

ССТ·УФРЪ

MÉTODOS COMPUTACIONAIS APLICADOS
À ANÁLISE E SÍNTESE ÓTIMA DE SIS-
TEMAS LINEARES

Carlos Alberto Holz

UNIVERSIDADE FEDERAL DA PARAIBA
CENTRO DE CIÊNCIAS E TECNOLOGIA

AVENIDA APRIGIO VELOSO, 882 - Cx. POSTAL 518
CAMPINA GRANDE - PB
BRASIL



H762m Holz, Carlos Alberto.
Métodos computacionais aplicados à análise e síntese
ótima de sistemas lineares / Carlos Alberto Holz. - Campina
Grande, 1975.
137 f.

Dissertação (Mestrado em Engenharia Elétrica) -
Universidade Federal da Paraíba, Centro de Ciências e
Tecnologia, 1975.
"Orientação : Prof. Janusz Stanislaw Lipowski".
Referências.

1. Métodos Computacionais - Linguagem de Programação. 2.
FORTRAN. 3. Sistemas Lineares. 4. Dissertação - Engenharia
Elétrica. I. Lipowski, Janusz Stanislaw. II. Universidade
Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.43(043)

MÉTODOS COMPUTACIONAIS APLICADOS À ANÁLISE E SÍNTESE
ÓTIMA DE SISTEMAS LINEARES

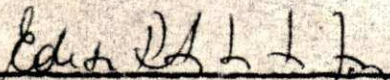
CARLOS ALBERTO HOLZ

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DE PÓS-GRADUAÇÃO DE ENGENHARIA DO CENTRO DE CIÊNCIAS E TECNOLOGIA DA UNIVERSIDADE FEDERAL DA PARAÍBA COMO PARTE DOS REQUISITOS NECESSÁRIOS A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

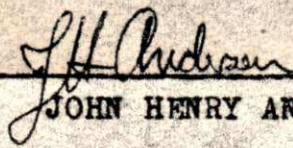
Aprovado por:



JANUSZ STANISŁAW LIPOWSKI
- Orientador -



EDISON ROBERTO CABRAL DA SILVA



JOHN HENRY ANDERSON

CAMPINA GRANDE
ESTADO DA PARAÍBA - BRASIL
FEVEREIRO DE 1975

RESUMO

Este trabalho apresenta um conjunto de programas escritos em FORTRAN para o IBM-1130, que permitem a análise e síntese ótima de sistemas lineares, com parâmetros constantes no tempo, contínuos, de malha aberta ou fechada, representados por equações de estado.

ABSTRACT

This work presents FORTRAN programs for the IBM-1130 for computational assistance in the analysis and optimal synthesis of linear, time-invariant, continuous-time, open-loop and closed-loop systems represented by state-space equations.

PP-5-11



INTRODUÇÃO

Os métodos clássicos de análise e síntese de sistemas no domínio do tempo ou no domínio da frequência, são bastante poderosos no caso de sistemas com apenas uma entrada e uma saída.

A análise e síntese de tais sistemas são feitas utilizando-se funções de transferência associadas a várias técnicas gráficas e numéricas.

Para sistemas de múltiplas entradas e/ou múltiplas saídas, esses métodos não são suficientes.

Uma técnica de representação de sistemas por equações de estado facilita a sua análise e síntese, especialmente com a utilização de computadores digitais.

No caso de sistemas muito complexos, esta técnica é uma das mais viáveis e bastante utilizada.

O objetivo deste trabalho é a adaptação de alguns programas existentes, que utilizam esta mencionada técnica, para o computador IBM - 1130 e desenvolvimento de outros necessários.

Os programas de interesse são programas para estudo do desempenho de sistemas lineares com parâmetros invariáveis no tempo, contínuo, de malha aberta e de malha fechada.

É também de interesse um programa para o cálculo da realimentação ótima do sistema de malha aberta, dado para determinado índice de performance.

O índice de performance é suposto ser da forma quadrática.

Os programas são divididos em três partes: a primeira, para os sistemas sem realimentação (Capítulo 2), a segunda para os sistemas com realimentação (Capítulo 3) e a última para a obtenção da realimentação - ótima (Capítulo 4).

No Capítulo 1 são apresentados os princípios matemáticos envolvidos em uma forma bastante resumida.

Os subprogramas utilizados são apresentados no Apêndice.

Í N D I C E

CAPÍTULO 1 - FUNDAMENTOS MATEMÁTICOS	1
1.1 - Definições Preliminares	1
1.2 - Representação de Sistemas no Espaço de Estado	2
1.3 - Matriz de Transição de Estado - $\phi(t)$	7
1.4 - Solução da Equação de Estado Não Homogênea	8
1.5 - Matriz Resolvente do Sistema - $\phi(s)$	9
1.6 - Controlabilidade e Observabilidade	10
1.7 - Sistemas de Malha Aberta e Fechada	12
1.8 - Otimização	13
CAPÍTULO 2 - SISTEMAS DE MALHA ABERTA	16
2.1 - Introdução	16
2.2 - Programa BASMT	16
2.3 - Programa CONOB	25
2.4 - Programa RESPT	32
CAPÍTULO 3 - SISTEMAS DE MALHA FECHADA	48
3.1 - Introdução	48
3.2 - Programa BASMF	50
3.3 - Programa COERM	60
3.4 - Programa RESPF	68

CAPÍTULO 4 - SISTEMAS DE MALHA FECHADA ÓTIMOS	82
4.1 - Introdução	82
4.2 - Alguns Métodos de Solução da Equação de Riccati	82
4.3 - Programa RICCI	88
4.4 - Comparação Entre Sistema de Malha Fechada Com Realimentação Qualquer e Com Realimentação Ótima ...	96
CONCLUSÃO -	107
APÊNDICE -	109
BIBLIOGRAFIA	136

CAPÍTULO 1

FUNDAMENTOS MATEMÁTICOS

1.1. Definições Preliminares

Neste Capítulo, são examinados os princípios matemáticos envolvidos na análise de sistemas, representados por equações de estado.

Depois de verificada a forma de se obter as equações de estado de um sistema, é analisada a solução das mesmas; a obtenção da matriz de transição e o estudo resumido da controlabilidade, da observabilidade e da otimização do sistema também são feitos.

Inicialmente, é necessária a definição de alguns termos⁽¹⁾:

Estado - menor conjunto de variáveis (chamadas variáveis de estado) tal que o conhecimento dessas variáveis em $t = t_0$ e também a entrada para $t \geq t_0$ determinam completamente o comportamento do sistema, para qualquer tempo $t \geq t_0$.

Variáveis de Estado - menor conjunto de variáveis que determinam o estado do sistema.

$$x_i(t), \quad i = 1, \dots, n$$

Vetor de Estado - vetor cujos elementos são as variáveis de estado - $X(t)$ ou X

Um vetor de estado é, então, um vetor que determina, unicamente, o estado de um sistema para qualquer $t \geq t_0$, desde que a entrada $U(t)$ para $t \geq t_0$ seja especificada e também as condições iniciais.

Espaço de Estado - espaço n-dimensional, cujos eixos coordenados consistem dos eixos x_1, x_2, \dots, x_n (variáveis de estado)

1.2. Representação de Sistemas no Espaço de Estado

Um sistema dinâmico, consistindo de um número finito de parâmetros concentrados, pode ser descrito por equações diferenciais ordinárias nas quais o tempo é a variável independente.

Uma equação diferencial de ordem "n" pode ser reduzida a um sistema de "n" equações de primeira ordem ⁽¹⁾.

Para a equação abaixo:

$$y^{(n)} + a_1 y^{(n-1)} + \dots + a_{n-1} \dot{y} + a_n y = u \quad (1.1)$$

Definindo:

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \\ &\vdots \\ x_n &= y^{(n-1)} \end{aligned}$$

A equação (1.1) pode ser escrita na forma seguinte:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\vdots \\
 \dot{x}_{n-1} &= x_n \\
 \dot{x}_n &= a_n x_1 - \dots - a_1 x_n + u
 \end{aligned}$$

Logo, um sistema dinâmico linear pode ser descrito por um sistema de equações da forma:

$$\begin{aligned}
 \dot{x}_1 &= a_{11}(t)x_1 + a_{12}(t)x_2 + \dots + a_{1n}(t)x_n + b_{11}(t)u_1 + b_{12}(t)u_2 + \dots + b_{1m}(t)u_m \\
 \dot{x}_2 &= a_{21}(t)x_1 + a_{22}(t)x_2 + \dots + a_{2n}(t)x_n + b_{21}(t)u_1 + b_{22}(t)u_2 + \dots + b_{2m}(t)u_m \\
 \dot{x}_n &= a_{n1}(t)x_1 + a_{n2}(t)x_2 + \dots + a_{nm}(t)x_n + b_{n1}(t)u_1 + b_{n2}(t)u_2 + \dots + b_{nm}(t)u_m
 \end{aligned} \tag{1.2}$$

onde:

u_i , $i = 1, \dots, n$ - variáveis de entrada

x_i , $i = 1, \dots, n$ - variáveis de estado (uma possível escolha)

Se a equação diferencial do sistema envolve derivadas da função de entrada, pode-se obter também um sistema da forma (1.2)⁽¹⁾.

Em notação matricial vetorial, o sistema pode ser escrito na for

ma:

$$\dot{X} = A(t) X + B(t) U \quad (1.3)$$

onde:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad - \text{ Vetor de estado}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \quad - \text{ Vetor de entrada}$$

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2n}(t) \\ \vdots & \vdots & \dots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \dots & a_{nn}(t) \end{bmatrix}$$

$$B(t) = \begin{bmatrix} b_{11}(t) & b_{12}(t) & \dots & b_{1m}(t) \\ b_{21}(t) & b_{22}(t) & \dots & b_{2m}(t) \\ \vdots & \vdots & \dots & \vdots \\ b_{n1}(t) & b_{n2}(t) & \dots & b_{nm}(t) \end{bmatrix}$$

Para a saída as equações são:

$$\begin{aligned}
 y_1 &= c_{11}(t) x_1 + c_{12}(t) x_2 + \dots + c_{1n}(t) x_n + d_{11}(t) u_1 + d_{12}(t) u_2 + \dots + d_{1m}(t) u_m \\
 y_2 &= c_{21}(t) x_1 + c_{22}(t) x_2 + \dots + c_{2n}(t) x_n + d_{21}(t) u_1 + d_{22}(t) u_2 + \dots + d_{2m}(t) u_m \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 y_p &= c_{p1}(t) x_1 + c_{p2}(t) x_2 + \dots + c_{pn}(t) x_n + d_{p1}(t) u_1 + d_{p2}(t) u_2 + \dots + d_{pm}(t) u_m
 \end{aligned}
 \tag{1.4}$$

Em notação matricial, pode-se escrever (1.4):

$$Y = C(t) X + D(t) U \tag{1.5}$$

onde:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad - \text{ Vetor de saída}$$

$$C(t) = \begin{bmatrix} c_{11}(t) & c_{12}(t) & \dots & c_{1n}(t) \\ c_{21}(t) & c_{22}(t) & \dots & c_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1}(t) & c_{p2}(t) & \dots & c_{pn}(t) \end{bmatrix}$$

$$D(t) = \begin{bmatrix} d_{11}(t) & d_{12}(t) & \dots & d_{1m}(t) \\ d_{21}(t) & d_{22}(t) & \dots & d_{2m}(t) \\ \vdots & \vdots & \ddots & \vdots \\ d_{p1}(t) & d_{p2}(t) & \dots & d_{pm}(t) \end{bmatrix}$$

A equação (1.3) é a equação de estado do sistema e a equação (1.5) é a equação de saída do sistema.

Essas duas equações e as condições iniciais definem totalmente o desempenho do sistema.

As matrizes $A(t)$, $B(t)$, $C(t)$ e $D(t)$ podem ser constantes ou funções do tempo. Este trabalho se limita a analisar o caso de A , B , C e D constantes.

Um diagrama de blocos do sistema, representado pelas equações (1.3) e (1.5) é dado pela Figura 1.1.

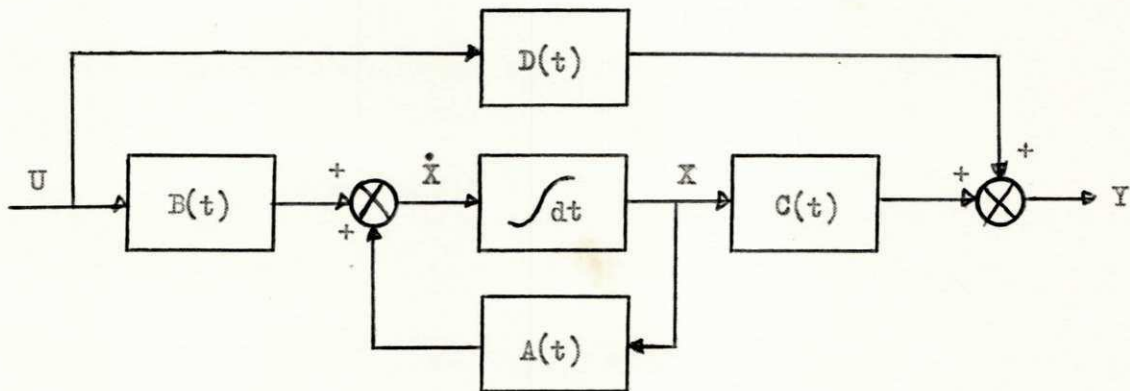


FIGURA 1.1. Diagrama de Blocos de um Sistema, Representado por Equações de Estado.

1.3. Matriz de Transição de Estado - $\phi(t)$ ⁽¹⁾

A homogênea da equação (1.3) é:

$$\dot{X} = A X \quad (1.6)$$

onde:

A - matriz n x n constante

A solução de (1.6) é:

$$X(t) = \left(I + At + \frac{1}{2!} A^2 t^2 + \dots + \frac{1}{k!} A^k t^k + \dots \right) X(0) \quad (1.7)$$

A expressão entre parênteses acima é uma matriz nxn e, devido à sua semelhança com a série exponencial infinita, pode-se mostrar que pode ser escrita na forma:

$$I + At + \frac{1}{2!} A^2 t^2 + \dots + \frac{1}{k!} A^k t^k + \dots = e^{At}$$

Logo, a expressão (1.7) toma a forma:

$$X(t) = e^{At} X(0) \quad (1.8)$$

onde:

$$e^{At} = \phi(t) \quad (1.9)$$

$\phi(t)$ é chamada matriz de transição de estado e contém toda informação sobre o movimento livre do sistema.

A solução de (1.6) é, então:

$$X(t) = \phi(t) X(0) \quad (1.10)$$

1.4. Solução da Equação de Estado não Homogênea ⁽¹⁾

A solução da equação de estado:

$$\dot{X} = AX + BU \quad (1.11)$$

onde:

- X - vetor n-dimensional
- U - vetor m-dimensional
- A - matriz nxn constante
- B - matriz nxm constante

pode ser obtida da seguinte forma:

Escrevendo (1.11) na forma

$$\dot{X}(t) - AX(t) = BU(t)$$

Premultiplicando ambos os lados por e^{-At}

$$e^{-At} \left[\dot{X}(t) - AX(t) \right] = \frac{d}{dt} \left[e^{-At} X(t) \right] = e^{-At} BU(t)$$

integrando entre os limites 0 e t:

$$e^{-At} X(t) = X(0) + \int_0^t e^{-A\tau} BU(\tau) d\tau$$

Pode-se obter, como resultado final:

$$X(t) = e^{At} X(0) + \int_0^t e^{A(t-\tau)} BU(\tau) d\tau \quad (1.12)$$

Pode-se ainda escrever (1.12) na forma:

$$X(t) = \phi(t) X(0) + \int_0^t \phi(t-\tau) BU(\tau) d\tau \quad (1.13)$$

1.5. Matriz Resolvente do Sistema - $\phi(S)$

Dado o sistema abaixo:

$$\dot{X} = AX + BU \quad (1.14)$$

$$Y = CX + DU \quad (1.15)$$

Se a transformada de Laplace é aplicada ao sistema (1.14) e (1.15) o seguinte resultado é obtido:

$$s X(S) - X(0^+) = AX(S) + BU(S) \quad (1.16)$$

$$Y(S) = CX(S) + DU(S) \quad (1.17)$$

Da equação (1.16):

$$X(S) = [SI - A]^{-1} X(0^+) + [SI - A]^{-1} BU(S) \quad (1.18)$$

A transformada inversa de (1.18) é:

$$X(t) = f(t) X(0^+) + \int_0^t f(t - \tau) BU(\tau) d\tau \quad (1.19)$$

Comparando (1.19) com (1.13) verifica-se que:

$$\phi(t) = f(t)$$

Logo:

$$\phi(t) = \mathcal{L}^{-1} [SI - A]^{-1} \quad (1.20)$$

$$\phi(S) = [SI - A]^{-1} \text{ é chamada de matriz resolvente do sistema.}$$

Substituindo (1.18) em (1.15) e admitindo que $X(0^+) = 0$:

$$Y(S) = \left[C (SI - A)^{-1} B + D \right] U(S)$$

$$G(S) = C (SI - A)^{-1} B + D \quad (1.21)$$

o lado direito de (1.21) envolve $(SI - A)^{-1}$ portanto, $G(S)$ pode ser escrito:

$$G(S) = \frac{Q(S)}{|SI - A|} \quad (1.22)$$

onde:

$Q(S)$ - matriz em S

Examinando (1.22) pode-se concluir que os autovalores de A são idênticos aos polos de $G(S)$.

Para o sistema ser estável, os autovalores de A devem estar todos no semi plano S da esquerda.

1.6. Controlabilidade e Observabilidade

Um processo ou sistema é dito controlável⁽²⁾ se é possível levar o sistema de um certo estado inicial $X(t_0)$ a um desejado estado $X(t_1)$ em um intervalo de tempo $t_1 - t_0$ finito. Em outras palavras, um processo é completamente controlável se existe um sinal controle $U(t)$ definido num intervalo finito $t_0 \leq t \leq t_1$ que move o processo do estado inicial $X(t_0)$ - ao desejado estado $X(t_1)$ no intervalo de tempo $t_1 - t_0$.

Um processo ou sistema é observável⁽²⁾ se os valores das variáveis de estado podem ser deduzidos a partir das variáveis de saída.

Em problemas típicos de controle ótimo, é necessário achar uma entrada controle $U(t)$ que leva o sistema de um dado estado inicial a um estado final desejado por algum caminho ótimo.

É claro que é necessário saber se o estado final pode ser alcançado e, então, o problema de controlabilidade aparece.

Muitas vezes também é necessário saber se o estado do sistema pode ser determinado, na prática, por medidas feitas na saída do sistema, surge, então, o problema da observabilidade.

Para o sistema definido pelas equações

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

pode-se mostrar^(1, 3, 4, 5) que a condição necessária e suficiente para o sistema ser controlável é que a matriz M , definida abaixo, tenha "rank" igual a " n " (ordem da matriz A).

$$M = \begin{bmatrix} B, AB, \dots, A^i B, \dots, A^{n-1} B \end{bmatrix} \quad (1.23)$$

Também se mostra^(1, 3, 4, 5) que a condição necessária e suficiente para o sistema ser observável é que a Matriz N , definida a seguir, tenha "rank" n (ordem da matriz A) onde C' e A' são as transpostas de C e A , respectivamente.

$$N = \begin{bmatrix} C', A' C', \dots, (A^{n-1})' C' \end{bmatrix} \quad (1.24)$$

Uma matriz A é de "rank" $n^{(1)}$ se existe uma sub matriz G de ordem $n \times n$ de A tal que o determinante de G é diferente de zero e o determinante de toda sub matriz $r \times r$ (onde $r \geq n + 1$) de A é zero.

1.7. Sistemas de Malha Aberta e Fechada

Por sistemas de malha aberta são denominados os sistemas representados pelas equações (1.14) e (1.15) sem malhas de realimentação.

O diagrama de blocos desses sistemas é o da Figura 1.1. Pode se verificar, na Figura, a inexistência de malha de realimentação.

Por sistemas de malha fechada são denominados os sistemas que apresentam malha de realimentação para a entrada.

O diagrama de blocos dos sistemas de malha fechada é dado pela Figura 1.2 seguinte.

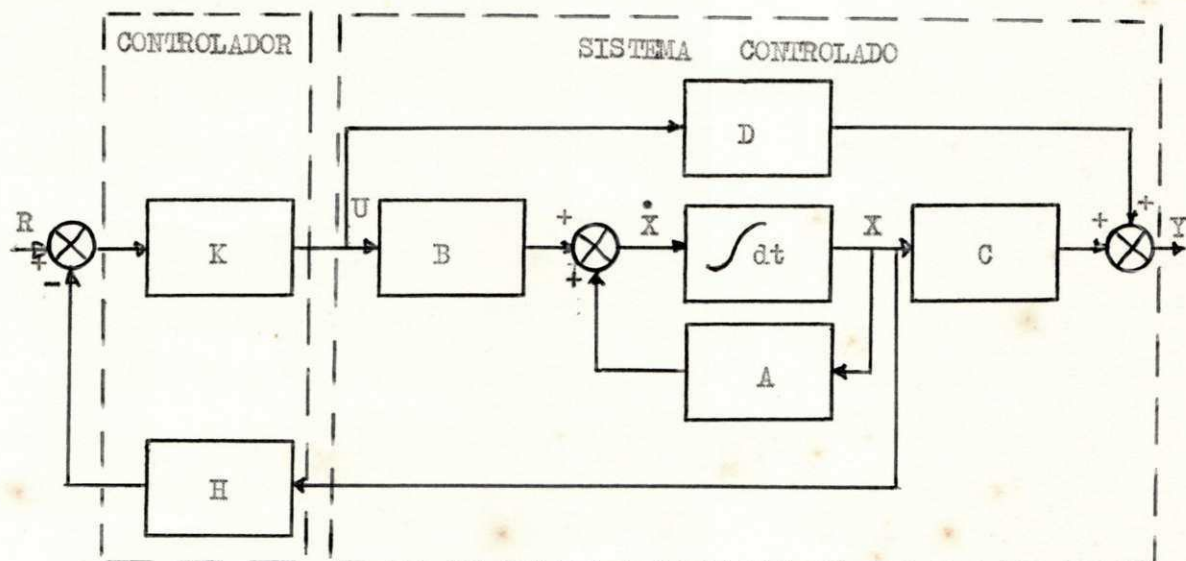


FIGURA 1.2. Diagrama de Blocos do Sistema de Malha Fechada

onde: $K = k I$
 k - ganho linear
 H - matriz de realimentação
 R - vetor de referência
 U - vetor de controle

1.8. Otimização (1, 4, 5, 7, 8)

A filosofia da teoria da otimização é projetar o "melhor" sistema. Isto implica em algum critério ou índice de performance, para julgar qual é o melhor.

Um sistema de controle deve satisfazer vários requisitos relacionados com o desempenho do mesmo, entradas e saídas, resposta desejada, limites do sistema impostos pela natureza do mesmo, ou por saturações, razão de oscilação etc.

O índice de performance é um modelo matemático dos requisitos necessários.

Ele é expresso em termos das entradas, saídas e estados dos elementos controlados.

Muitos índices de performance foram propostos, vários deles são casos particulares do índice:

$$I = \int_0^{\infty} f(t) g [e(t)] dt \quad (1.25)$$

onde:

$g [e(t)]$ - função do erro $e(t)$

$f(t)$ - fator peso de $g [e(t)]$ em função do tempo

$f(t)$ é geralmente uma das funções $1, t, t^2, \dots, t^n$ e $g[e(t)]$ é geralmente $e^2(t)$ ou $|e(t)|$.

Em problemas práticos de controle é muito empregado um índice da forma quadrática abaixo, que é de importância devido ao fato de que o sistema final satisfaz um compromisso, entre minimizar a integral do erro e minimizar a energia de controle:

$$I = \int_0^{\infty} \left[X'(t) Q_p X(t) + U'(t) R_p U(t) \right] dt \quad (1.26)$$

onde:

Q_p - matriz peso positiva definida (ou positiva semi definida) real e simétrica

R_p - matriz peso positiva definida real e simétrica

O problema de controle ótimo é, então, dado o sistema de equações (1.14) e (1.15) determinar a matriz H tal que minimize (1.26).

Da Figura (1.2):

$$\begin{aligned} \dot{X} &= AX + BU \\ U &= K [R - H X(t)] \end{aligned} \quad (1.27)$$

Assumindo $K = k \times I$ e $R = 0$ com $k = 1$, a equação (1.27) fica:

$$U = -HX(t) \quad (1.28)$$

Pode-se mostrar que a lei de controle dada por (1.28) é a Lei de controle ótima, desde que H seja dado pela expressão:

$$H = R_p^{-1} B'P \quad (1.29)$$

onde:

P - matriz solução da equação matricial de Riccati, na forma reduzida.

A equação matricial de Riccati, na forma reduzida, é dada por:

$$A'P + PA - PBR^{-1}B'P + Q = 0 \quad (1.30)$$

Logo, o problema da otimização de um sistema com parâmetros constantes é resolvido achando-se a solução de (1.30) e substituindo-se o resultado em (1.29) que é a matriz da malha de realimentação.

CAPÍTULO 2

SISTEMAS DE MALHA ABERTA

2.1. Introdução

Neste Capítulo são apresentados os programas para o estudo de sistemas de malha aberta, representados na forma:

$$\dot{X}(t) = A X(t) + B U(t) \quad (2.1)$$

$$X(t) = C X(t) + D U(t)$$

com condições iniciais $X(0)$.

É assumido que as matrizes A, B, C, D e as condições iniciais são conhecidas.

O estudo de tais sistemas é feito através de três programas - (BASMT, CONOB e RESPT) que permitem:

- determinar certas funções da matriz A, chamada matriz básica
- verificar se o sistema é observável e controlável
- obter os estados e saídas do sistema, em função do tempo sob a forma de tabelas e gráficos

2.2. Programa BASMT⁽⁹⁾

O programa BASMT permite achar:

- o determinante de A
- o inverso de A
- a matriz resolvente

- o polinômio característico
- os autovalores de A
- a matriz de transição

O programa lê os elementos de A linha por linha e também um cartão de opções que indica quais funções de A são desejadas.

O programa principal chama então sub programas que calculam as funções desejadas.

Estes programas são discutidos no Apêndice.

A disposição dos dados de entrada é explicada na Tabela 2.2.1.

Depois da identificação do problema e da leitura de A, a matriz é impressa e, dependendo do cartão de opções, o determinante de A, o seu inverso, a matriz resolvente, o polinômio característico, os autovalores e a matriz de transição são calculados e impressos.

A matriz resolvente é escrita como $\phi(S) = \text{adj}(SI - A)/\det(SI - A)$ e a matriz numerador, $\text{adj}(SI - A)$, é expressa como matriz dos coeficientes de potências de S, na forma:

$$\text{adj}(SI - A) = F_1 + F_2 S + \dots + F_n S^{n-1} \quad (2.2)$$

A matriz de transição é expressa numa forma similar a (2.2) como matriz coeficiente vezes o modo natural $\exp(\lambda_i t)$; valores complexos de autovalores são escritos como senos e cossenos amortecidos:

Exemplo para ilustrar o uso do programa:

Dada a matriz A abaixo, da qual é necessário o cálculo de todas as funções:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -3 \end{bmatrix}$$

TABELA 2.2.1.

DADOS DE ENTRADA PARA BASMT

Cartão Número	Coluna Número	Forma to	Valor Perfu rado	Nome da Variável, Descrição
1	1 - 20	10 A2	Título	Nome ou descrição arbitrária
	21 - 22	I2	1,2,....., 10	N - dimensão de A
2 até N + 1	1 - 10	E10.3	Valor de a_{11}	a_{11}
	11 - 20 etc	E10.3 : :	Valor de a_{12} : :	a_{12}
N + 2	1	IL	0	IDET - calcula determinante
	1	IL	1,2,....., 9	IDET - suprime determinante
	2	IL	0	INV - calcula inverso
	2	IL	1,2,....., 9	INV - suprime inverso
	3	IL	0	NRM - calcula resolvente
	3	IL	1,2,....., 9	NRM - suprime resolvente
	4	IL	0	ICP - calc. polin. característico
	4	IL	1,2,...., 9	ICP - sup. polin. característico
	5	IL	0	IEIG - calcula autovalores
	5	IL	1,2,....., 9	IEIG - suprime autovalores
6	IL	0	ISTM - calc. mat. transição	
	IL	1,2,....., 9	ISTM - sup. mat. transição	
N + 3				Branco - para o programa, igual ao primeiro para outro sistema.

De acordo com o objetivo, os dados de entrada estão na Tabela 2.2.2.

TABELA 2.2.2.

DADOS DE ENTRADA PARA O EXEMPLO DE BASMT

Cartão Número	Coluna Número																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30				
1	B	A	S	M	T	E X E M P L O										1										3								
2					0.	0									1.	0							0.	0										
3					0.	0									0.	0							1.	0										
4				-	2.	0									-	3.	0							-	3.	0								
5							C A R T A O										E M B R A N C O																	
6							C A R T A O										E M B R A N C O																	

Os resultados são dados na Tabela 2.2.3.

TABELA 2.2.3.

SAÍDA DO EXEMPLO DE BASMT

PROGRAMA MATRIZ BASICA
 IDENTIFICACAO DO PROBLEMA BASMAT EXEMPLO 1

A MATRIZ A

```

0.000E 00    0.100E 01    0.000E 00
0.000E 00    0.000E 00    0.100E 01
-0.200E 01   -0.300E 01   -0.300E 01
  
```

O DETERMINANTE DA MATRIZ

```

-0.200E 01
  
```

TABELA 2.2.3

(Cont.)

