

TESE

1095

EFEI

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

**Projeto e Ajuste Computacional de Controladores de
Sistemas Dinâmicos pelo Método das Inequações**

(Dissertação de Mestrado)

JUAN CARLOS BEJARANO LUJAN

ITAJUBÁ - MG - BRASIL

Dezembro de 2000

Projeto e Ajuste Computacional de Controladores de Sistemas Dinâmicos pelo Método das Inequações

Dissertação submetida à
ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
como requisito parcial para obtenção do grau de
Mestre em Ciências em Engenharia Elétrica

por

JUAN CARLOS BEJARANO LUJÁN

Orientador: Prof. Dr. Carlos Alberto Dias Coelho (EFEI)

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
ITAJUBÁ - MG - BRASIL
Dezembro de 2000

À minha amada esposa Rita, pela compreensão, paciência e dedicação inabaláveis.

Aos meus filhos Pablo Andres e Juan José, pelo constante incentivo.

Aos meus pais, pela bondade e amor, pela fé e confiança depositadas em mim e, principalmente, pelo exemplo de vida.

Agradecimentos

À ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ - EFEI, pela valiosa oportunidade de cursar o mestrado.

À COORDENAÇÃO DE APERFEIÇOAMENTO DE PESSOAL DE NÍVEL SUPERIOR – CAPES, por ser a entidade fornecedora dos meios materiais necessários para alcançar o título de Mestre.

Ao professor Carlos Alberto Dias Coelho, por sua paciência, compreensão, inestimável orientação e competência no desenvolvimento deste trabalho.

Aos professores dos Departamentos de Eletrônica e Eletrotécnica, pelos conhecimentos ministrados.

Aos colegas Gustavo Iribarren, Alexandre Aoki, pela amizade, ajuda e incentivo.

E acima de tudo a Deus.

RESUMO

O Método das Inequações (MI) é uma técnica de projeto assistido por computador de sistemas de controle. O MI está baseado no princípio de que os problemas de projeto de controladores de sistemas dinâmicos tem o objetivo de satisfazer especificações de desempenho e restrições de operação, as quais podem ser formuladas através de um conjunto de inequações.

A meta básica do projeto é a determinação de um conjunto de valores dos parâmetros do controlador que faça com que o sistema satisfaça as inequações que definem as especificações de desempenho e as restrições de operação. Estes parâmetros são determinados através de um algoritmo computacional que varia e ajusta estes valores automaticamente até que todas as inequações sejam satisfeitas simultaneamente.

Este trabalho descreve a estruturação e a solução computacional do problema de projeto de controladores em cascata, para sistemas de controle univariáveis com realimentação unitária, pelo Método das Inequações. A especificação básica de que o sistema deva ser estável é expressa por uma inequação que limita a abscissa de estabilidade do sistema a valores negativos e não-nulos, e as especificações de desempenho são expressas por inequações que impõem limites para as características da resposta a degrau, que são os índices de desempenho utilizados; as restrições de operação são expressas por inequações que confinam a variação de parâmetros dentro de faixas estabelecidas pelo projetista, e de uma inequação que limita a amplitude do sinal de controle. Esta última inequação tem a dupla função de limitar o esforço de controle e de estabelecer a condição de validade do modelo linear do sistema controlado, no caso deste sinal estar sujeito a algum tipo de saturação.

Para a solução dos problemas de controle que se enquadram na estrutura considerada foi desenvolvido um conjunto de rotinas e comandos computacionais, os quais formam um programa interativo, que torna a tarefa de projeto muito rápida e eficiente, e no qual

também foram incluídos alguns recursos que proporcionam também grande facilidade para avaliação dos resultados. A aplicação do MI e a eficácia do programa desenvolvido são demonstradas através de dois exemplos bem documentados. No capítulo de conclusões são destacadas as contribuições oferecidas à área de controle neste trabalho e as vantagens de se projetar sistemas de controle por este método. Também são mencionadas as possíveis alternativas de aplicação do MI a problemas de controle em que os índices de desempenho utilizados sejam de outros domínios que não o do tempo, e da extensão da aplicação do método a problemas de controle mais amplos e mais elaborados, tais como aqueles relativos a sistemas multivariáveis e não-lineares.

ABSTRACT

The Method of Inequalities (MI) is a technique of computer-aided design of control systems. The MI is based on the principle that the problem of designing controllers for dynamic systems is aimed at satisfying performance specifications under operation constraints, both of which can be formulated as a set of inequations.

The design basic goal is determining a set of values of the controller parameters that make the system to satisfy the inequalities that define the performance specifications and the operation constraints. These parameters are determined by a computational algorithm, which varies and sets those values automatically until all inequalities are simultaneously satisfied.

This work describes the framework and the computational solution of the problem of designing cascade controllers for Single-Input-Single-Output control systems with unity feedback, by the Method of Inequalities. The basic specification that the system must be stable is expressed by an inequality which constrains the abscissa of stability to negative nonzero values, and the performance specifications are expressed by inequalities that specify limits to the step response characteristics, which are the performance indices. The operation constraints are expressed by inequalities that confine the parameter variations within ranges prescribed by the designer, and by an inequality which limits the control signal magnitude. This last inequality has the double role of limiting the control effort and establishing the condition for the controlled plant linear model to be valid, in case this signal is subject to some kind of saturation.

For the solution of control problems that fit into the considered framework, a set of computer routines and commands was developed, in order to form an interactive program that makes the design procedure very fast and effective. This set of routines and commands also include some other facilities that enable a fast and effective analysis of the design results. The application of the MI and the program effectiveness are shown by means of two well-documented examples. In the Conclusions chapter the

contributions offered by this work to the control field and the advantages of designing control systems by this method are highlighted. Mentions are also made to possible alternative applications of the MI to control problems where the performance indices are of domains other than time, and to extending the application of the method to larger and more elaborate control problems, such as those involving multivariable and nonlinear systems.

PROMUNCIAMENTO DA BANCA EXAMINADORA

Banca Examinadora, criada especial, nomeada pela Portaria nº 342 de 14/2000, avaliando o resultado do Julgamento de Prova de Defesa Pública de Dissertação de Mestrado intitulada "Projeto e Análise Computacional de Controladores de Sistemas Não-Lineares pelo Método das Funções" apresentada pessoalmente no sentido de que o candidato dos Cursos de Pós-Graduação em Engenharia, Lúcio de Fátima Fátima de Aguiar, de origem sob o DPA (Departamento de Projetos Acadêmicos) e aprofundado o título de Mestre em Ciências em Engenharia Elétrica, na Área de Concentração em Controle e Sistemas Elétricos Industriais, mediante as demais condições previstas, a Juan Carlos Bejarano Luján.

Brasília, 18 de dezembro de 2000.

Prof. José Manoel Fernandes
1º Coordenador - UFU

Prof. Pedro Manoel Cavalho Mendes
2º Coordenador - UFU

Prof. Carlos Alberto Das Costa
3º Coordenador - UFU



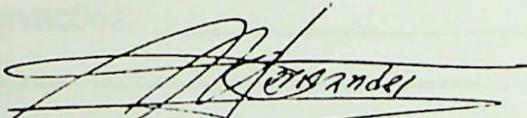
Ministério da Educação
ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
Reconhecida Lei 3232 - 05/01/1917

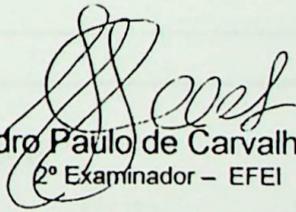
A N E X O I

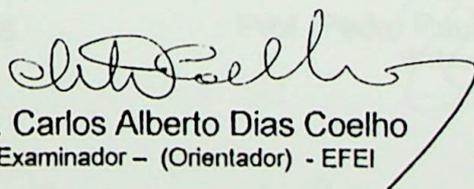
PRONUNCIAMENTO DA BANCA EXAMINADORA

Banca Examinadora, abaixo assinada, nomeada pela Portaria nº 342 de 14/12/00, considerando o resultado do Julgamento da Prova de Defesa Pública da Dissertação de Mestrado intitulada: **“Projeto e Ajuste Computacional de Controladores de Sistemas Dinâmicos pelo Método das Inequações”** apresenta pronunciamento no sentido de que o Coordenador dos Cursos de Pós-Graduação em Engenharia Elétrica da Escola Federal de Engenharia de Itajubá solicite ao DRA (Departamento de Registro Acadêmico) a expedição do título de **Mestre em Ciências em Engenharia Elétrica, na Área de Concentração Automação e Sistemas Elétricos Industriais**, satisfeitas as demais exigências regimentais, a **Juan Carlos Bejarano Luján**.

Itajubá, 18 de dezembro de 2000.


Prof. José Manoel Fernandes
1º Examinador - UFU


Prof. Pedro Paulo de Carvalho Mendes
2º Examinador - EFEI


Prof. Carlos Alberto Dias Coelho
3º Examinador - (Orientador) - EFEI



Ministério da Educação
ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
Reconhecida Lei 3232 - 05/01/1917

A N E X O II

FOLHA DE JULGAMENTO DA BANCA EXAMINADORA

Titulo da Dissertação: "Projeto e Ajuste Computacional de Controladores de Sistemas Dinâmicos pelo Método das Inequações"

Autor: Juan Carlos Bejarano Luján

JULGAMENTO

Examinadores	Conceito	Rubrica
1º	A+	
2º	A+	
3º	A+	

Resultado Médio: Conceito: A+, ou seja, aprovado com distinção.

Observações: _____

Itajubá, 18 de dezembro de 2000.

Prof. José Manoel Fernandes
1º Examinador - UFU

Prof. Pedro Paulo de Carvalho Mendes
2º Examinador - EFEI

Prof. Carlos Alberto Dias Coelho
3º Examinador - (Orientador) - EFEI

PROJETO E AJUSTE COMPUTACIONAL DE CONTROLADORES DE SISTEMAS DINÂMICOS PELO MÉTODO DAS INEQUAÇÕES

Por: Juan Carlos Bejarano Luján

Sumário

<i>Agradecimentos</i>	iii
<i>Resumo</i>	iv
<i>Abstract</i>	vi
<i>Pronunciamento da Banca Examinadora</i>	viii
<i>Folha de Julgamento da Banca Examinadora</i>	ix
<i>Sumário</i>	x
Capítulo 1 - PROJETO DE SISTEMAS DE CONTROLE LINEARES ATRAVÉS DO MÉTODO DAS INEQUAÇÕES	1
1.1 - <i>Introdução</i>	1
1.2 - <i>Descrição do texto</i>	4
Capítulo 2 - MODELO E ESTRATÉGIA DE PROJETO	5
2.1 - <i>Definições básicas sobre sistemas próprios e estritamente próprios</i>	5
2.1.1 - <i>Definição de função racional</i>	5
2.1.2 - <i>Excitações padronizadas</i>	5
2.2 - <i>Sistemas de controle com realimentação padrão</i>	6
2.2.1 - <i>Modelo condicionalmente linear</i>	6
2.2.2 - <i>As restrições no controle</i>	9
2.2.3 - <i>Representação de função de transferência</i>	13

2.2.4 - Estabilidade e comportamento dinâmico de sistemas lineares	15
2.2.5 - Definição de estabilidade	16
2.2.6 - Resposta a degrau e resposta ao impulso	19
2.2.7 - Sistemas estáveis e tempo de observação de transitório	21
Exemplo 2.1 : Sistema de primeira ordem.....	23
Exemplo 2.2 : Sistemas subamortecidos de segunda ordem.....	26
Exemplo 2.3 : Sistema de controle com realimentação padrão	29
Exemplo 2.4 : Pólos não dominantes localizados mais à direita na região de estabilidade.....	32
2.3 - Avaliação da saída do sistema a partir da resposta a degrau	35
2.4 - Critérios de desempenho	37
2.4.1 - Características da resposta a degrau	37
2.5 - O problema de projeto	39
2.6 - O Método das Inequações	42
Capítulo 3 - PROJETO DE SISTEMAS DE CONTROLE	44
3.1 - Aspectos computacionais	44
3.2 - O papel do projetista	50
3.3 - Projeto através das especificações da resposta a degrau	52
Capítulo 4 - RESOLVENDO AS INEQUAÇÕES	58
4.1 - O Processo dos Contornos Móveis (PCM)	58
4.2 - Ilustração do funcionamento do PCM para uma inequação e vetor de parâmetros bi-dimensional	66
Capítulo 5 - EXEMPLOS DE APLICAÇÃO DO MÉTODO DAS INEQUAÇÕES AO PROJETO DE SISTEMAS DE CONTROLE	70
Exemplo 5.1 : Compensação em cascata de um sistema tipo 1	70
Exemplo 5.2 : Compensação do regulador de velocidade de uma turbina hidráulica através do Método das Inequações	86

Capítulo 6 - CONCLUSÕES	98
6.1 – Trabalho desenvolvido	98
6.2 – Aspectos relevantes	99
6.3 – Sugestões para futuros trabalhos	101
 Bibliografia	 102
 Apêndice A - Guia das sub-rotinas utilizadas no projeto	 104
 Apêndice B - Comandos relevantes mais utilizados da “Toolbox” CACCON e da “Control System Toolbox” do MATLAB	 118
 Apêndice C - Guia para o usuário do programa de projeto.....	 121

CAPÍTULO I

PROJETO DE SISTEMAS DE CONTROLE LINEARES ATRAVÉS DO MÉTODO DAS INEQUAÇÕES

1.1 INTRODUÇÃO

Esta dissertação descreve o trabalho de implementação do Método das Inequações(MI), uma técnica criada por Zakian [17] para Projeto Assistido por Computador de Sistemas de Controle, em um ambiente computacional atual e amplamente disponível, a partir de parte de uma implementação anterior, em outra linguagem de programação, apresentada por Coelho [8] em sua tese de doutorado. Esta nova implementação foi desenvolvida em linguagem formal estruturada, que se aproveita das potencialidades de programação do aplicativo MATLAB [13], o qual, além de ser um aplicativo computacional que trabalha com matrizes, tem todas as características de linguagem de programação, além de possuir uma interface gráfica muito poderosa.

O projeto de sistemas de controle pelo MI parte da premissa de que as especificações de desempenho e as restrições de operação são estabelecidas através de faixas de valores dentro das quais deverão estar situados, ao final do projeto, os índices de desempenho, os parâmetros do controlador e os sinais de controle. O estabelecimento de tais faixas leva naturalmente o problema de projeto a ser formulado como um sistema de inequações que devem ser satisfeitas simultaneamente.

O Método das Inequações teve a sua concepção e o seu desenvolvimento inicial motivados pelos avanços na teoria de sistemas dinâmicos e do aumento contínuo da capacidade de armazenamento e de processamento de dados dos computadores. A aplicabilidade do MI gira principalmente em torno do uso eficaz de recursos computacionais para a resolução das inequações, através de algoritmos numéricos iterativos. Na primeira comunicação de grande circulação sobre o MI [17], além de ser apresentada a sua filosofia e suas premissas básicas, também foi apresentado o Processo dos Contornos Móveis - PCM, um algoritmo extremamente simples e eficaz para a solução de sistemas de inequações. O PCM, ao ser utilizado como parte de um programa para projeto assistido por computador de sistemas de controle, proporciona uma considerável economia de tempo e de esforço de cálculo, em comparação com os métodos

de projeto convencionais. Naquela primeira comunicação, a filosofia de projeto do MI foi aplicada ao projeto de sistemas de controle univariáveis e multivariáveis, considerando como índices de desempenho as características da resposta a degrau.

Em função dos bons resultados obtidos com o PCM [9,17,19], foi proposta uma Nova Formulação para o Método das Inequações - NFMI [16], onde foi proposto que os índices de desempenho deveriam contemplar o conjunto de todas as entradas possíveis a que os sistemas estariam sujeitos em sua operação, aí incluídos os sinais de referência e os distúrbios, e que, na formulação das inequações, também deveriam ser incluídas limitações nos sinais de controle, de modo a garantir a validade do modelo do sistema controlado e a limitar o esforço de controle. No aprofundamento destas idéias, Coelho [8] propôs que sistemas que cumprissem especificações de desempenho e respeitassem restrições de operação nessas condições, ou seja, para todas as entradas pertencentes ao conjunto de entradas possíveis, fossem denominados *Sistemas de Controle Robustos*.

Os comandos e rotinas computacionais desenvolvidos ao longo deste trabalho estão direcionados para o projeto de controladores de sistemas univariáveis, ou seja, do tipo SISO (do inglês, *Single-Input / Single-Output*), mas considerando que as entradas sejam degraus de amplitudes quaisquer, compatíveis com as condições de operação de cada sistema. Os índices de desempenho utilizados para estabelecimento das especificações são as características da resposta a degrau convencionais [19,10,12], mas o fato de se considerar degraus de amplitude quaisquer e a especificação de limites para os sinais de controle implica que as idéias lançadas na NFMI são parcialmente aplicadas, desde que tais degraus sejam realmente aqueles que normalmente vão ocorrer no sistema, durante sua operação.

Esse conjunto de comandos e rotinas computacionais foi concebido de modo a dar ao usuário o sentimento de estar executando um programa autônomo de projeto assistido por computador, a partir da ativação do comando principal. Assim, ao longo deste texto, esse conjunto de comandos e rotinas será denominado simplesmente *programa*.

Uma vez que as especificações de desempenho e as restrições de operação tenham sido formuladas como um sistema de inequações, é definida uma estrutura para o controlador, o que define um conjunto de parâmetros a serem ajustados. O ajuste desses parâmetros é feito

automaticamente através do PCM, que determina conjuntos de valores dos parâmetros para os quais o sistema de inequações é satisfeito.

Neste contexto, as tarefas envolvidas em um projeto assistido por computador pelo método das inequações são executadas de forma integrada através do programa interativo, através do qual o projetista fornece os dados do sistema, da estrutura do controlador, e dos limites estabelecidos para os índices de desempenho, para os parâmetros do controlador e para os sinais de controle. Executa-se então o PCM, através do qual o computador fará uma busca iterativa, variando automaticamente os parâmetros do controlador, até determinar um conjunto de valores que faça com que todas as inequações sejam satisfeitas simultaneamente. Ao final deste estágio o computador mostrará na tela os resultados obtidos pelo programa do projeto, juntamente com o conjunto de parâmetros responsáveis pela estabilidade e pelo comportamento dinâmico do sistema.

Da observação dos resultados o projetista poderá tomar varias decisões, entre elas a de traçar o gráfico da resposta a degrau do sistema e, se achar conveniente, a de tentar melhorar os resultados através de alterações das especificações do conjunto de inequações, ou da alteração da estrutura do controlador ou do compensador, incrementando o grau de complexidade, se for julgado necessário. Neste sentido o ciclo de alteração da estrutura do controlador, cálculos iterativos e exposição de resultados na tela do computador é repetido, até que o projetista fique satisfeito com os resultados alcançados .

Conforme já mencionado, a formulação básica do problema considera sistemas que possam ser representados através de modelos condicionalmente lineares, e sujeitos a entradas degrau de amplitudes quaisquer. Este modelo é especialmente adequado para trabalhar com sistemas sujeitos a saturação, a mais freqüente e a mais importante das não-linearidades.

Dentro do projeto de controle, as inequações expressam as condições de estabilidade, as especificações da resposta a degrau, e as restrições de operação. Na resolução destas inequações pelo PCM [17], implementado no ambiente MATLAB, são utilizados repetidas vezes comandos da CACCON Control Toolbox [4] para computação da abscissa de estabilidade do sistema, e das características de resposta a degrau.

Com o programa desenvolvido neste trabalho, os resultados comunicados em [8] foram reproduzidos, e são tão bons quanto ou melhores do que aqueles obtidos com técnicas de controle clássico e de otimização. Além disso os índices de desempenho são facilmente computados, só dependem da resposta a degrau de malha fechada, o que torna a solução computacional das inequações bastante rápida. Os fatores acima expostos, juntamente com as vantagens oferecidas pelo ambiente MATLAB, geraram a motivação para a implementação do Método das Inequações detalhada neste trabalho.

Outro objetivo desta dissertação é reiterar a eficácia do Método das Inequações no projeto de controladores para sistemas univariáveis, e motivar novas aplicações em todas as áreas em que existam problemas de controle de sistemas ou processos dinâmicos, cujas especificações possam ser formuladas através de inequações que possam ser satisfeitas simultaneamente através do ajuste de parâmetros dos controladores.

1.2 DESCRIÇÃO DO TEXTO

No capítulo 2 desta dissertação são apresentados os fundamentos teóricos e os desenvolvimentos matemáticos que são o ponto de partida para a compreensão do Método das Inequações; no capítulo 3 é feito o desenvolvimento da idéia central do método, de formulação do problema de projeto de sistemas de controle em termos de inequações; no capítulo 4 tem-se o detalhamento do Processo dos Contornos Móveis, o algoritmo utilizado para resolução das inequações no programa de projeto; e no capítulo 5 são apresentados exemplos de aplicação do MI ao projeto de sistemas de controle, com utilização do PCM para resolução do sistema de inequações. As conclusões são apresentadas no capítulo 6.

Para que este texto, na medida do possível, seja auto-suficiente, também são incluídos três apêndices: no Apêndice A se encontra a descrição das subrotinas utilizadas no programa e os objetivos de cada uma delas, respectivamente; no Apêndice B tem-se uma descrição resumida dos principais comandos da “*Control System Toolbox*” do MATLAB [22] e da “*CACCON Control Toolbox*” [4]. No Apêndice C tem-se o manual para o usuário do programa de Projeto de Sistemas de Controle pelo Método das Inequações.

CAPÍTULO 2

MODELO E ESTRATÉGIA DE PROJETO

2.1 DEFINIÇÕES BÁSICAS SOBRE SISTEMAS PRÓPRIOS E ESTRITAMENTE PRÓPRIOS

As seguintes duas subseções são uma revisão dos conceitos apresentados por Coelho em [19].

2.1.1 Definição de função racional

Considerando-se a seguinte função racional $W(s)$ como sendo a relação de dois polinômios dada pela forma :

$$W(s) = \frac{P(s)}{Q(s)} = \frac{b_w \cdot s^w + b_{w-1} \cdot s^{w-1} + \dots + b_2 \cdot s^2 + b_1 \cdot s + b_0}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_2 \cdot s^2 + a_1 \cdot s + a_0} \quad (2.1)$$

Uma característica importante das funções racionais é a diferença entre os graus dos polinômios, do numerador e do denominador.

Sob este ponto de vista, as funções racionais podem ser classificadas da seguinte forma :

Se $w < n$, $W(s)$ é uma função estritamente própria.

Se $w \leq n$, $W(s)$ é uma função própria.

Se $w > n$, $W(s)$ é uma função imprópria.

Seguindo esta linha de pensamento pode-se dizer que um sistema é próprio quando ele tiver uma função de transferência própria, e teremos um sistema estritamente próprio quando a sua função de transferência for uma função estritamente própria.

2.1.2 Excitações padronizadas

As excitações ou entradas para sistemas de controle e outros sistemas dinâmicos podem ter caráter aleatório e determinístico, de acordo com o tipo e a característica de funcionamento do sistema. Funções aleatórias não podem ser expressas analiticamente e não são repetitivas. As

funções determinísticas, por sua vez, têm características bem definidas por funções analíticas ou curvas específicas. Um exemplo de excitação de característica determinística é o molde utilizado numa operação de máquina onde a ferramenta de corte é controlada de modo a reproduzir o contorno do molde. E existem os sistemas de regulação automática, que têm o objetivo de manter as variáveis controladas em valores constantes estabelecidos pelos sinais de referência.

Entretanto, para se analisar, ajustar e controlar sistemas dinâmicos, é necessário haver uma base para sua identificação e especificação de desempenho. Isto é feito através da resposta do sistema a excitações padronizadas, sobre as quais são definidas as características de desempenho. O ajuste e o projeto dos controladores devem ter o objetivo de fazer com que essas características de desempenho cumpram as especificações exigidas pelo projetista.

As excitações padronizadas usadas para identificação e análise de desempenho do sistema de controle utilizados nesta dissertação, são os seguintes:

1. Degrau $R_0 u_{-1}(t)$
2. Impulso $A u_0(t) = A (d/dt) [u_{-1}(t)]$

O degrau é a excitação mais utilizada para a análise e a especificação de desempenho transitório.

2.2 SISTEMAS DE CONTROLE COM REALIMENTAÇÃO PADRÃO

2.2.1 Modelo condicionalmente linear

Ao longo deste texto será adotada a estrutura de realimentação padrão mostrada na Figura 2.1(a), que é amplamente usada em Projetos de Sistemas de Controle. Denomina-se um sistema que tenha tal estrutura como um sistema de controle de realimentação padrão (do inglês - SFCS - *Standard Feedback Control System*). O modelo deste sistema, para o caso de sistemas univariáveis, consiste das seguintes expressões matemáticas:

$$|u(t)| \leq K \quad (2.2)$$

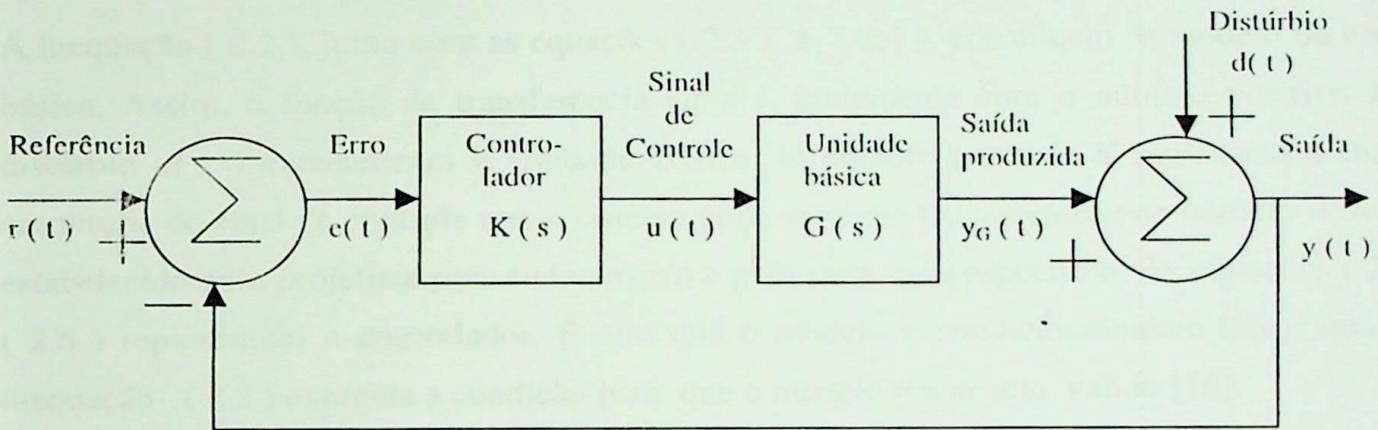
$$Y_G(s) = G(s)U(s) \quad (2.3)$$

$$y(t) = y_G(t) + d(t) \quad (2.4)$$

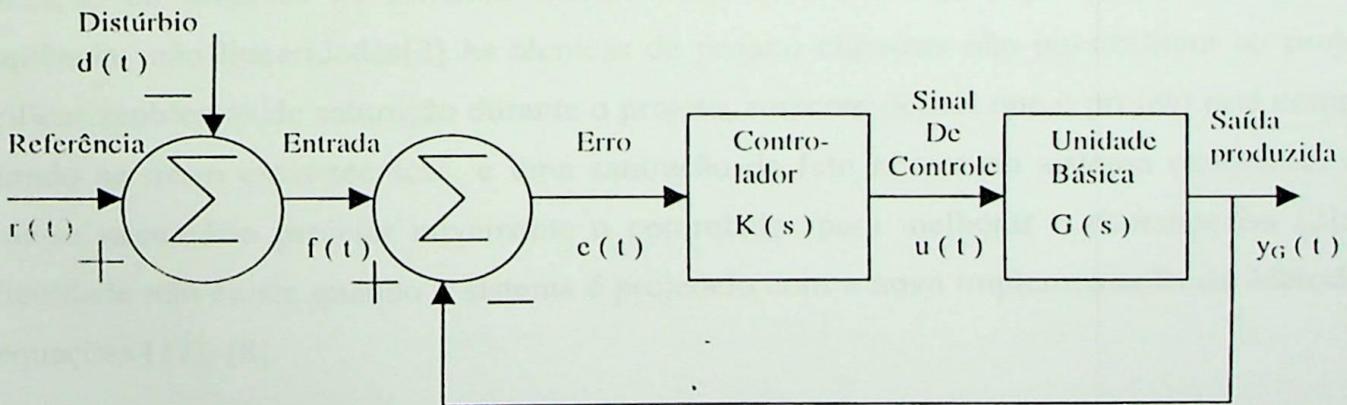
$$U(s) = K(s)E(s) \quad (2.5)$$

$$e(t) = r(t) - y(t) \quad (2.6)$$

A variável independente, t , denota tempo e todas as funções do tempo são definidas para todo $t \geq 0$. Este modelo foi proposto para o caso geral multivariável [17], mas a versão apresentada aqui refere-se ao caso de Sistemas Univariáveis, ou seja, de uma única entrada / única saída (do inglês- SISO – *Single Input / Single Output*), cujo modelo é dado em termos de funções de transferência, que é o assunto desta dissertação. Neste contexto, as variáveis $u(t)$, $e(t)$, $r(t)$, $d(t)$, $y_G(t)$, $y(t)$ serão escalares e serão denominadas respectivamente controle, erro, referência, distúrbio, saída produzida e saída.



(a)



(b)

Figura 2.1 Sistema de controle com realimentação padrão (a) estrutura atual ;

(b) representação equivalente

As equações (2.3) e (2.5) , $Y_G(s)$, $U(s)$ e $E(s)$ são as transformadas de Laplace de $y_G(t)$, $u(t)$ e $e(t)$ respectivamente. Por exemplo, $U(s)$ é definido pela expressão :

$$U(s) = \mathcal{L} [u(t)] = \int_0^{\infty} u(t) e^{-st} dt \quad (2.7)$$

Para ser coerente com os modelos de espaço de estado, todas as funções de transferência sempre serão dadas através da relação de polinômios racionais em s , os quais serão próprios, ou seja, o grau do numerador é menor, ou, no máximo, igual ao grau do denominador. No sistema de controle com realimentação padrão definido pela equações 2.3 a 2.6 (na figura 2.1) será considerado que $G(s)$ seja uma função estritamente própria.

A inequação (2.2), junto com as equações (2.3) e (2.4), constituem o modelo da unidade básica. Assim, a função de transferência $G(s)$, juntamente com o número positivo K e o distúrbio $d(t)$ caracterizam a Unidade Básica. O número positivo K representa a máxima amplitude do sinal de controle que o sistema pode suportar fisicamente; este número deverá ser estabelecido pelo projetista para cada projeto e para cada caso específico. As equações (2.5) e (2.6) representam o controlador. É dito que o modelo é condicionalmente linear quando a inequação (2.2) expressa a condição para que o modelo linear seja válido [16].

A restrição no controle , definida pela inequação (2.2), expressa corretamente a condição para a validação de modelos de sistemas lineares sujeitos a saturação, nos quais ocorrem com frequência não linearidades[2].As técnicas de projeto clássicas não possibilitam ao projetista verificar problemas de saturação durante o projeto, somente depois que o projeto está completo. Quando se usam estas técnicas, e uma saturação de fato ocorre no sistema resultante; então torna-se necessário projetar novamente o controlador para melhorar o desempenho [2]; esta dificuldade não existe quando o sistema é projetado com a nova implementação do Método das Inequações [17], [8].

A idéia de considerar o distúrbio como parte do modelo significa que a Unidade Básica e os distúrbios são inseparáveis, no sentido de que os distúrbios sempre estarão presentes durante a operação de sistema. Isto é verdade em muitos problemas de controle; como exemplos têm-se os conjugados de vento que agem em uma antena de posicionamento de radar, de forças que

surtem da operação de equipamento que age em uma ferramenta elétrica [16], ou as características elétricas da carga que está sendo alimentada por uma central elétrica [11].

Como a referência deve ser conhecida e o distúrbio surge como parte do modelo da Unidade Básica, não deve haver nenhum problema em combinarmos ambos os sinais para definição da entrada combinada $f(t)$ como segue:

$$f(t) = r(t) - d(t) \quad (2.8)$$

A combinação das equações (2.4), (2.6) e (2.8) dá uma expressão nova para o erro :

$$e(t) = f(t) - y_G(t) \quad (2.9)$$

A introdução das equações (2.8) e (2.9) estão ilustradas no diagrama de blocos da figura 2.1 (b), que é o modelo equivalente ao da figura 2.1 (a). Porém, se $f(t)$ é tratado como uma única entrada, a representação ilustrada na figura 2.1 (b) fica mais conveniente devido à sua forma mais conhecida, mas com $u(t)$ e $e(t)$ ainda aparecendo na forma explícita. A conclusão imediata obtida da equação (2.9) (ou figura 2.1(b)) é que para que o erro seja pequeno, a saída produzida $y_G(t)$ tem que acompanhar a entrada $f(t)$ com precisão satisfatória, e independentemente da entrada ser a referência, o distúrbio ou uma combinação de ambos.

2.2.2 As restrições no controle

A inequação (2.2) pode servir também para limitar a amplitude do esforço gerado pelo controlador ou para evitar o seu desgaste excessivo ou outros danos ao sistema, ou para tornar válida a linearidade do modelo da Unidade Básica. No último caso, é suposto que quando esta inequação não é satisfeita o sistema não pode ser representado pelo modelo linear [16]; neste caso, com tal modelo os resultados obtidos podem não ser confiáveis.

