

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA/ELETRÔNICA

IMPLEMENTAÇÃO DE REGULADORES AUTO-AJUSTÁVEIS
EM PEQUENOS PROCESSADORES

LÉO PINI MAGALHÃES

Orientador: Prof. MANUEL DE JESUS MENDES

Tese de Mestrado apresentada à
Faculdade de Engenharia da Uni
versidade Estadual de Campinas

Financiada pelo CNPq sob con -
trato nº 1561/74

MARÇO - 1977

UNICAMP
BIBLIOTECA CENTRAL

AGRADECIMENTOS

Em primeiro lugar, como não poderia deixar de ser, o agradecimento aos iniciadores e maiores responsáveis por esse trabalho, o qual iniciaram há 25 anos atrás ; meus pais Clélia e Geraldino.

Também meu agradecimento aos colaboradores mais diretos, Dr. Manuel de Jesus Mendes, meu orientador, os professores Dr. Márcio Luis de Andrade Netto e Wagner Caradori do Amaral, pelas discussões elucidativas e às Srtas. Elza Aoki e Maria Ângela Sigríst, responsáveis respectivamente pelos serviços de datilografia e desenho, executados com máxima dedicação e boa vontade.

Finalmente um agradecimento a todos que sintam ter colaborado de alguma forma para o êxito desse trabalho.

RESUMO

Esse trabalho trata da implementação de um regulador auto-ajustável, conhecido na literatura como regulador "self-tuning", em um mini-processador.

São apresentadas as principais características do controlador, analisadas através de simulações num computador de grande porte.

O controlador é implementado num mini-processador e os resultados são comparados com os obtidos no computador de grande porte.

Lista dos principais símbolos utilizados:

Um polinômio de ordem r , com o operador q é denotado:

$$P(q) = p_0 q^r + q^{r-1} + \dots + p_r$$

Com essa definição, onde for conveniente, $P(q)$ será denotado por P .

$A(q)$: polinômio de ordem n , com $a_0 = 1$

$B(q)$: polinômio de ordem $n-1$

$C(q)$: polinômio de ordem n , com $c_0 = 1$

$\epsilon\{\}$: operador esperança

$z(t)$: sequência aleatória de valores não correlatos, do tipo $N(0, 1)$

k : atraso no tempo do sistema

n : ordem do sistema

$\underline{P}(t)$: matriz proporcional a covariância dos parâmetros estimados

$\underline{S}$: raiz quadrada de $\underline{P}$

$\underline{K}(t)$: ganho do "least-square" recursivo

$u(t), y(t)$: saída e controle (entrada) no instante t

t : tempo em instantes de amostragem (inteiro)

ÍNDICE

CAPÍTULO I - INTRODUÇÃO	1
1.1 - Bibliografia	10
CAPÍTULO II - ESTUDO DO DESEMPENHO DOS ALGORÍTMOS DE CONTROLE . . .	11
2.1 - Análise dos parâmetros do algoritmo de controle	11
2.1.1 - Limitação no sinal de controle	11
2.1.2 - Fator de ponderação na matriz P	13
2.1.3 - Número de coeficientes do regulador	20
2.1.4 - Fator de escala do regulador (β_0)	22
2.2 - Algoritmo RAT, utilizando aproximação esto- cástica na estimação	24
2.3 - Algoritmo RAT, utilizando o método da raiz quadrada na estimação	26
2.4 - Simulação dos algoritmos RAT	28
2.4.1 - Algoritmo RAT básico	28
2.4.2 - Algoritmo RAT, utilizando aproxima- ção estocástica na estimação	32
2.4.3 - Algoritmo RAT, utilizando o método da raiz quadrada na estimação	36
2.4.4 - Conclusões	40
2.5 - Bibliografia	41
CAPÍTULO III - IMPLEMENTAÇÃO DO CONTROLADOR RAT NUM MINI-COM - PUTADOR	42
3.1 - Caracterização do PDP-11/05	42
3.2 - Rotinas aritméticas de ponto fixo e flutuante	52

3.2.1 - Rotinas de ponto fixo	52
3.2.2 - Rotinas de ponto flutuante	52
3.3 - Rotinas para transmissão de dados entre o PDP-11/05 e PDP-10	61
3.4 - Implementação do algoritmo RAT, utilizando aproximação estocástica na estimação	67
3.5 - Implementação do algoritmo RAT, utilizando o método da raiz quadrada na estimação	70
3.6 - Conclusões	74
3.7 - Bibliografia	77
CAPÍTULO IV - PROJEÇÃO PARA O USO DE MICRO-PROCESSADORES	78
4.1 - Bibliografia	83
BIBLIOGRAFIA COMPLEMENTAR	84
ANEXOS	85

CAPÍTULO 1

INTRODUÇÃO

Neste trabalho é apresentada a descrição de um regulador auto-ajustável, conhecido na literatura por regulador "self-tuning" e aqui abreviado por RAT.

Aplicar-se-ã o regulador a sistemas discretos, com uma entrada e uma saída e com coeficientes desconhecidos.

A idéia de identificar um processo enquanto ele ainda es_tã decorrendo, vem do fato do controle poder melhor se adaptar as condições de vizinhança do processo dinâmico.

O regulador RAT, é um controlador não-dual, que minimiza a variância da saída do sistema sob o qual atua e ao qual se aplica o princípio da certeza e equivalência [1].

O regulador é obtido separando-se controle e estimação, que são desenvolvidos em partes distintas do algoritmo, como se re_presenta esquematicamente na figura 1.1.

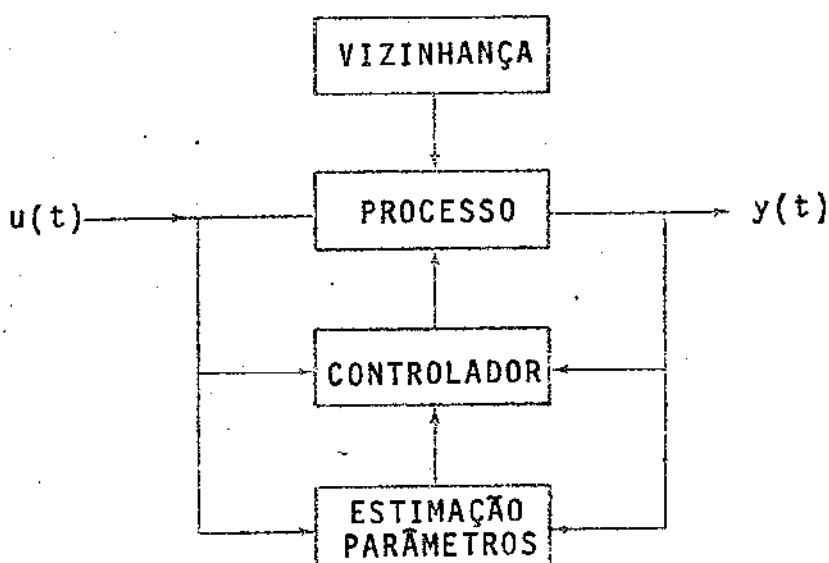


FIGURA 1.1

Na implementação do regulador, como foi proposto em [2], unem-se um estimador dos mínimos quadrados e um controlador que

minimiza uma certa função de custo. O regulador RAT, será tal que, assintoticamente, tenderá para aquele que minimiza a função de custo definida, se o sistema é conhecido.

Uma das possíveis funções de custo, a adotar, é aquela originalmente definida para o regulador de variância mínima, RVM, introduzido por ASTRÖM em [3]:

$$V = \epsilon \{y^2(t+k)\} \quad (1.1)$$

A seguir desenvolvem-se as equações que descrevem o RAT, minimizando a equação (1.1).

Seja um sistema descrito por:

$$A(q) y(t) = B(q) u(t-k) + C(q) z(t) \quad (1.2)$$

Para facilidade de notação, usar-se-á A, B, C ao invés de $A(q), B(q), C(q)$, sempre que não houver possibilidade de dúvida.

A propriedade do RVM, que induz o RAT, é que a escolha de um valor apropriado para $u(t)$, torna $y(t+k)$ zero, em cada passo.

Reescrevendo a equação (1.2), obtêm-se:

$$y(t+k) = \frac{B}{A} u(t) + \frac{C}{A} z(t+k) \quad (1.3)$$

E o termo $\frac{C}{A} z(t+k)$ pode ser extendido como:

$$\frac{C}{A} z(t+k) = (z(t+k) + r_1 z(t+k-1) + \dots + r_k z(t)) + (r_{k+1} z(t-1) + \dots) \quad (1.4)$$

Definindo:

$$F(q) = q^k + f_1 q^{k-1} + \dots + f_k \quad (1.5)$$

$$G(q) = g_0 q^{n-1} + \dots + q_{n-1}$$

pode-se reescrever a equação (1.4) por:

$$\frac{C}{A} \zeta(t + k) = F \zeta(t) + \frac{G}{A} \zeta(t) \quad (1.6)$$

em que:

$$\frac{q^k C}{A} = F + \frac{G}{A} \quad (1.7)$$

A equação (1.3) permite a seguinte formulação:

$$y(t + k) = \frac{BF}{C} q^{-k} u(t) + \frac{G}{C} y(t) + F \zeta(t) \quad (1.8)$$

Minimizando-se a função de custo descrita pela equação (1.1) e considerando-se a sequência $\{\zeta(t + k), \dots, \zeta(t)\}$, uma sequência aleatória de valores não correlatos e independente das sequências $\{u(t - k - 1), u(t - k - 2), \dots\}$, $\{y(t - 1), y(t - 2), \dots\}$ e $\epsilon \{\zeta(t)\} = 0$, obtêm-se:

$$\epsilon \left[y^2(t + k) \right] = \epsilon \left[(F \zeta(t))^2 \right] + \epsilon \left[\left(\frac{BF}{C} q^{-k} u(t) + \frac{G}{C} y(t) \right)^2 \right] \quad (1.9)$$

$$\text{Para o RVM, } \epsilon \left[y^2(t + k) \right] = \epsilon \left[(F \zeta(t))^2 \right] \quad (1.10)$$

Assim, obtêm-se a lei de controle:

$$u(t) = -q^k \frac{G(q)}{B(q) F(q)} y(t) \quad (1.11)$$

E comparando-se as equações (1.11) e (1.8), obtêm-se:

$$y(t + k) = F(q) \zeta(t) \quad (1.12)$$

Na equação (1.11) o termo $B(q) F(q)$, é um polinômio de ordem $n - 1 + k$, que pode ser reescrito como:

$$B(q) F(q) = d_0 q^{n+k-1} + d_1 q^{n+k-2} + \dots + d_{n+k-1} \quad (1.13)$$

Tomando-se $\ell = n + k - 1$, obtêm-se:

$$u(t) (d_0 q^\ell + \dots + d_\ell) = -q^k (g_0 q^{n-1} + \dots + g_{n-1}) y(t) \quad (1.14)$$

e redefinindo os coeficientes:

$$u(t) = \frac{1}{\beta_0} \left[\alpha_1 y(t) + \alpha_2 y(t-2) + \dots + \alpha_n y(t-n+1) \right] - \beta_1 u(t-1) - \dots - \beta_\ell u(t-\ell) \quad (1.15)$$

$$\text{ou:} \quad u(t) = q^{\ell-n+1} \frac{A(q)}{B(q)} y(t) \quad (1.16)$$

$$\text{onde} \quad A(q) = \alpha_1 q^{n-1} + \alpha_2 q^{n-2} + \dots + \alpha_n$$

$$B(q) = \beta_0 \left[q^\ell + \beta_1 q^{\ell-1} + \dots + \beta_\ell \right]$$

Esses coeficientes são os mesmos da equação (1.8), quando se supõe $C(q) = q^n$. Reescrevendo a equação (1.8), obtêm-se:

$$y(t) + \alpha_1 y(t-k-1) + \dots + \alpha_n y(t-k-n) = \beta_0 \left[u(t-k-1) + \beta_1 u(t-k-2) + \dots + \beta_\ell u(t-k-\ell-1) \right] + \varepsilon(t) \quad (1.17)$$

onde $\{\varepsilon(t)\}$, é uma sequência do tipo média móvel de ordem k da sequência $\{z(t)\}$.

A equação (1.17), descreve o sistema a ser controlado e então, seus coeficientes são os obtidos através do estimador dos mínimos quadrados.

Definindo-se os vetores:

$$\underline{\phi}(t) = \left[-y(t) \dots -y(t-n+1) \quad ; \quad \beta_0 u(t-1) \dots \beta_\ell u(t-\ell) \right] \quad (1.18)$$

$$\underline{\theta}^T(t) = \begin{bmatrix} \alpha_1 & \dots & \alpha_n & | & \beta_1 & \dots & \beta_\ell \end{bmatrix} \quad (1.18)$$

pode-se reescrever a equação (1.17) como:

$$y(t) = \beta_0 u(t - k - 1) + \underline{\phi}(t - k - 1) \underline{\theta}(t) + \varepsilon(t) \quad (1.19)$$

E definindo:

$$\underline{\tilde{y}}(t) = \begin{bmatrix} y(0) - \beta_0 u(-k-1) \\ \vdots \\ y(t-1) - \beta_0 u(t-k-2) \end{bmatrix} \quad (1.20)$$

$$\underline{Y}(t) = \begin{bmatrix} -y(-k-1) \dots -y(-k-n) & | & \beta_0 u(-k-2) \dots \beta_0 u(-k-\ell-1) \\ \vdots & & \vdots \\ -y(t-k-2) \dots -y(t-k-n-1) & | & \beta_0 u(t-k-3) \dots \beta_0 u(t-k-\ell-2) \end{bmatrix} \quad (1.21)$$

$$\underline{\varepsilon}(t) = \begin{bmatrix} \varepsilon(0) \\ \vdots \\ \varepsilon(t-1) \end{bmatrix} \quad (1.22)$$

Chega-se a: $\underline{\tilde{y}}(t) = \underline{Y}(t) \underline{\theta}(t) + \underline{\varepsilon}(t) \quad (1.23)$

e

$$\underline{\varepsilon}(t) = \underline{\tilde{y}}(t) - \underline{Y}(t) \underline{\theta}(t) \quad (1.24)$$

O estimador dos coeficientes $\underline{\theta}(t)$, é obtido minimizando o erro quadrático. Em [4], tem-se a dedução desse algoritmo em sua forma recursiva, que abaixo é transcrita:

$$\hat{\underline{\theta}}(t+1) = \hat{\underline{\theta}}(t) + \gamma(t) \underline{P}(t) \underline{\phi}^T(t-k-1) \left(y(t) - \beta_0 u(t-k-1) - \underline{\phi}(t-k-1) \hat{\underline{\theta}}(t) \right) \quad (1.25)$$

$$\gamma(t) = \left[1 + \underline{\phi}(t-k-1) \underline{P}(t) \underline{\phi}^T(t-k-1) \right]^{-1} \quad (1.26)$$

$$\underline{P}(t+1) = \underline{P}(t) - \gamma(t) \underline{P}(t) \underline{\phi}^T(t-k-1) \underline{\phi}(t-k-1) \underline{P}(t) \quad (1.27)$$

$$\underline{P}^{-1}(t+1) = \underline{P}^{-1}(t) + \underline{\phi}^T(t-k-1) \underline{\phi}(t-k-1) \quad (1.28)$$

Desse algoritmo básico, descrito pelas equações (1.25) a (1.27), pode-se derivar um outro que prevê pequenas variações nos coeficientes do sistema, no decorrer da simulação. Para isso, introduz-se na expressão de $\underline{P}(t+1)$, termo proporcional à matriz de covariância dos coeficientes estimados, um fator que pondere os instantes passados, denotado por RHO .

Então, obtêm-se:

$$\hat{\underline{\theta}}(t+1) = \hat{\underline{\theta}}(t) + \gamma(t) \underline{P}(t) \underline{\phi}^T(t-k-1) \left(y(t) - \beta_0 u(t-k-1) - \underline{\phi}(t-k-1) \hat{\underline{\theta}}(t) \right) \quad (1.29)$$

$$\underline{P}(t+1) = \frac{1}{RHO} \left[\underline{P}(t) - \gamma(t) \underline{P}(t) \underline{\phi}^T(t-k-1) \underline{\phi}(t-k-1) \underline{P}(t) \right] \quad (1.30)$$

$$\gamma(t) = \left[RHO + \underline{\phi}(t-k-1) \underline{P}(t) \underline{\phi}^T(t-k-1) \right]^{-1} \quad (1.31)$$

O regulador RAT, apresenta propriedades interessantes, que são apresentadas na forma de dois teoremas, cuja demonstração pode ser encontrada em [5].

TEOREMA 1

Assumindo que os valores estimados para os coeficientes

$\alpha_i(t)$, $i = 1, \dots, n$ e $\beta_i(t)$, $i = 1, \dots, \ell$ do modelo descrito pela equação (1.17) convergem quando $t \rightarrow \infty$ e que o sistema de malha fechada tem saída ergódica, nos momentos de segunda ordem, quando adotada a lei de controle dada pela equação (1.11), o sistema de malha fechada tem as seguintes propriedades:

$$E[y(t + \tau) y(t)] = r_y(\tau) = 0, \text{ para } \tau = k + 1, \dots, k + n \quad (1.32)$$

$$E[y(t + \tau) u(t)] = r_y u(\tau) = 0, \text{ para } \tau = k + 1, \dots, k + \ell + 1$$

NOTA: É interessante observar, que esse teorema fornece um índice (as correlações), para se avaliar o comportamento do controlador, já que a saída do processo, deve ser uma sequência não correlata de valores, conforme a equação (1.12).

TEOREMA 2

Seja o sistema a ser controlado, descrito pela equação (1.2). Supondo que o algoritmo RAT seja utilizado com n e $\ell = n + k - 1$ coeficientes e que os coeficientes estimados convergem para valores tais que os polinômios A e B não tem fatores comuns, então o regulador obtido convergirá para o de variância mínima.

NOTA: Observe-se que, por esse teorema, o algoritmo RAT pode convergir para o RVM, mesmo que a sequência do ruído seja correlata, contrariando o exigido para se obter a equação (1.17). Esse resultado é inesperado, pois o estimador dos mínimos quadrados apresenta, nesse caso, valores estimados polarizados.

Com a minimização da função de custo dada pela equação (1.1), não é possível levar a saída para um valor de referência diferente de zero. Assim, para resolver esse problema, propõe-se uma outra função de custo, a ser minimizada:

$$V = \epsilon \left((y(t+k) - w(t))^2 + \lambda u^2(t) \right) \quad (1.33)$$

onde: $w(t)$: ponto de operação no instante t
 λ : peso do controle $u(t)$

onde o fator de ponderação λ , limita o sinal de controle em troca de um acréscimo na variância da saída [6].

A minimização de (1.33) leva a:

$$u(t) = - \frac{1}{E} \left(G y(t) - C w(t) \right) \quad (1.34)$$

onde $E = F B + \lambda C$

Em [6], são apresentados diversos exemplos e a dedução detalhada da equação (1.34), que aqui aparece somente para ilustrar a possibilidade do uso de outras funções de custo para obter algoritmos para o controlador auto-ajustável.

Os controladores tratados, são não-duais, pois consideram para a obtenção da função de custo as observações feitas até o instante a controlar, assumindo não existir qualquer informação posterior. Necessariamente serão sub-ótimos com relação ao controlador dual, que considera as observações futuras, através de suas distribuições de probabilidade calculadas com a informação disponível no presente.

Então, enquanto no controlador não-dual minimiza-se a função de custo no passo seguinte, para o controlador dual minimiza-se essa função para todo o intervalo de controle.

A função de custo a minimizar para obter um controlador não-dual, é a descrita pela equação (1.1). Para o controlador dual, adota-se uma outra função de custo, dada por:

$$V = \epsilon \left\{ \sum_{s=0}^{N-1} f_s(y(s), u(s)) \right\} \quad (1.35)$$

A minimização dessa equação, em geral é um problema trabalhoso, não sendo abordado nesse trabalho, ver por exemplo [7].

No capítulo seguinte, analisa-se a influência de diversos parâmetros no comportamento transitório e assintótico do RAT. Assim estuda-se a influência de:

- fator de ponderação na matriz $P(t)$;
- limitação do sinal de controle;
- variação no número de coeficientes;
- fator de escala do regulador.

No capítulo 3, são apresentadas duas versões do algoritmo RAT, implementadas no mini-computador PDP-11/05 e controlando um sistema simulado no PDP-10, ao qual o mini-computador, está ligado em "time-sharing". É desenvolvido ainda um conjunto de rotinas para possibilitar o uso de operações em ponto flutuante, para permitir um melhor desempenho do regulador auto-ajustável.

Finalmente, no capítulo 4, são apresentadas as conclusões do trabalho, bem como uma projeção da implementação desses algoritmos por meio de micro-processadores.

1.1 - BIBLIOGRAFIA

1. Wittenmark ,B. - "Stochastic adaptive control methods : a survey".Int. Journal Control, 21(5) : 705-730,1975.
2. Wittenmark ,B. - "A self-tuning regulator".Lund Institute of Technology,Dep. Automatic Control, 1973. (Report 7311, abril 1973).
3. Astrom ,K.J. - "Introduction to Stochastic Control Theory." New York ,Academic Press ,1972. 299 p.(Mathematics in Science and Engineering ,v.70).
4. Amaral ,W.C. - "Identificação de Sistemas Discretos por Métodos Sequenciais".Tese de Mestrado - FEC , 1976.
5. Astrom ,K.J.; Wittenmark ,B. - "On self-tuning regulators!" Automatica 9 : 185-189,1973.
6. Clarke ,D.W.; Cope ,S.N.; Gawthrop ,P.J. - "Feasibility study of the application of micro-processors to self-tuning controllers".Oxford.(Report 1137, 1975).
7. Mendes ,M. - "An on-line Adaptive Control Method".Automatica 7 : 323-332,1971.

CAPÍTULO 2

ESTUDO DO DESEMPENHO DOS ALGORÍTMOS DE CONTROLE

2.1 - ANÁLISE DOS PARÂMETROS DO ALGORÍTMO DE CONTROLE

O desempenho do algoritmo RAT, anteriormente apresentado, é influenciado por diversos parâmetros, tanto durante o transi-
tório, como em condição de regime.

A seguir, apresenta-se uma breve discussão sobre os prin-
cipais parâmetros.

2.1.1 Limitação no sinal de controle

Exemplo 2.1: Seja um processo descrito por:

$$y(t) - 0,9y(t - 1) = 0,5u(t - 2) + \zeta(t) \quad (2.1)$$

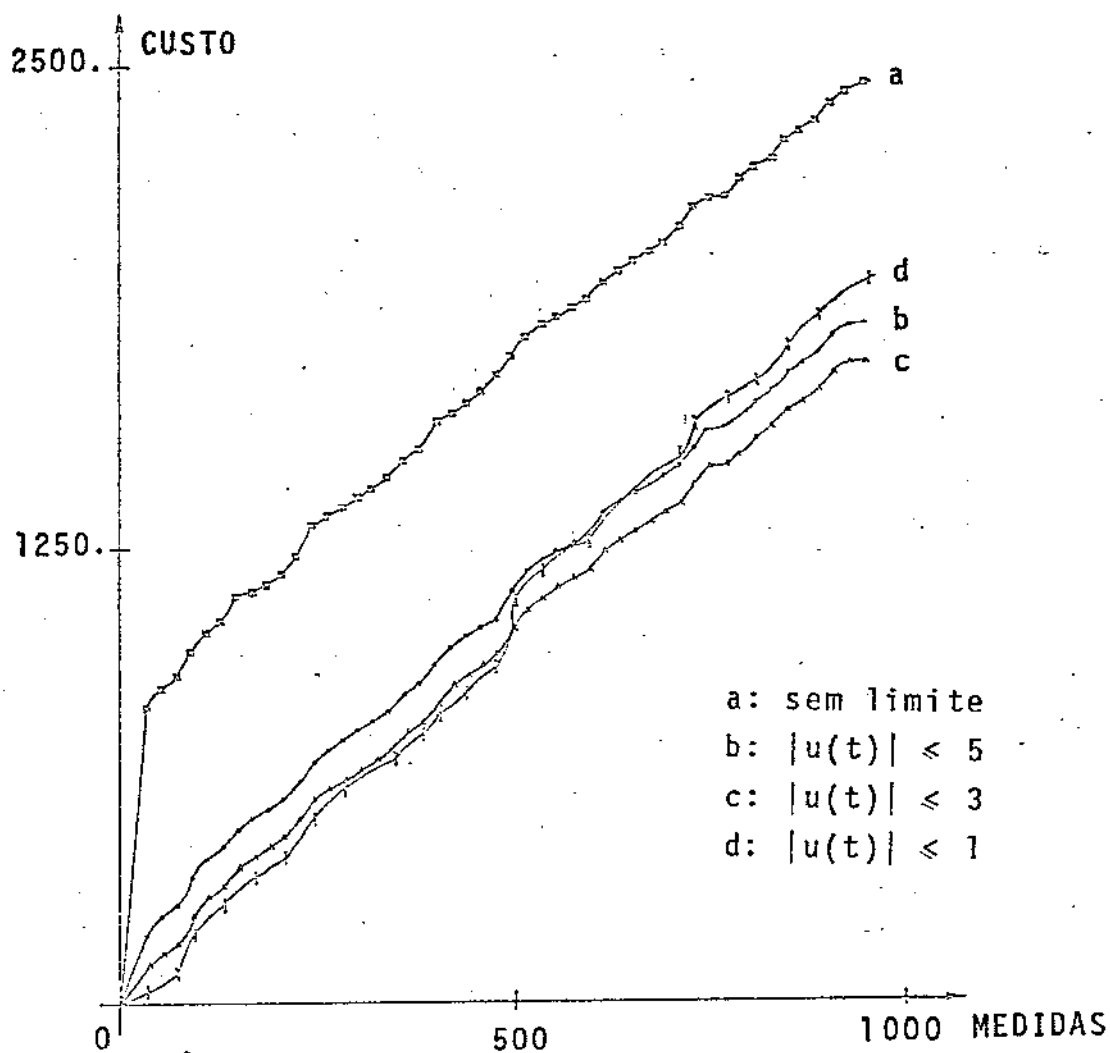


FIGURA 2.1 : Função de custo com vários limites no controle

Como no início da estimação tem-se pouca informação sobre o processo, o estimador poderá obter valores pouco significativos para os coeficientes do modelo, não atuando de maneira eficiente o controlador baseado nesses valores.

Isso pode ser visto na figura 2.1, onde na curva (a) tem-se um acréscimo muito grande no valor da função de custo logo ao início da simulação.

Uma forma de se melhorar o comportamento do algoritmo será limitar o sinal de controle, como demonstram as curvas (b), (c) e (d), da figura 2.1. Uma limitação excessiva pode ser igualmente prejudicial.

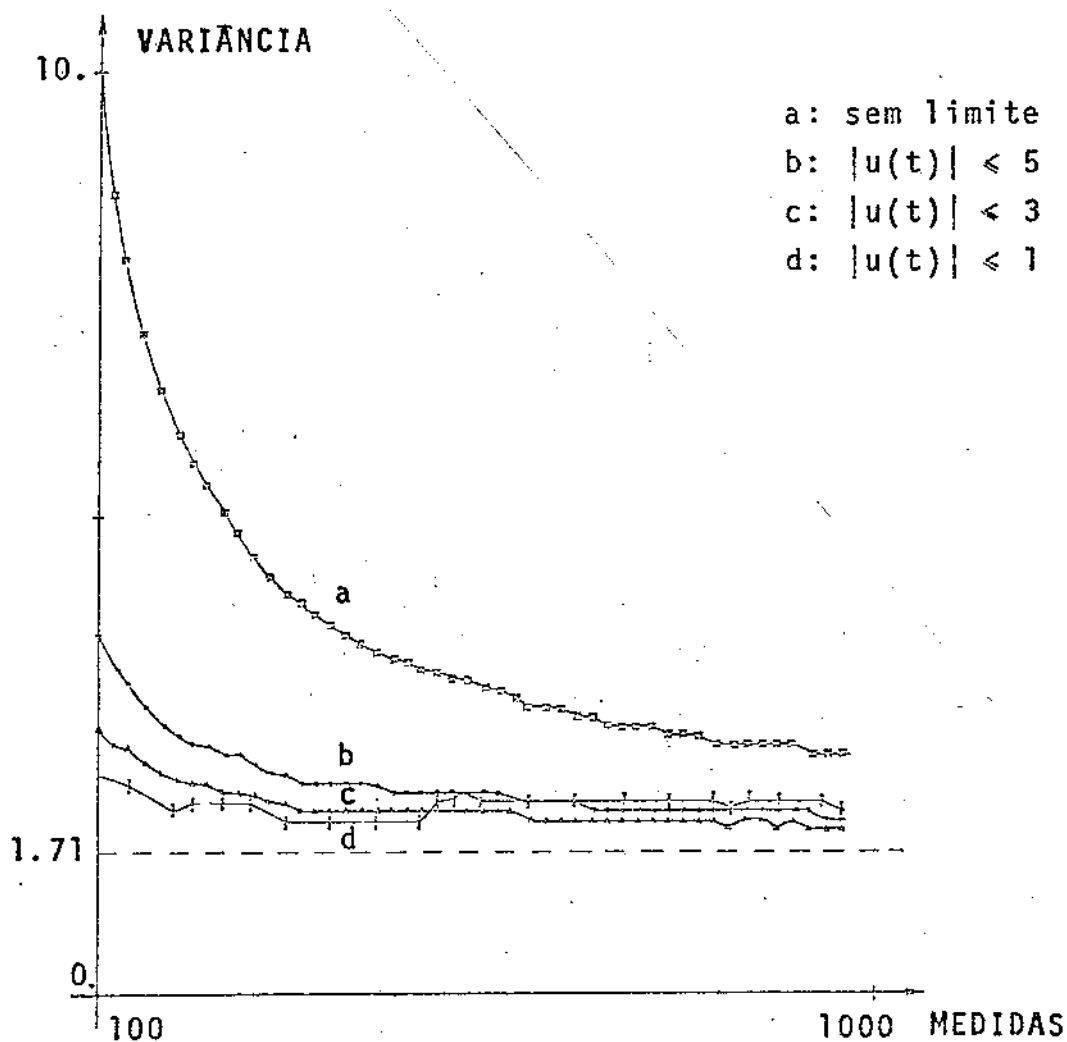


FIGURA 2.2 : Variância da saída para vários limites no controle

Límite	Variância da saída
sem	2,53
1	2,03
3	1,80
5	1,90
regulador de variância mínima	1,71

Tabela 2.1

A política de se limitar o sinal de controle, faz com que o controlador RAT, aproxime seu comportamento, do desempenho esperado de um controlador dual. O controle dual, que fornece um controlador ótimo, considera no presente, que existirão medidas no futuro, sendo portanto mais "cauteloso" quando a qualidade dos valores estimados é baixa.

Assim, limitar o sinal de controle no início da simulação, é uma forma de comunicar ao controlador que são esperados valores mais seguros para os coeficientes nos passos seguintes e então, controlar com um sinal muito forte, pode não corresponder à melhor ação.

2.1.2 Fator de ponderação na matriz P

Como foi visto no capítulo 1, adota-se um fator de ponderação nas equações (1.30) e (1.31), para possibilitar ao algoritmo acompanhar pequenas variações nos valores dos coeficientes estimados para o sistema, bem como para permitir, em troca de um acréscimo no valor da variância da saída, uma mais rápida convergência do algoritmo de estimação.

Exemplo 2.2: Seja um processo descrito por:

$$y(t) - 0,5y(t - 1) = u(t - 1) + \zeta(t) + 0,7\zeta(t - 1) \quad (2.2)$$

O regulador de variância mínima é:

$$u(t) = -1,2y(t) \quad (2.3)$$

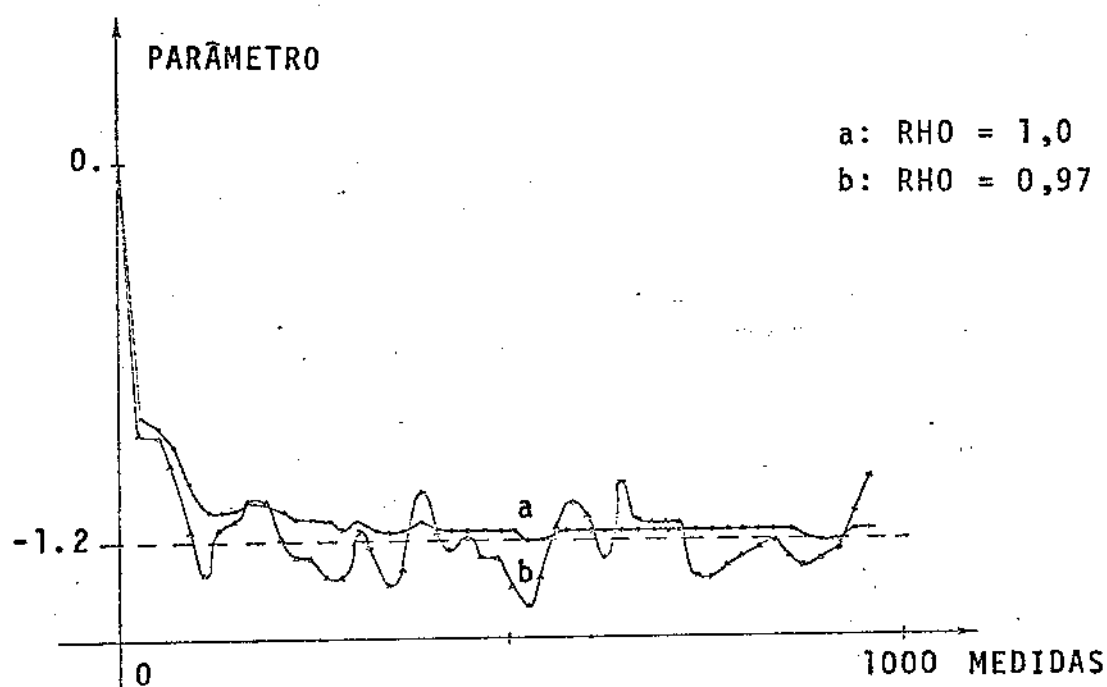


FIGURA 2.3 : Parâmetro estimado

Na figura 2.3, com $RHO < 1$ se consegue uma convergência mais rápida do(s) coeficiente(s) estimado(s), sendo ainda possível acompanhar ligeiras oscilações dos valores dos coeficientes.