O INVERSO DA MATRIZ

-0.150E 01	-0.150E 01	-0.500E 00
0.100E 01	0.000E 00	0.000E 00
0.000E 00	0.100E 01	0.000E 00

OS COEFICIENTES DA MATRIZ DO NUMERADOR DA MATRIZ RESOLVENTE

OS COEFICIENTES DA MATRIZ DE S**2

0.100E 01	0.000E 00	0.000E 00
0.000E 00	0.100E 01	0.000E 00
0.000E 00	0.000E 00	0.100E 01

OS COEFICIENTES DA MATRIZ DE S**1

0.300E 01	0.100E 01	0.000E 00
0.000E 00	0.300E 01	0.100E 01
-0.200E 01	-0.300E 01	0.000E 00

OS COEFICIENTES DA MATRIZ DE S**0

0.300E 01	0.300E 01	0.100E 01
-0.200E 01	0.000E 00	0.000E 00
0.000E 00	-0.200E 01	0.000E 00

POLINOMIO CARACTERISTICO EM POTENCIAS CRESCENTES DE S

0.200E 01	0.300E 01	0.300E 01	0.100E 01
-----------	-----------	-----------	-----------

OS AUTOVALORES DA MATRIZ A

PARTE REAL

PARTE IMAGINARIA

-0.50000E 00	-0.86602E 00
-0.50000E 00	0.86602E 00
-0.20000E 01	0.00000E 00

TABELA 2.2.3.

(Cont.)

OS ELEMENTOS DA MATRIZ TRANSICAO DE ESTADO

OS COEFICIENTES MATRICIAIS DE $\text{EXP}(-0.500000E 00)T * \text{COS}(0.866025E 00)T$

```

0.666E 00  -0.333E 00  -0.333E 00
0.666E 00   0.166E 01   0.666E 00
-0.133E 01 -0.133E 01  -0.333E 00

```

OS COEFICIENTES MATRICIAIS DE $\text{EXP}(-0.500000E 00)T * \text{SEN}(0.866025E 00)T$

```

0.115E 01   0.173E 01   0.577E 00
-0.115E 01 -0.577E 00  -0.296E-06
-0.953E-06 -0.115E 01  -0.577E 00

```

OS COEFICIENTES MATRICIAIS DE $\text{EXP}(-0.200000E 01)T$

```

0.333E 00   0.333E 00   0.333E 00
-0.666E 00 -0.666E 00  -0.666E 00
0.133E 01   0.133E 01   0.133E 01

```

Para este exemplo, verifica-se que:

$$\phi_{11}(s) = \frac{3 + 3s + s^2}{2 + 3s + 3s^2 + s^3}$$

$$\phi_{11}(t) = 0,667 e^{-0,5t} \cos 0,866t + 1,11 e^{-0,5t} \text{sen } 0,866t + \\ + 0,33 e^{-2t}$$

Os seguintes subprogramas são usados por BASMT:

1 - PCRS	3 - XMPRE	5 - DET
2 - EQLSI	4 - RAPOL	

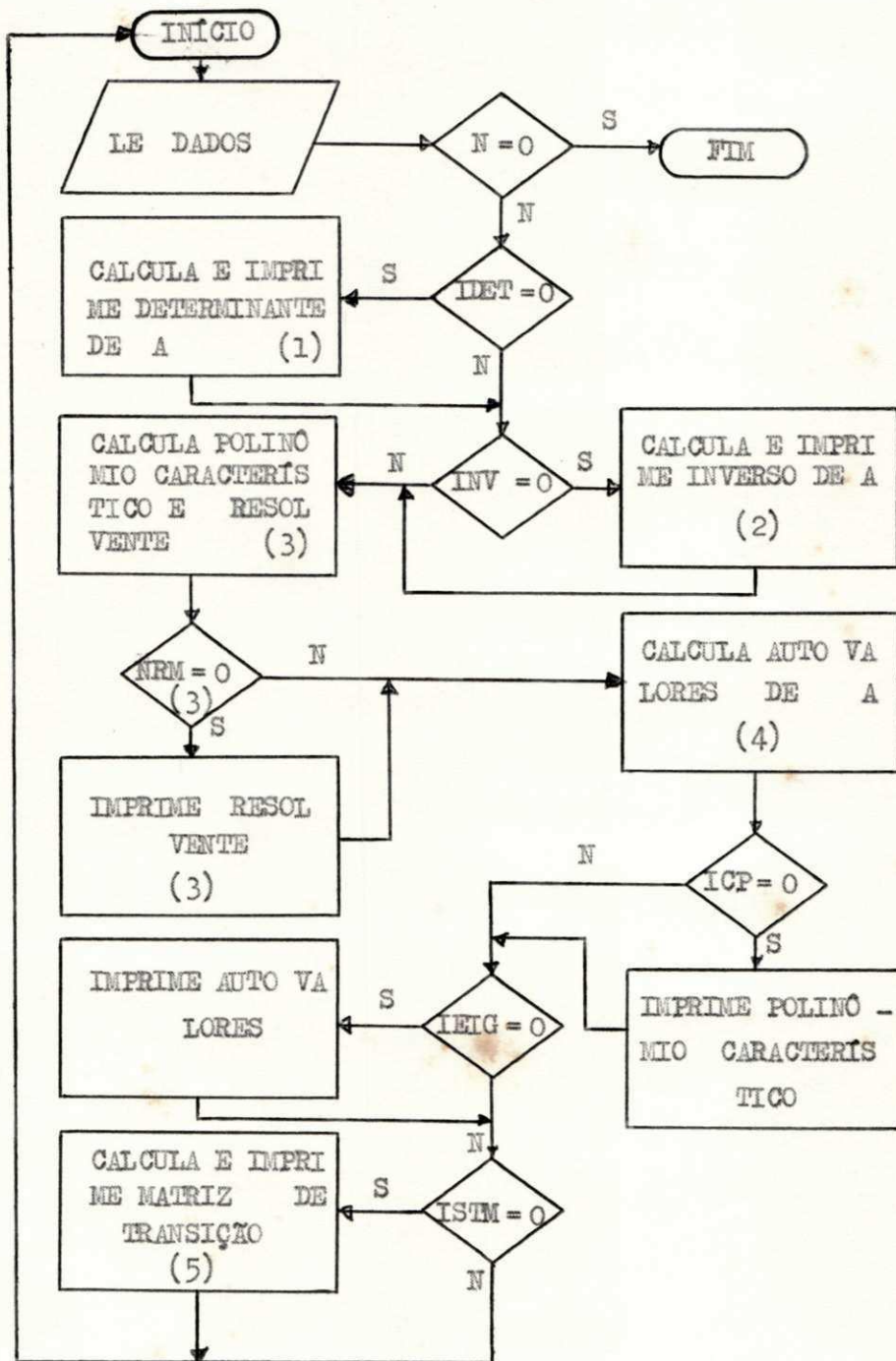


FIGURA 2.1. Fluxograma de BASMT

A descrição destes subprogramas é feita no Apêndice A.

O fluxograma de BASMT é dado pela Figura 2.1.

Os números dentro dos quadrinhos são para se fornecer algumas explicações adicionais da maneira pela qual é feita a operação do respectivo quadrinho. No fluxograma de BASMT, as respectivas operações são:

- (1) - Cálculo do determinante de A, feito através da função -
DET
- (2) - Cálculo do inverso de A, feito através da sub rotina
EQLSI
- (3) - Cálculo do polinômio característico e da matriz resolven-
te, feito pela sub rotina PCMRS
- (4) - Cálculo dos autovalores de A, feito pela sub rotina RA
POL
- (5) - Cálculo da matriz de transição feito pela sub rotina XM
TRE.

LISTAGEM DE BASMT

```

C PROGRAMA MATRIZ BASICA
C SUBPROGRAMAS USADOS PCMRS,EQLSI,XMTRE,RAPOL,DET
  DIMENSION A(10,10),EIGR(10),EIGI(10),C(11),AINV(10,10),NAME(5)
  COMMON CHI(10,10,10)
2001 FORMAT(10A2,I2)
2002 FORMAT(8E10.3)
2003 FORMAT(6X,10(E10.3,2X))
2004 FORMAT(1H0,5X,16HA MATRIZ A      ,/)
2005 FORMAT(1H0,5X,'POLINOMIO CARACTERISTICO EM POTENCIAS CRESCENTES DE
1  S',/)
2006 FORMAT(1H0,5X,'OS AUTOVALORES DA MATRIZ A')
2007 FORMAT(9X,'PARTE REAL',9X'PARTE IMAGINARIA',/)
2008 FORMAT(1H1,5X,'PROGRAMA MATRIZ BASICA')
2009 FORMAT(6X,'IDENTIFICACAO DO PROBLEMA ',5X,10A2)
2010 FORMAT(1H0,5X,'O DETERMINANTE DA MATRIZ',/)
2011 FORMAT(1H0,5X,'O INVERSO DA MATRIZ',/)
2012 FORMAT(1H0,60(1H*))
2013 FORMAT(6I1)
2014 FORMAT(8X,2(E12.5,10X))
  4 READ(2,2001) (NAME(I),I=1,10),N
  IF(N) 13,10,13
13 DO 1 I=1,N
  1 READ(2,2002) (A(I,K),K=1,N)

```


LISTAGEM DE BASMT

(Cont.)

```
READ(2,2013) IDET,INV,NRM,ICP,IEIG,ISTM
WRITE(3,2008)
WRITE(3,2009) (NAME(I),I=1,10)
WRITE(3,2012)
WRITE(3,2004)
DO 2 I=1,N
2 WRITE(3,2003) (A(I,K),K=1,N)
  IF(IDET)5,6,5
6 D=DET(A,N)
  WRITE(3,2010)
  WRITE(3,2003) D
5 IF(INV)15,7,15
7 WRITE(3,2011)
  CALL EQLSI(A,C,N,AINV,C,IERR)
  IF(IERR)8,15,8
8 DO 20 I=1,N
20 WRITE(3,2003) (AINV(I,J),J=1,N)
15 CALL PCMRS(A,N,C,NRM)
  CALL RAPOL(N,C,EIGR,EIGI,+1)
  IF(ICP)30,9,30
9 WRITE(3,2012)
  WRITE(3,2005)
  NN=N+1
  WRITE(3,2003) (C(I),I=1,NN)
30 IF(IEIG)35,11,35
11 WRITE(3,2012)
  WRITE(3,2006)
  WRITE(3,2007)
  DO 3 I=1,N
3 WRITE(3,2014) EIGR(I),EIGI(I)
35 IF(ISTM)25,12,25
12 CALL XMTRE(N,A,EIGR,EIGI,ISTM)
25 GO TO 4
10 CALL EXIT
END
```

2.3. Programa CONOB

O programa CONOB permite:

- verificar a observabilidade do sistema
- verificar a controlabilidade do sistema

O programa lê os elementos de A, B e C, fila por fila, imprime esses valores e, por meio de sub programas, forma as matrizes M (1.23) e N (1.24).

Calcula, então, o "rank" de M e de N e faz a comparação com a ordem da matriz A verificando, dessa forma, a observabilidade e controlabilidade.

A disposição dos dados de entrada é explicada na Tabela 2.3.1.

Exemplo para ilustrar o uso do programa

Dados:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Os dados de entrada estão na Tabela 2.3.2.

TABELA 2.3.1.
DADOS DE ENTRADA PARA CONOB

Cartão Número	Coluna Número	Formato	Valor Perfurado	Nome da Variável, Descrição
1	1 - 2	I2	1,2,....., 10	N - ordem da matriz A
	10 -11	I2	1,2,....., 10	M - número de colunas de B
	20 -21	I2	1,2,....., 10	P - número de linhas de C
N + 1	1 -10	E10.3	Valor de a_{11}	a_{11}
	11 -20	E10.3	Valor de a_{12}	a_{12}
	etc	⋮	⋮	⋮
N + 2 até 2N + 1	1 -10	E10.3	Valor de b_{11}	b_{11}
	11 -20	E10.3	Valor de b_{12}	b_{12}
	etc	⋮	⋮	⋮
2N + 2 até 2N+P+1	1 -10	E10.3	Valor de C_{11}	C_{11}
	11 -20	E10.3	Valor de C_{12}	C_{12}
	etc	⋮	⋮	⋮
2N+P+2	1 - 2	I2	-1,-2,....- 9	Número inteiro negativo para parar ou igual ao primeiro para outro sistema, repetindo tudo.

TABELA 2.3.2.

DADOS DE ENTRADA PARA O EXEMPLO DE CONOB

Cartão Número	Coluna Número																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1		2									2										1									
2					0	.	0							1	.	0														
3				-	2	.	0					-	3	.	0															
4					0	.	0							1	.	0														
5					1	.	0						-	2	.	0														
6					1	.	0							0	.	0														
7	-	1																												

Os resultados do exemplo estão na Tabela 2.3.3.

Verifica-se que o sistema é observável e controlável.

Os subprogramas utilizados são:

- | | | |
|-----------|-----------|-----------|
| 1 - SOMA | 4 - MATEQ | 7 - ADDIC |
| 2 - PRODU | 5 - TRNPZ | 8 - LER |
| 3 - RANK | 6 - KDENT | 9 - IMPR |

O fluxograma de CONOB é dado pela Figura 2.2.

TABELA 2.3.3.

SAÍDA DO EXEMPLO DE CONOB

EXEMPLO DE CONTROLABILIDADE E OBSERVABILIDADE
IDENTIFICACAO DO PROBLEMA CONOB EXEMPLO 1

MATRIZ A

0.000E 00 0.100E 01
-0.200E 01 -0.300E 01

MATRIZ B

0.000E 00 0.100E 01
0.100E 01 -0.200E 01

MATRIZ C

0.100E 01 0.000E 00

A ORDEM DO SISTEMA E 2

O SISTEMA E CONTROLAVEL,O RANK DE M E 2

O SISTEMA E OBSERVAVEL,O RANK DE N E 2

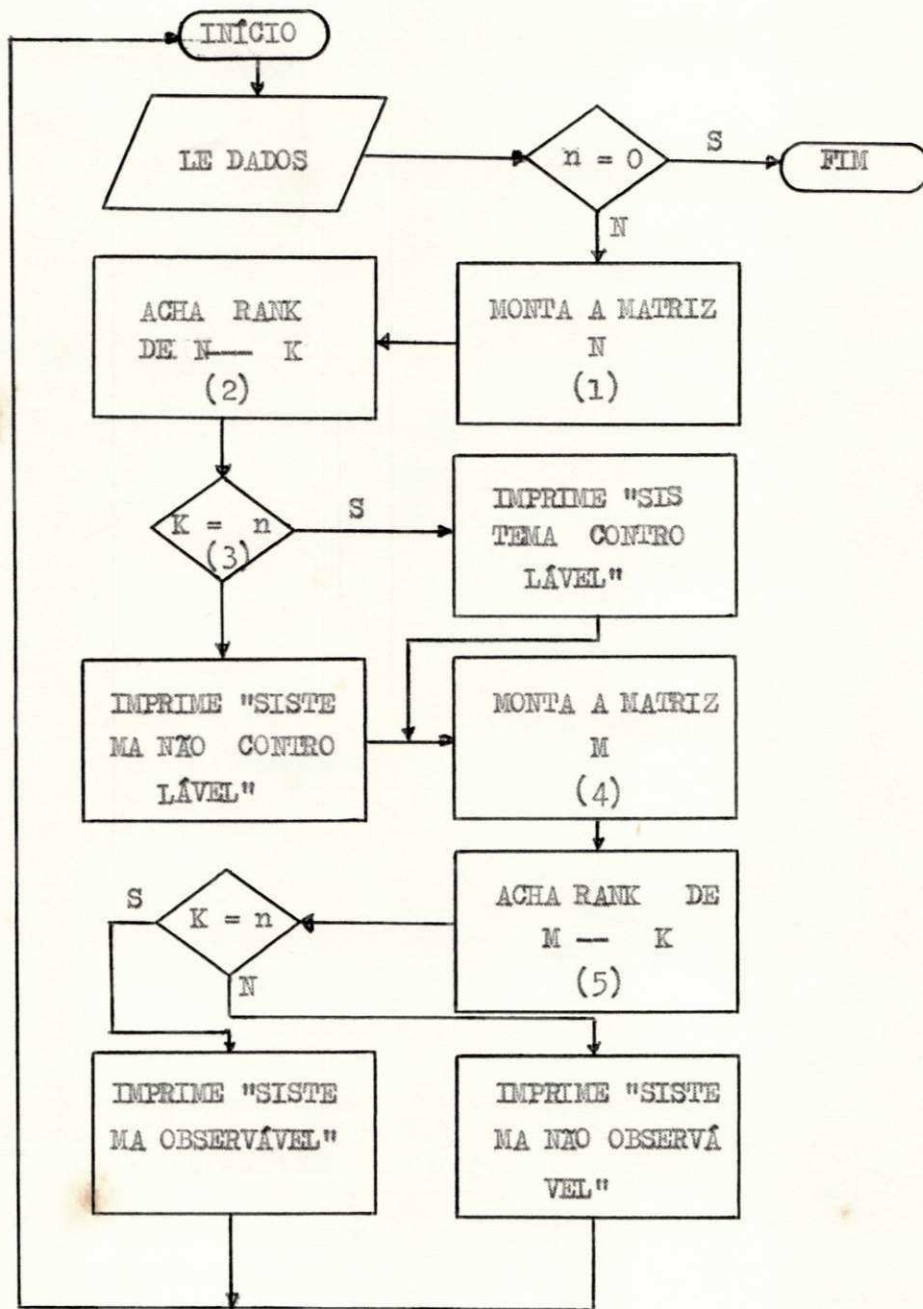


FIGURA 2.2. Fluxograma de CONOB

LISTAGEM DE CONOB

```
C  PROGRAMA PRINCIPAL
C  PROGRAMA CONTROLABILIDADE E OBSERVABILIDADE  CONOB
C  SUBPROGRAMAS USADOS  SOMA,PRODU,RANK,MATEQ,TRNPZ,
C  KIDENT,ADDIC,LER,IMPR
C
C  A(N,N)  B(N,M)  C(P,N)
C
C  INTEGER P
C  DIMENSION XM(10,100),XN(10,100),A(10,10),B(10,10),C(10,10),
1  E(10,10),F(10,10),G(10,10),Cl(10,10),Dl(10,10)
C  IND=1
C  12 READ(2,101) N,M,P
C  101 FORMAT(I2,7X,I2,8X,I2)
C  IF(N) 13,14,14
C  14 WRITE(3,107) IND
C  107 FORMAT(1H1,////,2X,'EXEMPLO DE CONTROLABILIDADE E OBSERVABILIDADE'
1  ,/,2X,'IDENTIFICACAO DO PROBLEMA  CONOB EXEMPLO',I4)
C  WRITE(3,108)
C  108 FORMAT(1H0,60(1H*),/)
C  CALL LER(A,N,N)
C  CALL LER(B,N,M)
C  CALL LER(C,P,N)
C  WRITE(3,109)
C  109 FORMAT(2X,'MATRIZ  A'//)
C  CALL IMPR(A,N,N)
C  WRITE(3,111)
C  111 FORMAT(/,2X,'MATRIZ  B',/)
C  CALL IMPR(B,N,M)
C  WRITE(3,112)
C  112 FORMAT(/,2X,'MATRIZ  C',/)
C  CALL IMPR(C,P,N)
C  WRITE(3,108)
C  WRITE(3,113) N
C  113 FORMAT(1H0,2X,'A ORDEM DO SISTEMA E',I5)
C  CALL KIDENT(F,N)
C  CALL KIDENT(G,N)
C  DO 4 I=1,N
C  CALL PRODU(G,B,E,N,N,M)
C  CALL ADDIC(XM,E,N,M,I)
C  CALL PRODU(F,A,G,N,N,N)
```

LISTAGEM DE CONOB

(Cont.)

```
CALL MATEQ(G,F,N,N)
4 CONTINUE
MM=N*M
CALL RANK(XM,N,MM,K)
IF(K-N) 6,5,6
5 WRITE(3,103) K
103 FORMAT(/3X,'O SISTEMA E CONTROLAVEL,O RANK DE M E',I5)
GO TO 7
6 WRITE(3,104) K
104 FORMAT(/3X,'O SISTEMA NAO E CONTROLAVEL,O RANK DE M E',I5)
7 CONTINUE
CALL KDENT(F,N)
CALL KDENT(G,P)
CALL TRNPZ(C,C1,P,N)
DO 8 I=1,N
CALL TRNPZ(G,D1,N,N)
CALL PRODU(D1,C1,E,N,N,P)
CALL ADDIC(XN,E,N,P,I)
CALL PRODU(F,A,G,N,N,N)
CALL MATEQ(G,F,N,N)
8 CONTINUE
MM=N*P
CALL RANK(XN,N,MM,K)
IF(K-N) 9,10,9
9 WRITE(3,105) K
105 FORMAT(/3X,'O SISTEMA NAO E OBSERVAVEL, O RANK DE N E',I5)
GO TO 11
10 WRITE(3,106) K
106 FORMAT(/3X,'O SISTEMA E OBSERVAVEL,O RANK DE N E',I5)
11 CONTINUE
IND=IND+1
GO TO 12
13 CALL EXIT
END
```


As operações dos quadrinhos numerados são feitas da seguinte maneira:

- (1) - Montagem da matriz N, feita através da subrotina ADDIC que aloca cada matriz da forma $(A^{k-1})'C$ no seu respectivo local, de acordo com a fórmula (1.24)
- (2) - Cálculo do "rank" de N, feito através da subrotina RANK
- (3) - Teste se o "rank" de N é igual à ordem da matriz A
- (4) - Montagem da matriz M, feita de maneira semelhante à de N, de acordo com a fórmula (1.23)
- (5) - Teste se o "rank" de M é igual à ordem da matriz A

2.4. Programa RESPT

O programa RESPT permite:

- Cálculo dos estados do sistema
- Cálculo das saídas do sistema
- Obtenção dos gráficos dos estados
- Obtenção dos gráficos das saídas

O algoritmo do programa foi baseado na equação (1.13)⁽¹⁰⁾:

$$X(t) = \phi(t) X(0) + \int_0^t \phi(t - \tau) B U(\tau) d\tau \quad (2.3)$$

Aplicando esta equação num intervalo de tempo $KT < t \leq (K+1)T$ onde T representa o intervalo de amostragem, obtém-se:

$$X_{k+1} = e^{AT} X_k + e^{AT} \int_0^T e^{-A\tau} BU(\tau) d\tau \quad (2.4)$$

onde:

$$\tau = t - K\tau$$

Considerando que a entrada $U(t)$ é constante no intervalo em questão e que a mudança na função de entrada é feita somente no fim do intervalo de tempo, a equação (2.4) pode ser integrada.

$$X_{k+1} = e^{AT} X_k + e^{AT} (I - e^{-AT}) A^{-1} BU_k$$

$$X_{k+1} = e^{AT} X_k + (e^{AT} - I) A^{-1} BU_k \quad (2.5)$$

(2.5) pode ser escrita na forma:

$$X_{k+1} = PX_k + QU_k \quad (2.6)$$

onde:

$$P = e^{AT} = \phi(T) \quad (2.7)$$

$$Q = (\phi(T) - I) A^{-1} B \quad (2.8)$$

$\phi(T)$ foi definido em (1.9) como:

$$\phi(T) = I + AT + \frac{1}{2!} A^2 T^2 + \dots + \frac{1}{n!} A^n T^n + \dots$$

Substituindo a expressão de $\phi(T)$ em (2.7) e (2.8):

$$P = I + AT + \frac{1}{2!} A^2 T^2 + \dots + \frac{1}{n!} A^n T^n + \dots \quad (2.9)$$

$$Q = T \left(I + \frac{1}{2} AT + \frac{1}{3!} (AT)^2 + \dots \right) B \quad (2.10)$$

As séries de P e Q são convergentes (para T pequeno) e se pode tomar o número de termos de acordo com a precisão requerida. Para um intervalo de tempo pequeno, bastam alguns termos iniciais.

Uma vez calculados P e Q, eles são constantes para um certo intervalo de tempo fixado a priori e, então, partindo do tempo inicial e das condições iniciais, aplicando (2.6) acham-se todos os estados subsequentes

As saídas podem ser calculadas por:

$$Y_k = C X_k + DU_k \quad (2.11)$$

O programa RESPT então calcula P e Q através de (2.9) e (2.10) aplicando (2.6) e (2.11) calcula os estados e saídas do sistema, desde o instante inicial até o instante final, com o intervalo de tempo especificado.

Internamente, o programa calcula P e Q, estados e saídas com um intervalo de 0,01 e que dá uma boa precisão com tempo de computação pequeno.

Para maior precisão (intervalo menor do que 0,01) basta diminuir o valor da variável DELT do programa principal mas o tempo de computação cresce bastante.

O intervalo de tempo para a saída é escolhido através da variável DEL, que é lida como dado.

A precisão das séries P e Q é da ordem de $1.0 E - 8$ o que é conseguido rapidamente com poucos termos, para T pequeno.

Os dados de entrada para o programa são especificados na Tabela 2.4.1.

A entrada $U(T)$ é definida separadamente, através de uma subrotina denominada ENTR de argumentos U e T. O modo de defini-la é:

$$\begin{aligned} U(1, 1) &= \dots \\ U(2, 1) &= \dots \\ &\vdots \\ U(10, 1) &= \dots \end{aligned}$$

A entrada é definida em forma de matriz bidimensional e não em forma de vetor para poder utilizar os mesmos subprogramas básicos em suas operações matriciais.

Exemplo para ilustrar o uso do programa

Dados:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} & B &= \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix} & D &= \begin{bmatrix} 1 \end{bmatrix} \end{aligned}$$

Condições iniciais:

$$X(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

TABELA 2.4.1.

DADOS DE ENTRADA PARA RESPT

Cartão Número	Coluna Número	Formato	Valor Perfu rado	Nome da Variável, Descrição
1	10 - 11	I2	1,2,....., 10	N - ordem de A
	10 - 21	I2	1,2,....., 10	M - número de colunas de B
	30 - 31	I2	1,2,....., 10	IP - número de linhas de C
	40 - 49	E10.3	Valor de DEL	DEL - intervalo de tempo saída
	50 - 59	E10.3	Valor de TO	TO - tempo inicial
	60 - 69	E10.3	Vr. de TMAX	TMAX - tempo final
	70 - 79	E10.3	Vr. de PLOTE	PLOTE - nº real qualquer - Sup. gráfico
2 até N + 1	1 - 10	E10.3	Valor de a_{11}	a_{11}
	11 - 20	E10.3	Valor de a_{12}	a_{12}
	etc	:	.	.
N + 2 até 2N + 1	1 - 10	E10.3	Valor de b_{11}	b_{11}
	11 - 20	E10.3	Valor de b_{12}	b_{12}
	etc	:	.	.
2N + 2 até 2N+IP+1	1 - 10	E10.3	Valor de C_{11}	C_{11}
	11 - 20	E10.3	Valor de C_{12}	C_{12}
	etc	:	.	.
2N+IP+2 até 2N+2IP+1	1 - 10	E10.3	Valor de d_{11}	d_{11}
	11 - 20	E10.3	Valor de d_{12}	d_{12}
	etc	:	.	.
2N+2IP+2 2N+2IP+3 :	1 - 10	E10.3	Valor de $x_{11}(0)$	$x_{11}(0)$
	1 - 10	E10.3	Valor de $x_{21}(0)$	$x_{21}(0)$
	:	:	.	.
3N+2IP+2	10 - 11	I2	-1,-2,.....,- 9	Para o programa ou igual ao primeiro, repetindo tudo pa ra outro sistema.

entrada $U(T) = \text{EXP}(-T)$

tempo inicial - 0,0 segundos

tempo final - 22,0 segundos

intervalo de tempo para a saída - 0,2 segundos

Os dados de entrada do programa estão na Tabela 2.4.2.

Os resultados do programa estão na Tabela 2.4.3.

Os gráficos são dados nas Figuras 2.4.1, 2.4.2. e 2.4.3.

Os subprogramas utilizados são:

1 - LER	5 - SOMA	8 - VALMX	11 - SCMAT
2 - IMPR	6 - SCMUT	9 - ENTR	12 - ADDM
3 - KDENT	7 - MATEQ	10 - FAT	13 - GPLTE
4 - PRODU			

O fluxograma de RESPT é dado na Figura 2.3.

As operações nos quadrinhos numerados são:

- (1) - Cálculo da matriz P de acordo com a fórmula (2.9) através das subrotinas KDENT, SCMUT, PRODU, MATEQ, ADDM e VALMX
- (2) - Cálculo da matriz Q, de acordo com a fórmula (2.10) através das subrotinas KDENT, SCMUT, PRODU, MATEQ, ADDM e VALMX
- (3) - Cálculo dos estados de acordo com a fórmula (2.6) através das subrotinas ENTR, PRODU, SOMA
- (4) - Cálculo das saídas de acordo com a fórmula (2.11) através das subrotinas ENTR, PRODU, SOMA
- (5) - Obtenção dos gráficos dos estados através da subrotina GPLTE
- (6) - Obtenção dos gráficos das saídas através da subrotina GPLTE

TABELA 2.4.3.

RESULTADOS DO EXEMPLO DO PROGRAMA RESPT

```

PROGRAMA RESPOSTA E ESTADO DO SISTEMA NO TEMPO
IDENTIFICACAO DO PROBLEMA      RESPT  EXEMPLO    1
*****
ORDEM DO SISTEMA      2
INTERVALO DE TEMPO PARA A SAIDA  0.20000
TEMPO INICIAL        0.0000
TEMPO FINAL         22.0000
*****
MATRIZ  A
  0.000E 00  0.100E 01
-0.200E 01 -0.300E 01
MATRIZ  B
  0.200E 01
  0.000E 00
MATRIZ  C
  0.100E 01  0.000E 00
MATRIZ  D
  0.100E 01
CONDICOES INICIAIS...X(0)...
  0.000E 00
  0.000E 00
*****
NUMERO DE TERMOS NAS SERIES DE P E Q
  5
*****

```


TABELA 2.4.3.

(Cont.)

