Based Mean Square Error with Minimum Control Effort*

I. INTRODUÇÃO

Os Controladores PID são bastante utilizados nas indústrias para controle de sistemas em geral, tais como pressão, nível, velocidade, entre outros. De fato, na indústria, 95% das malhas de controle utilizam controladores PID, e em sua maioria são controladores PI (ASTRÖM & HÄGGLUND, 1995).

O projeto desses controladores é classicamente realizado com a resposta de um modelo de referência de segunda ordem e, a partir do conhecimento da planta que deverá ser controlada, encontra-se os parâmetros de ganho proporcional, integral e derivativo do controlador PID. Este controlador, utilizado em cascata com esta planta, em malha fechada, possibilita que a mesma acompanhe a resposta do modelo da melhor forma possível, (ASTRÖM & HÄGGLUND, 1995).

O esforço realizado pelo controlador é de determinar à inversa da função de transferência da planta, de forma que o valor desejado para a operação do sistema seja transferido diretamente para a saída.

O Problema de Sintonia de Parâmetros de Controladores (PSPC) PID é um problema ainda pouco estudado pela área de otimização, visto os métodos clássicos de sintonia apresentarem resultados satisfatórios quanto à resposta dos sistemas. Na indústria, nota-se uma aplicação empírica de métodos baseados no conhecimento do operador para ajustar os parâmetros.

O desempenho do controle para os métodos clássicos é fortemente dependente do processo e o modelo matemático do sistema se faz necessário para a sua análise e posterior sintonia. Tais ajustes podem ser feitos de várias formas, dentre elas pode-se destacar a otimização abordada neste trabalho, que consiste, basicamente, em considerar os três parâmetros do Controlador PID (K_p, K_i e K_d) como variáveis independentes de um sistema quadridimensional desconhecido, que pode possuir vários mínimos locais representando valores ótimos, ou sub-ótimos, que proporcionam uma resposta do sistema a uma entrada padrão, e tentar minimizar um índice de desempenho $J(K_p, K_i, K_d)$.

Na Figura 1.1 mostra-se um sistema de controle na forma de Diagrama de Blocos.

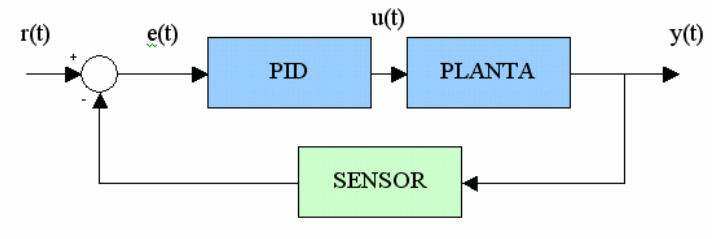

Figura 1.1 Diagrama de Blocos de um sistema de controle

Neste capítulo, além dessa breve descrição do problema abordado, são apontadas à origem e a importância do estudo do problema de sintonia de controladores PID, bem como os objetivos deste trabalho e sua estruturação.

1.1 Origem do Trabalho

O trabalho teve origem a partir do conhecimento adquirido pelo autor nas disciplinas cursadas na área de Teoria de Controle de sistemas contínuos e Teoria de Controle de sistemas discretos. Juntou-se a estes conhecimentos a motivação de unir o problema da sintonia de controladores, abordado nestas disciplinas, e os problemas abordados nas disciplinas Otimização Combinatória e Inteligência Computacional para Otimização, que necessitam de técnicas especiais para serem resolvidos e possuem um vasto campo de aplicações práticas.

1.2 Importância do Trabalho

O desempenho das malhas de controle está diretamente relacionado ao rendimento do processo associado ao controlador. A entrada do controlador, definida como o erro $e(t)$, ou $e(kT)$, para o caso discreto, com T definido como tempo de amostragem, está relacionado à perda de eficiência do processo, pois, dada à

referência $r(t)$, que define a saída desejada do sistema, o erro indica o quão "afastado" do desejado o sistema se encontra.

O correto ajuste dos controladores implica em uma resposta do sistema ajustada de forma a seguir a entrada desejada, sendo que, no caso do ajuste ótimo, o sistema responde de forma a ter o menor índice de desempenho possível. Desta forma a eficiência do processo se garante quanto à resposta à entrada de referência a ser seguida, resultando em um controlador mais adequado e rentável.

Neste contexto, o ajuste dos parâmetros do controlador PID passa a ser encarado como uma etapa importante nas decisões das empresas, pois respostas para as perguntas: "O quanto estamos perdendo de produção/rendimento com o ajuste dos controladores da planta?", ou "O quanto podemos melhorar a produção/rendimento com um melhor ajuste dos controladores?". Tomar a decisão correta pode acarretar em aumento direto do rendimento do processo.

Conjuntamente a este cenário é que está inserido o Problema de Sintonia de Parâmetros de Controladores (PSPC) PID. A resolução deste problema define a melhor sintonia dos controladores a se implementada no processo de forma a atender ao índice de desempenho definido. Desta forma, a resolução deste problema contribui para a redução dos custos das empresas com malhas de controle mal ajustadas.

Destaca-se, também, a existência de poucos sistemas computacionais para resolver tais problemas, grande parte deles muito fracos do ponto de vista da ferramenta de otimização utilizada.

Finalmente, ressalta-se que o desenvolvimento deste trabalho contribuirá com a divulgação de técnicas de otimização nas empresas, possibilitando a redução de seus custos com o processo e valorizando o papel do engenheiro de controle e automação nas atividades empresariais.

1.3 Objetivo

Nesta seção são apresentados os objetivos deste trabalho, tanto o geral quanto os específicos.

1.3.1 Objetivo Geral

A presente monografia tem como por objetivo desenvolver um programa computacional para resolver o Problema de Sintonia de Parâmetros de Controladores (PSPC) PID, através de técnicas de otimização combinatória e inteligência computacional.

1.3.2 Objetivos Específicos

Os objetivos específicos a serem alcançados são os seguintes:

- Revisar a literatura sobre as técnicas mais comumente utilizadas para resolver o Problema de Sintonia de Parâmetros de Controladores (PSPC) PID;
- Desenvolver uma metodologia heurística para resolver o Problema de Sintonia de Parâmetros de Controladores (PSPC) PID
- Estudar a linguagem de programação e o ambiente de programação do *software* C++ Builder 5.0 da Borland;
- Implementar um sistema computacional na linguagem C, e aplicando a metodologia desenvolvida, resolver o problema proposto;
- Validar o sistema desenvolvido, pela sua aplicação a problemas-teste reais.

1.4 Limitações do Trabalho

Sendo heurística a metodologia proposta, este trabalho limita-se a encontrar soluções aproximadas para o Problema de Sintonia de Parâmetros de Controladores (PSPC) PID. Dessa forma não se garante que o mesmo encontrará a

solução ótima para uma instância real qualquer. Outra limitação reside no fato de o sistema ser altamente dependente do modelo matemático do processo. Um módulo de reconhecimento de sistemas pode ser desenvolvido e incorporado facilmente ao projeto.

1.5 Metodologia Adotada

Como forma de representação da solução do problema, considerando-se S o conjunto de possíveis soluções, temos:

$$S = \{s = (K_p, K_i, T_d) : K_p \in \Re, K_i \in \Re \text{ e } T_d \in \Re) \} \quad (1)$$

Definiu-se para o *Simulated Annealing* o operador de geração da vizinhança de s, N(s), tal que o vizinho é gerado incrementando-se a solução corrente de um valor entre ± 5% do máximo valor do parâmetro, sendo esta operação realizada apenas sobre um parâmetro do controlador, de cada vez, possibilitando assim uma busca orientada.

Em relação ao controle, várias estratégias são abordadas, entre elas, o *Anti-Reset-Windup*, que limita a acumulação do termo integral, o pólo de limitação de ganho em altas freqüências para o controlador derivativo e a ponderação de referência do termo proporcional.

Por fim, como modelos de simulação, são apresentadas três funções de transferência não-lineares que representam processos fictícios, mas que exigem grande esforço de controle.

1.6 Estrutura do Trabalho

No capítulo 1 é feita uma breve descrição sobre o problema abordado, explicitando a origem, a importância, os objetivos e as limitações da presente monografia.

No capítulo 2 apresentam-se conceitos sobre controladores industriais de vários tipos e considerações sobre desempenho de cada um deles são abordadas.

No capítulo 3 apresenta-se a revisão bibliográfica da literatura acerca do assunto abordado, mais especificamente das técnicas apresentadas por vários autores para solucionar o Problema de Sintonia de Parâmetros de Controladores (PSPC) PID.

No capítulo 4 é apresentada a metodologia proposta para resolver o problema, incluindo a descrição detalhada do mesmo, as estruturas de vizinhança adotadas para o problema, determinação da solução inicial, os índices de desempenho que podem ser utilizados e, por fim, o método heurístico aplicado ao problema.

No Capítulo 5 são apresentados os resultados computacionais, bem como uma avaliação dos mesmos em relação aos vários índices empregados.

No Capítulo 6 conclui-se o trabalho desenvolvido, bem como se apresentam sugestões para possíveis trabalhos futuros.

II. CONTROLADORES INDUSTRIAIS

Os controladores industriais microprocessado possibilitam hoje, em pouco tempo, estabelecer uma malha de controle necessária para se manter o processo com o mínimo de erro em relação a um sinal de referência desejado. Em muitos casos, como na utilização de CLPs, é possível um fácil ajuste dos parâmetros destes controladores frente a variações ocorridas no processo e/ou a perturbações externas.

Vários são os tipos de controladores utilizados para se obter uma resposta desejável. Entre eles o de mais simples implementação é o controlador *On-Off*, que para um sinal de realimentação acima da referência mantém o sinal atuante no mínimo possível, enquanto que se o sinal de erro for negativo mantém o sinal de atuação no máximo possível. O diagrama de blocos deste controlador é apresentado na Figura 2.1, bem como sua resposta à entrada degrau. Nota-se que a saída apresenta uma grande variação em torno da referencia desejada, Figura 2.2.

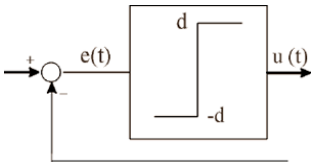

Figura 2.1 Diagrama de Blocos do Controlador On-Off

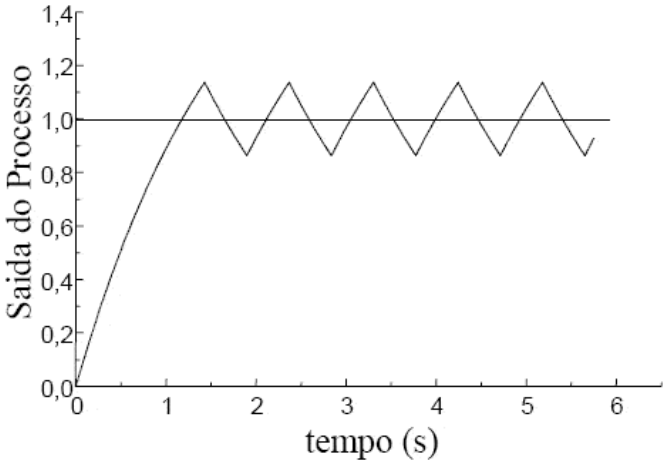

Figura 2.2 Forma de onda típica para um controlador On-Off

Para minimizar a freqüência de variação do controlador *On-Off* uma parâmetro, chamado de Histerese ou Intervalo Diferencial, que corresponde a uma

região em torno do valor de referência onde o sinal de saída é mantido no seu valor atual até que o erro saia da faixa determinada. O diagrama de blocos e a nova resposta do sistema podem ser visualizados, respectivamente, nas Figuras 2.3 e 2.4.

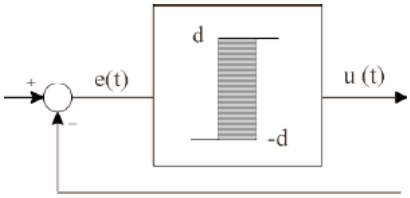

Figura 2.3 Diagrama de blocos do controlador On-Off com Histerese.

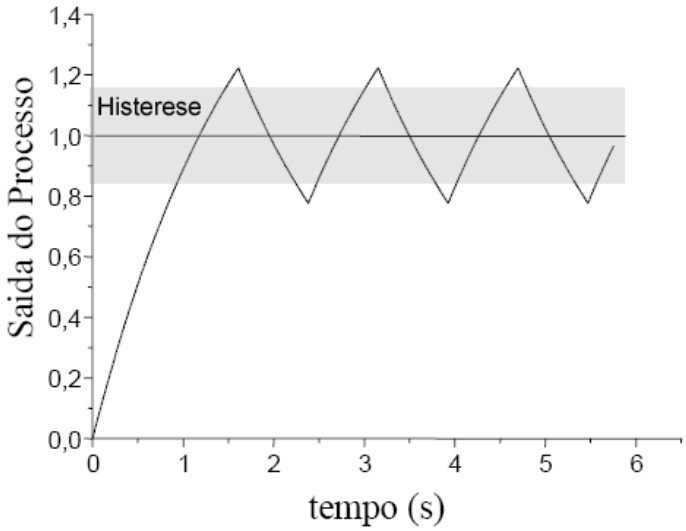

Figura 2.4 Forma de onda típica para um controlador *On-Off* com Histerese

Estes controladores são muito utilizados em dispositivos simples que não requerem uma sofisticação no controle, tais como máquinas de lavar, fornos em geral e ar-condicionado.

Para se ter uma melhor resposta de controle a alternativa é inserir um dispositivo atuador mais sofisticado e calcular com mais precisão a saída do controlador, inserindo uma relação proporcional ao erro da saída, em outras palavras, inserindo um ganho K_p ao controlador.

Este controlador é conhecido como Controlador Proporcional, e seu diagrama de blocos pode ser visualizado na Figura 2.5. Ele permite algumas melhoras na resposta do sistema, principalmente quanto à sensibilidade a variações de parâmetros internos e externos, mas por outro lado surge a possibilidade do sistema apresentar instabilidade. Qualquer que seja a natureza ou a forma de energia

utilizada na implementação do controlador proporcional ele será essencialmente um amplificador com ganho ajustável (OGATA, 1993).

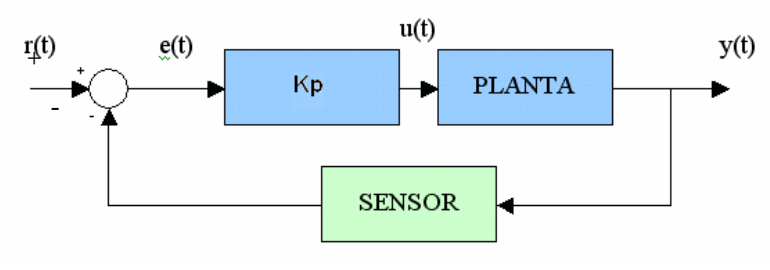

Figura 2.5 Diagrama de blocos do controlador proporcional

O sinal de controle do controlador proporcional é:

$$u(t) = K_p e(t) \tag{2}$$

Uma maneira mais didática de demonstrar a interligação dos diversos dispositivos necessários ao controle de um sistema é exemplificar com uma planta de controle de nível.

Um sistema de controle de nível pode ser assim descrito: um tanque onde um líquido qualquer deve ser mantido em um nível determinado, sendo este regulado pela válvula de entrada. A saída do tanque alimenta outro processo, mas para fins de exemplo a saída do líquido pode ser considera como uma perturbação constante na planta.

