

Cap. 9 — Exemplos Resolvidos no MATLAB[®]

Visão Geral do Capítulo.

Neste capítulo apresentaremos vários exercícios resolvidos via MATLAB[®], todos eles baseados no mesmo sistema mecânico. A idéia é ir examinando todos os tópicos da teoria através deste exemplo. O capítulo é dividido em seções cujo o nome identifica qual o capítulo e a teoria que está sendo aplicada.

Apenas o exemplo correspondente ao capítulo 4 não se baseia no sistema mecânico da figura 1. O capítulo se termina com um apêndice que explica como simular sistemas não lineares através do MATLAB SIMULINK[®].

O estudante absorverá melhor o conteúdo do capítulo se os resultados aqui mostrados forem reproduzidos no MATLAB.

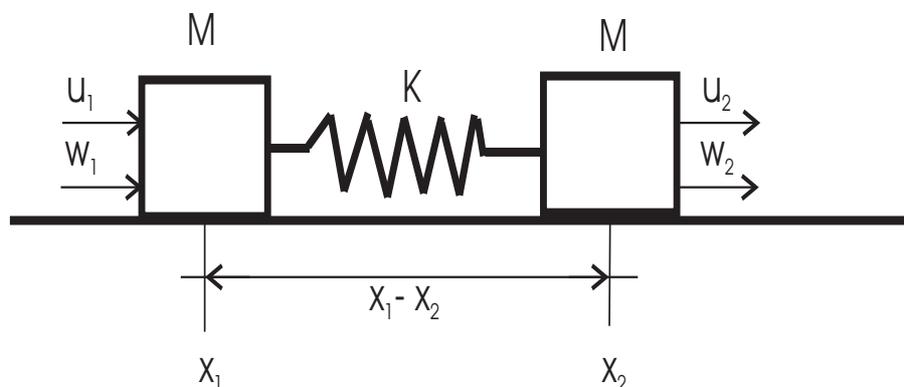


Figura 1: *Sistema mecânico considerado neste capítulo.*

Consideraremos que $w_1(t)$ e $w_2(t)$ são perturbações (forças) e $u_1(t)$, $u_2(t)$ são entradas. As massas são idênticas com posições dadas respectivamente por $x_1(t)$ e $x_2(t)$. A saída do sistema será $y(t) = (y_1(t), y_2(t))$ onde $y_1 = x_1$ e $y_2 = x_2$. A constante da mola que liga as duas massas é K . Assim é fácil mostrar que a dinâmica do sistema é

regida pelas equações diferenciais:

$$M\ddot{x}_1 + K(x_1 - x_2) = u_1 + w_1 \quad (1a)$$

$$M\ddot{x}_2 + K(x_2 - x_1) = u_2 + w_2 \quad (1b)$$

$$(1c)$$

Convertendo as equações para forma de estado teremos:

$$\dot{x} = Ax + Bu \quad (2a)$$

$$y = Cx + Du \quad (2b)$$

onde

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K/M & 0 & K/M & 0 \\ 0 & 0 & 0 & 1 \\ K/M & 0 & -K/M & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ K/M & 0 \\ 0 & 0 \\ 0 & K/M \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} \quad (2c)$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Adotaremos $M = 1$ Kg e $K = 1$ N/m.

1 Controlabilidade e Observabilidade (Caps. 2 e 3)

Exemplo 1 Mostremos que o sistema é controlável. Abaixo segue a lista dos comandos do Matlab e os seus ecos (somente nos casos interessantes):

```
>>K=1
```

```
>>M=1
```

```
>> A=[0 1 0 0;
```

```
>> -K/M 0 K/M 0;
```

```
>> 0 0 0 1;
```

```
>> K/M 0 -K/M 0]
```

```
>> B= [0 0;
```

```
>> K/M 0;
```

```
>> 0 0;
```

```
>> 0 K/M]
```

```

>> C=[1 0 0 0;
>>    0 0 1 0]

>> CC=CTRB(A,B) (matriz de controlabilidade)

CC =

    0    0    1    0    0    0   -1    1
    1    0    0    0   -1    1    0    0
    0    0    0    1    0    0    1   -1
    0    1    0    0    1   -1    0    0

>> s = svd(CC)  %(Decomposição em valores singulares - o
    % posto de uma matriz é o número de valores
    % singulares não nulos)

s =

    2.2361
    2.2361
    1.0000
    1.0000  % (vemos que o posto é quatro, porque há
    % 4 valores singulares não nulos)

```

Portanto o sistema é controlável porque o posto da matriz de controlabilidade coincide com a dimensão do estado. ♣

Exemplo 2 Mostremos agora que o sistema ainda é controlável se atuarmos apenas na entrada u_1 . Para isso seja B_1 a matriz obtida eliminando-se a segunda coluna de B :

```

>> B1= B(:,1)

B1 =

    0
    1
    0
    0

>> CC1=CTRB(A,B1)

```

```
>> s = svd(CC1)
```

```
s =
```

```
1.6180
1.6180
0.6180
0.6180
```

Como o posto da matriz de controlabilidade de (A, B_1) é quatro (4 valores singulares não nulos), segue-se que o sistema ainda é controlável. ♣

Exemplo 3 Suponhamos que um dispositivo é montado no sistema de maneira a provocar sempre $u_1 = u_2$. Mostremos que neste caso a controlabilidade é perdida. Note que isto é equivalente a fazermos

$$u = \begin{bmatrix} 1 \\ 1 \end{bmatrix} v$$

obtendo um sistema em malha fechada com a mesma matriz A mas com uma nova matriz de controle dada por $B_2 = B \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Abaixo mostramos que neste caso o sistema não é mais controlável.

```
>> r=[1;
      1 ]
```

```
>> B2=B*r
```

```
B2 =
```

```
0
1
0
1
```

```
>> CC2=CTRB(A,B2)
```

```
CC2 =
```

```
0    1    0    0
1    0    0    0
0    1    0    0
1    0    0    0
```

```
>> s = svd(CC2) % (nem precisava, já que duas colunas são nulas)
```

```
s =
    1.4142
    1.4142
         0
         0
```

Vemos que o sistema não é controlável porque o posto da matriz de controlabilidade é 2. 

Exemplo 4 Mostremos que o sistema (C, A) é observável.

```
>> O=OSV(A,C) % (matriz de observabilidade)

>> s = svd(O)

s =

    2.2361
    2.2361
    1.0000
    1.0000 %(vemos que o posto de O é 4!
            % já que temos 4 valores singulares
            % nulos)
```

Assim o sistema (C, A) é observável. Note que o sistema ainda é observável se medirmos apenas a saída y_1 . De fato, isto equivale a termos uma nova matriz C_1 obtida de C eliminando-se a segunda linha.

```
>> C1=C(1,:) % Obtém a primeira linha de C

C1 =

     1     0     0     0

>> O1=OSV(A,C1)

>> svd(O1)

ans =

    1.6180
```

```
1.6180
0.6180
0.6180
```

Vemos que o sistema continua observável (4 valores não singulares não nulos). ♣

Exemplo 5 Suponhamos que apenas as velocidades são medidas (e não as posições). Isto equivale a termos uma nova matriz:

$$C_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Neste caso:

```
>> C3=[0 1 0 0;
       0 0 0 1]
```

```
>> O3=obsv(A,C3)
```

```
>> s = svd(O3)
```

```
s =
```

```
4.4721
2.2361
1.0000
0 % (3 valores singulares não nulos)
```

Vemos que o posto da matriz de observabilidade é 3 e o sistema não é observável. ♣

2 Teoria da realização (Cap. 4)

Exemplo 6 Queremos fornecer uma realização minimal para a seguinte função de transferência:

$$G(s) = \begin{bmatrix} \frac{s^2-s+4}{(s+2)(s+3)} & \frac{s^2-s+4}{(s+2)(s+3)} \\ \frac{s^2+s+4}{(s+2)(s+3)} & \frac{s^2+s+4}{(s+2)(s+3)} \end{bmatrix}$$

Para isso vamos aplicar o algoritmo de realização por colunas do Cap. 4. O primeiro passo é obter o *m.m.c.* dos denominadores de cada coluna, fornecendo $d_1(s) = d_2(s) =$

