Campus de São Carlos

IMPLEMENTAÇÃO DE UM ALGORITMO PID AUTO-AJUSTÁVEL

EDSON STRADIOTTO

ORIENTAÇÃO: PROFª DRª VILMA ALVES DE OLIVEIRA

UNIVERSIDADE DE SÃO PAULO

**ESCOLA DE ENGENHARIA
DE SÃO CARLOS**

✓
UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ELÉTRICA

IMPLEMENTAÇÃO DE UM ALGORITMO PID AUTO-AJUSTÁVEL

Dissertação apresentada junto ao Departamento de Engenharia Elétrica da EESC/USP como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Autor: Eng. Edson Stradiotto
Orientadora: Dra. Vilma A. de Oliveira

São Carlos
- julho de 1996 -

Class.	TESP-EESC
Cutt.	5902
Tombo	203/96

Eletricidade

0745842

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC-USP

Stradiotto, Edson
S895i Implementação de um algoritmo PID auto-
ajustável. -- São Carlos, 1996.

Dissertação (Mestrado) -- Escola de Engenharia
de São Carlos-Universidade de São Paulo, 1996.
Orientador: Profa. Dra. Vilma Alves de
Oliveira

1. Controle adaptativo. 2. Estimção
paramétrica 3. Controle PID. I. Título

FOLHA DE APROVAÇÃO

Dissertação defendida e aprovada em 29-08-1996
pela Comissão Julgadora:

Profa. Doutora VILMA ALVES DE OLIVEIRA - Orientadora-
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Prof. Doutor MANOEL LUIS DE AGUIAR
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Prof. Doutor RONALDO GUIMARÃES CORRÊA
(Universidade Federal de São Carlos - UFSCar)

Coordenador da área "Engenharia Elétrica"
P/ Prof. Dr. GERALDO R. MARTINS DA COSTA

Presidente da Comissão de Pós-Graduação
Prof. Dr. JOSÉ CARLOS A. CINTRA

Para Meus Pais
Leonardo e Maria.

Agradecimentos

À Professora Vilma A. de Oliveira, pela sua orientação.

Aos Professores Azauri A. de Oliveira Jr., Manoel L. de Aguiar e Geraldo R. M. da Costa pelo seu auxílio e sugestões valiosas.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, pela bolsa concedida para realização deste trabalho.

À secretária Rosane, pela ajuda.

Aos técnicos Sílvio, Jesus, Rui Berto, Petrúcio, José Luis, João e Jerson, e aos desenhistas Odair e Marcelo Charaba pela colaboração.

Aos colegas, Annete, Amarildo, Armando, Bonato, Bia, Carlos, Kitty, Glória, José Roberto, Evelina, Marcio, Marcos, Celso, Tainha, Magda, Nelson, Flávio, Flajola, Márcia, Catanduva, Ali, Ben Hur, Lineu e todos aqueles cujos nomes não me recordo agora.

Aos meus irmãos Alex, Susana e Bete e meu cunhado Dema.

SUMÁRIO

1 - Introdução	1
2 - Controladores PID	5
2.1 - Modelos de Controladores PID	5
2.2 - Modelos de Sintonia Automática	7
2.2.1 - Controlador de Ganho Programado	7
2.2.2 - Controlador Auto-Ajustável	9
2.2.3 - Controlador Auto-Sintonizado	11
3 - Algoritmo PID Auto-Ajustável	16
3.1 - Formulação do Problema	18
3.2 - Algoritmo de Estimação Paramétrica	26
4 - Implementação dos Algoritmos	29
4.1 - Estimação de Parâmetros	30
4.2 - Auto-Ajuste do Controlador PID	33
4.3 - Projeção dos Parâmetros	36
4.4 - Algoritmos de Estimação e Controle	42
4.5 - Implementação em Linguagem C	43
5 - Implementação em Laboratório	45
5.1 - Interface de Potência	46
5.1.1 - Sistema <i>PWM</i> Digital	47

5.1.2 - Sistema <i>CHOPPER</i>	49
5.2 - <i>Encoder Óptico</i>	51
6 - Resultados de Simulação	52
6.1 - Ajuste de Parâmetros	53
6.2 - Ajuste do Controlador Sem Ajuste de Parâmetros	60
6.3 - Ajuste do Controlador	64
6.4 - Simulação em Linguagem C	89
7 - Resultados Experimentais	100
7.1 - Controle de Velocidade	101
7.2 - Controle de Posição	105
7.3 - Variação Paramétrica	109
7.4 - Variação de Carga	113
8 - Conclusão	117
Bibliografia	120

RESUMO

O presente trabalho trata da implementação de um sistema de controle PID auto-ajustável. O trabalho desenvolvido compreende as seguintes etapas: ajuste dos parâmetros da planta, ajuste do controlador sem ajuste dos parâmetros da planta e ajuste do controlador com projeção dos parâmetros do controlador. O algoritmo de ajuste do controlador é baseado no critério de variância mínima generalizada e o de ajuste de parâmetros da planta é baseado no método dos mínimos quadrados estendido. Os parâmetros do controlador são projetados na região factível de existência física dos parâmetros do controlador contínuo através de uma técnica de programação quadrática.

Os algoritmos foram implementados e testados em laboratório no controle de posição ou velocidade de um servomotor cc. Utilizou-se um computador pessoal AT-286, 25MHz, para executar os algoritmos, um sistema de potência com um modulador de largura de pulso digital associado a um recortador de onda e um encoder óptico para realimentação de velocidade.

ABSTRACT

The present work deals with the implementation of a self-tuning PID control. The work developed comprises the tuning of the plant parameters, the tuning of the controller without tuning the plant parameters and the tuning of the controller with projection of the controller parameters. The controller algorithm is based on the generalised minimum-variance criterion and the tuning of the plant parameters is based on the extended recursive least square method. The parameters of the controller are projected onto the region of physical existence of the continuous controller parameters by a quadratic programming technique. This guarantees a PID behavior for the digital controller.

The algorithms were implemented and tested in laboratory for position and velocity control of a servomotor dc. We used a personal computer AT-286, 25MHz, to run the algorithms, a power electronic driver with a digital pulse width modulate associate with a chopper and an optic encoder for velocity feedback.

Lista de Figuras

2.1	Estruturas de controladores PID.	6
2.2	Controlador de ganho programado.	8
2.3	Controlador auto-ajustável explícito.	9
2.4	Controlador auto-ajustável implícito.	10
2.5	Controlador auto-sintonizado.	11
2.6	Diagrama de blocos do controlador auto-sintonizado usando relé.	13
5.1	Diagrama de blocos do sistema de controle completo.	46
5.2	Esquema do <i>PWM</i> .	48
6.1	Sinais de entrada, saída real e estimada (curvas sobrepostas).	54
6.2	Parâmetros a_1 , a_2 e a_3 .	55
6.3	Parâmetros b_0 e b_1 .	56
6.4	Parâmetros c_1 e c_2 .	56
6.5	Sinais de entrada, saída real e estimada (curvas sobrepostas).	58
6.6	Parâmetros a_1 , a_2 e a_3 do motor.	59
6.7	Parâmetros b_0 e b_1 do motor.	59
6.8	Parâmetros c_1 e c_2 do motor.	60
6.9	Parâmetros do controlador.	61

6.10 Sinais de entrada, saída e referência.	62
6.11 Parâmetros g_0 , g_1 e g_2 do controlador.	63
6.12 Sinais de entrada, saída e referência do motor.	63
6.13 Sinais de entrada, saída e referência.	64
6.14 Parâmetros g_0 , g_1 e g_2 do controlador.	65
6.15 Parâmetros a_1 , a_2 e a_3 .	66
6.16 Parâmetros b_0 e b_1 .	66
6.17 Parâmetros c_1 e c_2 .	67
6.18 Sinais de entrada, saída e referência do motor	68
6.19 Parâmetros g_0 , g_1 e g_2 do controlador.	69
6.20 Parâmetros a_1 , a_2 e a_3 do motor.	69
6.21 Parâmetros b_0 e b_1 do motor.	69
6.22 Parâmetros c_1 e c_2 do motor.	70
6.23 Sinais de entrada, saída e referência (curvas sobrepostas).	72
6.24 Parâmetros a_1 , a_2 e a_3 .	73
6.25 Parâmetros b_0 e b_1 .	73
6.26 Parâmetros c_1 e c_2 .	74
6.27 Parâmetros g_0 , g_1 e g_2 do controlador.	74
6.28 Restrições dos parâmetros do controlador	75
6.29 Sinais de entrada, saída e referência do motor	76
6.30 Parâmetros a_1 , a_2 e a_3 do motor.	76
6.31 Parâmetros b_0 e b_1 do motor.	77
6.32 Parâmetros c_1 e c_2 do motor.	77

6.33 Parâmetros g_0 , g_1 e g_2 do controlador.	78
6.34 Restrições dos parâmetros do controlador.	78
6.35 Sinais de entrada, saída, referência e esforço conjugado (100xNm) do motor.	79
6.36 Parâmetros a_1 , a_2 e a_3 do motor.	80
6.37 Parâmetros b_0 e b_1 do motor.	81
6.38 Parâmetros c_1 e c_2 do motor.	81
6.39 Parâmetros g_0 , g_1 e g_2 do controlador.	82
6.40 Restrições controlador.	83
6.41 Sinais de entrada, saída e referência do motor (velocidade).	84
6.42 Sinal de saída e referência do motor (posição)	85
6.43 Parâmetros a_1 , a_2 e a_3 do motor.	86
6.44 Parâmetros b_0 e b_1 do motor.	87
6.45 Parâmetros c_1 e c_2 do motor.	87
6.46 Parâmetros g_0 , g_1 e g_2 do controlador.	88
6.47 Restrições do controlador.	89
6.48 Sinais de entrada, saída e referência.	90
6.49 Parâmetros g_0 , g_1 e g_2 do controlador.	91
6.50 Parâmetros a_1 , a_2 e a_3 .	91
6.51 Parâmetros b_0 e b_1 .	92
6.52 Parâmetros c_1 e c_2 .	92
6.53 Sinais de entrada, saída e referência do motor	93
6.54 Parâmetros a_1 , a_2 e a_3 do motor.	94
6.55 Parâmetros b_0 e b_1 do motor.	94

6.56	Parâmetros c_1 e c_2 do motor.	95
6.57	Parâmetros g_0 , g_1 e g_2 do controlador.	95
6.58	Sinal de saída e referência do motor (posição)	96
6.59	Sinais de entrada, saída e referência do motor (velocidade).	97
6.60	Parâmetros g_0 , g_1 e g_2 do controlador.	97
6.61	Parâmetros a_1 , a_2 e a_3 do motor.	98
6.62	Parâmetros b_0 e b_1 do motor.	98
6.63	Parâmetros c_1 e c_2 do motor.	99
7.1	Sinais de entrada, saída e referência.	102
7.2	Parâmetros a_1 , a_2 e a_3 .	103
7.3	Parâmetros b_0 e b_1 .	103
7.4	Parâmetros c_1 e c_2 .	104
7.5	Parâmetros g_0 , g_1 e g_2 do controlador.	105
7.6	Sinal de saída e referência de posição.	106
7.7	Sinais de entrada, saída e referência de velocidade.	106
7.8	Sinal de saída e referência de posição.	108
7.9	Sinais de entrada, saída e referência de velocidade.	108
7.10	Sinais de entrada, saída e referência de velocidade do motor com variação paramétrica	110
7.11	Sinais de entrada, saída e referência de velocidade do motor com variação paramétrica	111

7.12	Parâmetros g_0 , g_1 e g_2 do controlador com variação paramétrica.	111
7.13	Parâmetros a_1 , a_2 e a_3 do motor, com variação paramétrica.	112
7.14	Parâmetros b_0 e b_1 do motor, com variação paramétrica	112
7.15	Parâmetros c_1 e c_2 do motor com variação paramétrica.	113
7.16	Sinais de entrada, saída e referência de velocidade do motor com variação de carga	114
7.17	Sinal de saída e referência de posição com variação de carga.	115
7.18	Sinais de entrada, saída e referência de velocidade do motor com variação de carga	116

CAPÍTULO 1

INTRODUÇÃO

No atual estágio de desenvolvimento tecnológico, os sistemas de controle devem ser extremamente precisos e funcionais, devem ser versáteis, mantendo o desempenho mesmo com a troca de componentes danificados da planta controlada, ou ainda, mesmo com alterações significativas no meio ambiente de trabalho.

Quando o modelo de um sistema pode ser completamente especificado, com variações paramétricas desprezíveis em um ambiente de trabalho normal, podem ser usadas técnicas clássicas de projeto. Porém, quando isto não ocorre, técnicas de projeto adaptativas têm sido utilizadas.

O controle adaptativo é uma área importante do controle moderno, permitindo tratar sistemas na presença de incertezas, perturbações estruturais e mudanças no meio ambiente. Podendo ser utilizado diretamente no processo ou apenas para ajustar os parâmetros de controladores fixos.