O sensor/transdutor de nível transforma uma grandeza física em outra, que pode ser manipulada pelo controlador, chamada de PV (variável de processo). Normalmente esta grandeza é uma corrente variando de 4 a 20 mA (STEPHANOPOULOS, 1984). A saída do controlador também é um sinal, normalmente de 4 a 20 mA, que é enviado ao atuador para atuar no processo, chamada de VS (variável de saída). O *set-point* (SP) e a PV devem estar na mesma unidade de engenharia para que a comparação dos valores possa ser realizada.

Em suma, o controlador atuará na válvula de controle (atuador) de forma a minimizar o erro entre o nível atual e o nível desejado, sendo que esta ação é a de abertura ou fechamento da válvula de 0 a 100%. Normalmente o atuador é modelado juntamente com a planta, pois os tempos de resposta do mesmo influenciam no desempenho global. Sistemas de atuação rápida podem ser desconsiderados caso o controle apresente constantes de tempo elevadas ou atrasos na resposta, mas sempre o projetista deve julgar necessária a sua consideração.

Outro ponto importante é a resposta do elemento sensor/transdutor, que também possui um modelo que pode ser considerado na modelagem do sistema. Novamente cabe ao projetista o julgamento da modelagem do elemento sensor/atuador para o sistema.

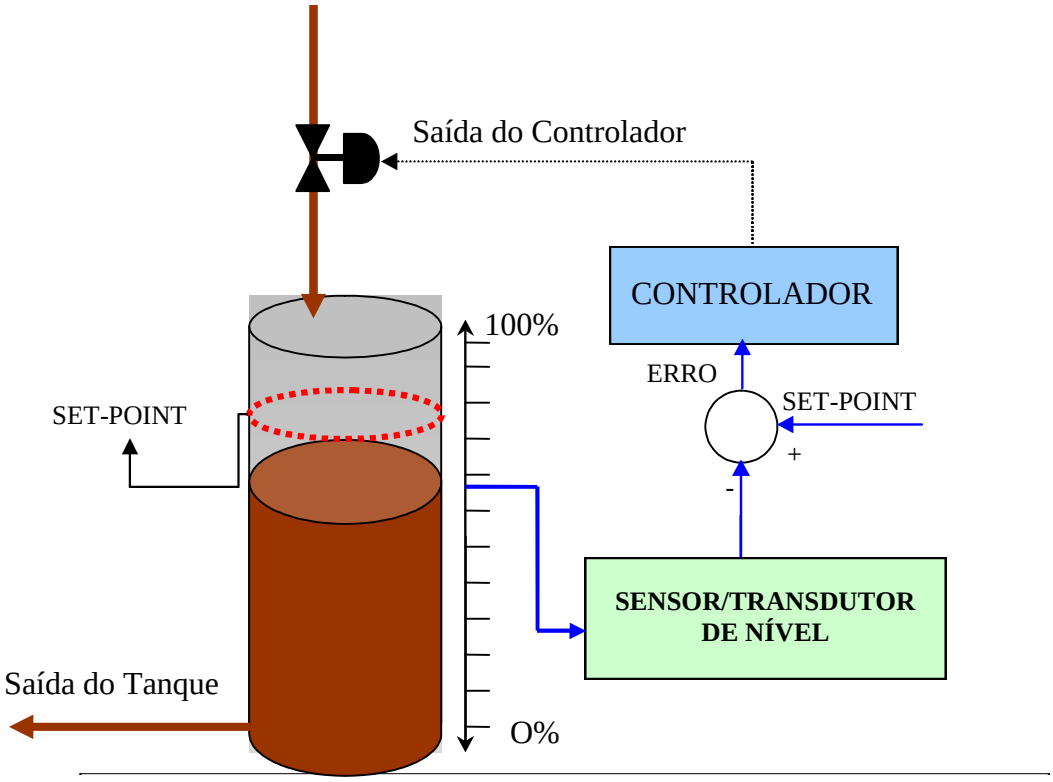

Figura 2.6 Sistema de Controle de Nível de Tanque

Simplesmente a implementação de uma ação de controle proporcional não garante um desempenho satisfatório para a maioria dos sistemas, principalmente aqueles que apresentam erro em regime estacionário. Para solucionar este problema a ação de controle integral é necessária.

No controle integral, o sinal de controle é, em qualquer instante, proporcional à área da curva do sinal de erro atuante naquele instante. O sinal de controle pode possuir um valor não nulo mesmo quando o erro for nulo, o que é impossível no caso do controle proporcional. O valor da saída do controlador integral é, considerando K_i como parâmetro de proporcionalidade:

$$u(t) = K_i \int_0^t e(t)dt \qquad (3)$$

Se o valor do erro for dobrado, o sinal de controle varia duas vezes mais rápido para o controle integral. Esta ação também é conhecida como controle de restabelecimento (*reset*), Figura 2.7.

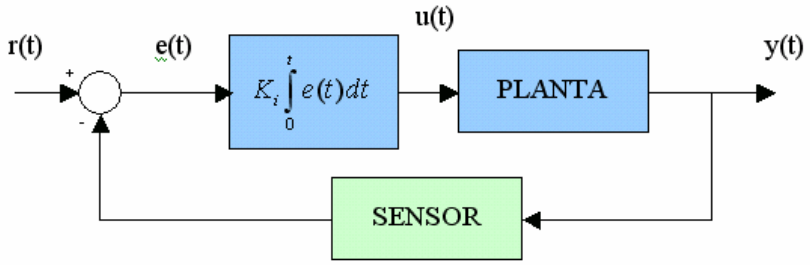

Figura 2.7 Diagrama de blocos do controlador integral

Muitas vezes a sensibilidade do controlador às variações da planta não atendem aos requisitos de controle. Para solucionar este problema a ação de controle derivativa deve ser utilizada. A vantagem de utilizar-la é que a resposta é proporcional à taxa de variação do erro atuante, o que pode garantir uma ação de controle que minimize o erro antes que o mesmo atinja valores demasiadamente grandes. Este controlador tem a característica de antecipar a ação corretiva do erro, tendendo a aumentar a estabilidade do sistema. O sinal de controle para o controlador derivativo é, considerando K_d como parâmetro de proporcionalidade:

$$u(t) = K_d \frac{de(t)}{dt} \tag{4}$$

Embora o controle derivativo não afete diretamente o erro em regime estacionário, ele produz um amortecimento no sistema, permitindo a utilização de ganhos proporcionais mais elevados, resultando em uma maior precisão em regime estacionário.

Contudo, o fato do controlador derivativo operar sobre a taxa de variação do erro, e não sobre o próprio erro atuante, ele deve ser utilizado em conjunto ao controlador proporcional ou proporcional-integral, Figura 2.8. As desvantagens deste controlador são a amplificação dos ruídos em alta freqüência e a freqüente saturação do atuador. Esta saturação pode ser eliminada com uma pequena alteração na entrada do controlador derivativo, retirando-se a ação sobre o erro e passando a atuar diretamente sobre o valor da variável de processo.

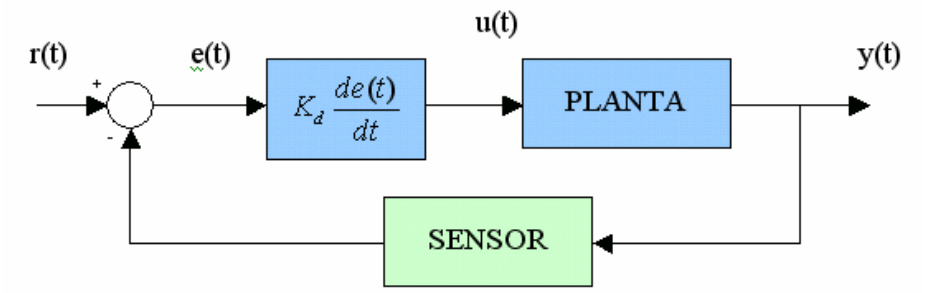

Figura 2.8 Diagrama de blocos do controlador derivativo

A partir das características apresentadas sobre cada controlador, pode-se estabelecer a ação de controle completa para o sistema, ou seja, a ação de controle PID (Proporcional-Integral-Derivativo), ou mesma a combinação de uma das partes com outra, resultando na soma de todas as ações de controle implementadas ao mesmo tempo, ou seja, conforme a figura abaixo, a ação de controle PID será:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{5}$$

2.1 Interpretação dos Parâmetros do controlador

Como sugere a equação (4), tem-se três parâmetros livres para a definição do controlador PID, sendo eles o K_p, K_i e o K_d. Estes parâmetros estão na forma conhecida como equação de termos independentes.

Outra representação desta equação, conhecida como equação ISA, pode ser definida em temos dos tempos de controle integral e derivativo que, em temos dos parâmetros acima, podem ser relacionados como:

$$T_i = \frac{K_p}{K_i} \tag{6}$$

$$T_d = \frac{K_d}{K_p} \tag{7}$$

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right) \tag{8}$$

Fisicamente, o inverso do Tempo Integral, denominado Taxa de Restabelecimento, representa o numero de vezes por unidade de tempo que a parte proporcional da ação de controle é duplicada. A taxa é normalmente medida em repetições por minuto (rep/min). Já o Tempo Derivativo, também denominado Controle de Taxa, é aonde a magnitude da saída do controlador é proporcional à taxa de variação do sinal de erro atuante. Este tempo T_d é o intervalo de tempo pelo qual a ação derivada avança o efeito da ação de controle proporcional. No entanto, a ação de controle derivativa nunca poderá antecipar uma ação que ainda não tenha ocorrido.

2.2 Considerações sobre o Controle PID

No momento da aplicação prática dos controladores PID algumas considerações devem ser ressaltadas para que se obtenha o desempenho adequado do sistema controlado. Sem elas, o efeito final do controlador pode ser degenerado ou mesmo inadequado. Os principais efeitos a serem considerados são a saturação do sinal atuante, a limitação do ganho do termo derivativo em altas freqüências e a ponderação do sinal de referência por um fator adequado.

2.2.1 O Efeito *Reset-Windup*

Na equação (7), observa-se que uma ação de controle será grande quando o sinal de erro ou a variação do sinal de erro for grande, causando, assim. a saturação do atuador. Este sinal demasiado pode causar um retardo na resposta do sistema, mantendo o erro em um sinal positivo, o que implica em um contínuo aumento do sinal de controle devido à ação do controlador integral.

Quando a resposta da planta atingir o sinal de referência, a inversão do sinal do erro pode demorar a ter efeito sobre a ação de controle devido ao alto valor acumulado pelo termo integral. Este efeito é denominado *Reset-Windup.*

Para evitar este fenômeno, utiliza-se um esquema denominado *Anti-Reset-Windup*. Este modelo pode ser implementado de diversas maneiras, entre as quais sugeridas por ISERMANN (1989):

- Cancelamento da integração: neste caso faz-se $\Delta u_I(t) = 0$ quando $u(t) > u_{\max}$, ou seja, quando a saída do controlador integral atingir o máximo permitido o acumulado da integral é resetado;
- Integração Condicional: neste caso faz-se $\Delta u_I(t) = 0$ se $|e(t)| > e_{\max}$, sendo o $e_{\max}$ determinado a partir de experimentos ou da própria função de transferência do processo.

2.2.2 Filtragem do Termo Derivativo

Como a ação de controle derivativa apresenta uma amplificação para sinais de alta freqüência, o mesmo pode levar o sistema à instabilidade devido ao crescente ganho apresentado nesta faixa do espectro.

Da função de transferência do controlador derivativo podemos verificar que ele se compara a um controlador por avanço de fase com pólo no infinito. De seu Diagrama de Bode, concluímos que sua característica é adicionar angulo de fase positivo a resposta em freqüência do sistema como também melhorar a resposta em regime transitório (OGATA, 1993).

O ganho da malha do controlador mais o processo tenderão ao infinito para altas freqüências, o que pode levar o sistema a instabilidade ou prejudicar o desempenho do sistema pela amplificação de ruídos. Este efeito é denominado *quick derivative*.

A solução para este problema é a adição de pelo menos um pólo ao termo derivativo, resultando na seguinte função de transferência para o controlador em questão:

$$D = \frac{K_p T_d s}{1 + \frac{T_d}{N} s} \tag{9}$$

com o valor de N entre 3 < N < 20, sendo usualmente utilizado N=10, na literatura, sem perda de generalidade.

A finalidade deste pólo é manter, em altas freqüências, o ganho em, no máximo, $K_p N$, limitando-se a amplificação dos ruídos.

2.2.3 Ponderação da referência no termo proporcional

Esta é a principal modificação na ação de controle do PID, e é a que traz resultados significativos ao desempenho do controlador. Esta modificação foi inicialmente proposta por HANG, ASTROM & HO (1991), que introduz um fator de ponderação b sobre a referência desejada no termo proporcional, ou seja:

$$u(t) = K_p \left(\left(b \cdot r(t) - y(t) \right) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right) \tag{10}$$

Mais informações sobre a formulação proposta para a ponderação pode ser encontrado em CAON, 1999.

Os benefícios desta modificação podem ser sentidos na alteração do sobre sinal na saída do sistema, pois o fator de ponderação ajusta os pólos do sistema em malha fechada, garantindo uma performance satisfatória.

2.3 Controlador PID discreto

Para a implementação da lei de controle contínua do controlador na forma digital é necessário adotar aproximações para o termo integral e o termo derivativo. Esta aproximação também é conhecida como discretização.

Para a ação de controle proporcional tem-se que a resposta discreta do controlador pode ser assim definida:

$$u(k) = K_p\left[b \cdot r(k) - y(k)\right] \tag{11}$$

A ação integral parte da idéia da aproximação pela diferença, ou seja:

$$I(t) = \frac{K_p}{T_i}\int_0^t e(t)dt \xrightarrow{\frac{d}{dt}} \frac{dI}{dt} = \frac{K_p}{T_i}e(t) \tag{12}$$

Aproximando a derivada por uma diferença, teremos:

$$\frac{I(k+1) - I(k)}{\Delta t} = \frac{K_p}{T_i}e(k) \tag{13}$$

Isolando o termo em k+1 temos a seguinte equação recursiva para o termo integral:

$$I(k+1) = I(k) + \frac{K_p \Delta t}{T_i}e(k) \tag{14}$$

Para a ação derivativa podemos ter três formas de discretização: Diferença Posterior, Diferença Anterior e a Aproximação de Tustin. Conforme CAON, 1999, apenas a aproximação por Diferença Posterior , que pode-se visualizar na equação (14), apresenta bons resultados para qualquer T_d. Logo, a utilizaremos para o estudo em questão:

$$D(k+1) = \left(1 - \frac{\Delta t \cdot N}{T_d}\right)D(k) - K_p N\left[y(k+1) - y(k)\right] \tag{15}$$

Nota-se que a ação derivativa está sendo empregada sobre a variável de processo e não sobre o erro. Esta realocação é a forma encontrada para evitar que a ação atue sobre a variação do erro, que pode ser muito grande, enquanto que a ação sobre a variável de processo é mais suave e evita a saturação do atuador.

III. REVISÃO BIBLIOGRÁFICA

3.1 Métodos de Sintonia

O maior problema com os controladores PID na indústria é a sintonia adequada de seus parâmetros, necessárias para se produzir uma resposta adequada do sistema, sem sobre-sinal e suficientemente rápidas. Comumente esta sintonia é realizada de forma manual através de gráficos de reposta do sistema, como proposto por ZIEGLER & NICHOLS (1942), que depende muito do responsável pelo projeto do controlador. Outra maneira muito utilizada era partir o sistema em modo manual e ir ajustando os parâmetros à medida que a saída do processo fosse se aproximando da saída desejada (CAON, 1999). No entanto este método requer muita experiência do operador e demanda muito tempo para o correto ajuste dos parâmetros.

Nos últimos anos, várias técnicas de auto-sintonia têm sido propostas para o ajuste de controladores PID. Este esforço é motivado por duas razões:

Os controladores PID são de implementação simples e rápida, sendo que, atualmente, todos os CLP de boa qualidade possuem interface para a sua implementação direta. Este controlador é muito aceito na indústria (LEVA, 1993).