$(s+2)(s+3)$. Depois escrevemos (através do algoritmo de divisão) para primeira coluna (e idem para a segunda):

$$g_{11}(s) = \underbrace{1}_{d_{11}} + \frac{\overbrace{-6}^{c_{12}} \underbrace{s}_{s} \overbrace{-2}^{c_{11}}}{\underbrace{(s+2)(s+3)}_{d_1(s)}}$$

$$g_{21}(s) = \underbrace{1}_{d_{11}} + \frac{\overbrace{-4}^{c_{12}} \underbrace{s}_{s} \overbrace{-2}^{c_{11}}}{\underbrace{(s+2)(s+3)}_{d_1(s)}}$$

Como $d_1(s) = s^2 + \underbrace{5}_{a_1} s + \underbrace{6}_{a_2}$, obtemos a seguinte realização para a primeira coluna:

$$A_1 = \begin{bmatrix} 0 & 1 \\ \underbrace{-6}_{-a_2} & \underbrace{-5}_{-a_1} \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} \underbrace{-2}_{c_{11}} & \underbrace{-6}_{c_{12}} \\ \underbrace{-2}_{c_{21}} & \underbrace{-4}_{c_{22}} \end{bmatrix}, D_1 = \begin{bmatrix} \underbrace{1}_{d_{11}} \\ \underbrace{1}_{d_{11}} \end{bmatrix}$$

Como a segunda coluna é idêntica, fazemos $A_2 = A_1$, $B_2 = B_1$, $C_2 = C_1$ e $D_2 = D_1$. Assim uma realização (não minimal) para $G(s)$ é dada por:

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}$$

$$C = [C_1 \ C_2], D = [D_1 \ D_2]$$

A realização minimal é obtida pela determinação da parte observável de (A, B, C, D) conforme ilustrado no arquivo *realiza.m* a seguir:

```
%%%%% arquivo realiza.m

a2=6
a1=5
A1=[ 0      1 ; % (monta a realização da primeira coluna)
     -a2   -a1 ]

B1=[0;
     1]
```

```

c11=-2 % monta a matriz C

c12=-6

c21=-2

c22=-4

C1=[c11 c12;
    c21 c22]

d11=1
d21=1

D1=[d11;
    d21]

A2=A1; % (realização da segunda coluna igual à primeira)
C2=C1;
D2=D1;
B2=B1;

A=[A1      zeros(2,2); % (monta realização não minimal)
   zeros(2,2) A2      ]

B=[B1      zeros(2,1);
   zeros(2,1) B2]

C=[C1 C2]

D=[D1 D2]

[U,S,V] = svd(observ(A,C)) % (Extrai parte observável de (A,B,C,D))
                        % Note que as duas últimas colunas de V
                        % fornecem uma base do subespaço não observável
                        % e as duas últimas completam esta base até
                        % uma base do espaço de estados

T=[V(:,3:4) V(:,1:2)] % monta matriz de mudança de base
                        % conforme observação acima

Atil = inv(T)*A*T % muda base para expor a decomposição

```

```

Btil = inv(T)*B    % em parte obs/não obs
Ctil = C*T
Dtil = D

Afinal = Atil(3:4,3:4) % (parte observável)
Bfinal = Btil(3:4,:)
Cfinal= Ctil(:,3:4)
Dfinal = Dtil

sys = ss(Afinal,Bfinal,Cfinal,Dfinal)

disp(' Confere a resposta');
tf(sys)

%%%%%%%%%%%%%% fim do arquivo realiza.m %%%%%%%%%%%%%%%

```

Tal programa fornece a seguinte realização minimal:

```

Afinal =

    -3.3647    -0.0718
     6.9282    -1.6353

Bfinal =

     0.3490     0.3490
    -0.6150    -0.6150

Cfinal =

    -6.6483     5.9833
    -5.2521     3.5235

Dfinal =

     1     1
     1     1

```



3 Estabilizabilidade e Detectabilidade (Cap. 5 e 6)

Exemplo 7 Mostremos agora que o exemplo 3 ilustra um caso onde o sistema também não é estabilizável.

```
[T,S,V]=svd(CC2) % (Chamada desta forma, a matriz T já
                  % é a transformação de base necessária
                  % para converter o sistema em sua parte
                  % controlável e não controlável.)
```

```
>> At = inv(T)*A*T
```

```
At =
```

```
    0.0000    0.0000   -0.0000    0.0000
   -1.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.3143    1.1111
    0.0000   -0.0000   -1.8889   -0.3143
```

```
>> Bt = inv(T)*B
```

```
Bt =
```

```
   -1.4142
         0
         0
         0
```

```
>> A22 = A(3:4,3:4) (Parte não controlável)
```

```
A22 =
```

```
    0.3143    1.1111
   -1.8889   -0.3143
```

```
>> eig(A22) % (autovalores da parte não controlável)
```

```
ans =
```

```
   -0.0000 + 1.4142i
   -0.0000 - 1.4142i
```

Vemos que a parte não controlável A_{22} não é assintoticamente estável (pólos imaginários puros). Portanto o sistema não é estabilizável. Fisicamente podemos interpretar tal fato da seguinte forma. Como a força aplicada nas duas massas é a mesma, não podemos controlar a deflexão da mola já que a deflexão da mola seria controlada pela diferença das forças. Note que os pólos encontrados em A_{22} (que são chamados de modos não controláveis) correspondem exatamente à frequência de ressonância do sistema. As direções controláveis correspondem à posição e velocidade do centro de massa e as direções não controláveis correspondem à elongação da mola e a sua taxa de variação.



Exemplo 8 Veremos agora que o caso do exemplo 5, onde verificamos que há perda de observabilidade, é um caso onde o sistema não é detectável.

```
>> [U,S,V]= svd(O3) % Como o posto de O3 é 3, as últimas n - 3 = 1
                    % colunas de V formam uma base do subespaço não observável
                    % e as primeiras 3 colunas complementam esta base
```

```
>> T = [V(:,4) V(:,1:3)]
```

```
>> At3 = inv(T)*A*T
```

```
At3 =
```

```

      0      1.0000     -0.0000         0
      0         0         0         0
  0.0000         0         0     -2.0000
      0      0.0000      1.0000         0
```

```
>> Ct3 = C*T
```

```
Ct3 =
```

```

      0     -0.7071     -0.7071         0
      0     -0.7071      0.7071         0
```

```
>> A11=A(1,1) % (parte não observável)
```

```
A11 =
```

```

0
>> eig(A11)

ans =

0 % (autovalores da parte não observável)

```

Vemos que o pólo da parte não observável A_{11} é nulo e portanto o sistema não é detectável. Note que a direção não observável está relacionada com a posição do centro de massa. Fisicamente podemos interpretar tal fato da seguinte maneira. Como só medimos as velocidades, a posição do centro de massa só pode ser deduzida por integração, que tem uma constante desconhecida. ♣

4 Imposição de pólos (Cap. 5)

Exemplo 9 Vamos considerar o exemplo onde eliminamos o atuador da segunda massa para obtendo o sistema $(A, B1)$. Neste caso fizemos um programa no Matlab (arquivo polos.m):

```

%%%%% CONTEÚDO DO ARQUIVO polos.m
%% F = realimentação (resposta)
%% (A,b) = sistema
%% pd =[p1 p2 ... pn] = vetor dos pólos desejados
%% N = n = ordem do sistema
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [F] = polos(A,b,pd,N)

p=poly(A)

pd=poly(pd)

for k = 1:N,
    a(k)= -p(k+1);    %(coeficientes do polinomio caracteristico de A)
    ad(k) = -pd(k+1); % (coef. do pol. caract. desejado de A+bF)
end

for k= 1:N,
    f(k)=ad(N-k+1)-a(N-k+1); % (Realimentacao f na base {e1, ... , en})
end

```

```

e(1:N,N) = b % ( faz en= b)

for k = 1:N-1,
    e(1:N,N-k) = A*e(1:N,N-k+1)-a(k)*b %%% (faz e{i-1} = A e{i} - a{i-1}b)
end

T=A; for k=1:N,
    T(1:N,k)=e(1:N,k); % (monta a matriz T= [e1...en])
end

At=inv(T)*A*T % conferencia (forma canônica controlável)
bt=inv(T)*b % idem acima
F=f*inv(T) % (resposta procurada = F)
%%%%%%%%%% Fim do arquivo polos.m %%%%%%%%%%%

