PÉRICLES REZENDE BARROS

PROJETO E IMPLEMENTAÇÃO DE CONTROLADORES DIGITAIS: A DETERMINAÇÃO DO COMPRIMENTO DE PALAVRA DE VARIÁVEIS E COEFICIENTES NA REPRESENTAÇÃO EM PONTO FIXO

Dissertação apresentada ao Curso de MESTRADO EM ENGENHARIA ELÉTRICA da Universidade Federal da Paraí ba, em cumprimento às exigências para obtenção do Grau de Mestre.

ÁREA DE CONCENTRAÇÃO: PROCESSAMENTO DA INFORMAÇÃO

GURDIP SINGH DEEP
Orientador

FERNANDO ANTONIO CAMPOS GOMIDE
Co-Orientador

CAMPINA GRANDE
FEVEREIRO-1985

B277p Barros, Péricles Rezende.
Projeto e implementação de controladores digitais : a determinação do comprimento de palavra de variáveis e coeficientes na representação em ponto fixo / Péricles Rezende Barros. - Campina Grande, 1985.
98 f.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1985.
"Orientação : Prof. Dr. Gurdip Singh Deep, Prof. Dr. Fernando Antonio Campos Gomide".
Referências.

1. Controladores Digitais - Projeto. 2. Processamento da Informação. 3. Ponto Fixo. 4. Dissertação - Engenharia Elétrica. I. Deep, Gurdip Singh. II. Gomide, Fernando Antonio Campos. III. Universidade Federal da Paraíba - Campina Grande (PB). IV. Título

CDU 621:004.2(043)

Aos meus pais,

Pedro e Eunice.

AGRADECIMENTOS

A todos aqueles que durante o período deste trabalho participaram através de orientações, discussões e esclarecimentos, formando um ambiente de incentivo à procura;

Em especial, aos meus orientadores, Prof.Dr. Gurdip Singh Deep e Prof. Dr. Fernando Antonio Campos Gomide, pelo incentivo continuado e pelo ambiente amigável proporcionado, tornando agradável a realização do mesmo.

Aos professores do Setor de Computação e Automação Industrial da Faculdade de Engenharia de Campinas, UNICAMP, pela gentileza na cessão dos equipamentos usados. Particularmente, aos Prof. Dr. Wagner Caradori do Amaral e Prof. José Raimundo de Oliveira.

Ao Centro Tecnológico para Informática pelo constante apoio na realização deste trabalho.

A Olga Regina S. Morales, Domingos Savio Xavier Cavalcanti e Roberto de Oliveira pela dedicação na elaboração desta dissertação.

RESUMO

Este trabalho apresenta o projeto de controladores digitais determinísticos ótimos, obtidos através da minimização de um critério de desempenho quadrático. Dois destes controladores usam observadores de estados, sendo diferenciados por suas estruturas. O terceiro controlador é projetado como uma aproximação dinâmica do controlador ótimo.

É apresentada uma técnica estatística para o cálculo do comprimento de palavra de variáveis, cujo parâmetro de projeto é o ruído máximo aceitável, devido à quantização de variáveis, na saída dos controladores. Em adição é proposta uma técnica, também estatística, para a determinação do comprimento de palavra de coeficientes a partir da definição da variação máxima aceitável da saída dos controladores quantizados em relação aos controladores ideais, não quantizados.

O projeto dos controladores, assistido por computador, é também apresentado.

A implementação em um sistema baseado em um microprocessador de oito bits de dois controladores com observadores é descrita. A representação numérica usada na implementação é complemento de dois, ponto fixo, com arredondamento. O processo utilizado como objeto de controle na im

plementação é um servomotor de corrente contínua com carga inercial.

ABSTRACT

In this thesis, design of linear optimal deterministic digital controllers based on minimization of a quadratic performance index is presented. In the design of two of these controllers, state observers have been employed, though their structures are different. The third controller is based on the dynamic approximation of the optimal controller.

A statistical technique for determining the wordlength of variables is presented. The quantization of variables is modelled as noise sources entering the controller. The definition of a maximum allowed noise at the controller's output is used for determining that wordlength. A technique, also statistical, for determining the wordlength of coefficients is proposed. In this technique, the maximum allowed control signal variation due to quantization determines the wordlength. In this case, coefficients variations are also modelled as noise sources.

The computer aided design of these controllers is described. Finally, implementation of two controllers using state observers, based on an 8 bit microprocessor is presented, wherein fixed point, two's complement number representation with rounding has been employed. The d.c. servomotor with inertial load constitutes the process under control.

S U M Á R I O

LISTA DE ILUSTRAÇÕES

1. INTRODUÇÃO

1.1. Motivação

No projeto de controladores industriais digitais, por razões históricas (substituição de controladores analógicos) e pela simplicidade aliada a um bom desempenho, os controladores P.I.D. (proporcional-integral-derivativo) foram e são utilizados largamente. Tal fato extende-se também ao projeto de controladores digitais de posição e velocidade de servomotores, como exemplificado por Gauen (1983).

No entanto, o aparecimento de microprocessadores, com capacidade computacional a baixo custo, possibilitou a implementação de técnicas de controle mais sofisticadas já desenvolvidas, e incentivou o desenvolvimento de outras técnicas. Diversos trabalhos apresentam a implementação destes controladores, como Ortega (1982), Jing-Ping & Marleau (1982), Stojič (1984) e Barros (1981).

O uso de microprocessadores na implementação de controladores evidenciou a necessidade da determinação do com primento de palavra de variáveis e coeficientes, tendo sido gerada uma série de trabalhos abordando o tema, como Moroney, Willsky & Houpt (1980), Fam (1982), Moroney, Willsky & Houpt (1983).

A teoria de controle ótimo digital mesmo sendo uma teoria bem estabelecida, apresentada em livros texto clássicos como Kwakernaak & Sivan (1972), Franklin & Powell (1980) e Isermann (1981), ainda desperta grande interesse na literatura, devido à sua potencialidade não explorada na prática industrial. Assim, trabalhos nesta área ainda são publicados, como o de Tsuchiya (1982).

Seguindo o objetivo de implementação de controladores mais sofisticados, este trabalho se propõe ao projeto de um controlador usando as técnicas de projeto de controle ótimo e de determinação de comprimento de palavra de variável e coeficientes a serem utilizados na implementação em microprocessador.

1.2. Objetivos

São objetivos deste trabalho:

. Apresentar a modelagem de um processo a ser utilizado, isto é, o modelo de um servomotor c.c. com carga inercial.

. Apresentar o projeto de um controlador linear digi tal ótimo, bem como técnicas de realizar simplificações em sua estrutura, de maneira a viabilizar sua implementação baseada em microprocessadores. Implementá-lo em um microprocessador Intel 8085.

. Apresentar e propor técnicas de determinação do comprimento de palavra de variáveis e coeficientes, usando a representação de números em ponto fixo.

. Realizar o projeto dos controladores auxiliado por computador.

. Analisar os resultados obtidos e apresentar as conclusões.

1.3. Revisão Bibliográfica

Este item apresenta uma revisão bibliográfica sobre os assuntos tratados neste trabalho. São apresentados a seguir os trabalhos mais representativos nas áreas de controle ótimo digital e estudo de efeito de comprimento de palavra de variáveis e coeficientes em filtros e controladores digitais, bem como são referenciados trabalhos sobre o controle de posição de servomotores de corrente contínua.

1.3.1. Controle Ótimo Digital

Em controle ótimo digital, o trabalho de Kalman &

Koepcke (1958) é colocado como o marco inicial no projeto de controladores digitais pela minimização de um índice de desempenho quadrático. Recentemente Dorato & Levis (1971) extenderam o trabalho de Kalman & Koepcke, baseando também em programação dinâmica para a determinação das equações de otimização. Livros texto como Kwakernaak & Sivan (1972) , Anderson & Moore (1971), Kuo (1980) e Isermann (1981), dentre outros, apresentam o projeto de controladores e reguladores através da minimização de um índice de desempenho quadrático.

Tsuchiya (1982) também desenvolve o projeto de reguladores ótimos.

Os observadores de estados tiveram nos trabalhos de Luenberger (1964)(1966)(1971) sua introdução e desenvolvimento posterior. Os livros texto mencionados anteriormente incluem o desenvolvimento de observadores.

Na transformação do projeto de um servomecanismo num problema de projeto de um regulador ótimo, o trabalho de Johnson (1971) é sempre referenciado pelos livros texto já citados, dentre outros.

1.3.2. Efeito de Comprimento de Palavra de Variáveis e Coeficientes em Filtros e Controladores Digitais

Os trabalhos de Knowles & Edwards (1965) Bertram -

(1958) e Slaughter (1964) foram os primeiros a introduzirem o aspecto de efeitos de quantização, sendo desenvolvidos para controladores digitais.

Baseados nestes trabalhos, pesquisadores na área de processamento digital de sinais extenderam as técnicas para filtros digitais, tendo os resultados sido publicados por Knowles & Alcayto (1968), Avenhaus (1972), Crochiere (1975), Crochiere & Oppenheim (1975), Oppenheim & Weisntein (1972),, Claasen & Mecklenbräuker (1975) , Jackson (1970) e outros.

Mais recentemente, trabalhos na área de controle digital têm sido apresentados, citando como exemplo Moroney , Willsky & Houpt (1980) e (1983), Fam (1982), Ahmed & Belanger (1984a) e (1984b).

1.3.3. Controle de Posição de Servomotores Corrente Contínua

No projeto de controle de posição de servomotores os controladores mais usados são dos tipos P,PI e PID, de acordo com Gauen (1983). Trabalhos como este baseiam-se nos desenvolvimentos de Chiu, Corripio & Smith (1973a), (1973b) e (1973c), Lopez, Murrill & Smith (1969).

Controladores do tipo auto-ajustáveis bem como aqueles baseados em estruturas variáveis foram desenvolvidas e implementados por Ortega (1982) e Jing-Ping & Marleau (1982) e Barros (1981).

1.4. Descrição do Trabalho

No próximo capítulo são apresentadas a descrição do sistema de controle e a modelagem do processo a ser usado neste trabalho. No capítulo 3 é apresentado o projeto dos controladores. As técnicas de determinação do comprimento de palavra de variáveis e coeficientes são apresentadas no capítulo 4. O projeto assistido por computador e a implementação dos controladores são discutidos nos capítulos 5 e 6, respectivamente. Por fim, no capítulo 7 são analisados os resultados e no capítulo 8 as conclusões são apresentadas.

A apresentação deste trabalho procura incluir o aspecto didático, de modo a prover ao leitor os subsídios necessários à absorção dos temas discutidos, de uma foram auto contida.

2. DESCRIÇÃO DO SISTEMA E MODELAGEM DO PROCESSO

2.1. Introdução

No controle de qualquer sistema é necessário o conhecimento de seu comportamento dinâmico. Este comportamento é muitas vezes descrito em termos de equações matemáticas as quais representam uma aproximação do sistema real. A obtenção destas equações é usualmente conhecida como modelagem.

Este capítulo introduz o processo a ser controlado , através de uma descrição do seu funcionamento. A seguir este processo é modelado em termos de equações dinâmicas contínuas. Após algumas considerações adicionais o modelo discreto no tempo é determinado.

2.2. - Descrição do Sistema

A figura 2.1 apresenta o sistema em diagrama de blocos. O processo a ser controlado consiste de um servomotor de corrente contínua (C.C.) acoplado a uma carga inercial . No microcomputador tem-se implementado o controlador. A interface no sentido controlador-processo é um conversor digi

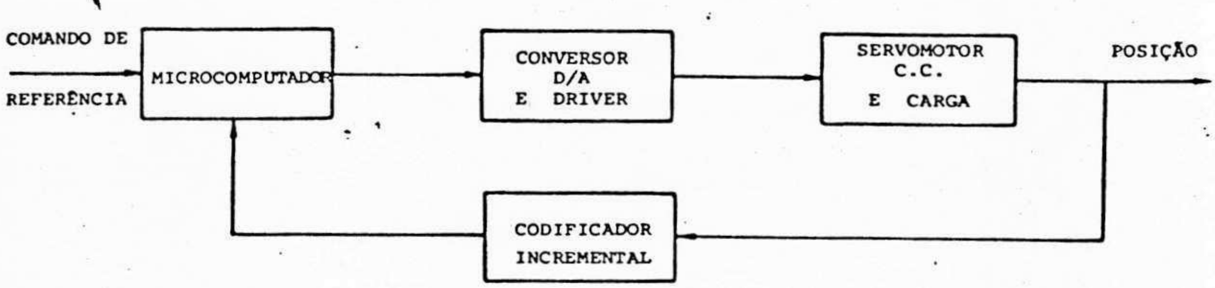


Figura 2.1 - O sistema em diagrama de blocos.

tal-analógico associado a um dispositivo de potência ("dri - ver"). No sentido processo-controlador a interface é um codificador incremental. A seguir o funcionamento do sistema é discutido suscintamente, baseando-se na figura 2.2.

O objetivo é que o controlador mantenha a posição angular da carga de acordo com um dado referêncial (comando de referência) fornecido através de uma interface de comunica - ção, mostrada na figura 2.2.. Se este referencial for constante, o sistema deve manter a posição angular da carga constante, compensando o efeito de perturbações externas. Se este referencial for variável, o sistema deve ter a posição angular da carga variando de acordo com o referencial, também compensando o efeito de perturbações externas.

O microprocessador, executando o programa correspon - dente ao controlador, determina e emite um sinal de controle Para tal, são comparados o referencial e a informação correspondente à posição angular. Esta informação é obtida utilizando-se como transdutor de deslocamento um codificador incremental bidirecional.

O codificador incremental bidirecional é um dispositivo que gera duas sequências de pulsos, cada sequência relativa a um sentido de rotação, com cada pulso correspondendo a um deslocamento de $2\pi/p$ rad naquele sentido de rotação, onde p é o número de pulsos por revolução. Cada sequência é utilizada para decrementar um contador correspondente, de ma

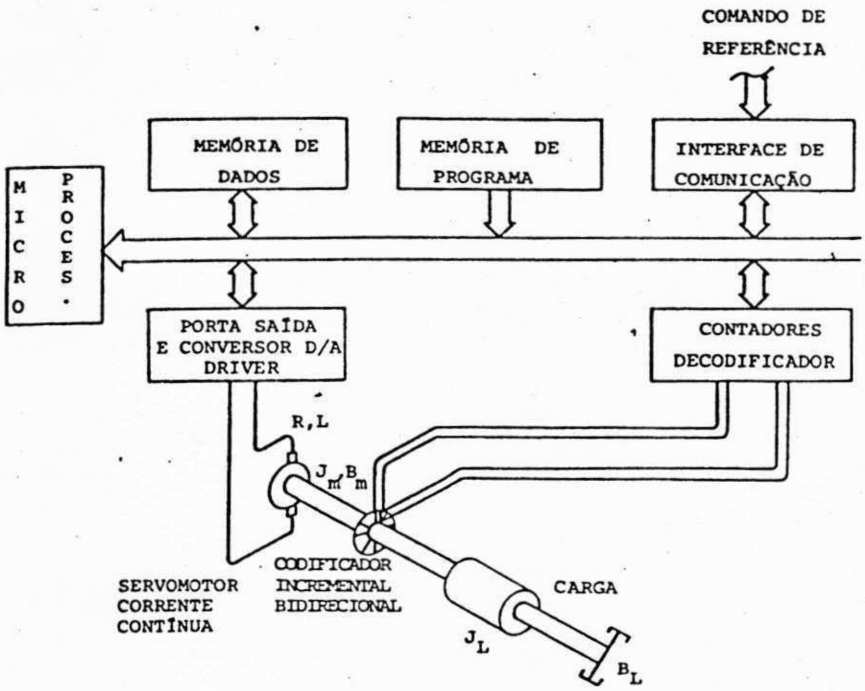


Figura 2.2 - Configuração do sistema.

neira que a cada medição pode-se calcular o deslocamento o corrido desde a última medição, em cada sentido, bem como o deslocamento total no mesmo período. Com este deslocamento calcula-se então a posição angular atual, a partir da posição angular anterior. As operações aritméticas necessárias são realizadas pelo microcomputador.

A diferença entre o referencial e a posição atual fornece o erro de posição, que é o sinal de entrada do controlador.

O controlador, com o erro de posição e outras variáveis e parâmetros internos presentes na memória de dados, calcula o sinal de controle.

O sinal de controle, de natureza digital, é convertido em um sinal de natureza analógica (conversor d/a) com energia suficiente ("driver" de potência) para compensar o erro existente.

O procedimento acima descrito é repetido a cada T segundos, onde T é o período de amostragem.

2.3. Modelagem do Processo

No projeto de controladores é frequentemente necessário determinar um modelo matemático que descreva o comportamento dinâmico do processo. Esta secção apresenta o modelo do processo a ser controlado.

2.3.1. Diagrama de Blocos e de Estados do Processo

O modelo associado ao processo considerado neste trabalho tem como entrada um sinal de controle, $u_c(kT)$, e como saída a posição angular, $d(kT)$, de acordo com o diagrama de blocos da figura 2.3. Este diagrama mostra as funções de transferência associadas ao processo, acrescido da função de transferência correspondente à operação de determinação da posição angular atual.

Como o sinal de controle é digital, e o processo analógico, é necessária a introdução de um bloco que execute a conversão entre as duas formas. Este bloco, denominado z.o.h. (de "zero-order hold"), tem como função tornar o sinal $u_c(kT)$ válido também durante o intervalo $[kT, (k+1)T)$. Assim, matematicamente,

$$u_k(t) = u_c(kT) \quad , \qquad kT \leqslant t < (k+1)T \tag{2.1}$$

A transformada de Laplace da equação acima, no intervalo $kT \leqslant t < (k+1)T$, é

$$U_k(s) = \frac{u_c(kT)}{s} \tag{2.2}$$

O diagrama de blocos do conjunto servomotor - carga é dado na figura 2.3 pela parte entre as variáveis $u_K(t)$ e $\omega(t)$.

Ao deslocamento realizado durante cada período de amostragem corresponde uma variável velocidade, expressa em

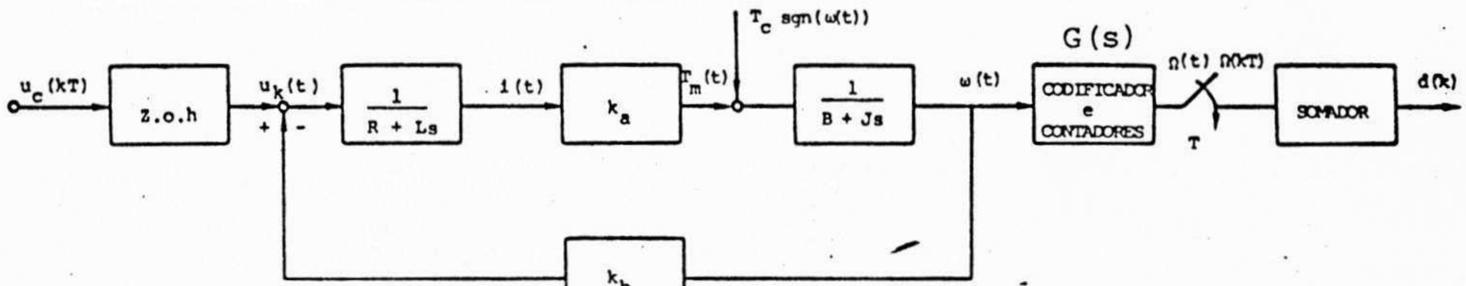


Figura 2.3 - Diagrama de blocos do processo.

número de pulsos por período de amostragem. Esta velocidade, $\Omega(kT)$, digital, é obtida através da leitura, a cada T segundos, do número de pulsos gerados a partir do codificador incremental com resolução de p pulsos/revolução. Para obtê-la da velocidade angular $\omega(t)$, em rad/s, usa-se o bloco codificador e contadores, cuja função de transferência é dada por:

$$G_s = \frac{pT}{2\pi} \tag{2.3}$$

pois, como uma velocidade de 2π rad/s corresponde a p pulsos/s, a velocidade em pulsos por período de amostragem é dada multiplicando-se por T a velocidade em pulsos/s.

A posição angular, $d(kT)$, em pulsos, é determinada pela adição, a cada período de amostragem, da posição no período anterior, $d((k-1)T)$, com o deslocamento ocorrido entre estes dois períodos de amostragem, numericamente igual à velocidade $\Omega(kT)$ pulsos por período de amostragem. Assim, o bloco correspondente, o somador, tem função de transferência no plano Z dada por

$$D(z) = \frac{1}{1-z^{-1}} \ \Omega\ (z) \tag{2.4}$$

Para o projeto dos controladores digitais é necessário expressar o processo em termos de um modelo discreto no tempo. Como na maioria dos servomotores de corrente contínua a indutância é muito pequena (Dorf (1973,p.40) e D'azzo

Houpis (1975, p. 58)), e como a fricção de Coulomb, dada por

$$T_C \text{ sgn } (\omega(t)), \tag{2.5}$$

onde

$$\text{sgn}(x) = \begin{cases} 1 & \text{se} \quad x > 0 \\ -1 & \text{se} \quad x < 0 \end{cases},$$

é constante em módulo, variando apenas em sinal, usualmente ela é compensada usando-se uma alimentação direta ("feedforward") de

$$\frac{T_C R}{k_a} \text{sgn } (\omega(t)) \tag{2.6}$$

na tensão de armadura do servomotor, já que o atraso introduzido pela parte elétrica do servomotor é despresível em relação à dinâmica da parte mecânica . Isto corresponde à aproximação mostrada na figura 2.4.

O sinal de controle a ser aplicado ao processo tem a seguinte forma

$$u_c(kT) = u(kT) + u_f(kT) \tag{2.7}$$

onde

$$u_f(kT) = - \frac{T_C \; R}{k_a} \; \text{sgn } (\Omega(kT)) \tag{2.8}$$

corresponde à discretização da equação (2.6). u(kT) é o controle a ser determinado para o sistema linear que representa o processo, baseado no sinal de realimentação.

Para pequenas indutâncias, o efeito de sua inclusão

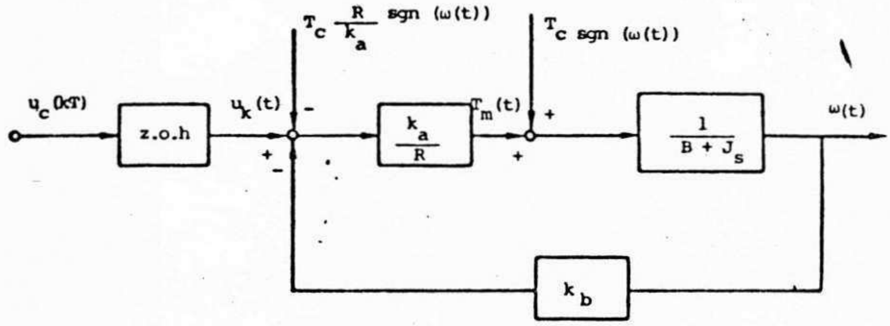


Figura 2.4 - Alimentação direta (Feedforward)

(ou não) é desprezível. Na determinação do comprimento de palavra de coeficientes, (Capítulo 4), e na implementação dos controladores (capítulo 6) será visto que alguns coeficientes podem ser eliminados se os seus arredondamentos dão como resultados zero. Neste trabalho tais coeficientes estão associadas à parte elétrica do servomotor. Por isso, não obstante a aproximação da figura 2.4, a indutância será reintroduzida na análise a seguir.

O diagrama de estados amostrado para a parte analógica (no plano s) do processo é apresentado pela figura 2.5.. A partir deste diagrama pode-se obter uma representação de natureza discreta no tempo para o processo (KUO (1980, p.231 a 237)). Uma representação é determinada a seguir.

2.3.2. Representação do Processo em Variáveis de Estado

Usando a regra de Mason para malhas digitais ao diagrama de estados amostrado da figura 2.5 tem-se as malhas fechadas

$$M_1 = -(R/L)\ s^{-1}$$
$$M_2 = -(B/J)\ s^{-1}$$
$$M_3 = -(K_a K_b/LJ)\ s^{-2}. \qquad (2.9)$$

Assim

$$\Delta = 1 - \left[-(R/L)s^{-1} - (B/J)s^{-1} - (K_a K_b/LJ)s^{-2}\right] + (RB/LJ)s^{-2} \qquad (2.10)$$

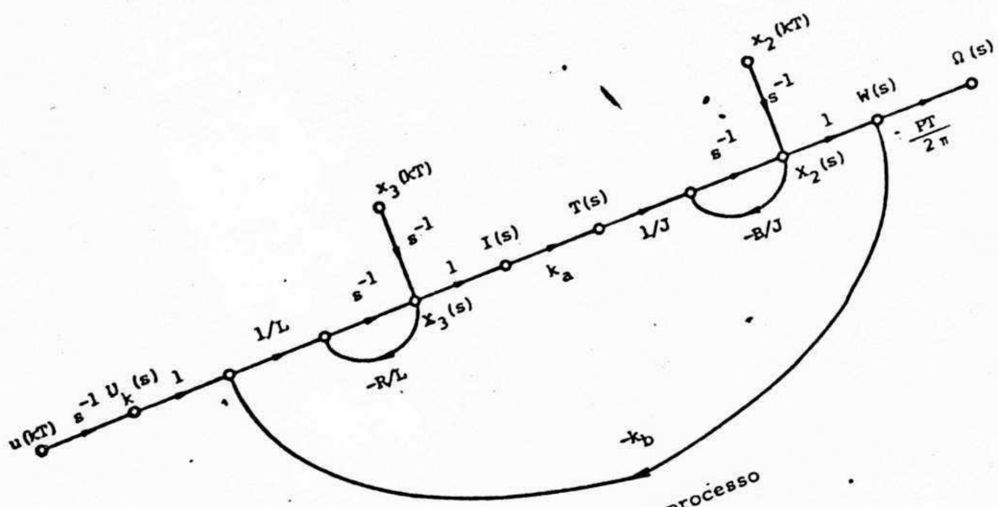


Figura 2.5 - Diagrama de estados do processo

ou

$$\Delta = 1 + \left[\frac{R}{L} + \frac{B}{J}\right] s^{-1} + \left[\frac{K_a K_b + RB}{LJ}\right] s^{-2} \quad (2.11)$$

Usando a fórmula do ganho de Mason (KUO (1980,p. 231 a 237)) obtêm-se as equações apresentadas na forma de variáveis de estado

$$\begin{bmatrix} X_2(s) \\ \\ X_3(s) \end{bmatrix} = \begin{bmatrix} \dfrac{s + R/L}{s^2 + as + b} & \dfrac{K_a/J}{s^2 + as + b} \\ \\ \dfrac{-K_b/L}{s^2 + as + b} & \dfrac{s + B/J}{s^2 + as + b} \end{bmatrix} \begin{bmatrix} x_2(kT) \\ \\ x_3(kT) \end{bmatrix} + \begin{bmatrix} \dfrac{K_a/LJ}{s(s^2 + as + b)} \\ \\ \dfrac{(s + B/J)/L}{s(s^2 + as + b)} \end{bmatrix} u(kT)$$

(2.12)

onde

$$a = \frac{R}{L} + B/J$$

$$b = \frac{RB + K_a K_b}{LJ} \quad , \text{ e} \qquad (2.13)$$

onde B é a fricção e J, a inércia do conjunto servomotor - carga , ou seja

$$B = B_m + B_\ell$$

$$J = J_m + J_\ell \qquad (2.14)$$

Observe que, $X_2(s)$ corresponde à velocidade angular e $X_3(s)$

à corrente de armadura.

Efetuando a transformada inversa de Laplace sobre a equação (2.8), no intervalo $kT \leqslant t \leqslant (k+1)T$, e fazendo $t = (k+1)T$, chega-se à representação em variáveis de estado discretas no tempo

$$\begin{bmatrix} x_2 \; [(k+1)T] \\ \\ x_3 \; [(k+1)T] \end{bmatrix} = \begin{bmatrix} a_{22} & a_{23} \\ \\ a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_2 \,(kT) \\ \\ x_3 \,(kT) \end{bmatrix} + \begin{bmatrix} b_2 \\ \\ b_3 \end{bmatrix} u(kT) \quad (2.15)$$

onde

$$a_{22} = \mathcal{L}^{-1} \left\{ \frac{s + R/L}{s^2 + as + b)} \right\} ,$$

$$a_{23} = \mathcal{L}^{-1} \left\{ \frac{K_a/J}{s^2 + as + b} \right\} ,$$

$$a_{32} = \mathcal{L}^{-1} \left\{ \frac{- K_b/L}{s^2 + as + b} \right\} ,$$

$$a_{33} = \mathcal{L}^{-1} \left\{ \frac{s + B/J}{s^2 + as + b} \right\} ,$$

$$b_2 = \mathcal{L}^{-1} \left\{ \frac{K_a/LJ}{s(s^2 + as + b)} \right\} ,$$

$$b_3 = \mathcal{L}^{-1} \left\{ \frac{(s + B/J)/L}{s(s^2 + as + b)} \right\} , \qquad (2.16)$$

O diagrama de estados do processo discretizado no tempo é mostrado na figura 2.6. Para a obtenção da representação em variáveis de estado para o processo incluindo o somador, note que a equação (2.4) é equivalente a

$$d(kT) = d[(k-1)T] + \Omega(kT) \tag{2.17}$$

Substituindo kT por (k+1)T na equação acima, e, como

$$\Omega(kT) = \frac{PT}{2\pi} \quad x_2(kT) \tag{2.18}$$

obtém-se

$$d\ [(k+1)T] = d(kT) + \frac{PT}{2\pi} \quad x_2\ [(k+1)T] \tag{2.19}$$

Da equação (2.15) chega-se a

$$d\ [(k+1)T] = d(kT) + \frac{PT}{2\pi}\, a_{22}\, x_2\,(kT) + \frac{PT}{2\pi}\, a_{23}\, x_3\,(kT)$$
$$+ \frac{PT}{2\pi}\, b_2 \quad u(kT) \tag{2.20}$$

Definindo-se uma nova variável de estado $x_1(kT) = d(kT)$, chega-se, finalmente, às equações dinâmicas

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u(k) \tag{2.21}$$

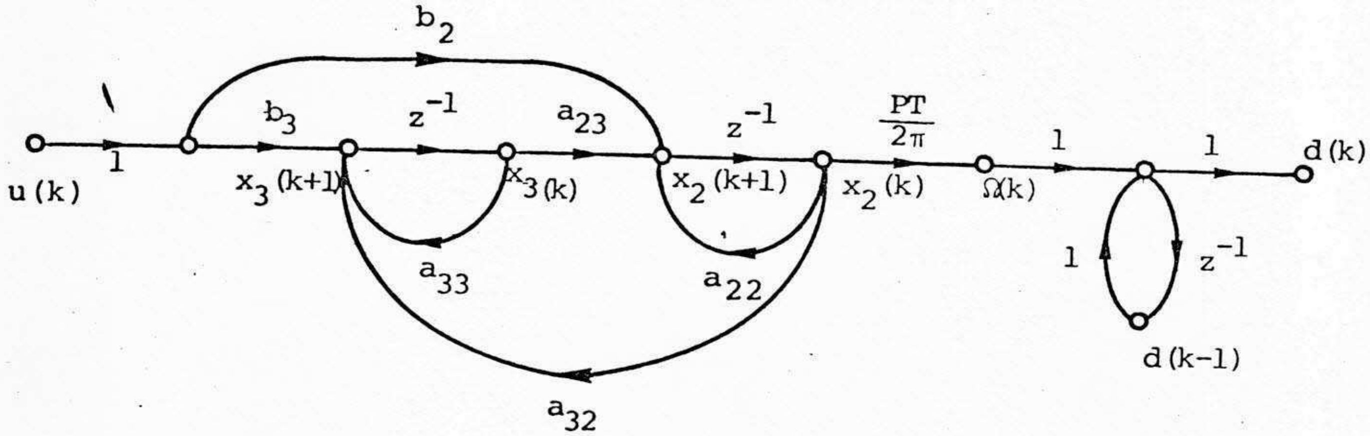


Figura 2.6 . Diagrama de estados do processo discretizado no tempo (incluindo somador).

e

$$y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \tag{2.22}$$

com

$$\begin{aligned} a_{12} &= \frac{PT}{2\pi} \quad a_{22} \;, \\ a_{13} &= \frac{PT}{2\pi} \quad a_{23} \;, \\ b_1 &= \frac{PT}{2\pi} \quad b_2 \;, \end{aligned} \tag{2.23}$$

e onde $y(k) = d(k)$ é a posição angular expressa em número de pulsos. Nas equações (2.21) e (2.22) o período de amostragem, T, foi incluído implicitamente de maneira que a amostragem $\underline{k}$ corresponde ao tempo $\underline{k}T$.

2.4 - Resumo

Este capítulo tratou da modelagem do processo a ser controlado usado neste trabalho. Inicialmente foi apresentada um descrição do sistema, objetivando a apresentação do problema de controle. A seguir o processo foi modelado, obtendo-se a partir do seu diagrama de blocos (contínuo) o diagrama de estados e a correspondente representação em variáveis de estados (discretos). A não linearidade existente foi compensada usando-se alimentação direta (feedforward).

3. PROJETO DOS CONTROLADORES

3.1. Introdução

Este capítulo trata do projeto de três controladores lineares digitais determinísticos obtidos através da minimização de um índice de desempenho quadrático. Os dois primeiros controladores utilizam observadores de estados, cujas estruturas são diferenciadas por transformações lineares aplicadas ao conjunto controlador ótimo-observador de estados. O terceiro é uma aproximação dinâmica do controlador ótimo que leva a uma realimentação da saída.

No item 3.2 é apresentado o problema do regulador digital linear ótimo e, no item seguinte o projeto do observador de estados. No item 3.4 é apresentado o projeto do controlador digital linear ótimo para o problema de um servomecanismo, a partir do projeto do regulador. No item 3.5. apresenta-se um modo de implementação baseado na diagonalização do conjunto controlador-observador, levando ao controlador I. O controlador II é determinado após algumas reduções na estrutura do conjunto controlador-observador e é apresentado no item 3.6. Finalmente, no item 3.7 é apresentado o controlador III (controlador dinâmico).

3.2. Regulador Digital Linear Ótimo

Considere um sistema linear invariante descrito por

$$\underline{x}(k+1) = A\ \underline{x}(k) + \underline{b}\ u(k) \tag{3.1}$$

$$y(k) = \underline{d}^t\ \underline{x}(k) \tag{3.2}$$

onde

$\underline{x}(k)$ é o vetor de variáveis de estado,

$u(k)$ é a variável de entrada,

$y(k)$ é a variável de saída,

sendo

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} , \qquad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

e

$$\underline{d}^t = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} , \tag{3.3}$$

com os coeficientes a_{ij}, b_i $(i,j = 1,2,3)$ obtidos a partir das equações (2.16) e (2.23).

As equações acima são representadas graficamente na figura 3.1, onde $\underline{x}(0)$ é o estado inicial (k=0).

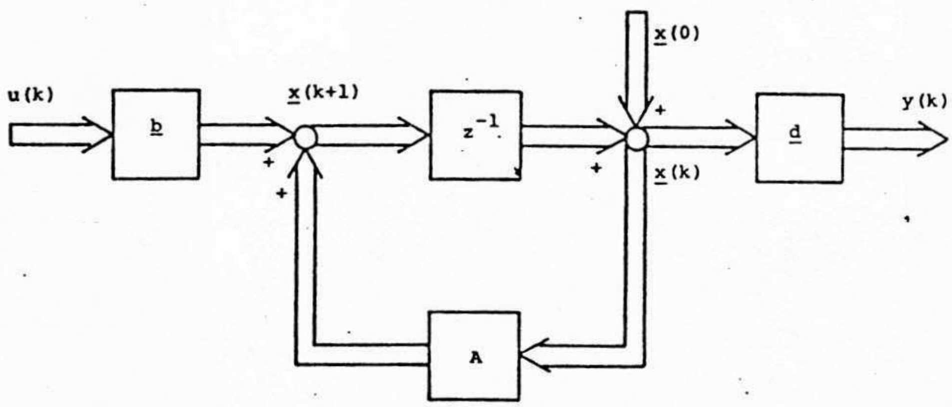


Figura 3.1 - Representação do sistema na forma de variáveis de estado.

O regulador digital linear ótimo é baseado na realimentação das variáveis de estado, $\underline{x}(k)$, de maneira a conduzir o sistema de um estado $\underline{x}(0)$ ao estado $\underline{x}(N) \simeq \underline{0}$, minimizando o índice de desempenho quadrático

$$I_x = \underline{x}^t(N) Q\, \underline{x}(N) + \sum_{k=0}^{N-1} \left[\underline{x}^t(k)\, Q\, \underline{x}(k) + r\, u^2(k) \right] \qquad (3.4)$$

onde

Q é uma matriz simétrica positiva definida, isto é, $\underline{x}^t(k)\, Q\, \underline{x}(k) > 0 \quad \forall k$, e

r escalar positivo $\quad (r\, u^2(k) > 0) \quad \forall k.$

Como mostrado no apêndice A1 (e em Isermann (1981 p. 136 a 143)) o regulador é dado por

$$u(N-j) = - \underline{K}_{N-j} \quad \underline{x}(N-j) \qquad j=1,\ldots,N \qquad (3.5)$$

onde $\underline{K}_{N-j}$ é o vetor de realimentação de estados obtido das equações recursivas

$$\underline{K}_{N-j} = (r + \underline{b}^t P_{N-j+1} \underline{b})^{-1}\, \underline{b}^t\, P_{N-j+1}\, A \qquad (3.6)$$

$$P_{N-J} = Q + A^t\, P_{N-j+1}\, A - \underline{K}^t_{N-j}\, (r + \underline{b}^t\, P_{N-j+1} b)\, \underline{K}_{N-j} \qquad (3.7)$$

com $P_N = Q$ como condição de contorno. A última equação é conhecida como equação à diferenças matricial de Riccati.

Fazendo $N \to \infty$, tem-se que $\underline{K}_{N-j}$ converge para um ve tor constante

$$\underline{K} = \lim_{N\to\infty} \underline{K}_{N-j}\ , \tag{3.8}$$

se o sistema é controlável e observável completamente, ou é estável exponencialmente.

Assim, o regulador torna-se

$$u(k) = -\underline{K}\ \underline{x}(k) \tag{3.9}$$

com $\underline{K}$ obtido das equações

$$\underline{K} = (r + \underline{b}^t\ \overline{P}\ \underline{b})^{-1}\ \underline{b}^t\ \overline{P}\ A \tag{3.10}$$

$$\overline{P} = \lim_{N\to\infty} \overline{P}_{N-j} = Q + A^t\ \overline{P}\left[I-\underline{b}(r+\underline{b}^t\ P\ \underline{b})^{-1}\ \underline{b}^t\ \overline{P}\right] A \tag{3.11}$$

obtidas das equações (3.6) e (3.7). A equação (3.11) é chamada equação estacionária de Riccati.

O sistema em malha fechada é então

$$\underline{x}(k+1) = \left[A - \underline{b}\ \underline{K}\right] \underline{x}(k), \tag{3.12}$$

sendo a equação característica associada

$$\det \left[z\,I - A + \underline{b}\ \underline{K}\right] = 0 \tag{3.13}$$

3.3. Observador de Estados

No projeto do regulador do item anterior, assumiu-se implicitamente que todas as variáveis de estado são diretamente mensuráveis. Na realidade, $x_2(k)$, correspondente à velocidade, é obtida diretamente do codificador incremental,e $x_1(k)$, correspondente à posição, é diretamente calculável a partir da posição no período anterior e do deslocamento (ou velocidade média) entre o período anterior e ao atual, como visto no capítulo anterior. No entanto, $x_3(k)$, correspondente à corrente de armadura, não é mensurável pelo hardware existente nem calculável diretamente. Assim, torna-se necessária a sua obtenção através de software. Isto é conseguido com o uso de um observador de estados.

Existem basicamente dois tipos de observadores de estados: os observadores de estados de ordem completa e os observadores de estados de ordem reduzida. No primeiro assume-se que todos os estados precisam ser obtidos, enquanto no segundo apenas os estados não disponíveis precisam ser determinados.

Os observadores de estado de ordem completa levam a um projeto mais transparente, e , se o número de variáveis disponíveis não for grande em relação ao número de variáveis restante, geralmente requerem esforço computacional (número de multiplicações exigidas) da ordem do esforço computacional dos observadores de ordem reduzida (Isermann (1981, p. 179)). Como será visto no item 3.5, o observador de ordem completa leva a uma solução para o problema de atraso no cálculo de u(k) a partir das variáveis na amostra k. E,além disso, como observadores de estados têm uma característica dinâmica, com o observador de ordem completa os ruídos presentes nas medidas das variáveis de saída são filtrados, fornecendo assintoticamente estados observados mais próximos dos reais. Assim, o observador de ordem completa é selecionado e de agora em diante será referenciado apenas como observador de estados.

O observador de estados para o sistema representado pelas equações (3.1), (3.2), e (3.3) tem a seguinte forma

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u(k) + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} [y(k) - \hat{y}(k)] \tag{3.14}$$

$$\hat{y}(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} \tag{3.15}$$

onde os termos $\hat{x}_i$ $(i=1,2,3)$ são os estados observados. Note que o observador acima compreende um termo com a mesma estrutura do sistema a ser observado, e um fator de correção, cujo objetivo é minimizar o efeito dos erros decorrentes da impossibilidade de obtenção de maneira precisa, dos coeficientes do sistema real.

Para a obtenção dos coeficiente h_i $(i=1,2,3)$, do fator de correção, considere o vetor erro entre as variáveis de estado do sistema e do observador, ou seja

$$\underline{\tilde{x}}(k) = \underline{x}(k) - \underline{\hat{x}}(k) \qquad (3.16)$$

Das equações (3.1), (3.3), (3.14) e (3.15) obtém-se

$$\begin{bmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \\ \tilde{x}_3(k+1) \end{bmatrix} = \left\{ \begin{bmatrix} 1 & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \right\} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \\ \tilde{x}_3(k) \end{bmatrix} \qquad (3.17)$$

ou, em forma compacta.

$$\underline{\tilde{x}}(k+1) = \left[A - \underline{h}\underline{d}^t\right] \underline{\tilde{x}}(k) \qquad (3.18)$$

Da equação acima conclui-se que o vetor erro entre as variáveis de estado do sistema e as variáveis de estado do observador depende apenas do estado inicial deste vetor erro, independendo da entrada u(k).

Obviamente, deseja-se que o erro convirja para um estado zero, ou

$$\lim_{k\to\infty} \underline{\tilde{x}}(k+1) = \underline{0} \qquad (3.19)$$

Para calcular os h_i's de maneira que a equação acima seja válida, considere que a equação característica do sistema da equação (3.18) é

$$\det\left[zI - A + \underline{h}\underline{d}^t\right] = 0 \qquad (3.20)$$

Como $\det[zI - A + \underline{h}\underline{d}^t] = \det[zI - A + \underline{h}\underline{d}^t]^t$ tem-se também que $\det[zI - A^t + \underline{d}\,\underline{h}^t] = 0.$ (3.21)

Comparando com a equação (3.13), a equação acima pode ser considerada como a equação característica do sistema em malha fechada obtido da regulação sobre o sistema

$$\underline{x}'(k+1) = A^t\underline{x}'(k) + \underline{d}\ u'(k) \qquad (3.22)$$

através da minimização do índice de desempenho quadrático

$$I_{\underline{x}'} = \lim_{N\to\infty}\left\{\underline{x}'^t(N)\ Q\underline{x}'(N) + \sum_{k=0}^{N-1}\left[\underline{x}'^t(k)\ Q\underline{x}'(k) + r(u'(k))\right]\right\} \qquad (3.23)$$

Assim, os parâmetros h_i's (i=1,2,3) são obtidos como a solução das equações

$$\underline{h}^t = (r + \underline{d}^t P'\underline{d})^{-1}\ \underline{d}^t\ P'\underline{A}^t \qquad (3.24)$$

$$P' = \lim_{N\to\infty} P' = Q + AP'\left[I-\underline{d}\ (r + \underline{d}^t P'\underline{d})^{-1}\ \underline{d}^t P'\right]A^t \qquad (3.25)$$

O procedimento acima, transformando o projeto do observador num projeto equivalente ao do regulador é possível devido à dualidade entre o observador e o regulador.

3.4. Controlador Ótimo Obtido a Partir do Regulador

Nos itens anteriores foi considerado o problema do regulador, ou seja, estando o sistema num estado inicial $\underline{x}(0)$, o objetivo é conduzí-lo ao estado $\underline{x}(N) \simeq 0$. No proje to de um servomecanismo deseja-se que a saída do sistema siga um determinado comando de referência. Este item trata da transformação do projeto do regulador no projeto de um controlador para um servomecanismo, baseado no apresentado em Franklin-Powell ((1980, p.155 e 156)).