Para a sintonia dos parâmetros do controlador PID vários autores propõem métodos distintos e que requerem informações variadas sobre o processo a ser controlado, tais como margem de ganho e fase, resposta ao degrau unitário em malha aberta, conhecimento de pontos da curva de Nyquist, constante de tempo de malha fechada, entre outros. Estes métodos derivam leis de sintonia simples para reguladores PID e muitos auto-sintonizadores comerciais baseados nestes métodos de sintonia estão disponíveis há mais de 20 anos.

Em NISHIKAWA *et al.* (1984) o conceito de áreas características da malha fechada é utilizado para identificação de processos de primeira ordem mais tempo morto, e os parâmetros PID são obtidos com o critério ISE, resolvido através do determinante de Hurwitz.

Em LEVA (1993) um procedimento de sintonia baseado na realimentação a relé com inclusão de dinâmicas de primeira ordem (filtros) na entrada e saída do processo é usado na obtenção dos parâmetros, mas com duração de auto-sintonia bastante elevada.

WANG, BARNES e CLUETT (1995) utilizaram dois pontos do gráfico de resposta em freqüência em malha fechada do processo e da resposta desejada para o sinal de controle através de uma função de ponderação complexa para o sinal aplicado como referência para obterem os parâmetros do controlador.

Com VODA & LANDAU (1995) a auto-sintonia é realizada combinando-se o experimento a relé com regras de sintonia baseados no princípio da simetria ótima de Kressler (1958). No entanto, o tempo de acomodação alcançado é o mais conservador entre os métodos apresentados.

KHAN & LEHMAN (1996) apresentam um método que seleciona os ganhos de um controlador PI que utiliza o cancelamento Pólo-Zero a partir da análise do efeito do controle proporcional sobre o sobre-sinal e tempo de subida. Este método resulta em um bom tempo de subida para tempos de atraso menores ou iguais a 10 segundos.

Em POULING *et al.* (1996) a sintonia é baseada na minimização do critério ITAE aproximada para controladores PI. Neste método, a ausência de sobre-sinal é alcançada à custa de um tempo de acomodação muito elevado.

FUNG, WANG e LEE (1998) apresentam um método gráfico para obter os parâmetros do controlador PI a partir das especificações de margem de fase e ganho que são desdobradas em duas funções complexas. O ponto de interseção destas duas funções fornece os parâmetros desejados. Embora o método resulte em bons resultados sua implementação não é trivial.

Todos os métodos acima citados são contínuos no tempo, sendo necessário para a realização do controlador a discretização do mesmo. Uma alternativa a esta solução é utilizar o modelo discreto do processo e sintonizar o controlador a partir deste, obtendo assim o controlador discreto diretamente.

Com a escolha do modelo discreto da planta e do controlador o problema de encontrar um método de discretização apropriado é evitado. Outra vantagem é que a implementação discreta do controlador permite realizar compensações de tempo morto elevados de forma mais direta através de técnicas de predição (ISERMANN, 1989).

O problema da sintonia ótima dos parâmetros do controlador PID pode ser encarado como um problema de otimização combinatória assim enunciado:

"Definido um índice de desempenho para o sistema de controle, determinar os parâmetros do controlador PID de modo que, dentro do intervalo de tempo especificado, a resposta do sistema de controle composto pela planta e o controlador minimize o índice $J(K_p, K_i, K_d)$".

Matematicamente:

$$Min\ \{J(K_p, K_i, K_d)\} \tag{16}$$

$$J(K_p, K_i, K_d) = I_k(e(t)) \tag{17}$$

$$e(t) = r(t) - y(t) \tag{18}$$

$$y(t) = C(e(t)) * G(t) \tag{19}$$

sendo $0 \le t \le t_\infty$, onde I_k é um índice de desempenho definido, $r(t)$ é a referência, que pode ser o *set-point* ou uma resposta padrão desejada para o sistema, $y(t)$ é a resposta do sistema à entrada de referência, $C(e(t))$ é à saída do controlador PID para o erro $e(t)$ no instante t e $G(t)$ é a resposta do sistema ou modelo do sistema.

Para o projeto de controladores, o mesmo deve satisfazer a especificações de desempenho definidas pelo projetista. Como os sistemas de controle tratam de sistemas dinâmicos, estas especificações de desempenho podem ser tratadas em termos da resposta transitória a determinadas entradas do sistema ou a índices de desempenho.

A modelagem e o reconhecimento de Sistemas Dinâmicos é um campo essencial para o desenvolvimento do controlador apropriado. O desempenho do controle clássico é altamente dependente da qualidade do modelo utilizado, visto que para certas aplicações o sistema em questão pode apresentar pouca variação em torno do ponto de linearização utilizado no modelamento.

Um ponto importante na análise de sistemas de controle é a não linearidade dos processos. Normalmente a aplicação do controle PID, que independe do sistema ser linear ou não, é utilizada desconsiderando-se as implicações do ponto de operação do sistema, podendo, em certos momentos, operar fora da região linear

ou do ponto de linearização, principalmente em sistemas onde a referência não é mais um Set-point, mas sim uma trajetória ou função de referência.

A simulação computacional surge, possibilitando a análise de controle robusto (variação dos parâmetros internos), tolerância a ruídos de qualquer espécie e, além disso, permite personalizar a sintonia para determinado processo, realçando características importantes para que a sintonia responda de forma ótima e dentro do desejado, pois nem sempre a característica de ótimo garante a satisfação dos requisitos do processo, o que dificilmente é alcançado pelos métodos empíricos e analíticos.

Recentemente, os esforços para se desenvolver algoritmos de sintonia *on-line* de parâmetros dos controladores têm se intensificado. Métodos heurísticos de busca ótima e sub-ótima se destacam pelo tempo de computação inferior às técnicas clássicas de sintonia, apresentando resultados que podem levar em conta variações de parâmetros, ruídos gaussianos, não-linearidades e interferências de forma rápida e robusta. A vantagem de se programar um sintonizador *on-line* reside no fato dos parâmetros dos processos apresentarem variações durante sua operação, o que garantiria a "otimalidade" do sistema mesmo em condições adversas e imprevisíveis.

Considerando-se o fato de que o controle hoje é realizado por dispositivos programáveis, a alteração "a quente" dos parâmetros não é mais um problema. A dificuldade que se encontra hoje para implementar tais sistemas de sintonia on-line está na criticidade de alguns processos ou na não necessidade de se obter um controle rígido, além de não se confiar em um algoritmo para comandar a operação de um sistema. Nestes casos, a solução off-line é conveniente, garantindo-se ao engenheiro ou projetista de controle a decisão final de alterar os parâmetros.

Entre os trabalhos aplicando meta-heuristicas para sintonia de controladores PID destacam-se HERREROS *et al.* (2002), MAITELLI & CARVALHO (2003) e HU *et al*. (1999), que mesclam Algoritmos Genéticos para sintonia de controladores Fuzzy-PID.

VAROL & BINGUL (2003) utilizaram o algoritmo Colônia de Formigas para determinar os parâmetros do controlador, considerando-se como referência a resposta de um sistema de primeira ordem e uma função de avaliação RBEMCE (*Reference Based Error with Minimum Control Effort*), que estipula o mínimo esforço de controle para o sistema e minimiza também o erro do sistema.

PORTER & JONES (1992) apresentam o projeto e sintonia de controle PID via AGs para processos monovariáveis. KRÖHLING (1997) apresenta o projeto de controle PID para rejeição de distúrbios pela utilização de funções de penalização e dois AGs aplicados à otimização e cooperando entre si.

O problema de regulação de sistemas hidráulicos via AGs é tratada por NORDVIK & RENDERS (1991). O controle de um pêndulo invertido pela utilização de controle *on-off* é apresentado por KAWAJI *et al.* (1994). DIMEO & LEE (1994) apresentam o projeto e sintonia de controle PID MIMO aplicado a uma caldeira. FOGEL (1995) apresenta um procedimento para identificação e controle visando o equilíbrio de um pêndulo invertido utilizando Programação Evolutiva.

MARRISON & STENGEL (1997) tratam o projeto de compensadores em sistemas controle robusto aplicando métodos de busca aleatória e AGs. PATTON & LIU (1994) propõem o projeto de controladores robustos MIMO combinado AGs à otimização baseada em gradiente. O projeto de controle de estrutura variável por AGs é proposto por LIN & CHEN (1995). Projeto de controladores que satisfaçam o critério de H_2 / H_∞ em processos SISO (*Single-Input Single-Output*) (CHEN *et al.* 1995) e MIMO (PARAG & LEWIN, 1996) também têm sido tratados na literatura.

A Programação Genética tem sido tratada em problemas de controle ótimo que envolvem a descrição de sistemas dinâmicos por variáveis de estado, como o tratamento diferenciado do clássico problema de controle do pêndulo invertido (KOZA, 1992, 1994).

Dentre os esforços para se aplicar meta-heurística na solução do Problema de Sintonia dos Parâmetros do Controlador (PSPC) PID destaca-se o trabalho de GRIFFIN (2003), que desenvolveu um algoritmo no ambiente MatLab utilizando Algoritmos Genéticos para sintonia *on-line* dos parâmetros, sendo o estudo aplicado a um sistema Bola e Cilindro considerando-se o critério MSE para o índice de desempenho.

3.2 Índices de Desempenho e algoritmos

A função de avaliação J é um índice de desempenho, $J \in \Re;[0,\infty[$, que define o quão próximo do ótimo um sistema se encontra, podendo, até mesmo, ser o ótimo.

Um sistema de controle é dito ótimo se os parâmetros do controlador são escolhidos de forma a minimizar ou maximizar o índice de desempenho escolhido. O índice de desempenho deve proporcionar seletividade, ou seja, um ajuste ótimo deve ser claramente distinguido de um ajuste não ótimo, deve ser uma função dos parâmetros do sistema e deve possuir pelo menos um ponto de ótimo, seja de máximo ou de mínimo, local ou global.

Encontram-se na literatura critério que podem definir um sistema como ótimo. Os mais amplamente utilizados são os índices de desempenho dados pela integral de uma função ponderada do desvio da saída em relação à entrada ou referência.

Considerando-se $e(t) = r(t) - y(t)$ e o caso limite, onde $\lim_{t\to\infty} e(t) = 0$ implica em um sistema estável e com índice de desempenho finito (OGATA, 1993), pode-se estabelecer os índices de desempenho mais conhecidos.

3.2.1 Integral do Erro Absoluto ponderada pelo Tempo (ITAE)

Neste critério, o erro absoluto é ponderado pelo tempo, resultando que para um erro inicial grande há uma pequena ponderação, enquanto que um erro pequeno na resposta para os tempos finais é muito penalizado. Matematicamente o ITAE é assim definido:

$$I_{ITAE} = \int_0^\infty t|e(t)|dt \tag{20}$$

O algoritmo de avaliação do índice de desempenho para o ITAE é implementado, conforme mostrado na Tabela 1:

Tabela 1. Algoritmo ITAE

```
Y é vetor de reais; //resposta do sistema
```

```
i é inteiro; //índice
Erro é vetor de reais; //erro do sistema
ITAE é real; //índice de desempenho
Referência é vetor de reais; //referência
T é vetor de reais; //tempo
Tempo_Simulação é inteiro; //tamanho do vetor T

Y = Simula_Sistema(T[]);
Tempo_Simulação = Tamanho(T);
PARA i = 1 ATÉ Tempo_Simulação
   Erro[i] = abs(Referência[i] – Y[i])*T(i);
FIM

ITAE = Soma(Erro);
ITAE = 1/ITAE;
```

3.2.2 Integral do Erro Quadrático (ISE)

O critério ISE é bastante utilizado devido à facilidade de ser computado, além de proporcionar grandes ponderações para erros grandes e pequenas ponderações para erros pequenos. No entanto, ele apresenta pouca seletividade, sendo que as variações nos parâmetros do processo correspondem a pequenas variações no índice ISE. Sistemas implementados para minimizar este critério tendem a apresentar uma rápida resposta, mas com baixa estabilidade relativa, apresentando oscilações. Matematicamente o ISE é assim definido:

$$I_{ISE} = \int_0^{\infty} e^2(t)dt \tag{21}$$

O algoritmo de avaliação é implementado, conforme apresentado na Tabela 2:

Tabela 2. Algoritmo ISE

```
Y é vetor de reais; //resposta do sistema
i é inteiro; //índice
Erro é vetor de reais; //erro do sistema
```

```
ISE é real; //índice de desempenho
Referência é vetor de reais; //referência
T é vetor de reais; //tempo
Tempo_Simulação é inteiro; //tamanho do vetor T

Y = Simula_Sistema(T[]);
Tempo_Simulação = Tamanho(T);
PARA i = 1 ATÉ Tempo_Simulação
   Erro[i] = Referência[i] – Y[i];
   Erro[i] = Erro[i] * Erro[i];
FIM

ISE = Soma(Erro);
```

3.2.3 Integral Absoluta do Erro (IAE)

O critério IAE considera o módulo do erro, sendo muito utilizado em simulações, é de fácil implementação e entendimento, mas não apresenta muita seletividade quanto a variações dos parâmetros. Matematicamente o IAE é assim definido:

$$I_{IAE} = \int_0^{\infty} |e(t)| dt \tag{22}$$

O algoritmo de avaliação é implementado, conforme apresentado na Tabela 3:

Tabela 3. Algoritmo IAE

```
Y é vetor de reais; //resposta do sistema
i é inteiro; //índice
Erro é vetor de reais; //erro do sistema
IAE é real; //índice de desempenho
Referência é vetor de reais; //referência
T é vetor de reais; //tempo
Tempo_Simulação é inteiro; //tamanho do vetor T

Y = Simula_Sistema(T[]);
```

```
Tempo_Simulação = Tamanho(T);
PARA i = 1 ATÉ Tempo_Simulação
   Erro[i] = abs(Referência[i] – Y[i]);
FIM

IAE = Soma(Erro);
IAE = 1/IAE;
```

3.2.4 Média do Erro Quadrático (MSE)

O critério MSE reflete 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 (GRIFFIN, 2003) e Colônia de Formigas (VAROL & BINGUL, 2003). Matematicamente o MSE é assim definido:

$$I_{MSE} = \frac{1}{n}\sum_{i-1}^{n}\left(e(t)\right)^2 \tag{23}$$

O algoritmo de avaliação é implementado, conforme apresentado na Tabela 4:

Tabela 4. Algoritmo MSE

```
Y é vetor de reais; //resposta do sistema
i é inteiro; //índice
Erro é vetor de reais; //erro do sistema
MSE é real; //índice de desempenho
Referência é vetor de reais; //referência
T é vetor de reais; //tempo
Tempo_Simulação é inteiro; //tamanho do vetor T

Y = Simula_Sistema(T[]);
Tempo_Simulação = Tamanho(T);
PARA i = 1 ATÉ Tempo_Simulação
   Erro[i] = Referência[i] – Y[i];
   Erro[i] = Erro[i] * Erro[i];
FIM
```

```
MSE = Soma(Erro)/Tempo_Simulação;
```

3.2.5 Referência Baseado no Erro com Mínimo de Esforço de Controle (RBEMCE)

O critério RBEMCE leva em consideração um fator muito importante na estratégia de controle: o esforço. Entende-se por esforço de controle o somatório da saída do controlador durante um período n, ou seja, o quanto de energia foi despendido para o controle, visto que, fisicamente falando, a saída do controlador é, em sua maioria, um sinal de 4 a 20 mA que excita um atuador eletrônico e/ou pneumático, resultando em um gasto de energia. Em outras palavras, define-se:

$$I_{RBEMCE} = \frac{1}{n}\sum_{i=1}^{n} r(i) - y(i) + \frac{b}{n}\sum_{i=1}^{n} u(i) \tag{24}$$