A inequação (2.2) é especialmente útil quando a saturação é a principal não linearidade na saída do controlador ou na entrada da Unidade Básica. O problema causado pela saturação em projeto de sistemas de controle é bem ilustrado por Chen [2] em parte do livro dele reproduzida aqui * * .

SATURAÇÃO.

A maioria dos sistemas físicos é, estritamente falando, não linear. As suas descrições matemáticas lineares são todas obtidas a partir de suposições de linearização. As não linearidades podem ser dos tipos saturação, folga, atrito de Coulomb, zona morta, ou outros. Entre eles, é encontrada freqüentemente saturação em sistemas de controle. Acontece em amplificadores eletrônicos, geradores DC e em motores AC, e motores hidráulicos. Primeiro usamos um exemplo para ilustrar o efeito da saturação.

EXEMPLO 1

Considere o sistema mostrado na figura 2.2. O elemento N é um amplificador com ganho 2. A função de transferência global é $(2s + 4)/(s^2 + s + 4)$. O sistema é claramente estável do tipo **BIBO**. Desde que os termos independentes no numerador e no denominador sejam os mesmos, o sistema tem zero de erro de regime permanente para uma entrada degrau. As respostas do sistema devido às duas funções degrau com magnitudes diferentes são mostradas na figura 2.4.

Na realidade, o amplificador pode ter a característica mostrada na figura 2.3. Para facilidade de simulação, a curva de saturação é aproximada pelas linhas pontilhadas. São mostradas as respostas do sistema com a curva de saturação aproximada devido à aplicação das duas funções degrau na figura 2.4. Isto é obtido usando a "toolbox" Simulink do programa MATLAB. Vemos que se o elemento N não está saturado, a análise linear dá um resultado muito preciso.

Por outro lado, se o elemento N está saturado, então a resposta do modelo linear do sistema físico real é bastante diferente. Na realidade, o sistema com saturação ficará instável se $y_d(t) = a$, com $a \geq 1,15$, embora o modelo linear seja sempre estável, no sentido BIBO.

Vemos a partir deste exemplo que no projeto de um sistema de controle usando modelos lineares, é importante manter os componentes fora de saturação; caso contrário o sistema não funcionará como esperado. Manter um componente fora da saturação, porém, é um problema muito complicado, porque, no processo de projeto, o conhecimento da resposta exata do sistema resultante não está disponível.

Conseqüentemente o problema de saturação só pode ser conferido depois do projeto ter sido completado. Se a saturação realmente acontece no sistema resultante, o sistema tem que ser reprojetoado para melhorar seu desempenho.

** No texto aqui reproduzido, BIBO (do inglês *Bounded Input / Bounded Output*) representa Entrada limitada / Saída limitada. Qualquer sistema linear que tenha função de transferência própria e seja assintoticamente estável é BIBO.

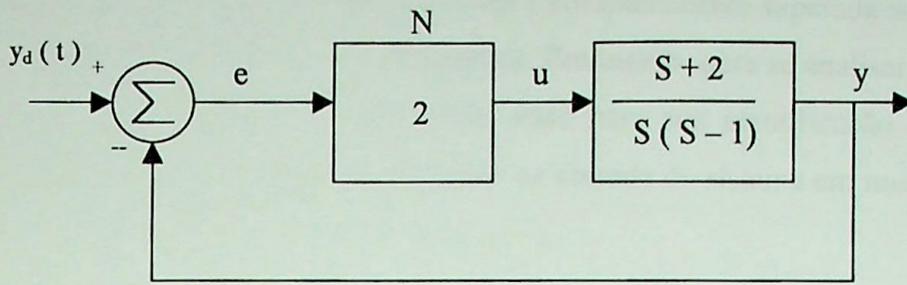


Figura 2.2 Bloco representativo do amplificador

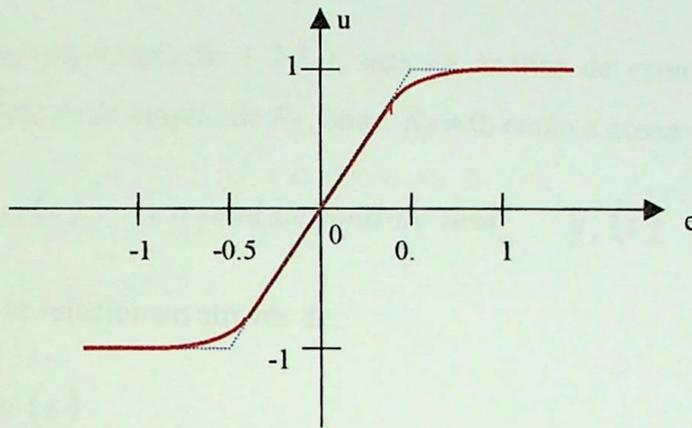


Figura 2.3 A saturação

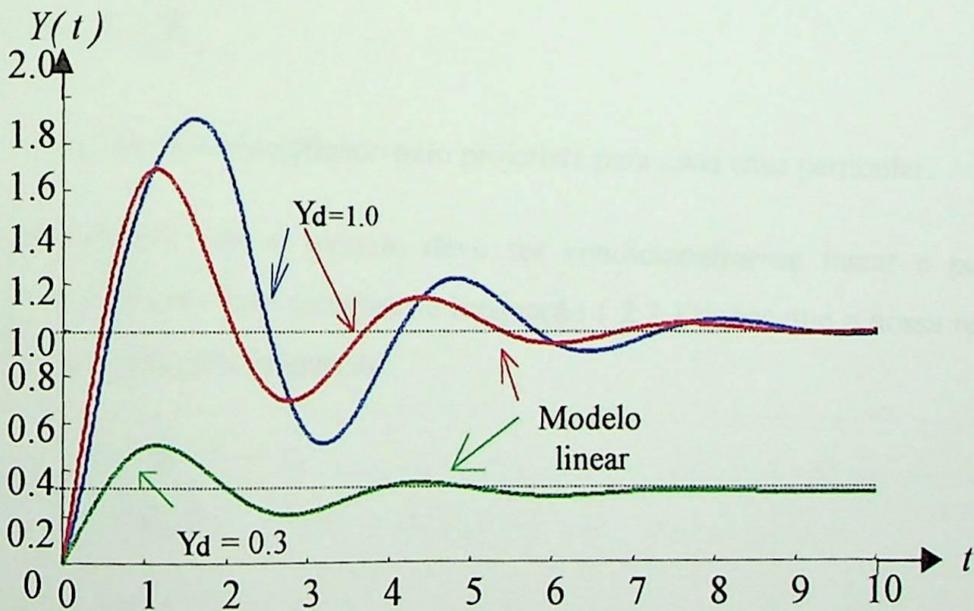


Figura 2.4 O efeito da saturação

A dificuldade do projeto mencionada por Chen é completamente superada ao projetar-se com a nova implementação do Método das Inequações. Entretanto, para se analisar, ajustar e controlar o sistema dinâmico é necessário haver uma base para sua identificação e especificação de desempenho. Isto foi feito através da excitação na entrada do sistema em nosso projeto por uma entrada degrau de amplitude R_0 , isto é $R_0 u_{-1}(t)$.

Temos que para uma entrada degrau de amplitude um:

$$r(t) = u_{-1}(t) \quad \text{o sinal de controle seria} \quad u(t) \quad (2.10)$$

que é o sinal de controle da inequação (2.2), mas na análise de estudo desta dissertação utilizamos uma entrada degrau de amplitude R_0 , onde $R_0 \neq 0$, então a nossa entrada será:

$$r(t) = R_0 u_{-1}(t) \quad \text{e o sinal de controle será} \quad u_x(t) \quad (2.11)$$

ambos sinais de controle se relacionam através de:

$$u_x(t) = R_0 u(t) \quad (2.12)$$

Onde o sinal de controle para uma entrada a degrau de amplitude R_0 também é menor que ou igual a K o que pode ser expresso por:

$$|u_x(t)| \leq K \quad (2.13)$$

lembrando que o número K é especificado pelo projetista para cada caso particular.

Levando em consideração que o modelo deve ser condicionalmente linear e partindo do pressuposto de que o projeto deve satisfazer a inequação (2.2) temos que a nossa restrição de controle será dada pela seguinte inequação:

$$|u(t)| \leq \frac{K}{|R_0|} \quad (2.14)$$

que satisfaz implicitamente a inequação (2.2), tornando válido o modelo linear.

Quando um projeto é realizado através do método proposto e se obtém sucesso, e ainda mantendo a inequação (2.14) satisfeita, pode-se ter certeza que não haverá nenhuma

saturação no sistema e pode-se garantir que não haverá necessidade de reprojeter-se o controlador.

A linearização de outros tipos de não linearidades também pode ter sua validade expressa pela inequação (2.14). Neste caso, a linearização de um elemento não linear é levada a cabo em torno de um ponto de operação [3] , e as funções de tempo se tornam variáveis incrementais. Em tais casos, a inequação (2.14) dá o limite dentro do qual o controle incremental tem que permanecer para que os resultados obtidos com o modelo linear tenham precisão aceitável.

2.2.3 Representação de função de transferência

Para sistemas do tipo univariáveis, os modelos da Unidade Básica e do controlador podem ser expressos convenientemente na forma de funções de transferência, as quais são obtidas a partir das equações (2.3) e (2.5), resultando em :

$$G(s) = \frac{Y_G(s)}{U(s)} \quad (2.15)$$

$$K(s) = \frac{U(s)}{E(s)} \quad (2.16)$$

A transformada de Laplace da equação (2.9) é:

$$E(s) = F(s) - Y_G(s) \quad (2.17)$$

$F(s)$, $Y_G(s)$ e $E(s)$ são as transformadas de Laplace de $f(t)$, $y_G(t)$ e $e(t)$, respectivamente.

Os modelos de espaço de estado de sistemas dinâmicos lineares invariantes sempre correspondem a funções de transferência racionais. Essas funções de transferência serão sempre próprias, ou seja, relações de dois polinômios, com o grau do polinômio do numerador sendo menor que, ou, no máximo, igual ao grau do polinômio do denominador.

A representação da Unidade Básica e do controlador através de funções de transferência é adotada nesta dissertação. Para ser consistente com a representação do sistema por equações de espaço de estado, todas as funções de transferência serão dadas por relações de polinômios racionais em s . A função de transferência da unidade básica $G(s)$ sempre terá o grau do

polinômio do numerador menor que o grau do polinômio do denominador. A função de transferência do controlador $K(s)$ pode ter o grau do polinômio do numerador com o mesmo grau do polinômio do denominador, mas não maior do que este.

Isto garantirá que, contanto que o sistema seja estável e as entradas sejam todas limitadas, todas as respostas dinâmicas do sistema ($u(t)$, $y_G(t)$, $y(t)$ e $e(t)$) também serão limitadas.

As respostas no tempo de um Sistema de Controle com Realimentação Padrão, como o da figura 2.1, são calculadas a partir das seguintes funções de transferência de malha fechada :

$$\frac{U(s)}{F(s)} = W_u(s) = \frac{K(s)}{1+K(s)G(s)} \quad (2.18)$$

$$\frac{E(s)}{F(s)} = W_e(s) = \frac{1}{1+K(s)G(s)} \quad (2.19)$$

$$\frac{Y_G(s)}{F(s)} = W_y(s) = \frac{K(s)G(s)}{1+K(s)G(s)} \quad (2.20)$$

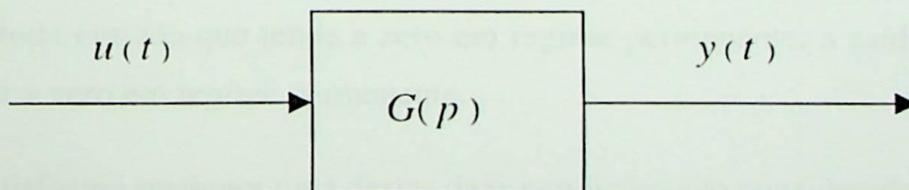
Pode-se ver que as funções de transferência das equações (2.3) a (2.6) são todas dadas por relações de polinômios racionais em s onde o grau do polinômio do numerador é menor que ou igual ao grau do polinômio do denominador. Quando se quer projetar funções de transferência de sistemas univariáveis do tipo SISO, essas funções são diretamente calculadas através das equações (2.18) a (2.20).

Considerando que as funções de transferência são todas próprias, é importante notar que a estabilidade do sistema só irá depender dos pólos do sistema, ou seja, dos zeros do denominador comum às funções de transferência de malha fechada, dadas pelas equações (2.18) a (2.20). Quando estas funções de transferência forem colocadas na forma racional, terão todas o mesmo polinômio característico, que levará a uma mesma equação de estado para todos os cálculos de resposta no tempo, sendo necessário que apenas as equações de saída sejam distintas para cada resposta a degrau, o que torna esta técnica muito eficaz para redução do tempo de computação.

2.2.4 Estabilidade e comportamento dinâmico de sistemas lineares

As seguintes duas subseções são uma revisão e uma adaptação da teoria exposta por Coelho em [19] para o problema de projeto com o propósito de deixar o texto auto-suficiente.

O critério para definição de estabilidade em sistemas dinâmicos se baseia na *resposta natural* de cada sistema e na resposta a excitações de amplitude limitada. Por resposta natural de um sistema dinâmico entende-se aquela componente da resposta cujas características de variação no tempo são sempre do mesmo tipo, quaisquer que sejam os sinais de entrada. A resposta natural aparece inclusive quando se retiram todas as excitações do sistema, e é justamente esta a situação mais favorável para observá-la: ao se retirar as excitações, os componentes estarão fora de seus estados de repouso, e naturalmente irão buscar estes estados com uma velocidade que irá depender tão somente dos parâmetros dos componentes e de que forma eles estão interligados. Nos sistemas lineares invariantes (SLI), os tipos de variação das diversas parcelas da resposta natural dependerão tão somente dos pólos da função de transferência (FT) do sistema. Considerando que as excitações são retiradas num instante t_0 , os valores das variáveis nesse instante são as *condições iniciais* do sistema. O instante t_0 corresponde ao início da observação da resposta do sistema, ou ainda o instante da ocorrência de qualquer outro distúrbio. Assim, um sistema pode ou não ter condições iniciais nesse instante. No caso mais geral de haver condições iniciais e excitação, a resposta terá duas componentes: uma é a parcela devida somente às condições iniciais, que é denominada *resposta livre*. A outra é a parcela devida somente às excitações, a qual é denominada *resposta forçada*. A *resposta total* é dada pela soma das respostas livre e forçada.



(a)

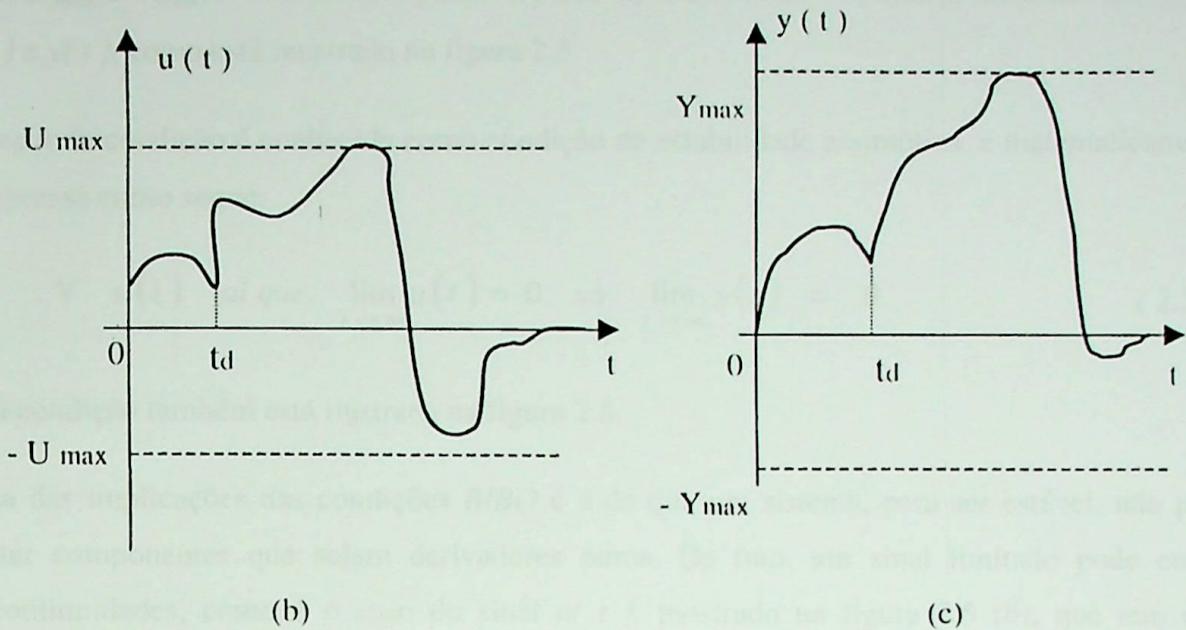


Figura 2.5 (a) Sistema linear; (b) entrada limitada com valor final nulo;
(c) resposta típica de um sistema estável.

Para definição de estabilidade, será considerado um sistema com uma entrada (excitação) $u(t)$ e com uma única saída (resposta) $y(t)$.

2.2.5 Definição de estabilidade

Diz-se que um sistema é estável se, e somente se, forem satisfeitas simultaneamente as duas seguintes condições:

- I) Para toda entrada limitada, a saída também deverá ser limitada.
- II) Para toda entrada que tende a zero em regime permanente, a saída também deve tender a zero em regime permanente.

Sistemas que não satisfaçam qualquer uma destas duas condições são considerados instáveis.

A primeira condição é conhecida como de *entrada limitada/saída limitada*, ou então pela sigla *BIBO*. Matematicamente a condição BIBO é expressa da seguinte forma:

$$\forall u(t) \text{ tal que } |u(t)| \leq U_{\max}, \exists \text{ um } Y_{\max} \text{ tal que } |y(t)| \leq Y_{\max} \quad (2.21)$$

onde U_{max} e Y_{max} são constantes positivas, correspondentes às amplitudes máximas dos sinais $u(t)$ e $y(t)$, como está mostrado na figura 2.5.

A segunda condição é conhecida como condição de estabilidade assintótica, e matematicamente é expressa como segue:

$$\forall u(t) \text{ tal que } \lim_{t \rightarrow \infty} u(t) = 0 \Rightarrow \lim_{t \rightarrow \infty} y(t) = 0 \quad (2.22)$$

Esta condição também está ilustrada na figura 2.5.

Uma das implicações das condições *BIBO* é a de que um sistema, para ser estável, não pode conter componentes que sejam derivadores puros. De fato, um sinal limitado pode conter discontinuidades, como é o caso do sinal $u(t)$, mostrado na figura 2.5 (b), que tem uma discontinuidade no instante t_d . Se tal sinal for a entrada de um derivador puro, a saída, teoricamente, terá uma parcela do tipo impulso, isto é, de duração tendendo a zero, mas de amplitude infinita; na prática, isto pode causar um dano sério ao próprio componente e aos componentes próximos.

A condição de estabilidade assintótica implica que a resposta livre do sistema deve ter valor final igual a zero. Isto é o mesmo que dizer que um sistema estável, estando em estado de equilíbrio, ao sofrer qualquer distúrbio temporário, deve retornar ao mesmo estado de equilíbrio após o término do distúrbio.

Conforme já definido, um sistema é assintoticamente estável quando, para uma entrada com valor final nulo, o valor final da saída também for nulo. Para que o valor final de uma variável dinâmica seja nulo, todos os pólos (ou raízes) de sua Transformada de Laplace devem ter parte real negativa. Assim se o valor final da entrada $u(t)$ for nulo, todos os pólos de $U(s)$ terão parte real negativa [19]. Para que o valor final da saída $y(t)$ seja nulo, será necessário que todos os pólos de $Y_C(s) = G(s)U(s)$ da equação (2.15) também tenham parte real negativa. Isto implica no seguinte:

“ Para um sistema linear racional ser assintoticamente estável, é necessário e suficiente que todos os pólos de sua função de transferência tenham parte real negativa.”

A condição de estabilidade de um sistema que tenha uma função de transferência como a da equação (2.20) pode ter sua estabilidade descrita pela seguinte inequação:

$$\sigma_s < 0 \quad (2.23)$$

onde σ_s é a abscissa de estabilidade do sistema, definida abaixo como sendo:

$$\sigma_s = \max_i \{ \text{Re}(\lambda_i), i=1, 2, \dots, n \} \quad (2.24)$$

Onde os λ_i denotam os n pólos de malha fechada do sistema, os quais são as raízes da equação característica :

$$1 + K(s)G(s) = 0 \quad (2.25)$$

que na realidade é o denominador comum às equações (2.18) a (2.20) igualado a zero.

O sistema será estável se, e somente se, a inequação (2.23) for satisfeita. No plano "s", a condição dada pela equação (2.23) é visualizada da seguinte forma: σ_s é a abscissa do pólo ou pólos da equação característica (2.25) que estiver(em) posicionado(s) mais à direita, conforme mostra a figura 2.6. A condição $\sigma_s < 0$ significa que todos os pólos devem estar no semiplano esquerdo aberto, isto é, à esquerda do eixo $j\omega$. Não há restrições quanto à posição de zeros.

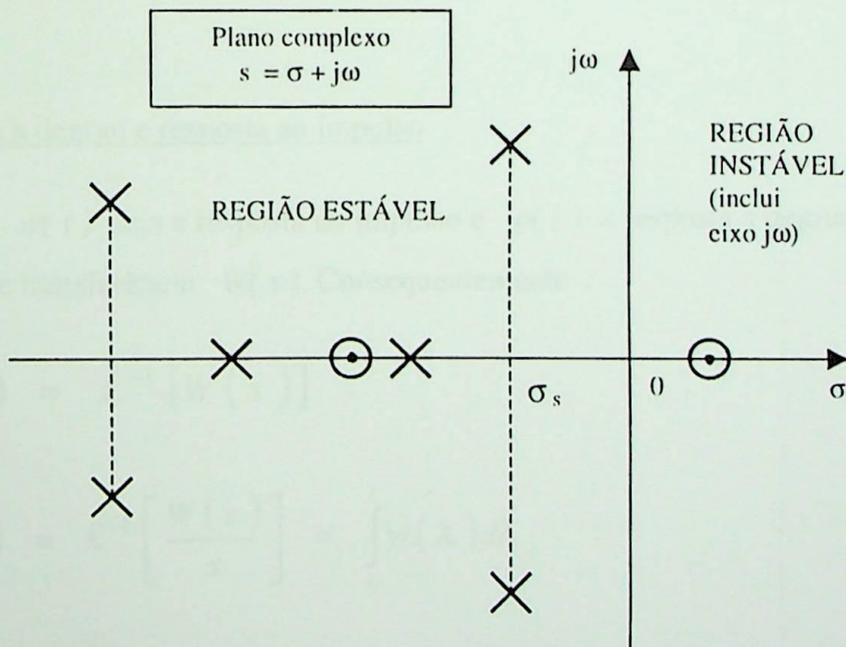


Figura 2.6 Pólos e zeros de um sistema e abscissa de estabilidade σ_s .

A estabilidade assintótica garante em parte a estabilidade BIBO, ao não permitir a existência de pólos reais positivos, que fazem aparecer exponenciais crescentes na resposta. Ou de pólos complexos com parte real positiva, que correspondem a oscilações de amplitude crescente. A garantia total de estabilidade BIBO vem com a exigência da função de transferência ser própria, não tendo, por isto, derivadores puros. A inexistência de derivadores puros impedirá que discontinuidades na entrada provoquem impulsos na saída.

Considera-se d_K como sendo o ganho de alta frequência do controlador, definido pela seguinte equação :

$$d_K = \lim_{s \rightarrow \infty} K(s) \quad (2.26)$$

Considera-se que $G(s)$ é uma função estritamente própria e sendo assim é possível demonstrar que :

$$u(0) = d_K f(0) \quad (2.27)$$

$$e(0) = f(0) \quad (2.28)$$

$$y_G(0) = 0 \quad (2.29)$$

2.2.6 Resposta a degrau e resposta ao impulso

Considere que $w(t)$ seja a resposta ao impulso e $\rho(t)$ a resposta a degrau do sistema tendo como função de transferência $W(s)$. Consequentemente :

$$w(t) = \mathcal{L}^{-1} [W(s)] \quad (2.30)$$

$$\rho(t) = \mathcal{L}^{-1} \left[\frac{W(s)}{s} \right] = \int_0^t w(\lambda) d\lambda \quad (2.31)$$

Utilizam-se as expressões própria e estritamente própria da resposta a degrau e da resposta ao impulso para referir-se às respostas ao impulso e ao degrau de sistemas próprios e estritamente próprios, respectivamente. Algum significado físico pode ser acrescentado à definição de

sistemas próprios e estritamente próprios através do exame das respostas ao impulso e a degrau $w(t)$ e $\rho(t)$.

As respostas a degrau de malha fechada são definidas respectivamente para o controle, o erro e a saída produzida pelas seguintes transformadas inversas de Laplace :

$$\rho_u(t) = \mathcal{L}^{-1} \left[\frac{W_u(s)}{s} \right] \quad (2.32)$$

$$\rho_e(t) = \mathcal{L}^{-1} \left[\frac{W_e(s)}{s} \right] \quad (2.33)$$

$$\rho_y(t) = \mathcal{L}^{-1} \left[\frac{W_y(s)}{s} \right] \quad (2.34)$$

onde \mathcal{L}^{-1} denota a transformada inversa de Laplace.

Por comparação com a equação (2.17), se obtém :

$$\rho_e(t) = 1 - \rho_y(t) \quad (2.35)$$

Os valores iniciais de cada uma das respostas a degrau mencionadas acima são os seguintes :

$$\rho_e(0) = 1 \quad (2.36)$$

$$\rho_u(0) = d_K \quad (2.37)$$

$$\rho_y(0) = 0 \quad (2.38)$$

Para uma função de transferência $W(s)$ do tipo da equação (2.1), tem-se que a respectiva resposta ao impulso estritamente própria não contém nenhum impulso. Uma resposta ao impulso que só é própria, mas sem ser estritamente própria, contém um impulso. A resposta a degrau de um sistema próprio (mas não estritamente próprio) tem valor inicial diferente de zero $\rho(0) = C$. Quando um sistema é estritamente próprio tem-se que $\rho(0) = 0$. Estas

afirmações podem ser conferidas aplicando-se o teorema do valor inicial à transformada de Laplace nas equações (2.30) e (2.31)

$$w(0) = \lim_{s \rightarrow \infty} sW(s) \quad (2.39)$$

$$\rho(0) = \lim_{s \rightarrow \infty} W(s) \quad (2.40)$$

Na equação (2.40), observa-se que o limite usado na última seção para definir se um sistema é próprio ou não ou estritamente próprio dá, na realidade, o valor inicial da resposta a degrau. Pode-se dizer então que um sistema é próprio quando o valor inicial da sua resposta a degrau existir e for finito. Um sistema é estritamente próprio se o valor inicial da sua resposta a degrau for zero. Esta definição tem mais sentido do que aquela dada na seção precedente, e pode, eventualmente, ser estendida a sistemas não lineares.

2.2.7 Sistemas estáveis e tempo de observação do transitório

Na análise de sistemas dinâmicos estáveis, é freqüente que a avaliação do desempenho seja feita com base na resposta $y(t)$ a uma entrada degrau, considerando que antes da aplicação do degrau o sistema está em repouso, isto é, com a saída e todas as suas derivadas nulas.

Supondo que o degrau seja aplicado no instante $t = 0$ e tenha amplitude R_0 , então a entrada do sistema é

$$r(t) = R_0 u_{-1}(t), \quad R_0 \neq 0 \quad (2.41)$$

Um sistema linear é estável quando sua abscissa de estabilidade é negativa, como expresso pela inequação (2.23). Em um sistema estável a resposta ao impulso tende a zero e a resposta a degrau tende a um valor constante,

$$\lim_{t \rightarrow \infty} w(t) = 0 \quad (2.42)$$

$$\lim_{t \rightarrow \infty} \rho(t) = \rho(\infty) \quad (2.43)$$

onde $\rho(\infty)$ é constante. O valor final da resposta a um degrau de amplitude R_0 pode ser calculado diretamente da função de transferência $W(s)$ usando-se o teorema de valor final da transformada de Laplace,

$$\rho(\infty) = \lim_{s \rightarrow 0} s \mathcal{L}[\rho(t)] = \lim_{s \rightarrow 0} W(s) \cdot R_0 \quad (2.44)$$

Para uma função de transferência da forma da equação (2.1) tem-se :

$$\rho(\infty) = \frac{b_0}{a_0} \cdot R_0 \quad (2.45)$$

Para sistemas estáveis, as respostas a degrau terão valores constantes de regime permanente constante, conforme segue :

$$\rho_u(\infty) = \lim_{t \rightarrow \infty} \rho_u(t) = \lim_{s \rightarrow 0} W_u(s) \cdot R_0 \quad (2.46)$$

$$\rho_e(\infty) = \lim_{t \rightarrow \infty} \rho_e(t) = \lim_{s \rightarrow 0} W_e(s) \cdot R_0 \quad (2.47)$$

$$\rho_Y(\infty) = \lim_{t \rightarrow \infty} \rho_Y(t) = \lim_{s \rightarrow 0} W_Y(s) \cdot R_0 \quad (2.48)$$

esse valor final será dado por

$$\rho_Y(\infty) = W(0) \cdot R_0 \quad (2.49)$$

$W(0)$ é o ganho estático do sistema e R_0 é a amplitude do degrau aplicado na entrada. Esta expressão do valor final é válida mesmo que haja condições iniciais não nulas, porque em sistemas estáveis o valor da resposta livre é sempre igual a zero.

A duração e a forma da resposta no período transitório dependem dos elementos dinâmicos que constituem o sistema. Em sistemas lineares, essa duração e forma dependem principalmente dos pólos da Função de Transferência. Teoricamente, a resposta transitória

$$\rho_N(t) = \rho(t) - \rho(\infty) \quad (2.50)$$

só se anula quando $t \rightarrow \infty$; na prática aceita-se que o transitório pode ser considerado nulo após decorrido um período igual a 5 vezes a maior constante de tempo do sistema, isto é,

$$\rho_N(\infty) = \rho(t) - \rho(\infty) \approx 0, \text{ para } t \geq 5T_{\max} \quad (2.51)$$

onde T_{\max} pode ser calculado como uma aproximação por:

$$T_{\max} = \frac{1}{|\sigma_s|}, \quad \sigma_s < 0 \quad (2.52)$$

onde $\frac{1}{|\sigma_s|}$ é a maior constante de tempo do sistema, e $\sigma_s < 0$ é a abscissa de estabilidade do sistema. O tempo de observação de transitório, como definido na equação (2.52), é utilizado nesta dissertação para fornecer uma orientação para a escolha do tempo final t_f de simulação para computar a resposta a degrau e a resposta ao impulso. O valor adotado para o tempo de observação do transitório nos exemplos numéricos desta dissertação é $t_f \geq 5T_{\max}$.

Exemplo 2.1: SISTEMA DE PRIMEIRA ORDEM

Este exemplo é uma reprodução do exemplo feito por Coelho em [19], que descreve muito bem o comportamento de um sistema de primeira ordem.

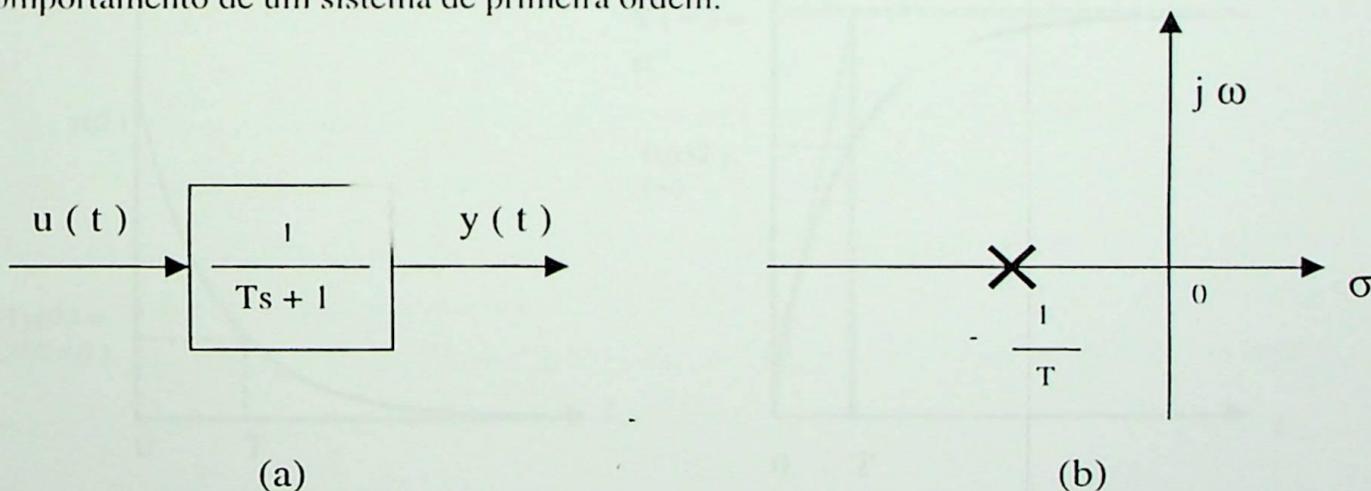


Figura 2.7 Sistema de primeira ordem estritamente próprio: (a) bloco representativo; (b) posição do pólo no plano s.