As equações associadas ao sistema são

$$\underline{x}(k+1) = A\,\underline{x}(k) + \underline{b}\,u(K) \tag{3.1}$$

e

$$y(k) = \underline{d}^t\,\underline{x}(k) \tag{3.2}$$

Para o conjunto observador-regulador tem-se das equações (3.14) e (3.15).

$$\hat{\underline{x}}(k+1) = A\,\hat{\underline{x}}(k) + \underline{b}\,u(k) + \underline{h}\,[y(k) - \hat{y}(k)] \tag{3.26}$$

$$\hat{y}(k) = \underline{d}^t\,\hat{\underline{x}}(k) \tag{3.27}$$

e

$$u(k) = -\underline{K}\,\hat{\underline{x}}(k) \tag{3.9}$$

Substituindo as equações (3.27) e (3.9) em (3.26) tem se

$$\hat{\underline{x}}(k+1) = [A - \underline{b}\underline{K} - \underline{h}\underline{d}^t]\,\hat{\underline{x}}(k) + \underline{h}\,y(k) \tag{3.28}$$

e

$$u(k) = -\underline{K}\,\hat{\underline{x}}(k) \tag{3.9}$$

Considerando uma entrada de comando v(k), tem-se que a maneira mais geral de transformar o problema do regulador no problema do servomecanismo é introduzindo $\underline{v}(k)$ nas equações acima. Assim

$$\hat{\underline{x}}(k+1) = [A - \underline{b}\,\underline{K} - \underline{h}\underline{d}^t]\,\hat{\underline{x}}(k) + \underline{h}\;y(k) + \underline{m}\;v(k) \tag{3.29}$$

e

$$u(k) = -\underline{K}\,\hat{\underline{x}}(k) + n\;v(k) \tag{3.30}$$

onde $\underline{m}$ é um vetor de dimensão 3 e n um escalar. O problema então torna-se em determinar $\underline{m}$ e n.

Existem três alternativas para a determinação de $\underline{m}$ e n. A primeira é selecioná-los de maneira que a equação do observador independa da entrada v(k). A segunda é que v(k) sempre apareça na forma do erro e(k) = y(k) - v(k). A terceira é que sejam escolhidas de modo que as respostas transitória e em regime do sistema em malha fechada satisfaçam especificações desejadas.

No sistema considerado neste trabalho tem-se que a saída, a posição na amostragem k (igual à variável de estado $x_1(k)$), é obtida através da adição do valor da posição na amostragem anterior, k-1, ao deslocamento medido entre as amostragens k-1 e k, como visto no capítulo anterior.

Como existe limitação no comprimento de palavra dos registradores que armazenam o comando total e a saída do sistema, em algumas aplicações torna-se interessante armazenar apenas a diferença entre o comando e a saída do sistema, ou seja, o erro

$$e(k) = y(k) - v(k) \quad , \tag{3.31}$$

de maneira a minimizar as possibilidades de "overflow" das variáveis. Com isso, é deixado para o gerador de comandos de referência a determinação do valor absoluto da posição. Isto pode ser feito a partir da distribuição de sensores em locais pré-determinados (como, por exemplo, em um ponto de zeragem) e da informação dos comandos emitidos, subtraído do erro existente no controlador.

Assim, a opção escolhida para a determinação de $\underline{m}$ e n nas equações (3.30) e (3.31) é a segunda, ou seja, o comando v(k) deve sempre aparecer na forma do erro e(k). Para tal, da equação (3.29) chega-se a

$$\underline{m} = -\underline{h} \tag{3.32}$$

e da equação (3.31) a

$$n = 0 \tag{3.33}$$

As equações do controlador são

$$\hat{\underline{x}}(k+1) = [A - \underline{b}\underline{K} - \underline{h}\underline{t}]\ \hat{\underline{x}}(k) + \underline{h}\ [y(k)-v(k)] \tag{3.34}$$

e

$$u(k) = -\ \underline{K}\ \hat{\underline{x}}(k) \tag{3.35}$$

Com esta escolha, o comportamento dinâmico do sistema em malha fechada dependerá apenas dos valores determina dos para $\underline{k}$ e $\underline{h}$.

3.5. Diagonalização: Controlador I

A equação (3.34) pode ser implementada diretamente, ou sofrer transformações de maneira a reduzir o esforço computacional (em número de multiplicações). Este item e o próximo tratam destas transformações. Aqui a equação acima citada é agrupada de maneira a manter a característica de correção do erro entre a saída do sistema e a saída do observador, presente na formulação da teoria do observador, como visto no item 3.4. Tal estrutura é obtida a partir da diagonalização do conjunto controlador-observador, tendo como resultado uma estrutura semelhante à estrutura paralela apresentada em textos de filtragem e controle digitais. O resultado é denominado Controlador I. No próximo item, com algumas reduções chega-se à estrutura paralela, denominada Controlador II.

A equação (3.34) pode ser escrita como

$$\underline{\hat{x}}(k+1) = [A - \underline{b}\underline{K}]\ \underline{\hat{x}}(k) + \underline{h}\ [y(k) - v(k) - \underline{d}^t \hat{x}(k)] \qquad (3.36)$$

Para a transformação da equação acima numa estrutura com características de estruturas paralelas, define-se um novo conjunto de variáveis de estado

$$\underline{s}(k) = R^{-1}\ \underline{\hat{x}}(k) \qquad (3.37)$$

obtido através da transformação linear R^{-1}, onde R é uma matriz 3x3. A matriz R é escolhida tal que a equação do novo sistema

$$\underline{s}(k+1) = R^{-1} [A - \underline{b}\underline{K}] R\, \underline{s}(k) + R^{-1} \underline{h} [y(k) - v(k) - \underline{d}^t R\, \underline{s}(k)] \tag{3.38}$$

tenha a matriz $R^{-1} [A - \underline{b}\,\underline{K}] R$ na forma diagonal, ou seja , na forma de Jordan. A matriz R é a matriz dos autovetores de $[A - \underline{b}\,\underline{K}]$, como mostrado por KUO((1980, p.191 a 198)).

Na equação (3.25) a variável de controle, u(k), é obtida a partir das variáveis de estado $\hat{x}(k)$. Como existe um tempo para o cálculo de $\hat{\underline{x}}(k)$ e u(k), a equação (3.35) não é realista. Uma alternativa para solucionar este problema é considerar a variável de controle na amostragem k+1, ou

$$u(k+1) = - \underline{K}\, \hat{\underline{x}}(k+1) \tag{3.39}$$

Da equação (3.37) tem-se

$$u(k+1) = -\underline{K}\, R\, \underline{s}(k+1) \tag{3.40}$$

Substituindo a equação (3.38) na equação (3.40) tem - se

$$u(k+1) = -\underline{K}\, R \left\{ R^{-1} [A - \underline{b}\,\underline{K}] R\, \underline{s}(k) + R^{-1} \underline{h} [y(k) - v(k) - \underline{d}^t R\, \underline{s}(k)] \right\} \tag{3.41}$$

ou

$$u(k+1) = -\underline{k} [A - \underline{b}\underline{K}] R\, \underline{s}(k) - \underline{K}\underline{h} [y(k) - v(k) - \underline{d}^t R\, \underline{s}(k)] \tag{3.42}$$

Como visto, com este procedimento a variável de controle é determinada a partir das variáveis amostradas anteriormente. No entanto, isto leva a um atraso de um período de amostragem para a atuação da entrada de comando v(k). Para minimizar este problema, geralmente usa-se como entrada de comando o valor presente no instante mais próximo da amostragem k+1 (Moroney-Willsky and Houpt (1980)).

Chamando

$$A_s = R^{-1} [\, A - \underline{b}\,\underline{K} \,]\, R \tag{3.43}$$

$$\underline{h}_s = R^{-1}\, h \tag{3.44}$$

$$\underline{K}_s = -\underline{K}\, [\, A - \underline{b}\,\underline{K} \,]\, R \tag{3.45}$$

$$\underline{d}_s^t = d^t R \tag{3.46}$$

e

$$r = -\underline{K}\,\underline{h} \quad , \tag{3.47}$$

as equações (3.38) e (3.41) tornam-se

$$\underline{s}(k+1) = A_s\, \underline{s}(k) + \underline{h}_s\, [\, y(k) - v(k) - \underline{d}_s^t\, \underline{s}(k)] \tag{3.48}$$

e

$$u(k+1) = \underline{K}_s\, \underline{s}(k) + r\, [\, y(k) - v(k) - \underline{d}^t{}_s\, \underline{s}(k)] \, , \tag{3.49}$$

onde A_s é uma matriz diagonal, com elementos na diagonal iguais a λ_i, correspondentes aos autovalores.

A estrutura acima, mostrada no diagrama de estados da figura 3.2, pode ser modificada. Usando as propriedades dos sistemas lineares, os ganhos podem ser deslocados no diagra

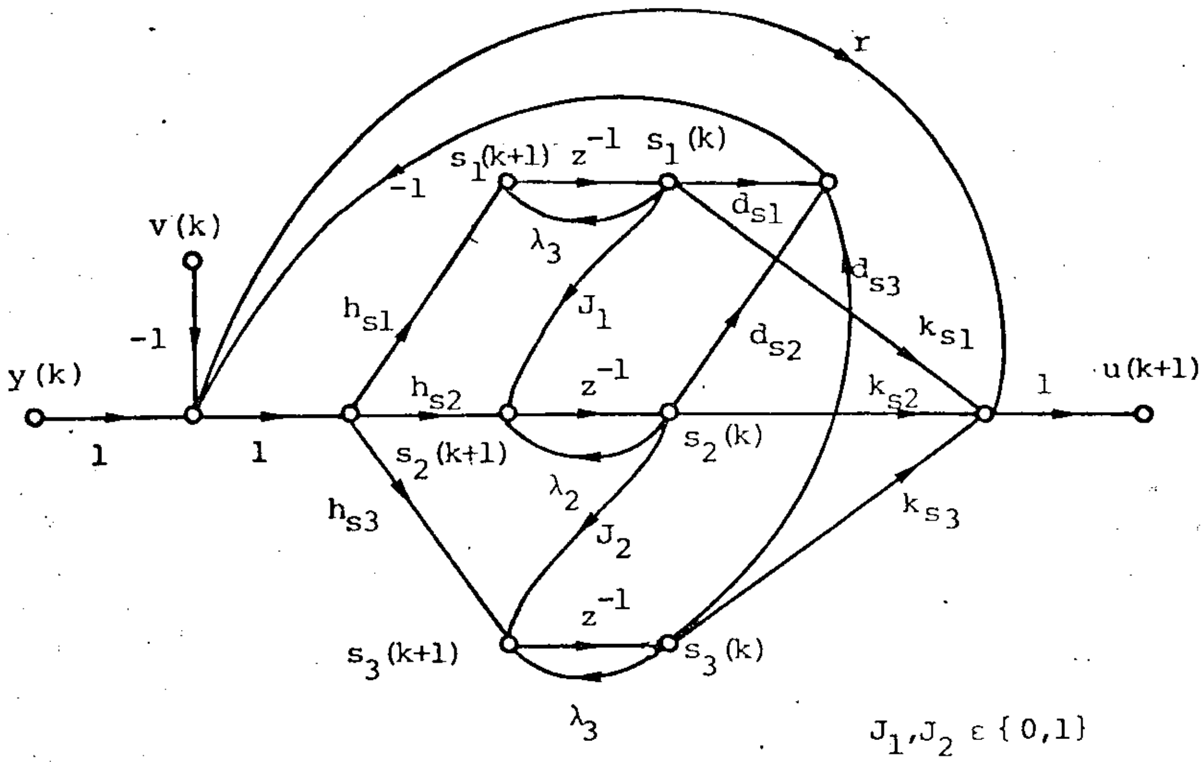
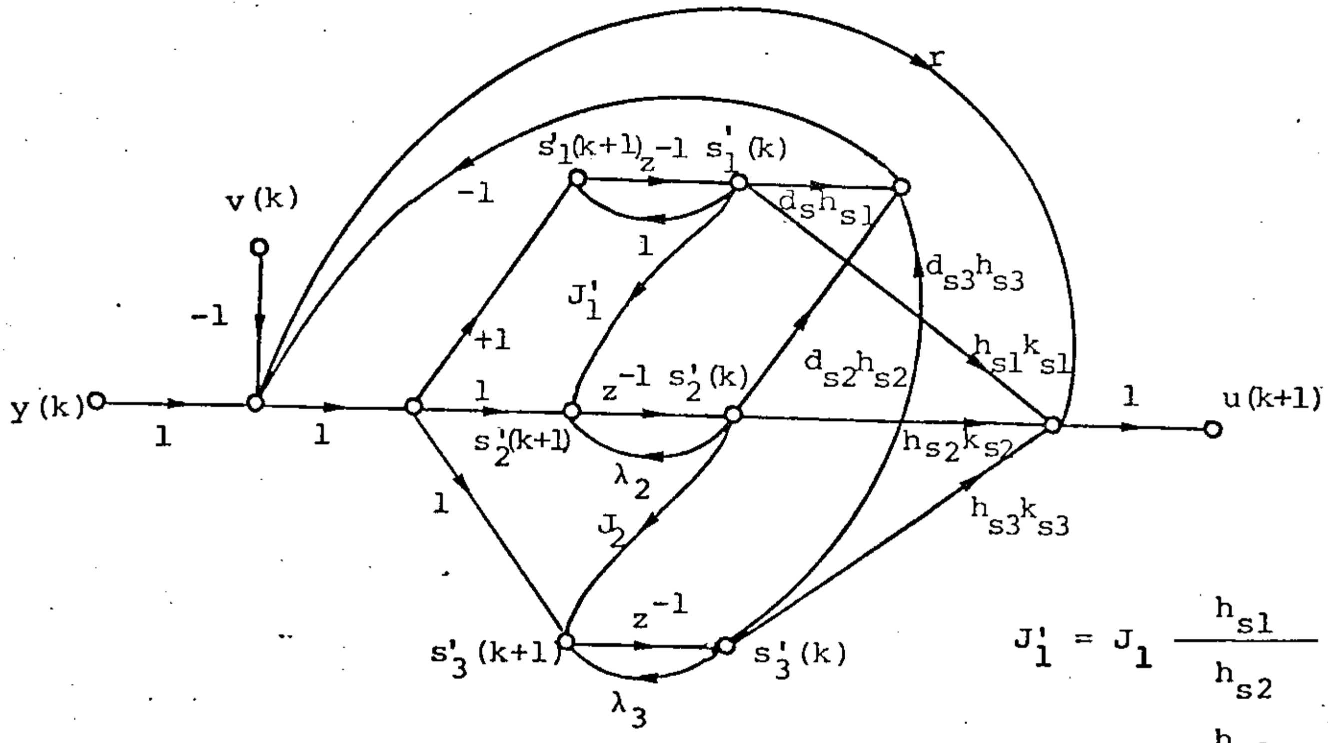


Figura 3.2 - Estrutura do controlador com características da estrutura paralela.



$$J'_1 = J_1 \frac{h_{s1}}{h_{s2}}$$

$$J'_2 = J_2 \frac{h_{s2}}{h_{s3}}$$

Figura 3.3 - Estrutura do controlador com menor número de multiplicações.

ma, de modo que a estrutura original transforma-se numa es trutura com um número menor de multiplicações, como mostrado na figura 3.3.

Finalmente, caso existam no máximo 2 (dois) autovalores idênticos, chega-se à estrutura mostrada no diagrama da figura 3.4, usando-se a equivalência entre os diagramas das figuras 3.5 e 3.6.

3.6 Reduções : Controlador II

A estrutura paralela é obtida através da transformação linear

$$\underline{s}(k) = R^{-1}\ \hat{\underline{x}}(k) \tag{3.37}$$

sobre as equações

$$\hat{\underline{x}}(k+1) = [A - \underline{b}\,\underline{K} - \underline{h}\underline{d}^t]\ \hat{\underline{x}}(k) + \underline{h}[y(k) - v(k)] \tag{3.34}$$

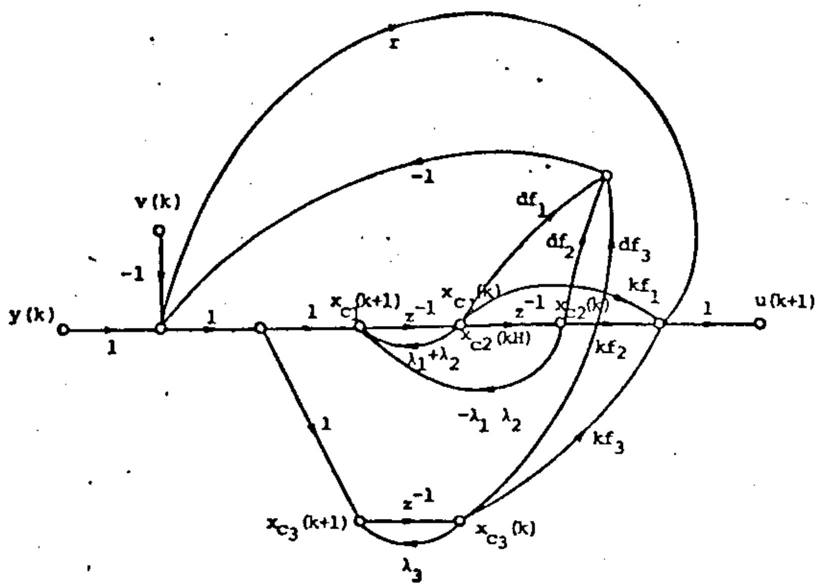
e

$$u(k) = \underline{K}\ \hat{\underline{x}}(k) \tag{3.35}$$

onde R é uma matriz de ordem 3x3 escolhida de maneira que a equação do novo sistema

$$\underline{s}(k+1) = R^{-1}\ [A - \underline{b}\,\underline{K} - \underline{h}\underline{d}^t]\ R\ s(k) + R^{-1}\underline{h}\ [y(k) - v(k)] \tag{3.50}$$

tenha a matriz $R^{-1}\ [A - \underline{b}\underline{K} - \underline{h}\underline{d}^t]\ R$ na forma diagonal. Como citado no item anterior, a matriz R é a matriz formada pelos



$df_1 = d_{s1}h_{s1} + d_{s2}h_{s2}$

$df_2 = d_{s2}h_{s1}J_1 - d_s h_s \lambda_2 - d_{s2}h_{s2}\lambda$

$df_3 = d_{s3}h_{s3}$

$kf_1 = k_{s1}h_{s1} + h_{s2}k_{s2}$

$k_{f2} = h_{s1}k_{s2}J_1 - k_{s1}h_{s1}\lambda_2 - k_{s2}h_s\lambda_1$

$kf_3 = h_{s3}k_{s3}$

Figura 3.4 - Estrutura do controlador I

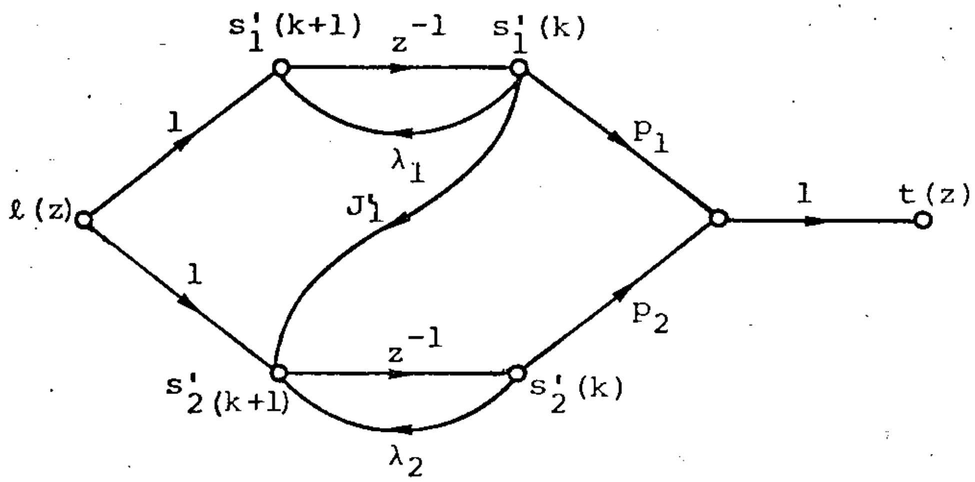


Figura 3.5 . Estrutura de segunda ordem.

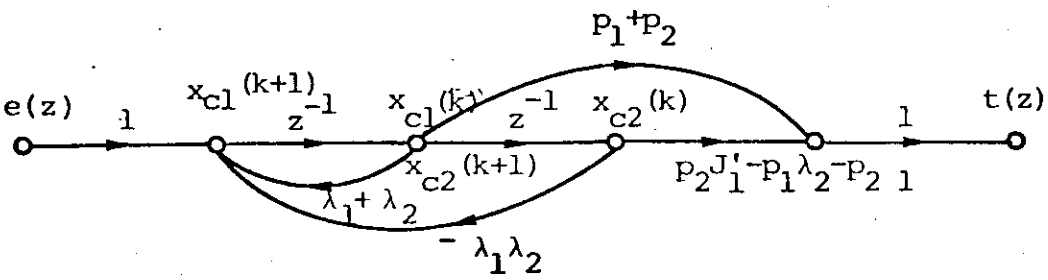


Figura 3.6 - Estrutura equivalente à da figura 3.5.

autovetores de $[A - \underline{b}\underline{K} - \underline{h}\underline{d}^t]$.

Da equação (3.35), tem-se

$$u(k+1) = -\underline{K}\,\underline{\hat{x}}(k+1) \quad (3.39)$$

e de (3.37)

$$u(k+1) = -\underline{K}\,R\,\underline{s}(k+1) \quad (3.40)$$

Substituindo a equação (3.50) na equação acima, tem-se

$$u(k+1) = -\underline{K}\,R\left\{R^{-1}\left[A - \underline{b}\underline{K} - \underline{h}\underline{d}^t\right] R\,\underline{s}(k) + R^{-1}\underline{h}\left[y(k) - v(k)\right]\right\} \quad (3.51)$$

$$u(k+1) = -\underline{K}\left[A - \underline{b}\underline{K} - \underline{h}\underline{d}^t\right] R\,\underline{s}(k) - \underline{K}\underline{h}\left[y(k) - v(k)\right] \quad (3.52)$$

Chamando

$$A_s = R^{-1}\left[A - \underline{b}\underline{K} - \underline{h}\underline{d}^t\right] R \quad (3.53)$$

$$\underline{h}_s = R^{-1}\,\underline{h} \quad (3.54)$$

$$\underline{K}_s = -\underline{K}\left[A - \underline{b}\underline{K} - hd^t\right] R \quad (3.55)$$

$$\underline{d}^t{}_s = \underline{d}^t\,R \quad (3.56)$$

e

$$r = -\underline{Kh}\ , \quad (3.57)$$

as equações (3.50) e (3.52) tornam-se

$$\underline{s}(k+1) = A_s\,\underline{s}(k) + \underline{h}_s\left[y(k) - v(k)\right] \quad (3.58)$$

e

$$u(k+1) = \underline{K}_s \, \underline{s}(k) + r\,[\,y(k) - v(k)] \qquad (3.59)$$

As equações acimas são mostradas no diagrama de estados da figura 3.7. Como no item anterior, usando as propriedades dos sistemas lineares para diminuir o número de multiplicações, chega-se à estrutura da figura 3.8.

Se existirem no máximo 2 (dois) autovalores idênticos, chega-se à estrutura paralela mostrada na figura 3.9, onde foi levada em consideração a equivalência entre as estruturas mostradas pelos diagramas das figuras 3.5 e 3.6.

3.7. Controlador Dinâmico: Controlador III

Seja o sistema dado pelas equações

$$\underline{x}(k+1) = A\,\underline{x}(k) + \underline{b}\,u(k) \qquad (3.60)$$

e

$$y(k) = \underline{d}^t\,\underline{x}(k) \qquad (3.61)$$

O controlador por realimentação de estados obtidos usando o índice de desempenho quadrático dado pela equação (3.9) é

$$u(k) = -\begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \qquad (3.62)$$

O conjunto de equações acima é mostrado na figura 3.10.

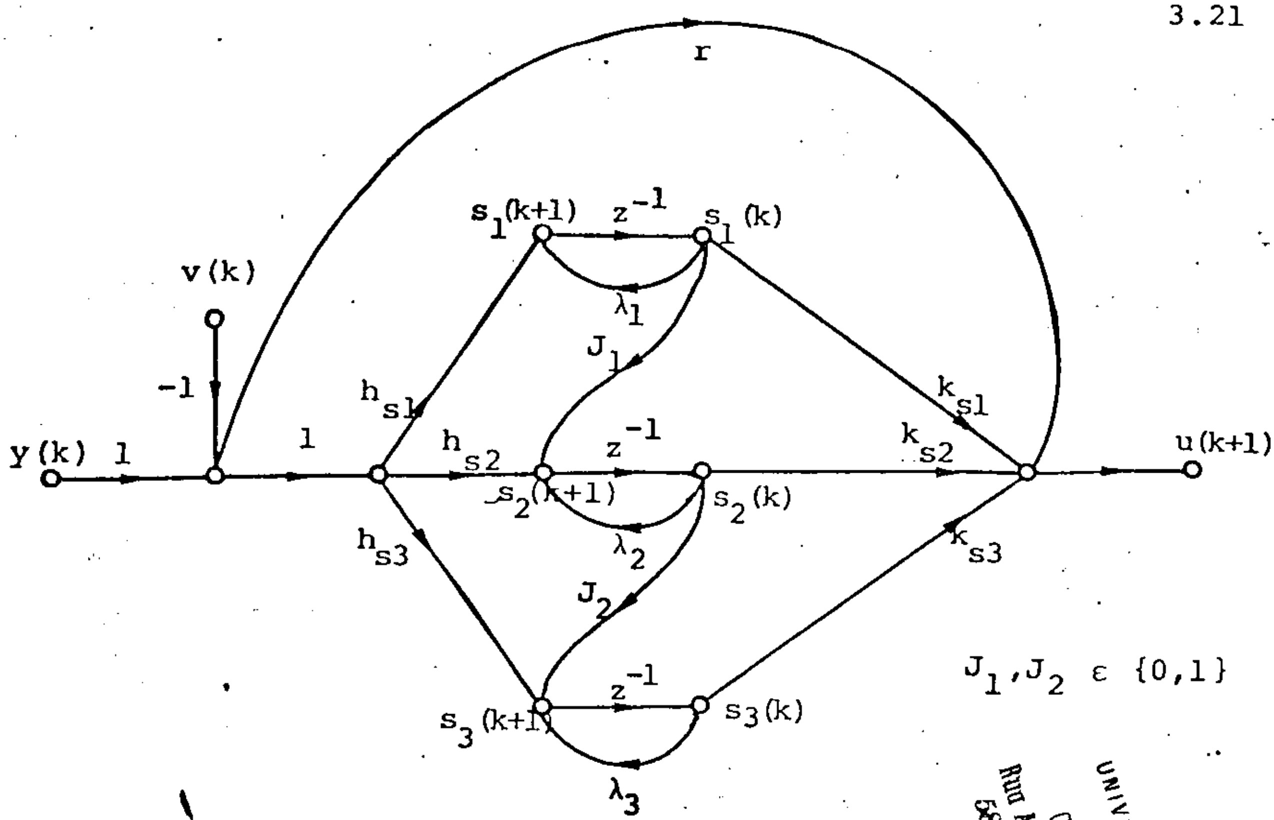


Figura 3.7 - Estrutura reduzida (paralela)

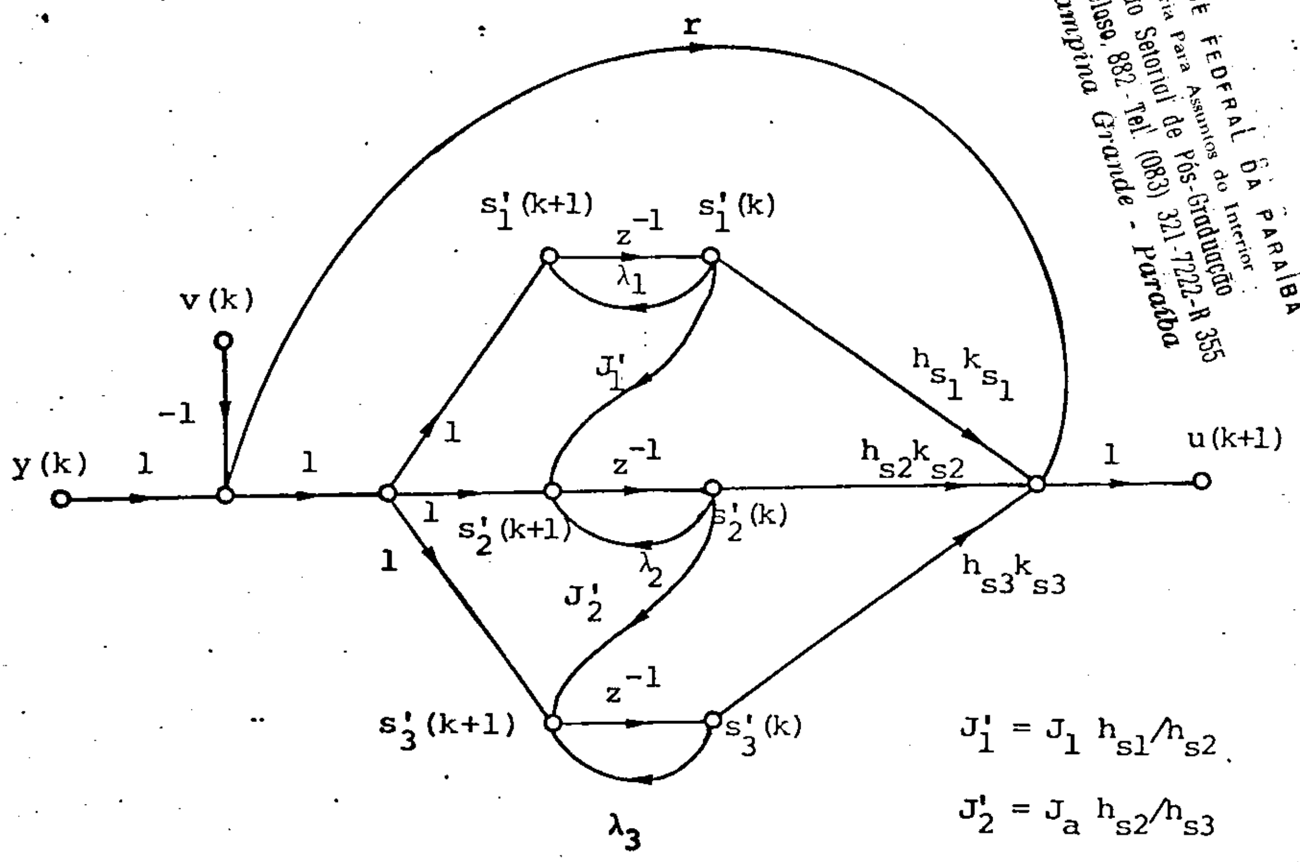
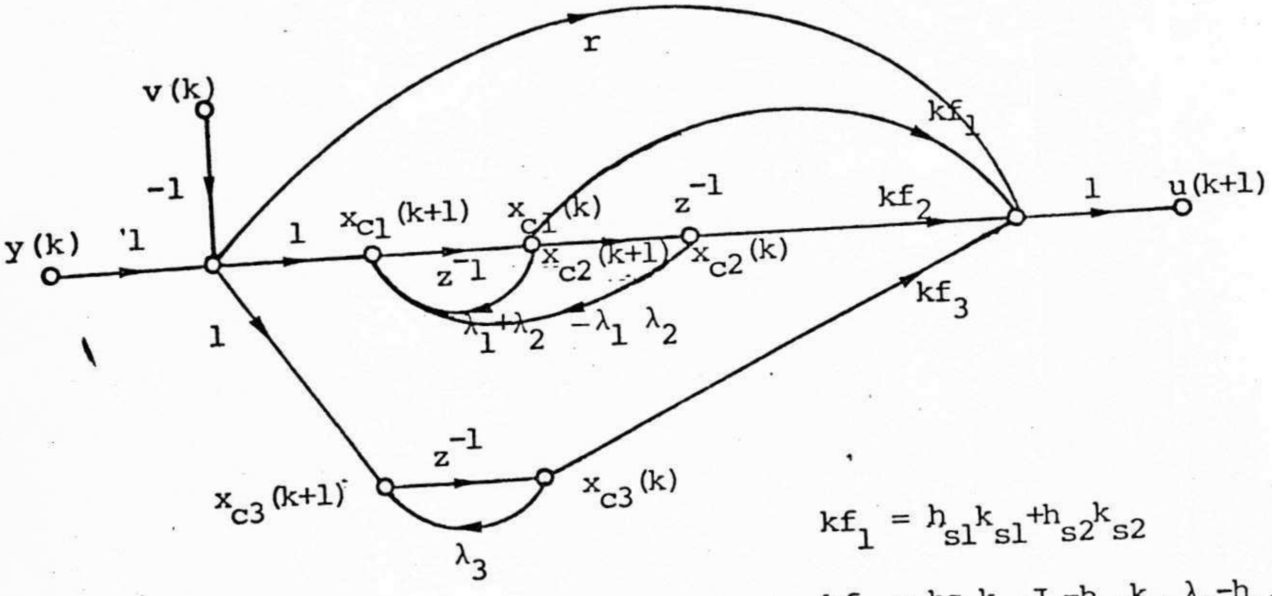


Figura 3.8 - Estrutura reduzida (paralela) com menor número de multiplicações.



$kf_1 = h_{s1}k_{s1}+h_{s2}k_{s2}$

$kf_2 = hs_1 k_{s2} J_1 - h_{s1}k_{s1}\lambda_2 - h_{s2}k_{s2}\lambda_1$

$kf_3 = h_{s3}k_{s3}$

Figura 3.9 - Estrutura do controlador II

Como os estados não são usualmente disponíveis em sua totalidade, o objetivo é aproximar o controlador com realimentação de estados da equação (3.62) por um controlador com entrada y(k). Isto é, um controlador com (KUO (1980, p. 555 a 566)) .

$$U(z) = -H(z)\ Y(z) \tag{3.63}$$

como mostrado na figura 3.11.

H(z) pode ser expressa por sua função de transferência:

$$H(z) = \frac{k_d\ (z^n + \alpha_1\ z^{n-1} + \ldots + \alpha_n)}{z^n + \beta_1\ z^{n-1} + \ldots + \beta_n} \tag{3.64}$$

Expandindo H(z) em série de Laurent em torno de z=0, tem-se

$$H(z) = k_d \sum_{k=0}^{\infty} h_k\ z^{-k} \tag{3.65}$$

onde

$$h_0 = 1$$

$$h_1 = \alpha_1 - \beta_1$$

$$h_k = \alpha_k - \beta_k - \sum_{\ell=1}^{k-1} h_k\ \beta_{k-\ell} \tag{3.66}$$

Assumindo que a série infinita converge, e truncando-a em m termos, tem-se a nova série

$$H_m(z) = K_d \sum_{k=0}^{m-1} h_k\ z^{-k} \tag{3.67}$$

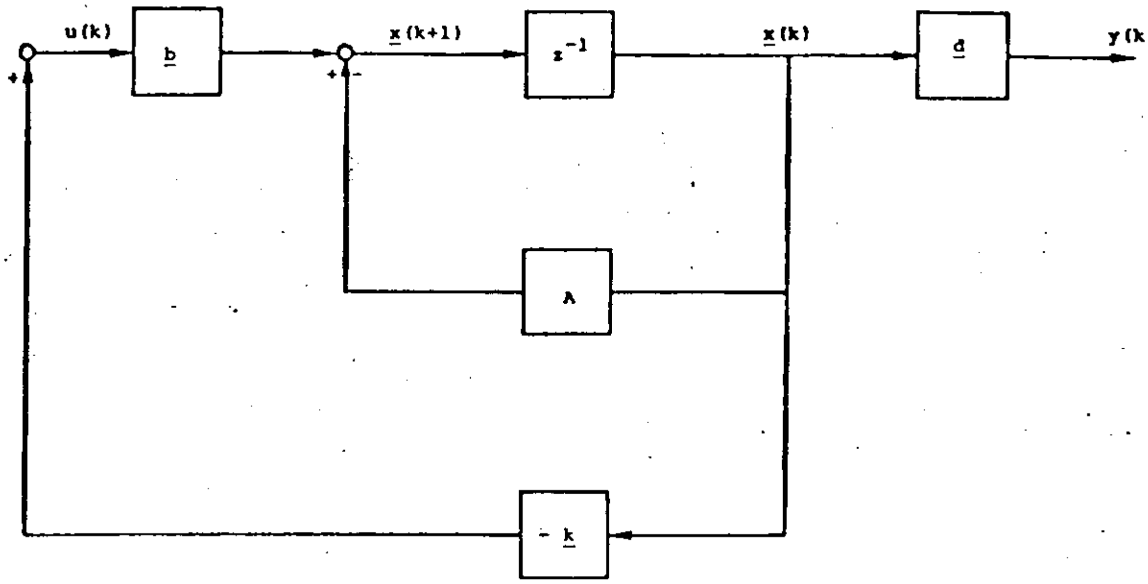


Figura 3.10 - Sistema de controle digital com realimentação de estado.

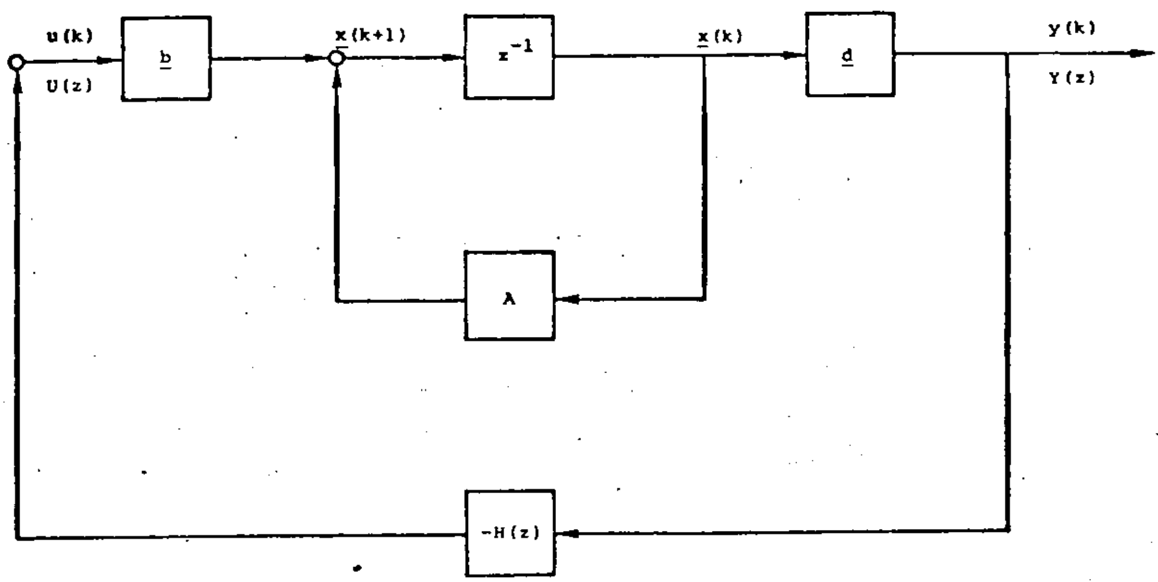


Figura 3.11 - Sistema de controle digital com realimentação dinâmica.

Substituindo-se a equação acima na equação (3.63), tem-se

$$U(z) \approx - \begin{bmatrix} k_d' h_0 & k_d h_1 \dots K_d h_{m-1} \end{bmatrix} \begin{bmatrix} Y(z) \\ z^{-1} Y(z) \\ \vdots \\ z^{-m+1} Y(z) \end{bmatrix} \tag{3.68}$$

Realizando a transformada inversa

$$u(k) \approx - F \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-m+1) \end{bmatrix} \tag{3.69}$$

Da equação (3.61) obtem-se

$$y(k-1) = \underline{d}^t \, \underline{x}\,(k-1) \tag{3.70}$$

e da equação (3.60)

$$A\,\underline{x}(k-1) = \underline{x}(k) - \underline{b}\,u(k-1) \tag{3.71}$$

Usando a equação (3.62)

$$A\,\underline{x}(k-1) = \underline{x}(k) + \underline{b}\,\underline{K}\;\underline{x}(k-1) \tag{3.72}$$

a qual pode ser reagrupada como

$$\underline{x}(k-1) = (A - \underline{b}\,\underline{K})^{-1} \cdot \underline{x}(k) \tag{3.73}$$

assumindo que a inversa da matriz existe.

Substituindo a equação (3.73) na equação (3.70) tem-se

$$y(k-1) = \underline{d}^t (A - \underline{b}\,\underline{K})^{-1} \,.\, \underline{x}(k) \tag{3.74}$$

Analogamente chega-se a

$$y(k-\ell) = d^t (A - \underline{b}\,\underline{K})^{-\ell}\, \underline{x}(k) \tag{3.75}$$

Assim, a equação (3.69) torna

$$u(k) \simeq -F \begin{bmatrix} \underline{d}^t \\ \underline{d}^t A' \\ \underline{d}^t (A')^2 \\ \vdots \\ \underline{d}^t (A')^{m-1} \end{bmatrix} \underline{x}(k) \tag{3.76}$$

onde

$$A' = (A - \underline{b}\,\underline{K})^{-1}$$

Comparando a equação acima com a equação (3.62) tem-se

$$-F \begin{bmatrix} \underline{d}^t \\ \underline{d}^t A' \\ \vdots \\ \underline{d}^t (A')^{m-1} \end{bmatrix} \approx -\begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \tag{3.77}$$

Para que o sistema de equações (3.77) tenha solução única, deve-se ter m=3.

O controlador dinâmico é então obtido a partir de

$$F \begin{bmatrix} 1 & 0 & 0 \\ [100] & A' & \\ [100] & (A')^2 & \end{bmatrix} = [\, k_1 \quad k_2 \quad k_3] \tag{3.78}$$

Da equação (3.68) temos que

$$F = \begin{bmatrix} k_d\, h_0 & k_d h_1 & k_d h_2 \end{bmatrix} \quad , \tag{3.79}$$

e da equação (3.66)

$$\begin{aligned} h_0 &= 1 \\ h_1 &= \alpha_1 - \beta_1 \\ h_2 &= \alpha_2 - \beta_2 - h_1\, \beta_1 \end{aligned} \tag{3.80}$$

Na equação acima, β_1 e β_2 são parâmetros a serem selecionados pelo projetista e devem ser pequenos de modo a manter os polos do controlador com efeitos despresíveis em relação aos zeros, (KUO (1980, p. 561)), já que eles não são obtidos diretamente da expansão em série dada pela equação (3.65) utilizada em todo o desenvolvimento do controlador dinâmico.

O controlador dinâmico torna-se

$$H(z) \simeq k_d \frac{(1 + \alpha_1\, z^{-1} + \alpha_2\, z^{-2})}{(1 + \beta_1\, z^{-1} + \beta_2\, z^{-2})} \tag{3.81}$$

A implementação direta do controlador acima é mostrada na figura 3.12.

Na amostragem k, para determinar a saída após o cálculo de e(k), são necessárias duas adições e uma multiplicação. Fazendo

$$H(z) = k_d + \frac{\delta_1 \; z^{-1} + \delta_2 z^{-2}}{1 + \beta_1 \; z^{-1} + \beta_2 z^{-2}} \tag{3.82}$$

onde

$$\delta_1 = k_d \; \alpha_1 - \beta_1$$

$$\delta_2 = k_d \; \alpha_2 - \beta_2 \tag{3.83}$$

uma adição é eliminada. Estrutura do controlador III é mostrada na figura 3.13.

3.8. Resumo

Neste capítulo foram apresentados os projetos de três controladores, utilizando de transformações lineares em suas estruturas, visando a diminuição no número de operações aritméticas necessárias nas implementações. Os controladores são projetados baseando-se na minimização de um índice de desempenho quadrático. Dois deles utilizam observadores de estados e são diferenciados pela estrutura obtida após as transformações lineares, sendo que no primeiro procurou-se manter a realimentação existente na formulação da teoria dos observadores. O terceiro controlador é uma aproximação dinâmica do regulador ótimo. Foi considerada também a extensão do projeto do regulador para o projeto de um servomecanismo.

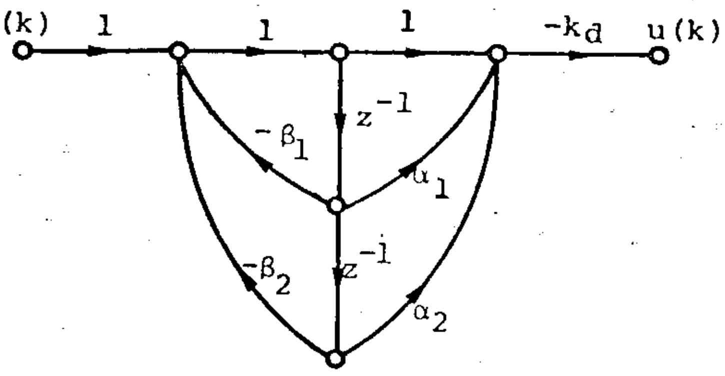


Figura 3.12 - Estrutura direta de 2ª ordem.

