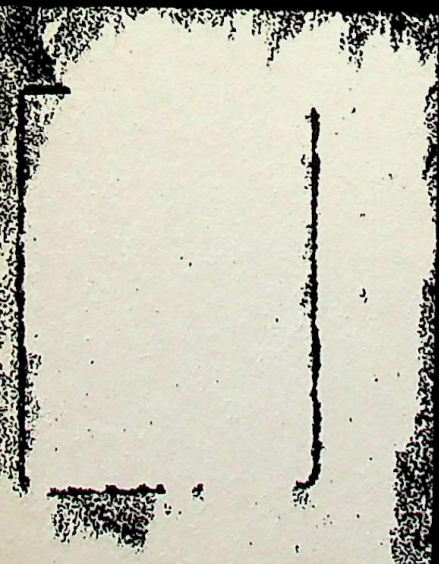
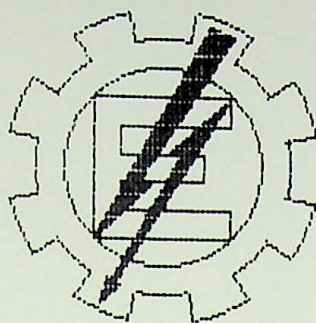


TESE

1194

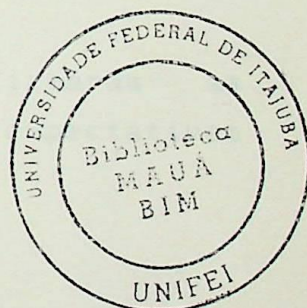




ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

DISSERTAÇÃO DE MESTRADO

IMPLEMENTAÇÃO DE SISTEMAS DE CONTROLE AUTO-AJUSTÁVEIS
E UMA APLICAÇÃO NO CONTROLE DE VELOCIDADE DE
MOTORES DE CORRENTE CONTÍNUA



Mestrando : José Vantuil Lemos Pinto
Orientador: Prof. Luiz Eduardo Borges da Silva

Itajubá, março de 1992

CLASS. 62-52(042.2)
COTTER. P659i
TOMBO. 1194

ESCOLA FEDERAL DE ENGENHARIA
DE ITAJUBA



INSTITUO DE PESQUISA

IMPLEMENTAÇÃO DE SISTEMAS DE CONTROLE AUTO-ALIMENTADOS
E SUA APLICAÇÃO NO CONTROLE DE VELOCIDADE DE
MOTORES DE CORRENTE CONTINUA



Coordenador: José Vander Leão Filho
Orientador: Prof. Luiz Eduardo Borges da Silva

RESUMO

Este trabalho propõe, e apresenta a implementação prática de um sistema de controle auto-ajustável para o controle de velocidade de um motor de corrente contínua, acionado por conversor tiristorizado.

A filosofia do controle consiste em deslocar os pólos da função de transferência de malha fechada do sistema para posições convenientes, a fim de obter um comportamento transitório específico, independentemente das variações que possam vir a ocorrer nos parâmetros do processo sob controle.

O ajuste global da malha de controle, dá-se pela modificação do sinal de referência. A identificação dos parâmetros do processo são obtidas pelo algoritmo dos mínimos quadrados recursivo.

Os resultados práticos obtidos confirmaram as simulações efetuadas, e satisfizeram plenamente as expectativas teóricas.

ABSTRACT

This work describes a practical implementation of a Self-Tuning Control System for speed control of a dc motor.

The control strategy is based on the idea of moving the closed-loop system poles to a specified position in order to get a required transient response, independently of parameter variations.

The control loop strategy consist in modifying the reference signal of the system. The identification of the process parameters is done by for Recursive Least Squares Algorithm.

The practical results obtained agree with simulations and satisfy the theory expectation.

INDICE

CAPÍTULO	1	INTRODUÇÃO	1
CAPÍTULO	2	FORMA CANÔNICA CONTROLÁVEL	6
	2.1	Introdução	6
	2.2	Representação na Forma Canônica Controlável	8
CAPÍTULO	3	ALOCAÇÃO DE PÓLOS	13
	3.1	Introdução	13
	3.2	Método de Alocação de Pólos	14
CAPÍTULO	4	ESTIMADOR DE PARÂMETROS	21
	4.1	Introdução	21
	4.2	Generalização do Método de Identificação de Parâmetros	22
	4.3	Método Recursivo de Estimação	32
	4.4	Algoritmo Recursivo de Estimação de Parâmetros	39
CAPÍTULO	5	SIMULAÇÕES	42
	5.1	Introdução	42
	5.2	Simulação do Algoritmo de Estimação de Parâmetros	42
	5.3	Simulação da Estrutura de Controle Auto-Ajustável	48

CAPÍTULO	6	IMPLEMENTAÇÃO DO CONTROLE AUTO-AJUSTÁVEL	52
	6.1	Introdução	52
	6.2	Diagrama em Blocos	55
	6.3	Implementação Prática	56
	6.4	Resultados Práticos	59
CAPÍTULO	7	SOFTWARE DE CONTROLE	78
	7.1	Introdução	78
	7.2	Programa	78
CAPÍTULO	8	CONCLUSÕES	82
CAPÍTULO	9	EXTENSÕES PARA TRABALHOS FUTUROS	86
BIBLIOGRAFIA			88
APÊNDICE	A	FÓRMULA DE SHERMAN-MORRISON-WOODBURY	91
APÊNDICE	B	RESPOSTA A UM DEGRAU NEGATIVO	94
APÊNDICE	C	LISTAGEM DO PROGRAMA	96

INTRODUÇÃO

Nas duas últimas décadas, a teoria de controle moderno tem se desenvolvido na direção do estudo, análise e implementação de sistemas de controle cada vez mais complexos, e com melhor desempenho. Dentre os vários princípios do controle moderno, encontram-se as técnicas de controle adaptativo, cujo interesse significativo surgiu no início de 1950, em vista das perspectivas que essas técnicas de controle podiam oferecer.

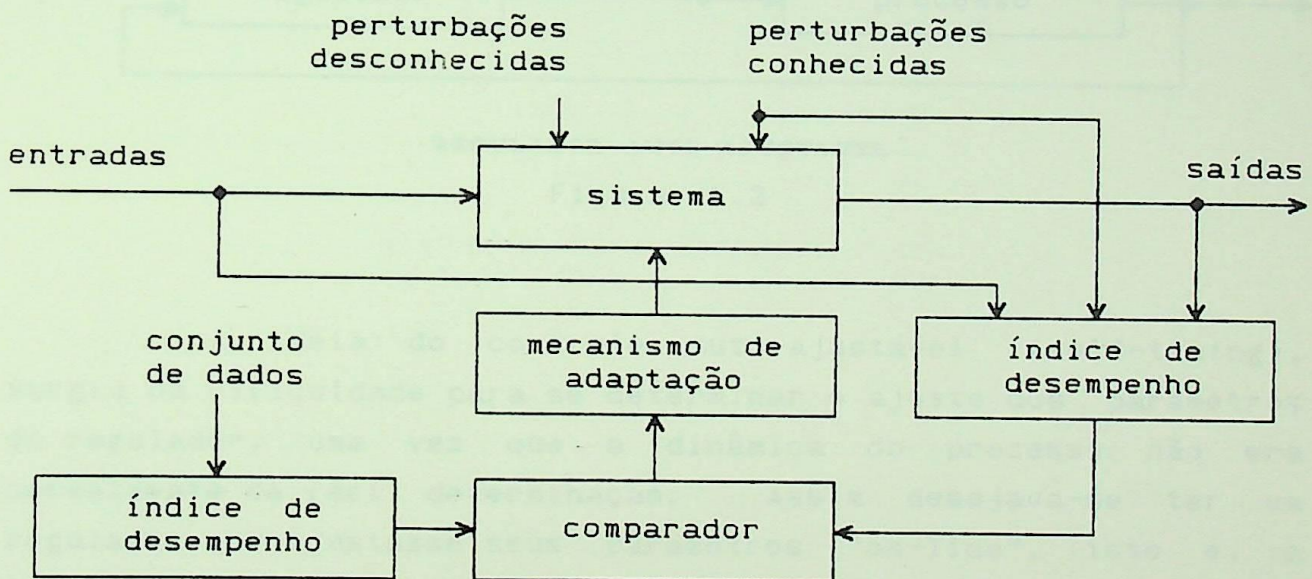
Com as contribuições feitas à teoria de controle na década de sessenta, e ao rápido desenvolvimento nos últimos anos, de tecnologias, como por exemplo, a da microeletrônica, as técnicas de controle adaptativo vem se desenvolvendo de forma vigorosa. Atualmente, estas técnicas contam com vários trabalhos publicados, e são motivo de intensas pesquisas nos estabelecimentos universitários e industriais.

As técnicas de controle adaptativo, têm sido utilizadas numa tentativa de implementar sistemas de controle com um alto índice de desempenho. Nos sistemas de controle adaptativo as características dinâmicas da planta, podem ser pouco conhecidas, ou até mesmo desconhecidas, além de poderem sofrer variações, devido a fatores ambientais.

Os sistemas de controle adaptativo são orientados, de modo a eliminar os efeitos causados, essencialmente, por variações dos parâmetros da planta sobre o desempenho global do controle. No entanto, a literatura não apresenta um consenso para a definição de controle adaptativo, dado que esse possui uma grande variedade de significados. Neste trabalho será usado a definição descrita a seguir [1]:

"Em um sistema de controle adaptativo, o índice de desempenho, obtido por meio de suas entradas, saídas e estados internos, é comparado com um outro índice de desempenho oriundo de um conjunto de dados especificado. Por meio de um mecanismo de adaptação, os parâmetros do sistema são modificados, ou uma entrada auxiliar é gerada, de forma a aproximar o índice de desempenho do sistema do índice do conjunto de dados especificado, isto é, colocá-lo dentro de uma faixa de valores aceitáveis."

No enunciado acima, ilustrado pela figura 1.1, todo sistema é capaz de ajustar seu desempenho, ou por modificação de seus parâmetros (estrutura interna) ou por modificação dos seus sinais de entrada (excitação).



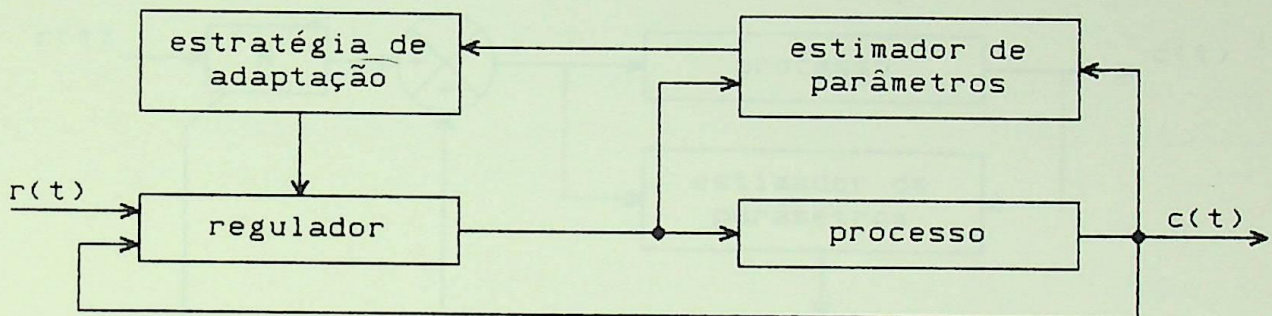
SISTEMA DE CONTROLE ADAPTATIVO

Figura 1.1

Das estratégias de controle adaptativo atualmente conhecidas, a que utiliza um regulador auto-ajustável (Self-Tuning Regulator - STR), cujo esquema foi inicialmente proposto por Kalman (1958), tem recebido muita atenção nos dias

atuais. A atração por esta técnica de controle é devido, principalmente, a sua flexibilidade, facilidade de entendimento e facilidade de implementação por meio de microprocessadores.

Os sistemas de controle adaptativo que utilizam um regulador auto-ajustável, são compostos basicamente por três partes: um estimador de parâmetros, um regulador e um dispositivo que especifica a estratégia de adaptação a ser utilizada no controle (figura 1.2).



REGULADOR AUTO-AJUSTAVEL

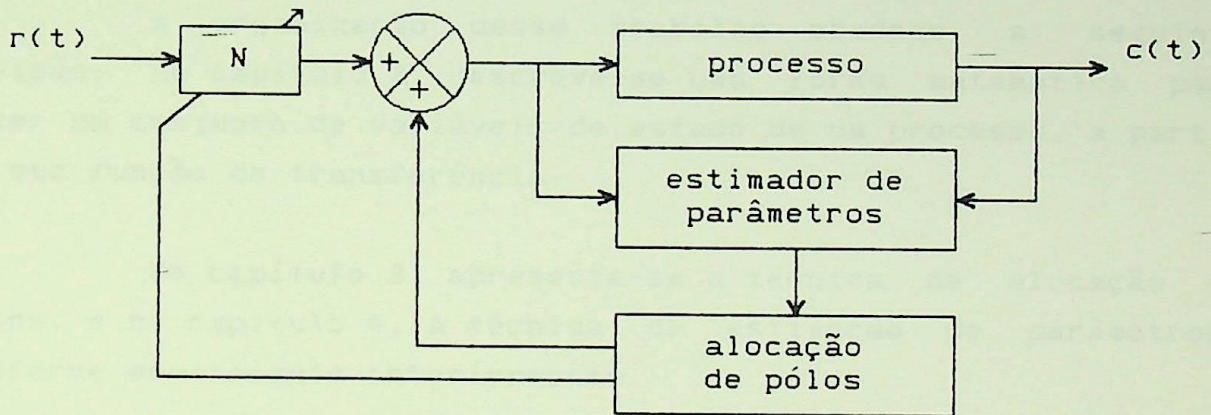
Figura 1.2

A idéia do controle auto-ajustável (self-tuning), surgiu da dificuldade para se determinar o ajuste dos parâmetros do regulador, uma vez que a dinâmica do processo não era normalmente de fácil determinação. Assim desejava-se ter um regulador que ajustasse seus parâmetros "on-line", isto é, a medida que o sistema estivesse em funcionamento.

O objetivo do controle auto-ajustável é controlar sistemas com parâmetros desconhecidos, porém constantes, ou que variam lentamente. O interesse teórico por este tipo de controle diz respeito à elegância dos algoritmos envolvidos na análise do desempenho e convergência, enquanto o interesse prático deriva de seu potencial como uma ferramenta simples e eficiente de controle.

O objetivo primordial desse trabalho, é a implementação prática de um sistema de controle auto-ajustável (self-tuning) de uma única entrada e uma única saída. Como implementação prática foi utilizado o controle de velocidade de um motor de corrente contínua acionado por conversor tiristorizado, a fim de obter uma análise comparativa entre os resultados práticos e teóricos.

A estrutura proposta para o controle é apresentada, na forma de diagrama de blocos, pela figura 1.3.



ESTRUTURA DE CONTROLE PROPOSTA

Figura 1.3

A fim de detectar as mudanças que possam vir a ocorrer nos parâmetros do processo sob controle, será utilizado o algoritmo dos mínimos quadrados recursivo (Recursive Least Squares - RLS) para estimar os parâmetros do processo. Este algoritmo, bem como a técnica de estimação de parâmetros empregada na implementação do controle, é descrita com maiores detalhes no quarto capítulo.

O controle do processo é determinado considerando que o modelo estimado, por meio do algoritmo de estimação de parâmetros, é igual ao do modelo real do processo. O ajuste global da malha de controle se dá por intermédio da modificação

de seu sinal de entrada, ao invés da modificação dos parâmetros de um regulador a ser incluído na malha de controle.

A técnica empregada para que o controle seja auto-ajustável, consiste em deslocar os pólos, da função de transferência de malha fechada do processo, para posições pré-definidas. Deste modo, é possível fazer com que o processo sempre apresente o mesmo comportamento transitório, independentemente das mudanças que possam vir a ocorrer em seus parâmetros. Esta técnica conhecida por alocação de pólos (Pole Assignment) é apresentada no terceiro capítulo.

A organização desse trabalho obedece a seguinte divisão: no capítulo 2, descreve-se uma forma matemática para obter um conjunto de variáveis de estado de um processo, a partir de sua função de transferência.

No capítulo 3, apresenta-se a técnica de alocação de pólos, e no capítulo 4, a técnica de estimação de parâmetros, conforme mencionados anteriormente.

Os resultados da simulação do sistema são apresentados no capítulo 5.

No capítulo 6, descreve-se a implementação prática do sistema de controle auto-ajustável. Neste capítulo também são apresentados os resultados experimentais do protótipo desenvolvido.

No capítulo 7, descreve-se o "software" desenvolvido para o sistema de controle proposto.

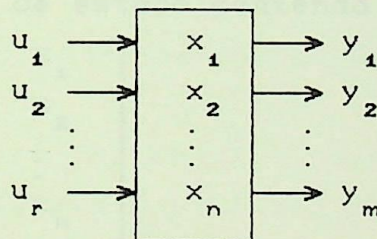
No capítulo 8, são apresentadas as conclusões a respeito da técnica de controle proposta. Finalmente, no capítulo 9, são descritas sugestões para a melhoria deste trabalho, juntamente com algumas idéias para novos trabalhos de pesquisa.

FORMA CANÔNICA CONTROLÁVEL

2.1 Introdução

As variáveis de estado de um processo descrevem completamente a sua dinâmica. Em outras palavras, elas representam o mínimo de informação necessária para se determinar os subseqüentes estados, e a resposta do processo para uma dada entrada.

A representação de um processo segundo suas variáveis de estado é ilustrada na figura a seguir:

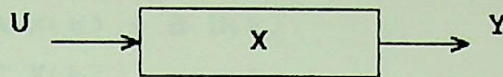


REPRESENTAÇÃO DE UM PROCESSO POR VARIÁVEIS DE ESTADO

Figura 2.1

Para uma completa generalização, a figura 2.1, representa um processo com mais de uma entrada e mais de uma saída. Nesta figura, u_i , $i = 1, 2, \dots, r$ são as entradas externas que comandam o processo, y_i , $i = 1, 2, \dots, m$ são as saídas ou as respostas do processo e x_i , $i = 1, 2, \dots, n$ são as variáveis internas ou variáveis de estado.

Uma forma mais conveniente para representar o processo da figura 2.1, é ilustrado pela figura 2.2.



REPRESENTAÇÃO DE UM PROCESSO POR VETOR DE ESTADO

Figura 2.2

onde: U é o vetor de entrada

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}$$

Y é o vetor de saída

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

X é o vetor de estado contendo as variáveis de estado

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Se um processo é linear e discretizado, pode-se representá-lo por equações de estado na forma:

$$\begin{aligned} X(k+1) &= A(k) X(k) + B(k) U(k) \\ Y(k) &= C(k) X(k) \end{aligned} \quad (2.1)$$

onde: X , U e Y são vetores de ordem n , r e m , respectivamente. $A(k)$, $B(k)$ e $C(k)$ são matrizes de dimensões apropriadas, e que variam no tempo

Caso o processo seja invariante no tempo, tem-se que:

$$\begin{aligned} X(k+1) &= A X(k) + B U(k) \\ Y(k) &= C X(k) \end{aligned} \quad (2.2)$$

onde: A , B e C são matrizes que não variam no tempo

As equações acima são as equações necessárias para representar um processo linear e invariante no tempo, segundo suas variáveis de estado.

Uma vez que um processo é geralmente especificado por meio de sua função de transferência, torna-se necessário em alguns casos obter de forma matemática suas equações de estado a partir da transformada Z desta função de transferência. Uma das maneiras de se conseguir isto, consiste em representar o processo na sua forma canônica controlável [2].

2.2 Representação na Forma Canônica Controlável

Um processo discretizado, linear e invariante no tempo com entrada $U(z)$ e saída $Y(z)$, pode ser representado pela função de transferência:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_{n-1} z^{n-1} + \dots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0} \quad (2.3)$$

onde: b_n e a_n são parâmetros que caracterizam o processo

$$\text{fazendo: } Y(z) = (b_{n-1} z^{n-1} + \dots + b_1 z + b_0) E(z)$$

$$U(z) = (z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0) E(z)$$

$$z^n E(z) = E(z+n)$$

resulta:
$$Y(z) = b_{n-1}E(z+n-1) + \dots + b_1E(z+1) + b_0E(z)$$

$$U(z) = E(z+n) + a_{n-1}E(z+n-1) + \dots + a_1E(z+1) + a_0E(z)$$

Considerando que o processo tenha uma única entrada e uma única saída, pode-se escrever:

$$\begin{aligned} Y(z) &\longrightarrow y(k) \\ U(z) &\longrightarrow u(k) \\ E(z) &\longrightarrow e(k) \end{aligned}$$

assim:

$$y(k) = b_{n-1}e(k+n-1) + \dots + b_1e(k+1) + b_0e(k) \tag{2.4}$$

$$u(k) = e(k+n) + a_{n-1}e(k+n-1) + \dots + a_1e(k+1) + a_0e(k) \tag{2.5}$$

Definindo as variáveis de estado:

$$\begin{aligned} x_1(k) &= e(k) \\ x_2(k) &= x_1(k+1) = e(k+1) \\ x_3(k) &= x_2(k+1) = e(k+2) \\ &\dots \\ x_n(k) &= x_{n-1}(k+1) = e(k+n-1) \end{aligned} \tag{2.6}$$

obtem-se das equações (2.5) e (2.6), as seguintes equações de estado:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ x_3(k+1) &= x_4(k) \\ &\dots \\ x_n(k+1) &= -a_0x_1(k) - a_1x_2(k) - \dots - a_{n-1}x_n(k) + u(k) \end{aligned}$$

que pode ser escrita na forma matricial:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \quad (2.7)$$

ou simplesmente:

$$X(k+1) = A X(k) + B u(k) \quad (2.8)$$

a equação de saída é obtida das equações (2.4) e (2.6), como:

$$y(k) = [b_0 \quad b_1 \quad \dots \quad b_{n-1}] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} \quad (2.9)$$

ou simplesmente:

$$y(k) = C X(k) \quad (2.10)$$

Uma forma conveniente de representar um processo discretizado, é através de seu diagrama de fluxo de sinal, ou de seu diagrama equivalente de simulação. Essas duas formas podem ser derivadas da equação (2.3), dividindo primeiramente o numerador e o denominador por z^n , com isso resulta:

$$\frac{Y(z)}{U(z)} = \frac{b_{n-1} z^{-1} + \dots + b_1 z^{n-1} + b_0 z^{-n}}{1 + a_{n-1} z^{-1} + \dots + a_1 z^{n-1} + a_0 z^{-n}} \quad (2.11)$$

dessa expressão, obtém-se:

$$Y(z) = (b_{n-1}z^{-1} + \dots + b_1z^{n-1} + b_0z^{-n}) E(z) \quad (2.12)$$

$$U(z) = (1 + a_{n-1}z^{-1} + \dots + a_1z^{n-1} + a_0z^{-n}) E(z) \quad (2.13)$$

a equação (2.13) pode ser reescrita na forma:

$$E(z) = U(z) - (a_{n-1}z^{-1} + \dots + a_1z^{n-1} + a_0z^{-n}) E(z) \quad (2.14)$$

A representação na forma de diagrama de fluxo de sinal da equação (2.3) é derivada das equações (2.12) e (2.14), sendo apresentada na figura 2.3.

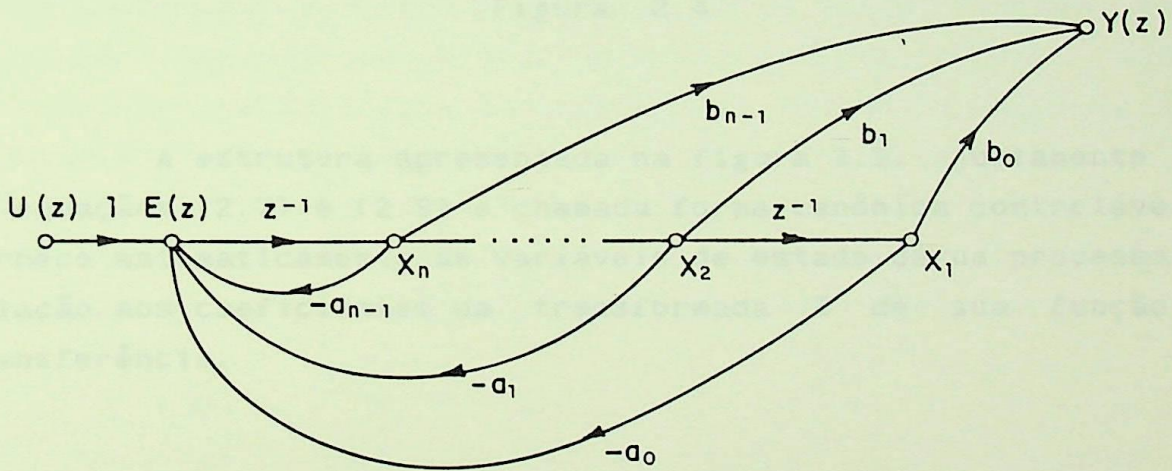


DIAGRAMA DE FLUXO DE SINAL

Figura 2.3

Lembrando que z^{-1} representa um atraso unitário T , onde T é o intervalo de amostragem do sinal de entrada. O diagrama de fluxo de sinal da figura 2.3 pode ser convertido para o diagrama equivalente, mostrado na figura 2.4, mais apropriado à simulação.

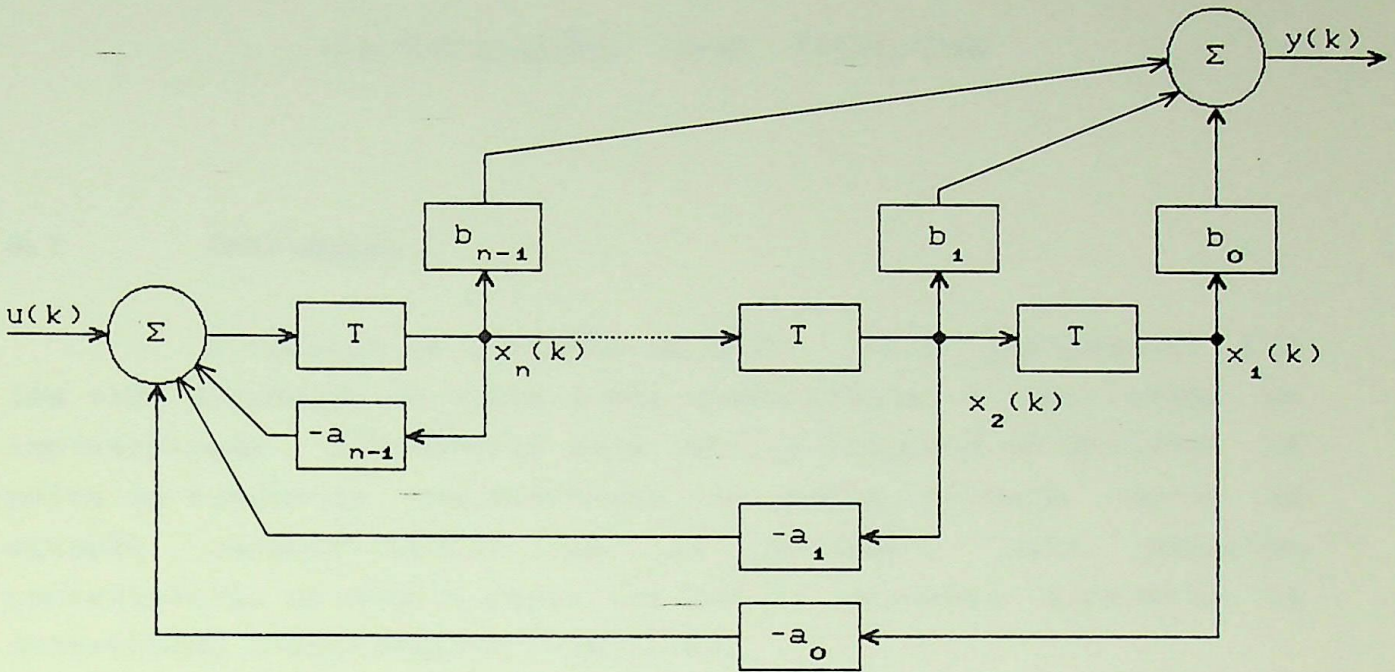


DIAGRAMA DE SIMULAÇÃO

Figura 2.4

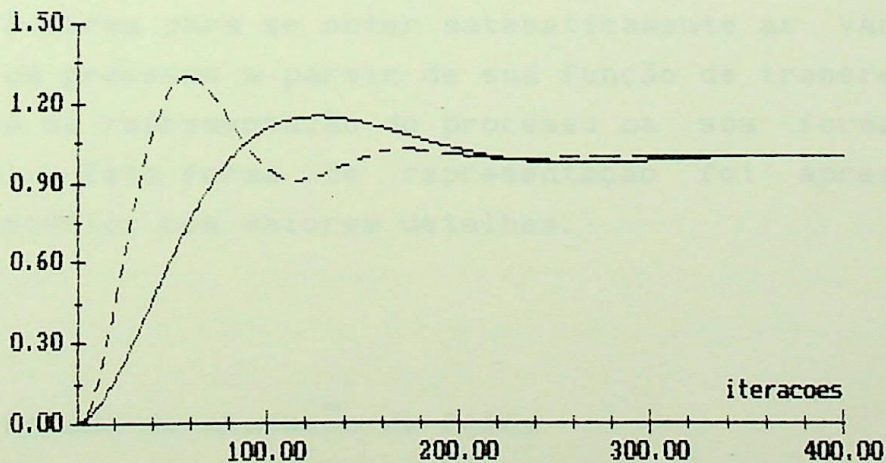
A estrutura apresentada na figura 2.3, juntamente com as equações (2.7) e (2.9) é chamada forma canônica controlável e fornece matematicamente as variáveis de estado de um processo em relação aos coeficientes da transformada Z de sua função de transferência.