Considere o sistema de primeira ordem dado pela função de transferência

$$W(s) = \frac{1}{Ts + 1} \quad (2.53)$$

Este sistema é estritamente próprio e tem um único pólo em $s = -1/T$, que é então, em consequência, a abscissa de estabilidade,

$$\sigma_s = -\frac{1}{T} \quad (2.54)$$

A resposta livre é:

$$\rho_L(t) = \rho(0)e^{-t/T} \quad (2.55)$$

A resposta forçada é:

$$\rho_F(t) = a\left(1 - e^{-t/T}\right), \quad t \geq 0 \quad (2.56)$$

A resposta total é:

$$\rho(t) = \rho(0)e^{-t/T} - a \cdot e^{-t/T} + a, \quad t \geq 0 \quad (2.57)$$

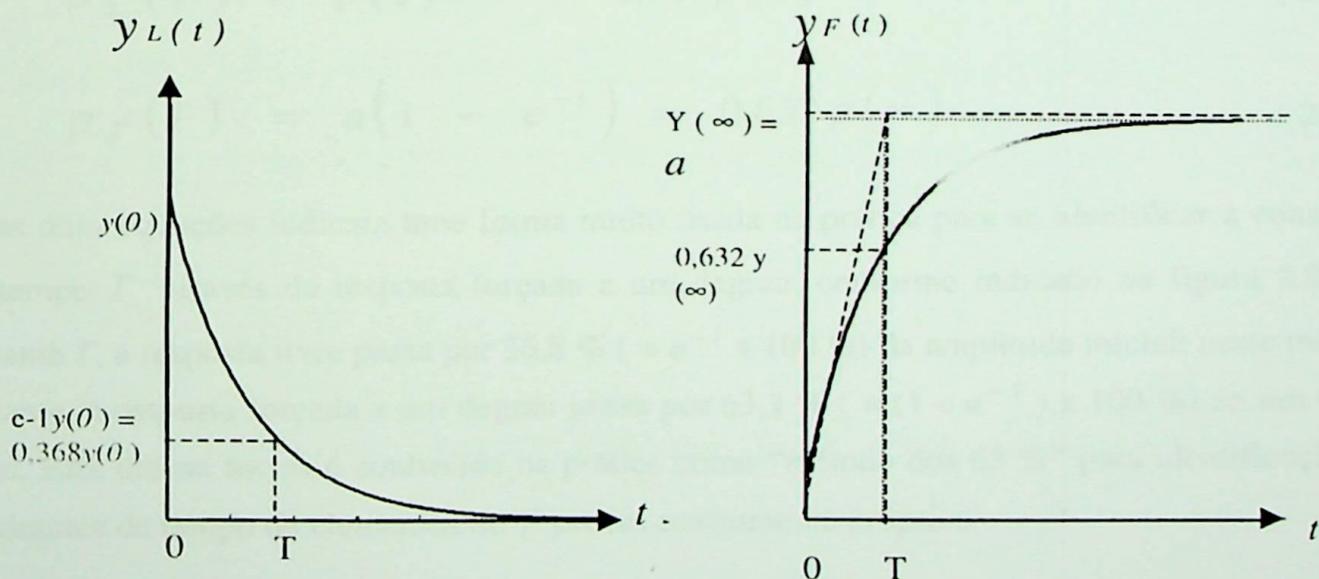


Figura 2.8 Parcelas da resposta de um sistema de primeira ordem estritamente próprio: (a) resposta livre; (b) resposta forçada para $u(t) = a u_{-1}(t)$.

Note-se que em regime permanente constante a resposta tende ao valor constante

$$\rho(\infty) = \lim_{t \rightarrow \infty} \rho(t) = a \quad (2.58)$$

Tabela 2.1 Valores da exponencial $e^{-t/T}$.

T	T	2T	3T	4T	5T	>5T
$e^{-t/T}$	0,368	0,135	0,0498	0,0183	0,0067	$\cong 0$

O pólo em $s = -1/T$ da função de transferência faz com que, tanto na resposta livre quanto na resposta forçada do sistema, apareça a exponencial $e^{-t/T}$. Valores desta exponencial em instantes múltiplos da constante de tempo são dados na Tabela 2.1.

Na prática, considera-se que o valor final é atingido em cinco vezes a constante de tempo (ver Tabela 2.1), o que dá um sentido prático a $t \rightarrow \infty$.

$t = T$ é um instante importante da resposta natural ou da resposta forçada a um degrau. Como pode ser visto nas equações (2.55) e (2.56) tem-se

$$\rho_L(T) = \rho(0) e^{-1} = 0,368 \rho(0) \quad (2.59)$$

$$\rho_F(T) = a(1 - e^{-1}) = 0,632 \rho(\infty) \quad (2.60)$$

Estas duas equações indicam uma forma muito usada na prática para se identificar a constante de tempo T através da resposta forçada a um degrau, conforme indicado na figura 2.8. No instante T , a resposta livre passa por 36,8 % ($= e^{-1} \times 100$ %) da amplitude inicial; neste mesmo instante, a resposta forçada a um degrau passa por 63,2 % ($= (1 - e^{-1}) \times 100$ %) de seu valor final. Esta última forma é conhecida na prática como “método dos 63 %” para identificação de constantes de tempo de elementos de 1ª ordem-estritamente próprios.

Para a entrada degrau, de acordo com as equações (2.57) e (2.58)

$$\rho(t) = \rho_N(t) + \rho(\infty) \quad (2.61)$$

como ilustra a figura 2.9.

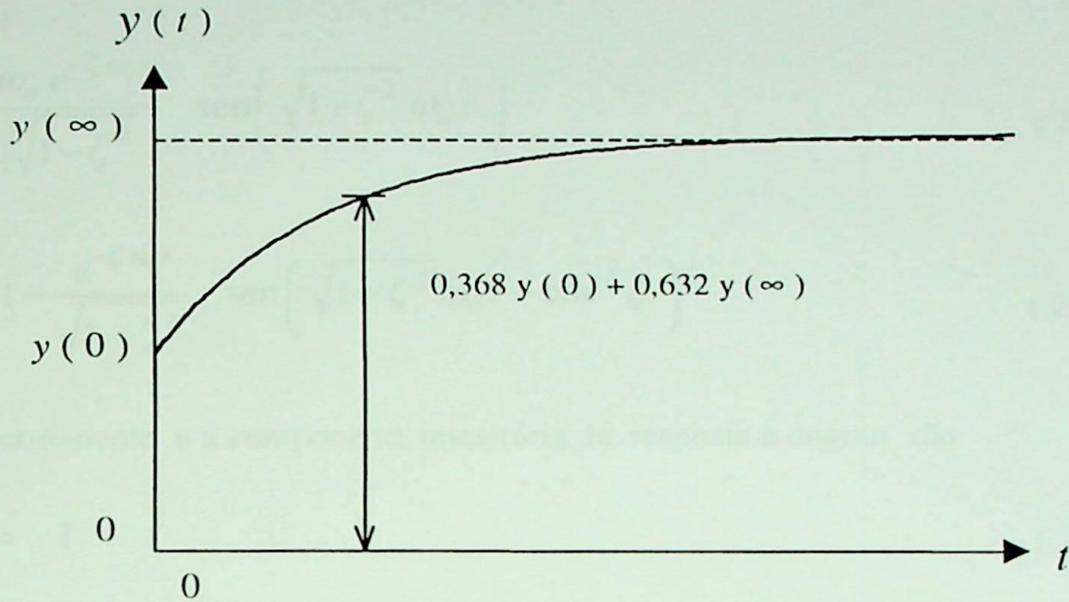


Figura 2.9 Resposta total de um sistema de 1ª ordem estritamente próprio, para entrada degrau:

Exemplo 2.2 SISTEMAS SUBAMORTECIDOS DE SEGUNDA ORDEM

Considere um sistema de segunda ordem dado pela função de transferência:

$$W(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.62)$$

Onde $\omega_n > 0$ e $\zeta > 0$ (para estabilidade). O sistema terá pólos complexos para $\zeta < 1$ (sistema sub-amortecido), e em tal caso pode-se escrever

$$W(s) = \frac{\omega_n^2}{\left(s + \zeta\omega_n + j\sqrt{1-\zeta^2}\omega_n\right)\left(s + \zeta\omega_n - j\sqrt{1-\zeta^2}\omega_n\right)} \quad (2.63)$$

onde $j = \sqrt{-1}$. É óbvio que

$$\sigma_s = -\zeta \omega_n \quad (2.64)$$

As respostas ao impulso e ao degrau são determinadas através de

$$w(t) = \frac{\omega_n e^{-\zeta \omega_n t}}{\sqrt{1-\zeta^2}} \operatorname{sen}\left(\sqrt{1-\zeta^2} \omega_n t\right) \quad (2.65)$$

$$\rho(t) = 1 - \frac{e^{-\zeta \omega_n t}}{\sqrt{1-\zeta^2}} \operatorname{sen}\left(\sqrt{1-\zeta^2} \omega_n t + \cos^{-1} \zeta\right) \quad (2.66)$$

O valor de regime permanente e a componente transitória da resposta a degrau são

$$\rho(\infty) = 1 \quad (2.67)$$

$$\rho(t) - \rho(\infty) = -\frac{e^{-\zeta \omega_n t}}{\sqrt{1-\zeta^2}} \operatorname{sen}\left(\sqrt{1-\zeta^2} \omega_n t + \cos^{-1} \zeta\right) \quad (2.68)$$

O sistema é estritamente próprio, e a resposta ao impulso e a componente transitória da resposta a degrau são limitadas pela mesma exponencial como segue:

$$|w(t)| \leq \frac{\omega_n e^{-\zeta \omega_n t}}{\sqrt{1-\zeta^2}} \quad (2.69)$$

$$|\rho(t) - \rho(\infty)| \leq \frac{e^{-\zeta \omega_n t}}{\sqrt{1-\zeta^2}} \quad (2.70)$$

O tempo de observação do transitório é calculado de acordo com equação (2.52) e pela conclusão de que o tempo final deve respeitar $t_f \geq 5 T_{max}$, a partir disso tem-se que o tempo T_s de observação do transitório será dado por:

$$T_s = \frac{5}{\zeta \omega_n} \quad (2.71)$$

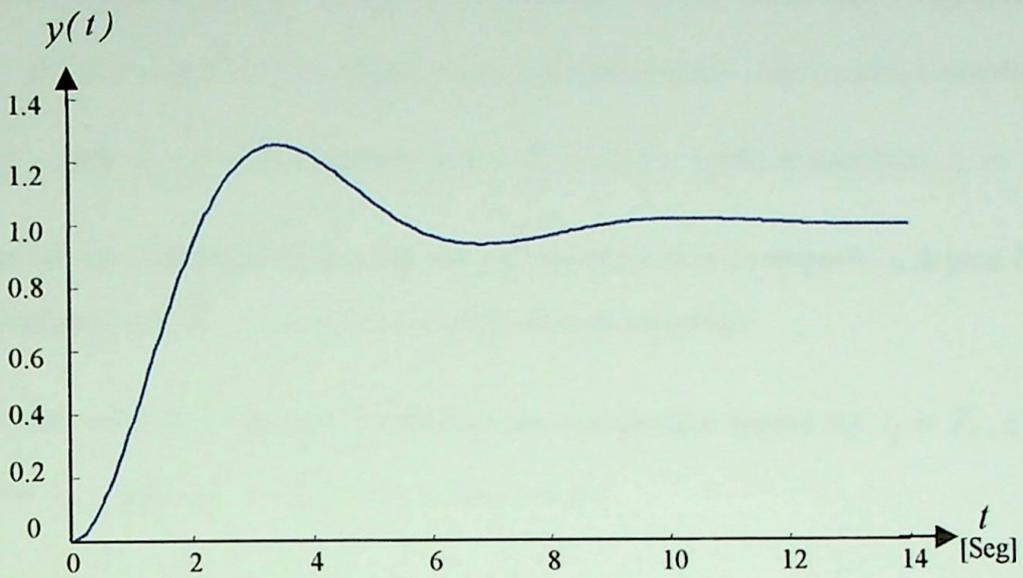


Figura 2.10 Resposta a degrau de um sistema de Segunda ordem

$$(\omega_n = 1, \zeta = 0,4);$$

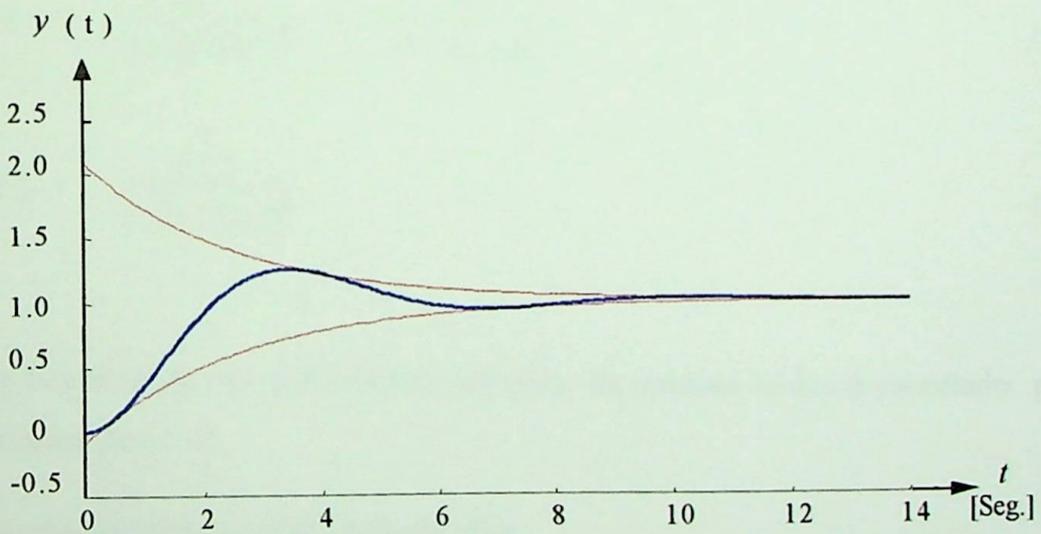


Figura 2.11 Transitório da resposta a degrau (linha azul),e envoltórias (linha vermelha).

Com a definição dada na equação 2.67 as equações 2.66 e 2.68 são equivalentes. A figura 2.10 mostra a resposta a degrau $\rho(t)$ e na figura 2.11 a resposta ao impulso $w(t)$ e a parcela transitória $\rho(t) - \rho(\infty)$ da resposta a degrau, juntamente com as curvas envoltórias $\pm e^{-\zeta t} / \sqrt{1-\zeta^2}$, para $\zeta = 0,4$, conseqüentemente, $T_s = 12,5$; então se escolheu $t_f = 14$.

Novamente se vê que a resposta ao impulso e a parcela transitória da resposta a degrau ficam muito pequenas para $t \geq T_s$, e podem ser consideradas desprezíveis.

Então é suficiente calcular a resposta ao impulso ou a resposta a degrau até $t_f = T_s$, e assumir depois que $t = T_s$ $\rho(t) = \rho(\infty)$ e $w(t) = 0$..

Esta conclusão é confirmada nos próximos dois exemplos.

Exemplo 2.3 SISTEMA DE CONTROLE COM REALIMENTAÇÃO PADRÃO

Um Sistema de Controle com Realimentação Padrão (do inglês SFCS- *Standard Feedback Control Systems*), como definido pelas equações (2.15) a (2.16) (figura 2.1), tem as seguintes funções de transferência :

$$G(s) = \frac{10}{(s+1)(s+5)} = \frac{10}{s^2 + 6s + 5} \quad (2.72)$$

$$K(s) = \frac{s+1}{s(0,5s+1)} \quad (2.73)$$

Na função de transferência de malha aberta, um pólo da unidade básica é cancelado por um zero do controlador em $s=1$.

As funções de transferência de malha fechada são:

$$\frac{Y_G(s)}{F(s)} = W_y(s) = \frac{20}{a(s)} \quad (2.74)$$

$$\frac{U(s)}{F(s)} = W_u(s) = \frac{2(s+1)(s+5)}{a(s)} \quad (2.75)$$

$$\frac{E(s)}{F(s)} = W_e(s) = \frac{s(s+2)(s+5)}{a(s)} \quad (2.76)$$

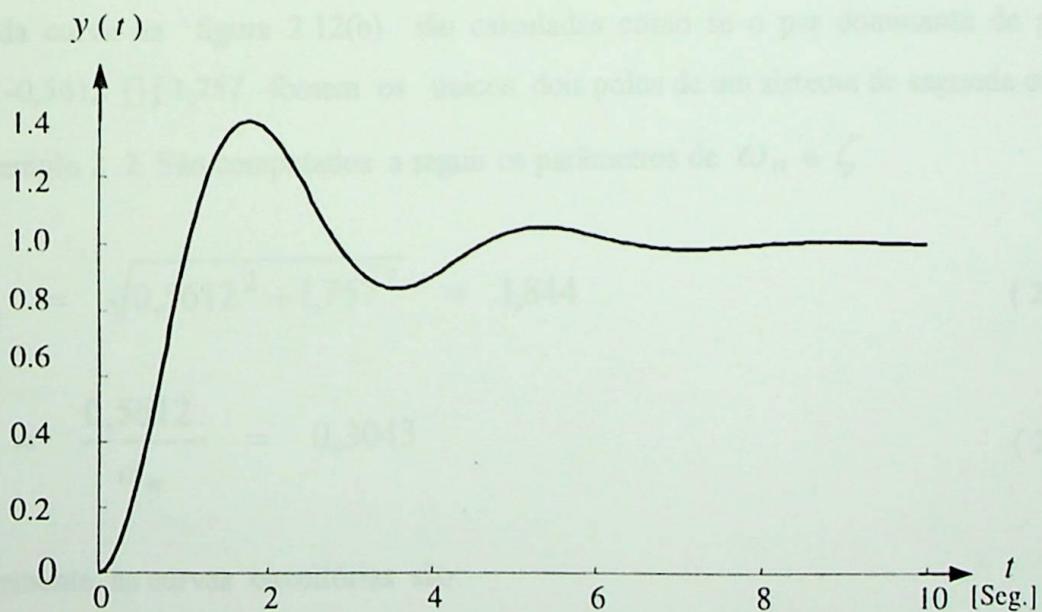
onde

$$\begin{aligned} a(s) &= s^3 + 7s^2 + 10s + 20 \\ &= (s + 0,5612 + j1,757)(s + 0,5612 - j1,757)(s + 5,878) \end{aligned} \quad (2.77)$$

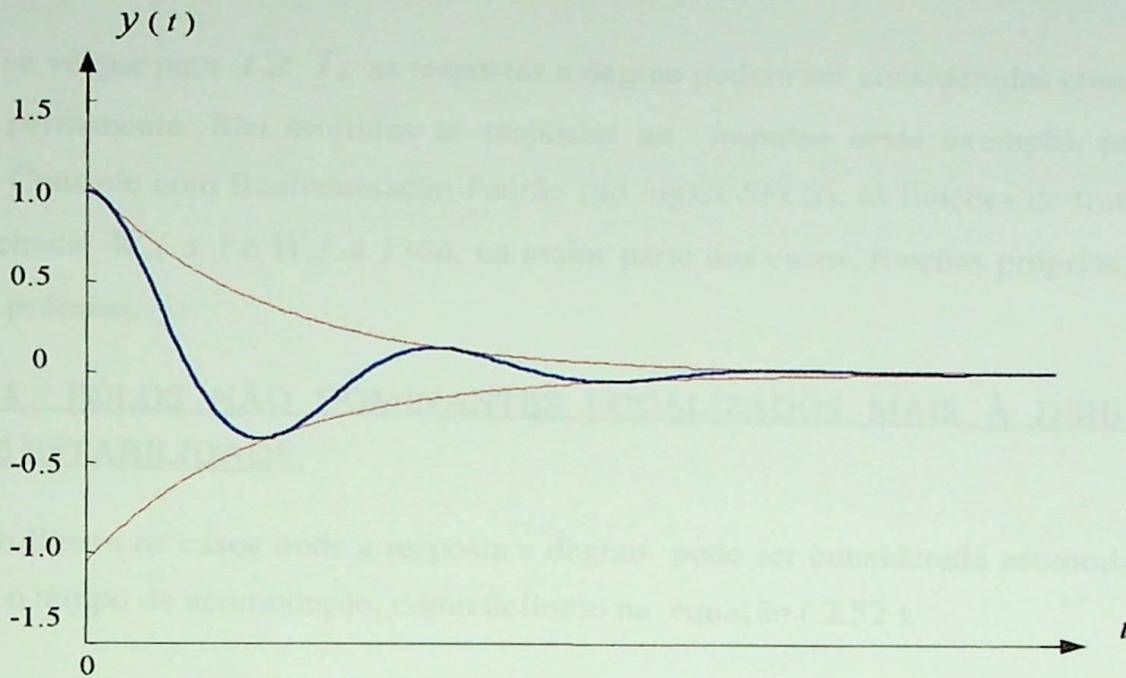
Usando-se a equação (2.40) pode-se calcular os valores de regime permanente das respostas de malha fechada das respostas a degrau, obtendo-se 1,0 para a saída produzida, 0,5 para o controle e zero para o erro (o controlador tem ação integral). A abscissa de estabilidade e o tempo de observação de transitório são:

$$\sigma_s = -0,5612 \quad (2.78)$$

$$T_s = \frac{5}{|0,5612|} = 8,91 \quad (2.79)$$



(a)



(b)

Figura 2.12 (a) Resposta a degrau de malha fechada ; (b) Transitório do Controle (linha azul) e envoltórias (linha vermelha)

As respostas a degrau de malha fechada são mostradas nas figuras 2.12(a) e 2.12(b). As envoltórias da curva na figura 2.12(b) são calculadas como se o par dominante de pólos complexos $-0,5612 \pm j 1,757$ fossem os únicos dois pólos de um sistema de segunda ordem como no Exemplo 2.2. São computados a seguir os parâmetros de ω_n e ζ :

$$\omega_n = \sqrt{0,5612^2 + 1,757^2} = 1,844 \quad (2.80)$$

$$\zeta = \frac{0,5612}{\omega_n} = 0,3043 \quad (2.81)$$

Conseqüentemente, as curvas envoltórias são:

$$\pm \frac{e^{-\zeta \omega_n t}}{\sqrt{1-\zeta^2}} = \pm 1,05 e^{-0,5612 t} \quad (2.82)$$

Novamente se vê que para $t \geq T_s$ as respostas a degrau podem ser consideradas como estando em regime permanente. São omitidas as respostas ao impulso neste exemplo, porque em Sistemas de Controle com Realimentação Padrão (do inglês *SFCS*), as funções de transferência de malha fechada $W_e(s)$ e $W_u(s)$ são, na maior parte dos casos, funções próprias, mas não estritamente próprias.

Exemplo 2.4 PÓLOS NÃO DOMINANTES LOCALIZADOS MAIS À DIREITA NA REGIÃO DE ESTABILIDADE.

Este exemplo ilustra os casos onde a resposta a degrau pode ser considerada acomodada antes de decorrido o tempo de acomodação, como definido na equação (2.52).

O sistema é novamente um Sistema de Controle com Realimentação Padrão (do inglês *SFCS*), e a unidade básica $G(s)$ é a mesma do último exemplo (equação (2.72)). O controlador para este exemplo é:

$$K(s) = \frac{0,9(1,1s + 1)}{s(0,02s + 1)} \quad (2.83)$$

As funções de transferência de malha fechada são as seguintes :

$$W_y(s) = \frac{9,9(s + 0,909)}{A(s)} \quad (2.84)$$

$$W_u(s) = \frac{0,99(s + 0,909)(s + 1)(s + 5)}{A(s)} \quad (2.85)$$

$$W_e(s) = \frac{0,02(s + 1)(s + 5)(s + 50)}{A(s)} \quad (2.86)$$

onde $A(s) = 0,02s^4 + 1,122s^3 + 6,1s^2 + 14,9s + 9$ (2.87)

$$A(s) = 0,02(s+0,8594)(s+2,461+j2,09)(s+2,461-j2,09)(s+50,22) \quad (2.88)$$

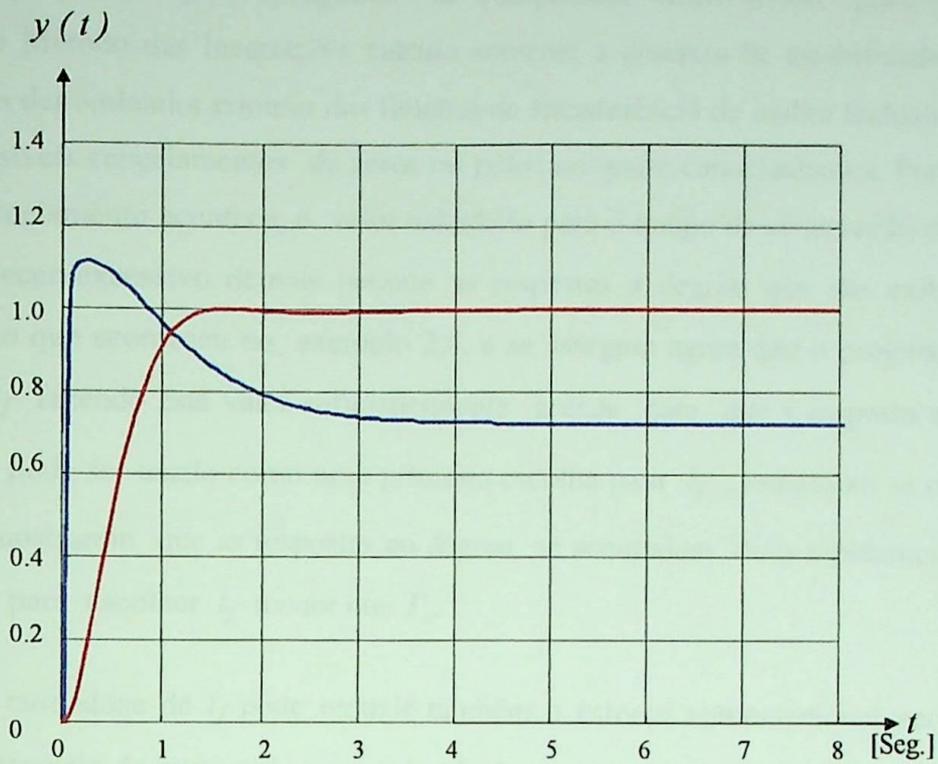
Devido ao controlador ter ação integral, os valores de regime permanente são iguais aos do exemplo 2.3. Porém, a abscissa de estabilidade e o tempo de acomodação são :

$$\sigma_s = -0,8594$$

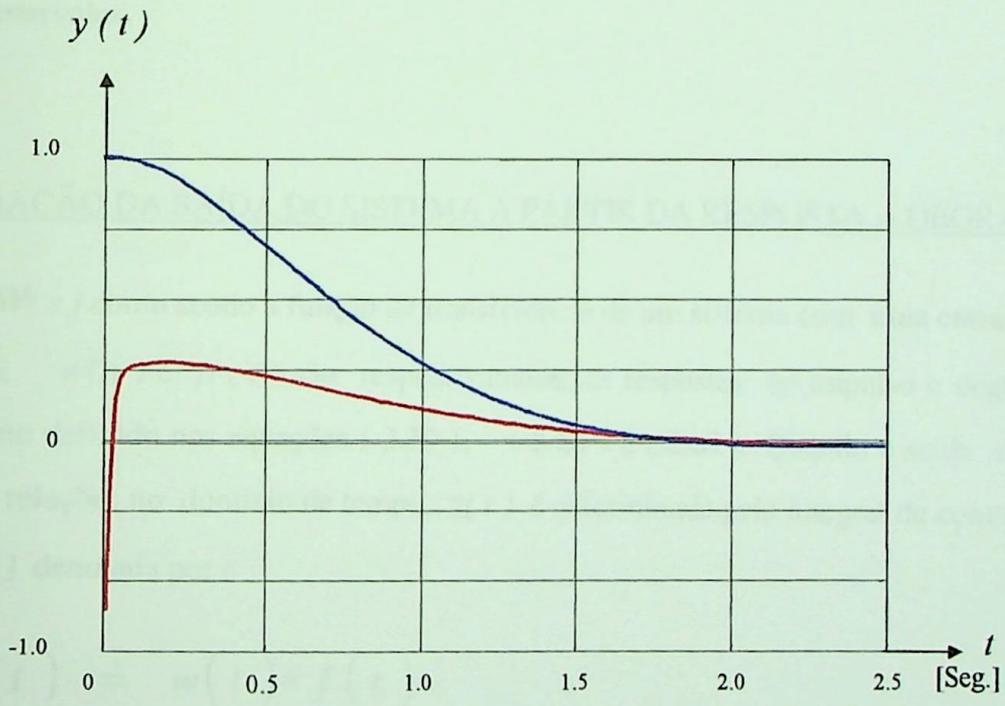
$$T_s = \frac{5}{|\sigma_s|} = 5,82$$

A figura 2.13 (a) mostra que a resposta a degrau da saída produzida e do controle para o tempo final de simulação é $t_f = 5$. Nesta figura as respostas a degrau parecem ter se acomodado em torno de $t = 2,0$, embora o tempo de observação do transitório previsto seja de $T_s = 5,82$. Então, traça-se os gráficos da parcela transitória das respostas a degrau (erro e transitório do controle), junto com as exponenciais $\pm e^{\sigma_s t}$, para $t_f = 2,5$. As curvas são mostradas na figura 2.13 (b), onde observa-se que as respostas a degrau podem ser consideradas acomodadas realmente para $t = 2,5$, embora a exponencial não esteja acomodada naquele momento e precisará do decurso total de T_s para acomodar-se e ficar desprezível.

Isto é explicado pela presença, nas funções de transferência de malha fechada, de zeros em $s = -0,909$ e $s = -1$, que estão na proximidade dos pólos em $s = \sigma_s = -0,8594$. Estes zeros reduzem a influência deste pólo de tal modo que ele tem pequena influência na resposta do sistema. Neste exemplo o segundo pólo mais à direita é o complexo em $s = -2,461 + j2,09$ e o seu conjugado (equação (2.88)); este par complexo é responsável pela parcela mais relevante do transitório. Pode-se ver que se $\sigma_s = -0,8594$ fosse substituído pelo valor de $-2,461$ no cálculo do tempo de acomodação, o resultado seria $T_s = 2,03$, um valor muito mais consistente com o aspecto da figura 2.13 (b).



(a)



(b)

Figura 2.13 Resposta degrau de malha fechada para o Exemplo 2.4:

- (a) Saída produzida (linha vermelha) e controle (linha azul)
- (b) Erro (linha vermelha), transitório do controle (linha azul)

Deve-se enfatizar aqui que o programa de computador desenvolvido para o Projeto de Controle pelo Método das Inequações calcula somente a abscissa de estabilidade a partir do polinômio do denominador comum das funções de transferência de malha fechada, e não leva em conta possíveis cancelamentos de zeros ou pólos ou quase cancelamentos. Por esta razão, quando o cancelamento acontece, o valor calculado para o tempo de observação do transitório T_s pode parecer excessivo demais perante as respostas a degrau que são exibidas. Isto é precisamente o que aconteceu no exemplo 2.4, e se assegura agora que o projetista escolhe o tempo final t_f fazendo este valor suficientemente grande para que a resposta ao degrau se acomode. T_s pode ser usado como uma primeira escolha para t_f , entretanto se os resultados do projetista mostrarem que as respostas ao degrau se acomodam mais rapidamente, ele deve ter a liberdade para escolher t_f menor que T_s .

Uma redução cuidadosa de t_f pode reduzir também o esforço computacional necessário no projeto, se o intervalo de integração para a simulação e o número de intervalos for reduzido, ou pode melhorar a precisão dos resultados se o número de intervalos é mantido, para reduzir o valor dos intervalos.

2.3 AVALIAÇÃO DA SAÍDA DO SISTEMA A PARTIR DA RESPOSTA A DEGRAU

Considere $W(s)$ como sendo a função de transferência de um sistema com uma entrada $f(t)$ e saída $z(t)$; $w(t)$ e $\rho(t)$ são respectivamente as respostas ao impulso e degrau de tal sistema, como definido nas equações (2.39), (2.66) e (2.67). Quando a saída é avaliada pelo uso de relações no domínio de tempo, $z(t)$ é determinada pela integral de convolução de $w(t)$ e $f(t)$ denotada por :

$$z(t) = w(t) * f(t) \quad (2.89)$$

$$w(t) * f(t) = \int_0^t w(\lambda) f(t-\lambda) d\lambda \quad (2.90)$$

Note-se que $w(t) * f(t) = f(t) * w(t)$ (2.91)

A transformada de Laplace de uma convolução é:

$$\mathcal{L} [w(t) * f(t)] = W(s)F(s) \quad (2.92)$$

Zakian [43,44] propôs expressar $z(t)$ em termos da resposta a degrau.