Apresenta-se em seguida uma tabela com os valores das variâncias da saída, para diferentes valores de RHO :

RHO	Variância da saída
1,0	0,971
0,97	0,982
regulador de variância mínima	0,944

Tabela 2.2

Observa-se que a variação dada no valor de RHO, para obter a figura 2.3, não modificou de maneira significativa o valor da função de custo (~ % 3 do valor final) e portanto, embora a variância cresça, não alcança valores elevados.

A estratégia ideal para o caso apresentado, segundo simulações executadas, é ter de início um valor para RHO inferior à unidade e, após um determinado número de passos igualá-lo a unidade, evitando desse modo um acréscimo na variância da saída. Isso, com certeza, exigirá algum conhecimento a priori, ou possibilidade de testes "off-line" no sistema.

No presente exemplo, uma boa política foi manter o valor de $RHO = 0,99$ até o passo 450 e, após, incrementá-lo a 1, como mostra a figura 2.4.

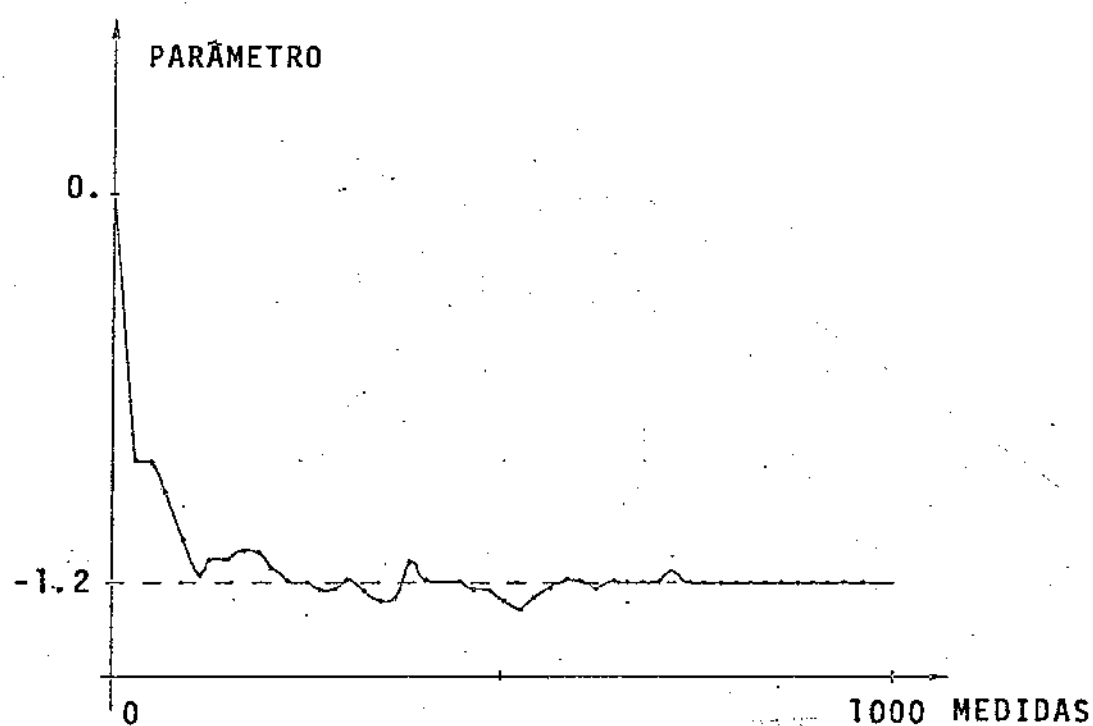


FIGURA 2.4 : Parâmetro estimado com variação de ρ
(variância = 0,966)

2.1.3 Número de coeficientes do regulador

Já foi mostrado que, se as ordens dos polinômios do regulador RAT forem corretas e se seus coeficientes convergirem, o regulador será de variância mínima. Entretanto, na prática, nem sempre a ordem é conhecida.

a. Variações nos valores de n e ℓ

Exemplo 2.3: Seja um processo descrito por:

$$y(t) - 1,9y(t-1) + 0,9y(t-2) = u(t-2) - u(t-3) + \zeta(t) - 0,5\zeta(t-1) \quad (2.4)$$

CASO 1 : $n = \ell = 2$, $k = 1$

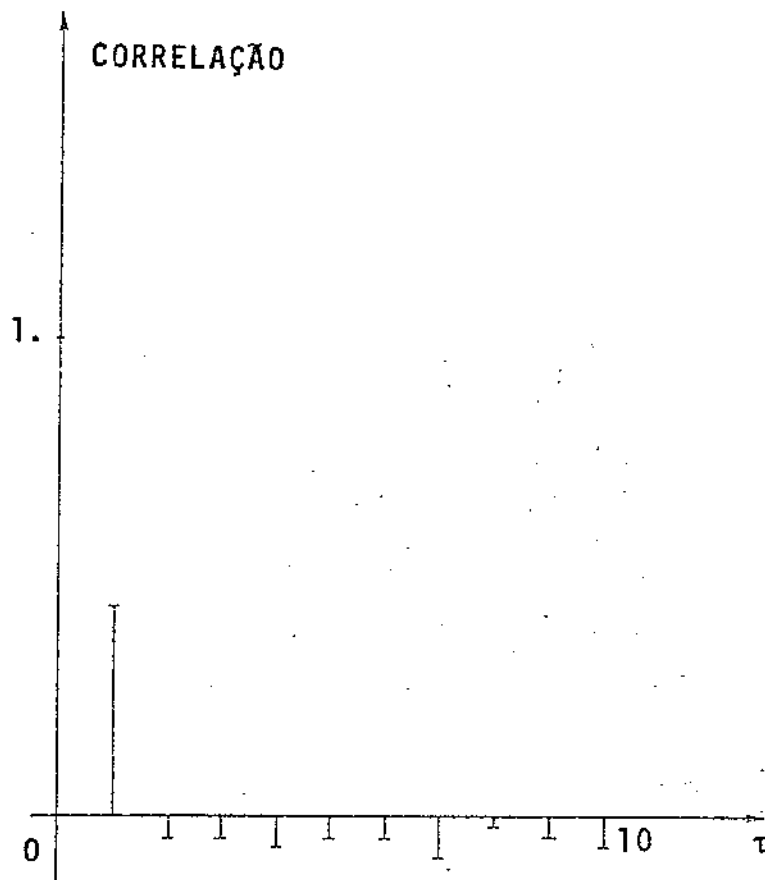


FIGURA 2.5 : Correlação $y(t) y(t + \tau)$

CASO 2 : $n = \ell = 3$, $k = 1$

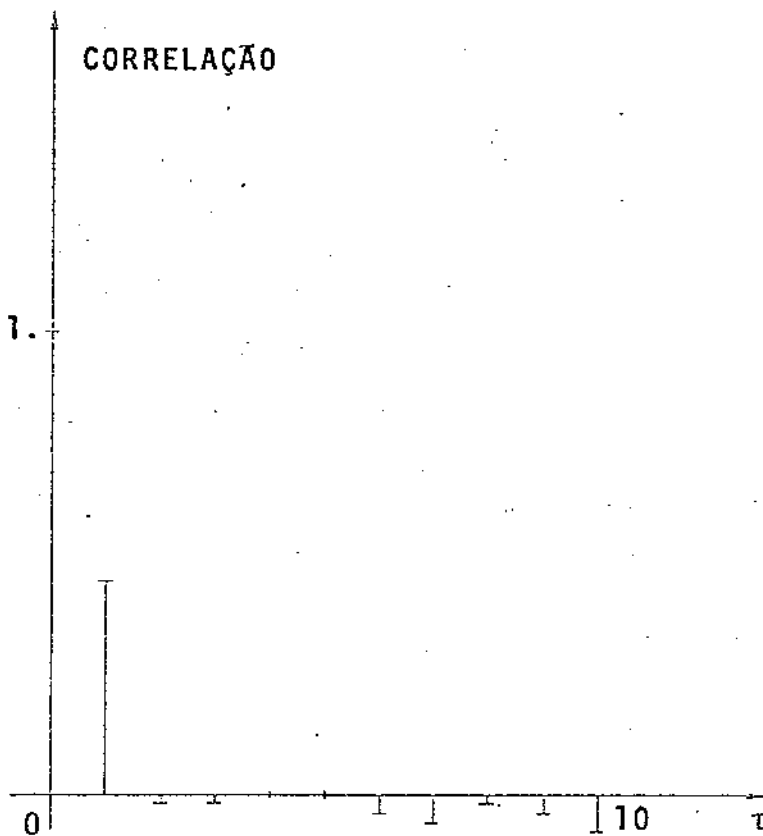


FIGURA 2.6 : Correlação $y(t) y(t + \tau)$

CASO 3 : $n = 2$, $\ell = 3$, $k = 1$

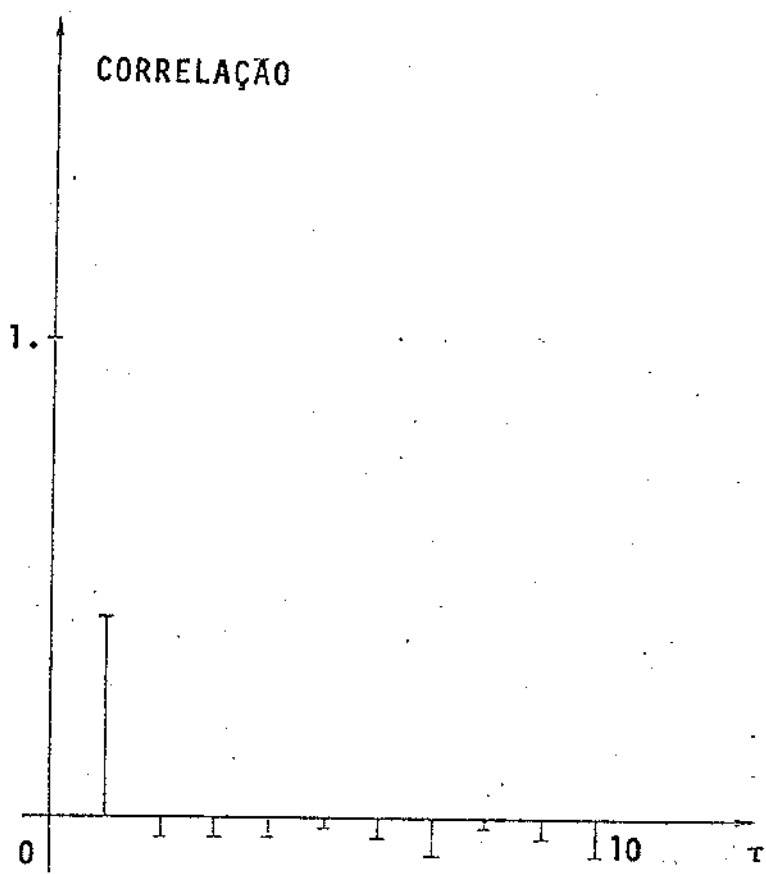


FIGURA 2.7 : Correlação $y(t) y(t + \tau)$

CASO 4 : $n = 1$, $\ell = 2$, $k = 1$

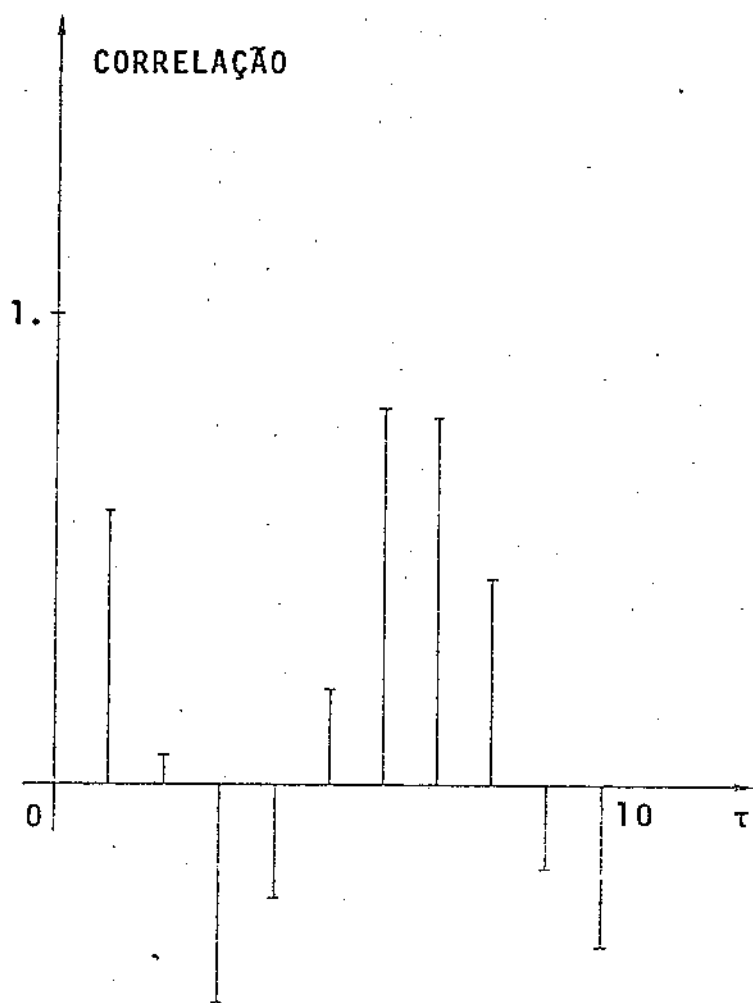


FIGURA 2.8 : Correlação $y(t) y(t + \tau)$

Em seguida apresenta-se uma tabela com os valores das variâncias da saída, para os casos anteriores

Caso	Variância da saída
1	2,98
2	3,01
3	3,05
4	40
regulador de variância mínima	2,79

Tabela 2.3

Na figura 2.5 tem-se a correlação entre os valores $y(t)$ e $y(t + \tau)$, com τ variando de 0 a 10. Essa figura servirá como ponto de referência para as comparações feitas abaixo, pois representa o comportamento do algoritmo RAT para o caso de número de coeficientes conhecido e ilustra os teoremas anteriormente apresentados.

Para o caso 2, o teorema 2 em geral não se aplica, pois há coeficientes em excesso nos polinômios A e B , que assim poderão não mais ser primos quando esses coeficientes convergirem para seus valores finais. Entretanto no exemplo apresentado, até o passo 1000, não havia fatores comuns.

O caso 3, ocorre quando um dos números de coeficiente é correto ($\hat{n}$ ou $\hat{\ell}$) e o outro é tomado como tendo um valor superior ao verdadeiro. É fácil ver que o teorema 2, continua a ser aplicável nesse caso.

Para o caso 4, onde um dos valores, $\hat{n}$ ou $\hat{\ell}$, é conhecido e o outro é inferior ao valor verdadeiro, em geral tem-se um acréscimo no valor da variância de saída, uma vez que estão sendo desprezados coeficientes significativos da equação.

b. Variação no valor do atraso de tempo (k):

As figuras (2.9) e (2.10), demonstram a importância do valor assumido para o atraso no tempo do sistema. A escolha de um valor $\hat{k}$, inferior ao valor verdadeiro k $\bar{e}$, em geral, mais desastroso que a escolha de um valor superior.

Para $\hat{k} < k$, o controle $u(t)$ tentará levar a saída no instante $t + \hat{k} + 1$ para zero. Assim o efeito de $u(t)$ em $y(t + \hat{k} + 1)$ $\bar{e}$ menor, sendo necessários maiores sinais de controle.

Na figura (2.9), assume-se um valor $\hat{k} = 0$, para o exemplo 2.3. Constata-se por essa figura, a má qualidade do controle exercido, expressa por elevados valores de correlação e variância e grande número de limitações no sinal de controle.

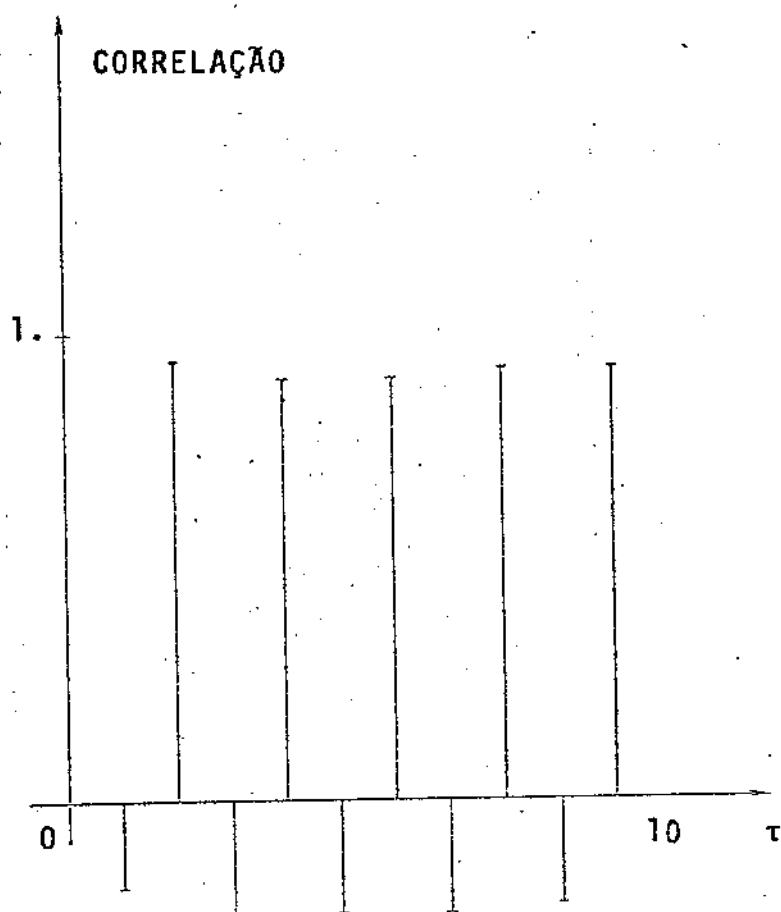


FIGURA 2.9 : correlação $y(t) y(t + \tau)$

Tomar-se $\tilde{k} > k$, não é tão crítico, embora em certos casos seja necessário alterar também o parâmetro β_0 , para se obter convergência. A figura 2.10, mostra o mesmo exemplo 2.3, usando-se $k = 2$.

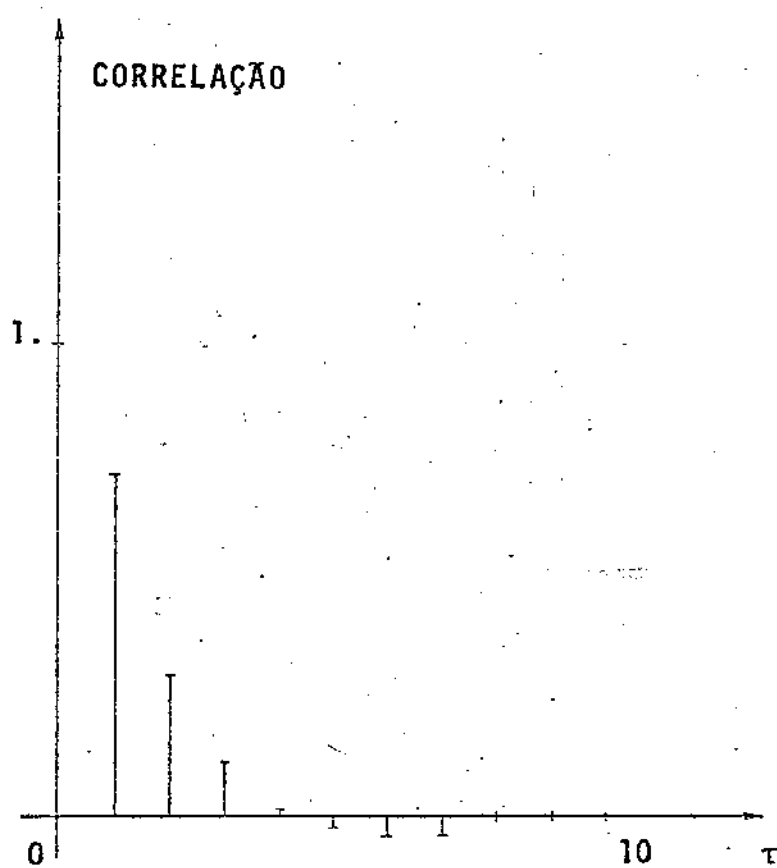


FIGURA 2.10 : correlação $y(t) y(t + \tau)$ $[\beta_0 = 5]$

Para diversos valores de k , com o sinal de controle limitado em $|u(t)| \leq 20$, obtém-se os seguintes valores para a variância da saída:

k	variância da saída	nº de limitações $ u \leq 20$
0	92	557
1	2,98	0
2	3,56	0

Tabela 2.4

2.1.4 Fator de escala do regulador (β_0)

O coeficiente β_0 seria originalmente o valor estimado do coeficiente b_1 da equação (1.2). Entretanto, como a sequência $\{u(t)\}$ não é independente da sequência de ruído $\{z(t)\}$, pode não ser possível determinar todos os coeficientes do modelo. O seguinte exemplo ilustra essa afirmação:

Exemplo 2.4: Seja o processo:

$$y(t) + a y(t - 1) = b u(t - 1) + z(t) \quad (2.5)$$

O controle para o regulador de variância mínima é:

$$u(t) = \frac{a}{b} y(t) \quad (2.6)$$

Se a e b forem desconhecidos, empregam-se então os valores $\hat{a}$ e $\hat{b}$ estimados por mínimos quadrados. Esses coeficientes são obtidos, minimizando:

$$E(a, b) = \sum_{t=1}^N \left\{ y(t+1) + a y(t) - b u(t) \right\}^2 \quad (2.7)$$

Se $u(t) = \alpha y(t)$, tem-se:

$$-\gamma(u(t) - \alpha y(t)) = 0, \quad \forall \gamma \quad (2.8)$$

E adicionando a equação (2.8) em (2.7), tem-se:

$$\begin{aligned} E(a, b) &= \sum (y(t+1) + a y(t) - b u(t) - \gamma u(t) + \gamma \alpha y(t))^2 \\ &= \sum \left[y(t+1) + (a + \gamma \alpha) y(t) - (b + \gamma) u(t) \right]^2 \\ &= E(a + \gamma \alpha, b + \gamma) \end{aligned}$$

$$\text{Então:} \quad E(a, b) = E(a + \gamma \alpha, b + \gamma) \quad (2.9)$$

E a função de custo se mantém constante, para diferentes valores do parâmetro a estimar, não possibilitando calcular a e b. Nesse exemplo o problema é solucionado modificando-se o controle para, por exemplo, $u(t) = \alpha y(t - 1)$.

A seguir, no exemplo 2.5, verifica-se que não é crucial o conhecimento do parâmetro β_0 , quando se conhecem outros parâmetros envolvidos.

Exemplo 2.5: Seja um processo descrito por:

$$y(t) - 0,9y(t - 1) = 0,5u(t - 1) + \zeta(t) \quad (2.10)$$

Seja $n = 1$, $\ell = k = 0$

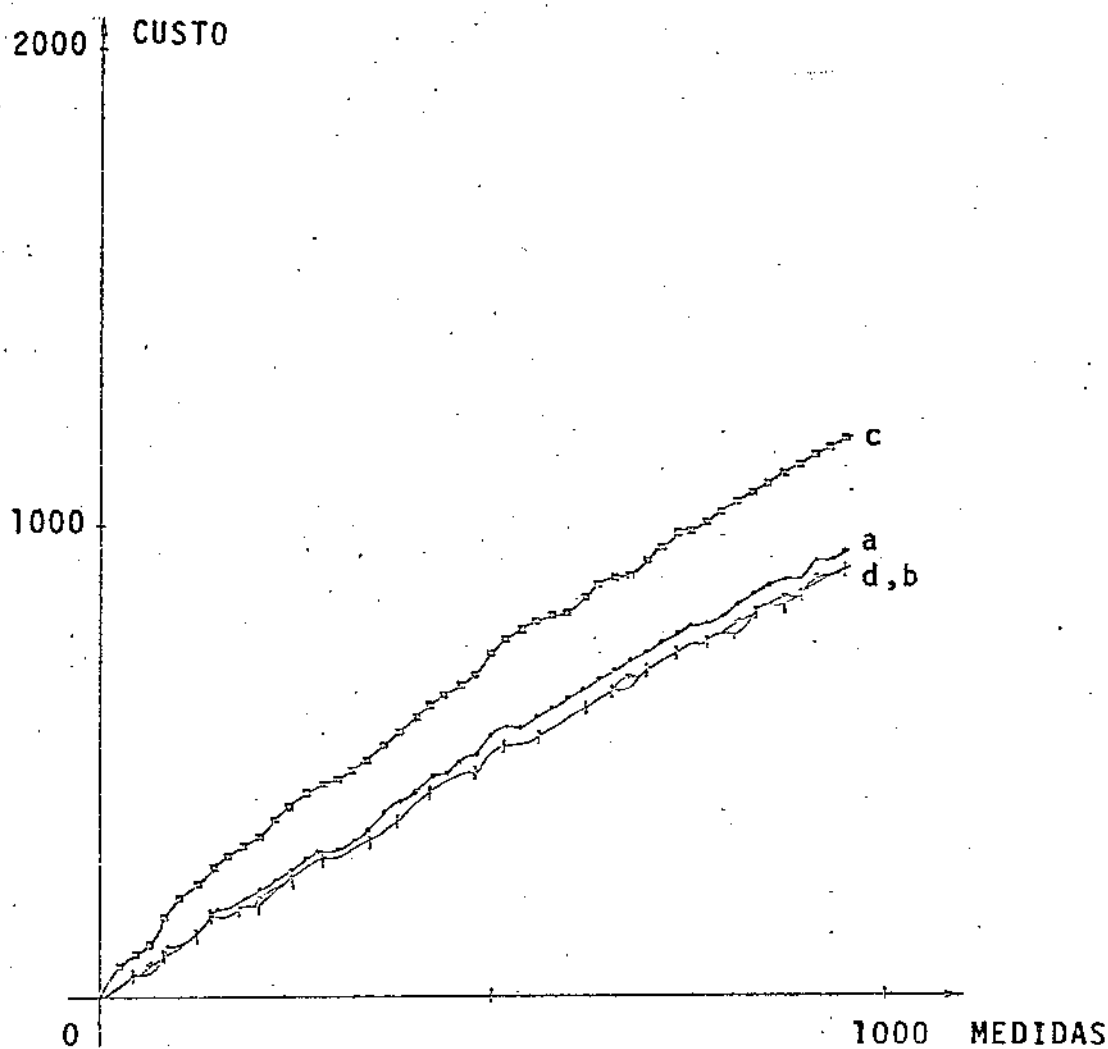


FIGURA 2.11 : Função de custo com $|u(t)| \leq 5$

Fig.	B ₀	Variância	nº limita- ções ocorridos
a	0,05	1,009	20
b	0,5	0,965	7
c	5	1,250	0
d	0,4	0,962	7
=	reg. var. mínima	0,944	=

Tabela 2.5

Pela tabela 2.5, vê-se que o número de limitações ocorridos, no valor do sinal de controle, é um bom índice para a escolha do valor a ser fixado para β_0 .

2.2 - ALGORÍTMO RAT , UTILIZANDO APROXIMAÇÃO ESTOCÁSTICA NA ESTIMAÇÃO

No capítulo 1, chegou-se ao seguinte conjunto de equações para descrever o estimador dos mínimos quadrados do algoritmo RAT básico:

$$\hat{\underline{\theta}}(t+1) = \hat{\underline{\theta}}(t) + \gamma(t) \underline{p}(t) \underline{\phi}^T(t-k-1) \{y(t) - \beta_0 u(t-k-1) - \underline{\phi}(t-k-1) \hat{\underline{\theta}}(t)\} \quad (1.25)$$

$$\gamma(t) = \left(1 + \underline{\phi}(t-k-1) \underline{p}(t) \underline{\phi}^T(t-k-1) \right)^{-1} \quad (1.26)$$

$$\underline{p}^{-1}(t+1) = \underline{p}^{-1}(t) + \underline{\phi}^T(t-k-1) \underline{\phi}(t-k-1) \quad (1.28)$$

$$\underline{\phi}(t-k-1) = \begin{bmatrix} -y(t-k-1) & -y(t-k-2) & \dots & -y(t-k-n) & \vdots & \beta_0 u(t-k-2) & \dots & \beta_0 u(t-k-2-1) \end{bmatrix} \quad (1.18)$$

Assim, a cada novo passo, é necessário atualizar a matriz $\underline{P}$, uma matriz quadrada, simétrica, com $(n + \ell)^2$ elementos.

Uma aproximação possível, que traz grandes simplificações no cálculo dos coeficientes, em troca de um decréscimo na velocidade de convergência, é substituir a matriz $\underline{P}^{-1}(t)$ pelo seu traço [2].

Considerando todos os intervalos discretos de $\emptyset$ a t na equação (1.27), chega-se a:

$$\underline{P}^{-1}(t + 1) = \underline{P}^{-1}(0) + \sum_{i=0}^t \underline{\phi}^T(i - k - 1) \underline{\phi}(i - k - 1) \quad (2.11)$$

Pode-se reescrever a equação (1.24) como:

$$\begin{aligned} \underline{\hat{\theta}}(t + 1) = \underline{\hat{\theta}}(t) + \frac{\underline{P}(t) \cdot \underline{\phi}^T(t - k - 1)}{1 + \underline{\phi}(t - k - 1) \underline{P}(t) \underline{\phi}^T(t - k - 1)} \left\{ y(t) - \right. \\ \left. - \beta_0 u(t - k - 1) - \underline{\phi}(t - k - 1) \underline{\hat{\theta}}(t) \right\} \quad (2.12) \end{aligned}$$

O termo $\frac{\underline{P}(t)}{1 + \underline{\phi}(t - k - 1) \underline{P}(t) \underline{\phi}^T(t - k - 1)}$ pode ser desenvolvido e obtém-se:

$$\frac{\underline{P}(t)}{1 + \underline{\phi}(t - k - 1) \underline{P}(t) \underline{\phi}^T(t - k - 1)} = \frac{1}{\underline{P}^{-1}(t) + \underline{\phi}(t - k - 1) \underline{\phi}^T(t - k - 1)} \quad (2.13)$$

Da equação (2.11) tem-se:

$$\text{traço} \left(\underline{P}^{-1}(t + 1) \right) = \text{TR}^{-1}(t + 1) = \left\{ \sum_{i=0}^t \left(\sum_{j=1}^{n+\ell} \underline{\phi}_j^2(i - k - 1) \right) \right\} \quad (2.14)$$

E a equação (2.13) fica assim:

$$\frac{P(t)}{1 + \phi(t-k-1) P(t) \phi^T(t-k-1)} = \frac{1}{\text{traço}(P^{-1}(t+1))} = TR(t+1) \quad (2.15)$$

Então, o algoritmo de estimação do regulador RAT passa a ser descrito por:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + TR(t+1) \phi^T(t-k-1) \left(y(t) - \beta_0 u(t-k-1) - \phi(t-k-1) \hat{\theta}(t) \right) \quad (2.16)$$

$$TR(t+1) = \left\{ \sum_{i=0}^t \left(\sum_{j=1}^{n+l} \phi_j^2(i-k-1) \right) \right\}^{-1} \quad (2.17)$$

ou ainda:

$$TR(t+1) = \left\{ \text{RHO} * TR(t) + \sum_{j=1}^{n+l} \phi_j^2(t-k-1) \right\}^{-1} \quad (2.18)$$

Onde a obtenção da equação (2.18) é feita heurísticamente, aplicando-se um fator de ponderação nos instantes passados, de forma a considerar a última informação como a melhor.

É bom reprimir, que a simplificação transformando a matriz $P(t)$ num escalar, resulta numa convergência mais lenta, exigindo maior número de iterações, embora cada iteração consuma menor tempo de computação.

Um estudo da estabilidade desse algoritmo é apresentada por Ljung [2], associando ao algoritmo estocástico uma equação diferencial e analisando sua estabilidade.

2.3 - ALGORÍTMO RAT, UTILIZANDO O MÉTODO DA RAIZ QUADRADA NA ESTIMAÇÃO

Anteriormente, chegou-se ao seguinte conjunto de equações para descrever o estimador do RAT :

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \gamma(t) P(t) \phi^T(t-k-1) \left(y(t) - \beta_0 u(t-k-1) - \phi(t-k-1) \hat{\theta}(t) \right) \quad (1.18)$$

$$\underline{P}(t+1) = \frac{1}{\text{RHO}} \left(\underline{P}(t) - \gamma(t) \underline{P}(t) \underline{\phi}^T(t-k-1) \underline{\phi}(t-k-1) \underline{P}(t) \right) \quad (1.29)$$

$$\gamma(t) = \left(1 + \underline{\phi}(t-k-1) \underline{P}(t) \underline{\phi}^T(t-k-1) \right)^{-1} \quad (1.30)$$

Reescrevendo as equações (1.28), (1.29) e (1.30), obtêm-se:

$$\hat{\underline{\theta}}(t+1) = \hat{\underline{\theta}}(t) + \underline{K}(t) \left(y(t) - \beta_0 u(t-k-1) - \underline{\phi}(t-k-1) \hat{\underline{\theta}}(t) \right) \quad (2.19)$$

$$\underline{P}(t+1) = \frac{1}{\text{RHO}} \left(\text{I} - \underline{K}(t) \underline{\phi}(t-k-1) \right) \underline{P}(t) \quad (2.20)$$

$$\underline{K}(t) = \frac{\underline{P}(t) \underline{\phi}^T(t-k-1)}{\text{RHO} + \underline{\phi}(t-k-1) \underline{P}(t) \underline{\phi}^T(t-k-1)} \quad (2.21)$$

No caso escalar, é fácil ver que se na equação (2.21), o elemento $\underline{P}(t) \underline{\phi}(t-k-1)$ crescer muito, então a expressão $1 - \underline{K}(t) \underline{\phi}(t-k-1)$ tenderá para zero e não mais será corrigido, o valor do parâmetro estimado. Para tentar solucionar esse erro de precisão, no cálculo do valor de $\underline{P}(t)$, é que foi introduzido o método da raiz quadrada, no algoritmo de estimação, já que esse método dobra a precisão dos cálculos [4]-[5] e garante que a matriz $\underline{P}(t)$ permanece semi-definida positiva.