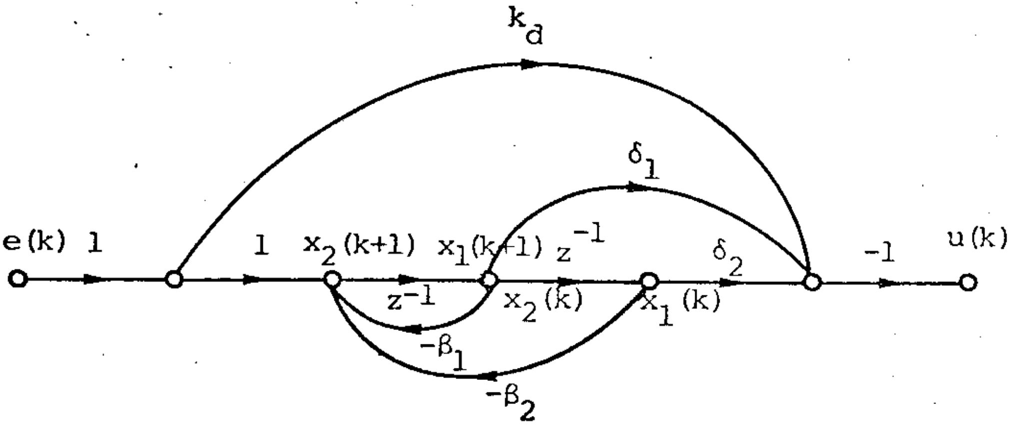


Figura 3.13 - Estrutura do controlador III.

4. DETERMINAÇÃO DO COMPRIMENTO DE PALAVRA DE VARIÁVEIS E COEFICIENTES NA REPRESENTAÇÃO EM PONTO FIXO

4.1. Introdução

Os controladores projetados nos capítulos anteriores assumem a representação de variáveis e coeficientes com precisão infinita.

Na implementação destes controladores em computadores é necessário que as variáveis e coeficientes sejam armazenados em registradores com comprimento finito. Consequentemente os controladores assim implementados (controladores digitais) não corresponderão exatamente aos controladores com precisão infinita (ideais).

O objetivo deste capítulo é apresentar métodos estatísticos de análise aplicáveis à determinação do comprimento de palavra das variáveis e coeficientes, de modo que os controladores implementados se aproximem dos controladores ideais dentro de limites considerados aceitáveis para fins práticos.

Os métodos apresentados são baseados em pesquisas de senvolvidas na área de Processamento de Sinais Digitais, relacionadas com a análise e implementação de filtros digitais (Oppenheim-Schafer (1975) Rabiner-Gold(1975) ,Crochiere-Oppenheim (1975), Crochiere (1975) , Oppenheim-Weinstein (1972)). Particularmente, a determinação do comprimento de palavra dos coeficientes é uma extensão da técnica apresentada para filtros digitais por Crochiere (1975) .

Introduzem-se a seguir: definições de diagrama de fluxo de sinais e da representação matricial para sistemas digitais, a resposta de sistemas lineares a sinais estocásticos, a análise estatística na determinação do comprimento de palavra de variáveis, a análise do ruído na saída devido à conversão D/A, o teorema de Tellegen para malhas digitais, a interreciprocidade entre malhas e o teorema da transposição, pois os conceitos envolvidos são essenciais para o desenvolvimento deste trabalho. Continuando, chega-se a uma fórmula para a sensibilidade de malhas. A seguir, é apresentada uma análise estatística para a determinação do comprimento de palavra dos coeficientes. Finalmente, são feitas considerações sobre a implementação numérica dos métodos de análise incluindo métodos para determinação da resposta em frequência usando a representação matricial.

4.2. Diagrama de Fluxo de Sinais e Representação Matricial para Sistemas Digitais

Uma das possíveis formas de representação de sistemas digitais é o diagrama de fluxo de sinais. Neste item é introduzida uma versão particularizada da apresentada por Crochière-Oppenheim (1975) , por ser mais adequado aos propósitos deste trabalho.

Um diagrama de fluxo de sinais é uma malha com ramos direcionados que se conectam em nós. Associado a cada nó existe uma variável (ou valor)do nó . A variável associada ao nó k é denominada ω_k. O ramo (Jk) é o ramo originado no nó J e terminado no nó k, com a direção de J a k sendo indicada pela seta existente no ramo (Fig. 4.1.a). Cada ramo tem associada uma variável de entrada e uma variável de saída. A variável de entrada associada com o ramo (Jk) é representada por ω_J e a variável de saída por v_{Jk}. A dependência da variável de saída do ramo em relação à entrada correspondente é dada por

$$v_{Jk} = f_{Jk}(\omega_k) \quad , \tag{4.1}$$

ou seja, $f_{Jk}(.)$ é uma transformação que o ramo exerce sobre a variável de entrada dando como resultado o sinal de saída.

Os sinais externos ao sistema digital são representados no diagrama de fluxo de sinais por nós denominados nós fonte. O valor do nó fonte J é representado por x_J, e a variável de saída de um ramo com origem no nó fonte J e dirigida ao nó k é representado por s_{Jk} (Fig. 4.1.b). A saída de sinais do sistema digital ocorre através de nós denominados

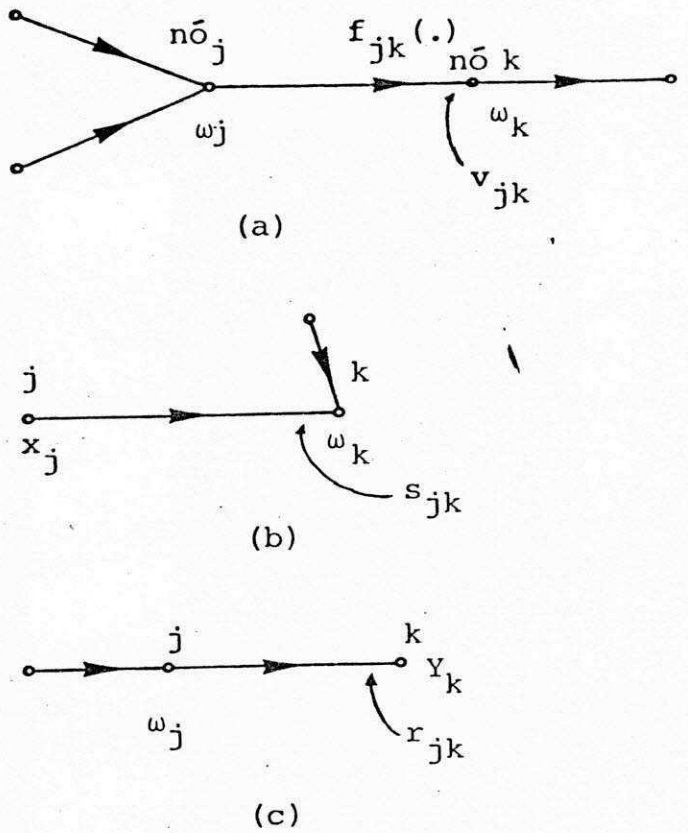


Figura 4.1 - Diagrama de fluxo de sinais.

nós sorvedouro . O valor do nó sorvedouro k é representado por y_k e a variável de saída do ramo conectando o nó J ao nó sorvedouro k é representada por r_{Jk} (Figura 4.1.c).

Por definição, o valor de cada nó da malha é dado pe a soma das variáveis de saída de todos os ramos que chegam àquele nó. De maneira geral podemos escrever

$$\omega_k = \sum_{J=1}^{N} v_{Jk} + \sum_{J=1}^{M} s_{Jk} \qquad k = 1,2,\ldots,N \qquad (4.2)$$

$$y_k = \sum_{J=1}^{N} r_{Jk} \qquad ; \quad k = 1,2,\ldots P \qquad (4.3)$$

onde

N é o número de nós da malha
M é o número de nós fonte, e
P é o número de nós sorvedouro.

Se o sistema digital é linear, então $f_{Jk}(.)$ a transformação associada ao ramo JK é caracterizada por uma função de transferência, denominada transmitância, de modo que

$$V_{Jk}(z) = F_{Jk}(z)\ W(z) \qquad (4.4)$$

Em sistemas digitais lineares o diagrama de fluxo de sinais pode ser definido de modo que as funções de transferência dos ramos sejam os elementos básicos multiplicação por uma constante ou atrasos (Crochiere-Oppenheim (1975) e Oppe-

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprigio Veloso, 882 - Tel. (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

nheim-Schaffer (1975 p. 136-148)). Assim

$$F_{Jk}(z) = F_{cJk} + F_{dJk}\, z^{-1} \tag{4.5}$$

onde

$$F_{dJk} = 1 \text{ ou } F_{dJk} = 0$$

Assumindo que para cada nó da malha existe apenas um nó fonte correspondente e que a cada nó sorvedouro corresponde um nó da malha e, ainda mais, que as funções de transferência associadas aos ramos respectivos são ganhos unitáios, então as equações (4.2) e (4.3) têm transformadas

$$W_k(z) = \sum_{J=1}^{N} (F_{cJk} + F_{dJk}\, z^{-1})\, W_J(z) + X_k(z), \tag{4.6}$$

$k = 1,2, \ldots, N$ e,

$$Y_k(z) = W_k(z). \tag{4.7}$$

Substituindo a equação (4.7) na equação (4.6) tem-se

$$Y_k(z) = \sum_{J=1}^{N} (F_{cJk} + F_{dJk}\, z^{-1})\, Y_J(z) + X_k(z) \tag{4.8}$$

para $k = 1,2,\ldots, N$.

O conjunto de N equações acima pode ser colocada na forma matricial,

$$\underline{Y}(z) = (F_c^t + F_d^t\, z^{-1})\, \underline{Y}(z) + \underline{X}(z) \tag{4.9}$$

onde

$\underline{Y}(z)$ é o vetor coluna (Nx1) com elementos $Y_k(z)$

$\underline{X}(z)$ é o vetor coluna (Nx1) com elementos $X_k(z)$

F_c^t é a matriz transposta (NxN) de

$$F_c = \{F_{ckJ}\}$$

F_d^t é a matriz transposta (NxN) de

$$F_d = \{F_{dkJ}\} \quad \text{com } F_{dkJ} \in \{0,1\} .$$

4.3. RESPOSTA DE SISTEMAS LINEARES A SINAIS ESTOCÁSTICOS

Seja h(n) a sequência resposta ao impulso unitário de um sistema digital linear invariante no tempo estável. (Este item segue o apresentado por Oppenheim-Schaffer (1975, p. 391-395)). Seja x(n) a sequência de entrada, real, a qual é uma amostra de um processo estocástico, discreto no tempo estacionário no sentido amplo. A resposta do sistema linear à sequência x(n) é dada por

$$y(n) = \sum_{k=-\infty}^{\infty} h(n-k)\ x(k) = \sum_{k=-\infty}^{\infty} h(k)\ x(n-k) \qquad (4.10)$$

a qual é uma amostra de um processo estocástico (de saída).

Se m_x é a média de x(n), então.

$$m_x = E\ \{x(n)\} = \int_{\infty}^{\infty} x\ P_{x_n}\ (x,n)\ dx \qquad (4.11)$$

e consequentemente

$$m_y = E\ \{y(n)\} = \sum_{k=-\infty}^{\infty} h(k)\ E\ \{x(n-k)\}$$

$$= m_x \sum_{k=-\infty}^{\infty} h(k) \qquad (4.12)$$

Se o sistema é estável e o processo de entrada é estacionário, a média do sinal de saída é constante.

Sendo $\phi_{xx}(n,m)$ a sequência autocorrelação, definida por

$$\phi_{xx}(n,m) = E\{x_n \, x_m^*\}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_n \, x_n^* \, P_{x_n, x_m}(x_n, n, x_m, m) \, d_{x_n} d_{x_m} \qquad (4.13)$$

(Nota: * denota complexo conjugado) ,

então

$$\phi_{yy}(n,n+m) = E\{y(n)\; y(n+m)\} \qquad (4.14)$$

$$= E\left\{\sum_{k=-\infty}^{\infty} \sum_{r=-\infty}^{\infty} h(k)\; h(r)\; x(n-k)\; x(n+m-r)\right\} \qquad (4.15)$$

$$= \sum_{k=-\infty}^{\infty} h(k) \sum_{r=-\infty}^{\infty} h(r)\; E\{x(n-k)\,x(n+m-r)\} \qquad (4.16)$$

Assumindo que x(n) é estacionário, o termo

$$E\{x(n-k)\; x(n+m-r)\}$$

depende somente da diferença m+k-r. Então

$$\phi_{yy}(n,n+m) = \phi_{yy}(m) = \sum_{k=-\infty}^{\infty} h(k) \sum_{r=-\infty}^{\infty} h(r)\; \phi_{xx}(m+k-r) \qquad (4.17)$$

Conclui-se da equação acima que, como a autocorrelação da saída depende somente da diferença no tempo m, então um sistema digital linear invariante no tempo e estável, excitado por um sinal estacionário, tem uma saída também estacionária.

Substituindo $\ell = r-k$ tem-se

$$\phi_{yy}(m) = \sum_{\ell=-\infty}^{\infty} \phi_{xx}(m-\ell) \sum_{k=-\infty}^{\infty} h(k)h(\ell+k) \qquad (4.18)$$

Se a média $m_x = 0$, então a transformada z da equação acima existe e é dada por

$$\Phi_{yy}(z) = H(z)\, H(z^{-1})\, \Phi_{xx}(z) \qquad (4.19)$$

Em termos do espectro de densidade de potência,

$$P_{yy}(\omega) = |H(e^{J\omega})|^2 P_{xx}(\omega) \qquad (4.20)$$

para $0 \leqslant \omega^{<} \leqslant 2\pi$.

Se o sinal de entrada é um ruído branco, então

$$\phi_{xx}(m) = \sigma_x^2\, \delta(m) \qquad (4.21)$$

onde σ_x^2 é a variância de x(n) e

$$P_{xx}(\omega) = \sigma_x^2 \qquad (4.22)$$

para $0 \leqslant \omega < 2\pi$. Assim, da equação (4.20)

$$P_{yy}(\omega) = \sigma_x^2\, |H(e^{J\omega})|^2 \qquad (4.23)$$

A variância de y(n) é dada por

$$\sigma_y^2 = \frac{1}{2\pi} \int_0^{2\pi} P_{yy}(\omega)\, d\omega \tag{4.24}$$

ou

$$\sigma_y^2 = \frac{\sigma_x^2}{2\pi} \int_0^{2\pi} |H(e^{J\omega})|^2\, d\omega \ . \tag{4.25}$$

4.4. Determinação do Comprimento de Palavra de Variáveis

O objetivo da análise do erro introduzido pela quantização de variáveis é o da escolha do comprimento dos registradores, onde estas variáveis serão armazenadas, de maneira a ter-se uma degradação aceitável no comportamento do sistema. A análise apresentada neste item basicamente segue aquela apresentada em Oppenheim-Schaffer (1975 p. 423-432) e Rabiner-Gold (1975, p. 309-315)) para filtros digitais.

Como o comprimento dos registradores só pode ser mudado em passos de um bit, uma análise precisa dos erros de arredondamento não é usualmente necessária em aplicações práticas. Como será visto na equação (4.48), a adição de um bit reduz a amplitude dos erros de quantização por um fator de um meio. Este fato possibilita o uso de modelos estatísticos na análise dos erros de quantização. No caso deste trabalho, a análise será limitada ao efeito de arredondamento, assumindo-se a representação de números em complemento de dois, pon to fixo.

Na formulação do modelo de análise considere inicialmente um sistema de primeira ordem mostrado na fig. 4.2.a , assumindo uma implementação com precisão infinita. Na figura 4.2.b é mostrado o mesmo sistema, assumindo uma implementação com precisão finita, com a variação na saída devido ao arredondamento dado por f(n).Q(.) denota a operação de arredondamento. Observe que, no caso de ponto fixo, só existe necessidade de arredondamento após a multiplicação por uma cosntante, já que na adição só pode ocorrer "overflow". Na figura 4.2.c a operação de arredondamento é substituída por uma fonte de ruído aditiva e(n) onde

$$e(n) = Q\,[\alpha\omega(n-1)] - \alpha\omega(n-1) \qquad (4.26)$$

Assumindo que:

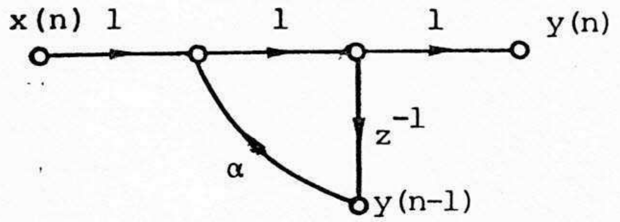
1) A sequência de erro e(n) é um ruído branco,

2) A sequência de erro tem uma distribuição uniforme sobre o intervalo de quantização, e

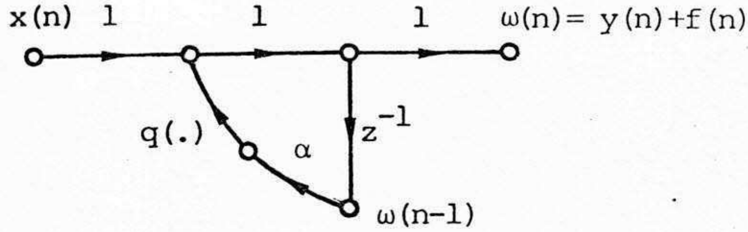
3) A sequência de erro é não correlata com a entrada x(n) e a variável $\alpha\omega(n-1)$, então, supondo o comprimento do registrador ser de b bits, tem-se que se o bit menos significativo tem valor 2^{-b}, então o erro estará na faixa.

$$-\frac{1}{2}\,2^{-b} < e(n) \leq \frac{1}{2}\,2^{-b} \qquad (4.27)$$

(a)



(b)



(c)

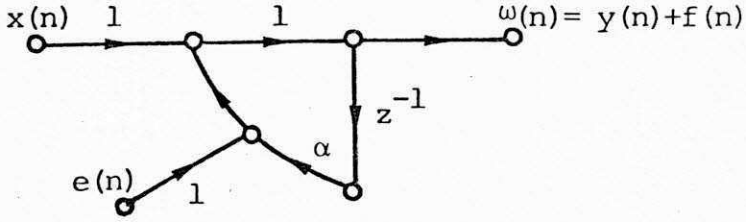


Figura 4.2 - Modelagem estatística do arredondamento.

Como a distribuição é uniforme,

$$m_e = 0$$

$$\sigma_e^2 = \frac{1}{12} \cdot 2^{-2b} \qquad (4.28)$$

Sendo y(n) a saída para a implementação com precisão infinita, a saída real pode ser representada por

$$\omega(n) = y(n) + f(n) \qquad \mathbf{e,} \qquad (4.29)$$

de acordo com as equações (4.12) e (4.25)

$$m_f = m_e \sum_{n=-\infty}^{\infty} h_e(n) = 0 \qquad (4.30)$$

e

$$\sigma_f^2 = \sigma_e^2 \sum_{h=-\infty}^{\infty} h_e^2(n) = \frac{\sigma_e^2}{2\pi} \int_0^{2\pi} |H_e(e^{J\omega})|^2 \, d\omega \qquad (4.31)$$

onde $H_e(z)$ é a função de transferência entre o nó onde a fonte de erro é introduzida, e o nó de saída da malha.

De uma maneira geral tem-se que os sistemas digitais lineares podem ser representados a partir de três elementos básicos: somadores, multiplicação por constantes e atrasos, como exemplificado pela figura 4.3.a, para um sistema de segunda ordem. Dois tipos de nós podem ser identificados: nós de soma, que correspondem a pontos de soma das variáveis de ramos e que têm uma ou mais entradas e apenas uma saída; e, nós de ramos, que correspondem a pontos de intercorrecção , com uma entrada e uma ou mais saídas.

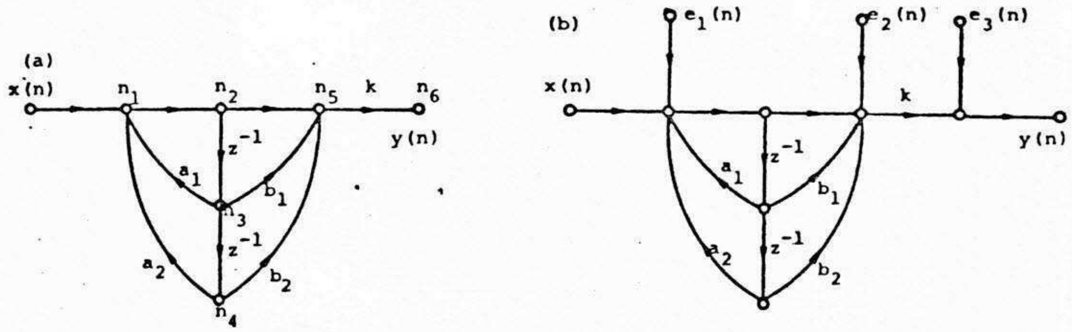


Figura 4.3 - Introdução de fontes de erro.

Conforme dito anteriormente, os erros de quantiza - ção introduzidos no sistema, para o caso de implementação em ponto fixo, aparecem apenas nos nós de soma que se seguem a multiplicadores por constantes não inteiras, como exemplificado na fig. 4.3.b.

Assumindo existirem k_i entradas de fontes de erro no nó de soma i, tem-se que

$$m_{e_i} = 0$$

e,

$$\sigma^2_{e_i} = k_i \ \sigma^2_e = k_i \ \frac{2^{-2b}}{12} \tag{4.32}$$

O ruído de arredondamento total que aparece na saída tem um espectro de densidade de potência (Jackson (1970, p. 166))

$$N_f(\omega) = \sum_i P_{ff_i}(\omega) = \sigma^2_e \sum_i k_i |H_i(e^{J\omega})|^2 \tag{4.33}$$

onde

$P_{ff_i}(\omega)$ é o espectro de densidade de potência na saída correspondente ao erro introduzido no nó i,

$\sigma^2_e = \frac{1}{12} 2^{-2b}$ é a variância do sinal de erro devido ao arredondamento para b bits,

k_i é o número de fontes de erro entrando no nó i, e

$H_i(e^{J\omega})$ é a função de transferência do nó i à saída, avaliada em $z = e^{J\omega}$ com $0 \leqslant \omega \leqslant 2\pi$.

Das equações (4.31) e (4.33) tem-se

$$\sigma_f^2 = \frac{\sigma_e^2}{2\pi} \int_0^{2\pi} \sum_i k_i \left| H_i(e^{J\omega}) \right|^2 d\omega \ . \qquad (4.34)$$

Se $\sum_i k_i$ é grande o suficiente para podermos considerar f(n) um processo com distribuição gaussiana, então pode-se escolher um valor $x\,\sigma_f$ onde x é um fator de multiplicação do desvio padrão, tal que

$$P\ \{ |f(n)| \leqslant x\,\sigma_f \} = p \qquad (4.35)$$

Se k_v é o limite superior aceitável para $|f(n)|$ com probabilidade p, ou seja,

$$k_v = x\,\sigma_f \quad , \qquad (4.36)$$

então o número de bits necessário será calculado a partir da substituição da equação (4.34) na equação acima,

$$k_v = x.\ \frac{\sigma_e}{\sqrt{2\pi}} \left[\int_0^{2\pi} \sum_i k_i \left| H_i(e^{J\omega}) \right|^2 d\omega \right]^{1/2} \qquad (4.37)$$

isto é,

$$\sigma_e = \frac{2^{-b}}{\sqrt{12}} = \frac{k_v \sqrt{2\pi}}{x.\left[\int_0^{2\pi} \sum_i k_i \left| H_i(e^{J\omega}) \right|^2 d\omega \right]^{1/2}} \qquad (4.38)$$

ou, finalmente,

$$b = \log_2 \frac{x.\left[\int_0^{2\pi} \sum_i k_i \left| H_i(e^{J\omega}) \right|^2 d\omega \right]^{1/2}}{k_v.\ \sqrt{24\pi}} \qquad (4.39)$$

onde não foi incluido o bit de sinal.

4.5. Ruído Devido à Conversão D/A

A técnica do item anterior, apresentada para a determinação do comprimento de palavra de variáveis, pode ser usada na análise do efeito de quantização devido à conversão D/A.

Caso seja desejado determinar o comprimento de palavra para o conversor D/A, utiliza-se a equação (4.39), entre o conversor D/A e a saída do sistema. No caso de desejar-se calcular o ruído que aparece na saída devido à escolha a priori de um dado conversor D/A, utiliza-se a equação (4.34).

4.6. Teorema de Tellegen, Interreciprocidade e Teorema da Transposição para Malhas Digitais

Na determinação da sensibilidade de uma malha à variação de um coeficiente, um conjunto de teoremas básicos é utilizado. Este item trata destes teoremas, e segue o apresentado em Crochiere-Oppenheim (1975) e Oppenheim-Schaffer (1975 p.171-181).

Sejam duas malhas digitais quaisquer com a mesma topologia, com N nós, com as variáveis de nó dadas por ω_k e ω_k'. Da equação (4.2) tem-se que

$$\omega_k = \sum_{J=1}^{N} v_{Jk} + x_k \tag{4.40}$$

e

$$\omega'_k = \sum_{J=1}^{N} v'_{Jk} + x'_k \tag{4.41}$$

Escrevendo a identidade

$$\sum_{k=1}^{N} (\omega_k \, \omega'_k - \omega'_k \, \omega_k) = 0 \tag{4.42}$$

tem-se

$$\sum_{k=1}^{N} \left[\left(\sum_{J=1}^{N} v_{Jk} + x_k \right) \omega'_k - \left(\sum_{J=1}^{N} v'_{Jk} + x'_k \right) \omega_k \right] = 0 \tag{4.43}$$

ou

$$\sum_{k=1}^{N} \sum_{J=1}^{N} (v_{Jk} \, \omega'_k - v'_{Jk} \, \omega_k) + \sum_{k=1}^{N} (\omega'_k x_k - \omega_k x'_k) = 0 \tag{4.44}$$

A equação acima é conhecida como o teorema de Tellegen para Sistemas Digitais.

Se o sistema é linear, pode-se escrever a equação (4.54) em termo das transformadas z. Então

$$\sum_{k=1}^{N} \sum_{J=1}^{N} (W'_k \, V_{Jk} - W_k V'_{Jk}) + \sum_{k=1}^{N} (W'_k \, X_k - W_k \, X'_k) = 0 \tag{4.45}$$

Por definição, duas malhas digitais são ditas inter-recíprocas se satisfazem a relação

$$\sum_{k=1}^{N} (W_k' X_k - W_k X_k') = 0 \qquad (4.46)$$

Considere uma malha de um sistema digital linear e sua transposta, isto é, uma malha onde a direção de todos os ramos é invertida. Tem-se então que se

$$V_{Jk} = F_{Jk} W_J \qquad (4.47)$$

e

$$V_{Jk}' = F_{Jk}' W_J' \qquad (4.48)$$

então

$$F_{Jk}' = F_{kJ} \qquad (4.49)$$

Substituindo as três equações acima na equação (4.45) tem-se

$$\sum_{J=1}^{N} \sum_{k=1}^{N} (W_k' W_J F_{Jk} - W_k W_J' F_{Jk}') + \sum_{k=1}^{N} (W_k' X_k - W_k X_k') = 0 \qquad (4.50)$$

ou

$$\sum_{J=1}^{N} \sum_{k=1}^{N} W_k' W_J F_{Jk} - \sum_{J=1}^{N} \sum_{k=1}^{N} W_k W_J' F_{Jk}' +$$

$$\sum_{k=1}^{N} (W_k' X_k - W_k X_k') = 0 \qquad (4.51)$$

Trocando-se J por k no primeiro termo da equação obtêm-se

$$\sum_{J=1}^{N} \sum_{k=1}^{N} (W_k\ W'_J\ F_{kJ} - W_k\ W'_J\ F'_{jK}) + \sum_{k=1}^{N} (W'_k\ X_k - W_k\ X'_k) = 0 \tag{4.52}$$

De acordo com a equação (4.49) a soma dupla é nula , de modo que

$$\sum_{k=1}^{N} (W'_k\ X_k - W_k\ X'_k) = 0 \tag{4.53}$$

O resultado acima indica que as malhas de um sistema digital e seu transposto são interrecíprocas. Para sistemas com uma entrada e uma saída isto significa que um sistema digital e seu transposto têm a mesma função de transferência. Isto é conhecido como o teorema da transposição. Este fato pode ser visto considerando-se dois nós quaisquer a e b. Assumindo todas as fontes na malha nulas, exceto em a para o sistema e em b para o seu transposto, tem-se da equação (4.53)

$$W_b\ X'_b = W'_a\ X_a \tag{4.54}$$

Se $X'_b = X_a$, então a mesma resposta que se obtem no nó b da malha original é vista no nó a da malha transposta.

4.7 - Sensibilidade em Malhas Digitais

A análise do efeito de quantização de coeficientes no comportamento de sistemas digitais baseia-se no estudo da sensibilidade de malhas a variações nos coeficientes. (Este item baseia-se no apresentado por Crochiere-Oppenheim, (1975) e Oppeinheim-Schaffer (1975 p. 173-181)). Assim, assuma que o sistema correspondente a uma dada malha tenha apenas uma entrada no nó a e saída no nó b. Então, a variável no nó b, $W_b(z)$ será dada por

$$Y_b(z) = W_b(z) = T_{ab}(z)\ X_a(z) = H(z)\ X(z) \tag{4.55}$$

onde $T_{ab}(z)$ é a função de transferência do nó a ao nó b.

Suponha que existam três malhas conforme definição a seguir. A primeira é a malha original com variáveis de nó W_k (Fig. 4.4.a) e transmitância entre os nós n e m F_{nm}. A segunda é a transposta da malha original, com variáveis de nó W'_k e transmitância $F'_{mn} = F_{nm}$ (Fig. 4.4.b). A terceira é idêntica à malha original, apenas diferindo na transmitância entre os nós n e m, F''_{nm} igual à do sistema original, mais uma perturbação ΔF_{nm}, ou seja

$$F''_{nm} = F_{nm} + \Delta F_{nm}\ .$$

Assuma também que todas malhas são excitadas pela mesma fonte X.

Pelo teorema de Tellegen para a segunda e terceira malhas.

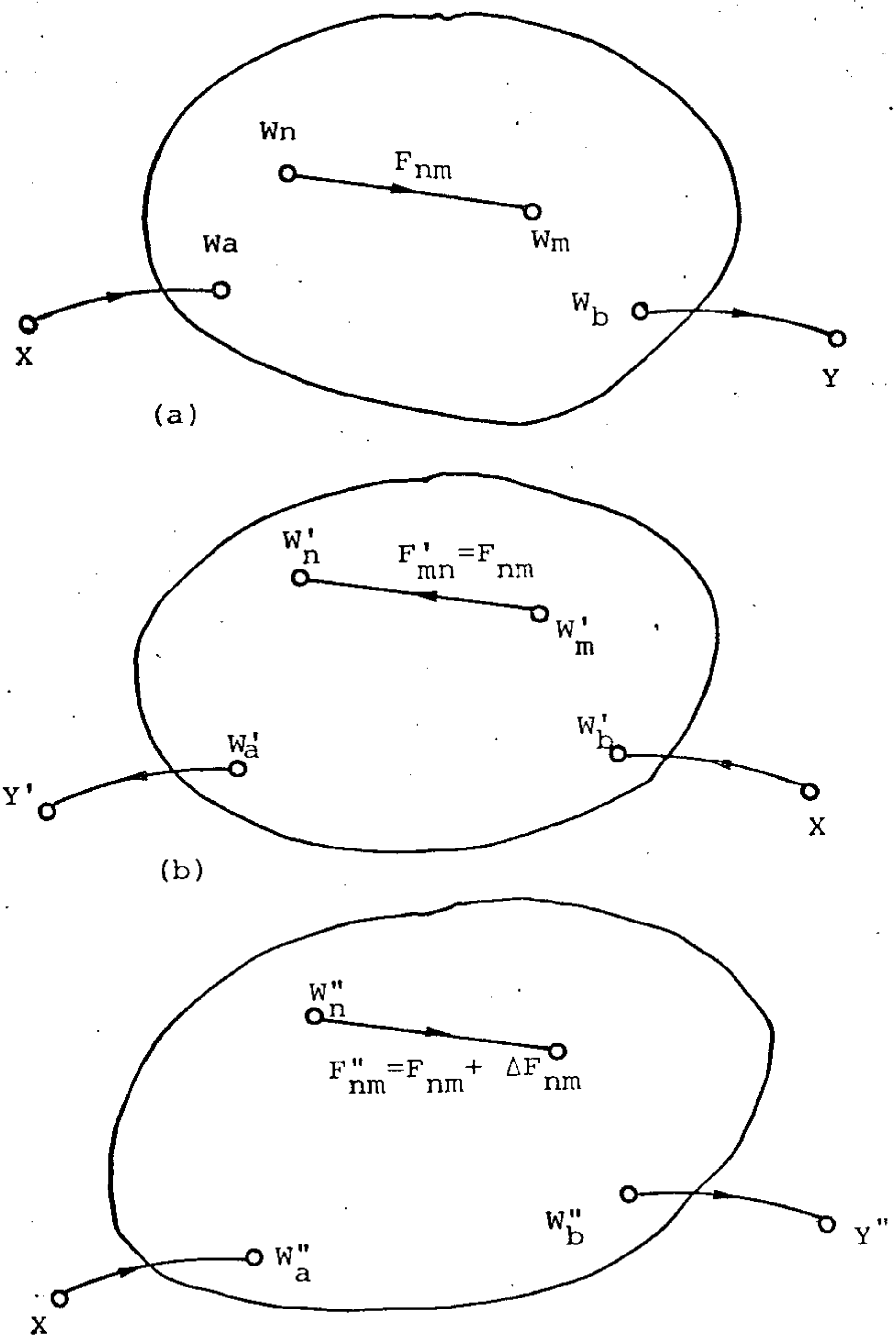


Figura 4.4 - Malhas usadas na obtenção da relação da sensibilidade.

$$\sum_{k=1}^{N} \sum_{J=1}^{N} (W_k' V_{Jk}'' - W_k'' V_{Jk}') + \sum_{k=1}^{N} (W_k' X_k'' - W_k'' X_k') = 0$$

(4.56)

ou

$$\sum_{k=1}^{N} \sum_{J=1}^{N} (W_k' V_{Jk}'') - \sum_{k=1}^{N} \sum_{J=1}^{N} W_k'' V_{Jk}' + \sum_{k=1}^{N} (W_k' X_k'' - W_k'' X_k') = 0$$

(4.57)

Trocando-se os índices do seguinte termo, chega-se a

$$\sum_{k=1}^{N} \sum_{J=1}^{N} (W_k' V_{Jk}'' - W_J'' V_{kj}') + \sum_{k=1}^{N} (W_k' X_k'' - W_k'' X_k') = 0$$

(4.58)

Como

$$V''_{Jk} = F''_{Jk} W''_J \qquad (4.59)$$

e

$$V'_{JK} = F'_{Jk} W'_J \qquad (4.60)$$

a equação (4.58) torna

$$\sum_{k=1}^{N} \sum_{J=1}^{N} W_J'' W_k' (F_{Jk}'' - F_{kJ}') + \sum_{k=1}^{N} (W_k' X_k'' - W_k'' X_k') = 0$$

(4.61)

A segunda e terceira malhas são transpostas, exceto para o ramo nm. Assim,

$$F''_{Jk} - F'_{kJ} = 0 \quad \forall\, k,J \;,\; J \neq n, \; k \neq m$$

e

$$F''_{nm} - F'_{mn} = \Delta F_{nm} \tag{4.62}$$

E, apenas uma fonte entra na malha, reduzindo a equação (4.61) a

$$W''_n \; W'_m \;\; \Delta F_{nm} + X(W'_a - W''_b) = 0 \tag{4.63}$$

Escrevendo a variável de nó W^i_k em termo da fonte de entrada X e da função de transferência correspondente, tem-se

$$W'_a = T'_{ba} \; X = T_{ab} \; X$$

$$W'_n = T'_{bm} \; X = T_{mb} \; X$$

$$W_n = T''_{an} \; X$$

$$W''_b = T''_{ab} \; X = [T_{ab} + \Delta T_{ab}] \; X \tag{4.64}$$

onde ΔT_{ab} denota a variação na função de transferência do sistema devido à variação ΔF_{nm} na transmitância F_{nm}.

Substituindo-se as equações acima na equação (4.63) obtem-se

$$\left[T''_{an} \; T_{mb} \;\; \Delta F_{nm} + T_{ab} - T_{ab} - \Delta T_{ab}\right] X^2 = 0 \tag{4.65}$$

e, como a equação acima é válida para qualquer fonte X, então

$$T''_{an} \; T_{mb} \;\; \Delta F_{nm} = \Delta T_{ab} \tag{4.66}$$

ou

$$\frac{\Delta T_{ab}}{\Delta F_{nm}} = T''_{an} \;\; T_{mb} \tag{4.67}$$

Tomando-se o limite quando $\Delta F_{nm} \to 0$, tem-se

$$\lim_{\Delta F_{nm} \to 0} \left[\frac{\Delta T_{ab}}{\Delta F_{nm}} \right] = \lim_{\Delta F_{nm} \to 0} \left[T''_{an} \; T_{mb} \right] \tag{4.68}$$

Quando $\Delta F_{nm} \to 0$, a terceira malha se aproxima da primeira, de maneira que

$$\frac{\partial T_{ab}}{\partial F_{nm}} = T_{an} \quad T_{mb} \tag{4.69}$$

A expressão acima estabelece que a sensibilidade da função de transferência de um sistema digital devido a variação na transmitância de um ramo entre dois nós $\underline{n}$ e $\underline{m}$ é expressa como o produto da função de transferência entre o nó $\underline{a}$ (de entrada) e o nó $\underline{n}$, pela função de transferência entre o nó $\underline{m}$ e o nó de saída $\underline{b}$.

A variação ΔT_{ab} na função de transferência do sistema T_{ab} devido a variações ΔF_{nm} na transmitância F_{nm}, expressa como uma expansão em série de Taylor é dada por

$$\Delta T_{ab} = \frac{\partial T_{ab}}{\partial F_{nm}} \Delta F_{nm} + \frac{1}{2} \frac{\partial^2 T_{ab}}{\partial F_{nm}^2} (\Delta F_{nm})^2 + \ldots \tag{4.70}$$

Finalmente, se ΔF_{nm} são os coeficientes c_i's do sistema digital

$$\Delta T_{ab_i} = \Delta H(z)_i = \frac{\partial H(z)}{\partial C_i} \Delta c_i + \frac{1}{2} \frac{\partial^2 H(z)}{\partial C_i} (\Delta c_i)^2 + \ldots \tag{4.71}$$

4.8 - Determinação do Comprimento de Palavra de Coeficientes

Neste item é proposta uma técnica estatística de determinação do comprimento de palavra de coeficientes de controladores digitais. Esta técnica utiliza como parâmetro de determinação do comprimento de palavra a variação no sinal de saída devido ao arredondamento. Esta técnica é uma extensão da técnica usada em filtragem digital e apresentada por Crochiere (1975) e Crochiere-Oppenheim (1975), onde o parâmetro de determinação do comprimento de palavra é a variação na magnitude da resposta em frequência.

Na equação (4.71) chegou-se a uma expressão para a variação na função de transferência de um sistema, devido à variação de um coeficiente qualquer. Numa aproximação de primeira ordem, tem-se que para m coeficientes arredondados,

$$\Delta H(\omega) = \sum_{i=1}^{m} \frac{\partial H(\omega)}{\partial c_i} \Delta c_i \quad , \qquad (4.72)$$

onde a variação da função de transferência é avaliada para z ao longo do círculo unitário, sendo

c_i o i-ésimo coeficiente arredondado

Δc_i a variação de c_i devido ao arredondamento,

e

$\frac{\partial H(\omega)}{\partial c_i}$ a sensibilidade de $H(\omega)$ em relação a c_i.

A função de transferência do sistema real, $H(\omega)$, pode ser interpretada como a função de transferência ideal , $H_0(\omega)$, (caso fosse implementada com precisão infinita),adicionada do termo representando a variação desta função de transferência (devido ao arredondamento de coeficientes) , $\Delta H(\omega)$. Ou seja,

$$H(\omega) = \Delta H(\omega) + H_0(\omega) \tag{4.73}$$

Aplicando ao sistema descrito pela equação acima um sinal $x(n)$, com transformada de Fourier para sinais discretos $X(\omega)$, então a saída, $y(n)$, terá a transformada

$$Y(\omega) = \left[\Delta H(\omega) + H_0(\omega)\right] X(\omega) \tag{4.74}$$

a qual pode ser reescrita como:

$$Y(\omega) = \Delta Y(\omega) + Y_0(\omega) \; . \tag{4.75}$$

Da equação (4.82) tem-se que

$$\Delta Y(\omega) = \left[\sum_{i=1}^{m} \frac{\partial H(\omega)}{\partial c_i} \Delta c_i\right] . X(\omega) \tag{4.76}$$

Assumindo que Δc_i seja uma variável aleatória , uniformente distribuída no intervalo de quantização (-q/2,q/2], onde q é o degrau de quantização, tem-se que (Oppenheim Schaffer (1975, p. 416)),

$$E\left[\Delta c_i\right] = 0 \tag{4.77}$$

$$\sigma^2_{\Delta c_i} = \frac{q^2}{12} \tag{4.78}$$

Da equação (4.76)

$$\Delta Y(\omega) = X(\omega)\ \frac{\partial H(\omega)}{\partial c_i}\ \Delta c_i + X(\omega)\ \frac{\partial H(\omega)}{\partial c_2}\ \Delta c_2 + \ldots + X(\omega)$$

$$\frac{\partial H(\omega)}{\partial c_m}\ \Delta c_m \qquad (4.79)$$

Assim,

$$E\ [Y(\omega)] = 0$$

e

$$\sigma^2_{\Delta Y(\omega)} = \sigma^2\ X(\omega)\ \frac{\partial H(\omega)}{\partial c_i}\ \Delta c_1 + X(\omega)\ \frac{\partial H(\omega)}{\partial c_2}\ \Delta c_2 + \ldots +$$

$$X(\omega)\ \frac{\partial H(\omega)}{\partial c_m}\ \Delta c_m \qquad (4.80)$$

$$= E\Big\{\Big[X(\omega)\ \frac{\partial H(\omega)}{\partial c_i}\ \Delta c_i + X(\omega)\ \frac{\partial H(\omega)}{\partial c_2}\ \Delta c_2 + \ldots +$$

$$X(\omega)\ \frac{\partial H(\omega)}{\partial c_m}\ \Delta c_m\Big]\ .$$

$$\Big[X(\omega)\ \frac{\partial H(\omega)}{\partial c_1}\ \Delta c_1 + \ldots + X(\omega)\ \frac{\partial H(\omega)}{\partial c_m}\ \Delta c_m\Big]^*\Big\}$$

$$(4.81)$$

Como para quaisquer números complexos a e b

$$(ab)^* = a^*b^* \text{ e } (a + b)^* = a^* + b^* \qquad (4.82)$$

obtém-se

$$\sigma^2_{\Delta Y(\omega)} = E\Big\{\Big[X(\omega)\ \frac{\partial H(\omega)}{\partial c_1}\ \Delta c_1 + \ldots + X(\omega)\ \frac{H(\omega)}{\partial c_m}\ \Delta c_m\Big]\ .$$

$$\Big[X^*(\omega)\left(\frac{\partial H(\omega)}{\partial c_1}\right)^*\ \Delta c_1 + \ldots + X^*(\omega)\left(\frac{\partial H(\omega)}{\partial c_m}\right)^*\ \Delta c_m\Big]\Big\} \tag{4.83}$$

ou

$$\sigma^2_{\Delta Y}(\omega) = E\left\{X(\omega)\ X^*(\omega)\left[\frac{\partial H(\omega)}{\partial c_1}\ \Delta c_1\left(\left(\frac{\partial H(\omega)}{\partial c_1}\right)^*\Delta c_1 + \ldots\right.\right.\right.$$

$$+\left.\left(\frac{\partial H(\omega)}{\partial c_m}\right)^*\Delta c_m\right) + \ldots + \frac{\partial H(\omega)}{\partial c_m}\ \Delta c_m\left(\left(\frac{\partial H(\omega)}{\partial c_1}\right)^*\Delta c_1 + \ldots\right.$$

$$\left.\left.\left.+\left(\frac{\partial H(\omega)}{\partial c_m}\right)^*\Delta c_m\right)\right]\right\} \tag{4.84}$$

ou,

$$\sigma^2_{\Delta Y(\omega)} = |X(\omega)|^2\left[E\left\{\frac{\partial H(\omega)}{\partial c_1}\left(\frac{\partial H(\omega)}{\partial c_1}\right)^*\Delta c_1^2 + \ldots\right.\right.$$

$$+\frac{\partial H(\omega)}{\partial c_1}\left(\frac{\partial H(\omega)}{\partial c_m}\right)^*\Delta c_1\ \Delta c_m + \ldots$$

$$+\frac{\partial H(\omega)}{\partial c_m}\left(\frac{\partial H(\omega)}{\partial c_1}\right)^*\Delta c_m\ \Delta c_1 + \ldots$$

$$\left.\left.+\frac{\partial H(\omega)}{\partial c_m}\left(\frac{\partial H(\omega)}{\partial c_m}\right)^*\Delta c_m^2\right\}\right] \tag{4.85}$$

ou,

$$\sigma^2_{\Delta Y(\omega)} = |X(\omega)|^2\left[E\left\{\frac{\partial H(\omega)}{\partial c_1}\left(\frac{\partial H(\omega)}{\partial c_1}\right)^*\Delta c_1^2\right\} + \ldots\right.$$

$$+ E\left\{\frac{\partial H(\omega)}{\partial c_1}\left(\frac{\partial H(\omega)}{\partial c_m}\right)^* \Delta c_1 \Delta c_m\right. + \ldots$$

$$+ E\left\{\frac{\partial H(\omega)}{\partial c_m}\left(\frac{\partial H(\omega)}{\partial c_1}\right)^* \Delta c_1 \Delta c_m + \ldots + E\left\{\frac{\partial H(\omega)}{\partial c_1}\left(\frac{\partial H(\omega)}{\partial c_m}\right)^* \Delta c_m^2\right\}\right]$$

(4.86)

Ainda mais, tem-se que

$$E\left\{\frac{\partial H(\omega)}{\partial c_i}\left(\frac{\partial H(\omega)}{\partial c_j}\right)^* \Delta c_i \Delta c_j\right\} = \frac{\partial H(\omega)}{\partial c_i}\left(\frac{\partial H(\omega)}{\partial c_j}\right)^*$$

$$E\{\Delta c_i \Delta c_j\} \quad (4.87)$$

Assumindo as variáveis aleatórias Δc_k independentes , então

$$E\{\Delta c_i \Delta c_j\} = \begin{cases} 0 & i \neq j \quad i,j = 1,\ldots,m \\ \sigma^2_{\Delta c_i} & i = j \quad i = 1,\ldots,m \end{cases} \quad (4.88)$$

Da equação acima e das equações (4.86) e (4.87) chega se a

$$\sigma^2_{\Delta Y(\omega)} = |X(\omega)|^2 \left[\left|\frac{\partial H(\omega)}{\partial c_1}\right|^2 \sigma^2_{\Delta c_1} + \ldots + \left|\frac{\partial H(\omega)}{\partial c_m}\right|^2 \sigma^2_{\Delta c_m}\right]$$

(4.89)

ou, em forma compacta, usando a equação (4.78),

$$\sigma^2_{\Delta Y(\omega)} = |X(\omega)|^2 \left[\sum_{i=1}^{m} \left|\frac{\partial H(\omega)}{\partial c_i}\right|^2\right] . \frac{q^2}{12} \quad (4.90)$$

Assumindo m grande o suficiente para poder-se considerar $\Delta Y(\omega)$ com distribuição gaussiana, então pode-se escolher um valor $x\ \sigma_{\Delta Y(\omega)}$, onde x é um fator de multiplicação do desvio padrão, tal que,

$$P\ \{\ |\Delta Y(\omega)|\ \leqslant\ x\ \sigma_{\Delta Y(\omega)}\} = g\ , \qquad (4.91)$$

ou seja, garante-se com probabilidade g que

$$|\Delta Y(\omega)|\ \leqslant\ x\ \sigma_{\Delta Y(\omega)} \qquad (4.92)$$

Na determinação do degrau de quantização (e consequentemente do comprimento de palavra) dos coeficientes, deseja-se, com uma certa probabilidade p, que a magnitude $|\Delta Y\ (\omega)|$ não exceda uma percentagem da magnitude $|Y_0(\omega)|$, ou

$$P\ \{|\Delta Y(\omega)|\ \leqslant\ k.\ |Y_0(\omega)|\}\ = p \qquad (4.93)$$

onde $0 < k < 1$.