Considerando que

$$sF(s) = \mathcal{L} [f^{(1)}(t)] + f(0) \quad (2.93)$$

onde $f^{(1)}(t)$ denota a primeira derivada de $f(t)$ Conseqüentemente, de acordo com equações (2.89) e (2.92) nós podemos escrever:

$$Z(s) = \frac{W(s)}{s} f(0) + \frac{W(s)}{s} \mathcal{L} [f^{(1)}(t)] \quad (2.94)$$

A transformada inversa de Laplace desta equação é:

$$z(t) = \rho(t)f(0) + \rho(t) * f^{(1)}(t) \quad (2.95)$$

Se o sistema é estável, $\rho(t)$ tem um valor de regime permanente $\rho(\infty)$ e a equação (2.95) pode ser reescrita como segue :

$$z(t) = [\rho(t) - \rho(\infty)]f(0) + \rho(\infty)f(t) + [\rho(t) - \rho(\infty)] * f^{(1)}(t) \quad (2.96)$$

ou

$$z(t) = [\rho(t) - \rho(\infty)]f(0) + \rho(\infty)f(t) + \int_0^t [\rho(\lambda) - \rho(\infty)]f^{(1)}(t-\lambda)d\lambda \quad (2.97)$$

Para sistemas próprios e estáveis a equação (2.97) pode ser usada mais convenientemente que a equação (2.89), porque a resposta a degrau transitória $\rho(t) - \rho(\infty)$ é finita para todo t e decai a zero quando $t \rightarrow \infty$. Em outras palavras, $\rho(t) - \rho(\infty)$ tem as propriedades de uma resposta a impulso estável e estritamente própria. Na equação (2.89), se $w(t)$ não for

estritamente própria ela terá um componente impulsivo o qual impediria a solução numérica da integral de convolução.

2.4 CRITÉRIOS DE DESEMPENHO

Os índices de desempenho utilizados nesta dissertação estão formalmente definidos em [19]. Estes índices são utilizados para avaliar o comportamento do sistema quando submetido a uma excitação padronizada, as características da resposta a degrau de um sistema dinâmico foram obtidas através da função CARACRD da Toolbox de controle CACCON [4], escrita para o ambiente MATLAB. O projeto é formulado em termos de um conjunto de inequações que definem as limitações de operação e especificações que devem ser cumpridas pelo sistema. Quando o programa de projeto consegue determinar valores dos parâmetros para os quais o sistema cumpre tais especificações e limitações, o programa baseado neste critério terá encontrado uma possível solução do problema de projeto. Estas características de avaliação são descritas na seguinte seção que é uma revisão dos conceitos formulados em [19].

2.4.1 Características da Resposta a Degrau

O desempenho de sistemas dinâmicos durante transitórios provocados por mudanças de pontos de operação e perturbações é avaliado e especificado através das características da resposta a degrau (CRD's).

Estas características são definidas com base em três considerações fundamentais:

1. O sistema possui elementos armazenadores de energia, que o leva a ter uma certa inércia, e, por isto, a não responder instantaneamente a variações na entrada. Esta consideração implica resposta com valor inicial nulo:

$$y(0) = 0 \quad (2.98)$$

Em sistemas racionais, esta consideração implica que a Função de Transferência $W(s)$ é estritamente própria ($\Rightarrow W(\infty) = 0$).

2. O sistema é estável; portanto, a resposta a degrau é finita para todo t . Em um sistema linear com Função de Transferência racional, esta consideração implica $W(s)$ com todos os pólos com parte real negativa.
3. As CRD's são definidas em termos de relações entre determinados instantes de tempo e frações do valor final, que por isto deve ser diferente de zero:

$$y(\infty) = \lim_{t \rightarrow \infty} y(t) \neq 0 \quad (2.99)$$

Em sistemas lineares, esta consideração implica $W(0) \neq 0$.

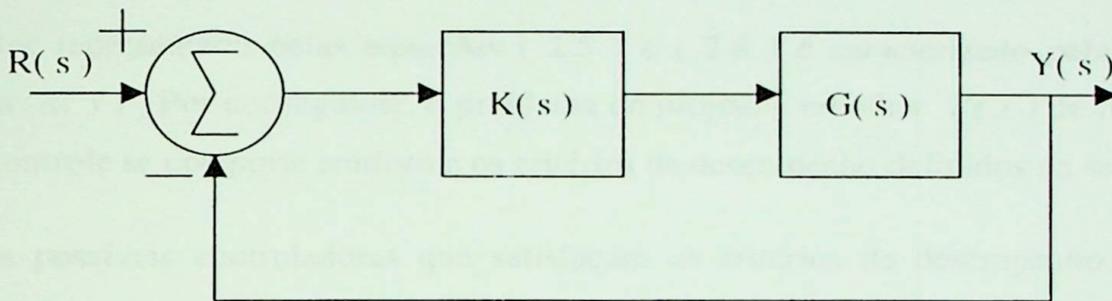


Figura 2.14 Estrutura básica do sistema de controle

Esta última consideração implica que o valor final $y(\infty)$ tem caráter de um objetivo a ser atingido, como acontece em sistemas de controle. Por exemplo, as CRD's são de fundamental importância para a especificação de sistemas de controle com a estrutura dada na figura 2.14.

Onde :

$$K(s) = \frac{N_K(s)}{D_K(s)} \quad (2.100)$$

$$G(s) = \frac{N_G(s)}{D_G(s)} \quad (2.101)$$

A Função de Transferência deste sistema é

$$W(s) = \frac{Y(s)}{R(s)} = \frac{N_K(s)N_G(s)}{D_K(s)D_G(s) + N_K(s)N_G(s)} \quad (2.102)$$

Normalmente, considera-se que $G(s)$ é estritamente própria e que $K(s)$ é própria. O que leva $W(s)$ a ser estritamente própria. Neste sistema a saída $y(t)$ é uma variável controlada, e a amplitude R_0 do degrau de entrada é uma referência que determina qual deverá ser o valor final dessa variável.

O programa desenvolvido nesta dissertação calcula os índices de desempenho da resposta a degrau a partir da Função de Transferência da equação (2.102). Estes índices de desempenho estão amplamente explicados no capítulo 3.

2.5 O PROBLEMA DE PROJETO

O controlador representado pelas equações (2.5) e (2.6) é caracterizado pela função de transferência $K(s)$. Por conseguinte, o problema de projeto é escolher $K(s)$ de forma que o sistema de controle se comporte conforme os critérios de desempenho definidos na seção 2.4.

De todos os possíveis controladores que satisfaçam os critérios de desempenho, prefere-se aquele que tenha a menor complexidade. A complexidade de um controlador é definida como o esforço computacional que é exigido ao programa para gerar o controle. Neste caso, o controlador é considerado como um programa em tempo real. Aquele esforço aumenta com a ordem do controlador, que é o grau do polinômio do denominador de $K(s)$. Então, associa-se a complexidade do controlador com sua ordem: quanto mais alta a ordem, mais complexo será o controlador. Obviamente, o controlador mais simples possível tem ordem zero, ou seja, é um controlador proporcional.

O controlador é caracterizado então por meio da sua ordem (grau de complexidade), os valores constantes na função de transferência do controlador e o vetor de projeto. Este vetor, que é denotado por p , tem componentes reais e dimensão $[p]$ como segue:

$$p = [p_1 \ p_2 \ \cdots \ p_{[p]}]^T \quad (2.103)$$

Onde o índice superior T denota a função transposição do vetor de projeto. Os componentes de $p_i, i = 1, 2, \dots, [p]$ são os parâmetros que deverão ser ajustados pelo programa de projeto.

Com os valores apropriados dos parâmetros do vetor de projeto, encontrados pelo programa, o controlador pode ter qualquer uma das seguintes estruturas de controle : proporcional integral, proporcional integral derivativo - amortecido, avanço de fase, atraso de fase, ou combinações destes.

O controlador pode estar sujeito a certas restrições que são expressas na forma de inequações, como segue :

$$Q_i(p) \leq q_i, \quad i = 1, 2, \dots, [Q] \quad (2.104)$$

onde os Q_i são funções de valores reais do vetor de projeto e os q_i são números reais. Essas restrições estão relacionadas com as possíveis limitações para implementação do controlador, devidas ao uso de componentes físicos e outras considerações que podem, de algum modo, restringir os valores dos parâmetros do vetor de projeto.

Nesta dissertação, representa-se sempre o controlador através de uma função de transferência que poderá ter uma das seguintes formas:

Forma No.1

$$K(s) = \frac{p_1 (1 + p_2 s) \cdots (1 + p_{N+1} s)}{s^L (1 + p_{N+2} s) \cdots (1 + p_{N+M+1} s^M)}, \quad N \leq L + M \quad (2.105)$$

Forma No.2

$$K(s) = \frac{p_1 + p_2 s + \cdots + p_{N+1} s^N}{1 + p_{N+2} s + \cdots + p_{N+M+1} s^M}, \quad N \leq M \quad (2.106)$$

As restrições $N \leq L + M$ e $N \leq M$ são impostas para assegurar que a função de $K(s)$ seja sempre própria.

É dito que um controlador é viável se, e somente se, houver um valor do vetor de projeto para o qual o sistema de controle satisfaça os critérios de desempenho e as inequações (2.104). Tal valor do vetor de projeto é considerado como um valor admissível.

Uma solução exata do problema de projeto é um controlador viável de menor complexidade e um valor admissível do vetor de projeto.

Resumindo, a solução do problema de projeto, como a atual proposta, consistirá em satisfazer simultaneamente as seguintes quatro condições :

I) O controlador estará sujeito às restrições expressas pelas inequações (2.104), que em consequência deverão ser respeitadas.

II) O sistema de malha fechada deve ser estável, caso contrário o controle $u(t)$ e o erro $e(t)$ não serão limitados. A condição para estabilidade pode ser expressa por

$$\sigma_s \leq 0$$

onde σ_s é a abscissa de estabilidade do sistema, definida como segue :

$$\sigma_s = \max_i \{ \text{Re}(\lambda_i), \quad i = 1, 2, \dots, n \}$$

onde os λ_i denotam os n pólos de malha fechada do sistema que são as raízes da equação característica

$$1 + K(s)G(s) = 0$$

III) Para que os resultados do projeto sejam confiáveis, deve-se ter a certeza de que o modelo linear seja um modelo válido e, em consequência, que a inequação (2.14) seja respeitada.

IV) O problema de projeto estará resolvido se a inequação (2.14) for satisfeita simultaneamente com as inequações (2.104), que representam as restrições de operação do sistema, as quais são calculadas através dos índices de desempenho das características da resposta a degrau.

É dito que um controlador é viável se, e somente se, existir um valor do vetor de projeto tal que o sistema de controle satisfaça estas quatro condições. Tal valor do vetor de projeto é chamado de solução.

2.6 O MÉTODO DAS INEQUAÇÕES

A nova implementação do Método das Inequações foi proposta para resolver o problema de projeto descrito na última seção. A proposta sugerida consiste em gerar um controlador de ordem de complexidade mínima. Claramente, o projeto começa com um controlador de grau de complexidade zero, significando que o controlador com a mais baixa complexidade possível é somente um controlador proporcional com ganho d_K .

Se uma sucessão de vários controladores é conseguida e ela contém controladores viáveis então uma solução exata para o problema de projeto será fornecida por um controlador viável que tenha o menor grau de complexidade.

Na prática, para se determinar um controlador que satisfaça às exigências da seção anterior, teria que ser projetado e necessariamente testado para ver se este controlador cumpriria com as exigências de desempenho desejadas. Na atual implementação do Método das Inequações é possível reduzir-se o esforço computacional, se a condição necessária for sacrificada, e uma condição suficientemente fácil de ser computada para viabilidade for desenvolvida.

O teste a ser usado faz uso de um conjunto de inequações da forma

$$\phi_i(p) \leq c_i, \quad i = 1, 2, \dots, M \quad (2.107)$$

onde os ϕ_i são funções reais, os c_i são números reais e p é o vetor do projeto. As inequações (2.107) são ditas possíveis de ser resolvidas se, e somente se, houver pelo menos um valor do vetor de projeto para o qual todas as inequações são satisfeitas. O valor de tal vetor é chamado de Φ -admissível.

As inequações (2.107) fornecem uma condição suficiente para viabilidade no sentido de que se as inequações são resolvíveis então o controlador é viável. Note-se que todos os pontos fundamentais do problema de projeto de controle, mencionados nas seções anteriores a este

capítulo, foram expressos pelas inequações (2.14), (2.23), (2.102) e (2.104) , as quais se ajustam na forma geral das inequações (2.107).

Um teste de suficiência, necessariamente, não leva a um controlador viável que seja uma solução exata do problema de projeto. Porém, se o teste de suficiência for suficientemente severo, então a solução do problema de projeto obtida pela sua utilização será satisfatória na prática.

O projeto através do cumprimento de especificações da resposta a degrau é o objeto de estudo desta dissertação. É esperado que a análise e o desenvolvimento junto com resultados obtidos dêem uma melhor noção das potencialidades desta nova proposta para projetos de sistemas de controle.

CAPÍTULO 3

PROJETO DE SISTEMAS DE CONTROLE

3.1 ASPECTOS COMPUTACIONAIS

O capítulo 2 é a base para o projeto de sistemas de controle pelo Método das Inequações (MI). Neste capítulo, descreve-se como todo aquele material é usado no projeto de sistemas de controle. Os detalhes computacionais e os algoritmos orientados para a implementação do MI são também apresentados neste capítulo. E exemplos de aplicação prática são apresentados no capítulo seguinte.

O projeto de sistemas de controle com realimentação padrão pelo MI, usando o programa desenvolvido junto com esta dissertação, tem como objetivo o cumprimento de especificações da resposta a degrau do sistema em malha fechada.

O problema do projeto consiste em determinar um controlador que consiga estabilizar um sistema de controle com realimentação padrão, que esteja submetido a um tipo de entrada padronizada; nesta dissertação foi utilizada a entrada a degrau como a da equação (2.12). O objetivo é a estabilização do sistema após a aplicação da entrada degrau, e isto somente é possível se o sistema global, com a Função de Transferência dada pela equação (2.102), for capaz de satisfazer um conjunto de inequações, as quais representam as especificações e as restrições que o sistema deverá satisfazer. O modelo linear foi discutido na seção 2.1 e o diagrama de blocos é o mesmo do capítulo 2, figura 2.1. É conveniente repetir aqui as equações essenciais do modelo linear. A função de transferência da Unidade básica é

$$G(s) = \frac{Y_G(s)}{U(s)} \quad (3.1)$$

sujeita às seguintes restrições

$$\frac{K_l}{|R_0|} \leq u(t) \leq \frac{K_u}{|R_0|}, \quad t \geq 0. \quad (3.2)$$

Uma aplicação do Método das Inequações [9] mostrou um caso de saturação assimétrica, no qual se sugere restringir o controle $u(t)$, como na inequação (3.2), em lugar da inequação (2.13) $\left(|u(t)| \leq \frac{K}{|R_0|} \right)$. Por esta razão no programa de projeto através de resposta a degrau, onde o controle $u(t)$ está disponível à limitação do sinal de controle dada pela inequação (3.2).

A função de transferência do controlador é

$$K(s) = \frac{U(s)}{E(s)} \quad (3.3)$$

$G(s)$ é uma função estritamente própria e $K(s)$ é uma função própria. $Y_G(s)$, $U(s)$ e $E(s)$ são as transformadas de Laplace da saída produzida $y_G(t)$, do controle $u(t)$ e do erro $e(t)$, respectivamente. O erro é dado por

$$e(t) = f(t) - y_G(t) \quad (3.4)$$

sendo

$$f(t) = r(t) - d(t) \quad (3.5)$$

onde $r(t)$ é a referência, $d(t)$ é o distúrbio e $f(t)$ é a entrada. $G(s)$ tem parâmetros constantes, e o problema de projeto consiste em achar um vetor de projeto para o controlador $K(s)$ que faça com que o sistema cumpra certas especificações de desempenho. O controlador poderá ter uma das seguintes formas:

Forma número 1:

$$K(s) = \frac{p_1 (1 + p_2 s) \cdots (1 + p_{N+1} s)}{s^L (1 + p_{N+2} s) \cdots (1 + p_{N+M+1} s^M)}, \quad N \leq L + M \quad (3.6)$$

Forma número 2:

$$K(s) = \frac{p_1 + p_2 s + \dots + p_{N+1} s^N}{1 + p_{N+2} s + \dots + p_{N+M+1} s^M}, \quad N \leq M \quad (3.7)$$

Os parâmetros do controlador são agrupados em um vetor de projeto

$$p = [p_1 \ p_2 \ \dots \ p_{[p]}]^T \quad (3.8)$$

onde o índice superior T indica transposição. Para ambas formas do controlador $[p] = 1+N+M$, os parâmetros do controlador terão as suas variações limitadas através da seguinte inequação composta

$$c_i' \leq p_i \leq c_i, \quad i = 1, 2, \dots, m \quad (3.9)$$

O programa perguntará ao projetista se deseja incluir ou não limites nos parâmetros do controlador; se a resposta for afirmativa, o programa pedirá os valores dos limites; caso a resposta seja negativa, estes limites serão automaticamente fixados pelo programa em $c_i' = -10^{-35}$ e $c_i = 10^{-35}$.

As restrições ao vetor de parâmetros p juntamente com as especificações de desempenho são expressas como um conjunto de inequações $\phi_i(p) \leq c_i$, para $i = 1, 2, \dots, [p]$. Como algumas das funções $\phi_i(p)$ exigirão um limite inferior c_i' e um limite superior c_i , é conveniente definir as inequações em pares, como segue

$$\left. \begin{array}{l} -\phi_i(p) \leq -c_i' \\ \phi_i(p) \leq c_i \end{array} \right\} \quad i = 1, 2, \dots, m \quad (3.10)$$

Neste sentido, evita-se definir funções extras como $\phi_j(p) = -\phi_i(p)$, o que representa uma economia de esforço computacional. Quando uma funcional $\phi_i(p)$ tiver um só limite, o outro limite será fixado internamente pelo programa em um valor constante apropriado, de modo que esse limite nunca venha a ser ultrapassado pelo

funcional. Os pares de inequações (3.10) podem ser expressos como as inequações compostas

$$c_i' \leq \phi_i(p) \leq c_i, \quad i = 1, 2, \dots, m \quad (3.11)$$

Esta forma será usada para expressar as restrições no controlador e as especificações de desempenho, deste ponto em diante, mas enfatiza-se que as inequações a serem resolvidas são aquelas dadas na expressão (3.10).

O número das inequações (3.11) será $m = [p] + \ell + 1$.

Onde ℓ representa o número de restrições e especificações que o projeto deverá satisfazer, no programa desenvolvido $\ell = 7$, e p representa a quantidade de parâmetros do vetor de projeto.

As $[p]$ primeiras inequações expressarão as restrições nos parâmetros do controlador dados pelas inequações (3.9). Claramente

$$\phi_i(p) = p_i, \quad i = 1, 2, \dots, [p] \quad (3.12)$$

O desempenho do sistema será caracterizado pelo desempenho dos funcionais $\phi_i(p)$, $i = 1, 2, \dots, [p] + \ell$ e dos índices de desempenho das características da resposta a degrau do sistema, as quais serão explicadas com detalhes nas seções seguintes deste capítulo, e que descrevem a proposta de projeto. O último funcional será a abscissa de estabilidade, definido pela inequação (2.23). Conseqüentemente tem-se

$$\phi_m(p) = \sigma_s \quad (3.13)$$

O algoritmo usado para resolver as inequações será o Processo dos Contornos Móveis (PCM) [17], descrito nesta seção. A ordem na qual as funcionais estão definidas não afeta os resultados, e conseqüentemente não há maior inconveniente em definir a equação (3.13) como a última funcional. Porém, os funcionais $\phi_i(p)$, $i = [p] + 1, \dots, [p] + \ell$ são definidos somente quando o sistema em malha fechada satisfaz a especificação de estabilidade, e por esta razão, a sub-rotina que calcula estes

funcionais é programada de modo a não efetuar os cálculos no caso da inequação de estabilidade (2.23) não ser satisfeita.

Pode acontecer que o ponto de partida p não satisfaça a condição de estabilidade, a qual é expressa pela inequação (2.23) ($\sigma_s < 0$), em geral aplicado para definir a estabilidade do sistema. Por esta razão, assim que o programa aceita o valor de p a inequação de estabilidade é testada, e quando ela não é satisfeita o programa chama a sub-rotina denominada de Processo dos Contornos Móveis para resolver o conjunto das $[p]$ inequações (3.9) e mais a inequação de estabilidade, totalizando $[p] + 1$ inequações. No contexto do Método das Inequações, a condição de estabilidade é expressa por:

$$-10^{35} \leq \sigma_s \leq a \quad (3.14)$$

onde a é um valor fixado internamente pelo programa em $a = -0,0001$. O valor de p que satisfaz as $[p] + 1$ inequações assim definidas tornar-se-á o novo ponto de partida p^0 .

O programa do projeto é elaborado de modo tal que o ponto de partida p só é aceito pelo programa se suas coordenadas satisfazem as $[p]$ inequações (3.9). Isto faz com que o PCM rejeite qualquer ponto p de tentativa para o qual qualquer dessas inequações não seja satisfeita.

Por isto, o PCM é programado para testar as inequações (3.11) em duas fases: na primeira fase as $[p]$ inequações são primeiro testadas, e se todas elas forem satisfeitas só então o PCM permite o calculo dos funcionais restantes $\phi_i(p)$, $i = [p] + 1, \dots, m$, e assim estes funcionais terão as suas respectivas inequações testadas. Por esta razão, é recomendável especificar limites nos parâmetros do controlador todas as vezes que isto seja possível, já que o programa pergunta se o projetista deseja estabelecer estes limites via teclado e também por causa da economia de esforço computacional que é feita quando se evita computar as funcionais $\phi_i(p)$, $i = [p] + 1, \dots, m$, para as tentativas que não satisfaçam quaisquer das primeiras $[p]$ inequações.

Nesta implementação do PCM, o processo pára automaticamente, quando todas as inequações forem satisfeitas simultaneamente dentro de um limite (número) máximo de iterações especificadas pelo projetista no começo da execução do programa, e também quando este limite for ultrapassado sem que as inequações tenham sido resolvidas. Portanto, se é especificado um limite máximo de iterações e o PCM avança em direção a um ponto qualquer p que, aparentemente, faz parte da solução depois de algumas iterações, e se a partir deste ponto p não se obtém nenhum novo avanço na direção da solução após outras iterações, todo esforço computacional despendido até a iteração atual é perdido.

Para evitar o esforço computacional desnecessário e a perda de tempo em procura de soluções infrutíferas na implementação do programa, o usuário pode monitorar o progresso feito pelo PCM, já que o programa mostra na tela os resultados do processamento a cada iteração, e também é possível especificar iterações adicionais, se as inequações não forem resolvidas ao término de um número máximo de iterações especificadas pelo projetista no início da execução do programa. Estas duas características dos programas interativos permitem ao projetista a monitoração do PCM a um número regular de iterações (digamos, a cada 10 ou 20 iterações); para isto basta ao projetista determinar o limite de iterações de 10 em 10, e decidir se o programa continua ou não com o processamento das inequações, ou se o projetista faz alguma modificação nos parâmetros do controlador, ou se ele segue uma outra alternativa. Da experiência em projeto de Sistemas de Controle através do PCM, pode-se dizer que quando nenhum progresso é feito na procura de uma solução dentro das primeiras 10 iterações, então a execução do PCM deverá ser interrompida. Outra situação é a do vetor de projeto p satisfazer algumas das inequações (3.10) e estar tão perto quanto possível de satisfazer as inequações restantes. É razoável dizer que tal valor de p é uma solução ótima aproximada para as inequações (3.10). Porém, esta solução depende das inequações que são satisfeitas e conseqüentemente das componentes do ponto de partida do vetor de projeto p e do método utilizado para gerar os pontos de tentativa das componentes do vetor de projeto p [18].

3.2 O PAPEL DO PROJETISTA

No projeto assistido por computador de sistemas de controle pelo Método das Inequações, muito do trabalho repetitivo envolvido nesta proposta para projetar o controlador é resolvido pelo computador, mas há certas tarefas que devem ser levadas a cabo pelo projetista. É apresentada a seguir uma descrição dessas tarefas, coerente com as características do programa.

Como primeiro passo, o projetista deve identificar a Unidade básica. Isto envolve a obtenção de informação satisfatória para calcular numericamente a função de transferência da Unidade Básica $G(s)$, e os limites K_u e K_l no controle e a amplitude $(R_{0u-l}(t))$ do degrau que será aplicado na entrada do sistema, para fazer o levantamento das características da resposta a degrau do mesmo.

O projetista também tem que fixar os limites das inequações que definem as especificações de desempenho e considerar a função do sistema de controle para determinar estes limites. Os valores dos limites são determinados para assegurar a qualidade desejada de operação do sistema maior.

Ao projetista cabe decidir que restrições o controlador tem que satisfazer e expressar estas restrições na forma das inequações (3.9).

Finalmente, o projetista tem que averiguar o grau máximo de complexidade permitido pelo controlador. Na representação da função de transferência, o grau de complexidade do controlador é determinado pelo grau do polinômio do denominador de $K(s)$. O programa pode ser usado para projetar controladores com grau de complexidade qualquer.

Durante a execução do PCM, o projetista tem que monitorar o progresso na solução das inequações. Se ele percebe que nenhum progresso está sendo feito, então este não deve continuar tentando novas iterações; neste caso, as opções disponíveis, a partir desta decisão, são as seguintes :

- I) Reiniciar o PCM a partir de um outro ponto de partida p : os resultados numéricos dados pelo PCM são fortemente dependentes no ponto de partida, particularmente se p satisfaz algumas das inequações. Sempre é aconselhável

tentar dois ou três pontos de partida diferentes antes de tomar outra atitude no casos em que as inequações não possam ser resolvidas.

- II) Modificar os limites das inequações: muito freqüentemente algumas das inequações podem ser conflitantes, a ponto de não ser possível satisfazê-las simultaneamente . Nestas circunstâncias, o projetista pode ampliar os limites das inequações menos importantes, de forma que as mais importantes possam ser satisfeitas.
- III) Mudar o tipo do controlador, escolhendo outra alternativa entre as oferecidas pelo programa.
- IV) Aumentar o grau de complexidade do controlador: um controlador mais sofisticado tem maior chance de levar a um desempenho melhor.
- V) Traçar o gráfico das respostas a degrau do sistema para verificar o comportamento do sistema.
- VI) Traçar o gráfico do controle aplicado ao sistema para analisar o comportamento ou aproximação da tentativa de solução com a solução desejada.
- VII) Aceitar o controlador atual como uma solução aproximada ; se as inequações mais importantes são satisfeitas e o grau de máxima complexidade permitido ao controlador for alcançado, sem uma melhoria adicional possível, o projetista pode ser levado a aceitar este controlador como uma solução aproximada para o problema de projeto.

Por outro lado, quando uma solução é alcançada , o projetista tem a possibilidade de tentar melhorar o desempenho do sistema, reduzindo os limites das inequações de interesse para obter novos valores para os parâmetros do vetor de projeto do controlador, mas sem aumentar o seu grau de complexidade. Isto não envolverá nenhum esforço extra na operação do controlador, e pode ser considerado como uma otimização (ou refinamento) deste.

O programa interativo permite ao projetista explorar efetivamente uma ampla gama de possibilidades de projeto, de forma que o problema possa ser reformulado várias vezes, até que seja encontrada uma solução que seja a mais satisfatória para o projetista.

3.3 PROJETO ATRAVÉS DAS ESPECIFICAÇÕES DA RESPOSTA A DEGRAU

No projeto através desta proposta, funcionais $\phi_{l p l+i}(p)$, $i = 1, 2, \dots, \ell$ das respostas a degrau de malha fechada, do erro, do controle e a saída produzida são denotadas por $\rho_e(t)$, $\rho_u(t)$ e $\rho_y(t)$, respectivamente, e definidas através das equações (2.32) a (2.34).

O desempenho de sistemas dinâmicos durante transitórios provocados por mudanças de pontos de operação e distúrbios é avaliado e especificado através das *características da resposta a degrau (CRD's)*.

É freqüente que a resposta de sistemas dinâmicos seja oscilatória. Assim, o conjunto completo das CRD's é definido para respostas oscilatórias. As definições estão ilustradas na figura 3.1, ver pág. 56, que mostra a resposta a degrau típica de um sistema oscilatório estável. O gráfico apresentado é o da relação $y(t)/y(\infty)$, isto é, da resposta normalizada em relação ao valor final. Este procedimento tem a conveniência de padronizar o gráfico, deixando-o independente dos sinais algébricos do degrau e do ganho estático.

Evidentemente, o valor final da resposta normalizada é igual a 1.

As CRD's estão ilustradas na figura 3.1, e são definidas a seguir.

1. Erro de estado permanente $\rho_e(\infty)$

$$\rho_e(\infty) = \lim_{t \rightarrow \infty} \rho_e(t) = \lim_{s \rightarrow 0} W_e(s) \quad (3.15)$$

2. Tempo de subida T_1 (rise time) (t_r)

Seja

$$t_\alpha = \min \left\{ t_a \left| \frac{\rho_y(t_a)}{\rho_y(\infty)} = \alpha \right. \right\}, \quad 0 < \alpha \leq 1 \quad (3.16)$$

Esta definição significa que t_α é o tempo necessário para a resposta $y(t)$ atingir uma fração α do valor de final pela primeira vez. Neste texto, por *tempo de subida* entender-se-á o tempo necessário para que a resposta $y(t)$ atinja 90% do valor final pela primeira vez, isto é

$$t_r = t_{0,9} \quad (3.17)$$

A definição de tempo de subida utilizada nesta dissertação é aquela apresentada na equação (3.16), alertando aqui ao leitor que a definição de tempo de subida não é única. Ogata [12], por exemplo, oferece as três alternativas a seguir:

$$\begin{aligned} t_{r,1} &= t_{0,9} - t_{0,1} \\ t_{r,2} &= t_{0,95} - t_{0,05} \\ t_{r,3} &= t_1 \end{aligned}$$

Ou seja, o tempo de subida também pode ser considerado o tempo necessário para a resposta a degrau variar de 10% a 90% do valor final pela primeira vez, ou então de 5% a 95%, ou ainda de 0 a 100%.

3. Ultrapassagem máxima (“Overshoot” (M_0))

É a maior amplitude que a resposta do sistema pode alcançar, no mesmo sentido do valor final, e o próprio valor final. A ultrapassagem máxima pode ser expressa em porcentagem, de forma bastante simples, como :

$$M_0\% = \left. \begin{aligned} & \frac{\hat{\rho}_y - \rho_y(\infty)}{\rho_y(\infty)} \times 100\%, \quad \text{se } \hat{\rho}_y > \rho_y(\infty) \\ & = 0\%, \quad \text{se } \hat{\rho}_y \leq \rho_y(\infty) \end{aligned} \right\} \quad (3.18)$$

$$\text{onde } \hat{\rho}_y = \sup_{t \geq 0} \rho_y(t) \quad (3.19)$$

onde “sup” denota supremo. +++++

4. Tempo de acomodação T_2 (*settling time*) (t_s)

É o tempo necessário para a resposta entrar dentro de uma faixa de tolerância em torno do valor final, dada por uma fração $\pm \varepsilon$ do valor final, e permanecer dentro dessa faixa após esse tempo.

Matematicamente, esta definição é expressa como segue:

$$t_s = \min \left\{ t_b \left| \left| \frac{\rho(t)}{\rho(\infty)} - 1 \right| \leq \varepsilon, \quad \forall t \geq t_b \right. \right\} \quad (3.20)$$

Normalmente faz-se $\varepsilon = 0,02$ (2 %) ou $\varepsilon = 0,05$ (5 %).

5. Retrocesso (“*Undershoot*”) (L_p)

É o valor absoluto da maior amplitude negativa da resposta, isto é, em sentido oposto ao valor final (ver a figura 3.1). O retrocesso é expresso como uma fração do valor final, da seguinte forma:

$$L_p = \left| \inf_t \left[\frac{\rho(t)}{\rho(\infty)} \right] \right| \quad (3.21)$$

onde “inf” denota ínfimo. ****

+++++ Supremo de uma função é definido como sendo o limite superior mínimo [1]. Por limite superior de uma função entende-se qualquer constante que seja sempre maior ou igual à função. Quando a função tem um máximo, para um valor finito da variável independente, então o supremo é também o máximo. Entretanto, existem funções que são limitadas, mas não possuem um máximo. É o caso, por exemplo, da função $f(t) = 1 - e^{-t/T}$. À primeira vista, o máximo desta função é igual a 1, mas este valor só seria atingido para $t \rightarrow \infty$. Assim, esta função não tem máximo. Mas não existe nenhuma constante menor do que 1 que seja um limite superior para $f(t)$. Portanto, $f(t)$, embora não tendo máximo, tem o seu supremo igual a 1.

Sob o ponto de vista de controle, sistemas cujas respostas a degrau possuem retrocesso diferente de zero têm um comportamento estranho, pois quando a referência sofre uma variação, a resposta inicialmente varia em sentido oposto ao determinado pela referência. O retrocesso aparece em sistemas com derivadores na entrada, com ganho de sinal oposto ao do ganho estático. Isto se reflete na função de transferência através da ocorrência de zeros reais positivos, isto é, no semiplano direito do plano s.

Este fenômeno ocorre, por exemplo, em turbinas hidráulicas, quando se abre as válvulas de admissão de água com a finalidade de aumentar a potência gerada: no momento da abertura da válvula, ocorre uma queda de pressão, mas a vazão não varia de imediato: como a potência hidráulica é igual ao produto da pressão pela vazão, a abertura da válvula é seguida de uma redução de potência, que só começa a aumentar depois do restabelecimento da pressão.