O algoritmo de avaliação é implementado, conforme apresentado na Tabela 5:

Tabela 5. Algoritmo RBEMCE

```
Y é vetor de reais; //resposta do sistema
U é vetor de reais; //saída do controlador
β é real;
i é inteiro; //índice
Erro é vetor de reais; //erro do sistema
Esforço é vetor de reais; //esforço de controle
MSE é real; //índice de desempenho
RBEMCE é real; //índice de desempenho global
Referência é vetor de reais; //referência
T é vetor de reais; //tempo
Tempo_Simulação é inteiro; //tamanho do vetor T

[Y U] = Simula_Sistema(T[]);
Tempo_Simulação = Tamanho(T);
PARA i = 1 ATÉ Tempo_Simulação
   Erro[i] = Referência[i] – Y[i];
   Esforço[i] = U[i]
FIM
```

```
MSE = Soma(Erro)/Tempo_Simulação;
RBEMCE = MSE + (β/Tempo_Simulação)*Soma(Esforço)
```

3.2.6 Referência Baseada na Média do Erro Quadrático com Mínimo de Esforço de Controle (RBMSEMCE)

Para o presente trabalho propõe-se o seguinte critério de avaliação de desempenho: RBMSEMCE (*Reference Based in Mean Square Error with Minimum Control Effort*), que provê duas características importantes dos critérios anteriormente apresentados. Primeiramente ele avalia a resposta do sistema baseado no critério MSE, e refina a função de avaliação levando-se em consideração o esforço de controle. Em outras palavras, define-se:

$$I_{\text{RBMSEMCE}} = \frac{1}{n}\sum_{i=1}^{n}\left(e(i)\right)^2 + \frac{b}{n}\sum_{i=1}^{n}u(i) \tag{25}$$

O algoritmo de avaliação é implementado, conforme apresentado na Tabela 6:

Tabela 6. Algoritmo RBMSEMCE

```
Y é vetor de reais; //resposta do sistema
U é vetor de reais; //saída do controlador
β é real;
i é inteiro; //índice
Erro é vetor de reais; //erro do sistema
Esforço é vetor de reais; //esforço de controle
MSE é real; //índice de desempenho
RBEMCE é real; //índice de desempenho global
Referência é vetor de reais; //referência
T é vetor de reais; //tempo
Tempo_Simulação é inteiro; //tamanho do vetor T

[Y U] = Simula_Sistema(T[]);
Tempo_Simulação = Tamanho(T);
PARA i = 1 ATÉ Tempo_Simulação
   Erro[i] = Referência[i] – Y[i];
```

```
   Erro[i] = Erro[i] * Erro[i];
   Esforço[i] = U[i]
FIM

MSE = Soma(Erro)/Tempo_Simulação;
RBEMCE = MSE + (β/Tempo_Simulação)*Soma(Esforço)
```

3.3 MÉTODOS DE SINTONIA EMPÍRICOS

Muitos métodos de sintonia de controladores se baseiam em conhecimentos adquiridos sobre uma planta específica. Outras se baseiam no estudo de respostas de sistemas de ordem inferior (normalmente primeira e segunda ordem) e extrapolam para sistemas de ordem superior.

Dentre os métodos mais utilizados está o proposto por Ziegler e Nichols que observaram que os parâmetros do controlador podem ser determinador pelo conhecimento de um ponto da curva de Nyquist do sistema em malha aberta. Muitos métodos surgiram para melhorar o desempenho deste método, mas todos possuem muita semelhança com o originalmente proposto. Segue uma descrição detalhada do método original.

3.3.1 Método de Ziegler-Nichols em malha fechada

As regras propostas por Ziegler & Nichols são bastante utilizadas em Sistemas de Controle para determinação dos valores do ganho proporcional *Kp*, do tempo integral *Ti* e do tempo derivativo *Td*, baseadas nas características da resposta transitória de uma dada planta, obtidas experimentalmente.

Há dois métodos para determinação dos parâmetros, um para plantas que possuem curva de resposta ao degrau em malha aberta na forma de “S” e outro para plantas que são instáveis em malha fechada, com o aumento do ganho de um

controlador proporcional. Ambos os métodos visam à obtenção de 25% de sobre-sinal máximo na resposta ao degrau, (OGATA, 1993).

O primeiro método consiste em obter experimentalmente a resposta do processo a uma entrada degrau. Quando a planta não envolve integradores nem pólos dominantes complexos conjugados (o que implica em oscilações na resposta ao degrau), a curva de resposta será na forma de "S". Se a resposta do sistema não apresentar este comportamento o método não se aplica.

A resposta do sistema pode ser representada por duas constantes, o tempo de retardo, L, e a constante de tempo, T. Estes parâmetros podem ser determinados a partir do ponto de inflexão da curva, passando por ele uma reta tangente e determinando as interseções com o eixo dos tempos e com a reta c(t)=K, conforme assinalado na figura abaixo.

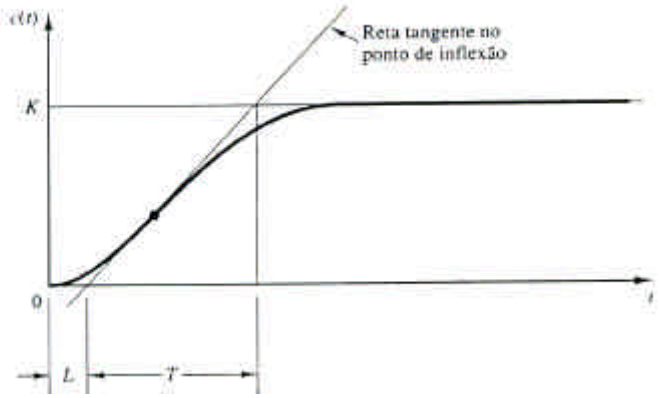

Figura 3.1. Curva de resposta em forma de "S". Fonte: OGATA, 1993

ZIEGLER-NICHOLS (1942) sugeriram ajuste dos parâmetros do controlador de acordo com o mostrado na Tabela 7:

Tabela 7. Método de ZIEGLER-NICHOLS para obter os parâmetros do controlador com resposta em "S"

	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0{,}9\frac{T}{L}$	$\frac{L}{3}$	0
PD	$1{,}2\frac{T}{L}$	$2L$	$0{,}5L$

No segundo método, os parâmetros do controlador em malha fechada são obtidos da seguinte maneira: um controlador proporcional deve ser colocado no

sistema e seu ganho deve ser ajustado até que se observem oscilações periódicas e de amplitude constante. Quando esta condição é atingida o ganho utilizado no controlador proporcional é o ganho crítico e o período da oscilação é o período crítico.

Para o sucesso deste método é necessário que o sistema em questão não apresente integradores. Em outras palavras, sistemas que possuem integradores (em termos da transformada de Laplace, implica na função $\frac{1}{s}$ multiplicada pela função de transferência) tendem a zerar o erro quando $t \to \infty$. É importante ressaltar que muitos sistemas se comportam desta maneira, limitando a aplicação deste método.

Obtido o ganho crítico e o período crítico pode-se determinar os parâmetros do controlador, Tabela 8:

Tabela 8. Método de ZIEGLER-NICHOLS para obter os parâmetros do controlador em malha fechada

	K_p	T_i	T_d
P	$\frac{K_{CR}}{2}$	∞	0
PI	$\frac{K_{CR}}{2,2}$	$\frac{t_{CR}}{1,2}$	0
PD	$\frac{K_{CR}}{1,7}$	$\frac{t_{CR}}{2}$	$\frac{t_{CR}}{8}$

A verdadeira utilidade do método de ZIEGLER-NICHOLS (1942) está quando não se conhece a dinâmica do processo a controlar, de modo a se poder utilizar abordagens analíticas, gráficas ou até computacionais para se projetar os controladores.

3.4 META-HEURÍSTICAS

As meta-heurísticas são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de

uma heurística subordinada, a qual tem que ser modelada para cada problema específico (RIBEIRO, 1996).

Contrariamente às heurísticas convencionais, as meta-heurísticas são de caráter geral e têm condições de escapar de ótimos locais. As meta-heurísticas, assim como os métodos de busca local tradicionais, diferenciam-se entre si basicamente pelas seguintes características (SOUZA, 2005):

- critério de escolha de uma solução inicial;
- definição da vizinhança N(s) de uma solução s;
- critério de seleção de uma solução vizinha dentro de N(s);
- critério de término.

3.4.1 Algoritmos Baseados em *Simulated Anneling*

TAN *et al*. (2001) definem *Simulated Annealing* (SA) como sendo uma técnica de relaxação estocástica que encontra sua origem em mecanismos estatísticos. A metodologia do SA é análoga ao processo de recozimento dos sólidos. Para evitar os estados meta-estáveis produzido pelo resfriamento em água, metais são esfriados muito lentamente, permitindo a eles tempo para tornarem-se estáveis, estruturalmente fortes, com baixas energias de configuração. Este processo é chamado recozimento. Esta analogia pode ser usada em otimização combinatória com os estados dos sólidos correspondendo a soluções factíveis, a energia em cada estado ao melhoramento da função objetivo e a energia mínima sendo a solução ótima. SA envolve um processo em que a temperatura é gradualmente reduzida durante a simulação. Geralmente, o sistema é primeiro aquecido e então resfriado. Assim, ao sistema é dada a oportunidade para superar as barreiras energéticas em uma busca por conformações com energias mais baixas que a energia mínima local. SA é uma heurística de otimização global baseado em probabilidade, então, sendo hábil a superar ótimos locais.

TAN *et al*. (2001) citam, ainda, algumas vantagens e inconvenientes advindos da utilização desta técnica, sendo:

- SA pode ser tratado com sistemas arbitrários e funções de custo;

- SA estatisticamente garante encontrar uma solução ótima;
- SA é relativamente fácil para codificar, mesmo para problemas complexos;
- SA geralmente dá uma boa solução.
- Os inconvenientes citados por TAN et al. (2001) são:
- Repetir o recozimento com uma função $\frac{1}{\log k}$ (onde k= constante de Boltzman) é muito lenta, especialmente se a função custo é cara para calcular.
- Para problemas onde a curva de energia é suave, ou há poucos mínimos locais, SA é uma "dose excessiva" – métodos simples e rápidos trabalharão melhor. Mas usualmente a curva de energia não é conhecida.
- Métodos heurísticos normais, que são específicos a um dado problema e tomam vantagem de informações extras sobre o problema, geralmente serão melhores que métodos gerais. No entanto, SA é freqüentemente competitivo com estas heurísticas clássicas.
- O método não pode dizer se a solução encontrada é uma solução ótima. Alguns outros métodos (ex.: Relaxação Lagrangeana) são requeridos para fazer isto.

3.4.2 *Simulated Annealing* proposto

O algoritmo SA proposto parte de uma boa solução inicial factível. Tal solução é obtida através de um algoritmo guloso de busca local partindo de uma solução aleatória. As diversas temperaturas utilizadas são então inicializadas.

Seja *S* uma solução do problema (parâmetros do controlador) e $f(S)$ o índice de desempenho da simulação do sistema com *S*, define-se a variação de energia $\Delta = f(S') - f(S)$ onde *S*'=vizinho da solução *S*. Se Δ for menor que zero, a

solução *S*’ é aceita, caso contrário, se Δ≥0 e $e^{-\Delta/T} \geq \Theta$ (número randomicamente escolhido entre 0 e 1), a solução *S*’ também será aceita. A temperatura T corrente então é “resfriada” pela fórmula, considerando t como taxa de resfriamento:

$$T_k = \frac{T_{k-1}}{1 + t\sqrt{T_{k-1}}} \tag{26}$$

e a busca é repetida IterMax vezes até que $T_k = T_f = 0{,}1$ (parâmetro definido pelo autor).

IV. METODOLOGIA

Neste capítulo são apresentadas as metodologias utilizadas para resolver o problema, bem como a forma de representação da solução, a obtenção de uma solução inicial, a estrutura de vizinhança, a obtenção da temperatura inicial, a função de avaliação e a descrição detalhada do algoritmo utilizado.

A meta-heurística escolhida para desenvolver o algoritmo foi o *Simulated Annealing* pela sua simplicidade e capacidade de encontrar boas soluções para problemas genéricos. Vários índices de desempenho foram implementados e a modelagem do problema é apresentada com foco no controle.

4.1 O Controlador PID

Para a implementação do controlador PID uma classe base foi criada e a partir dela mais três classes foram herdadas. Esta classe, denominada cl_Parametro, também implementa a geração do vizinho da solução, sendo portanto, representativa geral do parâmetro.

O controlador_KP é a classe que define o comportamento do controlador proporcional. O K_p deve se situar na faixa entre 0,0 e 2,0 %/%. O algoritmo do controlador foi implementado, conforme a equação (10), Tabel 9:

Tabela 9. Algoritmo Controlador Proporcional

```
Kp é real; //valor do parâmetro proporcional
sp é real; //valor do set-point
pv é real; //valor da variável de processo
_beta é real; //Ponderação da referência

Método Saida(sp,pv)
INÍCIO
        retorna Kp*(_beta*sp - pv);
FIM
```

O controlador_KI é a classe que define o comportamento do controlador integral. O K_i deve se situar na faixa entre 0,01 e 500,0 rep/seg. O reset-windup é realizado uma camada acima na estrutura de classes, sendo que apenas a saída é calculada, conforme a equação (13), Tabela 10:

Tabela 10. Algoritmo Controlador Integral

```
Ki é real; //valor do parâmetro integral
sp é real; //valor do set-point
pv é real; //valor da variável de processo
I_anterior é real; //Saida anterior
Ts é real; //periodo de amostragem

Método Saida(sp,pv)
INÍCIO
        saida = I_anterior+(Ts*Ki)*(sp-pv);
        I_anterior = saida;
        retorna saida;
FIM
```

O controlador_KP define o comportamento do controlador derivativo. Este controlador está implementado na forma de equação ISA, ou seja, utiliza-se o parâmetro Tempo Derivativo ao invés de Ganho Derivativo, e o ganho proporcional também entra no calculo da saída. O T_d deve se situar na faixa entre 0,01 e 10,0 seg. Nota-se que a taxa de variação é calculada sobre a saída da planta, e não sobre o erro, conforme explicitado na equação (14), Tabela 11:

Tabela 11. Algoritmo Controlador Derivativo

```
Kp é real; //valor do parâmetro proporcional
Td é real; //valor do parâmetro derivativo
sp é real; //valor do set-point
pv é real; //valor da variável de processo
PV_anterior é real; //pv anterior
D_anterior é real; //saida anterior
Ts é real; //periodo de amostragem
polo_filtragem é real; //filtro de freqüência

Método Saida(sp,pv)
```

```
INÍCIO
        saida = D_anterior;
        saida *= 1-(Ts*polo_filtragem/Td);
        saida  -=  Kp*polo_filtragem*(PV_anterior
- pv);
        PV_anterior = pv;
        D_anterior = saida;
        Retorna Saida;
FIM
```

4.2 Representação da Solução do Problema

Considera-se como a solução do problema o conjunto dos três valores dos parâmetros do controlador PID. A classe cl_PID implementa as funcionalidades do controlador e é responsável pelo cálculo da saída do mesmo, além de simular a saturação do atuador, com limites definidos de 0 a 100%, e de realizar *o anti-reset-windup* no controlador integral quando a saída do controlador for superior à faixa de –500 a + 500 (parâmetros definidos pelo autor), conforme proposto por ISERMANN (1989).