O controle adaptativo pode ser pensado como uma generalização do controle clássico, onde os coeficientes da lei de controle são invariantes no

tempo e pré-definidos pelo projetista, mas com alguns ou todos os coeficientes ajustados em resposta a medições em tempo real do sistema ou variações de perturbações [01].

Um sistema de controle adaptativo deve ser projetado para garantir estabilidade e robustez. Deve-se pensar também na necessidade de algoritmos de adaptação rápidos para compensar a variação dos parâmetros e na necessidade de utilizar pequenos intervalos de computação [02]. De forma contrária, o que seria uma solução tornar-se-ia um problema, e o longo tempo de processamento certamente prejudicaria o desempenho do sistema.

O presente trabalho aborda o desenvolvimento e implementação de um sistema de controle por computador utilizando-se um algoritmo de controle PID auto-ajustável.

Visando implementar o algoritmo de controle auto-ajustável, o sistema de controle foi dividido em duas partes distintas: unidade de processamento responsável pela execução dos programas de controle e estimação de parâmetros, e sistema de instrumentação responsável pela aquisição de dados e pela atuação na planta.

O controlador atuará na planta, um servomotor cc, através de um recortador de onda (*CHOPPER*)

transistorizado, acionado por um modulador de pulsos (*PWM*) digital. O *PWM* colocado junto à unidade de processamento, deve gerar os pulsos de funcionamento do *CHOPPER*, que por sua vez irá acionar a planta. Entre o *PWM* e o *CHOPPER* são utilizados acopladores-ópticos para evitar que falhas ocorridas na planta danifiquem o controlador. O sistema *CHOPPER-PWM* foi escolhido por ser utilizado em máquinas de grande potência, bem como em máquinas de baixa potência.

O projeto do PID auto-ajustável utilizado segue o desenvolvimento de Tousson e Lyung [03] e de Vega [04].

Neste trabalho o ajuste de parâmetros da planta foi realizado pelo método dos mínimos quadrados estendido. Um algoritmo de projeção dos parâmetros no espaço de sua existência baseado na técnica de programação quadrática [05] por multiplicadores de Lagrange foi utilizado. Resultados de simulação e de laboratório também são apresentados.

Esta dissertação está organizada da seguinte forma:

No capítulo dois é feita uma breve revisão bibliográfica sobre os controladores PID e suas formas de ajuste. No capítulo três é descrito um algoritmo auto-ajustável em associação com um algoritmo de ajuste de parâmetros da planta. No Capítulo quatro é tratado

detalhadamente os passos do trabalho desenvolvido onde cada item citado equivale a uma etapa de desenvolvimento do mesmo. No Capítulo cinco é feita uma descrição do equipamento usado na parte prática deste trabalho. No Capítulo seis são reunidos os resultados obtidos em simulação, no sete os resultados obtidos em experimentos práticos e no Capítulo oito são apresentadas as conclusões deste trabalho.

CAPÍTULO 2

CONTROLADORES PID

Basicamente os controladores discretos no tempo tiveram sua origem no tempo contínuo. Sua estrutura foi mantida sendo incorporado amostradores que discretiza o valor do dado no momento certo e seguradores que o mantém durante o tempo de amostragem, evitando a perda do mesmo. Dos modelos, considerados simples, os mais utilizados são os P(proporcional), PI(proporcional integral), PD(proporcional derivativo) e o PID(proporcional integral e derivativo).

O controlador P considera apenas o nível final a ser atingido, o PI considera o nível e a velocidade que o sinal desenvolve, o PD considera o nível e quanto falta para atingí-lo, e o PID considera todos os fatores, ou seja, a velocidade do sinal, o ponto final e quanto falta para atingí-lo.

2.1 Modelos de Controladores PID

Há muitas maneiras de se implementar um controlador PID, cada uma com vantagens e desvantagens inerentes à própria estrutura que devem ser analisadas

e ponderadas pelo projetista.

Estas estruturas consideram cada um dos termos do controlador PID dando destaque a uma ou mais particularidades da planta a ser controlada, ou a um desempenho particular de um sinal de referência pré-determinado. A Fig.2.1 mostra três variações da estrutura do controlador PID.

Fig.2.1: Estruturas de controladores PID.

Na estrutura da Fig.2.1a, a parte derivativa do controlador pode causar ganhos elevados quando ocorre mudanças abruptas no sinal de referência. Esta situação pode ser evitada aplicando-se a ação derivativa apenas na saída $y(t)$, como mostra a Fig.2.1b. Esta estrutura, da Fig.2.1b, é mais utilizada na literatura [06].

A estrutura da Fig.2.1c, onde os módulos proporcional e derivativo são aplicados apenas na saída

do sistema, não introduz zeros adicionais na função de transferência do sistema em malha fechada, o que é conveniente dado que no caso discreto os zeros podem ter grande influência na saída [07].

2.2 Modelos de Sistemas de Sintonia Automática

Há um grande número de produtos disponíveis na literatura que providenciam uma sintonia automática. Muitos destes podem ajustar continuamente os coeficientes do controlador para seguir a referência. Quase todos estes controladores usam diferentes aproximações para ajustar a malha de controle, e essas aproximações variam na concepção, bem como na complexidade.

Na maioria dos casos o controlador adaptativo é um controlador PID simples, associado a um algoritmo de ajuste de seus coeficientes. A diferença entre estes controladores está na forma em que os coeficientes do controlador são ajustados.

2.2.1 Controlador de Ganho Programado

No controlador de ganho programado ou controlador de ganho adaptativo, o ganho, e em alguns casos os termos derivativo e integrativo, são ajustados

automaticamente em tempo real, como é visto na Fig.2.2. Estes ajustes são pré-definidos pelo usuário como

Fig.2.2: Controlador de Ganho Programado.

funções simples das variáveis mensuráveis, da saída do controlador, do sinal de erro ou de algum sinal externo.

Uma vez definido, este esquema pré-programado permanece fixado e a programação é definida para ajustar os coeficientes da malha para compensar as não-linearidades do processo ou dos atuadores. Claramente todos os benefícios do ganho programado poderão apenas ser atingidos se as características do processo com respeito às variáveis mensuráveis são bem entendidas e

se estas relações não variam com o tempo.

2.2.2 Controlador Auto-Ajustável

Os controladores auto-ajustáveis têm recebido muita atenção nestes últimos vinte anos. Podem ser divididos em duas categorias distintas: auto-ajustável explícito e auto-ajustável implícito.

O auto-ajustável explícito tem duas funções distintas, como mostra a Fig.2.3. Primeiro, usando valores passados da entrada e da saída do processo,

Fig.2.3: Controlador auto-ajustável explícito.

desenvolve-se uma representação matemática (ou modelo)

do processo a ser controlado. Em seguida, os coeficientes do controlador são ajustados baseados no modelo do processo.

No auto-ajustável implícito, Fig.2.4, tanto o ajuste dos parâmetros do modelo do processo como a sintonia do controlador estão em um único algoritmo.

Fig.2.4: Controlador auto-ajustável implícito.

2.2.3 Controlador Auto-Sintonizado

Em contraste aos auto-ajustáveis, os controladores auto-sintonizados, Fig.2.5, têm sido desenvolvidos quase que independentemente dos trabalhos acadêmicos. Ao passo que os auto-ajustáveis usam uma descrição formal do processo e procedimentos de projeto do controlador que podem ser usados durante o desenvolvimento do projeto do sistema, o auto-sintonizado mimica os procedimentos práticos que seriam

Fig.2.5: Controlador auto-sintonizado.

realizados pelo engenheiro durante o comissionamento do sistema.

Os auto-sintonizados trabalham através da monitoração e análise do sinal de erro do processo em resposta a transientes de inicialização, e a mudanças do sinal de referência, ou a distúrbios de carga. Uma vez analisadas as respostas do sistema em termos de tempo de subida, sobrepassagem, tempo de acomodação, etc. regras heurísticas ou técnicas freqüenciais frequentemente baseadas nos procedimentos de Ziegler-Nichols [08] são utilizadas para modificar os coeficientes do controlador.

Aström e Hägglund [09] desenvolveram um algoritmo baseado no conhecimento de pontos da curva de Nyquist para determinar a dinâmica do sistema. Um relé ideal ou com histerese é usado para forçar o ciclo-limite de oscilação do sistema.

A ideia básica reside em usar a regra de Ziegler-Nichols para ajustar o controlador PID baseando-se na observação de que os parâmetros do controlador podem ser determinados a partir do conhecimento de um ponto da curva de Nyquist do sistema em malha aberta. Este ponto é a intersecção da curva de Nyquist com o eixo real negativo, que é descrito em termos do ganho crítico, K_c , e do período crítico, t_c .

No esquema original de Ziegler-Nichols [08], o ganho crítico e o período crítico são determinados da seguinte forma: um regulador proporcional é conectado

ao sistema; o ganho crítico é gradualmente incrementado até que uma oscilação é obtida; quando este ganho ocorre tem-se o ganho crítico e o período desta oscilação é o período crítico.

O método para a obtenção de K_c e t_c é baseado na observação de que um sistema com fase atrasada de pelo menos π pode oscilar com período t_c sob o controle de um relé. Para determinar o ganho crítico e o período crítico, o sistema é conectado em uma malha realimentada como mostra a Fig.2.6. Quando o período de oscilação do sinal de erro $e(t)$ é igual a t_c , e a amplitude do relé é d , o primeiro harmônico da série de Fourier da saída do relé tem amplitude de $4d/\pi$. Se a amplitude da saída do processo é a , o ganho crítico é aproximadamente igual a:

$$K_c \propto \frac{4d}{\pi a} \quad (2.01)$$

Fig.2.6: Diagrama de blocos do controlador auto-sintonizado usando relé.

Note que este resultado descreve uma função de transferência aproximada para o relê ideal, definida por $N(a)$ ou:

$$N(a) \propto \frac{4d}{\pi a} \quad (2.02)$$

Existem vantagens no uso de outras não linearidades além do relê ideal. Um relê com histerese dá ao sistema pouca sensibilidade à ruídos mensuráveis.

É possível modificar este procedimento a fim de determinar outros pontos na curva de Nyquist. Um integrador pode ser adicionado em série com o relê visando obter o ponto crítico. Outros métodos, baseados em outros procedimentos, são descritos em Aström e Hägglund [10].

Uma vez conhecido o ponto crítico e a frequência crítica dada por $\omega_c = 2\pi/t_c$ é simples determinar os parâmetros do controlador. No caso do controlador PID:

$$G_c = K \left(1 + sT_d + \frac{1}{sT_i} \right) \quad (2.03)$$

a condição seguinte é obtida considerando o argumento da função de transferência da malha aberta com $\omega_c = \phi_m - \pi$:

$$\omega_c T_d - \frac{1}{\omega_c T_i} = \operatorname{tg} \phi_m \quad (2.04)$$

onde ϕ_m representa a margem de fase com o sistema em

malha fechada

Existem muitos T_i e T_d que satisfazem (2.04).

Uma possibilidade de escolha é dada por:

$$T_i = \alpha T_d \quad (2.05)$$

Em combinação à condição (2.05) uma solução para T_d é dada por:

$$T_d = \frac{\operatorname{tg} \phi_n + \sqrt{\frac{d}{a} + \operatorname{tg}^2 \phi_n}}{2\omega_c} \quad (2.06)$$

Cálculos simples mostram que uma função de transferência tem ganho unitário para $\omega = \omega_c$ se o ganho do controlador for escolhido como:

$$K = \frac{\cos \phi_n}{|G(j\omega_c)|} = K_c \cos \phi_m \quad (2.07)$$

Há muitas outras possibilidades de escolha, uma delas é escolher:

$$T_d = \omega_c T_i \quad (2.08)$$

CAPÍTULO 3

ALGORITMO PID AUTO- AJUSTÁVEL

Os algoritmos de controle auto-ajustáveis podem ser uma solução melhor para os problemas de controle. Eles são hábeis para auto ajustar os parâmetros do controlador e adaptar-se a variações de carga quando certas condições são satisfeitas [04][06][11][12].

O projeto do controlador PID desenvolvido utiliza a abordagem de Toulsson e Ljung[03]. O método baseia-se na minimização de um índice quadrático de predição de erro e no desempenho do controlador sobre um período de tempo.