6. Saída mínima do controlador $\overset{\vee}{\rho}_u$:

$$\overset{\vee}{\rho}_u = \inf_{t \geq 0} \rho_u(t) \quad (3.22)$$

7. Saída máxima do controlador $\overset{\wedge}{\rho}_u$:

$$\overset{\wedge}{\rho}_u = \sup_{t \geq 0} \rho_u(t) \quad (3.23)$$

onde “sup” denota supremo. ****

**** Ínfimo de uma função é definido como sendo o seu limite inferior máximo [1]. É o oposto de supremo. Quando uma função tem um mínimo, então o ínfimo também é o mínimo. Entretanto, existem funções que têm um limite inferior, mas não possuem um mínimo. É o caso, por exemplo, da função $e^{-t/T}$: esta função não possui mínimo, mas o seu ínfimo é igual a zero.

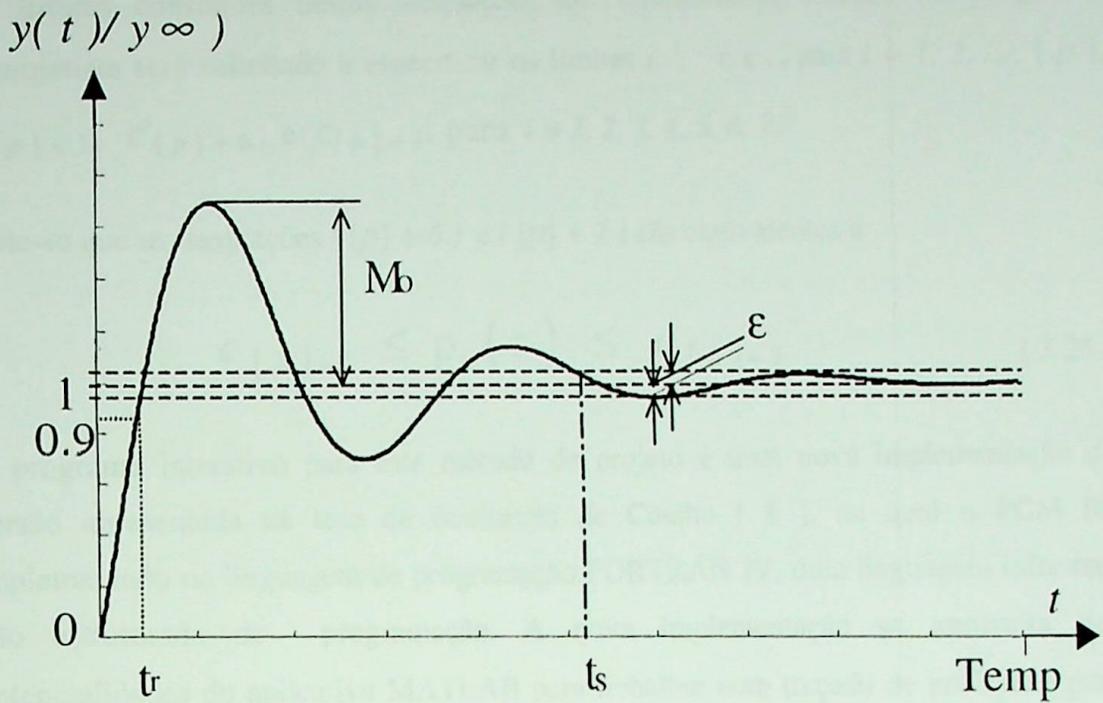


Figura 3.1 Resposta a degrau normalizada de um sistema estritamente próprio estável.

As $m = [p] + \ell + 1$ inequações (3.11) serão as seguintes

$$\left. \begin{aligned}
 c_i' \leq \phi_i(p) \leq c_i, \quad \phi_i(p) &= p_i, \quad i=1,2,\dots,[p] \\
 c_{[p]+1} \leq \phi_{[p]+1}(p) \leq c_{[p]+1}, \quad \phi_{[p]+1}(p) &= \rho_c(\infty) \\
 0 \leq \phi_{[p]+2}(p) \leq c_{[p]+2}, \quad \phi_{[p]+2}(p) &= T_1 \\
 0 \leq \phi_{[p]+3}(p) \leq c_{[p]+3}, \quad \phi_{[p]+3}(p) &= M_p \\
 0 \leq \phi_{[p]+4}(p) \leq c_{[p]+4}, \quad \phi_{[p]+4}(p) &= T_2 \\
 0 \leq \phi_{[p]+5}(p) \leq c_{[p]+5}, \quad \phi_{[p]+5}(p) &= L_p \\
 c_{[p]+6} \leq \phi_{[p]+6}(p) \leq 10^{36}, \quad \phi_{[p]+6}(p) &= \overset{\vee}{\rho}_u \\
 -10^{36} \leq \phi_{[p]+7}(p) \leq 10^{36}, \quad \phi_{[p]+7}(p) &= \hat{\rho}_u \\
 -10^{35} \leq \phi_{[p]+8}(p) \leq 10^{36}, \quad \phi_{[p]+8}(p) &= \sigma_s
 \end{aligned} \right\} \quad (3.24)$$

Os limites constantes nestas inequações são internamente fixados no programa. O projetista será solicitado a especificar os limites c'_i e c_i , para $i = 1, 2, \dots, [p]$, $c'_{[p]+1}$, $c'_{[p]+6}$, e $c_{[p]+i}$, para $i = 1, 2, 3, 4, 5, 6, 7$.

Note-se que as inequações $([p] + 6)$ e $([p] + 7)$ são equivalentes a

$$c_{[p]+6} \leq \rho_u(t) \leq c_{[p]+7} \quad (3.25)$$

O programa interativo para este método de projeto é uma nova implementação da versão apresentada na tese de doutorado de Coelho [8], na qual o PCM foi implementado na linguagem de programação FORTRAN IV, uma linguagem informal, não estruturada de programação. A nova implementação se aproveita das potencialidades do aplicativo MATLAB para trabalhar com traçado de gráficos e para acessar as bibliotecas de controle do MATLAB. O MATLAB tem uma interface usuário aplicativo amigável para trabalhar, o programa escrito no ambiente MATLAB dá ao usuário uma gama de opções, as quais foram descritas no item 3.2 deste capítulo. Todas as sub-rotinas para cálculo das respostas no tempo e as sub-rotinas de programação utilizadas nesta proposta de controle estão descritas no apêndice A desta dissertação.

CAPÍTULO 4

RESOLVENDO AS INEQUAÇÕES

4.1. O PROCESSO DOS CONTORNOS MÓVEIS (PCM)

A tarefa principal do computador durante o projeto de controle através do Método das Inequações é resolver numericamente um sistema de inequações do tipo:

$$\phi_i(p) \leq c_i, \quad i=1, 2, \dots, m \quad (4.1)$$

onde os c_i são números reais, e onde P denota um vetor de componentes reais como segue :

$$P = [p_1 \quad p_2 \quad \dots \quad p_{[p]}]^t \quad (4.2)$$

e onde os $\phi_{i(p)}$ são funções reais de P . Cada inequação de $\phi_{i(p)} \leq c_i$ define um conjunto S_i de pontos no espaço $[p]$ dimensional $R^{[p]}$. As coordenadas desse vetor são $p_1, p_2, \dots, p_{[p]}$. Cada conjunto pode ser escrito como:

$$S_i = \{ p : \phi_{i(p)} \leq c_i \} \quad (4.3)$$

Os limites de S_i são definidos através da equação $\phi_{i(p)} = c_i$.

Se existir um ponto em $R^{[p]}$ que satisfaça simultaneamente todas as inequações

$\phi_{i(p)} \leq c_i, i = 1, 2, \dots, m$ então P pertence ao conjunto S_i

Definindo-se S como o conjunto interseção de todos os possíveis conjuntos S_i pode-se escrever a seguinte definição:

$$S = \bigcap_{i=1}^m S_i \quad (4.4)$$

Evidentemente p satisfará todas as inequações se, e somente se, p estiver contido em S . Logo diz-se que S é um conjunto admissível, e qualquer p em S é chamado de ponto admissível.

Para resolver o sistema de inequações (4.1) Zakian e Al-Naib [18] desenvolveram um processo de busca iterativa, o qual parte de um ponto inicial arbitrário p^0 e faz a busca de um ponto admissível, ou seja, um ponto p que pertença ao conjunto S de possíveis soluções do sistema de inequações.

Uma descrição do Algoritmo que é denominado de Processo dos Contornos Móveis (PCM) é apresentada a seguir.

Considere-se o ponto p^k como sendo o valor do ponto p quando este ponto se encontra em sua k -ésima tentativa de movimento.

Então define-se que:

$$\left. \begin{aligned} c_i^k &= C_i, \text{ se } \phi_i(p^k) \leq c_i \\ &= \phi_i(p^k), \text{ se } \phi_i(p^k) > c_i \\ i &= 1, 2, \dots, m \end{aligned} \right\} \quad (4.5)$$

$$S_i^k = \{P: \phi_i(p) \leq c_i^k\} \quad (4.6)$$

$$S^k = \bigcap_{i=1}^m S_i^k \quad (4.7)$$

Uma mudança é gerada partindo do ponto p^k em direção ao ponto tentativa \tilde{p}^k se, e somente se :

$$\phi_i(\tilde{p}^k) \leq c_i^k, \text{ para } i = 1, 2, \dots, m \quad (4.8)$$

se esta inequação apresentada acima for satisfeita então o ponto tentativa \tilde{p}^k é chamado de ponto sucesso e então o ponto p^{k+1} recebe o valor do ponto tentativa

$\tilde{p}^k (p^{k+1} = \tilde{p}^k)$ tornando-se o ponto p^{k+1} o novo ponto tentativa. Se alguma das inequações (4.8) não for satisfeita, outra tentativa é feita a partir do ponto p^k até que um sucesso aconteça.

Quando todas as inequações (4.8) forem satisfeitas, os limites de S^{k+1} estarão a um passo dos limites de S . O processo é terminado quando, depois de um número suficiente de passos bem sucedidos, os limites de S_i^k coincidam com os limites de S_i para todo $i = 1, 2, \dots, m$, isto é :

$$C_i^k = C_i \quad \text{para } i = 1, 2, \dots, m \quad (4.9)$$

significando que os limites de S^k convergiram para os limites de S .

É conceitualmente muito útil imaginar o movimento dos limites $\phi_{i(p)} = C_i^k$, e por esta razão o processo de resolução das inequações é chamado de Processo dos Contornos Móveis.

Um ponto tentativa \tilde{p}^k é gerado através da utilização de algoritmo desenvolvido por Rosenbrock [13], onde as variações de p são geradas sobre uma base de vetores ortonormais, como segue:

$$\tilde{p}^k = P^k + e_j V_j^r \quad (4.10)$$

onde os $V_j^r, j = 1, 2, \dots, [p]$ são os vetores ortonormais, e os e_j são números reais. Se \tilde{p}^k não for um sucesso, o valor de e_j é substituído por βe_j , onde β denota um número escalar pertencente ao intervalo $-1 < \beta < 0$. Se o ponto \tilde{p}^k é um sucesso, o valor de e_j é substituído por αe_j , onde $\alpha > 1$. Em ambos os casos o valor de j é substituído por $j+1$ na equação (4.10); uma iteração do PCM é definida como $[P]$

tentativas consecutivas [18], portanto, depois da última tentativa ($j = [P]$) da n -ésima iteração, faz-se $j = 1$ e $N = N+1$ para a próxima tentativa.

Tão logo um sucesso é seguido de um fracasso para todo $j = 1, 2, \dots, [P]$, o valor de V_j^r é substituído V_j^{r+1} , $j = 1, 2, \dots, [P]$. Os vetores V_j^{r+1} são calculados da seguinte forma:

Considere-se a d_j como sendo igual a soma de todos os valores de e_j que levaram a um sucesso durante a r -ésima mudança (ou passo); seja :

$$\left. \begin{aligned} a_1 &= d_1 V_1^r + d_2 V_2^r + \dots + d_{[P]} V_{[P]}^r \\ a_2 &= d_2 V_2^r + \dots + d_{[P]} V_{[P]}^r \\ &\vdots \\ a_{[P]} &= d_{[P]} V_{[P]}^r \end{aligned} \right\} \quad (4.11)$$

Ortogonalizando os vetores a_j através do procedimento de Gram-Schmidt, tem-se

$$\left. \begin{aligned} b_1 &= a_1 \\ V_1^{r+1} &= b_1 / \| b_1 \| \\ b_2 &= a_2 - [a_2, V_1^{r+1}] V_1^{r+1} \\ V_2^{r+1} &= b_2 / \| b_2 \| \\ &\vdots \\ b_{[P]} &= a_{[P]} - \sum [a_{[P]}, V_k^{r+1}] V_k^{r+1} \\ V_{[P]}^{r+1} &= b_{[P]} / \| b_{[P]} \| \end{aligned} \right\} \quad (4.12)$$

Onde (x, y) denomina-se o produto escalar de dois vetores $[p]$ -dimensionais x e y

$$(x, y) = x^t y = x_1 y_1 + x_2 y_2 + \dots + x_{[p]} y_{[p]} \quad (4.13)$$

e $\| x \|$ denota o comprimento do vetor x definido através de

$$||x|| = (x, x)^{1/2} \quad (4.14)$$

Inicialmente, para $r = 0$, e_j e V_j^r são escolhidos arbitrariamente. Entretanto quando o valor de r é incrementado, a velocidade de convergência de S^k em direção a S tende a aumentar, visto que V_1^r torna-se progressivamente orientado ao longo da direção de um avanço mais rápido, V_2^r ao longo da melhor direção normal à direção de V_1^r e assim sucessivamente. O PCM foi utilizado com sucesso com os valores de $\alpha = 3$ e $\beta = -0.5$

Nesta dissertação, o Processo dos Contornos Móveis foi utilizado com os valores de α e β acima mencionados.

Os vetores iniciais V_j^0 são inicialmente posicionados como uma base ortonormal do vetor de espaço p , como segue:

$$V_j^0 = \left[\begin{array}{ccccccc} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{array} \right]^t \left. \begin{array}{l} \uparrow \\ j\text{-ésima componente} \end{array} \right\} \quad (4.15)$$

Os valores iniciais de e_j são adotados como segue:

$$\left. \begin{array}{l} e_j^0 = 0.1 |p_j|, \text{ para } |p_j^0| \geq 0.1 \\ = 0.01 \quad \text{para } |p_j^0| < 0.1 \\ j = 1, 2, \dots, [p] \end{array} \right\} \quad (4.16)$$

onde P_j^0 é a j -ésima componente do vetor inicial de projeto p^0

Quando o valor de V_j^r é substituído pelo valor de V_j^{r+1} , $j = 1, 2, \dots, [p]$ então

e_j recebe o valor de e_j^0 ($e_j = e_j^0$), e então inicia-se uma nova iteração incrementando-se o valor de N , fazendo-se $N = N + 1$ e $j = 1$.

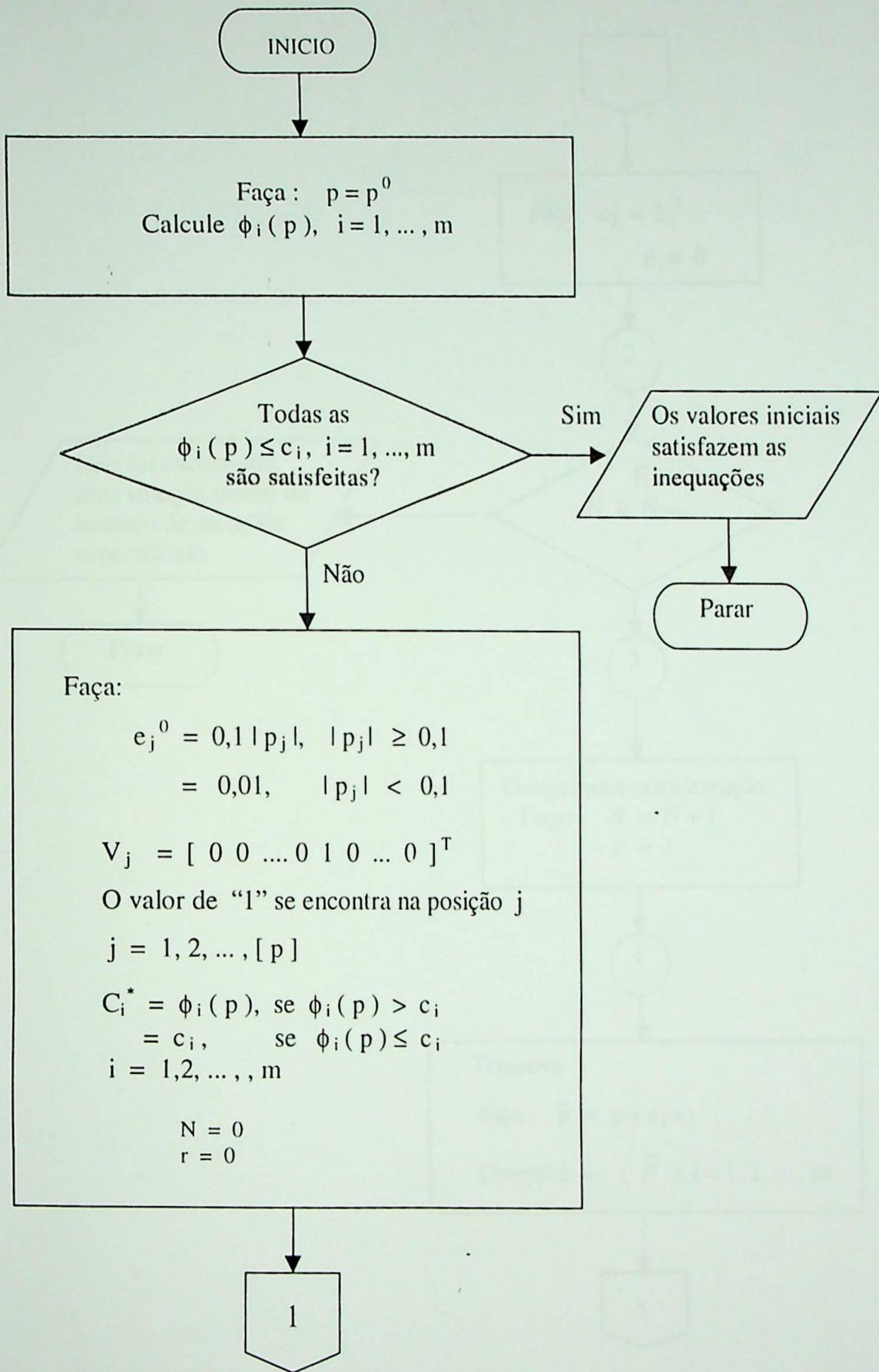


Figura 4.1 Fluxograma do Processo dos Contornos Móveis (PCM)

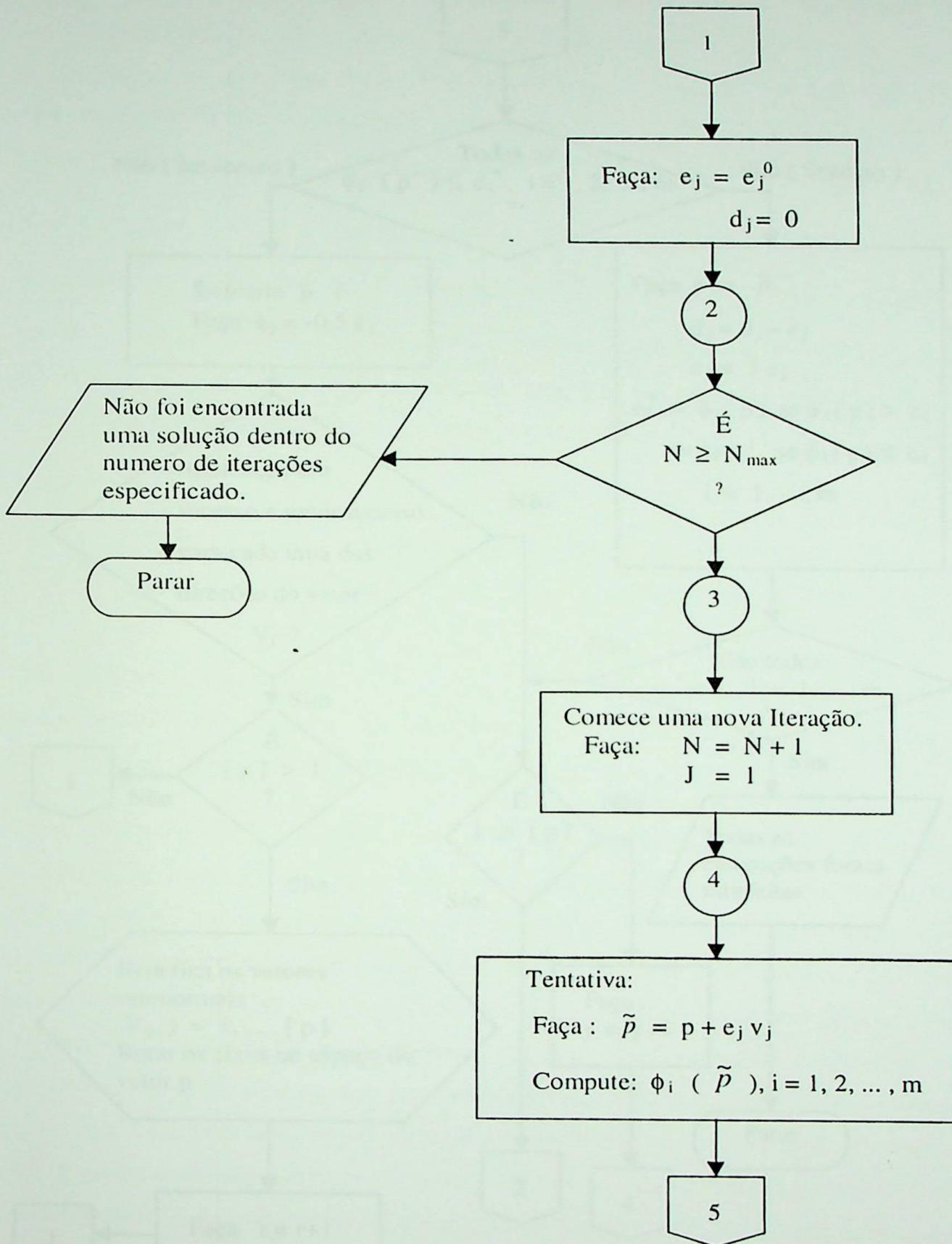


Figura 4.2 Fluxograma do Processo dos Contornos Móveis (continuação).

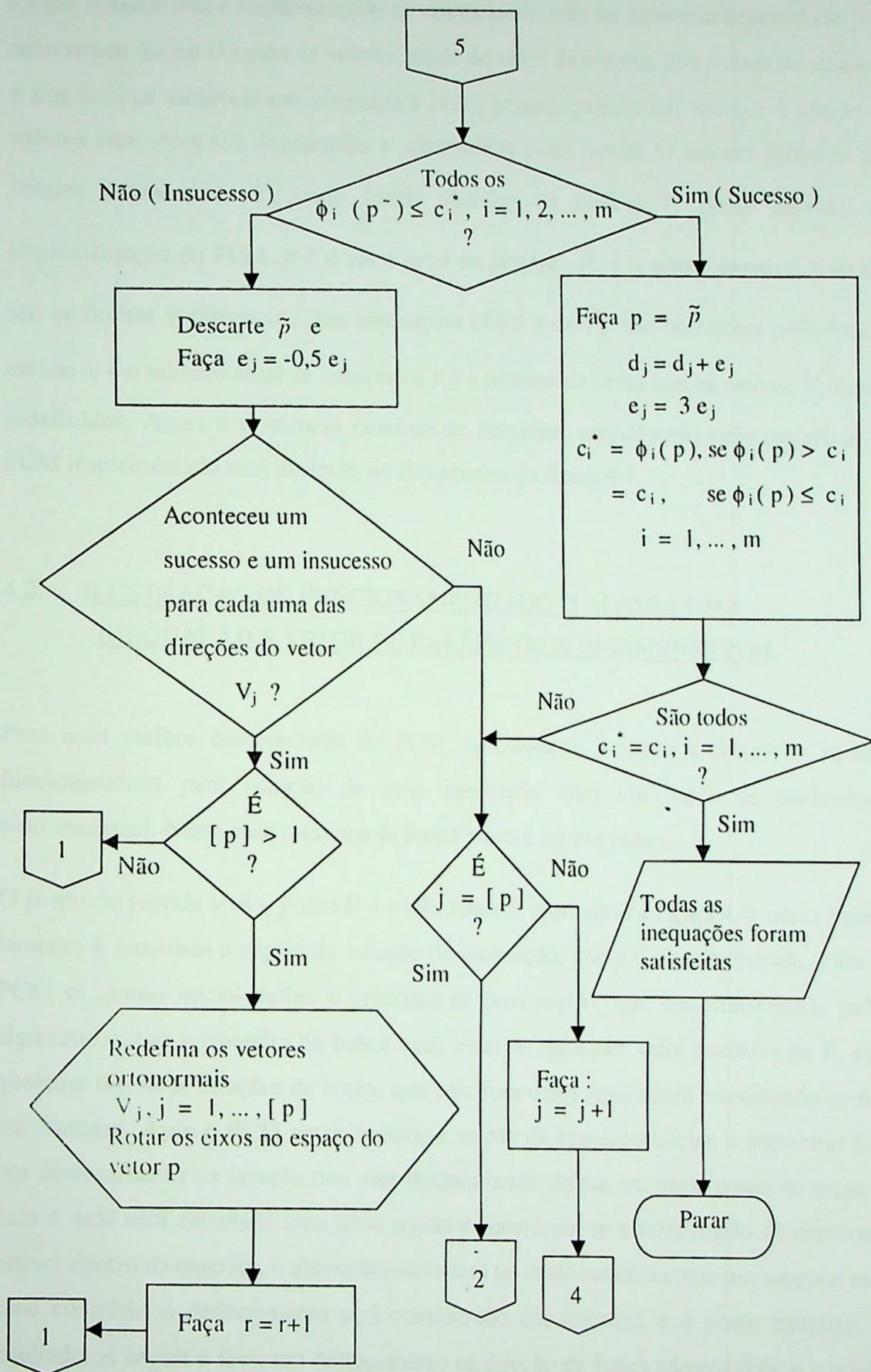


Figura 4.3 Fluxograma do Processo dos Contornos Móveis (continuação).

Como o algoritmo é implementado no computador, não há nenhuma necessidade para usar-se um índice (k) para os valores atuais do vetor de projeto, dos pontos de tentativa e dos limites variáveis nas inequações (4.8), porque quando um sucesso é obtido os valores anteriores são descartados e substituídos pelos novos. O mesmo aplica-se aos vetores ortonormais V_j onde foram abolidos os índices r . Neste sentido, na implementação do PCM, p é o vetor atual de projeto, \tilde{p} é o ponto tentativa, C_i^* são os limites variáveis (C_i^k nas inequações (4.8)) e os V_j são os vetores ortonormais atuais; N é o número atual de iterações e r é o número de vezes que os vetores V_j foram redefinidos. N_{max} é o número máximo de iterações, especificado pelo projetista. O PCM implementado está ilustrado no fluxograma da figura 4.1.

4.2. ILUSTRAÇÃO DO FUNCIONAMENTO DO PCM PARA UMA INEQUAÇÃO E VETOR DE PARÂMETROS BI-DIMENSIONAL

Para uma melhor compreensão do PCM, apresenta-se a seguir a descrição de seu funcionamento para solução de uma inequação com um vetor de parâmetros bidimensional. Neste caso o espaço de busca passa a ser um plano.

O ponto de partida será o ponto $P = (0,0)$, como é mostrado na figura 4.4; nesta figura também é mostrada a região da solução da inequação, que é a área hachurada. Para o PCM o ponto inicial define o contorno de uma região, que será considerado pelo algoritmo como a fronteira de busca mais externa: qualquer valor tentativa de P , em qualquer das duas direções de busca, que caia fora desta região será considerado como um fracasso. Após o PCM ter delimitado a região de contorno inicial, o algoritmo faz um deslocamento na direção um; este deslocamento define um novo ponto no espaço para o qual será calculada uma nova região de contorno; se a nova região de contorno estiver dentro da anterior, o algoritmo entenderá tal deslocamento como um sucesso; em caso contrário o deslocamento será considerado um fracasso, e o ponto tentativa é rejeitado. A seguir é feito um deslocamento na direção de busca número dois, e o novo ponto no espaço define uma nova região de contorno, e verifica-se se esta nova região está dentro da anterior, como foi feito na outra direção; na seqüência, um novo

deslocamento é feito na direção de busca número um com a diferença que desta vez o deslocamento será 3(três) vezes maior que o deslocamento da primeira vez (figura 4.5) já que na primeira vez o deslocamento nesta direção foi um sucesso; o processo de cálculo é repetido para este novo ponto; quando um fracasso acontece, o próximo deslocamento nesta direção será no sentido oposto, e a amplitude do passo será a metade da anterior; na seqüência é feito o deslocamento na segunda direção de busca (figura 4.5). As buscas continuarão sendo feitas nas duas direções até que o algoritmo tenha obtido pelo menos um sucesso e um fracasso em cada direção, e não se possa obter novos pontos de sucesso; quando se chegar a esta situação o algoritmo girará os eixos do plano, assim definindo novas direções de busca (figura 4.6), que serão sempre ortogonais entre si. Este processo de geração de deslocamentos continuará até que seja encontrado um ponto que esteja situado dentro da região de solução da inequação(figura 4.7).