Para ter-se uma estimativa para p, considere que

$$x\ \sigma_{\Delta Y(\omega)} \leqslant\ k\ |Y_0(\omega)| \qquad (4.94)$$

para todo ω tal que $0\ \leqslant\ \omega\ \leqslant\ \omega_s$.

Sendo $x > 0$, tem-se que

$$P\ \{|\Delta Y(\omega)|\ \leqslant\ x\ \sigma_{\Delta Y(\omega)}\} = P\ \{-\ x\ \sigma_{\Delta Y(\omega)}\ \leqslant\ \Delta Y(\omega)$$

$$\leqslant\ x\ \sigma_{\Delta Y(\omega)}\} \qquad (4.95)$$

ou

$$P\ \{|\Delta Y(\omega)|\ \leqslant\ x\ \sigma_{\Delta Y(\omega)}\} = F_{\Delta Y(\omega)}\ (x\ \sigma_{\Delta Y(\omega)}) -$$

$$F_{\Delta Y(\omega)}\ (-x\ .\sigma_{Y(\omega)}) \qquad (4.96)$$

onde $F_{\Delta Y(\omega)}$ (.) é a função distribuição de probabilidade de $\Delta Y(\omega)$, assumida gaussiana com média zero.

Então, como a função densidade de probabilidade $f_{\Delta Y(\omega)}$ é par, tem-se (Papoulis (1962 p. 131)).

$$F_{\Delta Y(\omega)}(-x\ \sigma_{\Delta Y(\omega)}) = 1 - F_{\Delta Y(\omega)}(x\ \sigma_{\Delta Y(\omega)}) \qquad (4.97)$$

Assim a equação (4.105) torna-se

$$q = P\{|\Delta Y(\omega)| \leqslant x\ \sigma_{\Delta Y(\omega)}\} = 2\ F_{\Delta Y(\omega)}(x\ \sigma_{\Delta Y(\omega)}) - 1 \qquad (4.98)$$

Das equações (4.94), (4.98) e do fato que

$$F_x(x_1) \leqslant F_x(x_2) \qquad \forall\ x_1 < x_2 \qquad (4.99)$$

tem-se que

$p \geqslant g.$

Isto significa que se a equação (4.94) é válida para todo $0 \leqslant \omega \leqslant \omega_s$, então garantindo, com probabilidade g, que

$$|\Delta Y(\omega)| \leqslant k\ |Y_0(\omega)| \qquad (4.100)$$

com definido pela equação (4.91).

Das equações (4.92), (4.93), (4.74) e (4.75)

$$x.\ |X(\omega)| \left[\sum_{i=1}^{m} \left|\frac{\partial H(\omega)}{\partial c_i}\right|^2\right]^{1/2} \frac{q}{\sqrt{12}} \leqslant k\ |X(\omega)|\ |H_0(\omega)| \qquad (4.101)$$

ou seja

$$x.\ \frac{q}{\sqrt{12}} \left[\sum_{i=1}^{m} \left| \frac{\partial\ H(\omega)}{\partial\ c_i} \right|^2 \right]^{1/2} \leqslant k\ |H_0(\omega)| \tag{4.102}$$

Para a determinação de um q que satisfaça a inequação acima para todo ω em $0 \leqslant \omega \leqslant \omega_s$, define-se a função

$$q(\omega) = \frac{\sqrt{12}\ .\ k\ .\ |H_0(\omega)|}{x.\left[\sum_{i=1}^{m} \left| \frac{\partial\ H(\omega)}{\partial\ c_i} \right|^2 \right]^{1/2}} \tag{4.103}$$

Determina-se, então, q, tal que

$$q = \min_{o\ \leqslant\ \omega\ \leqslant\ \omega_s} q(\omega) \tag{4.104}$$

O comprimento de palavra dos coeficientes incluindo o bit de sinal, é dado por

$$w = 2 + i_m - i_\ell \tag{4.105}$$

onde 2^{i_m} é o valor do bit mais significante, representando a grandeza do coeficiente, e 2^{i_ℓ} é o valor do bit menos significante (Crochiere-Oppenheim (1975, p. 589)), e dado por

$$i_\ell \approx \log_2 q \tag{4.106}$$

4.9 - Determinação da Resposta em Frequência Usando a Representação Matricial

Nos itens referentes à análise estatística para a determinação do comprimento de palavra de variáveis e coeficientes, foram obtidas as equações (4.39) e (4,103), as quais exigem o cálculo das funções de transferência entre a entrada e a saída, entre a entrada e alguns nós da malha, e entre alguns nós e a saída, avaliadas na faixa de frequências $0 \leqslant \omega \leqslant \omega_s$, onde $\omega_s = 2\pi$. A representação matricial de uma malha leva ao cálculo simultâneo destas respostas em frequência, como será visto a seguir, diminuindo o tempo de execução de programas.

Considere a equação (4.9), repetida abaixo na forma

$$\left[I - F_c^{\,t} - F_d^{\,t}\, z^{-1} \right] \underline{Y}(z) = \underline{X}(z). \tag{4.107}$$

Considere que um sistema tem apenas uma entrada no nó $\underline{a}$ e uma saída no nó $\underline{b}$. Seja $T_{jk}(z)$ a função de transferência entre o nó j e o nó k. Tem-se que

$$T_{ak}(z) = \frac{Y_k(z)}{X_a(z)} \qquad k = 1,2, \ldots, N \tag{4.108}$$

$$T_{jb}(z) = \frac{Y_b(z)}{Y_j(z)} \qquad j = 1,2,\ldots, N \tag{4.109}$$

e

$$H(z) = T_{ab}(z) = \frac{Y_b(z)}{X_a(z)} \tag{4.110}$$

A equação (4.107) pode ser vista como um sistema de N equações lineares simultâneas. Assumindo $X_a = 1$ e $X_k=0$ para $k \neq a$, então, a solução do sistema de equações, usando a - ritmética complexa, para uma dada frequência , com $0 \leqslant \omega \leqslant \omega_s$, dará o conjunto de resposta em frequência $T_{ak}(\omega)$ (ou $T_{ak}(e^{j\omega})$), para aquela frequência. Se $\underline{k}=\underline{b}$, então ter-se-á a resposta em frequência do sistema $H(\omega)$, para aquela frequência.

Para o cálculo da função de transferência entre qualquer nó $\underline{k}$ e o nó de saída $\underline{b}$, usando o teorema da transposição, calcula-se simultaneamente todas as respostas $T_{jb}(\omega)$, definindo o sistema transposto de (4.107), dado por

$$\left[I - F_c - F_d\, z^{-1}\right] W'(z) = X'(z) \qquad (4.111)$$

e escolhendo

$$X'_b = 1 \text{ e } X'_j = 0 \text{ para } j \neq b \text{ ,} \qquad (4.112)$$

tendo em mente que

$$T_{jb}(\omega) = T'_{bj}(\omega) \qquad j = 1,2,\ldots,N \qquad (4.113)$$

Com um método numérico, como por exemplo, a eliminação gaussiana, usando aritmética complexa, calculam-se as soluções das equações (4.107) e (4.111). Na integração um método numérico como a regra trapezoidal ou o método de Runge - Kutta pode ser usado.

Para sistemas onde a matriz das equações (4.107) e (4.111) são singulares em algumas frequências ω , o método apresentado para a determinação da resposta em frequência não funciona . No caso do sistema considerado tal fato ocorre para $\omega = 0$, e em casos gerais quanto o sistema tem polos no círculo unitário.

Como o método é poderoso, no sentido que permite o cálculo da resposta em frequência simultaneamente entre diversos nós e os nós de entrada ou de saída, optou-se pelo uso deste método. Assim, neste trabalho, as análises da determinação do comprimento de coeficientes e variáveis leva em conta apenas o controlador. Os estudos para a adaptação (ou extensão) do método para os casos de singularidade da matriz estão em andamento, e a sua apresentação será feita em trabalhos posteriores. Intuitivamente, espera-se que o uso do sistema em malha fechada leve a um número menor de bits necessários para ter-se uma mesma degradação no desempenho do sistema, sendo válida, portanto, a adaptação (ou extensão) pretendida.

4.10 - Resumo

Neste capítulo apresentou-se a determinação do comprimento de palavra de variáveis e coeficientes, na representação em ponto fixo, na implementação de controladores digitais. No caso das variáveis utilizou-se de uma técnica, desenvolvida para processamento de sinais digitais, enquanto que no caso dos coeficientes, uma técnica foi proposta, baseada também no usado na área citada.

Teoremas básicos, necessários ao desenvolvimento e apresentação das técnicas, foram também apresentados.

Finalmente, uma técnica computacional aplicada às técnicas citadas foi apresentada.

5. PROJETO ASSISTIDO POR COMPUTADOR

5.1. Introdução

Este capítulo descreve os programas responsáveis pelo projeto dos controladores assistido por computador, segundo a teoria desenvolvida nos capítulos anteriores, bem como os resultados do projeto do sistema.

Os programas são escritos na linguagem PASCAL, e são auto-suficientes, não utilizando nenhum subprograma externo. Eles foram implementados num computador PDP-10 da "Digital Equipment Corporation". Limitações na versão do Compilador PASCAL usado fizeram necessária pequenas desestruturações dos programas, bem como a separação em projeto e simulação dos controladores. No entanto, procurou-se indicar no corpo dos programas os locais onde ocorreram as citadas desestruturações.

A sequência de projeto recomendada é determinar os coeficientes dos controladores e realizar as simulações até obter-se uma resposta aceitável para o sistema em malha fe -

chada. Com o sistema satisfazendo as especificações, então calcular o comprimento de palavra de variáveis e coeficientes.

A seguir são descritos os três programas. Depois são mostrados os resultados de projeto para o sistema deste trabalho, de acordo com o visto pelo projetista em seu terminal. Os programas têm uma representação esquemática na qual um programa principal utiliza subprogramas agrupados como interface homem-máquina (entrada e saída de dados), subprogramas de métodos e subprogramas auxiliares, como mostrado na figura 5.1.

5.2. Projeto dos Controladores

O programa CONTROLADORES OTIMOS apresentado no apêndice A2 é responsável pelo projeto dos controladores desenvolvidos no capítulo 3.

Inicialmente os dados do conjunto servomotor c.c. - carga, bem como o período de amostragem desejado são introduzidos. Isto é feito pelas "procedures".

.LER PARAMETROS

.MUDAR PARAMETROS

.LER PERIODO

.MUDAR PERIODO

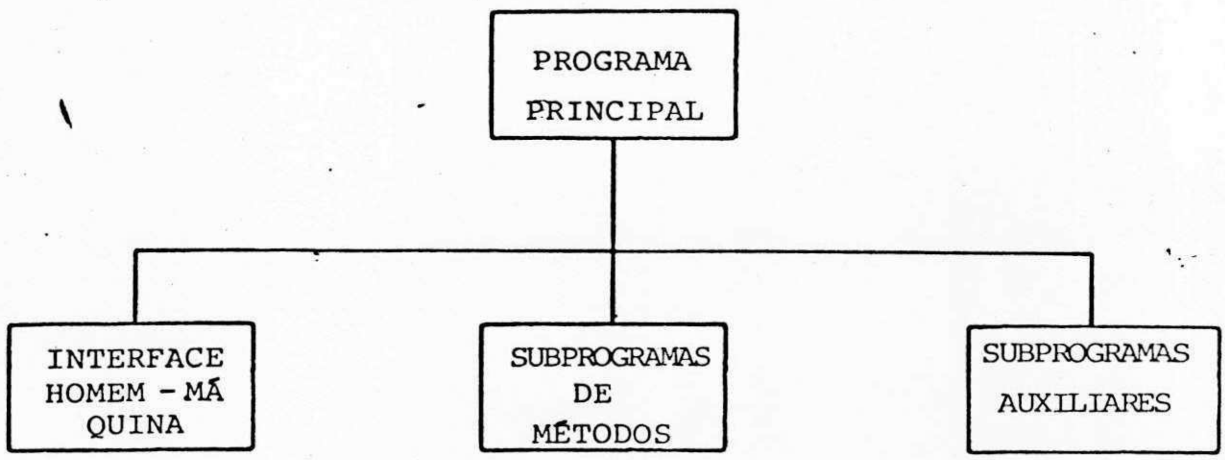


Figura 5.1 - Representação esquemática dos programas.

A seguir são calculados e impressos os polos do sistema contínuo em malha aberta e os coeficientes do sistema discreto, para o período de amostragem desejado. Também são calculados os coeficientes do sistema discreto para um período de amostragem real, a serem usados na simulação. São usadas as "procedures"

.RAIZES REAIS

.COEFICIENTES

São então calculados e impressos os coeficientes do regulador ótimo e do observador de estados, usando-se a "procedure" :

.RICCATI.

São entradas as matrizes de ponderação, e os coeficientes calculados; o vetor de ganhos do regulador e de realimentação do observador são impressos. Os coeficientes dos controladores são então calculados.

No caso dos controladores com observador, são calculadas as matrizes a serem diagonalizadas, utilizando-se as "procedures" de operações em matrizes. Os autovalores associados são calculados e impressos, a matriz dos autovetores e sua inversa são calculadas, bem como os coeficientes do sistema diagonalizado. E, são feitas as transformações lineares necessárias à obtenção dos controladores finais, sendo por fim, os coeficientes destes controladores impressos. Usam-se as "procedures"

.AUTOVALORES

.PAUTOVETORES

.PINVERSA ;

as de operações em matrizes e vetores complexos

.CMULTMTZNN

.CMULTMNN1,

e outras de operações em matrizes e vetores reais.

No caso do controlador dinâmico, são calculados: a inversa da matriz (A - bk), os coeficientes do controlador ,os quais são impressos, usando-se as "procedures"

.INVERAMBK

.KDALFABETA.

Finalmente, são impressos os coeficientes do sistema discreto usado na simulação.

O procedimento pode ser repetido para outro controlador, ou outro período de amostragem.

O restante das "procedures" e "functions" são auxiliares aos subprogramas principais. São elas de leitura e escrita de matrizes e vetores,

.EVETMATRIZ

.LERMATRIZ

.ESCRMATRIZ

.ESCVL

.ESCVC,

de cálculo de alguns parâmetros

.ALFA

.BETA

.A22

.A23

.A32

.A33

.B2

.B3,

operações em matrizes e entre matrizes, matriz-vetor, vetor-vetor,

.ZEREMATRIZ

.MIGUALM

.MTZTRANSP

.SOMAR NN

.SUBTRAI NN

.MULTVET 1N1

.MULTVET N1N

.MULTVETCNST

.MULTUM 1NN

.MUTMTZ NN ,

operações em números complexos,

.SOMAR

.SUBTR

.MULTI

.DIVID

.IGUAL

.MENOS

e outras operações auxiliares,

.TEST CONV

.CALCULEK

.CALCULEP

.VALORPOLINOMIO

.VALORDERIVADA

.RAIZESSEG

.RAIZNEWTON

.AUTOVETOR

.DISTAUTOVETORES

.IGUALVETORES

.TRANSF1DEGR

com os seus significados apresentados no fim dos respectivos corpos.

Os métodos numéricos usados são particularizados para a aplicação em questão. Por exemplo, no cálculo de autovalores do sistema de terceira ordem, um autovalor é calculado pelo método de Newton-Raphson, e os dois restantes resolvendo-se diretamente a equação quadrática resultante.

5.3. Simulação do Sistema em Malha Fechada

A simulação do sistema em malha fechada é executada pelo programa SIMULAÇÃO, apêndice A3.

Inicialmente o tipo do controlador a ser simulado é selecionado, e os respectivos dados são fornecidos como entrada. A seguir, o sistema em malha fechada é simulado para um período de amostragem de 1/8 (um oitavo) do período real, de maneira a se poder verificar a existência de oscilações entre os períodos de amostragem. Como entradas de simulação podem ser escolhidas degrau, rampa e parábola com amplitudes variáveis. E, por fim, são impressos os resultados da simulação.

As "procedures" usadas são

. SISTEMA

. CONTROL 1

. CONTROL 2

. CONTROL 3

. TRACECURVAS,

com os seus significados comentados no fim dos respectivos corpos.

Todo o procedimento de simulação pode ser repetido para outras entradas, outros parâmetros ou outros controladores, numa única execução do programa .

5.4. Determinação do Comprimento de Palavra de Variáveis e Coeficientes

O programa COEFICIENTES (Apêndice A4) é responsável pela determinação do comprimento de palavra de variáveis e coeficientes, usando as técnicas apresentadas no capítulo 4.

Inicialmente são fornecidos como entradas os coeficientes dos ramos do diagrama de fluxo de sinais do sistema em estudo, a seguir os nós de atrasos (podendo-se entrar coeficientes associados diferentes de 1), os nós de entrada, saída e os nós com variáveis quantizadas em suas entradas, bem como o número destas variáveis (resultantes da multiplicação por coeficientes não inteiros) e, finalmente, o fator de multiplicação da variância na saída do sistema (probabilidade de confiança nos resultados), o fator de variação máxima da magnitude da resposta em frequência do sistema (determinação do comprimento de palavra dos coeficientes), a amplitude máxima do ruído proveniente da quantização de variáveis, e o número de divisões no período entre 0 e 2 π (usado na integração). Os resultados são calculados e impressos.

A "procedure" principal é a que determina a solução do sistema de equações

. SOLUÇÃOSISTEMA ,

que usa o método de Gauss, com aritmética complexa. Como técnica de integração usa-se a integração trapezoidal.

O procedimento de cálculo pode ser repetido para outro número de divisões entre 0 e 2π , ou para outras especificações de projeto.

Neste programa a "procedure" SOLUÇÃOSISTEMA é transportada para o corpo do programa, de maneira a se solucionar as limitações impostas pelo sistema onde o programa foi implementado.

5.5. Estrutura de Interação e Resultados

A estrutura de interação dos programas apresentados nos itens anteriores é mostrada na figura 5.2.

Os programas interagem com o usuário, via terminal, através de uma estrutura de comunicação pergunta-resposta. O programa solicita consecutivamente os dados que necessita, esperando do usuário a sua entrada, até que ele tenha as informações necessárias. A seguir é feito o processamento destes dados com a impressão dos resultados, também no terminal. Este processamento é repetido até o fim do projeto.

Os resultados do projeto assistido por computador para o sistema implementado são apresentados no apêndice A9, onde pode ser verificada a estrutura de interação descrita acima.

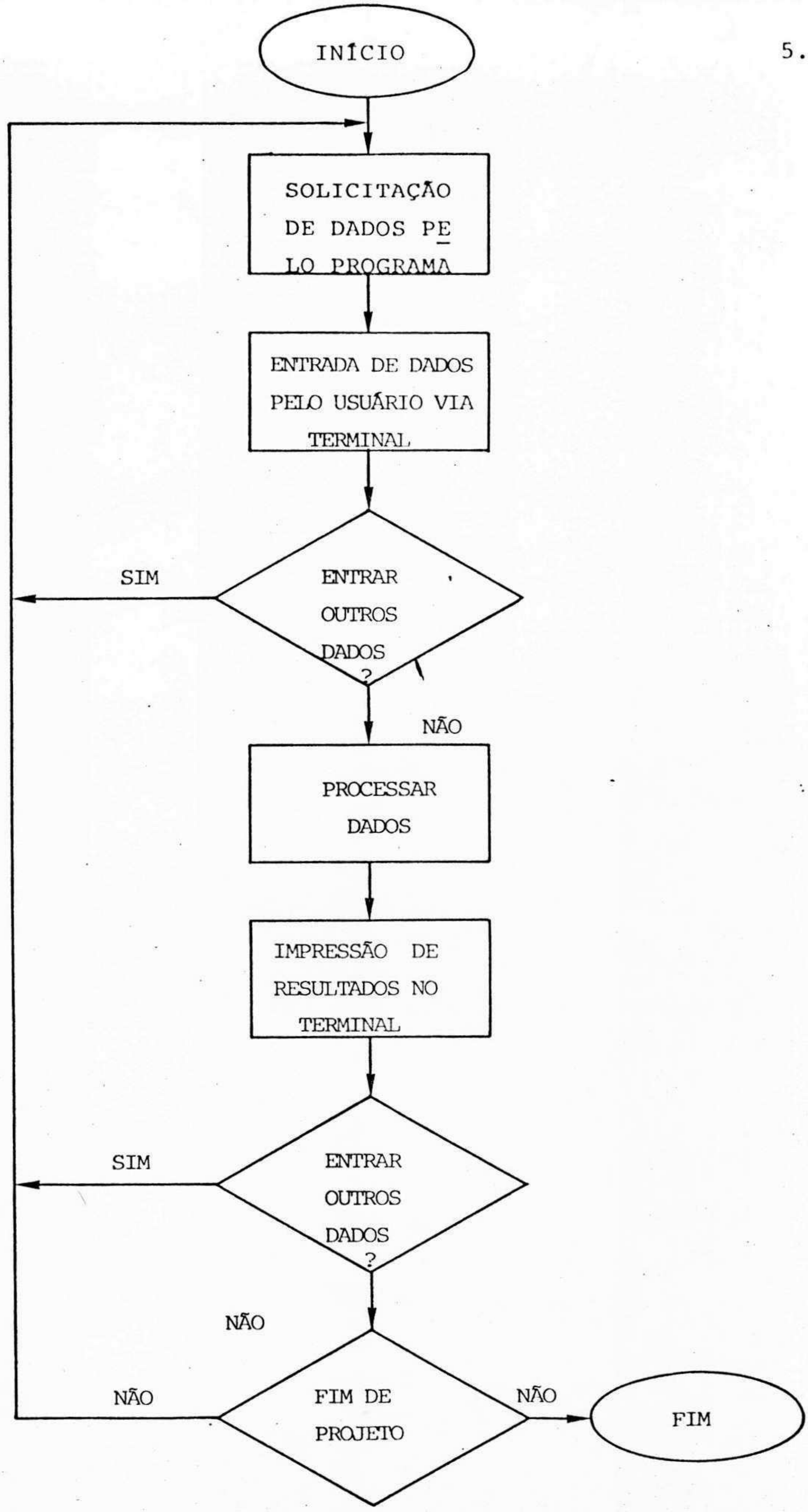


FIG. 5.2. ESTRUTURA DE INTERAÇÃO

5.6. Resumo

Este capítulo apresentou os programas responsáveis pelo projeto do sistema de controle deste trabalho, auxiliado por computador. Os resultados do projeto utilizados na implementação deste trabalho são também apresentados.

6. IMPLEMENTAÇÃO DOS CONTROLADORES

6.1. Introdução

Este capítulo apresenta a implementação de dois dos três controladores projetados no capítulo anterior. São implementados os controladores I e II. O controlador III (dinâmico) não foi implementado por apresentar na simulação ciclo limites considerados inaceitáveis.

São apresentados o hardware onde o sistema foi implementado, dando-se maior ênfase à parte desenvolvida, o software dos controladores, e os resultados obtidos para a resposta ao degrau.

6.2. Hardware

O sistema de controle foi implementado no Sistema de Desenvolvimento do Laboratório de Sistemas/Microprocessadores (LSM) do DEE-FEC-UNICAMP. A configuração utilizada, mostrada na Figura 6.1, compreende dos seguintes cartões:

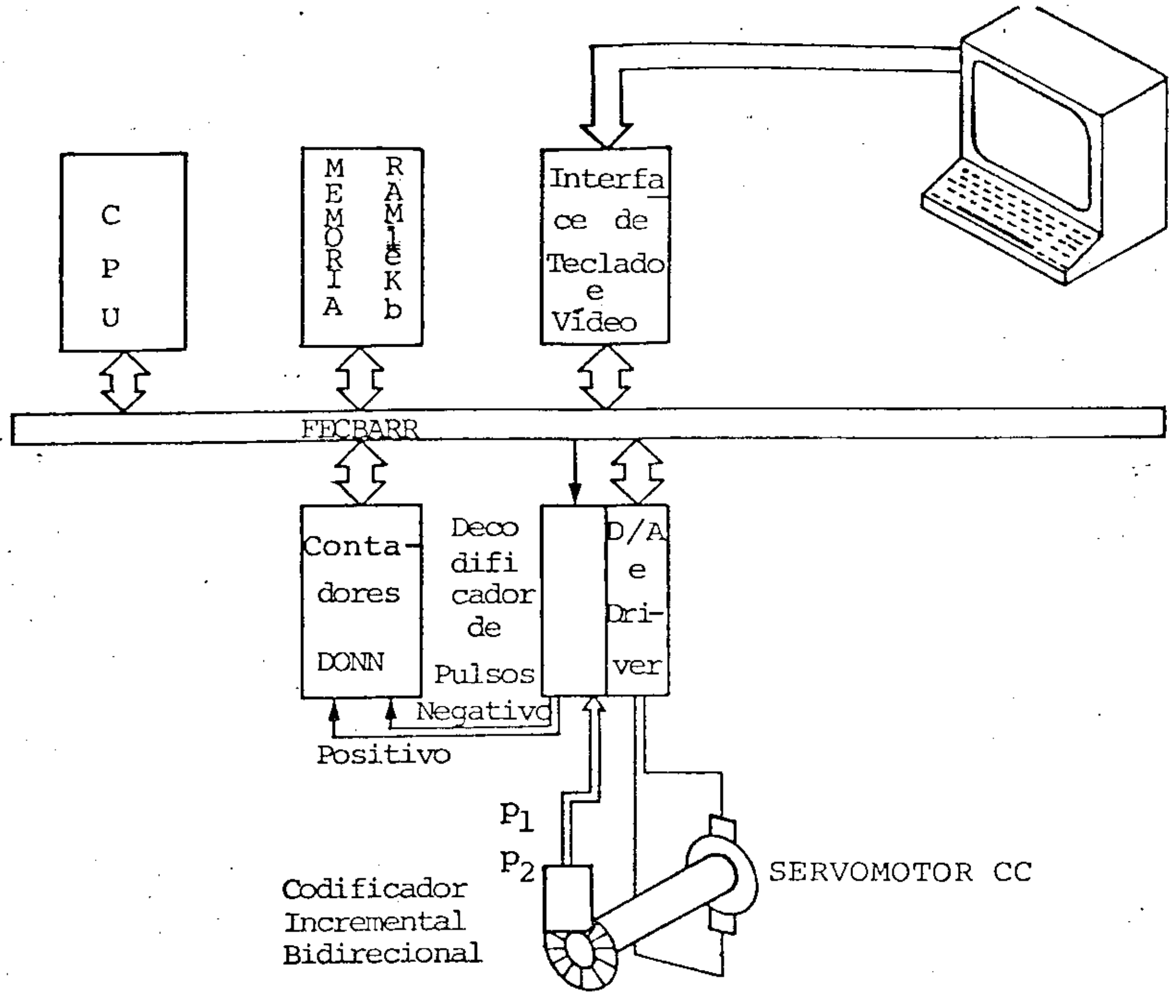


Figura 6.1 - Configuração do hardware utilizado.

.U.C.P. baseada no microprocessador INTEL 8085A, com monitor em ROM e 256 bytes de RAM interna;

.Interface para techado e vídeo;

.Memória RAM de 16 Kbytes;

.Contadores "down" de 16 bits, baseados no C.I. INTEL 8253;

Estes cartões são padronizados em torno de um barramento, o FECBARR, desenvolvido no LSM. Maiores detalhes sobre cartões e barramento podem ser encontrados nas referências LSM(28), LSM(29), LSM(30), LSM(31), LSM(32).

Também compatível com o FECBAR, foi desenvolvido um cartão de interface com o processo, contendo um conversor D/A, um amplificador operacional de potência e um decodificador de pulsos, mostrados no diagrama do Apêndice A5.

A parte de saída do cartão de interface é acoplada às linhas de dados do FECBARR através do "latch" SN74LS116 - (Signetics (44)), selecionado pelo circuito denominado decodificador de endereços. Os dados escritos neste "latch" são mantidos fixos por um período de amostragem do controlador. A conversão destes dados digitais para a forma analógica é realizada pelo conversor D/A SE5009 (Signetics (45)). O nível de potência adequado é fornecido pelo amplificador operacional de potência μA791 (Fairchild(16)), com corrente máxima de 1A.

O deslocamento é medido por um codificador incremental bidirecional. Ele consiste de um par de LED's infravermelhos TIL 32 (Texas (48)) excitando, através da reflexão em um disco graduado pela alternância entre uma faixa refletora e uma não refletora e acoplado ao eixo do servomotor, um par de fototransistores TIL 78 (Texas (48)). O circuito mostrado na figura 6.2 tem como saída um nível de corrente de 20 mA, quando há condução (faixa não refletora). O disco tem 36 faixas refletoras e 36 não refletoras. Como saídas do codificador obtêm-se duas sequências de pulsos defasadas entre si de $\pm$ 2,5^{o}, de acordo com o período de rotação.

As saídas do codificador imcremental bidirecional são entradas para o decodificador de pulsos mostrado no diagrama do apêndice A5. As entradas deste decodificador são desacopladas eletricamente do resto do circuito por acopladores óticos TIL 126 (Texas (48)). Depois, "Schimitt Trigger's" SN74LS14 (Signetics (44)) são usados para minimizar o efeito de ruídos e dar uma forma quadrada aos pulsos vindos dos acopladores. O decodificador tem duas saídas, uma para cada sentido de rotação. Pulsos de largura 4 vezes a do período de clock (CLK) do FECBARR são emitidos de acordo com o diagrama de tempo mostrado na figura 6.3. O diagrama de estados associados ao decodificador também é mostrado nesta figura.

As saídas do decodificador de pulsos decrementam dois dos três contadores do cartão dos Contadores. O terceiro é

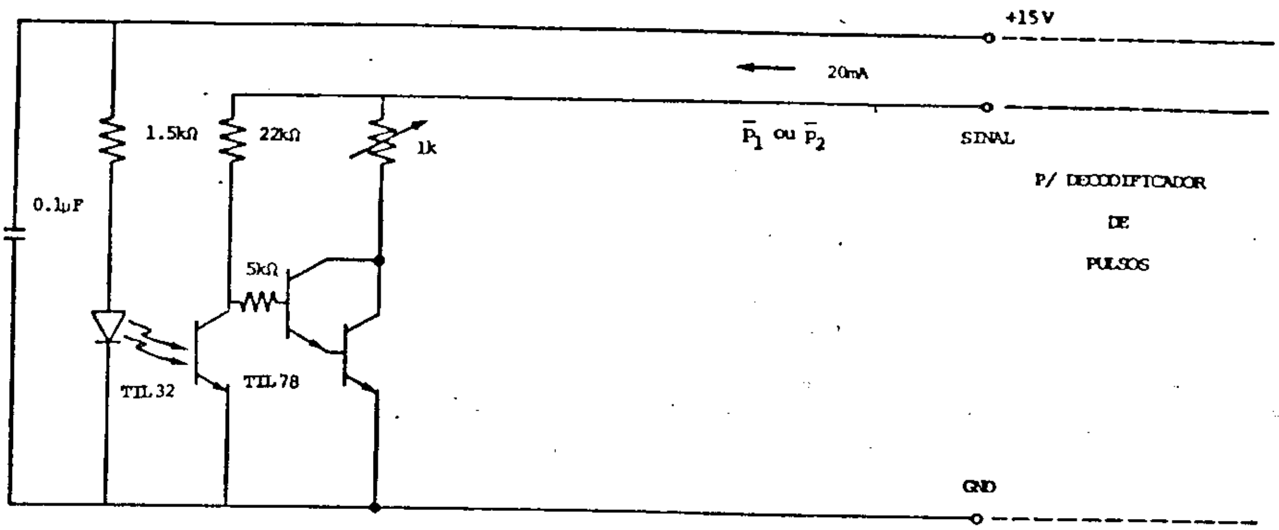


Figura 6.2 - Circuito do codificador incremental bidirecional.

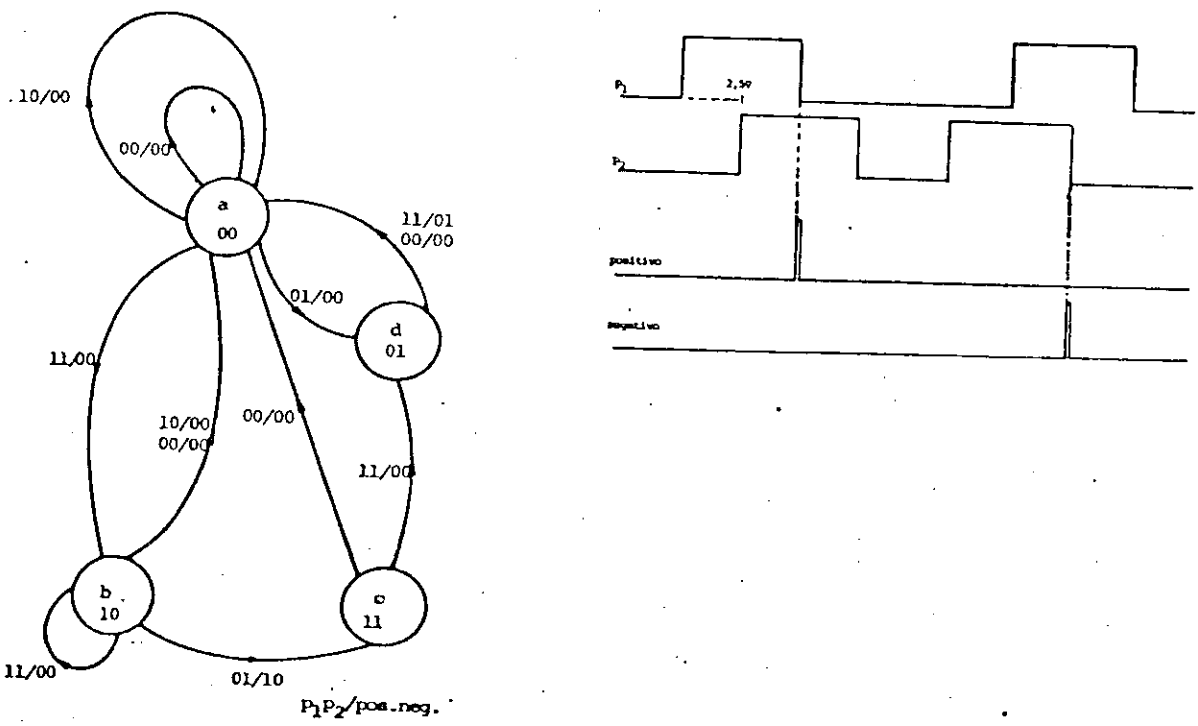


Figura 6.3 - Diagramas de estados e de tempo do decodificador de pulsos.

usado para determinar o período de amostragem do controlador, com uma interrupção não mascarável (NMI) no microprocessador ao fim de sua contagem decrescente.

6.3. Software

Este item descreve os programas em linguagem "assembley" do microprocessador Intel 8085 que implementam os dois controladores no sistema apresentado no item anterior. Para maiores detalhes sobre a família do microprocessador 8085A o leitor é referenciado ao manual Intel (20).

A primeira parte dos programas, mostrados nos apêndices A6 e A7 , idêntica em ambos é a inicialização. Quando no início de operação dos controladores a saída de controle é zerada, os contadores de deslocamento e as variáveis associadas são inicializados, as outras variáveis do controlador são zeradas, e , finalmente, o gerador de comandos é inicializado. Então, o controlador é executado.

Os controladores inicialmente escrevem a saída de controle calculada como a adição do resultado dos controladores lineares calculado no período anterior à alimentação direta ("feedforward") calculada a partir do sentido de rotação (sinal da velocidade).

A seguir o contador de período de amostragem é inicializado, há a aquisição de dados dos contadores de deslocamen to, e são feitos cálculos iniciais sibre estes dados, de modo a se obter o deslocamento total (ou posição). O diagrama de fluxo de sinais desta parte é mostrado nas figuras 6.4 e 6.5.

A seguir o restante das estruturas mostradas nas figuras (6.4) (controlador I) e (6.5) (controlador II), são implementadas. Elas são obtidas das figuras (3.4) e (3.9), respectivamente, com a eliminação dos ramos cujos coeficientes, devido ao comprimento de palavra obtida no capítulo anterior, para os mesmos, foram arrendondados para zero.

Estas estruturas têm basicamente como operações aritméticas a adição e a multiplicação por uma constante.

A adição é realizada em complemento de dois com precisão dupla. São feitos testes para a deteção de overflow e sua consequente correção é realizada pela substituição do resultado pelas representações máxima (positivo) e mínima (negativo) da representação usada.

A multiplicação também é realizada em complemento de dois (Duncan (1979)). O comprimento de palavra é reduzido por arredondamento, para a precisão calculada no capítulo anterior.

Por fim, as variáveis são atualizadas para o próximo período. E, para fins de teste, um gerador de comandos, de ondas quadradas de amplitude (AMP) e período (SP1) alteráveis, é implementado.

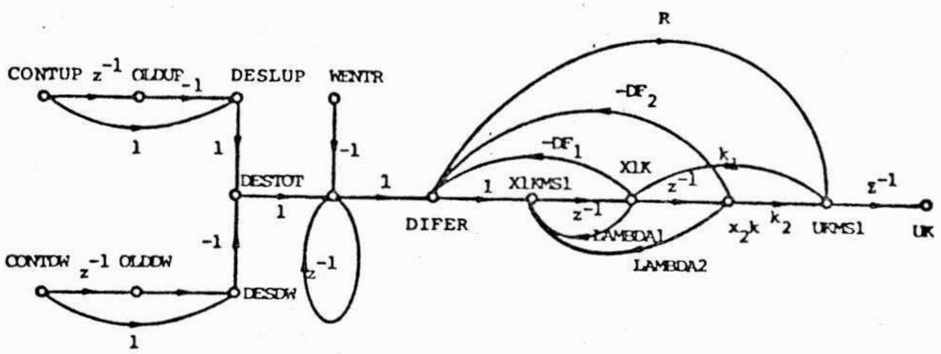


Figura 6.4 - Diagrama de fluxo de sinais do controlador I implementado.

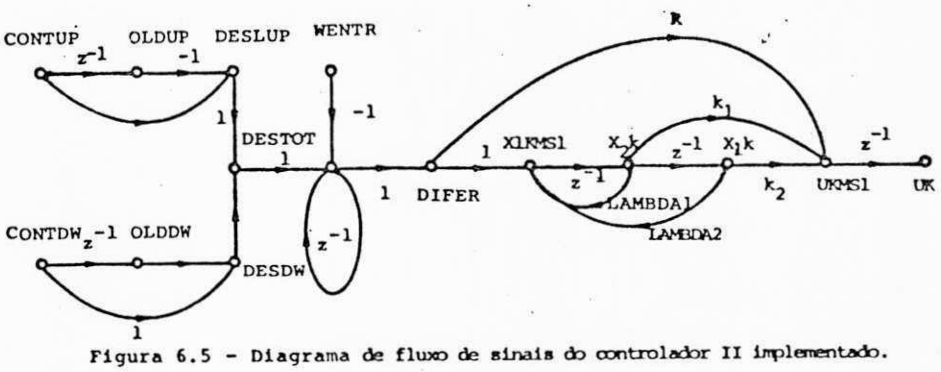


Figura 6.5 - Diagrama de fluxo de sinais do controlador II implementado.

Ao término desde processamento o microprocessador é parado ("Halt") até o próximo período de amostragem determinado pela interrupção no fim de contagem do contador de período de amostragem, quando o procedimento do controlador é repetido.

Como limitações na implementação dos controladores, devido à baixa velocidade de processamento do microprocessador disponível, eliminiu-se ao máximo o uso de desvios e chamadas a subrotinas, bem como limitou-se a representação máxima de inteiros para 8 bits. Com isso, os programas obtidos foram programas longos, com o maior deles (Controlador I) ocupando em torno de 4 Kbytes de memória com um tempo de execução máximo de aproximadamente 7,2ms para um período de amostragem de 7.8 ms.

6.4. Resultados

As figuras 6.6 e 6.7 mostram as respostas obtidas para uma entrada degrau de comando aos controladores implementados I e II, respectivamente. A análise dos resultados obtidos será feita no item 7.3.

6.5. Resumo

Este capítulo apresentou a implementação dos controladores I e II projetados no capítulo 5. Inicialmente des creveu-se o hardware utilizado, e a seguir o software dos controladores. Por fim, as respostas ao degrau obtidas para controlador foram apresentadas.

7. ANÁLISE DOS RESULTADOS

7.1. Introdução

Neste capítulo são apresentados a análise dos resultados obtidos no projeto e implementação dos controladores, apresentados nos ítens 5.5 e 6.4.

Inicialmente são analisados os resultados do projeto e a seguir os resultados da implementação, em comparação com as especificações de projeto.

7.2. Projeto

Neste item são comentados os resultados do projeto dos controladores, obtidos para as especificações apresentadas no item 5.5.

Através de simulação, para as matrizes de ponderação escolhidas, chegou-se a um período de amostragem no qual a resposta do sistema era aceitável. Com isso foram obtidos os controladores com observadores de estados. O controlador di nâmico, aproximação do controlador ótimo não fornecem, para os parâmetros do projeto dos controladores, uma resposta aceitável. A variação dos parâmetros de projeto na procura de uma resposta aceitável na simulação, levou às matrizes de ponderação e aos períodos de amostragem mostrados no item 5.5. Os resultados de projeto e simulação levaram então às decisões de projeto de implementar os controladores que tiveram respostas (ao degrau) aceitáveis. Assim, foram selecionados para implementação os controladores com observadores.

Na determinação do comprimento de palavra de coeficientes pode-se observar o seguinte:

. Um comprimento de palavra menor para os controladores mais complexos, aumentando para os menos complexos. Assim, o controlador I, com as características de realimentação do observador, exige um comprimento de palavra menor que o controlador II, onde, nas simplificações estruturais, eliminou-se a característica de realimentação. E ambos, um comprimento bem menor que a aproximação dinâmica.

. Com o comprimento de palavra obtido, os coeficientes cujo arredondamento dêem zero são desprezíveis no comportamento do controlador, para os níveis de perturbação (devido a quantização) aceitáveis, e por isso são eliminados da estrutura do controlador. Com este resultado, pode-se então projetar os controladores usando um modelo mais complexo para o sistema, ao invés de fazerem-se simplificações no modelo do sistema. Após o cálculo dos controladores que satisfaçam as especificações de projeto desejadas, simplifica-se a estrutura do controlador obtido. Para isso são definidas degradações máximas aceitáveis das especificações.