Os controladores PID são em geral implementados na forma digital e são discretizados a partir do controlador PID contínuo no tempo dado por:

$$u(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(r) dr + T_d \frac{de(t)}{dt}] \quad (3.01)$$

onde $u(t)$ é o sinal de controle e $e(t)$ é o sinal de erro entre a referência $r(t)$ e a saída $y(t)$ do sistema no instante t . Discretizando-se (3.01) por aproximação *forward* para integral temos:

$$u(t) = \frac{g_0 + g_1 z^{-1} + g_2 z^{-2}}{1 - z^{-1}} e(t) \quad (3.02)$$

ou

$$u(t) = \frac{G(z^{-1})}{1 - z^{-1}} e(t) \quad (3.03)$$

onde K , T_i , T_d são respectivamente o ganho, e os tempos integral e derivativo do controlador. Os parâmetros g_i são funções de K , T_i , T_d , e o tempo de amostragem T_s e, são representados por [04]:

$$g_0 = K \left(1 + \frac{T_d}{T_s} + \frac{T_s}{T_i} \right) \quad (3.04a)$$

$$g_1 = -K \left(1 + 2 \frac{T_d}{T_s} \right) \quad (3.04b)$$

$$g_2 = K \frac{T_d}{T_s} \quad (3.04c)$$

No tempo contínuo os parâmetros do controlador PID são escalares positivos. Desta forma, os parâmetros discretizados devem satisfazer algumas condições que são dadas por:

$$g_2 \geq 0 \quad (3.05a)$$

$$2g_2 + g_1 \leq 0 \quad (3.05b)$$

$$g_2 + g_1 + g_0 \geq 0 \quad (3.05c)$$

E se o ganho K for negativo, estas equações são

facilmente modificadas de forma adequada.

3.1 Formulação do Problema

Considere a planta representada pelo modelo Controlador Auto-Regressivo de Média Móvel, com iniciais em inglês CARMA [04][13]:

$$A(z^{-1})y(t) = z^{-k}B(z^{-1})u(t) + C(z^{-1})\xi(t) + d \quad (3.06)$$

onde

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_nz^{-n}$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + \dots + b_mz^{-m}$$

$$C(z^{-1}) = 1 + c_1z^{-1} + \dots + c_lz^{-l}$$

são polinômios em função do operador de atraso z^{-1} satisfazendo $A(0)=C(0)=1$ e $C(z)=0$ para $|z| > 1$. Os sinais $u(t)$ e $y(t)$ representam os sinais de entrada e saída do sistema, respectivamente. O distúrbio $\xi(t)$ é um ruído branco estacionário com média zero e variância σ^2 e d representa um termo de *offset*.

O nosso problema é formulado como em [04] onde os parâmetros discretos g_i são obtidos a partir da minimização do índice de desempenho

$$I_t(G) = \frac{1}{2} \sum_{j=0}^t E(\tau^{t-j} [y(j+k) - r(j)]^2 + \alpha [u(j) - u(j-1)]^2) | j \quad (3.07)$$

onde α representa um fator de ponderação do esforço de controle e τ um fator de esquecimento ambos no intervalo $(0,1)$, E é o operador estatístico esperança. O índice inclui $y(t+k)$, desde que, tenhamos um atraso de k amostras, no tempo do sistema. Usando a linearidade do operador esperança (3.07) fica:

$$I_t(G) = \frac{1}{2} \sum_{j=0}^t E(\tau^{t-j} [r(j) + y(j+k)]^2 | j + \alpha [u(j) - u(j-1)]^2 | j) \quad (3.08)$$

Considerando que a esperança de uma função determinística é ela mesma e que a esperança de uma função não determinística é o seu valor estimado, a Equação (3.08) fica:

$$I_t(G) = \frac{1}{2} \sum_{j=0}^t (\tau^{t-j} [r(j) + \hat{y}(j+k|j)]^2 + \alpha [u(j) - u(j-1)]^2) \quad (3.09)$$

onde $\hat{y}(j+k|j)$ representa o valor estimado de $y(t)$ adiantado de k amostras calculado no tempo j .

Um procedimento geral de determinação de $\hat{y}(j+k|j)$ em relação a entradas e saídas relacionadas pelo modelo CARMA, visto em Aström [14], segue o

seguinte raciocínio. Considere a equação Diophantine [15]:

$$C = DA + z^{-k}F \quad (3.10)$$

onde:

$$D(z^{-1}) = 1 + d_1 z^{-1} + \dots + d_{k-1} z^{-(k-1)}$$

$$F(z^{-1}) = f_0 + f_1 z^{-1} + \dots + f_{n-1} z^{-(n-1)}$$

A Equação (3.06), adiantada de k amostras pode ser reescrita como:

$$y(t+k) = \frac{C}{A} \xi(t+k) + \frac{B}{A} u(t) + \frac{z^k}{A} d \quad (3.11)$$

Usando a Equação (3.10) na Equação (3.11) temos:

$$y(t+k) = D\xi(t+k) + \frac{F}{A} \xi(t) + \frac{B}{A} u(t) + \frac{z^k}{A} d \quad (3.12)$$

Usando $\xi(t)$ da Equação (3.06) e re-agrupando seus termos temos:

$$\begin{aligned} y(t+k) &= D\xi(t+k) + \frac{F}{A} y(t) + \\ &+ (C - z^{-k}F) \frac{B}{CA} u(t) + (C - z^{-1}F) \frac{z^k}{CA} d \end{aligned} \quad (3.13)$$

Usando novamente a Equação (3.10) na equação (3.13) temos:

$$y(t+k) = D\xi(t+k) + \frac{F}{C} y(t) + \frac{DB}{C} u(t) + \frac{Dz^k}{C} d \quad (3.14)$$

Chamando

$$T = \frac{F}{C}y(t) + \frac{DB}{C}u(t) + \frac{DZ^k}{C}d \quad (3.15)$$

e escrevendo

$$E[y(t+k) - \hat{y}]^2 = E[D\xi(t+k)]^2 + E[T - \hat{y}]^2 + 2E[D\xi(t+k) [T - \hat{y}]] \quad (3.16)$$

e usando o fato que o último termo da Equação (3.16) é zero porque $\xi(t+k), \xi(t+k-1), \dots, \xi(t+1)$ têm média zero e são independentes de $y(t), y(t-1), \dots$, o preditor que minimiza o erro de predição $(y(t+k) - \hat{y})$ é dado por

$$\hat{y}(t+k/t) = \frac{F}{C}y(t) + \frac{DB}{C}u(t) + \frac{Z^k Dd}{C} \quad (3.17)$$

Quando um controlador é um PID, a Equação (3.11), será:

$$\begin{aligned} \hat{y}(t+k/t) = & \frac{BD}{C}u(t-1) + \frac{BDG}{C}r(t) + \\ & \frac{F-BDG}{C}y(t) + \frac{Z^k Dd}{C} \end{aligned} \quad (3.18)$$

Na minimização de (3.07) para obter um valor estimado de G, G^* , utilizamos uma rotina recursiva cujos passos são desenvolvidos a seguir.

Vamos admitir que $G^*(t-1)$ são parâmetros ótimos do índice minimizado $I(t-1, G)$, que é obtido através de:

$$\frac{\delta I(t-1, G)}{\delta G} \Big|_{G^*(t-1)} = 0 \quad (3.19)$$

Assim para obter o algoritmo recursivo de $G^*(t)$ em função de $G^*(t-1)$, usaremos uma expansão de Taylor de segunda ordem de $I(t, G)$ sobre $G^*(t-1)$, obtendo:

$$\begin{aligned} I(t, G) &= I(t, G^*(t-1)) + \\ &\frac{\delta I(t, G)}{\delta G} \Big|_{G^*(t-1)} (G - G^*(t-1)) + \\ &\frac{1}{2} (G - G^*(t-1))^2 \frac{\delta^2 I(t, G)}{\delta G^2} \Big|_{G^*(t-1)} (G - G^*(t-1)) \quad (3.20) \\ &+ O((G - G^*(t-1))^3) \end{aligned}$$

onde,

$$\frac{\delta I}{\delta G} = \left(\frac{\delta I}{\delta g_0}, \frac{\delta I}{\delta g_1}, \frac{\delta I}{\delta g_2} \right)$$

e $O(x)$ representa os termos de ordem superior da expansão.

Derivando a Equação (3.20) para minimizar $I(t, G)$ em relação a G , após manipulação algébrica chegamos à seguinte expressão:

$$\begin{aligned} G^*(t) &= G^*(t-1) - \\ &\left[\frac{\delta^2 I(t, G)}{\delta G^2} \Big|_{G^*(t-1)} \right]^{-1} \frac{\delta I(t, G)}{\delta G} \Big|_{G^*(t-1)} \quad (3.21) \\ &+ O((G^*(t) - G^*(t-1))^2) \end{aligned}$$

que é a expressão recursiva básica para $G^*(t)$.

Para determinar as derivadas na Equação (3.21), note que o critério dado pelo índice de desempenho pode ser reescrito como:

$$I(t, G) = \frac{[x(t) - \hat{y}(t+k/t)]^2}{2} + \frac{\alpha}{2} [G'e(t)]^2 + \tau I(t-1, G) \quad (3.22)$$

Derivando este índice $I(t, G)$ em função de G temos:

$$\begin{aligned} \frac{\delta I(t, G)}{\delta G} &= e(t, G) \frac{\delta e(t, G)}{\delta G} \\ &+ \alpha G'e(t) \frac{\delta (G'e(t))}{\delta G} + \tau \frac{\delta I(t-1, G)}{\delta G} \end{aligned} \quad (3.23)$$

e

$$\begin{aligned} \frac{\delta^2 I(t, G)}{\delta G^2} &= e(t, G) \frac{\delta^2 e(t, G)}{\delta G^2} + \\ &\frac{\delta e(t, G)}{\delta G} \frac{\delta e(t, G)}{\delta G} + \alpha e(t) e(t)' + \\ &\tau \frac{\delta^2 I(t-1, G)}{\delta G^2} \end{aligned} \quad (3.24)$$

onde

$$e(t, G) = x(t) - \hat{y}(t+k/t) \quad (3.25)$$

A Equação (3.25) representa o erro de predição. Na Equação (3.24) podemos usar

$$\frac{\delta G'e(t)}{\delta G} = e(t) \quad (3.26)$$

Lembrando que $G^*(t-1)$ é ótimo, temos que:

$$\frac{\delta I(t-1, G)}{\delta G} \Big|_{G^*(t-1)} = 0 \quad (3.27)$$

e a primeira derivada de $G^*(t-1)$ pode ser reescrita como,

$$\begin{aligned} \frac{\delta I(t, G)}{\delta G} \Big|_{G^*(t-1)} = \\ \varepsilon(t, G^*(t-1)) \frac{\delta \varepsilon(t, G)}{\delta G} \Big|_{G^*(t-1)} \quad (3.28) \\ + \alpha G^*(t-1)' e(t) e(t) \end{aligned}$$

Assumindo que $G^*(t)$ tende para $G^*(t-1)$, é razoável assumir que,

$$O[G^*(t) - G^*(t-1)] \rightarrow 0 \quad (3.29)$$

$$\frac{\delta^2 I(t, G)}{\delta G^2} \Big|_{G^*(t-1)} \approx \frac{\delta^2 I(t, G)}{\delta G^2} \Big|_{G^*(t)} \quad (3.30)$$

Neste caminho, usando as Equações (3.28), (3.29) e (3.30) podemos reescrever a Equação (3.21) como,

$$\begin{aligned} G^*(t) = G^*(t-1) + \\ R(t)^{-1} [\varepsilon(t, G^*(t-1)) \phi(t, G^*(t-1)) \quad (3.31) \\ - \alpha G^*(t-1) e(t) e(t)'] \end{aligned}$$

onde por simplicidade denotamos,

$$\phi(t, G^*(t-1)) = - \frac{\delta \varepsilon(t, G)}{\delta G} \Big|_{G^*(t-1)} \quad (3.32)$$

$$R(t) = \frac{\delta^2 I(t, G)}{\delta G^2} \Big|_{G^*(t)} \quad (3.33)$$

Particularizando a Equação (3.24) em $G^*(t-1)$ e usando a Equação (3.30) uma recursão adicional é obtida para $R(t)$

$$\begin{aligned} R(t) = & \tau R(t-1) + \\ & \phi(t, G^*(t-1)) \phi(t, G^*(t-1))' + \\ & \alpha e(t) e(t)' - \varepsilon(t, G^*(t-1)) \frac{\delta \phi(t, G)}{\delta G} \Big|_{G^*(t-1)} \end{aligned} \quad (3.34)$$

Utilizando um lema da inversão de matriz [04] obtemos o algoritmo abaixo:

$$\begin{aligned} G^*(t) = & G^*(t-1) + P(t) [\phi(t) \varepsilon(t) - \\ & \alpha (G^*(t-1))' e(t) e(t)'] \end{aligned} \quad (3.35)$$

$$P(t) = K(t) - \frac{K(t) e(t) e(t)' K(t)}{e(t)' K(t) e(t) - (1/\alpha)} \quad (3.36)$$

$$\begin{aligned} K(t) = & [P(t-1) - \\ & \frac{P(t-1) \phi(t) \phi(t)' P(t-1)}{\phi(t)' P(t-1) \phi(t) + \tau}] (1/\tau) \end{aligned} \quad (3.37)$$

$$\varepsilon(t) = r(t) - \hat{y}(t+k/t) \quad (3.38)$$

$$C\phi(t) = b_0 e(t) \quad (3.39)$$

$$\begin{aligned} C\hat{y}(t+k/t) = & Fy(t) + BD[u(t-1) + \\ & G^*(t-1)' e(t)] + Dd \end{aligned} \quad (3.40)$$

Assim obtém-se o valor dos parâmetros de G^* em um tempo

t , mas não se pode garantir que estes valores de $G^*(t)$ satisfaçam as restrições, Equações (3.05a), (3.05b) e (3.05c). No caso disto ocorrer, devemos obter $\hat{G}$ através da minimização do índice:

$$I = \|G^* - \hat{G}\|^2 \quad (3.41)$$

que é facilmente resolvido através de um algoritmo de projeção baseado em técnicas de programação quadrática, desenvolvido na Seção 4.3.