ALOCAÇÃO DE POLOS

3.1 Introdução

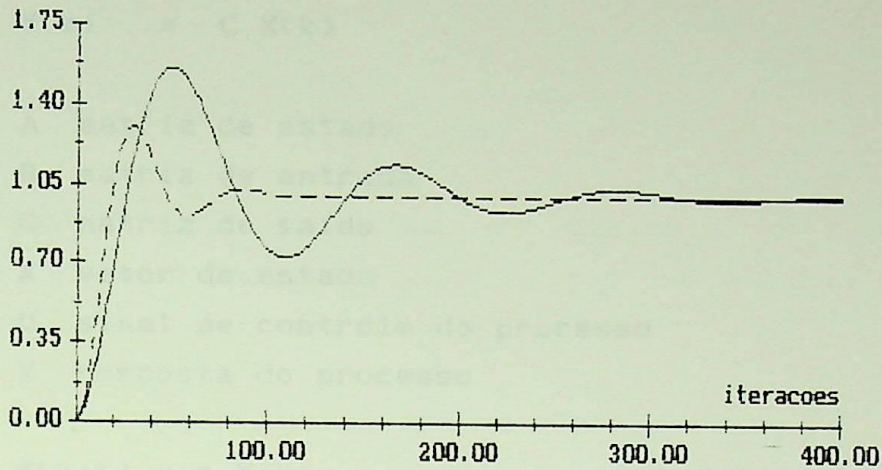
A técnica de alocação de pólos (Pole Assignment) [2] tem sido difundida em vista a sua simplicidade e facilidade de implementação. Basicamente essa técnica consiste em deslocar os pólos da função de transferência de malha fechada (zeros da equação característica) de um processo, para posições convenientes, de modo a fazer com que o processo apresente um determinado comportamento transitório.

Como exemplo a figura 3.1(a) mostra, por meio de uma linha tracejada, a resposta temporal de um dado sistema de 2^a ordem para uma entrada em degrau unitário. Uma possível resposta que pode ser obtida, após o deslocamento dos pólos da função de transferência do sistema em questão, é indicada nessa figura por uma linha cheia. Da mesma forma, a figura 3.1(b), apresenta as resposta de um outro sistema de 2^a ordem, antes e depois do deslocamento dos pólos da sua função de transferência.



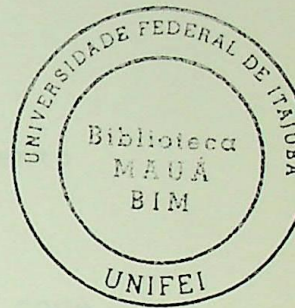
TECNICA DE ALOCAÇÃO DE POLOS

Figura 3.1(a)



TECNICA DE ALOCAÇÃO DE POLOS

Figura 3.1(b)



Uma das formas de obter o deslocamento desejado dos pólos da função de transferência de um processo, é mediante a realimentação das variáveis de estado do processo sob controle. Entretanto, para processos inicialmente desconhecidos, o método de alocação de pólos por realimentação das variáveis de estado se justifica, quando a este é associado um estimador de estado. Para processos previamente conhecidos, as variáveis de estado podem ser obtidas, matematicamente, por meio de sua função de transferência ou diretamente de suas equações de estado.

A forma para se obter matematicamente as variáveis de estado de um processo a partir de sua função de transferência se dá por meio da representação do processo na sua forma canônica controlável. Esta forma de representação foi apresentada no capítulo anterior com maiores detalhes.

3.2 Método de Alocação de Pólos

Os sistemas de controle lineares, discretizados, podem ser representados por uma equação de estado na forma:

$$\begin{aligned} X(k+1) &= A X(k) + B U(k) \\ Y(k) &= C X(k) \end{aligned} \quad (3.1)$$

onde:

- A matriz de estado
- B matriz de entrada
- C matriz de saída
- X vetor de estado
- U sinal de controle do processo
- Y resposta do processo

$$X(k+1) = z X(k) \quad \therefore \quad z \text{ operador deslocamento}$$

A equação de estado que descreve um sistema, onde os pólos da função de transferência de malha fechada se encontram em determinadas posições, será do tipo:

$$\begin{aligned} X(k+1) &= A_P X(k) + B_P U(k) \\ Y_P(k) &= C_P X(k) \end{aligned} \quad (3.2)$$

cuja equação característica é dada por [2]:

$$| z I - A_P | = 0$$

onde:

- z operador deslocamento
- I matriz identidade

Se os pólos da função de transferência de malha fechada se deslocarem, para posições diferentes daquelas em que se encontravam inicialmente, o sistema será representado por uma nova equação de estado, dada por:

$$\begin{aligned} X(k+1) &= A_M X(k) + B_M U(k) \\ Y_M(k) &= C_M X(k) \end{aligned} \quad (3.3)$$

neste caso, a equação característica do sistema será:

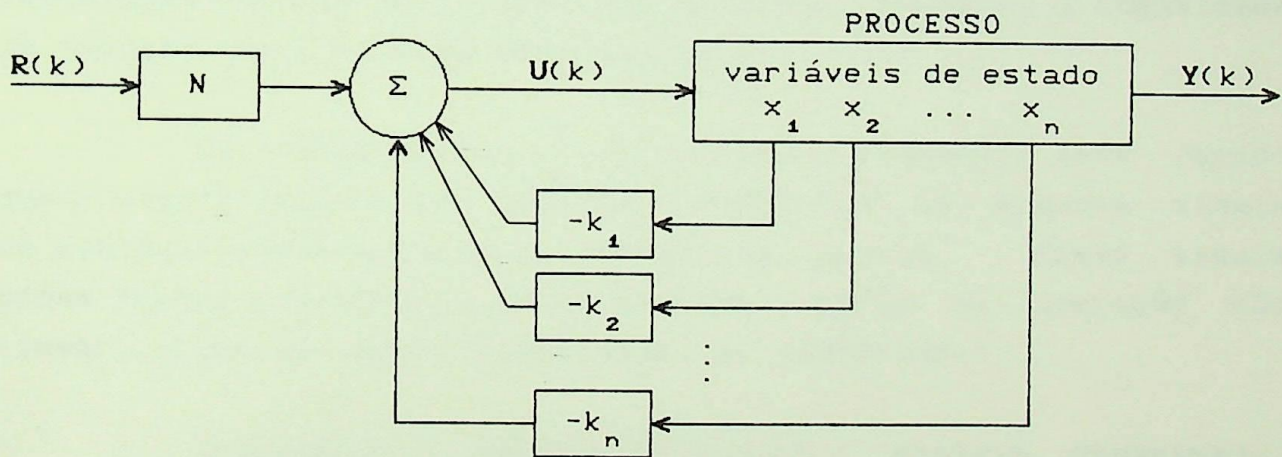
$$| z I - A_M | = 0 \quad (3.4)$$

Fazendo com que o sinal de controle do processo, na equação (3.2), seja uma combinação linear dos estados estimados e do sinal de referência (figura 3.2), vem:

$$U(k) = - K X(k) + N R(k) \quad (3.5)$$

sendo: N uma constante a ser determinada,
 K um vetor de ganho dado por:

$$K = [k_1 \quad k_2 \quad \dots \quad k_n]$$



REALIMENTAÇÃO POR VARIÁVEIS DE ESTADO

Figura 3.2

assim:

$$X(k+1) = (A_P - B_P K) X(k) + B_P N R(k) \quad (3.6)$$

$$Y_P(k) = C_P X(k)$$

Neste caso, a equação característica do sistema realimentado por suas variáveis de estado, será dada por:

$$| z I - (A_P - B_P K) | = 0 \quad (3.7)$$

Comparando as equações (3.4) e (3.7), podemos forçar:

$$A_M = A_P - B_P K \quad (3.8)$$

Dessa forma, calculando a matriz de ganho K é possível deslocar, teoricamente, os pólos do sistema para quaisquer posições, de modo que o sistema responda com certas características específicas. Naturalmente, não se pode aumentar a velocidade de resposta de um sistema, além de um certo limite, embora o método pareça indicar ser possível. Assim, os valores limites para a matriz K , devem levar em consideração as restrições físicas do sistema sob controle, ou seja, a quantidade de energia que o sistema pode manipular.

Se tentarmos forçar um sistema a responder mais rápido que o limite imposto por suas restrições físicas, grandes sinais de controle serão gerados na entrada da planta. Esses sinais podem forçar o sistema a entrar em uma região de operação não linear, o que em alguns casos pode ser indesejável.

A equação de transferência para o sistema compensado, resulta da equação (3.6):

$$Y_P(z) = C_P [z I - (A_P - B_P K)]^{-1} B_P N R(z) \quad (3.9)$$

Da mesma forma, obtém-se para o sistema representado pela equação (3.3), a equação de transferência:

$$Y_M(z) = C_M [z I - A_M]^{-1} B_M R(z) \quad (3.10)$$

Fazendo:

$$A = A_M = A_P - B_P K$$

e sabendo que:

$$[z I - A]^{-1} = \frac{1}{|z I - A|} \text{adj} [z I - A]$$

onde: adj denota a matriz adjunta

resulta:

$$Y_P(z) = \frac{D_P(z)}{|z I - A|} N R(z) \quad (3.11)$$

$$Y_M(z) = \frac{D_M(z)}{|z I - A|} R(z)$$

sendo: $D_P(z) = C_P \text{adj} [z I - A] B_P$

$$D_M(z) = C_M \text{adj} [z I - A] B_M$$

Para determinar os zeros da função de transferência do sistema compensado, considere a transformada Z da equação (3.6):

$$\begin{aligned} z X(z) &= (A_P - B_P K) X(z) + B_P N R(z) \\ Y_P(z) &= C_P X(z) \end{aligned}$$

donde resulta:

$$\begin{aligned} (z I - A_P + B_P K) X(z) - B_P N R(z) &= 0 \\ C_P X(z) &= Y_P(z) \end{aligned} \quad (3.12)$$

Se $z = z_0$ é um zero da equação de transferência (3.9), então $Y_P(z_0)$ é igual a zero. Conseqüentemente a equação (3.12), pode ser expressa por:

$$\begin{bmatrix} z_0 I - A_P + B_P K & - B_P N \\ C_P & 0 \end{bmatrix} \begin{bmatrix} X(z_0) \\ R(z_0) \end{bmatrix} = 0 \quad (3.13)$$

Essa equação possui uma solução não trivial, uma vez que, $X(z_0)$ e $R(z_0)$ são diferentes de zero. Assim o determinante da matriz de coeficientes, que é um polinômio em z_0 deve ser zero, ou seja:

$$\begin{vmatrix} z_0 I - A_P + B_P K & - B_P N \\ C_P & 0 \end{vmatrix} = 0 \quad (3.14)$$

as raízes do polinômio são então os zeros da função de transferência do sistema compensado [2].

Uma vez que os zeros da função de transferência não estão evidentes na equação (3.14), a equação (3.13) pode ser reescrita na forma:

$$\begin{bmatrix} z_0 I - A_P & - B_P \\ C_P & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -K & N \end{bmatrix} \begin{bmatrix} X(z_0) \\ R(z_0) \end{bmatrix} = 0$$

ou melhor:

$$\begin{bmatrix} z_0 I - A_P & - B_P \\ C_P & 0 \end{bmatrix} \begin{bmatrix} X(z_0) \\ N R(z_0) - K X(z_0) \end{bmatrix} = 0 \quad (3.15)$$

Assim sendo, a matriz de coeficientes é independente de K , e depende exclusivamente da planta. Os zeros da função de

transferência de malha fechada do sistema compensado, são os mesmos da função de transferência do processo não compensado, e não podem ser alterados pelo procedimento descrito acima. O que leva a concluir que o método de alocação de pólos é de valor limitado.

A constante N é calculada, de modo que a resposta estacionária do sistema compensado, equação (3.11), para uma entrada em degrau unitário, seja idêntica a que o sistema representado pela equação (3.3) teria, caso este fosse submetido ao mesmo sinal de entrada.

Para essa restrição o valor de N é dado por:

$$\lim_{t \rightarrow \infty} y_P(t) = \lim_{t \rightarrow \infty} y_M(t)$$

ou

$$\lim_{z=1} N D_P(z) = \lim_{z=1} D_M(z)$$

Observa-se que a constante N não afeta a resposta relativa em frequência do sistema, mas somente a amplitude da resposta em frequência. A função de transferência é independente de N , exceto como um fator multiplicativo.

ESTIMADOR DE PARÂMETROS

4.1 Introdução

O interesse na área de identificação de parâmetros tem sido, em parte, gerado pela dificuldade em se obter modelos matemáticos de processos industriais, sobretudo daqueles que apresentam variações dinâmicas dos parâmetros.

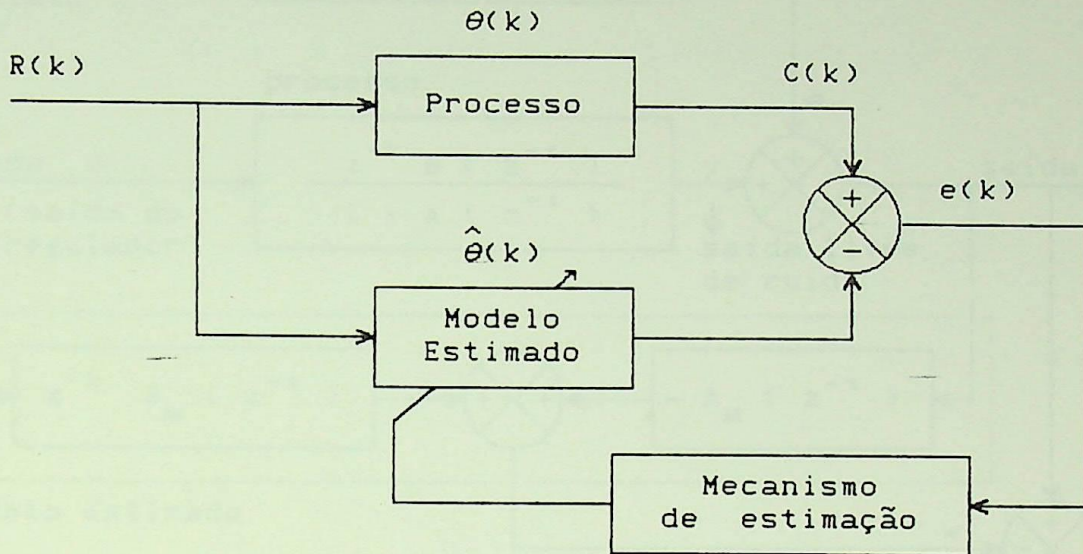
A identificação dos parâmetros de um processo dinâmico, consta basicamente de quatro etapas [1]:

- identificação estrutural, que permite caracterizar a estrutura de um modelo matemático para o processo.
- aquisição de dados de entrada e saída.
- estimação de parâmetros, a fim de determinar, o mais exato possível, os parâmetros do modelo matemático levantado para o processo a ser identificado.
- verificação da validade do modelo matemático obtido para o processo.

Entre os vários métodos de identificação de parâmetros, que surgiram ao longo dos tempos, o método dos mínimos quadrados (Least Squares - LS) destaca-se como um dos métodos mais importantes, visto que este constitui a base para vários outros algoritmos de identificação [3]. Atualmente o método dos mínimos quadrados é um dos mais usados, devido a sua simplicidade, facilidade de implementação e recursividade.

De uma forma geral, a idéia da recursividade, como mostra a figura 4.1, é obter uma nova estimativa para os parâmetros do processo a cada nova aquisição de dados, de modo que os parâmetros sejam constantemente atualizados com a evolução do processo. Sua principal vantagem é a de tornar desnecessária

a inversão de uma matriz de ordem elevada, inerente ao método dos mínimos quadrados, o que nesse caso conduz a um aumento na velocidade de execução do algoritmo de identificação de parâmetros.



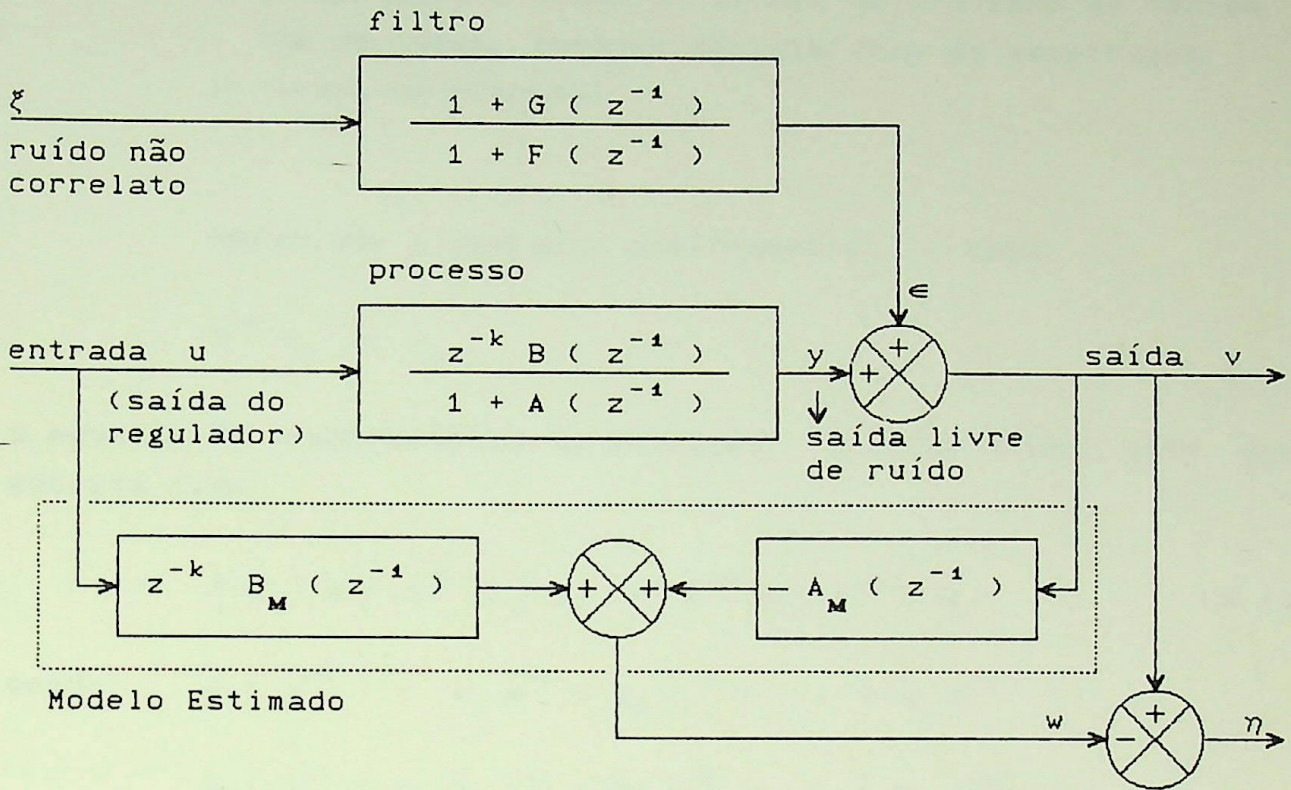
ESTIMADOR DE PARAMETROS

Figura 4.1

onde: $\hat{\theta}(k+1) = \hat{\theta}(k) + A(k) \cdot e(k)$
 $\hat{\theta}(k)$ - vetor de parâmetros do modelo estimado
 $A(k)$ - matriz de ganho no instante k
 $e(k)$ - erro entre a saída do processo e do modelo

4.2 Generalização do Método de Identificação de Parâmetros

O método recursivo dos mínimos quadrados (Recursive Least Squares - RLS) [4], devido as suas características, tem sido largamente utilizado na estimação de parâmetros de sistemas de controle, nos quais os parâmetros da planta são desconhecidos, ou sofrem mudanças bruscas ou lentas. Esse método é descrito a seguir, para o sistema genérico linear, discretizado, de ordem n , com uma única entrada e uma única saída, mostrado na figura 4.2.



SISTEMA LINEAR DISCRETIZADO

Figura 4.2

onde: $A_M(z^{-1}) = \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \dots + \hat{a}_n z^{-n}$

$B_M(z^{-1}) = \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + \dots + \hat{b}_n z^{-n}$

Em um dado instante de amostragem, a saída livre de ruído y , indicada na figura 4.2, é dada pela soma dos valores passados, dessa mesma saída ($y_{t-1}, y_{t-2}, \dots, y_{t-n}$), com os valores passados da entrada do processo ($u_{t-k-1}, u_{t-k-2}, \dots, u_{t-k-n}$), sendo cada fator multiplicado por um coeficiente adequado. Assim:

$$y_t = -a_1 y_{t-1} - a_2 y_{t-2} - \dots - a_n y_{t-n} + b_1 u_{t-k-1} + b_2 u_{t-k-2} + \dots + b_n u_{t-k-n}$$

onde: t instante de amostragem
 k representa o tempo de atraso do processo em termos de um número inteiro do intervalo de amostragem
 n ordem do processo

Definindo o operador deslocamento z , como:

$$z^{-n} y_t = y_{t-n}$$

a equação de transferência do processo, livre de ruído, pode ser escrita como:

$$[1 + A(z^{-1})] y_t = z^{-k} B(z^{-1}) u_t \quad (4.1)$$

sendo: $A(z^{-1}) = a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}$$

O ruído do processo é considerado aleatório, sendo representado na figura 4.2, por uma fonte equivalente de ruído introduzida na saída do processo. A fonte equivalente de ruído é composta basicamente de um filtro alimentado por uma fonte de ruído branco. Tal como para o processo, a equação de transferência do filtro é dada por:

$$[1 + F(z^{-1})] \epsilon_t = [1 + G(z^{-1})] \xi_t \quad (4.2)$$

onde: ξ_t e ϵ_t são os sinais amostrados, respectivamente, na entrada e na saída do filtro

A saída do processo v_t em um determinado instante de amostragem, é dada pela equação:

$$v_t = y_t + \epsilon_t \quad (4.3)$$

substituindo as equações (4.1) e (4.2) na equação (4.3), resulta:

$$[1 + A(z^{-1})] v_t = z^{-k} B(z^{-1}) u_t + e_t \quad (4.4)$$

ou

$$v_t = -A(z^{-1}) v_t + z^{-k} B(z^{-1}) u_t + e_t$$

onde:

$$e_t = \frac{[1 + A(z^{-1})][1 + G(z^{-1})]}{[1 + F(z^{-1})]} \xi_t$$

A fim de conseguir a recursividade desejada, considere a seqüência de observações (u,v), obtida na entrada e na saída do processo, respectivamente:

$$v_t = -a_1 v_{t-1} - a_2 v_{t-2} - \dots - a_n v_{t-n} + b_1 u_{t-k-1} + b_2 u_{t-k-2} + \dots + b_n u_{t-k-n} + e_t$$

$$v_{t+1} = -a_1 v_t - a_2 v_{t-1} - \dots - a_n v_{t-n+1} + b_1 u_{t-k} + b_2 u_{t-k-1} + \dots + b_n u_{t-k-n+1} + e_{t+1}$$

...

$$v_{t+N} = -a_1 v_{t-1+N} - a_2 v_{t-2+N} - \dots - a_n v_{t-n+N} + b_1 u_{t-k-1+N} + b_2 u_{t-k-2+N} + \dots + b_n u_{t-k-n+N} + e_{t+N}$$

que pode ser escrita na forma matricial:

$$V = X\theta + e \quad (4.5)$$

onde: $V^T = [v_t \ v_{t+1} \ \dots \ v_{t+N}]$

$$U^T = [u_t \ u_{t+1} \ \dots \ u_{t+N}]$$

$$e^T = [e_t \ e_{t+1} \ \dots \ e_{t+N}]$$

$$X = [-z^{-1}V \mid -z^{-2}V \mid \dots \mid -z^{-n}V \mid -z^{-k-1}U \mid -z^{-k-2}U \mid \dots \mid -z^{-k-n}U]$$

$$\theta^T = [a_1 \ a_2 \ \dots \ a_n \ b_1 \ b_2 \ \dots \ b_n]$$

v_t , u_t e e_t são valores amostrados da saída, da entrada e do sinal de ruído, respectivamente

A saída do modelo discretizado, que permite caracterizar o processo para o qual os parâmetros serão estimados, é dado conforme figura 4.2, pela equação:

$$w_t \equiv -A_M(z^{-1})v_t + z^{-k}B_M(z^{-1})u_t \quad (4.6)$$

sendo: $A_M(z^{-1}) = \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \dots + \hat{a}_n z^{-n}$

$$B_M(z^{-1}) = \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + \dots + \hat{b}_n z^{-n}$$

w_t saída do modelo em um instante de amostragem
 u_t sinal de entrada em um instante de amostragem,
 comum ao processo e ao modelo

da mesma forma que para a saída do processo v_t , a equação acima também pode ser escrita na forma matricial:

$$W = X \hat{\theta} \quad (4.7)$$

onde, $\hat{\theta}$ é o vetor de parâmetros do modelo matemático a ser estimado para o processo, sendo dado por:

$$\hat{\theta}^T = [\hat{a}_1 \ \hat{a}_2 \ \dots \ \hat{a}_n \ \hat{b}_1 \ \hat{b}_2 \ \dots \ \hat{b}_n]$$

O vetor de parâmetros $\hat{\theta}$ é obtido minimizando o erro residual entre a saída do processo e a saída prevista pelo modelo, após um conjunto de observações (u,v) . Isto é possível

através da minimização da função custo descrita pela equação (4.8), com relação ao vetor de parâmetros $\hat{\theta}$.

$$J = \eta^T \eta = \sum_{i=1}^N \eta_i^2 \quad (4.8)$$

sendo, η o vetor erro residual, definido como:

$$\eta = V - W \quad (4.9)$$

onde: $\eta^T = [\eta_1 \quad \eta_{l+1} \quad \dots \quad \eta_{l+N}]$

substituindo a equação (4.9) na equação (4.8), resulta:

$$J = [V - W]^T [V - W]$$

porém da equação (4.7), vem:

$$W = X \hat{\theta}$$

portanto:

$$J = [V - X \hat{\theta}]^T [V - X \hat{\theta}]$$

A literatura mostra que a minimização da função custo, com relação ao vetor de parâmetros $\hat{\theta}$, é dada por:

$$\frac{\partial J}{\partial \hat{\theta}} = 0$$

ou seja:

$$\frac{\partial}{\partial \hat{\theta}} (\eta^T \eta) = 0$$

$$\frac{\partial}{\partial \hat{\theta}} [V - X \hat{\theta}]^T [V - X \hat{\theta}] = 0$$

para $\eta^T = [\eta_t \quad \eta_{t+1} \quad \dots \quad \eta_{t+N}]$ tem-se que:

$$\frac{\partial}{\partial \hat{\theta}} (\eta^T \eta) = \frac{\partial \eta^T}{\partial \hat{\theta}} \eta + \eta^T \frac{\partial \eta}{\partial \hat{\theta}}$$

$$\frac{\partial}{\partial \hat{\theta}} (\eta^T \eta) = 2 \frac{\partial \eta^T}{\partial \hat{\theta}} \eta = 2 \left[\frac{\partial \eta}{\partial \hat{\theta}} \right]^T \eta$$

portanto:

$$\frac{\partial}{\partial \hat{\theta}} (\eta^T \eta) = -2 \mathbf{x}^T [\mathbf{v} - \mathbf{x} \hat{\theta}] = 0$$

$$\mathbf{x}^T [\mathbf{v} - \mathbf{x} \hat{\theta}] = 0$$

logo: $\hat{\theta} = [\mathbf{x}^T \mathbf{x}]^{-1} \mathbf{x}^T \mathbf{v}$ (4.10)

O vetor de parâmetros $\hat{\theta}$, estimado pela equação (4.10), tem a desvantagem de ser polarizado, isto é:

$$p \hat{\Delta} \theta - E \hat{\theta} \neq 0 \quad [5]$$

onde: p polarização
 $E \hat{\theta}$ valor esperado de $\hat{\theta}$ [6]

desde que:

$$\mathbf{v} = \mathbf{x} \theta + \mathbf{e}$$

a equação (4.10) ainda pode ser escrita:

$$\hat{\theta} = [\mathbf{x}^T \mathbf{x}]^{-1} \mathbf{x}^T (\mathbf{x} \theta + \mathbf{e})$$

$$\hat{\theta} = \theta + [\mathbf{x}^T \mathbf{x}]^{-1} \mathbf{x}^T \mathbf{e}$$

o que mostra que a polarização sobre a estimativa do vetor de parâmetros $\hat{\theta}$ é dado pelo termo:

$$p = E \left\{ [X^T X]^{-1} X^T e \right\}$$

para reduzir esse termo à zero, o que caracterizará um estimador não polarizado, é necessário reduzir e para uma seqüência de valores não correlatos, de tal modo que, os elementos de X , que consistem dos valores passados de U e V , se tornem independentes de e . Isso pode ser feito através de uma transformação, ou seja, por uma filtragem adequada nos valores de U e V .