ESTADOS E SAIDAS DO SISTEMA

SEQUENCIA NUM. ORDEM, TEMPO, ESTADOS NA ORDEM X(1)...X(2)
EM SEGUIDA AS SAIDAS NA ORDEM Y(1)...Y(1)

0.	0.000E 00	0.00000E 00	0.00000E 00	0.10000E 01
1.	0.199E 00	0.35637E 00	-0.61030E-01	0.11751E 01
2.	0.399E 00	0.62737E 00	-0.18759E 00	0.12976E 01
3.	0.599E 00	0.81780E 00	-0.32503E 00	0.13666E 01
4.	0.799E 00	0.93826E 00	-0.44587E 00	0.13875E 01
5.	0.999E 00	0.10013E 01	-0.53862E 00	0.13692E 01
6.	0.119E 01	0.10196E 01	-0.60079E 00	0.13208E 01
7.	0.139E 01	0.10043E 01	-0.63458E 00	0.12508E 01
8.	0.159E 01	0.96499E 00	-0.64434E 00	0.11668E 01
9.	0.179E 01	0.90960E 00	-0.63503E 00	0.10749E 01
10.	0.199E 01	0.84437E 00	-0.61150E 00	0.97970E 00
11.	0.219E 01	0.77409E 00	-0.57803E 00	0.88490E 00
12.	0.239E 01	0.70235E 00	-0.53820E 00	0.79307E 00
13.	0.259E 01	0.63172E 00	-0.49490E 00	0.70600E 00
14.	0.279E 01	0.56398E 00	-0.45033E 00	0.62479E 00
15.	0.299E 01	0.50028E 00	-0.40614E 00	0.55007E 00
16.	0.319E 01	0.44131E 00	-0.36350E 00	0.48207E 00
17.	0.339E 01	0.38738E 00	-0.32319E 00	0.42076E 00
18.	0.359E 01	0.33858E 00	-0.28569E 00	0.36590E 00
19.	0.379E 01	0.29479E 00	-0.25127E 00	0.31716E 00
20.	0.399E 01	0.25578E 00	-0.22000E 00	0.27410E 00
21.	0.419E 01	0.22125E 00	-0.19186E 00	0.23625E 00
22.	0.439E 01	0.19085E 00	-0.16672E 00	0.20313E 00
23.	0.459E 01	0.16421E 00	-0.14441E 00	0.17426E 00
24.	0.479E 01	0.14096E 00	-0.12472E 00	0.14919E 00
25.	0.499E 01	0.12075E 00	-0.10743E 00	0.12749E 00
26.	0.519E 01	0.10324E 00	-0.92326E-01	0.10876E 00
27.	0.539E 01	0.88114E-01	-0.79167E-01	0.92631E-01
28.	0.559E 01	0.75078E-01	-0.67747E-01	0.78777E-01
29.	0.579E 01	0.63874E-01	-0.57868E-01	0.66903E-01
30.	0.599E 01	0.54266E-01	-0.49346E-01	0.56745E-01
31.	0.619E 01	0.46043E-01	-0.42012E-01	0.48072E-01
32.	0.639E 01	0.39018E-01	-0.35717E-01	0.40680E-01
33.	0.659E 01	0.33027E-01	-0.30323E-01	0.34387E-01
34.	0.679E 01	0.27926E-01	-0.25712E-01	0.29040E-01
.
.
.

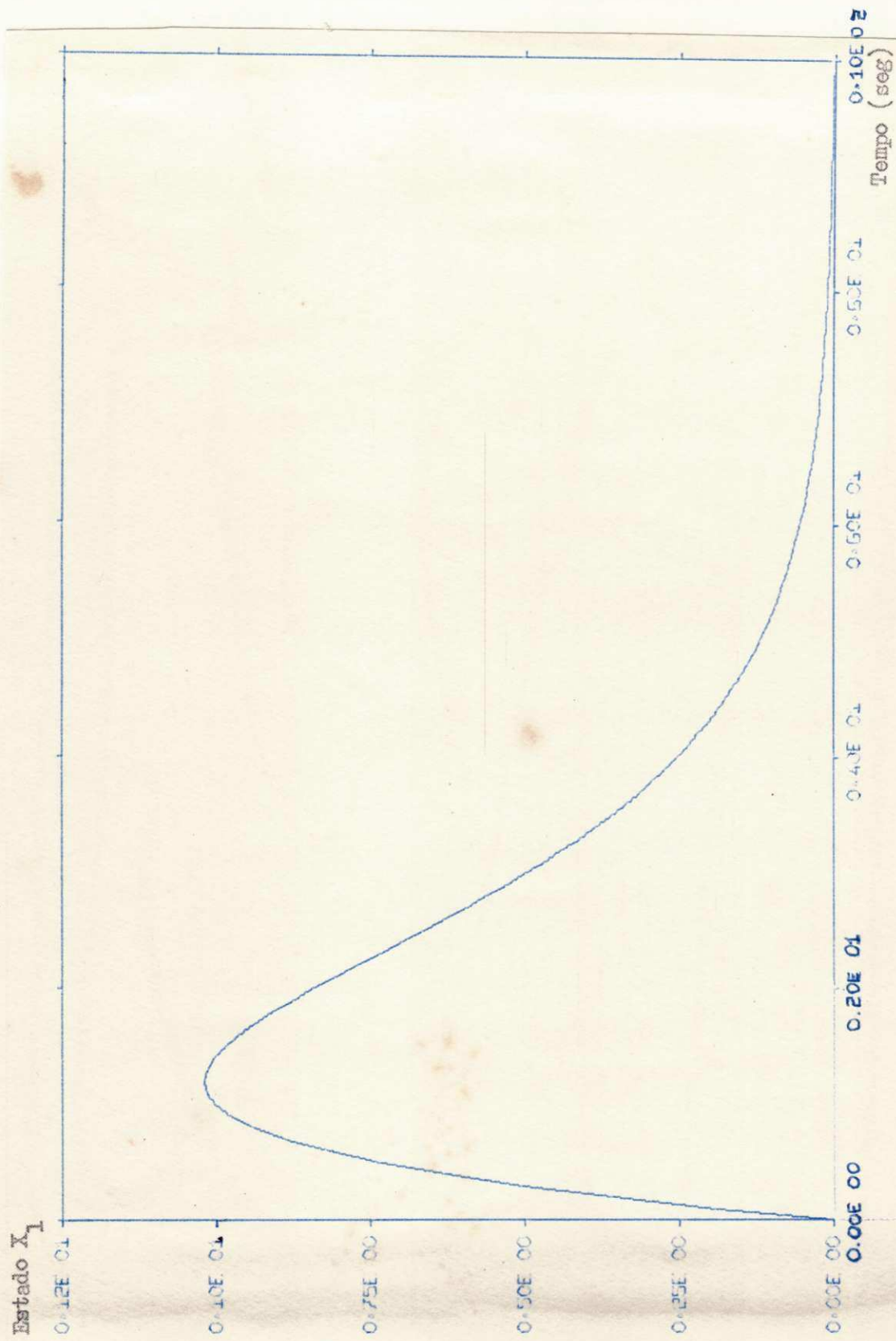


FIGURA 2.4.1. Estado X_1 do Exemplo de RESPT em Função do Tempo

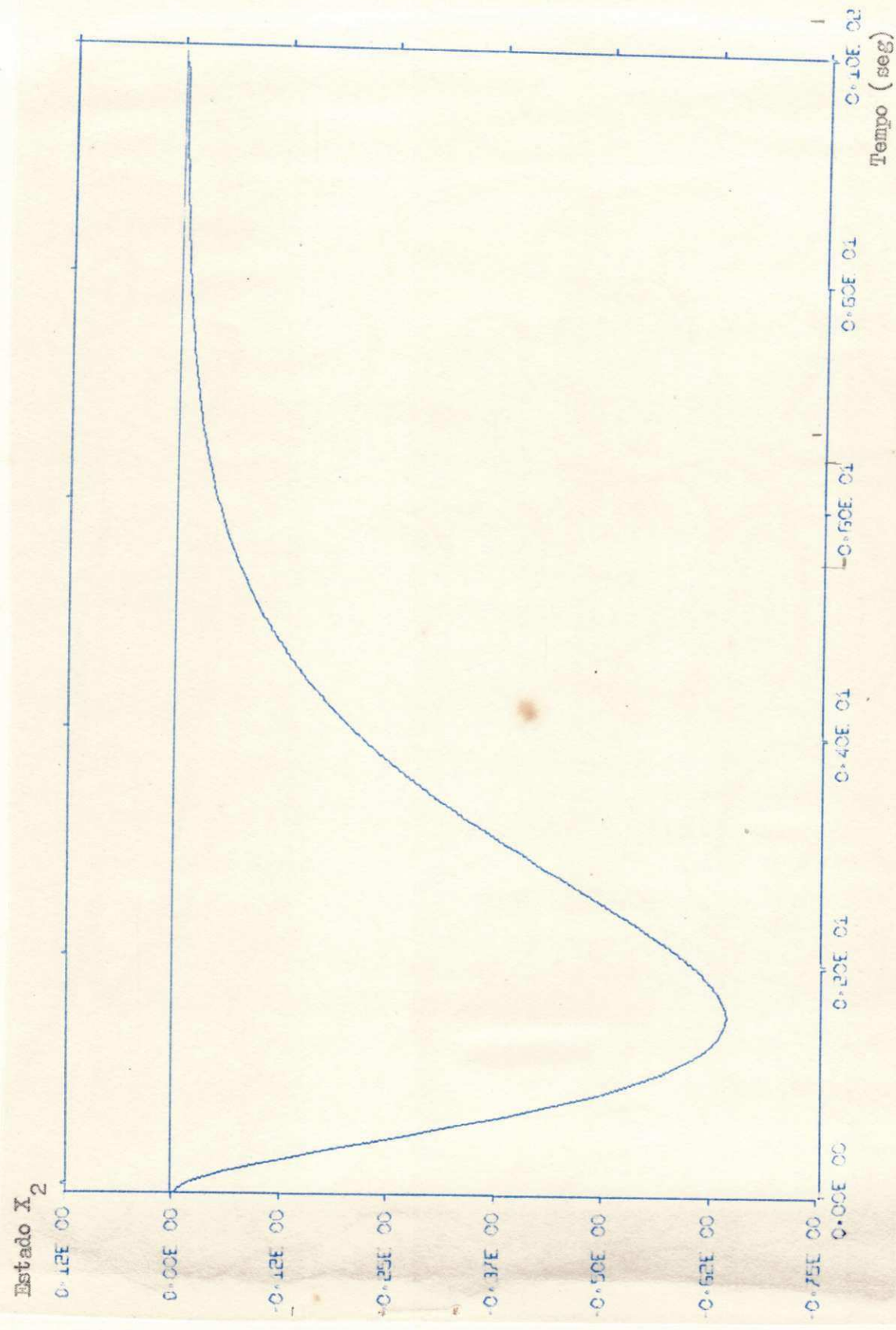


FIGURA 2.4.2. Estado X_2 do Exemplo de RESPT em Função do Tempo

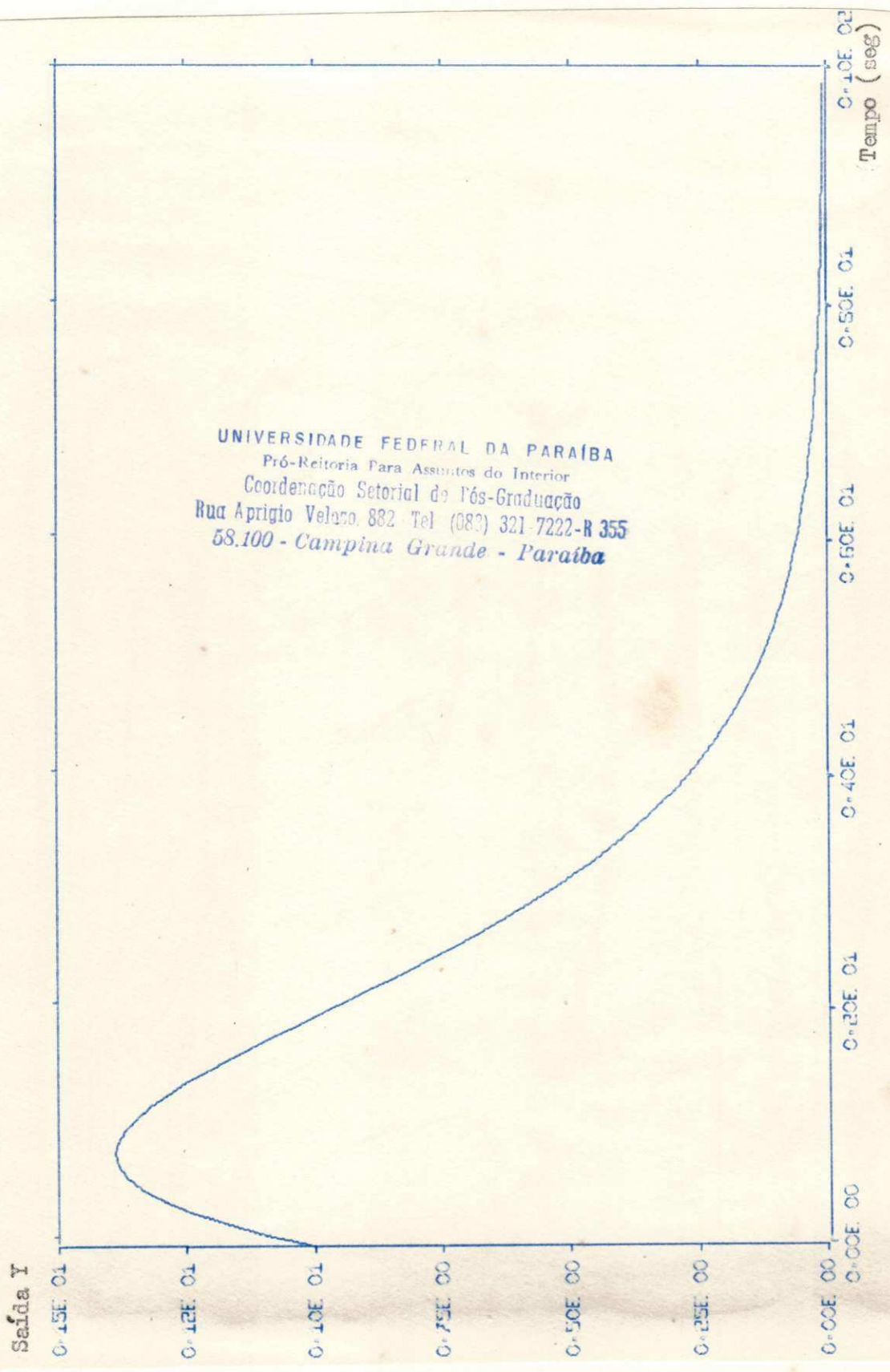


FIGURA 2.4.3. Saída Y do Exemplo de RESPT em função do Tempo

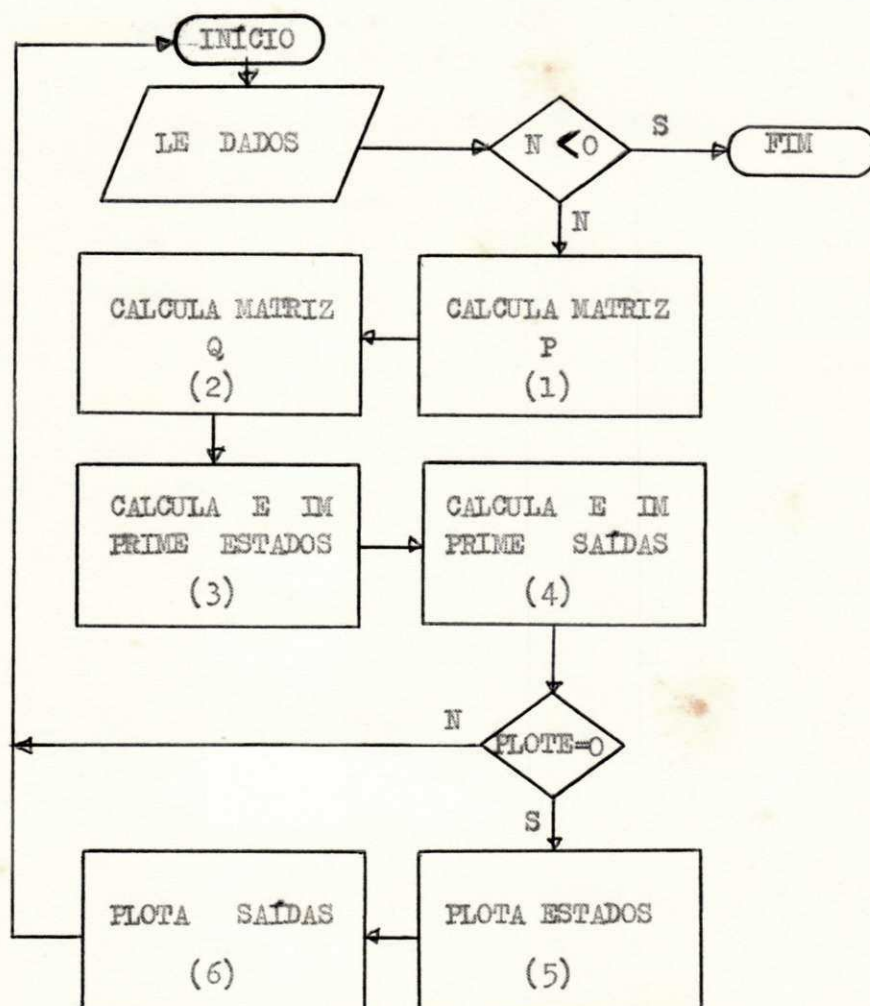


FIGURA 2.3. Fluxograma de RESPT

LISTAGEM DE RESPT:

```

C PROGRAMA ESTADO E RESPOSTA DO SISTEMA NO TEMPO RESPT
C
C SUBPROGRAMAS USADOS LER,IMPR,KDENT,PRODU,SOMA,
C SCMAT,MATEQ,VALMX,ENTR,FAT,
C SCMAT,ADDM,GPLTE
C
C A(N,N),B(N,M),C(IP,N),D(IP,M),U(M,1),X(N,1),Y(IP,1)
C
C DIMENSION A(10,10),B(10,10),C(10,10),D(10,10),QS(10,10),
1 P(10,10),T1(10,10),T2(10,10),T5(10,10),T6(10,10),
2 T(10,10),Q(10,10),X(10,1),Y(10,1),U(10,1),
3 XP(100,10),YP(100,10),XPLF(100),YPLF(100),TPL(100)
IND=1
200 READ(2,99) N,M,IP,DEL,TO,TMAX,PLOTE
99 FORMAT(9X,I2,8X,I2,8X,I2,8X,4E10.3)
IF(N) 202,201,201
202 CALL EXIT
201 WRITE(3,101) IND
101 FORMAT(1H1,///,2X,'PROGRAMA RESPOSTA E ESTADO',
1' DO SISTEMA NO TEMPO',/,2X,'IDENTIFICACAO DO',
2' PROBLEMA RESPT EXEMPLO',I5)
WRITE(3,100)
100 FORMAT(1H0,60(1H*))
WRITE(3,102) N,DEL,TO,TMAX
102 FORMAT(2X,'ORDEM DO SISTEMA',I4,/,2X,
1 'INTERVALO DE TEMPO PARA A SAIDA',F10.5,
2 //2X,'TEMPO INICIAL',F12.4,
3 //,2X,'TEMPO FINAL',F12.4)
WRITE(3,100)
CALL LER(A,N,N)
WRITE(3,103)
103 FORMAT(2X,'MATRIZ A',/)
CALL IMPR(A,N,N)
CALL LER(B,N,M)
WRITE(3,104)
104 FORMAT(/2X,'MATRIZ B',/)
CALL IMPR(B,N,M)
CALL LER(C,IP,N)
WRITE(3,105)
105 FORMAT(/2X,'MATRIZ C',/)
CALL IMPR(C,IP,N)
CALL LER(D,IP,M)
WRITE(3,106)
106 FORMAT(/2X,'MATRIZ D',/)
CALL IMPR(D,IP,M)
CALL LER(X,N,1)
WRITE(3,107)
107 FORMAT(/2X,'CONDICOES INICIAIS...X(0)...',/)

```

LISTAGEM DE RESPT:

(Cont.)

```

CALL IMPR(X,N,1)
WRITE(3,100)
CALL KDENT(QS,N)
CALL KDENT(P,N)
CALL KDENT(T1,N)
CALL KDENT(T2,N)
DELT=0.01
CALL SCMUT(DELT,A,T,N,N)
I=1
2 CALL PRODU(T,T1,T5,N,N,N)
CALL MATEQ(T5,T1,N,N)
TFAT=(1.0/FAT(I))
CALL SCMAT(TFAT,T5,N,N)
CALL ADDM(P,T5,N,N)
CALL PRODU(T,T2,T6,N,N,N)
CALL MATEQ(T6,T2,N,N)
TFAT=(1.0/FAT(I+1))
CALL SCMAT(TFAT,T6,N,N)
CALL ADDM(QS,T6,N,N)
CALL VALMX(T5,T5MAX,N,N)
IF(T5MAX-1.0E-10) 5,5,6
5 CALL VALMX(T6,T6MAX,N,N)
IF(T6MAX-1.0E-10) 7,7,6
6 I=I+1
IF(I-20) 11,11,7
11 GO TO 2
7 CONTINUE
WRITE(3,108) I
108 FORMAT(/2X,'NUMERO DE TERMOS NAS SERIES DE P E Q',/,I7)
WRITE(3,100)
CALL PRODU(QS,B,Q,N,N,M)
CALL SCMAT(DELT,Q,N,M)
WRITE(3,111) N,IP
111 FORMAT(/2X,'ESTADOS E SAIDAS DO SISTEMA',//,2X,
1 'SEQUENCIA NUM. ORDEM,TEMPO,ESTADOS NA ORDEM X(1)...X(',
2 I2,')',/,2X,'EM SEGUIDA AS SAIDAS NA ORDEM Y(1)...Y(',I2,')'//)
TX=TO
TVL=DEL/DELT +1.0
XC=0.0
9 XL=0.0
14 XL=XL+1.0
IF(XL-TVL) 15,15,9
15 CALL ENTR(U,TX)
CALL PRODU(C,X,T5,IP,N,1)
CALL PRODU(D,U,T6,IP,M,1)
CALL SOMA(T5,T6,Y,IP,1)
IF(XL-1) 12,12,13
12 WRITE(3,112) XC,TX,(X(J,1),J=1,N),(Y(J,1),J=1,IP)

```

LISTAGEM DE RESPT:

(Cont.)

```

112 FORMAT(1X,F5.0,2X,E10.3,2X,7(E12.5,2X),/,4X,
1      8(E12.5,2X),/,6X,5(E12.5,2X))

```

C
C

```

      IF(PLOTE) 17,18,17
18 IF(XC-99.0) 21,21,17
21 K=IFIX(XC+1.0)
      DO 19 J=1,N
19 XP(K,J)=X(J,1)
      DO 20 J=1,IP
20 YP(K,J)=Y(J,1)
      TPL(K)=TX
17 XC=XC+1.0

```

C
C

```

13 TX=TX+DELT
      IF(TX-TMAX)16,16,10
16 CALL ENTR(U,TX)

```

C
C

```

      CALL PRODU(Q,U,T5,N,M,1)
      CALL PRODU(P,X,T6,N,N,1)
      CALL SOMA(T5,T6,X,N,1)
      GO TO 14
10 CONTINUE

```

C
C

```

      IF(PLOTE) 22,23,22
23 DO 25 J=1,N
      DO 24 KN=1,K
24 XPLF(KN)=XP(KN,J)
      CALL GPLTE(K,TPL,XPLF)
25 PAUSE
      DO 26 J=1,IP
      DO 27 KN=1,K
27 YPLF(KN)=YP(KN,J)
      CALL GPLTE(K,TPL,YPLF)
      IF(J-IP) 26,28,28
26 PAUSE
22 CONTINUE
28 IND=IND+1
      GO TO 200

```

C
C

```

END

```

UNIVERSIDADE FEDERAL DA PARAIBA
 Pró-Reitoria Para Assuntos do Interior
 Coordenação Setorial de Pós-Graduação
 Rua Aprigio Veloso, 882 - Tel (083) 321-7222 - R 355
 58100 - Campina Grande - Paraíba

CAPÍTULO 3

SISTEMAS DE MALHA FECHADA

3.1. Introdução

Neste Capítulo são discutidos os programas para o estudo de sistemas com realimentação ou sistemas de malha fechada, cujo diagrama de blocos é dado pela Figura 1.2., repetida aqui.

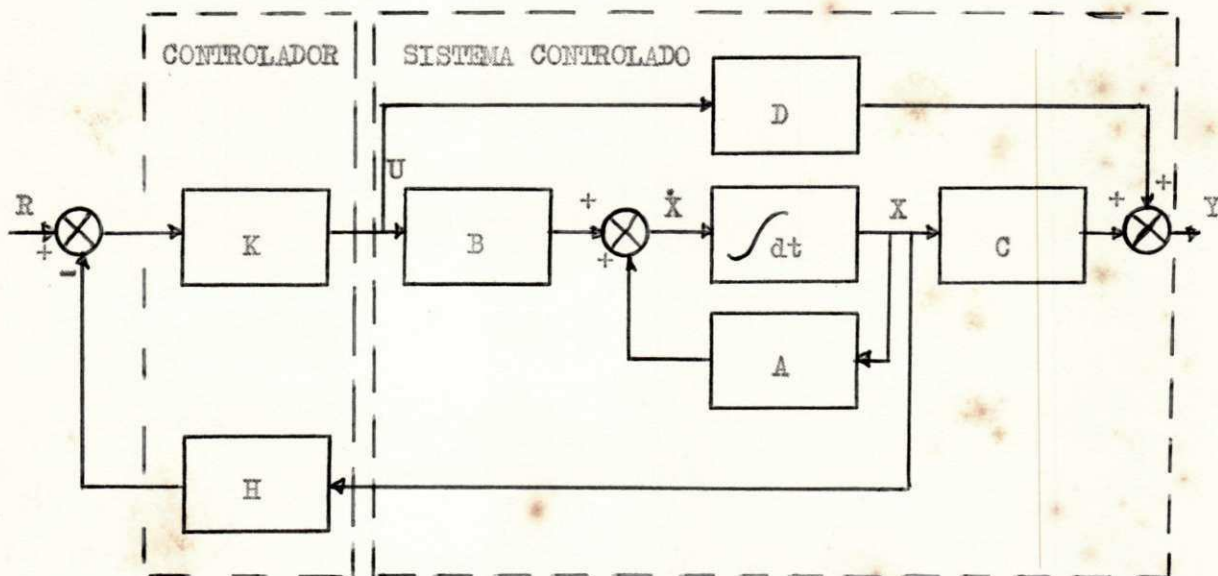


FIGURA 3.1. Diagrama de blocos dos sistemas de malha fechada

Pela Figura 3.1. pode-se escrever: ⁽³⁾

$$U = K [R - HX] \quad (3.1)$$

onde:

$$H = \begin{bmatrix} h_{11} & \dots & h_{1n} \\ h_{21} & & h_{2n} \\ \vdots & & \vdots \\ h_{m1} & & h_{m1} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$K = k I$$

k - ganho linear

Das equações (1.3) e (1.5):

$$\dot{X} = AX + BU \quad (3.2)$$

$$Y = CX + DU \quad (3.3)$$

Substituindo (3.1) em (3.2) tem-se:

$$\dot{X} = AX + B \left[K (R - HX) \right] = Ph X + Bh R \quad (3.4)$$

onde:

$$Ph = A - kBH \quad (3.5)$$

$$Bh = kB \quad (3.6)$$

Analogamente em (3.3):

$$Y = CX + D \left[K (R - HX) \right] = Ch X + Dh R \quad (3.7)$$

onde:

$$Ch = C - kDH \quad (3.8)$$

$$Dh = k D \quad (3.9)$$

Logo, o sistema com realimentação é equivalente a um sistema de malha aberta, representado pelas equações:

$$\dot{X} = Ph X + Bh R \quad (3.10)$$

$$Y = Ch X + Dh R \quad (3.11)$$

Para os sistemas de malha aberta $R = U$ e as equações (3.10) e (3.11) podem ser escritas

$$\dot{X} = Ph X + Bh U \quad (3.12)$$

$$Y = Ch X + Dh U \quad (3.13)$$

O estudo de sistemas de malha fechada pode então ser feito utilizando-se os programas do capítulo anterior, desde que se façam as transformações definidas pelas equações (3.5), (3.6), (3.8) e (3.9) quando então acha-se um sistema de malha aberta equivalente.

No parágrafo 3.2. é apresentado o programa BASMF que acha algumas funções da matriz Ph (mesmas funções que BASMT calcula), em 3.3. o programa COBRM que verifica se o sistema é controlável e observável e em 3.4. o programa RESPF que acha os estados e saídas do sistema.

3.2. Programa BASMF

O programa BASMF permite achar:

- o determinante de Ph
- o inverso de Ph
- o polinômio característico

- os autovalores de Ph
- a matriz resolvente
- a matriz de transição

O programa lê os elementos de A, B, H, o ganho linear K e então faz a transformação definida por (3.5).

Depois disso, o programa é semelhante ao BASMT.

O modo de entrada dos dados é explicado na Tabela 3.2.1.

Depois da identificação do problema e da leitura dos dados, estes valores são impressos e, dependendo do cartão de opções, o determinante de Ph, seu inverso, o polinômio característico, os autovalores de Ph, a matriz resolvente e a matriz de transição é calculada e impressa.

A matriz resolvente e a de transição são impressas de maneira semelhante à do programa BASMT.

Exemplo para ilustrar o uso do programa

Dados:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 3 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & 3 \\ 4 & 0 & 0 \\ 1 & -1 & -2,5 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ganho linear $k = 2$

TABELA 3.2.1.