Assim a matriz $\underline{P}$ é substituída por seu valor equivalente $\underline{P} = \underline{S}\underline{S}^T$, onde $\underline{S}$ é uma matriz triangular superior.

A seguir, apresentam-se as equações de algoritmo obtido por Clarke [3], no desenvolvimento do RAT, usando o método da raiz quadrada para estimação.

$$\hat{\underline{\theta}}(t+1) = \hat{\underline{\theta}}(t) + \underline{K}(t) \left(y(t) - \beta_0 u(t-k-1) - \underline{\phi}(t-k-1) \hat{\underline{\theta}}(t) \right) \quad (2.22)$$

$$[K(t)]_i = \frac{g_i^N}{u_N^2} \quad (2.23)$$

$$\mu_0 = \sqrt{\beta} \quad (2.24)$$

$$\mu_i = \sqrt{\beta + f_i^2}$$

$$g_i^N = \begin{cases} \sum_{k=1}^N [S(t)]_{ik} f_k & , \quad 1 \leq i \leq m \\ 0 & , \text{ nos outros casos} \end{cases} \quad (2.25)$$

$$[S(t+1)]_{ij} = \frac{1}{\sqrt{\beta}} \cdot \frac{u_{j-1}}{u_j} \left\{ [S(t)]_{ij} - \frac{f_i g_i^{j-1}}{u_{j-1}^2} \right\} \quad (2.26)$$

$$\underline{f} = \underline{S}^T \underline{\phi}(t) \quad (2.27)$$

2.4 - SIMULAÇÃO DOS ALGORÍTMOS RAT

Nesse parágrafo, ilustram-se as três versões do algoritmo RAT, através de um exemplo, utilizando um sistema de segunda ordem.

2.4.1 Algoritmo RAT básico

Exemplo 2.6: Seja um processo descrito por:

$$y(t) - 1,6y(t-1) + 0,8y(t-2) = u(t-2) + 0,5u(t-3) + \zeta(t) \quad (2.28)$$

Então : $n = l = 2$
 $k = 1$

O regulador de variância mínima para esse processo tem a forma:

$$u(t) = -1,76y(t) + 1,28y(t - 1) - 2,1u(t - 1) - 0,8u(t - 2) \quad (2.29)$$

E a variância, teórica, de saída é: 3,36 .

As figuras 2.12 a 2.17 fornecem os resultados obtidos com o algoritmo básico.

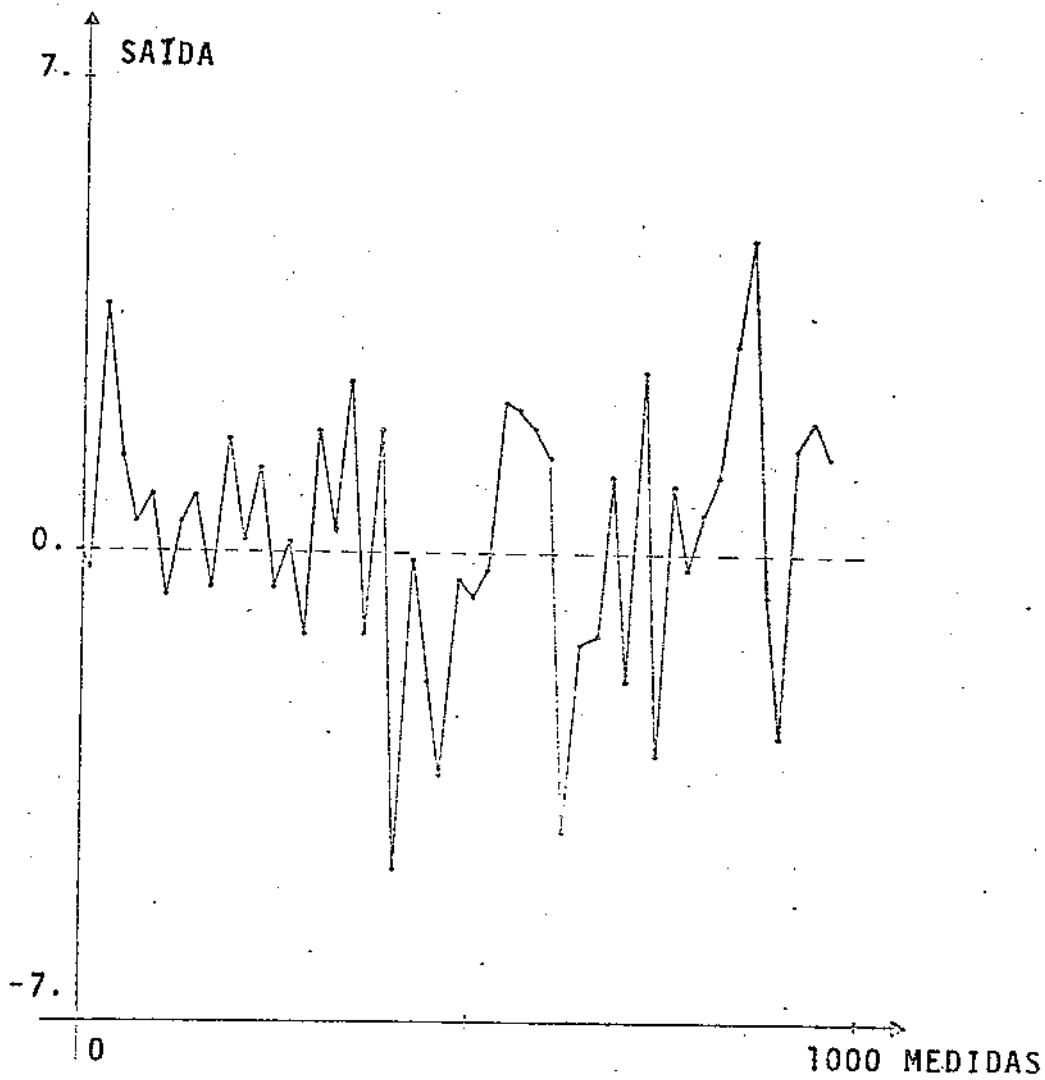


FIGURA 2.12 : Saída do processo

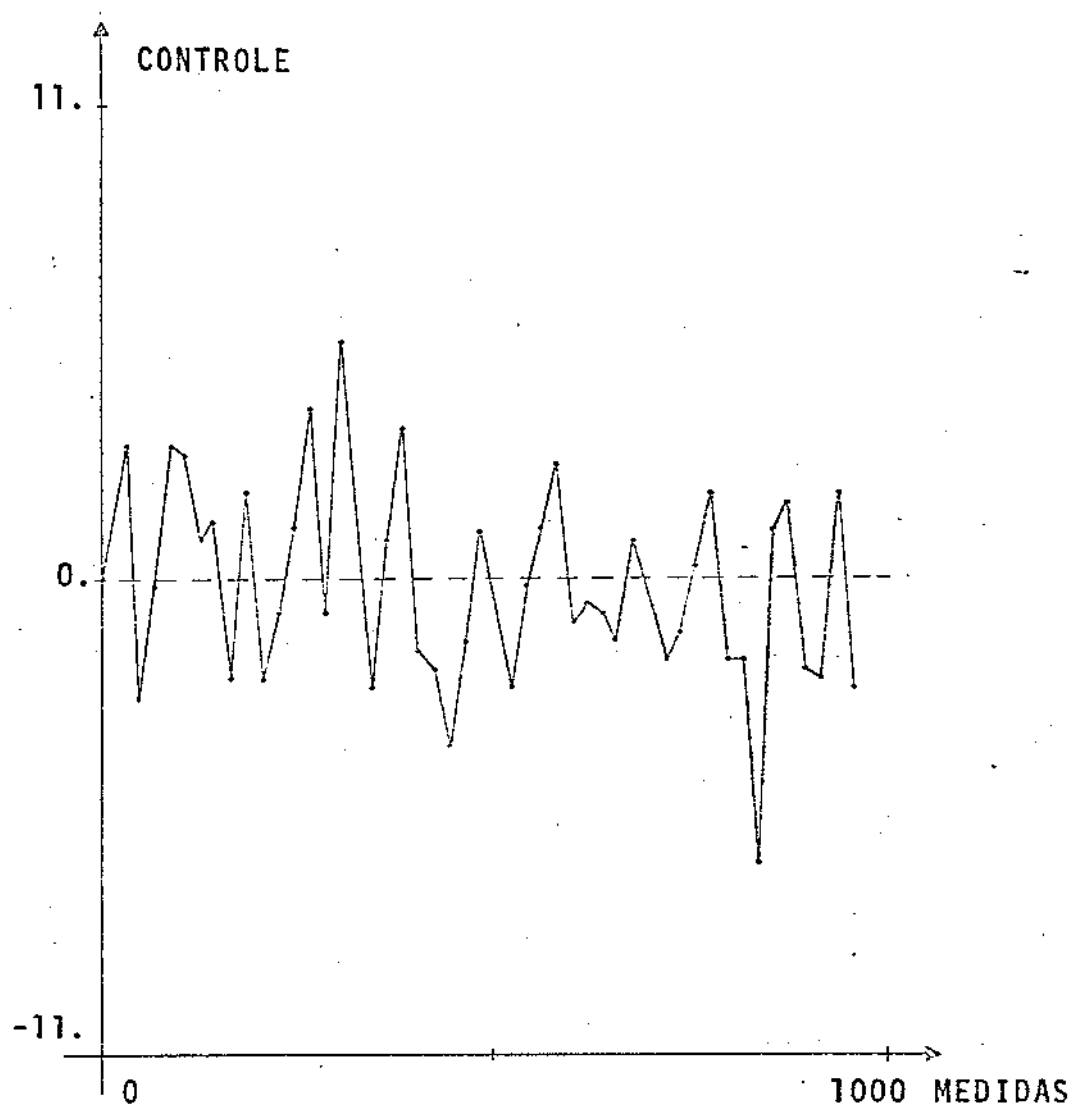


FIGURA 2.13 : Controle

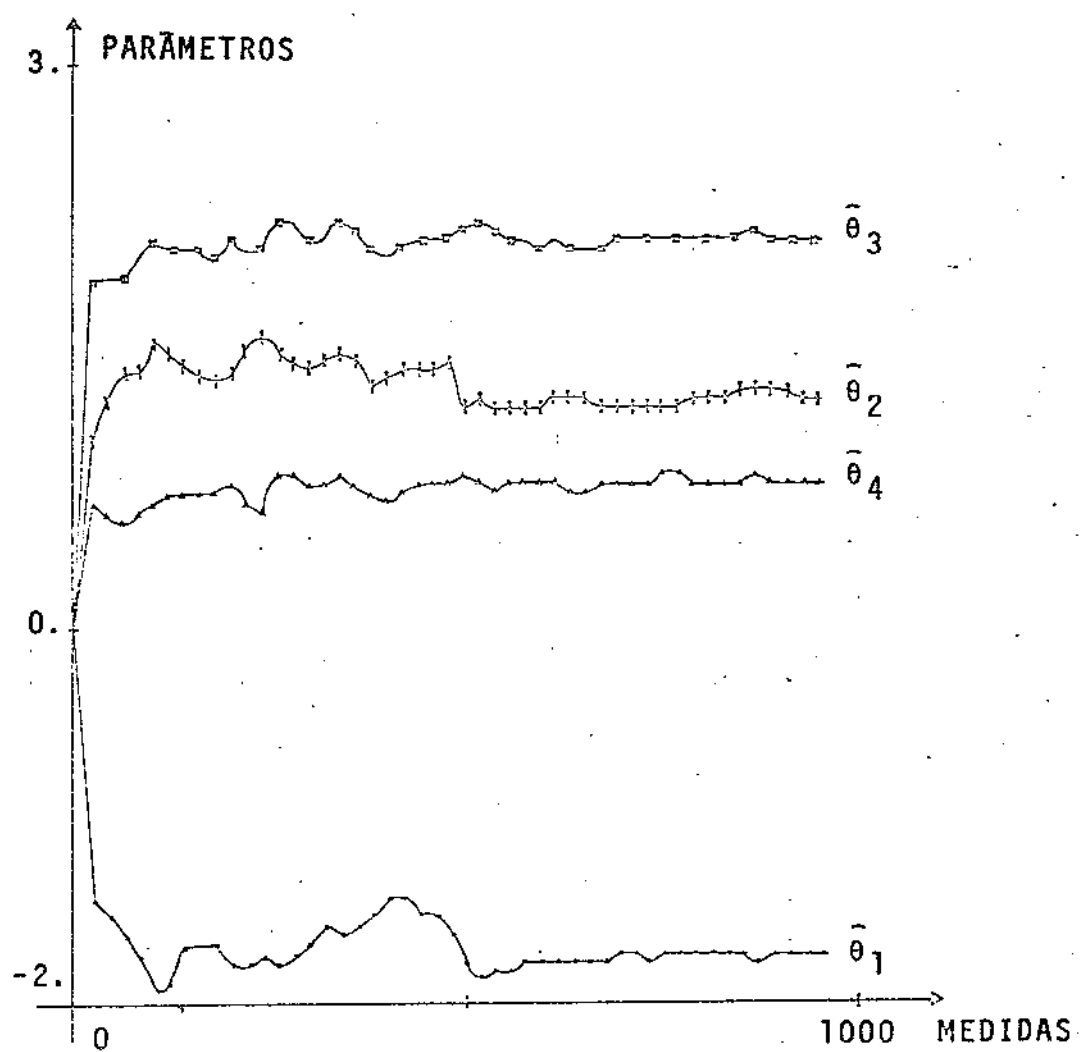


FIGURA 2.14 : Coeficientes estimados

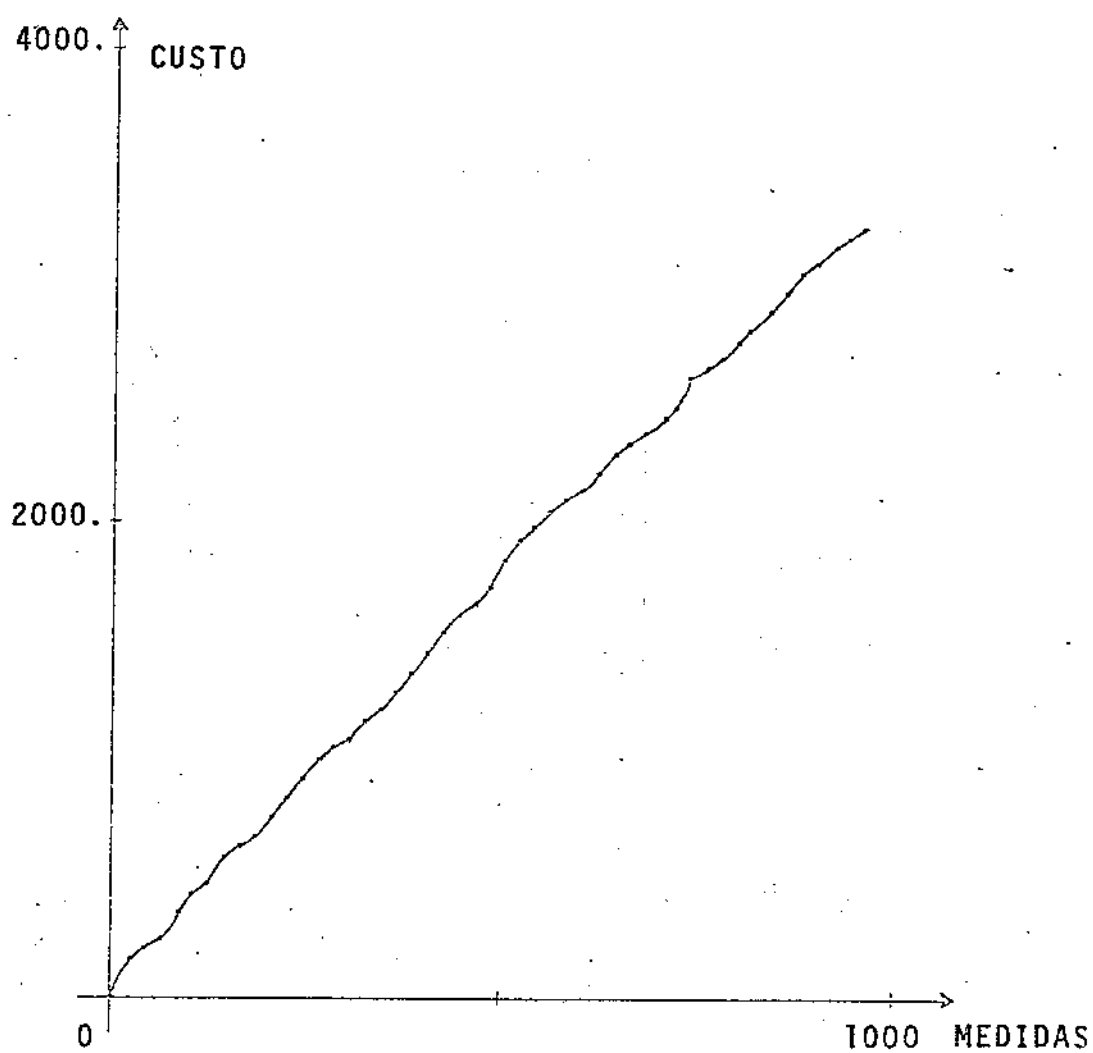


FIGURA 2.15 : Função de custo

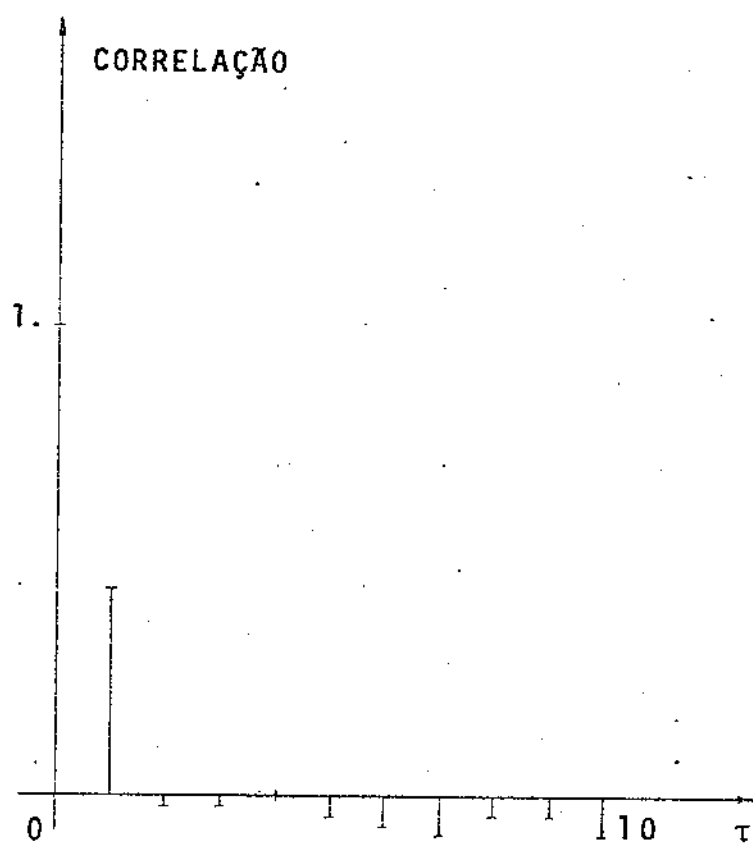


FIGURA 2.16 : Correlação $y(t) y(t + \tau)$

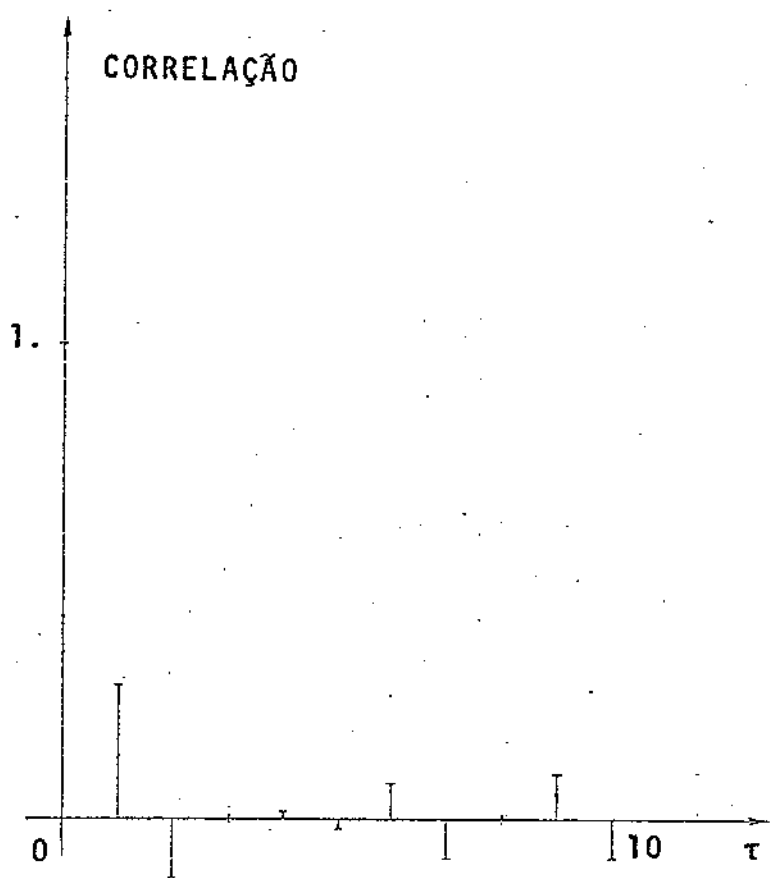


FIGURA 2.17 : Correlação $y(t)$ $u(t + \tau)$

Para este exemplo foi tomado o valor 0,986 para $RH0$, até o passo 500 e o controle foi limitado a $-10 \leq u(t) \leq 10$. Os coeficientes foram inicializados em 0 e foram assumidos valores corretos de n , l e k .

A variância obtida foi 3,47 ($\text{var}_{\text{teórico}} = 3,36$) e ocorreram 3 limitações no sinal de controle.

2.4.2 Algoritmo RAT utilizando aproximação estocástica na estimação

Aplicando-se o algoritmo apresentado no item 2.2, ao exemplo 2.6, utilizando os valores corretos para n , l e k e com uma inicialização para os valores dos parâmetros pressupondo algum conhecimento a priori, obtêm-se :