Para realizar essa filtragem, o ruído e , presente na saída do processo, é estimado como:

$$\hat{e} \cong \eta = V - X \hat{\theta} \quad (4.11)$$

Esta aproximação é válida, desde que no limite, quando $\hat{\theta}$ se aproxima de θ , o erro residual η se aproxime do ruído e .

O ruído e presente na saída V do processo, em um determinado instante, pode ser representado por um modelo que dependa do valor de ξ_t , amostrado no instante considerado, e dos valores passados do ruído e , ou seja:

$$[1 + C(z^{-1})] e_t = \xi_t \quad (4.12)$$

ou

$$e_t = -C(z^{-1}) e_t + \xi_t$$

onde: $C(z^{-1}) = c_1 z^{-1} + c_2 z^{-2} + \dots + c_m z^{-m}$

comparando as equações (4.12) e (4.4), assume-se que [4]:

$$[1 + C(z^{-1})] = \left[\frac{[1 + A(z^{-1})][1 + B(z^{-1})]}{[1 + F(z^{-1})]} \right]^{-1} \quad (4.13)$$

Escrevendo a equação (4.12) na forma matricial, resulta:

$$e = -EC + \xi \quad (4.14)$$

$$\text{com: } E = [z^{-1}e \mid z^{-2}e \mid \dots \mid z^{-m}e]$$

$$C^T = [c_1 \quad c_2 \quad \dots \quad c_m]$$

Tal como para o processo, a estimativa do vetor de parâmetros do ruído C , será dado pela equação:

$$\hat{C} = -[E^T E]^{-1} E^T e \quad (4.15)$$

sendo \hat{C} , o vetor estimado para os parâmetros do ruído dado por:

$$\hat{C}^T = [\hat{c}_1 \quad \hat{c}_2 \quad \dots \quad \hat{c}_m]$$

O resultado obtido na equação (4.15) é não polarizado, quando os elementos de E são independentes de ξ , o qual representa uma seqüência de valores não correlatos (figura 4.2).

A estimativa dos parâmetros do modelo do ruído, permite que seja feita a desejada filtragem sobre os valores de U e V , produzindo no limite, quando \hat{C} se aproxima de C , um estimador não polarizado dos parâmetros θ do processo.

Isto pode ser visto, substituindo os elementos de u e v , na equação (4.4), por:

$$u_t^F = [1 + C(z^{-1})] u_t \quad (4.16)$$

$$v_t^F = [1 + C(z^{-1})] v_t$$

onde: v_t^F e u_t^F são os valores filtrados da saída e da entrada em um instante de amostragem

considerando ainda a aproximação feita na equação (4.13), a equação (4.4) se reduz a:

$$[1 + A(z^{-1})] v_t^F = z^{-k} B(z^{-1}) u_t^F + \xi_t \quad (4.17)$$

que pode ser escrita na forma matricial:

$$V^F = X^F \theta + \xi \quad (4.18)$$

sendo: $V^{FT} = [v_t^F \ v_{t+1}^F \ \dots \ v_{t+N}^F]$

$$U^{FT} = [u_t^F \ u_{t+1}^F \ \dots \ u_{t+N}^F]$$

$$X^F = [-z^{-1}V^F \ | \ \dots \ | -z^{-n}V^F \ | \ \dots \ | -z^{-k-1}U^F \ | \ \dots \ | -z^{-k-n}U^F]$$

Isso mostra que um estimador, não polarizado, do vetor de parâmetros do processo θ , pode ser obtido com:

$$\hat{\theta} = [X^{FT} X^F]^{-1} X^{FT} V^F \quad (4.19)$$

onde: X^F matriz de regressão dos valores passados de U e V , filtrados

Assim, a partir de uma estimativa inicial de θ , pode-se obter um estimador não polarizado dos parâmetros do sistema iteragindo as estimativas dos parâmetros do ruído (equação 4.15)

com a dos parâmetros do processo (equação 4.19). Esse processo iterativo é encerrado, quando a soma dos quadrados do erro residual η alcança um valor mínimo.

Devido às características do procedimento generalizado do método dos mínimos quadrados, a convergência dos parâmetros estimados não é assegurada, a não ser que muitas considerações restritivas sejam feitas. Contudo tem sido afirmado que o método é satisfatoriamente eficiente [4], e a convergência dos parâmetros estimados, para a vizinhança de seus valores verdadeiros, tem sido obtida em todas as experiências usando dados simulados.

4.3 Método Recursivo de Estimação

O método dos mínimos quadrados pode, em geral, ser transformado em um procedimento recursivo, para isso, é conveniente considerar o efeito de uma observação adicional (u_{t+N+1}, v_{t+N+1}) sobre a N -ésima estimativa dos vetores $\hat{\theta}$ e \hat{C} , que são baseados em uma seqüência de N observações (u, v) .

Mostrou-se que, a fim de se chegar a um estimador não polarizado dos parâmetros θ do processo, na presença de ruído, é necessário efetuar uma filtragem adequada nos valores de U e V , antes de proceder com a estimação. Seguindo a filosofia do processo iterativo do método generalizado dos mínimos quadrados, onde o ruído é representado pela equação (4.12); a filtragem sobre a observação adicional (u_{t+N+1}, v_{t+N+1}) é realizada, como na equação (4.16), usando os parâmetros do modelo do ruído obtidos na N -ésima estimativa do vetor \hat{C} , ou seja:

$$u_{t+N+1}^F = [1 + \hat{C}_N(z^{-1})] u_t \quad (4.20)$$

$$v_{t+N+1}^F = [1 + \hat{C}_N(z^{-1})] v_t$$

sendo: $\hat{C}_N (z^{-1}) = \hat{c}_{1N} z^{-1} + \hat{c}_{2N} z^{-2} + \dots + \hat{c}_{mN} z^{-m}$

$\hat{c}_{1N} \dots \hat{c}_{mN}$ parâmetros do ruído obtidos na N-ésima estimativa do vetor \hat{C}

Considerando os novos dados filtrados, a equação (4.18) pode ser reescrita, para uma seqüência de N+1 observações (u,v), como:

$$\begin{bmatrix} \frac{V^F}{v_{t+N+1}} \end{bmatrix} = \begin{bmatrix} \frac{X^F}{\mathcal{X}_{N+1}^{FT}} \end{bmatrix} \theta + \begin{bmatrix} \frac{\xi}{\xi_{t+N+1}} \end{bmatrix} \quad (4.21)$$

onde:

$$V^{FT} = [v_t^F \quad v_{t+1}^F \quad \dots \quad v_{t+N}^F]$$

$$U^{FT} = [u_t^F \quad u_{t+1}^F \quad \dots \quad u_{t+N}^F]$$

$$\xi^T = [\xi_t \quad \xi_{t+1} \quad \dots \quad \xi_{t+N}]$$

$$\theta^T = [a_1 \quad a_2 \quad \dots \quad a_n \quad b_1 \quad b_2 \quad \dots \quad b_n]$$

$$X^F = [-z^{-1}V^F \quad \dots \quad -z^{-n}V^F \quad | \quad -z^{-k-1}U^F \quad \dots \quad -z^{-k-n}U^F]$$

$$\mathcal{X}_{N+1}^{FT} = [-z^{-1}v_{t+N+1}^F \quad \dots \quad -z^{-n}v_{t+N+1}^F \quad -z^{-k-1}u_{t+N+1}^F \quad \dots \quad -z^{-k-n}u_{t+N+1}^F]$$

Da mesma forma que na equação (4.19) a N+1-ésima estimativa do vetor $\hat{\theta}$, baseado em uma seqüência de N+1 observações, é obtida da equação (4.21) como:

$$\hat{\theta}_{N+1} = \left[\begin{bmatrix} \frac{X^F}{\mathcal{X}_{N+1}^{FT}} \end{bmatrix}^T \begin{bmatrix} \frac{X^F}{\mathcal{X}_{N+1}^{FT}} \end{bmatrix} \right]^{-1} \begin{bmatrix} \frac{X^F}{\mathcal{X}_{N+1}^{FT}} \end{bmatrix}^T \begin{bmatrix} \frac{V^F}{v_{t+N+1}} \end{bmatrix}$$

$$\hat{\theta}_{N+1} = \left[\begin{array}{c} \left[X^{FT} \quad \dots \quad \mathcal{X}_{N+1}^F \right] \left[\begin{array}{c} X^F \\ \hline \mathcal{X}_{N+1}^{FT} \end{array} \right] \\ \hline \end{array} \right]^{-1} \left[\begin{array}{c} X^{FT} \quad \dots \quad \mathcal{X}_{N+1}^F \\ \hline \end{array} \right] \left[\begin{array}{c} V^F \\ \hline v_{t+N+1} \end{array} \right]$$

$$\hat{\theta}_{N+1} = \left[X^{FT} X^F + \mathcal{X}_{N+1}^F \mathcal{X}_{N+1}^{FT} \right]^{-1} \left[X^{FT} V^F + \mathcal{X}_{N+1}^F v_{t+N+1} \right]$$

onde: $\hat{\theta}_{N+1}$ é a N+1-ésima estimativa do vetor de parâmetros θ do processo

Fazendo uso da fórmula de Sherman-Morrison [7], demonstrada no apêndice A e transcrita pela equação (4.22),

$$\left[A^T A + a a^T \right]^{-1} = \left[A^T A \right]^{-1} - \frac{\left[A^T A \right]^{-1} a a^T \left[A^T A \right]^{-1}}{1 + a^T \left[A^T A \right]^{-1} a} \quad (4.22)$$

onde: A matriz do tipo $N \times 2n$
 a^T vetor do tipo $1 \times 2n$

resulta:

$$\hat{\theta}_{N+1} = \left[X^{FT} X^F \right]^{-1} X^{FT} V^F + \frac{\left[X^{FT} X^F \right]^{-1} \mathcal{X}_{N+1}^F \left[v_{t+N+1} - \mathcal{X}_{N+1}^{FT} \left[X^{FT} X^F \right]^{-1} X^{FT} V^F \right]}{1 + \mathcal{X}_{N+1}^{FT} \left[X^{FT} X^F \right]^{-1} \mathcal{X}_{N+1}^F}$$

porém da equação (4.19), vem:

$$\hat{\theta}_N = \left[X^{FT} X^F \right]^{-1} X^{FT} V^F$$

sendo, $\hat{\theta}_N$ a N-ésima estimativa do vetor de parâmetros do processo

portanto:

$$\hat{\theta}_{N+1} = \hat{\theta}_N + \frac{[X^{FT} X^F]^{-1} \mathcal{X}_{N+1}^F [v_{t+N+1} - \mathcal{X}_{N+1}^{FT} \hat{\theta}_N]}{1 + \mathcal{X}_{N+1}^{FT} [X^{FT} X^F]^{-1} \mathcal{X}_{N+1}^F}$$

que pode ser reescrita na forma:

$$\hat{\theta}_{N+1} = \hat{\theta}_N + G_N P_N \mathcal{X}_{N+1}^F [v_{t+N+1} - \mathcal{X}_{N+1}^{FT} \hat{\theta}_N] \quad (4.23)$$

onde:

$$P_N = [X^{FT} X^F]^{-1}$$

$$G_N = [1 + \mathcal{X}_{N+1}^{FT} P_N \mathcal{X}_{N+1}^F]^{-1}$$

$$\mathcal{X}^{FT} = [v_{t+N}^F \dots v_{t-n+N+1}^F \quad u_{t-k+N}^F \dots u_{t-k-n+N+1}^F]$$

A equação (4.23) possui a forma necessária para estimar os parâmetros do processo θ de maneira recursiva. O vetor de parâmetros $\hat{\theta}$, obtido na equação acima, não é polarizado, quando o termo correspondente ao ruído, na equação (4.21), é uma seqüência de valores aleatórios não correlatos.

A equação recursiva para determinar os parâmetros do ruído, é obtida levando-se em consideração que o ruído aproxima do erro residual entre a saída do processo e a saída do modelo do processo, quando no limite, $\hat{\theta}$ se aproxima de θ . Portanto o ruído presente na N+1-ésima observação da saída do processo pode ser estimado, usando os parâmetros do processo obtidos na N+1-ésima estimativa do vetor $\hat{\theta}$, como:

$$\hat{e}_{t+N+1} \cong v_{t+N+1} - \mathcal{X}^T \hat{\theta}_{N+1} \quad (4.24)$$

sendo:

$$\mathcal{X}^T = \left[-z^{-1}v_{t+N+1} \dots -z^{-n}v_{t+N+1} \quad -z^{-k-1}u_{t+N+1} \dots -z^{-k-n}u_{t+N+1} \right]$$

Reescrevendo a equação (4.18), para uma seqüência de N+1 observações, resulta:

$$\begin{bmatrix} \hat{e} \\ \dots \\ \hat{e}_{t+N+1} \end{bmatrix} = \begin{bmatrix} E \\ \dots \\ \mathbb{E}_{N+1}^T \end{bmatrix} C + \begin{bmatrix} \xi \\ \dots \\ \xi_{t+N+1} \end{bmatrix} \quad (4.25)$$

onde:

$$\hat{e} = [e_t \quad e_{t+1} \quad \dots \quad e_{t+N}]$$

$$\xi^T = [\xi_t \quad \xi_{t+1} \quad \dots \quad \xi_{t+N}]$$

$$C^T = [c_1 \quad c_2 \quad \dots \quad c_m]$$

$$E = [z^{-1} e \quad | \quad z^{-2} e \quad | \quad \dots \quad | \quad z^{-m} e]$$

$$\mathbb{E}_{N+1}^T = [z^{-1} e_{t+N+1} \quad | \quad z^{-2} e_{t+N+1} \quad | \quad \dots \quad | \quad z^{-m} e_{t+N+1}]$$

Da mesma forma que na equação (4.15) a N+1-ésima estimativa do vetor \hat{C} , baseado em uma seqüência de N+1 observações, é obtido da equação (4.25) como:

$$\hat{C}_{N+1} = \left[\begin{bmatrix} E \\ \dots \\ \mathbb{E}_{N+1}^T \end{bmatrix}^T \begin{bmatrix} E \\ \dots \\ \mathbb{E}_{N+1}^T \end{bmatrix} \right]^{-1} \begin{bmatrix} E \\ \dots \\ \mathbb{E}_{N+1}^T \end{bmatrix}^T \begin{bmatrix} \hat{e} \\ \dots \\ e_{t+N+1} \end{bmatrix}$$

Fazendo uso da equação (4.22) e desenvolvendo de modo análogo ao caso anterior, obtém-se a equação recursiva para a estimação dos parâmetros do ruído, dada por:

$$\hat{C}_{N+1} = \hat{C}_N - H_N R_N E_{N+1} [e_{t+N+1} - E_{N+1}^T \hat{C}_N] \quad (4.26)$$

onde: $R_N = [E^T E]^{-1}$

$$H_N = [1 + E_{N+1}^T R_N E_{N+1}]^{-1}$$

$$E^T = [e_{t+N} \quad e_{t+N+1} \quad \dots \quad e_{t-n+N+1}]$$

As matrizes P_N e R_N , nas equações (4.23) e (4.26), são atualizadas usando a equação (4.22) de forma que:

$$P_{N+1} = P_N - G_N P_N X_{N+1}^F X_{N+1}^{FT} P_N \quad (4.27)$$

$$R_{N+1} = R_N - H_N R_N E_{N+1} E_{N+1}^T R_N \quad (4.28)$$

Quando os parâmetros do sistema variam no tempo, há necessidade de alterar as equações obtidas, de modo que as últimas medidas influenciem mais que as medidas anteriores, ou seja, que as últimas medidas tenham um peso maior em relação as anteriores. Desta forma, evita-se que os parâmetros estimados venham a convergir, o que só é desejado para o caso de sistemas invariantes no tempo.

A convergência dos parâmetros estimados é normalmente refletida pela diminuição dos elementos da matriz P_N [8]. Assim, nos sistemas variantes no tempo, torna-se necessário evitar que os elementos da matriz P_N se tornem muito pequenos. Pois, caso tal fato aconteça, não haverá correções nos valores estimados dos parâmetros do processo, conforme pode ser observado

na equação (4.23). Uma forma simples de resolver este problema, consiste em introduzir um fator de esquecimento de modo a controlar a magnitude dos elementos da matriz P_N . Isto é possível fazendo nas equações (4.23) e (4.27):

$$P_N = \frac{P_N}{\rho}$$

onde: ρ é um parâmetro escalar, tal que : $0 < \rho < 1$

desta forma as equações (4.23) e (4.27) podem ser reescrita como:

$$\hat{\theta}_{N+1} = \hat{\theta}_N + G_N P_N X_{N+1}^F [v_{t+N+1} - X_{N+1}^{FT} \hat{\theta}_N] \quad (4.29)$$

$$P_{N+1} = \frac{1}{\rho} [P_N - G_N P_N X_{N+1}^F X_{N+1}^{FT} P_N] \quad (4.30)$$

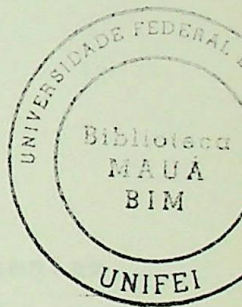
onde: $P_N = [X^{FT} X^F]^{-1}$

$$G_N = [\rho + X_{N+1}^{FT} P_N X_{N+1}^F]^{-1}$$

$$X^{FT} = [v_{t+N}^F \dots v_{t-n+N+1}^F \quad u_{t-k+N}^F \dots u_{t-k-n+N+1}^F]$$

Quando ρ diminui os elementos da matriz P_N aumentam a cada recursão, permitindo que os valores anteriores sejam esquecidos. O fator de esquecimento ρ influencia a velocidade de estimação dos parâmetros, sendo que, o valor mais conveniente para este, dependerá da ordem do processo e das espécies de distúrbios presente no processo. Na prática, procura-se um compromisso entre a velocidade de adaptação (elementos de P_N grandes), e a convergência dos parâmetros estimados (elementos de P_N pequenos).

De forma idêntica a estimação dos parâmetros do processo, a estimação dos parâmetros do ruído é descrita por:



$$\hat{C}_{N+1} = \hat{C}_N - H_N R_N E_{N+1} [e_{t+N+1} - E_{N+1}^T \hat{C}_N] \quad (4.31)$$

$$R_{N+1} = \frac{1}{\delta} [R_N - H_N R_N E_{N+1} E_{N+1}^T R_N] \quad (4.32)$$

onde: $R_N = [E^T E]^{-1}$

$$H_N = [\delta + E_{N+1}^T R_N E_{N+1}]^{-1}$$

$$E^T = [e_{t+N} \quad e_{t+N+1} \quad \dots \quad e_{t-n+N+1}]$$

δ é um parâmetro escalar, tal que : $0 < \delta < 1$

As equações descritas acima, permitem uma estimação "on-line" de parâmetros que variam no tempo. O algoritmo de estimação recursiva, para tal finalidade é descrito a seguir.

4.4 Algoritmo Recursivo de Estimação de Parâmetros

Os passos de "a" até "h" enumerados abaixo, apresentam de forma seqüencial as ações necessárias à implementação do algoritmo, recursivo, desenvolvido no item 4.3.

- a - filtrar a observação inicial (u_{t+N+1}, v_{t+N+1}) usando a equação (4.20)
- b - filtrar os elementos do vetor X_{N+1}^F usando os parâmetros do modelo do ruído, obtidos na N-ésima estimativa do vetor \hat{C} , assim:

$$X_{N+1}^{FT} = [1 + \hat{C}_N (z^{-1})] X_{N+1}^T$$

- c - atualizar $\hat{\theta}_N$ e P_N usando as equações (4.29) e (4.30), respectivamente.

- d - atualizar \mathcal{X}_{N+1}^T
- e - estimar o ruído presente na $N+1$ -ésima observação da saída do processo, usando a equação (4.24)
- f - atualizar \hat{C}_N e R_N usando as equações (4.31) e (4.32), respectivamente
- g - atualizar E_{N+1}^T
- h - repetir a seqüência acima

Para iniciar o algoritmo de estimação recursiva, pode-se fornecer uma estimativa inicial dos parâmetros dos vetores $\hat{\theta}$ e \hat{C} , sendo que, as matrizes P_N e R_N devem refletir, nesse caso, a confiança nos valores de $\hat{\theta}$ e \hat{C} , respectivamente. Outra forma de inicializar esse algoritmo consiste em determinar os valores iniciais dos vetores $\hat{\theta}$ e \hat{C} , por intermédio de cálculos que envolvam inversões matriciais.

Se os parâmetros dos vetores $\hat{\theta}$ e \hat{C} são completamente desconhecidos, estes podem ser feitos, arbitrariamente, iguais a zero. Neste caso, os elementos das matrizes P_N e R_N devem ter valores elevados, mas não o suficiente para causar uma instabilidade inicial.

Se existir uma aproximação inicial para os parâmetros dos vetores $\hat{\theta}$ e \hat{C} , esses valores podem ser usados como condição inicial destes vetores, com isto, a velocidade de convergência será melhorada. Neste caso, os elementos das matrizes P_N e R_N podem assumir valores pequenos, traduzindo a confiança inicial nos valores fornecidos para $\hat{\theta}$ e \hat{C} .

De uma forma geral, no início do algoritmo de estimação, as matrizes P_N e R_N , são escolhidas de modo a serem múltiplas da matriz identidade, ou seja:

$$P_N = \alpha I$$

$$R_N = \beta I$$

onde: I matriz identidade
 α e β são valores escalares que refletem a confiança nos valores iniciais de $\hat{\theta}$ e \hat{C} , respectivamente

Um melhor desempenho do algoritmo de estimação de parâmetros, tem sido obtido para valores de ρ compreendidos na faixa de 0,95 a 0,99. No entanto, valores de ρ menores do que a unidade ($\rho < 1$), podem levar a saída do processo à valores inaceitáveis na prática. Isto pode ser entendido de forma intuitiva da seguinte maneira: o termo negativo na equação (4.30), representa a redução na incerteza dos parâmetros estimados graças a última medida, quando não houver mudanças no sinal de entrada e nos parâmetros estimados, este termo será zero. Assim a equação (4.30) se reduz à forma:

$$P_{N+1} = \frac{1}{\rho} P_N$$

e a matriz P_N crescerá exponencialmente se $\rho < 1$, podendo gerar problemas numéricos na estimação dos parâmetros.

Uma das formas de contornar o problema, consiste em modificar o algoritmo de estimação. Uma possibilidade é parar de atualizar a matriz P_N , quando o erro entre a saída do processo e a saída do modelo estimado para o processo, for menor do que um dado valor. Outra possibilidade é escolher um fator de esquecimento, tal que uma função de P_N , como por exemplo, o traço da matriz P_N , seja constante.

Deve-se observar que as últimas considerações também se aplicam ao valor de δ na estimação dos parâmetros do ruído.

SIMULAÇÕES

5.1 Introdução

Visando a implementação prática do algoritmo de estimação de parâmetros apresentado no capítulo anterior, e da técnica de alocação de pólos (Pole Assignment) descrita no capítulo 3, foram realizadas simulações, considerando que o processo a ser controlado fosse descrito por uma função de transferência, discretizada, na forma:

$$F(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (5.1)$$

onde: a_1 , a_2 , b_1 e b_2 são parâmetros que caracterizam o comportamento transitório do processo

A seguir são apresentados alguns dos resultados obtidos nas simulações do algoritmo de estimação de parâmetros, e na simulação da estrutura de controle auto-ajustável proposta neste trabalho.

5.2 Simulação do Algoritmo de Estimação de Parâmetros

O algoritmo de estimação de parâmetros, que emprega a técnica dos mínimos quadrados recursivo, permite que os parâmetros de um processo sejam estimados simultaneamente com a evolução do processo, independentemente dos parâmetros serem ou não variantes no tempo.

Os parâmetros a serem estimados são obtidos por intermédio das equações recursivas (4.29) e (4.30), que constituem a base do algoritmo de estimação. Uma vez estimados os parâmetros, pode-se levantar um modelo matemático discretizado, para o processo, por meio da equação (4.6).

A fim de verificar o desempenho do algoritmo de estimação de parâmetros, considere inicialmente um processo descrito pela função de transferência (5.1), onde os parâmetros a_1 , a_2 , b_1 e b_2 variam segundo os dados da tabela 5.1.

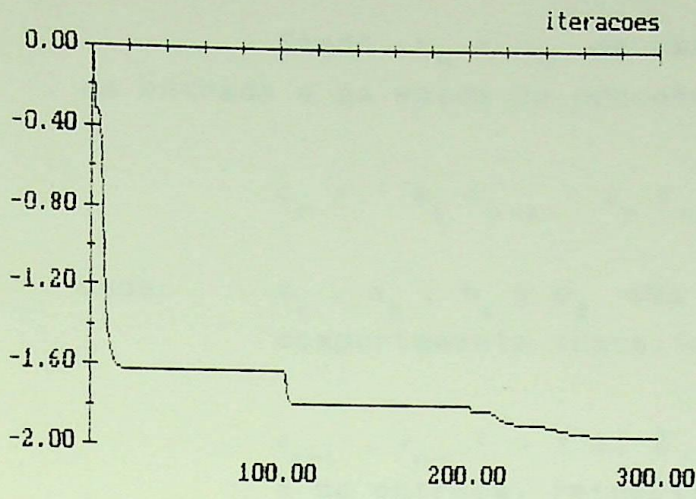
Número de iterações	a_1	a_2	b_1	b_2
início	-1.6274	0.6780	0.3419	-0.2913
80	-1.7968	0.8226	0.1858	-0.1599
180	-1.9480	0.9501	0.0479	-0.0457

Tabela 5.1

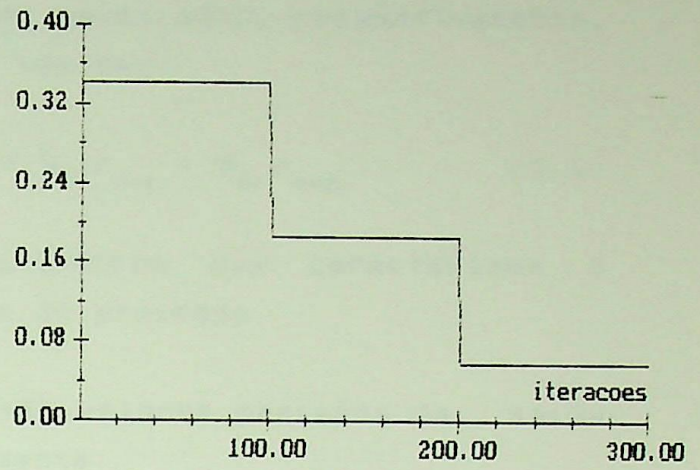
A evolução do algoritmo de estimação de parâmetros, para as variações da tabela 5.1, é apresentada na figura 5.1. Nessa figura \hat{a}_1 , \hat{a}_2 , \hat{b}_1 e \hat{b}_2 , são os valores estimados pelo algoritmo para os parâmetros a_1 , a_2 , b_1 e b_2 , respectivamente.