3.2 Algoritmo de Estimação Paramétrica

Nesta seção vamos considerar a Equação (3.42) como modelo para estimação paramétrica usando o método dos mínimos quadrados, deduzido por Ljung [16].

$$y(t) = \theta'X(t) + \xi(t) \quad (3.42)$$

O vetor de parâmetros θ é estimado a partir dos dados medidos inseridos no vetor $X(t)$. A partir de um índice quadrático Ljung [16] deduz o algoritmo que tem a seguinte estrutura:

$$\begin{aligned} \theta(t) = \theta(t-1) + \frac{1}{t} R^{-1}(t) X(t) \alpha_t [y(t) - \\ \theta'(t-1) X(t)] \end{aligned} \quad (3.43a)$$

$$R(t) = R(t-1) + \frac{1}{t} [\alpha_t X(t) X'(t) - R(t-1)] \quad (3.43b)$$

onde $\hat{\theta}$ representa o valor estimado de θ .

Usando o lema de inversão utilizado no algoritmo do controlador auto-ajustável, as Equações (3.43a) e (3.43b) podem ser reescritas da seguinte forma:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t) [y(t) - \hat{\theta}'(t-1) X(t)] \quad (3.44a)$$

$$L(t) = \frac{P(t-1) X(t)}{1/\alpha_t + X'(t) P(t-1) X(t)} \quad (3.44b)$$

$$P(t) = P(t-1) + \frac{P(t-1) X(t) X'(t) P(t-1)}{1/\alpha_t + X'(t) P(t-1) X(t)} \quad (3.44c)$$

O método dos mínimos quadrados é um dos mais utilizados, e tendo em vista a estrutura que ele é apresentado pode ser utilizado com qualquer modelo. Quando o modelo usado é o CARMA, ou similar, temos:

$$\begin{aligned} \theta' &= [-a_1, \dots, -a_n, b_0, \dots, b_m, c_1, \dots, c_l] \\ X(t) &= [y(t-1), \dots, y(t-n), \\ &u(t-k), \dots, u(t-k-m), \xi(t-1), \dots, \xi(t-l)] \end{aligned} \quad (3.45)$$

o método utilizado é o método dos mínimos quadrados estendido ou o método de máxima verossimilhança

recursivo [17].

Existem muitas maneiras de se estimar o valor do parâmetro d , do modelo CARMA. Clarke [11] estima este parâmetro inserindo-o no algoritmo de estimação paramétrica, desta forma as Equações (3.45) ficam:

$$\begin{aligned}\theta' &= [-a_1, \dots, -a_n, b_0, \dots, b_m, c_1, \dots, c_l, d] \\ X(t) &= [y(t-1), \dots, y(t-n), u(t-k), \dots, \\ &\quad u(t-k-m), \xi(t-1), \dots, \xi(t-l), 1]\end{aligned}\quad (3.46)$$

A desvantagem deste método reside no fato de que se o sistema não possuir este parâmetro, no processo iterativo, haverá um erro de estimação nos outros parâmetros devido a sua presença.

Outra maneira de estimá-lo é considerar este parâmetro como parte integrante do ruído inerente ao sistema, como sugere Tuffs [11], desta forma se utiliza o método dado pelas Equações (3.45) e o parâmetro d não é considerado no equacionamento. Este procedimento permite uma maior exatidão na estimação dos demais parâmetros, quando este não existe, e na eventualidade de sua existência seu valor é diluído nos demais parâmetros.

CAPÍTULO 4

IMPLEMENTAÇÃO DOS ALGORITMOS

Os algoritmos foram implementados em primeiro lugar na forma de simulação, no ambiente de programação MATLAB [18], e em seguida em linguagem C, visando sua implementação prática. Tendo em vista a análise de um problema de cada vez, o processo de implementação foi dividido em fases.

Na primeira fase, o algoritmo a ser implementado foi o de mínimos quadrados estendido, que ajusta os parâmetros da planta. Nesta fase não houve preocupação em se controlar a planta, mas apenas o de implementar o algoritmo e identificar os parâmetros da planta, segundo o modelo matemático adotado, o CARMA. Desta forma o sistema foi trabalhado em malha aberta, utilizando como sinal de excitação uma onda quadrada.

Na segunda fase, também a nível de simulação, foi implementado o algoritmo de ajuste dos parâmetros do controlador, o auto-ajustável. Nesta fase os parâmetros da planta foram tidos como conhecidos e também não se visou analisar o desempenho do controlador, embora já se trabalhe em malha fechada usando como sinal de referência uma onda quadrada.

Em uma terceira fase, uniu-se as duas rotinas desenvolvidas em separado, gerando uma estrutura básica do controlador. Para melhor analisar os resultados utilizou-se, inicialmente, como sinal de referência uma onda quadrada, que realça melhor as características da planta. Posteriormente com a rotina melhor ajustada, foi utilizado uma onda sigmoidal para analisar o desempenho do controlador.

A última fase da simulação tratou da implementação do algoritmo que garanta a existência do controlador no tempo contínuo. Embora Vega [04] considere este item opcional, neste trabalho ele foi considerado importante, por se tratar de um trabalho de cunho acadêmico e permitir relacionar os parâmetros discretos com seus equivalentes no tempo contínuo. A técnica de projeção utilizada foi a de programação quadrática.

4.1 Estimação de Parâmetros

Os parâmetros são ajustados utilizando o algoritmo dos mínimos quadrados recursivo estendido, e a planta é modelada segundo o modelo CARMA:

$$\begin{aligned} A(z^{-1})y(t) = z^{-1}B(z^{-1})u(t) + \\ C(z^{-1})\xi(t) + d \end{aligned} \tag{4.01}$$

onde

$y(t)$ é o sinal de saída da planta

$u(t)$ é o sinal de entrada da planta

$\xi(t)$ é um ruído branco

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2}$$

O parâmetro d do modelo CARMA representa o nível de *offset* da planta e neste modelamento ele é considerado como parte integrante da figura de ruído $\xi(t)$, desta forma podemos reescrever a equação, do modelo CARMA, na forma matricial da seguinte maneira:

$$Y(t) = \theta' X(t) + \xi(t) \quad (4.02)$$

onde

$$\theta' = [-a_1 \ -a_2 \ -a_3 \ b_1 \ b_2 \ c_1 \ c_2] \quad (4.03)$$

$$X(t)' = [y(t-1) \ y(t-2) \ y(t-3) \ u(t-1) \ u(t-2) \ \xi(t-1) \ \xi(t-2)] \quad (4.04)$$

O ruído estimado $\hat{\xi}(t)$ é assumido ser igual ao erro de estimativa a menos de sua amplitude que é restrita a um intervalo apropriado, de -0.3 a 0.3. O algoritmo dos mínimos quadrados recursivo estendido tem a seguinte estrutura:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t) \hat{\epsilon}(t) \quad (4.05)$$

onde:

$$\hat{e}(t) = y(t) - \hat{\theta}(t-1)X(t) \quad (4.06)$$

representa o erro de estimativa,

$$K(t) = \frac{P(t-1)X(t)}{\beta + X(t)'P(t-1)X(t)} \quad (4.07)$$

com β um fator escalar de ganho e

$$P(t) = [I - K(t)X(t)']P(t-1) \quad (4.08)$$

Desta forma o algoritmo assume a seguinte forma:

- 01 - Define-se o valor inicial das variáveis e das constantes;
- 02 - Determina-se o valor da entrada da planta $u(t)$;
- 03 - Adiciona-se um ruído na saída da planta;
- 04 - Calcula-se o valor da saída $y(t)$ usando este ruído;
- Algoritmo dos Mínimos Quadrados Recursivos estendido;
 - i - Calcula-se o valor do erro de estimativa;
 - ii - Calcula-se o valor de $K(t)$;
 - iii - Calcula-se o valor de $\hat{\theta}(t)$;
 - iv - Salva-se os valores de $\hat{\theta}(t)$;
- 05 - Partindo do erro estima-se o ruído, verificando se o erro está dentro dos seus limites;
- 06 - Retorna-se a 02.

4.2 Auto-Ajuste do Controlador PID

A rotina implementada foi apresentada na Seção 3.2, e sua estrutura baseou-se naquela do ajuste dos parâmetros da planta apresentada na seção 4.1. Antes de implementar a rotina, propriamente dita, algumas de suas equações tiveram de ser reescritas e alguns valores deduzidos.

Partindo do modelo CARMA adotado, tem-se que a equação:

$$C = DA + Z^{-k}F \quad (4.09)$$

pode ser reescrita como

$$1 + C_1 Z^{-1} + C_2 Z^{-2} = 1 (1 + a_1 Z^{-1} + a_2 Z^{-2} + a_3 Z^{-3}) + Z^{-1} (f_0 + f_1 Z^{-1} + f_2 Z^{-2}) \quad (4.10)$$

Da equação (4.10) os valores das componentes do vetor F são:

$$f_0 = C_1 - a_1 \quad (4.11a)$$

$$f_1 = C_2 - a_2 \quad (4.11b)$$

$$f_2 = \quad - a_3 \quad (4.11c)$$

Da Equação (3.39)

$$C\phi(t) = b_0 e(t) \quad (4.12)$$

onde

$$e(t) = [e(t) \ e(t-1) \ e(t-2)] \quad (4.13)$$

tem-se que

$$(1 + c_1 z^{-1} + c_2 z^{-2})\phi(t) = b_0 e(t) \quad (4.14)$$

$$\phi(t) + c_1 \phi(t-1) + c_2 \phi(t-2) = b_0 e(t) \quad (4.15)$$

$$\phi(t) = b_0 e(t) - c_1 \phi(t-1) - c_2 \phi(t-2) \quad (4.16)$$

onde $\phi(t)$ é uma matriz 3×1 . Da equação:

$$\begin{aligned} C\hat{y}(t+1/t) &= Fy(t) + \\ &BD[u(t-1) + G^*(t-1)'e(t)] \end{aligned} \quad (4.17)$$

sabendo que $D=1$ esta equação fica:

$$\begin{aligned} C\hat{y}(t+1/t) &= Fy(t) + \\ &B[u(t-1) + G^*(t-1)'e(t)] \end{aligned} \quad (4.18)$$

por uma questão de conforto faz-se

$$u1(t) = u(t) + G^*(t-1)'e(t) \quad (4.19)$$

assim

$$C\hat{y}(t+1/t) = Fy(t) + Bu1(t) \quad (4.20)$$

$$\begin{aligned} (1 + c_1 z^{-1} + c_2 z^{-2})\hat{y}(t+1/t) &= \\ (f_0 + f_1 z^{-1} + f_2 z^{-2})y(t) + \\ (b_0 + b_1 z^{-1})u1(t) \end{aligned} \quad (4.21)$$

$$\begin{aligned} \hat{y}(t+1/t) = & f_0 y(t) + f_1 y(t-1) + f_2 y(t-2) + \\ & b_0 u_1(t) + b_1 u_1(t-1) - c_1 \hat{y}(t/t) - c_2 \hat{y}(t-1/t) \end{aligned} \quad (4.22)$$

ou na forma matricial

$$\hat{y}(t+1/t) = F'Y + B'A_1 - C'Y_t \quad (4.23)$$

onde

$$F' = [f_0 \ f_1 \ f_2]$$

$$B' = [b_0 \ b_1]$$

$$C' = [c_1 \ c_2]$$

$$Y' = [y(t) \ y(t-1) \ y(t-2)]$$

$$A_1' = [u_1(t) \ u_1(t-1)]$$

$$Y_t' = [\hat{y}(t|t) \ \hat{y}(t-1|t)]$$

Desta forma o algoritmo assume a seguinte forma:

- 01 - Define-se o valor inicial das variáveis e das constantes;
- 02 - Determina-se o valor da referência $r(t)$;
- 03 - Determina-se o valor da saída $y(t)$ e atualiza-se os vetores;
- Algoritmo de Ajuste dos Parâmetros do Controlador PID
 - i - Calcula-se o valor de $\hat{y}(t+1|t)$;
 - ii - Calcula-se o valor de $\epsilon(t)$;
 - iii - Calcula-se o valor de $\Phi(t)$;
 - iv - Calcula-se o valor de $K(t)$;

- v - Calcula-se o valor de $P(t)$;
- v_i - Calcula-se os novos valores dos parâmetros $G(t)$;
- 04 - Calcula-se o sinal de controle $u(t)$;
- 05 - Retorna-se a 2.