```

Note que este programa implementa o método de imposição de pólos do Cap. 5 para $m = 1$. Executando o arquivo polos.m para impor os pólos $\{-1, -2, -1 + j, -1 - j\}$ vamos obter:

```

>>f=polos(A,B1,[-1, -2, -1+j, -1-j],4)

f =

    -8.0000    -5.0000     4.0000    -5.0000

```

Note que isto já é implementado de outra forma através da fórmula de Ackermann no Matlab (neste caso temos que inverter o sinal pois a convenção do Matlab é $(A - BF)$ ao invés de $(A + BF)$):

```

>> f=-acker(A,B1,[-1, -2, -1+j, -1-j])

f =

    -8    -5     4    -5

```

A resposta é a mesma porque para $m = 1$ a solução da imposição de pólos é única. ♣

Exemplo 10 Para o caso onde consideramos o sistema completo (A, B) com duas entradas, vamos determinar \hat{F} e \hat{u} tal que $(A + B\hat{F}, \hat{b})$ seja controlável com $\hat{b} = B\hat{u}$. Para

isso chutamos $\hat{F} = 0$ e $\hat{u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. O leitor pode facilmente verificar que recaímos no caso anterior, obtendo

$$F1 = \hat{F} + \hat{u}f = \begin{bmatrix} -8.0000 & -5.0000 & 4.0000 & -5.0000 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

O MATLAB possui ainda uma outra rotina que permite a solução do problema de imposição para $m > 1$. É a rotina *place* utilizada conforme exemplo abaixo:

```
>> F2=-place(A,B,[-1, -2, -1+j, -1-j])
place: ndigits= 15
```

F2 =

```
-0.9834    -2.6900    -1.0810     0.4002
-0.9035     0.5250    -1.0128    -2.3100
```

Note que os valores dos ganhos de $F2$ são bastante inferiores aos obtidos na realimentação $F1$. De fato a rotina *place* utiliza um algoritmo de otimização, que acaba gerando ganhos menores que o método (ruim) de “chutar” \hat{F} e \hat{u} . Assim, as duas entradas do sistema são efetivamente usadas, com possibilidade de gerar realimentações mais aplicáveis na prática. ♣

Exemplo 11 Chamar as rotinas *acker* e *place* para sistemas não controláveis produz, obviamente, um erro! O máximo que podemos fazer é impor pólos na parte controlável, como prevê a teoria.

```
>> f=-acker(A,B2,[-1, -2, -1+j, -1-j])
Warning: Matrix is singular to working precision.
> In C:\MATLABR12\toolbox\control\control\acker.m at line 42
??? Error using ==> eig NaN or Inf prevents convergence.
```

```
>> f=-place(A,B2,[-1, -2, -1+j, -1-j])
??? Error using ==> place Can't place eigenvalues there.
```

♣

5 Observadores e Compensadores (Cap.6) e Controle Ótimo (Cap.8)

Projetar observadores consiste em determinar injeções da saída K estabilizantes. Assim dado um par (C, A) deve-se determinar o sistema dual $A_1 = A^T$, $B_1 = C^T$ e determinar uma realimentação estabilizante F_1 . Como visto na teoria, fazemos $K = -F^T$.

Exemplo 12 Determinamos uma injeção da saída K tal que $\sigma(A-KC) = \{-1, -2, -1-j, -1+j\}$:

```
>> F=-place(A1,B1,[-1, -2, -1+j, -1-j])
place: ndigits= 15 Warning: Pole locations are more than 10% in error.
```

F =

```
-2.0645    -1.0066     0.2370    -0.9950
 0.2549    -0.9805    -2.9355    -0.9935
```

```
>> K = -F'
```

K =

```
 2.0645   -0.2549
 1.0066    0.9805
-0.2370    2.9355
 0.9950    0.9935
```

Note a sensibilidade do problema. Por questões numéricas, o problema de imposição de pólos sofre erros que são indicados pela rotina. Neste caso deve-se determinar o ganho do observador via controle ótimo, pois a sensibilidade alta indica também uma baixa robustez:

```
>> Q=eye(4)
```

Q =

```
 1    0    0    0
 0    1    0    0
 0    0    1    0
 0    0    0    1
```

```
>> R=eye(2)
```

R =

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

>> [F] = -lqr(A1,B1,Q,R)

F =

$$\begin{bmatrix} -1.4727 & -0.6180 & -0.2594 & -0.3820 \\ -0.2594 & -0.3820 & -1.4727 & -0.6180 \end{bmatrix}$$

>> K = -F'

K =

$$\begin{bmatrix} 1.4727 & 0.2594 \\ 0.6180 & 0.3820 \\ 0.2594 & 1.4727 \\ 0.3820 & 0.6180 \end{bmatrix}$$



6 Teoria da regulação e rastreamento (Cap.7) e controle ótimo (Cap. 8)

Considere que o sistema (2a)–(2c) é submetido a uma perturbação da forma $w_1(t) = \alpha_1 r(t) + \beta_1 \cos(0.5t + \phi_1)$ e $w_2(t) = \alpha_2 r(t) + \beta_2 \cos(0.5t + \phi_2)$, onde as amplitudes α_i, β_i , $i = 1, 2$ são desconhecidas e $r(t)$ é um degrau unitário.

Queremos projetar um compensador de regulação para este sistema. A teoria do capítulo 7 nos diz que devemos inicialmente construir o modelo interno. Neste caso teremos (para $i = 1, 2$):

$$\begin{aligned} A_i &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -0.25 & 0 \end{bmatrix}, B_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ C_i &= [1 \ 0 \ 0] \end{aligned}$$

Assim o modelo interno será (justifique!):

$$\begin{aligned} A_\Omega &= \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, B_\Omega = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \\ C_\Omega &= \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \end{aligned}$$

A teoria do Capítulo 7 nos mostra que qualquer compensador de estabilização para o sistema aumentado $(\tilde{A}, \tilde{B}, \tilde{C})$ dado por:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}}_\Omega \end{bmatrix} = \underbrace{\begin{bmatrix} A & BC_\Omega \\ 0 & A_\Omega \end{bmatrix}}_{\tilde{A}} \begin{bmatrix} x \\ \tilde{x}_\Omega \end{bmatrix} + \underbrace{\begin{bmatrix} B & 0 \\ 0 & B_\Omega \end{bmatrix}}_{\tilde{B}} \begin{bmatrix} v \\ u_\Omega \end{bmatrix} \quad (3a)$$

$$\tilde{y} = y = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{\tilde{C}} \begin{bmatrix} x \\ \tilde{x}_\Omega \end{bmatrix} \quad (3b)$$

é solução do problema de regulação. O arquivo *masmol.m* a seguir é utilizado para fazer o projeto deste compensador de regulação:

```

%%%% Conteúdo de masmol.m
%%%% A,B,C,D = sistema massa/mola
%%%% Ai,Bi,Ci = modelo interno
%%%% Ao,Bo,Co = sistema auxiliar (índice Omega)
%%%% At,Bt,Ct = sistema aumentado (com til)
K=1
M=1

A=[ 0 1 0 0 ;
   -K/M 0 K/M 0 ;
    0 0 0 1 ;
    K/M 0 -K/M 0 ]

B= [0 0;
    K/M 0;
    0 0;
    0 K/M ]

C=[1 0 0 0;
   0 0 1 0]

D= zeros(2,2);

Ai = [ 0 1 0; % (modelo interno)
       0 0 1;
       0 -0.25 0]

Bi=[0 ;
    0 ;
    1 ]

```

```

Ci=[1 0 0]

Ao= [ Ai      zeros(3,3); % (sistema auxiliar Ao, Bo, Co)
      zeros(3,3)  Ai      ]

Co = [Ci      zeros(1,3);
      zeros(1,3)  Ci      ]

Do= zeros(2,2)

Bo = [ Bi      zeros(3,1);
      zeros(3,1)  Bi      ]

At = [ A      B*Co; %(sistema aumentado At,Bt,Ct)
      zeros(6,4)  Ao]

Bt=[B      zeros(4,2);
     zeros(6,2)  Bo      ]

Ct=[C      zeros(2,6)]

Contr = svd(CTRB(At,Bt)) %(confirma a controlabilidade do sistema
                          % aumentado)
                          % Pelo menos a estabilizabilidade seria
                          % necessária
Observ = svd(OBSV(At,Ct)) %(confirma a observabilidade do sistema
                            % aumentado)
                            % Pelo menos a detectabilidade seria
                            % necessária

[K,S,E] = lqr(At,Bt,1e3*eye(10),eye(4)) % (Calcula realimentação de
                                          % estado por controle ótimo)

Ft = -K % Realimentação de estado

[H,S,E] = lqr(At',Ct',1e3*eye(10),eye(2)) % (Calcula injeção da
                                              % saída por controle
                                              % ótimo)

Kt = H' % injeção da saída