Deve-se observar que o algoritmo de estimação somente passou a identificar os parâmetros a_1 , a_2 , b_1 e b_2 , após a ocorrência de perturbações no sistema, o que se deu no início do processo e nos instantes em que o algoritmo realizou a 100^a e a 200^a iteração. Essas perturbações foram produzidas por um sinal de referência na forma indicada pela figura 5.2.

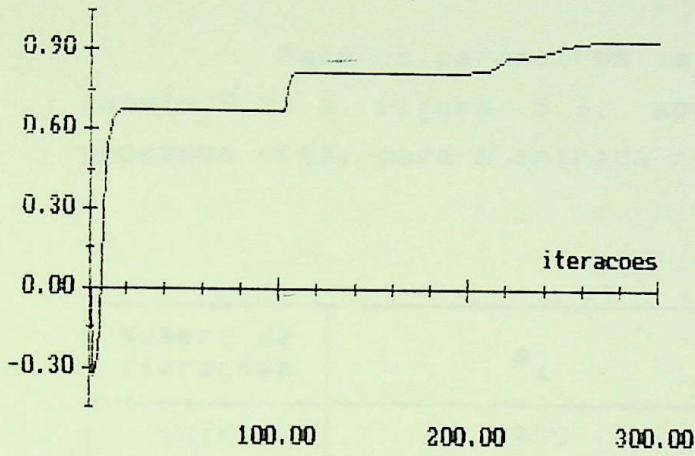
O período de amostragem usado na simulação do algoritmo de estimação de parâmetros, foi igual a 20 [ms], que corresponde ao tempo necessário para o algoritmo efetuar uma iteração.



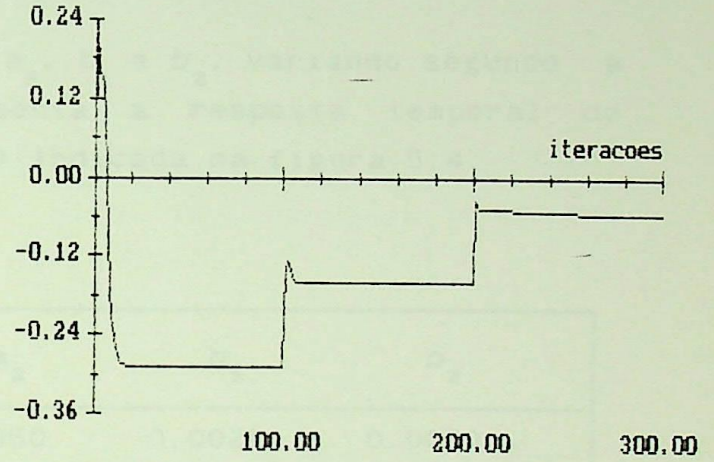
(a) Parâmetro \hat{a}_1



(c) Parâmetro \hat{b}_1



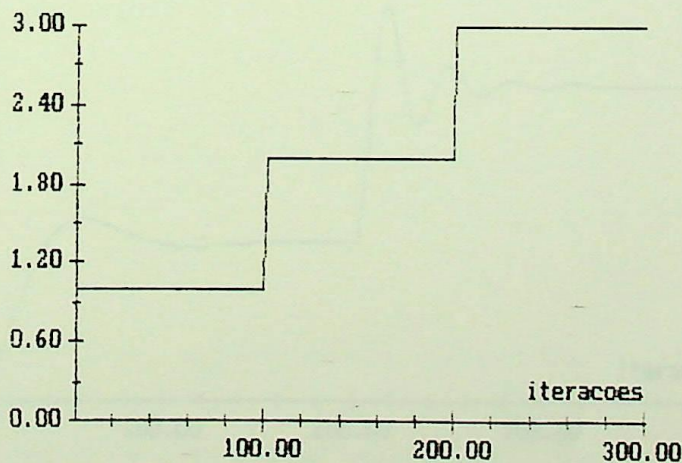
(b) Parâmetro \hat{a}_2



(d) Parâmetro \hat{b}_2

EVOLUÇÃO DO ALGORITMO DE ESTIMAÇÃO DE PARAMETROS

Figura 5.1



SINAL DE REFERENCIA

Figura 5.2

Sendo r_n e c_n os valores amostrados, respectivamente, na entrada e na saída do processo, tem-se:

$$c_n = -a_1 c_{n-1} - a_2 c_{n-2} + b_1 r_{n-1} + b_2 r_{n-2} \quad (5.2)$$

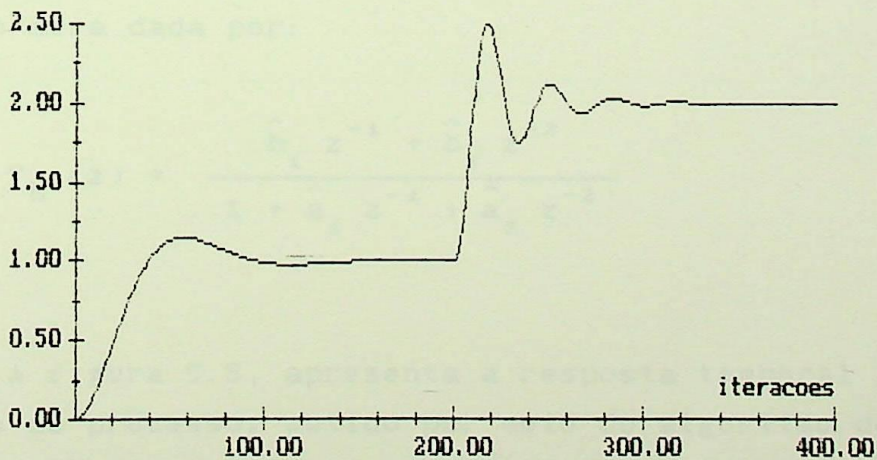
onde: a_1 , a_2 , b_1 e b_2 são parâmetros que caracterizam o comportamento transitório do processo

c_{n-i} , r_{n-i} ($i = 1, 2$) são valores passados da saída e da entrada, respectivamente

Para os parâmetros a_1 , a_2 , b_1 e b_2 , variando segundo a tabela 5.2, a figura 5.3, apresenta a resposta temporal do processo $c(t)$, para a entrada $r(t)$ indicada na figura 5.4.

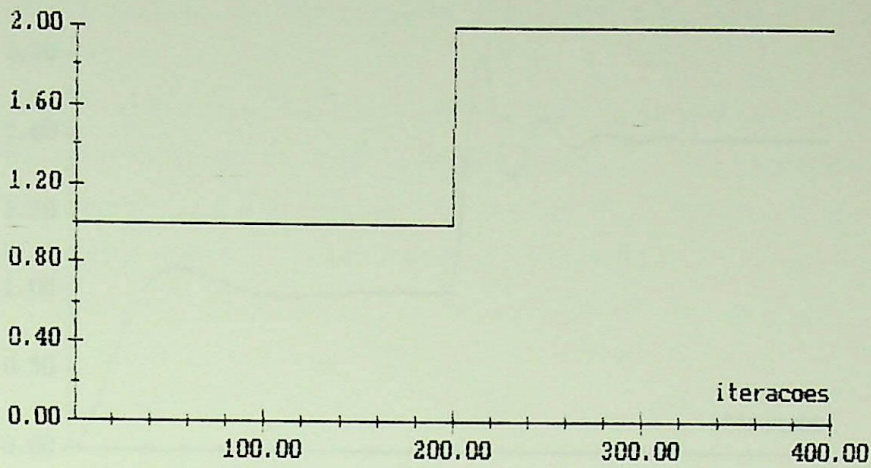
Número de iterações	a_1	a_2	b_1	b_2
início	-1.9309	0.9350	0.0021	0.0020
180	-1.8831	0.9194	0.0104	0.0179

Tabela 5.2



RESPOSTA TEMPORAL DO PROCESSO SIMULADO

Figura 5.3



SINAL DE REFERENCIA DO PROCESSO SIMULADO

Figura 5.4

Neste último caso o modelo matemático do processo, cujos parâmetros serão estimados por meio do algoritmo dos mínimos quadrados recursivo, será da forma:

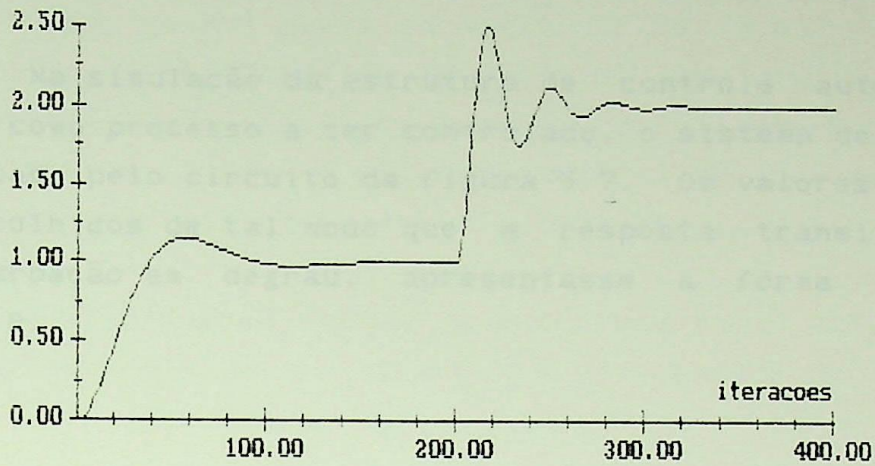
$$c_n = -\hat{a}_1 c_{n-1} - \hat{a}_2 c_{n-2} + \hat{b}_1 r_{n-1} + \hat{b}_2 r_{n-2} \quad (5.3)$$

onde: \hat{a}_1 , \hat{a}_2 , \hat{b}_1 e \hat{b}_2 são os valores estimados para os parâmetros a_1 , a_2 , b_1 e b_2 , respectivamente

Por conseguinte, a função de transferência do modelo matemático será dada por:

$$F_M(z) = \frac{\hat{b}_1 z^{-1} + \hat{b}_2 z^{-2}}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2}} \quad (5.4)$$

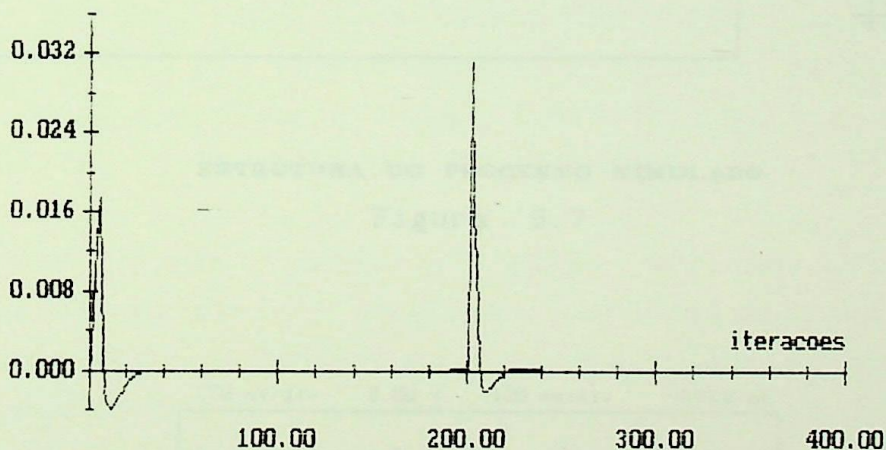
A figura 5.5, apresenta a resposta temporal do modelo matemático do processo, obtido por meio do algoritmo de estimação de parâmetros, para a entrada indicada na figura 5.4. Os parâmetros a_1 , a_2 , b_1 e b_2 , variam no tempo de acordo com os valores da tabela 5.2,



RESPOSTA DO MODELO MATEMATICO DO PROCESSO SIMULADO

Figura 5.5

O erro entre a saída real do processo e a saída do modelo estimado, é apresentado na figura 5.6.



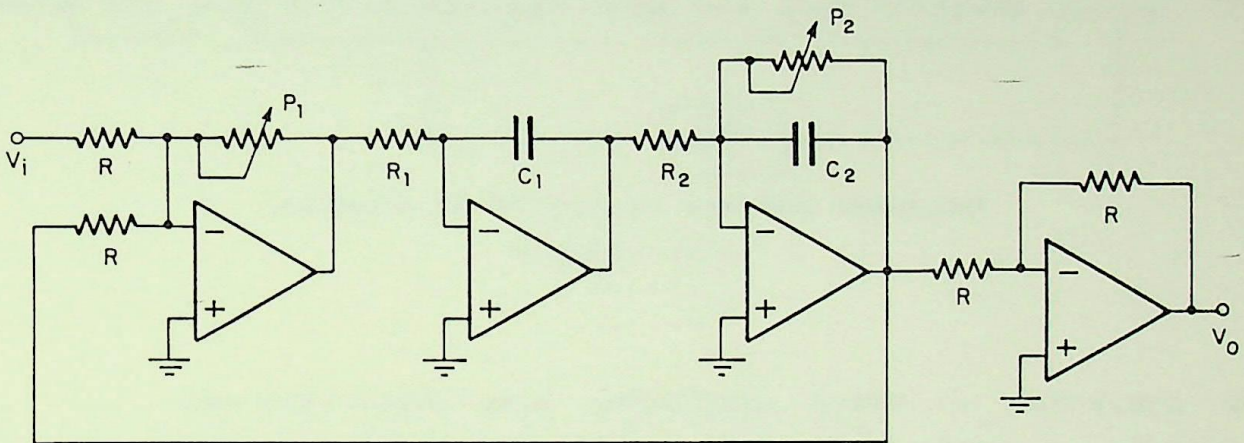
ERRO ENTRE A SAIDA DO PROCESSO E DO MODELO

Figura 5.6

No caso simulado acima, o fator de esquecimento ρ , utilizado na equação (4.30), foi feito igual a 0.75 ($\rho = 0.75$). O período de amostragem foi de 20 [ms], e o efeito do ruído sobre o processo não foi considerado.

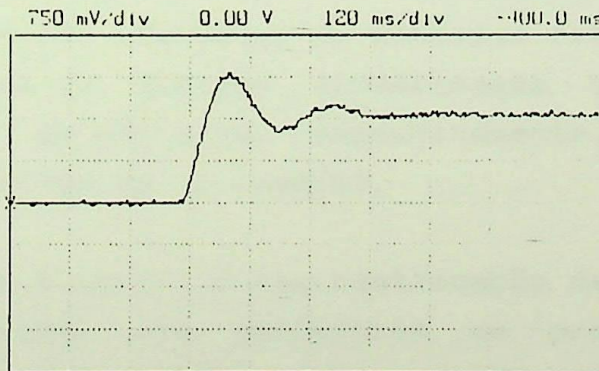
5.3 Simulação da Estrutura de Controle Auto-Ajustável

Na simulação da estrutura de controle auto-ajustável, tomou-se como processo a ser controlado, o sistema de 2^a ordem, implementado pelo circuito da figura 5.7. Os valores de R_i e C_i foram escolhidos de tal modo que a resposta transitória, para uma perturbação em degrau, apresentasse a forma indicada na figura 5.8.



ESTRUTURA DO PROCESSO SIMULADO

Figura 5.7



RESPOSTA DO PROCESSO A UMA ENTRADA EM DEGRAU

Figura 5.8

Para efetuar a amostragem dos sinais de entrada e de saída, utilizou-se uma placa de aquisição de dados. Esta placa, além de todos os subsistemas necessários ao seu funcionamento (fontes, sample hold, multiplexadores, etc), possuía ainda um conversor A/D de 12 bits e um conversor D/A de 10 bits, cada qual podendo comportar até oito canais multiplexados.

O diagrama em blocos da estrutura final utilizada na simulação do controle auto-ajustável, é apresentado na figura 5.9.

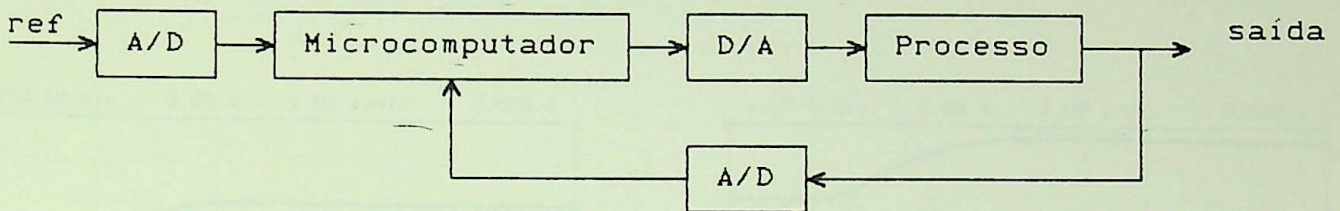


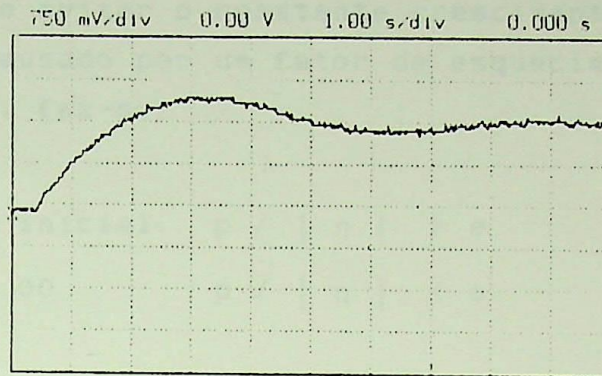
DIAGRAMA EM BLOCOS DO SISTEMA SIMULADO

Figura 5.9

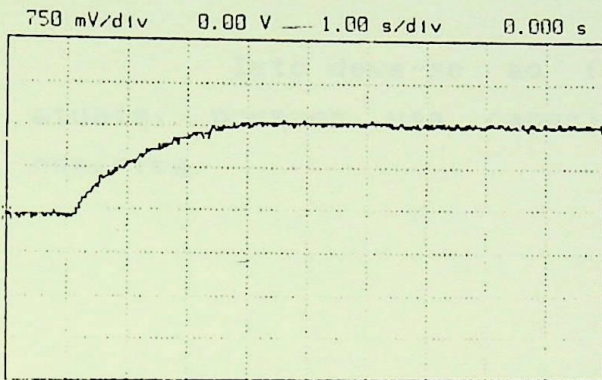
Uma explicação mais detalhada sobre a estrutura de controle proposta por esse trabalho, é deixada para o capítulo seguinte, onde descreve-se a implementação prática do controle auto-ajustável, aplicado a um processo real.

As figuras 5.10(a) e 5.10(b), mostram a resposta do sistema auto-ajustável (figura 5.9), para uma entrada em degrau, após a estimação dos parâmetros e o deslocamento dos pólos do processo sob controle. Os pólos do processo foram deslocados, de modo que a resposta do sistema apresentasse uma ultrapassagem máxima (overshoot) de 15% e 4%, respectivamente, com frequência natural não amortecida de 1 [rad/s].

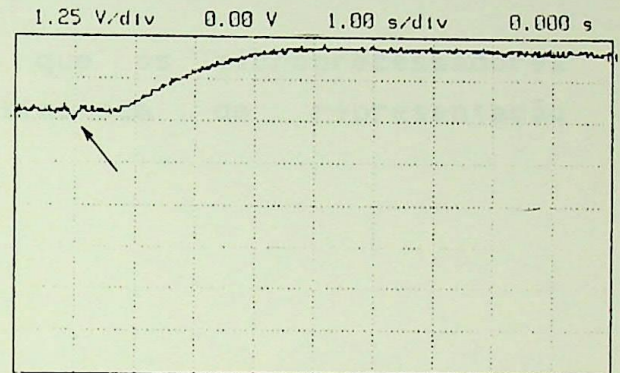
A figura 5.10(c), é uma continuação da figura 5.10(b), porém com uma variação dos parâmetros do processo, antes da ocorrência de um novo degrau na entrada. A variação dos parâmetros deu-se pela modificação de valores de resistência e capacitância no circuito da figura 5.7.



(a) overshoot de 15%



(b) overshoot de 4%



(c) variação de parâmetros

RESPOSTA DO CONTROLE AUTO-AJUSTAVEL

Figura 5.10

No caso acima, o fator de esquecimento ρ , utilizado na equação (4.30), foi feito igual a 0.75 ($\rho = 0.75$), e o efeito do ruído sobre o processo não foi considerado.

Em todos os casos simulados anteriormente, o valor inicial da matriz P_N nas equações (4.29) e (4.30), usado para refletir a confiança nos valores iniciais dos parâmetros estimados, foi igual a $P_N = \alpha I$, onde $\alpha = 1000$ e I é a matriz identidade.

A fim de evitar o constante crescimento dos elementos da matriz P_N , causado por um fator de esquecimento menor do que a unidade ($\rho < 1$), fez-se:

$$\rho = \begin{cases} \rho \text{ inicial} & p / |\eta| > e \\ 1.00 & p / |\eta| < e \end{cases}$$

onde: η é o erro entre a saída do processo e a saída do modelo matemático estimado
 e valor de erro limite

Isto deve-se ao fato de que os microprocessadores atuais, possuem uma capacidade limitada de representação numérica.

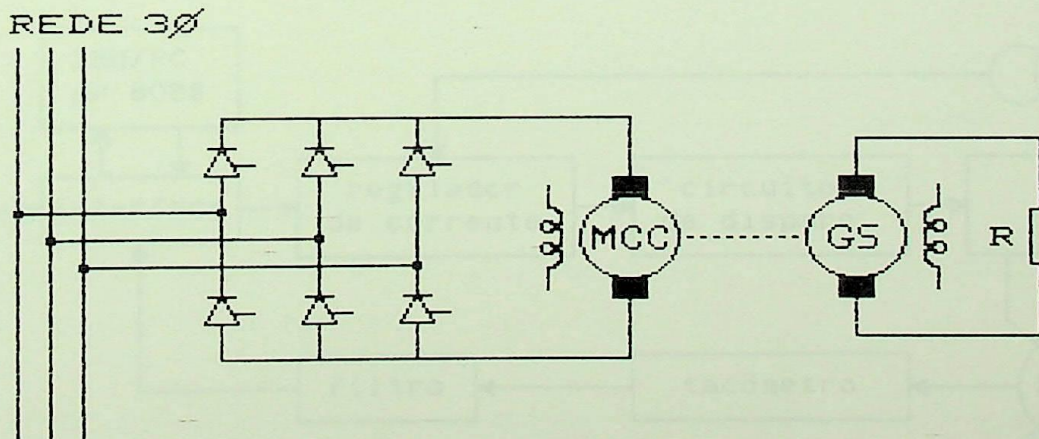
IMPLEMENTAÇÃO DO CONTROLE
AUTO - AJUSTAVEL

6.1 Introdução

O sistema de controle auto-ajustável (self-tuning), proposto neste trabalho, foi aplicado no controle de velocidade de um motor de corrente contínua, acionado por conversor tiristorizado. O motor de corrente contínua utilizado possuía as seguintes características:

potência	1 Hp		
tensão	220 V	resistência	3.07 Ω
corrente a vazio	0.5 A	indutância	34.2 mH
corrente nominal	4.2 A	velocidade	1800 rpm

Como carga para o motor de corrente contínua, foi acoplado no seu eixo, um gerador síncrono de 1 HP alimentando uma carga puramente resistiva. A figura 6.1, apresenta o esquema utilizado na montagem entre conversor tiristorizado, motor de corrente contínua e carga.



MOTOR DE CORRENTE CONTINUA ACIONADO
POR CONVERSOR TIRISTORIZADO

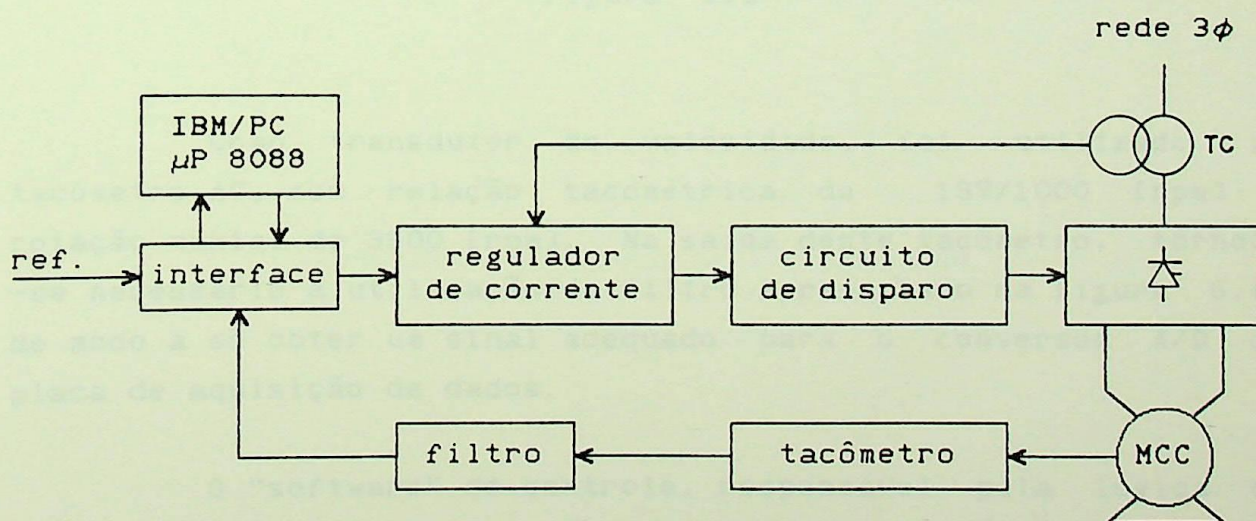
Figura 6.1

Toda a lógica do controle auto-ajustável foi realizada por um IBM/PC, com CPU 8088 trabalhando a 4.77 MHz e co-processador aritmético 8087. A comunicação do IBM/PC com o sistema a ser controlado foi feita por intermédio de uma placa de aquisição de dados, que possuía um conversor A/D de 12 bits e um conversor D/A de 10 bits, cada qual podendo comportar até oito canais multiplexados.

A figura 6.2, fornece um esquema da malha de controle auto-ajustável, para o motor de corrente contínua.

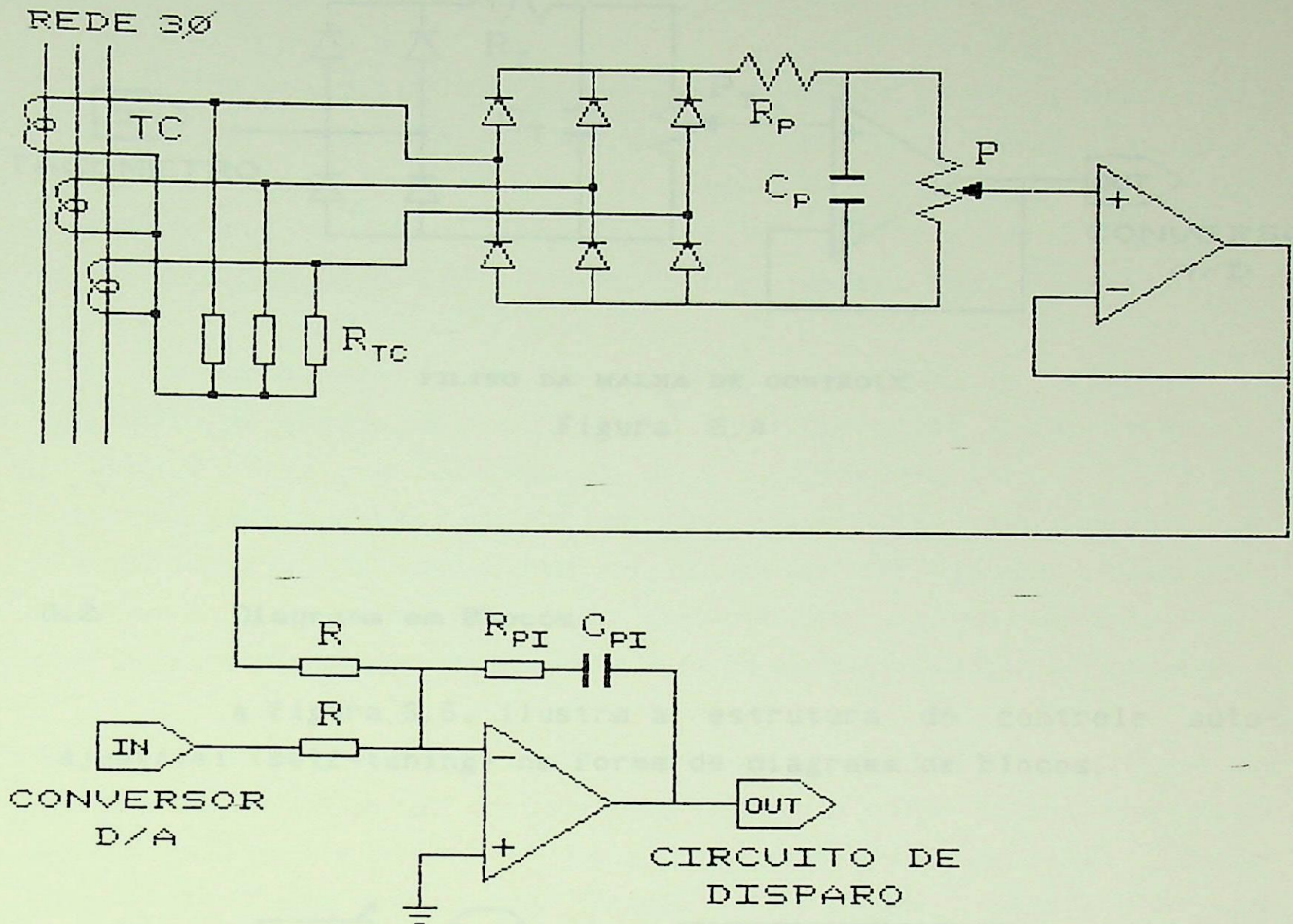
O regulador de corrente é um regulador PI, conforme ilustrado na figura 6.3. Os resistores conectados nos terminais dos transformadores de corrente (TC), são exclusivamente para proteção e não afetam a precisão da malha de corrente.