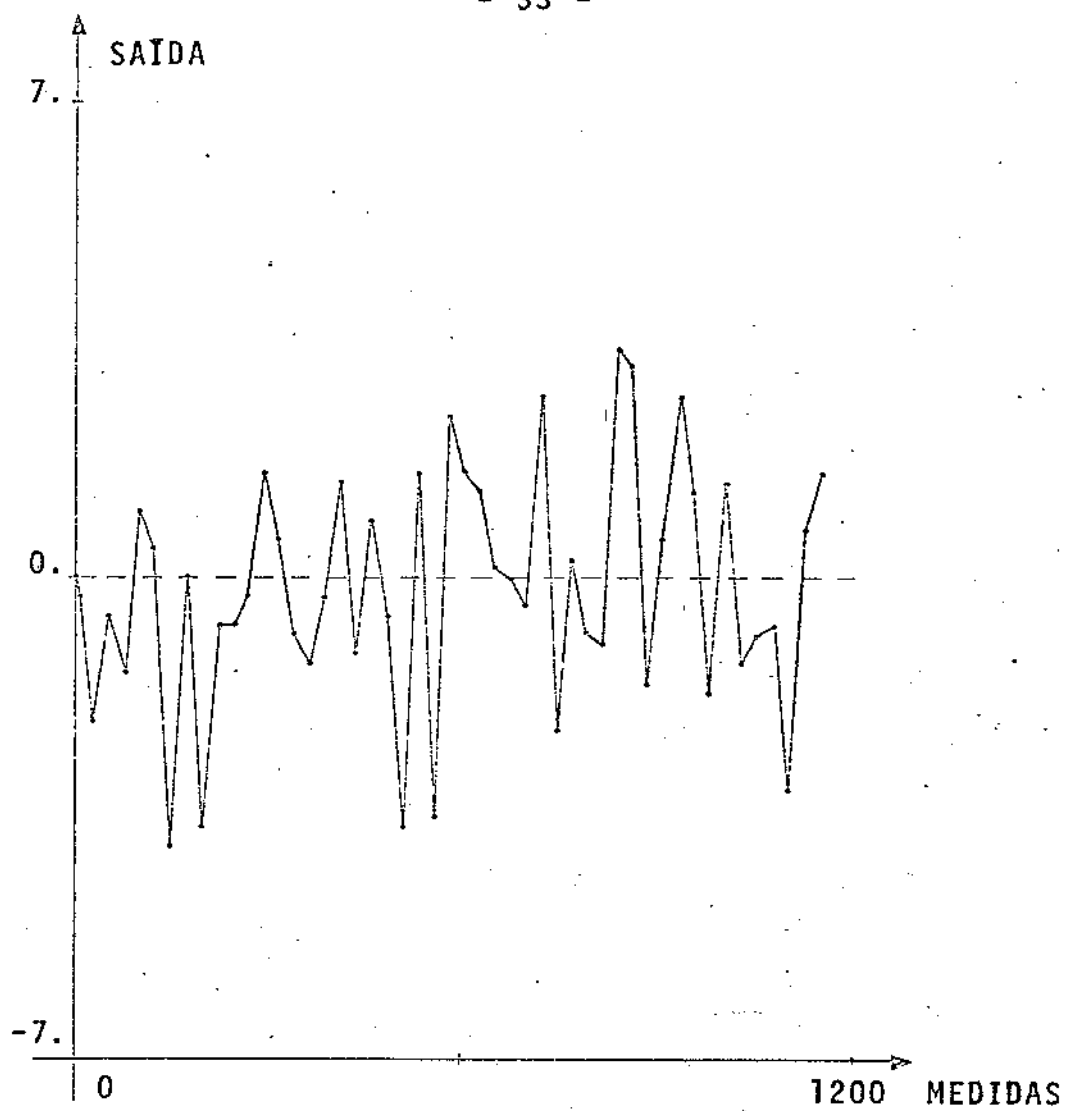


FIGURA 2.18 : Saída do processo

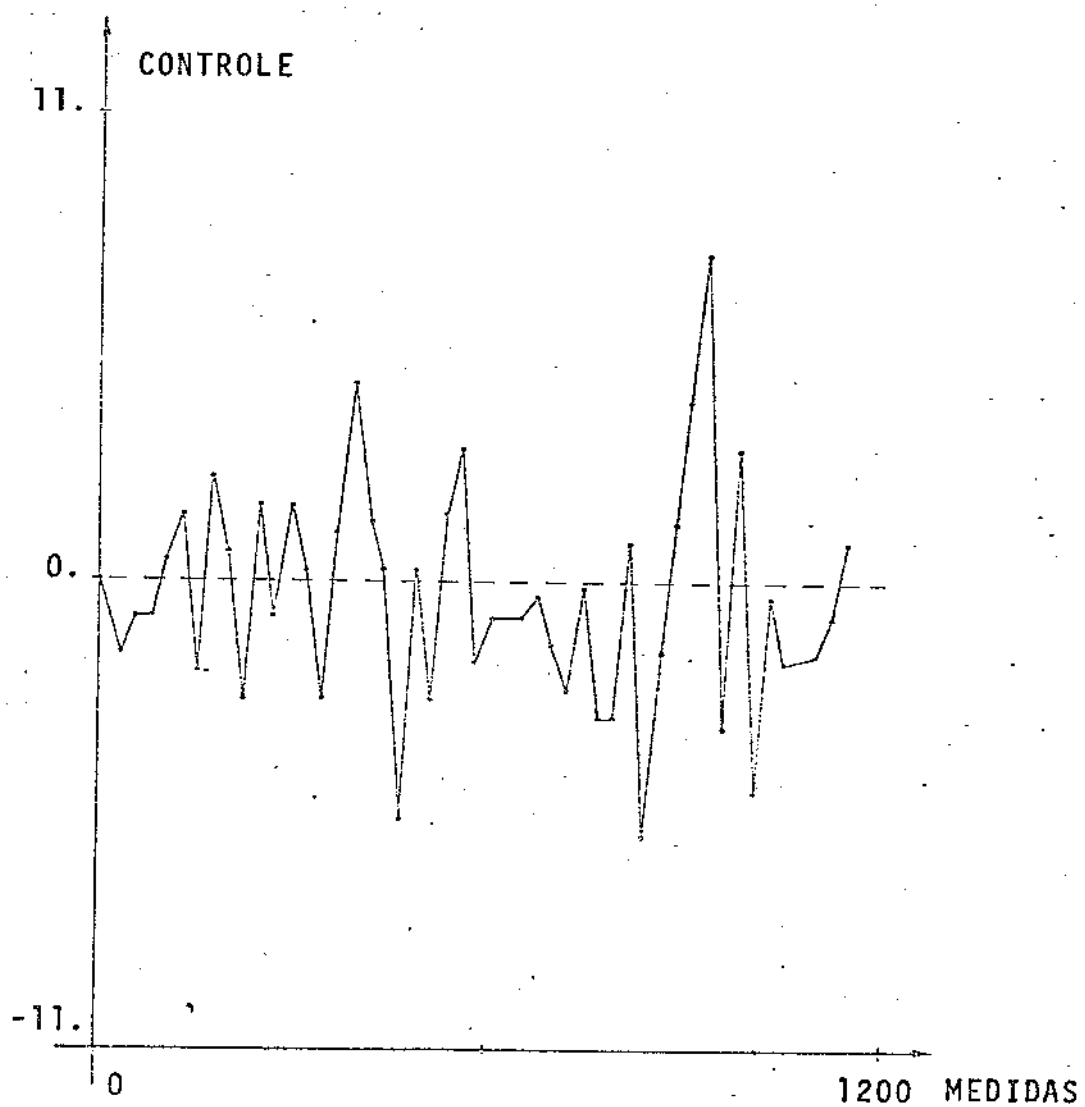


FIGURA 2.19 : Controle

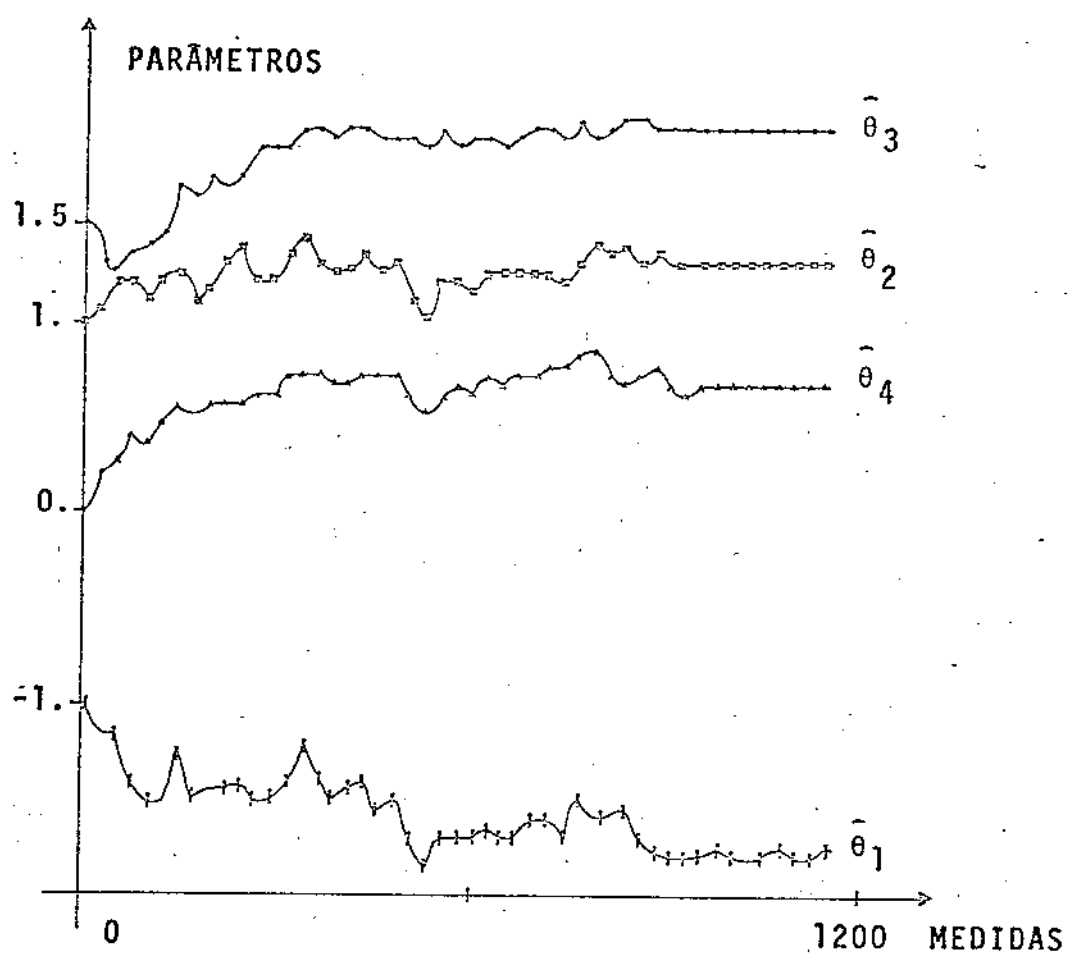


FIGURA 2.20 : Coeficientes estimados

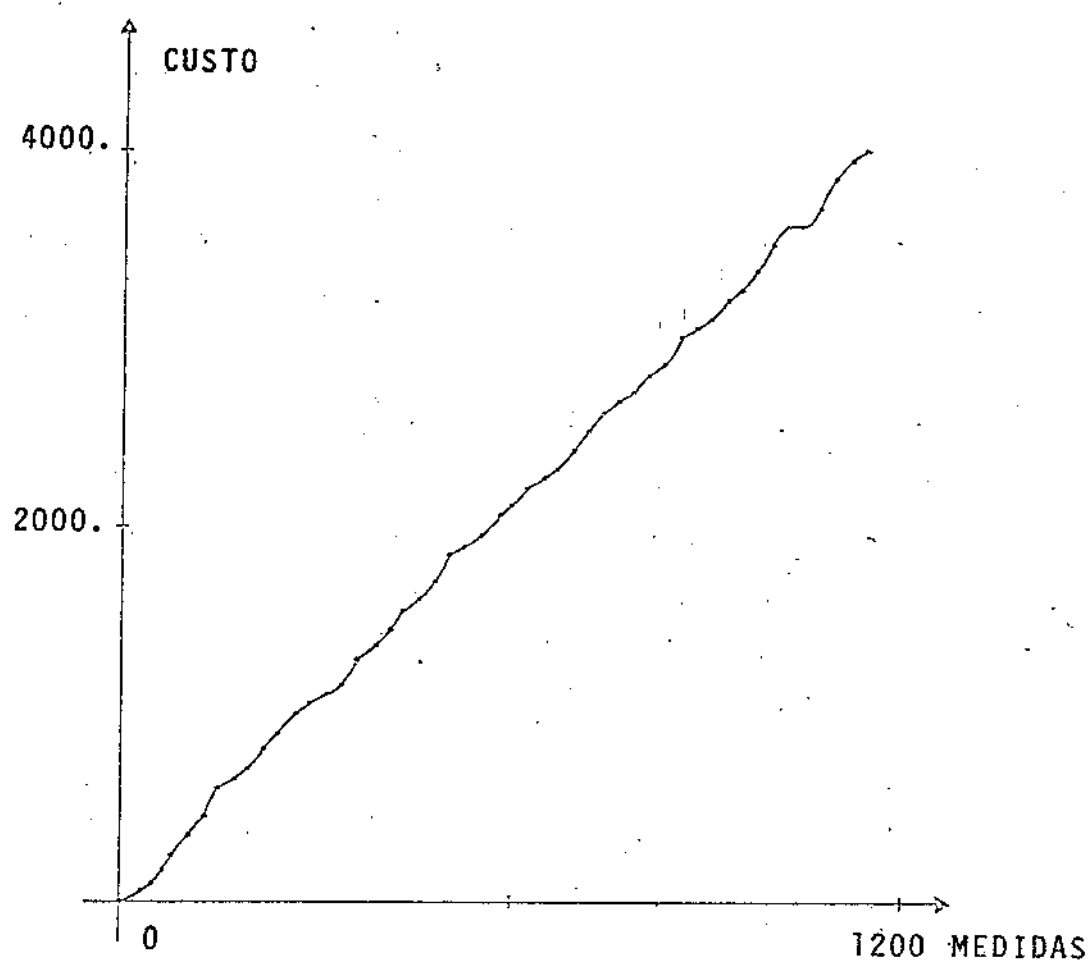


FIGURA 2.21 : Função de custo

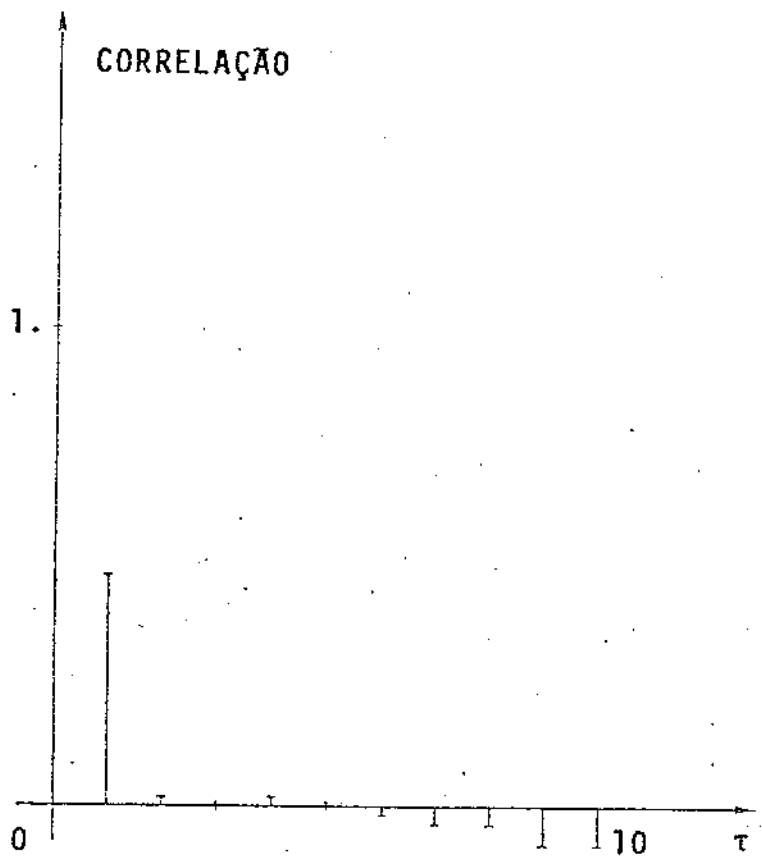


FIGURA 2.22 : Correlação $y(t) y(t + \tau)$

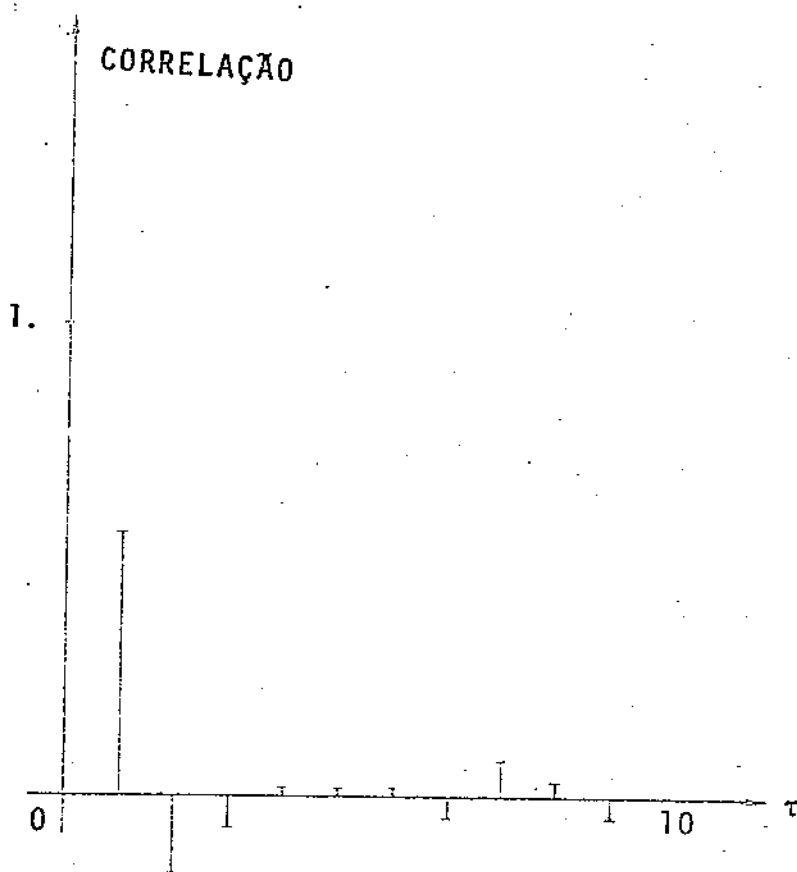


FIGURA 2.23 : Correlação $y(t) u(t + \tau)$

COEFICIENTES

Inicialização	Valor Correto
-1,0	-1,76
1,0	1,28
1,5	2,1
0,0	0,8

Tabela 2.5

Para este exemplo foi tomado o valor de 0,9 para ρ até o passo 900, dentre os 1200 passos da simulação e o controle foi limitada a $-10 \leq u(t) \leq 10$, não tendo ocorrida nenhuma limitação durante a simulação.

A variância obtida foi 3,56 ($\text{var}_{\text{teórica}} = 3,36$).

Se a inicialização fosse a mesma utilizada no item 2.4.1, o controlador obtido teria uma variância da saída de 4,19.

Comparando-se o controlador usando aproximação estocástica com o algoritmo RAT básico, vê-se que dependendo da inicialização o comportamento obtido com o RAT básico é também obtido com o algoritmo de aproximação estocástica, com a vantagem de usar um tempo muito menor em cada iteração.

2.4.3 Algoritmo RAT, utilizando o método da raiz quadrada na estimação:

Aplicando-se o algoritmo apresentado no item 2.3 ao exemplo 2.6, utilizando valores corretos para n , ℓ e k obtiveram-se os seguintes resultados:

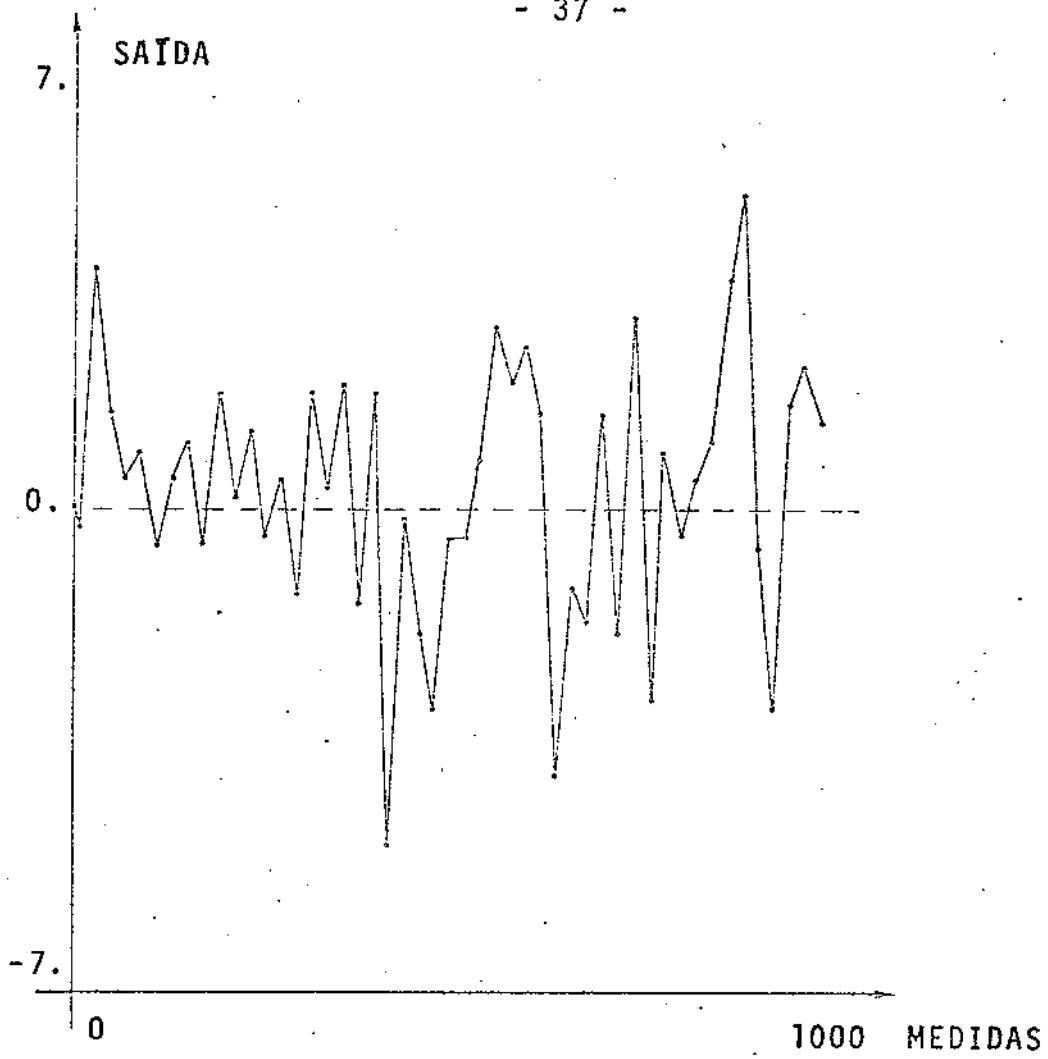


FIGURA 2.24 : Saída do processo

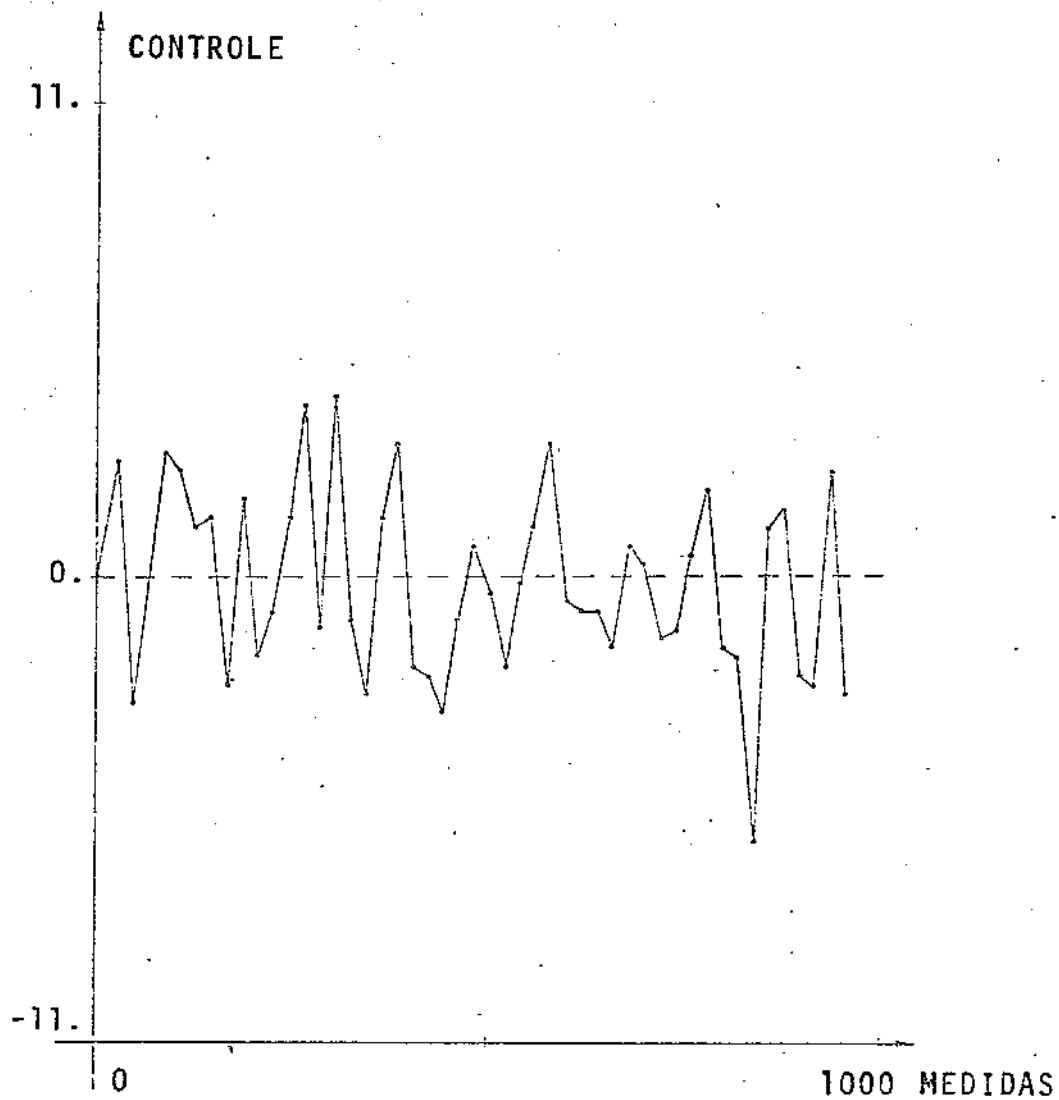


FIGURA 2.25 : Controle

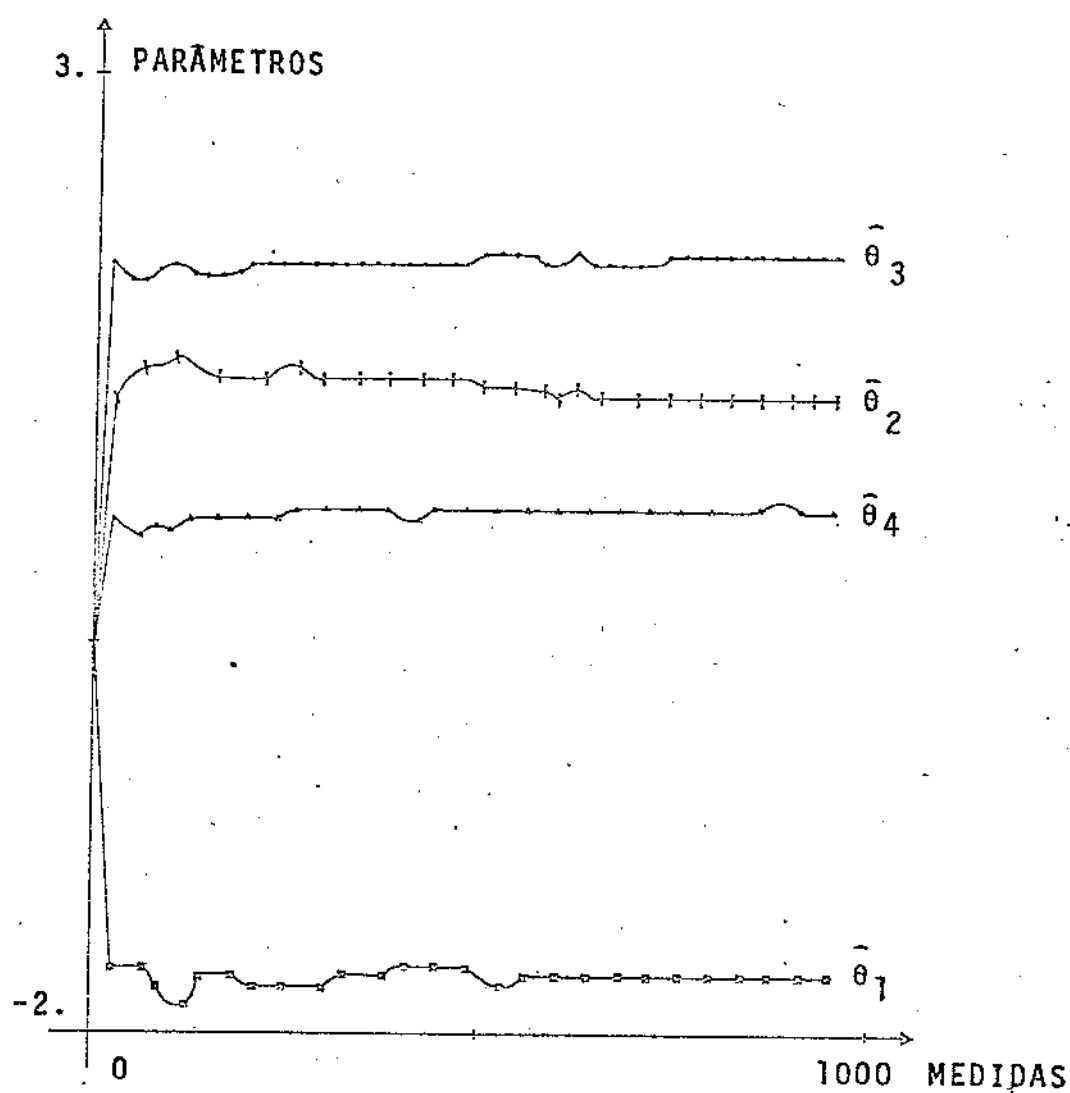


FIGURA 2.26 : Coeficientes estimados

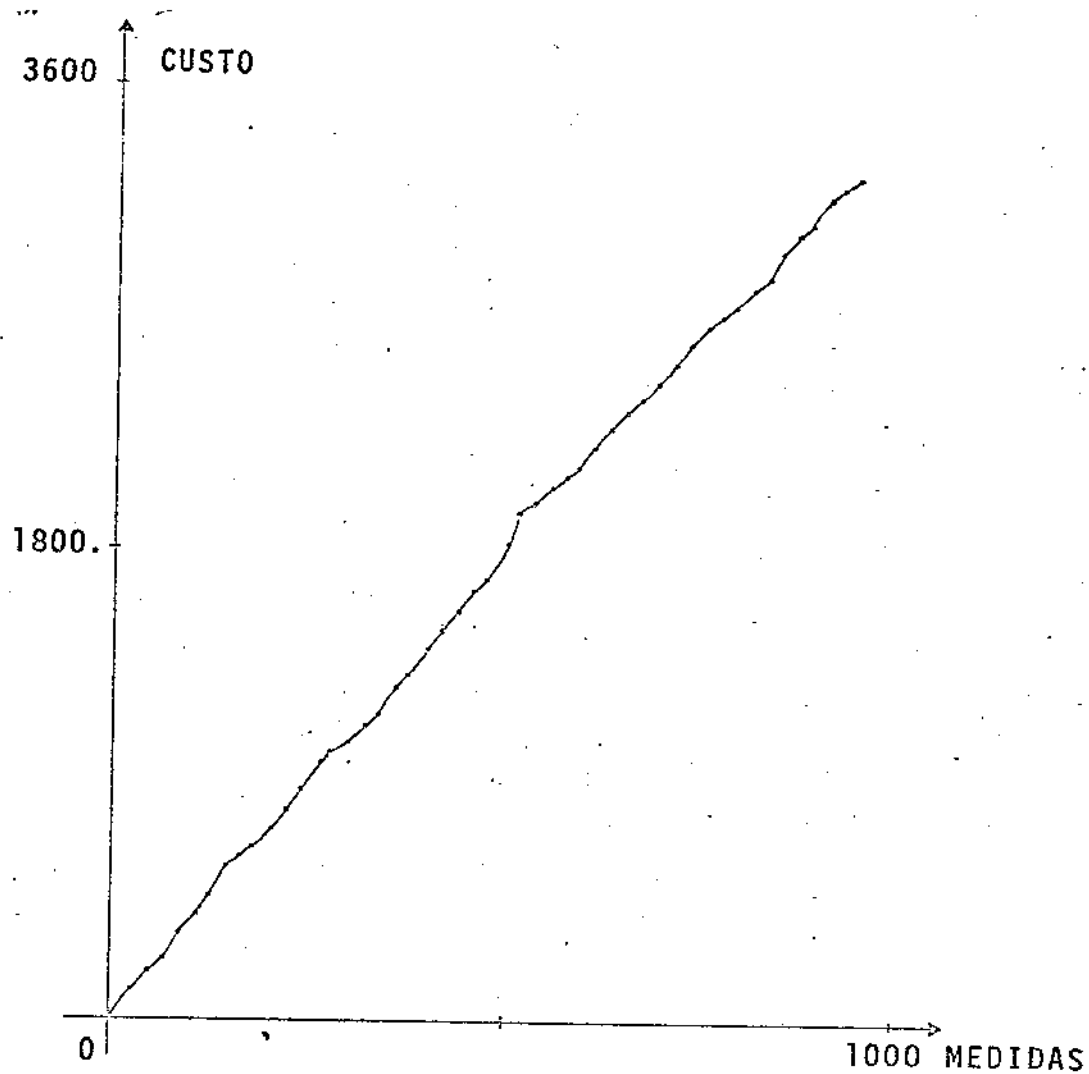


FIGURA 2.27 : Função de custo

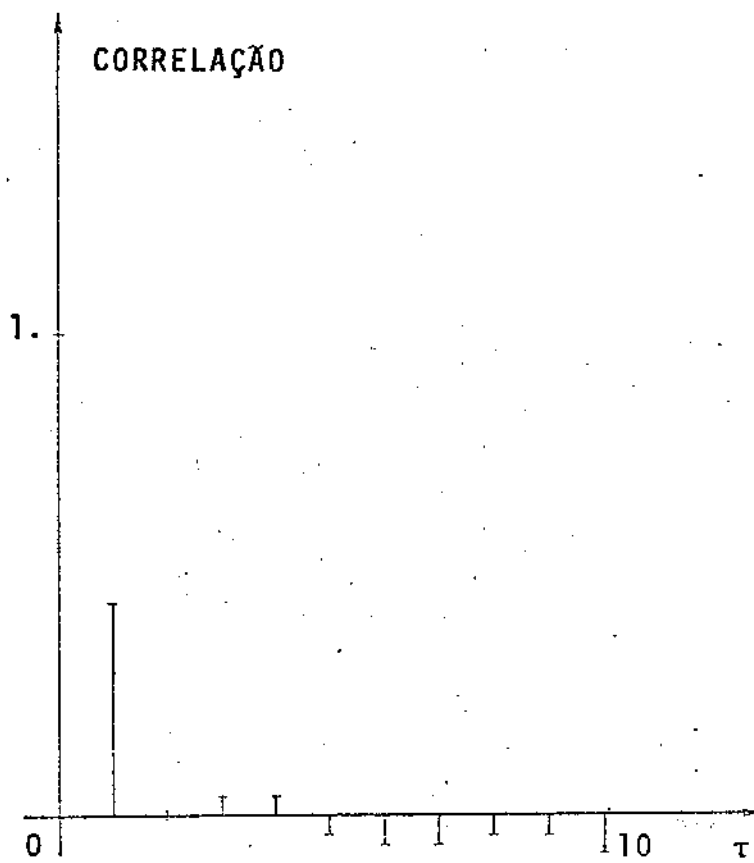


FIGURA 2.28 : Correlação $y(t) y(t + \tau)$

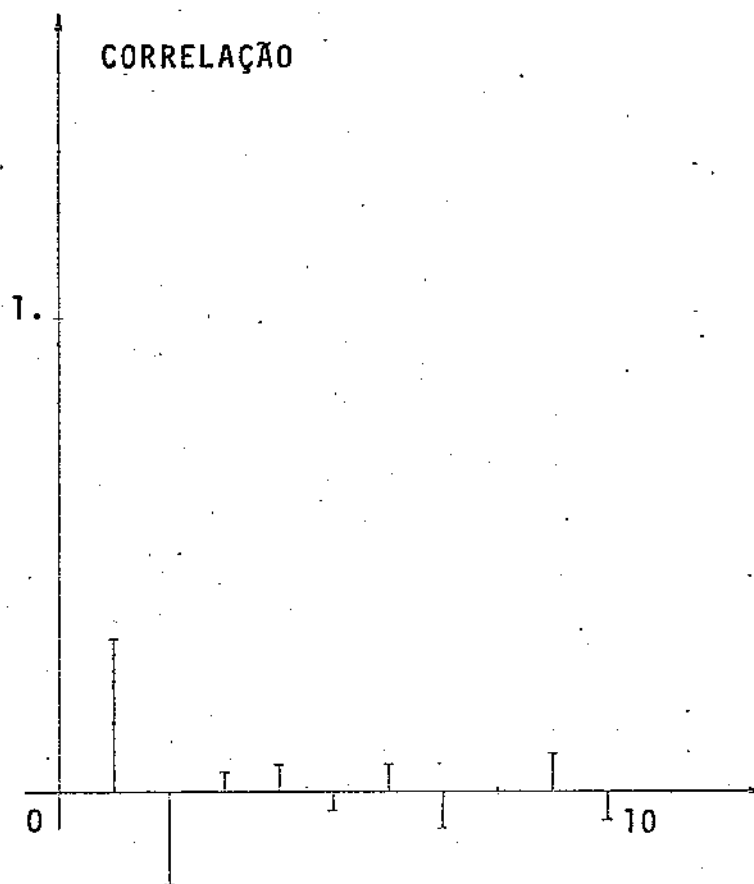


FIGURA 2.29 : Correlação $y(t) u(t + \tau)$

Os valores iniciais dados aos coeficientes foi zero, o valor tomado para RHO foi 1. O sinal de controle foi limitado em $-10 \leq u(t) \leq 10$, tendo ocorrido 3 limitações durante a simulação. A variância obtida foi 3,46.

2.4.4 Conclusões

Como era esperado, os resultados dos itens 2.4.1 e 2.4.3 são comparáveis em todos os sentidos, desde que a fatoração efetuada na matriz P , não acarreta nenhuma mudança no algoritmo, só cooperando na melhora da precisão das operações a efetuar.

É interessante destacar a relação entre conhecimento a priori do sistema e a aplicação do algoritmo do item 2.4.2, que por exigir menor esforço computacional, é de interesse em aplicações "on-line". Uma idéia, será utilizar um algoritmo mais complexo enquanto o sistema ainda estiver em regime transitório e após este atingir o regime permanente, passar a utilizar um controlador mais simples, por exemplo o RAT com aproximação estocástica.

Na implementação presente, feita num PDP-10, os tempos de execução por passo para o algoritmo básico e o que usa raiz quadrada se equivaleram, enquanto que o algoritmo usando a aproximação estocástica consumiu a metade do tempo por passo.

2.5 - BIBLIOGRAFIA

1. Wittenmark ,B. - "Stochastic adaptive control methods : a survey".Int. Journal Control, 21(5) : 705-730,1975.
2. Ljung ,L.; Wittenmark ,B. - "Asymptotic Properties of self-tuning regulators".Lund Institute of Technology ,Dep. Automatic Control, 1974.(Report 7404, fevereiro 1974).
3. Clarke ,D.W.;Cope ,S.N.; Gawthrop ,P.J. - "Feasibility study of the application of micro-processors to self-tuning controllers".Oxford.(Report 1137, 1975).
4. Peterka ,V. - "A square-root filter for real-time multi-variable regression".Kybernetika ,11: 53-67,1975.
5. Kaminsky ,P.G.; Bryson ,Jr.,A.E.; Schmidt ,S.F. - "Discrete Square-Root Filtering : A survey of current techniques".IEEE Transactions on Automatic Control ,16(6) : 727-736,dezembro 1971.

CAPÍTULO 3

IMPLEMENTAÇÃO DO CONTROLADOR RAT NUM MINI-COMPUTADOR

A seguir será apresentada a simulação do controle de um processo, através do algoritmo RAT implementado no mini-computador PDP-11/05 . O processo, simulado, estará em desenvolvimento em um PDP-10 , ao qual está ligado o PDP-11/05 em "time-sharing".

A implementação do processo num computador central e o controlador num mini-computador, possibilita ter-se um controlador operando em situação próxima da real, sem no entanto perder as vantagens de se operar com um simulador desenvolvido exclusivamente num computador de grande porte.

No decorrer desse capítulo, será desenvolvido um conjunto de rotinas para operações aritméticas no mini-computador, bem como rotinas para a comunicação entre o computador central e o controlador.

3.1 - CARACTERIZAÇÃO DO PDP-11/05

O mini-computador PDP-11/05 , é produzido pela "Digital Equipment Corporation" e foi introduzido no mercado por volta de 1970.

O modelo aqui utilizado, possui 16k - bytes e permite processar operandos de um ou dois bytes.

Possui esquematicamente a seguinte estrutura:

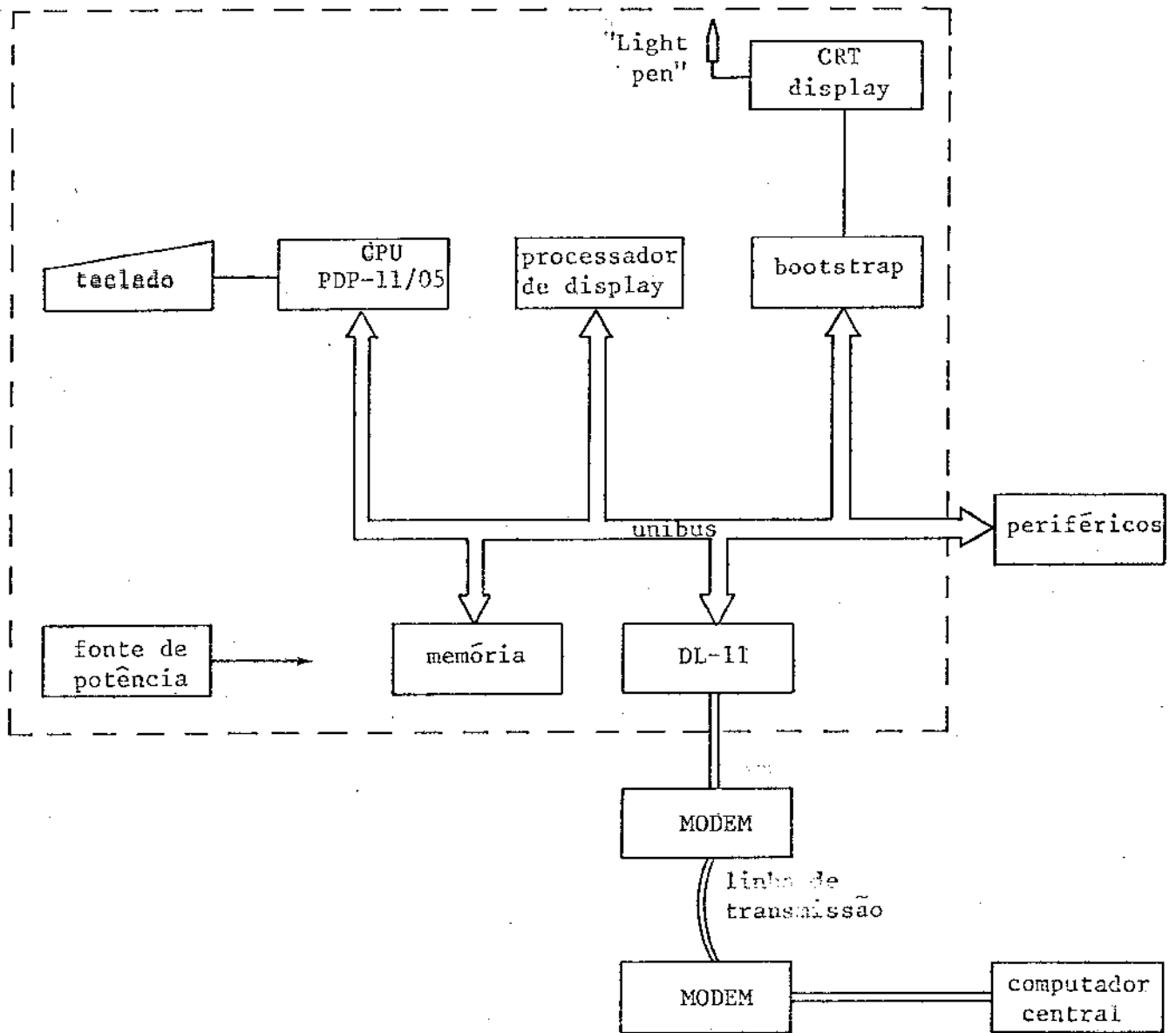


FIGURA 3.1

Essa estrutura que considera os elementos formadores do mini-computador como periféricos associados por um "BUS", permite entre outras coisas uma padronização nos acessos a quaisquer dados, dispensando instruções especiais para cada periférico.