4.3 Determinação da Solução Inicial

Com a finalidade de se gerar uma solução inicial, criou-se um procedimento capaz de gerar soluções iniciais factíveis. Este procedimento executa uma “busca gulosa” por novas soluções a partir de uma solução aleatória, ou seja, a cada movimento, um vizinho será aceito como solução somente se melhorar o índice de desempenho em relação à solução atual. São executadas 500 iterações para se determinar uma solução inicial para o SA.

4.4 Estrutura de Vizinhança

O movimento utilizado para geração de uma solução *S*', vizinha à *S*, é um movimento de incremento nos parâmetros. A utilização de parâmetros do tipo real para a representação do problema implica significativamente na performance do algoritmo.

Como forma de representação da solução do problema, considerando S o conjunto de possíveis soluções, temos:

$$S = \{s = (K_p, K_i, T_d) : K_p \in \Re, K_i \in \Re \text{ e } T_d \in \Re) \} \quad (27)$$

Definiu-se para o *Simulated Annealing* o operador de geração da vizinhança de s, N(s), tal que:

$$N(s) = \{s' = (K_p', K_i', T_d') : s' = s \oplus m \} \quad (28)$$

onde s'= (K_p', K_i', T_d') e m = $(\Delta K_p, \Delta K_i, \Delta T_d)$, sendo que, dos três parâmetros acima referidos $(\Delta K_p, \Delta K_i, \Delta T_d)$, somente um é diferente de zero e seu valor pode ser calculado, em cada caso pelas seguintes fórmulas, considerando r um número aleatório entre –5,0 e +5,0:

$$\Delta K_p = \pm 0{,}01 * r * \text{Max}(K_p) \quad (29)$$

$$\Delta K_i = \pm 0{,}01 * r * \text{Max}(K_i) \quad (30)$$

$$\Delta T_d = \pm 0{,}01 * r * \text{Max}(T_d) \quad (31)$$

4.5 O Algoritmo *Simulated Annealing*

A partir da solução inicial gerada para o PSPC, é aplicado o algoritmo SA proposto por Souza (2004). O procedimento principal deste algoritmo consiste de

um *loop* que gera aleatoriamente, em cada iteração, um único vizinho *s'* da solução corrente *s*.

Para o PSPC, seja Δ a variação de valor da função objetivo ao mover-se para uma solução vizinha candidata, isto é, $\Delta = f(s') - f(s)$. O método aceita o movimento e a solução vizinha passa a ser a nova solução corrente se $\Delta \geq 0$. Caso $\Delta > 0$ a solução vizinha candidata também poderá ser aceita mas, neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de temperatura e que regula a probabilidade de se aceitar soluções de pior custo.

A temperatura T assume, inicialmente, um valor elevado T_0 estipulado em 500,0 (parâmetro definido pelo autor). Após um número fixo de iterações (o qual representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma função de resfriamento (equação (25)). Com esse procedimento, dá-se, no início uma chance maior para escapar de mínimos locais e, à medida que T aproxima-se de zero, o algoritmo comporta-se como um método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora ($T \rightarrow 0 \Rightarrow e^{-\Delta/T} \rightarrow 0$).

O procedimento pára quando a temperatura chega a um valor próximo de zero e nenhuma solução de piora da solução corrente é mais aceita, isto é, quando o sistema está estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local.

Os parâmetros de controle do procedimento são a razão de resfriamento $t = 0{,}15$ e o número de iterações para cada temperatura (SAmax=90) e a temperatura inicial T_0.

Aa Tabela 12 apresenta-se o pseudo-código para o procedimento SA.

Tabela 12. Pseudo-Código para o procedimento SA

```
Procedimento SA (f(.), N(.), a, SAmax, T0, s)
INÍCIO
s* = s; //melhor solução obtida até então
IterT = 0;//Número de iterações na temperatura T
T = T0; //Temperatura Corrente
ENQUANTO (T > 0) faça
  ENQUANTO (IterT <SAmax) FAÇA
```

```
    IterT IterT+1;
    Gere um vizinho qualquer s' ∈ N(s);
    D = f(s')-f(s);
    SE (D<0) ENTÃO
       s = s';
       SE (f(s')<f(s*)) ENTÃO
           s* = s';
       FIM-SE
    SENÃO
       Tome x ∈ [0, 1];
       SE (x < exponencial(-D/T)) ENTÃO
          s = s';
       FIM-SE;
    FIM-SENÃO;
  FIM-ENQUANTO;
  Resfria(t);
  IterT = 0;
FIM-ENQUANTO;
s = s*;
Retorna s;
FIM
```

Resumidamente, o algoritmo SA aplicado pode ser descrito da seguinte forma:

- Passo 1: Gerar uma solução inicial s;
- Passo 2: Aplicar o SA ao problema;
- Passo 3: Verificar os resultados.

4.6 Ambiente de Simulação

Para o desenvolvimento do projeto, um ambiente de simulação denominado SA_PID foi elaborado. Normalmente, os trabalhos que utilizam algoritmos complexos para a solução de problemas NP - Difíceis não apresentam uma interface gráfica amigável para o usuário. Para o presente trabalho foi implementada uma plataforma onde é possível se definir alguns parâmetros da

simulação e visualizar a evolução do algoritmo AS, bem como os resultados alcançados. Um arquivo com os dados relevantes da simulação pode ser salvo, caso o usuário ache necessário, Figura 4.1.

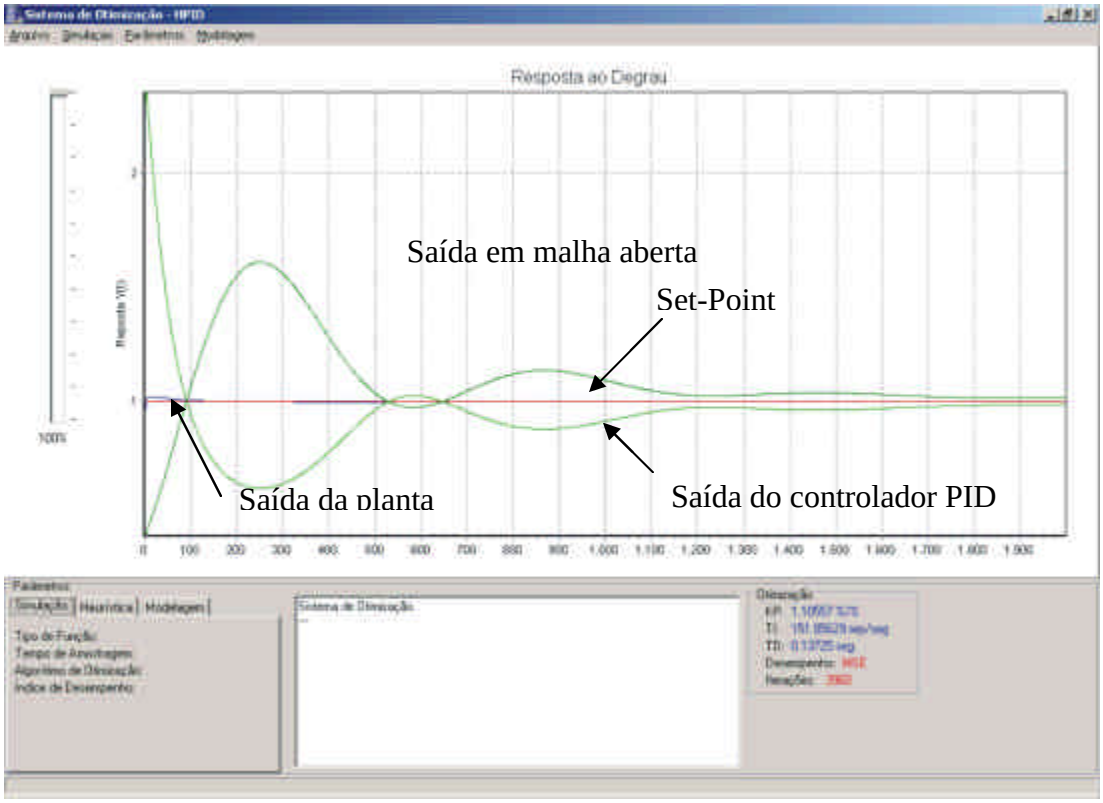

Figura 4.1. Janela principal do ambiente de desenvolvimento

Neste exemplo a linha vermelha representa o *Set-Point*, a linha verde escuro representa a saída em malha aberta da planta, a linha verde claro representa a saída do controlador (esforço de controle) e a linha azul é a saída da planta após o ajuste dos parâmetros do controlador PID.

Pode-se observar, no canto inferior direito, os parâmetros finais da simulação, bem como o índice de desempenho utilizado e o numero total de iterações do algoritmo SA.

É possível também, por meio do *software*, alterar o índice de desempenho utilizado para a simulação, Figuras 4.2 e 4.3.

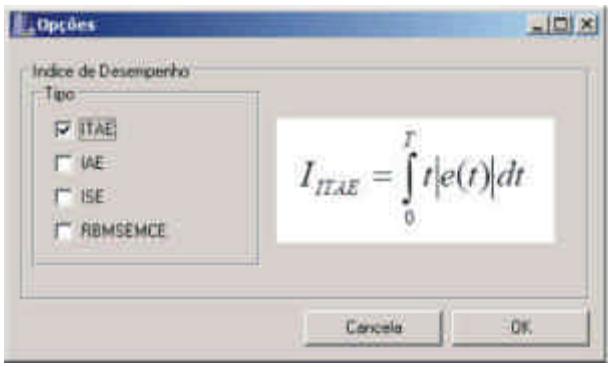

Figura 4.2. Definição do Índice de Desempenho a ser utilizado.

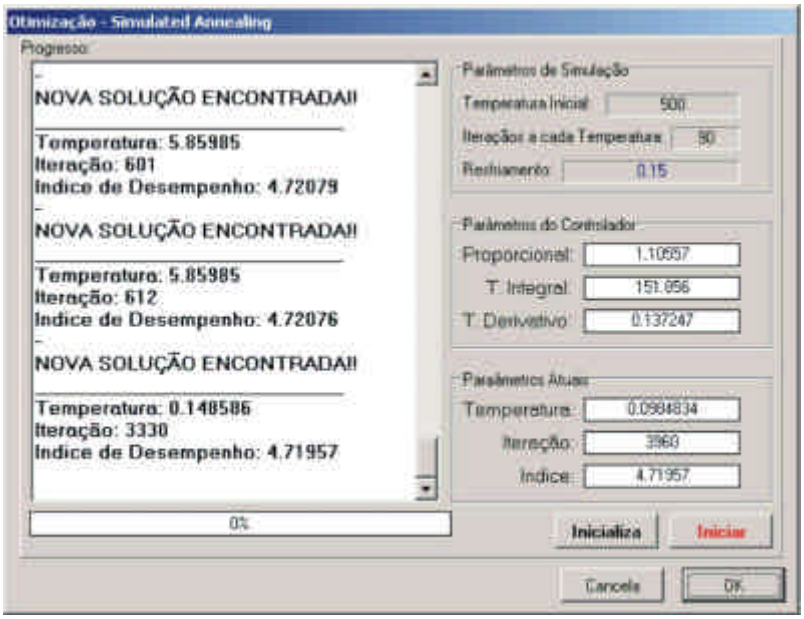

Figura 4.3. Janela de simulação

4.7 Função de Transferência

Para a simulação de um sistema físico, sua função de transferência deve ser modelada. A Função de Transferência define como o sistema transforma a variável de entrada na variável de saída, utilizando para isso a energia cedida ao sistema.

No presente trabalho foi utilizada uma função de transferência que exigia um esforço de controle maior, mas que não é representativa de um sistema físico real. A não linearidade imposta pela raiz quadrada restringe os métodos de

sintonia a linearizar o sistema em um ponto de operação e depois aplicar a metodologia, fato este que não é necessário dentro do ambiente de simulação proposto, tratando-se, assim, a não-linearidade de forma direta.

A Função de Transferência temporal utilizada pode ser visualizada abaixo:

$$\frac{y(t)}{u(t)} = A + 0.31 \cdot \sqrt{e^{-0,31 \cdot t}} + 0.25 * e^{-0,31 \cdot t} * \sin(t) + 0.9 * e^{-0,31 \cdot t} * \cos(t) \qquad (32)$$

onde A define a proporção do set-point (SP) no qual o processo se estabiliza, ou seja:

$$\lim_{t \to \infty} y(t) = A * SP \qquad (33)$$

Para as simulações foram utilizados A = 1,0 e A=0,8, além de variações desta formula que serão demonstradas na apresentação dos resultados.

Com a ajuda do MatLab, pode-se calcular a Transformada de Laplace de Y(t). A não-linearidade apresentada é linearizada, resultando no seguinte polinômio em s, Figuras 4.4 e 4.5:

$$\frac{Y(s)}{U(s)} = \frac{0,41 s^5 + 1,053 s^4 + 2,472 s^3 + 1,992 s^2 + 1,78 s + 0,1862}{s^6 + 1,395 s^5 + 2,769 s^4 + 1,759 s^3 + 1,412 s^2 + 0,1862 s} \qquad (34)$$

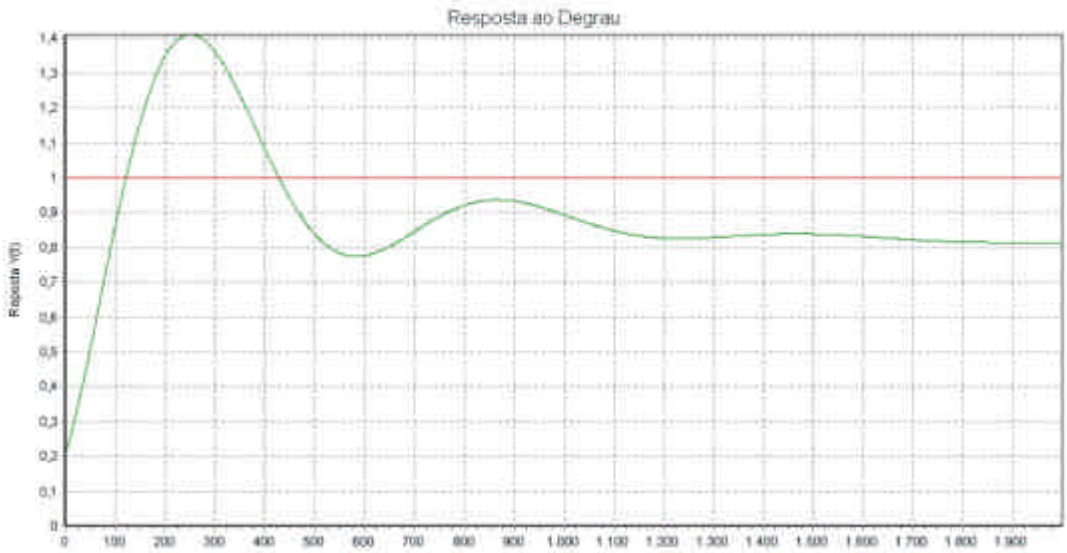

Figura 4.4. Resposta da Planta para A=0,8

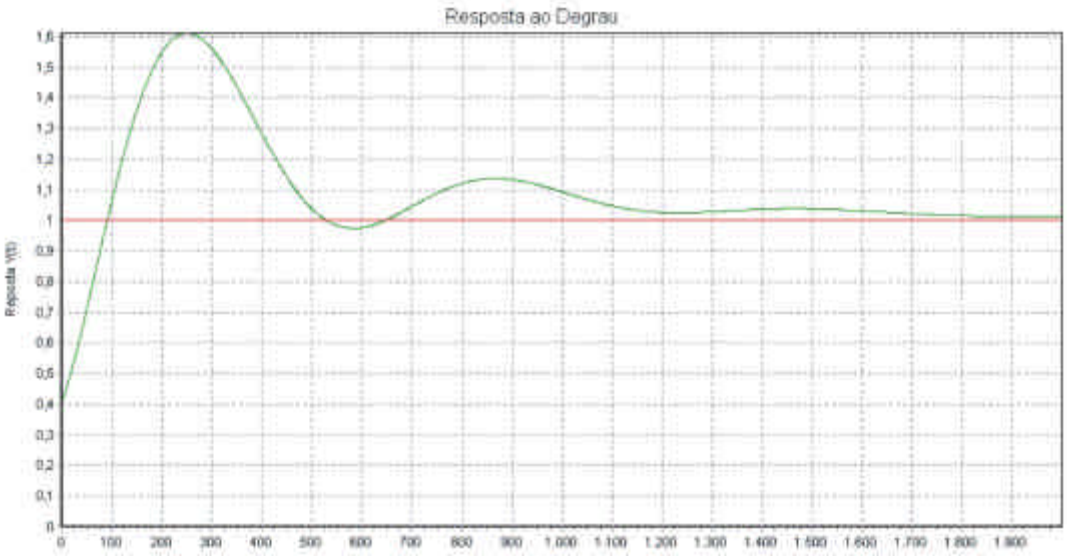

Figura 4.5. Resposta da Planta para A=1,0

4.8 Parâmetros de Controle

Alguns parâmetros de controle foram definidos para a simulação dos problemas, Tabela 3:

Tabela 13. Parâmetros de Simulação

Nome	Descrição	Valor
Período de Amostragem	Define o período de amostragem do valor de saída da planta	0,01 s
Tempo de Simulação	Define o tempo máximo da simulação	2,0 s
Set-Point	*Set-Point* da malha de controle	1,0
Temperatura Inicial	Define a temperatura inicial para o SA	500,0º
Taxa de resfriamento	Define a taxa de resfriamento do SA	0,15
Temperatura Final	Define a temperatura final para o SA parâmetro.	0,1º
Iterações	Define o numero de iterações do SA em uma dada temperatura.	90
Ponderação do Termo Proporcional	Define o termo de ponderação do controlador proporcional, conforme equação (10)	0,80

Taxa de incremento da saída	Define a porcentagem da saída calculada que é passada para o processo, evitando variações bruscas no atuador.	70%
Pólo de Filtragem do derivativo	Define o pólo de filtragem de alta freqüência para o controlador derivativo, conforme equação (8).	1,0
Ponderação do Índice de Desempenho	Define o valor da ponderação do índice de desempenho na função de avaliação.	30,0 * Set - Point
Ponderação do Esforço de Controle	Define o valor da ponderação do esforço de controle na função de avaliação.	$\frac{5,0}{\text{Set - Point}}$
Pólo de Filtragem do Derivativo	Define o pólo de filtragem de alta freqüência para o controlador derivativo, conforme equação (8).	1,0

4.9 Diagrama de Blocos do Sistema

O diagrama de blocos de um sistema permite uma visão mais detalhada do processo e de suas partes constituintes. Para o problema apresentado, pode-se visualizar a interligação dos diversos módulos na Figura 4.6:

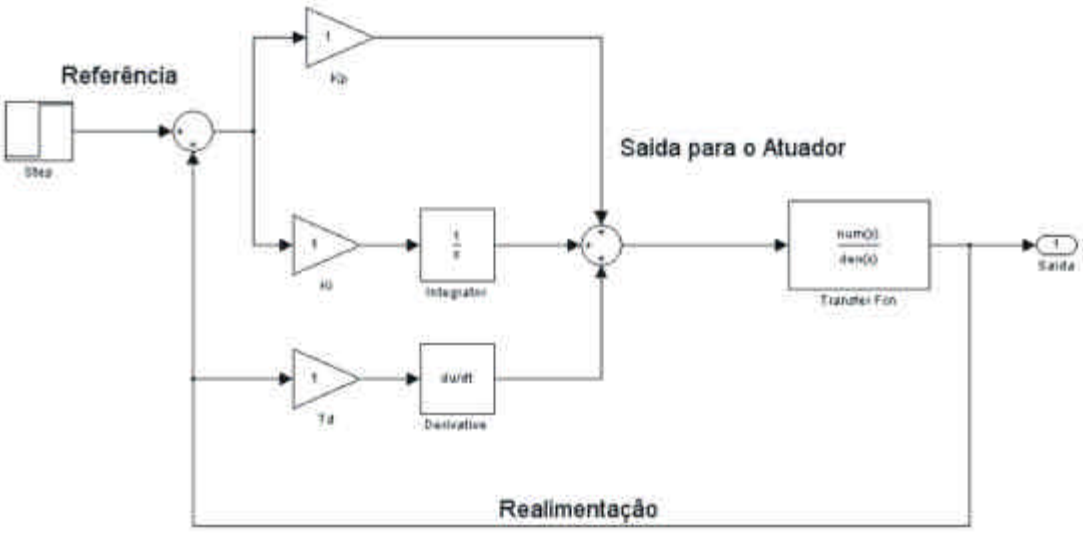

Figura 4.6. Diagrama de Blocos do sistema

V. RESULTADOS OBTIDOS

Os algoritmos propostos foram desenvolvidos em linguagem C++, utilizando-se o compilador C++ Builder da Borland, e executados em um computador AMD® Atlon XP 2400+ com 512 MB de memória RAM, sob plataforma Windows XP Pro.

5.1 APRESENTAÇÃO DOS RESULTADOS

Foram realizadas 5 simulações para cada configuração do sistema, sendo definido o desvio padrão e a média do erro como meio de comparação dos resultados. As configurações divergem entre si em vários aspectos, sempre em relação a configuração original apresentada na tópico 5.1.1.

5.1.1 Simulação do Sistema 01

Para esta simulação utilizaram-se os parâmetros de configuração do sistema apresentados na Tabela 14 e os resultados são apresentado na Tabela 15, bem como a convergência dos parâmetros (Figura 5.1, 5.2 e 5.3).

Tabela 14. Configuração básica do Sistema 1

Parâmetro	Figura 1.2 Valor
Set-Point	Figura 1.3 1,0
A	0,80
Ponderação do Índice de Desempenho	30,0
Ponderação do Esforço de Controle	5,0
Ponderação do Termo Proporcional	0,8

Pólo Derivativo	1,0
Temperatura Inicial	500,0
Taxa de resfriamento	0,15
Taxa de incremento da saída	70%
Índice de Desempenho	RBMSEMCE

Tabela 15. Resultados da Simulação do Sistema 1

	Média Erro	**Desvio Padrão**	**Esforço**
Simulação 1	-0,0005	0,0112	1,1806
Simulação 2	-0,0005	0,0106	1,1807
Simulação 3	-0,0005	0,0346	1,1747
Simulação 4	-0,0005	0,0098	1,1808
Simulação 5	-0,0005	0,0100	1,1808
Média	**-0,0005**	**0,0152**	**1,1795**

Parâmetros

	KP	**KI**	**TD**
Simulação 1	Kp = 1,10974	Ki = 249,279	Td = 0,124353
Simulação 2	Kp = 1,32747	Ki = 270,589	Td = 0,204967
Simulação 3	Kp = 1,68068	Ki = 30,6292	Td = 0,0657573
Simulação 4	Kp = 0,625629	Ki = 306,771	Td = 0,555779
Simulação 5	Kp = 1,32427	Ki = 304,727	Td = 0,156199

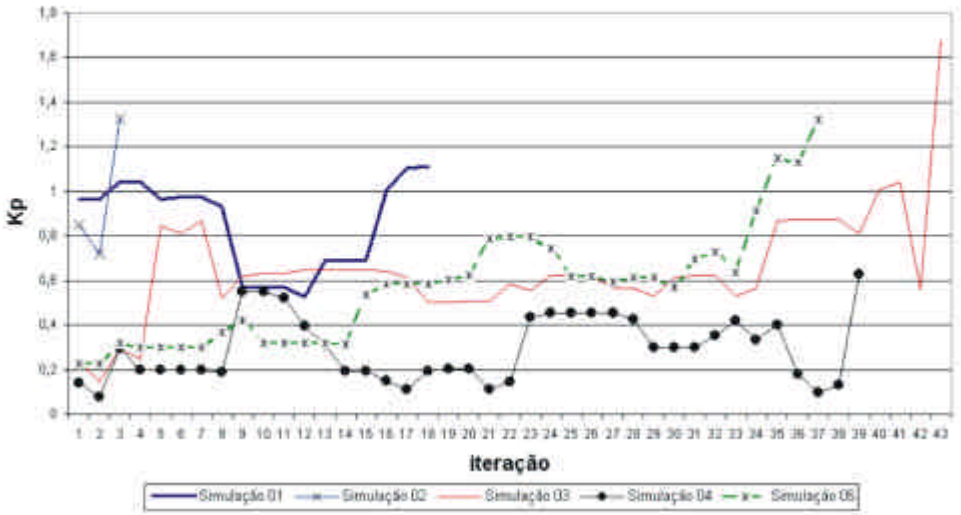

Figura 5.1 Convergência do parâmetro K_p do Sistema 1

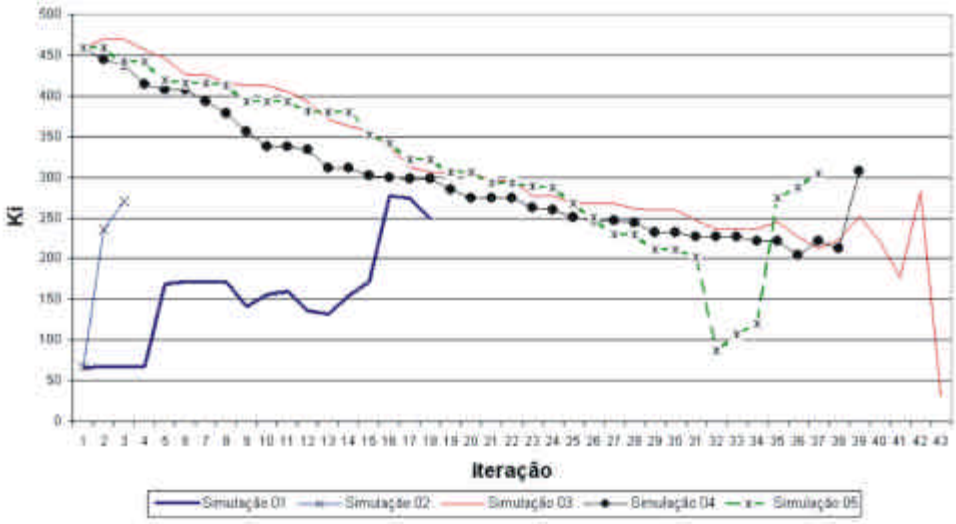

Figura 5.2 Convergência do parâmetro Ki do Sistema 1

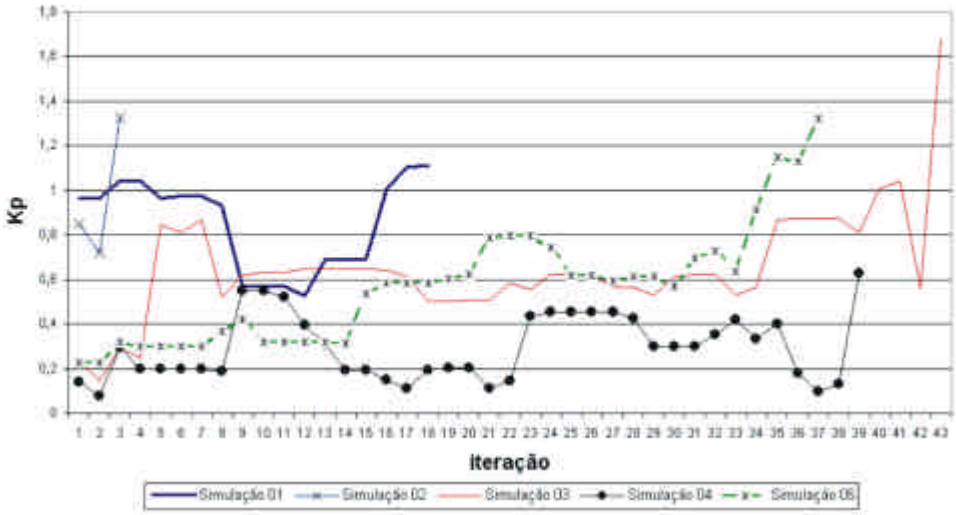

Figura 5.3 Convergência do parâmetro Td do Sistema 1

5.1.2 Simulação do Sistema 2

Para esta simulação a função de transferência , chamada de Sistema 2, é definida como:

$$\frac{y(t)}{u(t)} = A + 0.91 \cdot \sqrt{0{,}1 * e^{-0{,}21 \cdot t}} + 0.1 * e^{-0{,}21 \cdot t} * \sin(0{,}22 * t) + 0.9 * e^{-0{,}21 \cdot t} * \cos(2{,}2 * t) \quad (35)$$

cuja resposta pode ser visualizada na Figura 5.4.

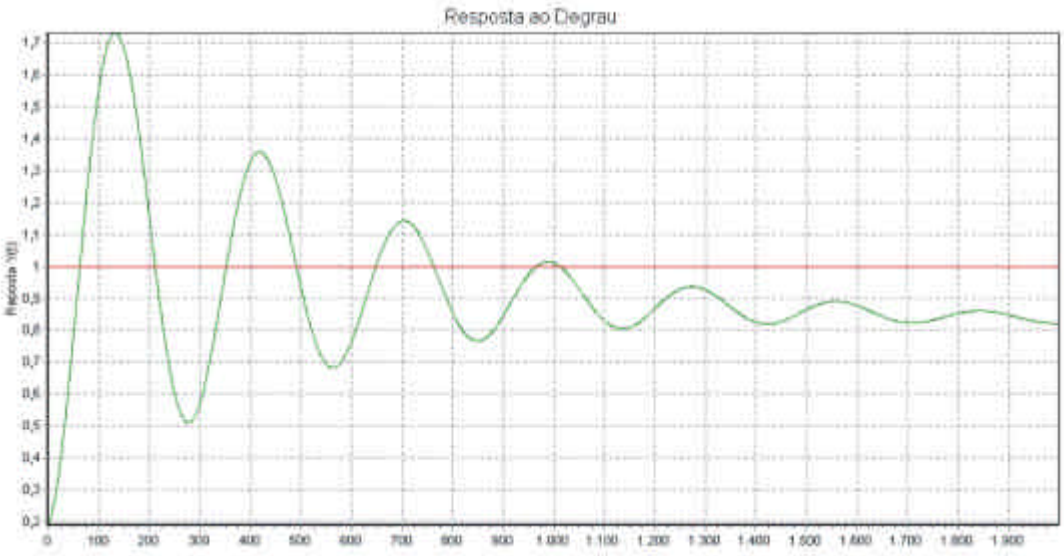

Figura 5.4 Resposta do Sistema 2

Os parâmetros da simulação são iguais aos apresentados anteriormente, menos pela Ponderação do Termo Proporcional, Tabela 16. Os resultados são apresentados na Tabela 17, bem como a convergência dos parâmetros (Figura 5.5, 5.6 e 5.7).