DADOS DE ENTRADA PARA BASMF

Cartão Número	Coluna Número	Formato	Valor Per furado	Nome da Variável, Descrição
1	1 - 20 21 - 22 31 - 32	A2 I2 I2	Título 1,2,..., 10 1,2,..., 10	Nome ou descrição arbitrária N - ordem da matriz A M - n° de colunas de B
2 até N + 1	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de a_{11} Valor de a_{12} :	a_{11} a_{12} :
N + 2	1 1 2 2 3 3 4 4 5 5 6 6	II II II II II II II II II II II II	0 1,2,..., 9 0 1,2,..., 9 1,2,..., 9 0 0 1,2,..., 9 0 1,2,..., 9 0 1,2,..., 9	IDET - calcula determinante IDET - suprime determinante INV - calcula inverso INV - suprime inverso NRM - suprime resolvente NRM - calcula resolvente ICP - calcula pol. característi. ICP - sup. pol. característico IEIG - calcula autovalores IEIG - suprime autovalores ISIM - calc. matriz transição ISIM - sup. matriz transição
N + 3 até 2N + 2	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de b_{11} Valor de b_{12} :	b_{11} b_{12} :
2N + 3 até 2N+M+ 2	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de h_{11} Valor de h_{12} :	h_{11} h_{12} :
2N+M+ 3	1 - 10	E10.3	Valor de GANHO	Ganho - valor do ganho linear
2N+M+ 4				Cartão em branco para o programa para outro sistema repetir tudo

É necessário o cálculo de todas as funções e, para isso, os dados de entrada estão na Tabela 3.2.2.

TABELA 3.2.2.

DADOS DE ENTRADA PARA O EXEMPLO DE BASMF

Cartão Número	Coluna Número																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
1	B	A	S	M	F		E	X	E	M	P	L	O	1							3											3		
2				0	.	0							1	.	0										0	.	0							
3				0	.	0							0	.	0										1	.	0							
4				-	2	.	0						-	3	.	0									-	3	.	0						
5				B	R	A	N	C	O																									
6				0	.	0							1	.	0										3	.	0							
7				4	.	0							0	.	0										0	.	0							
8				1	.	0							-	1	.	0									-	2	.	5						
9				1	.	0							0	.	0										0	.	0							
10				0	.	0							1	.	0										0	.	0							
11				0	.	0							0	.	0										1	.	0							
12					2	.	0																											
13				B	R	A	N	C	O																									

Os resultados são dados na Tabela 3.2.3.

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

TABELA 3.2.3.

SAÍDA DO EXEMPLO DE BASMF

```

PROGRAMA MATRIZ BASICA COM REALIMENTACAO
IDENTIFICACAO DO PROBLEMA          BASMFB EXEMPLO  3
*****

A MATRIZ A
  0.000E 00  0.100E 01  0.000E 00
  0.000E 00  0.000E 00  0.100E 01
 -0.200E 01 -0.300E 01 -0.300E 01

A MATRIZ B
  0.400E 01  0.000E 00  0.000E 00
  0.100E 01 -0.100E 01 -0.250E 01
  0.100E 01  0.000E 00  0.000E 00

A MATRIZ H
  0.000E 00  0.100E 01  0.000E 00
  0.000E 00  0.000E 00  0.100E 01
  0.200E 01  0.000E 00  0.000E 00

O GANHO LINEAR DO SISTEMA
0.00000E 00

A MATRIZ PH
  0.000E 00  0.100E 01  0.000E 00
  0.000E 00  0.000E 00  0.100E 01
 -0.200E 01 -0.300E 01 -0.300E 01

*****

O DETERMINANTE DA MATRIZ PH
-0.200E 01

O INVERSO DA MATRIZ PH
 -0.150E 01 -0.150E 01 -0.500E 00
  0.100E 01  0.000E 00  0.000E 00
  0.000E 00  0.100E 01  0.000E 00
*****

```

TABELA 3.2.3.

(Cont.)

OS COEFICIENTES DA MATRIZ DO NUMERADOR DA MATRIZ RESOLVENTE

OS COEFICIENTES DA MATRIZ DE S**2

0.100E 01	0.000E 00	0.000E 00
0.000E 00	0.100E 01	0.000E 00
0.000E 00	0.000E 00	0.100E 01

OS COEFICIENTES DA MATRIZ DE S**1

0.300E 01	0.100E 01	0.000E 00
0.000E 00	0.300E 01	0.100E 01
-0.200E 01	-0.300E 01	0.000E 00

OS COEFICIENTES DA MATRIZ DE S**0

0.300E 01	0.300E 01	0.100E 01
-0.200E 01	0.000E 00	0.000E 00
0.000E 00	-0.200E 01	0.000E 00

POLINOMIO CARACTERISTICO EM POTENCIAS CRESCENTES DE S

0.200E 01 0.300E 01 0.300E 01 0.100E 01

OS AUTOVALORES DA MATRIZ PH

PARTE REAL	PARTE IMAGINARIA
-0.50000E 00	-0.86602E 00
-0.50000E 00	0.86602E 00
-0.20000E 01	0.00000E 00

TABELA 3.2.3.

(Cont.)

OS ELEMENTOS DA MATRIZ TRANSICAO DE ESTADO

OS COEFICIENTES MATRICIAIS DE $\text{EXP}(-0.500000E 00)T * \text{COS}(0.866025E 00)T$

0.666E 00	-0.333E 00	-0.333E 00
0.666E 00	0.166E 01	0.666E 00
-0.133E 01	-0.133E 01	-0.333E 00

OS COEFICIENTES MATRICIAIS DE $\text{EXP}(-0.500000E 00)T * \text{SEN}(0.866025E 00)T$

0.115E 01	0.173E 01	0.577E 00
-0.115E 01	-0.577E 00	-0.296E-06
-0.953E-06	-0.115E 01	-0.577E 00

OS COEFICIENTES MATRICIAIS DE $\text{EXP}(-0.200000E 01)T$

0.333E 00	0.333E 00	0.333E 00
-0.666E 00	-0.666E 00	-0.666E 00
0.133E 01	0.133E 01	0.133E 01

Os seguintes subprogramas são utilizados:

1 - PCMRS	4 - RAPOL	7 - PRODU
2 - EQLSI	5 - DET	8 - SCMAT
3 - XMTRE	6 - MATSU	9 - MATEQ

O fluxograma de BASMP é dado pela Figura 3.2.

As operações nos quadrinhos numerados são respectivamente:

(1) - Cálculo da matriz Ph do sistema de malha aberta equivalente dado pela transformação (3.5) feito pelas subrotinas PRODU, SCMAT, MATSU, MATEQ

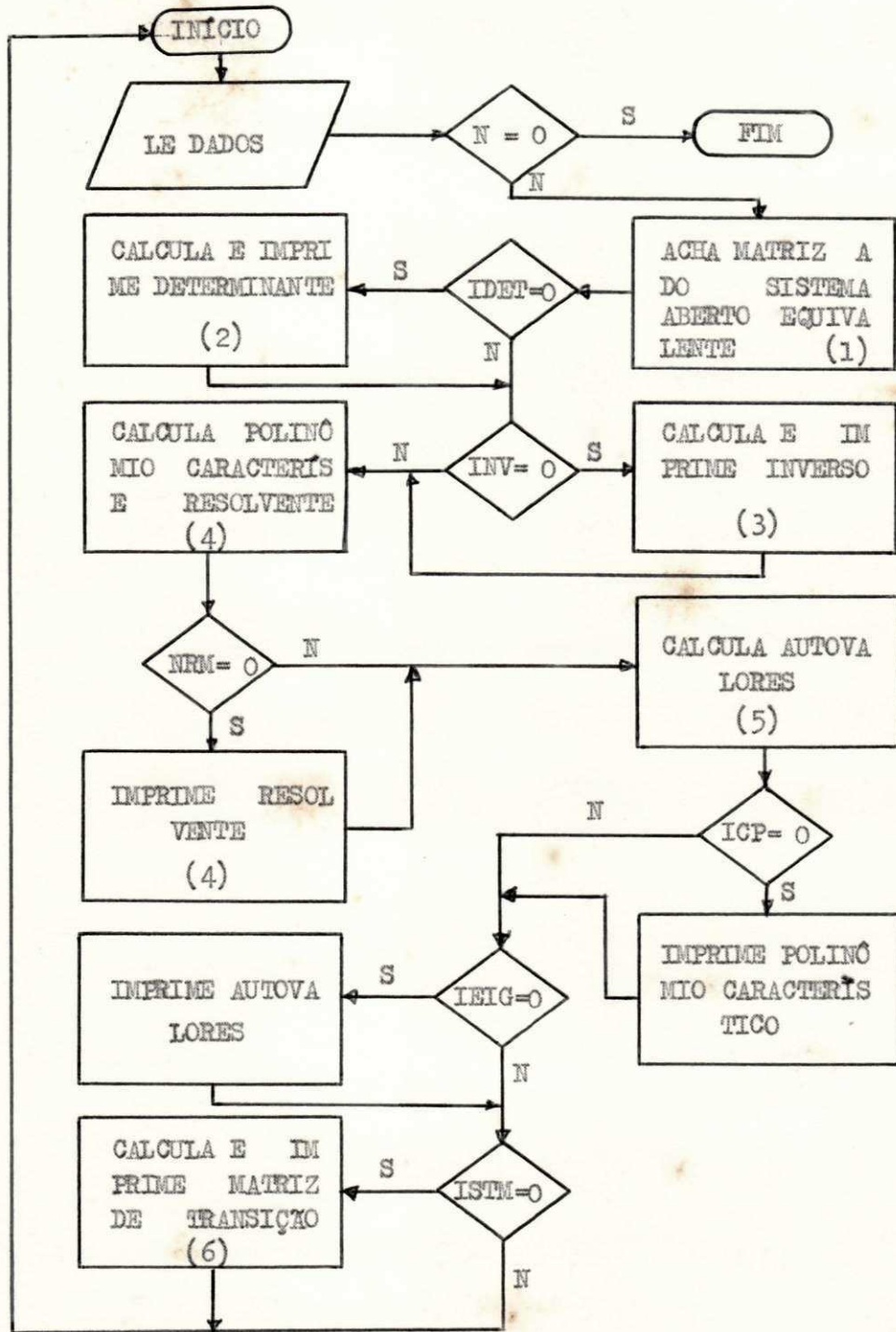


FIGURA 3.2. Fluxograma de BASMF

- (2) - Cálculo do determinante de Ph através da função DET
- (3) - Cálculo do inverso de Ph através da subrotina EQLSI
- (4) - Cálculo do polinômio característico e da matriz resolvente feito pela subrotina PCMRS
- (5) - Cálculo dos autovalores de Ph feito pela subrotina RAPOL
- (6) - Cálculo da matriz de transição feito pela subrotina XMTRE

LISTAGEM DE BASMF

C

```

DIMENSION A(10,10),EIGR(10),EIGI(10),C(11),AINV(10,10),
1      NAME(5),H(10,10),PH(10,10),T1(10,10),B(10,10)
COMMON CHI(10,10,10)
2001 FORMAT(10A2,I2,8X,I2,8X,I2)
2002 FORMAT(10E10.3)
2003 FORMAT(6X,10(E10.3,1X))
2004 FORMAT(1H0,5X,16HA MATRIZ A      ,/)
2005 FORMAT(1H0,5X,'POLINOMIO CARACTERISTICO EM POTENCIAS CRESCENTES DE
1  S',/)
2006 FORMAT(/5X,'OS AUTOVALORES DA MATRIZ PH'/)
2007 FORMAT(9X,'PARTE REAL',9X'PARTE IMAGINARIA',/)
2008 FORMAT(1H1,5X,'PROGRAMA MATRIZ BASICA COM REALIMENTACAO')
2009 FORMAT(6X,'IDENTIFICACAO DO PROBLEMA ',5X,10A2)
2010 FORMAT(/5X,'O DETERMINANTE DA MATRIZ PH'/)
2011 FORMAT(/5X,'O INVERSO DA MATRIZ PH'/)
2012 FORMAT(1H0,60(1H*))
2013 FORMAT(6I1)
2014 FORMAT(8X,2(E12.5,10X))
2015 FORMAT(/5X,'A MATRIZ B'/)
2016 FORMAT(/5X,'O GANHO LINEAR DO SISTEMA'/5X,E12.5)
2017 FORMAT(/5X,'A MATRIZ H'/)
2018 FORMAT(/5X,'A MATRIZ PH'/)
4 READ(2,2001) (NAME(I),I=1,10),N,M
IF(N) 13,10,13

13 DO 1 I=1,N
1 READ(2,2002) (A(I,K),K=1,N)
READ(2,2013) IDET,INV,NRM,ICP,IEIG,ISTM
WRITE(3,2008)
WRITE(3,2009) (NAME(I),I=1,10)
WRITE(3,2012)
WRITE(3,2004)
DO 2 I=1,N
2 WRITE(3,2003) (A(I,K),K=1,N)

```

LISTAGEM DE BASMF

(Cont.)

```

DO 50 I=1,N
50 READ(2,2002) (B(I,K),K=1,M)
DO 51 I=1,M
51 READ(2,2002) (H(I,K),K=1,N)
READ(2,2002) GANHO
WRITE(3,2015)
DO 52 I=1,N
52 WRITE(3,2003) (B(I,K),K=1,M)
WRITE(3,2017)
DO 53 I=1,M
53 WRITE(3,2003) (H(I,K),K=1,N)
WRITE(3,2016) GANHO
CALL PRODU(B,H,T1,N,M,N)
CALL SCMAT(GANHO,T1,N,N)
CALL MATSU(A,T1,PH,N,N)
CALL MATEQ(PH,A,N,N)
WRITE(3,2018)
DO 54 I=1,N
54 WRITE(3,2003) (PH(I,K),K=1,M)
WRITE(3,2012)
IF(IDET)5,6,5
6 D=DET(A,N)
WRITE(3,2010)
WRITE(3,2003) D
5 IF(INV)15,7,15
7 WRITE(3,2011)
CALL EQLSI(A,C,N,AINV,C,IERR)
IF(IERR)8,15,8
8 DO 20 I=1,N
20 WRITE(3,2003) (AINV(I,J),J=1,N)
15 CALL PCMRS(A,N,C,NRM)
CALL RAPOL(N,C,EIGR,EIGI,+1)
IF(ICP)30,9,30
9 WRITE(3,2012)
WRITE(3,2005)
NN=N+1
WRITE(3,2003) (C(I),I=1,NN)
30 IF(IEIG)35,11,35
11 WRITE(3,2012)
WRITE(3,2006)
WRITE(3,2007)
DO 3 I=1,N
3 WRITE(3,2014) EIGR(I),EIGI(I)
35 IF(ISTM)25,12,25
12 CALL XMTRE(N,A,EIGR,EIGI,ISTM)
25 GO TO 4
10 CALL EXIT
END

```

3.3. Programa COBRM

O programa COBRM permite:

- Verificar a observabilidade do sistema
- Verificar a controlabilidade do sistema

O programa lê as matrizes A, B, C, D, H, o ganho linear K e faz as transformações definidas pelas equações (3.5), (3.6) e (3.8) para achar as matrizes Ph, Bh e Ch, imprime os dados de entrada e as matrizes Ph, Bh e Ch.

Depois disso, o programa monta as matrizes M e N definidas por (1.23) e (1.24) respectivamente e acha o "rank" delas, verificando, então, a observabilidade e controlabilidade do sistema, de acordo com o item (1.6).

A disposição dos dados de entrada é explicada na Tabela 3.3.1.

Exemplo para ilustrar o uso do programa

Dados:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}$$

$$k \quad - \quad \text{ganho linear} = 1.0$$

TABELA 3.3.1.

DADOS DE ENTRADA PARA COBRM

Cartão Número	Coluna Número	Formato	Valor Perfurado	Nome da Variável, Descrição
1	1 - 2 10 - 11 20 - 21	I2 I2 I2	1,2,...., 10 1,2,...., 10 1,2,...., 10	N - ordem da matriz A M - número de colunas de B P - número de linhas de C
2 até N + 1	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de a_{11} Valor de a_{12} :	a_{11} a_{12} :
N + 2 até 2N + 1	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de b_{11} Valor de b_{12} :	b_{11} b_{12} :
2N + 2 até 2N+P+1	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de c_{11} Valor de c_{12} :	c_{11} c_{12} :
2N+P+2 até 2N+2P+1	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de d_{11} Valor de d_{12} :	d_{11} d_{12} :
2N+2P+2 até 2N+2P+m+1	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de h_{11} Valor de h_{12} :	h_{11} h_{12} :
2N+2P+M+2	1 - 10	E10.3	Valor de k	k - ganho linear
2N+2P+M+3	1 - 2	I2	-1,-2,....,-9	Número inteiro negativo para o programa; para outro sistema igual ao primeiro repetindo tudo

A disposição dos dados de entrada está na Tabela 3.3.2.

TABELA 3.3.2.

DADOS DE ENTRADA DO EXEMPLO DE COBRM

Cartão Número	Coluna Número																													
	-	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1		2								1										1										
2					0.0							1.0																		
3					-2.0							-3.0																		
4					0.0							1.0																		
5					1.0							-2.0																		
6					1.0							0.0																		
7					-1.0							0.0																		
8					1.0							-2.0																		
9					-1.0							3.0																		
10					1.0																									
11		-1																												

Os resultados do programa são dados na Tabela 3.3.3.

TABELA 3.3.3.

SAÍDA DO EXEMPLO DE COBRM

EXEMPLO DE CONTROLABILIDADE E OBSERVABILIDADE
COM REALIMENTAÇÃO
IDENTIFICAÇÃO DO PROBLEMA COBRM EXEMPLO 1

MATRIZ A

0.000E 00 0.100E 01
-0.200E 01 -0.300E 01

MATRIZ B

0.000E 00 0.100E 01
0.100E 01 -0.200E 01

TABELA 3.3.3.

(Cont.)

MATRIZ C

0.100E 01 0.000E 00

MATRIZ D

0.200E 06 0.100E 07

MATRIZ H

-0.100E 01 0.000E 00

0.100E 01 -0.200E 01

GANHO LINEAR DO SISTEMA

0.10000E 01

MATRIZ PH

-0.100E 01 0.300E 01

0.100E 01 -0.700E 01

MATRIZ BH

0.000E 00 0.100E 01

0.100E 01 -0.200E 01

MATRIZ CH

-0.799E 06 0.200E 07

A ORDEM DO SISTEMA E 2

O SISTEMA E CONTROLAVEL, O RANK DE M E 2

O SISTEMA E OBSERVAVEL, O RANK DE N E 2

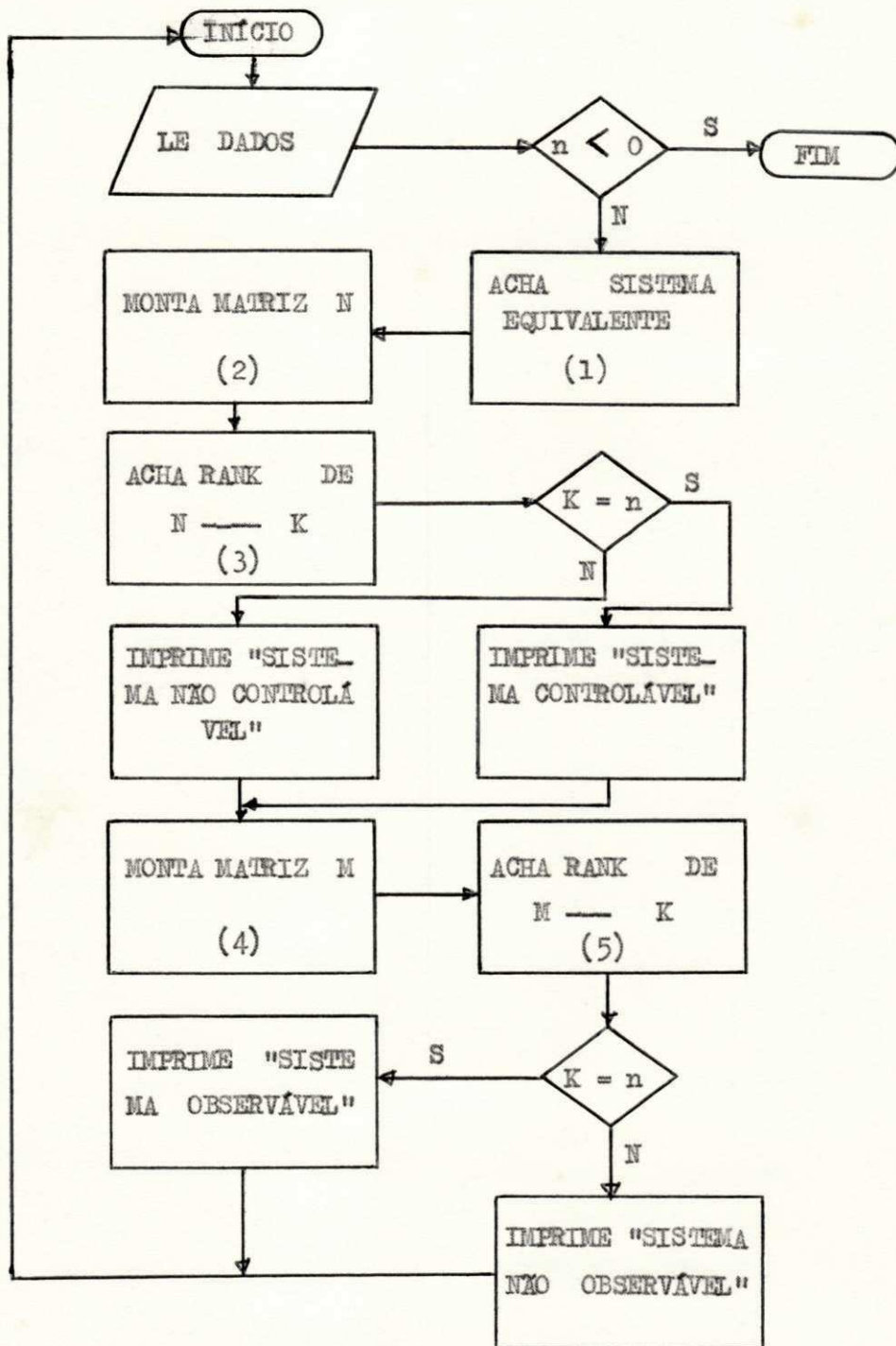


FIGURA 3.3. Fluxograma de COBRM

Verifica-se que o sistema é observável e controlável.

Os subprogramas utilizados são:

1 - SOMA	4 - MATEQ	7 - ADDIC	10 - MATSU
2 - PRODU	5 - TRNPZ	8 - LER	11 - SCMAT
3 - RANK	6 - KDENT	9 - IMPR	

O fluxograma de COBRM é dado pela Figura 3.3.

As operações dos quadrinhos numerados são:

- (1) - Cálculo de Ph , Eh e Ch do sistema de malha aberta equivalente dadas pelas equações (3.5), (3.6) e (3.8) respectivamente
- (2) - Montagem da matriz N de acordo com a fórmula (1.24) através da subrotina ADDIC
- (3) - Cálculo de "rank" de N através da subrotina RANK
- (4) - Montagem da matriz M de acordo com a fórmula (1.23) através da subrotina ADDIC
- (5) - Cálculo do "rank" de M através da subrotina RANK

LISTAGEM DE COBRM

```

INTEGER P
DIMENSION XM(10,100),XN(10,100),A(10,10),B(10,10),C(10,10),
1      E(10,10),F(10,10),G(10,10),Cl(10,10),Dl(10,10),
2      D(10,10),H(10,10)
IND=1
12 READ(2,101) N,M,P
101 FORMAT(I2,7X,I2,8X,I2)
   IF(N) 13,14,14
14 WRITE(3,107) IND
107 FORMAT(1H1,///,2X,'EXEMPLO DE CONTROLABILIDADE E OBSERVABILIDADE',
1/2X,'COM REALIMENTACAO'/2X,'IDENTIFICACAO DO PROBLEMA',
2' COBRM EXEMPLO',I4)
   WRITE(3,108)
108 FORMAT(1H0,60(1H*),/)
   CALL LER(A,N,N)
   CALL LER(B,N,M)
   CALL LER(C,P,N)
   CALL LER(D,P,M)
   CALL LER(H,M,N)

```

LISTAGEM DE COBRM

(Cont.)

```

      READ(2,120) GANHO
120  FORMAT(E10.3)
      WRITE(3,109)
109  FORMAT(2X,'MATRIZ  A'//)
      CALL IMPR(A,N,N)
      WRITE(3,111)
111  FORMAT(/,2X,'MATRIZ  B',//)
      CALL IMPR(B,N,M)
      WRITE(3,112)
112  FORMAT(/,2X,'MATRIZ  C',//)
      CALL IMPR(C,P,N)
      WRITE(3,114)
114  FORMAT(/2X,'MATRIZ  D',//)
      CALL IMPR(D,P,M)
      WRITE(3,115)
115  FORMAT(/2X,'MATRIZ  H',//)
      CALL IMPR(H,M,N)
      WRITE(3,116) GANHO
116  FORMAT(/,2X,'GANHO LINEAR DO SISTEMA'/2X,E12.5)
      CALL PRODU(B,H,F,N,M,N)
      CALL SCMAT(GANHO,F,N,N)
      CALL MATSU(A,F,G,N,N)
      CALL MATEQ(G,A,N,N)
      WRITE(3,117)
117  FORMAT(/2X,'MATRIZ  PH',//)
      CALL IMPR(A,N,N)
      CALL SCMAT(GANHO,B,N,M)
      WRITE(3,118)
118  FORMAT(/2X,'MATRIZ  BH',//)
      CALL IMPR(B,N,M)
      CALL PRODU(D,H,F,P,M,N)
      CALL SCMAT(GANHO,F,P,N)
      CALL MATSU(C,F,G,P,N)
      CALL MATEQ(G,C,P,N)
      WRITE(3,119)
119  FORMAT(/2X,'MATRIZ  CH',//)
      CALL IMPR(C,P,N)
      WRITE(3,108)
      WRITE(3,113) N
113  FORMAT(1H0,2X,'A ORDEM DO SISTEMA E',15)
      CALL KIDENT(F,N)
      CALL KIDENT(G,N)
      DO 4 I=1,N
      CALL PRODU(G,B,E,N,N,M)
      CALL ADDIC(XM,E,N,M,I)
      CALL PRODU(F,A,G,N,N,N)
      CALL MATEQ(G,F,N,N)
4  CONTINUE

```

LISTAGEM DE COBRM

(Cont.)

```
MM=N*M
CALL RANK(XM,N,MM,K)
IF(K-N) 6,5,6
5 WRITE(3,103) K
103 FORMAT(/3X,'O SISTEMA E CONTROLAVEL,O RANK DE M E',I5)
GO TO 7
6 WRITE(3,104) K
104 FORMAT(/3X,'O SISTEMA NAO E CONTROLAVEL,O RANK DE M E',I5)
7 CONTINUE
CALL KIDENT(F,N)
CALL KIDENT(G,P)
CALL TRNPZ(C,C1,P,N)
DO 8 I=1,N
CALL TRNPZ(G,D1,N,N)
CALL PRODU(D1,C1,E,N,N,P)
CALL ADDIC(XN,E,N,P,I)
CALL PRODU(F,A,G,N,N,N)
CALL MATEQ(G,F,N,N)
8 CONTINUE
MM=N*P
CALL RANK(XN,N,MM,K)
IF(K-N) 9,10,9
9 WRITE(3,105) K
105 FORMAT(/3X,'O SISTEMA NAO E OBSERVAVEL, O RANK DE N E',I5)
GO TO 11
10 WRITE(3,106) K
106 FORMAT(/3X,'O SISTEMA E OBSERVAVEL,O RANK DE N E',I5)
11 CONTINUE
IND=IND+1
GO TO 12
13 CALL EXIT
END
```

3.4. Programa RESPF

O programa RESPF permite:

- Cálculo dos estados do sistema
- Cálculo das saídas do sistema
- Obtenção dos gráficos dos estados do sistema
- Obtenção dos gráficos das saídas do sistema

O programa lê as matrizes A, B, C, D, H, as condições iniciais, o ganho linear e então transforma o sistema de malha fechada em um de malha aberta, equivalente através de (3.5), (3.6) e (3.9).

Depois disso, o programa é semelhante ao programa RESPT.

Os dados de entrada do programa são especificados na Tabela número 3.4.1.

A entrada U (T) é definida como no programa RESPT.

Exemplo para ilustrar o uso do programa

Dados:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 \end{bmatrix}$$

Condições iniciais:

$$x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

TABELA 3.4.1.

DADOS DE ENTRADA PARA RESPF

Cartão Número	Coluna Número	Formato	Valor Perfurado	Nome da Variável, Descrição
1	10 - 11	I2	1,2,..., 10	N - ordem da matriz A
	20 - 21	I2	1,2,..., 10	M - número de colunas de B
	30 - 31	I2	1,2,..., 10	IP - número de linhas de C
	40 - 49	E10.3	Valor de DEL	DEL - interv. de tempo saída 0,1
	50 - 59	E10.3	Valor de TO	TO - tempo inicial
	60 - 69	E10.3	Vr. de TMAX	TMAX - tempo final
	70 - 79	E10.3	Vr. de FLOTE	FLOTE - nº real $\neq 0$ - sup. gráficos
2 até N + 1	1 - 10	E10.3	Valor de a_{11}	a_{11}
	11 - 20	E10.3	Valor de a_{12}	a_{12}
	etc	:	:	:
N + 2 até 2N + 1	1 - 10	E10.3	Valor de b_{11}	b_{11}
	11 - 20	E10.3	Valor de b_{12}	b_{12}
	etc	:	:	:
2N + 2 até 2N+IP+ 1	1 - 10	E10.3	Valor de c_{11}	c_{11}
	11 - 20	E10.3	Valor de c_{12}	c_{12}
	etc	:	:	:
2N+IP+ 2 até 2N+2IP+1	1 - 10	E10.3	Valor de d_{11}	d_{11}
	11 - 20	E10.3	Valor de d_{12}	d_{12}
	etc	:	:	:
2N+2IP+2 2N+2IP+3 :	1 - 10	E10.3	Valor $x_{11}(0)$	$x_{11}(0)$
	1 - 10	E10.3	Valor $x_{12}(0)$	$x_{12}(0)$
	etc	:	:	:
3N+2IP+2 até 3N+2IP+1	1 - 10	E10.3	Valor de h_{11}	h_{11}
	11 - 20	E10.3	Valor de h_{12}	h_{12}
	etc	:	:	:
3N+2IP+3	1 - 10	E10.3	Vr. de GANHO	GANHO - Ganho linear
3N+2IP+4	10 - 11	I2	-1,-2,..., -9	Número inteiro negativo para o programa; para outro sistema igual ao primeiro, repetindo tudo.

tempo inicial = 0,0 segundos
tempo final = 22,0 segundos
intervalo de tempo para a saída = 0,2 segundos
ganho linear = 1,0

De acordo com as condições acima, os dados de entrada estão na Tabela 3.4.2.