Para auxílio nas operações, existem 8 registros aritméticos de 16 bits, numerados de 0 a 7:

- 0 → 5 : registros para uso geral
- 6 : registro usado como "stack pointer"
- 7 : registro usado como "program counter" (PC)

No PDP-11/05, são permitidos "interrupts", "subroutines" e instruções "trap". Para o processamento dessas, é utilizado o conceito de "stack" (pilha), para o armazenamento da "PSW" antiga e do endereço de retorno ou do conteúdo de um registro, conforme a instrução em andamento.

Cada periférico conectado ao "BUS", tem uma prioridade associada. As rotinas, de serviço, ligadas a cada periférico, tem uma outra prioridade, que é fornecida quando da definição da "PSW". Essa prioridade, varia em ordem crescente de 0 a 7 e é usada para, comparada com a prioridade do programa principal, permitir ou não uma "interrupt" requerida.

Para o tratamento dos operandos envolvidos num programa, são definidos tres modos de endereçamento, conforme a seguir é descrito:

a) Modo direto

Código	Nome	Sintaxe	Descrição
000	registro	RN	registro RN contém o operando
010	auto-incremento	(RN)+	registro RN contém endereço do operando. Registro é incrementado de dois, após a referência.
100	auto-decremento	-(RN)	registro RN é decrementado de dois e após contém o endereço do operando.
110	índice	X(RN)	valor X, armazenado na palavra seguinte à instrução, é adicionado ao conteúdo do registro RN para produzir o endereço do operando.

b) Modo indireto

Código	Nome	Sintaxe	Descrição
001	registro indireto	@RN	registro RN contém o endereço do operando
011	auto-incremento	@(RN)+	registro é um apontador para o endereço do operando e após sua referência é incrementado de dois.
101	auto-decremento indireto	@-(RN)	registro é decrementado de dois e após é um apontador para o endereço do operando.
111	índice indireto	@X(RN)	a soma dos conteúdos de X, armazenado na palavra seguinte à instrução, e do registro RN é usada como apontador para o endereço do operando.

c) Modo usando o PC (RN = 7)

010	imediato	# n	operando segue a instrução
011	absoluto	@# A	endereço absoluto do operando segue a instrução
110	relativo	A	endereço relativo do operando segue a instrução
111	relativo indireto	@A	endereço relativo, do endereço que contém a localização de A, segue a instrução

Exemplos :

```
BR      VOLTA      :   desvio incondicional para a po  
                        sição VOLTA  
  
JMP     %5,S0      :   desvio para a subrotina S0 ar-  
                        mazenando no registro %5 o  
                        endereço de volta
```

d. Instruções para operar com o "condition code"

São instruções que ativam ou desativam cada um ou todos os bits do "condition code", existindo um código de operação para cada uma e nenhum operando.

Exemplo :

```
CLC     -      clear carry
```

e. Instruções para operação

São tres instruções com o formato : (CÓDIGO OPERAÇÃO)

E são as seguintes:

```
HALT    :   interrompe o processamento  
  
WAIT    :   interrompe o processamento, aguardando uma  
            interrupt  
  
RESET   :   "reset" de todas unidades ligadas ao "BUS"
```

O PDP-11/05 , permite processamento com dados de um ou dois bytes. As instruções tem tamanho padrão de 2 bytes.

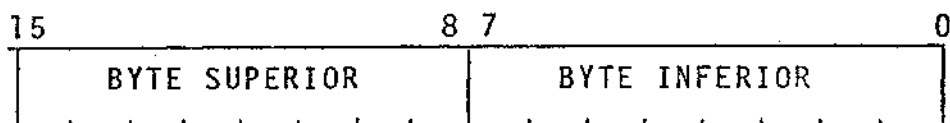


FIGURA 3.2

O repertório de instruções assembler, do modelo original,

não conta com multiplicação ou divisão inteiras e não permite operação entre números reais por hardware.

Como já foi dito, o PDP-11/05 está conectado em "time-sharing" ao PDP-10 e então para o processamento e "debug" de programas no 11 é utilizado um software residente no 10 (compilador, carregador, montador).

A tabela 1, fornece o conjunto de instruções disponível ao usuário do PDP-11/05.

Um maior detalhamento do exposto acima, encontra-se nas referências [1], [2], [3], [4].

Instruções de um operando

CLR (B) x	zera	$x \leftarrow 0$
COM (B) x	complementa (de 1)	$x \leftarrow \bar{x}$
INC (B) x	incrementa	$x \leftarrow x + 1$
DEC (B) x	decrementa	$x \leftarrow x - 1$
NEG (B) x	complemento (de 2)	$x \leftarrow -x$
TST (B) x	testa operando e ativa "condition code"	
ASR (B) x	deslocamento aritmético de 1 bit para a direita	$x \leftarrow \frac{x}{2}$
ASL (B) x	deslocamento aritmético de 1 bit para a esquerda	$x \leftarrow 2 * x$
ROR (B) x	rotação (inclusive "carry") de 1 bit para a direita	
ROL (B) x	rotação (inclusive "carry") de 1 bit para a esquerda	
SWAB x	troca de "bytes"	
ADC (B) x	soma "carry"	$x \leftarrow x + c$
SBC (B) x	subtrai "carry"	$x \leftarrow x - c$

Instruções de dois operandos

MOV (B) x,y	substituição	$y \leftarrow x$
CMP (B) x,y	compara x e y e ativa "condition code"	
ADD x,y	soma	$y \leftarrow y + x$
SUB x,y	subtrai	$y \leftarrow y - x$
BIT (B) x,y	produto lógico	$y \leftarrow y \wedge x$
BIC (B) x,y	complemento do produto lógico	$y \leftarrow \overline{y \wedge x}$
BIS (B) x,y	soma lógica	$y \leftarrow y \vee x$

Tabela 1 : Repertório de instruções

Instruções de controle

Bxx loc	transferência	PC ← loc
JMP x	transferência	PC ← x
JSR RN, SUB	execute a subrotina SUB	
RTS RN	retorne ao programa principal	
EMT	instrução de trap	
TRAP	instrução de trap	
BPT	instrução de trap	
IOT	instrução de trap	
RTI	retorno de interrupt	

Nota: Bxx representa as 17 instruções de quebra de sequência conforme a condição apresentada pelos bits do "condition code".

Instruções de modificação do "condition code"

CLx	zera "condition code"
SEx	ativa "condition code"

Tabela 1 : Repertório de instruções (continuação)

Nota: x representa uma das seguintes letras: V, Z, N, C que opera com os bits indicativos respectivamente de over flow, resultado zero, resultado negativo, "carry".

Instruções de operação

HALT	interrompe o processamento
WAIT	interrompe o processamento a espera de uma "interrupt"
RESET	inicializa todas as unidades, ligadas ao "BUS"

Tabela 1 : Repertório de instruções (continuação)

3.2 - ROTINAS ARITMÉTICAS DE PONTO FIXO E PONTO FLUTUANTE

Como o mini-computador utilizado como controlador, não dispõe de operações aritméticas em ponto flutuante, assim como não efetua divisão, nem multiplicação entre inteiros, foi desenvolvido um conjunto de rotinas para efetuar essas operações. Listagens das subrotinas são fornecidas no fim do trabalho, em anexo.

3.2.1 Rotinas de ponto fixo

Multiplicação:

É efetuada através da subrotina MULT, que multiplica o conteúdo dos registros % 2 e % 3 e apresenta o resultado em 31 bits nos registros % 4 e % 3, sendo esse último o dos bits menos significativos.

A multiplicação é efetuada através de "shifts" e somas e o algoritmo utilizado é apresentado no diagrama 3.1.

Divisão:

É efetuada através da subrotina DIVIDA, que divide o conteúdo do registro % 0 pelo conteúdo do registro % 2, ficando o resultado no registro % 3.

A divisão é efetuada com "shifts" e subtrações e o diagrama 3.2 fornece o algoritmo utilizado na implementação.

3.2.2 Rotinas de ponto flutuante

Nas rotinas desenvolvidas foi usada a notação de mantissa e expoente para os números envolvidos nas operações. Assim esses números tem suas mantissas normalizadas em valores entre 0 e 1.

A mantissa será representada em 15 bits mais um de sinal e o expoente em 7 bits mais um de sinal.

Com essa configuração, teremos nas mantissas normalizadas e ajustadas à esquerda, números no intervalo 0,5 a $(1 - 2^{-15})$ na forma de complemento de 2 e o expoente estará no intervalo -128 a 127.

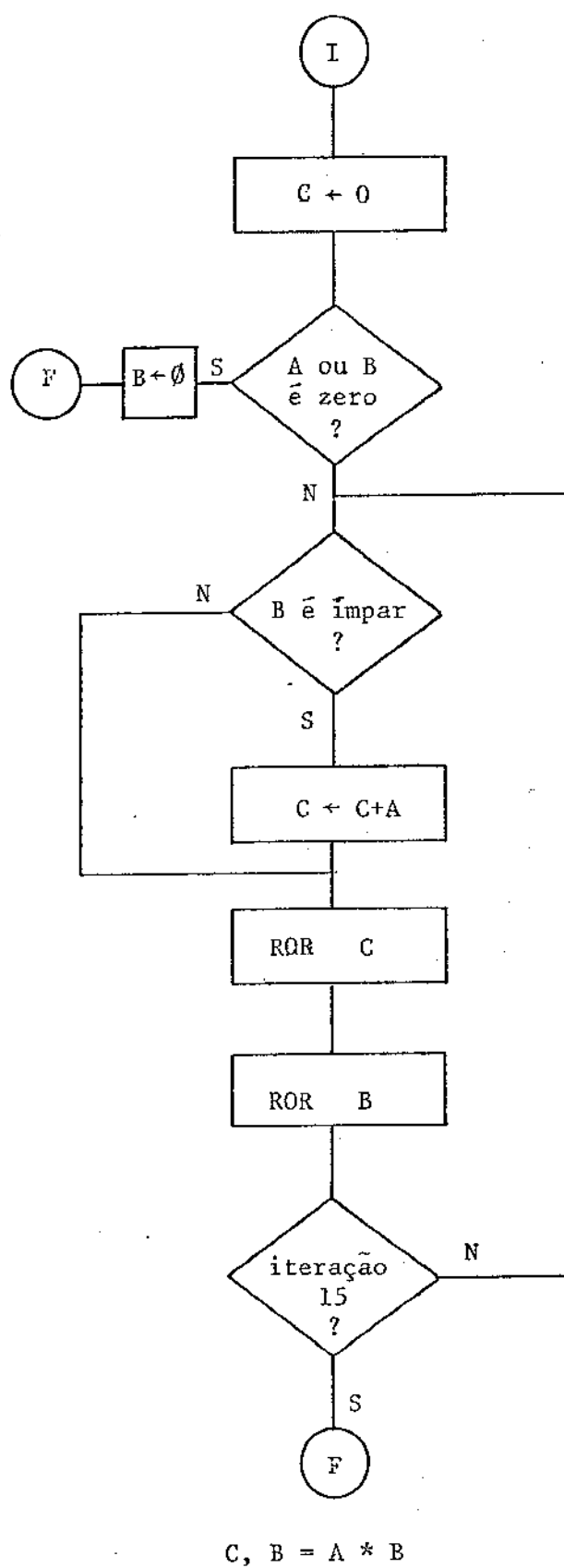
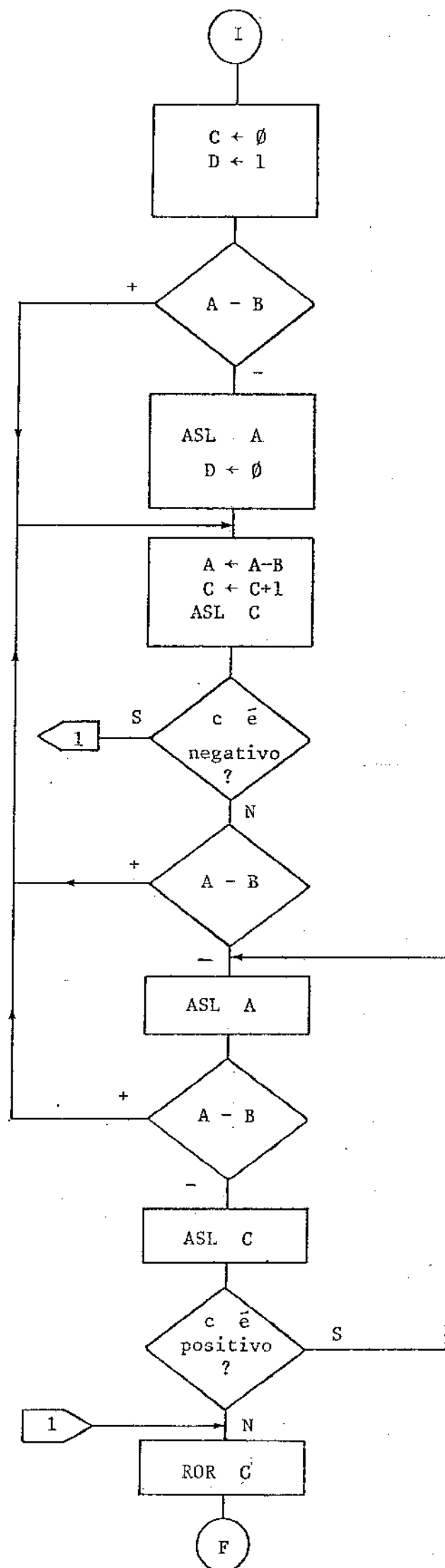


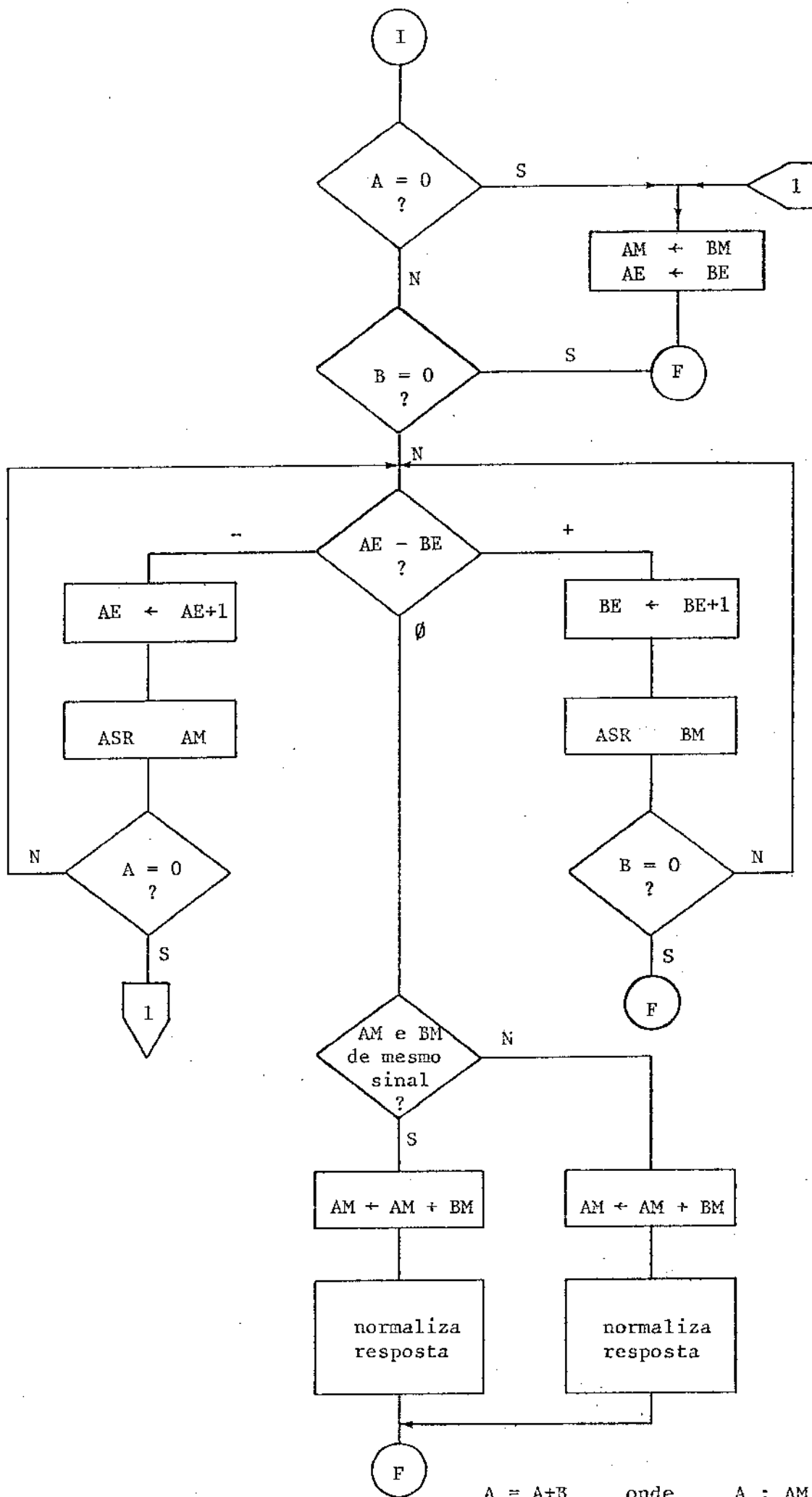
Diagrama 3.1 : Produto de dois números inteiros (MULT)



$C_1 = A \div B$ resto em A

D : expoente a somar ao expoente da resposta normalizada

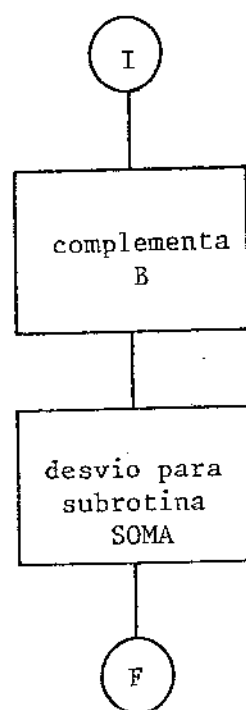
Diagrama 3.2 : Quociente de dois números inteiros (DIVIDA)



$A = A + B$ onde

- A : AM - mantissa de A
- AE - expoente de A
- B : BM - mantissa de B
- BE - expoente de B

Diagrama 3.3 : Adição de dois números reais (SOMA)



$$A = A + (-B)$$

Diagrama 3.4 : Subtração de dois números reais (SUBT)

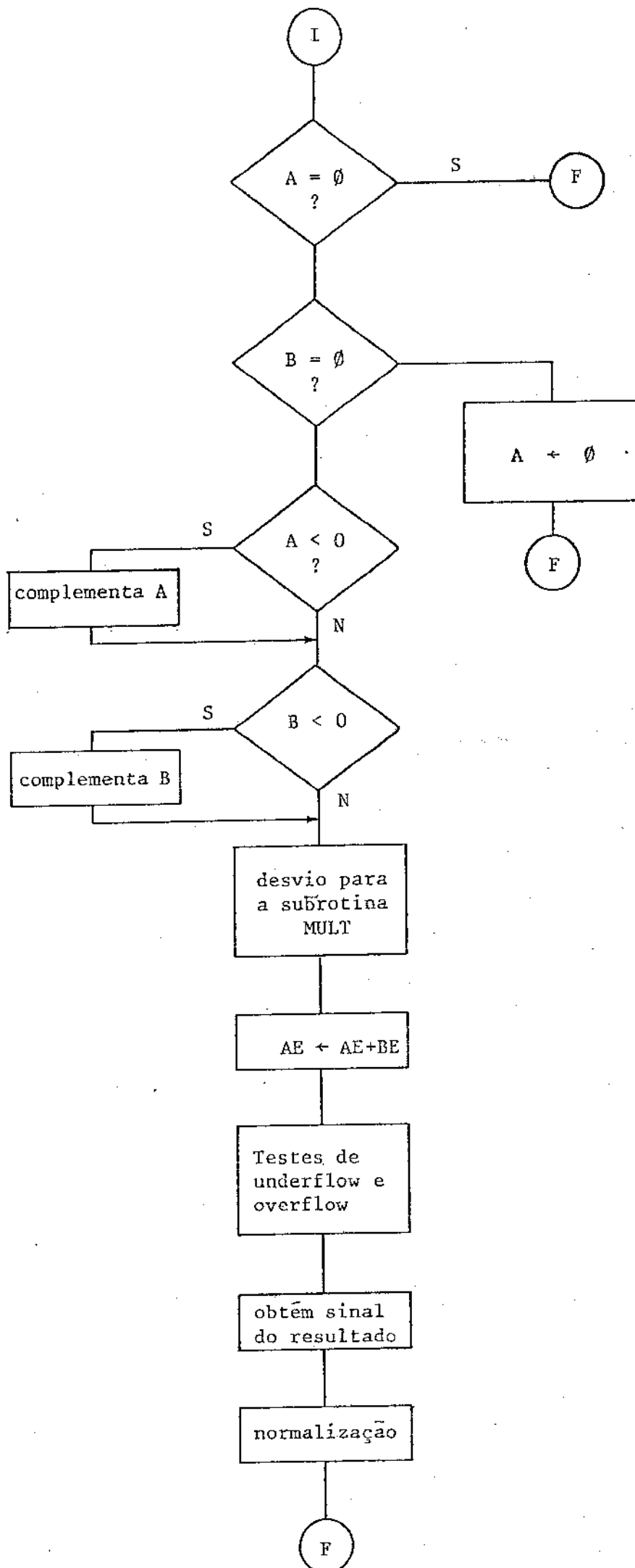


Diagrama 3.5 : Produto de dois números reais (MULTI)

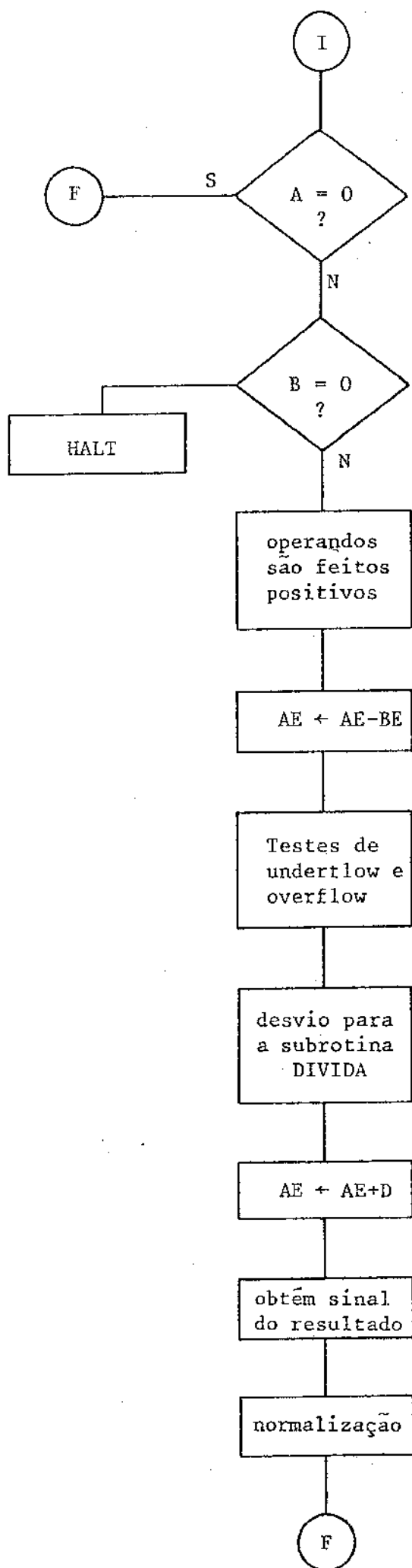


Diagrama 3.6 : Quociente de dois números reais (DIVIS)

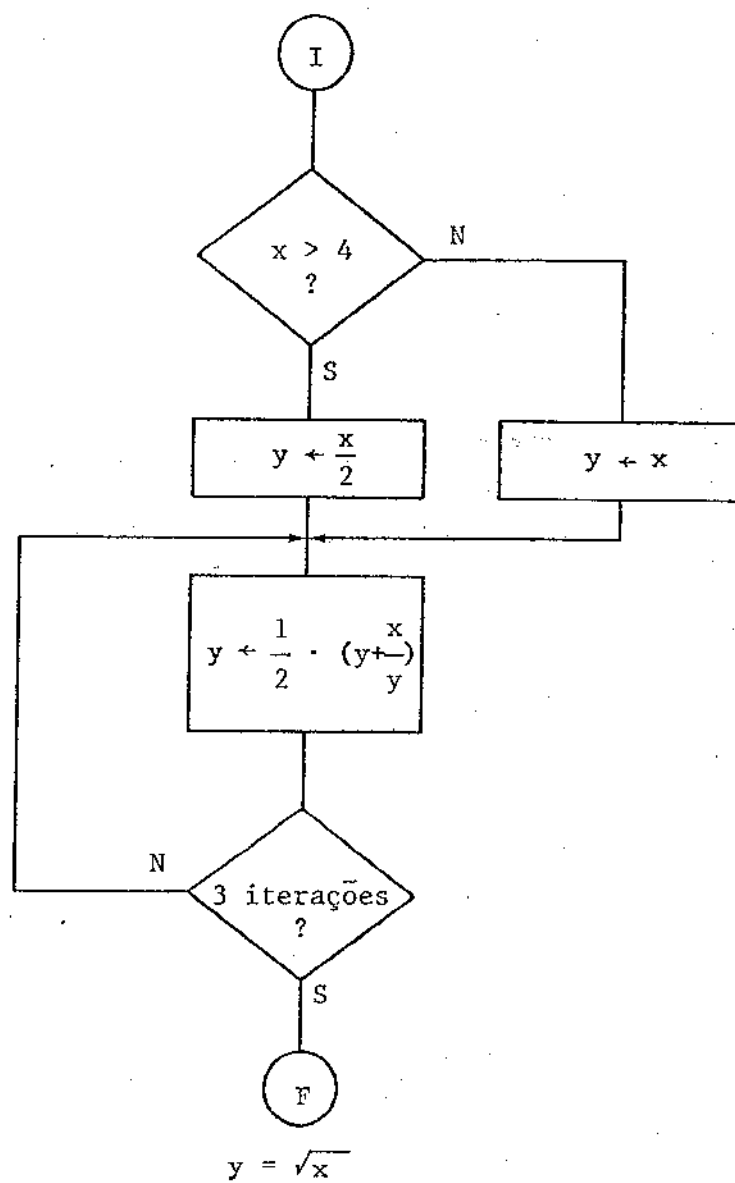
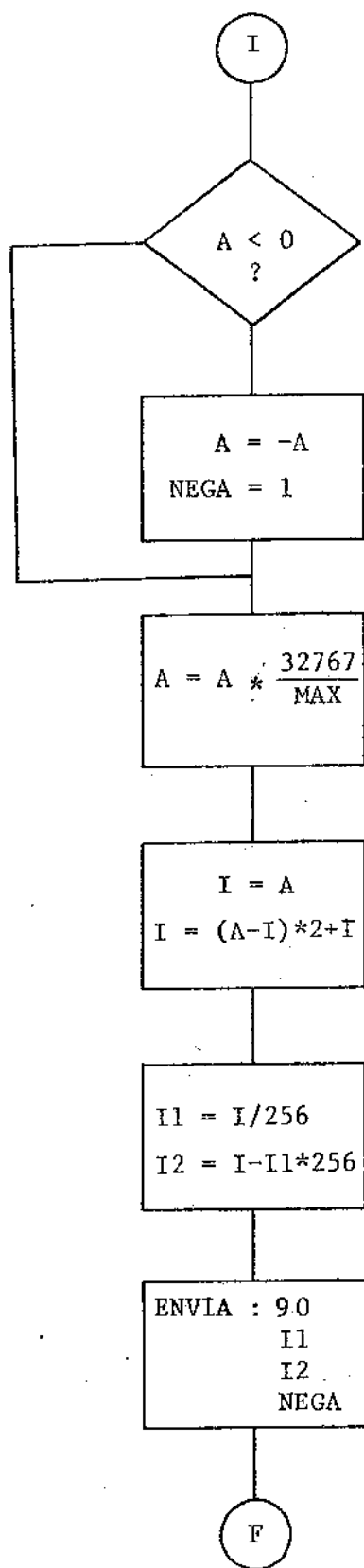


Diagrama 3.7 : Raiz quadrada de um número real positivo (RAIZ)



A : sinal analógico a ser enviado ao controlador
 MAX : valor máximo de A.
 90 : sinal de início de transmissão
 NEGA: indicador de sinal positivo ou negativo

Diagrama 3.8 : Simulador do conversor A/D (ENVIA.F4)

Para a manipulação dos operandos, usando qualquer dos tres modos de endereçamento, existem cinco tipos diferentes de instruções :

a. Instruções de um operando

Apresentam o seguinte formato :

(CÓDIGO OPERAÇÃO) (OPERANDO)

Possibilitam operações com palavras ou bytes.

Exemplos :

CLR	% 0	-	zera registro 0
INCB	UM	-	incrementa byte menos significativo da posição UM

b. Instruções com dois operandos

Apresentam o seguinte formato :

(CÓDIGO OPERAÇÃO) (OPERANDO 1, OPERANDO 2)

Possibilitam operações com palavras ou bytes, com exceção das instruções ADD e SUB.

Exemplos :

MOV	x,y	-	move conteúdo da posição x para a posição y
ADD	x,y	-	soma ao conteúdo da posição x o conteúdo da posição y

c. Instruções para controle de programas

Podem ser subdivididas em tres sub-grupos :

- instruções de desvio
- instruções para tratamento de subrotinas
- instruções para tratamento de interrupts e traps.

Com esses valores de mantissa e expoente, obtêm-se então números em ponto flutuante no intervalo:

$$\pm(0,1469 \times 10^{-38} \text{ a } 0,1714 \times 10^{39})$$

A precisão da representação é de 1 parte em 32768, ou 4 dígitos decimais.

Os diagramas 3.3, 3.4, 3.5, 3.6 descrevem algoritmos de soma, subtração, multiplicação e divisão entre números reais.

Para completar o conjunto de rotinas aritméticas necessárias à implementação do controlador, apresenta-se a rotina de extração da raiz quadrada de um número real positivo. A entrada e saída dessa rotina é feita pelos registros %0 e %1, que contêm mantissa e expoente tanto do número a operar como da resposta.

O algoritmo empregado para a raiz quadrada é descrito pelo diagrama 3.7.

3.3 - ROTINAS PARA TRANSMISSÃO DE DADOS ENTRE PDP-11/05 e PDP-10

O esquema existente para a conexão entre os computadores PDP-11/05 e PDP-10 é apresentado na figura 3.3.

O quadro (1) da figura 3.3, é o controlador e o quadro (2) representa o simulador do processo a ser controlado.

O elemento DL11E, é uma interface assíncrona que transforma dados em série para a forma paralela na recepção e na transmissão transforma os dados da forma paralela para a série. Por características do sistema, o envio de dados do processo para o controlador é feito em conjuntos de 8 bits (1 byte) enquanto a comunicação em sentido inverso é feita também com conjuntos, de 8 bits, porém em código ASCII.

As saídas de um processo a controlar, normalmente são analógicas, por exemplo tensões. Essas tensões então devem ser codificadas para poderem ser utilizadas pelo controlador, isso é feito por um conversor A/D. Da mesma forma a saída do controlador deve ser transformada num sinal analógico pelo conversor D/A.

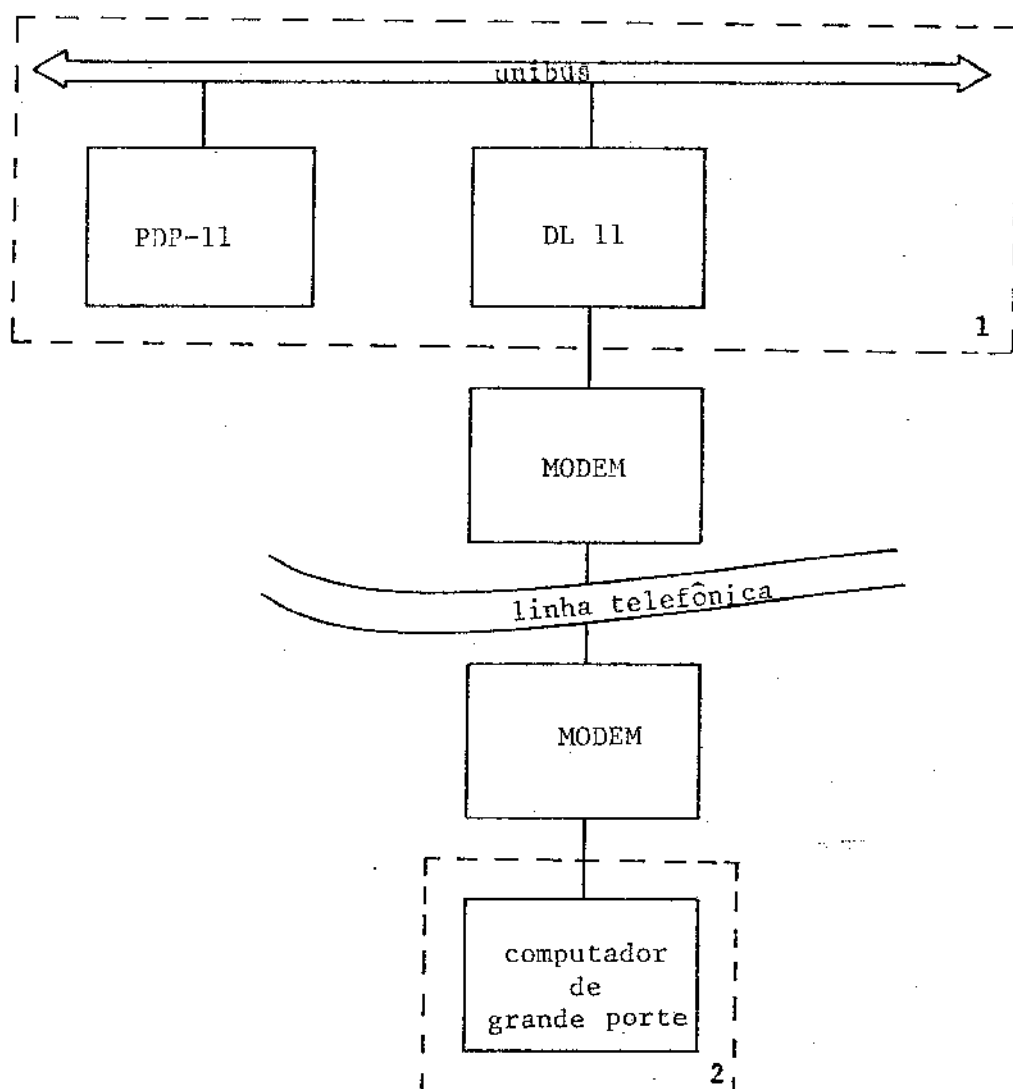


FIGURA 3.3

Na simulação aqui apresentada o conversor A/D é simulado no programa ENVIA.F4, enquanto o conversor D/A é simulado na subrotina DA no PDP-11/05, entretanto de maneira diversa da normalmente utilizada pela característica especial do envio de dados ao PDP-10 ser em ASCII .