Tabela 16. Configuração básica de simulação do Sistema 2

Parâmetro	Figura 1.4 Valor
Ponderação do Termo Proporcional	0,19
Índice de Desempenho	RBMSEMCE

Tabela 17. Resultados da Simulação Básica do Sistema 2

	Média Erro	**Desvio Padrão**	**Esforço**
Simulação 1	-0,0005	0,0129	1,1639
Simulação 2	-0,0005	0,0134	1,1638
Simulação 3	-0,0005	0,0124	1,1640
Simulação 4	-0,0005	0,0124	1,1640
Simulação 5	-0,0005	0,0133	1,1638
Média	**-0,0005**	**0,0129**	**1,1639**

	Parâmetros		
	KP	**KI**	**TD**
Simulação 1	Kp = 1,21162	Ki = 255,409	Td = 0,114022
Simulação 2	Kp = 0,133815	Ki = 245,164	Td = 1,58412
Simulação 3	Kp = 1,1585	Ki = 269,504	Td = 0,0886157
Simulação 4	Kp = 1,55938	Ki = 269,969	Td = 0,140757
Simulação 5	Kp = 1,27386	Ki = 246,859	Td = 6,92166

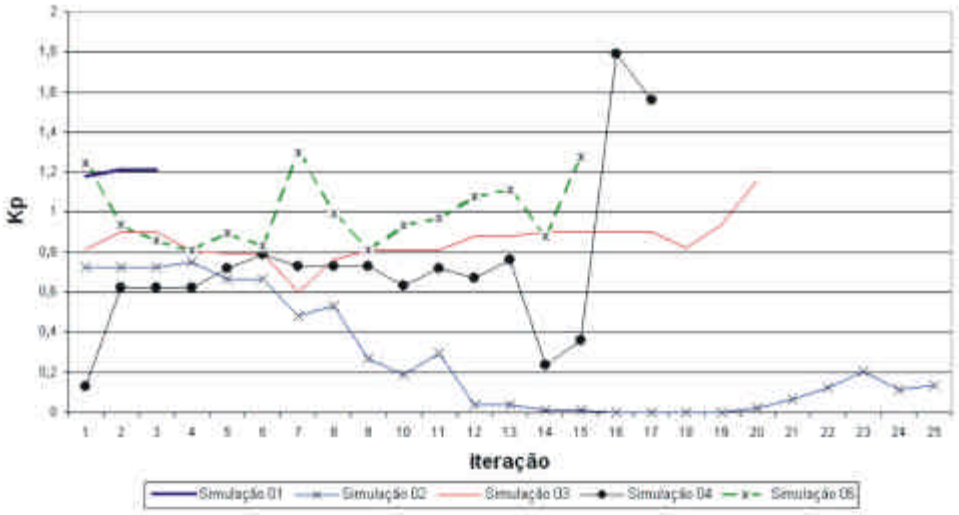

Figura 5.5 Convergência do parâmetro Kp do Sistema 2

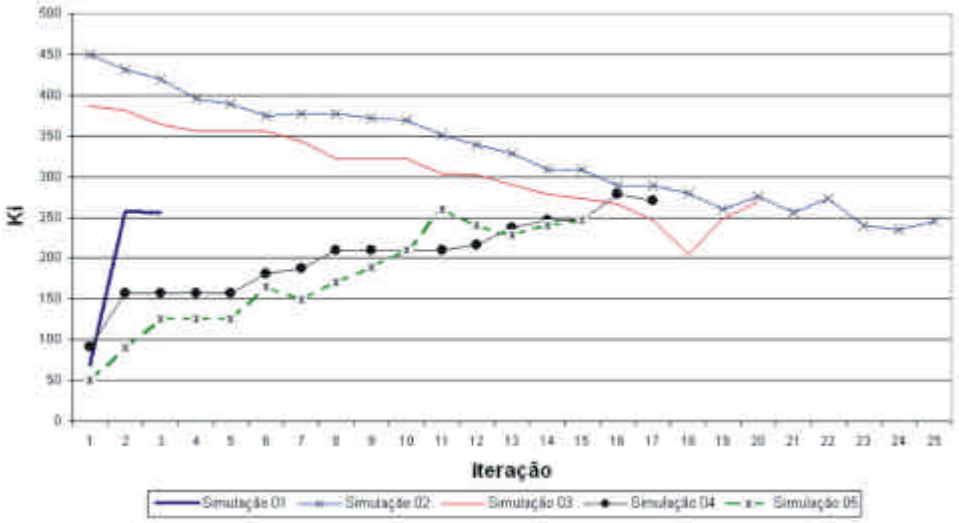

Figura 5.6 Convergência do parâmetro Ki do Sistema 2

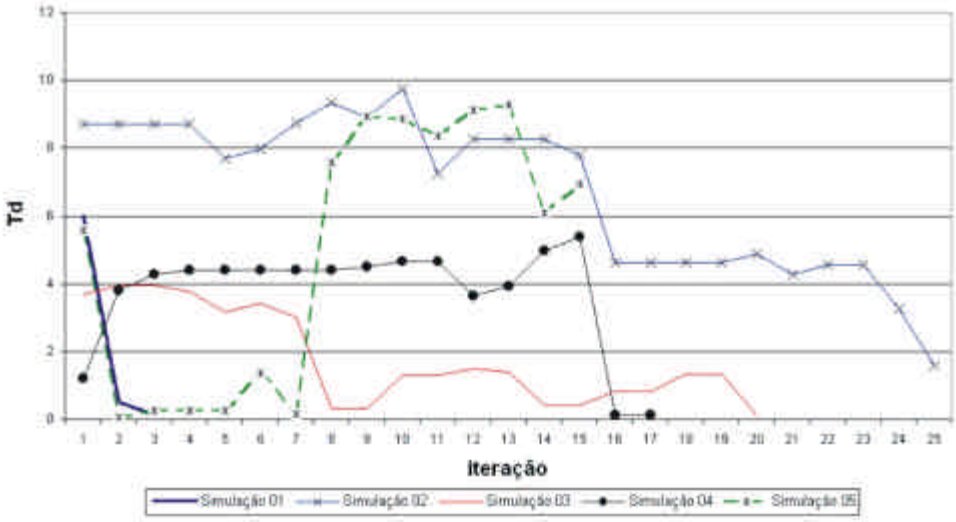

Figura 5.7 Convergência do parâmetro Td do Sistema 2

5.1.3 Simulação do Sistema 3

Para esta simulação a função de transferência , chamada de Sistema 3, é definida como:

$$\frac{y(t)}{u(t)} = A + 0.11 \cdot \sqrt{e^{-0,31 \cdot t}} + 0.4 * e^{-0,31 \cdot t} * \sin(0,8 * t) + 0.4 * e^{-0,31 \cdot t} * \cos(0,8 * t) \qquad (36)$$

cuja resposta pode ser visualizada na Figura 5.8.

Os parâmetros da simulação são iguais aos apresentados anteriormente, menos pela Ponderação do Termo Proporcional, Tabel 18. Os resultados são apresentados na Tabela 19, bem como a convergência dos parâmetros (Figura 5.9, 5.10 e 5.11).

Tabela 18. Configuração básica de simulação do Sistema 3

Parâmetro	Figura 1.5 Valor
Set-Point	Figura 1.6 1,0
A	1,1
Ponderação do Índice de Desempenho	30,0

Ponderação do Esforço de Controle	15,0
Ponderação do Termo Proporcional	0,9
Pólo Derivativo	1,0
Índice de Desempenho	RBMSEMCE

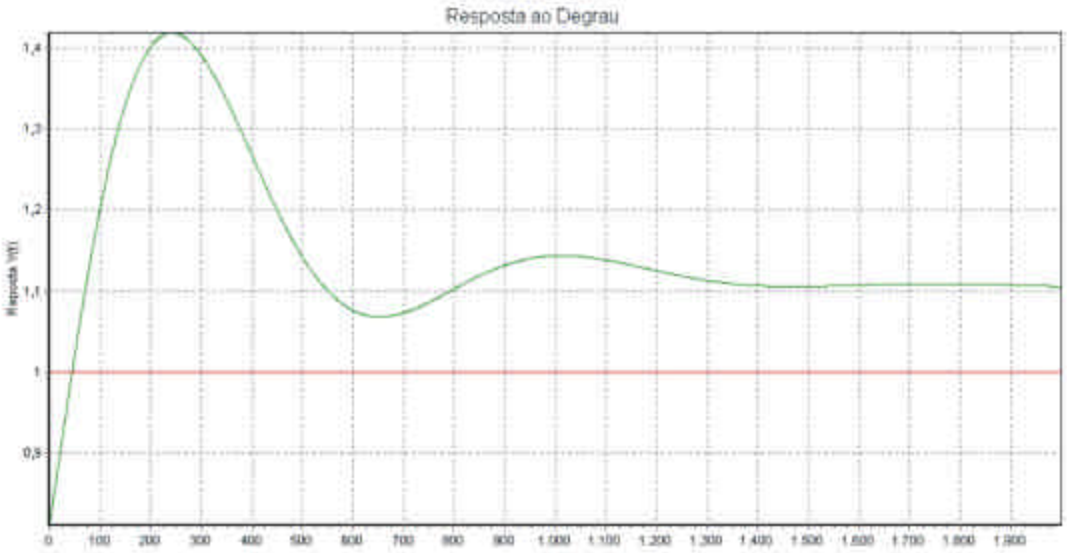

Figura 5.8 Resposta do Sistema 3

Tabela 19. Resultados da Simulação Básica do Sistema 3

	Média Erro	**Desvio Padrão**	**Esforço**
Simulação 1	-0,0005	0,0129	1,1639
Simulação 2	-0,0005	0,0134	1,1638
Simulação 3	-0,0005	0,0124	1,1640
Simulação 4	-0,0005	0,0124	1,1640
Simulação 5	-0,0005	0,0133	1,1638
Média	**-0,0005**	**0,0129**	**1,1639**

Parâmetros

	KP	**KI**	**TD**
Simulação 1	Kp = 1,21162	Ki = 255,409	Td = 0,114022
Simulação 2	Kp = 0,133815	Ki = 245,164	Td = 1,58412
Simulação 3	Kp = 1,1585	Ki = 269,504	Td = 0,0886157
Simulação 4	Kp = 1,55938	Ki = 269,969	Td = 0,140757
Simulação 5	Kp = 1,27386	Ki = 246,859	Td = 6,92166

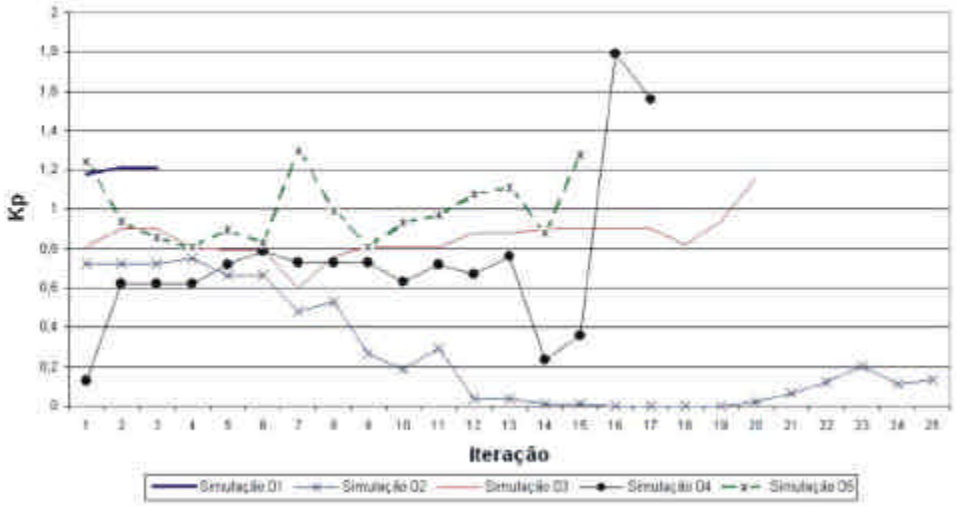

Figura 5.9 Convergência do parâmetro Kp do Sistema 3

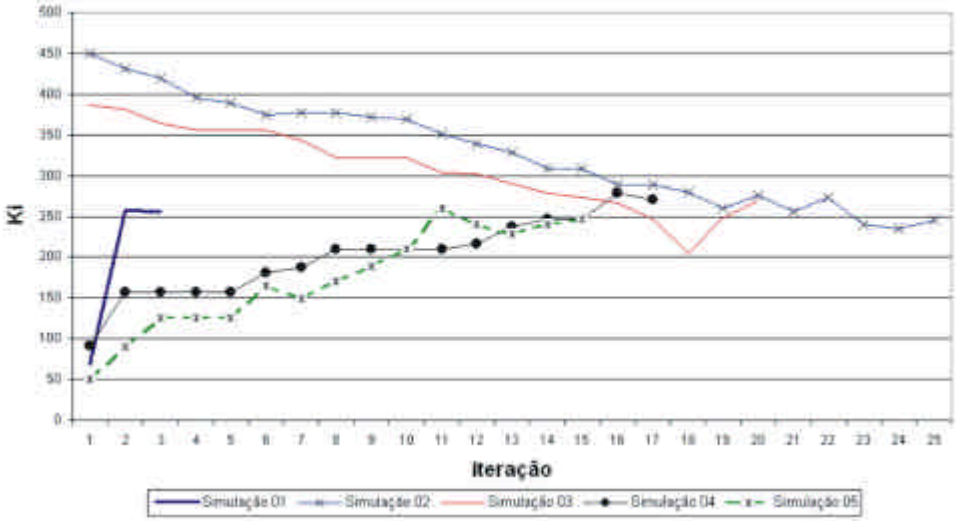

Figura 5.10 Convergência do parâmetro Ki do Sistema 3

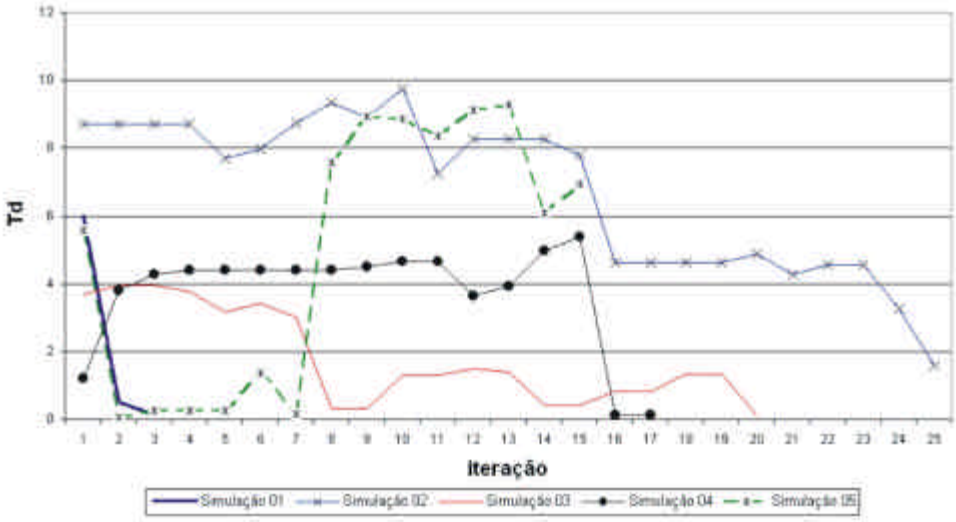

Figura 5.11 Convergência do parâmetro Td do Sistema 3

5.2 ANÁLISE DOS RESULTADOS

Conforme se observa nas tabelas 15, 17 e 19, o algoritmo SA básico comportou-se bem com os problemas apresentados, conseguindo-se alcançar a média de -0,0005 para o erro em todas as simulações.

Para o Sistema 01 a dificuldade apresentada para o algoritmo SA estava no fato da planta apresentar erro em regime permanente de 20% abaixo do valor do *set-point*. Para superar isso o controle deve manter uma saída acima da referência em regime permanente, garantindo a equalização do sistema. Observa-se também um ajuste para que a saída do controlador represente a inversa da função de transferência da planta, o que garante o desacoplamento, estado este que é causa da minimização da média do erro quadrático (MSE).

O algoritmo apresentou uma variação na terceira simulação do sistema 01, onde a média do erro foi a mesma, mas o desvio padrão aumentou, tendo como uma boa conseqüência a redução do esforço de controle, mas mesmo assim garantindo uma boa resposta do sistema, Figura 5.12.