Neste estágio as variáveis tiveram seus valores iniciais fixados em zero, exceto as matrizes P e K que tiveram seus valores fixados em múltiplos da identidade.

4.3 Projeção de Parâmetros

Como foi discutido na seção 3.3, existe a possibilidade dos parâmetros G^+ não satisfazer as condições de existência no tempo contínuo ($K > 0$, $T_i > 0$ e $T_d > 0$), dadas pelas Equações (3.05a), (3.05b) e (3.05c).

Para isto os parâmetros estimados G^+ devem ser projetados dentro da região de existência, chamada região factível, usando o índice da Equação (3.41).

Para implementar o algoritmo foi tomado como base que apenas uma desigualdade é violada por vez, e que a projeção é feita em todos os ciclos que haja violação. Esta limitação foi usada tendo em vista que o objetivo deste algoritmo de projeção deve ter resultados em regime permanente, onde esta idéia de uma violação por vez é válida. Em regime transitório,

início do processo, ocorrem mais de uma violação por vez, mas se este problema fosse tomado com rigor o objetivo principal não seria obtido.

Desenvolvendo o índice tem-se:

$$I = \hat{g}_0^2 - 2g_0^* \hat{g}_0 + \hat{g}_1^2 - 2g_1^* \hat{g}_1 + \hat{g}_2^2 - 2g_2^* \hat{g}_2 + (g_0^{*2} + g_1^{*2} + g_2^{*2}) \quad (4.24)$$

Utilizando técnicas de programação quadrática [05], baseadas nos multiplicadores de Lagrange, constroeu-se:

$$L = I + \lambda \Lambda_n \quad (4.25)$$

onde

$$\Lambda_1 = -(\hat{g}_0 + \hat{g}_1 + \hat{g}_2) \leq 0 \quad (4.26a)$$

$$\Lambda_2 = \hat{g}_1 + 2\hat{g}_2 \leq 0 \quad (4.26b)$$

$$\Lambda_3 = -\hat{g}_2 \leq 0 \quad (4.26c)$$

Para solucionar o problema tem-se que:

$$\frac{\delta L}{\delta \hat{g}_0} = 0 ; \frac{\delta L}{\delta \hat{g}_1} = 0 ; \frac{\delta L}{\delta \hat{g}_2} = 0 ; \frac{\delta L}{\delta \lambda} = 0 \quad (4.27)$$

para cada caso. Assim, para o caso de violação de Λ_1 tem-se:

$$\frac{\delta L}{\delta \hat{g}_0} = 2\hat{g}_0 - 2g_0^* - \lambda = 0 \quad (4.28a)$$

$$\frac{\delta L}{\delta \hat{g}_1} = 2\hat{g}_1 - 2g_1^* - \lambda = 0 \quad (4.28b)$$

$$\frac{\delta L}{\delta \hat{g}_2} = 2\hat{g}_2 + 2g_2^* - \lambda = 0 \quad (4.28c)$$

$$\frac{\delta L}{\delta \lambda} = -(\hat{g}_0 + \hat{g}_1 + \hat{g}_2) = 0 \quad (4.28d)$$

ou na forma de matriz:

$$\begin{bmatrix} 2 & 0 & 0 & -1 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 2 & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 2g_0^* \\ 2g_1^* \\ 2g_2^* \\ 0 \end{bmatrix} \quad (4.29)$$

Tendo como solução

$$\begin{bmatrix} \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \lambda \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 2 & -1 & -1 & 2 \\ -1 & 2 & -1 & 2 \\ -1 & -1 & 2 & 2 \\ -2 & -2 & -2 & 4 \end{bmatrix} \begin{bmatrix} 2g_0^* \\ 2g_1^* \\ 2g_2^* \\ 0 \end{bmatrix} \quad (4.30)$$

Para o caso de violação de Λ_2 tem-se

$$\frac{\delta L}{\delta \hat{g}_0} = 2\hat{g}_0 - 2g_0^* = 0 \quad (4.31a)$$

$$\frac{\delta L}{\delta \hat{g}_1} = 2\hat{g}_1 - 2g_1^* - \lambda = 0 \quad (4.31b)$$

$$\frac{\delta L}{\delta \hat{g}_2} = 2\hat{g}_2 - 2g_2^* + 2\lambda = 0 \quad (4.31c)$$

$$\frac{\delta L}{\delta \lambda} = \hat{g}_1 + 2\hat{g}_2 = 0 \quad (4.31d)$$

ou na forma de matriz

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 2g_0^* \\ 2g_1^* \\ 2g_2^* \\ 0 \end{bmatrix} \quad (4.32)$$

tendo como solução

$$\begin{bmatrix} \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \lambda \end{bmatrix} = \frac{1}{10} \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 4 & -2 & 2 \\ 0 & -2 & 1 & 4 \\ 0 & 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 2g_0^* \\ 2g_1^* \\ 2g_2^* \\ 0 \end{bmatrix} \quad (4.33)$$

E para o caso de violação de Λ_3 tem-se

$$\frac{\delta L}{\delta \hat{g}_0} = 2\hat{g}_0 - 2g_0^* = 0 \quad (4.34a)$$

$$\frac{\delta L}{\delta \hat{g}_1} = 2\hat{g}_1 - 2g_1^* = 0 \quad (4.34b)$$

$$\frac{\delta L}{\delta \hat{g}_2} = 2\hat{g}_2 - 2g_2^* - \lambda = 0 \quad (4.34c)$$

$$\frac{\delta L}{\delta \lambda} = -\hat{g}_2 = 0 \quad (4.34d)$$

na forma de matriz

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 2g_0^* \\ 2g_1^* \\ 2g_2^* \\ 0 \end{bmatrix} \quad (4.35)$$

tendo como solução

$$\begin{bmatrix} \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \lambda \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2 & 4 \end{bmatrix} \begin{bmatrix} 2g_0^* \\ 2g_1^* \\ 2g_2^* \\ 0 \end{bmatrix} \quad (4.36)$$

4.4 Algoritmos de Estimação e Controle

Uma vez definidos os algoritmos, estes são associados. A primeira fase deste item deu-se com a união do algoritmo de estimação paramétrica com o algoritmo de ajuste dos coeficientes do controlador, sempre evitando a superposição de efeitos não estudados, e buscando a posição correta de cada parte do algoritmo. As constantes e os valores iniciais das variáveis foram acertados visando um controle eficiente e abrangente, de modo a se conseguir eficiência com diversos tipos de plantas sem haver necessidade de reajustar estas constantes e conseguir ao mesmo tempo um controle eficiente.

Em uma segunda fase foi acrescentado o algoritmo de projeção de parâmetros, visando atender, por completo, a proposta inicial de trabalho, atingindo a construção final do algoritmo, a nível de simulação. Assim o algoritmo adquiriu a seguinte forma:

- 01 - Define-se o valor inicial das constantes e variáveis;
- 02 - Calcula-se a referência;
- 03 - Calcula-se a saída da planta;
- 04 - Ajusta-se os parâmetros da planta;
- 05 - Estima-se o ruído da planta;

- 06 - Atualiza-se os vetores;
- 07 - Ajusta-se os parâmetros do controlador PID;
- 08 - Determina-se a necessidade de projetar os parâmetros da planta e se necessário fazê-la;
- 09 - Calcula-se o sinal de excitação da planta;
- 10 - Retorna-se a 2.

4.5 Implementação em Linguagem C

O algoritmo em C foi obtido a partir do algoritmo de simulação na sua versão final implementado no MATLAB. Sua reconfiguração não foi imediata pois a linguagem C não possui os mesmos recursos computacionais da utilizada no MATLAB.

O fato da simulação ser montada de forma organizada e estruturada, permitiu uma análise simples das operações e dos desdobramentos necessários. Isto foi possível devido ao fato da linguagem do MATLAB ser estruturada e ter muitos pontos em comum com a linguagem C, principalmente nos comandos condicionais. Por outro lado a linguagem C não possui em sua estrutura operadores matriciais, como os existentes no MATLAB, exigindo que estas operações fossem realizadas utilizando funções de repetições com a finalidade de se conseguir os mesmos resultados algébricos obtidos a nível de simulação.

Os operadores que foram construídos como a soma, subtração, multiplicação por um escalar e multiplicação matricial foram montados com auxílio de um comando condicional repetitivo. Não se gerou sub-rotinas para realizar cada operação em particular devido ao fato das matrizes não terem uma dimensão padrão, e a generalização demandar um custo operacional muito grande. Desta forma cada vez que uma operação era necessária, ela era realizada em particular, com seus parâmetros particulares. O operador em linguagem C utilizado, nestas operações, inicialmente, foi o comando FOR, que realiza um número fixo de iterações, possibilitando construir os operadores matriciais. Para implementar o algoritmo em sua versão final o comando FOR foi desdobrado e todas as operações realizadas através dele foram escritas passo a passo, devido ao alto consumo de tempo que este operador demanda.

CAPÍTULO 5

IMPLEMENTAÇÃO EM LABORATÓRIO

Neste capítulo é tratado a parte prática deste projeto: O acionamento da planta, neste caso um servo-motor CC, e a coleta de dados para realimentação do sistema. Os diagramas de blocos de controle apenas indicam como o sistema deve ser interligado, formando uma cadeia lógica. Na prática principalmente se pensarmos em plantas dotadas de potência e controladores residentes em micro-computadores que utilizam sinais extremamente pequenos devemos incluir interfaces de potência e acondicionamento de sinal.

Desta forma em trabalhos práticos o controlador não atua diretamente na planta, mas através de um sistema intermediário, interface de potência, que transforma o sinal, de baixa potência, em um sinal de potência adequado para a planta. A realimentação do sistema não se dá diretamente, devido ao fato dos sistemas não gerarem os sinais de forma adequada ao controlador, normalmente estes sinais ou são mecânicos ou elétricos incompatíveis com o controlador. Assim a comunicação entre a planta e o controlador se dá através de um conversor de sinal, que converte,

geralmente um sinal mecânico, rotação ou posição, em um sinal elétrico, normalmente tensão contínua. E o sistema de controle, prático, assume a forma representada na Fig.5.1.

Fig.5.1: Diagrama de blocos do sistema de controle completo.

5.1 Interface de Potência

A interface de potência tem como objetivo adequar o sinal de controle do controlador à planta. Normalmente as plantas não podem ser ativadas

diretamente pelo controlador, desta forma adota-se um sistema intermediário que executa esta tarefa. Outra função desta interface é gerar uma isolação física entre o controlador e a planta, evitando que falhas na planta danifiquem o controlador.

Esta interface não é padrão e pode variar conforme a necessidade e também com a conveniência. Para esta aplicação em particular adotou-se o sistema *PWM-CHOPPER*.

5.1.1 Sistema *PWM* Digital

O sistema *PWM*, Fig.5.2, foi projetado para interfacear o controlador com o sistema *CHOPPER*, e este em particular foi concebido inicialmente para trabalhar com um *CHOPPER* tiristorizado. Por esta razão além dos sinais usuais ao *PWM*, este possui um sinal de direção e dois sinais de controle manual do *CHOPPER*.

Sua função básica é gerar uma onda quadrada de frequência constante, mas com duração do nível alto variável. Para isto são utilizados um contador, um armazenador e um comparador.

A função do contador é gerar uma onda tipo rampa, que determina a frequência do *PWM*.

O armazenador tem função de resguardar o sinal de controle do controlador, enquanto este executa

Fig.5.2: Esquema do PWM.

outra tarefa.

O comparador tem função de gerar a onda quadrada, propriamente dita. O sinal do comparador é resultado da comparação do sinal do contador com o sinal do armazenador.

O funcionamento do *PWM* é bem simples e seus sinais bem confiáveis. Basicamente o sinal do contador é comparado com o armazenador, se o sinal do contador for menor ou igual ao do armazenador a saída do comparador é alto. Se o contrário ocorre, o sinal do contador é maior que o sinal do armazenador, a saída do comparador é baixa.

Para o sistema *CHOPPER* transistorizado o sinal do *PWM* pode ser utilizado diretamente mas para o sistema tiristorizado há necessidade de um estágio a mais, o disparador, que gera os pulsos para os tiristores, que foi colocado em uma placa interface *PWM-CHOPPER*. Este *PWM* em particular opera com uma palavra de 12 *bits* e como o microcomputador (controlador) opera com apenas 8 *bits* nos seus *slots* utiliza-se um procedimento que monta uma palavra de 16 *bits*.

Para uso exclusivo no *CHOPPER* tiristorizado existe a possibilidade de disparos manuais, visando testes de funcionamento do próprio sistema *CHOPPER*, que tem comutação forçada. E para ambos os *PWM*, utiliza-se um sinal de inversão de rotação de 1 *bit*, somando 15 *bits* e deixando 1 *bit* de reserva.

5.1.2 Sistema *CHOPPER*

O sistema *CHOPPER* é o último estágio da interface de potência e sua função é transformar o sinal de baixa potência do *PWM* para a potência adequada à planta. Seu funcionamento é não linear e a variação do sinal para planta se dá pelo nível médio do sinal. O *CHOPPER* é alimentado com uma tensão contínua e sua função é transformar esta tensão em uma onda quadrada

de frequência constante.