Os resultados do programa estão na Tabela 3.4.3.

Os gráficos dos estados e saídas estão nas Figuras 3.4.1., 3.4.2 e 3.4.3.

O fluxograma é dado na Figura 3.4.4.

As operações nos quadrinhos numerados são:

- (1) - Cálculo do sistema de malha aberta equivalente através das relações (3.5), (3.6), (3.8) e (3.9) e das subrotinas PRODU, SCMAT, MATSU, MATEQ
- (2) - Cálculo da matriz P de acordo com a fórmula (2.9) através das subrotinas KDENT, SCMUT, PRODU, MATEQ, SCMAT, VALMX
- (3) - Cálculo da matriz Q de acordo com a fórmula (2.10) através das subrotinas KDENT, SCMUT, PRODU, MATEQ, SCMAT, VALMX
- (4) - Cálculo dos estados de acordo com a fórmula (2.6) através das subrotinas ENTR, PRODU, SCMAT, SOMA
- (5) - Cálculo das saídas de acordo com a fórmula (2.11) através das subrotinas PRODU, SOMA, ENTR, SCMAT
- (6) - Obtenção dos gráficos dos estados através da subrotina GPLTE
- (7) - Obtenção dos gráficos das saídas através da subrotina GPLTE

TABELA 3.4.3.

RESULTADOS DO EXEMPLO DE RESPF

```

PROGRAMA RESPOSTA E ESTADO DO SISTEMA NO TEMPO COM REALIMENTACAO
IDENTIFICACAO DO PROBLEMA          RESPF EXEMPLO    1
*****
ORDEM DO SISTEMA      2
INTERVALO DE TEMPO PARA A SAIDA  0.20000
TEMPO INICIAL        0.0000
TEMPO FINAL          22.0000
*****
MATRIZ  A
  0.000E 00  0.100E 01
-0.200E 01 -0.300E 01
MATRIZ  B
  0.200E 01
  0.000E 00
MATRIZ  C
  0.100E 01  0.000E 00
MATRIZ  D
  0.100E 01
CONDICOES INICIAIS...X(0)...
  0.000E 00
  0.000E 00
MATRIZ  H
  0.000E 00  0.100E 01
GANHO LINEAR DO SISTEMA
0.10000E 01

```

TABELA 3.4.3.

(Cont.)

MATRIZ PH

0.000E 00 -0.100E 01
 -0.200E 01 -0.300E 01

MATRIZ BH

0.200E 01
 0.000E 00

MATRIZ CH

0.100E 01 -0.100E 01

MATRIZ DH

0.100E 01

NUMERO DE TERMOS NAS SERIES DE P E Q

6

ESTADOS E SAIDAS DO SISTEMA

SEQUENCIA NUM. ORDEM, TEMPO, ESTADOS NA ORDEM X(1)...X(2)

EM SEGUIDA AS SAIDAS NA ORDEM Y(1)...Y(1)

0.	0.000E 00	0.00000E 00	-0.00000E 00	0.10000E 01
1.	0.199E 00	0.36511E 00	-0.61837E-01	0.12456E 01
2.	0.399E 00	0.68561E 00	-0.19753E 00	0.15534E 01
3.	0.599E 00	0.98327E 00	-0.36421E 00	0.18962E 01
4.	0.799E 00	0.12718E 01	-0.54344E 00	0.22646E 01
5.	0.999E 00	0.15610E 01	-0.72851E 00	0.26574E 01
6.	0.119E 01	0.18583E 01	-0.91823E 00	0.30777E 01
7.	0.139E 01	0.21700E 01	-0.11140E 01	0.35306E 01
8.	0.159E 01	0.25020E 01	-0.13184E 01	0.40223E 01
9.	0.179E 01	0.28599E 01	-0.15346E 01	0.45598E 01
10.	0.199E 01	0.32493E 01	-0.17659E 01	0.51506E 01

ESTADO x_1

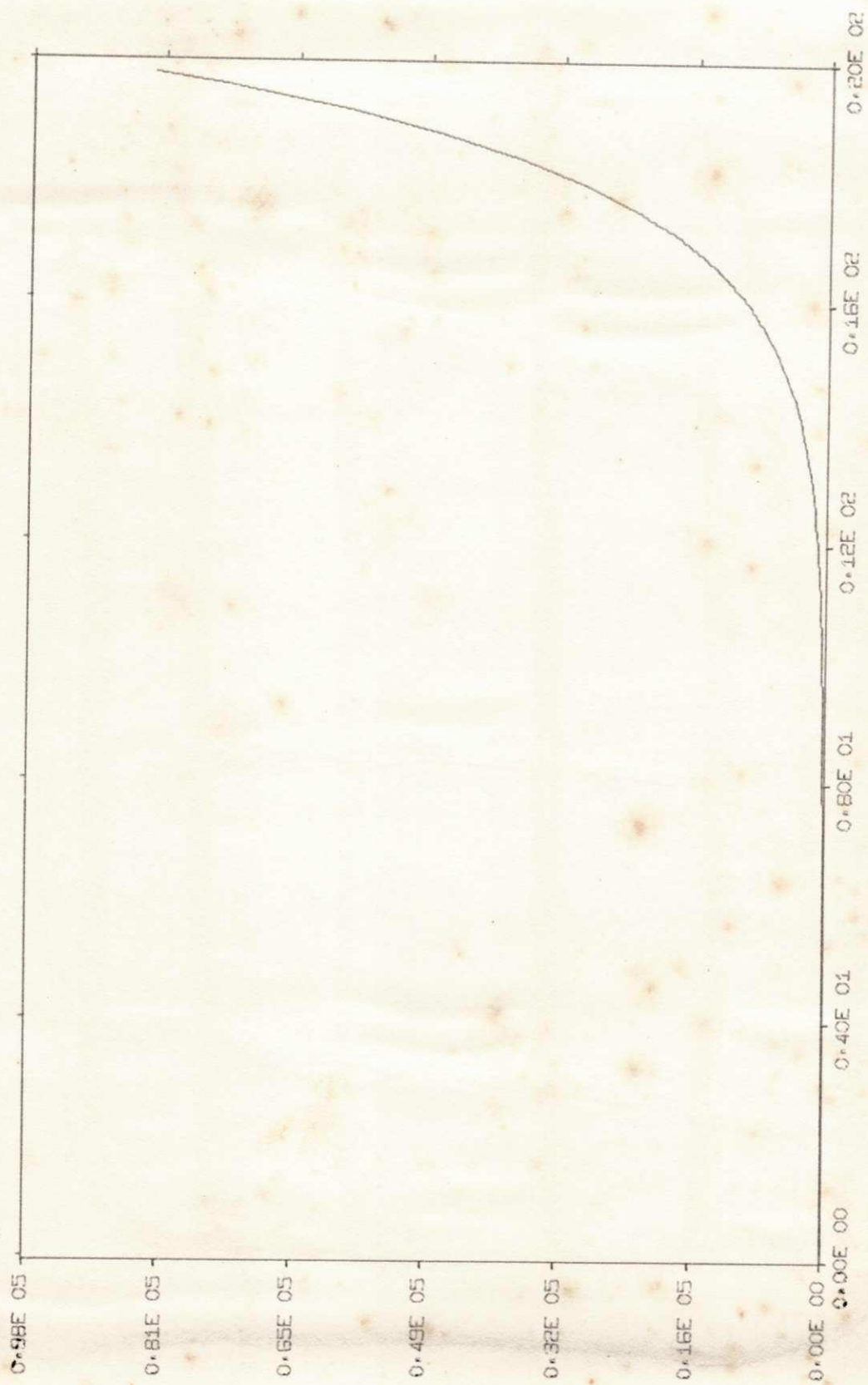


FIGURA 3.4.1. ESTADO x_1 EM FUNÇÃO DO TEMPO DO EXEMPLO DE RESPF

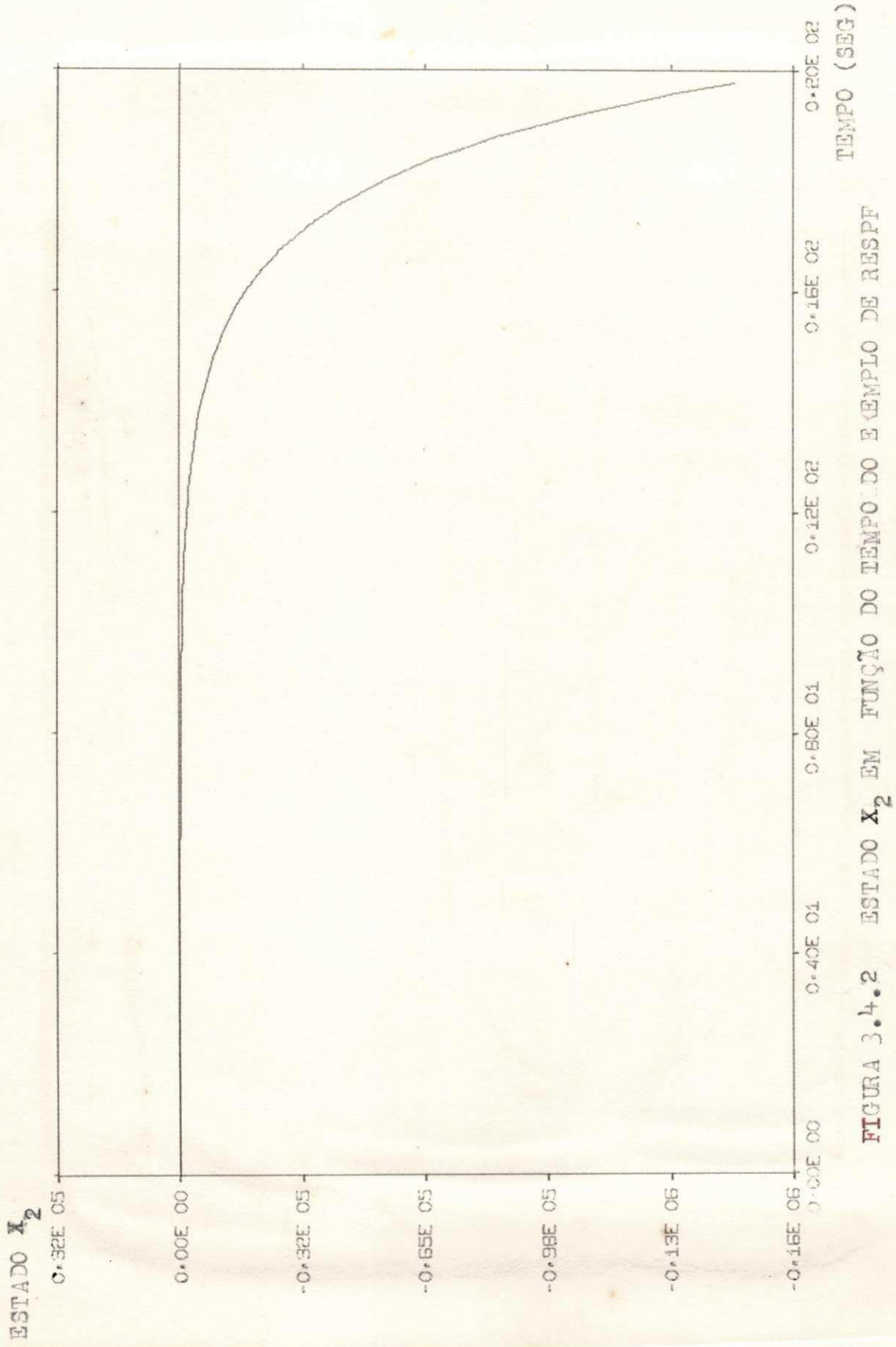


FIGURA 3.4.2 ESTADO X_2 EM FUNÇÃO DO TEMPO DO EXEMPLO DE RESPF

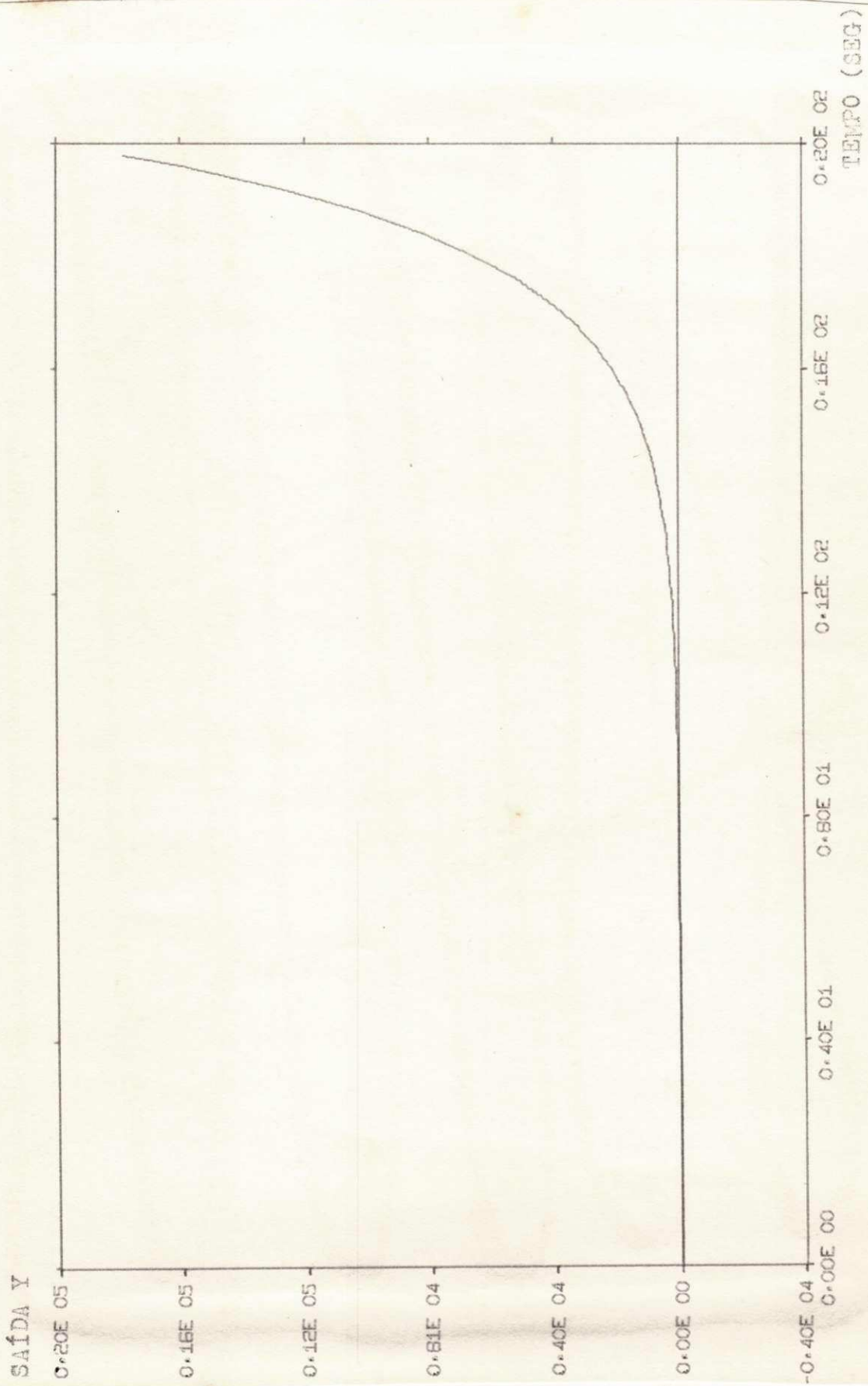


FIGURA 3.4.3 SAÍDA Y EM FUNÇÃO DO TEMPO DO EXEMPLO RESPP

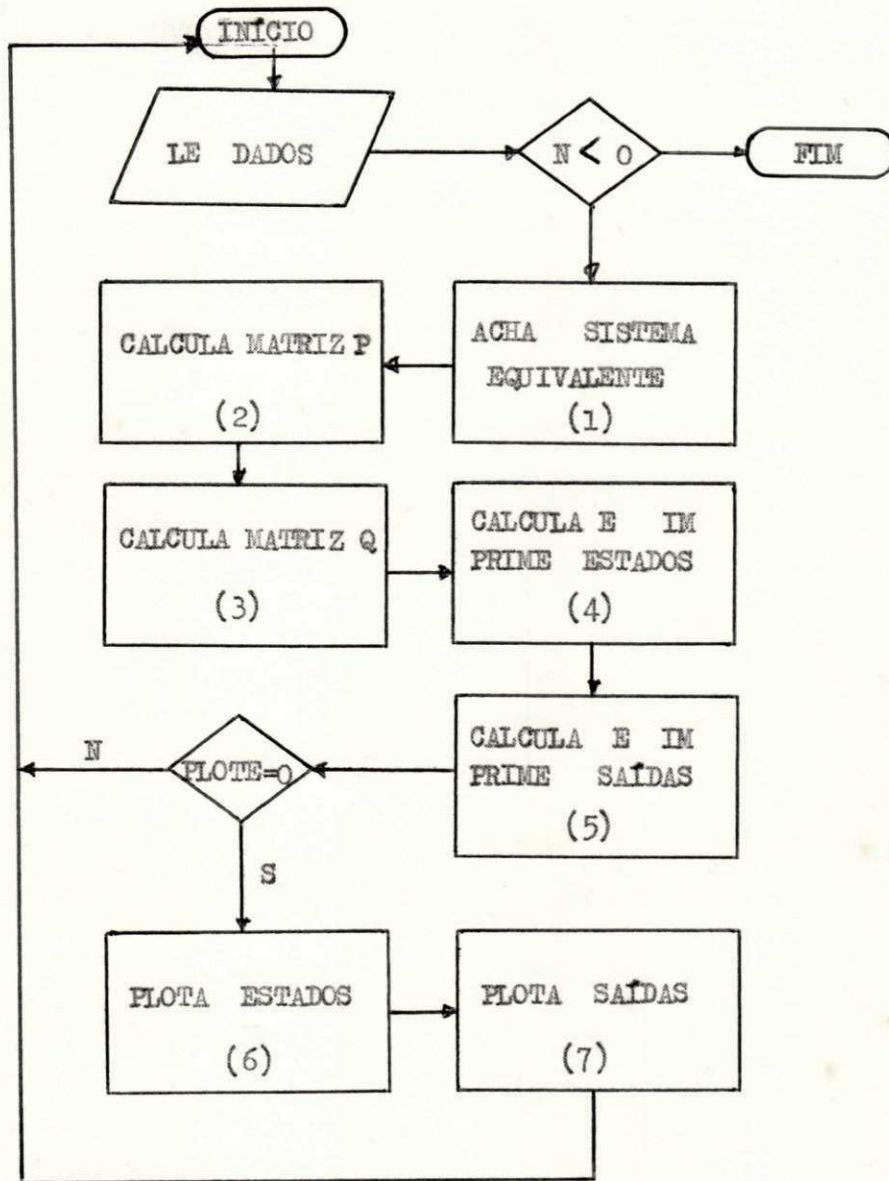


FIGURA 3.4.4 Fluxograma de RESPF

Os subprogramas utilizados são:

1 - LER	5 - SOMA	9 - ENTR	13 - GPLTE
2 - IMPR	6 - SCMUT	10 - FAT	14 - MATSU
3 - KDENT	7 - MATEQ	11 - SCMAT	
4 - PRODU	8 - VALMX	12 - ADIM	

LISTAGEM DE RESPF

```

DIMENSION A(10,10),B(10,10),C(10,10),D(10,10),QS(10,10),
1      P(10,10),T1(10,10),T2(10,10),T5(10,10),T6(10,10),
2      T(10,10),Q(10,10),X(10,1),Y(10,1),U(10,1),T3(10,10),
3      XP(100,10),YP(100,10),XPLF(100),YPLF(100),TPL(100)
      IND=1
200 READ(2,99) N,M,IP,DEL,TO,TMAX,PLOTE
99  FORMAT(9X,I2,8X,I2,8X,I2,8X,4E10.3)
      IF(N) 202,201,201
202 CALL EXIT
201 WRITE(3,101) IND
101 FORMAT(1H1,///,2X,'PROGRAMA RESPOSTA E ESTADO',
1' DO SISTEMA NO TEMPO COM REALIMENTACAO',/2X,
2 'IDENTIFICACAO DO PROBLEMA          RESPF EXEMPLO',I5)
      WRITE(3,100)
100 FORMAT(1H0,60(1H*)/)
      WRITE(3,102) N,DEL,TO,TMAX
102 FORMAT(2X,'ORDEM DO SISTEMA',I4,///,2X,
1      'INTERVALO DE TEMPO PARA A SAIDA',F10.5,
2      //2X,'TEMPO INICIAL',F12.4,
3      //,2X,'TEMPO FINAL',F12.4)
      WRITE(3,100)
      CALL LER(A,N,N)
      WRITE(3,103)
103 FORMAT(2X,'MATRIZ  A',/)
      CALL IMPR(A,N,N)
      CALL LER(B,N,M)
      WRITE(3,104)
104 FORMAT(/2X,'MATRIZ  B',/)
      CALL IMPR(B,N,M)
      CALL LER(C,IP,N)
      WRITE(3,105)
105 FORMAT(/2X,'MATRIZ  C',/)
      CALL IMPR(C,IP,N)
      CALL LER(D,IP,M)

```

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

LISTAGEM DE RESPF

(Cont.)

```

WRITE(3,106)
106 FORMAT(/2X,'MATRIZ D',/)
CALL IMPR(D,IP,M)
CALL LER(X,N,1)
WRITE(3,107)
107 FORMAT(/2X,'CONDICOES INICIAIS...X(0)...',/)
CALL IMPR(X,N,1)
CALL LER(T1,M,N)
WRITE(3,109)
109 FORMAT(/2X,'MATRIZ H',/)
CALL IMPR(T1,M,N)
READ(2,110) GANHO
110 FORMAT(E10.3)
WRITE(3,113) GANHO
113 FORMAT(/2X,'GANHO LINEAR DO SISTEMA',//,2X,E12.5)
CALL PRODU(B,T1,T2,N,M,N)
CALL SCMAT(GANHO,T2,N,N)
CALL MATSU(A,T2,T3,N,N)
CALL MATEQ(T3,A,N,N)
WRITE(3,114)
114 FORMAT(/2X,'MATRIZ PH',/)
CALL IMPR(A,N,N)
CALL SCMAT(GANHO,B,N,M)
WRITE(3,115)
115 FORMAT(/2X,'MATRIZ BH',/)
CALL IMPR(B,N,M)
CALL PRODU(D,T1,T5,IP,M,N)
CALL SCMAT(GANHO,T5,IP,N)
CALL MATSU(C,T5,T6,IP,N)
CALL MATEQ(T6,C,IP,N)
WRITE(3,116)
116 FORMAT(/2X,'MATRIZ CH',/)
CALL IMPR(C,IP,N)
CALL SCMAT(GANHO,D,IP,M)
WRITE(3,117)
117 FORMAT(/2X,'MATRIZ DH',/)
CALL IMPR(D,IP,M)
WRITE(3,100)
CALL KIDENT(QS,N)
CALL KIDENT(P,N)
CALL KIDENT(T1,N)
CALL KIDENT(T2,N)

DELT=0.01
CALL SCMUT(DELT,A,T,N,N)
I=1
2 CALL PRODU(T,T1,T5,N,N,N)

```


LISTAGEM DE RESPF

(Cont.)

```

CALL MATEQ(T5,T1,N,N)
TFAT=(1.0/FAT(I))
CALL SCMAT(TFAT,T5,N,N)
CALL ADDM(P,T5,N,N)
CALL PRODU(T,T2,T6,N,N,N)
CALL MATEQ(T6,T2,N,N)
TFAT=(1.0/FAT(I+1))
CALL SCMAT(TFAT,T6,N,N)
CALL ADDM(QS,T6,N,N)
CALL VALMX(T5,T5MAX,N,N)
IF(T5MAX-1.0E-10) 5,5,6
5 CALL VALMX(T6,T6MAX,N,N)
IF(T6MAX-1.0E-10) 7,7,6
6 I=I+1
IF(I-20) 11,11,7
11 GO TO 2
7 CONTINUE
WRITE(3,108) I
108 FORMAT(2X,'NUMERO DE TERMOS NAS SERIES DE P E Q',//,I7)
WRITE(3,100)
CALL PRODU(QS,B,Q,N,N,M)
CALL SCMAT(DELT,Q,N,M)
WRITE(3,111) N,IP
111 FORMAT(2X,'ESTADOS E SAIDAS DO SISTEMA',//,2X,
1 'SEQUENCIA NUM. ORDEM,TEMPO,ESTADOS NA ORDEM X(1)...X(',
2 I2,')',/,2X,'EM SEGUIDA AS SAIDAS NA ORDEM Y(1)...Y(',I2,')'//)
TX=TO
TVL=DEL/DELT +1.0
XC=0.0
9 XL=0.0
14 XL=XL+1.0
IF(XL-TVL) 15,15,9
15 CALL ENTR(U,TX)
CALL PRODU(C,X,T5,IP,N,1)
CALL PRODU(D,U,T6,IP,M,1)
CALL SOMA(T5,T6,Y,IP,1)
IF(XL-1) 12,12,13
12 WRITE(3,112) XC,TX,(X(J,1),J=1,N),(Y(J,1),J=1,IP)
112 FORMAT(1X,F5.0,2X,E10.3,2X,7(E12.5,2X),/,4X,
1 8(E12.5,2X),/,6X,5(E12.5,2X))

```

C

LISTAGEM DE RESPF

(Cont.)

```

C
  IF(PLOTE) 17,18,17
18 IF(XC-99.0) 21,21,17
21 K=IFIX(XC+1.0)
  DO 19 J=1,N
19 XP(K,J)=X(J,1)
  DO 20 J=1,IP
20 YP(K,J)=Y(J,1)
  TPL(K)=TX
17 XC=XC+1.0
C
C
13 TX=TX+DELT
  IF(TX-TMAX)16,16,10
16 CALL ENTR(U,TX)
C
C
  CALL PRODU(Q,U,T5,N,M,1)
  CALL PRODU(P,X,T6,N,N,1)
  CALL SOMA(T5,T6,X,N,1)
  GO TO 14
10 CONTINUE
C
C
  IF(PLOTE) 22,23,22
23 DO 25 J=1,N
  DO 24 KN=1,K
24 XPLF(KN)=XP(KN,J)
  CALL GPLTE(K,TPL,XPLF)
25 PAUSE
  DO 26 J=1,IP
  DO 27 KN=1,K
27 YPLF(KN)=YP(KN,J)
  CALL GPLTE(K,TPL,YPLF)
  IF(J-IP) 26,28,28
26 PAUSE
22 CONTINUE
28 IND=IND+1
  GO TO 200
C
C
  END

```

CAPÍTULO 4

SISTEMAS DE MALHA FECHADA ÓTIMOS

4.1. Introdução

Neste Capítulo, são discutidos alguns métodos numéricos de se resolver a equação de Riccati, o que é necessário para se otimizar o sistema, como foi visto no Capítulo 1.

Para um sistema com parâmetros invariáveis no tempo, chega-se à conclusão de que é necessário resolver a equação de Riccati na forma reduzida, para $T \rightarrow \infty$.

Depois da discussão resumida de alguns métodos de se resolvê-la, é apresentado um programa, chamado RICCI, baseado em um dos métodos analisados (método 4). O programa é, então, utilizado na resolução de um exemplo e o resultado é apresentado.

No item 4.4. é feita uma comparação dos estados e saídas de um sistema com realização qualquer, com estados e saídas do mesmo sistema com realimentação otimizada.

4.2. Alguns Métodos de Solução da Equação de Riccati

A equação de Riccati, na forma reduzida, é:

$$A'P + PA - PBR^{-1}B'P + Q = 0 \quad (4.1)$$

Método 1⁽¹¹⁾

Este método é baseado na fatorização espectral (construção da matriz W abaixo) e consiste nos seguintes passos:

- Construção da matriz W a partir dos coeficientes da equação (4.1) definida por:

$$W = \begin{bmatrix} A & -BR^{-1}B' \\ -Q & -A' \end{bmatrix} \quad (4.2)$$

A matriz W, pode-se mostrar, tem a propriedade de não possuir autovalores imaginários puros e que se λ é um autovalor $-\lambda^*$ também o é e λ^* e $-\lambda^*$

- Construção da Matriz

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \quad (4.3)$$

Que leva W a uma forma quase diagonal de acordo com a fórmula :

$$T^{-1}WT = \begin{bmatrix} -F & 0 \\ 0 & F \end{bmatrix} \quad (4.4)$$

onde:

F - Matriz diagonal dos autovalores $\lambda_i > 0$ de W.

- Cálculo de P dado por:

$$P = T_{21} T_{11}^{-1} \quad (4.5)$$

Método 2⁽¹¹⁾

Este método também é baseado na fatorização espectral e consiste, dos seguintes passos:

- Construção da matriz W definida em (4.2)
- Cálculo dos autovalores de W
- Construção do polinômio $P(S)$ cujos zeros consistem dos autovalores de W no semi-plano esquerdo (parte real negativa)

$$P(S) = S^n + a_n S^{n-1} + \dots + a_1 \quad (4.6)$$

$P(S)$ deve ter grau igual à ordem da matriz A .

- Construção da matriz $2n \times 2n$ definida por:

$$P(W) = W^n + a_n W^{n-1} + \dots + a_1 I$$

- Cálculo de P , definido por:

$$P(W) \begin{bmatrix} I \\ P \end{bmatrix} = 0 \quad (4.7)$$

A equação (4.7) é um conjunto de equações lineares simultâneas para os elementos de P , logo, é fácil de ser resolvida:

Método 3⁽¹¹⁾

Este método é iterativo, baseado na equação (4.1) e consiste dos passos:

- Escolha de uma matriz K_0 tal que $A + B K_0'$ tenha todos os autovalores com a parte real negativa.