Os diagramas 3.8 e 3.10 descrevem respectivamente os algoritmos utilizados nas conversões A/D e D/A.

A seguir são apresentadas as rotinas desenvolvidas para o mini-processador, bem como os programas de simulação no computador de grande porte.

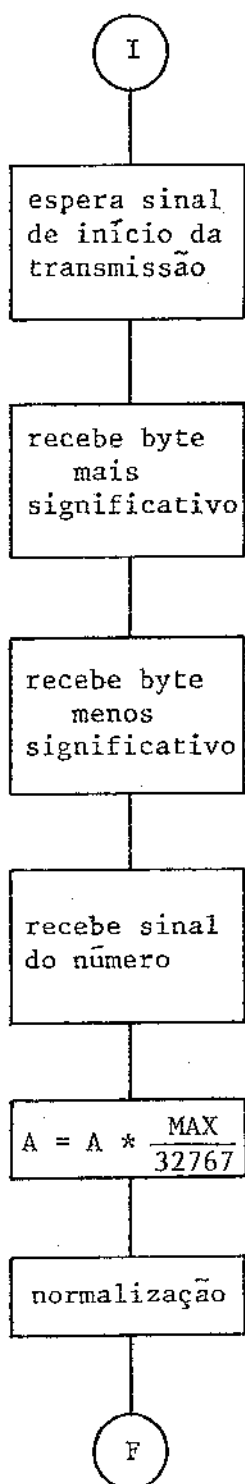
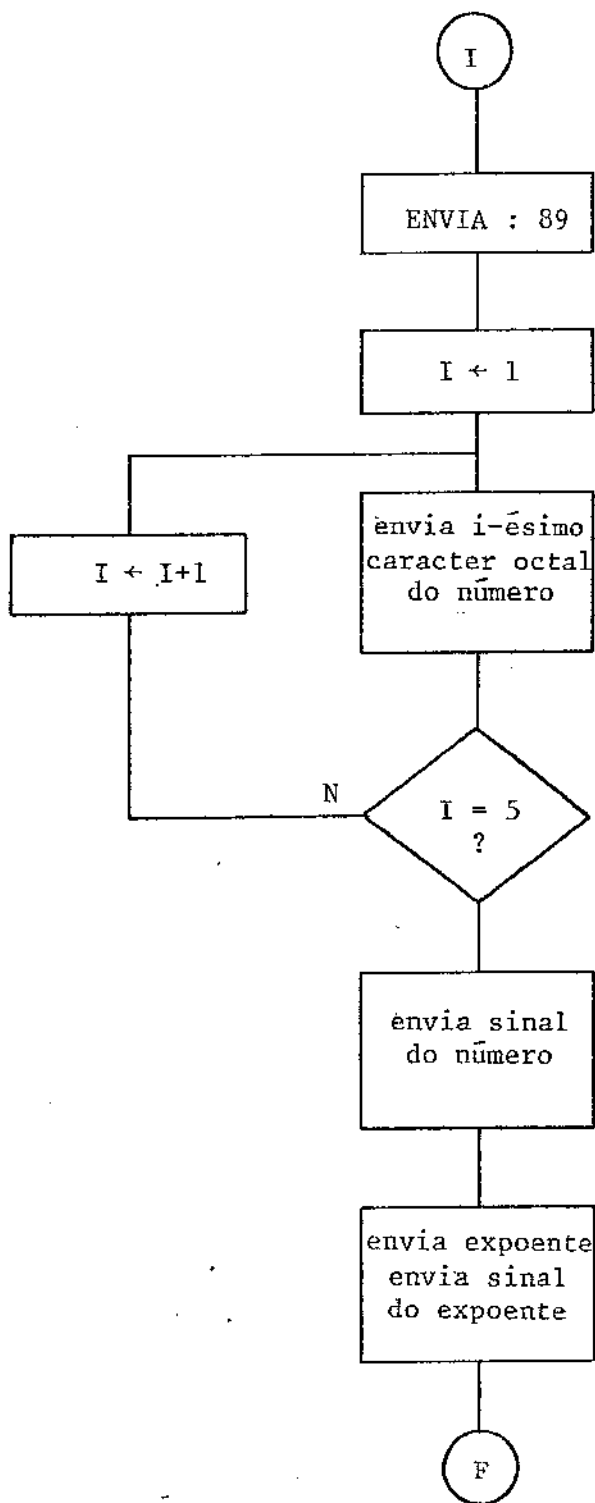


Diagrama 3.9 : Recebimento do sinal enviado pelo conversor A/D (AD)



89 : sinal de início de transmissão

Diagrama 3.10 : Simulador do conversor D/A (DA)

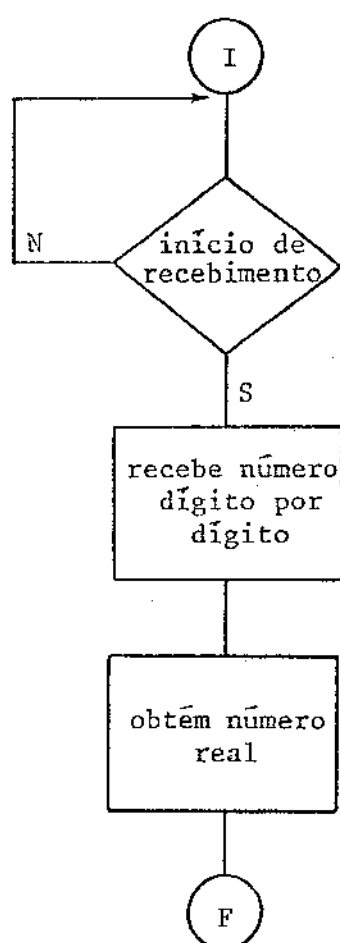


Diagrama 3.11 : Recebimento do sinal enviado pelo conversor D/A
(ENTRA.F4)

A tabela 3.1 ilustra a memória requerida e o tempo de processamento média das rotinas aritméticas.

ROTINAS	TEMPO (μ s)	MEMÓRIA (PALAVRAS)
SOMA	170	56
SUBT	190	18
MULTI	420	80
MULT	290	22
DIVIS	455	69
DIVIDA	320	20
RAIZ	1500	32
TOTAL	=	297

Tabela 3.1

3.4 - IMPLEMENTAÇÃO DO ALGORÍTMO RAT UTILIZANDO APROXIMAÇÃO ESTO-CÁSTICA NA ESTIMAÇÃO

Nesse ítem é apresentada a implementação do algoritmo descrito no ítem 2.2, no mini-computador PDP-11/05.

O exemplo utilizado é o mesmo do ítem 2.4.2, que servirá como objetivo a ser alcançado por essa implementação no mini-processor.

As figuras 3.4 a 3.8 fornecem os resultados obtidos com o algoritmo.

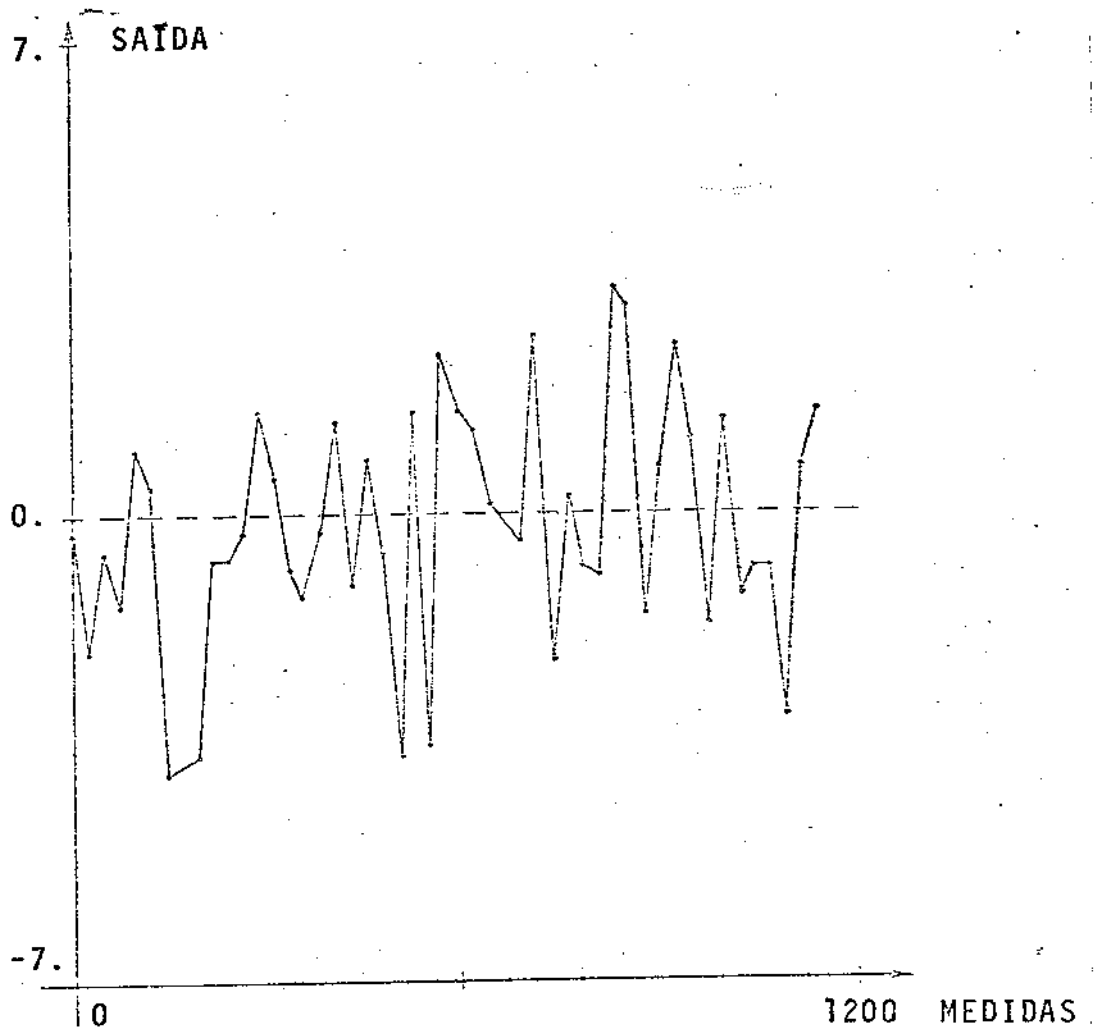


FIGURA 3.4

FIGURA 3.4 : Saída do processo

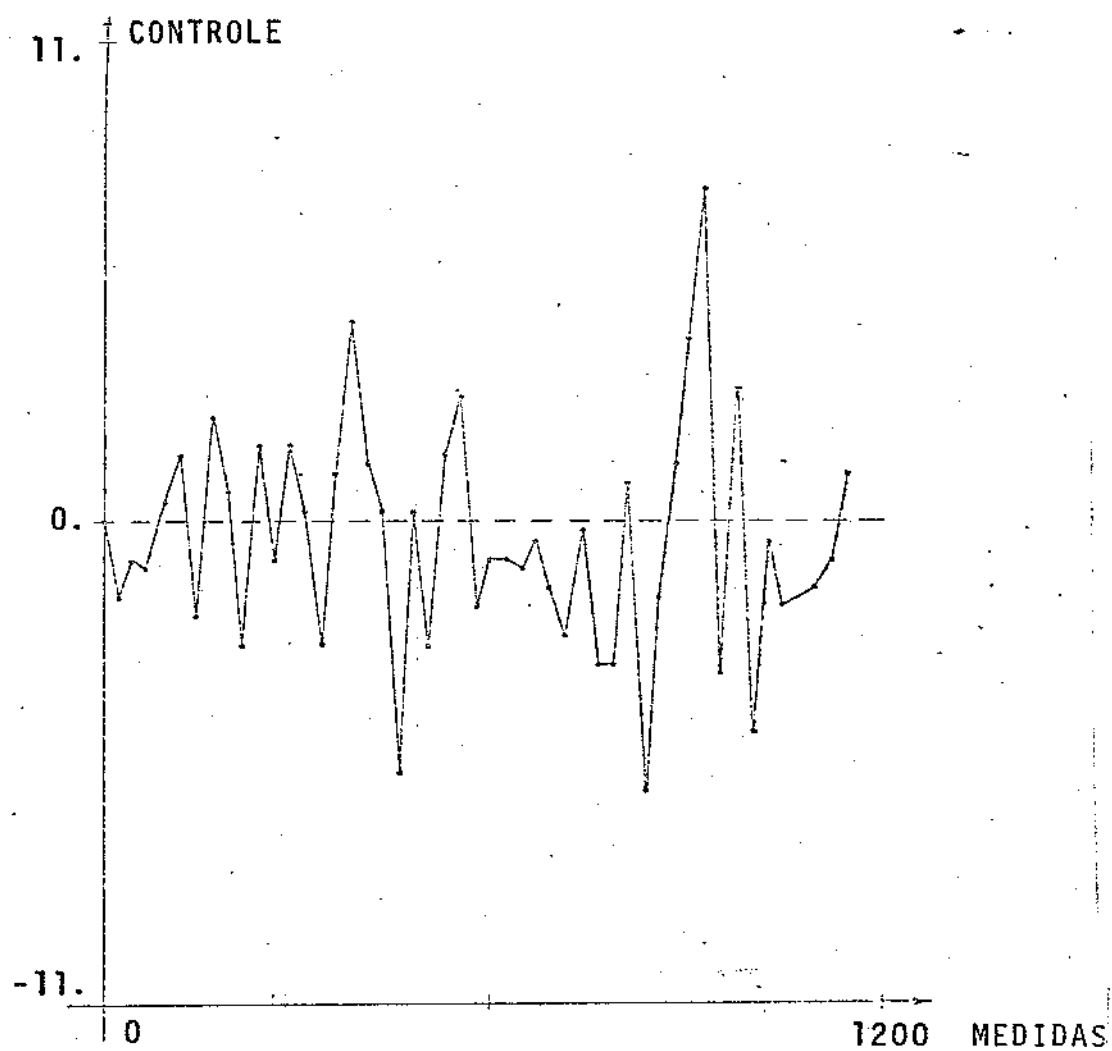


FIGURA 3.5 : Sinal de controle

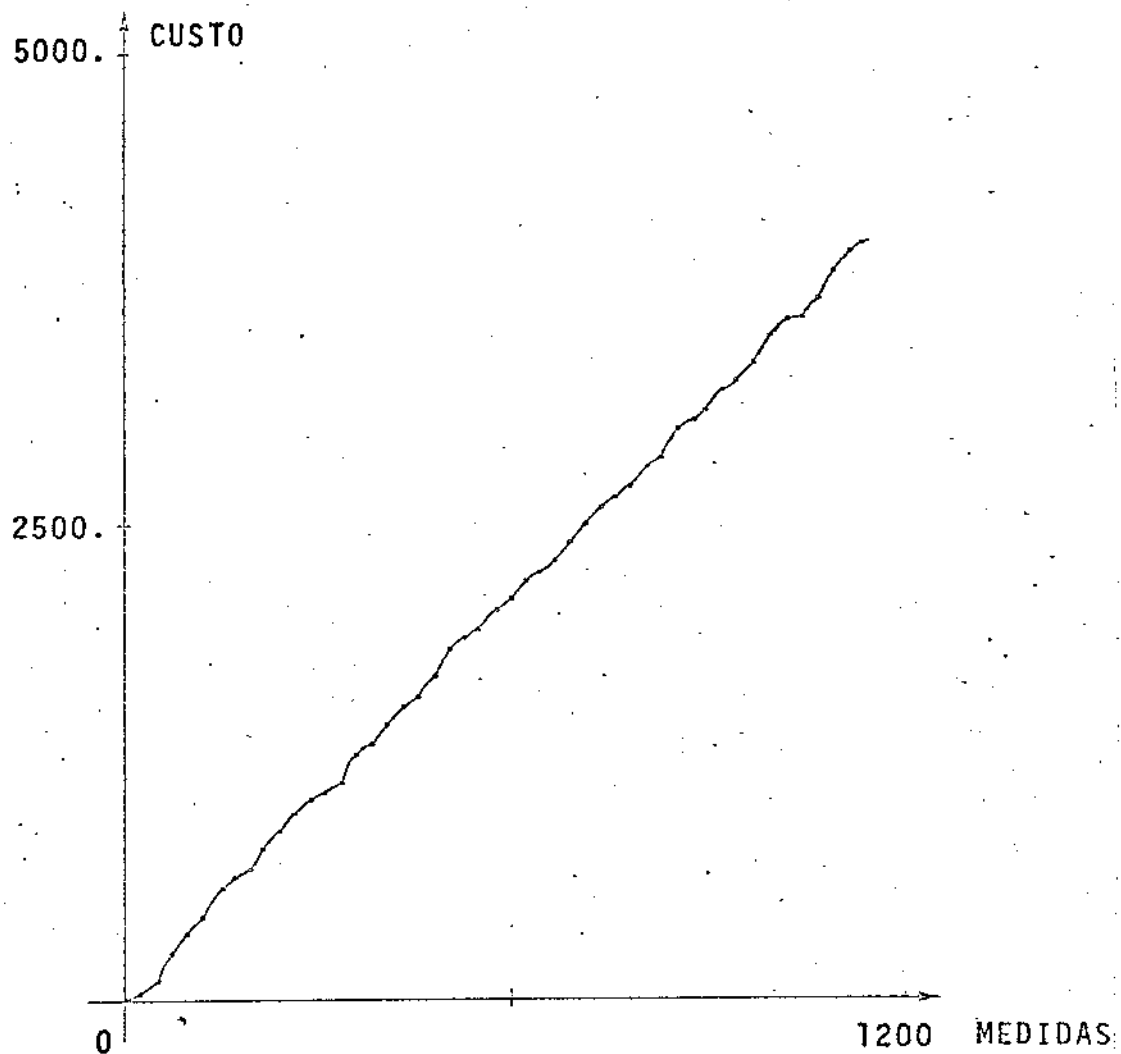


FIGURA 3.6 : Função de custo

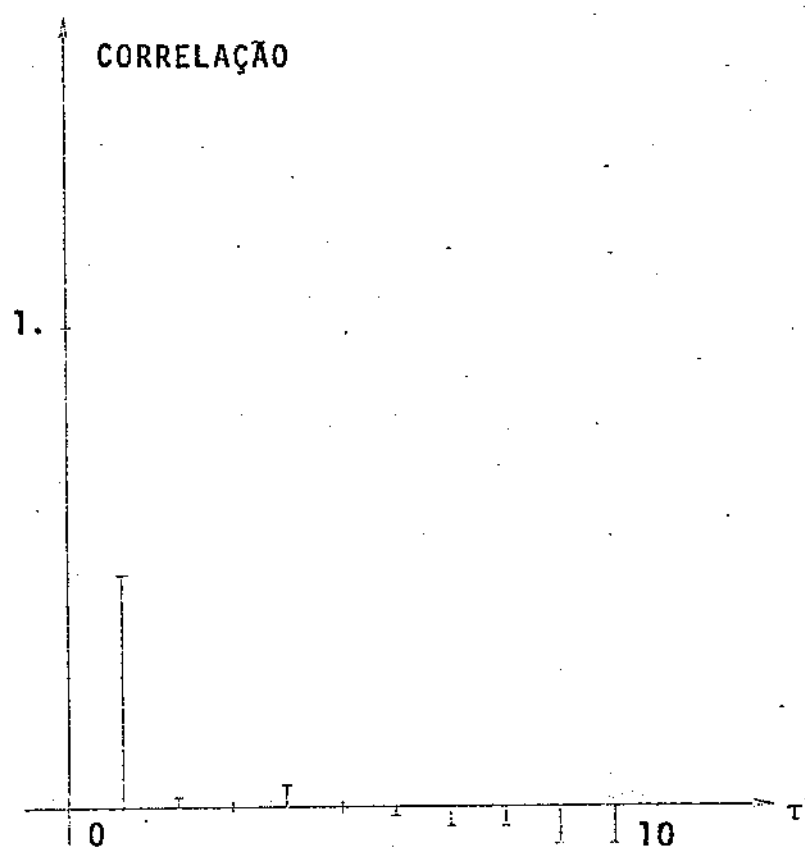


FIGURA 3.7 : Correlação $y(t)$ $y(t + \tau)$

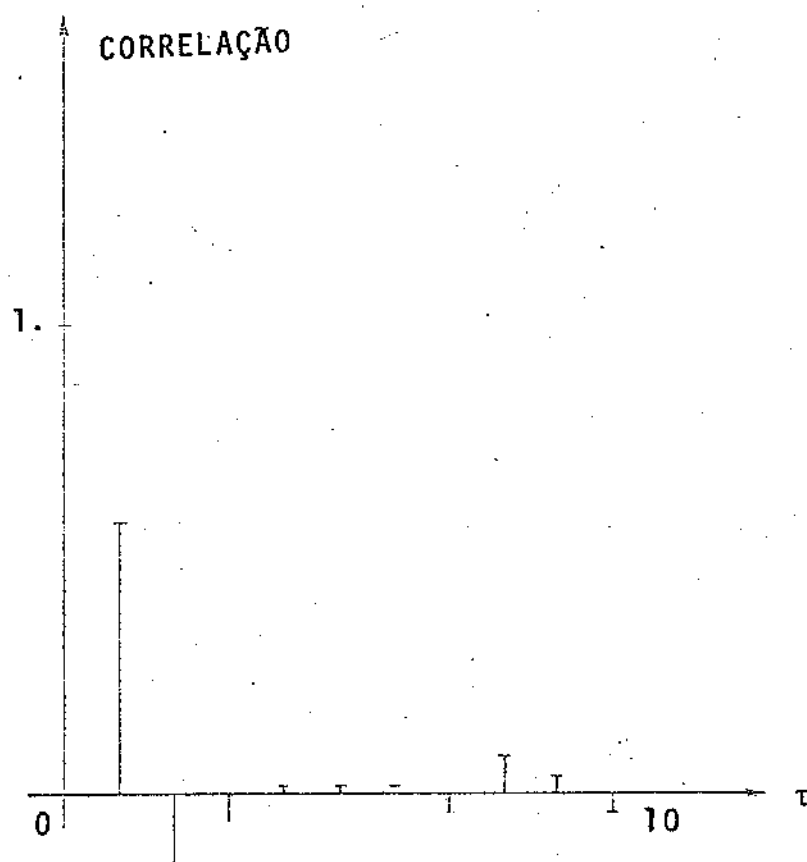


FIGURA 3.8 : Correlação $y(t)$ $u(t + \tau)$

Foram adotados os valores iniciais dados pela tabela 2.5 e o valor adotado para $RH0$ foi 0,9 até o passo 900, sendo a seguir elevado para 1. O sinal de controle foi limitado ao intervalo $-10 \leq u(t) \leq 10$, não ocorrendo nenhuma violação dos limites durante a simulação. Essas condições são as mesmas adotadas para o exemplo do item 2.4.2.

A variância obtida foi 3,56.

Comparando as figuras obtidas no item 2.4.2 com as figuras 3.4 a 3.8, vê-se que o controlador desenvolvido no mini-computador, apresenta características de funcionamento bem próximas do algoritmo implementado no computador de grande porte.

O tempo médio requerido para cada iteração do controlador, onde se faz estimativa e controle é de aproximadamente 11,5 ms.

3.5 - IMPLEMENTAÇÃO DO ALGORITMO RAT, UTILIZANDO O MÉTODO DA RAIZ QUADRADA NA ESTIMAÇÃO

Nesse item é apresentada a implementação do algoritmo descrito no item 2.3, no mini-computador PDP-11/05.

O exemplo utilizado é o mesmo do item 2.4.3, que servirá como objetivo a ser alcançado por essa implementação.

As figuras 3.9 a 3.13, fornecem os resultados obtidos com o algoritmo.