Este *CHOPPER* em particular foi projetado a partir de inversor de frequência trifásico, visando alimentar um motor *brushless*. E sua conversão para monofásico se deu pela eliminação de um estágio. Sua estrutura básica pode ser dividida em dois estágios, excitação e potência.

O estágio de excitação tem função de adequar o sinal do *PWM* e conter o circuito de isolamento físico e proteção, visando evitar que danos ocorridos no estágio de potência danifiquem o controlador. O sistema de proteção é feito por transformador de pulso.

O estágio de potência constitui a parte principal da interface de potência, sua função é agir diretamente na planta. Neste estágio está colocado o circuito *snubber* e sua função é reduzir as perdas de chaveamento e as sobretensões protegendo os componentes durante o processo de comutação. Sua estrutura elétrica é constituída por transistores que são chaveados dois a dois.

Foi implementado um sistema de proteção que evita que transistores alinhados entrem em condução no processo de inversão através da introdução de um atraso no envio do sinal para o novo par de transistores. Isto garante que a corrente residual provocada pela indutância do motor tenha nível suficientemente pequena

e que o transistor possa interrompê-la.

5.2 *Encoder Óptico*

A realimentação da malha de controle é feita por um sinal gerado a partir de um *encoder* óptico. Este *encoder* é constituído por um disco de 1000 posições e o sentido de rotação é determinado pelo primeiro giro do motor; se o primeiro movimento ocorrer no sentido horário este será considerado como sentido positivo.

O *encoder* está acoplado diretamente ao eixo do motor, na posição oposta de uso. Permitindo que este seja utilizado de forma prática. Suas dimensões são reduzidas e o seu atrito interno é totalmente desprezível, de tal forma que o *encoder* pode ser desprezado no modelo do motor em estudo.

Os componentes eletrônicos do *encoder* são alimentados pelo microcomputador que envia a tensão de alimentação através de uma placa de comunicação colocada diretamente no *slot* do microcomputador. Nesta placa está alocado um *chip* que tem a função de totalizar os pulsos emitidos pelo *encoder*, com uma capacidade de 16 *bits*.

CAPÍTULO 6

RESULTADOS DE SIMULAÇÃO

Neste capítulo serão apresentados os resultados de simulação, de todas as etapas desenvolvidas neste trabalho. Como já foi discutido anteriormente, cada nova etapa foi implementada separadamente com o objetivo de evitar superposição de efeitos.

A simulação foi realizada usando-se inicialmente o *software* MATLAB e posteriormente, visando a parte prática, utilizou-se a linguagem de programação C, nas suas versões Turbo C++ e Borland C++. Em MATLAB implementou-se o ajuste de parâmetros, o controlador PID sem ajuste de parâmetros, o controlador PID com ajuste de parâmetros e o controlador PID com ajuste de parâmetros e com projeção na região factível dos parâmetros do controlador PID. Em linguagem C implementou-se apenas a fase final da simulação em MATLAB: o controlador PID com ajuste de parâmetros e com projeção na região factível dos parâmetros do controlador PID.

6.1 Ajuste de Parâmetros

O ajuste de parâmetros implementado neste trabalho foi baseado no método dos mínimos quadrados estendido. O desenvolvimento do algoritmo visou obter o melhor resultado possível nesta etapa, sem se preocupar que os ajustes obtidos agora sejam preservados, ou alterados, nas etapas futuras. Baseado neste ponto de vista e levando em conta que o sinal de saída não é relevante e não será utilizado em nenhum sistema, o sinal de entrada utilizado foi uma onda quadrada com um sinal inicial pseudo aleatório que possui um grande número de harmônicos, excitando todos os estados da planta. Nesta seção apresentam-se resultados de simulação para dois casos. A ordem dos polinômios A, B e C do modelo CARMA são fixados em 3, 2 e 2 respectivamente

A primeira planta foi:

$$G(s) = \frac{-2.81s + 21.9}{s^2 + 3.22s + 4.39} \quad (6.01)$$

Discretizando (6.01) com um tempo de amostragem de 0.5s temos:

$$y(t) = \frac{z^{-1}(1 + 1.5z^{-1})}{1 - 0.7z^{-1} + 0.18z^{-2}} u(t) + \frac{1 - 0.49z^{-1}}{1 - 0.7z^{-1} + 0.18z^{-2}} \xi(t) \quad (6.02)$$

Utilizou-se $\beta = 1.0$ e $P = 10^9 I$, onde I representa a

matriz identidade. O valor inicial dos parâmetros foram adotados como sendo nulos. Para realizar esta simulação a planta foi excitada com um sinal de entrada e sua saída calculada, usando o modelo da Equação (6.02). Por se tratar de uma planta hipotética os resultados apresentados não possuem grandezas físicas. Obtendo como resultado as Figuras 6.01, 6.02, 6.03 e 6.04.

Na Figura 6.01 o sinal de entrada da planta no período de 0 a 100 amostras é representado por um sinal pseudo aleatório que tem o objetivo de acelerar o processo de convergência do algoritmo. No período de 101 à 1000 amostras o sinal de entrada é representado

Fig.6.01: Sinais de entrada, saída real e estimada (curvas sobrepostas).

por uma onda quadrada e tem o objetivo de verificar a exatidão do processo de convergência.

Nas Figuras 6.02, 6.03 e 6.04 são apresentados os parâmetros da planta, reais e estimados do modelo CARMA. Nestas figuras podemos verificar a convergência correta dos parâmetros do modelo: $a_1 = -0.7$, $a_2 = 0.18$, $a_3 = 0$, $b_0 = 1$, $b_1 = 1.5$, $c_1 = -0.49$ e $c_2 = 0$

Fig.6.02: Parâmetros a_1 , a_2 e a_3 .

Fig.6.03: Parâmetros b_0 e b_1 .

Fig.6.04: Parâmetros c_1 e c_2 .

A segunda planta testada foi o servo motor cc tendo como modelo a equação abaixo:

$$y(s) = \frac{b}{s + a} u(s) \quad (6.03)$$

$$\text{onde } a = \frac{fR_a + K_b K_a}{J_m R_a}, \quad b = \frac{K_a}{J_m R_a},$$

R_a representa a resistência de enrolamento, J_m o momento de inercia do motor, K_m a constante do conjugado do motor e K_b a constante da força contra eletromotriz. Discretizando (6.03) em relação ao tempo usando a bilinear com amostrador de ordem zero a equação fica:

$$y(t) = \frac{b}{a} \frac{(1 - e^{-aT}) z^{-1}}{1 - e^{-aT} z^{-1}} u(t) + \frac{C(z^{-1})}{1 - e^{-aT} z^{-1}} \xi(t) \quad (6.04)$$

utilizando os dados, levantados por Fernández [19] que utilizou o método de identificação descrito por Lord [20], calcula-se $a = 24.3978$, $b = 54.5466$ e os valores de β e P são os mesmos do teste anterior. O sinal de entrada $u(t)$ é dado em volts (V) e o sinal de saída $y(t)$ é dado em radianos por segundo (rad/s). Foi adotado um tempo de amostragem, $T = 5\text{ms}$, para se ter uma condição no limite de estabilidade do algoritmo, tendo uma visão clara do comportamento do algoritmo em relação a uma planta real. Adotando $C(z^{-1}) = 1$ obtemos os resultados dos Figuras 6.05, 6.06, 6.07 e 6.08.

Na Figura 6.05 é apresentado os sinais de entrada, saída real e saída estimada do motor. Neste teste foi utilizado as mesmas constantes e condições

iniciais do teste anterior. Assim pôde-se verificar a exatidão do modelo estimado.

Fig.6.05: Sinais de entrada e saída real e estimada (curvas sobrepostas).

De forma análoga ao teste anterior, as Figuras 6.06, 6.07 e 6.08 apresentam os parâmetros do modelo CARMA do motor: $a_1 = -0.8852$, $a_2 = 0$, $a_3 = 0$, $b_0 = 0.2568$, $b_1 = 0$, $c_1 = 0$ e $c_2 = 0$. Ao contrário do que aconteceu no teste anterior verificou-se que cada um dos parâmetros em particular não converge de forma tão exata para seus valores reais e sim para valores aproximados. Este fato não indica que o algoritmo seja falho, mas constata que ele tende a diluir o valor dos parâmetros em todos os parâmetros disponíveis no modelo utilizado, que apresenta uma ordem aumentada, procurando obter o menor erro possível.

Fig.6.06: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.07: Parâmetros b_0 e b_1 do motor.

Fig.6.08: Parâmetros c_1 e c_2 do motor.

6.2 Ajuste do controlador Sem Ajuste de Parâmetros

Nesta etapa foi implementado o algoritmo auto-ajustável (*self-tuning*) considerando os parâmetros da planta como conhecidos. O algoritmo utilizado foi apresentado, neste trabalho, na Seção 3.1 e sua implementação na seção 4.2.

Utilizou-se os valores de $\alpha = 1.25$ e $\tau = 1$. O sinal de referência utilizado tem a mesma forma de onda da Seção 6.1. Talvez usando uma referência mais suave se obteriam resultados melhores e mais adequados, contudo o objetivo de dividir o processo de

implementação do algoritmo em etapas seria seriamente prejudicado, uma vez que efeitos não desejados poderiam ser mascarados prejudicando as etapas futuras.

A primeira planta testada, Equação (6.02), deu como resultado as Figuras 6.09 e 6.10. Na Figura 6.09 é apresentado o comportamento dos parâmetros do controlador e na Figura 6.10 os sinais de referência, entrada da planta e saída da planta.

Fig.6.09: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.10: Sinais de entrada, saída e referência.

A segunda planta testada foi o servo motor cc, Equação (6.04), com os mesmos valores de α e P do teste anterior, dando como resultado os Figuras 6.11 e 6.12. Na Figura 6.11 pode ser visto o comportamento do controlador e na Figura 6.12 os sinais de entrada, saída e referência relativo ao motor CC.

Neste teste em particular, o controlador, apresentou um comportamento sub-amortecido. Uma razão para este comportamento pode ser atribuída aos parâmetros, α e τ , de ajuste dos parâmetros do controlador $G(z^{-1})$. O valor de α tem influência nos polos do sistema a malha fechada. Valores altos levam a um comportamento sobre-amortecido e valores pequenos a um comportamento sub-amortecido.

Fig.6.11: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.12: Sinais de entrada, saída e referência do motor.

6.3 Ajuste do controlador

Nesta etapa do trabalho o sistema de controle foi realizado tanto com ajuste dos parâmetros da planta como também com ajuste dos parâmetros do controlador. Os primeiros testes foram feitos apenas com a planta da Equação (6.02), obtendo como resultado as Figuras 6.13, 6.14, 6.15, 6.16 e 6.17.

Usou-se $\alpha = 2.5$, $\beta = 0.1$ e $\tau = 1$ com $P=I$, tanto para ajuste de parâmetros do controlador como para ajuste dos parâmetros da planta.

Com a inclusão de mais um processo de minimização, ajuste paramétrico, pode-se visualizar na Figura 6.13 que a resposta do sistema se tornou bem

Fig.6.13: Sinais de entrada, saída e referência.

mais abrupta tendendo a um processo oscilatório.

Em comparação aos resultados obtidos na primeira etapa deste trabalho, seção 6.1, pode-se nas Figuras 6.15, 6.16 e 6.17 verificar que o processo de convergência dos parâmetros da planta se tornou mais lento mas em contra partida a convergência dos parâmetros do controlador, como pode ser visto na Figura 6.14, tornou-se mais rápida em comparação aos da etapa anterior, Figura 6.09.

Fig.6.14: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.15: Parâmetros a_1 , a_2 e a_3 .

Fig.6.16: Parâmetros b_0 e b_1 .

Fig.6.17: Parâmetros c_1 e c_2 .

A segunda planta testada foi o servo motor CC, Equação 6.04, com os mesmos valores de α , θ , τ e P , para o ajuste dos parâmetros da planta e do controlador, do teste anterior, dando como resultado as Figuras 6.18, 6.19, 6.20, 6.21 e 6.22. Na Figura 6.18 podem ser vistos os sinais de entrada, saída e referência para o motor CC. Neste teste o controlador apresentou o mesmo comportamento da planta da equação 6.02 no teste anterior. As conclusões tiradas no teste anterior foram confirmadas e a mesma tendência de oscilação também foi verificada.

Fig.6.18: Sinais de entrada, saída e referência do motor

Da mesma forma os resultados obtidos para o ajuste dos parâmetros do controlador, Figura 6.19, e para o ajuste dos parâmetros do motor, Figuras 6.20, 6.21 e 6.22 tiveram comportamento semelhante ao teste da planta da Equação 6.02 no teste anterior e da mesma forma como ocorreu o ajuste de parâmetros nos testes anteriores, com o modelo do motor, os parâmetros estimados não se ajustam de forma exata aos valores dos parâmetros do modelo do motor. Quanto ao ajuste dos parâmetros do controlador este teve o mesmo comportamento do teste anterior.