Se os autovalores de A têm todos a parte real negativa, $K_0 = 0$ é uma escolha imediatamente aceitável.

- Cálculo de P através das iterações

$$A_i = A + B K_i' \quad (4.8)$$

$$P_i A_i + A_i P_i + K_i R K_i' + Q = 0 \quad (4.9)$$

$$K_{i+1} = -P_i B R^{-1} \quad (4.10)$$

$$P = \lim_{i \rightarrow \infty} P_i$$

A equação (4.9) é facilmente resolvida⁽¹²⁾

Método 4⁽¹¹⁾

Este método, como o 3, também é iterativo e consiste dos seguintes passos:

- Cálculo das relações, a partir dos elementos da equação (4.1) .

$$E = (I - A)^{-1} (I + A) \quad (4.11)$$

$$F = 2 (I - A)^{-2} B \quad (4.12)$$

$$G = R + B (I - A')^{-1} Q (I - A)^{-1} B \quad (4.13)$$

$$T = Q (I - A)^{-1} B \quad (4.14)$$

- Cálculo da matriz ϕ dada pela expressão:

$$\phi = \lim_{i \rightarrow \infty} \phi_i$$

onde:

$$\phi_{i+1} = E' \phi_i E - \left[E' \phi_i F + T \right] \left[G + F' \phi_i F \right]^{-1} \left[E' \phi_i F + T \right] + Q \quad (4.15)$$

e

$$\phi_0 = 0$$

- Cálculo de P dado por:

$$P = 2 (I - A')^{-1} \phi (I - A)^{-1} \quad (4.16)$$

Método 5⁽¹³⁾

Este método também é iterativo e consiste em:

- Cálculo de P da relação abaixo, por iteração:

$$P(t + \Delta) = P(t) + \Delta \left\{ -P(t) A - A' P(t) + P(t) B B^{-1} B' P(t) - Q \right\} \quad (4.17)$$

Condição limite:

$$\lim_{t \rightarrow \infty} P(t) = 0$$

P(t) é então integrado para trás no tempo com Δ negativo.

Método 6⁽¹⁴⁾

Este método também é baseado na fatorização espectral e consiste nos passos:

- Montagem da matriz W (4.2)

- Cálculo de C dado por:

$$C = \left\{ I + \frac{h}{2} W + \frac{h^2}{12} W^2 \right\}^{-1} \left\{ I + \frac{h}{2} W + \frac{h^2}{12} W^2 \right\} \quad (4.18)$$

onde h é o intervalo de integração escolhido tal que $h = \frac{T}{2^k}$ e k é escolhido como um inteiro que minimiza o erro de truncamento.

- Cálculo da matriz

$$P(0) = -\phi_{22}^{-1}(T) \phi_{21}(T) \quad (4.19)$$

onde

$$\exp(Wt) = \begin{bmatrix} \phi_{11}(t) & \phi_{12}(t) \\ \phi_{21}(t) & \phi_{22}(t) \end{bmatrix} \quad (4.20)$$

e

$$\exp(Wh) = C + O(h^5) \quad (4.21)$$

$O(h^5)$ significa um erro de truncamento da ordem de h^5 .

- Cálculo da $P(t)$ dado por:

$$P(t) = \left\{ \phi_{21}(t) + \phi_{22}(t) P(0) \right\} \left\{ \phi_{11}(t) + \phi_{12}(t) P(0) \right\}^{-1} \quad (4.22)$$

$$0 \leq t \leq T$$

Para este método o tempo final T é finito e a condição final é:

$$P(T) = 0$$

Vários métodos foram desenvolvidos ultimamente, para se resolver a equação de Riccati.

Nas referências (16), (17), (18), (19), (20), (21) e (22) são discutidos alguns deles.

4.3. Programa RICTI

O programa para se resolver a equação de Riccati foi feito baseado no método 4.

O programa, chamado RICTI, lê as matrizes A, B, R e Q, faz as transformações definidas pelas equações (4.11), (4.12), (4.13) e (4.14) e, começando com $\phi_0 = 0$, aplica o processo iterativo até que a maior diferença entre um elemento de ϕ_i e o correspondente de ϕ_{i+1} seja menor do que uma tolerância especificada, lida como dado.

Uma vez calculado o valor de ϕ_i , P é calculado através da equação (4.16) e H é achado através da equação (1.29).

Os dados de entrada são especificados na Tabela (4.1).

Exemplo para ilustrar o uso do programa

Dados:

$$A = \begin{bmatrix} 0 & 0 \\ 1 & -\sqrt{2} \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 \end{bmatrix}$$

Tolerância para $\phi = 1,0 \times 10^{-5}$

TABELA 4.1.

ENTRADA DE DADOS PARA RICTI

Cartão Número	Coluna Número	Formato	Valor Perfurado	Nome da Variável, Descrição
1	1 - 2 10 - 11	I2 I2	1,2,.....,10 1,2,.....,10	N - ordem da matriz A M - nº de colunas de B
2	1 - 10	E10.3	Vr. de TOL	TOL-tolerânc. para $ \phi_i - \phi_{i+1} $
3 até N + 2	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de a_{11} Valor de a_{12} :	a_{11} a_{12} :
N + 3 até 2N + 2	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de b_{11} Valor de b_{12} :	b_{11} b_{12} :
2N + 3 até 3N + 2	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de Q_{11} Valor de Q_{12} :	Q_{11} Q_{12} :
3N + 3 até 3N + M + 2	1 - 10 11 - 20 etc	E10.3 E10.3 :	Valor de r_{11} Valor de r_{12} :	r_{11} r_{12} :
3N + M + 3	1 - 2	I2	-1,-2,....,-9	Número inteiro negativo para o programa; para outro sistema igual ao primeiro, repetindo tudo.

Com estas especificações os dados de entrada são especificados na Tabela 4.2.

Os resultados estão na Tabela 4.3.

TABELA 4.2.

ENTRADA DE DADOS PARA O EXEMPLO DE RICCI

CARTÃO NÚMERO	COLUNA NÚMERO																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2									1																				
2	1.	000	E	-0	5																									
3			0.	0									0.	0																
4			1.	0									-1.	4	1	4														
5			1.	0																										
6			0.	0																										
7			0.	0										0.	0															
8			0.	0										1.	0															
9			1.	0																										
10	-1																													

TABELA 4.3.

RESULTADOS DO EXEMPLO DE RICCI

PROGRAMA RESOLUCAO DA EQUACAO DE RICCATI
 IDENTIFICACAO DO PROBLEMA RICCI EXEMPLO 2

MATRIZ A

0.000E 00 0.000E 00
 0.100E 01 -0.141E 01

MATRIZ B

0.100E 01
 0.000E 00

MATRIZ Q

0.000E 00 0.000E 00
 0.000E 00 0.100E 01

TABELA 4.3.
(Cont.)

```

MATRIZ R
0.100E 01
TOLERANCIA PARA A MATRIZ PHI
0.100E-05
*****
NUMERO DE INTERACOES PARA A MATRIZ PHI
3
MATRIZ P
0.585E 00 0.171E 00
0.171E 00 0.343E 00
MATRIZ H
0.585E 00 0.171E 00

```

O fluxograma de NICTH é dado na Figura 4.1.

As operações nos quadrinhos numerados são:

- (1) - Cálculo das matrizes E, F, G e T através das relações (4.11), (4.12), (4.13) e (4.14) por meio das subrotinas KIDENT, MATSU, EQLSI, SOMA, PRODU, MATEQ, SCMAT, TRNPZ
- (2) - Cálculo da matriz ϕ de acordo com a relação (4.15) através das subrotinas TRNPZ, PRODU, SOMA, EQLSI, MATSU, MATEQ
- (3) - Cálculo da matriz P de acordo com a relação (4.16) através das subrotinas MATEQ, PRODU, SCMAT

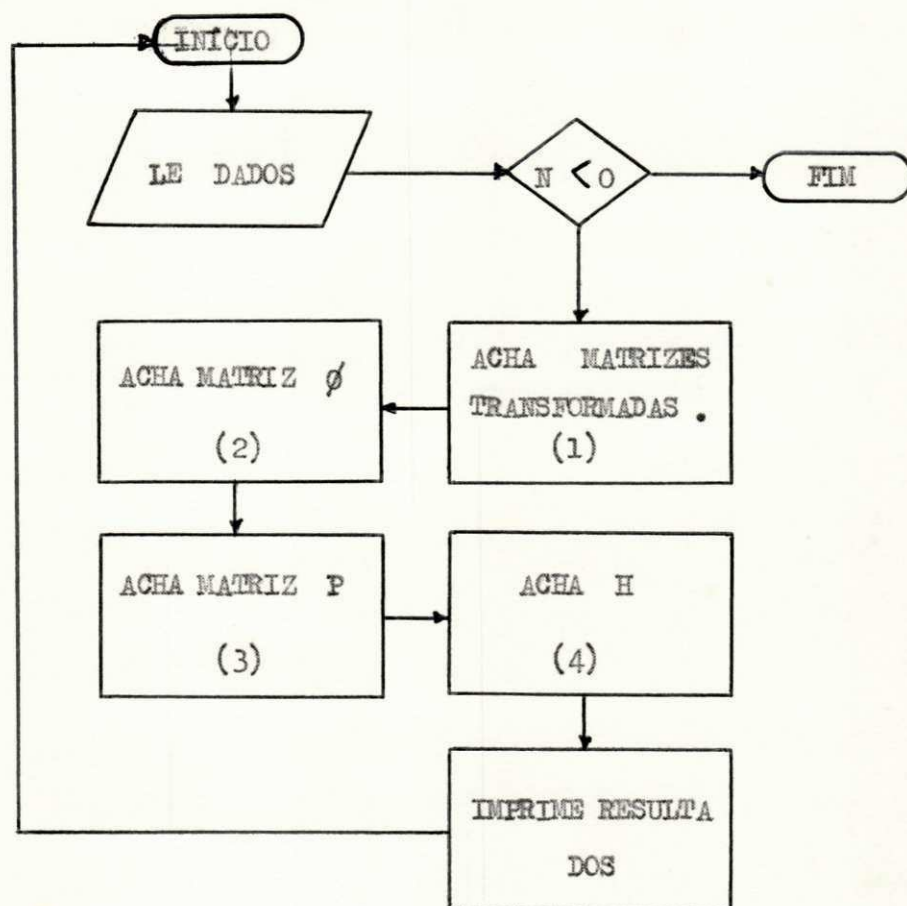


FIGURA 4.1. Fluxograma de RICCI

(4) - Cálculo da matriz H de acordo com a relação (1.29) através das subrotinas EQLSI, PRODU.

Os subprogramas usados são:

1 - LER	6 - SOMA
2 - IMPR	7 - PRODU
3 - KDENT	8 - MATEQ
4 - MATSU	9 - SCMAT
5 - EQLSI	10 - TRNPZ

LISTAGEM DE RICCI

```

C
  DIMENSION A(10,10),B(10,10),E(10,10),F(10,10),G(10,10),
1          T(10,10),P(10,10),R(10,10),Q(10,10),PHI(10,10),
2          H(10,10),BT(10,10),ET(10,10),FT(10,10),T1(10,10),
3          T2(10,10),T3(10,10),T4(10,10),T1INV(10,10),
4          T2INV(10,10),T4INV(10,10),XDOT(11),X(11)
  IND=1
  1 READ(2,101) N,M
101 FORMAT(I2,7X,I2)
  IF(N) 2,3,3
  2 CALL EXIT
  3 WRITE(3,102) IND
102 FORMAT(1H1//2X,'PROGRAMA RESOLUCAO DA EQUACAO DE RICCATI'
1 /2X,'IDENTIFICACAO DO PROBLEMA  RICCI EXEMPLO',I5)
  WRITE(3,100)
100 FORMAT(1H0,60(1H*))//
  READ(2,103) TOL
103 FORMAT(E10.3)
  CALL LER(A,N,N)
  WRITE(3,104)
104 FORMAT(2X,'MATRIZ  A',/)
  CALL IMPR(A,N,N)
  CALL LER(B,N,M)
  WRITE(3,105)
105 FORMAT(/2X,'MATRIZ  B',/)
  CALL IMPR(B,N,M)
  CALL LER(Q,N,N)
  WRITE(3,106)
106 FORMAT(/2X,'MATRIZ  Q',/)

```

LISTAGEM DE RICTI

(Cont.)

```

CALL IMPR(Q,N,N)
CALL LER(R,M,M)
WRITE(3,107)
107 FORMAT(/2X,'MATRIZ R',/)
CALL IMPR(R,M,M)
WRITE(3,108) TOL
108 FORMAT(/2X,'TOLERANCIA PARA A MATRIZ PHI'//2X,E10.3)
WRITE(3,100)
CALL KDENT(T1,N,N)
CALL MATSU(T1,A,T2,N,N)
CALL EQLSI(T2,XDOT,N,T2INV,X,IER)
IF(IER) 4,5,4
5 WRITE(3,109)
109 FORMAT(/2X,'NAO FOI POSSIVEL ACHAR O INVERSO DE (I-A)')
CALL EXIT
4 CALL SOMA(T1,A,T3,N,N)
CALL PRODU(T2INV,T3,E,N,N,N)
CALL MATEQ(T2INV,T4,N,N)
CALL PRODU(T2INV,T4,T2,N,N,N)
CALL PRODU(T2,B,F,N,N,M)
CALL SCMAT(2.0,F,N,M)
CALL TRNPZ(B,BT,N,M)
CALL TRNPZ(A,T2,N,N)
CALL MATSU(T1,T2,T4,N,N,N)
CALL EQLSI(T4,XDOT,N,T4INV,X,IER)
IF(IER) 6,7,6
7 WRITE(3,110)
110 FORMAT(/2X,'NAO FOI POSSIVEL ACHAR O INVERSO DE (I-AT)')
CALL EXIT
6 CALL PRODU(BT,T4INV,T4,M,N,N)
CALL PRODU(T4,Q,T3,M,N,N)
CALL PRODU(T3,T2INV,T2,M,N,N)
CALL PRODU(T2,B,T3,M,N,M)
CALL SOMA(R,T3,G,M,M)
CALL PRODU(Q,T2INV,T2,N,N,N)
CALL PRODU(T2,B,T,N,N,M)
DO 8 I=1,N
DO 8 J=1,N
8 PHI(I,J)=0.0
DO 13 K=1,200
CALL TRNPZ(E,ET,N,N)
CALL PRODU(ET,PHI,T2,N,N,N)
CALL PRODU(T2,E,T3,N,N,N)
CALL PRODU(T2,F,T4,N,N,M)
CALL SOMA(T4,T,T2,N,M)
CALL TRNPZ(F,FT,N,M)

```

LISTAGEM DE RICTI

(Cont.)

```

CALL PRODÜ(FT,PHI,T1,M,N,N)
CALL PRODÜ(T1,F,T4,M,N,M)
CALL SOMA(G,T4,T1,M,M)
CALL EQLSI(T1,XDOT,M,T1INV,X,IER)
IF(IER) 9,10,9
10 WRITE(3,111)
111 FORMAT(/2X,'NAO FOI POSSIVEL ACHAR O INVERSO DE (G+FT*PHI*F)')
CALL EXIT
9 CALL PRODÜ(T2,T1INV,T1,N,M,M)
CALL TRNPZ(T2,T4,N,M)
CALL PRODÜ(T1,T4,T2,N,M,N)
CALL MATSU(T3,T2,T4,N,N)
CALL SOMA(T4,Q,T1,N,N)
DO 11 I=1,N
DO 11 J=1,N
DIF=ABS(PHI(I,J)-T1(I,J))
IF(DIF-TOL) 11,12,12
12 CALL MATEQ(T1,PHI,N,N)
GO TO 13
11 CONTINUE
GO TO 14
13 CONTINUE
14 WRITE(3,112) K
112 FORMAT(2X,'NUMERO DE INTERACOES PARA A MATRIZ PHI',//,6X,I3)
CALL MATEQ(T1,PHI,N,N)
CALL PRODÜ(T4INV,PHI,T2,N,N,N)
CALL PRODÜ(T2,T2INV,P,N,N,N)
CALL SCMAT(2.0,P,N,N)
WRITE(3,115)
115 FORMAT(/2X,'MATRIZ P',/)
CALL IMPR(P,N,N)
CALL EQLSI(R,XDOT,M,RINV,X,IER)
IF(IER) 15,16,15
16 WRITE(3,116)
116 FORMAT(/2X,'NAO FOI POSSIVEL ACHAR O INVERSO DE R')
CALL EXIT
15 CALL PRODÜ(RINV,BT,T4,M,M,N)
CALL PRODÜ(T4,P,H,M,N,N)
WRITE(3,117)
117 FORMAT(/2X,'MATRIZ H',/)
CALL IMPR(H,M,N)
IND=IND+1
GO TO 1
END

```


4.4. Comparação Entre Sistema de Malha Fechada Com Realimentação
Qualquer e Com Realimentação Ótima

Para um sistema de malha aberta definido abaixo:

$$A = \begin{bmatrix} 0 & 0 \\ 1 & -\sqrt{2} \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 2 \end{bmatrix} \quad D = \begin{bmatrix} 2 \end{bmatrix}$$

Ganho linear $k = 1$

$$X(0) = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

e para um índice de performance definido por:

$$I = \int_0^{\infty} \left[X'(t) Q_p X(t) + U'(t) R_p U(t) \right] dt$$

onde:

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 \end{bmatrix}$$

Calcula-se o valor de I para três casos:

a - matriz H dada por

$$H = \begin{bmatrix} 0,585 & 0,171 \end{bmatrix}$$

matriz de realimentação ótima calculada para o sistema de malha aberta e índice de performance definidos acima (calculada no exemplo dado no parágrafo 4.3)

b - matriz H dada por

$$H = \begin{bmatrix} 0,05 & 0,005 \end{bmatrix}$$

c - matriz H dada por

$$H = \begin{bmatrix} 5 & 7,5 \end{bmatrix}$$

Os valores de I são calculados e para os três casos são:

$$a - I = 14,214$$

$$b - I = 3718,630$$

$$c - I = 31,392$$

Conclusão - O valor do índice de performance para o caso "a" é menor, o que confirma ser a matriz H do caso "a" a matriz ótima.

Os valores do índice de performance foram calculados algebricamente baseados nas equações abaixo.

$$I = \int_0^{\infty} \left[X' \left(Q_p + K' R_p K \right) X \right] dt = -V(0) = -X'(0) P X(0)$$

onde:

$$V = X' P X$$

$$(A + BK)' P + P (A + BK) = Q_p + K' R_p K$$

$$K = -k \cdot H$$

Substituindo-se as matrizes na equação acima:

$$\left\{ \begin{bmatrix} 0 & 0 \\ 1 & -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & k_1 \ k_2 \end{bmatrix} \right\}' \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} + \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \left\{ \begin{bmatrix} 0 & 0 \\ 1 & -\sqrt{2} \end{bmatrix} + \begin{bmatrix} k_1 & k_2 \\ 0 & 0 \end{bmatrix} \right\} =$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

obtem-se:

$$k_1 P_1 + P_2 + k_1 P_1 + P_2 = k_1^2$$

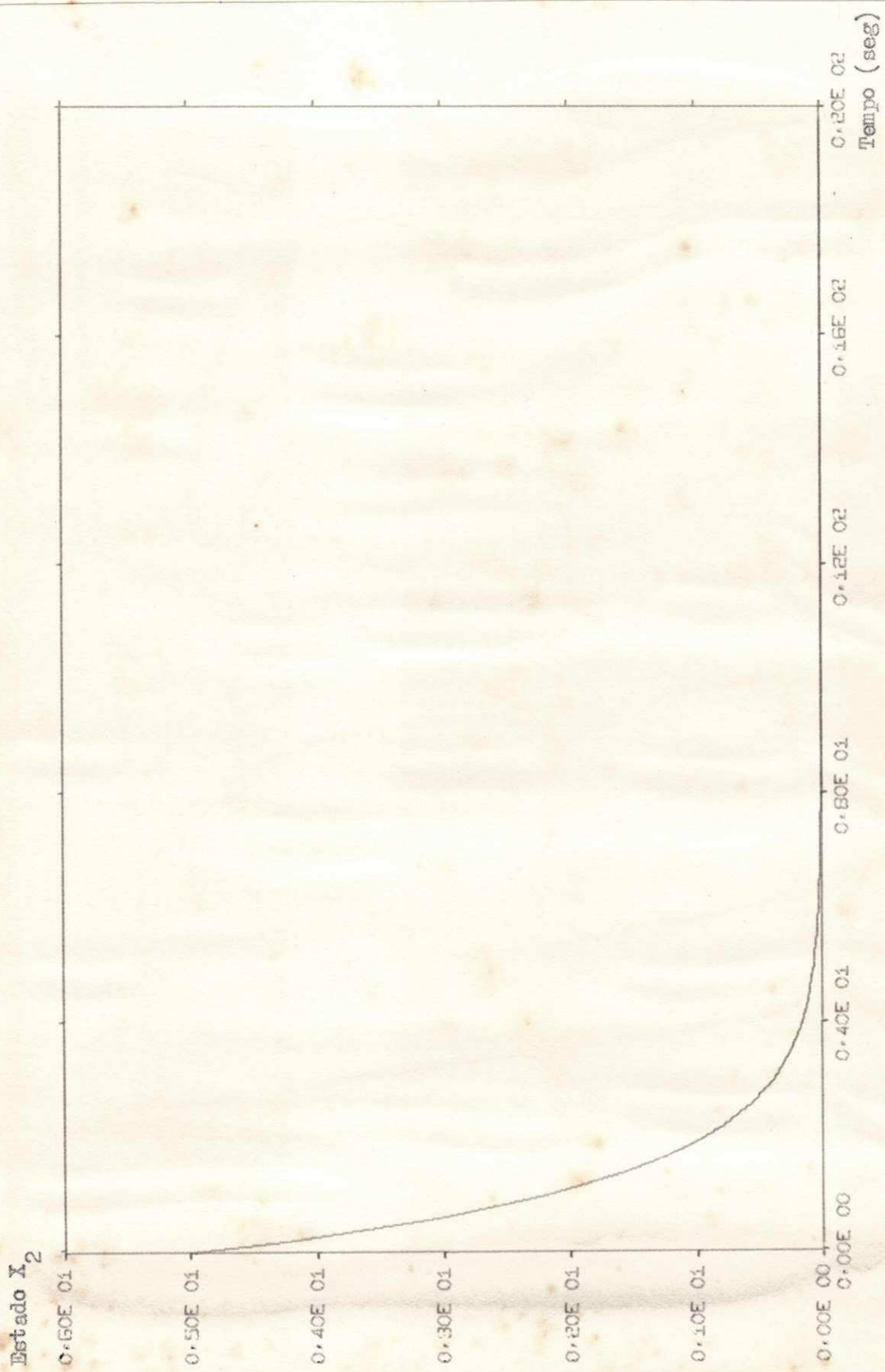
$$k_1 P_2 + P_3 + k_2 P_1 - \sqrt{2} P_2 = k_1 k_2$$

$$2 k_2 P_2 - 2\sqrt{2} P_3 = 1 + k_2^2$$

Os estados e saída para o caso "a" são dados nas Figuras 4.2. , 4.3 e 4.4; para o caso "b" nas Figuras 4.5., 4.6 e 4.7; para o caso "c" nas Figuras 4.8, 4.9 e 4.10.