7.3. Implementação

Os parâmetros do sistema usados no projeto dos controladores são nominais, não tendo sido feita a identificação. Além disso, o servomotor utilizado, por ter sido usado anteriormente (em controle de velocidade) provavelmente teve suas características alteradas durantes este período. Por exemplo, constatou-se que o atrito de Coulomb era diferente em módulo, para cada sentido de rotação. Também as escovas já estavam parcialmente desgastadas. Assim, sob estas condições, são comparados neste item os resultados obtidos pela implementação dos controladores em relação aos resultados da simulação.

Nas figuras 7.1 e 7.2 são mostradas as respostas simuladas e reais para o sistema em malha fechada, controladores I e II, respectivamente. O controlador I, tem uma resposta melhor que o controlador II. Este resultado decorre do fato do controlador I manter em sua estrutura a característica de realimentação no observador de estados, enquanto o controlador II não. Assim, variações dos parâmetros do sistema ou perturbações levam a um erro entre as variáveis de estado dos observadores e as variáveis de estado do sistema real. A presença da característica de realimentação do observador de estados leva a uma correção deste erro, de maneira que o controlador que mantém esta característica é menos sensível a perturbações, em relação ao que não a mantém.

Deve-se também observar que o sensor discretiza (em amplitude) o deslocamento. A existência de uma zona morta associada à discretinação faz com que o controlador só tome uma ação quando os limites desta zona morta são atingidos. Assim, no caso dos "overshoots" obtidos nos resultados da implementação, provavelmente seriam diminuídos para uma resolução maior do codificador incremental.

7.4. Resumo

Este capítulo apresentou a análise dos resultados de projeto, bem como os resultados de implementação em relação as especificações de projeto.

8. CONCLUSÕES

A seguir são apresentadas conclusões deste trabalho, bem como sugestões para trabalhos posteriores.

Conforme apresentado na introdução deste trabalho, são normalmente implementados em microprocessadores controladores que exijam um esforço computacional baixo, notadamente controladores com características proporcional, proporcional-integral, proporcional-integral-derivativa. Neste trabalho apresentou-se a implementação de controladores mais sofisticados. Com simplificações na estrutura dos controladores, bem como com a utilização de técnicas de determinação de comprimento de palavra de variáveis e coeficientes, pôde-se fazer a implementação em microprocessadores de oito bits, para o controle de uma classe de servomotores.

A aplicabilidade dos controladores desenvolvidos neste trabalho, a nível industrial (com sua utilização para servomotores mais rápidos) depende da utilização de: ou implementação em hardware das multiplicações, ou a utilização de microprocessadores com capacidade de processamento maior (por exemplo, 16 bits). Tais sugestões seriam aplicáveis em casos onde fosse justificado o custo adicional, ou em casos onde mais de um servomotor devesse ser controlado, o que seria feito paralelamente pelo mesmo sistema computacional.

Como sugestões para trabalhos posteriores, relacionados com o apresentado neste, aparecem:

. Estudo comparativo entre os controladores aqui implementados e os controladores P, PI, PID, com análise de sensibilidade a variações de parâmetros do sistema e outras perturbações. Neste caso usar-se-iam parâmetros obtidos através de identificação.

. Implementação dos controladores aqui desenvolvidos para outros processos.

. Extensão do estudo comparativo a outros processos.

. Extensão da análise do comprimento de palavra de coeficientes e variáveis a processos com polos no círculo unitário.

. Comparação das técnicas de determinação do comprimento de palavra com outras técnicas existentes, e com a simulação das respostas para diversos comprimentos de palavra.

. Extensão da análise do comprimento de palavra de coeficientes e variáveis para a representação em ponto flutuante.

. Estudo das condições de aplicabilidade da aproximação dinâmica para os controladores ótimos.

9. BIBLIOGRAFIA

1. ANDERSON, B.D.O. & MOORE, J.B., *Linear Optimal Control*. Englewood Cliffs, N.J., Prentice-Hall, 1971.

2. AHMED, M.E. & BELANGER, P.R.: Scalling and Roundoff in Fixed-Point Implementation of Control Algorithms. *IEEE Trans. on Industrial Electronics*, Vol. IE-31, nº 3, 228 - 234, Aug. 1984a.

3. AHMED, M.E. & BELANGER, P.R.: Limit Cycles in Fixed-Point Implementation of Control Algorithms. *IEEE Trans. on Industrial Electronics*, Vol.IE-31, nº 3, 235-242, Aug 1984b.

4. AVENHAUS, E.: On the Design of Digital Filters with Coefficients of Limited Word Length. *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-20, nº 3, 206-212, Aug. 1972.

5. BARROS, P.R., *The Design of a Position Digital Controller Based on the Theory of Variable Structure Systems for the Programmable Systems*. Eindhoven, Holland, P.I.I. Report nº 926, Dec., 1981.

6. BELLMAN, R.E., *Dynamic Programming*: Princeton, Princeton University Press, 1957.

7. BERTRAM, J.E.: The Effect of Quantization in Sampled - Feedback Systems. *Trans. AIEE*, Vol. 77, Part 2 , 177 - 182, Sep. 1958.

8. CLAASEN, T. MECKLENBRÄUKER, W.F.G & PEEK, J.B.H.: Frequency Domain Criteria for the Abscence of Zero-Input Limit Cycles in Nonlinear Discrete-Time Systems, with Applications to Digital Filters. *IEEE Trans. on Circuits and Systems*, Vol. CAS-22, nº 3, 232-239, Mar. 1975.

9. CHIU, K.C., CORRIPIO, A.B. & SMITH, C.L.: Digital Control Algorithms, Part I, II e III. *Instruments and Control Systems*, Oct. 1973a., Nov. 1973b, Dec. 1973c.

10. CROCHIERE, Ronald E.: A New Statistical Approach to the Coefficient Word Length Problem for Digital Filters. *IEEE Transactions on Circuits and Systems*, Vol. CAS-22, nº 3, 190-196, Mar. 1975.

11. CROCHIERE, Ronald E. & OPPENHEIM, Alan V.: Analysis of Linear Digital Networks. *Proceedings of the IEEE*, Vol. 63, nº 4, 581-595, Apr. 1975.

12. D'AZZO, John J. & HOUPIS, Constantine H., *Linear Control System Analysis and Design;* Kogakusha, McGraw-Hill Kogakusha, 1975. 635p.

13. DORATO, P. & LEVIS, A.H.: Optimal Linear Regulators: The Discrete-Time Case. *IEEE Trans. on Automatic Control*, Vol. AC-16, nº 6, 613-620, Dec. 1971.

14. DORF, Richard C., *Modern Control Systems*; Massachusetts, Addison-Wesley Publishing Company, 1974. 405 p.

15. DUNCAN, F.G., *Microprocessor Programming & Software Development*, London, Prentice-Hall International, 1979. 320 p.

16. FAIRCHILD, *Linear Integrated Circuits Manual*; USA.

17. FAM, A.T.: Word Length and Memory Requirements of the Integer Parts of Some Digital Control Parameters. *IEEE Trans. on Automatic Control*, Vol. AC-27, nº 2, 496-498, apr. 1982.

18. FRANKLIN, Gene F. & POWELL, J. David, *Digital Control of Dynamic Systems*; Massachusetts, Addison-Wesley Publishing Company, 1980. 325 p.

19. GAUEN, K.; Designing a DC Servo Position Control Using a Microcomputer. *Control Engineering*, Vol. 30, nº 7, 80-83, July 1983.

20. INTEL, *MCS-85 User's Manual*; Oregon, Intel Corporation, 1978.

21. ISERMANN, Rolf, *Digital Control Systems*; Springer-Verlag, 1981. 566 p.

22. JACKSON, Leland B.: On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters. *The Bell System Technical Journal*, Vol. 49, nº 2, 159-184, Feb. 1970.

23. JING-PING, J. & MARLEAU, R.S.: Digitally Controlled DC Drive Motors. *IEEE Transactions on Industry Applications*, Vol. IA-18, nº 6, 728-735, Nov. Dec. 1982.

24. JOHNSON, C.D.: Accommodation of External Disturbances in Linear Regulator and Servomechanism Problems. *IEEE Trans. on Automatic Control*, Vol. AC-16, nº 6, 535-644, Dec 1971.

25. KALMAN, R. & KOEPCKE, R.V.: Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes. *Trans. ASME*, nº 80, pp. 1820-1826, 1958.

26. KUO, Benjamin C., *Digital Control Systems*; USA, Holt - Saunders International Editions, 1980. 730 p.

27. KWAKERNAAK, H. & SIVAN, R., *Linear Optimal Control Systems*, New York, Wiley-Interscience, 1972, 575p.

28. LABORATÓRIO DE SISTEMAS E MICROPROCESSADORES: MIC 85 - Microcomputador Baseado na CPU 8085 Compatível com o Barramento FECBAR. *Relatório Interno LSM/FEC/UNICAMP.*

29. L.S.M.: Tópicos 83 - Interface Terminal de Vídeo Monocromático. *Relatório Interno LSM/FEC/UNICAMP.*

30. L.S.M.: Memória MK2141 V.2. *Relatório Interno LSM/FEC/UNICAMP.*

31. L.S.M.: Timer 8253. *Relatório Interno LSM/FEC/UNICAMP.*

32. L.S.M.: FECBAR. *Relatório Interno LSM/FEC/UNICAMP.*

33. LOPES, A.M., MURRILL, P.W. & SMITH, C.L.: Tuning PI and PID Digital Controllers. *Instruments and Control Systems.* Vol. 42, 89-95, Feb. 1969.

34. LUENBERGER ,D.G.: Observers for Multivariable Systems . *IEEE Trans. on Automatic Control*, Vol. AC-16, nº 6, 596-602, Dec. 1971.

35. LUENBERGER, D.G.: Observers for Multivariable Systems. *IEEE Trans. on Automatic Control*, Vol. AC-11, nº 2, 190-197, Apr. 1966.

36. LUENBERGER, D.G.: An Introduction to Observer. *IEEE Trans. on Automatic Control*,. Vol. AC-16, nº 6, 596-602, Dec 1971.

37. MORONEY, Paul & WILLSKY, Alan S. & HOUPT, Paul K.: The Digital Implementation of Control Compensators: The Coefficient Wordlength Issue. *IEEE Transactions on Automatic Control*, Vol. AC-25, nº 4, 621-630, Aug. 1980.

38. MORONEY, P., WILLSKY, A.S. & HOUPT, P.K.: Roudoff Noise and Scaling in the Digital Implementation of Control Compensators. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, nº 6, 1464-1477, Dec. 1983.

39. OPPENHEIM, Alan V. & WEINSTEIN, Clifford J.: Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform. *Proceedings of the IEEE*, Vol. 60, nº 8, 957-976, Aug. 1972.

40. OPPENHEIM, Alan V. & SCHAFER, Ronald W., *Digital Signal Processing*; Englewood Cliffs, New Jersey, Prentice-Hall,

Inc., 1975, 585p.

41. ORTEGA, R.: Experimental Evaluation of Four Microprocessor-Based Advanced Control Algorithms. *Microprocessing and Microprogramming*, Vol. 10, 229-245, 1982.

42. PAPOULIS, Athanasius, *Probability, Random Variables, and Stochastic Processes*; New York, McGraw-Hill Book Company, 1965.

43. RABINER, Lawrence R. & GOLD, Bernard; *Theory and Application of Digital Signal Processing*; Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1975. 762p.

44. SIGNETICS, *Signetics Integrated Circuits-Logic-TTL*; Sunnyvale, California, 1978.

45. SIGNETICS, *Signetics Integrated Circuits-Analogue Circuits*; Sunnyvale, California, 1978.

46. SLAUGHTER, J.B.: Quantization Errors in Digital Control Systems. *IEEE Trans. on Automatic Control*, Vol. AC-9, 70-74, Jan. 1964.

47. STOJIČ, M.R.: Design of the Microprocessor Based Digital

System For DC Motor Speed Control. *IEEE Trans. on Industrial Electronics*, Vol. IE-31, nº 3, 243-248, Aug. 1984.

48. TEXAS, *The Optoelectronics Data Book;* Texas, Texas Instruments Incorporated, 1978.

49. TSUCHIYA, T.: Improved Direct Digital Control Algorithm for Microprocessor Implementation. *IEEE Trans. on Automatic Control*, Vol. AC-27, nº 2, 295-306, Apr. 1982.

APÊNDICE A1 - REGULADOR ÓTIMO PARA SISTEMAS DIGITAIS

APÊNDICE A1. REGULADOR ÓTIMO PARA SISTEMAS DIGITAIS

Este apêndice é uma transcrição do apresentado por Isermann (1981 p. 135a 141).

É assumido que a equação de estados do processo

$$\underline{x}(k+1) = A\underline{x}(k) + B\underline{u}(k) \qquad (A1.1)$$

com matrizes de parâmetros constantes A e B é dada, junto com uma condição inicial $\underline{x}(0)$. É assumido inicialmente que todas variáveis de estado $\underline{x}(k)$ são medíveis exatamente.

Um regulador tem que ser determinado de maneira a gerar o vetor de variáveis manipuladas $\underline{u}(k)$ a partir do vetor de variáveis de estados $\underline{x}(k)$ de modo que o sistema total é controlado até o estado final $\underline{x}(N) \approx 0$ e que o critério de desempenho quadrático

$$I = \underline{x}^T(N)\, S\, \underline{x}(N) + \sum_{k=0}^{N-1} [\, \underline{x}^T(k)\; Q\; \underline{x}(k) + \underline{u}^T(k)\; R\; \underline{u}(k)] \qquad (A1.2)$$

é minimizado . Aqui

S é simétrica e semi-dèfinida positiva,

Q é simétrica e semi-definida positiva,

R é simetrica e definida positiva,

isto é,

$$\underline{x}^T \; S \; \underline{x} \geqslant 0 \quad , \quad \underline{x}^T \; Q \; \underline{x} \geqslant 0 \quad e \quad \underline{u}^T \; R \; \underline{u} > 0$$

Estas condições nas matrizes de ponderação S, Q e R resultam das condições para existência do ótimo de I, e pode ser discutida como segue. Soluções consistentes no sentido de engenharia de controle podem ser obtidas se todos os ter mos têm o mesmo sinal, por exemplo, um sinal positivo. Assim, todas matrizes devem ser pelo menos semidefinida positiva . Se S = $\underline{0}$, isto é o estado final $\underline{x}(N)$ não é ponderado, mas Q=0, isto é todos estados $\underline{x}(0)$, ... $\underline{x}(N-1)$ são ponderados , um ótimo consistente também existe. Isto significa que se Q é definida positiva S pode ser semidefinida positiva. O ca so contrário também é válido. Deve-se, no entanto, excluir o caso onde S=$\underline{0}$ e Q= $\underline{0}$, pois então os estados x(k) não seriam ponderados e somente a variável manipulada seria ponderada por R$\neq$0, o que não tem sentido. R deve ser definida positiva para reguladores de estado contínuos, pois R^{-1} é envolvida na lei de controle. Para reguladores de estado discretos no tempo, no entanto esta exigência pode ser relaxada, como se rá descrito posteriormente.

Como somente o caso onde $\underline{x}(N) \approx 0$ é considerado aqui , S=Q é escolhida. Neste caso, Q deve ser também definida positiva. Note que neste problema a influência das variáveis de referência e perturbações externas é ignorado, e as variáveis de saída

$$\underline{y}(k) = C\ \underline{x}(k) \qquad \text{(A1.3)}$$

não são realimentadas. No lugar disso, nós consideramos a estabilização e modificação do comportamento próprio do processo através de realimentação de estado. Se a variável manipulada ótima $\underline{u}(k)$ é encontrada, então

$$\min I = \min_{\underline{u}(k)} \left\{ \underline{x}^T(N) Q\ \underline{x}(k) + \sum_{k=0}^{N+1} [\ \underline{x}^T(k)\ Q\ \underline{x}(k) + \underline{u}^T(k)\ R\ \underline{u}(k)] \right\}$$

$$k = 0,\ 1,\ 2,\ \ldots,\ N-1 \qquad \text{(A1.4)}$$

O cálculo da variável manipulada ótima é um problema de otimização dinâmica que pode ser resolvido, pelo cálculo variacional, aplicando o princípio do máximo de Pontryagin ou princípio de otimização de Bellman- (Bellman (1957)). A solução apresentada abaixo foi dada por Kalman & Koepcke (1958) e usa o princípio de otimalidade.

Comentários

a) De acordo com o princípio de otimalidade de Bellman cada elemento final de uma trajetória é também ótima. Isto significa que se o ponto final é conhecida, pode-se determinar a trajetória ótima na direção do ponto final para trás.

b) Da equação de estados (A1.1), $\underline{u}(k)$ influência os estados futuros $\underline{x}(k+1)$, $\underline{x}(k+2)$, ... Assim, pode-se calcular o $\underline{u}(k)$ ótimo por cálculo no sentido contrário. Assim, a equação (A1.4) é reescrita como:

$$\min I = \min_{\substack{\underline{u}(k) \\ k=0,1,\ldots,N-2}} \left[\min_{\underline{u}(N-1)} \left\{ \underline{x}^T(N)\, Q\, \underline{x}(N) + \sum_{k=0}^{N-1} \left[\underline{x}^T(k)\, Q\, \underline{x}(k) + \underline{u}^T(k)\, R\, \underline{u}(k) \right] \right\} \right] \tag{A1.5}$$

Segue que

$$\min_{\underline{u}(N-1)} \{\ldots\} = \sum_{k=0}^{N-1} \underline{x}^T(k)\, Q\, \underline{x}(k) + \sum_{k=0}^{N-2} \underline{u}^T(k)\, R \underline{u}(k) + \underbrace{\min_{\underline{u}(N-1)} \left\{ \underline{x}^T(N)\, Q\, \underline{x}(N) + \underline{u}^T(N-1)\, R\, \underline{u}(N-1) \right\}}_{I_{N-1,N}} \tag{A1.6}$$

desde que os dois primeiros termos não são influenciados por $\underline{u}(N-1)$ e $I_{N-1,N}$ são os custos de $k = N-1$ a $k=N$ resultante de $\underline{u}(N-1)$. Se a equação de estados

$$\underline{x}(N) = A\ \underline{x}(N-1) + B\ \underline{u}(N-1)$$

ou

$$\underline{x}^T(N) = \underline{x}^T(N-1)\ A^T + \underline{u}^T(N-1)\ B^T \qquad (A1.7)$$

é considerada como uma condição a mais, segue da equação (A1.6) que

$$I_{N-1,N} = \min_{\underline{u}(N-1)} \{\underline{x}^T(N)\ Q\ \underline{x}(N) + \underline{u}^T(N-1)\ R\ \underline{u}(N-1)\} =$$

$$= \min_{\underline{u}(N-1)} \{\underline{x}^T(N-1)\ A^T\ Q\ A\ \underline{x}(N-1) + 2\ \underline{x}^T(N-1) A^T\ Q\ B\ \underline{u}(N-1)$$

$$+ \underline{u}^T(N-1)\ B^T\ Q\ B\ \underline{u}(N-1) + \underline{u}^T(N-1)\ \underline{R}\ \underline{u}(N-1)\} =$$

$$= \underline{x}^T(N-1)\ A^T\ Q\ A\ \underline{x}(N-1) + \min_{\underline{u}(N+1)} \{2\underline{x}^T(N-1)\ A^T\ Q\ B\ \underline{u}(N-1)$$

$$+ \underline{u}^T(N-1)\ (B^T\ Q\ B + R)\ \underline{u}(N-1)\}\ . \qquad (A1.8)$$

Para minimizar a equação (A1.8) as seguintes relações são válidas

$$\min_{\underline{u}(N-1)} \{\dots\} = \frac{\partial}{\partial u(N-1)} \{\dots\} = \underline{0}$$

$$\frac{\partial^2}{\partial\ u(N-1)^2} \{\dots\} > \underline{0} \qquad (A1.9)$$

Assim, usando as regras de derivação de vetores e matrizes.

$$\frac{\partial}{\partial \underline{u}(N-1)} \{\ldots\} = 2\, B^T Q\, A\, \underline{x}(N-1) + 2(B^T Q\, B + R)$$

$$\underline{u}(N-1) = 0$$

e

$$u^o(N-1) = -(B^T Q\, B + R)^{-1} B^T Q\, A\, \underline{x}(N-1)$$

$$= -\, \underline{K}(N-1)\, \underline{x}(N-1). \tag{A1.10}$$

Aqui

$$\underline{K}(N-1) = (B^T Q\, B + R)^{-1} B^T Q\, A \tag{A1.11}$$

e

$$\frac{\partial^2 \{\cdots\}}{\partial \underline{u}(N-1)^2} = 2(B^T Q\, B + R) > 0 \tag{A1.12}$$

Os custos $I_{N-1,N}$ resultantes de u(N-1) podem então ser formulados como uma função da condição inicial x(N-1) para aquele estágio:

$$I_{N-1,N} = \underline{x}^T(N-1)\, A^T Q\, A\, \underline{x}(N-1) - 2\, \underline{x}^T(N-1)\, .$$

$$A^T Q\, B\, (B^T Q\, B + R)^{-1} . B^T Q\, A\, \underline{x}(N-1) +$$

$$\underline{x}^T(N-1)\, A^T Q\, B\, (B^T QB + R)^{-1} B^T Q\, A\, \underline{x}(N-1)$$

$$= \underline{x}^T(N-1) \left[A^T Q A - A^T Q B (B^T Q B + R)^{-1} B^T Q A \right] \underline{x}(N-1)$$

$$= \underline{x}^T(N-1) \left[A^T QA - \underline{K}^T(N-1)(B^T QB + R)\underline{K}(N-1) \right] \underline{x}(N-1)$$

$$= \underline{x}^T(N-1)\, P_{N-1,\,N}\, \underline{x}(N-1). \qquad (A1.13)$$

Aqui

$$P_{N-1,N} = A^T Q [I - B (B^T Q B + R)^{-1} B^T Q] A \quad ,$$

$$= A^T Q A - \underline{K}^T(N-1) (B^T Q B + R) \underline{K}(N-1) \qquad (A1.14)$$

I, ou min I, de acordo com as equações (A1.5) (A1.6) podem ser expressas como função de $\underline{x}(k)$, K=0,..., N-1 e $\underline{u}(k)$, K=0,..., N-2. Assim os termos desconhecidos x(N) e $\underline{u}(N-1)$ podem ser eliminados. Para realizar esta eliminação, inicialmente $I_{N-1,N}$ da equação (A1.13) é substituída na equação (A1.6), resultanto em

$$\min_{\underline{u}(N-1)} \{\underline{x}^T(N)\, Q\, \underline{x}(N) + \sum_{k=0}^{N-1} [\underline{x}^T(K)\, Q\, \underline{x}(K) + \underline{u}^T(k)\, R\, \underline{u}(k)]\} = \sum_{k=0}^{N-1} \underline{x}^T(k)\, Q\, \underline{x}(k) + \sum_{k=0}^{N-2} \underline{u}^T(k)\, R\, \underline{u}(k) + \underline{x}^T(N-1)\, P_{N-1,N}\, \underline{x}(N-1) = \underbrace{\sum_{k=0}^{N-2} [\underline{x}^T(k)\, Q\, \underline{x}(k) + \underline{u}^T(k)\, R\, \underline{u}(k)] + \underline{x}^T(N-1)\, (P_{N-1,N} + Q)\, \underline{x}(N-1)}_{I_{N-1}} \qquad (A1.15)$$

A abreviação

$$P_{N-1} = P_{N-1,N} + Q \tag{A1.16}$$

é introduzida de modo que na equação (A1.15)

$$I_{N-1} = I_{N-1,N} + \underline{x}^T(N-1)\ Q\ \underline{x}(N-1) = \underline{x}^T(N-1)\ (P_{N+,N}+Q)$$

$$\underline{x}(N-1) = \underline{x}^T(N-1)\ P_{N-1}\ \underline{x}(N-1). \tag{A1.17}$$

Nesta abreviação o custo do último passo e a avaliação do desvio inicial correspondente $\underline{x}(N-1)$ são incluídos. (Esta compressão permite uma formulação mais simples das equações que seguem). Se a equação (A1.16) é introduzida na equação (A1.5), segue que

$$\min I = \min_{\substack{\underline{u}(k)\\ k=0,\dots,N-3}} \left[\min_{\underline{u}(N-2)} \left\{ \sum_{k=0}^{N-2} [\,\underline{x}^T(k)\ Q\ \underline{x}(k) + \underline{u}^T(k)\, R\, \underline{u}(k)] + \underline{x}^T(N-1)\ P_{N-1}\ \underline{x}(N-1) \right\} \right]. \tag{A1.18}$$

Ao invés do $\min\limits_{\underline{u}(N-1)}$ agora aparece $\min\limits_{\underline{u}(N-2)}$ como o ótimo $\underline{u}(N-1)$ e o estado resultante $\underline{x}(N)$ foi calculado e substituído. Para o termo $\min\limits_{\underline{u}(N-2)}$ ... obtêm-se por analogia à equação (A1.6).

$$\min_{\underline{u}(N-2)} \{\dots\} = \sum_{k=0}^{N-2} \underline{x}^T(K)\ Q\ \underline{x}(k) + \sum_{k=0}^{N-3} \underline{u}^T(K)\ R\ \underline{u}(K) +$$

$$\underbrace{+ \min_{\underline{u}(N-2)} \left\{ \underline{u}^T(N-2)\ R\ \underline{u}(N-2) + \underline{x}^T(N-1)\ P_{N-1}\underline{x}(N-1) \right\}}_{I_{N-2,N}}. \tag{Al.19}$$

$I_{N-2,N}$ descreve o custo resultantes dos dois últimos está - gios

$$I_{N-2,N} = \underline{u}^T(N-2)\ R\ \underline{u}(N-2) + \underline{x}^T(N-1)\ Q\ \underline{x}(N-1) + I_{N-1,N} \tag{Al.20}$$

Se agora a equação de estados é considerada novamente

$$\underline{x}(N-1) = A\ \underline{x}(N-2) + B\ \underline{u}(N-2)$$

segue que

$$\begin{aligned} I_{N-2,N} = \min_{\underline{u}(N-2)} \Big\{ & \underline{u}^T(N-2)(R+B^T P_{N-1} B)\ \underline{u}(N-2) + \\ & + 2\ \underline{u}^T(N-2)\ B^T P_{N-1} A\ \underline{x}(N-2) + \\ & \underline{x}^T(N-2)\ A^T P_{N-1} A\ \underline{x}(N-2) = \\ = \underline{x}^T(N-2)\ A^T P_{N-1} & A\ \underline{x}(N-2) + \min_{\underline{u}(N-2)} \Big\{ \underline{u}^T(N-2) \\ (R + B^T P_{N-1} B)\ \underline{u}(N-2) & + 2\ \underline{u}^T(N-2) B^T P_{N-1} A\underline{x}(N-2) \Big\} \end{aligned} \tag{Al.21}$$

Isto resulta por analogia com a equação (Al.10), em

$$\underline{u}^o(N-2) = -(R + B^T P_{N-1} B)^{-1}\ B^T\ P_{N-1}\ A\ \underline{x}(N-2)$$

$$= - K_{N-2}\ \underline{x}(N-2) \qquad (A1.22)$$

Assim, o regulador $\underline{K}_{N-2}$ torna-se

$$\underline{K}_{N-2} = (R + B^T P_{N-1} B)^{-1} B^T P_{N-1} A. \qquad (A1.23)$$

Portanto, o custo mínimo $I_{N-2,N}$ para os dois últimos estágios torna-se, usando a equação (A1.21):

$$\begin{aligned}
I_{N-2,N} &= \underline{x}^T(N-2)\ A^T P_{N-1} A\ \underline{x}(N-2) \\
&+ \underline{x}^T(N-2)\ A^T P_{N-1} B(R+B^TP_{N-1}B)^{-1} B^TP_{N-1}A\ \underline{x}(N-2) \\
&- 2\ \underline{x}^T(N-2)\ A^TP_{N-1} B(R+B^TP_{N-1}B)^{-1}B^TP_{N-1}A\ \underline{x}(N-2) \\
&= \underline{x}^T(N-2)\ [\ A^TP_{N-1} A-A^TP_{N-1} B(R+B^TP_{N-1}B)^{-1} \\
&\quad B^T P_{N-1} A\ ]\ \underline{x}(N-2) \\
&= \underline{x}^T(N-2)\ [\ A^T P_{N-1} A - \underline{K}^T_{N-2} (R + B^T P_{N-1}B)\ \underline{K}_{N-2}] \\
&\quad \underline{x}(N-2) \\
&= \underline{x}^T(N-2)\ P_{N-2,N}\ \underline{x}(N-2) \qquad (A1.24)
\end{aligned}$$

com

$$\begin{aligned}
P_{N-2,N} &= A^T P_{N-1} [I - B(R+B^TP_{N-1}B)^{-1} B^T P_{N-1}] A \\
&= A^T P_{N-1} A - \underline{K}^T_{N-2} (R+B^T P_{N-1} B)\ \underline{K}_{N-2} \qquad (A1.25)
\end{aligned}$$

Agora, o mínimo de I com respeito a u(N-2) pode ser formulado de acordo com a equação (A1.19)

$$\min_{\underline{u}(N-2)} \quad I = \sum_{k=0}^{N=2} \underline{x}^T(k)\; Q\; \underline{x}(k) + \sum_{k=0}^{N-3} \underline{u}^T(k)\; R\; \underline{u}(k) +$$

$$+ \underline{x}^T(N-2)\; P_{N-2,N}\; \underline{x}(N-2) =$$

$$= \sum_{k=0}^{N-3} [\, \underline{x}^T(k)\; Q\; \underline{x}(k) + \underline{u}^T(k)\; R\; \underline{u}(k)]$$

$$+ \underbrace{\underline{x}^T(N-2)\; (P_{N-2,N} + Q)\; \underline{x}(N-2)}_{I_{N-2}} \qquad \text{(A1.26)}$$

Se a abreviação

$$P_{N-2} = P_{N-2,N} + Q \qquad \text{(A1.27)}$$

é introduzida novamente, os custos dos dois últimos estágios incluindo desvio inicial $\underline{x}(N-2)$ resultam em

$$I_{N-2} = I_{N-2,N} + \underline{x}^T(N-2)\; Q\; \underline{x}(N-2) = \underline{x}^T(N-2)\,(P_{N-2,N}+Q)\,\underline{x}(N-2)$$

$$= \underline{x}^T(N-2)\; \underline{P}_{N-2}\; \underline{x}(N-2). \qquad \text{(A1.28)}$$

Considerando a equação de estados, I pode agora ser expresso como uma função de $\underline{x}(k)$ e $\underline{u}(k)$ com $k=0,\ldots, N-3$. Então $u^o(N-3)$ pode ser determinado, etc.

Em termos gerais, obtem-se um regulador de estados linear variante no tempo

$$\underline{u}^o(N-j) = -\, \underline{K}_{N-j}\; \underline{x}(N-j) \qquad j=1,2,\ldots,N \qquad \text{(A1.29)}$$

que é uma realimentação negativa, de ação proporcional, a entrada através da matriz $\underline{K}_{N-j}$.

Seus parâmetros são obtidos das equações recursivas

$$K_{N-j} = (R + B^T P_{N-j+1} B)^{-1} B^T P_{N-j+1} A \qquad \text{(A1.30)}$$

$$\begin{aligned} P_{N-j} &= Q + A^T P_{N-j+1} A - \underline{K}_{N-j} (R + B^T P_{N-j+1} B) \underline{K}_{N-j} \\ &= Q - \underline{K}_{N-j} R \underline{K}_{N-j} + [A - B\underline{K}_{N-j}]^T P_{N-j+1} [A + BK_{N-j}] \\ &= Q + A^T P_{N-j+1} [I - B(R + B^T P_{N-j+1} B)^{-1} B^T P_{N-j+1}] A \end{aligned}$$

(A1.31)

com $P_N = Q$ como matriz inicial. A última equação é a equação matricial à diferenças de Riccati. Para o valor do critério de desempenho da equação (A1.2) nos termos:

$$\min_{\underline{u}(k)} I = I_0 = \underline{x}^T(0) P_0 x(0) \qquad \text{(A1.32)}$$

com $k = 0,1,\ldots,N-1$.

Nas equações (A1.30) e (A1.31) se o termo $B^T P_{N-j+1} B > 0$, então a exigência que R seja definida positiva pode ser relaxada nos termos onde há inversas de matrizes.

APÊNDICE A2 - CONTROLADORES ÓTIMOS

```
PROGRAM CONTROLADORESOTIMOS;
(*****************************************************************

PROGRAMA PARA O CALCULO DOS CONTROLADORES OTIMOS:
      -COM OBSERVADOR COM REALIMENTACAO
      -COM OBSERVADOR SEM REALIMENTACAO
      -DINAMICO

AUTOR:PERICLES REZENDE BARROS

TESE DE MESTRADO EM ENGENHARIA ELETRICA APRESENTADA NA
        UNIVERSIDADE FEDERAL DA PARAIBA
        CENTRO DE CIENCIAS E TECNOLOGIA
        DEPARTAMENTO DE ENGENHARIA ELETRICA
        CAMPINA GRANDE - PARAIBA
                                      DESENVOLVIDA NA
        UNIVERSIDADE ESTADUAL DE CAMPINAS
        FACULDADE DE ENGENHARIA DE CAMPINAS
        DEPARTAMENTO DE ENGENHARIA ELETRICA
        CAMPINAS - SAO PAULO
                                                   1984

*****************************************************************)

CONST
        ORDEM=3;
        N=3;
        M=80;
        CONVERG=1.0E-10;

TYPE
        COMP=(R,I);
        ZEROPOLO=(ZE,PO);
        COMPLEXO=ARRAY[COMP] OF REAL;
        VETOR=ARRAY[1..N] OF REAL;
        XVETOR=ARRAY[1..5] OF REAL;
        MATRIZ=ARRAY[1..N,1..N] OF REAL;
        CVETOR=ARRAY[1..N] OF COMPLEXO;
        CXVETOR=ARRAY[1..5] OF COMPLEXO;
        CMATRIZ=ARRAY[1..N,1..N] OF COMPLEXO;
        RESPOSTA=ARRAY[0..M] OF REAL;
        CONVS=ARRAY[1..N,1..N] OF BOOLEAN;
VAR
        FINAL,CONT:(SIM,NAO);
        IFS,FIM:BOOLEAN;
        RES,IMG,RE1,IM1,CKA,CKB,PERIODO,RAIZ1,P,IZ2,GANHODIRETO,J1,J2:REAL;
        PASSOCORI:INTEGER;
        VTRENTR,VTRGANHO,VTRSIMU:VETOR;
        MIZSIST,MIZSIMU:MATRIZ;
        VTRREALIMENT,VTRDIRETO,REALIM:CVETOR;
        ENTRADA,CONTROLADOR,SAIDA:RESPOSTA;

        D,VTROBSERVADOR:VETOR;
        BK,AT,AMENOSBK,HC,AMENOSHC:MATRIZ;
        AUTOVALI1,AUTOVALI2,AUTOVALI3,AUX1:COMPLEXO;
        J,K,L:INTEGER;
        ERPS,VTRJORDAN,VTRDOJORDAN,CVTRGANHO,CGANHOESTADO:CVETOR;
        MIZAUTOVETORES,INVERSAP,AMBKVZP,MIZJORDAN,CAMENOSBK:CMATRIZ;

        KD,DEN1,DEN21,DEN22,DEN31,DEN32,NUM1,NUM21,NUM22,NUM31,NUM32:REAL;
```

```
        ALFAS,BETAS,MS,NS,GAMA,ZIS:VETOR;
        INVAMBK:MATRIZ;
        ZERO:ZERODOBU;
        TIPOCONTROL:(OTIMOREA,OTIMODIR,DINAMICO);

PROCEDURE LERPARAMETROS(VAR P:INTEGER;
                        VAR R,L,B,J,KA,KB:REAL);


BEGIN
WRITE(TTY,'RESISTENCIA = '); BREAK;
READ(TTY,R);
WRITE(TTY,'INDUTANCIA = ');BREAK;
READ(TTY,L);
WRITE(TTY,'FRICCAO MOTOR-CARGA = ');BREAK;
READ(TTY,B);
WRITE(TTY,'INERCIA MOTOR-CARGA = ');BREAK;
READ(TTY,J);
WRITE(TTY,'CONSTANTES ELETRICA E MECANICA =');BREAK;
READ(TTY,KA);
KB:=KA;
WRITE(TTY,'NUMERO DE PULSOS POR ROTACAO DO CODIFICADOR INCREMENTAL =');BREAK;
READ(TTY,P)
END;


FUNCTION ALFA(R,L,B,J:REAL): REAL;

BEGIN
ALFA:= (R/L)+(B/J)
END;


FUNCTION BETA(R,L,B,J,KA,KB:REAL): REAL;

BEGIN
BETA:=(R*B+KA*KB)/(L*J)
END;(* BETA *)

PROCEDURE LERPERIODO(VAR T:REAL);
BEGIN
WRITE(TTY,'PERIODO DE AMOSTRAGEM = ');BREAK;
READ(TTY,T);
WRITELN(TTY)
END;(* LER PERIODO DE AMOSTRAGEM          *)

PROCEDURE MUDARPERIODO(VAR T:REAL);

VAR
        CH:(N,S);

BEGIN
WRITELN(TTY);
WRITE(TTY,'PERIODO DE AMOSTRAGEM = ',T,'  ? (S OU N) =  ');BREAK;
READ(TTY,CH);
IF CH=N THEN
        BEGIN
        WRITE(TTY,'T = ');BREAK;
        READ(TTY,T)
        END;
```

```
WRITELN(TTY,'PERIODO DE AMOSTRAGEM = ',T);BREAK
END;(* MUDAR O PERIODO *)


PROCEDURE MUDARPARAMETROS(VAR P:INTEGER;
                          VAR R,L,B,J,KA,KB:REAL);

VAR
        CH:(S,N);

BEGIN
WRITE(TTY,'DESEJA CONFERIR OU MUDAR ALGUM PARAMETRO ? (S OU N)');BREAK;
READ(TTY,CH);
WHILE CH=S DO
        BEGIN
        WRITELN(TTY,'CASO DESEJE MUDAR O PARAMETRO ESPECIFICADO');
        WRITELN(TTY,'ESCREVA N , CASO CONTRARIO ESCREVA S.');
        WRITE(TTY,'RESISTENCIA R = ',R,' ? ');BREAK;
        READ(TTY,CH);
        IF CH=N THEN
                BEGIN
                WRITE(TTY,'R = ');BREAK;
                READ(TTY,R)
                END;
        WRITELN(TTY,'R = ',R);BREAK;
        WRITE(TTY,'INDUTANCIA L = ',L,' ? ');BREAK;
        READ(TTY,CH);
        IF CH=N THEN
                BEGIN
                WRITE(TTY,'L = ');BREAK;
                READ(TTY,L)
                END;
        WRITELN(TTY,'L = ',L);BREAK;
        WRITE(TTY,'FRICCAO B = ',B,' ? ');BREAK;
        READ(TTY,CH);
        IF CH=N THEN
                BEGIN
                WRITE(TTY,'B = ');BREAK;
                READ(TTY,B)
                END;
        WRITELN(TTY,'B = ',B);BREAK;
        WRITE(TTY,'INERCIA = ',J,' ? ');BREAK;
        READ(TTY,CH);
        IF CH=N THEN
                BEGIN
                WRITE(TTY,'J = ');BREAK;
                READ(TTY,J)
                END;
        WRITELN(TTY,'J = ',J);BREAK;
        WRITE(TTY,'CONSTANTE = ',KA,' ? ');BREAK;
        READ(TTY,CH);
        IF CH=N THEN
                BEGIN
                WRITE(TTY,'KA = ');BREAK;
                READ(TTY,KA)
                END;
        KB:=KA;
        WRITELN(TTY,'KA,KB = ',KA);
        WRITE(TTY,'NUMERO DE PULSOS DO CODIFICADOR INCREMENTAL = ',P,' ? ');BREAK;
        READ(TTY,CH);
```

```
        IF CH=N THEN
                BEGIN
                WRITE(TTY,'P = ');BREAK;
                READ(TTY,P)
                END;
        WRITELN(TTY,'P = ',P);BREAK;
        WRITE(TTY,'AINDA DESEJA MUDAR ALGUM PARAMETRO ?(S OU N) ');BREAK;
        READLN(TTY,CH)
        END
END;(* MUDAR PARAMETROS *)


PROCEDURE RAIZESREAIS(R,L,B,J,KA,KB:REAL;
                      VAR RAIZ1,RAIZ2:REAL);

VAR
        ALF,BET,DELTA:REAL;

BEGIN
WRITELN(TTY);
ALF:=ALFA(R,L,B,J);
BET:=BETA(R,L,B,J,KA,KB);
DELTA:=SQR(ALF)-4*BET;
IF DELTA< 0.0 THEN
        BEGIN
        WRITELN(TTY,'RAIZES COMPLEXAS , ENCERRE SIMULACAO');BREAK
        END
ELSE
        IF DELTA=0.0 THEN
                BEGIN
                WRITELN(TTY,'RAIZES IGUAIS , ENCERRE SIMULACAO ');BREAK
                END

        ELSE
                BEGIN
                RAIZ1:=((-ALF)+SQRT(DELTA))/2.0;
                RAIZ2:=((-ALF)-SQRT(DELTA))/2.0
                END;
WRITELN(TTY,'POLO CONTINUO 1 =',RAIZ1,'POLO CONTINUO2= ',RAIZ2);BREAK
END;(* RAIZES REAIS *)


FUNCTION A22(R,L,R1,R2,E1,E2:REAL):REAL;

VAR
        AUX:REAL;

BEGIN
AUX:=R1-R2;
A22:=((R1+(R/L))*E1-(R2+(R/L))*E2)/AUX
END;(*A22*)

FUNCTION A32(L,KB,R1,R2,E1,E2:REAL):REAL;

VAR
        AUX:REAL;
```

```
A32:=-(KB*(E1-E2))/(AUX*L)
END;(*A32*)

FUNCTION A23(J,KA,R1,R2,E1,E2:REAL):REAL;

VAR
        AUX:REAL;
BEGIN
AUX:=R1-R2;
A23:=(KA*(E1-E2))/(AUX*J)
END;(*A23*)

FUNCTION A33(B,J,R1,R2,E1,E2:REAL):REAL;

VAR
        AUX:REAL;

BEGIN
AUX:=R1-R2;
A33:=((R1+(B/J))*E1-(R2+(B/J))*E2)/AUX
END;(*A33*)

FUNCTION B2(L,J,KA,BET,R1,R2,E1,E2:REAL):REAL;

VAR
        AUX:REAL;
BEGIN
AUX:=R1-R2;
B2:=(1/BET+E1/(R1*AUX)-E2/(R2*AUX))*KA/(L*J)
END;(*B2*)

FUNCTION B3(L,B,J,BET,R1,R2,E1,E2:REAL):REAL;

VAR
        AUX1,AUX2:REAL;

BEGIN
AUX1:=R1-R2;
AUX2:=B/J;
B3:=((AUX2/BET)+(E1*(R1+AUX2)/R1-E2*(R2+AUX2)/R2)/AUX1)/L
END;(*B3*)

PROCEDURE COEFICIENTES(P:INTEGER;
                       K,L,B,J,KA,KB,T,R1,R2:REAL;
                       VAR VC:VETOR;
                       VAR A:MATRIZ);

CONST
        PI=3.14159265;
VAR
        EXP1,EXP2,AUXI,BET:REAL;

BEGIN
BET:=R1*R2;
EXP1:=EXP(R1*T);
EXP2:=EXP(R2*T);
AUXI:=(P*T)/(2*PI);
A[2,2]:=A22(P,L,R1,R2,EXP1,EXP2);
A[1,2]:=AUXI*A[2,2];
A[3,2]:=A32(L,KB,R1,R2,EXP1,EXP2);
```

```
A[2,3]:=A23(J,KA,R1,R2,EXP1,EXP2);
A[1,3]:=AUXI*A[2,3];
A[1,1]:=1.0;
A[2,1]:=0.0;
A[3,1]:=0.0;
A[3,3]:=A33(B,J,R1,R2,EXP1,EXP2);
VB[2]:=B2(B,J,KA,BET,R1,R2,EXP1,EXP2);
VB[1]:=AUXI*VB[2];
VB[3]:=B3(B,H,J,BET,R1,R2,EXP1,EXP2)
END;(* COEFICIENTES DO SISTEMA DISCRETO *)

PROCEDURE EVETMATRIZ(B:VETOR; A:MATRIZ);

CONST
        ORDEM=3;

VAR
        I,K:1..N;

BEGIN
WRITELN(TTY);
WRITELN(TTY,'MATRIZ DO SISTEMA DISCRETO');
FOR I:=1 TO ORDEM DO
        BEGIN
        WRITE(TTY,'[ ');
        FOR K:=1 TO ORDEM DO
                WRITE(TTY,A[I,K]);
        WRITELN(TTY,' ]')
        END;
WRITELN(TTY);
WRITELN(TTY,'VETOR DO SISTEMA DISCRETO');
FOR I:=1 TO ORDEM DO
        WRITELN(TTY,'[ ',B[I],' ] ');BREAK
END;(* ESCREVE VETOR E MATRIZ DISCRETOS *)

PROCEDURE LERMATRIZ(ORD:INTEGER;
                    VAR A:MATRIZ);

VAR
        I,J:1..N;
        CH:(S,N);
BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                IF I=J THEN
                        A[I,I]:=1.0
                ELSE
                        A[I,J]:=0.0;
WRITELN(TTY,'MATRIZ DE PONDERACAO DOS ESTADOS');
WRITELN(TTY);
FOR I:=1 TO ORD DO
        BEGIN
        FOR J:=I TO ORD DO
                BEGIN
                IF J=I THEN
                        BEGIN
                        WRITE(TTY,'Q[',I:2,',',I:2,']= ',A[I,I],' ? (S OU N) = ');BREAK;
                        READ(TTY,CH);
```

```
                IF CH=N THEN
                        BEGIN
                        WRITE(TTY,'Q[',T:2,',',I:2,'] = ');BREAK;
                        READ(TTY,A[I,T])
                        END
                END
            ELSE
                BEGIN
                WRITE(TTY,'Q[',T:2,',',T:2,'] = Q[',J:2,',',I:2,'] = ',A[I,J],' ? (S OU N) = ');BREAK;
                READ(TTY,CH);
                IF CH=N THEN
                        BEGIN
                        WRITE(TTY,'Q[',T:2,',',J:2,'] = Q[',J:2,',',I:2,'] = ');BREAK;
                        READ(TTY,A[I,T]);
                        A[J,I]:=A[I,J]
                        END
                END
            END;
        WRITELN
        END
END;(* LER MATRIZ *)

PROCEDURE ESCMATRIZ(ORD:INTEGER;
                    CH:CHAR;
                    VAR A:MATRIZ);

VAR
        I,J:1..N;
BEGIN
WRITELN(TTY);
CASE CH OF
        'Q':WRITELN(TTY,'MATRIZ DE PONDERACAO DOS ESTADOS');
        'D':WRITELN(TTY,'MATRIZ DE DESEMPENHO')
END;
WRITELN(TTY);
FOR I:=1 TO ORD DO
        BEGIN
        WRITE(TTY,'[ ');
        FOR J:=1 TO ORD DO
                WRITE(TTY,A[I,J]);
        WRITELN(TTY,' ]')
        END;
WRITELN(TTY);BREAK
END;(* ESCREVER MATRIZ *)

PROCEDURE ESCVL(ORD:INTEGER;
                VAR A:VETOR);

VAR
        I:1..N;
BEGIN
WRITELN(TTY);
WRITE(TTY,'[ ');
FOR I:=1 TO ORD DO
        WRITE(TTY,A[I]);
WRITELN(TTY,' ]');BREAK;
WRITELN(TTY)
END;(* ESCREVE VETOR LINHA *)
```

```
PROCEDURE ESCVC(ORD:INTEGER;
                VAR A:VETOR);
VAR
        I:1..N;
BEGIN
WRITELN(TTY);
FOR I:=1 TO ORD DO
        WRITELN(TTY,'[ ',A[I],' ]');BREAK
END;(* ESCREVE VETOR COLUNA *)

PROCEDURE ZEREMATRIZ(ORD:INTEGER;
                VAR A :MATRIZ);

VAR
        I,J:1..N;
BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                A[I,J]:=0.0
END; (*ZERAR MATRIZ*)

PROCEDURE MIGUALM(ORD:INTEGER;
                B:MATRIZ;
                VAR A:MATRIZ);

VAR
        I,J:1..N;

BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                A[I,J]:=B[I,J]
END;(* IGUALAR MATRIZES *)

PROCEDURE MTZTRANSP(ORD:INTEGER;
                        B:MATRIZ;
                        VAR A:MATRIZ);

VAR
        I,J:1..N;
BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                A[I,J]:=B[J,I]
END;(* TRANSPOSTA DE UMA MATRIZ *)

PROCEDURE SOMARMM(ORD:INTEGER;
                B,C:MATRIZ;
                VAR A:MATRIZ);
VAR
        I,J:1..N;
BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                A[I,J]:=B[I,J]+C[I,J]
END;(* SOMAR MATRIZES *)

PROCEDURE SUBTRAIMM(ORD:INTEGER;
                B,C:MATRIZ;
                VAR A:MATRIZ);
```