```

Exemplo 13 Considere no exemplo anterior que sinais do mesmo tipo de $w_1(t)$ e $w_2(t)$ podem ser usados como sinais de referência. Vamos mostrar que neste caso o esquema da figura 10 do Capítulo 7 resolve o problema da regulação e rastreamento simultâneos. Para isso precisamos mostrar que as condições da Prop. 4 do apêndice do Cap. 7 são satisfeitas. De fato, neste caso o modelo da referência $(\tilde{A}_\Omega, \tilde{B}_\Omega, \tilde{C}_\Omega)$ é o mesmo sistema auxiliar $(A_\Omega, B_\Omega, C_\Omega)$ anterior. Neste caso as condições da Proposição 4 equivalem a verificar se o espaço não observável \mathcal{N}_0 do par:

$$\bar{A} = \begin{bmatrix} \tilde{A} & 0 \\ 0 & \tilde{A}_\Omega \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} \tilde{C} & \tilde{C}_\Omega \end{bmatrix}$$

é tal que a dimensão do subespaço $\dim([0_{6 \times 10} \ I_6]\mathcal{N}_0)$ seja igual à 6. O próximo programa no MATLAB permite esta verificação:

```

%%% arquivo testa4.m
%%% Programa que verifica condição da Prop. 4, Cap. 7
%%%
%%%
Abarra=[ At          zeros(10,6);
         zeros(6,10) Ao      ];

Cbarra=[Ct Co];

Cbarra=[Ct Co];

s1 = svd (Obsv(Abarra,Cbarra)) %% note abaixo que S tem 6 valores
                                %% singulares nulos, corres-
                                %% pondendo à dimensão do
                                %% espaço não observável).

(Eco do MATLAB
s1 =

170.5511
170.2259
  1.4391
  1.4391
  1.4142
  1.2109
  1.1528
  1.0842
  1.0307

```

```

1.0307
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000)

[U,S,V] = svd (Obsv(Abarra,Cbarra));

No=V(:,11:16); % (as últimas seis colunas de V geram o
               % subespaço não observável)

Pi=[zeros(6,10) eye(6)]; % (Projeção)

s2 = svd(Pi*No) %(se há seis valores não singulares não nulos
                %então a condição procurada é satisfeita)
(eco do MATLAB
s2 =

0.7071
0.6963
0.6963
0.5363
0.4444
0.3631)

```

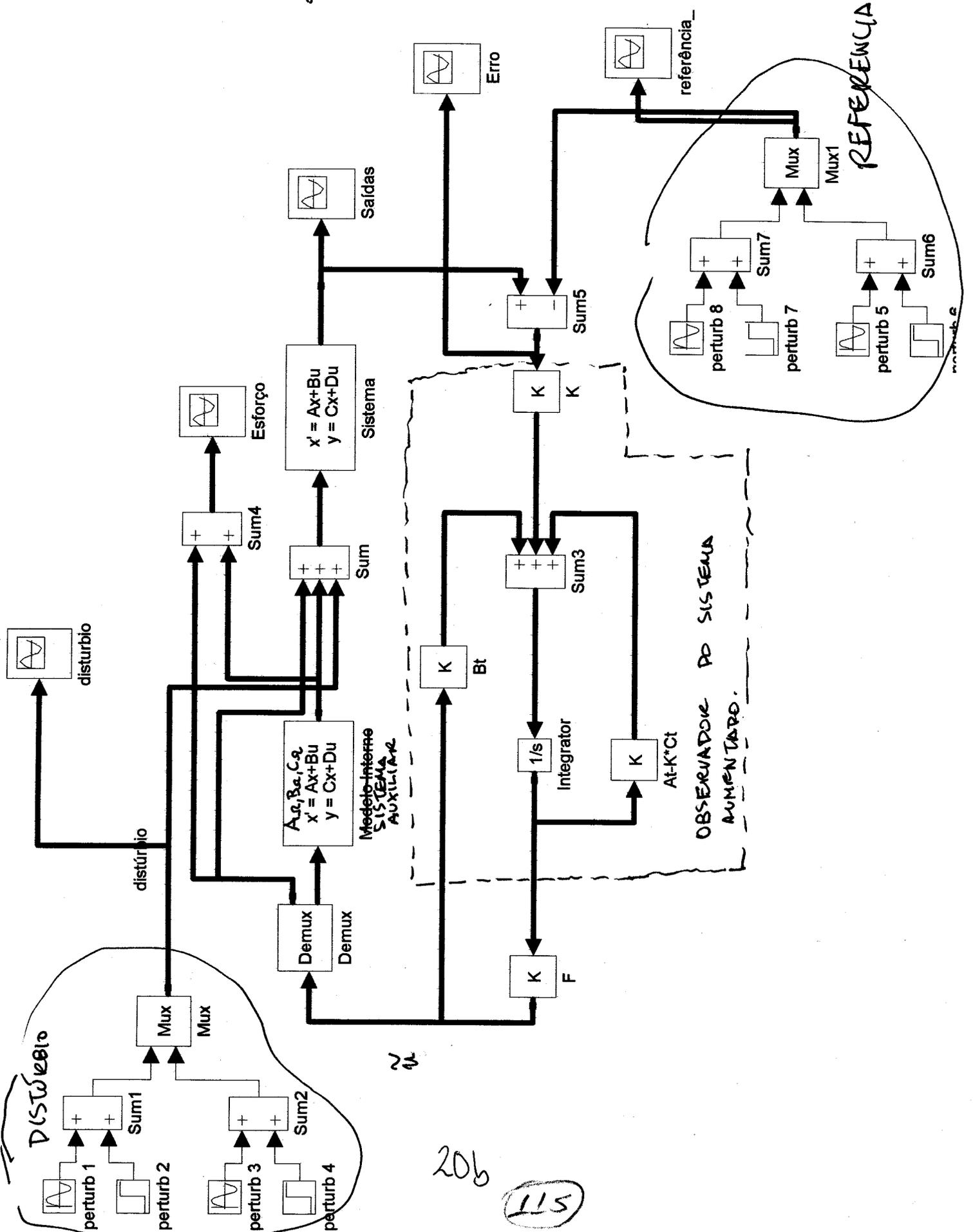
Como $s2$ fornece 6 valores singulares não nulos, conclui-se que as condições da proposição 4 são satisfeitas.

Simulações no simulink foram feitas para condições iniciais nulas mostrando os resultados das figuras a seguir, mostrando que há regulação e rastreamento simultâneo. Algumas simulações foram feitas reprojando o ganho Ft através da imposição dos pólos $\{-10, -20, -30, -11, -21, -31, -12, -22, -32, -33\}$. Os esforços de controle com o controle ótimo foram bem menores e a performance foi superior à imposição de pólos, como podemos observar nas simulações. Em anexo também apresentamos o diagrama do simulink utilizado.