Toda a lógica de disparo do conversor tiristorizado foi deixado a cargo de um módulo dedicado, desenvolvido nos laboratórios de eletrônica da Escola Federal de Engenharia de Itajubá - EFEI.



ESQUEMA DA MALHA DE CONTROLE

Figura 6.2

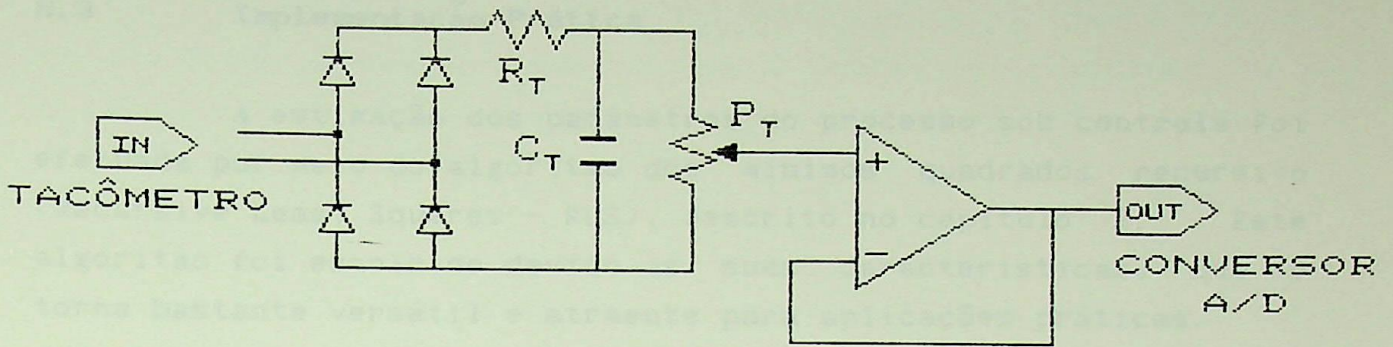


REGULADOR DE CORRENTE

Figura 6.3

Como transdutor de velocidade, foi utilizado um tacômetro AC, com relação tacométrica de $19V/1000$ [rpm] e rotação máxima de 3600 [rpm]. Na saída deste tacômetro, tornou-se necessário a utilização do filtro apresentado na figura 6.4, de modo a se obter um sinal adequado para o conversor A/D da placa de aquisição de dados.

O "software" de controle, responsável pela lógica do sistema implementado, foi totalmente desenvolvido em linguagem C e leva em torno de 20 [ms] para efetuar cada iteração. Este "software" é analisado com mais detalhes no capítulo 7.



FILTRO DA MALHA DE CONTROLE

Figura 6.4

6.2 Diagrama em Blocos

A figura 6.5, ilustra a estrutura do controle auto-ajustável (self-tuning) na forma de diagrama de blocos.

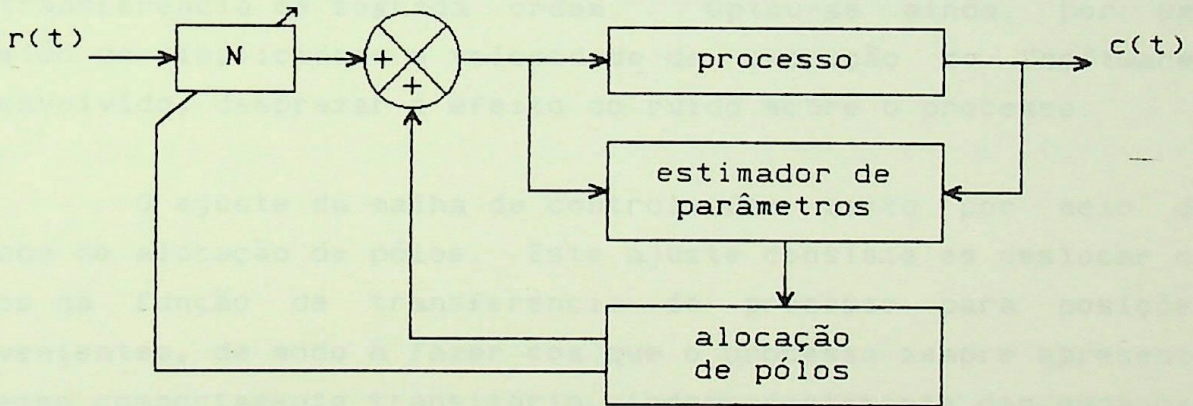


DIAGRAMA EM BLOCOS DA ESTRUTURA DE CONTROLE AUTO-AJUSTAVEL

Figura 6.5

Na figura acima o bloco referente ao processo, representa toda a malha física do controle (ponte retificadora, motor de corrente contínua, etc), enquanto que os demais blocos foram implementados, por meio de "software" no IBM/PC.

6.3 Implementação Prática

A estimação dos parâmetros do processo sob controle foi efetuada por meio do algoritmo dos mínimos quadrados recursivo (Recursive Least Squares - RLS), descrito no capítulo 2. Este algoritmo foi escolhido devido as suas características, que o torna bastante versátil e atraente para aplicações práticas.

Após ocorrer uma perturbação na entrada do sistema, o algoritmo de estimação levou de 10 a 20 iterações para convergir, de modo a obter uma nova função de transferência para o processo. A necessidade da presença de perturbações no sistema, resulta de uma característica estrutural do algoritmo dos mínimos quadrados recursivo, que não consegue identificar os parâmetros sob condições estacionárias, embora esteja permanentemente tentando.

Como o sistema auto-ajustável é aplicado no controle de velocidade de um motor de corrente contínua, foi escolhido como modelo base à ser estimado, para o motor mais carga, uma função de transferência de segunda ordem. Optou-se ainda, por uma questão de simplicidade e velocidade de execução do "software" desenvolvido, desprezar o efeito do ruído sobre o processo.

O ajuste da malha de controle foi feito por meio do método de alocação de pólos. Este ajuste consiste em deslocar os pólos da função de transferência do processo para posições convenientes, de modo a fazer com que o processo sempre apresente o mesmo comportamento transitório, independentemente das mudanças que possam vir a ocorrer em seus parâmetros.

Uma vez que para implementar a técnica de alocação de pólos se faz necessário, no nosso caso, o conhecimento prévio das variáveis de estado do processo sob controle, utilizou-se a função de transferência estimada do processo para obter, de forma matemática, as variáveis de estado necessárias. Deve-se observar que o modelo estimado para o processo, é considerado igual ao seu modelo real.

A forma para se obter matematicamente as variáveis de estado de um processo, a partir de sua função de transferência, e a técnica de alocação de pólos, foram descritas nos capítulos 2 e 3, respectivamente.

A ação de controle foi realizada por intermédio da modificação do sinal de entrada do processo e pela adaptação da constante N , representada no diagrama de blocos da figura 6.5. Um caso alternativo seria a modificação dos parâmetros de um regulador a ser incluído na malha de controle.

O método de estimação de estados do processo, associado à técnica de alocação de pólos, constitui o mecanismo de adaptação usado no ajuste do comportamento global da malha de controle. A escolha por esta filosofia de controle, visa a facilidade de implementação prática.

Como não se tinha uma estimativa inicial para os parâmetros do processo, a inicialização do mesmo deu-se de forma lenta e gradual, até se obter uma função de transferência que o representasse. Nessa etapa o valor da constante N , apresentada na figura 6.5, foi feita igual a um ($N = 1$). Embora, não tenha sido implementado na prática, poderia-se armazenar os parâmetros estimados do processo, para reutilizá-los numa futura inicialização do sistema.

A localização desejada para os pólos do processo, a fim de se ter um comportamento transitório desejado, foi dada pela função de transferência (6.1), visto que, o processo sob controle era estimado por um modelo matemático, também de segunda ordem.

$$F(S) = \frac{\omega_n^2}{S^2 + 2 \zeta \omega_n S + \omega_n^2} \quad (6.1)$$

onde: ζ coeficiente de amortecimento
 ω_n frequência natural não amortecida

Quando a equação (6.1) é discretizada, resulta em:

$$F(Z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6.2)$$

onde: z operador deslocamento

$$a_1 = -2 e^{-\sigma T} \cos \omega_d T$$

$$a_2 = e^{-2\sigma T}$$

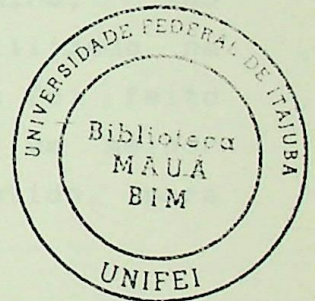
$$b_1 = \begin{cases} 1 - e^{-\sigma T} \cos \omega_d T - \frac{\sigma}{\omega_d} e^{-\sigma T} \sin \omega_d T & p / \xi \neq 1 \\ 1 - e^{-\sigma T} - \sigma T e^{-\sigma T} & p / \xi = 1 \end{cases}$$

$$b_2 = \begin{cases} e^{-2\sigma T} - e^{-\sigma T} \cos \omega_d T + \frac{\sigma}{\omega_d} e^{-\sigma T} \sin \omega_d T & p / \xi \neq 1 \\ e^{-2\sigma T} - e^{-\sigma T} + \sigma T e^{-\sigma T} & p / \xi = 1 \end{cases}$$

com: $\sigma = \xi \omega_n$

$$\omega_d = \omega_n \sqrt{1 - \xi^2}$$

T = período de amostragem



Uma vez especificado o comportamento transitório desejado para o processo, considerando naturalmente as suas limitações físicas, é possível obter uma função de transferência do tipo dado pela equação (6.1), cujos coeficientes de sua transformada Z , podem ser calculados (equação 6.2).

A partir da representação, na forma canônica controlável, da função de transferência, discretizada, que especifica o comportamento transitório desejado para o processo, obtém-se uma equação de estado na forma da equação (3.3), que é

transcrita pela equação (6.3). Através da técnica de alocação de pólos, determina-se um vetor de ganho K , a ser utilizado na realimentação das variáveis de estado do processo, de modo a deslocar seu pólos para as posições desejadas.

$$\begin{aligned} X(k+1) &= A_M X(k) + B_M U(k) \\ Y_M(k) &= C_M X(k) \end{aligned} \quad (6.3)$$

O vetor de ganho K é dado por:

$$K = [k_1 \quad k_2 \quad \dots \quad k_n]$$

onde: n representa a ordem do processo

6.4 Resultados Práticos

Alguns dos resultados da implementação prática do controle auto-ajustável, proposto neste trabalho, são apresentados a seguir. O fator de esquecimento ρ utilizado na equação (4.30) do algoritmo de estimação de parâmetros foi feito igual a 0.75 ($\rho = 0.75$), embora sabendo, que um melhor desempenho do algoritmo de estimação, tem sido obtido, para valores de ρ compreendidos na faixa de 0.95 a 0.99.

Nos gráficos, obtidos com auxílio de um osciloscópio digital, a curva que aparece na parte superior corresponde ao sinal de referência que foi aplicado ao sistema, e a curva na parte inferior corresponde à resposta do sistema ao sinal de referência aplicado.

Inicialmente, apresenta-se a resposta a um degrau para diferentes características da planta, e diferentes especificações de comportamento transitório. As diferentes características da planta são obtidas por variação da tensão do campo do motor de corrente contínua.

A figura 6.6, mostra o caso em que os pólos estimados para o processo são deslocados de forma a se ter sempre uma resposta ao degrau com ultrapassagem máxima (overshoot) de 4% e tempo de acomodação, para faixa de erro de $\pm 2\%$, igual a 30 [s], independentemente das características da planta.

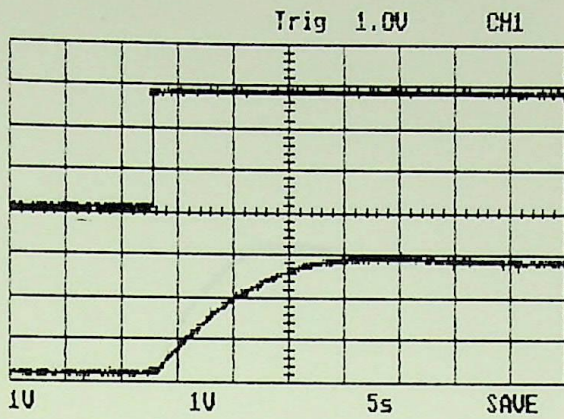
Nas figuras 6.7, 6.8 e 6.9, os pólos foram deslocados para posições de forma que o tempo de acomodação da resposta ao degrau, para faixa de erro de $\pm 2\%$, também fosse igual a 30 [s], porém com uma ultrapassagem máxima de 15%, 30% e 40%, respectivamente.

No caso da figura 6.10, os pólos foram deslocados de modo que a resposta ao degrau tenha um amortecimento crítico [9], e constante de tempo de 7.5 [s].

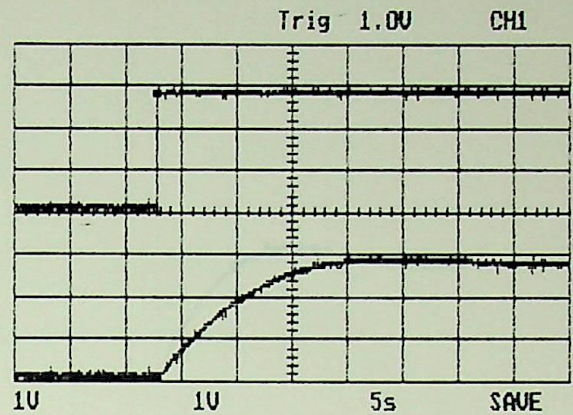
A tabela 6.1, fornece um resumo das curvas de resposta ao degrau apresentadas nas figuras mencionadas acima.

OVERSHOOT	FIGURA				
4 %	6.6 a	6.6 b	6.6 c	6.6 d	6.6 e
15 %	6.7 a	6.7 b	6.7 c	6.7 d	6.7 e
30 %	6.8 a	6.8 b	6.8 c	6.8 d	6.8 e
40 %	6.9 a	6.9 b	6.9 c	6.9 d	6.9 e
amort. crit.	6.10 a	6.10 b	6.10 c	6.10 d	6.10 e
CAMPO [V]	220	110	50	220	110
CARGA	VAZIO			NOMINAL	

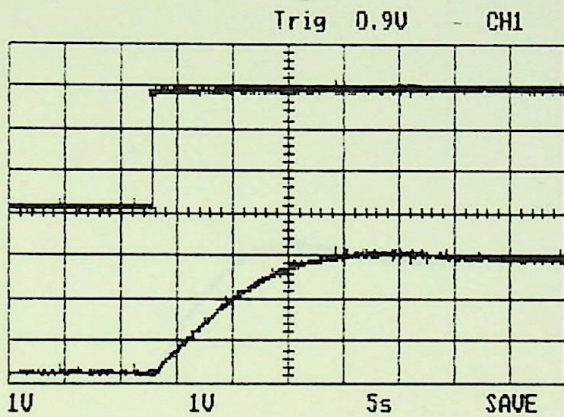
Tabela 6.1



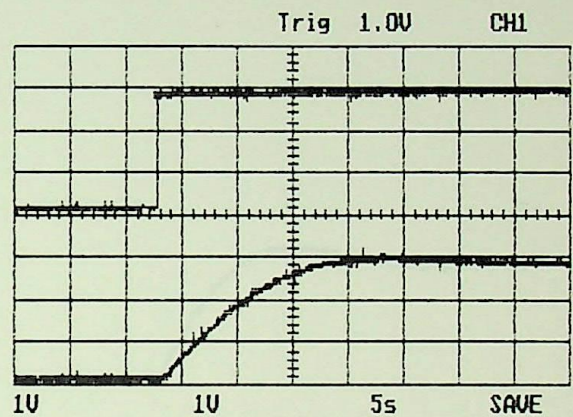
(a) motor a vazio
tensão do campo = 220 [V]



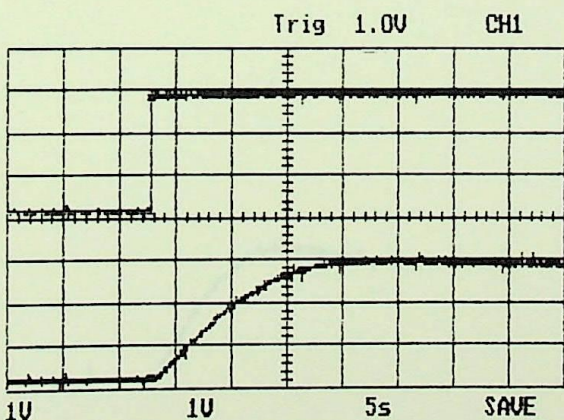
(d) motor com carga nominal
tensão do campo = 220 [V]



(b) motor a vazio
tensão do campo = 110 [V]



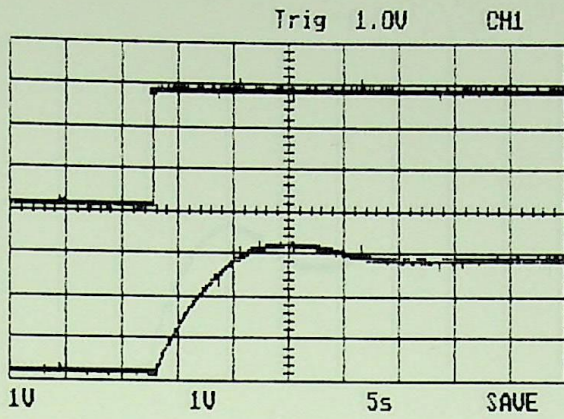
(e) motor com carga nominal
tensão do campo = 110 [V]



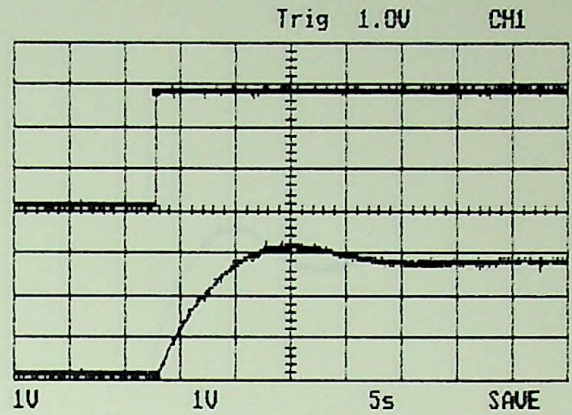
(c) motor a vazio
tensão do campo = 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 4% E TEMPO DE ACOMODAÇÃO DE 30(S)

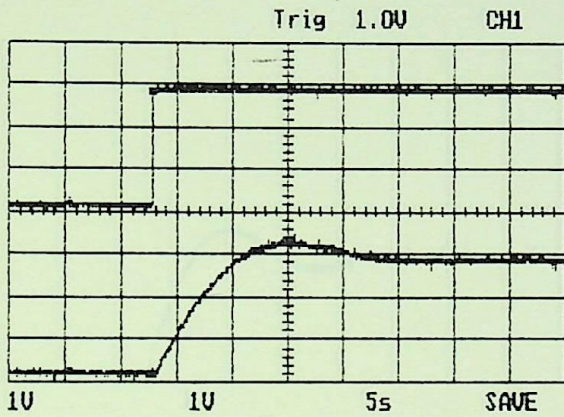
Figura 6.6



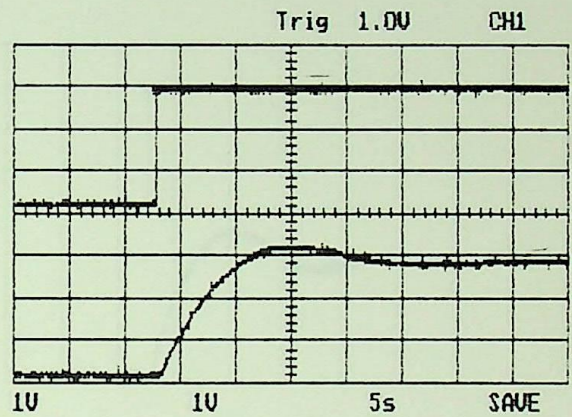
(a) motor a vazio
tensão do campo = 220 [V]



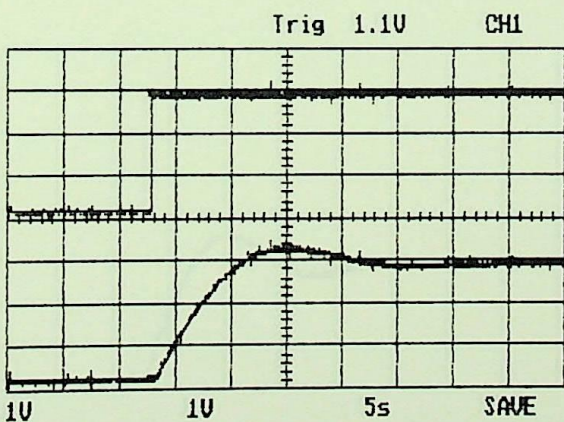
(d) motor com carga nominal
tensão do campo = 220 [V]



(b) motor a vazio
tensão do campo = 110 [V]



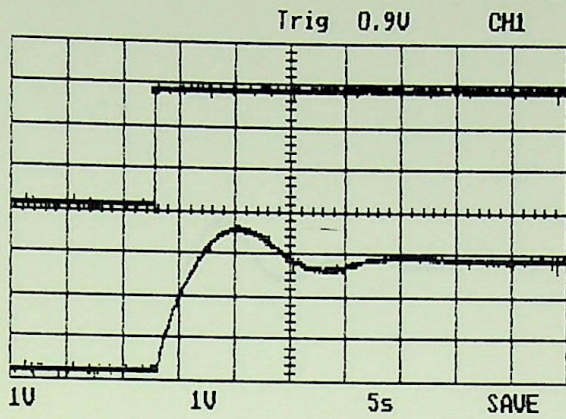
(e) motor com carga nominal
tensão do campo = 110 [V]



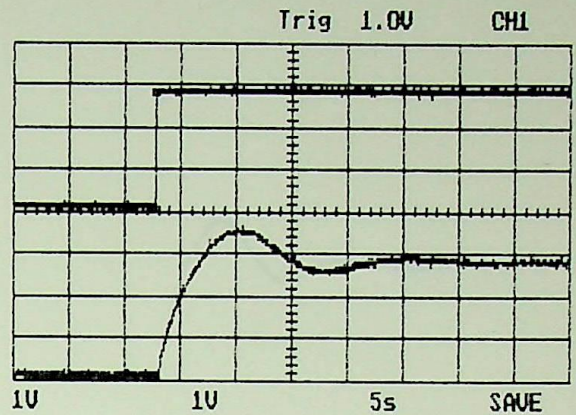
(c) motor a vazio
tensão do campo = 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 15% E TEMPO DE ACOMODAÇÃO DE 30(S)

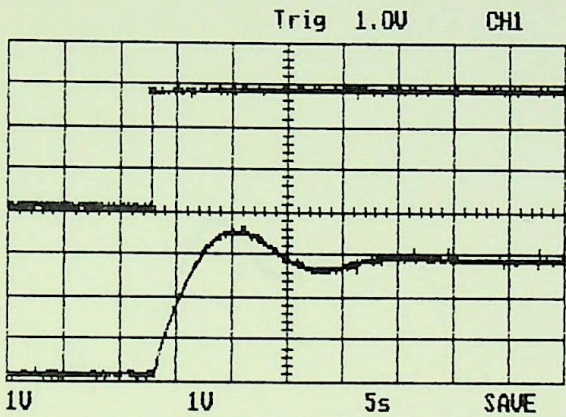
Figura 6.7



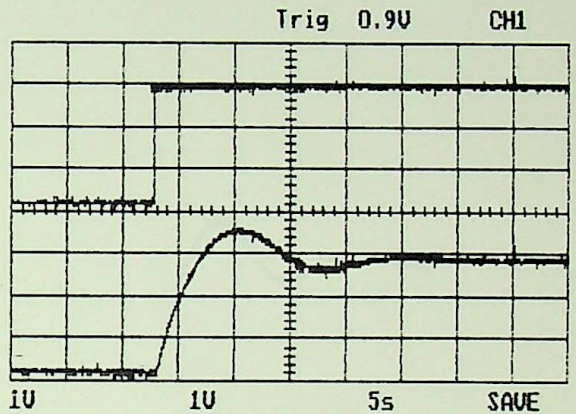
(a) motor a vazio
tensão do campo = 220 [V]



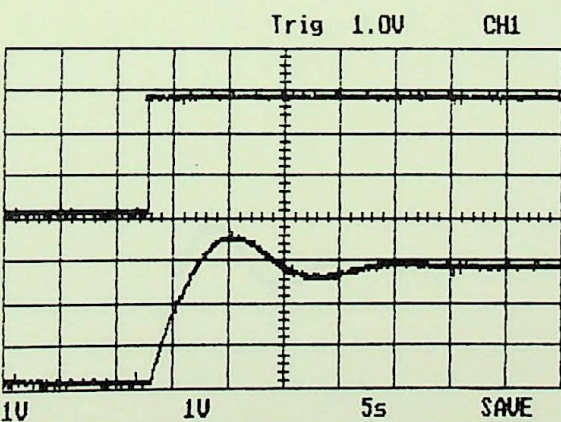
(d) motor com carga nominal
tensão do campo = 220 [V]



(b) motor a vazio
tensão do campo = 110 [V]



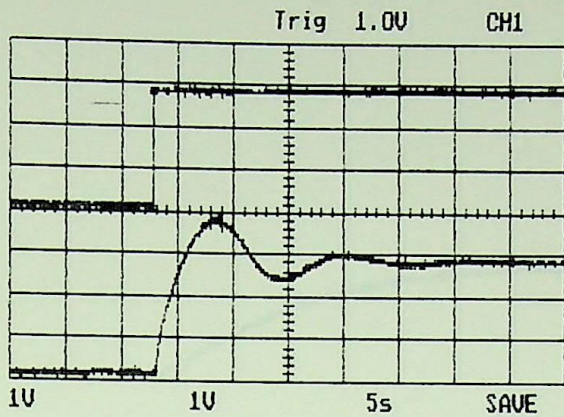
(e) motor com carga nominal
tensão do campo = 110 [V]



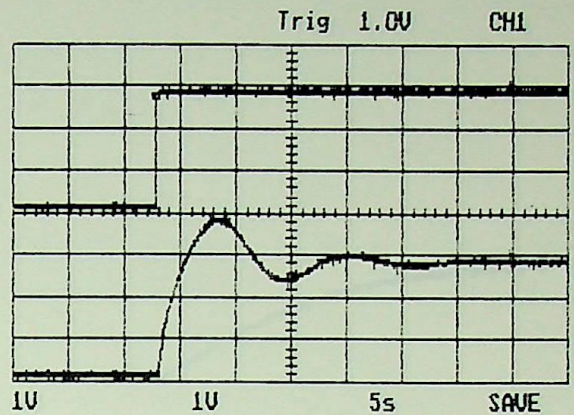
(c) motor a vazio
tensão do campo = 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 30% E TEMPO DE ACOMODAÇÃO DE 30(S)