UNICAMP
Centro de Computação

```
VAR
        I,J:1..N;

BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                A[I,J]:=B[I,J]-C[I,J]
END;(* SUBTRAIR MATRIZES *)

PROCEDURE MULTVETIN1(ORD:INTEGER;
                     B,C:VETOR;
                     VAR A:REAL);

VAR
        I:1..N;
        SOMA:REAL;

BEGIN
SOMA:=0.0;
FOR I:=1 TO ORD DO
        SOMA:=SOMA+B[I]*C[I];
A:=SOMA
END;(* MULTIPLICAR VETORES 1XN NX1 *)

PROCEDURE MULTVETN1N(ORD:INTEGER;
                     B,C:VETOR;
                     VAR A:MATRIZ);

VAR
        I,J:1..N;

BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                A[I,J]:=B[I]*C[J]
END;(* MULTIPLICAR VETORES NX1 1XN *)

PROCEDURE MULTVETCNST(ORD:INTEGER;
                      B:REAL;
                      C:VETOR;
                      VAR A:VETOR);

VAR
        I:1..N;

BEGIN
FOR I:=1 TO ORD DO
        A[I]:=B*C[I]
END;(* MULTIPLICAR VETOR POR CONSTANTE *)

PROCEDURE MULTVM1NN(ORD:INTEGER;
                    B:VETOR;
                    C:MATRIZ;
                    VAR A:VETOR);

VAR
        I,J:1..N;
        SOMA:REAL;
```

```
BEGIN
FOR I:=1 TO ORD DO
        BEGIN
        SOMA:=0.0;
        FOR J:=1 TO ORD DO
                SOMA:=SOMA+B[J]*C[J,I];
        A[I]:=SOMA
        END
END;(* MULTIPLICAR VETOR 1XN MATRIZ NXN *)

PROCEDURE MULTMTZNN(ORD:INTEGER;
                    B,C:MATRIZ;
                    VAR A:MATRIZ);

VAR
        I,J,K:1..N;
        SOMA:REAL;
BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                BEGIN
                SOMA:=0.0;
                FOR K:=1 TO ORD DO
                        SOMA:=SOMA+B[I,K]*C[K,J];
                A[I,J]:=SOMA
                END
END;(* MULTIPLICAR MATRIZES *)

PROCEDURE TESTCONV(ORD:INTEGER;
                   CV:REAL;
                   VAR CON:BOOLEAN;
                   VAR MG,MA:MATRIZ);

VAR
        I,J:1..N;
        CONVER:CONVE;
BEGIN
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                BEGIN
                IF (ABS(MG[I,J]-MA[I,J])<=ABS(CV*MA[I,J])) THEN
                        CONVER[I,J]:=TRUE
                ELSE
                        CONVER[I,J]:=FALSE;
                MA[I,J]:=MG[I,J]
                END;
CON:=TRUE;
FOR I:=1 TO ORD DO
        FOR J:=1 TO ORD DO
                CON:=CON AND CONVER[I,J]
END;(* TESTAR CONVERGENCIA E ATUALIZAR MATRIZ *)

PROCEDURE CALCULEN(ORD:INTEGER;
                   PR:REAL;VE:VETOR;
                        MO,MF:MATRIZ; VAR PP:REAL;
                        VAR VG:VETOR; VAR MA:MATRIZ);

VAR
        VR1:VETOR;
        PP1:REAL;
```

```
BEGIN
MULTVETMAN(ORD,VE,MD,VR1);
MULTVETMAN(ORD,VR1,VE,PP1);
PP:=PP1+PR;
MULTMTZMAN(ORD,MD,MA,MA);
MULTVMTMAN(ORD,VE,MA,VR1);
MULTVETCNST(ORD,1/PP,VR1,VG)
END;(* CALCULA VETOR DE GANHO *)


PROCEDURE CALCULEP(ORD:INTEGER;PP:REAL;VG:VETOR;
                   MA,AT,D:MATRIZ;VAR MD:MATRIZ);

VAR
        VR1:VETOR;
        MR1,MR2,MR3:MATRIZ;

BEGIN
MULTVETCNST(ORD,PP,VG,VR1);
MULTVETMIN(ORD,VG,VR1,MR1);
MULTMTZMN(ORD,AT,MA,MR2);
SUBTRAIMN(ORD,MR2,MR1,MR3);
SOMARMN(ORD,D,MR3,MD)
END; (* CALCULAR MATRIZ DE DESEMPENHO *)

PROCEDURE RICCATI(TIPO:CHAR;
                  VTRENTR:VETOR;
                  MTZSISTEM:MATRIZ;
                  VAR VTRGANHO:VETOR);

CONST
        ORDEM=3;

TYPE
        CONVE=ARRAY[1..N,1..N] OF BOOLEAN;
VAR
        CONV:BOOLEAN;
        MTZAUX,MTZAUXIL,TRANSP,MTZDESM,MTZPESO:MATRIZ;
        PESO,PESOP,CONVERG:REAL;

BEGIN
LERMATRIZ(ORDEM,MTZPESO);
ESCRMATRIZ(ORDEM,'Q',MTZPESO);
WRITE(TTY,'PESO DA VARIAVEL DE ENTRADA = ');BREAK;
READ(TTY,PESO);
WRITELN(TTY);
WRITE(TTY,'CONVERGENCIA PARA : ');BREAK;
READ(TTY,CONVERG);
ZEREMATRIZ(ORDEM,MTZAUXIL);
MIGUALM(ORDEM,MTZPESO,MTZDESM);
CONV:=FALSE;
MTZTRANSP(ORDEM,MTZSIST,TRANSP);
WHILE (NOT CONV) DO
        BEGIN
        CALCULEK(ORDEM,PESO,VTRENTR,MTZDESM,MTZSIST,PESOP,VTRGANHO,MTZAUX);
        TESTCONV(ORDEM,CONVERG,CONV,MTZDESM,MTZAUXIL);
        CALCULEP(ORDEM,PESOP,VTRGANHO,MTZAUX,TRANSP,MTZPESO,MTZDESM)
(*      ESCRMATRIZ(ORDEM,'P',MTZDESM)*)
        END;
```

```
WRITELN(TTY);
CASE TIPO OF
        'K':BEGIN
            WRITELN(TTY,'VETOR D. GANHOS DO CONTROLADOR ');
            ESCVL(ORDEM,VTRGANHO)
            END;
        'H':BEGIN
            WRITELN(TTY,'VETOR DE REALIMENTACAO DO OBSERVADOR');
            ESCVC(ORDEM,VTRGANHO)
            END
        END;
WRITELN(TTY)
END;(* CALCULO DA EQUACAO DE RICCATI *)

PROCEDURE VALORPOLINOMIO(PONTO,A2,A1,A0:REAL;
                         VAR B1,B0,VALOR:REAL);

BEGIN
B1:=A2 + PONTO;
B0:=A1+PONTO*B1;
VALOR:=A0+PONTO*B0
END;(* CALCULA VALOR DO POLINOMIO *)

FUNCTION VALORDERIVADA(PONTO,B1,B0:REAL):REAL;

VAR
        C1:REAL;

BEGIN
VALORDERIVADA:=3*SQR(PONTO)+2*B1*PONTO+B0
END;(* CALCULA VALOR PARA A DERIVADA NO PONTO *)


PROCEDURE RAIZESSEG(B1,B0:REAL;
                    VAR R1,R2:COMPLEXO);

VAR
        DELTA:REAL;

BEGIN
DELTA:=SQR(B1)-4*B0;
R1[I]:=0.0;
R2[I]:=0.0;
IF DELTA>=0.0 THEN
        BEGIN
        IF B1 > 0.0 THEN
                R1[R]:= (-B1-SQRT(DELTA))/2.0
        ELSE
                R1[R]:= (-B1+SQRT(DELTA))/2.0;
        R2[R]:=B0/R1[R]
        END
ELSE
        BEGIN
        R1[I]:=SQRT(-DELTA)/2.0;
        R2[I]:=-R1[I];
        R1[R]:=-B1/2;
        R2[R]:=R1[R]
        END
END;(* RESOLUCAO DA EQUACAO QUADRATICA *)
```

```
PROCEDURE RAIZNEWTON (A2,A1,A0:REAL;
                VAR R1,R0:REAL;
                VAR RAIZ:COMPLEXO);

CONST
        CONVERG=1.0E-15;
        CONV1=1.0E-15;
TYPE
        ECONV=(S,N);

VAR
        REPITA:ECONV;
        RAIZPRONTA:BOOLEAN;
        K:INTEGER;
        PONTO1,PONTO2,DERIVADA,POLINOMIO:REAL;
BEGIN
REPITA:=N;
WRITELN(TTY);
WHILE REPITA = N DO
        BEGIN
        WRITE(TTY,'PONTO INICIAL PARA CALCULO DO AUTOVALOR REAL = ');BREAK;
        READ(TTY,PONTO1);
        K:=0;
        RAIZPRONTA:=FALSE;
        WHILE NOT RAIZPRONTA DO
                BEGIN
(*              WRITE(TTY,'ZEROS DE NEWTON ',PONTO1); *)
                DERIVADA:=VALORDERIVADA(PONTO1,A2,A1);
                VALORPOLINOMIO(PONTO1,A2,A1,A0,R1,R0,POLINOMIO);
                WRITELN(TTY,'   VALOR DO POLINOMIO ',POLINOMIO);
                IF DERIVADA <> 0.0 THEN
                        PONTO2:=PONTO1-POLINOMIO/DERIVADA
                ELSE
                        PONTO2:=PONTO1*1.001;
                IF (ABS(PONTO2-PONTO1) <= CONVERG) THEN
                        BEGIN
                        WRITELN(TTY,'AUTOVALOR = ',PONTO2,'   VALOR DO POLINOMIO = ',POLINOMIO);BREAK;
                        RAIZ[R]:=PONTO2;
                        RAIZ[I]:=0.0;
                        RAIZPRONTA:=TRUE
                        END
                ELSE
                        PONTO1:=PONTO2;
                K:=SUCC(K);
                IF K=20 THEN
                        BEGIN
                        RAIZPRONTA:=TRUE;
                        RAIZ[R]:=PONTO2;
                        RAIZ[I]:=0.0
                        END
                END;
        WRITE(TTY,'AUTOVALOR E ACEITAVEL ? (S OU N) = ');BREAK;
        READ(TTY,REPITA);
        WRITELN(TTY)

        END
END;(* CALCULO DA RAIZ REAL METODO DE NEWTON *)

PROCEDURE AUTOVALORES(A:MATRIZ;
                        VAR RAIZ1,RAIZ2,RAIZ3:COMPLEXO);
```

```
VAR
        A2,A1,A0,B1,B0:REAL;
        REPETIR:(S,N);
BEGIN
A2:=-A[1,1]-A[2,2]-A[3,3];
A1:=A[1,1]*A[3,3] + A[2,2]*A[3,3] + A[1,1]*A[2,2]          -(A[3,2]*A[2,3] + A[3,1]*A[1,3] + A[1,2]*A[2,1]);
A0:= -(A[3,3]*A[2,2]*A[1,1] + A[3,1]*A[1,2]*A[2,3]        + A[1,3]*A[2,1]*A[3,2]) + A[3,1]*A[1,3]*A[2,2]
            + A[1,2]*A[2,1]*A[3,3] + A[3,2]*A[2,3]*A[1,1];
REPETIR := S;
WHILE REPETIR = S DO
        BEGIN
        RAIZNEWTON(A2,A1,A0,B1,B0,RAIZ3);
        RAIZESSEG(B1,B0,RAIZ1,RAIZ2);
        WRITELN(TTY,'AUTOVALOR 1 = ',RAIZ1[R],' + J ',RAIZ1[I]);
        WRITELN(TTY,'AUTOVALOR 2 = ',RAIZ2[R],' + J ',RAIZ2[I]);
        WRITELN(TTY,'AUTOVALOR 3 = ',RAIZ3[R],' + J ',RAIZ3[I]);
        WRITELN(TTY);
        WRITE(TTY,'REPETIR CALCULO DOS AUTOVALORES ? (S OU N) ');BREAK;
        READ(TTY,REPETIR)
        END
END;(* DETERMINACAO DOS AUTOVALORES *)

PROCEDURE SOMAR(R1,R2:COMPLEXO;
                VAR R3:COMPLEXO);
BEGIN
R3[R]:=R1[R]+R2[R];
R3[I]:=R1[I]+R2[I]
END;(* SOMAR COMPLEXOS *)

PROCEDURE MULTI(R1,R2:COMPLEXO;
                VAR R3:COMPLEXO);
BEGIN
R3[R]:=R1[R]*R2[R] - R1[I]*R2[I];
R3[I]:=R1[I]*R2[R] + R2[I]*R1[R]
END;(* MULTIPLICAR COMPLEXOS *)

PROCEDURE DIVID(R1,R2:COMPLEXO;
                VAR R3:COMPLEXO);

VAR
        DUMMY:BOOLEAN;
        MODUL:REAL;
        R4:COMPLEXO;
BEGIN
MODUL:=SQR(R2[R]) + SQR(R2[I]);
R4[R]:=R2[R];
R4[I]:=-R2[I];
IF MODUL <> 0 THEN
        BEGIN
        MULTI(R1,R4,R3);
        R3[R]:=R3[R]/MODUL;
        R3[I]:=R3[I]/MODUL
        END
ELSE
        BEGIN
        WRITELN(TTY,'DIVISAO POR ZERO - COMPLEXO !');
        READLN(TTY,DUMMY)
        END
END;(* DIVISAO DE COMPLEXOS *)
```

```
PROCEDURE IGUAL(R1:COMPLEXO;
                VAR R2:COMPLEXO);
BEGIN
R2[R]:=R1[R];
R2[I]:=R1[I]
END;(* IGUALAR COMPLEXOS *)

PROCEDURE MENOS(R1:COMPLEXO;
                VAR R2:COMPLEXO);
BEGIN
R2[R]:=-R1[R];
R2[I]:=-R1[I]
END;(* NEGAR COMPLEXO *)

PROCEDURE SUBTR(R1,R2:COMPLEXO;
                VAR R3:COMPLEXO);
BEGIN
R3[R]:=R1[R]-R2[R];
R3[I]:=R1[I]-R2[I]
END;(* SUBTRAIR COMPLEXOS *)


PROCEDURE AUTOVETOR(AUTOVALOR:COMPLEXO;
                    ELIMINA:CVETOR;
                    A:MATRIZ;
                    VAR P:CVETOR);

VAR
        N1,N2,N3,NN2,NN3,L,J:INTEGER;
        A1,A2,A3:REAL;
        PIVOT,AUX1,AUX2,PIVOT1,PIVOT2:COMPLEXO;
        C:CMATRIZ;

BEGIN
FOR L:=1 TO ORDEM DO
        FOR J:=1 TO ORDEM DO
                IF L=J THEN
                        BEGIN
                        C[L,J,R]:=AUTOVALOR[R] - A[L,J];
                        C[L,J,I]:=AUTOVALOR[I]
                        END
                ELSE
                        BEGIN
                        C[L,J,R]:=-A[L,J];
                        C[L,J,I]:=0.0
                        END;
A1:=ABS(C[1,1,R]);
A2:=ABS(C[2,1,R]);
A3:=ABS(C[3,1,R]);
IF A2 > A1 THEN
        IF A2 >= A3 THEN
                IF A1 >= A3 THEN
                        BEGIN
                        N1:=2;N2:=1;N3:=3
                        END
                ELSE
                        BEGIN
                        N1:=2;N2:=3;N3:=1
                        END
```

```
        ELSE
                BEGIN
                N1:=3;N2:=2;N3:=1
                END
ELSE
        IF A1 >= A3 THEN
                IF A2 >= A3 THEN
                        BEGIN
                        N1:=1;N2:=2;N3:=3
                        END
                ELSE
                        BEGIN
                        N1:=1;N2:=3;N3:=2
                        END
        ELSE
                BEGIN
                N1:=3;N2:=1;N3:=2
                END;
IF (C[N1,1,R] <> 0.0) OR (C[N1,1,I] <> 0.0) THEN
                BEGIN
                DIVID(C[N2,1],C[N1,1],PIVOT);
                PIVOT1:=PIVOT;
                MULTI(C[N1,2],PIVOT,AUX1);
                SUBTR(C[N2,2],AUX1,AUX2);
                IGUAL(AUX2,C[N2,2]);
                MULTI(C[N1,3],PIVOT,AUX1);
                SUBTR(C[N2,3],AUX1,AUX2);
                IGUAL(AUX2,C[N2,3]);
                MULTI(PLINHA[N1],PIVOT,AUX1);
                SUBTR(PLINHA[N2],AUX1,AUX2);
                IGUAL(AUX2,PLINHA[N2]);
                DIVID(C[N3,1],C[N1,1],PIVOT);
                PIVOT2:=PIVOT;
                MULTI(C[N1,2],PIVOT,AUX1);
                SUBTR(C[N3,2],AUX1,AUX2);
                IGUAL(AUX2,C[N3,2]);
                MULTI(C[N1,3],PIVOT,AUX1);
                SUBTR(C[N3,3],AUX1,AUX2);
                IGUAL(AUX2,C[N3,3]);
                MULTI(PLINHA[N3],PIVOT,AUX1);
                SUBTR(PLINHA[N3],AUX1,AUX2);
                IGUAL(AUX2,PLINHA[N3]);
                A1:= ABS(C[N2,2,R]);
                A2:= ABS(C[N3,2,R]);
                IF ( (A1 >= A2) AND (A1<> 0.0)) THEN
                        BEGIN
                        P[3,R]:=1.0;
                        P[3,I]:=0.0;
                        NN2:=N2;
                        NN3:=N3;
                        SUBTR(PLINHA[N2],C[N2,3],AUX1);
                        DIVID(AUX1,C[N2,2],P[2]);
                        MULTI(C[N1,2],P[2],AUX1);
                        SUBTR(PLINHA[N1],AUX1,AUX2);
                        SUBTR(AUX2,C[N1,3],AUX1);
                        DIVID(AUX1,C[N1,1],P[1])
                        END
                ELSE
                        IF A2 <> 0.0 THEN
                        BEGIN
```

```
                    P[3,R]:=1.0;
                    P[3,I]:=0.0;
                    N2:=N3;
                    N3:=N2;
                    SUBTR(PLINHA[N3],C[N3,3],AUX1);
                    DIVID(AUX1,C[N3,2],P[2]);
                    MULTI(C[N1,2],P[2],AUX1);
                    SUBTR(PLINHA[N1],AUX1,AUX2);
                    SUBTR(AUX2,C[N1,3],AUX1);
                    DIVID(AUX1,C[N1,1],P[1])
                    END
            END
ELSE
            WRITELN(TTY,' PROBLEMA COM AUTOVETORES - C[2,2] = 0.0')
END;(* RESOLVER EQUACAO DE AUTOVETORES *)

PROCEDURE DISTAUTOVETORES(RAIZ1,RAIZ2,RAIZ3:COMPLEXO;
                        A:MATRIZ;
                        VAR V:CMATRIZ);
VAR
        J:INTEGER;
        AUX,AV1,AV2,AV3:CVETOR;
BEGIN
FOR J:=1 TO ORDEM DO
        BEGIN
        AUX[J,R]:=0.0;
        AUX[J,I]:=0.0
        END;
AUTOVETOR(RAIZ1,AUX,A,AV1);
AUTOVETOR(RAIZ2,AUX,A,AV2);
AUTOVETOR(RAIZ3,AUX,A,AV3);
FOR J:=1 TO ORDEM DO
        BEGIN
        IGUAL(AV1[J],V[J,1]);
        IGUAL(AV2[J],V[J,2]);
        IGUAL(AV3[J],V[J,3])
        END
END; (* AUTOVALORES DISTINTOS *)

PROCEDURE IGUALVETORES(NUM:INTEGER;
                        RAIZ1,RAIZ2,RAIZ3:COMPLEXO;
                        A:MATRIZ;
                        VAR V:CMATRIZ);
VAR
        K,J:INTEGER;
        AUX,AV1,AV2,AV3:CVETOR;
BEGIN
FOR J:=1 TO ORDEM DO
        BEGIN
        AUX[J,R]:=0.0;
        AUX[J,I]:=0.0
        END;
IF NUM=1 THEN
        BEGIN
        AUTOVETOR(RAIZ3,AUX,A,AV1);
        AUTOVETOR(RAIZ1,AUX,A,AV2);
        FOR J:=1 TO ORDEM DO
                MENOS(AV2[J],AUX[J]);
                    AUTOVETOR(RAIZ2,AUX,A,AV3)
        END
```

```
                                        NUM:=0
                                        END;
IF (UM<> 0 THEN
        BEGIN
        IGUALVETORES(NUM,RAIZ1,RAIZ2,RAIZ3,A,AUVT);
        FOR K:=1 TO ORDEM DO
                FOR J:=1 TO ORDEM DO
                        AUVT[K,J,I]:=0.0
        END
(*WRITELN(ITY,'MATRIZ DE AUTOVETORES');
FOR J:=1 TO ORDEM DO
        BEGIN
        WRITE(ITY,'[ ');
        FOR K:=1  TO ORDEM DO
                WRITE(ITY,AUVT[J,K,R],' + ',AUVT[J,K,I],' J');
        WRITELN(ITY,' ]')
        END*)
END;(* AUTOVETORES *)

PROCEDURE PINVERSA(P:CMATRIZ;
                   VAR INVP:CMATRIZ);

VAR
        DETERMI,AUX1,AUX2,AUX3:COMPLEXO;

BEGIN
MULTI(P[1,1],P[2,2],AUX1);
MULTI(AUX1,P[3,3],DETERMI);
MULTI(P[1,2],P[2,3],AUX1);
MULTI(AUX1,P[3,1],AUX2);
SOMAR(DETERMI,AUX2,AUX1);
IGUAL(AUX1,DETERMI);
MULTI(P[2,1],P[3,2],AUX1);
MULTI(AUX1,P[1,3],AUX2);
SOMAR(DETERMI,AUX2,AUX1);
IGUAL(AUX1,DETERMI);
MULTI(P[3,1],P[2,2],AUX1);
MULTI(AUX1,P[1,3],AUX2);
SUBTR(DETERMI,AUX2,AUX1);
IGUAL(AUX1,DETERMI);
MULTI(P[2,1],P[1,2],AUX1);
MULTI(AUX1,P[3,3],AUX2);
SUBTR(DETERMI,AUX2,AUX1);
IGUAL(AUX1,DETERMI);
MULTI(P[1,1],P[3,2],AUX1);
MULTI(AUX1,P[2,3],AUX2);
SUBTR(DETERMI,AUX2,AUX1);
IGUAL(AUX1,DETERMI);(* DETERMINANTE *)

IF (DETERMI[R] <> 0.0) OR (DETERMI[I] <>0.0) THEN
        BEGIN(* MATRIZ DOS COFATORES *)
        MULTI(P[2,2],P[3,3],AUX1);
        MULTI(P[2,3],P[3,2],AUX2);
        SUBTR(AUX1,AUX2,INVP[1,1]);
        MULTI(P[2,1],P[3,3],AUX1);
        MULTI(P[3,1],P[2,3],AUX2);
        SUBTR(AUX1,AUX2,AUX3);
        MENOS(AUX3,INVP[2,1]);
        MULTI(P[2,1],P[3,2],AUX1);
        MULTI(P[3,1],P[2,2],AUX2);
```

```
ELSE
        IF NUM=2 THEN
                BEGIN
                AUTOVETOR(RAIZ2,AUX,A,AV1);
                AUTOVETOR(RAIZ1,AUX,A,AV2);
                FOR J:=1 TO ORDEM DO
                        MENOS(AV1[J],AUX[J]);
                AUTOVETOR(RAIZ3,AUX,A,AV3)
                END
        ELSE
                IF NUM=3 THEN
                        BEGIN
                        AUTOVETOR(RAIZ1,AUX,A,AV1);
                        AUTOVETOR(RAIZ2,AUX,A,AV2);
                        FOR J:=1 TO ORDEM DO
                                MENOS(AV2[J],AUX[J]);
                        AUTOVETOR(RAIZ3,AUX,A,AV3)
                        END
                ELSE
                        IF NUM=4 THEN
                                BEGIN
                                AUTOVETOR(RAIZ1,AUX,A,AV1);
                                FOR J:=1 TO ORDEM DO
                                        MENOS(AV1[J],AUX[J]);
                                AUTOVETOR(RAIZ2,AUX,A,AV2);
                                FOR J:=1 TO ORDEM DO
                                        MENOS(AV2[J],AUX[J]);
                                AUTOVETOR(RAIZ3,AUX,A,AV3)
                                END;
FOR J:=1 TO ORDEM DO
        BEGIN
        IGUAL(AV1[J],V[J,1]);
        IGUAL(AV2[J],V[J,2]);
        IGUAL(AV3[J],V[J,3])
        END
END;(* AUTOVALORES IGUAIS *)

PROCEDURE RAUTOVETORES(RAIZ1,RAIZ2,RAIZ3:COMPLEXO;
                        A:MATRIZ;
                        VAR AUVI:CMATRIZ);
CONST
        CONVERG=1.0E-12;
VAR
        J,K,NUM:INTEGER;
BEGIN
IF ( (ABS(RAIZ1[R] - RAIZ2[R]) <= CONVERG) AND (ABS(RAIZ1[I] - RAIZ2[I]) <=CONVERG) AND (ABS(RAIZ1[R] - RAIZ3[R]) <=CONVERG) AND (AB
S(RAIZ1[I] - RAIZ3[I]) <= CONVERG) AND (ABS(RAIZ2[R] - RAIZ3[R]) <= CONVERG) AND (ABS(RAIZ2[I] - RAIZ3[I]) <= CONVERG )) THEN
        NUM:=1
ELSE
        IF ((ABS(RAIZ1[R] - RAIZ2[R]) <= CONVERG) AND (ABS(RAIZ1[I] - RAIZ2[I]) <= CONVERG )) THEN
                NUM:=1
        ELSE
                IF ( (ABS(RAIZ1[R] - RAIZ3[R]) <= CONVERG) AND (ABS(RAIZ1[I] - RAIZ3[I]) <=CONVERG) ) THEN
                        NUM:=2
                ELSE
                        IF ( (ABS(RAIZ2[R] - RAIZ3[R]) <= CONVERG) AND (ABS(RAIZ2[I] - RAIZ3[I]) <= CONVERG) ) THEN
                                NUM:=3
                        ELSE
                                BEGIN
                                DISTAUTOVETORES(RAIZ1,RAIZ2,RAIZ3,A,AUVI);
```

```
        SUBTR(AUX1,AUX2,INVP[3,1]);
        MULTI(P[1,2],P[3,3],AUX1);
        MULTI(P[1,3],P[3,2],AUX2);
        SUBTR(AUX1,AUX2,AUX3);
        MENOS(AUX3,INVP[1,2]);
        MULTI(P[1,1],P[3,3],AUX1);
        MULTI(P[3,1],P[1,3],AUX2);
        SUBTR(AUX1,AUX2,INVP[2,2]);
        MULTI(P[1,1],P[3,2],AUX1);
        MULTI(P[3,1],P[1,2],AUX2);
        SUBTR(AUX1,AUX2,AUX3);
        MENOS(AUX3,INVP[3,2]);
        MULTI(P[1,2],P[2,3],AUX1);
        MULTI(P[2,2],P[1,3],AUX2);
        SUBTR(AUX1,AUX2,INVP[1,3]);
        MULTI(P[1,1],P[2,3],AUX1);
        MULTI(P[2,1],P[1,3],AUX2);
        SUBTR(AUX1,AUX2,AUX3);
        MENOS(AUX3,INVP[2,3]);
        MULTI(P[1,1],P[2,2],AUX1);
        MULTI(P[2,1],P[1,2],AUX2);
        SUBTR(AUX1,AUX2,INVP[3,3]);
        FOR J:=1 TO ORDEM DO
                FOR K:=1 TO ORDEM DO
                        BEGIN
                        DIVID(INVP[J,K],DETERMI,AUX1);
                        IGUAL(AUX1,INVP[J,K])
                        END
        END
ELSE
        BEGIN
        WRITELN(TTY,'MATRIZ DE AUTOVETORES NAO ADMITE INVERSA');BREAK
        END;
WRITELN(TTY)
(*WRITELN(TTY,'INVERSA DA MATRIZ DOS AUTOVETORES');
FOR J:=1 TO ORDEM DO
        BEGIN
        WRITE(TTY,'[ ');
        FOR K:=1 TO ORDEM DO
                WRITE(TTY,INVP[J,K,R],' + ',INVP[J,K,I],' J');
        WRITELN(TTY,' ]')
        END*)
END; (* INVERSA DE MATRIZ COMPLEXA *)

PROCEDURE CMULTMTZNN(X,Y:CMATRIZ;
                        VAR Z:CMATRIZ);
VAR
        J,K,L:INTEGER;
        AUX,AUX1,AUX2:COMPLEXO;
BEGIN
FOR J:=1 TO ORDEM DO
        FOR L:=1 TO ORDEM DO
                BEGIN
                AUX[R]:=0.0;
                AUX[I]:=0.0;
                FOR K:=1 TO ORDEM DO
                        BEGIN
                        MULTI(X[J,K],Y[K,L],AUX1);
                        SOMAR(AUX,AUX1,AUX2);
                        IGUAL(AUX2,AUX)
```

```
                    END;
            IGUAL(AUX2,Z[J,L])
            END
END; (* MULTIPLICAR MATRIZES COMPLEXAS *)

PROCEDURE CMULTMNN1(Y:CVETOR;
                    X:CMATRIZ;
                    VAR Z:CVETOR);
VAR
        J,K:INTEGER;
        AUX,AUX1,AUX2:COMPLEXO;
BEGIN
FOR J:=1 TO ORDEM DO
        BEGIN
        AUX[R]:=0.0;
        AUX[I]:=0.0;
        FOR K:=1 TO ORDEM DO
                BEGIN
                MULTI(X[J,K],Y[K],AUX1);
                SOMAR(AUX,AUX1,AUX2);
                IGUAL(AUX2,AUX)
                END;
        IGUAL(AUX,Z[J])
        END
END;(* MULTIPLICAR  MATRIZ POR VETOR COMPLEXOS *)

PROCEDURE CMULTV1NN(X:CVETOR;
                    Y:CMATRIZ;
                    VAR Z: CVETOR);
VAR
        J,K:INTEGER;
        AUX,AUX1,AUX2:COMPLEXO;
BEGIN
FOR J:=1 TO ORDEM DO
        BEGIN
        AUX[R]:=0.0;
        AUX[I]:=0.0;
        FOR K:=1 TO ORDEM DO
                BEGIN
                MULTI(X[K],Y[K,J],AUX1);
                SOMAR(AUX,AUX1,AUX2);
                IGUAL(AUX2,AUX)
                END;
        IGUAL(AUX2,Z[J])
        END
END;(* MULTIPLICAR VETOR POR MATRIZ COMPLEXOS *)

PROCEDURE TRANSFILEGH(J1:REAL;KDS,HS:CVETOR;
                         AJ:CMATRIZ;
                         VAR KDF:CVETOR);
VAR
        AUX1,AUX2,AUX3:COMPLEXO;
BEGIN
MULTI(KDS[1],HS[1],AUX1);
MULTI(KDS[2],HS[2],AUX2);
SOMAR(AUX1,AUX2,KDF[1]);
MULTI(KDS[2],HS[1],AUX1);
KDF[2,R]:=J1 * AUX1[R];
KDF[2,I]:=J1 * AUX1[I];
MULTI(KDS[1],HS[1],AUX1);
```

```
MULTI(AUX1,AJ[2,2],AUX2);
MULTI(KDS[2],HS[2],AUX1);
MULTI(AUX1,AJ[1,1],AUX3);
SUBTR(KDF[2],AUX2,AUX1);
SUBTR(AUX1,AUX3,KDF[2]);
MULTI(KDS[3],HS[3],KDF[3])
END;(* TRANSFORMACAO LINEAR DO GRAFICO DE FLUXO DE SINAIS *)

PROCEDURE INVERAMBK(C:MATRIZ;
                    VAR IA:MATRIZ);
CONST
        CONVERG=1.0E-30;
VAR
        DETERMINANTE:REAL;
        J,L:INTEGER;
BEGIN
DETERMINANTE:=C[1,1]*C[2,2]*C[3,3] + C[1,2]*C[2,3]*C[3,1] + C[2,1]*C[3,2]*C[1,3] - C[3,1]*C[2,2]*C[1,3] - C[2,1]*C[1,2]*C[3,3] - C[1
,1]*C[3,2]*C[2,3];
IF ABS(DETERMINANTE) >= CONVERG THEN
        BEGIN
        IA[1,1]:=C[2,2]*C[3,3] - C[2,3]*C[3,2];
        IA[2,1]:=-(C[2,1]*C[3,3] - C[3,1]*C[2,3]);
        IA[3,1]:=C[2,1]*C[3,2] - C[3,1]*C[2,2];
        IA[1,2]:=-(C[1,2]*C[3,3] - C[1,3]*C[3,2]);
        IA[2,2]:=C[1,1]*C[3,3] - C[3,1]*C[1,3];
        IA[3,2]:=-(C[1,1]*C[3,2] - C[3,1]*C[1,2]);
        IA[1,3]:=C[1,2]*C[2,3] - C[2,2]*C[1,3];
        IA[2,3]:=-(C[1,1]*C[2,3] - C[2,1]*C[1,3]);
        IA[3,3]:=C[1,1]*C[2,2] - C[2,1]*C[1,2];
        FOR J:=1 TO ORDEM DO
                FOR L:=1 TO ORDEM DO
                        IA[J,L]:=IA[J,L]/DETERMINANTE
        END
ELSE
        WRITELN(TTY,'INVERSA DE A-BK NAO EXISTE')
END;(*CALCULAR INVERSA DE A-BK*)

PROCEDURE KOALFABETA(INVAMBK:MATRIZ;
                     K:VETOR;
                     VAR KO:REAL;
                     VAR ALFA,BETA:VETOR);
CONST
        CONVERG=1.0E-30;
VAR
        L,J:INTEGER;
        AUX1,AUX2:MATRIZ;
        KOM,H:VETOR;
        POLOS:REAL;
BEGIN
FOR J:=2 TO ORDEM DO
        AUX1[1,J]:=0.0;
AUX1[1,1]:=1.0;
FOR J:=1 TO ORDEM DO
        AUX1[2,J]:=INVAMBK[1,J];
MULTMIZAN(ORDEM,INVAMBK,INVAMBK,AUX2);
FOR J:=1 TO ORDEM DO
        AUX1[3,J]:=AUX2[1,J];
FOR L:=1 TO ORDEM DO
        FOR J:=1 TO ORDEM DO
                AUX2[L,J]:=AUX1[L,J];
```

UNICAMP
Centro de Computação

```
IF ABS(AUX2[2,2]) > CONVERG THEN
        BEGIN
        AUX2[3,3]:=AUX2[3,3] - AUX2[3,2]*AUX2[2,3]/AUX2[2,2];
        IF ABS(AUX2[3,3]) > CONVERG THEN
                BEGIN
                KOH[3]:=(K[3] - K[2]*AUX2[2,3]/AUX2[2,2])/AUX2[3,3];
                KOH[2]:=(K[2] - AUX2[3,2]*KOH[3])/AUX2[2,2];
                KOH[1]:=(K[1] - AUX2[2,1]*KOH[2] - AUX2[3,1]*KOH[3])
                END
        ELSE
              WRITELN(TTY,'OS PARAMETROS DA SERIE NAO EXISTEM')
        END
ELSE
        IF (ABS(AUX2[3,2]) >= CONVERG) AND (ABS(AUX2[2,3]) >= CONVERG) THEN
                BEGIN
                KOH[3]:=K[3]/AUX2[3,2];
                KOH[2]:=(K[2] - AUX2[3,3]*KOH[3])/AUX2[2,3];
                KOH[1]:=K[1] - AUX2[2,1]*KOH[2] - AUX2[3,1]*KOH[3]
                END
        ELSE
                WRITELN(TTY,'OS PARAMETROS DA SERIE NAO EXISTEM');
IF ABS(KOH[1]) >= CONVERG THEN
        BEGIN
        KD:=KOH[1];
        H[1]:=1.0;
        H[2]:=KOH[2]/KD;
        H[3]:=KOH[3]/KD
        END
ELSE
        IF ABS(KOH[2]) >= CONVERG THEN
                BEGIN
                KD:=KOH[2];
                H[1]:=0.0;
                H[2]:=1.0;
                H[3]:=KOH[3]/KD
                END
        ELSE
                BEGIN
                KD:=1.0;
                H[1]:=0.0;
                H[2]:=0.0;
                H[3]:=KOH[3]
                END;
(*CALCULO DE ALFA E BETA*)
WRITELN(TTY,'QUAIS OS POLOS DO REGULADOR DINAMICO ?');
WRITE(TTY,'(MULTIPLICIDADE 2 E PROXIMO DE ZERO) = ');BREAK;
READ(TTY,POLOS);
BETA[1]:=-2*POLOS;
BETA[2]:=SQR(POLOS);
ALFA[1]:=BETA[1] + H[2];
ALFA[2]:=BETA[2] + H[3] + BETA[1]*H[2]
END;(* CALCULO DE KD ALFA E BETA *)

(*CORPO DO PROGRAMA *)
BEGIN
WRITELN(TTY,'PROGRAMA PARA O PROJETO DOS CONTROLADORES OTIMOS');
WRITELN(TTY);
WRITELN(TTY);
LERPARAMETROS(PULSOQUAD,PES,IND,PRI,IND,CNA,CK6);
LERPERIODO(PERIODO);
```

```
MUDARPARAMETROS(PULSOCODI,RES,IND,FRI,INF,CKA,CKB);
FIM:=FALSE;
WHILE NOT FIM DO

(* PROJETO DO REGULADOR E OBSERVADOR  *)

        BEGIN
        MUDARPERIODO(PERIODO);
        RAIZESREAIS(RES,IND,FRI,INE,CKA,CKB,RAIZ1,RAIZ2);
        COEFICIENTES(PULSOCODI,RES,IND,FRI,INF,CKA,CKB,PERIODO,RAIZ1,RAIZ2,VTRENTR,MTZSIST);
        EXECMATSIS(VTRENTR,MTZSIST);
        COEFICIENTES(PULSOCODI,RES,IND,FRI,INF,CKA,CKB,PERIODO/8,RAIZ1,RAIZ2,VTRSIMU,MTZSIMU);
        WRITELN(TTY);
        WRITELN(TTY,'PROJETO DO REGULADOR');
        WRITELN(TTY);
        RICCATI('K',VTRENTR,MTZSIST,VTRGANHO);
        WRITELN(TTY);
        WRITELN(TTY,'PROJETO DO OBSERVADOR');
        WRITELN(TTY);
        D[1]:=1.0;D[2]:=0.0;D[3]:=0.0;
        MTZTRANSP(ORDEM,MTZSIST,AT);
        RICCATI('L',D,AT,VTROBSERVADOR);
        MULTVETMTR(ORDEM,VTRENTR,VTRGANHO,BK);
        SUBTRAIMA(ORDEM,MTZSIST,BK,AMENOSBK);
(*      WRITELN(TTY,'MATRIZ A-BK');
        WRITELN(TTY);
        FOR K:=1 TO ORDEM DO
                BEGIN
                WRITE(TTY,'[ ');
                FOR J:=1 TO ORDEM DO
                        WRITE(TTY,AMENOSBK[K,J],'    ');
                WRITELN(TTY,' ] ')
                END;
        WRITELN(TTY); *)
        CONT:=SIM;

(*PROJETO DOS CONTROLADORES *)

        WHILE CONT=SIM DO
                BEGIN
                WRITELN(TTY,'TIPO DE CONTROLADOR  OTIMO 1) COM OBSERVADOR COM REALIMENTACAO =OTIMOREA');
                WRITELN(TTY,'                           2) COM OBSERVADOR SEM REALIMENTACAO=OTIMODIR');
                WRITELN(TTY,'                           3) DINAMICO                        =DINAMICO');
                WRITE(TTY,' = ');BREAK;
                READ(TTY,TIPOCONTROL);
                WRITELN(TTY);
(*
PROJETO DOS CONTROLADORES COM OBSERVADOR *)

                IF (TIPOCONTROL=OTIMOREA) OR (TIPOCONTROL=OTIMODIR) THEN
                        BEGIN
                        IF TIPOCONTROL=OTIMODIR THEN
                                BEGIN
                                MULTVETMTR(ORDEM,VTROBSERVADOR,D,HD);
                                SUBTRAIMA(ORDEM,AMENOSBK,HD,AMBKMHD);
(*                              WRITELN(TTY,'MATRIZ A-BK-HD');
                                FOR K:=1 TO ORDEM DO
                                        BEGIN
                                        WRITE(TTY,'[ ');
                                        FOR J:=1 TO ORDEM DO
```

```
                        WRITE(TTY,AMBKMHD(K,J));
                  WRITELN(TTY,' )')
                  END);
         WRITELN(TTY); *)
         FOR K:=1 TO ORDEM DO
                  FOR L:=1 TO ORDEM DO
                           AMENOSBK[K,L]:=AMBKMHD[K,L]
         END;
AUTOVALORES(AMENOSBK,AUTOVALI1,AUTOVALI2,AUTOVALI3);
PAUTOVETORES(AUTOVALI1,AUTOVALI2,AUTOVALI3,AMENOSBK,MTZAUTOVETORES);
PINVERSA(MTZAUTOVETORES,INVERSAP);
FOR J:=1 TO ORDEM DO
         FOR K:=1 TO ORDEM DO
                  BEGIN
                  CAMENOSBK[J,K,R]:=AMENOSBK[J,K];
                  CAMENOSBK[J,K,I]:=0.0
                  END;
CMULTMTZMN(CAMENOSBK,MTZAUTOVETORES,AMBKVZP);
CMULTMTZMN(INVERSAP,AMBKVZP,MTZJORDAN);
WRITELN(TTY);
(* WRITELN(TTY,'MATRIZ NA FORMA DIAGONAL OU DE JORDAN');
FOR J:=1 TO ORDEM DO
         BEGIN
         WRITE(TTY,'( ');
         FOR K:=1 TO ORDEM DO
                  WRITE(TTY,MTZJORDAN[J,K,R],' + ',MTZJORDAN[J,K,I],' J  ');
         WRITELN(TTY,' )')
         END;
WRITELN(TTY);*)
FOR J:=1 TO ORDEM DO
         BEGIN
         MEPS[J,R]:=VTROBSERVADOR[J];
         MEPS[J,I]:=0.0
         END;
CMULTMMV(MEPS,INVERSAP,VTRJORDAN);
(* WRITELN(TTY,'VETOR DE ENTRADA DA FORMA DIAGONAL OU DE JORDAN');
FOR J:=1 TO ORDEM DO
         WRITELN(TTY,'[ ',VTRJORDAN[J,R],' + ',VTRJORDAN[J,I],' J  ]');*)
FOR J:=1 TO ORDEM DO
         BEGIN
         VTROSJORDAN[J,R]:=MTZAUTOVETORES[1,J,R];
         VTROSJORDAN[J,I]:=MTZAUTOVETORES[1,J,I]
         END;
MULTVETMAT(ORDEM,VTRGANHO,VTROBSERVADOR,GANHODIRETO);
GANHODIRETO:= -GANHODIRETO;
FOR J:=1 TO ORDEM DO
         BEGIN
         CVTRGANHO[J,R]:=-VTRGANHO[J];
         CVTRGANHO[J,I]:=0.0
         END;
CMULTVMM(CVTRGANHO,AMBKVZP,CGANHOESTADO);
WRITELN(TTY);
FOR J:=1 TO 70 DO
         WRITE(TTY,'-');
WRITELN(TTY);
WRITELN(TTY);
WRITE(TTY,'PARAMETROS DO CONTROLADOR DIGITAL OTIMO COM OBSERVADOR');
IF TIPOCONTROLE=OTIMOREA THEN
         WRITELN(TTY,' COM REALIMENTACAO ')
ELSE
```

```
                IF TIPOCONTROLE=OTIMOQUIR THEN
                        WRITELN(TTY,' SEM REALIMENTACAO');
        WRITELN(TTY);
        IF NOT((AUTOVALI1=AUTOVALI2) AND (AUTOVALI2=AUTOVALI3) ) THEN
                BEGIN
                IF AUTOVALI1 = AUTOVALI2 THEN
                        J1:=1.0
                ELSE
                        J1:=0.0;
                J2:=0.0;
(*              WRITELN(TTY,'DIAGRAMA DE FLUXO DE SINAIS NO. 1'); *)
                TRANSFINEGR(J1,VTRDSJORDAN,VTRJORDAN,MTZJORDAN,VTRREALIMENT);
                TRANSFINEGR(J1,-GANHOESTADO,VTRJORDAN,MTZJORDAN,VTRDIRETO);
                SOMAR(MTZJORDAN[1,1],MTZJORDAN[2,2],REALIM[1]);
                MULTI(MTZJORDAN[1,1],MTZJORDAN[2,2],AUX1);
                MENOS(AUX1,REALIM[2]);
                IGUAL(MTZJORDAN[3,3],REALIM[3])
                END
        ELSE
                BEGIN
(*              WRITELN(TTY,' TRES AUTOVALORES IGUAIS, DIAGRAMA NO. 2'); *)
                FOR J:=1 TO ORDEM DO
                        BEGIN
                        VTRREALIMENT[J,R]:=VTRDSJORDAN[J,R]*VTRJORDAN[J,R];
                        VTRREALIMENT[J,I]:=0.0;
                        VTRDIRETO[J,R]:=-GANHOESTADO[J,R]*VTRJORDAN[J,R];
                        VTRDIRETO[J,I]:=0.0;
                        REALIM[J,R]:=MTZJORDAN[J,J,R];
                        REALIM[J,I]:=0.0;
                        J1:=VTRJORDAN[1,R]/VTRJORDAN[2,R];
                        J2:=VTRJORDAN[2,R]/VTRJORDAN[3,R]
                        END
                END;
        WRITELN(TTY,'R = ',GANHODIRETO);
        WRITELN(TTY);
        FOR J:=1 TO ORDEM DO
                WRITELN(TTY,'LAMBDA',J:1,' = ',REALIM[J,R]);(*,' + ',REALIM[J,I],' J'); *)
        WRITELN(TTY);
        FOR J:=1 TO ORDEM DO
                WRITELN(TTY,'KF',J:1,' = ',VTRDIRETO[J,R]);(*,' + ',VTRDIRETO[J,I],' J'); *)
        WRITELN(TTY);
        IF TIPOCONTROLE=OTIMOREA THEN
                FOR J:=1 TO ORDEM DO
                        WRITELN(TTY,'DF',J:1,' = ',VTRREALIMENT[J,R]);(*,' + ',VTRREALIMENT[J,I],' J');*)
        WRITELN(TTY);
        WRITELN(TTY,'J1= ',J1,'    J2 = ',J2);
        FOR K:=1 TO 72 DO
                WRITE(TTY,'-');
        WRITELN(TTY);
        WRITELN(TTY)
        END
    ELSE

(*PROJETO DO CONTROLADOR DINAMICO *)

        IF TIPOCONTROLE=DINAMICO THEN
                BEGIN
                INVERAMBK(A-EMBSBE,INVAMBK);
                CDALFABETA(INVAMBK,VTRGANHO,KO,ALFAS,BETAS);
                GAMA[1]:=KO*ALFAS[1] - BETAS[1];
```