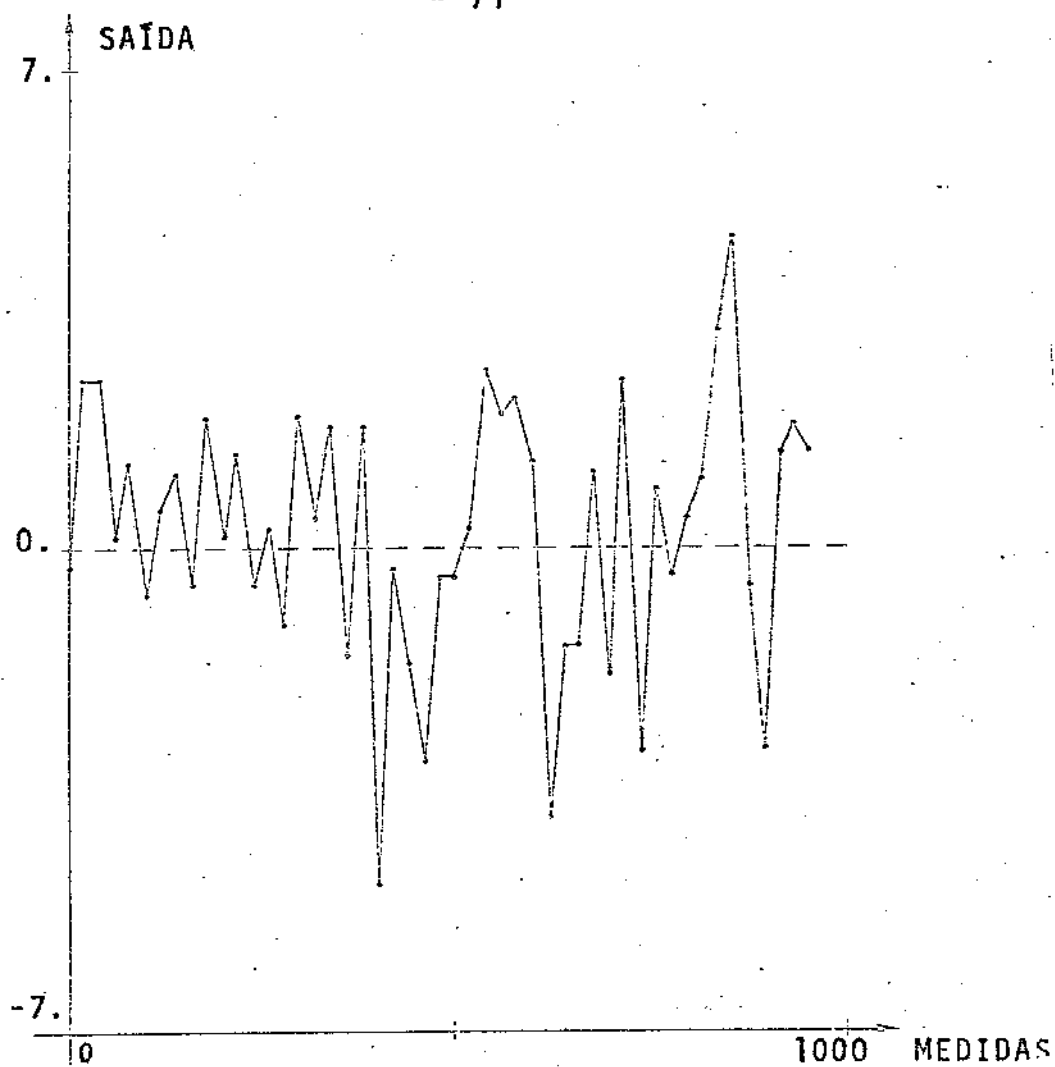


FIGURA 3.9 : Saída do processo

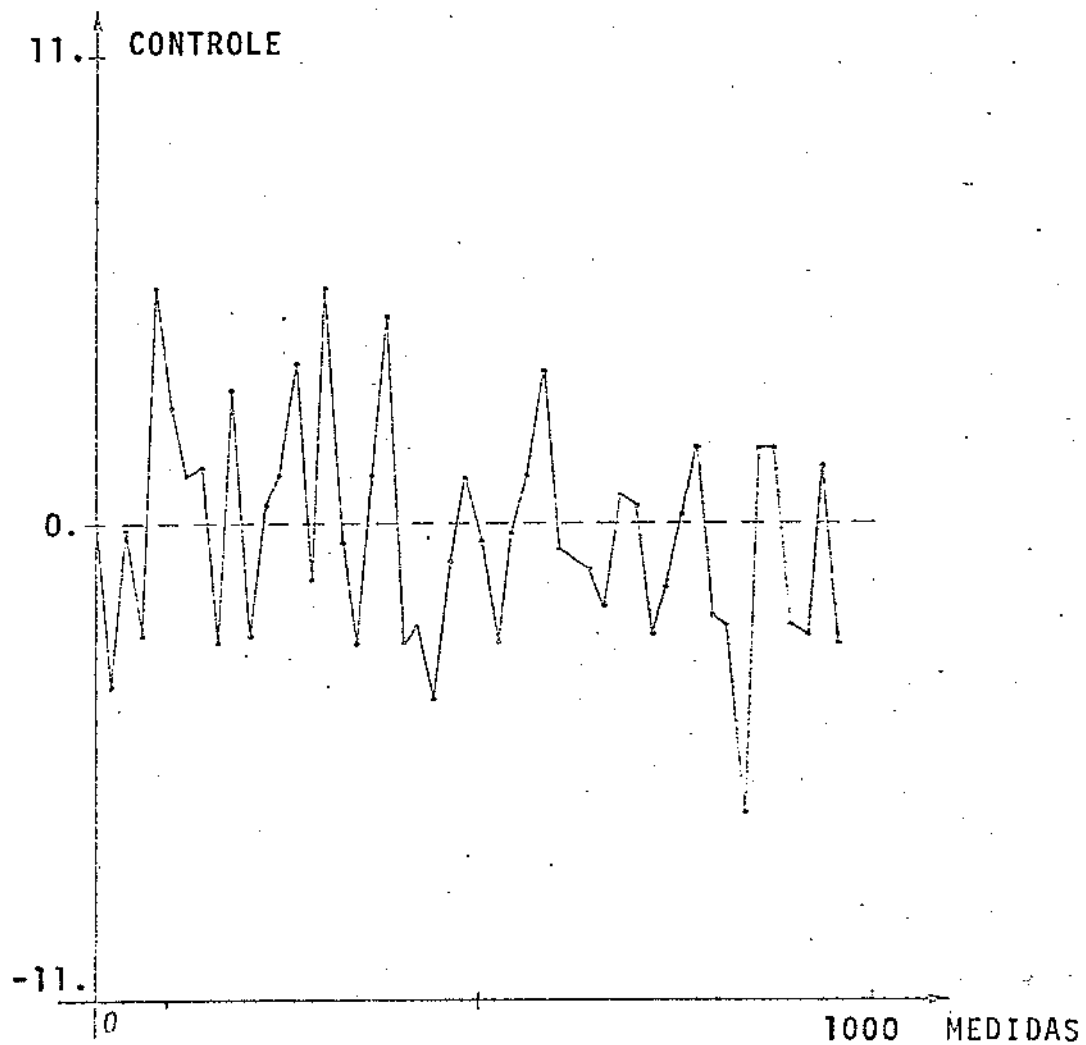


FIGURA 3.10 : Sinal de controle

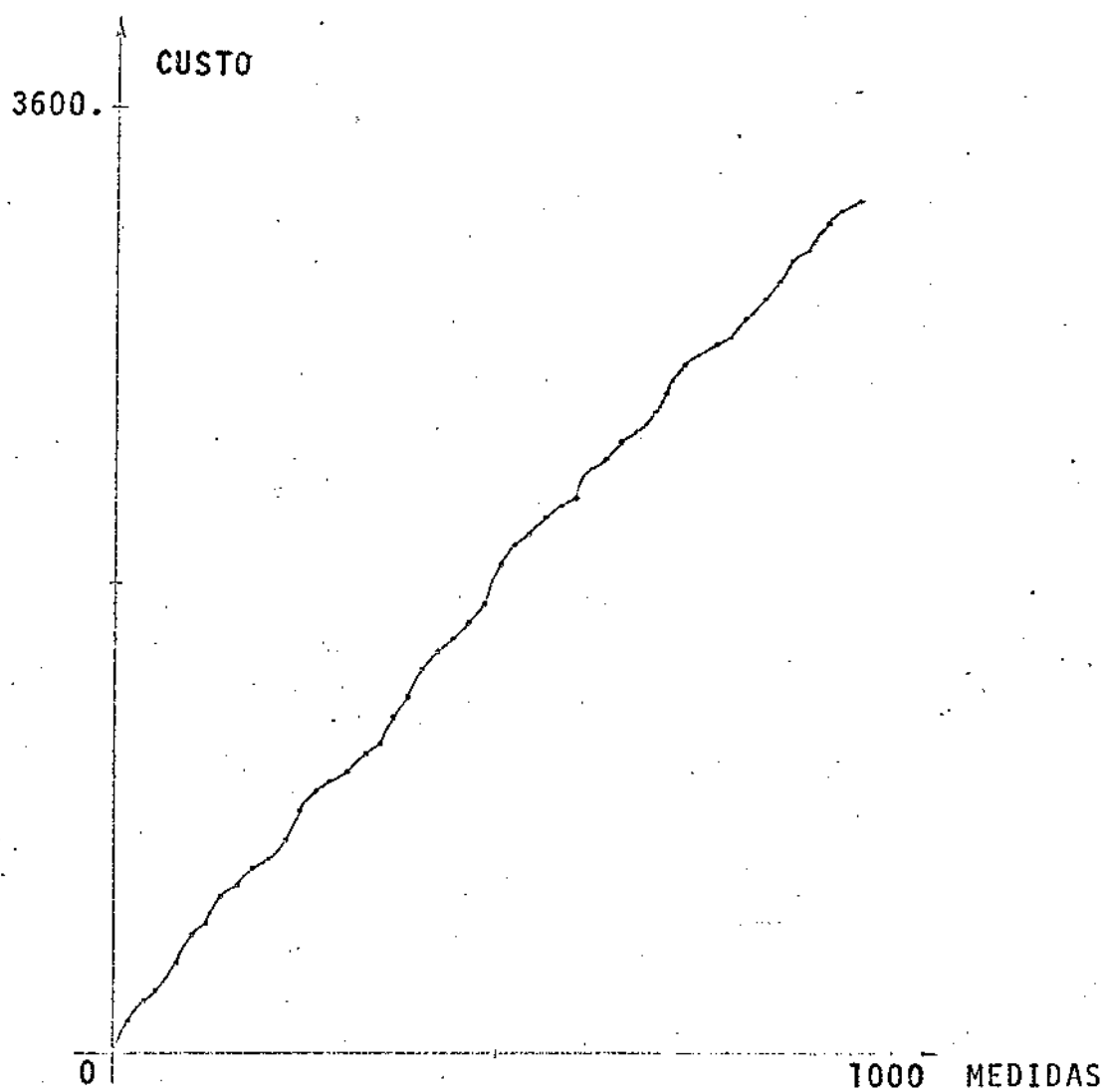


FIGURA 3.11 : Função de custo

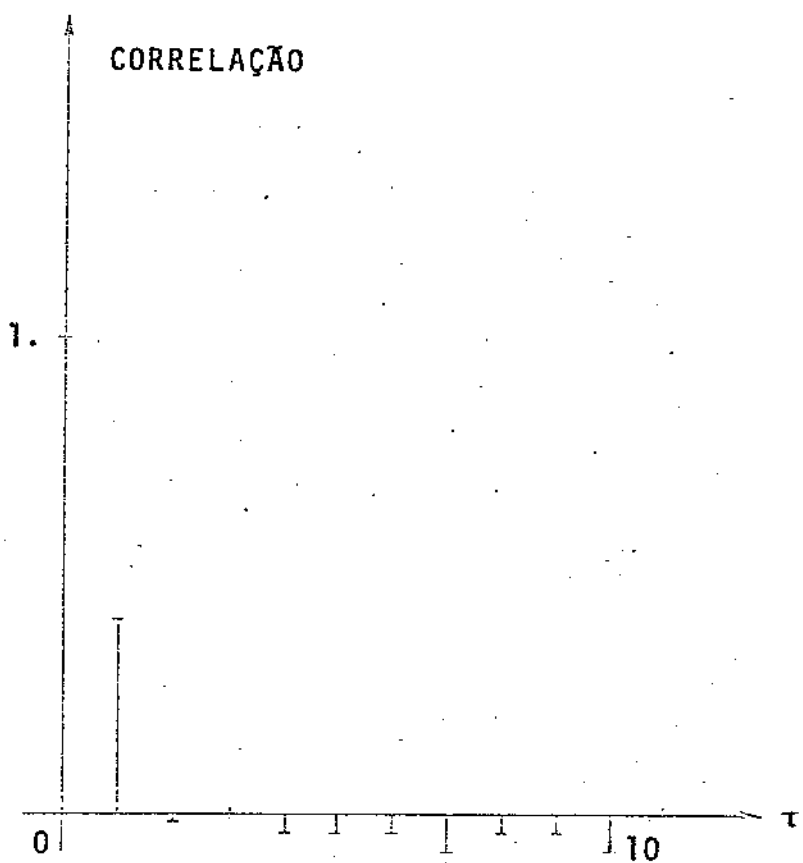


FIGURA 3.12 : Correlação $y(t)$ $y(t + \tau)$

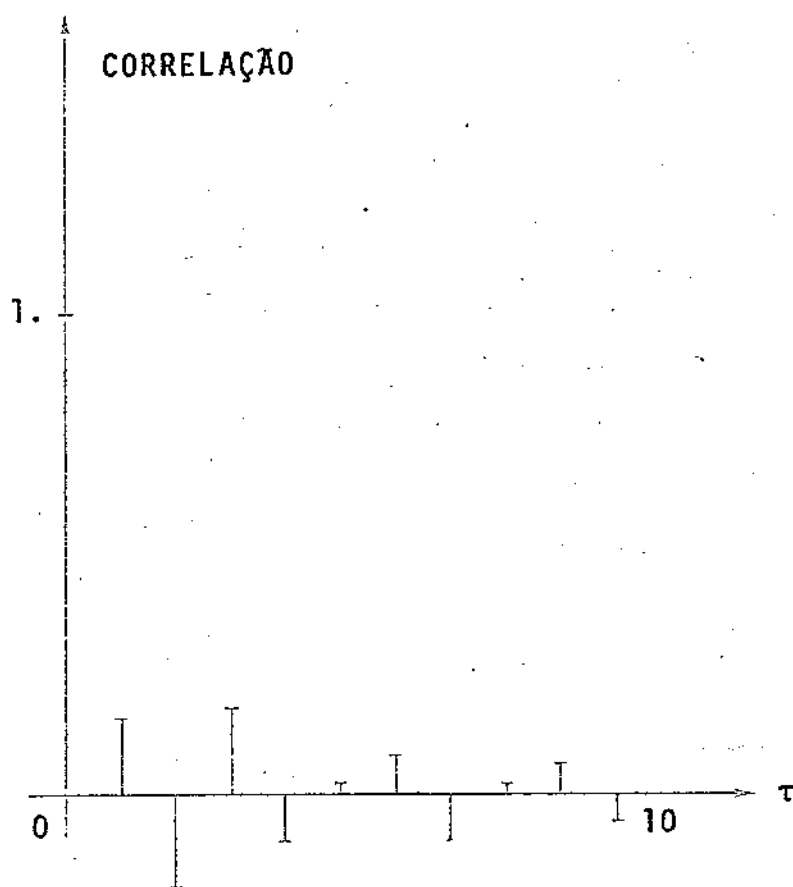


FIGURA 3.13 : Correlação $y(t)$ $u(t + \tau)$

Foram adotados valores iniciais nulos para os parâmetros, o valor de RHO foi mantido em 1 durante todo o processamento e o valor inicial para a matriz covariância foi $3 * I$, onde I é a matriz identidade. O sinal de controle foi limitado ao intervalo $-11 \leq u(t) \leq 11$, tendo ocorrido uma violação dos limites durante a simulação.

A variância obtida foi 3,47.

Comparando as figuras obtidas no item 2.4.3 com as figuras desse item, vê-se que o controlador implementado no mini-computador, apresentou um comportamento bem próximo do apresentado pelo controlador desenvolvido no computador de grande porte, apesar de no pequeno processador, trabalhar-se com menor número de dígitos significativos.

Aliás a razão de se usar a triangularização superior e inferior da matriz de covariância dos parâmetros, é justamente evitar que a menor precisão dos cálculos existente num mini-computador,

leve a erros significativos no desenvolvimento do controle. Isso já foi anteriormente focalizado no capítulo 2 e em [5].

O tempo médio requerido para cada iteração do controlador, onde se faz estimação e controle é de aproximadamente 41,5 ms.

3.6 - CONCLUSÕES

Primeiramente se resumirá em uma tabela as características dos dois algoritmos implementados no PDP-11/05.

	RAT COM APROX. ESTOCÁSTICA	RAT COM RAIZ QUADRADA
<u>ALGORÍTMO DE</u> <u>CONTROLE</u> MEMÓRIA (*)	97	97
TEMPO (* *)	2,5	2,5
<u>ALGORÍTMO DE</u> <u>ESTIMAÇÃO</u> MEMÓRIA (*)	378	967
TEMPO (* *)	9	39
VARIÁVEIS UTILIZADAS MEMÓRIA (*)	42	78
TOTAL DE MEMÓRIA	$\sim \frac{1}{2} \text{ k}$	$\sim 1,2 \text{ k}$

(*) = em nº de palavras de 16 bits

(* *) = em ms, médio em cada iteração

Tabela 3.2

Um outro parâmetro importante, não citado no capítulo 2, é o tempo de amostragem. Esse parâmetro está diretamente ligado as constantes de tempo do sistema a controlar e assim depende de cada aplicação particular.

Os algoritmos aqui propostos, apresentam tempos de execução de 11,5 ms para o RAT com aproximação estocástica e 41,5 ms para o RAT com raiz quadrada.

Deve-se ainda acrescentar a esse tempo, o intervalo necessário ao envio de dados entre processo e controlador. Supondo-se uma capacidade de transmissão de 1200 bits/seg (como existe no caso aqui simulado entre PDP-10 e 11) e ainda que as comunicações nos dois sentidos são realizadas segundo o diagrama 3.8, tem-se um total de 27 ms para cada transmissão.

Esquemáticamente pode-se representar o tempo decorrido em cada fase do controle, pela figura :

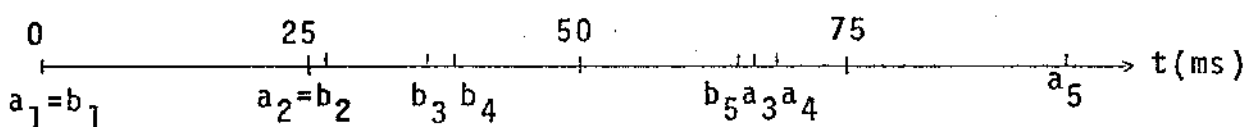


FIGURA 3.14

onde a_i = RAT com raiz quadrada

b_i = RAT com aproximação estocástica

$i=1$: início da transmissão de dados ao controlador

$i=2$: fim da transmissão de dados ao controlador e início do algoritmo de estimação

$i=3$: fim do algoritmo de estimação e início do algoritmo de controle

$i=4$: fim do algoritmo de controle e início da transmissão do controle ao processo

$i=5$: fim da transmissão do controle ao processo

Assim, utilizando uma linha de transmissão de 1200 bits/seg, o tempo mínimo para duas amostragens sucessivas do processo é:

ALGORÍTMO COM APROX. ESTOCÁSTICA : 65,5 ms

ALGORÍTMO COM RAIZ QUADRADA : 95,5 ms

É necessário ter presente o fato de que a aparente vantagem em termos de memória e tempo de execução do algoritmo RAT utilizando aproximação estocástica, pode ser totalmente anulada, pelo fato desse algoritmo exigir um relativo conhecimento a-priori, dos valores iniciais dos parâmetros, para que se tenha uma boa taxa de convergência. Uma possível aplicação interessante para esse tipo de algoritmo, é usá-lo em conjunto com o algoritmo RAT da raiz quadrada, usando o primeiro em situações de regime estacionário e o segundo em regime transitório. Essa versão conjunta poderá existir em casos onde haja um computador de médio porte controlando diversos processos simultaneamente e por isso mesmo necessitando um máximo de memória principal (RAM) disponível.

Finalizando notou-se, principalmente quando da implementação do RAT com o método da raiz quadrada, que podem ocorrer maus desempenhos do algoritmo de controle, por problemas de precisão numérica. Em casos extremos, quando a variação dos parâmetros do algoritmo de controle não for suficiente para corrigir esses erros, uma medida que pode ser tomada, será a de aumentar o número de dígitos significativos, aproveitando o "byte" mais significativo da palavra que contém o expoente para expandir a mantissa até 23 bits, o que acrescentará dois dígitos significativos a qualquer operação aritmética. Tal providência aumentará, no entanto, a memória requerida e tempo de execução dos algoritmos de controle.

3.7 - BIBLIOGRAFIA

1. Processor Handbook PDP-11/05 . Digital Equipment Corpora -
tion . 1973.
2. GT-40 Graphic Display Terminal ,vol. 1 e 2 , DEC-11-HGTVA-
B-D e DEC-11-HGTMA-A-D . Digital Equipment Corporation
1974.
3. Daltrini ,B.M. - "Sistema para representação da informação
em terminais gráficos de pequeno porte".Tese de Mes -
trado - FEC , 1976.
4. Stone ,H.S.; Siewiorek ,D.P. - "Introduction to Computer
Organization and Data Structures : PDP-11".Edition
McGraw-Hill , 1975
5. Clarke ,D.W.; Cope ,S.N.; Gawthrop ,P.J. - "Feasibility
study of the application of micro-processors to self-
-tuning controllers".Oxford.(Report 1137, 1975).

CAPÍTULO IV

PROJEÇÃO PARA O USO DE MICRO-PROCESSADORES

Durante longo tempo, empregou-se para controle de processos "on line", controladores analógicos não programáveis e acoplados a um único processo.

Com o desenvolvimento tecnológico surgiram controladores do tipo digital e híbridos, em substituição aos controladores puramente analógicos; no entanto esses controladores ainda continuaram não programáveis.

Uma fase posterior, foi alcançada com a miniaturização dos circuitos integrados por diversas técnicas LSI e assim uma outra opção surgiu para o uso em controle de processos, a utilização de mini-computadores. Esse tipo de controlador, programável, permitia uma grande flexibilidade no desempenho do controle, embora por seu elevado custo, só justificasse sua aplicação em controle simultâneo ("time-sharing") de diversos processos.

Ao mesmo tempo e independentemente desse desenvolvimento, começaram a surgir processadores digitais de pequeno porte, originando os micro-processadores, ou processadores contidos em um único "chip".

Inicialmente esses micro-processadores possuíam somente memória de leitura (ROM), permitindo controladores de algoritmos fixos, uma vez que sua memória, fornecida pelo fabricante não podia ser facilmente modificável pelo usuário comum.

Nos últimos dois anos, a utilização de memórias de leitura e escrita (RAMs), já agora realizáveis, por exemplo pela técnica NMOS, e apresentando dimensões razoáveis, 2k-bytes, levou a utilização em micro-processadores de algoritmos de controle antes implementáveis somente em mini-computadores, com a vantagem de por seu reduzido tamanho físico, resistência e baixo custo, poderem ser amplamente utilizados.

Com esse último desenvolvimento tecnológico, chegou-se à idéia de utilizar redes de computadores, em controle, empregando midi, mini e micro-computadores.

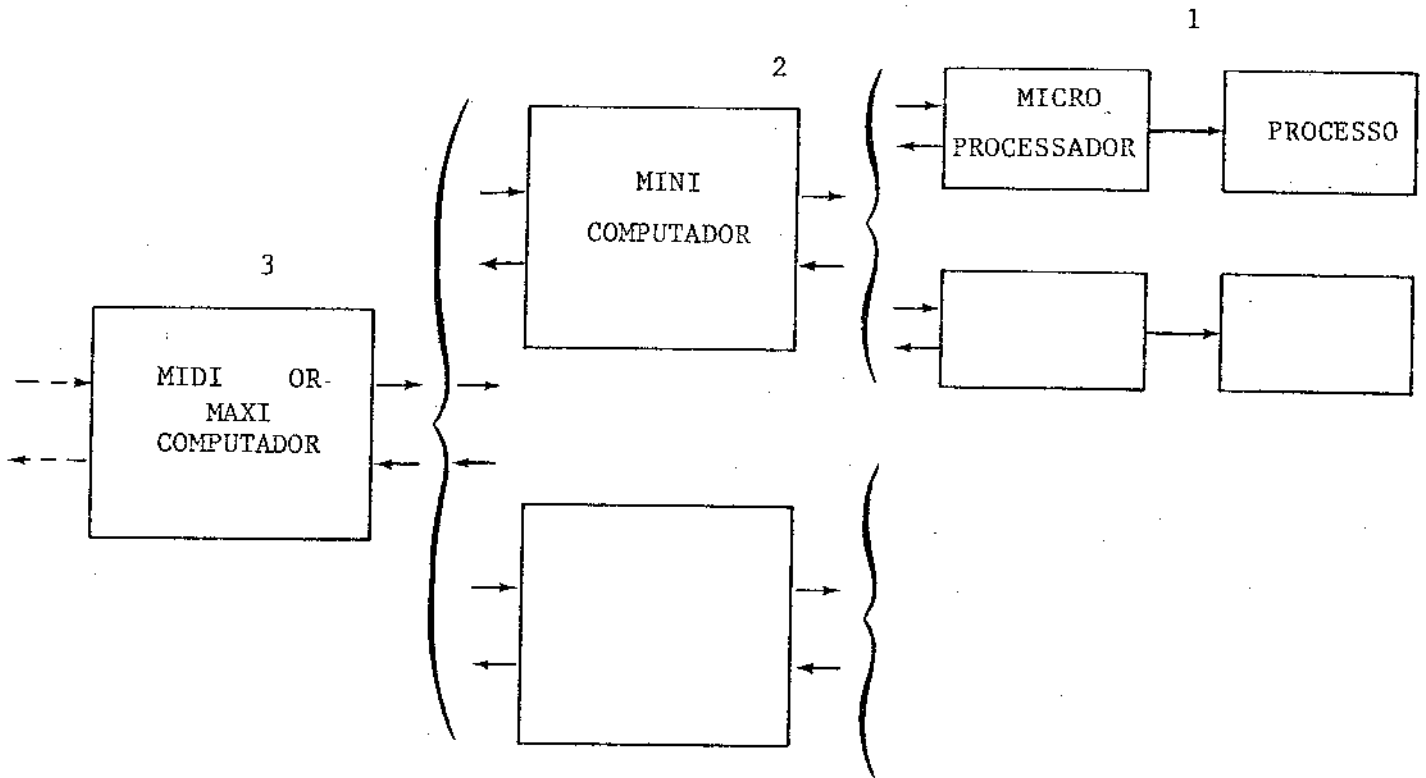


FIGURA 4.1

No nível 1, da figura 4.1, há o(s) processo(s) a controlar e como controlador aplicado diretamente a ele(s), um micro-processador, substituindo neste nível os antigos controladores analógico-digitais.

Na utilização de micro-processadores, tem-se um elevado custo para entrada e saída de dados, visto o preço de um "keyboard" ser mais elevado que o próprio preço do micro-processador e acessórios. Assim no nível 2, tem-se um mini-computador controlando e dialogando com diversos micro-processadores, para viabilizar economicamente o controlador. Conforme a dimensão do sistema a controlar, pode-se justificar o surgimento de um terceiro nível, onde se teria um midi (ou macro) - computador, controlando os níveis mais inferiores.

Com a finalidade de mostrar a extrema potencialidade dos micro-processadores existentes atualmente no mercado, na tabela 4.1, compara-se um típico representante da classe, o INTEL 8080,

com o mini-computador PDP-11/05 utilizado nesse trabalho e que se situa na fronteira entre micro e mini-processadores.

	PDP-11/05	INTEL-8080
comprimento da palavra (bits)	16	8
conjunto de instruções	~ 60	~ 69
modos de endereçamento	direto: com registro auto-incremento auto-decremento índice indireto usando PC: imediato absoluto relativo relativo indireto	direto registro registro indireto imediato
memória endereçável (palavras)	32 k	64 k
ciclo de memória	900 ns	2 μ s
tempo de execução ADD R1, R2	3.7 μ s	8 μ s (4 instruções)
instruções em ponto flutuante	chip opcional	não
instruções de multiplicação e divisão inteiras	chip opcional	não
processamento de stack	software (stack pointer)	software (stack pointer)
interrupts	multi-line multi-level vectored	1 nível
registros	8 de 16 bits	6 de 8bits, concatenáveis 1 de 16bits (acumulados)
preço em \$	2000	200

TABELA 4.1

Pelo exposto na tabela 4.1, conclue-se que a utilização de um micro-processador da classe do INTEL 8080, levará a resultados equivalentes aos obtidos no capítulo 3. E embora o tempo de processamento exigido para o mesmo algoritmo possa se elevar, o custo se reduzirá drasticamente.

Ao lado de todas essas vantagens do uso de micro-processadores, há um impedimento devido ao fato do algoritmo de controle RAT por vezes exigir o ajuste de certos parâmetros durante o processamento, o que requereria a conexão de "keyboard" ao micro-processador, periférico que seria mais dispendioso que o próprio sistema controlador. Isso enfatiza mais uma vez o uso de sistemas de controle do tipo exemplificado na figura 4.1 .

Finalizando, apresenta-se um resumo das principais características do controlador RAT implementado, bem como dos resultados obtidos.

CONCLUSÕES:

- o algoritmo RAT desenvolvido a partir de um regulador de variância mínima, embora exija grandes esforços matemáticos para estudos relativos a sua convergência e estabilidade, do ponto de vista de implementação em computadores é altamente interessante por não requerer grande quantidade de memória ou sofisticadas expressões aritméticas.
- a aferição do comportamento do controle é facilmente feita em qualquer fase do processo, conforme atestam os teoremas 1 e 2 do capítulo 1 , bastando para isso ter-se as correlações entre entrada e saída, entre entradas nos diversos instantes e a variância da saída.
- o uso do método de "square root" para a estimação, conforme estudos feitos por Clarke e aqui confirmados, diminui os erros devido a truncamentos e operações aritméticas feitas no mini-computador, sendo portanto de grande interesse em aplicações práticas com computadores caracterizados por uma palavra pequena (~12+16bits).

- a hipótese formulada de se usar, em cadeias de controle, chaveamento de algoritmos, utilizando, por exemplo, o "square-root" para as fases transitórias do desenvolvimento do processo e o método de aproximação estocástica para condições de regime, é de interesse para controladores em "time-sharing", onde o máximo de memória rápida disponível é um requisito interessante.
- a substituição de uma simulação que poderia, em princípio, ter sido totalmente feita no computador de grande porte, por uma simulação mais próxima da real onde mini-computador e macro-computador estão conectados tal como controlador e processo, permitiu antecipar problemas de conexão controle-processo, para a fase da simulação, mantendo grande flexibilidade na implementação e testes de "software".
- destaque-se o fato do tempo de amostragem poder ser sensivelmente reduzido, adotando "chips" de operações aritméticas por hardware, pois o maior consumo de tempo do controlador vem do algoritmo de estimação que envolve inúmeros cálculos aritméticos.
- propõe-se como continuação deste trabalho, a realização de um estudo detalhado do uso de redes de mini, micro-processadores, visto ser esse um dos principais problemas industriais da atualidade, consumindo divisas essenciais de um país carente delas como o Brasil.

4.1 - BIBLIOGRAFIA

1. Forés ,J.A. - "Introducción a la estructura de los micro-computadores".Mundo Electrónico, 1976.
2. Zaks ,R.; Steingart ,D. - "Microcomputers : from chips to systems".Sybex, 1976.
3. Barros ,P.M.R. - "The design and performance of a modular CNC System".Thesis for Doctor Philosophy, Victoria University of Manchester.Fevereiro, 1976.
4. Intel 8080, User's Manual. Intel Corporation.

BIBLIOGRAFIA COMPLEMENTAR

- :Borisson ,U. - "Self-tuning regulators-Industrial Application and Multi-variable Theory".Lund Institute of Technology, Dep. Automatic Control, 1975.(Report 7513, outubro 1975).
- :Amaral ,W.C.; Magalhães ,L.P.; Mendes, M.J. - "Identificação de processos, através de diferentes modelos de ruído". SBA, 1976.
- :Amaral, W.C.; Magalhães, L.P.; Mendes, M.J. - Determinação da ordem do modelo".SBPC, 1976.
- :Rosenbrock ,H.H - "State-space and Multivariable Theory". Thomas Nelson and Sons Ltd, 1970. 257 p.
- :PDP-11 Peripherals Handbook. Digital Equipment Corporation. 1975.
- :Components Group. Option Bulletin.Digital Equipment Corporation, agosto 1975.

ANEXOS : Principais listagens dos programas fontes

1. Programas de envio de dados ao PDP-11/05

ENVIA.F4
SUBROTINA PARA ENVIAR DADOS AO 11

SUBROUTINE ENVIA(R)

NUMERO R ENVIAR DEVE SER POSITIVO

NEGA=0
VRIA=R
IF(R)1,2,2
NEGA=1
VRIA=-R

IF(VRIA.GT.9) WRITE(44,44)VRIA
FORMAT(F)
XMAX=9.
VRIA=(32767./XMAX)*VRIA

IV=VRIA
IVRIA=IV
IVRIA=IV+(VRIA-IV)*2.

DIVISAO O NUM. EM DOIS BYTES
BYTE MAIS SIGNIFICATIVO

IVRIA1=IVRIA/256

BYTE MENOS SIGNIFICATIVO

IVRIA2=IVRIA-(IVRIA1*256)

WRITE(60,10)IVRIA,IVRIA1,IVRIA2,NEGA
FORMAT(I)

CARACTER DE AVISO DE INICIO DE TRANSM. 198(10) OU 132(8)

CALL OUT(90)
CALL OUT(IVRIA1)
CALL OUT(IVRIA2)
CALL OUT(NEGA)
RETURN

TESE3.MAC

MACRO PARA ENVIAR DADOS AO 11

ENTRY OUT
UT: JFCL
MOVE 0,0(16)
IONEQU 0
JRA 16,(16)
END

2. Programas de envio de dados ao PDP-10

C ENTRR.F4
C SUBROTINA PARA RECEBER DADOS DO 11

SUBROUTINE ENTRR(XU)
DIMENSION IR(8)

TESTE SE HA' RECEPCAO

CONTINUE
CALL IN(IH,LL)
IF(IH.EQ.FALSE) GOTO 1

E' INICIO DO ENVID DE DADOS ?

IF(LL.NE.89) GOTO 1

RECEPCAO DE DADOS EM ASCII

DO 2 I=1,8

CONTINUE
CALL IN(IH,LL)
IF(IH.EQ.FALSE) GOTO 3
IR(I)=LL
WRITE(40,10)LL
CONTINUE

OBTENCAO DO NUMERO REAL

XU=0
DO 4 I=1,5
XU=XU+(IR(I)-40)*B.**(-I)
FORMAT(6)
IF(IR(6).EQ.49)XU=-XU
XJ=1.
IF(IR(6).EQ.49) XJ=-1.
XU=XU*2.**(XJ*(IR(7)-40))
RETURN

;
;TESES.MAC
;
;MACRO PARA RECEBER DADOS DO 11

ENTRY IN

IN: JFCL
 SETZM 0(16)
 INCHPS @1(16)
 JRST RETU
 SETOM 0(16)
RETU: JRR 16,2(16)
 END

3. Programa principal do simulador no PDP-10

```

C      PROGRAMA PRINCIPAL DO SIMULADOR DO PROCESSO
C
C      IMPRESSAO DE VARIAVEIS
C      XU LIMITADO E ITERACAO FOR48
C      Y      FOR56
C      XU      FOR57
C      CORRY   FOR58
C      CORRXV  FOR58
C      VARIY   FOR61
C      CUSTO   FOR62
C      RUINO LEITRA DO FOR59

      DIMENSION Y(-2/2000), XU(-2, 2000), CORRY(11), CORRXV(11)
      Y(-2)=0.
      Y(-1)=0.
      XU(-3)=0.
      XU(-2)=0.
      XU(-1)=0.

      NHED=1000
      DO 1 L=0, NHED

      READ(59, 2) GE
      CTE1=-1.6*(-Y(L-1))+0.8*(-Y(L-2))
      Y(L)=1.6*XU(L-2)+0.5*XU(L-3)+GE+CTE1

      A=Y(L)

C      ENVIAR DADO AO CONTROLADOR
      CALL ENVIA(A)

C      RECEBE DADO DO CONTROLADOR
      CALL ENTRA(XU1)

      XU(L)=XU1
      WRITE(56, 2)Y(L)
      WRITE(57, 2)XU1
48      FORMAT(3X, F, I)
      IF(XU1.EQ. 11. OR. XU1.EQ. -11)WRITE(48, 48)XU1, L
      IF(L.LT. 20) GOTO 1020
      DO 3 I=1, 11
      CORRY(I)=CORRY(I)+Y(L)*Y(L-I+1)
3      CORRXV(I)CORRXV(I)+XU(L)*Y(L-I+1)

1020      CUSTO=CUSTO+Y(L)**2.
      WRITE(62, 2)CUSTO

1      CONTINUE

      DO 4 I=2, 11
      CORRY(I)=CORRY(I)/CORRY(1)
4      CORRXV(I)=CORRXV(I)/CORRXV(1)
      CORRY(1)=1.
      CORRXV(1)=1.

2      FORMAT(6)
      WRITE(58, 2)(CORRY(I), I=1, 11)
      WRITE(60, 2)(CORRXV(I), I=1, 11)

      VARIY=CUSTO/NHED
      WRITE(61, 2)VARIY
      STOP
      END

```

4. Programa de estimação utilizando o método da raiz quadrada

```
;PROGRAMA DE CONTROLE UTILIZANDO
; O METODO DE 'SQUARE ROOT' NA
; ESTIMACAO
```

```
%=1000
```

```
;DISPLAY DA ITERACAO ATUAL
```

```
WORD 116400
WORD 100
WORD 500
WORD 100000
WORD 0
WORD 0
WORD 0
WORD 0
WORD 100000
WORD 1000
```

```
;ENVIO DE EXECUTE AO 10
```

```
INI:  MOV    #1000,172000
      JSR     %5,EXE
```

```
;ATUALIZACAO DO SINAL DE SAIDA DO PROCESSO
```

```
RET0: JSR     %5,AD
      MOV     YM7H,YM3M
      MOV     YM2E,YM3E
      MOV     YM1M,YM2M
      MOV     YM1E,YM2E
      MOV     YM0M,YM1M
      MOV     YM0E,YM1E
      MOV     %7,YM0M
      MOV     %1,YM0E
      JSR     %5,GRAPH
      JSR     %5,QUADRA
      BR      RET0
```

```
; SEQUENCIAS
```

```
PM1:  MOV     FJM,%0
      MOV     FJE,%1
      MOV     SIGSQM,%2
      MOV     SIGSQE,%3
      JSR     %5,DIVIS
      MOV     %0,BM
      MOV     %1,BE
DOIS: MOV     SIGM,AM
      MOV     SIGE,AE
TRES: MOV     FJM,%0
      MOV     FJE,%1
      MOV     %0,%2
      MOV     %1,%3
      JSR     %5,MULTI
      MOV     SIGSQM,%2
      MOV     SIGSQE,%3
      JSR     %5,SOMA
      MOV     %0,SIGSQM
```

```

      MOV      %1,SIGSOE
0TRP: MOV      #761,0*RAIZ3
      JSR      %5,RAIZ
      MOV      %7,SIGM
      MOV      %1,SIGE
CNCD: MOV      %2,%2
      MOV      %1,%3
      MOV      AM,%7
      MOV      AE,%1
      JSR      %5,DIVIS
      MOV      %0,AM
      MOV      %1,AE
      RTS      %5

```

;SUBROTINA DO DISPLAY

```

GRAFI1: MOV      #NUM,%7
      INC      GRAFI1
      CMP      GRAFI1,#72
      BNE      GR1
      MOV      #64,GRAFI1
      INC      GRAFI2
      CMP      GRAFI2,#72
      BNE      GR1
      MOV      #64,GRAFI2
      INC      GRAFI3
      CMP      #72,GRAFI3
      BNE      GR1
      MOV      #64,GRAFI3
      INC      GRAFI4
GR1:  MOV      GRAFI4, (%7)+
      MOV      GRAFI3, (%7)+
      MOV      GRAFI2, (%7)+
      MOV      GRAFI1, (%7)+
      RTS      %5

```

;ALGORITMO DE CONTROLE E ESTIMACAO

```

;ESTIMACAO
;PEER=YM*UXUM1

```

```

QUADRA: MOV      YMAH,%7
      MOV      YME,%1
      MOV      XUM2H,%2
      MOV      XUM2E,%3
      JSR      %5,SUBT
      MOV      %7,AUXH
      MOV      %1,AUXE

```

```

;PEER=PEER-TETA(1)*U(1)

```

```

      MOV      TETA1H,%7
      MOV      TETA1E,%1
      MOV      YH2H,%2
      MOV      YH2E,%3
      JSR      %5,MULTI
      MOV      AUXH,%2
      MOV      AUXE,%1
      JSR      %5,SONA

```


MOV %0,AUXM
MOV %1,AUXE

!PEPD=PEPR-TETA(2)*U(2)

MOV TETA2M,%0
MOV TETA2E,%1
MOV YM3M,%2
MOV YM3E,%3
JSR %5,MULTI
MOV AUXM,%2
MOV AUXE,%3
JSR %5,SOMA
MOV %0,AUXM
MOV %1,AUXE

!PERA=PEPR-TETA(3)*U(3)

MOV TETA3M,%0
MOV TETA3E,%1
MOV XUM3M,%2
MOV XUM3E,%3
JSR %5,MULTI
MOV %0,%2
MOV %1,%3
MOV AUXM,%0
MOV AUXE,%1
JSR %5,SUBT
MOV %0,AUXM
MOV %1,AUXE

!PERB=PEPR-TETA(4)*U(4)

MOV TETA4M,%0
MOV TETA4E,%1
MOV XUM4M,%2
MOV XUM4E,%3
JSR %5,MULTI
MOV %0,%2
MOV %1,%3
MOV AUXM,%0
MOV AUXE,%1
JSR %5,SUBT
MOV %0,PERRM
MOV %1,PERRE

!FJ=S(1)*U(1)

MOV S1M,%0
MOV S1E,%1
MOV YM2M,%2
MOV YM2E,%3
JSR %5,MULTI
CMP #100000,%0
BEQ FJ1
CMP #40000,%0
BEQ FJ2
NEG
BR FJ3
MOV #40000,%0

```

112: INC      %1
      BR      FJ3
      MOV     #1000000,%2
      DEC     %1
113: MOV     %2,FJM
      MOV     %1,FJE

```

;SECUENCIA 1

```

MOV     #400000,SIGSQM
MOV     #1,SIGSQE
MOV     #500000,AM
MOV     #1,AE
JSR     %5,TRES

```

;G(1)=S(1)*FJ

```

MOV     S1M,%2
MOV     S1E,%1
MOV     FJM,%2
MOV     FJE,%3
JSR     %5,MULTI
MOV     %2,G1M
MOV     %1,G1E

```

;S(1)=A*5(1)

```

MOV     AM,%2
MOV     AE,%1
MOV     S1M,%2
MOV     S1E,%3
JSR     %5,MULTI
MOV     %2,S1M
MOV     %1,S1E

```

;FJ=S(2)*U(1)+S(3)*U(2)

```

MOV     S2M,%2
MOV     S2E,%1
MOV     YN2M,%2
MOV     YM2E,%3
JSR     %5,MULTI
MOV     %2,AUXM
MOV     %1,AUXE
MOV     S3M,%2
MOV     S3E,%1
MOV     YM3M,%2
MOV     YM3E,%3
JSR     %5,MULTI
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,SOMA
CMP     #1000000,%2
BEQ     FJ4
CMP     #400000,%2
BEQ     FJ5
NEG     %2
BR      FJ6
114: MOV     #400000,%2
      INC     %1