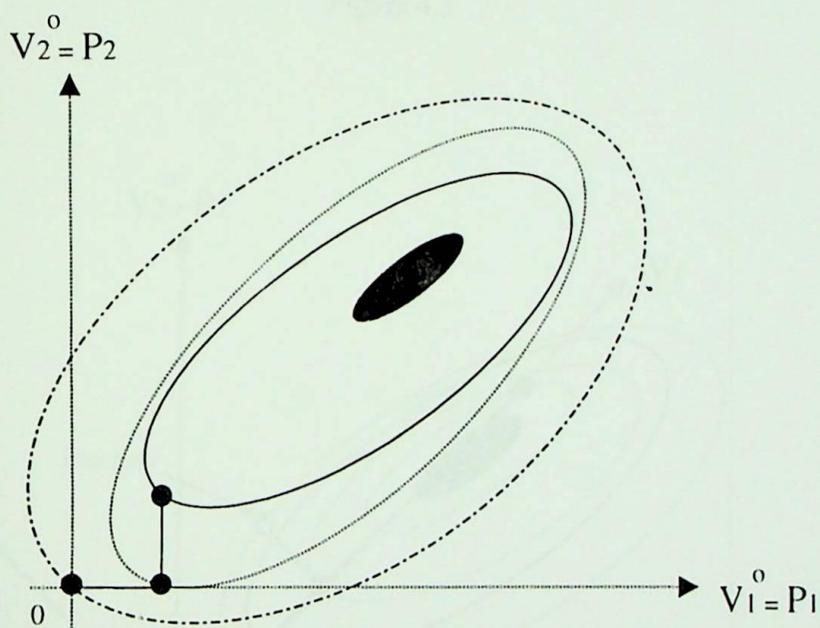


Figura 4.4

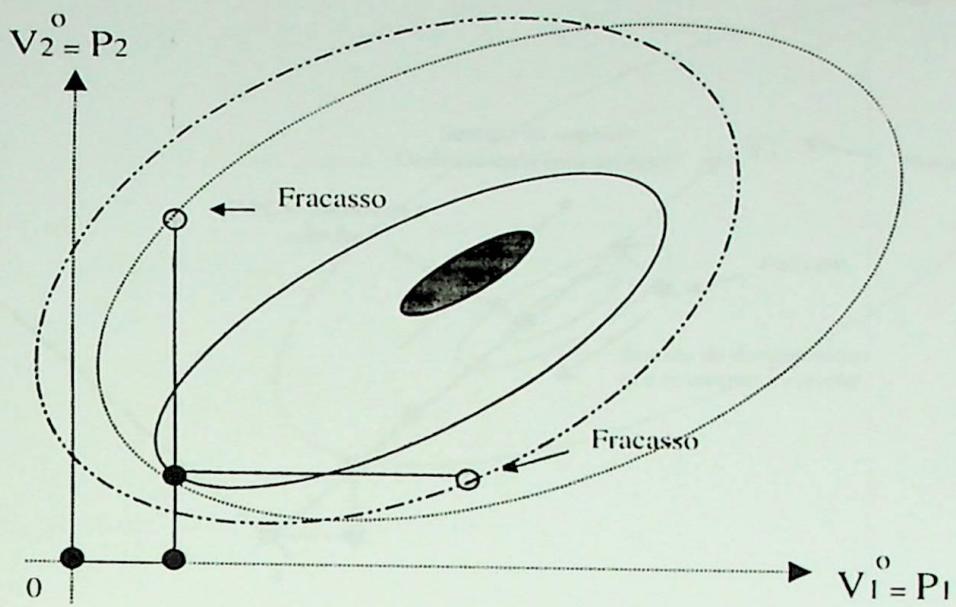


Figura 4.5

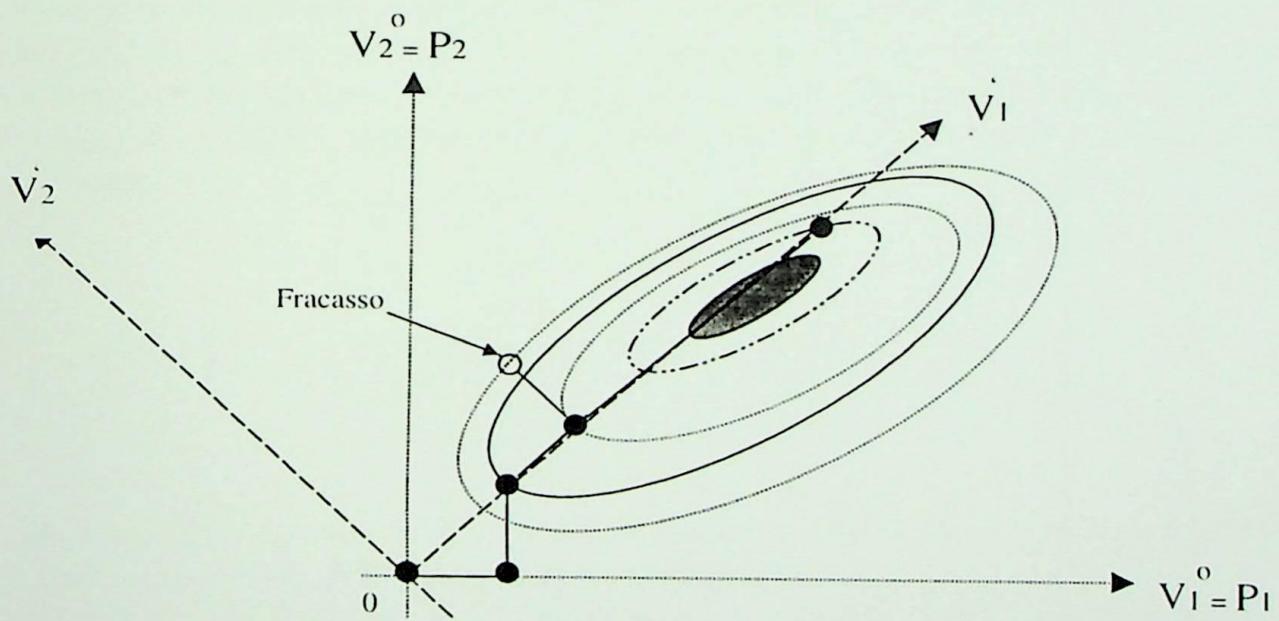


Figura 4.6

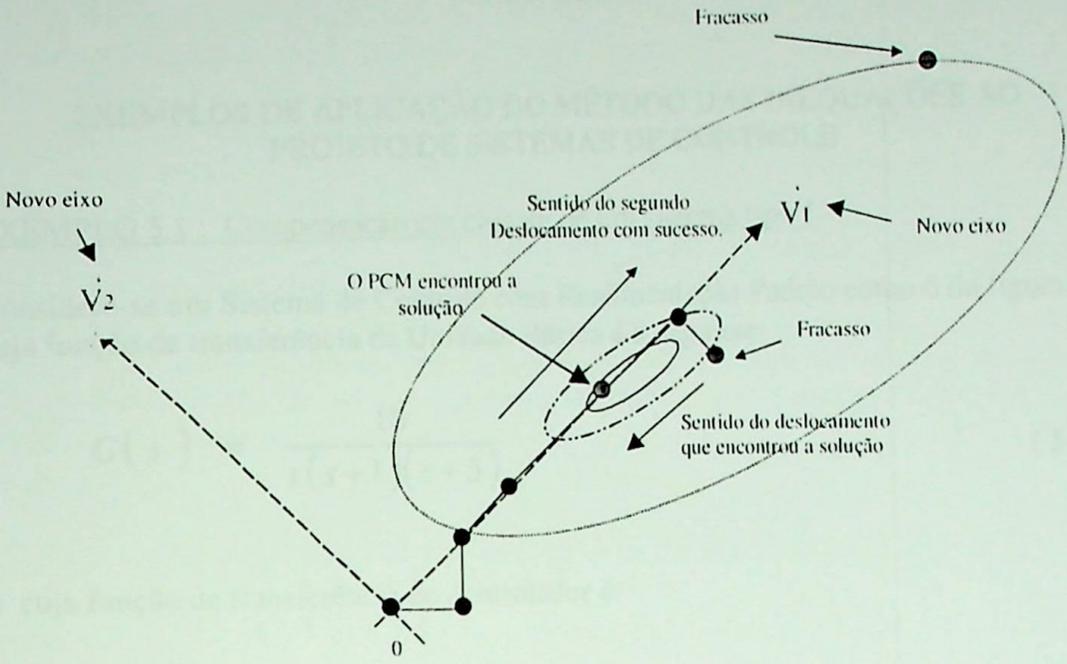


Figura 4.7

Os resultados obtidos por investigação por partes sucessivas com aplicação de novas soluções na procura do valor do vector \vec{v} para a obtenção de uma boa resposta a regime de estado estacionado. Usando a metodologia de busca descrita, D'Azaro e Hwang (1981) obtiveram o valor $\mu = (1,475; 1,0 - 1)$. Tendo em conta, entretanto, o valor $\mu = (1,475; 1,0 - 1)$ a qual representa a solução.

$$f(\mu, \sigma) = \mu^2 + \sigma^2$$

Na introdução do método dos tempos (de Talbot e At-Nash (1971) usando o PCM para localizar um ponto μ que satisfizesse as condições:

$$\begin{aligned} 0,00 &\leq P_1 \leq 100 \\ 0 &\leq P_2 \leq 20 \\ 0,01 &\leq P_3/P_2 \leq 10 \\ T_1 &\leq 10 \\ M_1 &\leq 0,01 \end{aligned}$$

CAPITULO 5

EXEMPLOS DE APLICAÇÃO DO MÉTODO DAS INEQUAÇÕES AO PROJETO DE SISTEMAS DE CONTROLE

EXEMPLO 5.1 : Compensação em cascata de um sistema tipo 1

Considere-se um Sistema de Controle com Realimentação Padrão como o da figura 2.1, cuja função de transferência da Unidade Básica é a seguinte:

$$G(s) = \frac{10}{s(s+1)(s+5)} \quad (5.1)$$

e cuja função de transferência do controlador é:

$$K(s) = \frac{p_1(1+p_2 \cdot s)}{1+p_3 \cdot s} \quad (5.2)$$

O mesmo problema foi investigado por vários projetistas que apontaram outras soluções na procura do valor do vetor p para a obtenção de uma boa resposta a degrau de malha fechada. Usando o método do lugar das raízes, D'Azzo e Houpis [10] obtiveram o valor $p = (1.475, 1, 0.1)$. Bach [20] obteve o valor $p = (1.7, 1.2, 0.144)$ o qual minimiza a integral

$$\int_0^{\infty} t [\rho_e(t)]^2 dt \quad (5.3)$$

Na introdução do método das inequações, Zakian e Al-Naib [17] usaram o PCM para localizar um ponto p que satisfizesse as inequações

$$\left. \begin{array}{l} 0.01 \leq P_1 \leq 100 \\ 0 \leq P_2 \leq 20 \\ 0.01 \leq \frac{P_3}{P_2} \leq 10 \\ T_1 \leq 1.0 \\ M_p \leq 0.01 \end{array} \right\} \quad (5.4)$$

O ponto de partida $p_o = (1, 1, 1)$ foi determinado de modo a pertencer à região de estabilidade do sistema. Um ponto $p = (0.997, 1.119, 0.0125)$ satisfazendo estas inequações foi achado depois de 9 iterações através da utilização do PCM. O tempo de execução do programa foi de 14 segundos em um computador DEC PDP-10. A atual versão do programa levou 17 segundos em um PC compatível Pentium II de 400 MHz de frequência para localizar o ponto $p = (0.9047, 1.147, 0.01963)$, satisfazendo as inequações (3.26) em 12 iterações. Os limites utilizados nas inequações que representam os critérios de desempenho são aqueles mostrados na tabela 5.1; na mesma tabela encontram-se os valores encontrados através da utilização dos outros métodos de projeto mencionados anteriormente, todos esses dados resumem também os resultados do projeto. Na figura 5.4 são mostrados os gráficos das respostas a degrau de todos os métodos comparados de projeto.

Tabela 5.1 Resposta a degrau para as funções de transferência

$$G(s) = 10 / [s (s + 1) (s + 5)], \quad K(s) = P_1 (1 + P_2 s) / (1 + P_3 s)$$

	LIMITES		Ponto de partida	D'Azzo e Houpis	Bach	PCM anterior	PCM novo
	Inferior	Superior					
P_1	0.01	100	1	1.475	1.7	0.997	0.9047
P_2	0	20	1	1	1.2	1.119	1.147
P_3	0	200	1	0.1	0.144	0.01399	0.01963
$\rho_c(\infty)$	0	0	0	0	0	0	0
T_1	0	1	1.428	0.8996	0.5799	0.8665	0.9406
M_p	0	0.01	0.4858	0.0849	0.3263	0.090	0
T_2	0	3	5	2.105	3.642	1.084	1.231
L_p	0	0	0	0	0	0	0
ρ_u	-80	10^{36}	-0.4858	-0.866	-3.59	-1.040	-0.8828
$\hat{\rho}_u$	-10^{36}	80	1	10	14.17	79.75	52.85
$\sigma(\lambda)$	-10^{35}	-0.0001	-0.2911	-1	-0.8048	-0.8456	-0.08109
No. Itera.	-	-	0	-	-	9	12

A seguir é mostrada uma listagem dos resultados obtidos para o projeto do Controlador através do programa elaborado para a resolução de inequações baseado no Processo dos Contornos Móveis; os comandos foram executados com a versão 5.3.1 do programa MATLAB.

» princípio

PROJETO E AJUSTE COMPUTACIONAL DE CONTROLADORES DE SISTEMAS DINÂMICOS PELO MÉTODO DAS INEQUAÇÕES

Digite os dados da relação $A(S)/B(S)$ dos dois polinômios de $G(S)$

São permitidos somente polinômios racionais em " s "

Entre com o grau do polinômio do numerador: **0**

Entre com o grau do polinômio do denominador: **3**

Coefficientes do Numerador

Digite os coeficientes em ordem decrescente .

Entre com os coeficientes de S^0 : **10**

Coefficientes do Denominador

Digite os coeficientes em ordem decrescente

Entre com os coeficientes de S^3 : **1**

Entre com os coeficientes de S^2 : **6**

Entre com os coeficientes de S^1 : **5**

Entre com os coeficientes de S^0 : **0**

Os tipos de estrutura dos controladores disponíveis são:

$$1.-) K(S,P) = \frac{P(1) (1+P(2)*S).....(1+P(N+1)*S)}{S**L (1+P(N+2)*S).....(1+P(N+M+1)*S)}$$

e

$$2.-) K(S,P) = \frac{P(1)+P(2)*S+.....+P(N+1)*S**N}{1+P(N+2)*S+.....+P(N+M+1)*S**M}$$

Selecione o tipo de estrutura do controlador (tipo 1 ou 2): **1**

Entre com o valor de **L**: **0**

Entre com o valor de **M**: **1**

Entre com o valor de **N**: **1**

se existem limites digite " 1 " senão digite " 0 ": **1**

Limites nos parâmetros

Limite inferior de **P^1**: **0.01**

Limite superior de **P^1**: **100**

Limite inferior de **P^2**: **0**

Limite superior de **P^2**: **20**

Limite inferior de **P^3**: **0**

Limite superior de **P^3**: **200**

Entre com os valores do Ponto de partida

O ponto **P^1**: **1**

O ponto **P^2**: **1**

O ponto **P³: 1**

Digite a amplitude do degrau a ser aplicado ao sistema

Digite valores entre 0.0 e 1.0 (+ > 0)

São permitidos somente valores positivos

Entre com a amplitude do degrau a ser aplicado ao sistema: 1

Na continuação, dê entrada aos limites das inequações

LIMITES NO DESEMPENHO DAS INEQUAÇÕES

Dê entrada aos limites do **ERRO DE ESTADO PERMANENTE**

Limite inferior : **0**

Limite superior : **0**

Dê entrada ao limite do **TEMPO DE SUBIDA**

Digite o limite superior como sendo positivo (**>=0**)

Limite superior : **1.0**

Dê entrada ao limite do **OVERSHOOT**

Digite o limite superior como sendo positivo (**>=0**)

Limite superior : **0.01**

Dê entrada ao limite do **TEMPO DE ACOMODAÇÃO**

Digite o limite superior como sendo positivo (**>=0**)

Limite superior : **3.0**

Dê entrada ao limite do **UNDERSHOOT**

Digite o limite superior como sendo positivo (**>=0**)

Limite superior : **0**

Dê entrada aos limites da SAIDA DO CONTROLADOR

Limite inferior : -80.0

Dê entrada aos limites da SAIDA DO CONTROLADOR

Limite superior : 80.0

O TEMPO MÍNIMO SUGERIDO DE RESPOSTA É DE =

1.717629e+001

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDCMOSTRAR A RESPOSTA DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 3

PROJETO DE CONTROLE

ENTRE COM O NÚMERO DE ITERAÇÕES = 15

Funcional P1:=1
Funcional P2:=1
Funcional P3:=1
Funcional P4:=0
Funcional P5:=1.427915e+000
Funcional P6:=4.858037e-001
Funcional P7:=1.270936e+001
Funcional P8:=0
Funcional P9:=-4.858037e-001
Funcional P10:=1
Funcional P11:=-2.910990e-001

Detalhes da iteração Número 1:

Funcional P1:=1.100000e+000
Funcional P2:=1
Funcional P3:=1
Funcional P4:=0
Funcional P5:=1.354119e+000
Funcional P6:=5.212184e-001
Funcional P7:=1.413713e+001
Funcional P8:=0
Funcional P9:=-5.733403e-001
Funcional P10:=1.100000e+000
Funcional P11:=-2.734972e-001

Detalhes da iteração Número 2:

Funcional P1:=9.500000e-001
Funcional P2:=1.100000e+000
Funcional P3:=1

Fracasso

Fracasso

Funcional P4:=0
Funcional P5:=1.421754e+000
Funcional P6:=4.502125e-001
Funcional P7:=1.211833e+001
Funcional P8:=0
Funcional P9:=-4.650701e-001
Funcional P10:=1.045000e+000
Funcional P11:=-3.263874e-001

Detalhes da iteração Número 3:

Funcional P1:=1.025000e+000
Funcional P2:=1.400000e+000
Funcional P3:=9.500000e-001
Funcional P4:=0
Funcional P5:=1.217163e+000
Funcional P6:=4.459045e-001
Funcional P7:=1.031531e+001
Funcional P8:=0
Funcional P9:=-6.735705e-001
Funcional P10:=1.510526e+000
Funcional P11:=-3.681748e-001

Detalhes da iteração Número 4:

Funcional P1:=9.875000e-001
Funcional P2:=1.400000e+000
Funcional P3:=8.000000e-001
Funcional P4:=0
Funcional P5:=1.170089e+000
Funcional P6:=4.067031e-001
Funcional P7:=8.139527e+000
Funcional P8:=0
Funcional P9:=-7.089470e-001
Funcional P10:=1.728125e+000
Funcional P11:=-4.243886e-001

Detalhes da iteração Número 5:

Funcional P1:=1.006250e+000
Funcional P2:=1.400000e+000

Funcional P3:=3.500000e-001
Funcional P4:=0
Funcional P5:=8.989938e-001
Funcional P6:=2.820129e-001
Funcional P7:=4.463423e+000
Funcional P8:=0
Funcional P9:=-1.210743e+000
Funcional P10:=4.025000e+000
Funcional P11:=-6.612656e-001

SISTEMA DE MALHA FECHADA INSTÁVEL

Detalhes da iteração Número 6:

Funcional P1:=9.968750e-001
Funcional P2:=1.400000e+000
Funcional P3:=3.500000e-001
Funcional P4:=0
Funcional P5:=9.043058e-001
Funcional P6:=2.780211e-001
Funcional P7:=4.471511e+000
Funcional P8:=0
Funcional P9:=-1.188450e+000
Funcional P10:=3.987500e+000
Inequação P11:=-6.607031e-001

Detalhes da iteração Número 7:

Funcional P1:=1.001562e+000
Funcional P2:=1.400000e+000
Funcional P3:=3.500000e-001
Funcional P4:=0
Funcional P5:=9.016287e-001
Funcional P6:=2.800258e-001
Funcional P7:=4.467539e+000
Funcional P8:=0
Funcional P9:=-1.199583e+000
Funcional P10:=4.006250e+000
Funcional P11:=-6.609858e-001

Detalhes da iteração Número 8:

Funcional P1:=1.006250e+000
Funcional P2:=1.400000e+000

Funcional P3:=3.500000e-001
Funcional P4:=0
Funcional P5:=8.989938e-001
Funcional P6:=2.820129e-001
Funcional P7:=4.463423e+000
Funcional P8:=0
Funcional P9:=-1.210743e+000
Funcional P10:=4.025000e+000
Funcional P11:=-6.612656e-001

Detalhes da iteração Número 9:

Funcional P1:=1.001767e+000
Funcional P2:=1.452410e+000
Funcional P3:=2.648343e-001
Funcional P4:=0
Funcional P5:=8.273329e-001
Funcional P6:=2.364614e-001
Funcional P7:=3.775163e+000
Funcional P8:=0
Funcional P9:=-1.455368e+000
Funcional P10:=5.493911e+000
Funcional P11:=-6.302002e-001

SISTEMA DE MALHA FECHADA INSTÁVEL

Detalhes da iteração Número 10:

Funcional P1:=9.023816e-001
Funcional P2:=1.524580e+000
Funcional P3:=-4.324671e-002
Funcional P4:=0
Funcional P5:=8.795937e-001
Funcional P6:=1.514282e-001
Funcional P7:=3.203817e+000
Funcional P8:=0
Funcional P9:=-1.147290e+000
Funcional P10:=5.809335e+000
Funcional P11:=2.358828e+001

SISTEMA DE MALHA FECHADA INSTÁVEL

Detalhes da iteração Número 11:

Funcional P1:=8.782455e-001
Funcional P2:=1.043937e+000
Funcional P3:=-1.450619e-002
Funcional P4:=0
Funcional P5:=9.348550e-001
Funcional P6:=1.871596e-002
Funcional P7:=1.130855e+000
Funcional P8:=0
Funcional P9:=-8.534214e-001
Funcional P10:=1.822273e+001
Funcional P11:=6.905958e+001

Detalhes da iteração Número 12:

Funcional P1:=9.135284e-001
Funcional P2:=1.145812e+000
Funcional P3:=8.978463e-002
Funcional P4:=0
Funcional P5:=9.001383e-001
Funcional P6:=4.591376e-002
Funcional P7:=1.760697e+000
Funcional P8:=0
Funcional P9:=-9.527037e-001
Funcional P10:=1.165825e+001
Funcional P11:=-8.175064e-001

O ALGORITMO TERMINOU COM TODAS AS INEQUAÇÕES

SATISFEITAS SIMULTANEAMENTE

NÚMERO DE ITERAÇÕES REALIZADAS =12

Funcional P1:=9.047252e-001
Funcional P2:=1.147054e+000
Funcional P3:=1.963507e-002
Funcional P4:=0
Funcional P5:=9.399930e-001
Funcional P6:=0
Funcional P7:=1.229860e+000
Funcional P8:=0
Funcional P9:=-8.856939e-001
Funcional P10:=5.285283e+001
Funcional P11:=-8.108804e-001

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA A DEGRAU DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

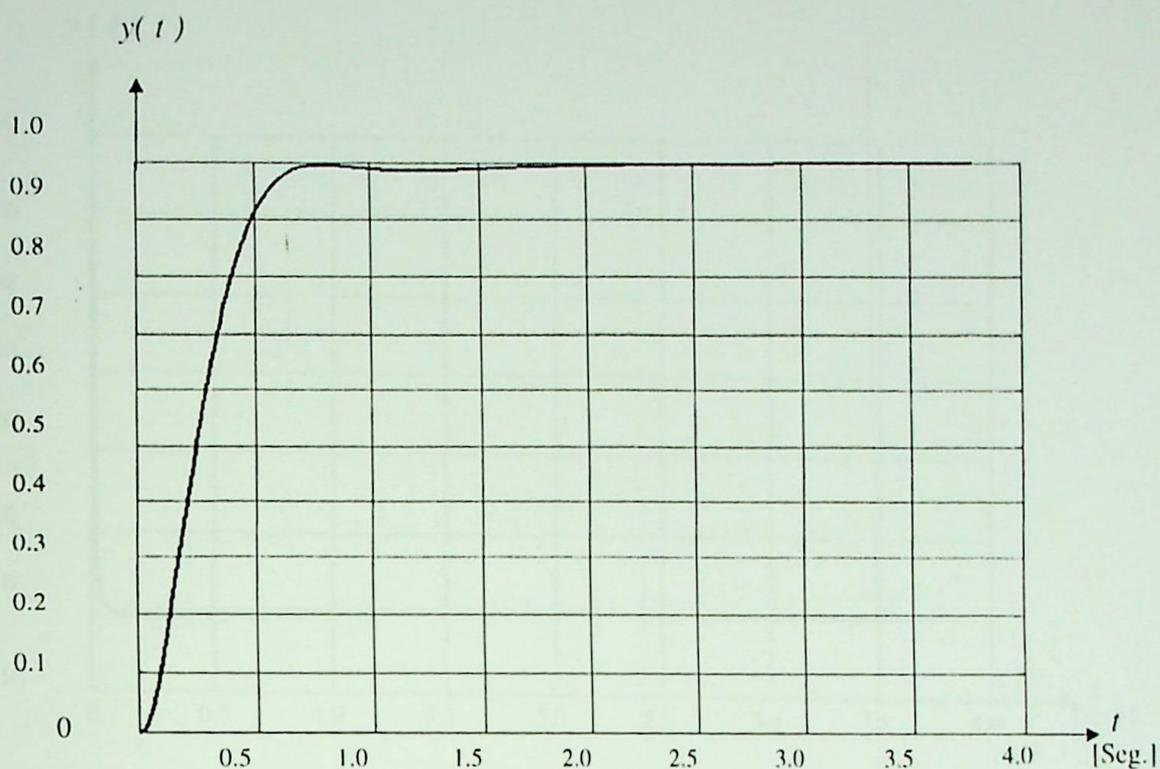


Figura 5.1 Resposta a degrau do sistema

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA A DEGRAU DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 7

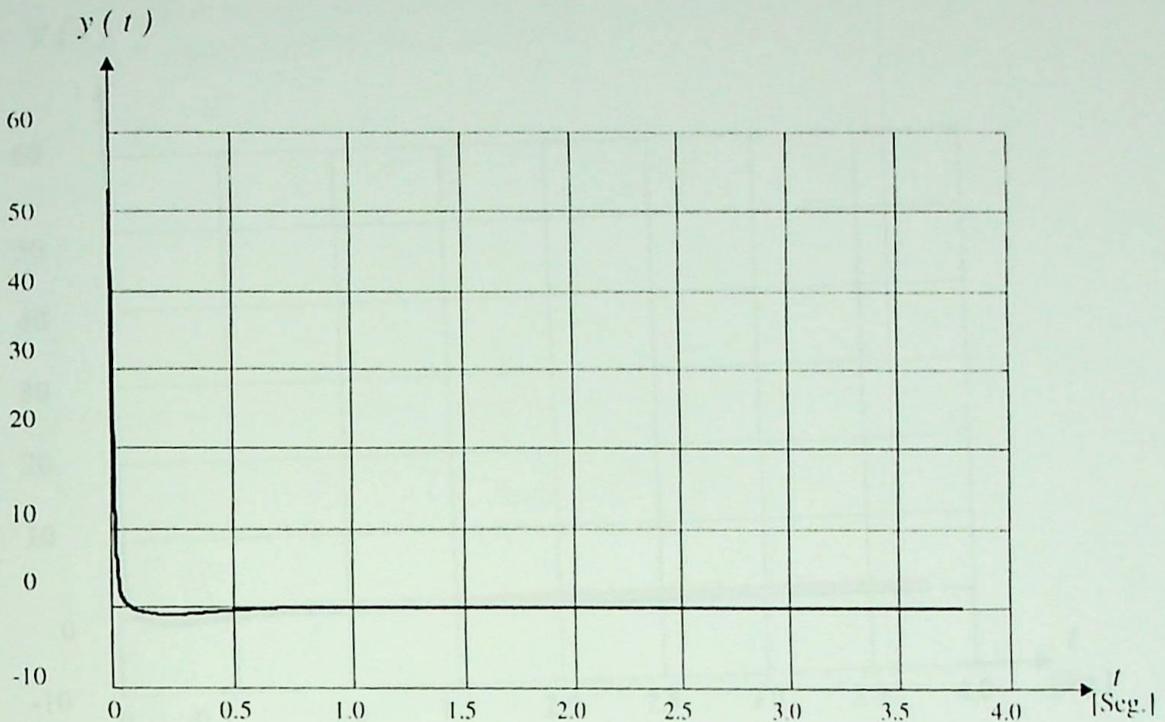


Figura 5.2 Resposta a degrau do Controlador.

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA A DEGRAU DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 8

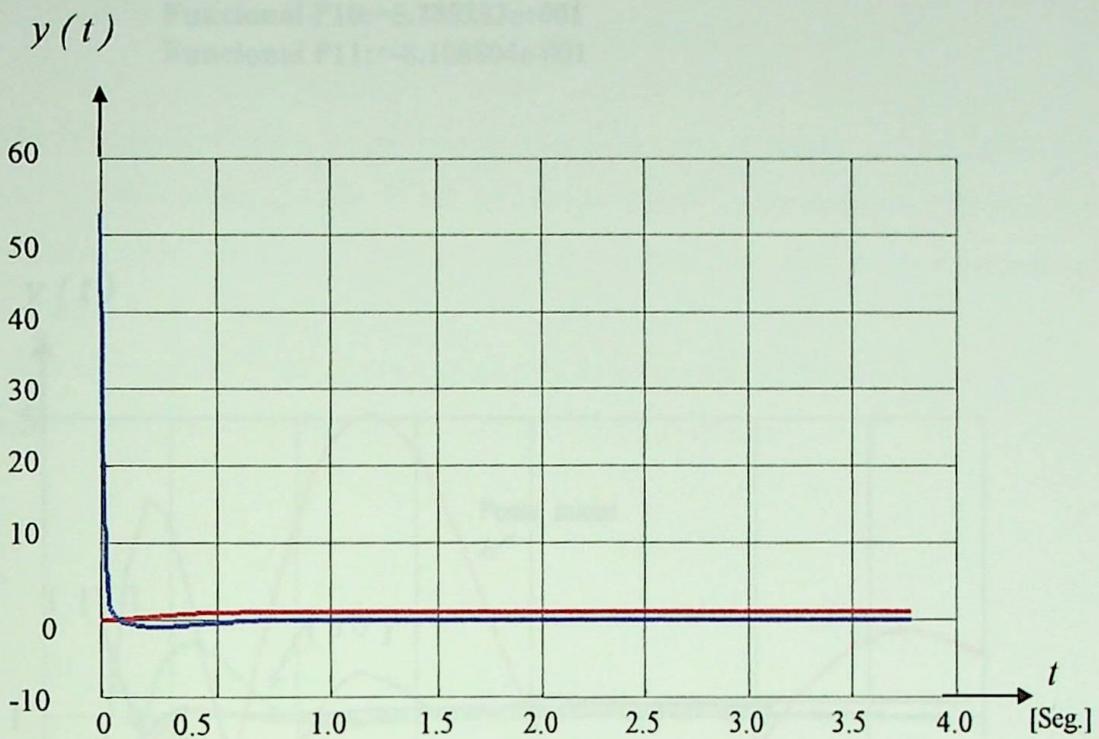


Figura 5.3 Resposta a degrau do Sistema e do Controlador.

RESPOSTA A DEGRAU PARA O SISTEMA DE MALHA FECHADA

RESULTADOS FINAIS

CRITÉRIO DE PROJETO: CARACTERÍSTICAS DA RESPOSTA A DEGRAU

**TODAS AS INEQUAÇÕES FORAM SATISFEITAS
SIMULTANEAMENTE**

O NÚMERO DE ITERAÇÕES LEVADAS A CABO FORAM:

12 ITERAÇÕES

E OS VALORES FINAIS DOS FUNCIONAIS SÃO :

Funcional P1:=9.047252e-001
Funcional P2:=1.147054e+000
Funcional P3:=1.963507e-002
Funcional P4:=0
Funcional P5:=9.399930e-001
Funcional P6:=0
Funcional P7:=1.229860e+000

Funcional P8:=0
Funcional P9:=-8.856939e-001
Funcional P10:=5.285283e+001
Funcional P11:=-8.108804e-001

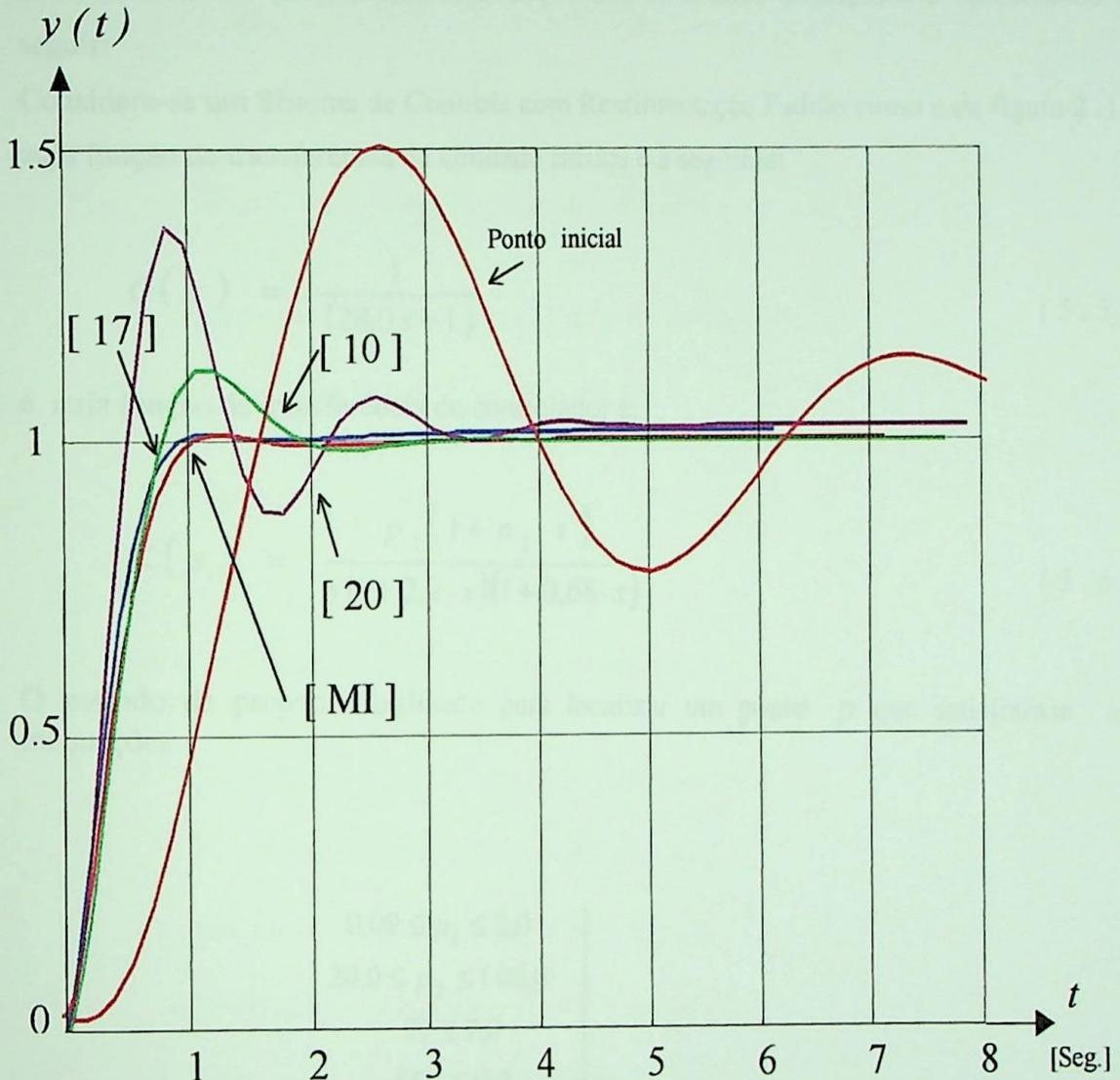


Figura 5.4 Respostas a degrau do Exemplo 5.1;

- Linha vermelha : $p = (1, 1, 1)$ (ponto inicial)
- Linha verde : $p = (1.475, 1, 0, 1)$ (D'Azzo and Houpis [10])
- Linha rouxa : $p = (1.7, 1.2, 0.144)$ (Bach [20])
- Linha azul : $p = (0.997, 1.119, 0.01399)$ (Zakian and Al-Naib [17])
- Linha vermelha : $p = (0.9047, 1.147, 0.01963)$ (Método das - Inequações).