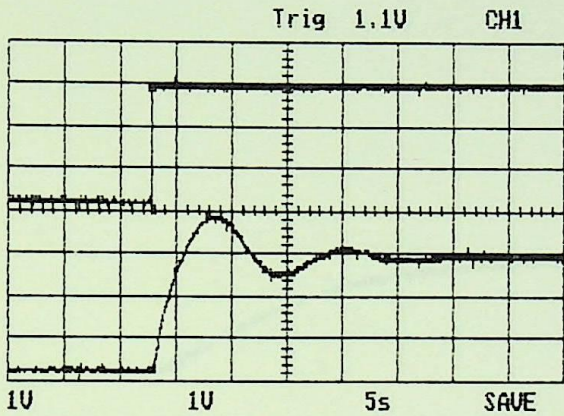
Figura 6.8



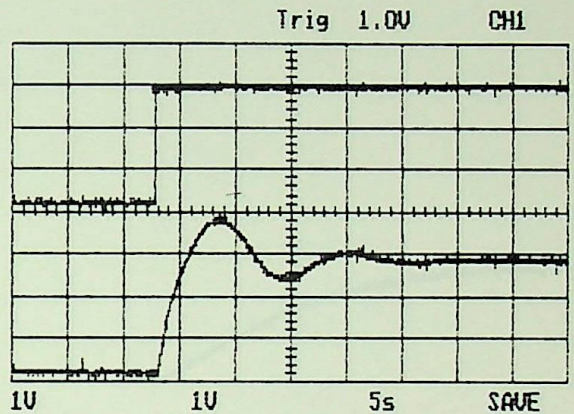
(a) motor a vazio
tensão do campo = 220 [V]



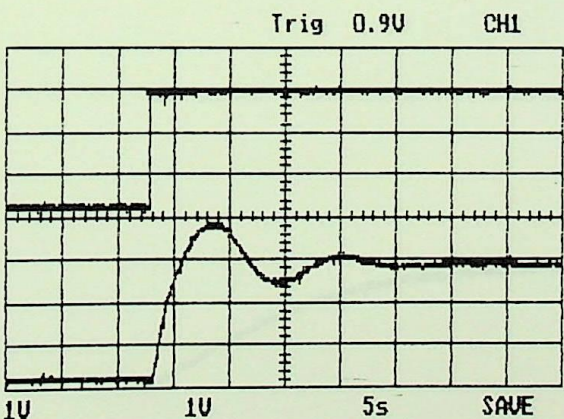
(d) motor com carga nominal
tensão do campo = 220 [V]



(b) motor a vazio
tensão do campo = 110 [V]



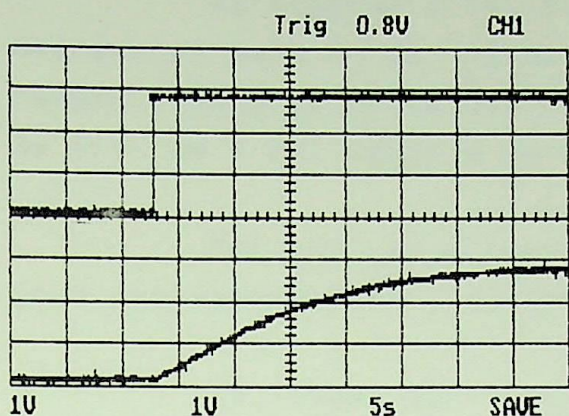
(e) motor com carga nominal
tensão do campo = 110 [V]



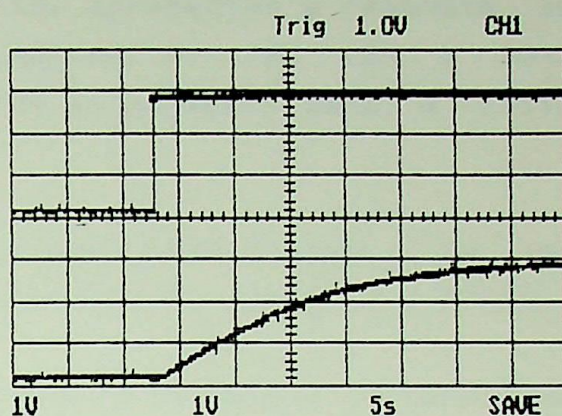
(c) motor a vazio
tensão do campo = 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 40% E TEMPO DE ACOMODAÇÃO DE 30IS

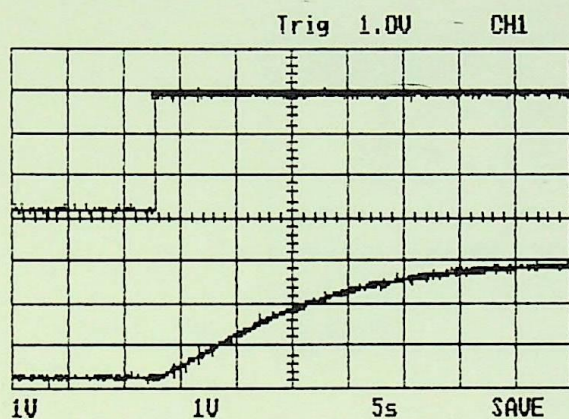
Figura 6.9



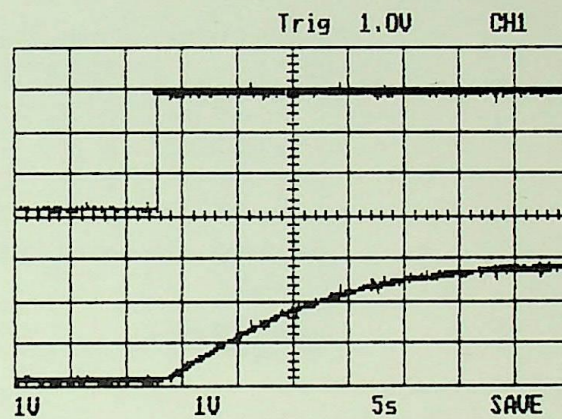
(a) motor a vazio
tensão do campo = 220 [V]



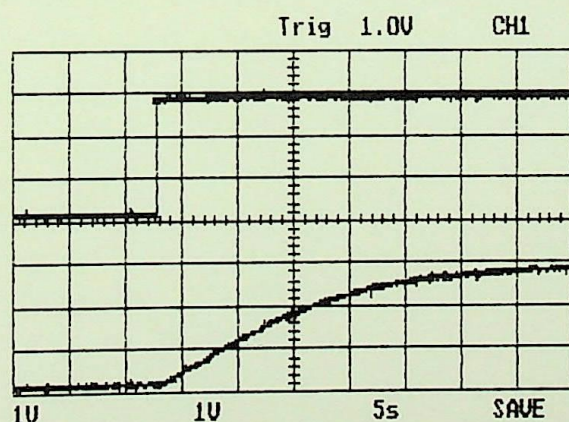
(d) motor com carga nominal
tensão do campo = 220 [V]



(b) motor a vazio
tensão do campo = 110 [V]



(e) motor com carga nominal
tensão do campo = 110 [V]



(c) motor a vazio
tensão do campo = 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE A UM AMORTECIMENTO CRÍTICO E CONSTANTE DE TEMPO DE 7.5[S]

Figura 6.10

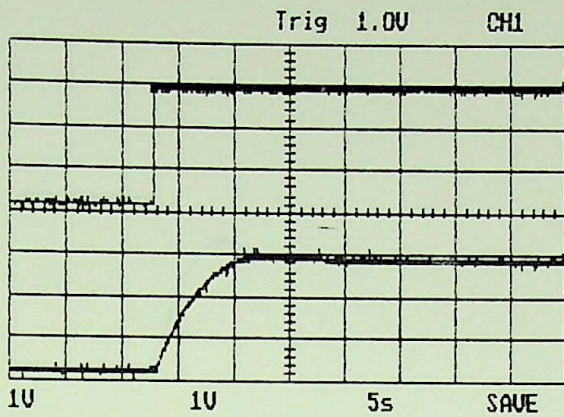
As figuras 6.11, 6.12 e 6.13, apresentam a resposta ao degrau com valores de ultrapassagem máxima de 4%, 15% e 30%, respectivamente, porém com um tempo de acomodação, para a faixa de erro de $\pm 2\%$, igual a 15 [s].

Por sua vez a figura 6.14, mostra uma resposta ao degrau com amortecimento crítico, e constante de tempo de 3.75 [s].

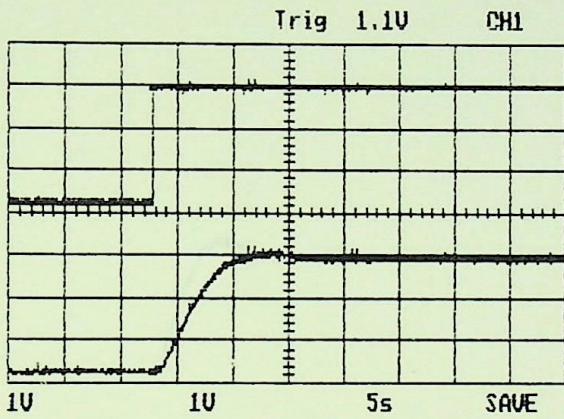
Um resumo das curvas de resposta ao degrau apresentadas nas figuras mencionadas acima, é fornecido na tabela 6.2.

OVERSHOOT	FIGURA		
4 %	6.11 a	6.11 b	6.11 c
15 %	6.12 a	6.12 b	6.12 c
30 %	6.13 a	6.13 b	6.13 c
amort.crit.	6.14 a	6.14 b	6.14 c
CAMPO [V]	220	50	220
CARGA	VAZIO		NOMINAL

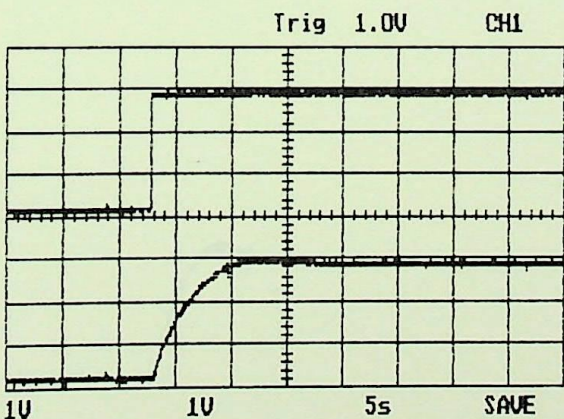
Tabela 6.2



(a) motor a vazio
tensão do campo = 220 [V]



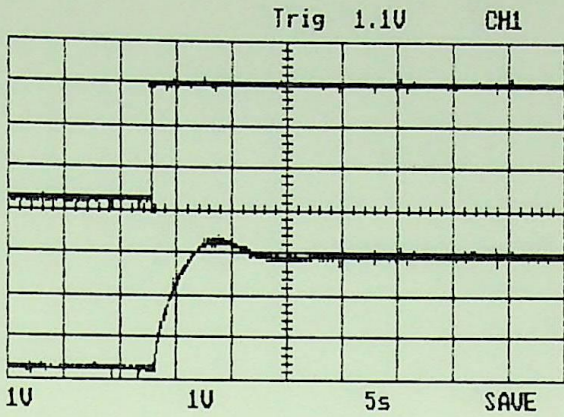
(b) motor a vazio
tensão do campo = 50 [V]



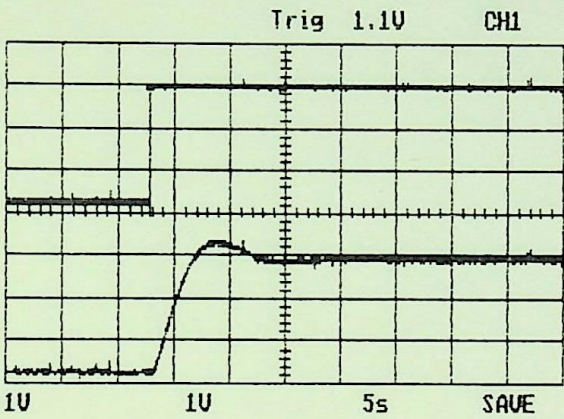
(c) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 4% E TEMPO DE ACOMODAÇÃO DE 15[S]

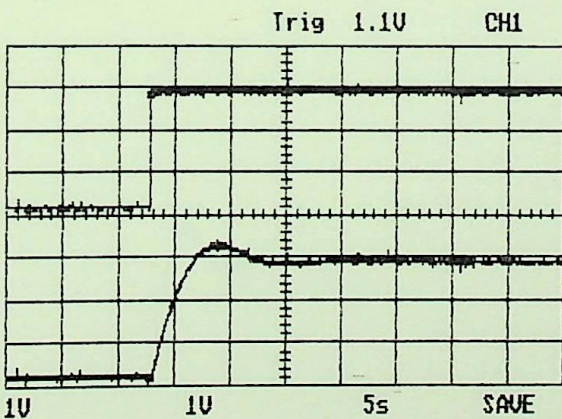
Figura 6.11



(a) motor a vazio
tensão do campo = 220 [V]



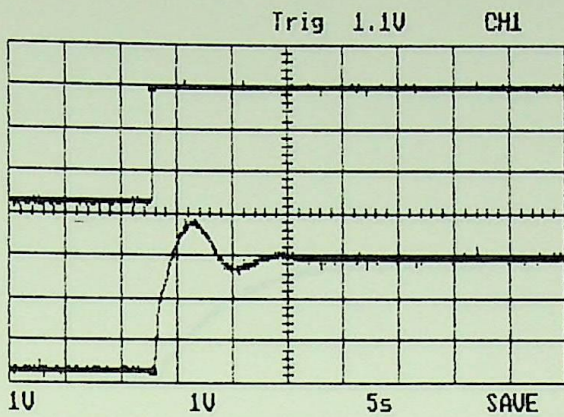
(b) motor a vazio
tensão do campo = 50 [V]



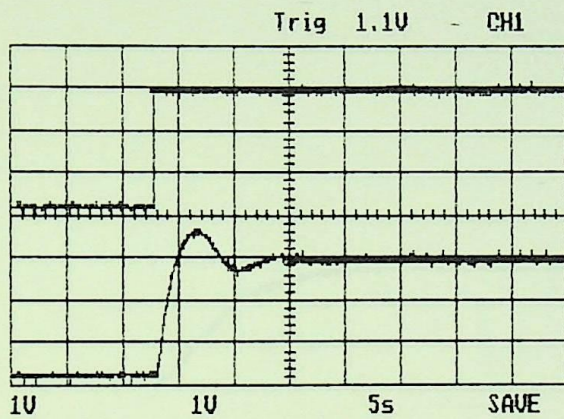
(c) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 15% E TEMPO DE ACOMODAÇÃO DE 15(S)

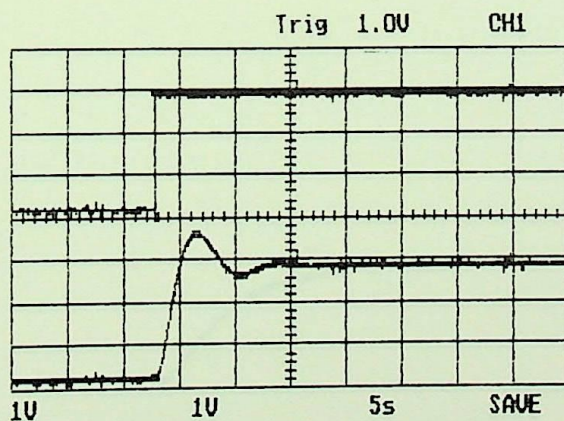
Figura 6.12



(a) motor a vazio
tensão do campo = 220 [V]



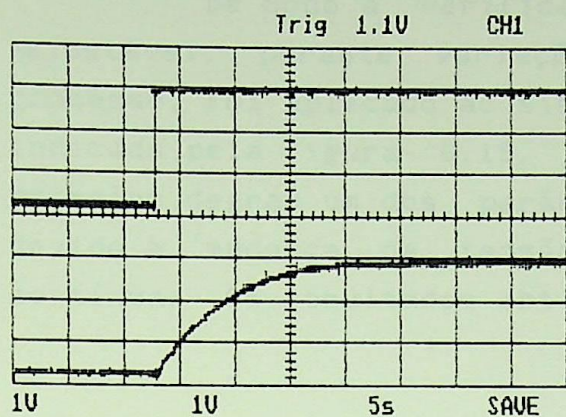
(b) motor a vazio
tensão do campo = 50 [V]



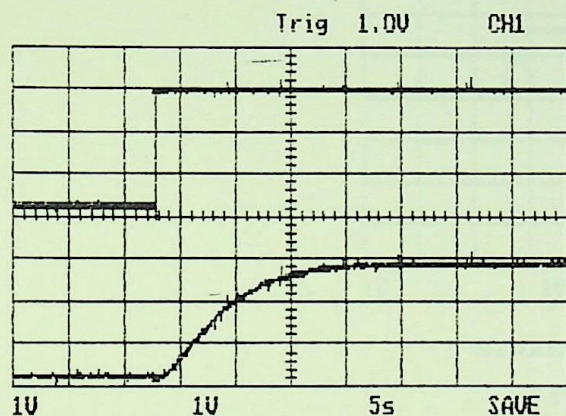
(c) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 30% E TEMPO DE ACOMODAÇÃO DE 15[S]

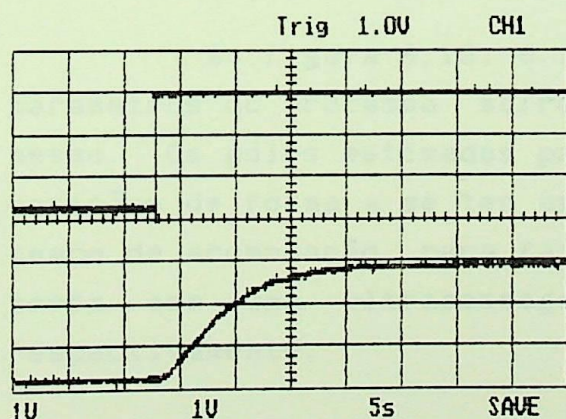
Figura 6.13



(a) motor a vazio
tensão do campo = 220 [V]



(b) motor a vazio
tensão do campo = 50 [V]



(c) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE A UM AMORTECIMENTO CRÍTICO E CONSTANTE DE TEMPO DE 3.75[S]

Figura 6.14

De modo a verificar o desempenho do sistema auto-ajustável, perante variações dinâmicas dos parâmetros do processo, foi aplicado ao sistema um sinal de referência na forma indicada pela figura 6.15. Aproximadamente, 30 [s] após o primeiro degrau um dos parâmetros da planta sofreu variação, devido à mudança da tensão no campo do motor de corrente contínua. Os resultados obtidos são apresentados abaixo.

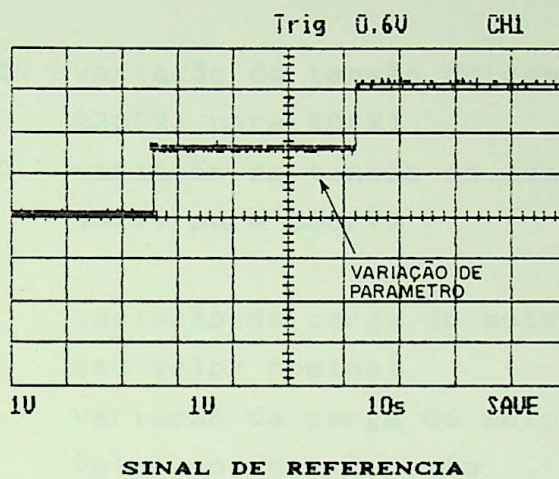


Figura 6.15

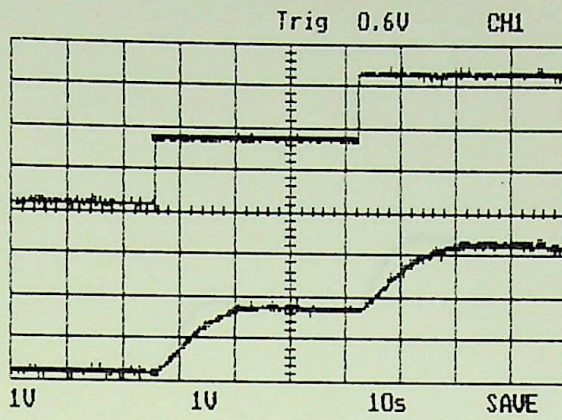
As figuras 6.16, 6.17 e 6.18, mostram casos, onde um dos parâmetros do processo sofre variação durante a evolução do mesmo. Os pólos estimados para o processo foram deslocados para posições de forma a se ter uma resposta ao degrau sempre com um tempo de acomodação, para faixa de erro de $\pm 2\%$, igual a 30 [s], porém com uma ultrapassagem máxima de 4%, 15% e 30%, respectivamente.

A tabela 6.3, fornece um resumo das curvas de resposta ao degrau apresentadas nas figuras mencionadas acima.

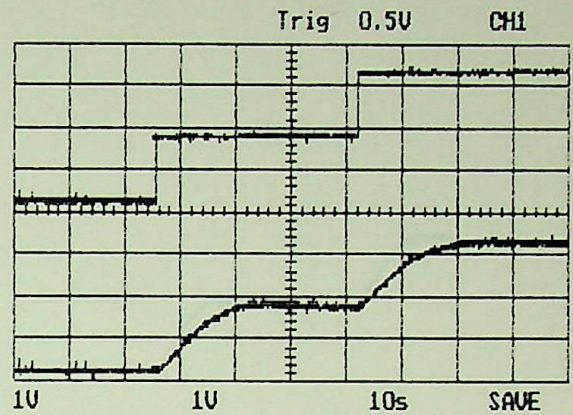
OVERSHOOT	FIGURA					
4 %	6.16 a	6.16 b	6.16 c	6.16 d	6.16 e	6.16 f
15 %	6.17 a	6.17 b	6.17 c	6.17 d	6.17 e	6.17 f
30 %	6.18 a	6.18 b	6.18 c	6.18 d	6.18 e	6.18 f
CAMPO [V]	220 → 50	220	50 → 220	50 → 220	220	220 → 50
CARGA	VAZIO	NL → VZ	VZ → NL	VAZIO	VZ → NL	50% NL

Tabela 6.3

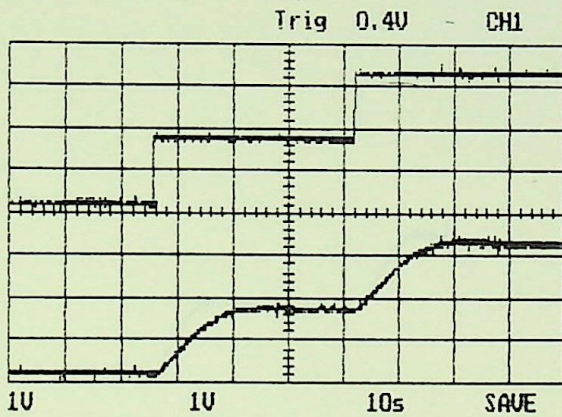
- onde:
- 220 → 50 variação da tensão do campo do motor CC de 220[V] para 50[V]
 - 50 → 220 variação da tensão do campo do motor CC de 50[V] para 220[V]
 - VZ → NL variação da carga do motor CC de vazio à seu valor nominal
 - NL → VZ variação da carga do motor CC de seu valor nominal à vazio
 - 50% NL motor operando com 50% de sua carga nominal



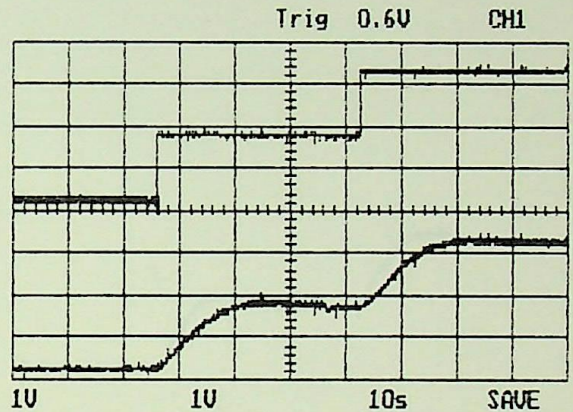
(a) motor a vazio
campo = 220 [V] → 50 [V]



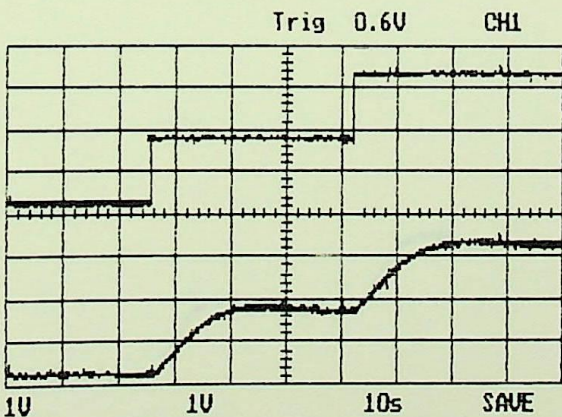
(d) motor a vazio
campo = 50 [V] → 220 [V]



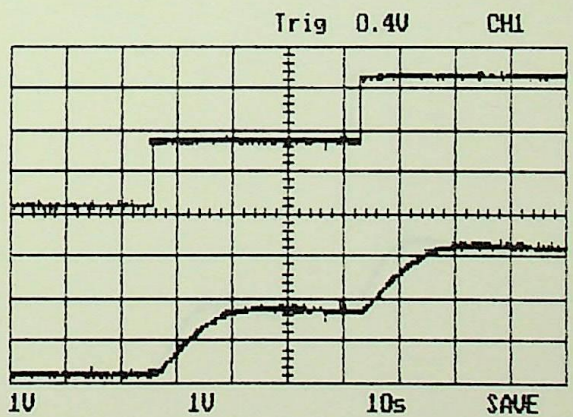
(b) carga = nominal → vazio
tensão do campo = 220 [V]



(e) carga = vazio → nominal
tensão do campo = 220 [V]



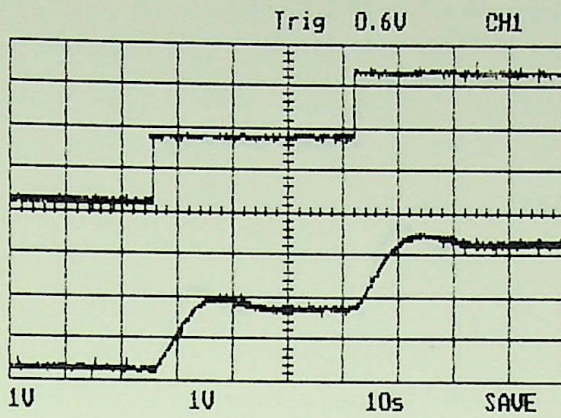
(c) carga = vazio → nominal
campo = 50 [V] → 220 [V]



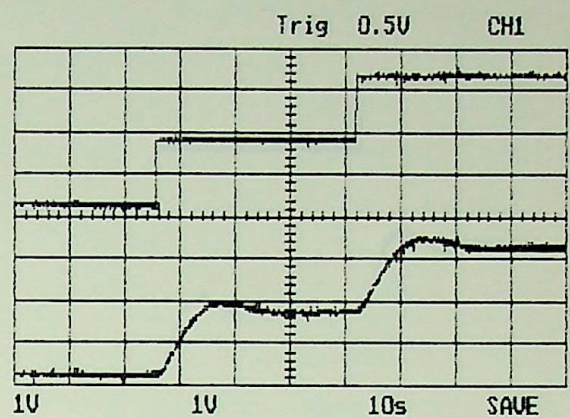
(f) carga = 50% da nominal
campo = 220 [V] → 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 4% E VARIAÇÃO DE PARÂMETROS

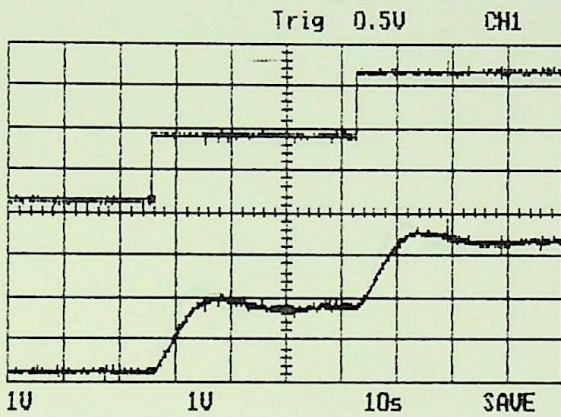
Figura 6.16



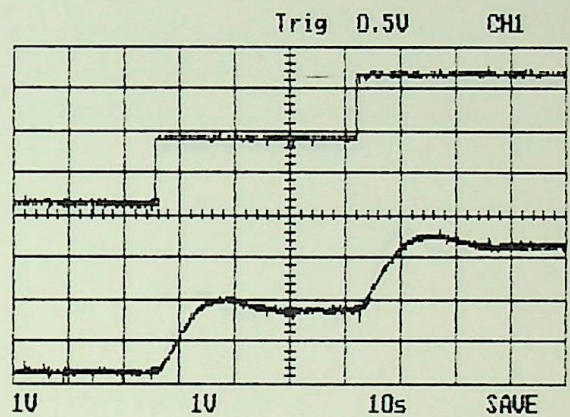
(a) motor a vazio
campo = 220 [V] → 50 [V]