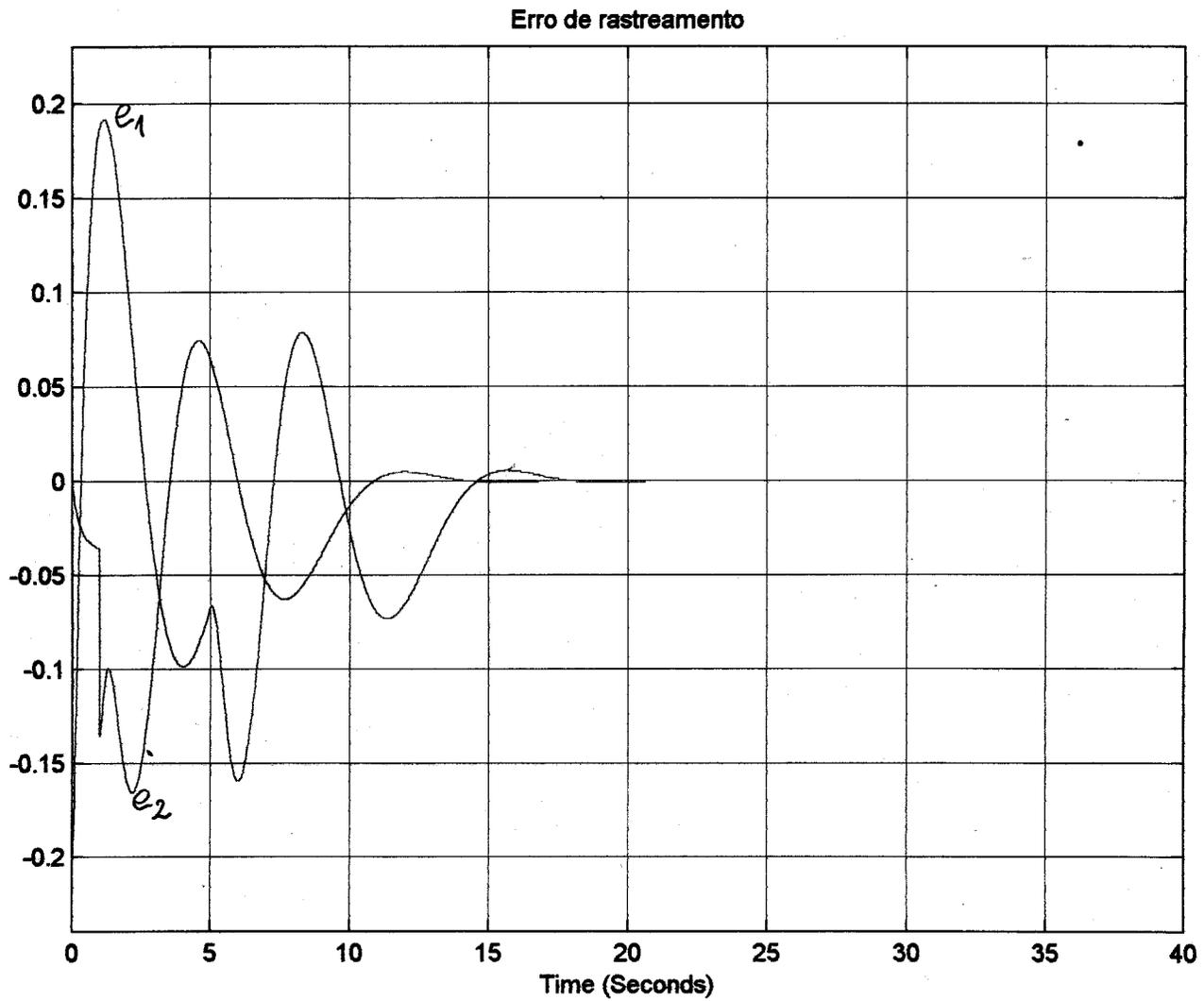
EXEMPLO 12

Diagrama do Simulink



EXEMPLO - 12

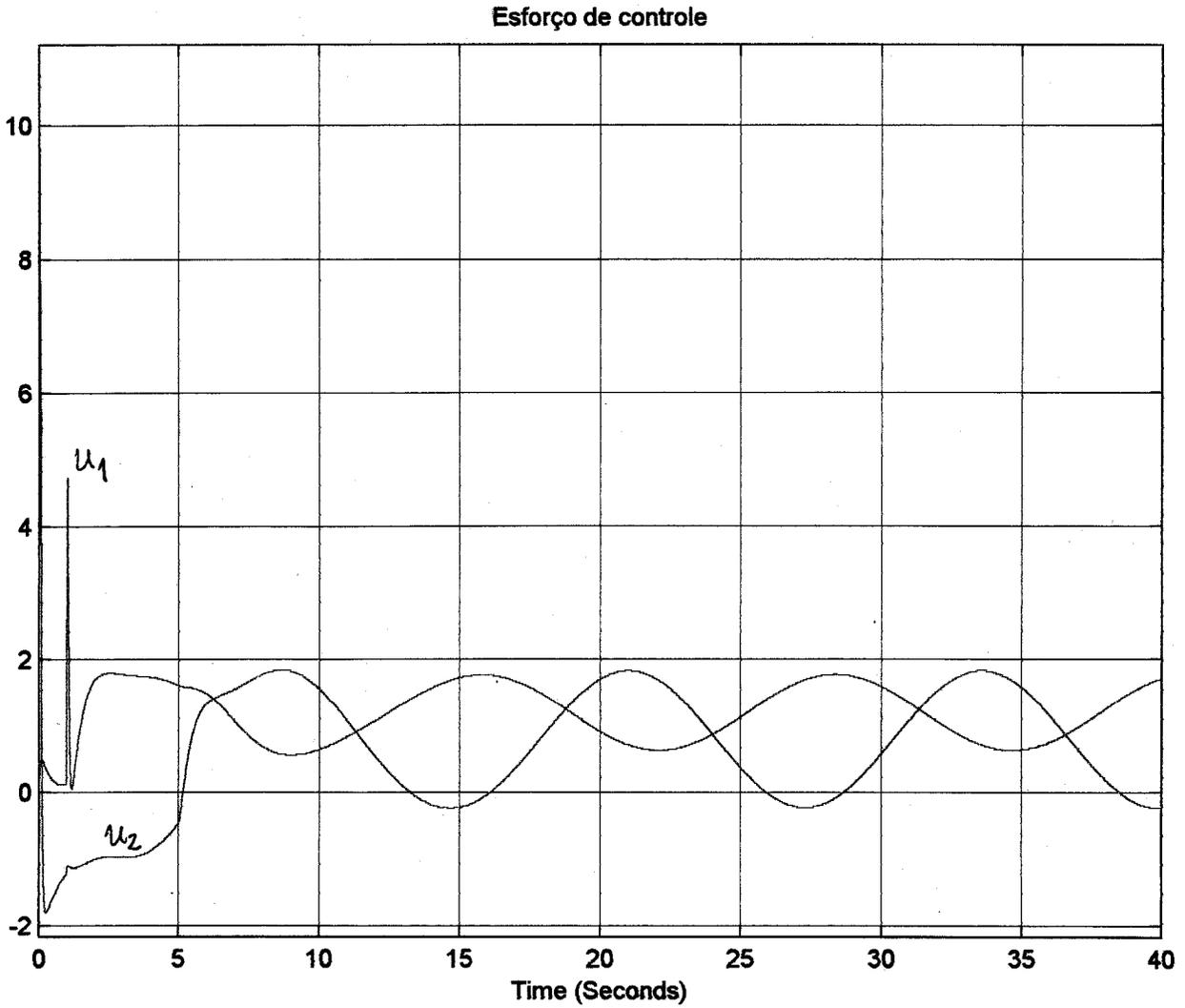
Regulação e Rastreamento



200
116

EXEMPLO 12

Regulação e Rastreamento

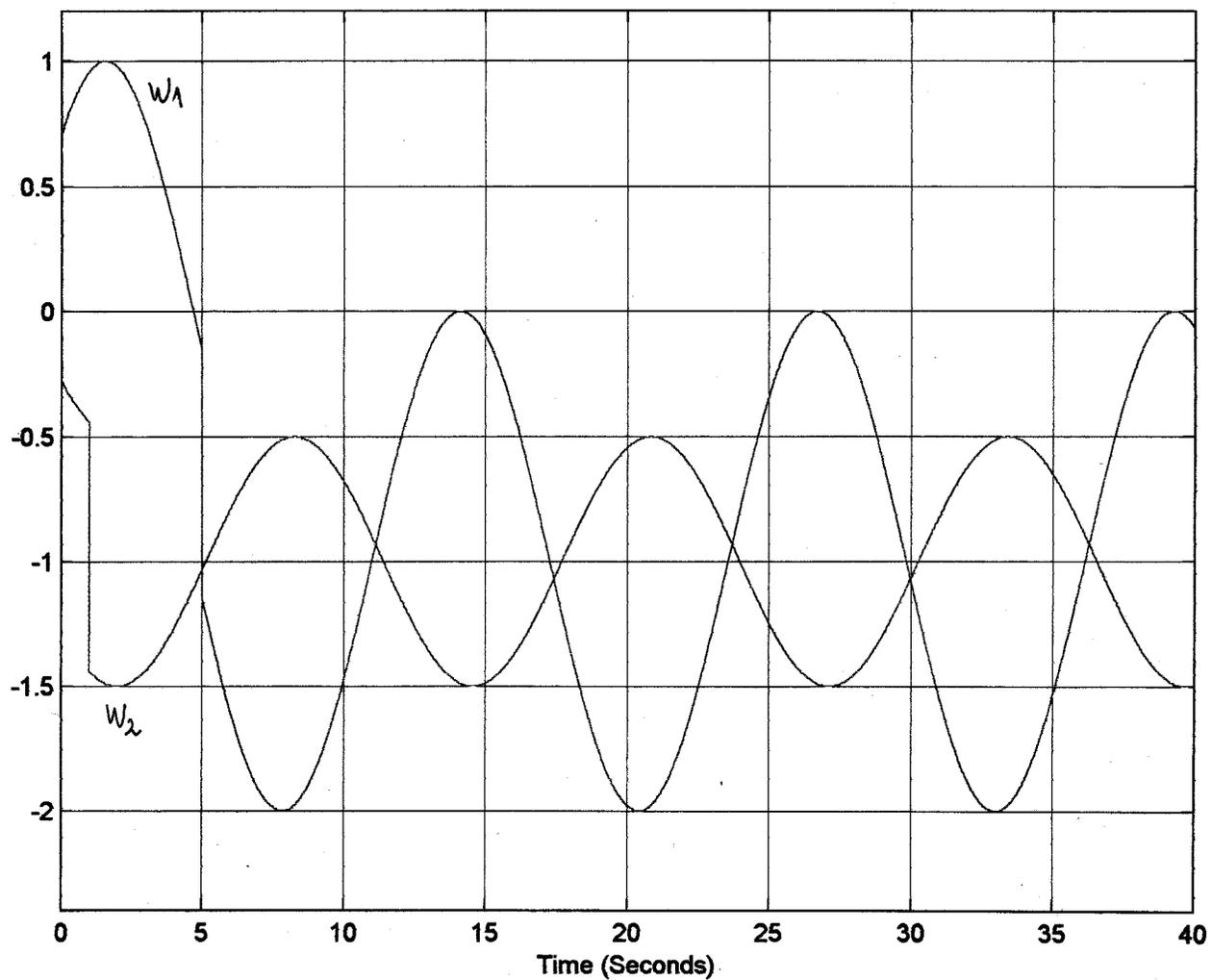


20d

117

EXEMPLO 12 - Regulação e Rastreamento