Fig.6.19: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.20: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.21: Parâmetros b_0 e b_1 do motor.

Fig.6.22: Parâmetros c_1 e c_2 do motor.

O passo seguinte do desenvolvimento foi incorporar na rotina de controle o algoritmo que projeta os parâmetros do controlador dentro da região factível, permitindo que o controlador exista no tempo contínuo. Esta rotina foi apresentada na Seção 3.3, onde os parâmetros factíveis são apresentados em função dos parâmetros calculados. A verificação de violação é feita em todos os ciclos de controle, e quando esta é detectada os parâmetros do controlador são projetados dentro da região factível.

Considerando que esta é a última etapa de construção do algoritmo de controle e que todos os processos de ajuste estão agrupados, o objetivo agora é obter um desempenho adequado do sistema de controle. Tendo em vista este fato e considerando que a onda quadrada não existe em processos reais o sinal de referência usado a partir de agora passa a ser a onda sigmoidal. Outra alteração em relação ao sinal de referência usado nas etapas anteriores se encontra na utilização de um nível constante somado ao sinal.

Utilizando a planta da Equação (6.02), com parâmetros $\alpha = 0.8$, $\beta = 0.4$ e $\tau = 1.0$ e $P = 0.1$ I (parâmetros da planta) e $P = 10^5$ I (parâmetros do controlador) obtivemos os resultados das Figuras 6.23, 6.24, 6.25, 6.26, 6.27 e 6.28.

Na Figura 6.23, que apresenta os sinais de referência, entrada e saída da planta, pode ser visto que o desempenho do controlador teve o desempenho ótimo.

Fig.6.23: Sinais de entrada, saída e referência (curvas sobrepostas).

Nas Figuras 6.24, 6.25 e 6.26 são apresentados os parâmetros da planta e na Figura 6.27 os do controlador. Na Figura 6.28 pode-se observar que as restrições impostas ao controlador são respeitadas, lembrando que o objetivo é o regime permanente.

Fig.6.24: Parâmetros a_1 , a_2 e a_3 .

Fig.6.25: Parâmetros b_0 e b_1 .

Fig.6.26: Parâmetros c_1 e c_2 .

Fig.6.27: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.28: Restrições dos parâmetros do controlador.

Com os mesmos valores do teste anterior o algoritmo foi testado com o servo motor cc, planta da Equação (6.04), obtendo como resultado as Figuras 6.29, 6.30, 6.31, 6.32, 6.33 e 6.34.

Em comparação com os resultados planta da Equação (6.02), pode-se observar, na Figura da 6.29, que o desempenho do sistema se manteve, mas ocorreu uma piora na convergência dos parâmetros da planta, como pode ser visto nas Figuras 6.30, 6.31 e 6.32. Em relação aos parâmetros do controlador, Figura 6.33, o mesmo teve desempenho semelhante ao da planta da Equação (6.02), podendo ser dito o mesmo das restrições, Figura 6.34.

Fig.6.29: Sinais de entrada, saída e referência do motor.

Fig.6.30: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.31: Parâmetros b_0 , b_1 do motor.

Fig.6.32: Parâmetros c_1 e c_2 do motor.

Fig. 6.33: Parâmetros g_0 , g_1 e g_2 .

Fig. 6.34: Restrições do controlador

Utilizando as mesmas constantes e condições iniciais utilizadas anteriormente com o modelo do motor, realizou-se o teste de efeito de carga, através do seguinte modelo:

$$y(t) = \frac{1}{a} \frac{(1 - e^{-aT})z^{-1}}{1 - e^{-aT}z^{-1}} [bu(t) - \frac{1}{J_m}T_L(t)] + \frac{C(z^{-1})}{1 - e^{-aT}z^{-1}}\xi(t)$$

onde T_L é esforço conjugado aplicado no motor, J_m é o momento de inércia do motor. Alterando apenas a matriz de minimização de 10^5 para 10^2 obteve-se os resultados das Figuras 6.35, 6.36, 6.37, 6.38, 6.39 e 6.40.

Na Figura 6.35 é mostrado o ponto em que se aplica o esforço T_L e o seu efeito no sinal de entrada da planta. Nesta mesma figura é mostrado que o desempenho do sistema se manteve mesmo com a introdução do esforço.

Fig.6.35: Sinais de entrada, saída, referência e esforço conjugado(100xNm) do motor.

Nas Figuras 6.36, 6.37 e 6.38 são apresentados os parâmetros do motor com efeito de carga. Nestas figuras podem ser visualizados o efeito do esforço nos parâmetros do motor. O efeito de carga é visto pelo programa de ajuste dos parâmetros do motor como uma variação dos parâmetros do motor. Desta forma quando o esforço é introduzido os parâmetros são reajustados para representar a nova planta. Quando o esforço é retirado os parâmetros estimados não retornam aos valores iniciais, pois o controlador adequou-se à nova situação.

Fig.6.36: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.37: Parâmetros b_0 e b_1 do motor.

Fig.6.38: Parâmetros c_1 e c_2 do motor.

Nas Figuras 6.39 e 6.40 são mostrados os parâmetros do controlador e as restrições dos parâmetros do controlador. A alteração dos parâmetros do controlador e as restrições dos parâmetros do controlador se dam em resposta às variações da carga. A alteração dos parâmetros do controlador se dá de forma mais lenta que o dos parâmetros do modelo do motor.

Fig.6.39: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.40: Retrições controlador.

O controle de posição foi realizado através do controle de velocidade. Entrando com uma referência de posição, ela é derivada numericamente, gerando uma referência de velocidade, que é usada pelo programa. A saída do motor, no caso a velocidade, é integrada numericamente gerando a saída de posição, para efeito de armazenamento de resultados.

Para esta simulação usou-se o modelo da Equação (6.04) e os valores de $\alpha = 0.8$, $\beta = 0.4$, $\tau = 1.0$, e $P = I$ (parâmetros do motor) e $P = 0.1I$ (parâmetros do controlador), obtendo como resultados as Figuras 6.41, 6.42, 6.43, 6.44, 6.45, 6.46e 6.47.

Na Figura 6.41 são visualizados os sinais de entrada, saída e referência da planta de velocidade.

Nesta Figura pode ser visualizado que a saída acompanhou a referência de velocidade, apresentando apenas um pequeno atraso que não interferiu no processo de seguimento da trajetória de posição. Os picos dos sinais de referência bem como os de saída apresentam os mesmos níveis. E no final de cada ciclo, onde a referência é zerada, o sinal de saída apresenta uma pequena oscilação, quase imperceptível.

Fig.6.41: Sinais de entrada, saída e referência do motor (velocidade).

Na Figura 6.42 é apresentado o sinal de saída e o de referência de posição. Como o sinal de saída e referência de posição são as integrais dos sinais de saída e referência de velocidade, nesta figura não aparece o sinal de entrada, que é igual para ambos os casos aparecendo apenas na Figura 6.41. Como o operador integral é um filtro, pequenas falhas no sinal de velocidade, oscilações, ruídos e pequenas perturbações, não aparecem no sinal de saída de posição. Desta forma o controle de posição se mostrou preciso e adequado.

Fig.6.42: Sinal de saída e referência do motor (posição).

Nas Figuras 6.43, 6.44 e 6.45 são apresentados os parâmetros do motor, estimados e reais. De modo análogo ao ocorrido nos testes anteriores com o motor, os parâmetros estimados não convergem para os valores calculados e da mesma forma, como ocorreu anteriormante, não houve prejuízo no desempenho do controlador e do sistema que utilizou estes valores no ajuste do controlador.

Fig.3.43: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.44: Parâmetros b_0 e b_1 do motor.

Fig.6.45: Parâmetros c_1 e c_2 do motor.

Nas Figuras 6.46 e 6.47 podem ser vistos os parâmetros do controlador e suas restrições. Neste ensaio ocorreu, devido a atuação das restrições, que os valores de g_1 e g_2 do controlador tendem para zero. Se as restrições não fossem utilizadas os parâmetros g_0 , g_1 e g_2 do controlador assumiriam valores distintos e não nulos. Mas, por outro lado, ocorreria uma piora na convergência dos algoritmos, e no tempo contínuo o termo T_d assumiria valores negativos, não tendo um significado físico.

Fig.6.46: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.47: Retrições do controlador.

6.4 Simulação em Linguagem C

Antes de implementar o algoritmo de controle na versão final, visando realizar os testes práticos, criou-se um estágio intermediário que tem o objetivo de verificar se a mudança do sistema de cálculos, passagem do MATLAB para C, afeta a estabilidade do algoritmo. Outra preocupação foi verificar se o tempo de processamento, que neste estágio é praticamente igual ao tempo de amostragem, é compatível para implementação em laboratório.

Em uma primeira fase o algoritmo do MATLAB foi transformado diretamente para linguagem C, mantendo totalmente a sua estrutura lógica e sistemas de

armazenamento de dados que são alocados diretamente em arquivos localizados no disco rígido do computador.

Os parâmetros de ajuste para esta sequência foram adotados como sendo $\alpha = 0.8$, $\beta = 0.4$ e $\tau = 1.0$. A condição inicial da matriz P dos parâmetros do controlador foi adotada como sendo $10^3 I$ e a matriz de minimização dos parâmetros da planta como sendo $10^5 I$.

A primeira planta testada, planta da Equação (6.02), teve os seus resultados mostrados nas Figuras 6.48, 6.49, 6.50, 6.51 e 6.52. Na Figura 6.48 pode ser visto os sinais de entrada, saída e referência da planta, na Figura 6.49 os parâmetros do controlador e nas Figuras 6.50, 6.51 e 6.52 os parâmetros da planta.

Fig.6.48: Sinais de entrada, saída e referência.

Fig.6.49: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.50: Parâmetros a_1 , a_2 e a_3 .

Fig.6.51: Parâmetros b_0 e b_1 .

Fig.6.52: Parâmetros c_1 e c_2 .

Com o motor, planta da Equação (6.04), foram realizados testes de controle de velocidade com os resultados mostrados nos Figuras 6.53, 6.54, 6.55, 6.56 e 6.57. Na Figura 6.53 é apresentado os sinais de entrada, saída e referência do motor, na Figura 6.54 os parâmetros do controlador e nas Figuras 6.55, 6.56 e 6.57 os parâmetros do motor.

Fig.6.53: Sinais de entrada, saída e referência do motor.

Fig.6.54: Parâmetros g_0 , g_1 e g_2 do controlador.

Fig.6.55: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.56: Parâmetros b_0 e b_1 do motor.

Fig.6.57: Parâmetros c_1 e c_2 do motor.

O controle de posição, através do controle de velocidade, com os resultados mostrados nos Figuras 6.58, 6.59, 6.60, 6.61, 6.62 e 6.63. Na Figura 6.58 pode ser visto o sinal de saída e referência de posição, na Figura 6.59 os sinais de entrada, saída e referência de velocidade e nas Figuras 6.61, 6.62 e 6.63 os parâmetros do motor.

Fig.6.58: Sinal de saída e referência do motor (posição).

Fig.6.59: Sinais de entrada, saída e referência do motor (velocidade).

Fig.6.60: Parâmetros g_0 , g_1 e g_2 .

Fig.6.61: Parâmetros a_1 , a_2 e a_3 do motor.

Fig.6.62: Parâmetros b_0 e b_1 do motor.

Fig.6.63: Parâmetros c_1 e c_2 do motor.

CAPÍTULO 7

RESULTADOS EXPERIMENTAIS

Neste capítulo são apresentados os resultados práticos deste trabalho. Para obtê-los foi utilizado o equipamento descrito no capítulo 5. A planta utilizada foi um servo-motor cc, o qual é representado por um modelo CARMA .

Ao contrário da simulação, nesta fase do trabalho implementou-se apenas a última etapa, ou seja, o controlador completo, por se considerar esta etapa a mais importante do trabalho e também por se julgar desnecessário a implementação das etapas intermediárias.

Para evitar que falhas do equipamento interferissem na estabilidade do controlador e no seu desempenho foi adicionado ao programa uma rotina com a finalidade de evitar que erros de contagem realizados pelo encoder sejam considerados. Assim, toda vez que a leitura do *encoder*, em um ciclo, fosse maior que uma volta, esta leitura seria ignorada e considerada a leitura anterior.

Outro problema encontrado foi em relação a coleta de dados, uma vez que o compilador C reserva uma

área de 64Kbytes para armazenar dados, não sendo possível coletar todos os dados, num mesmo ensaio, como foi realizado na simulação. Desta forma os ensaios foram repetidos para a coleta dos dados necessários.

7.1 Controle de Velocidade

Nesta seção serão apresentados os resultados relativos ao controle de velocidade. Os parâmetros utilizados foram ajustados da seguinte forma: $\alpha = 0.8$, $\beta = 0.4$ e $\tau = 1.0$, $P = I$ e $P = 0.001I$ no algoritmo de estimação de parâmetros e do controlador, respectivamente. Tendo como resultado as Figuras 7.01, 7.02, 7.03, 7.04 e 7.05.