EXEMPLO 5.2 : Compensação do regulador de velocidade de uma turbina hidráulica através do Método das Inequações

O exemplo ilustrado a seguir trata da compensação do regulador de velocidade de uma turbina hidráulica através do Método das Inequações. Este exemplo está muito bem documentado em [21], e uma reprodução dos resultados alcançados é apresentada a seguir:

Considere-se um Sistema de Controle com Realimentação Padrão como o da figura 2.1, cuja função de transferência da Unidade Básica é a seguinte:

$$G(s) = \frac{1}{(28,0s + 1)} \quad (5.5)$$

e cuja função de transferência do controlador é:

$$K(s) = \frac{p_1(1 + p_2 \cdot s)}{s(1 + 2,2 \cdot s)(1 + 0,68 \cdot s)} \quad (5.6)$$

O método de projeto é utilizado para localizar um ponto p que satisfizesse as inequações

$$\left. \begin{array}{l} 0,09 \leq p_1 \leq 2,0 \\ 20,0 \leq p_2 \leq 100,0 \\ T_1 \leq 7,0 \\ M_p \leq 0,2 \\ T_s \leq 20 \\ L_p = 0 \end{array} \right\} \quad (5.7)$$

»
» princípio

PROJETO E AJUSTE COMPUTACIONAL DE CONTROLADORES DE SISTEMAS DINÂMICOS PELO MÉTODO DAS INEQUAÇÕES

Digite os dados da relação $A(S)/B(S)$ dos dois polinômios de $G(S)$

São permitidos polinômios racionais somente em "s"

Entre com a ordem do polinômio do numerador: 0

Entre com a ordem do polinômio do denominador: 3

Coeficientes do Numerador

Digite os coeficientes em ordem decrescente

Entre com os coeficientes de S^0 : 1

Coeficientes do Denominador

Digite os coeficientes em ordem decrescente

Entre com os coeficientes de S^3 : 41.888

Entre com os coeficientes de S^2 : 82.136

Entre com os coeficientes de S^1 : 30.88

Entre com os coeficientes de S^0 : 1

Os tipos de estrutura dos controladores disponíveis são:

$$1) K(S,P) = \frac{P(1) (1+P(2)*S).....(1+P(N+1)*S)}{S^{**}L (1+P(N+2)*S).....(1+P(N+M+1)*S)}$$

e

$$2) K(S,P) = \frac{P(1)+P(2)*S+\dots+P(N+1)*S^{**N}}{1+P(N+2)*S+\dots+P(N+M+1)*S^{**M}}$$

Selecione o tipo de estrutura do controlador (tipo 1 ou 2): **1**

Entre com o valor de L: **1**

Entre com o valor de M: **0**

Entre com o valor de N: **1**

Existem limites nos parâmetros do controlador

se existem limites digite " 1 " se não digite " 0 ": **1**

Limites nos parâmetros

Limite inferior de **P^1: 0.09**

Limite superior de **P^1: 1.75**

Limite inferior de **P^2: 20.0**

Limite superior de **P^2: 100.0**

Entre com os valores do Ponto de partida

O ponto **P^1: 0.3**

O ponto **P^2: 20.0**

Digite a amplitude do degrau a ser aplicado ao sistema

Digite valores entre 0.0 e 1.0 (+ > 0)

São permitidos somente valores positivos

Entre com a amplitude do degrau a ser aplicado ao sistema: 1

Em continuação, dê entrada aos limites das inequações

limites no desempenho das inequações

Dê entrada aos limites do **ERRO DE ESTADO PERMANENTE**

Limite inferior: **0**

Limite superior: **0**

Dê entrada ao limite do **TEMPO DE SUBIDA**

Digite o limite superior como sendo positivo (≥ 0)

Limite superior: **7.0**

Dê entrada ao limite do **OVERSHOOT**

Digite o limite superior como sendo positivo (≥ 0)

Limite superior: **0.2**

Dê entrada ao limite do **TEMPO DE ACOMODAÇÃO**

Digite o limite superior como sendo positivo (≥ 0)

Limite superior: **20.0**

Dê entrada ao limite do **UNDERSHOOT**

Digite o limite superior como sendo positivo (≥ 0)

Limite superior: **0**

Dê entrada aos limites da **SAIDA DO CONTROLADOR**

Limite inferior **-10.0**

Dê entrada aos limites da **SAIDA DO CONTROLADOR**

Limite superior **9.8**

O TEMPO MÍNIMO SUGERIDO DE RESPOSTA É DE = **9.280201e+001**

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA = 3

DESENHO DO PROGRAMA

ENTRE COM O NÚMERO DE ITERAÇÕES = 5

Funcional P1:=3.000000e-001
Funcional P2:=20
Funcional P3:=0
Funcional P4:=8.149808e+000
Funcional P5:=1.458995e-001
Funcional P6:=2.502646e+001
Funcional P7:=0
Funcional P8:=4.985510e-001
Funcional P9:=6.211609e+000
Funcional P10:=-5.387814e-002

Detalhes da iteração Número 1:

Funcional P1:=3.300000e-001
Funcional P2:=20
Funcional P3:=0
Funcional P4:=7.574917e+000
Funcional P5:=1.664626e-001
Funcional P6:=2.233346e+001
Funcional P7:=0
Funcional P8:=3.025246e-001
Funcional P9:=6.818980e+000
Funcional P10:=-5.340997e-002

Detalhes da iteração Número 2:

Funcional P1:=4.200000e-001
Funcional P2:=22
Funcional P3:=0
Funcional P4:=6.006165e+000
Funcional P5:=2.370271e-001
Funcional P6:=2.225618e+001
Funcional P7:=0
Funcional P8:=-7.977635e-001
Funcional P9:=9.453782e+000
Funcional P10:=-4.682342e-002

Detalhes da iteração Número 3:

Funcional P1:=3.304499e-001
Funcional P2:=2.203000e+001
Funcional P3:=0
Funcional P4:=7.117552e+000
Funcional P5:=1.718988e-001
Funcional P6:=1.959373e+001
Funcional P7:=0
Funcional P8:=8.122276e-002
Funcional P9:=7.474173e+000
Funcional P10:=-4.720162e-002

SISTEMA DE MALHA FECHADA INSTÁVEL

Detalhes da iteração Número 4:

Funcional P1:=3.317998e-001
Funcional P2:=2.211999e+001
Funcional P3:=0
Funcional P4:=7.077592e+000
Funcional P5:=1.732390e-001
Funcional P6:=1.943086e+001
Funcional P7:=0
Funcional P8:=5.917615e-002
Funcional P9:=7.533077e+000
Funcional P10:=-4.695614e-002

Detalhes da iteração Número 5:

Funcional P1:=3.358493e-001
Funcional P2:=2.238996e+001
Funcional P3:=0
Funcional P4:=6.960551e+000
Funcional P5:=1.773349e-001
Funcional P6:=1.896685e+001
Funcional P7:=0
Funcional P8:=-8.923200e-003
Funcional P9:=7.711122e+000
Funcional P10:=-4.623955e-002

O ALGORITMO TERMINOU COM TODAS AS INEQUAÇÕES

SATISFEITAS SIMULTANEAMENTE

NÚMERO DE ITERAÇÕES REALIZADAS =5

**Funcional(1):=3.317998e-001
Funcional(2):=2.211999e+001
Funcional(3):=0
Funcional(4):=7.077592e+000
Funcional(5):=1.732390e-001
Funcional(6):=1.943086e+001**

Funcional(7):=0
Funcional(8):=5.917615e-002
Funcional(9):=7.533077e+000
Funcional(10):=-4.695614e-002

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 6

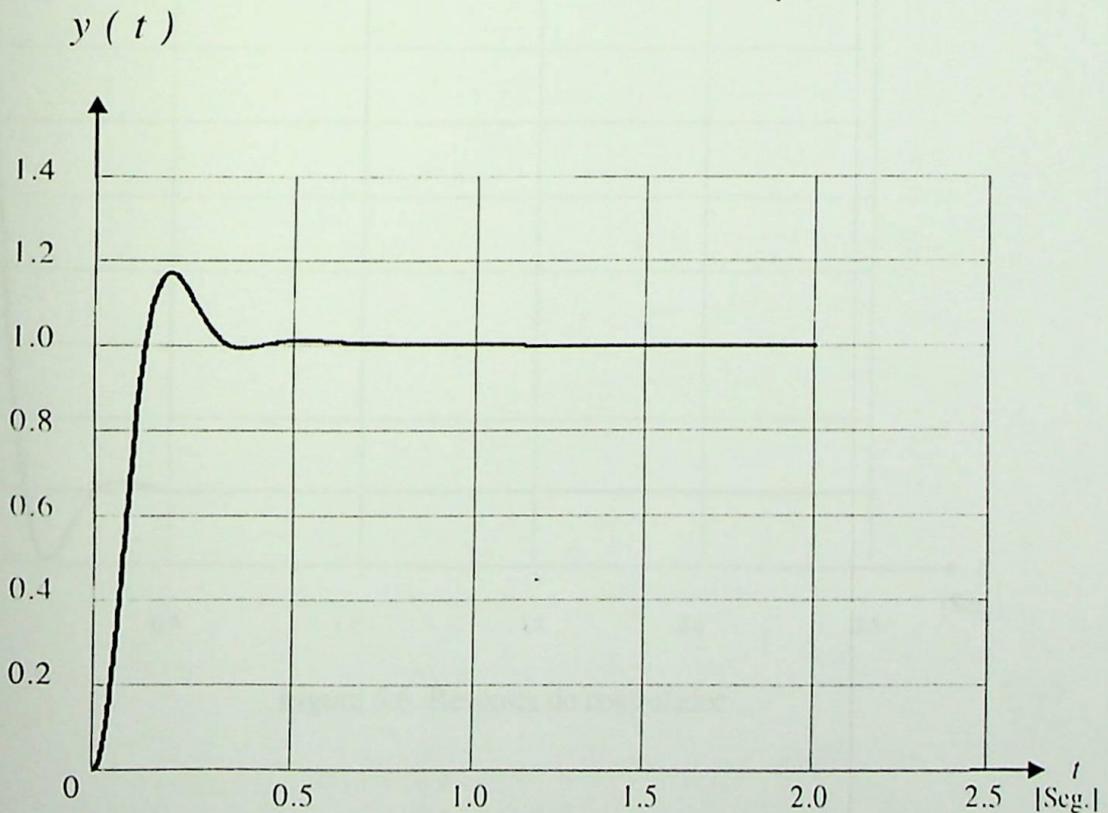


Figura 5.5 Resposta a degrau do Sistema

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 7

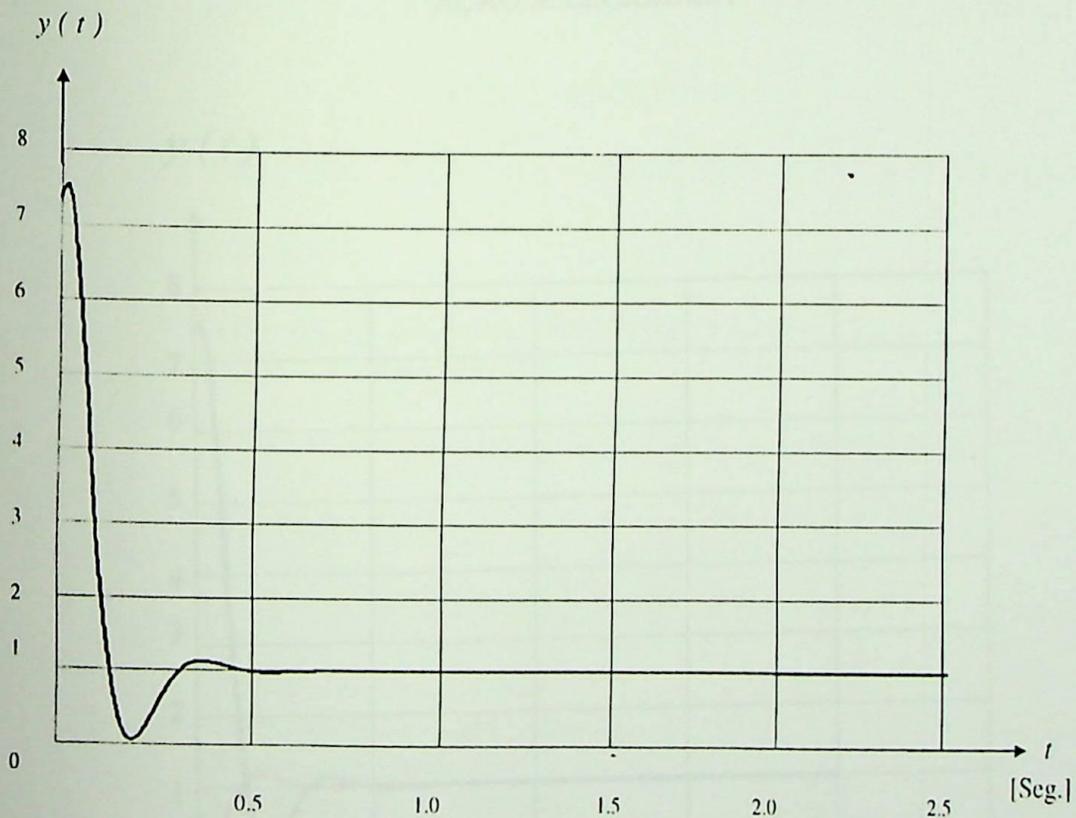


Figura 5.6 Resposta do controlador

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 8

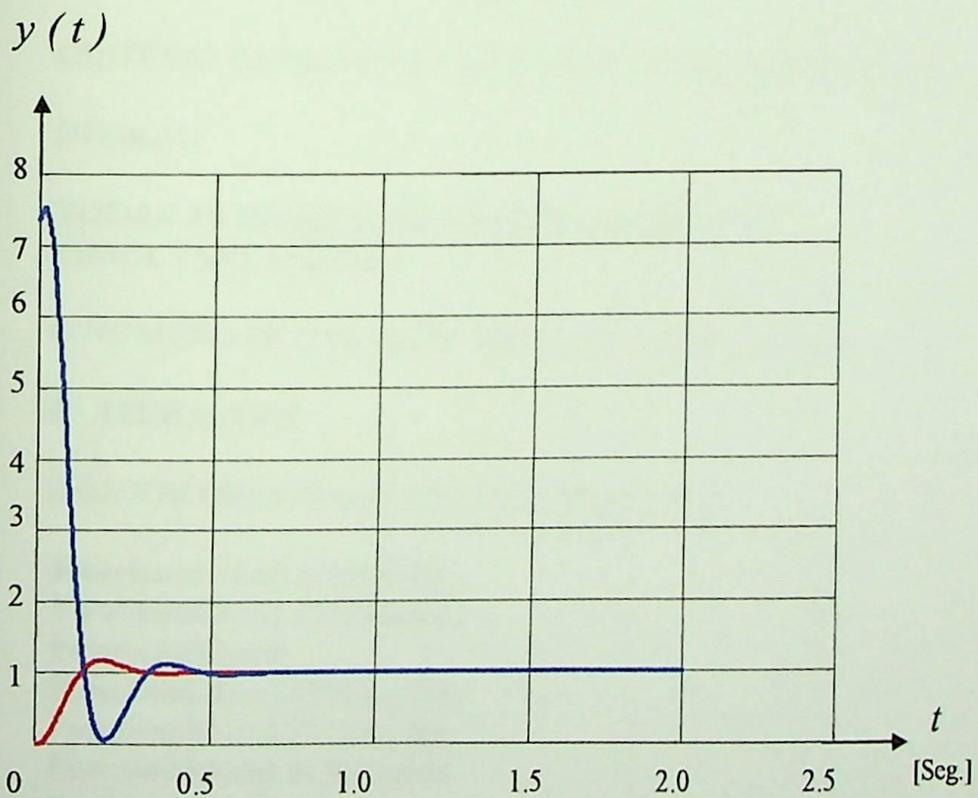


Figura 5.7 Resposta a degrau do Sistema e do Controlador.

SELECIONE AS SEGUINTE OPÇÕES

- 1.- ETDI ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES
- 2.- MDIN MUDAR OS LIMITES DAS INEQUAÇÕES
- 3.- TRIN TENTAR RESOLVER AS INEQUAÇÕES
- 4.- MCPP MUDAR AS COORDENADAS DO PONTO DE PARTIDA
- 5.- MFDC MUDAR A FORMA DO CONTROLADOR
- 6.- MRDS MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA
- 7.- MRDC MOSTRAR A RESPOSTA DO CONTROLADOR
- 8.- MSSC MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR
- 9.- EXIT SAÍDA DO PROGRAMA

AÇÃO SELECIONADA= 9

RESPOSTA A DEGRAU PARA O SISTEMA DE MALHA FECHADA

RESULTADOS FINAIS

CRITÉRIO DE PROJETO: CARACTERÍSTICAS DA RESPOSTA A
DEGRAU

**TODAS AS INEQUAÇÕES FORAM SATISFEITAS
SIMULTANEAMENTE**

O NÚMERO DE ITERAÇÕES LEVADAS A CABO FORAM:

5 ITERAÇÕES

E OS VALORES FINAIS DOS FUNCIONAIS SÃO :

Funcional(1):=3.317998e-001
Funcional(2):=2.211999e+001
Funcional(3):=0
Funcional(4):=7.077592e+000
Funcional(5):=1.732390e-001
Funcional(6):=1.943086e+001
Funcional(7):=0
Funcional(8):=5.917615e-002
Funcional(9):=7.533077e+000
Funcional(10):=-4.695614e-002

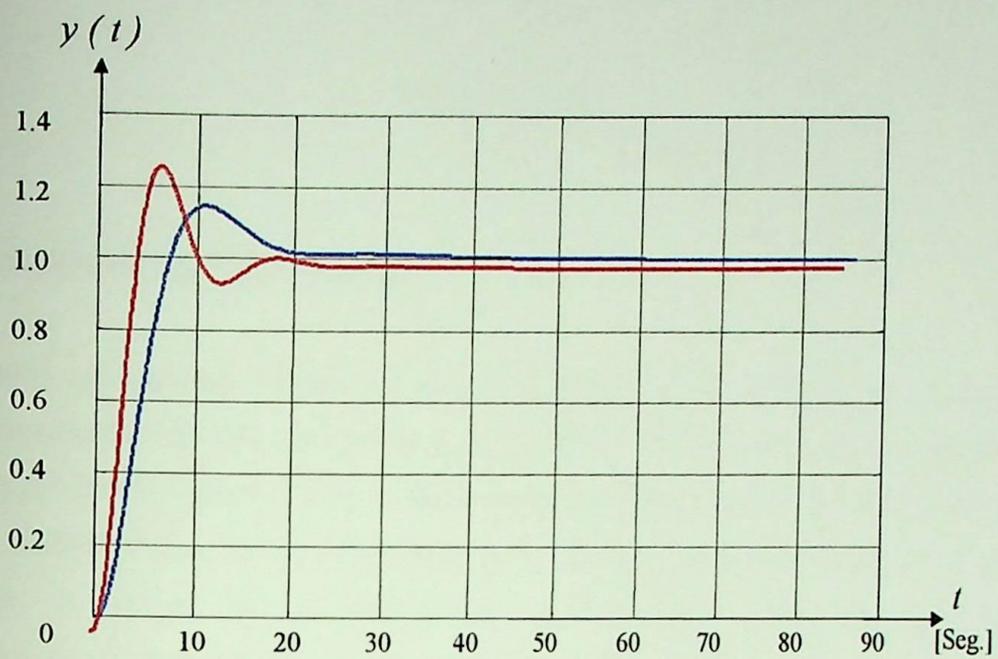


Figura 5.8 Gráfico comparativo das respostas a degrau do sistema compensado (linha azul) e do sistema sem compensação (linha vermelha)

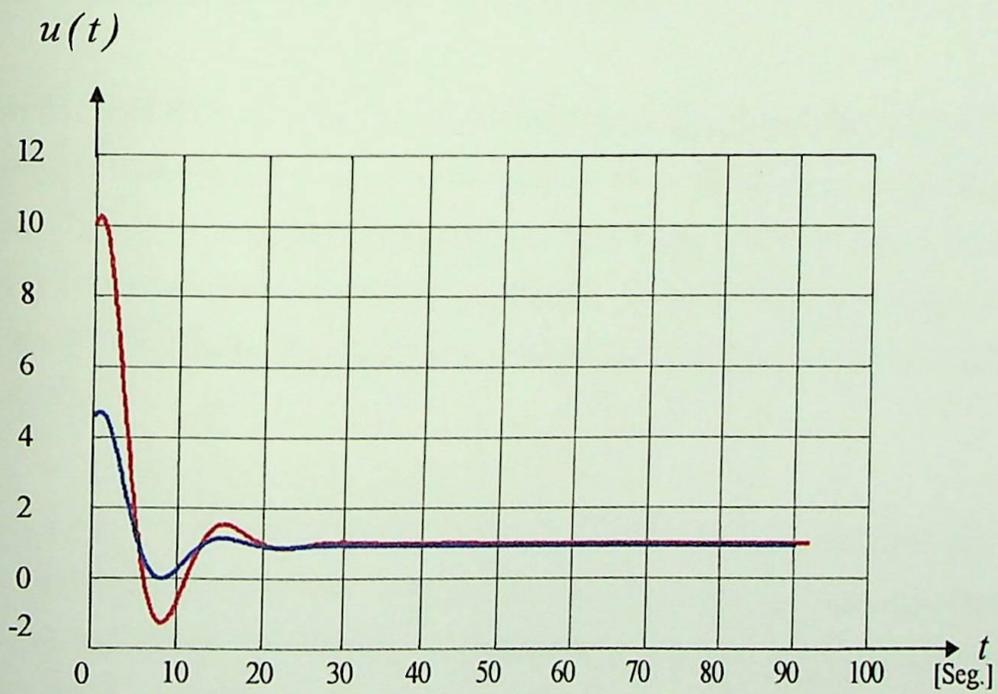


Figura 5.9 Saída do Controlador com compensação (linha azul).
Saída do Controlador sem compensação (linha vermelha).

CONCLUSÕES

6.1 TRABALHO DESENVOLVIDO

O trabalho apresentado tratou do desenvolvimento de um conjunto de rotinas e comandos computacionais para projeto e ajuste de controladores de sistemas dinâmicos pelo Método das Inequações (MI), voltado para a estrutura convencional de sistemas de controle realimentados, para realimentação unitária e com controlador em cascata. Esses comandos e rotinas foram desenvolvidos para execução dentro do ambiente computacional MATLAB, mas que são todos ativados a partir de um comando principal do tipo "script" [13], a partir do qual são feitas a entrada de dados do sistema e todas as demais operações de projeto e ajuste do controlador e de análise do sistema global. Dentro deste contexto, o conjunto de comandos e rotinas desenvolvido, como parte deste trabalho, será denominado simplesmente *programa*.

A referência ao MATLAB como sendo um ambiente computacional é feita por ser este um programa aplicativo e, ao mesmo tempo, uma linguagem de programação estruturada. Assim, o programa para projeto e ajuste de controladores pelo MI, por ter sido desenvolvido em uma linguagem estruturada, apresenta todas as vantagens de transparência e facilidade de compreensão, inerentes aos programas desenvolvidos em tais linguagens.

A base para o desenvolvimento deste trabalho foi a tese de doutorado de Coelho [8], e parte dos programas associados a esse documento. Deve ser ressaltado que esses programas foram escritos em linguagem FORTRAN IV, que é não estruturada, além de conter comandos específicos do computador utilizado, um DEC PDP-10, e de seus terminais gráficos. Portanto, o uso dos programas vinculados àquele trabalho tinham seu uso limitado àquele sistema computacional, o que evidentemente era uma dificuldade para que eventuais interessados no MI viessem a aplicá-lo de forma imediata.

6.2 ASPECTOS RELEVANTES

Em vista do exposto na seção anterior, é considerado que a contribuição relevante oferecida neste trabalho é uma nova versão de programa para projeto e ajuste computacional de controladores pelo Método das Inequações, o qual é executável em microcomputadores produzidos em larga escala e existentes em qualquer instituição ou empresa ligada à engenharia, e em um ambiente computacional amplamente conhecido dos engenheiros de controle. Por tanto, este trabalho torna o MI disponível para aplicação em larga escala.

A vantagem do MI é a sua proposta básica de formulação do problema de projeto através de um conjunto de inequações a serem satisfeitas; essas inequações definem as limitações de operação a serem observadas e as especificações a serem cumpridas pelo sistema. A tarefa básica a ser executada pelo programa de projeto é a determinação dos valores dos parâmetros do controlador que levam o sistema a respeitar as limitações de operação e a cumprir as especificações. O programa de projeto se mostrou eficiente na determinação dos valores dos parâmetros. Também constatou-se que o programa, baseado nestes critérios conseguiu encontrar uma solução ótima para os problemas de projeto formulados no capítulo 5, usados para testar o programa e para destacar as suas potencialidades.

O programa oferece ao usuário várias alternativas de execução para a busca da solução do problema de projeto, possibilitando-lhe ainda, se assim o desejar, fazer um refinamento dos resultados obtidos, e de confirmar os resultados através do traçado dos gráficos da resposta a degrau da saída e do erro do sistema. Com a ajuda destas e de outras opções disponíveis o usuário tem melhores condições de analisar a qualidade dos resultados alcançados, e de decidir pela aceitação ou não da solução determinada pelo programa, ou então tentar nova solução, através de uma nova entrada de dados, para reformulação das restrições de operação e das especificações do funcionamento do sistema. Este ciclo deve se repetir até que o usuário fique satisfeito com os resultados obtidos.

Nos testes do programa, o algoritmo utilizado para solução das inequações, conhecido como Processo dos Contornos Móveis (PCM) (ou *Moving Boundaries Process - MBP*,

em Inglês), se mostrou mais uma vez ser um método muito eficiente para a resolução de sistemas de inequações. Além disso, o programa como um todo funcionou bem, tendo sido repetidos com exatidão resultados obtidos com as versões anteriores desenvolvidas em FORTRAN [17,8,21]. Assim, o programa desenvolvido durante este trabalho traz consigo os meios, e por conseqüência a motivação, para que o MI seja aplicado em larga escala, para a resolução de problemas de controle de sistemas ou de processos dinâmicos, sempre que as limitações e as especificações de operação possam ser formuladas através de inequações que possam ser satisfeitas simultaneamente através do ajuste de parâmetros.

O programa tem alguns melhoramentos em relação às antigas versões em FORTRAN, entre elas o cálculo automático do tempo de simulação e do intervalo de integração, em função dos pólos de malha fechada do sistema, para cálculo da resposta a degrau, durante o processo iterativo de solução das inequações. O programa também leva em consideração, para o caso de perturbações do tipo degrau, uma das propostas básicas da Nova Formulação do Método das Inequações [17], de que o sistema controlado é *condicionalmente linear*, no sentido de que sua função de transferência só é válida se o sinal de controle estiver dentro de determinados limites. Este aspecto é contemplado pelo programa através da opção oferecida ao usuário de especificar a amplitude do degrau a ser aplicado ao sistema, e da inclusão das inequações onde são especificados os limites que devem ser respeitados pelo sinal de controle. Há uma razoável variedade de sistemas de controle em que as perturbações normalmente são do tipo degrau. Nestes casos, o projeto pelo MI, com a utilização deste recurso, levará a controladores bastante compatíveis com a realidade do sistema, podendo, portanto, ser implementados no sistema físico.

O programa desenvolvido para projeto e ajuste de controladores pelo MI contempla o problema básico de projeto de controladores em cascata com base em especificações da resposta a degrau. Vários problemas práticos se enquadram neste problema básico, e a disponibilidade de um programa que resolva este problema diretamente no domínio do tempo é, sem dúvida nenhuma, um recurso que poderá agilizar bastante os trabalhos de projeto de sistemas de controle, principalmente quando comparado aos métodos convencionais de lugar das raízes e resposta em frequência.

6.3 SUGESTÕES PARA FUTUROS TRABALHOS

Uma possível extensão do programa para problemas de controle mais elaborados poderá usar vários dos comandos computacionais auxiliares desenvolvidos, com destaque para o comando em que foi implementado o PCM. Por exemplo, futuros trabalhos poderão ser direcionados para ajuste de controladores que, além do controlador em cascata, também incluam um compensador de realimentação. Outra possibilidade seria a de formular as especificações em outros domínios que não o do tempo; tais domínios poderiam ser o da resposta em frequência, onde poderiam ser especificados limites para margem de fase, margem de ganho, pico e frequência de ressonância, frequência de corte, etc.; para sistemas lineares racionais, as especificações poderiam ser no plano complexo " s ", da transformada de Laplace, onde poderiam ser estabelecidos limites para as taxas de decaimento e relações de amortecimento dos pólos de malha fechada. E, em nível mais avançado, estender o MI para o projeto de controladores para sistemas multivariáveis, não lineares e para conjuntos quaisquer de sinais de entrada (referências e distúrbios), dentro das propostas apresentadas na Nova Formulação para o Método das Inequações [16] e de Projeto de Sistemas de Controle Robusto pelo Método das Inequações [8].

A filosofia do Método das Inequações, e a sua aplicação, podem se estender para muito além da área de sistemas de controle, pois qualquer problema que possa ser formulado em termos de um conjunto de inequações, a ser resolvido através de ajuste de parâmetros, pode ser resolvido pelo MI. As adaptações para outras áreas consistirão na identificação e separação adequada das partes do sistema com estruturas e valores de parâmetros fixos, e daquelas cujos parâmetros possam ser modificados, ou que possam receber novos componentes também ajustáveis. O outro passo importante é a definição das grandezas e índices de desempenho sujeitos a limites, das formas de calculá-los, e de seus limites, o que naturalmente levará à definição das inequações a serem satisfeitas através do ajuste de parâmetros. Cumpridas estas etapas, o problema estará pronto para ser resolvido pelo MI, com a utilização do Processo dos Contornos Móveis, ou de outro algoritmo para solução numérica de sistemas de inequações. Acredita-se que, sem grande esforço, é possível imaginar e identificar uma grande variedade de problemas de engenharia, bem como de outras áreas das Ciências Exatas, que possam ser formulados em termos de conjuntos de inequações.

Bibliografia

- [1] **APOSTOL, T.M.:** *Mathematical Analysis*, 2^a edição, Addison-Wesley Publishing Co., Reading, MA, EUA, 1974.
- [2] **CHEN, C.T.:** *Analysis and Synthesis of Linear Control Systems*, Holt, Rinehart and Wilson, Inc., New York, EUA, 1975.
- [3] **CLOSE, M.C., and FREDERICK, D.K.:** *Modeling and Analysis of Dynamic Systems*, Houghton Mifflin, Boston, EUA, 1978.
- [4] **COELHO, C.A.D.:** *CACCON Control Toolbox for the MATLAB Program (Version 3.2) – User’s Manual*, EFEI – Escola Federal de Engenharia de Itajubá, Itajubá, MG, Outubro de 1997.
- [5] **COELHO, C.A.D.:** “Root-Locus Based MATLAB Commands for Control System Design”, *1997 MATLAB Conference*, San Jose, CA, EUA, publicado por the MathWorks, Inc. na World Wide Web, no endereço . . .
http://www.mathworks.com/conf/user_contrib/coelho/, Natick, MA, EUA, 1997.
- [6] **COELHO, C.A.D.:** “Computer-Aided Control System Analysis and Parameter Setting Based on Root-Locus Properties”, *Anais do XII Congresso Brasileiro de Automática – XII CBA*, 14 a 18/09/1998, Uberlândia, MG, Vol. VI, pp. 1923-1928.
- [7] **COELHO, C.A.D.:** “Dominant pole Placement with Maximum Zero/Pole Ratio Phase-Lead Controllers”, *Proceedings of the 1998 American Control Conference*, Philadelphia, PA, EUA, 24 a 26/06/1998, Vol. 2, pp. 1159-1164.
- [8] **COELHO, C.A.D.:** *Design of Robust Control Systems by The Method of Inequalities*, Ph.D. Thesis, The University of Manchester Institute of Science and Technology, Manchester, Inglaterra, Julho de 1980.
- [9] **DAKESSIAN, V.G.:** *Design of Dynamic Control Systems by the Method of Inequalities*, Design Exercise Report, UMIST, Control Systems Centre, Manchester, Inglaterra, Março de 1980.
- [10] **D’AZZO, J.J. & HOUPIIS, C.H.:** *Análise e Projeto de Sistemas de controle Lineares* (tradução do Inglês), 2^a edição, Editora Guanabara Dois S.A., Rio de Janeiro, RJ, 1984.
- [11] **ELGERD, O.E.:** *Electric Energy Systems Theory*, T.M.H. Edition, Tata McGraw-Hill, New Delhi, Índia, 1973.
- [12] **OGATA, K.:** *Engenharia de Controle Moderno*, (tradução do Inglês), 3^a edição, Editora Prentice /Hall do Brasil Ltda., Rio de Janeiro, RJ, 1998.
- [13] **ROSENBROCK, H.H.:** “An Automatic Method for Finding the Greatest or Least Value of a Function”, *The Computer Journal*, Vol. 3, 1960, pp. 175-184.

- [13] **THE MATHWORKS, INC.:** *The Student Edition of MATLAB: Version 5.3 User's Guide*, Prentice Hall, Englewood Cliffs, NJ, 1999.
- [14] **THE MATHWORKS, INC.:** *Simulink User's Guide, The Student Edition of MATLAB: Version 5.3 User's Guide*, The MathWorks, Inc., Natick, MA, EUA, Abril/ 1999.
- [15] **WILSON, D.R.:** *Modern Practice in Servo Design*, Pergamon Press, Oxford, Inglaterra, 1970.
- [16] **ZAKIAN, V.:** "New Formulation for the Method of Inequalities", *Proceedings IEE*, Vol. 126, No. 6, June 1979, pp. 579-584.
- X [17] **ZAKIAN, V. and AL-NAIB, U.:** "Design of Dynamical and Control Systems by the Method of Inequalities", *Proceedings IEE*, Vol. 120, Nov. 1973, pp. 1421-1427.
- [18] **ZAKIAN, V.:** "The performance and Sensivity of Classical Control Systems", *International Journal of Systems Science*, Vol. 9, No. 3, 1977, pp. 343-355.
- [19] **COELHO, C.A.D.:** *Análise e Simulação de Sistemas Dinâmicos* (apostila), 1ª edição, EFEI – Escola Federal de Engenharia de Itajubá, Itajubá, MG, Brasil, Dezembro/1996.
- [20] **BACH, R.E.Jun.:** "A Practical Approach to Control System Optimization", *Proceedings of the IFAC Tokio Symposium on Systems Engineering for Control-System Design*, Japan, 1965, pp. 129-135.
- X [21] **COELHO, C.A.D.:** "Compensation of the Speed Governor of a Water Turbine by the Method of Inequalities", *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 101, No. 3, Sep. 1979, pp.205-211.
- [22] **LAUB, A.J. & LITTLE, J.N.:** *Control System Toolbox for Use with MATLAB - User's Guide*, The MathWorks, Inc., Natick, MA, EUA, 1986.