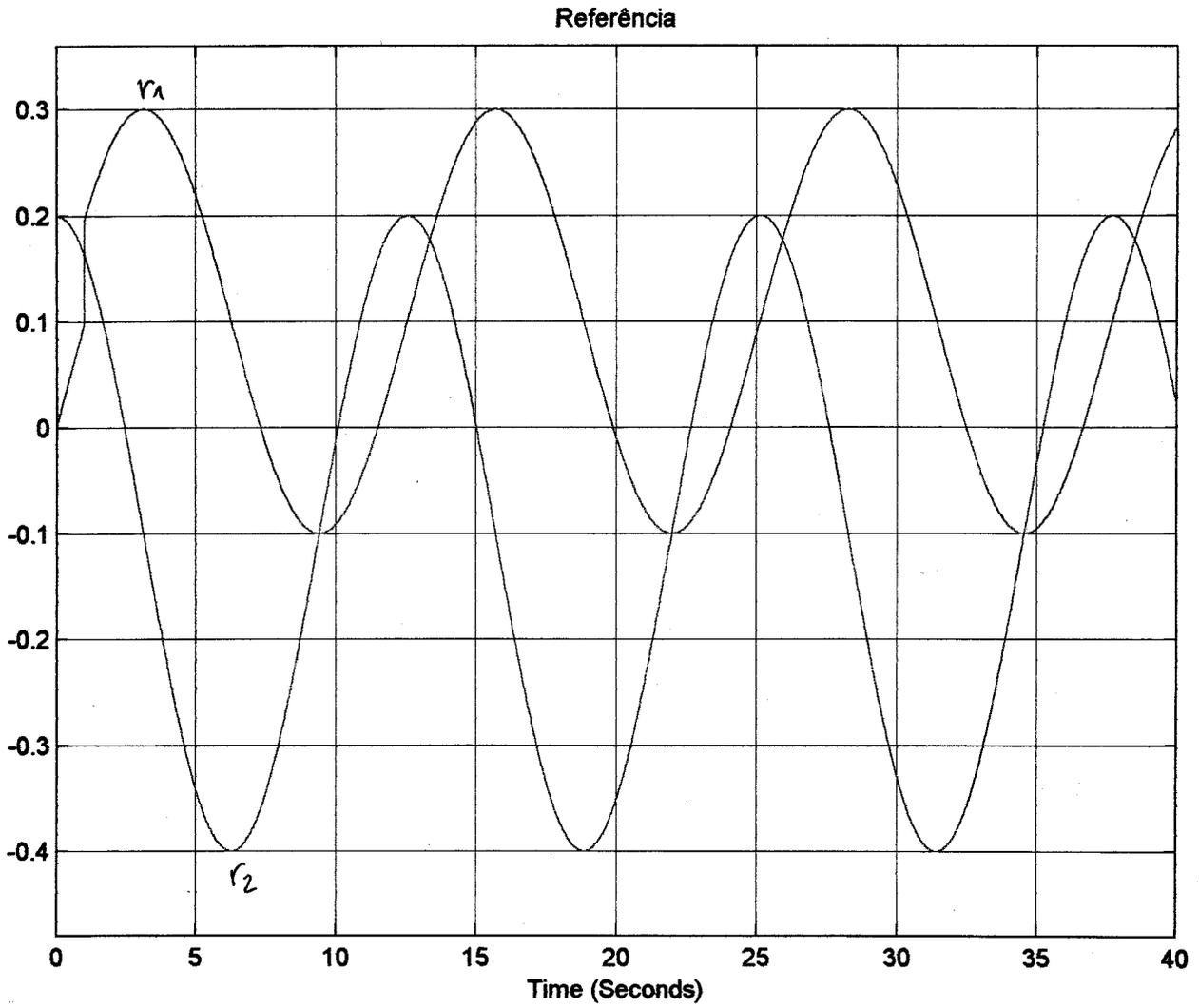
Distúrbio



20e

118

EXEMPLO 12 - Regulação e Rastreamento



20f

119

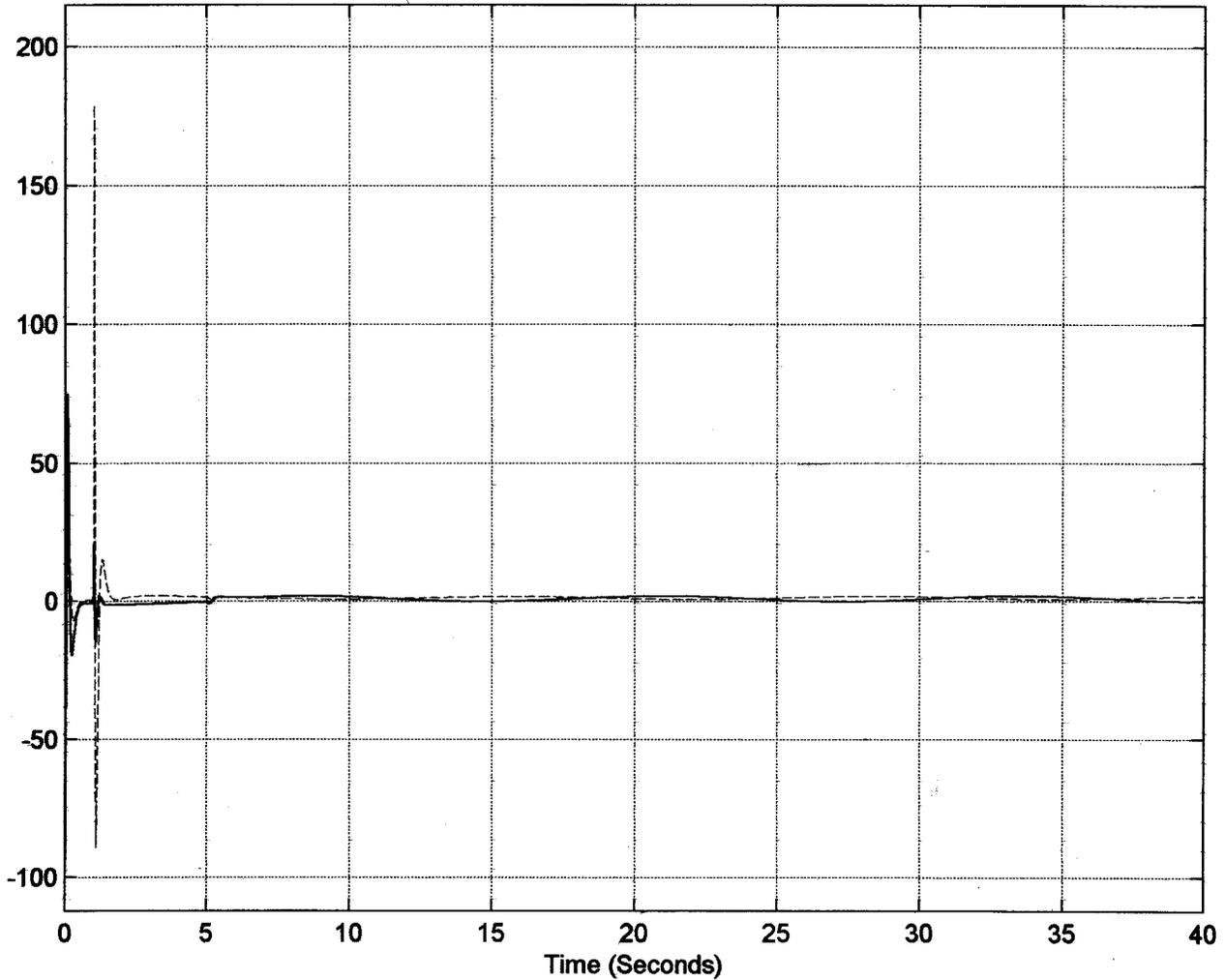


Esforço de Controle IMPOSIÇÃO DE PÓLOS

$$P = \begin{bmatrix} -10 & -20 & -30 & -11 & -21 & -31 & -12 & -22 & -32 & -33 \end{bmatrix}$$

pólos de $\tilde{A} + B\tilde{E}$