FIGURA 4.2. Estado X_1 do Sistema Otimizeado

FIGURA 4.3. Estado X_2 do Sistema Otimizeado

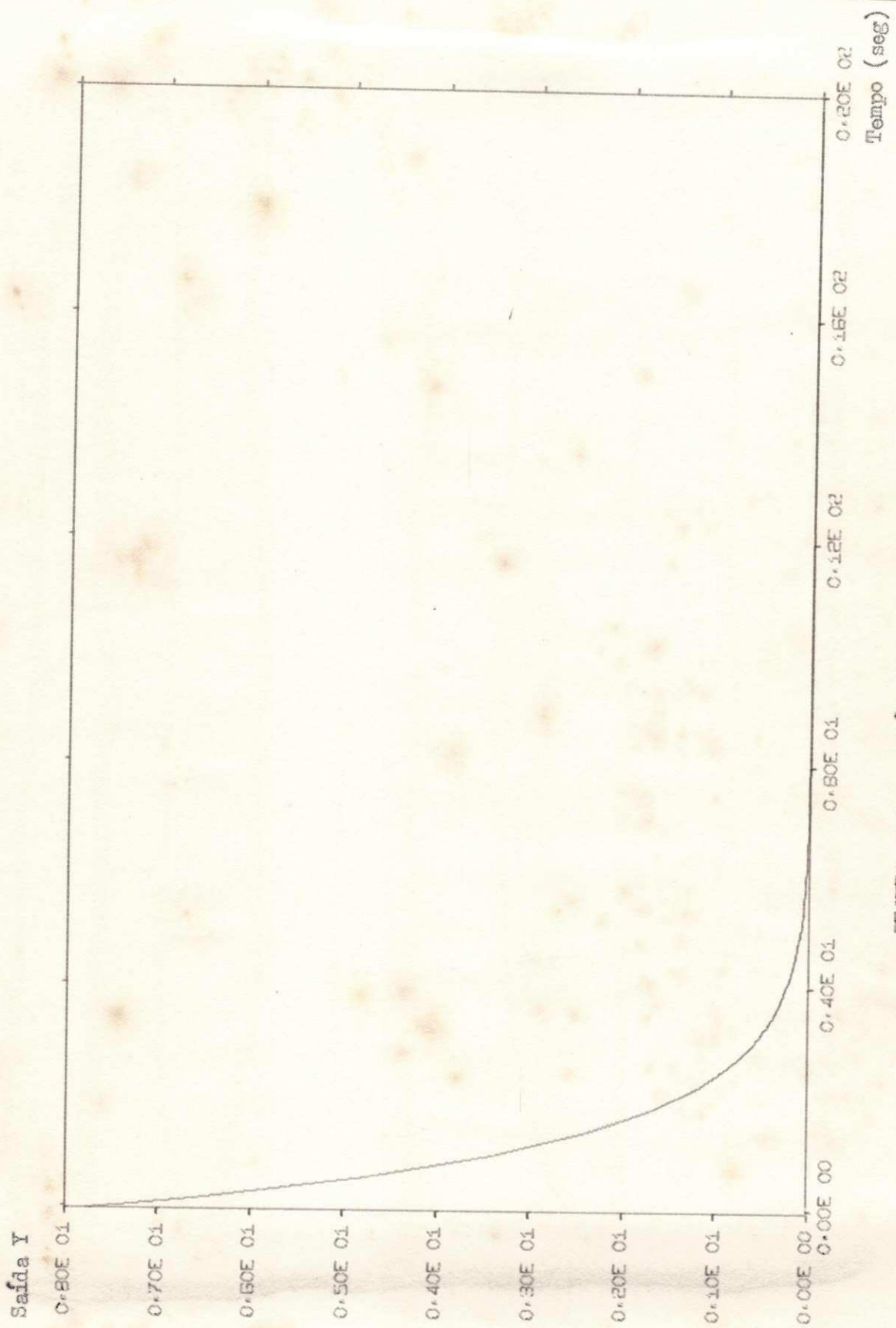


FIGURA 4.4. Saída Y do Sistema Otimizado

PPPh / BIBLIOTECA / PRAI

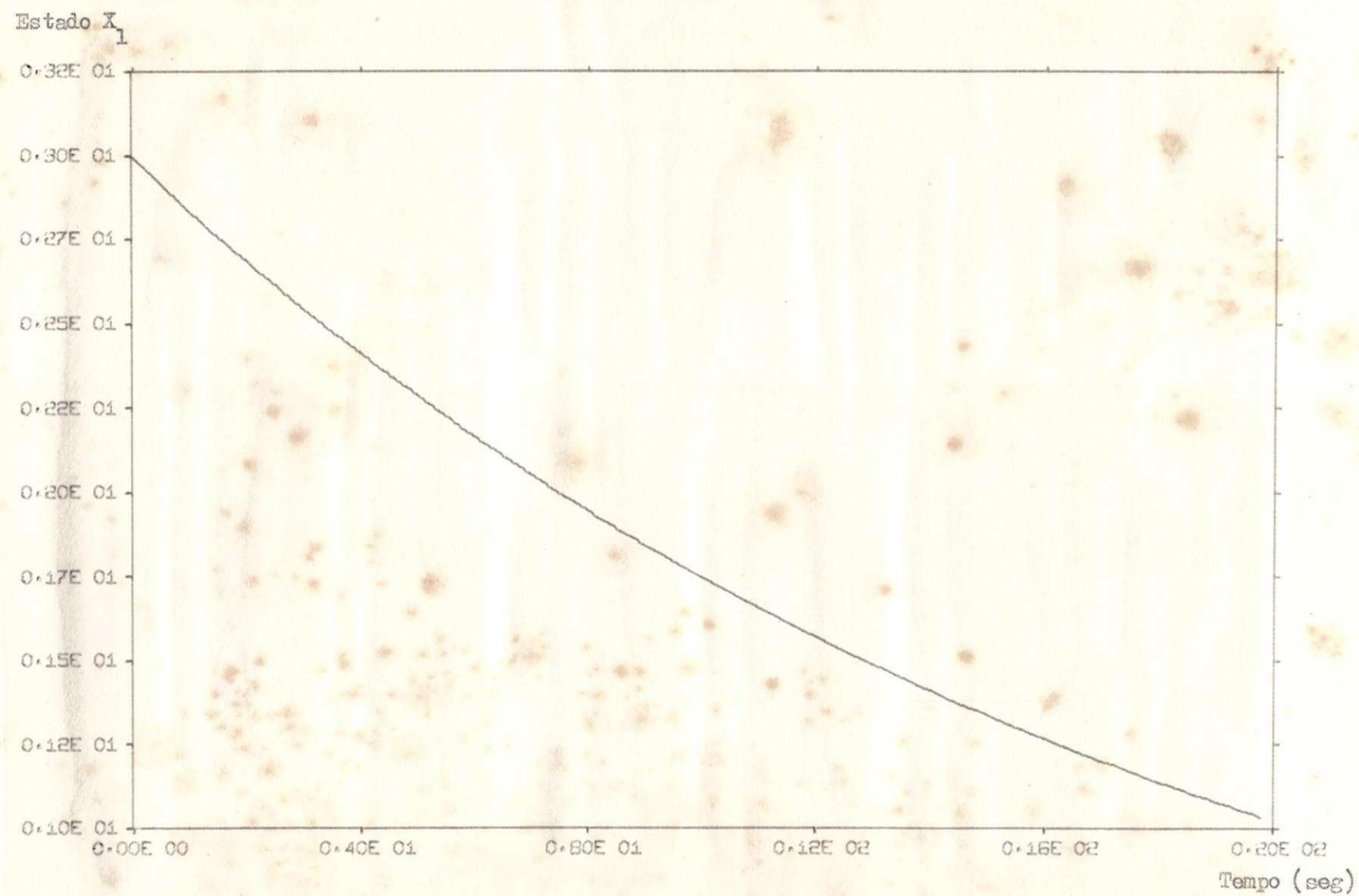
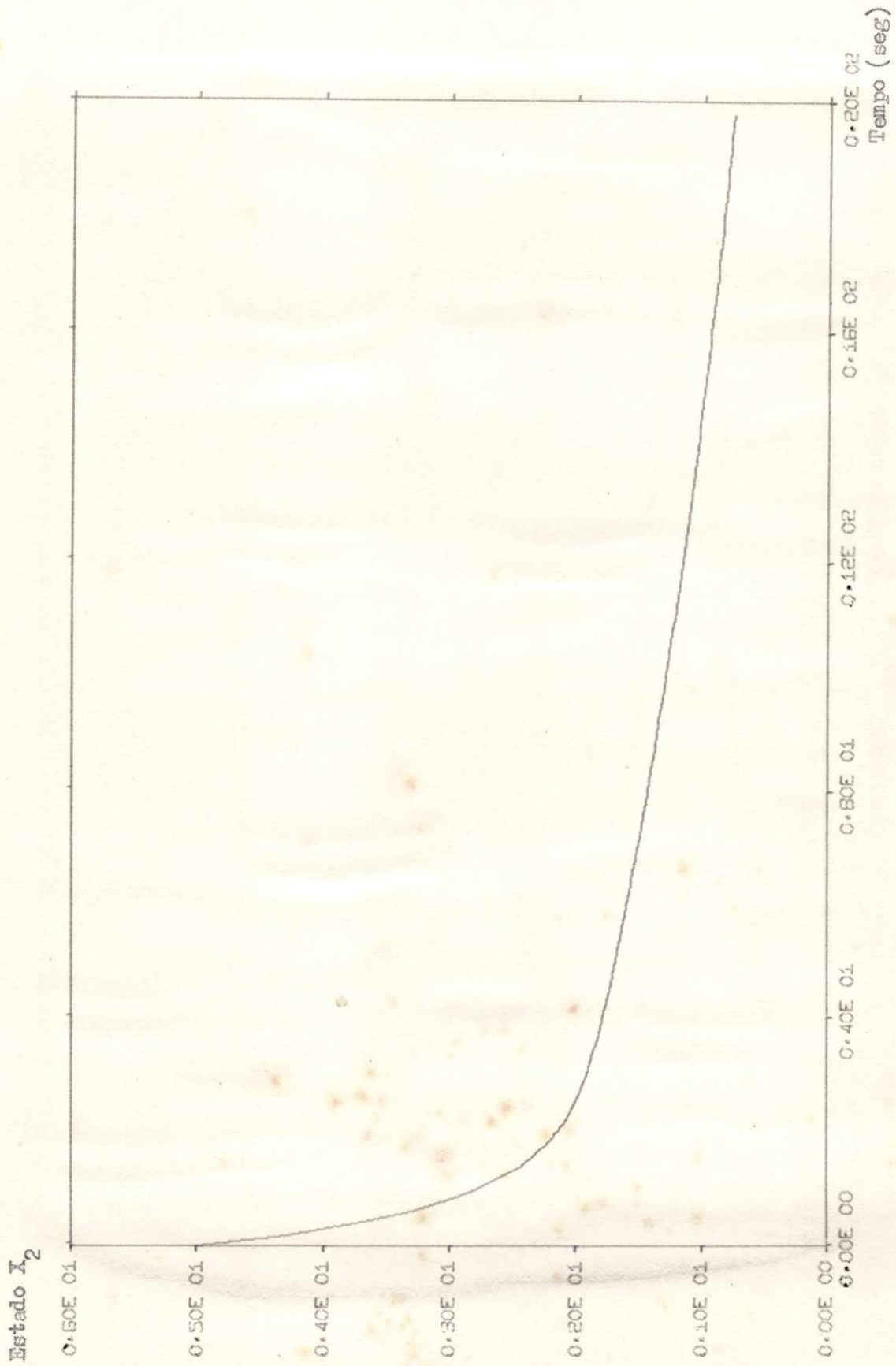


FIGURA 4.5. Estado X_1 do Sistema Não Otimizado

FIGURA 4.6. Estado X_2 do Sistema Não Otimizado

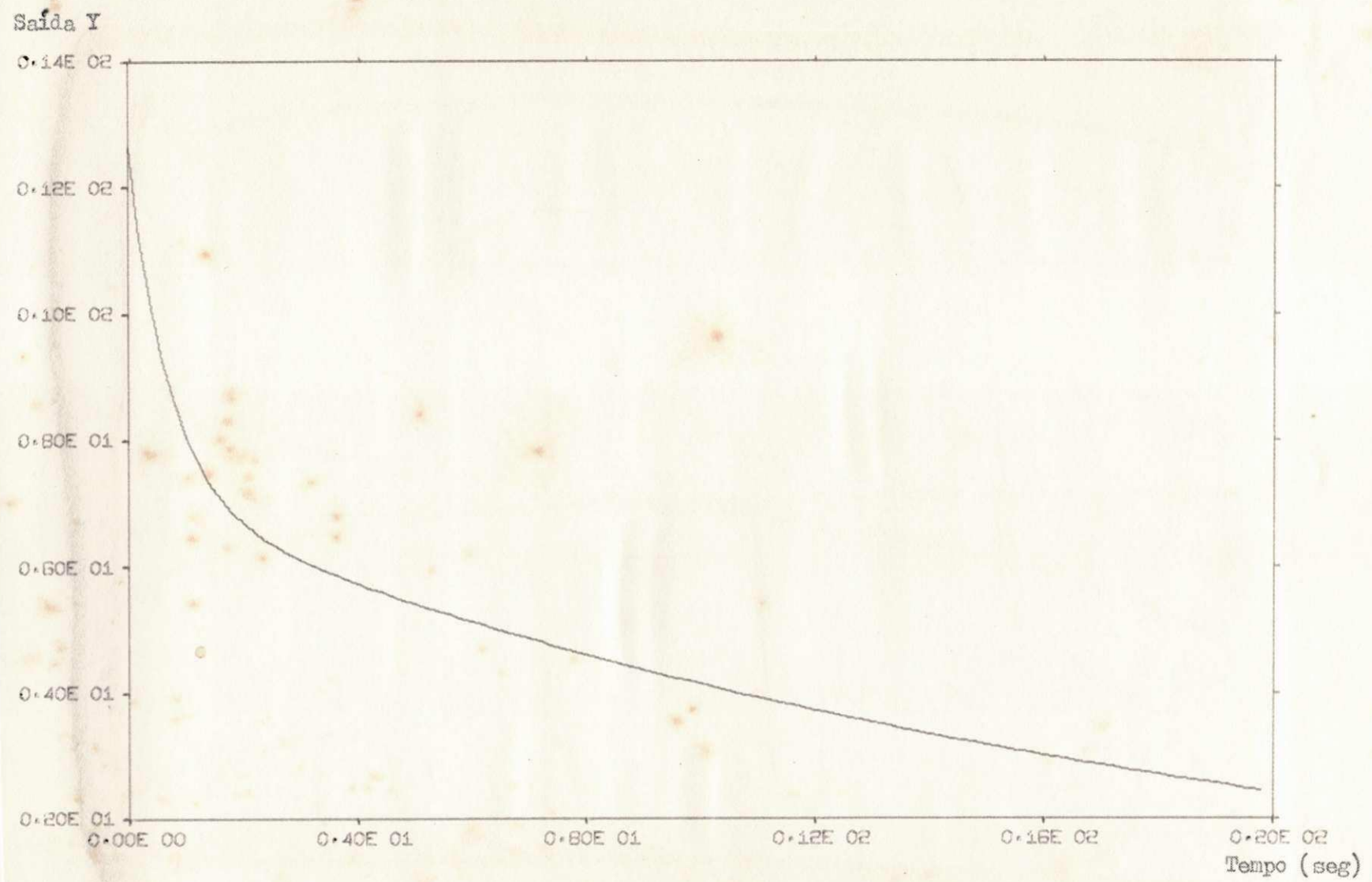


FIGURA 4.7. Saída Y do Sistema Não Otimizado

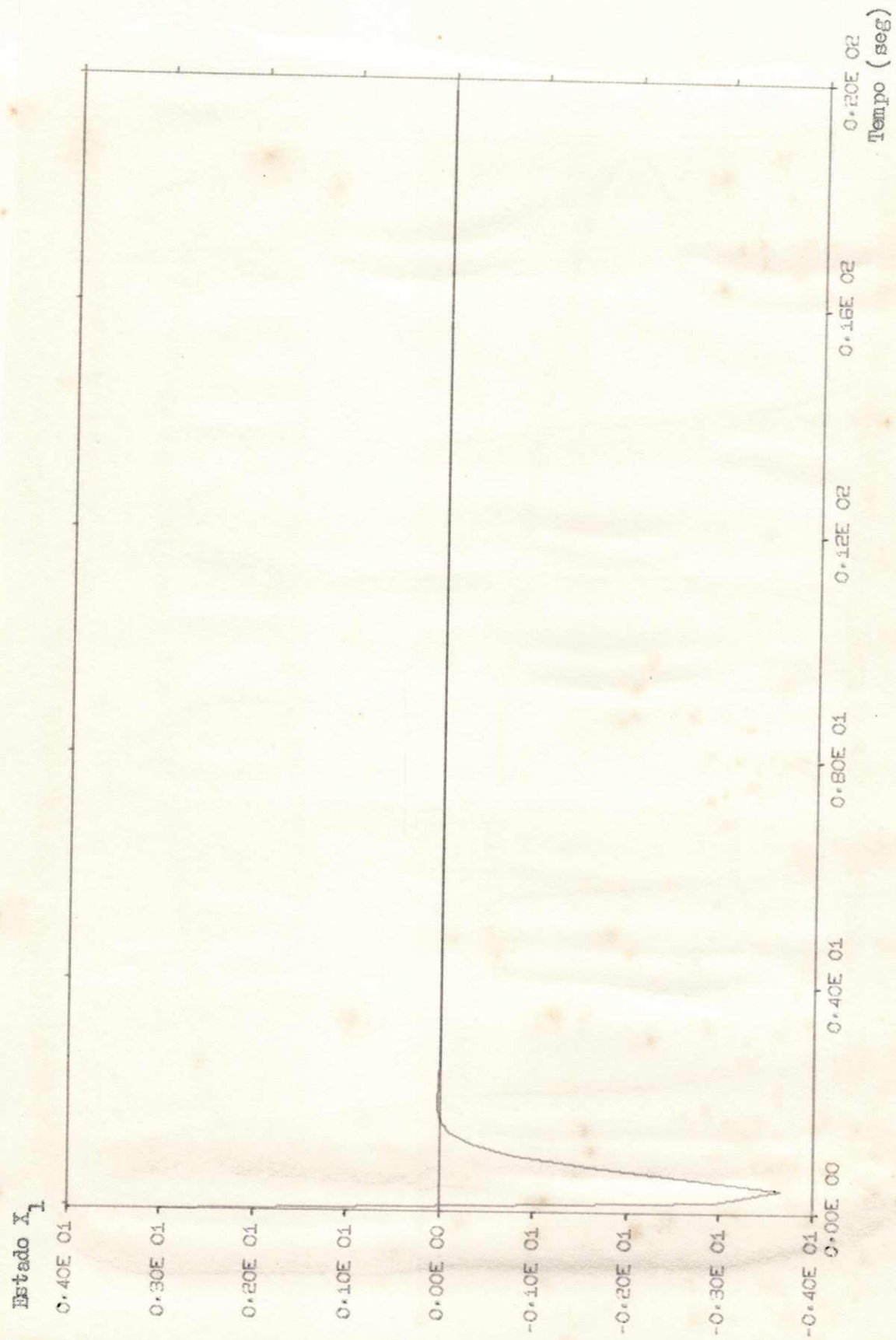
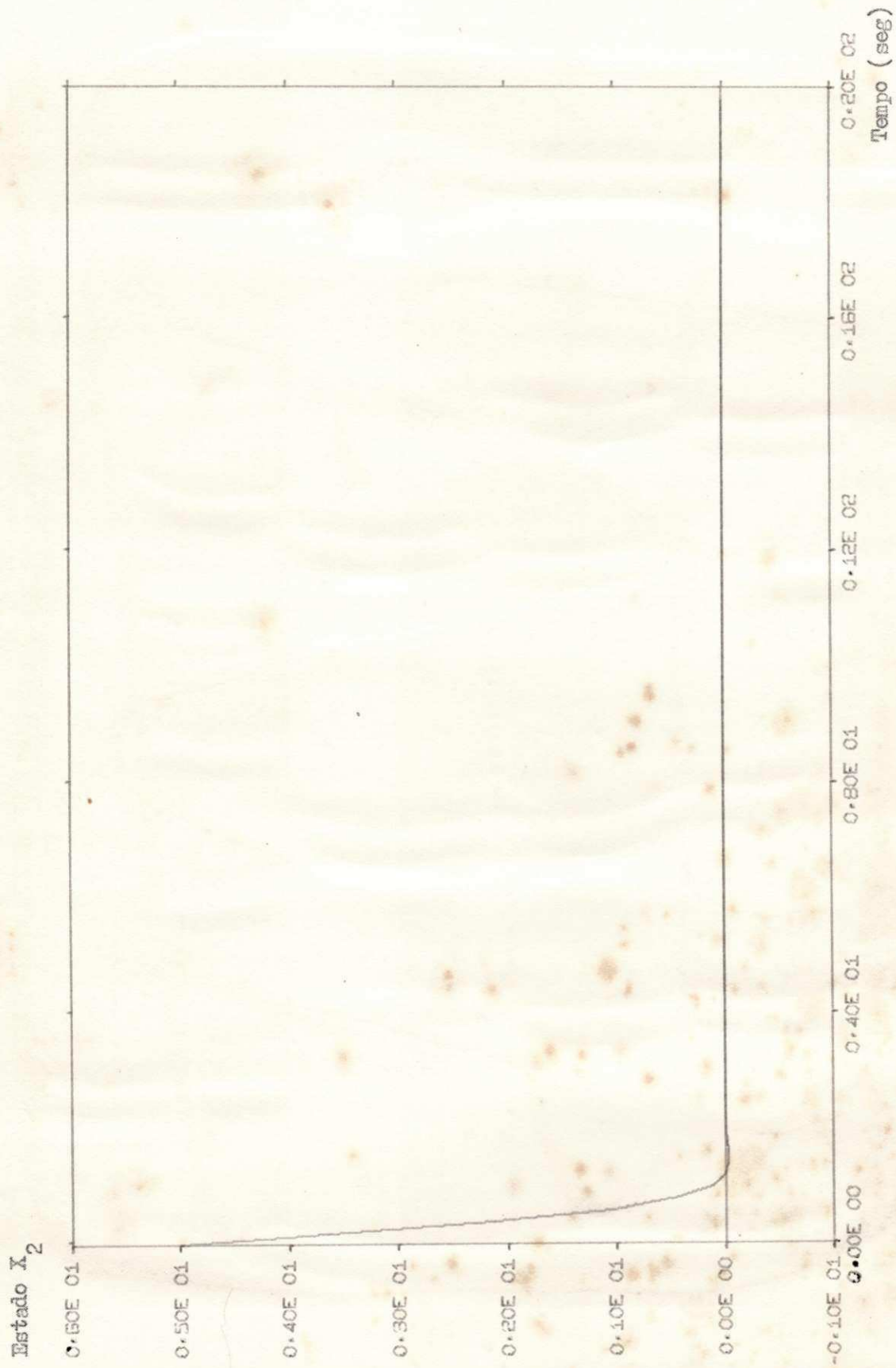


FIGURA 4.8. Estado X_1 do Sistema Não Otimizado

FIGURA 4.9. Estado X_2 do Sistema Não Otimizeado

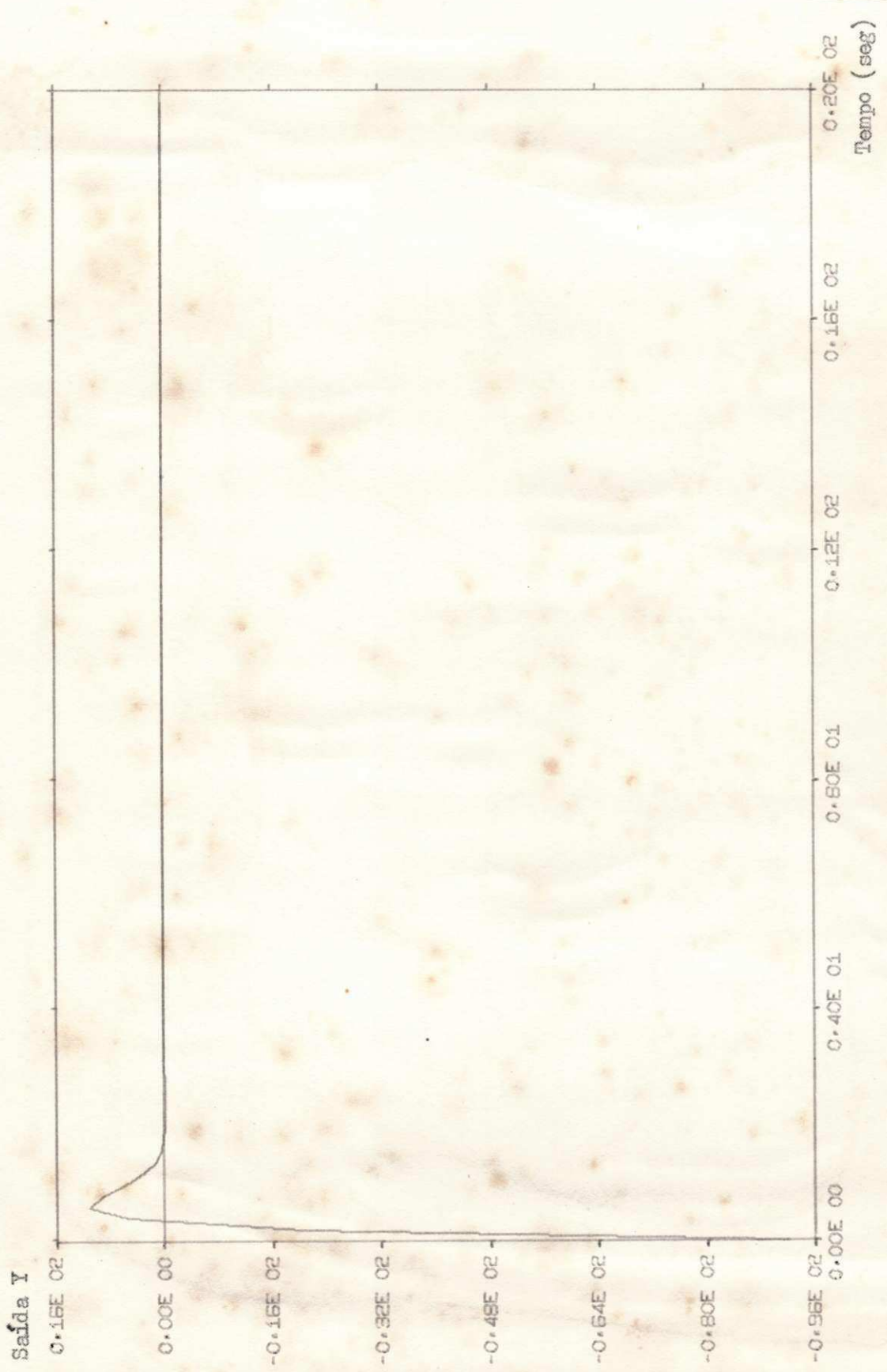


FIGURA 4.10. Saída Y do Sistema Não Otimizado

CONCLUSÃO

Os programas estão nas seções correspondentes a cada um deles.

O programa BASMP foi adaptado para o IBM-1130 baseado na referência (9) e os demais programas foram feitos pelo autor da tese.

Para o programa RESPT, bem como para o RESPTF, a aproximação é da ordem da quatro algarismos significativos. Para maior precisão é suficiente diminuir o valor da variável DELTA definida nos programas, mas o tempo de computação aumenta bastante, mesmo em sistemas pequenos.

Dos vários possíveis métodos de solução foi escolhido um para se resolver cada um dos problemas particulares exigidos.

Foi discutida, com mais ênfase, a solução da equação de Riccati por ser a parte mais importante do trabalho.

Foram examinados alguns métodos de solução da equação de Riccati existentes na bibliografia e se escolheu um deles para a programação.

Um estudo detalhado das vantagens e desvantagens de cada um deles foge ao objetivo deste trabalho.

Pode-se analisar resumidamente cada um deles.

O método 1 pode ser complicado e demorado para os sistemas de ordem um pouco elevada, porque exige o cálculo de autovetores e autovalores.

O método 2 já elimina o cálculo de autovetores mas existe ainda o problema de se achar os autovalores o que pode ser ainda trabalhoso e demorado para sistemas de ordem elevada.

O método 3 para os sistemas estáveis é fácil e simples mas no

caso de sistemas instáveis o cálculo da matriz K_0 pode ser problemático.

O método 4 é fácil de ser programado e requer apenas operações matriciais, o que não leva muito tempo, a não ser as inversões, é de convergência rápida e boa precisão.

O método 5 é fácil de ser programado, mas a precisão depende do intervalo escolhido para a integração e, se for necessário, grande precisão, pode demorar bastante tempo.

O método 6 é relativamente fácil de ser programado e é rápido e, para um tempo T finito é um método bastante recomendável.

Em vista dessa análise foi escolhido o método 4 para a programação.

O resultado foi de boa precisão em comparação com resultados exatos⁽¹⁾.

Analisando-se os resultados do trabalho, chega-se à conclusão de que os objetivos do trabalho foram plenamente alcançados.

Uma continuação lógica deste trabalho poderia ser o desenvolvimento de vários outros programas baseados em outros algoritmos possíveis e comparação dos programas com relação à precisão, tempo de computação, quantidade de memória de computador exigida etc.

Depois disso, poderiam ser desenvolvidos conjuntos de programas para análise e síntese de sistemas lineares, contínuos, com parâmetros variáveis no tempo e, em seguida, conjuntos de programas com objetivos semelhantes para sistemas discretos.

APÊNDICE

Neste Apêndice são descritos os subprogramas utilizados nos programas dos Capítulos anteriores.

É apresentado uma discussão resumida do método utilizado em cada uma das subrotinas e, em seguida, a listagem da mesma.

Algumas subrotinas são utilizadas em vários programas.

A.1 - Subrotina ADDIC

Esta subrotina é utilizada para se montar as matrizes M (1.23) e N (1.24).

Ela é chamada na seguinte forma:

CALL ADDIC (XM, E, N, M, IX)

O significado dos argumentos da subrotina é:

XM - matriz M ou N final

E - parte da matriz N ou M das formas $A^i B$ ou $(A^i)' C'$

N - ordem da matriz A do sistema ≤ 10

M - número de colunas de B ≤ 10

IX - inteiro que dá a posição em que vai ficar a matriz E na matriz XM

LISTAGEM DE ADDIC

```

SUBROUTINE ADDIC(XM,E,N,M,IX)
DIMENSION XM(10,100),E(10,10)
K=(IX-1)*M+1
L=IX*M
DO 1 I=1,N
  IY=0
  DO 1 J=K,L
    IY=IY+1
  1 XM(I,J)=E(I,IY)
RETURN
END

```

A.2 - Subrotina ADDM

Esta subrotina é utilizada para se formar duas matrizes. A
maneira de se chamá-la é:

CALL ADDM (XX, YY, N, M)

O significado dos argumentos é:

- XX - matriz de entrada que se quer somar com YY e, ao mesmo tempo, saída da soma de XX e YY
- YY - matriz que se quer somar com XX
- N - número de linhas de XX e de YY ≤ 10
- M - número de colunas de XX e de YY ≤ 10

Neste subrotina, a matriz XX de entrada é perdida e é substituída pela soma de XX de entrada com YY.

LISTAGEM DE ADDM

```

SUBROUTINE ADDM(XX,YY,N,M)
DIMENSION XX(10,10),YY(10,10)
DO 1 I=1,N
DO 1 K=1,M
1 XX(I,K)=XX(I,K)+YY(I,K)
RETURN
END

```

A.3 - Função DET

Esta função é utilizada para calcular o determinante de uma matriz.

O modo de chamá-la é:

$X = \text{DET}(A, KC)$

Onde os símbolos ou argumentos significam:

A - matriz da qual se deseja o valor do determinante

KC - ordem da matriz, $KC \leq 10$

Se a matriz é singular, o programa colocará $\text{DET} = 0$

LISTAGEM DE DET

```

FUNCTION DET(A,KC)
CARLOS ALBERTO HOLZ
C ESTE SUBPROGRAMA ACHA O DETERMINANTE DE UMA MATRIZ
C USANDO PROCESSO DE DIAGONALIZACAO
DIMENSION A(10,10),B(10,10)
IREV=0
DO 1 I=1,KC
DO 1 J=1,KC
1 B(I,J)=A(I,J)
DO 20 I=1,KC
K=I
9 IF(B(K,I))10,11,10
11 K=K+1
IF(K-KC)9,9,51
10 IF(I-K)12,14,51
12 DO 13 M=1,KC

```

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (031) 321 7222-R 355
58.100 - Campina Grande - Paraíba

LISTAGEM DE DET

(Cont.)

```

TEMP=B(I,M)
B(I,M)=B(K,M)
13 B(K,M)=TEMP
IREV=IREV+1
14 II=I+1
DO 17 M=II,KC
IF(B(M,I))19,17,19
19 TEMP=B(M,I)/B(I,I)
DO 16 N=I,KC
16 B(M,N)=B(M,N)-B(I,N)*TEMP
17 CONTINUE
20 CONTINUE
DET=1.
DO 2 I=1,KC
2 DET=DET*B(I,I)
DET=(-1.)**IREV*DET
RETURN
51 DET=0
RETURN
END

```

A.4 - Subrotina ENTR

Esta subrotina é utilizada para se definir a entrada do sistema.

O modo de chamá-la é:

CALL ENTR (U, T)

onde os argumentos são:

U - matriz entrada do sistema, definida na forma U(1,1),

U(N, 1) N ≤ 10

T - tempo

LISTAGEM DE ENTR

```

SUBROUTINE ENTR(U,T)
DIMENSION U(10,1)
U(1,1)=EXP(-T)
RETURN
END

```

A.5. - Subrotina EQLSI

Esta subrotina é usada para se resolver um conjunto de equações lineares na forma $AX = XDOT$ ou para se inverter uma matriz.

O modo de se chamá-la é:

CALL EQLSI (A, XDOT, KC, AINV, X, IERR)

O significado dos argumentos é:

A - matriz de entrada da equação acima

XDOT - coeficientes vectoriais da equação acima

KC - ordem da matriz A, ≤ 10

AINV - inverso da matriz A

X - solução da equação, ou seja, $X = (A \text{ INV}) XDOT$

IERR = 1 se A é não singular

= se A é singular

LISTAGEM DE EQLSI

```

SUBROUTINE EQLSI(A,XDOT,KC,AINV,X,IERR)
C ESTA SUBROTINA ACHA O INVERSO DA MATRIA A USANDO PROCESSO
C  DIAGONALIZACAO
DIMENSION A(10,10),B(10,10),XDOT(11),X(11),AINV(10,10)
N=1
IERR=1
DO 1 I=1,KC
DO 1 J=1,KC
AINV(I,J)=0
1 B(I,J)=A(I,J)
DO 2 I=1,KC
AINV(I,I)=1
2 X(I)=XDOT(I)
DO 3 I=1,KC
COMP=0
K=I
6 IF(ABS(B(K,I))-ABS(COMP))5,5,4
4 COMP=B(K,I)
N=K
5 K=K+1
IF(K-KC)6,6,7

```

LISTAGEM DE EQLSI

(Cont.)

```
7 IF(B(N,I))8,51,8
8 IF(N-I)51,12,9
9 DO 10 M=1,KC
  TEMP=B(I,M)
  B(I,M)=B(N,M)
  B(N,M)=TEMP
  TEMP=AINV(I,M)
  AINV(I,M)=AINV(N,M)
10 AINV(N,M)=TEMP

  TEMP=X(I)
  X(I)=X(N)
  X(N)=TEMP
12 X(I)=X(I)/B(I,I)
  TEMP=B(I,I)
  DO 13 M=1,KC
    AINV(I,M)=AINV(I,M)/TEMP
13 B(I,M)=B(I,M)/TEMP
  DO 16 J=1,KC
    IF(J-I)14,16,14
14 IF(B(J,I))15,16,15
15 X(J)=X(J)-B(J,I)*X(I)
  TEMP=B(J,I)
  DO 17 N=1,KC
    AINV(J,N)=AINV(J,N)-TEMP*AINV(I,N)
17 B(J,N)=B(J,N)-TEMP*B(I,N)
16 CONTINUE
  3 CONTINUE
  RETURN
51 WRITE(3,52)
52 FORMAT(6X,'A MATRIZ E SINGULAR')
  IERR=0
  RETURN
END
```

A.6 - Função FAT

Esta função é utilizada para se achar o fatorial de um número dado. O modo de se chamá-la é:

X = FAT (M)

onde,

M - número do qual se deseja o fatorial.