```

```

      BR      FJ6
{ JS:  MOV     #1*88889,%0
      DEC     %1
{ JS:  MOV     %0,FJM
      MOV     %1,FJE

```

;SECUENCIA 2

```
JSR     %5,UM
```

;G(2)=S(3)*FJ

```

MOV     FJM,%0
MOV     FJE,%1
MOV     S3M,%2
MOV     S3E,%3
JSR     %5,MULTI
MOV     %0,C2M
MOV     %1,C2E

```

;S(3)=A*S(3)

```

MOV     S3M,%0
MOV     S3E,%1
MOV     AM,%2
MOV     AE,%3
JSR     %5,MULTI
MOV     %0,S3M
MOV     %1,S3E

```

;AUX=S(2)

```

MOV     S2M,AUXM
MOV     S2E,AUXE

```

;S(2)=A*(S(2)+B*C(1))

```

MOV     BM,%0
MOV     BE,%1
MOV     G1M,%2
MOV     G1E,%3
JSR     %5,MULTI
MOV     %0,%2
MOV     %1,%3
MOV     S2M,%0
MOV     S2E,%1
JSR     %5,SUBT
MOV     AM,%2
MOV     AE,%3
JSR     %5,MULTI
MOV     %0,S2M
MOV     %1,S2E

```

;G(1)=G(1)+AUX*FJ

```

MOV     AUXM,%0
MOV     AUXE,%1
MOV     FJM,%2
MOV     FJE,%3
JSR     %5,MULTI

```

```

MOV    G1M,%2
MOV    G1E,%3
JSR    %5,SOMA
MOV    %2,G1M
MOV    %1,G1E

```

;FJ=S(4)*U(1)+S(5)*U(2)+S(6)*U(3)

```

MOV    S4M,%2
MOV    S4E,%1
MOV    YM2M,%2
MOV    YM2E,%3
JSR    %5,MULTI
MOV    %2,AUXM
MOV    %1,AUXE
MOV    S5M,%2
MOV    S5E,%1
MOV    YM3M,%2
MOV    YM3E,%3
JSR    %5,MULTI
MOV    AUXM,%2
MOV    AUXE,%3
JSR    %5,SOMA
MOV    %2,AUXM
MOV    %1,AUXE
MOV    S6M,%2
MOV    S6E,%1
MOV    XUM3M,%2
MOV    XUM3E,%3
JSR    %5,MULTI
MOV    AUXM,%2
MOV    AUXE,%3
JSR    %5,SUBT
MOV    %2,FJM
MOV    %1,FJE

```

;SEQUENCIA 3

```
JSR    %5,UM
```

;G(3)=S(6)*FJ

```

MOV    S6M,%2
MOV    S6E,%1
MOV    FJM,%2
MOV    FJE,%3
JSR    %5,MULTI
MOV    %2,G3M
MOV    %1,G3E

```

;S(6)=S(6)*A

```

MOV    AM,%2
MOV    AE,%1
MOV    S6M,%2
MOV    S6E,%1
JSR    %5,MULTI
MOV    %2,S6M
MOV    %1,S6E

```

!AUX=S4

MOV S4M,AUXM
MOV S4E,AUXE

!S(4)=A*(AUX*B*G(1))

MOV G1M,%2
MOV G1E,%1
MOV B2,%2
MOV BE,%3
JSR %5,MULTI
MOV %2,%2
MOV %1,%3
MOV AUXM,%2
MOV AUXE,%1
JSR %5,SUBT
MOV AM,%2
MOV AE,%3
JSR %5,MULTI
MOV %2,S4M
MOV %1,S4E

!G(1)=G(1)+AUX*FJ

MOV AUXM,%2
MOV AUXE,%1
MOV FJM,%2
MOV FJE,%3
JSR %5,MULTI
MOV G1M,%2
MOV G1E,%3
JSR %5,SOMA
MOV %2,G1M
MOV %1,G1E

!AUX=S(5)

MOV S5M,AUXM
MOV S5E,AUXE

!S(5)=A*(AUX*B*G(2))

MOV B2,%2
MOV BE,%1
MOV G2M,%2
MOV G2E,%3
JSR %5,MULTI
MOV %2,%2
MOV %1,%3
MOV AUXM,%2
MOV AUXE,%1
JSR %5,SUBT
MOV AM,%2
MOV AE,%3
JSR %5,MULTI
MOV %2,S5M
MOV %1,S5E

;G(2)=G(2)+AUX*FJ

```
MOV    AUXM,%2
MOV    AUXE,%1
MOV    FJM,%2
MOV    FJE,%1
JSR    %5,MULTI
MOV    G2M,%2
MOV    G2E,%3
JSR    %5,SOMA
MOV    %0,G2M
MOV    %1,G2E
```

;FJ=S(7)*U(1)+S(8)*U(2)+S(9)*U(3)+S(10)*U(4)

```
MOV    S7M,%2
MOV    S7E,%1
MOV    YM7M,%2
MOV    YM7E,%3
JSR    %5,MULTI
MOV    %0,AUXM
MOV    %1,AUXE
MOV    S8M,%0
MOV    S8E,%1
MOV    YM8M,%2
MOV    YM8E,%3
JSR    %5,MULTI
MOV    AUXM,%2
MOV    AUXE,%3
JSR    %5,SOMA
MOV    %0,AUXM
MOV    %1,AUXE
MOV    S9M,%0
MOV    S9E,%1
MOV    XUM3M,%2
MOV    XUM3E,%3
JSR    %5,MULTI
MOV    AUXM,%2
MOV    AUXE,%3
JSR    %5,SUBP
MOV    %0,AUXM
MOV    %1,AUXE
MOV    S10M,%0
MOV    S10E,%1
MOV    XUM4M,%2
MOV    XUM4E,%3
JSR    %5,MULTI
MOV    AUXM,%2
MOV    AUXE,%3
JSR    %5,SOMA
MOV    %0,FJM
MOV    %1,FJE
```

;SEQUENCIA 4

JSR %5,UM

;G(4)=S(10)*FJ

```
MOV    S10M,%0
MOV    S10E,%1
```

```

MOV    FJM,%2
MOV    FJE,%3
JSR    %5,MULTI
MOV    %9,G4M
MOV    %1,G4E

```

$I5(I^2)=A*S(I^2)$

```

MOV    S1M,%2
MOV    S1E,%1
MOV    AM,%2
MOV    AE,%3
JSR    %5,MULTI
MOV    %9,S1M
MOV    %1,S1E

```

$I AUX=S(7)$

```

MOV    S7M,AUXM
MOV    S7E,AUXE

```

$I3(7)=A*(AUX*B*G(1))$

```

MOV    BM,%2
MOV    BE,%1
MOV    G1M,%2
MOV    G1E,%3
JSR    %5,MULTI
MOV    %7,%2
MOV    %1,%3
MOV    AUXM,%2
MOV    AUXE,%1
JSR    %5,SOMA
MOV    AM,%2
MOV    AE,%3
JSR    %5,MULTI
MOV    %9,S7M
MOV    %1,S7E

```

$I G(1)=G(1)+AUX*FJ$

```

MOV    AUXM,%2
MOV    AUXE,%1
MOV    FJM,%2
MOV    FJE,%3
JSR    %5,MULTI
MOV    G1M,%2
MOV    G1E,%3
JSR    %5,SOMA
MOV    %9,G1M
MOV    %1,G1E

```

$I AUX=S(8)$

```

MOV    S8M,AUXM
MOV    S8E,AUXE

```

$I5(8)=A*(AUX*B*G(2))$

```

MOV    G2M,%2

```

```

MOV     G2E,%1
MOV     BM,%2
MOV     BE,%3
JSR     %5,MULTI
MOV     %4,%2
MOV     %1,%3
MOV     AUXM,%4
MOV     AUXE,%1
JSR     %5,SUBT
MOV     AM,%2
MOV     AE,%3
JSR     %5,MULTI
MOV     %4,S9M
MOV     %1,S9E

```

;G(2)=G(2)+AUX*FJ

```

MOV     FJM,%4
MOV     FJE,%1
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,MULTI
MOV     G2M,%2
MOV     G2E,%3
JSR     %5,SOMA
MOV     %4,G2M
MOV     %1,G2E

```

;AUX=S(9)

```

MOV     S9M,AUXM
MOV     S9E,AUXE

```

;S(9)=A*(AUX*B*G(2))

```

MOV     G2M,%4
MOV     G2E,%1
MOV     BM,%2
MOV     BE,%3
JSR     %5,MULTI
MOV     %4,%2
MOV     %1,%3
MOV     AUXM,%4
MOV     AUXE,%1
JSR     %5,SUBT
MOV     AM,%2
MOV     AE,%3
JSR     %5,MULTI
MOV     %4,S9M
MOV     %1,S9E

```

;G(3)=G(3)+AUX*FJ

```

MOV     AUXM,%4
MOV     AUXE,%1
MOV     FJM,%2
MOV     FJE,%3
JSR     %5,MULTI
MOV     G3M,%2

```



```

MOV     G3E,%3
JSR     %5,SOMA
MOV     %2,G3M
MOV     %1,G3E

```

```

;TETA(I)=TETA(I)+G(I)*AUX
;AUX=PERP/SIGSQ

```

```

MOV     PERPM,%2
MOV     PERPE,%1
MOV     SIGSQM,%2
MOV     SIGSQE,%3
JSR     %5,DIVIS
MOV     %2,AUXM
MOV     %1,AUXE

```

```

;TETA(1)

```

```

MOV     G1M,%2
MOV     G1E,%1
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,MULTI
MOV     TETA1M,%2
MOV     TETA1E,%3
JSR     %5,SOMA
MOV     %2,TETA1M
MOV     %1,TETA1E

```

```

;TETA(2)

```

```

MOV     G2M,%2
MOV     G2E,%1
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,MULTI
MOV     TETA2M,%2
MOV     TETA2E,%3
JSR     %5,SOMA
MOV     %2,TETA2M
MOV     %1,TETA2E

```

```

;TETA(3)

```

```

MOV     G3M,%2
MOV     G3E,%1
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,MULTI
MOV     TETA3M,%2
MOV     TETA3E,%3
JSR     %5,SOMA
MOV     %2,TETA3M
MOV     %1,TETA3E

```

```

;TETA(4)

```

```

MOV     G4M,%2
MOV     G4E,%1
MOV     AUXM,%2

```

```

MOV     AUXE,%3
JSR     %5,MULTI
MOV     TETA4M,%2
MOV     TETA4E,%3
JSR     %5,SOMA
MOV     %0,TETA4M
MOV     %1,TETA4E

```

: CALCULO DO CONTROLE

; ATUALIZACAO DOS XU

```

MOV     XUM3M,XUM4M
MOV     XUM3E,XUM4E
MOV     XUM2M,XUM3M
MOV     XUM2E,XUM3E
MOV     XUM1M,XUM2M
MOV     XUM1E,XUM2E
MOV     %0,XUM1M
MOV     %1,XUM1E

```

; ENVIO DE XU ATUAL

```

JSR     %5,DA

```

```

RTS     %5

```

5. Programa de estimação utilizando o método de aproximação estocástica

;PROGRAMA DE APROXIM. ESTOCASTICA NO 11

DE=1000

;DISPLAY DA ITERACAO ATUAL

ENWORD 116400
ENWORD 100
ENWORD 1360
ENWORD 100000
ENWORD 0
ENWORD 0
ENWORD 0
ENWORD 0
ENWORD 100000
ENWORD 1000

;ENVIO DE EXECUTE AO 10

```
INI:  MOV    #1000,172000
      JSR    %5,EXE
      ;ATUALIZACAO DO SINAL DE SAIDA DO PROCESSO
RETU: JSR    %5,AD
      MOV    YM2M,YM3M
      MOV    YM2E,YM3E
      MOV    YM1M,YM2M
      MOV    YM1E,YM2E
      MOV    YMMM,YM1M
      MOV    YMME,YM1E
      MOV    %0,YMMM
      MOV    %1,YMME
      JSR    %5,GRAFI
      JSR    %5,ALGOR
      BR     RETU
```

;SUBROTINA DE DISPLAY

```
GRAFI1: MOV    #NUM,%0
      INC     GRAFI1
      CMP     #72,GRAFI1
      BNE     GR1
      MOV     #50,GRAFI1
      INC     GRAFI1
      CMP     GRAFI1,#72
      BNE     GR1
      MOV     #60,GRAFI1
      INC     GRAFI1
      CMP     #72,GRAFI1
      BNE     GR1
      MOV     #60,GRAFI1
      INC     GRAFI1
      MOV     GRAFI4,(%0)+
      MOV     GRAFI3,(%0)+
      MOV     GRAFI2,(%0)+
      MOV     GRAFI1,(%0)+
      RTS     %5
GR1:   MOV     GRAFI4,(%0)+
      MOV     GRAFI3,(%0)+
      MOV     GRAFI2,(%0)+
      MOV     GRAFI1,(%0)+
      RTS     %5
```

ALGORITMO DE CONTROLE E ESTIMACAO

ESTIMACAO

AUX=U(1)**2

```
ALGOR:  MOV     YM2M,%0
        MOV     YM2E,%1
        MOV     %0,%2
        MOV     %1,%3
        JSR     %5,MULTI
        MOV     %2,AUXM
        MOV     %1,AUXE
```

U(2)**2+AUX=AUX

```
MOV     YM3M,%0
MOV     YM3E,%1
MOV     %0,%2
MOV     %1,%3
JSR     %5,MULTI
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,SOMA
MOV     %0,AUXM
MOV     %1,AUXE
```

U(3)**2+AUX=AUX

```
MOV     XUM3M,%0
MOV     XUM3E,%1
MOV     %0,%2
MOV     %1,%3
JSR     %5,MULTI
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,SOMA
MOV     %0,AUXM
MOV     %1,AUXE
```

U(4)**2+AUX=AUX

```
MOV     XUM4M,%0
MOV     XUM4E,%1
MOV     %0,%2
MOV     %1,%3
JSR     %5,MULTI
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,SOMA
MOV     %0,AUXM
MOV     %1,AUXE
```

AUXL1=AUXL1*RHO+AUX

```
MOV     RHOH,%0
MOV     RHOE,%1
MOV     AUXL1M,%2
MOV     AUXL1E,%3
JSR     %5,MULTI
MOV     AUXM,%4
MOV     AUXE,%3
JSR     %5,SOMA
```

```
MOV    %0,AUXL1M
MOV    %1,AUXL1E
```

```
; GAMMA=12/AUXL1
```

```
INC    ELE
CMP    ELE,#5
BPL    ELE1
CLR    %0
CLR    %1
BR     ELE2
```

```
ELE1:  CMP    ELE,#16#4                      ; L GT 900 ; RHO=1
```

```
BNE    ELE3
MOV    #42222,RHOM
MOV    #1,RHOM
```

```
ELE3:  MOV    %0,%2
MOV    %1,%3
MOV    #42222,%0
MOV    #1,%1
```

```
ELE2:  JSR    %5,DIVIS
MOV    %0,GAMMA
MOV    %1,GAMMAE
```

```
; AUX=TETA(1)*U(1)+TETA(2)*U(2)+TETA(3)*U(3)+TETA(4)*U(4)
```

```
; AUX=TETA(1)*U(1)
```

```
MOV    TETA1M,%0
MOV    TETA1E,%1
MOV    YM2M,%2
MOV    YM2E,%3
JSR    %5,MULTI
MOV    %0,AUXM
MOV    %1,AUXE
```

```
; AUX=AUX+TETA(2)*U(2)
```

```
MOV    YM3M,%0
MOV    YM3E,%1
MOV    TETA2M,%2
MOV    TETA2E,%3
JSR    %5,MULTI
MOV    AUXM,%0
MOV    AUXE,%1
JSR    %5,SOMA
MOV    %0,AUXM
MOV    %1,AUXE
```

```
; AUX=AUX+TETA(3)*U(3)
```

```
MOV    XOM3M,%0
MOV    XOM3E,%1
MOV    TETA3M,%2
MOV    TETA3E,%3
JSR    %5,MULTI
MOV    AUXM,%0
MOV    AUXE,%1
JSR    %5,SUBT
MOV    %0,AUXM
MOV    %1,AUXE
```

```
; AUX=AUX+TETA(4)*U(4)
```

```

MOV     XUM4M,%0
MOV     XUM4E,%1
MOV     TETA4M,%2
MOV     TETA4E,%3
JSR     %5,MULTI
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,SOMA

```

```

; AUX+XU(L-2)
MOV     XUM2M,%2
MOV     XUM2E,%3
JSR     %5,SOMA

```

```

; Y(L)=(AUX+XU(L-2))*EPSILON
MOV     %0,%2
MOV     %1,%3
MOV     YM0M,%0
MOV     YM0E,%1
JSR     %5,SUBT

```

```

; AUX=GAMA*EPSILON
MOV     GAMAM,%2
MOV     GAMAE,%3
JSR     %5,MULTI
MOV     %0,AUXM
MOV     %1,AUXE

```

```

; ATUALIZACAO DE TETA
; TETA(1)=TETA(1)+AUX*U(1)
MOV     YM2M,%2
MOV     YM2E,%3
JSR     %5,MULTI
MOV     %0,%2
MOV     %1,%3
MOV     TETA1M,%0
MOV     TETA1E,%1
JSR     %5,SUBT
MOV     %0,TETA1M
MOV     %1,TETA1E

```

```

; TETA(2)=TETA(2)+AUX*U(2)
MOV     YM3M,%0
MOV     YM3E,%1
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,MULTI
MOV     %0,%2
MOV     %1,%3
MOV     TETA2M,%0
MOV     TETA2E,%1
JSR     %5,SUBT
MOV     %0,TETA2M
MOV     %1,TETA2E

```

```

; TETA(3)=TETA(3)+AUX*U(3)
MOV     XUM3M,%0
MOV     XUM3E,%1
MOV     AUXM,%2
MOV     AUXE,%3

```

```

JSR    %5,MULTI
MOV    TETA3M,%2
MOV    TETA3E,%3
JSR    %5,SOMA
MOV    %0,TETA3M
MOV    %1,TETA3E

```

; TETA(4)=TETA(4)+AUX*U(4)

```

MOV    XUM4M,%0
MOV    XUM4E,%1
MOV    AUXM,%2
MOV    AUXE,%3
JSR    %5,MULTI
MOV    TETA4M,%2
MOV    TETA4E,%3
JSR    %5,SOMA
MOV    %0,TETA4M
MOV    %1,TETA4E

```

6. Programa de controle

```

; CALCULO DO CONTROLE
; AUX=TETA(3)*XU(L=1)+TETA(4)*XU(L=2)
MOV     TETA3M,%0
MOV     TETA3E,%1
MOV     XUM1M,%2
MOV     XUM1E,%3
JSR     %5,MULT1
MOV     %0,AUXM
MOV     %1,AUXE
MOV     TETA4M,%0
MOV     TETA4E,%1
MOV     XUM2M,%2
MOV     XUM2E,%3
JSR     %5,MULT1
MOV     AUXM,%4
MOV     AUXE,%5
JSR     %5,SOMA
MOV     %0,AUXM
MOV     %1,AUXE

; XU1=TETA(1)*Y(L)+TETA(2)*Y(L=1)+AUX
MOV     TETA1M,%0
MOV     TETA1E,%1
MOV     YMM,%2
MOV     YME,%3
JSR     %5,MULT1
MOV     AUXM,%4
MOV     AUXE,%5
JSR     %5,SUBT
MOV     %0,AUXM
MOV     %1,AUXE
MOV     YM1M,%0
MOV     YM1E,%1
MOV     TETA2M,%2
MOV     TETA2E,%3
JSR     %5,MULT1
MOV     AUXM,%2
MOV     AUXE,%3
JSR     %5,SOMA

; LIMITACAO
JSR     %5,LIMITA
;SUBROTINA PARA LIMITACAO DO SINAL DE CONTROLE

```

```

LIMITA: TST     %0
BEQ     NAOLIM
CMP     %1,%4
BMI     NAOLIM
BEQ     TSTLIM

```

;LIMITACAO EXPO MAIOR QUE 4

```

TST     %0
BPL     LIMPOS
LIMNEG: MOV     #130000,%0
LIM1:   MOV     #4,%1
NAOLIM: RTS
LIMPOS: MOV     #50000,%0
BR      LIM1

```

;EXPO E' 4 , TESTA MANTISSA

```

TSTLIM: TST     %0
BPL     LIM2
CMP     %0,#130000
BMI     LIMNEG
RTS     %0
LIM2:   CMP     %0,#50000
BPL     LIMPOS
RTS     %0

```


7. Subrotinas aritméticas e subrotinas de entrada/saída

; SUBROTINAS ARITMÉTICAS

;RAIZ QUADRADA

;R0 MANTISSA DA ENTRADA E SAIDA

;R1 EXPOENTE DA ENTRADA E SAIDA

```

RAIZ:  CMP      R1,R0
      BMI      RAIZ4
      MOV      #4,R0,R0
      MOV      #2,R0
      DEC      R1
      BR       RAIZ5

RAIZ4: MOV      #4,R0,R0
      MOV      #1,R0
RAIZ5: JCR      R5,SOMA
      DEC      R1
      BR       RAIZ1
RAIZ2: MOV      #5,R0,R0
RAIZ1: MOV      R4,R0
      MOV      R1,R0
      MOV      SIGSQ,R0
      MOV      SIGSQ,R1
      JCR      R5,DIVIS
      JCR      R5,SOMA
      DEC      R1
RAIZ3: BR       RAIZ2
  
```

;SOMA DE DOIS NUM. USANDO MANTISSA E EXPO

;A=A+B

;A: MANTISSA EM R0 E EXPO EM R1

;B: MANTISSA EM R2 E EXPO EM R3

;R4 E AUX0 USADOS COMO AUXILIARES

```

SOMA:  CLR      AUX0
      ; ALGUM OPERANDO É ZERO ?
      TST      R0
      BNE      SOMA17
      MOV      R2,R0
      MOV      R3,R1
      RTS      R5
SOMA10: TST      R2
      BNE      SOMA11
      RTS      R5
      ; ACERTOS DE EXPOENTES
SOMA11: CMP      R1,R3
      BEQ      SOMA1
  
```

```

BPL      SOMA2      ;EXPO %1 E' MAIOR
INC      %1         ;EXPO %3 E' MAIOR
ASH      %%
BR       SOMA11
SOMA2:   INC      %3      ;EXPO %1 E' MAIOR
ASH      %2
BR       SOMA11
;VERIFICACAO DOS SINAIS DOS NUMEROS
SOMA1:   TST      %%
BPL      SOMA8
INC      AUX%
SOMA8:   TST      %2
BPL      S..MA9
INC      AUX%
SOMA9:   BIT      #1,AUX%
BEQ      S..MA3
;NUMEROS DE SINAIS DIFERENTES
ADD      %2,%2      ;NOS DE SINAIS DIFERENTES
BNE      SOMA4%
CLR      %1
RTS      %3
SOMA4:   ASL      %2
DEC      %1
SOMA4:   TST      %%
BNI      SOMA5
BIT      #4%%,%2      ;BIT 14 E' 1 7
BEQ      SOMA4      ;NAO 1
RTS      %5
SOMA5:   BIT      #4%%,%2      ;BIT 14 E' 0 2
BNE      SOMA4      ;NAO 1
RTS      %5
;NUMEROS DE SINAIS IGUAIS
SOMA3:   ADD      %2,%2
BCC      SOMA3%
SOMA31:  BNI      SOMA32
SOMA33:  ROR      %%
INC      %1
SOMA32:  RTS      %5
SOMA32:  TST      %%
BNI      SOMA33
RTS      %5

;SUBTRACAO USANDO ADICAO
;A=A-B=A+(-B)
;A: MANT% %% E EXPO %1
;B: MANT% %2 E EXPO %3
SUBT:    CMP      #1%%,%2      ;TST SE NUM. E' 100000, POIS
                                ;ESTE NUM. TEM COMPLES ESPECIAL
BEQ      SUBT1
CMP      #4%%,%2
BEQ      SUBT2
TST      %2
BEQ      SUBT6
NEG      %2
SUBT3:   JSR      %5,SOMA
SUBT6:   RTS      %5
SUBT1:   MOV      #4%%,%2
INC      %3

```

```

SUBT2: BR      SUBT3
      MOV      #1000000,%2
      DEC      %2
      BR      SUBT3

```

```

;MULTIPLICACAO DE INTEIROS
;NAO GARANTE SALVACAO DOS OPERANDOS
;%0 AUXILIAR
;%1 CONTADOR
;%2 * %3 = %4,%5

```

```

MULT:  CLR      %4
      CLR      %5
      TST      %3
      BEQ      M3
      MOV      #16,%1
M1:    BIT      #1,%3
      BEQ      M1
      ADD      %2,%4
M1:    ROR      %4
      BCC      M2
      MOV      #40000,%2
M2:    ASR      %3
      BTC      #10000,%3
      BIS      %2,%3
      CLR      %7
      DEC      %1
      BPL      M1
M3:    RTS      %5

```

```

;MULTIPLICACAO DE REAIS
;A=A*B
;A: MANTENHA EM %0 E EXPO EM %1
;B: MANTENHA EM %2 E EXPO EM %3
;AUX0 DA O SINAL DO RESULTADO; AUX0 PAR = , IMPAR -

```

```

;OPERANDOS SAO FEITOS POSITIVOS
MULTI: CLR      AUX0
      TST      %0
      BEQ      FIMM1
      BPL      MULTI1
      INC      AUX0
      NEG      %0
      CMP      #1000000,%0
      BNE      MULTI1
      MOV      #40000,%0
      INC      %1
MULTI1: TST      %2
      BNE      FIMM2
      CLR      %0
FIMM1:  CLR      %1
      RTS      %5
FIMM2:  BPL      MULTI3
      INC      AUX0
      NEG      %2
      CMP      #1000000,%2
      BNE      MULTI3
      MOV      #40000,%2
      INC      %3
MULTI3: MOV      %1,AUX1

```

```

MOV    %3,AUX2
MOV    %2,%3
JSR    %5,MULT
MOV    %4,%2
MOV    AUX1,%1
ADD    AUX2,%1

```

TESTE DE RESULTADO ZERO

```

TST    %2
BNE    MOD4
TST    %3
BNE    MOD4
CLR    %1
RTS    %5

```

```

MOD4:  DEC    %1
        ASL    %2
        BMI    MOD2
        ASL    %3
        BPL    MOD4
        INC    %2
        BR     MOD4

```

```

MOD2:  ROR    %2
        INC    %1

```

TESTE DO EXPOENTE

```

CMP    #177,%1          ;TST DE OVERFLOW DE EXPO
BPL    MULT15
HALT

```

```

MULT15: CMP    #177600,%1      ;TST DE UNDERFLOW DE EXPO
        BMI    MULT14
        HALT

```

```

MULT14: BIT    #1,AUX2          ;AUX2 E' IMPAR?
        BNE    MULT19          ;SIM! PULA

```

```

MOD1:  MOV    AUX2,%3
        RTS    %5

```

```

MULT19: CMP    #40000,%2
        BEQ    MOD3
        NEG    %2
        BR     MOD1

```

```

MOD3:  MOV    #100000,%2
        DEC    %1
        BR     MOD1

```

DIVISAO ENTRE INTEIROS

%2/%2=%3 E RESTO EM %2

%1: EXPO A SOMAR AO EXPO DO RESULT.

```

DIVIDA: CLR    %3
        MOV    #1,%4
MAIS3:  CMP    %2,%2
        BPL    MAIS1
        ASL    %2
        CLR    %4

```

```

MAIS1:  SUB    %2,%2
        INC    %3
        ASL    %3
        BMI    MAIS4
        CMP    %2,%2
        BPL    MAIS1

```

```

MAIS6:  ASL    %2

```

```

CMP      %2,%2
BPL      MAIS1
ASL      %3
BPL      MAIS6
MAIS4:   ROR      %3
RTS      %5

;DIVISAO DE REAIS USANDO DIVIDA COMO AUXILIAR
;A=A/B  A: MANT, %2 E EXPO %1
;      B: MANT, %2 E EXPO %3
;AUXILIO  AUX0,1,2,3
DIVIS4:  MOV      %2,AUX1      ;SALVA DIVISOR
MOV      %3,AUX2

TST      %2
BNE      DIVIS4
CLR      %1
MOV      AUX1,%2
MOV      AUX2,%3
RTS      %5

;TESTA SE DIVISOR E' ZERO
DIVIS0:  TST      %2
BNE      DIVIS7
HALT

DIVIS7:  CLR      AUX0
TST      %2
BPL      DIVIS1
NEG      %4
INC      AUX0      ;AUX0 INDICARA SE RESP. E' NEG.
CMP      #100000,%2
BNE      DIVIS1
MOV      #40000,%2
INC      %1
DIVIS1:  TST      %2
BPL      DIVIS3
NEG      %2
INC      AUX0
CMP      #100000,%2
BNE      DIVIS3
MOV      #40000,%2
INC      %3
;AMBOS NUMS POSITIVOS E NORMAIS
;DECIDIMOS ENTAO SOBRE EXPO DA RESP.
DIVIS3:  SUB      %3,%1
CMP      #177,%1
BPL      DIVIS5
HALT
DIVIS5:  CMP      #177600,%1
BMI      DIVIS4
HALT      ;HOVE UNDERFLOW
DIVIS4:  JSR      %5,DIVIDA
MOV      %3,%2
ADD      %4,%1
MOV      AUX2,%3
MOV      AUX1,%2
BIT      #1,AUX0
BEQ      DIVIS6
DIVIS2:  CMP      #40000,%2

```

```

BNE     MOD11
MOV     #12345678,%R
DEC     %R
RTS     %R
MOD11:  NEG     %R
DIVIS6: RTS     %R

```

```

;CONVERSOR A/D
;COMUNICACAO 1º PARA 11
;R2 NUMº QUE VEM DO 1º
;AUX SINAL DESSE NUMº
;R0 MANTISSA DA RESPº
;R1 EXPO DA RESPº

```

```

AD2:    CLR     AUX
        CLR     %R2

```

```

AD1:    ;ESPERA SINAL INICIO DE TRANSMISSÃO(19)
        TSTB    175612
        BPL     AD1
        CMPB    #132,175612
        BNE     AD1

```

```

AD2:    ;RECEBE PARTE MAIS SIGN.
        TSTB    175612
        BPL     AD2
        MOVB    175612,%R2
        SWAB    %R2

```

```

AD3:    ;RECEBE PARTE MENOS SIGN.
        TSTB    175612
        BPL     AD3
        BIC     #255,175612
        BIS     175612,%R2

```

```

AD4:    ;RECEBE SINAL DO NUMº
        TSTB    175612
        BPL     AD4
        MOVB    175612,AUX

```

```

;TRANSFORMACAO DO NUMº
TST     %R2
BNE     AD5
CLR     %R0
CLR     %R1
RTS     %R3
; E' ZERO ?

```

```

AD5:    MOV     CTEM,%R3
        JSR     %R5,MULP

```

```

;Y(L) ESTA' N %R4,%R3
;ACERTO DA MANTISSA E EXPO      12*33=34,%R31

```

```

AD6:    MOV     CTEE,%R1
        ASL     %R4
        BMI     AD7
        DEC     %R1
        ASL     %R3
        BMI     AD3

```

```

AD8:  BR      AD6
      INC      %4
      BR      AD6
AD7:  ROR      %4
      MOV      %4,%8
      ADD      #17,%1
      TST      AUX
      BEQ      AD9
      CMP      #40000,%8
      BEQ      AD18
      NEG      %8
      BR      AD9
AD18: MOV      #100000,%8
      DEC      %1
AD9:  RTS      %5

```

```

;CONVERSOR D/A
;COMUNICACAO 11 PARA 10
;XU ESTA ARMazenado em %8 e %1
;TESTA SE XU E' NEGATIVO

```

```

DA5:  CLR      AUX
      TST      %8
      BPL      DA1
      INC      AUX
      NEG      %8
      CMP      #100000,%8
      BNE      DA1
      MOV      #40000,%8
      INC      %1

```

```

;ENVIO DA MANFISSA O NUMO JA' POSITIVO
; PARA O 10 , DE TRES EM TRES BITS

```

```

DA11: ROL      %8          ; SINAL NAO E' ENVIADO
      CLR      %2
      CLR      %3
      MOV      #3,%4

```

```

;ENVIO DE SINAL DE INICIO DE TRANSMISSAO 1891

```

```

TU1:  TSTB     175614
      BPL      TU1
      MOVB     #31,175616
TU2:  TSTB     175618
      BPL      TU2
      CMPSB    #131,175612
      BNE      TU2

```

```

;PREPARACAO DO DIGITO

```

```

PA3:  CLR      %3
PA9:  ASL      %2
      ROL      %8
      BCC      PA2
      INC      %3,
PA2:  INC      %2
      CMP      %4,%2          ; FIM DO DIGITO ?
      BNE      PA9

```

```

103:  ADD      #6*,%3
      TSTB   175614
      BPL    TU3
      MOVB   %1,175616      ; ENVIO DE 3 BITS
104:  TSTB   175618
      BPL    TU4
      CMPEB  %2,175612      ; TESTA ECO
      BNE    TU4
      ADD     #3,%4
      CMP     #22,%4        ; FIM DA MANTISSA ?
      BNE    DA3

```

; ENVIO DO SINAL DO NUM.

```

105:  ADD     #5*,AUX
      TSTB   175614
      BPL    TU5
      MOVB   AUX,175616      ; ENVIO DO SINAL DO NUM
106:  TSTB   175618
      BPL    TU6
      CMPEB  AUX,175612      ; TESTA ECO
      BNE    TU6

```

; ENVIO DO EXPO

```

      CLR     AUX
      TST     %1             ; SINAL E' NEG? ?
      BPL     DA4
      NEG     %1
      INC     AUX
DA4:  ADD     #2*,AUX
      ADD     #6*,%1
107:  TSTB   175614
      BPL    TU7
      MOVB   %1,175616      ; ENVIO DO EXPO
108:  TSTB   175618
      BPL    TU8
      CMPEB  %1,175612      ; TESTA ECO
      BNE    TU8

```

; ENVIO DO SINAL DO EXPO

```

109:  TSTB   175614
      BPL    TU9
      MOVB   AUX,175616      ; ENVIO DO SINAL DO EXPO
1010: TSTB   175618
      BPL    TU10
      CMPEB  AUX,175612
      BNE    TU10
      RLS     %5

```