masmoll



$$F_{\text{ctrl}} = -\text{place}(A_t, B_t, P)$$

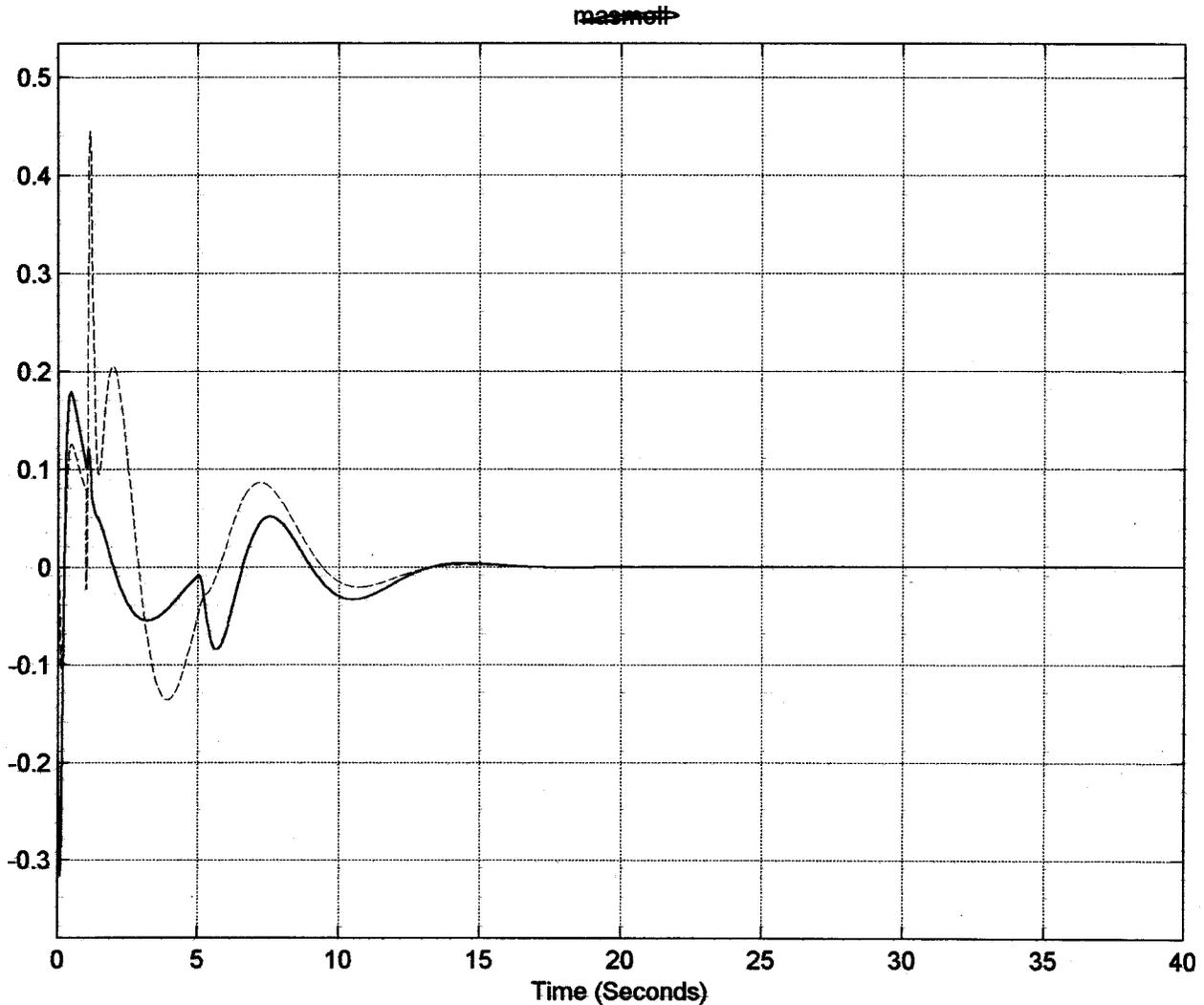
20g
120

(6)

$$\text{Erros: } (y_{1ref} - y_1) \\ (y_{2ref} - y_2)$$

IMPOSIÇÃO DE PÓLOS

Erro de rastreamento



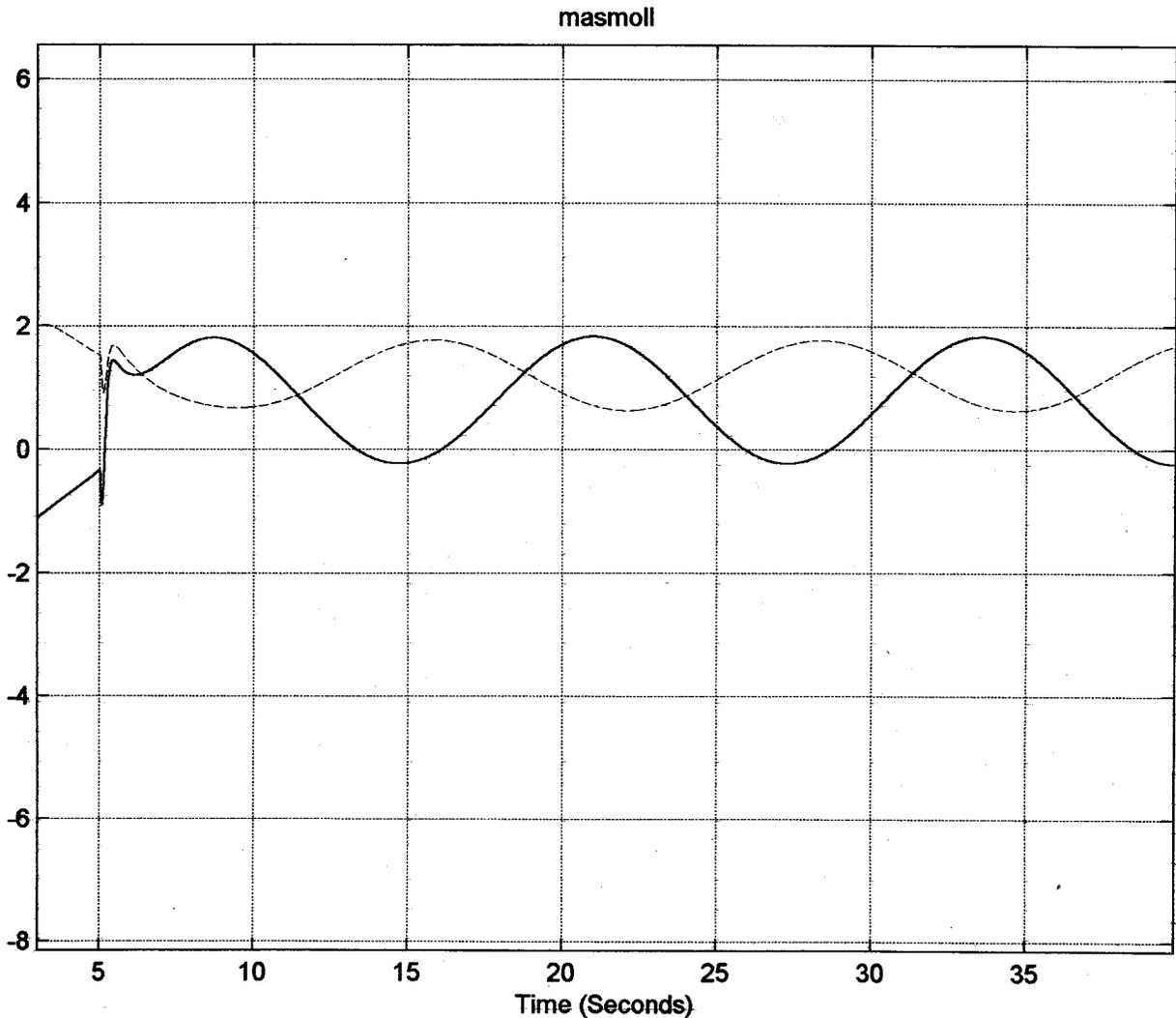
OBS: O erro máximo do controle ótimo foi de ± 0.2 .