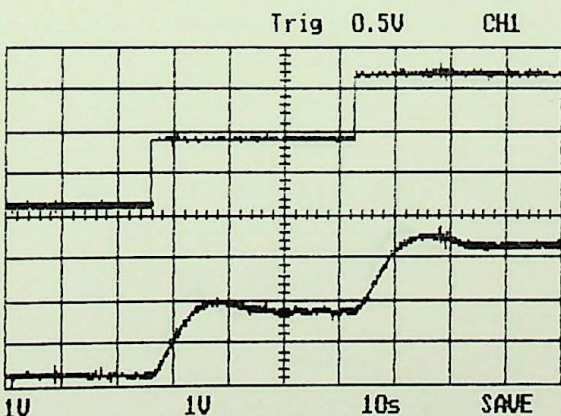
(d) motor a vazio
campo = 50 [V] → 220 [V]



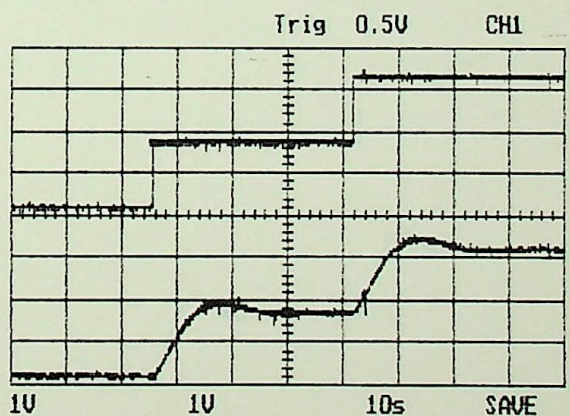
(b) carga = nominal → vazio
tensão do campo = 220 [V]



(e) carga = vazio → nominal
tensão do campo = 220 [V]



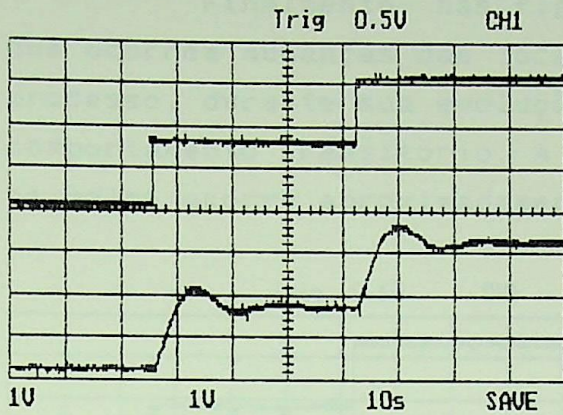
(c) carga = vazio → nominal
campo = 50 [V] → 220 [V]



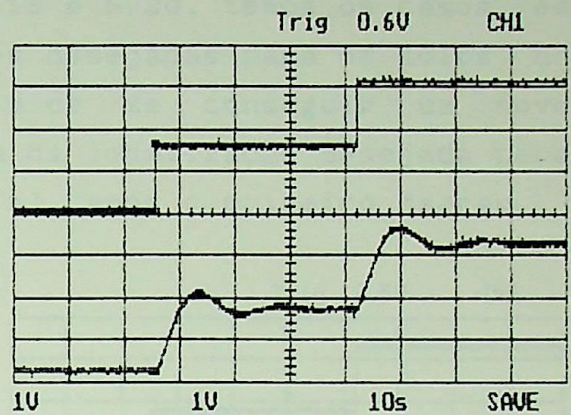
(f) carga = 50% da nominal
campo = 220 [V] → 50 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 15% E VARIAÇÃO DE PARÂMETROS

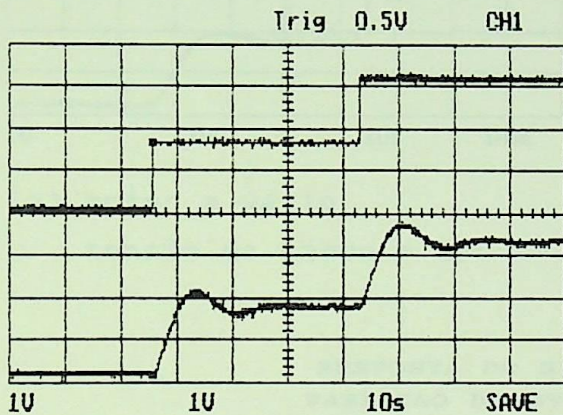
Figura 6.17



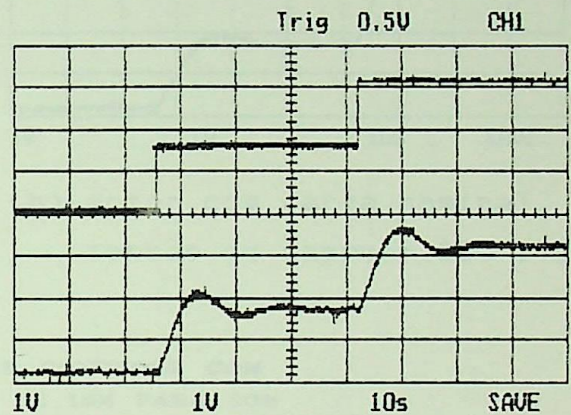
(a) motor a vazio
campo = 220 [V] → 50 [V]



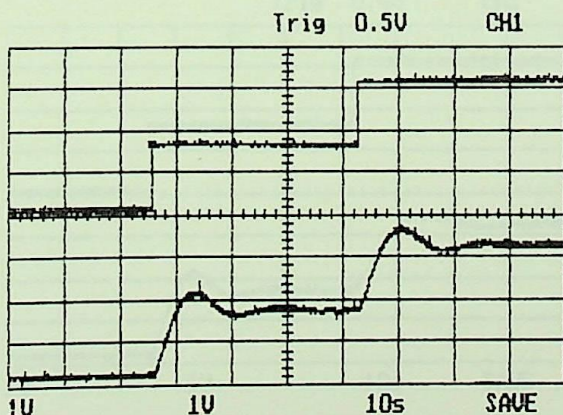
(d) motor a vazio
campo = 50 [V] → 220 [V]



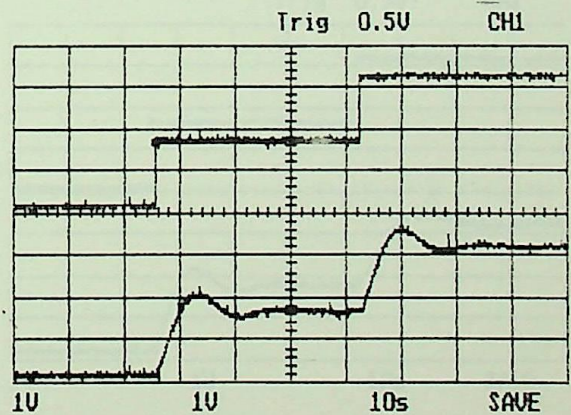
(b) carga = nominal → vazio
tensão do campo = 220 [V]



(e) carga = vazio → nominal
tensão do campo = 220 [V]



(c) carga = vazio → nominal
campo = 50 [V] → 220 [V]

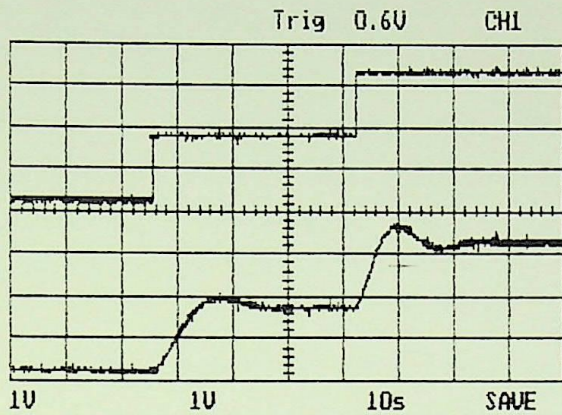


(f) carga = 50% da nominal
campo = 220 [V] → 50 [V]

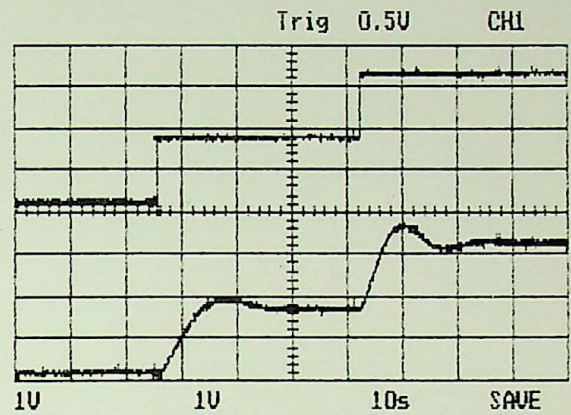
RESPOSTA DO SISTEMA DE CONTROLE COM
OVERSHOOT DE 30% E VARIAÇÃO DE PARAMETROS

Figura 6.18

Finalmente, nas figuras 6.19 e 6.20, temos os casos em que ocorrem mudanças das localizações desejadas para os pólos do processo, durante sua evolução, a fim de se conseguir um novo comportamento transitório. A mudança da localização desejada para os pólos ocorre aproximadamente 30 [s], após o primeiro degrau.



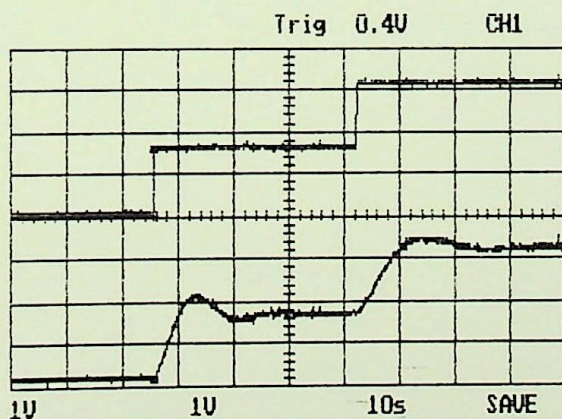
(a) motor a vazio
tensão do campo = 220 [V]



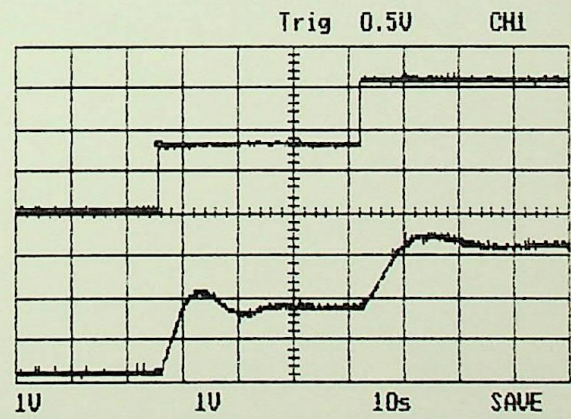
(b) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
VARIACÃO DO OVERSHOOT DE 15% PARA 30%

Figura 6.19



(a) motor a vazio
tensão do campo = 220 [V]



(d) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE COM
VARIACÃO DO OVERSHOOT DE 30% PARA 15%

Figura 6.20

Este último caso vem demonstrar uma das maiores vantagens de um sistema de controle digital, implementado através de microprocessadores, ou seja, a alteração da estratégia de controle por meio de uma simples mudança nos parâmetros do "software" desenvolvido.

7.1 Introdução

O programa desenvolvido para o controle auto-ajustável foi desenvolvido em linguagem C, e implementado em um computador IBM PC (CPU 486) de 4,77 MHz, equipado com o processador analógico AD574.

O programa leva ao longo de 20 (ms) a cada iteração do controle digital. Sua arquitetura consiste das seguintes partes:

- aquisição de dados de entrada (valor de referência e velocidade do motor de corrente contínua);
- implementação do algoritmo de estimação de parâmetros (utilizando o método dos mínimos quadrados recursivos);
- cálculo dos pólos da função de transferência para posições convenientes (Pole Assignment);
- envio do sinal de controle para o processo.

7.2 Programa

A figura 7.1, apresenta o diagrama de "software" desenvolvido para o controle auto-ajustável. A linguagem de programação utilizada é o C.

As aquisições de dados de entrada (valor de referência e velocidade do motor) são feitas pelo controle digital por intermédio de uma placa de aquisição de dados, equipada com o conversor AD574 do IBM PC. Esta placa atua

SOFTWARE DE CONTROLE

7.1 Introdução

O programa responsável pelo controle auto-ajustável foi totalmente desenvolvido em linguagem C, e implementado em um IBM/PC (CPU 8088) de 4.77 MHz, equipado com co-processador aritmético 8087.

O programa leva em torno de 20 [ms] à cada iteração do controle digital. Sua estrutura consiste basicamente de quatro partes:

- aquisição de dados de entrada (sinal de referência e de velocidade do motor de corrente contínua)
- implementação do algoritmo de estimação de parâmetros (método dos mínimos quadrados recursivo)
- alocação dos pólos da função de transferência para posições convenientes (Pole Assignment)
- envio dos sinais de controle para o processo

7.2 Programa

A figura 7.1, apresenta o fluxograma do "software" desenvolvido para o controle auto-ajustável. A listagem deste "software" é fornecida no apêndice C.

As aquisições dos dados de entrada (sinal de referência e de velocidade do motor), e o envio dos sinais de controle foram feitos por intermédio de uma placa de aquisição de dados, acoplada em um dos "slots" do IBM/PC. Esta placa continha além

de todos os subsistemas necessários ao seu funcionamento (fontes, sample hold, multiplexadores, etc), um conversor A/D de 12 bits e um conversor D/A de 10 bits, cada qual podendo comportar oito canais multiplexados. A comunicação desta placa com o IBM/PC se deu através de um endereço de entrada/saída (input/output - I/O), e a leitura de dados da mesma foi feita por "pooling".

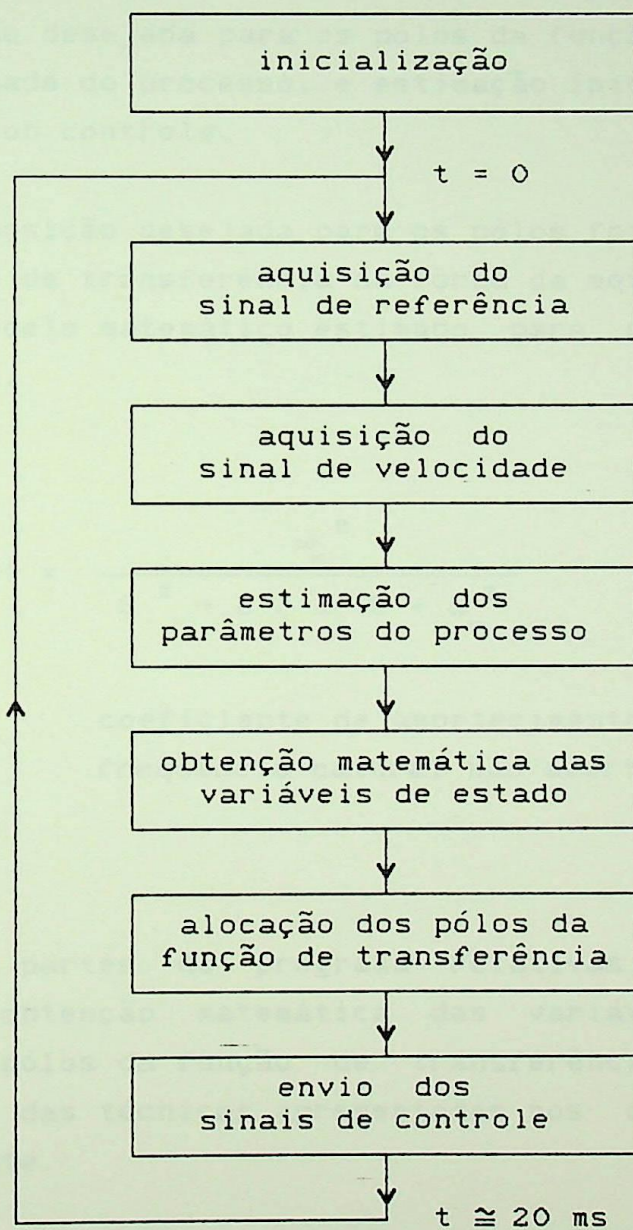


Figura 7.1

Toda as comunicações entre a placa e o IBM/PC, assim como o tratamento dos dados de entrada (referência e feedback), e dos dados de controle (sinal de controle), foram feitas por rotinas dedicadas, desenvolvidas em linguagem C.

A parte de inicialização do programa é responsável pela: seleção dos canais da placa de aquisição de dados (entrada e saída), ajuste do "offset" dos canais selecionados, inicialização do algoritmo de estimação de parâmetros, definição da localização desejada para os pólos da função de transferência de malha fechada do processo, e estimação inicial dos parâmetros do processo sob controle.

A posição desejada para os pólos foi fornecida por meio de uma função de transferência na forma da equação (7.1), devido ao fato do modelo matemático estimado para o processo ser de segunda ordem.

$$F(S) = \frac{\omega_n^2}{S^2 + 2\xi\omega_n S + \omega_n^2} \quad (7.1)$$

onde: ξ coeficiente de amortecimento
 ω_n frequência natural não amortecida

As partes do programa relativas a estimação de parâmetros, obtenção matemática das variáveis de estado, e alocação dos pólos da função de transferência, correspondem a implementação das técnicas apresentadas nos capítulos 4, 2 e 3, respectivamente.

A fim de evitar o constante crescimento dos elementos da matriz P_N do algoritmo de estimação, causado por um fator de esquecimento ρ menor do que a unidade ($\rho < 1$), fez-se:

$$\rho = \begin{cases} \rho \text{ inicial} & p / |\eta| > e \\ 1.00 & p / |\eta| < e \end{cases}$$

onde: η é o erro entre a saída do processo e a saída do modelo matemático estimado
 e valor de erro limite

A necessidade acima, é devido ao fato dos microprocessadores atuais possuírem uma capacidade limitada de representação numérica.

CONCLUSÕES

Os sistemas de controle auto-ajustável vem se tornando a cada dia uma realidade, principalmente nas indústrias, onde os sistemas são cada vez mais complexos, e exige-se um melhor desempenho.

A área de acionamento industrial representa hoje um dos campos mais férteis, para este tipo de controle, devido a sua flexibilidade, simplicidade, facilidade de implementação por meio de microprocessadores, e sobretudo baixo custo, quando comparado com outras técnicas de controle adaptativo.

A utilização da estrutura proposta (figura 1.3), para o controle de velocidade de motores de corrente contínua, acionados por conversores tiristorizados, embora com os problemas apresentados na prática, nos parece bastante viável.

Os problemas que surgiram na fase de implementação e as possíveis soluções, são discutidas a seguir. Assim, com uma análise das técnicas utilizadas para compor as várias partes do sistema de controle proposto neste trabalho.

Com relação ao método discutido no segundo capítulo, que trata da obtenção das variáveis de estado de um processo, a partir de sua função de transferência, constatou-se sobretudo a sua eficiência, facilidade para o entendimento e simplicidade de implementação à nível de "software".

A técnica de alocação de pólos, responsável pelo deslocamento dos pólos do processo para posições convenientes (capítulo 3), embora bastante simples e satisfazendo às exigências da estrutura de controle proposta, mostrou-se

limitada, principalmente no fato de não se poder alterar também as posições dos zeros do processo sob controle.

A análise da estabilidade do algoritmo de alocação de pólos não faz parte do contexto deste trabalho, entretanto, esta tem sido largamente investigada [10].

O algoritmo de estimação de parâmetros, descrito no quarto capítulo, provou ser eficaz na obtenção dos parâmetros de um processo variante no tempo e ou desconhecido, conforme resultados obtidos por meio de simulações e apresentados no capítulo 5.

Dois problemas de ordem prática surgiram, com a utilização deste algoritmo, na implementação do sistema de controle auto-ajustável proposto. O primeiro se refere ao fator de esquecimento empregado por este algoritmo, de modo a fazer com que o mesmo não venha a convergir, quando aplicado a processos variantes no tempo. A simples utilização de um fator de esquecimento, menor do que a unidade, pode levar a saída do processo à valores inaceitáveis na prática, e também a erros de computação, devido à limitada capacidade de representação numérica dos microprocessadores atuais.

Como solução optou-se por fazer com que o fator de esquecimento somente fosse menor que a unidade, nos momentos em que o erro entre a saída real do processo, e a saída do modelo matemático, obtido com os parâmetros estimados, fosse superior a um determinado valor limite. No entanto, constatou-se que, quanto maior for este limite para o erro entre as saídas, menor será a precisão dos parâmetros estimados, e maior será o número de iterações necessárias para obter um modelo matemático satisfatório para o processo.

Sugere-se desta forma para trabalhos futuros, utilizar outras técnicas, a fim de solucionar o problema. Como possibilidade, podemos mencionar a utilização de um fator de

esquecimento, cujo valor varie de forma linear com o número de iterações, até se obter uma precisão aceitável para os parâmetros estimados. Outras sugestões são mencionadas no final do quarto capítulo.

O segundo problema relacionado ao algoritmo de estimação refere-se ao fato do algoritmo implementado não conseguir estimar os parâmetros do processo sob condições estacionárias, embora esteja permanentemente tentando. Assim, torna-se necessário introduzir perturbações no sistema, ou atualizar a estimação dos parâmetros nos instantes em que a entrada for estimulada.

Como o objetivo visado era a implementação prática e a simples utilização do algoritmo de estimação de parâmetros, uma análise sobre a sua convergência [5] não constitui uma parte do escopo deste trabalho. A convergência dos parâmetros estimados pelo algoritmo é assegurada somente quando muitas considerações restritivas são feitas [4].

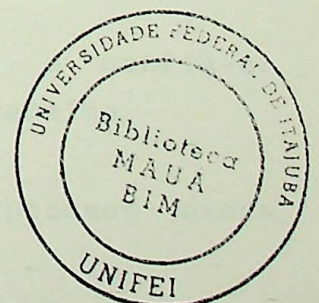
O processo a ser controlado mostrou-se bastante adequado, visto tratar-se de um processo já conhecido na área de controle. Desta forma os resultados obtidos por meio do controle proposto, podiam ser imediatamente questionados sobre a sua validade.

O principal inconveniente no processo sob controle foi devido ao fato deste utilizar somente um conversor tiristorizado, ao invés de um conversor dual. Assim, quando o sistema era submetido a um sinal de referência na forma de um degrau negativo, a única maneira do motor de corrente contínua consumir a sua energia cinética armazenada, era por meio de atrito. Neste caso, o sistema não conseguia se auto-ajustar, visto que o processo sob controle, passava a operar em uma região não linear. A nível de exemplificação, alguns resultados práticos foram obtidos nesta situação, onde ocorre um degrau negativo no sinal de referência. Estes ensaios são apresentados no apêndice B.

Finalmente com relação a velocidade da malha de controle, constatou-se que esta ficou limitada frente a velocidade de operação do microprocessador utilizado, e a quantidade de cálculos aritméticos em ponto flutuante envolvida. Uma solução para este impasse, consiste em utilizar um microprocessador mais veloz, associado a um co-processador aritmético mais rápido.

A estrutura proposta, embora apresentando limitações nos casos em que os parâmetros do processo variam muito rapidamente, ou com frequência muito grande, mostrou-se bastante simples e de baixo custo para o controle de velocidade de motores de corrente contínua, acionados por conversor tiristorizado.

Mesmo com os problemas apresentados, podemos dizer de um modo geral, que os resultados obtidos com este trabalho condizeram totalmente com as expectativas teóricas, além de nos fornecer um conhecimento básico, necessário para uma melhor compreensão das técnicas de controle auto-ajustável, e sobretudo de controle adaptativo.



EXTENSÕES PARA TRABALHOS FUTUROS

A área de engenharia industrial permite um grande número de aplicações das técnicas de controle auto-ajustável, e abre vários campos para pesquisa, aprimoramento e desenvolvimento deste tipo de controle.

A estrutura de controle proposta por este trabalho, embora simples e de fácil implementação, o que a torna bastante atrativa, também abre por si mesma várias possibilidades de aprimoramento, e de pesquisa, tanto a nível prático como teórico.

Dentre as possibilidades para o aprimoramento do sistema de controle de velocidade implementado, podemos citar:

- implementação de um sistema dedicado de controle de velocidade, utilizando a estrutura proposta neste trabalho.

- desenvolvimento do software de controle a nível de linguagem de máquina.

- divisão da parte lógica do sistema proposto em dois subsistemas dedicados e independentes, sendo um responsável unicamente pela estimação dos parâmetros do processo sob controle, e o outro, responsável por toda a lógica do controle.

- implementação de uma técnica aceitável e confiável para inicialização do sistema implementado neste trabalho.

Visando um aprimoramento deste trabalho podemos ainda sugerir:

- análise da confiabilidade do sistema implementado.

- análise da estabilidade da estrutura de controle desenvolvida.

- melhoramento das técnicas utilizadas neste trabalho e estudo de técnicas alternativas, a fim de obter um melhor desempenho do sistema implementado.

Dentre as possibilidades para futuros trabalhos de pesquisa podemos sugerir:

- implementação da estrutura proposta no controle de velocidade de motores de corrente contínua, acionados por conversores duais.

- aplicação e análise da estrutura de controle proposta em um sistema industrial operando sob condições reais.

- análise e implementação da estrutura de controle desenvolvida para uma malha de corrente.

- aplicação das técnicas apresentadas neste trabalho, à sistemas multi-variáveis.

BIBLIOGRAFIA

- [1] Yoan D. Landau (1979)
Adaptive Control - The Model Reference Approach
Marcel Dekker, Inc.
- [2] Charles L. Phillips, H. Troy Nagle Jr. (1984)
Digital Control System Analysis and Design
Prentice-Hall, Inc.
- [3] D. Graupe, V.K. Jain and J. Salahi (1980)
A Comparative Analysis of Various Least-Squares
Identification Algorithms
Automatica, vol. 16
- [4] R. Hastings-James and M.W. Sage (1969)
Recursive Generalised-Least-Squares Procedure for Online
Identification of Process Parameters
IEE Proceedings. vol. 116, no 12
- [5] Wagner Caradori do Amaral
Identificação de Sistemas Discretos Por
Métodos Sequenciais
UNICAMP - Tese de Mestrado, setembro 1976
- [6] Arthur E. Bryson Jr., Yu-Chi Ho (1975)
Applied Optimal Control
Hemisphere Publishing Corporation
- [7] Andrew P. Sage, Chelsea C. White III (1977)
Optimum Systems Control
Prentice-Hall, Inc.