UNICAMP
Centro de Computação

```
                    GAMA[2]:=KD*ALFAS[2] - BETAS[2];
                    EPS[1]:= - GAMA[1]*BETAS[2];
                    EPS[2]:= GAMA[2] - GAMA[1]*BETAS[1];
                    FOR J:=1 TO 72 DO
                            WRITE(TTY,'-');
                    WRITELN(TTY);
                    WRITELN(TTY);
                    WRITELN(TTY,'PARAMETROS DO CONTROLADOR DINAMICO');
                    WRITELN(TTY,'KD = ',KD);
                    WRITELN(TTY,'BETA1 = ',BETAS[1]);
                    WRITELN(TTY,'BETA2 = ',BETAS[2]);
                    WRITELN(TTY,'GAMA1 = ',GAMA[1]);
                    WRITELN(TTY,'GAMA2 = ',GAMA[2]);
(*                  WRITELN(TTY,'EPS1 = ',EPS[1]);
                    WRITELN(TTY,'EPS2 = ',EPS[2]); *)
                    WRITELN(TTY);
                    FOR K:=1 TO 72 DO
                            WRITE(TTY,'-');
                    WRITELN(TTY);
                    WRITELN(TTY)
                    END;
            WRITELN(TTY);
            WRITE(TTY,'CALCULAR OUTRO CONTROLADOR ? (SIM OU NAO) = ');BREAK;
            READ(TTY,CONT)
            END;
        WRITELN(TTY);
        WRITELN(TTY,'PARAMETROS PARA SIMULACAO');
        EVETMATRIZ(VTRSIMU,MTZSIMU);
        WRITELN(TTY);
        FOR J:=1 TO 70 DO
                WRITE(TTY,'-');
        WRITELN(TTY);
        WRITELN(TTY);
        WRITE(TTY,'REPETIR CALCULO DOS CONTROLADORES PARA OUTRO PERIODO  ? (SIM OU NAO) = ');BREAK;
        READ(TTY,FINAL);
        WRITELN(TTY);
        IF FINAL=NO THEN
                FIM:=TRUE
        END;
WRITELN(TTY,'FIM DE PROGRAMA')
END.(* FIM DO PROGRAMA *)
```

```
NXT56   MOV    A,B
        CPI    01H                ;NEGATIVO
        JNZ    SFIM6
        MVI    H,07FH             ;OVERFLOW POSITIVO
Press RETURN to Continue
        MVI    L,0FCH
;
SFIM6   SHLD   X1KMS1             ;FIM DE SOMA
;
;UPDATE VARIAVEIS DE ESTADO
;
        LHLD   X1K
        SHLD   X2K
        LHLD   X1KMS1
        SHLD   X1K
;
;
;GERACAO DA ENTRADA DE COMANDO
;
;
        LHLD   SPR
        DCX    H
        SHLD   SPR
        MOV    A,H
        ANA    A
        JNZ    CONTIN
        MOV    A,L
        ANA    A
Press RETURN to Continue
        JNZ    CONTIN
        LHLD   SP1
        SHLD   SPR
        LDA    CHAVE
        CPI    ZERO
        JZ     SOME3
        JMP    SUBT3
;
SUBT3   MVI    A,ZERO
        STA    CHAVE
        LHLD   AMP
        LDA    WENTR
        SUB    L
        STA    WENTR
        LDA    WENTR+1
        SBB    H
        STA    WENTR+1
        JMP    CONTIN
;
SOME3   MVI    A,01
        STA    CHAVE
        LHLD   AMP
        XCHG
Press RETURN to Continue
        LHLD   WENTR
        DAD    D
        SHLD   WENTR
        JMP    CONTIN
;
CONTIN  HLT
;
        END
```

APÊNDICE A7 - CONTROLADOR II

```
;PROGRAMA CONTROLADOR II
;
;AUTOR:PERICLES REZENDE BARROS
;
;TESE DE MESTRADO EM ENGENHARIA ELETRICA APRESENTADA NA
;        UNIVERSIDADE FEDERAL DA PARAIBA
;        CENTRO DE CIENCIAS E TECNOLOGIA
;        DEPARTAMENTO DE ENGENHARIA ELETRICA
;        CAMPINA GRANDE - PARAIBA
;                                          E DESLNVOLVIDA NA
;        UNIVERSIDADE ESTADUAL DE CAMPINAS
;        FACULDADE DE ENGENHARIA DE CAMPINAS
;        DEPARTAMENTO DE ENGENHARIA ELETRICA
;        CAMPINAS - SAO PAULO
;
;
;PROGRAMA CONTROLADOR
;SIMBOLOS USADOS
DACONV  EQU     80H             ;ENDERECO CONVERSOR D/A
LTIMER  EQU     034H            ;PERIODO DE
HTIMER  EQU     2DH             ;AMOSTRAGEM
TIMER   EQU     50H             ;ENDERECO CONTO

Press RETURN to Continue
CNTWRD  EQU     53H             ;ENDERECO CONTROL WORD
UPADDR  EQU     51H             ;ENDERECO CONT1
DWADDR  EQU     52H             ;ENDERECO CONT2
CONTUP  EQU     100H            ;CONTAGEM UP
CONTDW  EQU     102H            ;CONTAGEM DOWN
LTCHUP  EQU     40H
LTCHDW  EQU     80H
OLDUP   EQU     104H
OLDDW   EQU     106H
DESLUP  EQU     108H            ;DESLOCAMENTO POSITIVO
DESLDW  EQU     10AH            ;DESLOCAMENTO NEGATIVO
DESTOT  EQU     10CH            ;DESLOCAMENTO TOTAL NO PERIODO
DESLOC  EQU     122H            ;DESLOCAMENTO TOTAL
ERRO    EQU     10EH
WENTR   EQU     110H            ;ENTRADA DE COMANDO
X1KMS1  EQU     118H
X1K     EQU     112H
X2K     EQU     114H
UK      EQU     11AH
UKMS1   EQU     116H
DIFER   EQU     11CH
INICIA  EQU     8000H
RSTADD  EQU     0000H

Press RETURN to Continue
TRPADD  EQU     0024H           ;8085 = 0024H  Z80 = 0066H
INITIM  EQU     34H             ;INICIALIZACAO TIMER  MODO 2
INITUP  EQU     70H             ;INICIALIZACAO CONT1  MODO 0
INITDW  EQU     0B0H            ;INICIALIZACAO CONT2  MODO 0
ZERO    EQU     00H
DZERO   EQU     0000H
ATRITO  EQU     12EH
;
AMP     EQU     120H
SPR     EQU     124H
SP1     EQU     126H
CHAVE   EQU     128H
;
;R
HCOEF2  EQU     0A8H
LCOEF2  EQU     58H
;K1
HCOEF3  EQU     00FH
LCOEF3  EQU     0F8H

Press RETURN to Continue
;K2
HCOEF4  EQU     0EH
LCOEF4  EQU     08H
;LAMBDA1
HCOEF5  EQU     00CH
LCOEF5  EQU     58H
;LAMBDA2
HCOEF6  EQU     0FDH
LCOEF6  EQU     70H
;
;
;INICIALIZACAO
;
;       ORG RSTADD
;RSTADD JMP     INICIA
;
        ORG TRPADD
TRPADD  JMP     CONTR                   ;REINICIO DE PERIODO DE AMOSTRAGEM
;
        ORG     INICIA
INICIA  MVI     A,080H                  ;ZERAR D/A
        OUT     DACONV
;

Press RETURN to Continue
;INICIALIZAR CONTADORES
        MVI     A,INITIM
        OUT     CNTWRD                  ;CONTO INICIALIZADO
        MVI     A,INITUP
        OUT     CNTWRD                  ;CONT1 INICIALIZADO
        MVI     A,INITDW
        OUT     CNTWRD                  ;CONT2 INICIALIZADO
;
;PROGRAMAR CONTADORES UP E DOWN
;
        MVI     A,LTCHUP                ;LATCH C1
        OUT     CNTWRD
        MVI     A,LTCHDW                ;LATCH C2
        OUT     CNTWRD
        IN      UPADDR
        STA     CONTUP
        STA     OLDUP
        IN      UPADDR
        STA     CONTUP+1
        STA     OLDUP+1
        LHLD    OLDUP
        DCX     H
```

```
        OUT     UPADDR          ;CONTAGEM UP PROGRAMADA
;
        IN      DWADDR
        STA     CONTDW
        STA     OLDDW
        IN      DWADDR
        STA     CONTDW+1
        STA     OLDDW+1
        LHLD    OLDDW
        DCX     H
        MOV     A,L
        OUT     DWADDR
        MOV     A,H
        OUT     DWADDR          ;CONTAGEM DOWN PROGRAMADA
;
;ZERAR VARIAVEIS
;
        LXI     H,DZERO
        SHLD    X1K
        SHLD    X2K
        SHLD    WENTR

Press RETURN to Continue
        SHLD    ERRO
        SHLD    UK
        SHLD    UKMS1
        SHLD    DESLOC
;
;INICIALIZACAO DE FEEDFORWARD PARA COMPENSAR ATRITO DE COULOMB
        MVI     A,ATRCOU
        STA     ATRITO
;
;INICIALIZACAO DO GERADOR DE COMANDOS
;
        LXI     H,AMPLT
        SHLD    AMP
        LXI     H,SUBPER
        SHLD    SPR
        SHLD    SP1
        MVI     A,ZERO
        STA     CHAVE
        JMP     CONTR
;
;
;CONTROLADOR
;

Press RETURN to Continue
        ORG CONTR
;
;ESCREVER SAIDA DE CONTROLE
;
CONTR   LDA     ERRO+1          ;TESTAR SENTIDO DE VELOCIDADE
        CPI     ZERO
        JM      NEGER
        JNZ     POSER
        LDA     ERRO
        CPI     ZERO
        JZ      ZERERR
POSER   LDA     UKMS1+1
        ADI     OUTMAX
        JP      POSATR
        MVI     A,080H
        JMP     OUTPUT
```

```
        SUB     M
        JMP     OUTPUT
;
NEGER   LDA     UKMS1+1
        SBI     OUTMAX

Press RETURN to Continue
        JM      NEGATR
        MVI     A,7FH
        JMP     OUTPUT
NEGATR  LDA     UKMS1+1
        LXI     H,ATRITO
        ADD     M
        JMP     OUTPUT
;
ZERERR  LDA     UKMS1+1
;
OUTPUT  XRI     080H            ;COMPLEMENTE    A7
        OUT     DACONV          ;CONVERSOR D/A RESCRITO
;
        LXI     SP,9FFFH        ;STACK  POINTER CONHECIDO
;
;PROGRAMAR TIMER
;
        MVI     A,LTIMER
        OUT     TIMER
        MVI     A,HTIMER
        OUT     TIMER           ;TIMER PROGRAMADO
;
;LER DESLOCAMENTOS

Press RETURN to Continue
;
        MVI     A,LTCHUP
        OUT     CNTWRD          ;CONT1 LATCHED
        MVI     A,LTCHDW
        OUT     CNTWRD          ;CONT2 LATCHED
        IN      UPADDR
        STA     CONTUP
        IN      UPADDR
        STA     CONTUP+1        ;CONTADOR UP LIDO
        IN      DWADDR
        STA     CONTDW
        IN      DWADDR
        STA     CONTDW+1        ;CONTADOR DOWN LIDO
;
;CALCULAR DESLOCAMENTO TOTAL
;
        LHLD    CONTUP
        LDA     OLDUP
        SUB     L
        STA     DESLUP
        LDA     OLDUP+1
        SBB     H
        STA     DESLUP+1        ;DESLUP := CONTUP - OLDUP

Press RETURN to Continue
        LHLD    CONTUP
        SHLD    OLDUP           ;UPDATE OLDUP
;
        LHLD    CONTDW
        LDA     OLDDW
        SUB     L
        STA     DESLDW
```

```
        STA     DESLDW+1           ;DESLDW := CONTDW - OLDDW
        LHLD    CONTDW
        SHLD    OLDDW              ;UPDATE OLDDW
;
        LHLD    DESLDW
        LDA     DESLUP
        SUB     L
        STA     DESTOT
        LDA     DESLUP+1
        SBB     H
        STA     DESTOT+1           ;DESTOT := DESLUP - DESLDW
                                   DESLOCAMENTO TOTAL CALCULADO
;
;
;CALCULAR ERRO

Press RETURN to Continue     ε
        LHLD    DESTOT
        XCHG
        LHLD    DESLOC
        DAD     D
        SHLD    DESLOC             ;DESLOC := DESTOT + DESLOC
;
        LHLD    WENTR
        LDA     DESLOC
        SUB     L
        STA     ERRO
        LDA     DESLOC+1
        SBB     H
        STA     ERRO+1             ;ERRO := DESLOC - WENTR
                                   ERRO CALCULADO
;
;
;
;REDUZIR COMPRIMENTO DE ERRO
;
        STC
        CMC
        LDA     ERRO+1
        RAL

Press RETURN to Continue
        JC      REDNEG             ;TESTAR ER15
        CPI     ZERO               ;ER15=0
        JZ      POSZER             ;TESTAR ER14..ER8
        MVI     A,07FH             ;ER14..ER8 <> 0
        STA     DIFER+1            ;SATURACAO POSITIVA
        JMP     REDFIM
POSZER  LDA     ERRO
        RAL
        JNC     POSOK              ;TESTAR ER7
        MVI     A,07FH
        STA     DIFER+1            ;SATURACAO POSITIVA
        JMP     REDFIM
POSOK   RAR
        STA     DIFER+1            ;NUMERO OK
        JMP     REDFIM
;
REDNEG  CPI     0FEH               ;ER15=1
        JZ      NEGUM              ;TESTAR ER14..ER8
        MVI     A,80H              ;ER14..ER8 <> 0FEH
        STA     DIFER+1            ;SATURACAO NEGATIVA
        JMP     REDFIM
```

UNICAMP-FEC
LABORATÓRIO
DE
SIMULAÇÃO

```
        JC      NEGOK
        MVI     A,80H
        STA     DIFER+1
        JMP     REDFIM
NEGOK   RAR
        STA     DIFER+1
;
REDFIM  XRA     A                  ;ZERE ACCUMULADOR
        STA DIFER
;                                  FIM DE REDUCAO
;
;
;UKMS1 := R * DIFER
;                                  R = 1010.1000 0101 1
;MULTIPLICACAO DE UMA VARIAVEL DE 13 BITS POR UMA CONSTANTE
;DE 13BITS
;
;
MULT2   LHLD    DIFER              ;DE := DIFER
        XCHG
        LXI     H,DZERO            ;HL := 0000H
;
;C0 = 1

Press RETURN to Continue _
;
;       STC
;       CMC
        DAD     D                  ;HL := HL + DE
        MOV     A,H                ;A := H
        RAR
        MOV     H,A                ;H := A
        MOV     A,L                ;A := L
        RAR
        MOV     L,A                ;L := A
;
;C1 = 1
;
;       STC
;       CMC
        DAD     D                  ;HL := HL + DE
        MOV     A,H                ;A := H
        RAR
        MOV     H,A                ;H := A
        MOV     A,L                ;A := L
        RAR
        MOV     L,A                ;L := A
;

Press RETURN to Continue
;C2 = 0
;
        STC
        CMC
;       DAD     D                  ;HL := HL + DE
        MOV     A,H                ;A := H
        RAR
        MOV     H,A                ;H := A
        MOV     A,L                ;A := L
        RAR
        MOV     L,A                ;L := A
;
```

```
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A

Press RETURN to Continue
;
;C4 = 0
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A
;
;C5 = 0
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR

Press RETURN to Continue
        MOV     L,A         ;L := A
;
;C6 = 0
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A
;
;C7 = 0
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L

Press RETURN to Continue
        RAR
        MOV     L,A         ;L := A
;
;C8 = 1
;
        STC
```

```
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A
        MOV     A,B         ;A := B
        RAR
        MOV     B,A         ;B := A;
;C9 = 0
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H

Press RETURN to Continue
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A
        MOV     A,B         ;A := B
        RAR
        MOV     B,A         ;B := A;
;C10 = 1
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A
        MOV     A,B         ;A := B
        RAR
        MOV     B,A         ;B := A;
;C11 = 0

Press RETURN to Continue
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
        RAR
        MOV     L,A         ;L := A
        MOV     A,B         ;A := B
        RAR
        MOV     B,A         ;B := A;
;C12 = 1
;
        STC
        CMC
        DAD     D           ;HL := HL + DE
        MOV     A,H         ;A := H
        RAR
        MOV     H,A         ;H := A
        MOV     A,L         ;A := L
```

```
Press RETURN to Continue
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A
                                FIM DE MULTIPLICACAO SEM SINAL
;C12=1  CORRECAO DE SINAL
        MOV     A,E             ;CORRIGIR E
        ANI     0F8H
        MOV     E,A
        MOV     A,L             ;HL := HL - DE
        SUB     E
        MOV     L,A
        MOV     A,H
        SBB     D
        MOV     H,A             ;HL     CORRETO PARA C < 0
;
;V13    CORRECAO DE SINAL
        MOV     A,D
        RAL
        JNC     CORR2
        MOV     A,L             ;HL := HL - COEF**

Press RETURN to Continue
        SUI     LCOEF2
        MOV     L,A
        MOV     A,H
        SBI     HCOEF2
        MOV     H,A             ;HL     CORRETO PARA V < 0
;
;REDUCAO DE COMPRIMENTO DE PALAVRA
;
CORR2   MOV     A,H
        RAL
        JC      RNEG2           ;TESTAR H7
        ANI     0E0H            ;H7 = 0
        JZ      PZER2
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM2
;
PZER2   MOV     A,H
        ANI     08H             ;TESTAR H3
        JZ      RRND2
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM2
;
RNEG2   ANI     0E0H            ;H7=1

Press RETURN to Continue
        CPI     0E0H
        JZ      NZER2
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM2
;
NZER2   MOV     A,H
        ANI     08H             ;TESTAR H3
        JNZ     RRND2
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM2
;
;ARREDONDAR
;
RRND2   MOV     A,B             ;DESLOCAR HLB LEFT UMA VEZ
```



```
        RAL
        MOV     B,A
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
;
Press RETURN to Continue
        MOV     A,B             ;TESTAR SE SOMA
        RAL
        JNC     RFIM2
        MOV     A,L             ;B5=1
        CPI     0FFH
        JNZ     ROND2           ;TESTAR SE OVERFLOW E POSSIVEL
        MOV     A,H             ;L=0FFH
        CPI     0FH
        JZ      RFIM2
;
ROND2   INX     H               ;ARREDONDAR
;
RFIM2   STC                     ;DESLOCAR HL TRES VEZES
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L

Press RETURN to Continue
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
;
MFIM2   SHLD    UKMS1           ;UKMS1 := RESULTADO DA MULT
;
;RESUL := K1 * X1K
;                                       K1 = 0000.1111 1111 1
;MULTIPLICACAO DE UMA VARIAVEL DE 13 BITS POR UMA CONSTANTE
;DE 13BITS
;
;
MULT3   LHLD    X1K             ;DE := X1K
Press RETURN to Continue
        XCHG
        LXI     H,??ZERO        ;HL := 0000H
;
;CO = 1
```

```
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C1 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
Press RETURN to Continue
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C2 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C3 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
Press RETURN to Continue
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C4 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C5 = 1
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
Press RETURN to Continue
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C6 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C7 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
Press RETURN to Continue
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
;
;C8 = 1
;
;        STC
;        CMC
         DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A             ;L := A
         MOV     A,B             ;A := B
         RAR
         MOV     B,A             ;B := A;
;C9 = 0
;
Press RETURN to Continue
         STC
         CMC
;        DAD     D               ;HL := HL + DE
         MOV     A,H             ;A := H
         RAR
         MOV     H,A             ;H := A
         MOV     A,L             ;A := L
         RAR
         MOV     L,A
```

```
        RAL
        MOV     L,A
        MOV     A,H
        RAL
Press RETURN to Continue
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
;
MFIM3   XCHG                    ;DE := RESULTADO DA MULTIPLICACAO
;
;SOMA3          UKMS1 := UKMS1 + RESUL
;
SOMA3   LHLD    UKMS1
Press RETURN to Continue
        MOV     A,D
        ADI     ZERO
        JM      NXT13           ;DE POSITIVO OU NEGATIVO
        MVI     B,01H           ;POSITIVO
        JMP     NXT23
NXT13   MVI     B,02H   ;NEGATIVO
;
NXT23   MOV     A,H
        ADI     ZERO
        JM      NXT33           ;HL POSITIVO OU NEGATIVO
        MOV     A,B             ;POSITIVO
        CPI     02H
        JZ      OVOK3
        MVI     B,01H           ;OVERFLOW POSITIVO POSSIVEL
        JMP     NXT43
;
OVOK3   MVI     B,ZERO
        JMP     NXT43           ;OVERFLOW IMPOSSIVEL
NXT33   MOV     A,B             ;NEGATIVO
        CPI     01H
        JZ      OVOK3
        MVI     B,02H           ;OVERFLOW NEGATIVO POSSIVEL
;
Press RETURN to Continue
NXT43   DAD     D               ;HL := HL + DE
        MOV     A,H
        ADI     ZERO
        JM      NXT53           ;RESULTADO POSITIVO OU NEGATIVO
        MOV     A,B             ;POSITIVO
        CPI     02H

;
NXT53   MOV     A,B
        CPI     01H             ;NEGATIVO
        JNZ     SFIM3
        MVI     H,07FH          ;OVERFLOW POSITIVO
        MVI     L,0F8H
;
SFIM3   SHLD    UKMS1           ;FIM DE SOMA
;
;RESUL := K2 * X2K
;                                K2 = 0000.1110 0000 1
;MULTIPLICACAO DE UMA VARIAVEL DE 13 BITS POR UMA CONSTANTE
;DE 13BITS
Press RETURN to Continue
;
;
MULT4   LHLD    X2K             ;DE := X2K
        XCHG
        LXI     H,DZERO         ;HL := 0000H
;
;C0 = 1
;
;       STC
;       CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C1 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
Press RETURN to Continue
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C2 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C3 = 0
;
```

```
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C4 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C5 = 0
;
        STC

Press RETURN to Continue
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C6 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C7 = 1
;

Press RETURN to Continue
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C8 = 1
;
        STC
```

```
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR

Press RETURN to Continue
        MOV     B,A             ;B := A;
;C9 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C10 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A

Press RETURN to Continue
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C11 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C12 = 0
;
        STC

Press RETURN to Continue
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
```

```
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A
                                FIM DE MULTIPLICACAO SEM SINAL
;
;C1R=0  CORRECAO DE SINAL
;
        MOV     A,E             ;CORRIGIR E
        ANI     OF8H
        MOV     E,A
        MOV     A,L             ;HL := HL - DE
        SUB     E
        MOV     L,A
        MOV     A,H
        SBB     D

Press RETURN to Continue
        MOV     H,A             ;HL     CORRETO PARA C < 0
;
;V13    CORRECAO DE SINAL
        MOV     A,D
        RAL
        JNC     CORR4
        MOV     A,L             ;HL := HL - COEF**
        SUI     LCOEF4
        MOV     L,A
        MOV     A,H
        SBI     HCOEF4
        MOV     H,A             ;HL     CORRETO PARA V < 0
;
;REDUCAO DE COMPRIMENTO DE PALAVRA
;
CORR4   MOV     A,H
        RAL
        JC      RNEG4           ;TESTAR H7
        ANI     OEOH            ;H7 = 0
        JZ      PZER4
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM4
;

Press RETURN to Continue
PZER4   MOV     A,H
        ANI     08H             ;TESTAR H3
        JZ      RRND4
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM4
;
RNEG4   ANI     OEOH            ;H7=1
        CPI     OEOH
        JZ      NZER4
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM4
;
NZER4   MOV     A,H
        ANI     08H             ;TESTAR H3
        JNZ     RRND4
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM4
;
;ARREDONDAR
;
```

```
Press RETURN to Continue
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
;
        MOV     A,B             ;TESTAR SE SOMA
        RAL
        JNC     RFIM4
        MOV     A,L             ;B5=1
        CPI     OFFH
        JNZ     ROND4           ;TESTAR SE OVERFLOW E POSSIVEL
        MOV     A,H             ;L=OFFH
        CPI     OFH
        JZ      RFIM4
;
ROND4   INX     H               ;ARREDONDAR
;
RFIM4   STC                     ;DESLOCAR HL TRES VEZES
        CMC
        MOV     A,L
        RAL

Press RETURN to Continue
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
;
MFIM4   XCHG                    ;DE := RESULTADO DA MULTIPLICACAO
;

Press RETURN to Continue
;SOMA4          UKMS1 := UKMS1 + RESUL
;
SOMA4   LHLD    UKMS1
        MOV     A,D
        ADI     ZERO
        JM      NXT14           ;DE POSITIVO OU NEGATIVO
        MVI     B,01H           ;POSITIVO
        JMP     NXT24
NXT14   MVI     B,02H   ;NEGATIVO
;
NXT24   MOV     A,H
```

```
         JZ      OVOK4
         MVI     B,01H          ;OVERFLOW POSITIVO POSSIVEL
         JMP     NXT44

OVOK4    MVI     B,ZERO
         JMP     NXT44          ;OVERFLOW IMPOSSIVEL
NXT34    MOV     A,B            ;NEGATIVO
         CPI     01H

Press RETURN to Continue
         JZ      OVOK4
         MVI     B,02H          ;OVERFLOW NEGATIVO POSSIVEL

NXT44    DAD     D              ;HL := HL + DE
         MOV     A,H
         ADI     ZERO
         JM      NXT54          ;RESULTADO POSITIVO OU NEGATIVO
         MOV     A,B            ;POSITIVO
         CPI     02H
         JNZ     SFIM4
         MVI     H,80H          ;OVERFLOW NEGATIVO
         MVI     L,ZERO
         JMP     SFIM4
;
NXT54    MOV     A,B
         CPI     01H            ;NEGATIVO
         JNZ     SFIM4
         MVI     H,07FH         ;OVERFLOW POSITIVO
         MVI     L,0FBH

SFIM4    SHLD    UKMS1          ;FIM DE SOMA
;
;RESUL := LAMBDA1 * X1K

Press RETURN to Continue
;                               LAMBDA1 = 0000.1100 0101 1
;MULTIPLICACAO DE UMA VARIAVEL DE 13 BITS POR UMA CONSTANTE
;DE 13BITS
;
;
MULT5    LHLD    X1K            ;DE := X1K
         XCHG
         LXI     H,DZERO        ;HL := 0000H

;C0 = 1
;
;        STC
;        CMC
         DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     H,A            ;H := A
         MOV     A,L            ;A := L
         RAR
         MOV     L,A            ;L := A
;
;C1 = 1
;

Press RETURN to Continue
;        STC
;        CMC
         DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     A,L            ;A := L
         RAR
         MOV     L,A            ;L := A

;C2 = 0
;
         STC
         CMC
;        DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     H,A            ;H := A
         MOV     A,L            ;A := L
         RAR
         MOV     L,A            ;L := A
;
;C3 = 1

Press RETURN to Continue
;
;        STC
;        CMC
         DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     H,A            ;H := A
         MOV     A,L            ;A := L
         RAR
         MOV     L,A            ;L := A
;
;C4 = 0
;
         STC
         CMC
;        DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     H,A            ;H := A
         MOV     A,L            ;A := L
         RAR
         MOV     L,A            ;L := A
;

Press RETURN to Continue
;C5 = 0
;
         STC
         CMC
;        DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     H,A            ;H := A
         MOV     A,L            ;A := L
         RAR
         MOV     L,A            ;L := A
;
;C6 = 0
;
         STC
         CMC
;        DAD     D              ;HL := HL + DE
         MOV     A,H            ;A := H
         RAR
         MOV     H,A            ;H := A
```

```
        MOV     L,A             ;L := A
Press RETURN to Continue
;
;C7 = 1;
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C8 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
Press RETURN to Continue
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C9 = 0;
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C10 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
Press RETURN to Continue
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C11 = 0
;
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
Press RETURN to Continue
;C12 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A
;                               FIM DE MULTIPLICACAO SEM SINAL
;
;C12=0  CORRECAO DE SINAL
;
;       MOV     A,E             ;CORRIGIR E
;       ANI     0FBH
;       MOV     E,A
;       MOV     A,L             ;HL := HL - DE
;       SUB     E
Press RETURN to Continue
;       MOV     L,A
;       MOV     A,H
;       SBB     D
;       MOV     H,A             ;HL     CORRETO PARA C < 0
;
;V13    CORRECAO DE SINAL
        MOV     A,D
        RAL
        JNC     CORR5
        MOV     A,L             ;HL := HL - COEF**
        SUI     LCOEF5
        MOV     L,A
        MOV     A,H
        SBI     HCOEF5
        MOV     H,A             ;HL     CORRETO PARA V < 0
;
;REDUCAO DE COMPRIMENTO DE PALAVRA
;
CORR5   MOV     A,H
        RAL
        JC      RNEG5           ;TESTAR H7
        ANI     0F0H            ;H7 = 0
        JZ      PZER5
Press RETURN to Continue
        LXI     H,07FF0H        ;SATURACAO POSITIVA
        JMP     MFIM3
```

```
        ANI     08H                 ;TESTAR H3
        JZ      RRND5
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM5
;
RNEG5   ANI     0E0H                ;H7=1
        CPI     0E0H
        JZ      NZER5
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM5
;
NZER5   MOV     A,H
        ANI     08H                 ;TESTAR H3
        JNZ     RRND5
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM5
;
;ARREDONDAR
;
Press RETURN to Continue
RRND5   MOV     A,B                 ;DESLOCAR HLB LEFT UMA VEZ
        RAL
        MOV     B,A
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A

        MOV     A,B                 ;TESTAR SE SOMA
        RAL
        JNC     RFIM5
        MOV     A,L                 ;B5=1
        CPI     0FFH
        JNZ     ROND5               ;TESTAR SE OVERFLOW E POSSIVEL
        MOV     A,H                 ;L=0FFH
        CPI     0FH
        JZ      RFIM5
;
ROND5   INX     H                   ;ARREDONDAR
;
RFIM5   STC                         ;DESLOCAR HL TRES VEZES
Press RETURN to Continue
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
        STC
        CMC
        MOV     A,L
        RAL
        MOV     L,A
```

```
        MOV     A,H
        RAL
        MOV     H,A
Press RETURN to Continue
;
MFIM5   XCHG                        ;DE := RESULTADO DA MULTIPLICACAO
;
;SOMA5          X1KMS1 := DIFER + RESUL
;
SOMA5   LHLD    DIFER
        MOV     A,D
        ADI     ZERO
        JM      NXT15               ;DE POSITIVO OU NEGATIVO
        MVI     B,01H               ;POSITIVO
        JMP     NXT25
NXT15   MVI     B,02H   ;NEGATIVO
;
NXT25   MOV     A,H
        ADI     ZERO
        JM      NXT35               ;HL POSITIVO OU NEGATIVO
        MOV     A,B                 ;POSITIVO
        CPI     02H
        JZ      OVOK5
        MVI     B,01H               ;OVERFLOW POSITIVO POSSIVEL
        JMP     NXT45
;
OVOK5   MVI     B,ZERO
Press RETURN to Continue
        JMP     NXT45               ;OVERFLOW IMPOSSIVEL
NXT35   MOV     A,B                 ;NEGATIVO
        CPI     01H
        JZ      OVOK5
        MVI     B,02H               ;OVERFLOW NEGATIVO POSSIVEL
;
NXT45   DAD     D                   ;HL := HL + DE
        MOV     A,H
        ADI     ZERO
        JM      NXT55               ;RESULTADO POSITIVO OU NEGATIVO
        MOV     A,B                 ;POSITIVO
        CPI     02H
        JNZ     SFIM5
        MVI     H,80H               ;OVERFLOW NEGATIVO
        MVI     L,ZERO
        JMP     SFIM5
;
NXT55   MOV     A,B
        CPI     01H                 ;NEGATIVO
        JNZ     SFIM5
        MVI     H,07FH              ;OVERFLOW POSITIVO
        MVI     L,0F8H
;
Press RETURN to Continue
SFIM5   SHLD    X1KMS1              ;FIM DE SOMA
;
;RESUL := LAMBDA2 * X2K
;                                   LAMBDA2 = 1111.1101  0111 0
;MULTIPLICACAO DE UMA VARIAVEL DE 13 BITS POR UMA CONSTANTE
;DE 13BITS
;
;
MULT6   LHLD    X2K                 ;DE := X2K
        XCHG
```

```
;C0 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A

Press RETURN to Continue
;
;C1 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C2 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR

Press RETURN to Continue
        MOV     L,A             ;L := A
;
;C3 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C4 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L

Press RETURN to Continue
        RAR
        MOV     L,A             ;L := A
```



```
;
;C5 = 1
;
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C6 = 0
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A

Press RETURN to Continue
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C7 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
;
;C8 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR

Press RETURN to Continue
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;
;C9 = 1
;
        STC
        CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
```

```
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C10 = 1
;
Press RETURN to Continue
;       STC
;       CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C11 = 1
;
;       STC
;       CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
Press RETURN to Continue
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A;
;C12 = 1
;
;       STC
;       CMC
        DAD     D               ;HL := HL + DE
        MOV     A,H             ;A := H
        RAR
        MOV     H,A             ;H := A
        MOV     A,L             ;A := L
        RAR
        MOV     L,A             ;L := A
        MOV     A,B             ;A := B
        RAR
        MOV     B,A             ;B := A
                                FIM DE MULTIPLICACAO SEM SINAL
;
;C12=1  CORRECAO DE SINAL
;
        MOV     A,E             ;CORRIGIR E
        ANI     OFBH
Press RETURN to Continue
        MOV     E,A
        MOV     A,L             ;HL := HL - DE
        SUB     E
        MOV     L,A
        MOV     A,H
        SBB     D
        MOV     H,A             ;HL     CORRETO PARA C < 0
;
;V13    CORRECAO DE SINAL
```

```
        MOV     A,D
        RAL
        JNC     CORR6
        MOV     A,L             ;HL := HL - COEF6
        SUI     LCOEF6
        MOV     L,A
        MOV     A,H
        SBI     HCOEF6
        MOV     H,A             ;HL     CORRETO PARA V < 0
;
;REDUCAO DE COMPRIMENTO DE PALAVRA
;
CORR6   MOV     A,H
        RAL
Press RETURN to Continue
        JC      RNEG6           ;TESTAR H7
        ANI     0E0H            ;H7 = 0
        JZ      PZER6
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM6
;
PZER6   MOV     A,H
        ANI     08H             ;TESTAR H3
        JZ      RRND6
        LXI     H,07FF8H                ;SATURACAO POSITIVA
        JMP     MFIM6
;
RNEG6   ANI     0E0H            ;H7=1
        CPI     0E0H
        JZ      NZER6
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM6
;
NZER6   MOV     A,H
        ANI     08H             ;TESTAR H3
        JNZ     RRND6
        LXI     H,08000H                ;SATURACAO NEGATIVA
        JMP     MFIM6
Press RETURN to Continue
;
;ARREDONDAR
;
RRND6   MOV     A,B             ;DESLOCAR HLB LEFT UMA VEZ
        RAL
        MOV     B,A
        MOV     A,L
        RAL
        MOV     L,A
        MOV     A,H
        RAL
        MOV     H,A
;
        MOV     A,B             ;TESTAR SE SOMA
        RAL
        JNC     RFIM6
        MOV     A,L             ;B5=1
        CPI     0FFH
        JNZ     ROND6           ;TESTAR SE OVERFLOW E POSSIVEL
        MOV     A,H             ;L=0FFH
        CPI     0FH
        JZ      RFIM6
Press RETURN to Continue
```

```
ROND6   INX   H                ;ARREDONDAR
RFIM6   STC                    ;DESLOCAR HL TRES VEZES
        CMC
        MOV   A,L
        RAL
        MOV   L,A
        MOV   A,H
        RAL
        MOV   H,A
        STC
        CMC
        MOV   A,L
        RAL
        MOV   L,A
        MOV   A,H
        RAL
        MOV   H,A
        STC
        CMC
        MOV   A,L
        RAL
        MOV   L,A

Press RETURN to Continue
        MOV   A,H
        RAL
        MOV   H,A
;
MFIM6   XCHG                   ;DE := RESULTADO DA MULTIPLICACAO
;
;SOMA6        X1KMS1 := X1KMS1 + RESUL
;
SOMA6   LHLD  X1KMS1
        MOV   A,D
        ADI   ZERO
        JM    NXT16            ;DE POSITIVO OU NEGATIVO
        MVI   B,01H            ;POSITIVO
        JMP   NXT26
NXT16   MVI   B,02H     ;NEGATIVO
;
NXT26   MOV   A,H
        ADI   ZERO
        JM    NXT36            ;HL POSITIVO OU NEGATIVO
        MOV   A,B              ;POSITIVO
        CPI   02H
        JZ    OVOK6
        MVI   B,01H            ;OVERFLOW POSITIVO POSSIVEL

Press RETURN to Continue
        JMP   NXT46
;
OVOK6   MVI   B,ZERO
        JMP   NXT46            ;OVERFLOW IMPOSSIVEL
NXT36   MOV   A,B              ;NEGATIVO
        CPI   01H
        JZ    OVOK6
        MVI   B,02H            ;OVERFLOW NEGATIVO POSSIVEL
;
NXT46   DAD   D                ;HL := HL + DE
        MOV   A,H
        ADI   ZERO
        JM    NXT56            ;RESULTADO POSITIVO OU NEGATIVO
        MOV   A,B              ;POSITIVO
        CPI   02H
        MVI   H,80H            ;OVERFLOW NEGATIVO
        MVI   L,ZERO
        JMP   SFIM6
;
NXT56   MOV   A,B
        CPI   01H              ;NEGATIVO
        JNZ   SFIM6

Press RETURN to Continue
        MVI   H,07FH           ;OVERFLOW POSITIVO
        MVI   L,0F8H
;
SFIM6   SHLD  X1KMS1           ;FIM DE SOMA
;
;UPDATE VARIAVEIS DE ESTADO
;
        LHLD  X1K
        SHLD  X2K
        LHLD  X1KMS1
        SHLD  X1K
;
;
;GERACAO DA ENTRADA DE COMANDO
;
        LHLD  SPR
        DCX   H
        SHLD  SPR
        MOV   A,H
        ANA   A
        JNZ   CONTIN
        MOV   A,L

Press RETURN to Continue
        ANA   A
        JNZ   CONTIN
        LHLD  SP1
        SHLD  SPR
        LDA   CHAVE
        CPI   ZERO
        JZ    SOME3
        JMP   SUBT3
;
SUBT3   MVI   A,ZERO
        STA   CHAVE
        LHLD  AMP
        LDA   WENTR
        SUB   L
        STA   WENTR
        LDA   WENTR+1
        SBB   H
        STA   WENTR+1
        JMP   CONTIN
;
SOME3   MVI   A,01H
        STA   CHAVE
        LHLD  AMP

Press RETURN to Continue
        XCHG
        LHLD  WENTR
        DAD   D
        SHLD  WENTR
        JMP   CONTIN
```