Na Figura 7.01 é mostrado os sinais de entrada, saída e referência do motor, nas Figura 7.02, 7.03 e 7.04 os parâmetros do motor e na Figura 7.05 os parâmetros do controlador.

Como resultado deste teste obteve-se os seguintes resultados para um tempo de amostragem = 20.7 ms:

Parâmetros do controlador:

$$g_0 = 0.016291$$

$$g_1 = 0$$

$$g_2 = 0$$

Parâmetros do motor:

$$a_1 = -0.999979 \quad b_0 = 0.000115 \quad c_1 = 0.000161$$

$$a_2 = -0.000013 \quad b_1 = -0.000128 \quad c_2 = 0.001547$$

$$a_3 = -0.000015$$

Fig.7.01: Sinais de entrada, saída e referência.

Fig.7.02: Parâmetros a_1 , a_2 e a_3 .

Fig.7.03: Parâmetros b_0 e b_1 .

Fig.7.04: Parâmetros c_1 e c_2 .

Fig.7.05: Parâmetros g_0 , g_1 e g_2 .

7.2 Controle de Posição

Da mesma forma que foi realizado a nível de simulação, o controle de posição é realizado através do controle de velocidade. A referência de posição é derivada gerando a referência de velocidade que é utilizada no algoritmo de controle. A saída de posição é a integral da velocidade.

Para realizar o controle de posição apenas a referência foi alterada, todos os outros parâmetros foram mantidos iguais ao controle de velocidade.

Nesta seção a Figura 7.06 mostra a referência e a saída de posição, a Figura 7.07 mostra referência de velocidade, a saída de velocidade e o sinal de entrada do motor. Obteve-se os seguintes resultados, com um tempo de amostragem = 20.7 ms:

Parâmetros do controlador:

$$g_0 = 0.022877$$

$$g_1 = 0$$

$$g_2 = 0$$

Parâmetros do motor:

$$a_1 = -0.999897 \quad b_0 = -0.000112 \quad c_1 = 0.000229$$

$$a_2 = -0.000121 \quad b_1 = -0.000121 \quad c_2 = 0.000181$$

$$a_3 = 0.000010$$

Fig.7.06: Sinal de saída e referência de posição.

Fig.7.07: Sinais de entrada, saída e referência de velocidade.

O teste de controle de posição foi repetido usando um outro tipo de referência. O objetivo deste experimento foi testar o sistema de controle com uma referência mais complexa e verificar o comportamento do controlador quando a referência, no caso a de velocidade, apresenta amplitudes distintas durante o teste.

A Figura 7.08 mostra a referência e a saída de posição, a Figura 7.09 mostra referência de velocidade, a saída de velocidade e o sinal de entrada do motor. Obteve-se os seguintes resultados, com um tempo de amostragem = 20.7 ms:

Parâmetros do controlador:

$$g_0 = 0.017199$$

$$g_1 = 0$$

$$g_2 = 0$$

Parâmetros do motor:

$$a_1 = -0.999848 \quad b_0 = -0.000713 \quad c_1 = 0.000149$$

$$a_2 = -0.000198 \quad b_1 = -0.000720 \quad c_2 = 0.000160$$

$$a_3 = 0.000045$$

Fig.7.08: Sinal de saída e referência de posição

Fig.7.09: Sinais de entrada, saída e referência de velocidade.

7.3 Variação paramétrica

Nesta seção serão apresentados os resultados de teste de variação de parâmetros do motor. Para realizar estes testes foi simulado uma variação de parâmetros colocando-se, ou removendo-se, um resistor de 6.8Ω na entrada do motor através de uma chave de comutação, simulando assim um aumento, ou diminuição, da resistência da bobina de campo do motor, que representa uma variação de 100% da mesma.

Em um primeiro teste o motor foi ligado com o resistor ativado e na sequência, quando não existia movimento no motor, este foi desativado.

Na Figura 7.10 é mostrado o resultado deste primeiro teste, podendo-se observar um pequeno aumento na tensão de alimentação do motor ocasionado pelo aumento da resistência do enrolamento do motor. Também pode-se observar que o desempenho do controlador se manteve durante todo o teste.

Fig.7.10: Sinais de entrada, saída e referência de velocidade do motor com variação paramétrica.

O teste seguinte simula uma redução da resistência de enrolamento. Com base neste procedimento foram colhidos dados sobre o comportamento dos parâmetros do motor, bem como do controlador.

Na Figura 7.11 é mostrado os sinais de entrada, saída e referência do motor com variação de parâmetros, onde pode-se visualizar uma pequena redução da tensão de alimentação do motor. Na Figura 7.12 é mostrado os parâmetros do controlador e nas Figuras 7.13, 7.14 e 7.15 os parâmetros do motor.

Fig.7.11: Sinais de entrada, saída e referência de velocidade do motor com variação paramétrica.

Fig.7.12: Parâmetros g_0 , g_1 e g_2 do controlador com variação paramétrica.

Fig.7.13: Parâmetros a_1 , a_2 e a_3 do motor, com variação paramétrica.

Fig.7.14: Parâmetros b_0 e b_1 do motor com variação paramétrica.

Fig.7.15: Parâmetros c_1 e c_2 do motor com variação paramétrica.

7.4 Variação de Carga

Nesta seção serão apresentados os resultados de teste da variação de carga do motor. Para realizar este teste o eixo do motor foi freado com auxílio de um cano de PVC.

Em um primeiro teste, controle de velocidade, o motor foi freado no primeiro ciclo e mantido até o final do teste.

Na Figura 7.16 é mostrado o resultado deste primeiro teste, podendo-se observar um pequeno aumento da tensão de alimentação do motor ocasionado pelo efeito de carga exercido pelo freio. Também pode-se observar que o desempenho do controlador se manteve durante o teste.

Fig.7.16: Sinais de entrada, saída e referência de velocidade do motor com variação de carga.

No teste seguinte, controle de posição através do controle de velocidade, o motor foi freado no segundo intervalo, com o motor parado, e mantido até o início do quarto ciclo de controle, onde o motor retorna para a posição inicial.

Na Figura 7.17 é mostrado os sinais de referência e saída de posição e na Figura 7.18 os sinais de entrada, saída e referência de velocidade. Pode-se observar, na Figura 7.18, um pequeno aumento da tensão de alimentação do motor ocasionado pelo efeito de carga exercido pelo freio e da mesma forma que o primeiro teste desta seção o desempenho do controlador se manteve durante o teste.

Fig.7.17: Sinal de saída e referência de posição com variação de carga.

Fig.7.18: Sinais de entrada, saída e referência de velocidade do motor com variação de carga.

CAPÍTULO 8

CONCLUSÃO

A nível de simulação pôde-se observar que o ajuste de parâmetros, individualmente, necessita de um sinal de excitação, para planta, com um grande número de harmônicos para se ter uma resposta rápida e precisa. Devido a isto utilizou-se a onda quadrada precedida de um sinal pseudo aleatório de pequeno nível.

A simulação em MATLAB mostrou também que a inclusão de novos sistemas de ajuste, tais como ajuste de parâmetros da planta e projeção dos parâmetros do controlador, na região factível, permitindo sua existência no tempo contínuo, melhora o desempenho global do algoritmo completo bem como o de cada algoritmo que compõe o sistema de controle.

Os parâmetros da planta relativo ao ruído, parâmetros c_1 e c_2 , foram estimados de forma aceitável apenas após o uso do recurso que limita o sinal de erro, que é utilizado como ruído estimado, no intervalo de -0.3 à 0.3 . Este recurso permitiu também uma aceleração do ajuste dos parâmetros, tanto da planta como do controlador.

Ao se transcrever o programa da linguagem do

MATLAB para linguagem C pode-se observar que o desempenho do algoritmo sofreu uma melhora extremamente significativa na convergência dos parâmetros. A razão deste fato não pode ser identificada com certeza, mas a causa mais provável se encontra no fato do programa MATLAB, que é feito em linguagem C, ter um número de interações muito maior e também pelo fato do programa em MATLAB, ser compilado novamente em todos os ciclos, o que não ocorre com a linguagem C que é compilado apenas uma vez.

A simulação em linguagem C foi utilizada para remover todos os possíveis erros de lógica que poderiam ser levados para o algoritmo que seria utilizado no trabalho prático. Desta forma pode-se observar que o uso dos recursos computacionais tais como comando FOR e também o salvamento de dados diretamente para um arquivo tipo texto durante a execução do programa é inadequado e consomem um tempo de processamento extremamente longo, exigindo que estes recursos não fossem utilizados. Em consequência disto devido a limitação da área de memória reservada para variáveis, apenas 64K *bytes*, os dados foram salvos em testes distintos tendo o objetivo de formar um conjunto completo de dados como sinais do motor, parâmetros do motor e do controlador.

Ao se implementar o algoritmo em laboratório verificou-se que os valores de P utilizados na simulação não foram adequados. As matrizes P de ajuste que foram definidas com um nível alto, na simulação, tiveram que ser substituídas por valores extremamente baixos para que o algoritmo viesse a funcionar.

Os resultados práticos mostraram-se bastante aceitáveis, embora tenham tido uma piora em relação aos simulados. O uso de um número maior de parâmetros no modelo *CARMA* da planta, visando abranger condições não previstas no projeto inicial, do que o necessário não interferiu no desempenho do controlador, mas estes são ligeiramente diferentes dos parâmetros reais da planta utilizada.

Bibliografia

- [01] C. J. Harris, S. A. Billings: *Self-Tuning and Adaptive Control: Theory and Applications*, Peter Peregrinus Ltd., 1985
- [02] V. V. Chalam: *Adaptive Control Systems: "Techniques and Applications*, Marcel Dekker, Inc., 1987.
- [03] E. Toulsson, L. Ljung: "Adaptive Control Based on Explicit Criterium Minimization", *Automatica*, vol. 21, n° 4, pp.385-398, 1985.
- [04] P. Vega, C. Prada, V. Alexandre: "Self-Tuning Predictive PID Controller", *IEE Proceedings-D*, Vol 138, n° 3, pp.303-311, may 1991.
- [05] M. S. Bazaraa, C. M. Shetty: *Nonlinear Programming*, New York, Wiley, 1979.
- [06] B. Wittenmark, K. J. Aström: "Self-Tuning PID Controllers Based on Pole Placed", Report LUTFD2/(TFRT-7179)/1-037/(1979). Lund Institute of

Technology, 1979.

- [07] K. J. Aström, T. Hängglund: "Automatic Tuning of Simple Regulators", Proc 9th IFAC World Congress on Automatic Control, Budapest, Hungary, 1984.
- [08] J. G. Ziegler, N. B. Nichols: "Optimum Settings for Automatic Controllers", Trans. ASME, 65, pp.433-444, 1943.
- [09] J. K. Aström, T. Hängglund: "Automatic Tuning of Simple Regulators With Specifications on Phase and Amplitude Margins", Automatica, Vol 20, n° 5, pp.645-652, 1984.
- [10] J. K. Aström, T. Hängglund: "A Frequency Domain Approach to Analysis and Design of Simple Feedback Loops", Proceedings 23rd IEEE Conference on Decision and Control, Las Vegas, 1984.
- [11] D. W. Clarke, P. J. Gawthrop: "Self-Tuning Controller", Proc. IEE, Vol 122, n° 9, pp.929-934, september 1975.
- [12] P. J. Gawthrop: "Self-Tuning PID controllers: Algorithms and Implementation", IEEE Transactions

on Automatic Control, Vol AC-31, n° 3, pp.201-209, mach 1986.

[13] P. S. Tuffs, D. W. Clarke: "Self-Tuning Control of offset: a Unified approach", IEE Proceedings, vol. 132, Pt.D., n° 3, pp.100-132, may 1985.

[14] K. J. Aström, B. Wittenmark: *Computer Controlled System Theory and Design*, Pretince Hall, 1984.

[15] M. J. Grimble, M. Johnson: *Optimal Control and Stochastic Estimation Theory and Applications*, vol. 2, Wiley-Inters Cience Publication, 1988.

[16] L. Ljung, T. Soderstrom: *Theory and Practice of Recursive Identification*, MIT Press, 1983.

[17] J. T. Moir: "Frequency-Domain Aproach to State-Space LQG Self-Tuning Control", IEE Proceedings-D, vol. 138 n° 4, pp.372-380, july 1991.

[18] The Math Works Inc., Matlab Users Gride, Matlab the Math Works Inc., 1994.

[19] R. G. V. Fernández: "Algoritmo da Variância Mínima para Controle Auto-Ajustável de Servomotor CC",

Dissertação de Mestrado apresentada na EESC, 1992.

- [20] W. Lord, J. H. Hwang: "DC Servomotors - Modelling and Parameter Determination", IEEE Transactions on Industry Applications, vol. 1a-13, nº 3, pp.234-243, may/june 1977.