A simulação 4 apresentou os melhores resultados globais, tendo como parâmetros K_p = 0.625629, K_i = 306.771 e T_d = 0.555779.

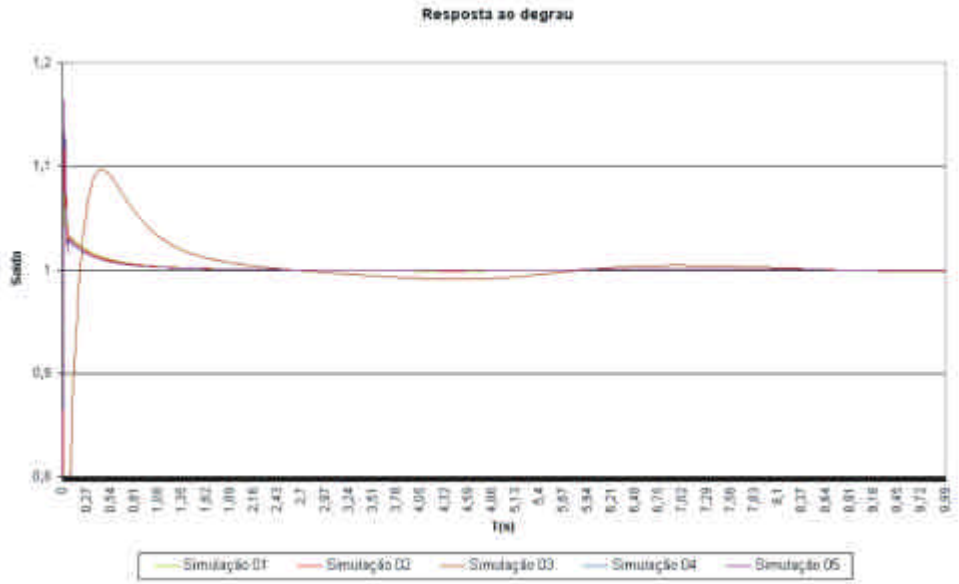

Figura 5.12 Respostas das simulações do sistema 1

Para o Sistema 02, a ponderação do termo proporcional foi reduzida para 0,19, forçando a saída do controlador proporcional para uma valor menor. As dificuldades apresentadas para o algoritmo SA estavam no erro em regime permanente de 20% abaixo do valor do *set-point* e na baixa estabilidade do sistema, que apresentava grandes oscilações antes de estabilizar abaixo do valor desejado. Observa-se também o desacoplamento do sistema para os parâmetros do PID encontrados pelo algoritmo.

O algoritmo apresentou uma baixa variabilidade nos resultados, garantindo uma boa solução nas cinco simulações realizadas, Figura 5.13.

Para o terceiro sistema a saída desta vez estabilizava 10% acima do valor desejado, exigindo do controlador a regulagem abaixo da referência para o regime permanente. O parâmetro de ponderação do esforço de controle igual a 15,0 garantiu a operação mais suave do sistema, apresentando o menor esforço de controle e ainda um desvio padrão do erro baixo.

Sistemas com as características do sistema 03 representam um desperdício de energia ao manterem um erro em regime permanente acima do desejado, sendo que, para os parâmetros encontrados para o controlador, conseguiu-se uma boa resposta para o sistema, com *overshot* máximo de 0,05, Figura 5.14.

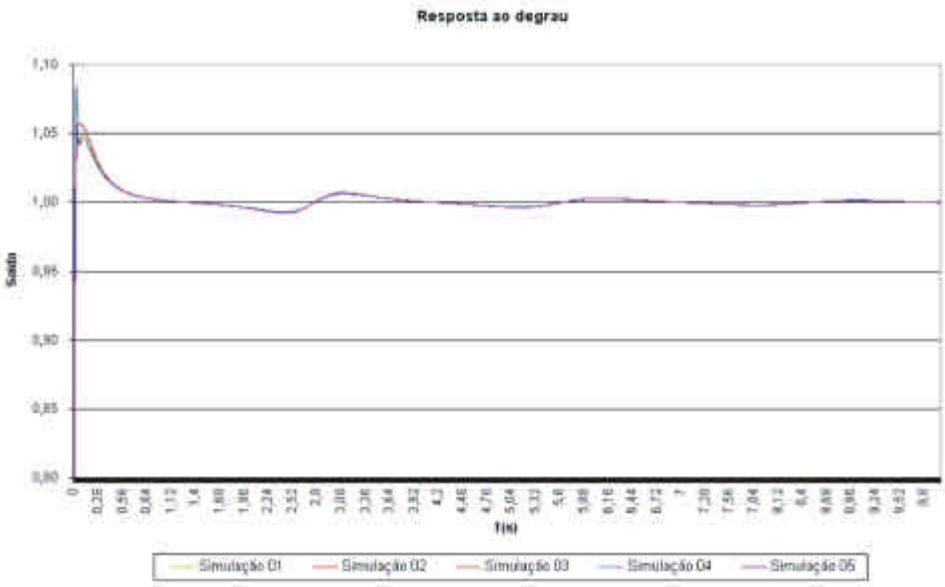

Figura 5.13 Respostas das simulações do sistema 2

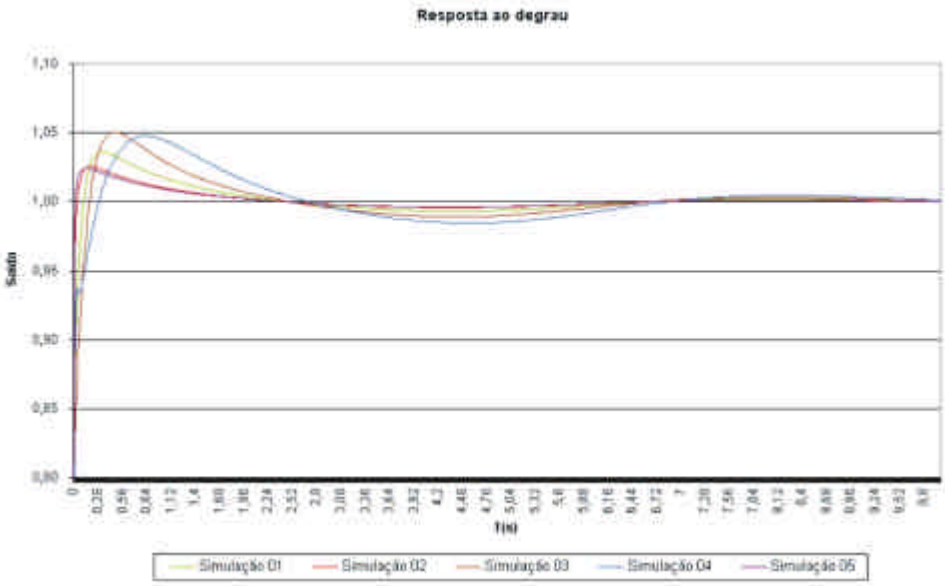

Figura 5.14 Respostas das simulações do sistema 3

VI. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho contribui com o desenvolvimento de metodologias heurísticas para resolver o Problema de Sintonia de Parâmetros de Controladores PID. O algoritmo *Simulated Annealing* (SA) se mostrou bastante eficiente para encontrar boas soluções e resposta rápida mesmo para o elevado numero de iterações necessárias para simulação do sistema (2000 para cada avaliação). O tempo médio de execução do algoritmo é de 20 segundos para todas as simulações apresentadas.

O método de busca gulosa para a determinação de uma boa solução inicial foi de grande ajuda para o SA convergir rapidamente para boas soluções.

Como trabalho futuro, aponta-se a inclusão de um módulo de reconhecimento de sistemas para determinar a resposta da planta, tornando o sistema independente da função de transferência e tratando a resposta do problema o mais próximo possível da resposta real da planta.

VII. REFERÊNCIAS BIBLIOGRÁFICAS

ASTRÖM, K. J.; T. HÄGGLUND. PID Controllers: Theory, Design, and Tuning, Instrument Society of America, 1995.

OGATA, K. Engenharia de Controle Moderno. Ed. Prentice-Hall, 1993.

STEPHANOPOULOS, G. Chemical Process Control: an introduction to theory and practice, Ed. Prentice-Hall, 1984.

ISERMANN, R. Digital Control Systems, v. 1, Springer-Verlag, 1989.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum Settings for Automatic Controllers, ASME Transaction, v. 64, p. 759-768, Nov. 1942.

CAON, J. R. Controladores PID Industriais com Sintonia Automática por Realimentação a Relé. USP, 1999.

LEVA, A. PID Autotuning Algorithm based on Relay Feedback, IEE Proceedings-D, v. 140, n. 5, p 328-338, 1993.

NISHIKAWA, Y; SONNOMIYA, N; OHTA, T; TANAKA, H. A Method for Auto-Tuning of PID Control Parameters, Automatica, v. 20, n. 3, p 321-332, 1984.

WANG, L; BARNES, T. J. D.; CLUETT, W. R. New Frequency-Domain Design Method for PID Controllers. IEE Proc.- Control Theory Appl., v. 142, n. 4, p 265-271, 1995.

VODA, A. A., LANDAU, I. D. A Method for the Auto-calibration of PID Controllers, Automatica, v. 31, n. 1, p 41-53, 1995.

KHAN, B. Z.; LEHMAN, B. Setpoint PI Controllers for System with Large Normalized Time, IEEE Transc. On Control Syst. Technology, v. 4, n. 4, p 459-400, 1996.

POULING, E.; POMERLEAU, A.; DESBIENS, A.; HODOUIN, D. Deployment and Evaluation of an Auto-Tuning and Adaptive PID Controller, Automatica, v. 32, n. 1, p 71-82, 1996.

FUNG, H. W; WANG, Q. G.; LEE, T. H. PI tuning in Terms of Gain and Phase Margins, Automatica, vol. 34, No. 9, pp. 1145-1149, 1998.

VAROL, H. A.; BINGUL, Z. A New PID Tuning Technique Using Ant Algorithm, University of Sabanci and Kocaeli, Turkey, 2003.

GRIFFIN, I. On-line PID ControllerTuning using Genetic Algorithms, DCU, ID: 98072358, 2003.

HERREROS, J. M.; BLASCO, M.; MARTÍNEZ, M.; SALCEDO, J. V. Optimal PID Tuning With Genetic Algorithms For Non Linear Process Models, 15th Trienal World Congress, 2002.

MAITELLI, A. L; CARVALHO, F. C. A. Programa Computacional Interativo para Simulação e otimização de controladores PID, COBENGE, 2003.

PORTER, B. & A.H. JONES. Genetic Tuning of Digital PID Controllers. Electronics Letters, v.28, n. 9, p. 843- 844, 1992.

KRÖHLING, R. Design of PID Controller for Disturbance Rejection: A Genetic Optimization Approach. Proc. of 2nd Int. Conf. on GA in Eng. Systems: Innovations and Applications, Glasgow, U.K., p. 498-503, 1997.

NORDVIK, J.-P. & J.-M. RENDERS. Genetic Algorithms and their Potential for Use in Process Control: A Case Study. Proc. of the 4th Int. Conf. on Genetic Algorithms, R.K. Belew (ed), Morgan-Kauffman, San Mateo, CA, USA, p. 480-486, 1991.

KAWAJI, S., K. OGASAWARA & H. HONDA. Swing up Control of a Pendulum using Genetic Algorithms. Proc. of the 33rd Conf. on Decision and Control, Lake Buena Vista, FL, USA, p. 3530-3532, 1994.

DIMEO, R.M. & K.Y. LEE. Genetic-Based Control of MIMO Boiler-Turbine Plant. Proc. of the 33rd Conf. on Decision and Control, Lake Buena Vista, FL, USA, pp. 3512-3517. SBA Controle & Automação, v. 27, n. 01, 1994.

FOGEL, D.B. An Introduction to Simulated Evolutionary Optimization. IEEE Trans. on Neural Networks, v. 5, n. 1, p. 3-14, 1994.

CHIPPERFIELD, A. & P. FLEMING. Evolutionary Algorithms for Control Engineering. 13th Triennial World Congress IFAC, San Francisco, CA, USA, p. 181-186, 1996.

MARRISON, C.I. & R.F. STENGEL. Robust Control System Design using Random Search and Genetic Algorithms. IEEE Trans. on Automatic Control, v. 42, n. 6, p. 835-839, 1997.

PATTON, R.J. & G.P. LIU. Robust Control Design via Eigenstructure Assignment, Genetic Algorithms and Gradient-based Optimisation. IEE Proc.-Control Theory Appl., v. 141, n. 3, p. 202-207, 1994.

LIN, S.-C. & Y.-Y. CHEN. A GA-Based Fuzzy Controller with Sliding Mode. Proc. of the Int. Joint Conf. of the 4th Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Eng. Symposium, Yokohama, Japan, vol. III, p. 1103-1110, 1995.

PARAG, A. & D.R. LEWIN. Decentralized Control System Synthesis using a Genetic Algorithm. 13th Triennial World Congress IFAC, San Francisco, CA, USA, p.193-198, 1996.

KOZA, J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, 1992.

RIBEIRO, C. C. Metaheuristics and Applications. In Advanced School on Artificial Intelligence, Estoril, Portugal, 1996.

TAN, K.C.; LEE, L. H.; ZHU, Q. L.; OU, K. (2001) Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering, v. 15, p. 281-295

SOUZA, M. J. F. Inteligência Computacional para Otimização: Notas de aula. UFOP, Ouro Preto, 2005, Disponível em http://www.decom.ufop.br/prof/marcone/InteligenciaComputacional/InteligênciaComputacional.ps. Acesso em 01/04/2005.

GLOSSÁRIO

Algoritmos Genéticos	Trata-se de uma meta-heurística que se fundamenta em uma analogia com processos naturais de evolução, nos quais, dada uma população, os indivíduos com características genéticas melhores têm maiores chances de sobrevivência e de produzirem filhos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer. Para mais detalhes ver Souza (2005).
Função ou procedimento guloso	É uma função ou um procedimento que escolhe o melhor elemento, segundo uma função de avaliação, dentre um conjunto de candidatos.
Heurística	Heurística é uma técnica que procura boas soluções (próximas da otimalidade) a um custo computacional razoável, sem, no entanto, estar capacitada a garantir a otimalidade, bem como garantir quão próximo uma determinada solução está da solução ótima. (SOUZA, 2005)
Meta-heurística	Meta-heurística são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual tem que ser modelada para cada problema específico. (SOUZA, 2005)
Otimalidade	Qualidade do que é ótimo.
Relaxação Lagrangeana	Método exato de resolução de um problema de programação linear, consistindo na decomposição do problema em diversos subproblemas.
Vizinhança de uma solução	A vizinhança de uma solução s é o conjunto de soluções s' que diferem de s por um movimento.
Malha Aberta	Processo sem laço de realimentação, ou seja, sem nenhuma ação de controle baseada no amostragem da saída.
Malha Fechada	Processo com laço de realimentação, sendo possível a ação de controle baseada na amostragem da saída.

Ordem do Sistema	Grau do polinômio do denominador, resultante da transformada de Lapalace da função de transferência da planta.
Entrada Degrau	Excitação da planta quando a entrada varia de uma unidade subitamente.
Sobre-Sinal ou overshot	Maior valor alcançado pela variável de processo, considerando-se o sistema estável.
Tempo de Subida	Tempo necessário para a variável de processo alcançar 90% do valor final.
Tempo de Acomodação	Tempo necessário para a variável de processo estabilizar em torno de 2% do valor final.
Zero	Raiz do polinômio do numerador da função de transferência da planta.
Pólo	Raiz do polinômio do denominador da função de transferência da planta.