A convergência p/ zero se deu em 20 segundos para um erro imperceptível.

20h

(121)

Esforço de controle no Regime (imposição de pólo)



OBS: NO REGIME O ESFORÇO
É O MESMO INDEPENDENTE
DO PROJETO, JÁ QUE O
CONTROLE CANCELA A
PERTURBAÇÃO.

20/

122

Apêndice — Como simular sistemas não-lineares

Nesta seção vamos apresentar brevemente um método de simulação de sistemas não lineares no MATLAB. A idéia é escrever a dinâmica não linear através de uma *function* do matlab. Por exemplo, suponhamos que queremos simular o sistema não-linear:

$$\begin{aligned}\dot{x}_i &= f_i(x, u), i = 1, \dots, n \\ y_j &= h_j(x), j = 1, \dots, p\end{aligned}$$

A idéia é fazer duas “functions”. A primeira que chamaremos “dinamica.m” e a segunda que chamaremos “saida.m”, possuindo a seguinte estrutura:

```
%(conteúdo do arquivo dinamica.m)

function [xp] = dinamica(z)

% O vetor z = (x, u)' - Extrai-se a informacao de (x, u)
% nos proximos comandos
for I = 1:n,
    x(I) = z(I); % Extraindo o vetor x
end
for I = (n+1):(n+m),
    u(I-n) = z(I); % Extraindo o vetor u
end

% Dinamica propriamente dita

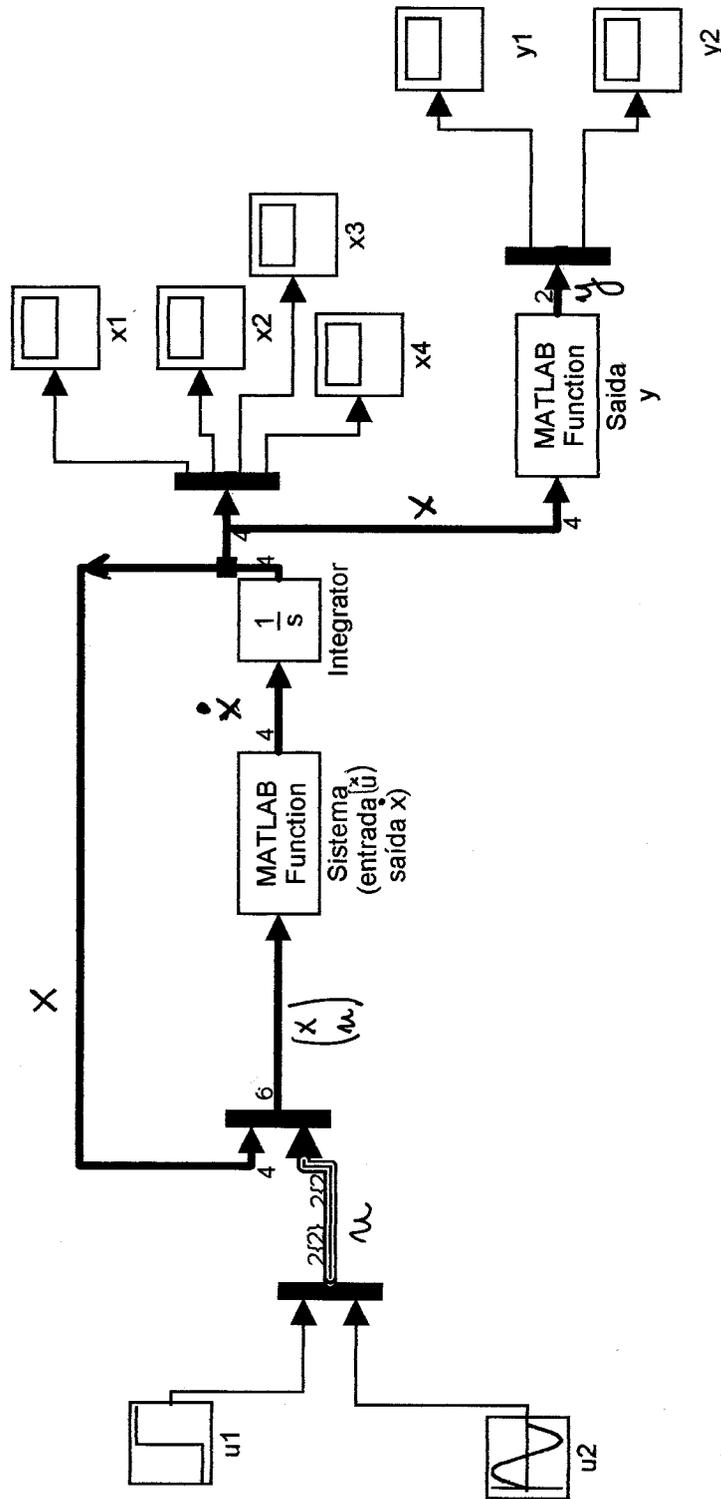
xp(1) = f1(x,u);
xp(2) = f2(x,u);
.....
xp(n) = fn(x,u);
%-----

% (conteúdo do arquivo saida.m)
function [y] = saida(x)
y(1) = h1(x);
y(2) = h2(x);
.....
y(p) = hp(x);
```

Depois fazemos um diagrama do simulink utilizando as MATLAB “functions” conforme a figura da página seguinte. Note que é preciso clicar cada bloco de *Matlab function*, definir o número de saídas do bloco e o nome do arquivo “.m” que será chamado. Lembre-se também que as condições iniciais podem ser dadas clicando o integrador e preenchendo esta informação no campo apropriado na forma de vetor linha $[x_1^0 \ x_2^0 \ \dots \ x_n^0]$.

Apêndice - Como Simular Sistemas não Lineares

Exemplo de simulação não linear
(Exemplo com $n=4$ e $m=2$)



```
function [xp] = dinamica(z)
```

```
n=4;
```

```
m=2;
```

```
for I = 1:n,
```

```
    x(I) = z(I);
```

```
end
```

```
for I = (n+1):(n+m),
```

```
    u(I-n) = z(I);
```

```
end
```

```
xp(1) = x(2);
```

```
xp(2) = -x(1) - x(2) - (x(1))^3 - (x(2))^3 + u(1);
```

```
xp(3) = x(4);
```

```
xp(4) = -x(3) - x(4) - (x(3))^3 - (x(4))^3 + u(2);
```

EXEMPLO

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_1 - x_2 - x_1^3 - x_2^3 + u_1$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -x_3 - x_4 - x_3^3 - x_4^3 + u_2$$

2/c

125