APÊNDICE A8 - CARACTERÍSTICAS DO SERVOMOTOR

System Performance Specs	EC 500	EC 550 System		
Speed Range	1000:1 typically 5 to 5000 RPM down to .001 RPM with gearheads			
Torque Range	0 - 10 oz. in. up to 200 in.-lb. with suitable gear ratios			
System Input Sensitivity	4.6 V/KRPM	4.8 V/KRPM	4.5 V/KRPM	1.5 V/KRPM
Load Regulation	See Graph	See Graph		
System Bandwidth (Remote Inputs) (No Load)	10 Hz	12 Hz.	7 Hz.	7 Hz.
Controller Specs.	E-500 M or O Reversible	E-550 M or O Reversible	E-550 BV-001 Bi-Directional	E-550-BP-001 Bi-Directional
Input Power	115 or 230 V AC 50/60 Hz	115 or 230 VAC 50/60 Hz.	115 or 230 VAC 50/60 Hz.	115 or 230 VAC 50/60 Hz.
Input Impedance (Remote Control) Inputs)	10K	10K	10K Shunted by .01 MF	10K Shunted by .01 MF ± 1.5/1000
Open Loop Gain	200 V/V	750 V/V	750 V/V	750 V/V
Maximum Output Voltage (no load)	33V	33V	33V	33V
Output Current ± Peak	5 A	5 A	Amp for 100 msec	3 Amp for 100 msec
RMS	1.5A	2 A	2 A	2 A
Stall	Fuse Limited	Circuit Breaker Limited	1.5 A	1.5A
Motor Specs	E-500 MG	E-550 MG		
Rated				
Output	1/25 hp	1/20 hp		
Speed	4750 RPM	5000 RPM		
Torque	8 oz.-in.	10 oz. in		
Volts	28	28		
Armature Current	1.4 amps	2 amps		
Armature Resistance	4 ohms	2.7 ohms		
Intrinsic Data				
Voltage Constant	4.6 Volts/KRPM	4.8 Volts/KRPM		
Torque Constant	6.2 oz.-in/Amp	6.5 oz.-in./amp		
Regulation	140 RPM/oz.-in	90 RPM/oz.-in.		
Inertia	4×10^{3} oz.-in sec.^{2}	4×10^{3} oz.-in.-sec^2		
Derived Constants				
Mechanical Time Constant	60 msec	30 msec		
Electrical Time Constant	0.5 msec	0.4 msec		
Generator		Temperature Compensated		
Voltage Gradient	4.6 V/1000 RPM	4.6V/1000 RPM		
Ripple	3% RMS	2.5% RPM		

APÊNDICE A.9 - RESULTADOS DO PROJETO ASSISTIDO POR COMPUTADOR

```
PROGRAMA PARA O PROJETO DOS CONTROLADORES OTIMOS

RESISTENCIA = 4.[illegible]
INDUTANCIA = 2.0E-3
FRICCAO MOTOR-CARGA = 1.200[illegible]
INERCIA MOTOR-CARGA = 5.875[illegible]
CONSTANTES ELETRICA E MECANICA =4.392[illegible]
NUMERO DE PULSOS POR ROTACAO DO CODIFICADOR INCREMENTAL =[illegible]
PERIODO DE AMOSTRAGEM = 5.0[illegible]

DESEJA CONFERIR OU MUDAR ALGUM PARAMETRO ? (S OU N)?

PERIODO DE AMOSTRAGEM = 5.000000000E-0[illegible] ? (S OU N) = S
PERIODO DE AMOSTRAGEM = 5.000000000E-0[illegible]

POLO CONTINUO 1 =-6.1160125[illegible]    POLO CONTINUO 2 =-[illegible]55102569E+0[illegible]

MATRIZ DO SISTEMA DISCRETO
[ 1.000000000   [illegible]   5.210[illegible]E-0[illegible] ]
[ 0.000000000   9.[illegible]E-0[illegible]   2.1[illegible] ]
[ 0.000000000   -1.07[illegible]   [illegible] ]

VETOR DO SISTEMA DISCRETO
[ 1.416220858E-02 ]
[ 4.943543076E-01 ]
[ 2.4[illegible]566[illegible] ]

PROJETO DO REGULADOR

MATRIZ DE PONDERACAO [illegible]

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

```
MATRIZ DE PONDERACAO DOS ESTADOS

[   1.000000000        0.000000000        0.000000000        ]
[   0.000000000        1.000000000        0.000000000        ]
[   0.000000000        0.000000000        1.000000000        ]

PESO DA VARIAVEL DE ENTRADA = 20.0

CONVERGENCIA PARA : 0.001

VETOR DE GANHOS DO CONTROLADOR

[   2.068748593E-01 2.040592581E-01 4.550220191E-02 ]



PROJETO DO OBSERVADOR

MATRIZ DE PONDERACAO DOS ESTADOS

Q[ 1, 1]=   1.000000000          ? (S OU N) = S
Q[ 1, 2] = Q[ 2, 1] =   0.000000000          ? (S OU N) = S
Q[ 1, 3] = Q[ 3, 1] =   0.000000000          ? (S OU N) = S
Q[ 2, 2]=   1.000000000          ? (S OU N) = S
Q[ 2, 3] = Q[ 3, 2] =   0.000000000          ? (S OU N) = S
Q[ 3, 3]=   1.000000000          ? (S OU N) = S

MATRIZ DE PONDERACAO DOS ESTADOS

[   1.000000000        0.000000000        0.000000000        ]
[   0.000000000        1.000000000        0.000000000        ]
[   0.000000000        0.000000000        1.000000000        ]

PESO DA VARIAVEL DE ENTRADA = 1.0

CONVERGENCIA PARA : 0.001

VETOR DE REALIMENTACAO DO OBSERVADOR

[   6.186039512E-01   ]
[   2.734408370E-02   ]
[   6.097114413E-02   ]
```

```
TIPO DE CONTROLADOR  OTIMO 1> COM OBSERVADOR COM REALIMENTACAO -OTIMORE
                           2> COM OBSERVADOR SEM REALIMENTACAO-OTIMOOF
                           3> DINAMICO                        -DINAMICO
 = DINAMICO

QUAIS OS POLOS DO REGULADOR DINAMICO ?
(MULTIPLICIDADE 2 E PROXIMO DE ZERO) = 0.00
-------------------------------------------------------------------------

PARAMETROS DO CONTROLADOR DINAMICO
KD =  7.219577729
BETA1 = -2.000000000E-03
BETA2 =  1.000000029E-06
GAMA1 = -7.028098220
GAMA2 =  1.436338499E-02

-------------------------------------------------------------------------

CALCULAR OUTRO CONTROLADOR ? (SIM OU NAO) = NAO

PARAMETROS PARA SIMULACAO

MATRIZ DO SISTEMA DISCRETO
[  1.000000000      2.573569297E-03 5.673730671E-04 ]
[  0.000000000      9.979287505E-01 1.084404408E-01 ]
[  0.000000000     -7.826799511E-03 2.806378525E-01 ]

VETOR DO SISTEMA DISCRETO
[  1.067.20664E-04  ]
[  2.979625511E-03  ]
[  1.782632961E-03  ]

-------------------------------------------------------------------------

REFETIR CALCULO DOS CONTROLADORES PARA OUTRO PERIODO ? (SIM OU NAO)


PERIODO DE AMOSTRAGEM = 7.8E-2

DESEJA CONFERIR OU MUDAR ALGUM PARAMETRO ? (S OU N)N

PERIODO DE AMOSTRAGEM =  7.800000011E-02  ? (S OU N) =  N
PERIODO DE AMOSTRAGEM =  7.800000011E-02

POLO CONTINUO 1 =-6.118072509     POLO CONTINUO2= -1 995102569E+0

MATRIZ DO SISTEMA DISCRETO
[  1.000000000      4.271302080E-02 5 524227014E-03 ]
[  0.000000000      9.557472544E-01 2 131164938E-01 ]
[  0.000000000     -1 054774175E-02-2 347344735E-01 ]

VETOR DO SISTEMA DISCRETO
[  3.554184729E-0
[  7.952300695E-0
[  2.416524068E-0
```

```
PROJETO DO REGULADOR

MATRIZ DE PONDERACAO DOS ESTADOS

Q[ 1, 1]=  1.000000000       ? (S OU N) = S
Q[ 1, 2] = Q[ 2, 1] =  0.000000000       ? (S OU N) = S
Q[ 1, 3] = Q[ 3, 1] =  0.000000000       ? (S OU N) = S
Q[ 2, 2]=  1.000000000       ? (S OU N) = S
Q[ 2, 3] = Q[ 3, 2] =  0.000000000       ? (S OU N) = S
Q[ 3, 3]=  1.000000000       ? (S OU N) = S

MATRIZ DE PONDERACAO DOS ESTADOS

[   1.000000000        0.000000000        0.000000000        ]
[   0.000000000        1.000000000        0.000000000        ]
[   0.000000000        0.000000000        1.000000000        ]

PESO DA VARIAVEL DE ENTRADA = 1.0

CONVERGENCIA PARA : 0.001

VETOR DE GANHOS DO CONTROLADOR

[   6.476702570E-01 6.369965314E-01 1.420401558E-01 ]


PROJETO DO OBSERVADOR

MATRIZ DE PONDERACAO DOS ESTADOS

Q[ 1, 1]=  1.000000000       ? (S OU N) = S
Q[ 1, 2] = Q[ 2, 1] =  0.000000000       ? (S OU N) = S
Q[ 1, 3] = Q[ 3, 1] =  0.000000000       ? (S OU N) = S
Q[ 2, 2]=  1.000000000       ? (S OU N) = S
Q[ 2, 3] = Q[ 3, 2] =  0.000000000       ? (S OU N) = S
Q[ 3, 3]=  1.000000000       ? (S OU N) = S

MATRIZ DE PONDERACAO DOS ESTADOS

[   1.000000000        0.000000000        0.000000000        ]
[   0.000000000        1.000000000        0.000000000        ]
[   0.000000000        0.000000000        1.000000000        ]

PESO DA VARIAVEL DE ENTRADA = 1.0

CONVERGENCIA PARA : 0.001

VETOR DE REALIMENTACAO DO OBSERVADOR

[   6.160338522E-00   ]
[   4.103730527E-01   ]
[   2.147010166E-01   ]
```

```
TIPO DE CONTROLADOR  OTIMO 1) COM OBSERVADOR COM REALIMENTACAO -OTIMO
                           2) COM OBSERVADOR SEM REALIMENTACAO-OTIMOD
                           3) DINAMICO                              -DINAMI
 =     OTIMOREA

PONTO INICIAL PARA CALCULO DO AUTOVALOR REAL = 0.0
   VALOR DO POLINOMIO -7.124617695E-08
   VALOR DO POLINOMIO -4.085620820E-14
   VALOR DO POLINOMIO  0.000000000
AUTOVALOR =  1.720926359E-07   VALOR DO POLINOMIO =  0.000000000
AUTOVALOR E ACEITAVEL ? (S OU N) = S

AUTOVALOR 1 =  9.566948056E-01 + J  0.000000000
AUTOVALOR 2 =  4.327388346E-01 + J  0.000000000
AUTOVALOR 3 =  1.720926359E-07 + J  0.000000000

REPETIR CALCULO DOS AUTOVALORES ? (S OU N) N

--------------------------------------------------------------------------

PARAMETROS DO CONTROLADOR DIGITAL OTIMO COM OBSERVADOR COM REALIMENT

R = -4.280435740E-01

LAMBDA1 =  1.38943163[illegible]
LAMBDA2 = -4.139989652E-01
LAMBDA3 =  1.717203263E-07

KF1 = -1.866534454E-01
KF2 =  1.772096002E-01
KF3 = -1.571658948E-10

DF1 =  6.180339992E-01
DF2 = -2.540351977E-01
DF3 =  4.448060631E-10

J1=  0.000000000        J2 =  0.000000000
--------------------------------------------------------------------------
```

```
CALCULAR OUTRO CONTROLADOR ? (SIM OU NAO) = SIM
TIPO DE CONTROLADOR  OTIMO 1) COM OBSERVADOR COM REALIMENTACAO -OTIMOR
                           2) COM OBSERVADOR SEM REALIMENTACAO-OTIMODI
                           3) DINAMICO                              -DINAMIC
 = OTIMODIR


PONTO INICIAL PARA CALCULO DO AUTOVALOR REAL = 0.0
    VALOR DO POLINOMIO -2.735760062E-08
    VALOR DO POLINOMIO -2.264854967E-14
    VALOR DO POLINOMIO  0.000000000
AUTOVALOR =  1.710236966E-07   VALOR DO POLINOMIO =  0.000000000
AUTOVALOR E ACEITAVEL ? (S OU N) = S

AUTOVALOR 1 =  3.856998224E-01 + J  1.058274284E-01
AUTOVALOR 2 =  3.856998224E-01 + J -1.058274284E-01
AUTOVALOR 3 =  1.710236966E-07 + J  0.000000000

REPETIR CALCULO DOS AUTOVALORES ? (S OU N) N


-------------------------------------------------------------------------

PARAMETROS DO CONTROLADOR DIGITAL OTIMO COM OBSERVADOR SEM REALIMENTAC

D = -4.280435740E-01

LAMBDA1 =  7.713996648E-01
LAMBDA2 = -1.599638015E-01
LAMBDA3 =  1.707135349E-07

KF1 =  7.799201488E-02
KF2 =  6.847147583E-02
KF3 = -2.309466868E-10

J1=  0.000000000          J2 =  0.000000000
-------------------------------------------------------------------------


CALCULAR OUTRO CONTROLADOR ? (SIM OU NAO) = NAO

PARAMETROS PARA SIMULACAO

MATRIZ DO SISTEMA DISCRETO
[  1.000000000      5.564521606E-03  1.062793061E-01  ]
[  0.000000000      9.961482524E-01  1.902485862E-01  ]
[  0.000000000     -9.398090124E-03  1.406594310E-01  ]

VETOR DO SISTEMA DISCRETO
[  3.367329583E-04 ]
[  6.061612401E-02 ]
[  2.141192734E-01 ]

-------------------------------------------------------------------------

REPETIR CALCULO DOS CONTROLADORES PARA OUTRO PERIODO  ? (SIM OU NAO) =

FIM DE PROGRAMA
```

PROGRAMA PARA SIMULAR OS CONTROLADORES OTIMOS

TIPO DO CONTROLADOR ? (OTIMOREA,OTIMODIR,DINAMICO) = DINAMICO

PARAMATROS DO SISTEMA A SER SIMULADO

KD = 7.219577
BETA1 = -2.0E-5
BETA2 = 1.0E-6
GAMA1 = -7.028098
GAMA2 = 1.436338E-2

MATRIZ DE SIMULACAO [1, 1] = 1.0
MATRIZ DE SIMULACAO [1, 2] = 3.573569E-3
MATRIZ DE SIMULACAO [1, 3] = 5.673730E-4
MATRIZ DE SIMULACAO [2, 1] = 0.0
MATRIZ DE SIMULACAO [2, 2] = 5.979287E-4
MATRIZ DE SIMULACAO [2, 3] = 1.584404E-1
MATRIZ DE SIMULACAO [3, 1] = 0.0
MATRIZ DE SIMULACAO [3, 2] = -7.026799E-3
MATRIZ DE SIMULACAO [3, 3] = 2.806378E-1

VETOR DE SIMULACAO [1] = 1.067120E-4
VETOR DE SIMULACAO [2] = 2.979963E-2
VETOR DE SIMULACAO [3] = 1.708632E-1

TIPO DE ENTRADA: DEGRAU RAMPA OU PARABOLA ? = DEGRAU
AMPLITUDE DA ENTRADA (EM NUMERO DE PULSOS) = [illegible]
DEVE TRUNCAR A SAIDA ? (S OU N) = S
DEVE IMPRIMIR OS RESULTADOS DA SIMULACAO ? (S OU N) = [illegible]

PERIODO	SUBPERIODO	ENTRADA	CONTROLADOR	SAIDA
0	0	1.000000000E+02	7.21[illegible]E+01	0.0[illegible]
[illegible]	4	1.000000000E+02	[illegible]	[illegible]
1	0	1.000000000E+02	-[illegible]	[illegible]
[illegible]	[illegible]	1.000000000E+02	-6.7[illegible]	1.2[illegible]
[illegible]	[illegible]	1.000000000E+02	-2.[illegible]	[illegible]
[illegible]	4	1.000000000E+02	-2.[illegible]	6.10[illegible]
[illegible]	0	1.000000000E+02	-1.7695052[illegible]E+02	7.2062[illegible]
[illegible]	[illegible]	1.000000000E+02	-1.7695052[illegible]E+02	8.1[illegible]E+01
[illegible]	0	1.000000000E+02	-1.0252140[illegible]E+02	8.72[illegible]
[illegible]	4	1.000000000E+02	-1.0252140[illegible]E+02	9.1721[illegible]
[illegible]	0	1.000000000E+02	-6.2094365[illegible]E+01	[illegible]
[illegible]	4	1.000000000E+02	-6.20943[illegible]E+01	[illegible]
[illegible]	0	1.000000000E+02	-2.0000[illegible]	[illegible]
[illegible]	4	1.000000000E+02	-2.0000[illegible]E+01	[illegible]
[illegible]	[illegible]	1.000000000E+02	-1.3662[illegible]	[illegible]
[illegible]	[illegible]	1.000000000E+02	-1.3662[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	1.[illegible]E-01	[illegible]
[illegible]	[illegible]	[illegible]	1.[illegible]E-01	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]E-01	[illegible]7610[illegible]
10	[illegible]	[illegible]	[illegible]E-01	9.984020[illegible]
10	[illegible]	[illegible]	[illegible]E-02	9.9967[illegible]
11	[illegible]	[illegible]	[illegible]E-02	9.99[illegible]
11	[illegible]	[illegible]	[illegible]E-01	1.0001[illegible]
[illegible]	[illegible]	[illegible]E+02	-7.0277[illegible]	1.000463739E+02
[illegible]	[illegible]	[illegible]	-7.[illegible]	1.000381439E+02

[illegible]	0	1.000000000E+02	7.219891846	9.994826078E+01
[illegible]	4	1.000000000E+02	7.219891846	9.985170841E+01
[illegible]	0	1.000000000E+02	1.914866387E-01	9.983903169E+01
[illegible]	4	1.000000000E+02	1.914866387E-01	9.987506628E+01
[illegible]	0	1.000000000E+02	1.917862445E-01	9.991448521E+01
[illegible]	4	1.000000000E+02	1.917862445E-01	9.995559096E+01
[illegible]	0	1.000000000E+02	1.917938888E-01	9.999835252E+01
[illegible]	4	1.000000000E+02	1.917938888E-01	1.000427439E+02
[illegible]	0	1.000000000E+02	-7.027783215	1.000687379E+02
[illegible]	4	1.000000000E+02	-7.027783215	1.000980943E+02
[illegible]	0	1.000000000E+02	3.148417472E-04	1.000255256E+02
[illegible]	4	1.000000000E+02	3.148417472E-04	9.990779628E+01
[illegible]	0	1.000000000E+02	7.219584703	9.975020116E+01
[illegible]	4	1.000000000E+02	7.219584703	9.971261262E+01
[illegible]	0	1.000000000E+02	1.914790704E-01	9.972026348E+01
[illegible]	4	1.000000000E+02	1.914790704E-01	9.977612761E+01
[illegible]	0	1.000000000E+02	1.917862012E-01	9.983546972E+01
[illegible]	4	1.000000000E+02	1.917862012E-01	9.989600181E+01
[illegible]	0	1.000000000E+02	1.917938888E-01	9.995789527E+01
[illegible]	4	1.000000000E+02	1.917938888E-01	1.000211268E+02
[illegible]	0	1.000000000E+02	-7.027783215	1.000856757E+02
[illegible]	4	1.000000000E+02	-7.027783215	1.001133039E+02
[illegible]	0	1.000000000E+02	3.148417472E-04	1.000587329E+02
[illegible]	4	1.000000000E-02	3.148417472E-04	9.995872735E+01
[illegible]	0	1.000000000E+02	7.219584703	9.985858440E+01
[illegible]	4	1.000000000E+02	7.219584703	9.979818582E+01
[illegible]	0	1.000000000E+02	1.914790704E-01	9.982276439E+01
[illegible]	4	1.000000000E+02	1.914790704E-01	9.985549875E+01
[illegible]	0	1.000000000E+02	1.917862012E-01	9.997106194E+01
[illegible]	4	1.000000000E+02	1.917862012E-01	1.000477626E+02
[illegible]	0	1.000000000E+02	-7.027783155	1.001255780E+02
[illegible]	4	1.000000000E+02	-7.027783155	1.001662761E+02
[illegible]	0	1.000000000E+02	3.148415118E-04	1.00129[illegible]E+02
[illegible]	4	1.000000000E+02	3.148415118E-04	1.000722290E+02
[illegible]	0	1.000000000E+02	7.219584703	[illegible]
[illegible]	4	1.000000000E+02	7.219584703	[illegible]
[illegible]	0	1.000000000E+02	1.914790704E-01	[illegible]E+01
[illegible]	4	1.000000000E+02	1.914790704E-01	1.00[illegible]E+02
[illegible]	0	1.000000000E+02	-7.027790784	1.0011[illegible]
[illegible]	4	1.000000000E+02	-7.027790784	1.001601338E+02
[illegible]	0	1.000000000E+02	3.148266106E-04	1.001276314E+02
[illegible]	4	1.000000000E+02	3.148266106E-04	1.000493615E+02
[illegible]	0	1.000000000E+02	7.219584703	9.997062802E+01
[illegible]	4	1.000000000E+02	7.219584703	9.993131279E+01
[illegible]	0	1.000000000E+02	1.914790704E-01	9.997665405E+01
[illegible]	4	1.000000000E+02	1.914790704E-01	1.00069835[illegible]E+02
[illegible]	0	1.000000000E+02	-7.027790784	1.00165[illegible]E+02
[illegible]	4	1.000000000E+02	-7.027790784	1.00223[illegible]
[illegible]	0	1.000000000E+02	3.148266106E-04	1.00199[illegible]E+02
[illegible]	4	[illegible]	[illegible]E-04	1.00129[illegible]E+02
[illegible]	0	[illegible]	[illegible]E-06	1.0005[illegible]E+02
[illegible]	4	[illegible]	[illegible]E-05	9.998650008E+01
[illegible]	0	[illegible]	7.2[illegible]	9.9919[illegible]E+01
[illegible]	4	[illegible]	[illegible]	9.98501[illegible]E-01
[illegible]	0	[illegible]	1.91[illegible]E-01	9.9945[illegible]E+01
[illegible]	4	[illegible]	[illegible]E-01	1.0004[illegible]527E+02

TIPO DO CONTROLADOR ? (OTIMOREA,OTIMODIN,DINAMICO) = OTIMOREA

PARAMATROS DO SISTEMA A SER SIMULADO

R = -4.280435E-1
LAMBDA1 = 1.389431
LAMBDA2 = -4.139989E-1
LAMBDA3 = 1.717283E-7
KF1 = -1.865534E-1
KF2 = 1.772096E-1
KF3 = -1.571698E-10
DF1 = 6.180339E-1
DF2 = -2.540351E-1
DF3 = 4.4488N8N06E-10
J1 = 0.0
J2 = 0.0
MATRIZDESIMULACAO [1, 1] = 1.0
MATRIZDESIMULACAO [1, 2] = 5.564821E-3
MATRIZDESIMULACAO [1, 3] = 1.0627930E-3
MATRIZDESIMULACAO [2, 1] = 0.0
MATRIZDESIMULACAO [2, 2] = 9.961482E-1
MATRIZDESIMULACAO [2, 3] = 1.902480E-1
MATRIZDESIMULACAO [3, 1] = 0.0
MATRIZDESIMULACAO [3, 2] = -9.398090E-3
MATRIZDESIMULACAO [3, 3] = 1.408594E-1

VETORDESIMULACAO [1] = 3.387338E-4
VETORDESIMULACAO [2] = 6.063612E-2
VETORDESIMULACAO [3] = 2.141192E-1
TIPO DE ENTRADA: DEGRAU RAMPA OU PARABOLA ? = DEGRAU
AMPLITUDE DA ENTRADA (EM NUMERO DE PULSOS) = 100
DEVE TRUNCAR A SAIDA ? (S OU N) = S
DEVE IMPRIMIR OS RESULTADOS DA SIMULACAO ? (S OU N) = S

PERIODO	SUBPERIODO	ENTRADA	CONTROLADOR	SAIDA
0	0	1.000000000E+02	0.000000000	0.000000000
0	4	1.000000000E+02	0.000000000	0.000000000
1	0	1.000000000E+02	4.280434966E+01	0.000000000
1	4	1.000000000E+02	4.280434966E+01	6.20728820[illegible]E-01
2	0	1.000000000E+02	3.500515073E+01	2.452446[illegible]
2	4	1.000000000E+02	3.500515073E+01	5.360036[illegible]
3	0	1.000000000E+02	2.171365916E+01	9.20791[illegible]
3	4	1.000000000E+02	2.171365916E+01	1.377958411E+01
4	0	1.000000000E+02	1.012008368E+01	1.887132793E+01
4	4	1.000000000E+02	1.012008368E+01	2.429955005E+01
5	0	1.000000000E+02	2.307865738	2.989207[illegible]
5	4	1.000000000E+02	2.307865738	3.552253[illegible]
6	0	1.000000000E+02	-2.898737847	4.110462[illegible]
6	4	1.000000000E+02	-2.898737847	4.654009[illegible]
7	0	1.000000000E+02	-5.883866356	5.176276[illegible]
7	4	1.000000000E+02	-5.883866356	5.671595[illegible]
8	0	1.000000000E+02	-7.572206280	6.142276644E+01
8	4	1.000000000E+02	-7.572206280	6.580621683E+01
9	0	1.000000000E+02	-8.187466526	6.986072[illegible]E+01
9	4	1.000000000E+02	-8.187466526	7.360375227E+01
10	0	1.000000000E+02	-7.734511044	7.70263[illegible]E+01
10	4	1.000000000E+02	-7.734511044	8.012575264E+01
11	0	1.000000000E+02	-7.445556640	8.295520052E+01
11	4	1.000000000E+02	-7.445556640	8.545993924E+01
12	0	1.000000000E+02	-6.597897529	8.771077752E+01
12	4	1.000000000E+02	-6.597897529	8.970651865E+01

13	0	1.000000000E+02	-5.681348025	9.146514692E+01
13	4	1.000000000E+02	-5.681348025	9.300567030E+01
14	0	1.000000000E+02	-4.781168580	9.434618711E+01
14	4	1.000000000E+02	-4.781168580	9.550460934E+01
15	0	1.000000000E+02	-3.884778410	9.649795055E+01
15	4	1.000000000E+02	-3.884778410	9.734323620E+01
16	0	1.000000000E+02	-2.968833714	9.805662393E+01
16	4	1.000000000E+02	-2.968833714	9.865465641E+01
17	0	1.000000000E+02	-2.017339199	9.915298700E+01
17	4	1.000000000E+02	-2.017339199	9.956789493E+01
18	0	1.000000000E+02	-1.877777695	9.991480350E+01
18	4	1.000000000E+02	-1.877777695	1.001974895E+02
19	0	1.000000000E+02	-1.105955883	1.004194378E+02
19	4	1.000000000E+02	-1.105955883	1.005932882E+02
20	0	1.000000000E+02	-9.609414458E-01	1.007310837E+02
20	4	1.000000000E+02	-9.609414458E-01	1.008359074E+02
21	0	1.000000000E+02	-5.643565416E-01	1.009105727E+02
21	4	1.000000000E+02	-5.643565416E-01	1.009615689E+02
22	0	1.000000000E+02	-2.816283047E-01	1.009950561E+02
22	4	1.000000000E+02	-2.816283047E-01	1.010156050E+02
23	0	1.000000000E+02	-1.269712120E-01	1.010275185E+02
23	4	1.000000000E+02	-1.269712120E-01	1.010332852E+02
24	0	1.000000000E+02	-4.609386491E-01	1.010352402E+02
24	4	1.000000000E+02	-4.809386491E-01	1.010283589E+02
25	0	1.000000000E+02	-3.705439835E-01	1.010078459E+02
25	4	1.000000000E+02	-3.705439835E-01	1.009755745E+02
26	0	1.000000000E+02	-2.287635922E-01	1.009331774E+02
26	4	1.000000000E+02	-2.287635922E-01	1.008835762E+02
27	0	1.000000000E+02	2.909900128E-01	1.008283406E+02
27	4	1.000000000E+02	2.909900128E-01	1.007751516E+02
28	0	1.000000000E+02	2.610634058E-01	1.007151071E+02
28	4	1.000000000E+02	2.610634058E-01	1.006974413E+02
29	0	1.000000000E+02	1.548362150E-01	1.006713062E+02
29	4	1.000000000E+02	1.548362150E-01	1.006517559E+02
30	0	1.000000000E+02	7.767982780E-02	1.006371408E+02
30	4	1.000000000E+02	7.767982780E-02	1.006262153E+02
31	0	1.000000000E+02	3.515396296E-02	1.006177946E+02
31	4	1.000000000E+02	3.515396296E-02	1.006111806E+02
32	0	1.000000000E+02	1.469177618E-02	1.006057425E+02
32	4	1.000000000E+02	1.469177618E-02	1.006011566E+02
33	0	1.000000000E+02	5.709862291E-03	1.005970925E+02
33	4	1.000000000E+02	5.709862291E-03	1.005934238E+02
34	0	1.000000000E+02	2.054430454E-03	1.005900684E+02
34	4	1.000000000E+02	2.054430454E-03	1.005867861E+02
35	0	1.000000000E+02	6.714145898E-04	1.005837021E+02
35	4	1.000000000E+02	6.714145898E-04	1.005807215E+02
36	0	1.000000000E+02	1.852941430E-04	1.005778342E+02
36	4	1.000000000E+02	1.852941430E-04	1.005750266E+02
37	0	1.000000000E+02	3.861930996E-05	1.005722925E+02
37	4	1.000000000E+02	3.861930996E-05	1.005696266E+02
38	0	1.000000000E+02	-4.652072426E-07	1.005670219E+02
38	4	1.000000000E+02	-4.652072426E-07	1.005644822E+02
39	0	1.000000000E+02	-6.555142641E-06	1.005619972E+02
39	4	1.000000000E+02	-6.555142641E-06	1.005595756E+02
40	0	1.000000000E+02	-4.978359818E-06	1.005572080E+02
41	4	1.000000000E+02	-4.978359818E-06	1.005546961E+02

TIPO DO CONTROLADOR ? (OTIMOREA,OTIMODIR,DINAMICO) = OTIMODIR

PARAMATROS DO SISTEMA A SER SIMULADO

R = -4.280435E-1
LAMBDA1 = 7.713996E-1
LAMBDA2 = -1.599638E-1
LAMBDA3 = 0.0
KF1 = 7.799201E-2
KF2 = 6.847147E-2
KF3 = 0.0
J1 = 0.0
J2 = 0.0
MATRIZDESIMULACAO [1, 1] = 1.0
MATRIZDESIMULACAO [1, 2] = 5.564821E-3
MATRIZDESIMULACAO [1, 3] = 1.062793E-3
MATRIZDESIMULACAO [2, 1] = 0.0
MATRIZDESIMULACAO [2, 2] = 9.961482E-1
MATRIZDESIMULACAO [2, 3] = 1.902485E-1
MATRIZDESIMULACAO [3, 1] = 0.0
MATRIZDESIMULACAO [3, 2] = -9.398090E-3
MATRIZDESIMULACAO [3, 3] = 1.408594E-1

VETORDESIMULACAO [1] = 3.387339E-4
VETORDESIMULACAO [2] = 6.063612E-2
VETORDESIMULACAO [3] = 2.141119E-1
TIPO DE ENTRADA: DEGRAU RAMPA OU PARABOLA ? = DEGRAU
AMPLITUDE DA ENTRADA (EM NUMERO DE PULSOS) = 100
DEVE TRUNCAR A SAIDA ? (S OU N) = S
DEVE IMPRIMIR OS RESULTADOS DA SIMULACAO ? (S OU N) = S

PERIODO	SUBPERIODO	ENTRADA	CONTROLADOR	SAIDA
0	0	1.000000000E+02	0.000000000	0.000000000
0	4	1.000000000E+02	0.000000000	0.000000000
1	0	1.000000000E+02	4.280434966E+01	0.000000000
1	4	1.000000000E+02	4.280434966E+01	6.207224120E-01
2	0	1.000000000E+02	3.500514835E+01	2.452415674
2	4	1.000000000E+02	3.500514835E+01	5.355964605
3	0	1.000000000E+02	2.171365737E+01	9.207791666
3	4	1.000000000E+02	2.171365737E+01	1.377978995E+01
4	0	1.000000000E+02	1.012007668E+01	1.887105867E+01
4	4	1.000000000E+02	1.012007668E+01	2.429920017E+01
5	0	1.000000000E+02	2.307851314	2.989243865E+01
5	4	1.000000000E+02	2.307851314	3.553182125E+01
6	0	1.000000000E+02	-2.898760557	4.110586225E+01
6	4	1.000000000E+02	-2.898760557	4.653941035E+01
7	0	1.000000000E+02	-5.883913278	5.176219761E+01
7	4	1.000000000E+02	-5.883913278	5.673510670E+01
8	0	1.000000000E+02	-7.572237253	6.142184795E+01
8	4	1.000000000E+02	-7.572237253	6.580422997E+01
9	0	1.000000000E+02	-8.187497615	6.986556649E+01
9	4	1.000000000E+02	-8.187497615	7.360420021E+01
10	0	1.000000000E+02	-7.734540581	7.701920508E+01
10	4	1.000000000E+02	-7.734540581	8.012452662E+01
11	0	1.000000000E+02	-7.445581436	8.293291560E+01
11	4	1.000000000E+02	-7.445581436	8.545650268E+01
12	0	1.000000000E+02	-6.597919404	8.770939626E+01
12	4	1.000000000E+02	-6.597919404	8.970508613E+01

13	0	1.000000000E+02	-5.681366205	9.146366834E+01
13	4	1.000000000E+02	-5.681366205	9.300415635E+01
14	0	1.000000000E+02	-4.781163660	9.434464097E+01
14	4	1.000000000E+02	-4.781163660	9.550303459E+01
15	0	1.000000000E+02	-3.884790539	9.649634957E+01
15	4	1.000000000E+02	-3.884790539	9.734160900E+01
16	0	1.000000000E+02	-2.968843191	9.805497407E+01
16	4	1.000000000E+02	-2.968843191	9.865298628E+01
17	0	1.000000000E+02	-2.017346471	9.915129899E+01
17	4	1.000000000E+02	-2.017346471	9.956619024E+01
18	0	1.000000000E+02	-1.877783134	9.991308212E+01
18	4	1.000000000E+02	-1.877783134	1.001957535E+02
19	0	1.000000000E+02	-1.105959632	1.004176855E+02
19	4	1.000000000E+02	-1.105959632	1.005915239E+02
20	0	1.000000000E+02	-9.609442472E-01	1.007293075E+02
20	4	1.000000000E+02	-9.609442472E-01	1.008341163E+02
21	0	1.000000000E+02	-5.643584668E-01	1.009087711E+02
21	4	1.000000000E+02	-5.643584668E-01	1.009597569E+02
22	0	1.000000000E+02	-2.816296041E-01	1.009932339E+02
22	4	1.000000000E+02	-2.816296041E-01	1.010137706E+02
23	0	1.000000000E+02	-1.269720360E-01	1.010256737E+02
23	4	1.000000000E+02	-1.269720360E-01	1.010314345E+02
24	0	1.000000000E+02	-4.809391319E-01	1.010333806E+02
24	4	1.000000000E+02	-4.809391319E-01	1.010264888E+02
25	0	1.000000000E+02	-3.705442279E-01	1.010059684E+02
25	4	1.000000000E+02	-3.705442279E-01	1.009736880E+02
26	0	1.000000000E+02	-2.287637174E-01	1.009314849E+02
26	4	1.000000000E+02	-2.287637174E-01	1.008816748E+02
27	0	1.000000000E+02	2.909900218E-01	1.008264318E+02
27	4	1.000000000E+02	2.909900218E-01	1.007734373E+02
28	0	1.000000000E+02	2.610635042E-01	1.007299870E+02
28	4	1.000000000E+02	2.610635042E-01	1.006955116E+02
29	0	1.000000000E+02	1.548364132E-01	1.006693691E+02
29	4	1.000000000E+02	1.548364132E-01	1.006498128E+02
30	0	1.000000000E+02	7.768003642E-02	1.006351917E+02
30	4	1.000000000E+02	7.768003642E-02	1.006242573E+02
31	0	1.000000000E+02	3.515412668E-02	1.006158292E+02
31	4	1.000000000E+02	3.515412668E-02	1.006092101E+02
32	0	1.000000000E+02	1.469188570E-02	1.006037637E+02
32	4	1.000000000E+02	1.469188570E-02	1.005991637E+02
33	0	1.000000000E+02	5.709926843E-03	1.005951017E+02
33	4	1.000000000E+02	5.709926843E-03	1.005914285E+02
34	0	1.000000000E+02	2.054465442E-03	1.005880087E+02
34	4	1.000000000E+02	2.054465442E-03	1.005847811E+02
35	0	1.000000000E+02	6.714321732E-04	1.005816906E+02
35	4	1.000000000E+02	6.714321732E-04	1.005787044E+02
36	0	1.000000000E+02	1.893024161E-04	1.005758091E+02
36	4	1.000000000E+02	1.893024161E-04	1.005729988E+02
37	0	1.000000000E+02	3.862256432E-05	1.005702599E+02
37	4	1.000000000E+02	3.862256432E-05	1.005675897E+02
38	0	1.000000000E+02	4.877949535E-07	1.005649805E+02
38	4	1.000000000E+02	-4.877949535E-07	1.005624324E+02
39	0	1.000000000E+02	-6.554561018E-06	1.005599454E+02
39	4	1.000000000E+02	-6.554561018E-06	1.005575165E+02
40	0	1.000000000E+02	-4.978156149E-06	1.005551457E+02
40	4	1.000000000E+02	-4.978156149E-06	1.005528315E+02

```
[illegible] E VARIAVEIS

NUMEROS DE NOS DO SISTEMA = [illegible]

NUMERO TOTAL DE RAMOS DE COEFICIENTES NO SISTEMA = [illegible]

MATRIZ DE COEFICIENTES

COEFICIENTE DO NO = [illegible]
            AO NO = [illegible]
 = [illegible]

COEFICIENTE DO NO = [illegible]
            AO NO = [illegible]
 = [illegible]

COEFICIENTE DO NO = [illegible]
            AO NO = [illegible]
 = [illegible]

COEFICIENTE DO NO = [illegible]
            AO NO = [illegible]
 = [illegible]

COEFICIENTE DO NO = [illegible]
            AO NO = [illegible]
 = [illegible]

[illegible]
            [illegible]
 = [illegible]

[illegible]
            [illegible]
 = [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]
```

```
NO DE ENTRADA = [illegible]

NO DE SAIDA = [illegible]

NUMERO DE COEFICIENTES QUANTIZADOS = [illegible]

COEFICIENTE 1   NO DE ENTRADA = [illegible]
                        NO DE SAIDA = [illegible]
COEFICIENTE 2   NO DE ENTRADA = [illegible]
                        NO DE SAIDA = [illegible]
COEFICIENTE 3   NO DE ENTRADA = [illegible]
                        NO DE SAIDA = [illegible]
COEFICIENTE 4   NO DE ENTRADA = 4
                        NO DE SAIDA = [illegible]
COEFICIENTE 5   NO DE ENTRADA = 5
                        NO DE SAIDA = [illegible]

NUMERO DE NOS COM VARIAVEIS QUANTIZADAS = [illegible]

NO QUANTIZADO [illegible] = [illegible]
                    NUMERO DE FONTES DE RUIDO = [illegible]

NO QUANTIZADO [illegible] = [illegible]
                    NUMERO DE FONTES DE RUIDO = [illegible]

[illegible]

[illegible]

[illegible]
[illegible] EFEITO DA QUANTIZACAO DE VARIAVEIS [illegible]

[illegible] = [illegible]

[illegible] ESTATISTICO DOS COEFICIENTES [illegible]

COMPRIMENTO ESTATISTICO DE VARIAVEIS = [illegible]
```

[illegible]

NUMERO TOTAL DE RAMOS DE COEFICIENTES N. SISTEMA = [illegible]

MATRIZ DE COEFICIENTES

COEFICIENTE DO NO = 1
            NO NO = 2
 = 1.0

COEFICIENTE DO NO = 2
            NO NO = 1
 = 1.0

COEFICIENTE DO NO = 4
            NO NO = 1
 = [illegible]

COEFICIENTE DO NO = 2
            NO NO = 1
 = [illegible]

COEFICIENTE DO NO = 4
            NO NO = 2
 = [illegible]

COEFICIENTE DO NO = [illegible]
            NO NO = [illegible]
 = [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
NO DE ENTRADA = 1

NO DE SAIDA = 6

NUMERO DE COEFICIENTES QUANTIZADOS = 7

COEFICIENTE 1  NO DE ENTRADA = 2
                   NO DE SAIDA = 6
COEFICIENTE 2  NO DE ENTRADA = 4
                   NO DE SAIDA = 1
COEFICIENTE 3  NO DE ENTRADA = 5
                   NO DE SAIDA = 1
COEFICIENTE 4  NO DE ENTRADA = 4
                   NO DE SAIDA = 2
COEFICIENTE 5  NO DE ENTRADA = 5
                   NO DE SAIDA = 2
COEFICIENTE 6  NO DE ENTRADA = 4
                   NO DE SAIDA = 6
COEFICIENTE 7  NO DE ENTRADA = 5
                   NO DE SAIDA = 6

NUMERO DE NOS COM VARIAVEIS QUANTIZADAS = 3

NO QUANTIZADO 1 = 2
                  NUMERO DE FONTES DE RUIDO = 2

NO QUANTIZADO 2 = 1
                  NUMERO DE FONTES DE RUIDO = 2

NO QUANTIZADO 3 = 6
                  NUMERO DE FONTES DE RUIDO = 3

FATOR DE MULTIPLICACAO DA VARIANCIA DA SAIDA (A) = 2.0

FATOR DE VARIACAO MAXIMO DA MAGNITUDE DA RESPOSTA EM FREQUENCIA =
 [illegible]

AMPLITUDE MAXIMA PERMITIDA PARA A MAGNITUDE DO ESPECTRO DE POTENCI
VIDO AO RUIDO PROVENIENTE DA QUANTIZACAO DE VARIAVEIS (B) = 1. DE

NUMERO DE DIVISOES ENTRE 0 E PI = 200

[illegible] DOS COEFICIENTES = [illegible]

[illegible]
```

```
*NUMEROS DE NOS DO SISTEMA = 6

NUMERO TOTAL DE RAMOS DE COEFICIENTES NO SISTEMA = 4

MATRIZ DE COEFICIENTES

COEFICIENTE DO NO = 1
              AO NO = 2
 = 1.0

COEFICIENTE DO NO = 2
              AO NO = 3
 = 1.0

COEFICIENTE DO NO = 4
              AO NO = 3
 = 7.113996E-1

COEFICIENTE DO NO = 5
              AO NO = [illegible]
 = -    1.[illegible]E-1

DESEJA MUDAR OU ACRESCENTAR ALGUM COEFICIENTE ? (S OU N) = N

NUMERO TOTAL DE ATRASOS NO SISTEMA = 5

MATRIZ DE ATRASOS

ATRASO DO NO = 3
         AO NO = 4
= 1.0
ATRASO DO NO = 4
         AO NO = 5
= 1.0
ATRASO DO NO = 2
         AO NO = 6
= -4.280435E-1
ATRASO DO NO = 4
         AO NO = 6
= 7.[illegible]201E-2
ATRASO DO NO = 5
         AO NO = 6
= 6.647147E-2

DESEJA MUDAR OU ACRESCENTAR ALGUM NO DE ATRASO? (S OU N) = N
```

```
NO DE ENTRADA = 1

NO DE SAIDA = 6

NUMERO DE COEFICIENTES QUANTIZADOS = 5

COEFICIENTE 1  NO DE ENTRADA = 2
                   NO DE SAIDA = 6
COEFICIENTE 2  NO DE ENTRADA = 4
                   NO DE SAIDA = 3
COEFICIENTE 3  NO DE ENTRADA = 5
                   NO DE SAIDA = 3
COEFICIENTE 4  NO DE ENTRADA = 4
                   NO DE SAIDA = 6
COEFICIENTE 5  NO DE ENTRADA = 5
                   NO DE SAIDA = 6

NUMERO DE NOS COM VARIAVEIS QUANTIZADAS = 2

NO QUANTIZADO  1 = 1
               NUMERO DE FONTES DE RUIDO = 2

NO QUANTIZADO  2 = 6
               NUMERO DE FONTES DE RUIDO = 1

FATOR DE MULTIPLICACAO DA VARIANCIA DA SAIDA [illegible] = 2.0

FATOR DE VARIACAO MAXIMO DA MAGNITUDE DA RESPOSTA EM FREQUENCIA = 1
 0.00

AMPLITUDE MAXIMA PERMITIDA PARA A MAGNITUDE DO ESPECTRO DE POTENC
VIDO AO RUIDO PROVENIENTE DA QUANTIZACAO DE VARIAVEIS (KV) = 39.0

NUMERO DE DIVISOES ENTRE 0 E 2*PI = 256

COMPRIMENTO DE PALAVRA ESTATISTICO DOS COEFICIENTES = 6.68077O  [illegible]

COMPRIMENTO ESTATISTICO DE VARIAVEIS = 4.79[illegible] BIT
```