APRESENTAÇÃO

Este relatório é elaborado pela Direção de Engenharia e pela Comissão Interdisciplinar dos Estudos de Pós-graduação da Universidade Federal do Rio de Janeiro, com o objetivo de apresentar os resultados da pesquisa realizada no âmbito do projeto de pesquisa "Análise de estabilidade de estruturas de concreto armado sob o efeito de cargas dinâmicas". O trabalho foi desenvolvido sob a orientação do Prof. Dr. [nome], com a colaboração dos alunos de pós-graduação [nomes].

APÊNDICE A

GUIA DAS SUB-ROTINAS UTILIZADAS NO PROJETO

OBJETIVO: Apresentar uma seqüência de rotinas de cálculo para a análise de estabilidade de estruturas de concreto armado sob o efeito de cargas dinâmicas. Cada rotina realiza uma operação específica, como a leitura dos dados, a montagem da matriz de rigidez, a solução das equações de movimento, etc.

SÍMBOLOS

- BAZ = Índice do BAZ (BAZ = 0 quando a resposta é em relação ao deslocamento lateral)
- BAV = Índice da velocidade não é a solução
- BO = Número de iterações levadas a efeito
- BP = Valor de [parâmetro] que determina um critério de parada
- BS = Índice do BAZ; BSW = 1 indica que o programa deve parar quando a resposta de estabilidade; BSF = 1 indica que o programa deve continuar a trabalhar
- CB = [parâmetro] = Número de iterações que serão realizadas
- CF = Número máximo de iterações especificado pelo usuário
- EX = [parâmetro] = Índice do X, quando EX = 0 indica a solução de movimento para um estado específico P = P0; EX = 1 indica a solução de movimento para o estado de equilíbrio P = P0 + [parâmetro] = direção de P0

A.1 INTRODUÇÃO

Este apêndice é composto pela descrição do programa principal e das sub-rotinas individualmente. São dados detalhes e objetivos do programa principal e das suas sub-rotinas, as variáveis que os compõem, como as sub-rotinas são chamadas, o tipo de processamento que elas fazem e os dados que elas retornam ao programa principal, de modo a dar uma visão da estrutura da “toolbox” desenvolvida para projeto de sistemas de controle pelo Método das Inequações e permitir uma avaliação do trabalho de programação envolvido no desenvolvimento desta dissertação. Este conjunto de informações combinado com o guia do usuário, que contém as instruções para entrada de dados e execução do programa, são suficientes para a compreensão do funcionamento do programa.

A.2 Programa PRINCIPAL

OBJETIVO: Resolver um conjunto de inequações as quais representam as restrições e especificações do projeto, visando encontrar através de um vetor projeto (cujas coordenadas consigam satisfazer as inequações mencionadas) um controlador para um dado sistema particular.

ARGUMENTOS:

ISAT = Indicador ISAT: ISAT = 0 quando as inequações têm solução; caso contrário

ISAT = 1, as inequações não têm solução.

INO = Número de iterações levadas a cabo.

IOPT = Variável que armazena um comando dado pelo usuário.

ISW = Indicador ISW: ISW = 1, indica que o programa deve resolver somente a inequação de estabilidade (2.22); ISW = 2, indica que o programa deve resolver todas as inequações.

IC = m = Número de inequações que estão sendo resolvidas.

ITE = Número máximo de iterações especificadas pelo usuário.

LX = LX = Indicador LX, quando LX = 0, inicia-se a solução das inequações a partir de um ponto específico $P = P_0$; LX = 1, continua-se com a solução das inequações a partir da posição atual P e se procura as direções do vetor V_j (equações.(4.12));

LX=2, igual a LX = 1, mais a diferença está em que algumas inequações têm alterados os seus limites.

ITAD = Número extra de iterações especificadas pelo usuário, para continuar o processamento.

SUB-ROTINAS CHAMADAS INTERNAMENTE:

A partir do programa principal: IRLT, REGUL, PBND, BD, STAR, CLP, DEG, FORM, SFORM, PCM, DESIGN, YVAL, ROTAX, PRBND, LIDI, RESULTADOS.

A partir de PBND : BD.

A partir de CLP : FORM, SFORM.

A partir de DESIGN : YVAL.

A partir de PCM : DESIGN, ROTAX.

A partir de PRBND : LIDI.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

305 linhas de programa.

A.2.1 Sub-rotina IRLT

SINTAXE: [N,M,A,B]= irlt

OBJETIVO: O objetivo do programa é aceitar os valores do grau dos polinômios A(s)/B(s) da função de transferência e aceitar os dados de entrada dos coeficientes de ambos polinômios, tanto do numerador como do denominador da Função de Transferência.

ARGUMENTOS:

NUM = grau do polinômio do numerador de G(S).

DEN = grau do polinômio do denominador de G(S).

A = vetor formado pelos coeficientes do numerador da Função de Transferência.

B = vetor formado pelos coeficientes do denominador da Função de Transferência.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

82 linhas de programa.

A.2.2 Sub-rotina REGUL

SINTAXE: [Tipo,LR,MR,NR,L,NK,MK] = regul

OBJETIVO: O objetivo deste programa é aceitar os dados que definem o tipo de estrutura do Controlador $K(S)$, para logo utilizar este controlador no projeto através do Método das Inequações (MI).

ARGUMENTOS:

Tipo = variável que armazena o Tipo de controlador escolhido.

Tipo1 = tipo fatorado.

Tipo2 = tipo polinomial.

LR = numero de parâmetros do controlador.

MR = grau do Denominador de $K(S)$.

NR = grau do Numerador de $K(S)$.

NK = N na estrutura do controlador (Tipo 1 ou 2).

MK = M na estrutura do controlador (Tipo 1 ou 2).

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

108 linhas de programa.

A.2.3 Subrotina PBND

SINTAXE: [G,H]= pbnd (Modo,LR)

OBJETIVO: O objetivo deste programa é aceitar, ler ou modificar os limites inferior ou superior dos parâmetros do controlador.

ARGUMENTOS:

G = Vetor que contém os limites inferiores de (c'_i).

H = Vetor que contém os limites superiores de (c_i).

Modo = indicador: se Modo=1 se aceita a entrada de dados ou a leitura dos valores do ponto p de um arquivo.

LR = Variável que indica o número de parâmetros do controlador.

SUB-ROTINAS CHAMADAS INTERNAMENTE:

A partir de PBND : BD.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

78 linhas de programa.

A.2.4 Sub-rotina BD

SINTAXE: [G,H] = bd (LB,LR)

OBJETIVO: O objetivo deste programa é de fazer a leitura dos limites de entrada do ponto de partida P[p1, p2, p3]. Esses limites serão superiores e inferiores.

ARGUMENTOS:

G = Vetor que contém os limites inferiores de (c'_i)

H = Vetor que contém os limites superiores de (c_i)

LB = indicador, que quando LB = 0, coloca G = 0, e força a $H \geq 0$; e quando LB = 1, aceita a entrada de valores de G e H.

LR = variável que indica o número de parâmetros do controlador.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

62 linhas de programa.

A.2.5 Sub-rotina STAR

SINTAXE: P = pstar (G,H,Modo,LR)

OBJETIVO: O objetivo deste programa é ler ou modificar os valores iniciais do ponto de partida P.

ARGUMENTOS:

Modo = variável que define se os valores do ponto de partida estão armazenados em arquivo ou serão digitados na execução do programa, Modo = 1 os valores serão lidos dum arquivo ou poderão ser digitados durante a execução, Modo = 2 serve para modificar os valores destes parâmetros.

LR = variável que contém o número de parâmetros do controlador a ser utilizado.

P (i) = vetor que armazena os valores do ponto de partida Po.

G = Vetor que contém os limites inferiores de (c' i).

H = Vetor que contém os limites superiores de (c i).

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

66 linhas de programa.

A.2.6 Sub-rotina CLP

SINTAXE:

[NQ,MQ,NU,QN,QD,UN,EN,STM]= clp (Tipo,L,NK,MK,NR,MR,A,B,P)

OBJETIVO: Calcula a função de transferência em malha fechada de $W_e(s)$, $W_u(s)$ e $W_y(s)$ e a abscissa de estabilidade do Sistema.

ARGUMENTOS:

Tipo = variável que armazena o Tipo de controlador escolhido.

L = número de integradores no denominador da FT de $K(S)$.
N = grau do polinômio do numerador.
M = grau do polinômio do denominador.
MR = grau do Denominador de $K(S)$.
NR = grau do Numerador de $K(S)$.
A = vetor formado pelos coeficientes do numerador da função de transferência.
B = vetor formado pelos coeficientes do denominador da função de transferência.
P = Vetor que contém as coordenadas do ponto de partida.
NQ = grau do numerador de $W_y(S)$.
MQ = grau do comum denominador de $W_e(S)$, $W_u(S)$ e $W_y(S)$.
NU = grau do numerador de $W_u(S)$.
QN = conjunto dos coeficientes do polinômio do numerador de $W_y(S)$.
QD = conjunto dos coeficientes do comum denominador dos polinômios $W_e(S)$,
 $W_u(S)$ e $W_y(S)$.
UN = conjunto dos coeficientes do numerador de $W_u(S)$.
EN = conjunto dos coeficientes do numerador do polinômio $W_e(S)$.
STM = $\sigma(A)$ = abscissa de estabilidade do sistema em malha fechada.
NK = valor de N na estrutura do controlador.
MK = valor de M na estrutura do controlador.

SUB-ROTINAS CHAMADAS INTERNAMENTE:

A partir de CLP : FORM, SFORM.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

51 linhas de programa.

A.2.7 Sub-rotina DEGRAU

SINTAXE: [DEG]= degrau

OBJETIVO: O objetivo da sub-rotina é aceitar o valor da amplitude do degrau que será aplicado ao sistema para levantar as características de resposta do mesmo à excitação padronizada degrau.

ARGUMENTOS:

DEG = amplitude do degrau a ser aplicado ao sistema.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

33 linhas de programa.

A.2.8 Sub-rotina FORM

SINTAXE: [RN,RD] = form (NI,IPN,IPD,P)

OBJETIVO: O objetivo desta sub-rotina é a de calcular a função de transferência do Controlador do Tipo número 1, escolhido na sub-rotina REGUL.

ARGUMENTOS:

NR = grau do numerador de $K(S)$.

MR = grau do denominador de $K(S)$.

NI = L na estrutura do controlador $K(S)$ e representa o número de controladores.

IPN = NK na estrutura do controlador de $K(S)$ e representa grau do numerador de $K(S)$.

IPD = MK na estrutura do controlador de $K(S)$.

RN = Conjunto dos coeficientes do polinômio do numerador de $K(S)$, $RN(i)$, $i = 1, 2, \dots, NR + 1$.

RD = Conjunto dos coeficientes do polinômio do denominador de $K(S)$, $RD(i)$, $i = 1, 2, \dots, MR + 1$.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

134 linhas de programa.

A.2.9 Sub-rotina SFORM

SINTAXE: [RN, RD] = sform (IPN, IPD)

OBJETIVO: O objetivo deste programa é o de calcular s FT de $K(S)$ do tipo 2.

ARGUMENTOS:

IPN = NK na estrutura do controlador de $K(S)$ e representa grau do numerador de $K(S)$.

IPD = MK na estrutura do controlador de $K(S)$.

RN = Conjunto dos coeficientes do polinômio do numerador de $K(S)$, $RN(i)$,
 $i=1,2,\dots,NR+1$.

RD = Conjunto dos coeficientes do polinômio do denominador de $K(S)$, $RD(i)$,
 $i=1,2,\dots,MR+1$.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

27 linhas de programa.

A.2.10 Sub-rotina PCM

SINTAXE:

[ISAT,INO,FK]=pcm(LX,LR,P,G,H,IC,STM,ISW,Tipo,L,NK,MK,NR,MR,A,B,ITE,-
DEG)

OBJETIVO: O objetivo deste programa é de resolver o sistema de inequações da forma (3.10).

ARGUMENTOS:

LX = Indicador LX, quando $LX = 0$, inicia-se a solução das inequações a partir de um ponto específico $P = P_0$; $LX = 1$, continua-se com a solução das inequações a partir da posição atual P e se procura as direções do vetor V_j (equações.(4.12));

$LX=2$, igual a $LX = 1$, mais a diferença está em que algumas inequações têm alterados os seus limites.

ISAT = Indicador ISAT: $ISAT = 0$ quando as inequações têm solução e caso contrário

$ISAT = 1$, as inequações não têm solução.

INO = Número de iterações levadas a cabo.

LR = variável que indica o número de parâmetros do controlador.

P = Vetor que contém as coordenadas do ponto de partida.

G = Vetor que contém os limites inferiores de (c'_i) .

H = Vetor que contém os limites superiores de (c_i) .

FK = Vetor, $FK(i) = \phi_i(p^k)$, onde p^k é o último ponto de sucesso que se obteve com o PCM.

SUB-ROTINAS CHAMADAS INTERNAMENTE:

A partir de PCM : DESIGN, ROTAX.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

395 linhas de programa.

A.2.11 Sub-rotina DESIGN

SINTAXE:

[FK,STM] = design (P,H,G,IC,ISW,Tipo,L,NK,MK,NR,MR,A,B,LR,DEG)

OBJETIVO: O objetivo desta sub-rotina é o de calcular a transformada de LaPlace das funções de transferência de malha fechada da saída $Yg(s)$ e de $Yu(s)$.

ARGUMENTOS:

FK = Vetor, $FK(i) = \phi_i(p^k)$, aonde p^k é o último ponto de sucesso que se obteve com o PCM.

STM = Abscissa de estabilidade do sistema.

SUB ROTINAS CHAMADAS INTERNAMENTE:

A partir de DESIGN : YVAL.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

27 linhas de programa.

A.2.12 Sub-rotina YVAL

SINTAXE:

[RDY,RDU,RMIN,RMAX,desempRDY,SVY,SVE,NT]=yval(QD,QN,EN,UN,MQ,NQ)

OBJETIVO: Objetivo desta sub-rotina é o de computar a resposta a degrau para o controlador e a saída do sistema.

ARGUMENTOS:

QN = conjunto dos coeficientes do polinômio do numerador de $W_y(S)$.

QD = conjunto dos coeficientes do comum denominador dos polinômios $W_e(S)$, $W_u(S)$ e $W_y(S)$.

EN = conjunto dos coeficientes do numerador do polinômio $W_e(S)$.

NQ = grau do numerador de $W_y(S)$.

MQ = grau do comum denominador de $W_e(S)$, $W_u(S)$ e $W_y(S)$.

RDY = Vetor de coeficientes $RDY(i)$, $i=1,2,\dots$ da resposta a degrau da saída do sistema $Yg(s)$.

AA = Vetor dos coeficientes do comum denominador dos polinômios $W_e(s)$, $W_u(s)$ e $W_y(s)$.

RDU = Vetor dos coeficientes $RDU(i)$, $i=1,2,\dots$ da resposta a degrau do controlador $W_u(t)$.

RMIN = mínimo valor da posição i do vetor RDU.

RMAX = máximo valor da posição i do vetor RDU.

TF = Tempo final de simulação.

TRDY = Número de passos para computar a resposta a degrau de Y .

HMAX = Largura dos passos de processamento.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

51 linhas de programa.

A.2.13 Sub-rotina ROTAX

SINTAXE: [V] = ROTAX(LR,V,D)

OBJETIVO: O objetivo desta sub-rotina é computar um novo conjunto de vetores ortonormais V_j , definindo uma nova direção de busca para o PCM (equações (4.11) e (4.12)).

ARGUMENTOS:

LR = Variável que indica o número de parâmetros do controlador [p].

V = Vetor que armazena os componentes do vetor V_j .

D = Vetor que armazena a direção de busca do PCM.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

89 linhas de programa.

A.2.14 Sub-rotina PRBND

SINTAXE: [NCH,INEQ,GN,HN]= prbnd (Modo,LR,G,H)

OBJETIVO: O objetivo deste programa é aceitar, ler ou modificar os limites inferior ou superior dos parâmetros do controlador.

ARGUMENTOS:

G = Vetor que contém os limites inferiores de (c'_i).

H = Vetor que contém os limites superiores de (c_i).

Modo = indicador: se Modo=1 se aceita a entrada de dados ou a leitura dos valores do ponto p de um arquivo.

LR = Variável que indica o número de parâmetros do controlador.

SUB-ROTINAS CHAMADAS INTERNAMENTE:

A partir de PRBND : LIDI.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

78 linhas de programa.

A.2.15 Sub-rotina LIDI

SINTAXE: [G,H]= lidi (LR,IN)

OBJETIVO: O objetivo deste programa é de fazer a leitura dos parâmetros de entrada do ponto P[p1,p2,p3] superiores e inferiores.

ARGUMENTOS:

G = Vetor que contém os limites inferiores de (c'_i).

H = Vetor que contém os limites superiores de (c_i).

LB = indicador, que quando LB=0, coloca G=0, e força a H \geq 0; e quando LB=1, aceita a entrada de valores de G e H.

LR = variável que indica o número de parâmetros do controlador.

IN = índice que vem do programa PRBND.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

116 linhas de programa.

A.2.16 Sub-rotina RESULTADOS

SINTAXE: resultados (FK,G,H)

OBJETIVO: O objetivo desta sub-rotina é a de apresentar os resultados do processamento do programa na tela do computador num formato específico.

ARGUMENTOS:

G = Vetor que contém os limites inferiores de (c'_i).

H = Vetor que contém os limites superiores de (c_i).

FK = Vetor, $FK(i) = \phi_i(p^k)$, onde p^k é o último ponto de sucesso que se obteve com o PCM.

NÚMERO DE LINHAS DE PROGRAMAÇÃO:

18 linhas de programa.

RESUMO :

O número total de linhas de programação realizadas no projeto é de 1698 linhas, entre o programa principal e as 15 sub-rotinas todas escritas na linguagem do aplicativo MATLAB que é uma linguagem muito parecida com a linguagem C.

APÊNDICE B

Comandos relevantes mais utilizados da "Toolbox" CACCOB
e da "Control System Toolbox" do MATLAB.

Este apêndice indica o número de ocorrências dos comandos relevantes MATLAB da Toolbox CACCOB (14) e da Control System Toolbox (22) do MATLAB, aplicados ao programa de Projeto de Controlador.

APÊNDICE B

Comandos relevantes mais utilizados da “Toolbox” CACCON e da “Control System Toolbox” do MATLAB.

Este apêndice inclui a descrição resumida dos principais comandos MATLAB da Toolbox CACCON [4] e da Control System toolbox [22] do MATLAB, usados no programa de Projeto de Controle.

1 - Comando "SOMAPOLI"

SINTAXE : $c = somapoli (a, b, epse)$

AÇÃO : Soma de polinômios: $c (s) = a (s) + b (s)$

a, b = vetores com os coeficientes de cada parcela.

c = vetor com os coeficientes do polinômio soma.

$epse$ = Tolerância (2% ou 5%).

2 - Comando "CARACRD"

SINTAXE : $desemp = caracrd (t, v, yrp, epse, indt)$

AÇÃO : Determina os índices de desempenho da resposta a degrau $y(t)$

t = vetor de tempos; supõe-se que $t(1) = 0, y(1) = 0$

$yrp = y (\infty)$ não pode ser nulo

$desemp = [tr tp Mo\% ts Td Lp\%]$

tr = Tempo de subida (tempo necessário para a resposta atingir 90% do valor final pela primeira vez).

tp = Tempo de pico.

$Mo\%$ = Ultrapassagem máxima (percentual)

Ts = tempo de acomodação (tempo requerido para a resposta alcançar e permanecer dentro de uma faixa de 2% ou 5% do valor final).

$epse$ = Tolerância (2% ou 5%).

Td = Período de oscilação.

$Lp\%$ = Ultrapassagem mínima (percentual).

3 - Comando "STEP"

Este comando é da "Control System Toolbox"

SINTAXE: $[yout] = step(Num, Den, TY)$

AÇÃO: Calcula a resposta a Degrau da Função de Transferência
 $G(s) = Num(s) / Den(s)$, onde Num e Den contêm os coeficientes do polinômio em potências decrescentes de "s".

Num = Conjunto de coeficientes do polinômio do numerador da FT.

Den = Conjunto de coeficientes do polinômio do comum denominador de FT.

TY = Vetor $TY(i)$, $i = 1, 2, \dots$, dos $t(i)$ tempos de resposta de $y(t)$

GUIA PARA O USUÁRIO DO PROGRAMA DE PROJETO

Este aplicativo realiza o projeto de um sistema de controle por meio de um algoritmo de projeto no computador. A fim de facilitar o uso do aplicativo, este guia descreve os procedimentos e as definições necessárias que o usuário precisa conhecer.

PROJETO DE CONTROLE E DE SISTEMAS LINEARES

APLICAÇÃO DO MÉTODO DAS INSCRIÇÕES

APÊNDICE C

GUIA PARA O USUARIO DO PROGRAMA DE PROJETO

Este apêndice inclui um pequeno manual de instruções para o usuário do programa de projeto no computador. Nele se detalha como devem ser introduzidos os dados no programa e os diferentes recursos que o programa oferece ao usuário.

Para começar a executar o programa deve-se digitar desde o MATLAB o nome PRINCIPAL em letras minúsculas, para que apareça a máscara do programa

principal

PROJETO DE CONTROLE DE SISTEMAS LINEARES

ATRAVÉS DO MÉTODO DAS INEQUAÇÕES

Logo aparecerá a mensagem de :

Digite os dados da relação $A(S)/B(S)$ dos dois polinômios de $G(S)$

São permitidos somente polinômios racionais em " s "

Isto quer dizer que a Função de Transferência da Unidade básica deverá estar no formato de um quociente de dois polinômios, ou seja, um numerador e um denominador, deverá ser digitado o maior grau de ambos polinômios.

Entre com o grau do polinômio do numerador: _?_

Entre com o grau do polinômio do denominador: _?_

Logo se deverá dar entrada aos coeficientes dos polinômios acima mencionados, com o cuidado de ingressar esses dados em ordem decrescente, ou seja, primeiro os coeficientes de mais alto grau e assim sucessivamente. Se você digitar algum dado a mais, este não será aceito pelo programa.

Coefficientes do Numerador

Digite os coeficientes em ordem decrescente

Entre com os coeficientes de S^0 : _?_

Para os coeficientes do denominador a entrada de dados é igual a entrada de dados do numerador, sempre digitando-se primeiro os coeficientes de mais alto grau e assim sucessivamente.

Coeficientes do Denominador

Digite os coeficientes em ordem decrescente

Entre com os coeficientes de S³: _?_

Entre com os coeficientes de S²: _?_

Entre com os coeficientes de S¹: _?_

Entre com os coeficientes de S⁰: _?_

Logo aparecerão na tela os tipos de controladores que o usuário poderá escolher.

Os tipos de estrutura dos controladores disponíveis são:

$$1.-) K(S, P) = \frac{P(1)(1+P(2)*S).....(1+P(N+1)*S)}{S**L(1+P(N+2)*S).....(1+P(N+M+1)*S)}$$

e

$$2.-) K(S, P) = \frac{P(1)+P(2)*S+.....+P(N+1)*S**N}{1+P(N+2)*S+.....+P(N+M+1)*S**M}$$

Digitando os números 1 (um) ou 2 (dois), o usuário poderá escolher o tipo de controlador que ele deseja utilizar, para logo dar entrada no valor de N, que somado de uma unidade é equivalente ao grau do polinômio do numerador do controlador. Depois dar entrada no valor de M, que somado ao valor de N e de mais uma unidade corresponderá ao grau do polinômio do denominador do controlador. Por último se deverá dar entrada ao valor de L, que corresponde a existência de integradores na estrutura do controlador.

Selecione o tipo de estrutura do controlador (tipo 1 ou 2): _?_

Entre com o valor de L: _?_

Entre com o valor de M: _?_

Entre com o valor de N: _?_

Após ter dado entrada nesses dados, o programa perguntará ao usuário se os parâmetros do controlador têm limites, ou seja, se os $[P_1, P_2, \dots, P_N]$ estão obrigados a satisfazer algumas restrições dadas através de limites. Caso os parâmetros tenham limites, estes devem ser digitados em ordem crescente, primeiro o limite inferior, para logo ser digitado o limite superior, com o devido cuidado de que o limite inferior deverá ser menor que o limite superior, se por acaso ocorrer um engano e o usuário tentar dar entrada no limite superior com um valor menor que o limite inferior o programa retornará uma mensagem de erro e aguardará a correta entrada de dados.

Existem limites nos parâmetros do controlador ?

se existem limites digite " 1 " senão digite " 0 ": _?_

Limites nos parâmetros

Limite inferior de P^1 : _?_

Limite superior de P^1 : _?_

Limite inferior de P^2 : _?_

Limite superior de P^2 : _?_

Limite inferior de P^3 : _?_

Limite superior de P^3 : _?_

O programa perguntará ao usuário para dar entrada nas coordenadas do ponto de partida, ou dito de outra maneira, nos parâmetros do vetor de projeto $P = [P_1, P_2, \dots, P_N]$.

Entre com os valores do Ponto de partida

O ponto P^1 : _?_

O ponto P^2 : _?_

O ponto P^3 : _?_

Em continuação, o programa perguntará ao usuário sobre o valor da amplitude do degrau que será aplicado ao sistema.

Digite a amplitude do degrau a ser aplicado ao sistema

Digite valores entre 0.0 e 1.0 ($+ > 0$)

São permitidos somente valores positivos

Entre com a amplitude do degrau a ser aplicado ao sistema: _?_

O seguinte passo será dar entrada aos valores dos limites dos índices de desempenho que correspondem às inequações que o programa deve satisfazer, sempre com o cuidado de digitar o limite inferior, menor que o limite superior.

LIMITES NO DESEMPENHO DAS INEQUAÇÕES

DÊ ENTRADA AOS LIMITES DO ERRO DE ESTADO

PERMANENTE

Limite inferior : _?_

Limite superior : _?_

DÊ ENTRADA AO LIMITE DO TEMPO DE SUBIDA

Digite o limite superior como sendo positivo (≥ 0)

Limite superior : _?_

DÊ ENTRADA AO LIMITE DO OVERSHOOT

Digite o limite superior como sendo positivo (≥ 0)

Limite superior : _?_

DÊ ENTRADA AO LIMITE DO TEMPO DE ACOMODAÇÃO

Digite o limite superior como sendo positivo (≥ 0)

Limite superior : _?_

DÊ ENTRADA AO LIMITE DO UNDERSHOOT

Digite o limite superior como sendo positivo (≥ 0)

Limite superior : _?_

DÊ ENTRADA AOS LIMITES DA SAÍDA DO CONTROLADOR

Limite inferior : _?_

DÊ ENTRADA AOS LIMITES DA SAÍDA DO CONTROLADOR

Limite superior : _?_

O TEMPO MÍNIMO SUGERIDO DE RESPOSTA É DE =

1.717629e+001

Uma vez que todos os limites foram digitados, o programa mostrará o seguinte menu na tela do computador.

SELECIONE AS SEGUINTE OPÇÕES

- | | | |
|-----|------|--|
| 1.- | ETDI | ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES |
| 2.- | MDIN | MUDAR O DESEMPENHO DAS INEQUAÇÕES |
| 3.- | TRIN | TENTAR RESOLVER AS INEQUAÇÕES |
| 4.- | MCPP | MUDAR AS COORDENADAS DO PONTO DE PARTIDA |
| 5.- | MFDC | MUDAR A FORMA DO CONTROLADOR |
| 6.- | MRDS | MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA |

- | | | |
|-----|------|---|
| 7.- | MRDC | MOSTRAR A RESPOSTA A DEGRAU DO CONTROLADOR |
| 8.- | MSSC | MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR |
| 9.- | EXIT | SAÍDA DO PROGRAMA |

Vamos simular que a opção a seguir tenha sido a de tentar resolver as inequações, então temos que:

AÇÃO SELECIONADA= 3

PROJETO DE CONTROLE

O programa perguntará ao usuário para que ele dê entrada a um número de iterações que o programa deverá realizar na procura de uma solução. Pela prática em projetos, deverá se dar um número no máximo igual a 20. Se a solução não for alcançada dentro desse número de iterações, o usuário poderá dar entrada a um número extra de iterações, para tentar achar uma solução.

ENTRE COM O NÚMERO DE ITERAÇÕES = _?_

Logo o programa mostrará o número total de inequações com os respectivos valores calculados para os índices de desempenho, a partir dos componentes do ponto de partida escolhido.

Funcional P1:=1
 Funcional P2:=1
 Funcional P3:=1
 Funcional P4:=0
 Funcional P5:=1.427915e+000
 Funcional P6:=4.858037e-001
 Funcional P7:=1.270936e+001
 Funcional P8:=0
 Funcional P9:=-4.858037e-001
 Funcional P10:=1
 Funcional P11:=-2.910990e-001

À medida que o processamento for realizado, o programa irá imprimindo um resultado a cada iteração levada a cabo.

Detalhes da interação Número 1:

Funcional P1:= **
Funcional P2:= **
Funcional P3:= **
Funcional P4:= **
Funcional P5:= **
Funcional P6:=**
Funcional P7:= **
Funcional P8:= **
Funcional P9:= **
Funcional P10:=**
Funcional P11:= **

Detalhes da interação Número 2:

Funcional P1:=**
Funcional P2:=**
Funcional P3:= **
Funcional P4:= **
Funcional P5:= **
Funcional P6:= **
Funcional P7:= **
Funcional P8:= **
Funcional P9:= **
Funcional P10:= **
Funcional P11:= **

E assim sucessivamente.

Se com os valores ingressados e as especificações requeridas o programa encontrar uma solução, será mostrada uma mensagem na tela do computador como a seguinte :

O ALGORITMO TERMINOU COM TODAS AS INEQUAÇÕES

SATISFEITAS SIMULTANEAMENTE

NÚMERO DE ITERAÇÕES REALIZADAS = _?_

e logo mostrará na tela o conteúdo de todas as funcionais com os pontos do vetor do projeto que conseguiram satisfazer todas as inequações.

Funcional (1)=9.047252e-001
Funcional (2)=1.147054e+000
Funcional (3)=1.963507e-002
Funcional (4)=0
Funcional (5)=9.399930e-001
Funcional (6)=0
Funcional (7)=1.229860e+000
Funcional (8)=0
Funcional (9)=-8.856939e-001
Funcional (10)=5.285283e+001
Funcional (11)=-8.108804e-001

Em seqüência, o programa voltará a mostrar na tela o menu principal, para que o usuário possa escolher outras opções de processamento ou para dar por finalizada a seção.

Dentre as alternativas que o usuário pode escolher, tem-se as de mudar os limites das inequações, verificar o sistema de dados, mudar as coordenadas do ponto de partida, mudar o formato do controlador, mostrar a resposta A degrau através de um gráfico do sistema, o mesmo para o controlador ou mostrar ambos gráficos juntos, e, para finalizar, sair do programa. O usuário poderá tentar toda sorte de opções, até que ele se sinta satisfeito com as respostas ao seu problema.

SELECIONE AS SEGUINTE OPÇÕES

- | | | |
|-----|------|--|
| 1.- | ETDI | ESCREVER NA TELA O DESEMPENHO DAS INEQUAÇÕES |
| 2.- | MDIN | MUDAR O DESEMPENHO DAS INEQUAÇÕES |
| 3.- | TRIN | TENTAR RESOLVER AS INEQUAÇÕES |
| 4.- | MCPP | MUDAR AS COORDENADAS DO PONTO DE PARTIDA |
| 5.- | MFDC | MUDAR A FORMA DO CONTROLADOR |
| 6.- | MRDS | MOSTRAR A RESPOSTA A DEGRAU DO SISTEMA |

- | | | |
|-----|------|---|
| 7.- | MRDC | MOSTRAR A RESPOSTA A DEGRAU DO CONTROLADOR |
| 8.- | MSSC | MOSTRAR A SAÍDA DO SISTEMA E DO CONTROLADOR |
| 9.- | EXIT | SAÍDA DO PROGRAMA |

AÇÃO SELECIONADA= 9

RESPOSTA A DEGRAU PARA O SISTEMA FINAL DE MALHA FECHADA

RESULTADOS FINAIS

CRITÉRIO DE PROJETO: CARACTERÍSTICAS DA RESPOSTA A DEGRAU

TODAS AS INEQUAÇÕES FORAM SATISFEITAS SIMULTANEAMENTE

O NÚMERO DE ITERAÇÕES LEVADAS A CABO FORAM:

12 ITERAÇÕES

E OS VALORES FINAIS DOS FUNCIONAIS SÃO :

Funcional (1)=9.047252e-001
Funcional (2)=1.147054e+000
Funcional (3)=1.963507e-002
Funcional (4)=0
Funcional (5)=9.399930e-001
Funcional (6)=0
Funcional (7)=1.229860e+000
Funcional (8)=0
Funcional (9)=-8.856939e-001
Funcional (10)=5.285283e+001
Funcional (11)=-8.108804e-001