- [8] Antonio A. R. Coelho e Lúcia V. R. Arruda
Técnicas para a Estimaco Paramétrica de Sistemas
Variantes no Tempo
Rev. Tecnol., Belém, Jul/dez 1988
- [9] Katsuhiko Ogata
Engenharia de Controle Moderno
Prentice-Hall, Inc.
- [10] Howard Elliott, R. Cristi and Manohar Das (1985)
Global Stability of Adaptive Pole Placement Algorithms
IEEE Transactions on Automatic Control, vol. AC-30, no 4
- [11] F. Harashima, Y. Dote, et al.
Status Review Applications of Modern Control Theory
IECON 84
- [12] K.J. Astrom (1983)
Theory and Applications of Adaptive Control - A Survey
Automatica, vol. 19, no 5
- [13] H. Kurz, R. Isermann and R. Schumann (1980)
Experimental Comparison and Application of Various
Parameter-Adaptive Control Algorithms
Automatica, vol. 16
- [14] Axel Brickwedde (1985)
Microprocessor-Based Adaptive Speed and Position
Control for Electrical Drives
IEEE Trans. on Industry Applications, vol. IA-21, no 5

- [15] K.J. Astrom, U. Borisson, L. Ljung and B. Wittenmark (1977)
Theory and Applications of Self-Tuning Regulators
Automatica, vol. 13
- [16] D.W. Clarke and P.J. Gawthrop (1979)
Self-Tuning Control
IEE Proceedings. vol. 126, no 6
- [17] K.J. Astrom and B. Wittenmark (1980)
Self-Tuning Controllers Based on Pole-Zero Placement
IEE Proceedings. vol. 127, no 3
- [18] P.E. Wellstead, D. Prager and P. Zanker (1979)
Pole Assignment Self-Tuning Regulator
IEE Proceedings. vol. 126, no 8
- [19] Rolf Isermann (1982)
Parameter Adaptive Control Algorithms - A Tutorial
Automatica, vol. 18, no 5
- [20] R. Isermann (1980)
Practical Aspects of Process Identification
Automatica, vol. 16

APÊNDICE A

FORMULA DE SHERMAN-MORRISON-WOODBURY

Seja: A uma matriz do tipo $n \times m$
 U^T e V^T vetores do tipo $k \times n$

$$P_N^{-1} = A^T A \quad \longrightarrow \quad P_N = [A^T A]^{-1} \quad (A.1)$$

$$P_{N+1}^{-1} = \left[\begin{array}{c} A \\ \hline U^T \end{array} \right]^T \left[\begin{array}{c} A \\ \hline V^T \end{array} \right] \longrightarrow P_{N+1} = \left[\left[\begin{array}{c} A \\ \hline U^T \end{array} \right]^T \left[\begin{array}{c} A \\ \hline V^T \end{array} \right] \right]^{-1} \quad (A.2)$$

Da equação (A.2), vem:

$$P_{N+1}^{-1} = [A^T \mid U] \left[\begin{array}{c} A \\ \hline V^T \end{array} \right] \longrightarrow P_{N+1}^{-1} = A^T A + U V^T$$

$$P_{N+1}^{-1} = P_N^{-1} + U V^T \quad (A.3)$$

pré-multiplicando (A.3) por P_{N+1} , vem:

$$I = P_{N+1} P_N^{-1} + P_{N+1} U V^T \quad (A.4)$$

onde: I é a matriz Identidade

pós-multiplicando (A.4) por P_N , resulta:

$$P_N = P_{N+1} + P_{N+1} U V^T P_N \quad (A.5)$$

pós-multiplicando (A.5) por U , obtém-se:

$$P_N U = P_{N+1} U + P_{N+1} U V^T P_N U$$

$$P_N U = P_{N+1} U [I + V^T P_N U] \quad (A.6)$$

pós-multiplicando (A.6) por $[I + V^T P_N U]^{-1}$, vem:

$$P_N U [I + V^T P_N U]^{-1} = P_{N+1} U \quad (A.7)$$

pós-multiplicando (A.7) por $V^T P_N$, resulta:

$$P_N U [I + V^T P_N U]^{-1} V^T P_N = P_{N+1} U V^T P_N \quad (A.8)$$

porém da equação (A.5), obtém-se que:

$$P_{N+1} U V^T P_N = P_N - P_{N+1}$$

portanto:

$$P_N U [I + V^T P_N U]^{-1} V^T P_N = P_N - P_{N+1}$$

logo:

$$P_{N+1} = P_N - P_N U [I + V^T P_N U]^{-1} V^T P_N \quad (A.9)$$

onde: A uma matriz do tipo $n \times m$
 U^T e V^T vetores do tipo $k \times n$

$$P_N = [A^T A]^{-1}$$

$$P_{N+1} = [A^T A + U V^T]^{-1}$$

A equação (A.9), é conhecida como fórmula de Sherman - Morrison - Woodbury, sendo necessário que P_N^{-1} e $[I + V^T P_N U]$ sejam matrizes não singulares.

Para o caso particular, em que U^T e V^T sejam vetores do tipo $1 \times n$, ter-se-á:

$$[I + V^T P_N U]^{-1} = \frac{1}{1 + V^T P_N U}$$

logo:

$$P_{N+1} = P_N - \frac{P_N U V^T P_N}{1 + V^T P_N U} \quad (A.10)$$

onde: $P_N = [A^T A]^{-1}$

$$P_{N+1} = [A^T A + U V^T]^{-1}$$

A equação (A.10) é conhecida como fórmula de Sherman - Morrison, sendo necessário que: $1 + V^T P_N U \neq 0$.

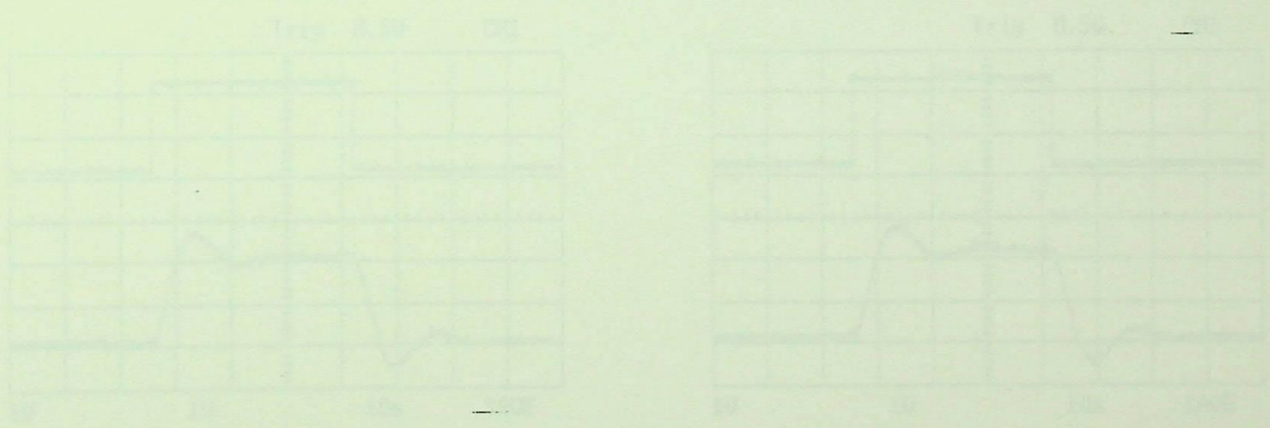
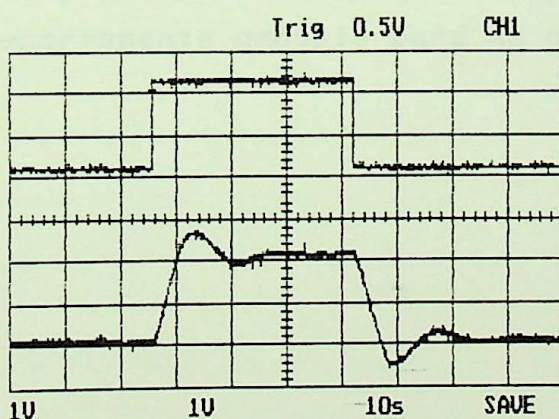


Figura 2.1

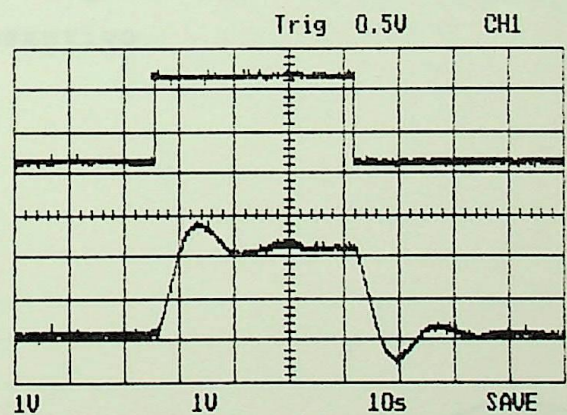
RESPOSTA A UM DEGRAU
NEGATIVO

Na estrutura de controle proposta neste trabalho, o motor de corrente contínua era acionado por um conversor tiristorizado. Assim, o consumo da energia cinética do processo sob controle dava-se unicamente por meio de atrito e ventilação, levando o processo a operar em uma região não linear, e conseqüentemente fazendo com que o sistema não conseguisse se auto-ajustar.

A fim de exemplificar tal fato, foi aplicado um degrau negativo no sinal de referência do sistema. Alguns dos resultados práticos que foram obtidos, são apresentados nas figuras B.1 e B.2.



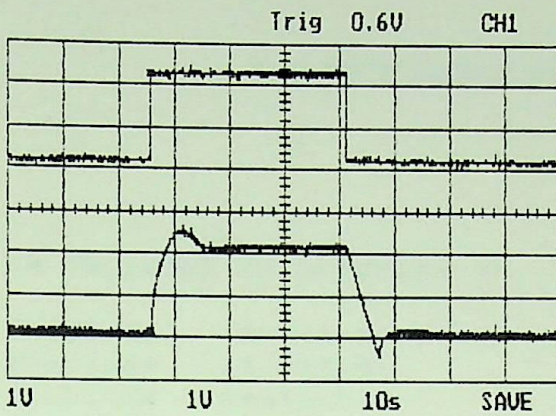
(a) motor a vazio
tensão do campo = 220 [V]



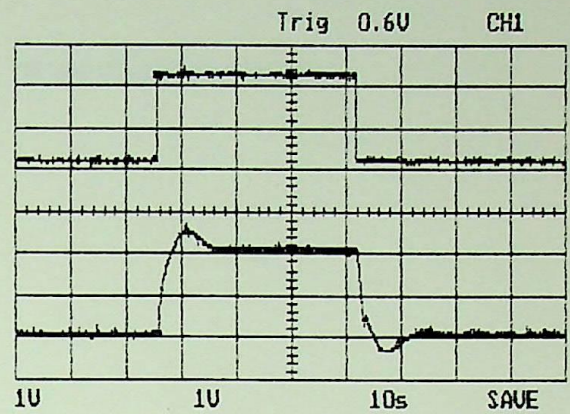
(b) motor com carga nominal
tensão do campo = 220 [V]

RESPOSTA DO SISTEMA DE CONTROLE A UM DEGRAU NEGATIVO
OVERSHOOT DE 30% E TEMPO DE ACOMODAÇÃO DE 30(S)

Figura B.1



(a) motor a vazio
tensão do campo = 220 [V]

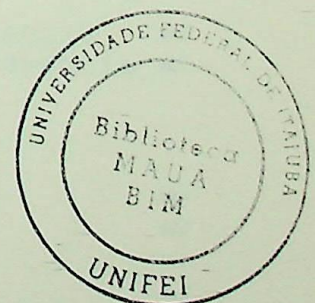


(d) motor com carga nominal
tensão do campo = 220 [V]

**RESPOSTA DO SISTEMA DE CONTROLE A UM DEGRAU NEGATIVO
OVERSHOOT DE 15% E TEMPO DE ACOMODAÇÃO DE 15[S]**

Figura B.2

Nas figuras acima, pode-se observar uma melhor resposta do sistema, quando o motor opera com carga nominal, e que o comportamento transitório para um degrau positivo, difere enormemente daquele para um degrau negativo.



APÊNDICE C

LISTAGEM DO PROGRAMA

```
/* INCLUDES do programa */
```

```
#include <stdio.h>
#include <conio.h>
#include <math.h >
#include <dos.h >
```

```
/* DEFINES da placa de aquisicao de dados */
```

```
#define ADLSB 0
#define ADMSB 1
#define ADOFF 2
#define ADSTS 4
#define DAMSB 6
#define DALSB 7
#define CTL 8
#define IODIG 10
#define TIMERO 12
#define TIMER1 13
#define TIMER2 14
#define TIMCTL 15

#define BSHSA 0x08
#define BSHEAN 0x10

#define Time_Out 25

#define in_port inp
#define out_port outp
```

```
/* DEFINES do programa */
```

```
#define Ord_FT 5 /* ordem da funcao de transferencia */
#define Num_FT 5 /* numero de funcoes de transferencia */
```

```
/* definicao de variaveis globais */
/* E Estimador */
/* P Planta */
/* M Modelo */
/* T Teta */
/* Xt X_transposta */
```



```

        placa_AD ( &canal_Feed, &offset_Feed );
        printf ( "\n\n\ncanal da planta: \n" );
        placa_DA ( &canal_Planta );
        break;
    }
}

placa_AD ( canal, offset )      /* conversor A/D */
unsigned char  *canal, *offset;
{
char          key;
unsigned int   dado;

printf ( "\ncanal          : " ); *canal = getche () - 48;
printf ( "\noffset inicial : " ); scanf ( "%d", offset );
printf ( "\n\n" );

do {
    key = ' ';
    dado = read_anl ( *canal, *offset );
    printf ( "offset = %2x  dado HEX = %3x  tensao [V] = %6.3f\r",
            *offset, dado, 5.0 * dado / 4096.0 );
    if ( kbhit () ) key = getch ();
    switch ( key ) {
        case '+' : *offset = *offset + 1;
                   break;
        case '-' : *offset = *offset - 1;
                   break;
    }
}
while ( key != 13 );

printf ( "\n\n" );
}

placa_DA ( canal )      /* conversor D/A */
unsigned char  *canal;
{
char          key;
float         tensao;
unsigned int   dado;

printf ( "\ncanal : " ); *canal = getche () - 48;

do {
    printf ( "\n\n" );
    printf ( "tensao [V] = " ); scanf ( "%f", &tensao );
    if ( tensao > 4.99512 ) tensao = 4.99512;
    if ( tensao < 0 )      tensao = 0;
    dado = (int) ( 1024.0 * tensao / 5.0 );
    printf ( "valor na saida: %6.3f", 5.0 * dado / 1024.0 );
    do {
        key = ' ';
    }
}
}

```

```

write_anl ( dado, *canal );
if ( kbhit ( ) ) key = getch ( );
}
while ( key == ' ' );
}
while ( key != 27 );

printf ( "\n\n" );
}

/**** entrada de dados e localização de polos ****/
Entrada_Dados ( )
{
char key, i, j, NModelo, Ord_M;

printf ( "numero de modelos : " ); scanf ( "%d", &NModelo );
printf ( "\n\n" );

for ( i = 1; i <= NModelo; i++ )
{
printf ( "\n\n" );
printf ( "\nordem do modelo %2d : ", i );
scanf ( "%d", &Ord_M );
printf ( "\n\nparametros do modelo %2d \n\n", i );
Ord_Modelo [i-1] = Ord_M;
switch ( Ord_M )
{
case 2 : printf ( "1 - dominio S \n2 - dominio Z" );
key = getch ( );
if ( key == '1' ) {
dominio_S ( i );
break;
}
printf ( "\n\n" );
default : for ( j = 1; j <= 2 * Ord_M; j++ )
{
if ( j <= Ord_M )
printf ( "a %2d = ", j );
else
printf ( "b %2d = ", j-Ord_M );
scanf ( "%f", &TM [i-1][j-1] );
}
}
}

printf ( "\n\n\n\n" );
printf ( "\ndiagonal da matriz P = " );
scanf ( "%f", &diag_P );
printf ( "\nvalor de ro = " );
scanf ( "%f", &ro_inicial );
printf ( "\nerro entre Planta e Estimador = " );
scanf ( "%f", &erro_PE );
printf ( "\n\n" );

```

```

printf ( "digite qualquer tecla" );
getch ();
}

dominio_S ( i )
char    i;
{
float    sigma, omega_d, csi, omega_n, ax, Pi;
float    tr, tp, f_osc, over, ts3, ts4;
char    key, jb;

Pi = 4.0 * atan ( 1.0 );
printf ( "\n\n" );
printf ( "1 - csi            omega_n            \n" );
printf ( "2 - sigma          omega_d            \n" );
printf ( "3 - overshoot      tempo de acomodacao " );
key = getch ();
printf ( "\n\n\n" );

switch ( key )
{
case '1' : printf ( "csi      = " ); scanf ( "%f", &csi );
           printf ( "omega_n = " ); scanf ( "%f", &omega_n );
           sigma = csi * omega_n;
           ax = 1 - csi * csi;
           omega_d = omega_n * sqrt( fabs( ax ) );
           if ( ax < 0 ) omega_d = - omega_d;
           break;

case '2' : printf ( "sigma >= 0                \n" );
           printf ( "omega_d < 0 ==> j * omega_d \n" );
           printf ( "raizes = - sigma +/- j * omega_d \n" );
           printf ( "\n\n" );
           printf ( "sigma    = " ); scanf ( "%f", &sigma );
           printf ( "omega_d = " ); scanf ( "%f", &omega_d );
           break;

case '3' : printf ( "overshoot [%%] = " );
           scanf ( "%f", &over );
           printf ( "\ntempo de acomodacao : \n" );
           printf ( "1 - erro de 2%% \n2 - erro de 5%% " );
           key = getch ();
           printf ( "\n\n" );
           switch ( key )
           {
           case '1' : printf ( "erro de 2%% = " );
                       scanf ( "%f", &ts4 );
                       sigma = 4.0 / ts4;
                       break;

           case '2' : printf ( "erro de 5%% = " );
                       scanf ( "%f", &ts3 );
                       sigma = 3.0 / ts3;
                       break;
           }
           omega_d = - sigma * Pi / log( over / 100.0 );
           break;
}
}

```

```

    }

ax = omega_d * omega_d;
if ( omega_d < 0 ) ax = - ax;
omega_n = sqrt( sigma * sigma + ax );
csi      = sigma / omega_n;

if ( omega_d > 0 ) {
    jb = 0;
    ax = atan( omega_d / sigma );
    tr = ( Pi - ax ) / omega_d;
    tp = Pi / omega_d;
    over = exp( - Pi * sigma / omega_d );
    f_osc = omega_d / ( 2 * Pi );
    ts3 = 3.0 / sigma;
    ts4 = 4.0 / sigma;
}
else {
    jb = 1;
    omega_d = fabs( omega_d );
}

printf ( "\n\n\n\n" );
printf ( "coeficiente de amortecimento = %10.5f \n", csi );
printf ( "freq. natural nao amortecida = %10.5f \n", omega_n );
printf ( "\n" );
printf ( "atenuacao                = %10.5f \n", sigma );
printf ( "freq. natural amortecida    = %10.5f \n", omega_d );
printf ( "raizes : " );

if ( jb ) {
    printf ( "%10.5f", - sigma - omega_d );
    printf ( "\n          " );
    printf ( "%10.5f", - sigma + omega_d );
}
else {
    printf ( "%10.5f - j %10.5f", - sigma, omega_d );
    printf ( "\n          " );
    printf ( "%10.5f + j %10.5f", - sigma, omega_d );
}

printf ( "\n\n\n" );

if ( !jb )
{
    printf ( "tempo de subida      = %10.5f \n", tr );
    printf ( "tempo de pico        = %10.5f \n", tp );
    printf ( "overshoot            = %10.5f [%%%] \n", over*100.0 );
    printf ( "freq. oscilacao      = %10.5f \n", f_osc );
    printf ( "\n" );
    printf ( "tempo de acomodacao p/ erro de 2%% = %10.5f\n", ts4 );
    printf ( "                               erro de 5%% = %10.5f\n", ts3 );
}

printf ( "\n\n\n\ntecle <ENTER>" );  getch ();

```

```

func_trans_Z ( i, sigma, omega_d, - sigma, jb );
}

func_trans_Z ( i, a, b, c, jb )
char    i, jb;
float   a, b, c;
{
float   T, k;
float   exp_aT, exp_2aT, cos_bT, sen_bT;

printf ( "\n\n\n\n" );
printf ( "Periodo de amostragem = " ); scanf ( "%f", &T );
printf ( "Ganho                    = " ); scanf ( "%f", &k );

exp_aT = exp( - a * T );
exp_2aT = exp( - 2 * a * T );

if ( jb ) {
    cos_bT = ( exp( b*T ) + exp( -b*T ) ) / 2.0;
    sen_bT = ( exp( b*T ) - exp( -b*T ) ) / 2.0;
}
else {
    cos_bT = cos( b*T );
    sen_bT = sin( b*T );
}

if ( b == 0 ) {
    TM [i-1][2] = c * T * exp_aT;
    TM [i-1][3] = c * T * exp_aT;
}
else {
    TM [i-1][2] = c * exp_aT * sen_bT / b;
    TM [i-1][3] = c * exp_aT * sen_bT / b;
}

TM [i-1][2] = k * ( 1 - exp_aT * cos_bT + TM [i-1][2] );
TM [i-1][3] = k * ( exp_2aT - exp_aT * cos_bT - TM [i-1][3] );
TM [i-1][0] = - 2 * exp_aT * cos_bT;
TM [i-1][1] = exp_2aT;

printf ( "\n\n\n\n\n\n" );
printf ( "a1 = %15.10f \n", TM [i-1][0] );
printf ( "a2 = %15.10f \n", TM [i-1][1] );
printf ( "b1 = %15.10f \n", TM [i-1][2] );
printf ( "b2 = %15.10f \n", TM [i-1][3] );
printf ( "\n\n\ntecle <ENTER>" ); getch ();
}

/***** inicializacao de variaveis *****/
Inicializacao ()
{
register char    i, j;

```



```

/* Resposta da planta */
dado = read_anl ( canal_Feed, offset_Feed );
yP = 5.0 * dado / 4096.0;

/* Resposta do estimador */
yE = 0;
for ( i = 0; i <= 2 * Ord_M - 1; i++ ) yE += XtE [i] * TE [i];

do {
/* resposta do modelo */
yM = 0;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    yM += XtM [i] * TM [NModelo-1][i];

/* atualizacao do modelo */
for ( i = 2 * Ord_M - 1; i >= 1; i-- ) XtM [i] = XtM [i-1];
XtM [0] = - yM;
XtM [Ord_M] = r;

/* tensao de controle da planta */
if ( r > 4.99512 ) r = 4.99512;
if ( r < 0 ) r = 0;
dado = (int) ( 1024.0 * r / 5.0 );
write_anl ( dado, canal_Planta );

/* atualizacao do estimador */
for ( i = 2 * Ord_M - 1; i >= 1; i-- ) XtE [i] = XtE [i-1];
XtE [0] = - yP;
XtE [Ord_M] = r;

/* calculo de A1 = P(K) * X */
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
{
    A1 [i] = 0;
    for ( j = 0; j <= 2 * Ord_M - 1; j++ )
        A1 [i] += P [i][j] * XtE [j];
}

/* calculo de G(K) */
G = ro;
for ( i = 0; i <= 2 * Ord_M - 1; i++ ) G += XtE [i] * A1 [i];
G = 1 / G;

/* calculo de A2 = X_transp * P(K) */
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
{
    A2 [i] = 0;
    for ( j = 0; j <= 2 * Ord_M - 1; j++ )
        A2 [i] += XtE [j] * P [j][i];
}

/* calculo de P(K+1) */
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    for ( j = 0; j <= 2 * Ord_M - 1; j++ )
        P [i][j] = ( P [i][j] - G * A1[i] * A2 [j] ) / ro;

```

```

/**** calculo de Teta_E ****/

/* Resposta da planta */
dado = read_anl ( canal_Feed, offset_Feed );
yP = 5.0 * dado / 4096.0;

/* Resposta do estimador */
yE = 0;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    yE += XtE [i] * TE [i];
erro = yP - yE;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    TE [i] += G * A1 [i] * erro;

if ((erro <= erro_PE) && (erro >= - erro_PE)) ro = 1;
                                     else ro = ro_inicial;

/* leitura da referencia */
dado = read_anl ( canal_Ref, offset_Ref );
r = 5.0 * dado / 4096.0;

key = ' ';
if ( kbhit ( ) ) key = getch ( );

switch ( key ) {
    case 'r' : Inicializacao ( );
              break;
}

while ( key != 27 );

printf ( "\n\nfim da estimacao de parametros\n\n" );

/**** controle adaptativo ****/

printf ( "modelo : %c\n", NModelo+48 );

/* leitura da referencia */
dado = read_anl ( canal_Ref, offset_Ref );
r = 5.0 * dado / 4096.0;
u = N * r;

/* tensao de controle da planta */
if ( u > 4.99512 ) u = 4.99512;
if ( u < 0 ) u = 0;
dado = (int) ( 1024.0 * u / 5.0 );
write_anl ( dado, canal_Planta );

/* Resposta da planta */
dado = read_anl ( canal_Feed, offset_Feed );
yP = 5.0 * dado / 4096.0;

/* Resposta do estimador */
yE = 0;
for ( i = 0; i <= 2 * Ord_M - 1; i++ ) yE += XtE [i] * TE [i];

```

```

do (
/* resposta do modelo */
yM = 0;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    yM += XtM [i] * TM [NModelo-1][i];

/* Resposta_C */
yC = N * r;

/* calculo de Kn e Xn */
for ( i = 0; i <= Ord_M - 1; i++ )
    Kn [i] = TM [NModelo-1][i] - TE [i];
dc_M = 0;
for ( i = 0; i <= Ord_M - 1; i++ )
    dc_M += TM [NModelo-1][i+Ord_M];
dc_E = 0;
for ( i = 0; i <= Ord_M - 1; i++ )    dc_E += TE [i+Ord_M];

if ( (dc_M == 0) || (dc_E == 0) ) N = 1;
    else N = dc_M / dc_E;

Xn1 = u;
for ( i = 0; i <= Ord_M - 1; i++ )    Xn1 -= TE [i] * Xn [i];
for ( i = Ord_M - 1; i >= 1; i-- )    Xn [i] = Xn [i - 1];
Xn [0] = Xn1;

/* atualizacao do modelo */
for ( i = 2 * Ord_M - 1; i >= 1; i-- )    XtM [i] = XtM [i-1];
XtM [0] = - yM;
XtM [Ord_M] = r;

/* atualizacao de u */
u = yC;
for ( i = 0; i <= Ord_M - 1; i++ )    u -= Kn [i] * Xn [i];

/* tensao de controle da planta */
if ( u > 4.99512 )    u = 4.99512;
if ( u < 0 )        u = 0;
dado = (int) ( 1024.0 * u / 5.0 );
write_anl ( dado, canal_Planta );

/* atualizacao do estimador */
for ( i = 2 * Ord_M - 1; i >= 1; i-- )    XtE [i] = XtE [i-1];
XtE [0] = - yP;
XtE [Ord_M] = u;

/* calculo de A1 = P(K) * X */
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    (
    A1 [i] = 0;
    for ( j = 0; j <= 2 * Ord_M - 1; j++ )
        A1 [i] += P [i][j] * XtE [j];
    )

/* calculo de G(K) */

```

```

G = ro;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )    G += XtE [i] * A1 [i];
G = 1 / G;

/* calculo de A2 = X_transp * P(K) */
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
{
    A2 [i] = 0;
    for ( j = 0; j <= 2 * Ord_M - 1; j++ )
        A2 [i] += XtE [j] * P [j][i];
}

/* calculo de P(K+1) */
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    for ( j = 0; j <= 2 * Ord_M - 1; j++ )
        P [i][j] = ( P [i][j] - G * A1[i] * A2 [j] ) / ro;

/** calculo de Teta_E */

/* Resposta da planta */
dado = read_anl ( canal_Feed, offset_Feed );
yP = 5.0 * dado / 4096.0;

/* Resposta do estimador */
yE = 0;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    yE += XtE [i] * TE [i];
erro = yP - yE;
for ( i = 0; i <= 2 * Ord_M - 1; i++ )
    TE [i] += G * A1 [i] * erro;

if ((erro <= erro_PE) && (erro >= - erro_PE)) ro = 1;
    else ro = ro_inicial;

/* leitura da referencia */
dado = read_anl ( canal_Ref, offset_Ref );
r = 5.0 * dado / 4096.0;

key = ' ';
if ( kbhit ( ) )    key = getch ( );

switch ( key )
{
    case '1' :
    case '2' :
    case '3' :
    case '4' :
    case '5' : NModelo = key - 48;
                printf ( "modelo : %c\n", NModelo+48 );
                Ord_M = Ord_Modelo [NModelo -1];
                break;
    case 'r' : Inicializacao ( );
                break;
}
}

```



```
unsigned char canal;
{
unsigned char ch_DA;

canal &= 0x07;
ch_DA = canal << 5;
ch_DA != canal;
ch_DA != BSHSA;
out_port ( base + CTL, ch_DA );
ch_DA &= ~BSHSA;
out_port ( base + CTL, ch_DA );
}

write_anl ( dado, canal )
unsigned int dado;
unsigned char canal;
{
unsigned char_ dado_MSB, dado_LSB;

dado_LSB = dado;
dado_LSB &= 0xFC;
dado_MSB = dado >> 8;
out_port ( base + DALSB, dado_LSB );
out_port ( base + DAMSB, dado_MSB );
sel_ch_DA ( canal );
}
```

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
BIBLIOTECA MAUÁ - BIM
ESTE LIVRO DEVE SER DEVOLVIDO NA ÚLTIMA
DATA ABAIXO:

03.02.03		
<u>13/03/03</u>		
<u>29/03/03</u>		
<u>17/04/03</u>		
<u>25.4.2003</u>		
<u>03/5/2003</u>		
10.05.03		
<u>23.5.2003</u>		
<u>30.5.2003</u>		
<u>06/06/03</u>		
<u>13/06/03</u>		
<u>21.6.2003</u>		
<u>05.11.2003</u>		
<u>12/11/03</u>		

UNIFEI - BIBLIOTECA MAUÁ
8201194



NÃO DANIFIQUE ESTA ETIQUETA