LISTAGEM DE FAT

```

FUNCTION FAT(M)
  FAT=1.0
  DO 1 I=1,M
1  FAT=FAT*I
  RETURN
  END

```

A.7 - Subrotina GPLTE

Esta subrotina é utilizada para se plotar um determinado conjunto de pontos. O modo de chamá-la é:

CALL GPLTE (N, X, Y)

onde,

N - número de pontos a ser plotado

X - vetor de ordem N que dá as abcissas dos pontos

Y - vetor de ordem N que dá as ordenadas dos pontos

LISTAGEM DE GPLTE

```

SUBROUTINE GPLTE(N,X,Y)
  DIMENSION X(1),Y(1)
  XMIN = X(1)
  XMAX = X(1)
  YMIN = Y(1)
  YMAX = Y(1)
  DO 17 I=2,N
  IF (X(I)-XMIN) 10,11,11

```

LISTAGEM DE GELTE

(Cont.)

```
10 XMIN = X(I)
11 IF (X(I)-XMAX) 13,13,12
12 XMAX=X(I)
13 IF (Y(I)-YMIN) 14,15,15
14 YMIN = Y(I)
15 IF (Y(I)-YMAX) 17,17,16
16 YMAX = Y(I)
17 CONTINUE
   DX = XMAX-XMIN
   IF (DX) 18,18,19
18 DX = .000001
19 IX = 0
20 IF (DX-4.) 21,22,22
21 DX = DX*2.
   IX = IX-1
   GO TO 20
22 IF (DX-8.) 24,23,23
23 DX = DX/2.
   IX = IX+1
   GO TO 22
24 XMIN = XMIN*2.**(-IX)
   IF (XMIN) 26,25,25
25 XMIN = FLOAT(IFIX(XMIN))
   GO TO 27
26 XMIN = FLOAT(IFIX(XMIN)-1)
27 XMIN = XMIN*2.**IX
   XMAX = XMAX*2.**(-IX)
   IF (XMAX) 29,28,28
28 XMAX = FLOAT(IFIX(XMAX)+1)
   GO TO 30
29 XMAX = FLOAT(IFIX(XMAX))
30 XMAX = XMAX*2.**IX
   DX = XMAX-XMIN
   DY = YMAX-YMIN
   IF (DY) 43,43,31
43 DY = .000001
31 IY = 0
32 IF (DY-4.) 33,34,34
33 DY = DY*2.
   IY = IY-1
   GO TO 32
34 IF (DY-8.) 36,35,35
35 DY = DY/2.
   IY = IY+1
   GO TO 34
36 YMIN = YMIN*2.**(-IY)
   IF (YMIN) 38,37,37
37 YMIN = FLOAT(IFIX(YMIN))
```

LISTAGEM DE GPLTE

(Cont.)

```
GO TO 39
38 YMIN = FLOAT(IFIX(YMIN)-1)
39 YMIN = YMIN*2.**IY
   YMAX = YMAX*2.**(-IY)
   IF (YMAX) 41,40,40
40 YMAX = FLOAT(IFIX(YMAX)+1)
   GO TO 42
41 YMAX = FLOAT(IFIX(YMAX))
42 YMAX = YMAX*2.**IY
   DY = YMAX-YMIN
   CALL SCALF(7.5/DX,5./DY,XMIN,YMIN)
   XX = XMIN
   YY = YMIN
55 CALL FPLOT(2,XX,YY)
   CALL FPLOT(2,XX,YY-DY/180.)
   CALL FCHAR(XX-DX/21.,YY-DY/30.,0.08,0.08,0.0)
   WRITE(7,51) XX
51 FORMAT(E9.2)
   CALL FPLOT(1,XX,YY-DY/180.)
   CALL FPLOT(2,XX,YY)
   IF (XX-XMAX) 44,56,56
44 XX = XX+2.**IX
   GO TO 55
56 CALL FPLOT(2,XX,YY)
   CALL FPLOT(2,XX+DX/180.,YY)
   CALL FPLOT(2,XX,YY)
   IF (YY-YMAX) 45,57,57
45 YY = YY+2.**IY
   GO TO 56
57 CALL FPLOT(2,XX,YY)
   CALL FPLOT(2,XX,YY+DY/180.)
   CALL FPLOT(2,XX,YY)
   IF (XX-XMIN) 58,58,46
46 XX = XX-2.**IX
   GO TO 57
58 CALL FPLOT(2,XX,YY)
   CALL FPLOT(2,XX-DX/180.,YY)
   CALL FCHAR(XX-DX/9.2,YY-DY/150.,0.08,0.08,0.0)
   WRITE(7,51) YY
   CALL FPLOT(1,XX-DX/180.,YY)
   CALL FPLOT(2,XX,YY)
   IF (YY-YMIN) 48,48,47
47 YY = YY-2.**IY
   GO TO 58
48 CALL FPLOT(1,X(1),Y(1))
   DO 49 I=2,N
```

LISTAGEM DE GFLTE

(Cont.)

```

49 CALL FPLOTT(2,X(I),Y(I))
   IF (XMIN*XMAX) 52,53,53
52 CALL FPLOTT(1,0.0,YMIN)
   CALL FPLOTT(2,0.0,YMAX)
53 IF (YMIN*YMAX) 54,50,50
54 CALL FPLOTT(1,XMIN,0.0)
   CALL FPLOTT(2,XMAX,0.0)
50 CALL FPLOTT(1,XMIN+11./9.*DX,YMIN)
   RETURN
   END

```

OBS: Subrotina feita pelo Senhor YAN WELSH

A.8 - Subrotina IMPR

Esta subrotina é utilizada para se imprimir uma matriz qualquer.

O modo de se chamá-la é:

CALL IMPR (A, K, L)

onde os argumentos são:

- A - matriz a ser impressa
- K - número de linhas de A
- L - número de colunas de A

LISTAGEM DE IMPR

```

SUBROUTINE IMPR(A,K,L)
DIMENSION A(10,10)
DO 1 I=1,K
1 WRITE(3,2) (A(I,J),J=1,L)
2 FORMAT(2X,10(E10.3,1X))
RETURN
END

```


A.9 - Subrotina KDENT

Esta subrotina é utilizada para se gerar uma matriz identidade.

O modo de se chamá-la é:

CALL KDENT (A, N)

onde

A - matriz identidade

N - ordem da matriz A

LISTAGEM DE KDENT

```

SUBROUTINE KDENT(A,N)
DIMENSION A(10,10)
DO 1 I=1,N
DO 1 J=1,N
1 A(I,J)=0.0
DO 2 K=1,N
2 A(K,K)=1.0
RETURN
END

```

A.10 - Subrotina LER

A subrotina LER é utilizada para se ler uma matriz cujos elementos estejam perfurados em cartões.

O modo de chamá-la é:

CALL LER (A, K, L)

onde

A - matriz a ser lida

K - número de linhas de A

L - número de colunas de A

LISTAGEM DE LER

```

SUBROUTINE LER(A,K,L)
DIMENSION A(10,10)
DO 1 I=1,K
1 READ(2,3) (A(I,J),J=1,L)
3 FORMAT(10E10.3)
RETURN
END

```

A.11 - Subrotina MATEQ

A subrotina MATEQ é utilizada para se substituir uma matriz por outra. O modo de chamá-la é:

```
CALL MATEQ (A, B, M, N)
```

O significado dos argumentos é:

- A - matriz que substituirá B
- B - matriz que será apagada entrando A em seu lugar
- M - número de linhas de A
- N - número de colunas de A

LISTAGEM DE MATEQ

```

SUBROUTINE MATEQ(A,B,M,N)
DIMENSION A(10,10),B(10,10)
DO 11 I=1,M
DO 11 J=1,N
11 B(I,J)=A(I,J)
RETURN
END

```

A.12 - Subrotina MATSU

Esta subrotina é utilizada para se diminuir uma matriz de outra dada. O modo de chamá-la é:

CALL MATSU (A, B, C, L, M)

onde:

A - B - matrizes nas quais é feita a operação A - B

C - matriz onde retornará o resultado da operação

L - número de linhas de A, B e C

M - número de colunas de A, B e C

LISTAGEM DE MATSU

```

SUBROUTINE MATSU(A,B,C,L,M)
DIMENSION A(10,10),B(10,10),C(10,10)
DO 1 I=1,L
DO 1 J=1,M
1 C(I,J)=A(I,J)-B(I,J)
RETURN
END

```

A.13 - Subrotina PCMRS

Esta subrotina é utilizada para se determinar o polinômio característico $\det(SI - A)$ e a matriz resolvente $(SI - A)^{-1}$ da matriz A.

O modo de chamá-la é:

CALL PCMRS (A, N, C, NRM)

O significado dos argumentos é:

A - matriz A do sistema

N - ordem da matriz A

C - vetor coeficiente do polinômio característico

NRM = 0 os coeficientes da matriz resolvente são impressos

≠ 0 os coeficientes da matriz resolvente não são impressos.

LISTAGEM DE PCMRS

```

SUBROUTINE PCMRS(A,N,C,NRM)
  DIMENSION A(10,10),C(11),ATEMP(10,10),PROD(10,10)
  COMMON ZED(10,10,10)
  DATA ATEMP/100*0.0/
1000 FORMAT(1H0,5X,'OS COEFICIENTES DA MATRIZ DO NUMERADOR DA MATRIZ RE
  1SOLVENTE')
1001 FORMAT(1H0,5X,'OS COEFICIENTES DA MATRIZ DE S**',11,/)
1002 FORMAT(6X,10(E10.3,2X))
1003 FORMAT(1H0,60(1H*))
  DO 65 I=1,N
65  ATEMP(I,I)=1.0
  DO 80 I=1,N
  DO 80 J=1,N
80  ZED(N,I,J)=ATEMP(I,J)
  IF(NRM)71,72,71
72  WRITE(3,1003)
  WRITE(3,1000)
  M=N-1
  WRITE(3,1001) M
  DO 35 I=1,N
35  WRITE(3,1002) (ATEMP(I,J),J=1,N)
71  DO 40 I=1,N
  DO 40 J=1,N
40  ATEMP(I,J)=A(I,J)
  C(1)=1.0
  DO 10 I=1,N
  NNN=N-I
  IF(I-1)73,55,73
73  IF(NRM)60,74,60
74  WRITE(3,1001) NNN
  DO 45 J=1,N
45  WRITE(3,1002) (ATEMP(J,K),K=1,N)
60  NP=NNN+1
  DO 90 II=1,N
  DO 90 J=1,N
90  ZED(NP,II,J)=ATEMP(II,J)
  DO 15 J=1,N
  DO 15 K=1,N
  PROD(J,K)=0.0
  DO 15 L=1,N
15  PROD(J,K)=PROD(J,K)+(A(J,L)*ATEMP(L,K))
  DO 13 J=1,N
  DO 13 K=1,N
13  ATEMP(J,K)=PROD(J,K)
55  TR=0.0
  DO 5 J=1,N
5  TR=TR+ATEMP(J,J)
  C(I+1)=(-1.0/I)*TR
  DO 10 J=1,N
10  ATEMP(J,J)=ATEMP(J,J)+C(I+1)

```

LISTAGEM DE RANK

(Cont.)

```

4 CONTINUE
  IF(A(K,K)) 51,7,51
51 IF(K-N) 52,8,52
52 IF(K-M) 5,8,5
5 CONTINUE
  DO 6 J=KK,M
  A(K,J)=A(K,J)/A(K,K)
  DO 6 I=KK,N
  W=A(I,K)*A(K,J)
  A(I,J)=A(I,J)-W
  IF(ABS(A(I,J))-0.0001*ABS(W)) 61,6,6
61 A(I,J)=0.0
6 CONTINUE
  K=KK
  GO TO 1
7 K=K-1
8 RETURN
END

```

A.16 - Subrotina RAPOL

Esta subrotina é usada para se achar as raízes de um polinômio com coeficientes reais. A maneira de se chamá-la é:

CALL RAPOL (N, A, U, V, IR)

Os argumentos são:

N - grau do polinômio $N \leq 19$

A - vetor coeficiente do polinômio

U - vetor coeficiente da parte real das raízes

V - vetor coeficiente da parte imaginária das raízes

IR = +1 se o polinômio é escrito na forma:

$$A(1) + A(2)S + \dots + A(N+1)S^N$$

-1 se o polinômio é escrito na forma:

$$A(1)S^N + A(2)S^{N-1} + \dots + A(N+1)$$

LISTAGEM DE RAPOL

```

SUBROUTINE RAPOL(N,A,U,V,IR)
C ESTA SUBROUTINA USA UM METODO DE BAIRSTOW MODIFICADO PARA
C ACHAR AS RAIZES DE UM POLINOMIO
DIMENSION A(20),U(20),V(20),H(21),B(21),C(21)
IREV=IR
NC=N+1
DO 1 I=1,NC
1 H(I)=A(I)
P=0.
Q=0.
R=0.
3 IF(H(I))4,2,4
2 NC=NC-1
V(NC)=0.
U(NC)=0.
DO 1002 I=1,NC
1002 H(I)=H(I+1)
GO TO 3
4 IF(NC-1)5,100,5
5 IF(NC-2)7,6,7
6 R=-H(1)/H(2)
GO TO 50
7 IF(NC-3)9,8,9
8 P=H(2)/H(3)
Q=H(1)/H(3)
GO TO 70
9 XX=ABS(H(NC-1)/H(NC))-ABS(H(2)/H(1))
IF(XX) 10,19,19
10 IREV=-IREV
M=NC/2
DO 11 I=1,M
NL=NC+1-I
F=H(NL)
H(NL)=H(I)
11 H(I)=F
IF(Q)13,12,13
12 P=0.
GO TO 15
13 P=P/Q
Q=1./Q
15 IF(R)16,19,16
16 R=1./R
19 E=5.E-10
B(NC)=H(NC)
C(NC)=H(NC)

```

UNIVERSIDADE FEDERAL DA PARAÍBA
 Pró-Reitoria Para Assuntos do Interior
 Coordenação Setorial de Pós-Graduação
 Rua Aprígio Veloso, 882 - Tel. (083) 321 7222-R 375
 58 100 - Campina Grande - Paraíba

LISTAGEM DE RAPOL

(Cont.)

```

B(NC+1)=0.
C(NC+1)=0.
NP=NC-1
20 DO 49 J=1,1000
DO 21 II=1,NP
I=NC-II
B(I)=H(I)+R*B(I+1)
21 C(I)=B(I)+R*C(I+1)
IF (ABS(B(1)/H(1))-E)50,50,24
24 IF (C(2))23,22,23
22 R=R+1
GO TO 30
23 R=R-B(1)/C(2)
30 DO 37 II=1,NP
I=NC-II
B(I)=H(I)-P*B(I+1)-Q*B(I+2)
37 C(I)=B(I)-P*C(I+1)-Q*C(I+2)
IF (H(2))32,31,32
31 IF (ABS(B(2)/H(1))-E)33,33,34
32 IF (ABS(B(2)/H(2))-E)33,33,34
33 IF (ABS(B(1)/H(1))-E)70,70,34
34 CBAR=C(2)-B(2)
D=C(3)**2-CBAR*C(4)
IF (D)36,35,36
35 P=P-2.
Q=Q*(Q+1.)
GO TO 49
36 P=P+(B(2)*C(3)-B(1)*C(4))/D
49 CONTINUE
Q=Q+(-B(2)*CBAR+B(1)*C(3))/D
E=E*10.
GO TO 20
50 NC=NC-1
V(NC)=0.
IF (IREV)51,52,52
51 U(NC)=1./R
GO TO 53
52 U(NC)=R
53 DO 54 I=1,NC
54 H(I)=B(I+1)
GO TO 4
70 NC=NC-2
IF (IREV)71,72,72
71 QP=1./Q
PP=P/(Q*2.0)
GO TO 73
72 QP=Q

```

LISTAGEM DE RAPOL

(Cont.)

```

PP=P/2.0
73 F=(PP)**2-QP
   IF(F)74,75,75
74 U(NC+1)=-PP
   U(NC)=-PP
   V(NC+1)=SQRT(-F)
   V(NC)=-V(NC+1)
   GO TO 76
75 U(NC+1)=- (PP/ABS(PP))*(ABS(PP)+SQRT(F))
   V(NC+1)=0.

```

```

   U(NC)=QP/U(NC+1)
   V(NC)=0.
76 DO 77 I=1,NC
77 H(I)=B(I+2)
   GO TO 4
100 RETURN
   END

```

A.17 - Subrotina SCMAT

Esta subrotina é utilizada para se substituir uma matriz pela mesma multiplicada por uma constante.

O modo de chamá-la é:

```
CALL SCMAT (SCAL, XSC, N, N)
```

Os argumentos são:

SCAL - escalar pelo qual será multiplicada a matriz

XSC - matriz que será multiplicada por SCAL

N - número de linhas de XSC

M - número de colunas de XSC

LISTAGEM DE SCMAT

```

SUBROUTINE SCMAT(SCAL,XSC,N,M)
DIMENSION XSC(10,10)
DO 1 I=1,N
DO 1 K=1,M
1 XSC(I,K)=SCAL*XSC(I,K)
RETURN
END

```

A.18 - Subrotina SCMUT

Esta subrotina é utilizada para se multiplicar uma matriz por uma constante e retornar o resultado em outra matriz, sem alterar a primeira.

O modo de chamá-la é:

```
CALL SCMUT (SCALA, A, B, L, M)
```

O significado dos argumentos é:

SCALA - constante pela qual será multiplicada a matriz

A - matriz que será multiplicada por SCALA

B - matriz que retornará com o produto

L - número de linhas de A

M - número de colunas de A

LISTAGEM DE SCMUT

```

SUBROUTINE SCMUT(SCALA,A,B,L,M)
DIMENSION A(10,10),B(10,10)
DO 1 I=1,L
DO 1 J=1,M
1 B(I,J)=SCALA*A(I,J)
RETURN
END

```

A.19 - Subrotina SOMA

Esta subrotina é utilizada para se somar duas matrizes e o resultado retornar em uma terceira matriz. O modo de chamá-la é:

CALL SOMA (A, B, C, M, N)

O significado dos argumentos é:

A - B - matrizes que serão somadas

C - matriz que retorna com a soma de A e B

M - número de linhas de A e B

N - número de colunas de A e B

LISTAGEM DE SOMA

```

SUBROUTINE SOMA(A,B,C,M,N)
DIMENSION A(10,10),B(10,10),C(10,10)
DO 1 I=1,M
DO 1 J=1,N
1 C(I,J)=A(I,J)+B(I,J)
RETURN
END

```

A.20 - Subrotina TRNPZ

A subrotina TRNPZ é utilizada para se achar a transposta de uma dada matriz, retornando a transposta em uma segunda matriz.

O modo de chamá-la é:

SCALL TRNPZ (A, V, M, N)

Onde os argumentos são:

A - matriz da qual se quer a transposta

V - matriz que retorna com a transposta de A

M - número de linhas de A

N - número de colunas de A

LISTAGEM DE TRNPZ

```

SUBROUTINE TRNPZ(A,V,M,N)
DIMENSION A(10,10),V(10,10)
DO 14 I=1,M
DO 14 J=1,N
14 V(J,I)=A(I,J)
RETURN
END

```

A.21 - Subrotina XMIRE

A subrotina XMIRE é utilizada para se determinar a matriz de transição, $\exp (At)$, da matriz A.

O modo de chamá-la é:

```
CALL XMIRE (N, A, EIGR, EIGI, IKNOW)
```

O significado dos argumentos é:

N - ordem da matriz A

A - matriz da qual se quer a matriz de transição

EIGR - vetor que representa a parte real dos autovalores da matriz A

EIGI - vetor que representa a parte imaginária dos autovalores da matriz A

IKNOW - = 0 a matriz de transição é impressa

≠ 0 a matriz de transição não é impressa

LISTAGEM DE XMTRE

```

SUBROUTINE XMTRE (N,A,EIGR,EIGI,IKNOW)
C ESTA SUBROTINA DETERMINA A MATRIZ ESTADO DE TRANSICAO USANDO
C O TEOREMA DA ESPANSAO DE SYLVESTER
  DIMENSION A(10,10),EIGR(10),EIGI(10),SPS(10,10),CAI(10,10),
  1 CAR(10,10),CA1I(10,10),CA1R(10,10),CA2I(10,10),CA2R(10,10),
  2 TCAI(10,10),TCAR(10,10),DENOI(10),DENOR(10),CEIGI(10),CEIGR(10)
  COMMON CHI(10,10,10)
1000 FORMAT(1H0,5X,'OS ELEMENTOS DA MATRIZ TRANSICAO DE ESTADO')
1001 FORMAT(1H0,5X,'OS COEFICIENTES MATRICIAIS DE EXP('',E13.6,
  3')T*COS('',E13.6,'')T'//)
1002 FORMAT(6X,10(E10.3,2X))
1003 FORMAT(1H0,5X,'OS COEFICIENTES MATRICIAIS DE EXP('',E13.6,
  4')T*SEN('',E13.6,'')T'//)
1004 FORMAT(1H0,5X,'OS COEFICIENTES MATRICIAIS DE EXP('',E13.6,'')T'//)
1005 FORMAT(1H0,60(1H*))
  IF(IKNOW)800,900,800
  900 WRITE(3,1005)
  800 DO 10 K=1,N
    CEIGR(K)=EIGR(K)
    CEIGI(K)=EIGI(K)
    DO 10 L=1,N
      CAR(K,L)=A(K,L)
  10 CAI(K,L)=0.0
    I=1
    IF(IKNOW)700,701,700
  701 WRITE(3,1000)
  700 DO 15 K=1,N
    DENOI(K)=CEIGI(I)-CEIGI(K)
  15 DENOR(K)=CEIGR(I)-CEIGR(K)
    DO 500 J=1,N
      IF(J-I)100,500,200
  100 IF(J-1)110,110,150
  200 IF(I-1)300,300,400
  300 IF(J-I-1)110,110,150
  400 IF(J-I-1)110,150,150
  110 DO 5 K=1,N
    DO 5 L=1,N
      CA1I(K,L)=CAI(K,L)
  5 CA1R(K,L)=CAR(K,L)
    DO 20 K=1,N
      CA1I(K,K)=CAI(K,K)-CEIGI(J)
      CA1R(K,K)=CAR(K,K)-CEIGR(J)
    DO 20 L=1,N
      STORE=CA1R(K,L)
      CA1R(K,L)=(CA1R(K,L)*DENOR(J)+CA1I(K,L)*DENOI(J))/(DENOR(J)**2 +
  5 DENOI(J)**2)
  20 CA1I(K,L)=(CA1I(K,L)*DENOR(J)-STORE*DENOI(J))/(DENOR(J)**2+

```

UNIVERSIDADE FEDERAL DA PARAÍBA

Pró-Reitoria Para Assuntos do Interior

Coordenação Setorial de Pós-Graduação

Rua Antônio Veloso 892 Tel (83) 321-7222-R 355

53109 - Campina Grande - Paraíba

LISTAGEM DE XMIRE

(Cont.)

```

6  DENOI(J)**2)
   GO TO 500
150 DO 40 K=1,N
     DO 40 L=1,N
       CA2R(K,L)=CAR(K,L)
40  CA2I(K,L)=CAI(K,L)
     DO 25 K=1,N
       CA2R(K,K)=CAR(K,K)-CEIGR(J)
       CA2I(K,K)=CAI(K,K)-CEIGI(J)
     DO 25 L=1,N
       TEMP=CA2R(K,L)
       CA2R(K,L)=(CA2R(K,L)*DENOR(J)+CA2I(K,L)*DENOI(J))/(DENOR(J)**2 +
7  DENOI(J)**2)
25  CA2I(K,L)=(CA2I(K,L)*DENOR(J)-TEMP*DENOI(J))/(DENOR(J)**2+
8  DENOI(J)**2)
     DO 30 K=1,N
       DO 30 L=1,N
         TCAI(K,L)=0.0
         TCAR(K,L)=0.0
     DO 30 M=1,N
       TCAR(K,L)=TCAR(K,L)+CA1R(K,M)*CA2R(M,L)-CA1I(K,M)*CA2I(M,L)
30  TCAI(K,L)=TCAI(K,L)+CA1I(K,M)*CA2R(M,L)+CA1R(K,M)*CA2I(M,L)
     DO 35 K=1,N
       DO 35 L=1,N
         CA1I(K,L)=TCAI(K,L)
35  CA1R(K,L)=TCAR(K,L)
500 CONTINUE
     IF(CEIGI(I))45,50,45
45  IM=I
     I=I+1
     IF(IKNOW) 801,11,801
11  WRITE(3,1001) EIGR(I),EIGI(I)
801 DO 65 K=1,N
     DO 65 L=1,N
65  SPS(K,L)=CA1R(K,L)*2.0
     DO 66 K=1,N
       DO 66 L=1,N
         CHI(IM,K,L)=SPS(K,L)
66  CONTINUE
     IF(IKNOW) 802,12,802
12  DO 80 K=1,N
80  WRITE(3,1002) (SPS(K,L),L=1,N)

```

LISTAGEM DE XMIRE

(Cont.)

```
WRITE(3,1003) EIGR(I),EIGI(I)
802 DO 55 K=1,N
    DO 55 L=1,N
55 SPS(K,L)=CALI(K,L)*2.0
    DO 56 K=1,N
    DO 56 L=1,N
    CHI(IM,K,L)=SPS(K,L)
56 CONTINUE
    IF(IKNOW) 600,13,600
13 DO 85 K=1,N
85 WRITE(3,1002) (SPS(K,L),L=1,N)
    GO TO 600
50 IF(IKNOW) 804,14,804
14 WRITE(3,1004) EIGR(I)
804 DO 60 K=1,N
    DO 60 L=1,N
60 SPS(K,L)=CALR(K,L)
    DO 61 K=1,N
    DO 61 L=1,N
    CHI(I,K,L)=SPS(K,L)
61 CONTINUE
    IF(IKNOW) 600,16,600
16 DO 75 K=1,N
75 WRITE(3,1002) (SPS(K,L),L=1,N)
600 III=I-N
    IF(III) 17,18,18
17 I=I+1
    GO TO 700
18 CONTINUE
    RETURN
    END
```

A.22 - Subrotina VALMX

Esta subrotina é utilizada para se achar o maior elemento, em valor absoluto, de uma matriz.

O modo de chamá-la é:

CALL VALMX (T, TMAX, N, M)

O significado dos argumentos é:

T - matriz da qual se quer o maior elemento

TMAX - maior elemento, em valor absoluto, da matriz T

N - número de linhas de T

M - número de colunas de T

LISTAGEM DE VALMX

```
SUBROUTINE VALMX(T,TMAX,N,M)
  DIMENSION T(10,10)
  TMAX=ABS(T(1,1))
  DO 2 I=1,N
  DO 2 J=1,M
  IF(ABS(T(I,J))-TMAX)2,2,3
  3 TMAX=ABS(T(I,J))
  2 CONTINUE
  RETURN
  END
```

BIBLIOGRAFIA

1. OGATA, K. Modern control engineering. Englewood Cliffs, N.J. Prentice-Hall, 1970.
2. Tou, J.T. Modern control theory. New York, McGraw-Hill Book, 1964.
3. SHINNERS, S.M. Modern control system theory and application. Reading, Massachusetts, Addison - Wesley, 1973.
4. DERRUSSO, P.M., ROY, R.J. and CLOSE, C.M. State variables for engineers. New York, John Wiley & Sons, 1965.
5. OGATA, K. State space analysis of control systems. Englewood Cliffs, N.J., Prentice Hall, 1967.
6. PENNINGTON, R.H. Introductory computer methods and numerical analysis. New York, The Macmillan Company, 1968.
7. PIERRE, D.A. Optimization theory with applications. New York, John Wiley & Sons, 1969.
8. MACCAUSLAND, J. Introduction to optimal control. New York, John Wiley & Sons, 1969.
9. MELSA, J.L. Computational assistance in the study of linear control theory. New York, MacGraw-Hill Book, 1970.
10. TAKAHASHI, Y., RABINS, M.J. and AUSLANDER, L.M. Control and dynamic systems. New York, Addison-Wesley Publishing company, 1970.
11. ANDERSON, B.D.O. & MOORE, J.B. Linear optimal control. Englewood Cliffs, N.J., Prentice Hall, 1971.

12. GANTMACHER, F.R. The theory of matrices. New York, Chelsea Publishing Company, 1960. V.1.
13. ATHANS, M. & FALB, P.L. Optimal control: an introduction to the theory and its applications. New York, McGraw-Hill, 1966.
14. DAVISON, E.J. & MAKY, M.C. The numerical solution of the matrix ricatti differential equation. IEEE-Transactions on automatic control New York, 1973. n° 1 Vol. 18 : 71-73, feb. 1973
15. TAYLOR, F. Coments on the numerical solution of the matrix ricatti differential equation. IEEE - Transactions on automatic control . New York, no. 1, V. 19 : 92-93, feb. 1974.
16. MAN, F.T. The davidon method of solution of the algebraic matrix riccati equation. Int. J. Control. New York, No. 6, V. 10 : 713-719 , 1969.
17. MEYER, G.G.J. & PAYNE, H.J. An Interative method of solution of the algebraic riccati, equation. IEEE - Transactions on automatic control. New York, 550-551, aug. 1972.
18. VIT, K. Iterative solution of the riccati equation. IEEE-Transactions on automatic control. New York 258-259, apr. 1972.
19. PRUSSING, J.E. A simplified method for solving the matrix riccati equation. Int. J. Control. New York, no. 5, V. 15: 995-1.000, 1972
20. MAKI, M.C. Numerical solution of the algebraic riccati for single - input systems. IEEE-Transactions on automatic control. New York 264-265.
21. VAUGHAN, D.R. A negative exponential solution for the matrix riccati equation. IEEE Transactions on automatic control. New York, 72-75 . feb. 1969.

22. CRUZ Jr, J.B. Feedback systems. New York, McGraw Hill, 1971.
23. PACCITTI, T. Fortran monitor princípios. Rio de Janeiro, Ao Livro Técnico, 1969.
24. CONTE, S.D. Elementos de análise numérica. Porto Alegre, Ed. Globo 1971.
25. HAMMING, R.W. Numerical methods for scientists and engineers. New York, McGraw Hill, 1962.