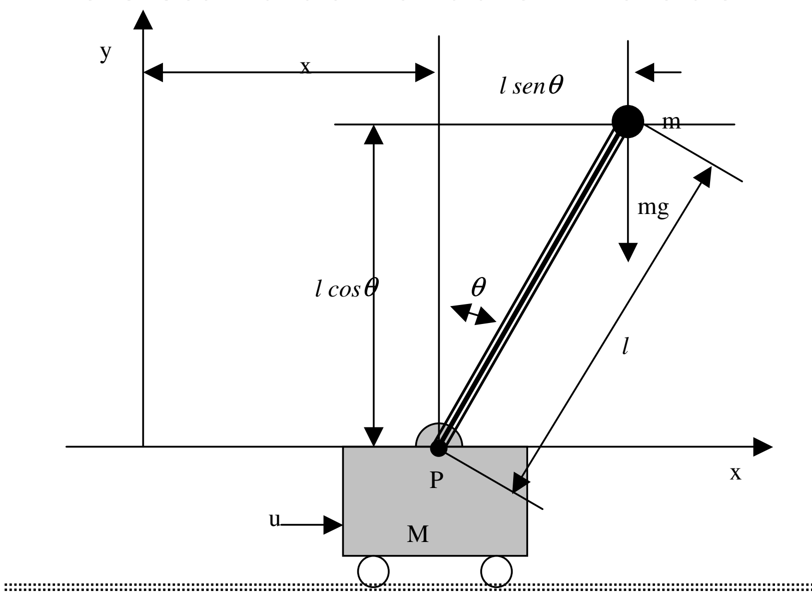
Universidade Estadual Paulista
“Júlio de Mesquita Filho”
“Campus” de Rio Claro
Instituto de Geociências e Ciências Exatas

Controle de Processo

Prof. Dr. José M. Balthazar

Projeto de um controlador digital para o sistema de Pêndulo Invertido



Grupo:

Daniel Gollo Aceto

Erich Gebrin Bachion

Fábio Côa

Marco Aurélio do Amaral Rocha

Reinaldo Yuji Ogassawara

Rômulo Caires dos Santos

Estudo de Caso

Considera-se o projeto de um controlador digital para um processo tipo 0. O controlador digital envolve retroação de estado e controle integral. Primeiro são deduzidas as equações para o sistema mostrado na Fig. 1, em que o processo não envolve um integrador. Em seguida, serão discutidos os detalhes de projeto de um controlador digital para o sistema pêndulo invertido.

Projeto de um servossistema no qual o processo não envolve um integrador

Considera-se o servossistema mostrado na Fig. 1 no qual o processo não envolve um integrador. Este sistema usa retroação de estado e controle integral.

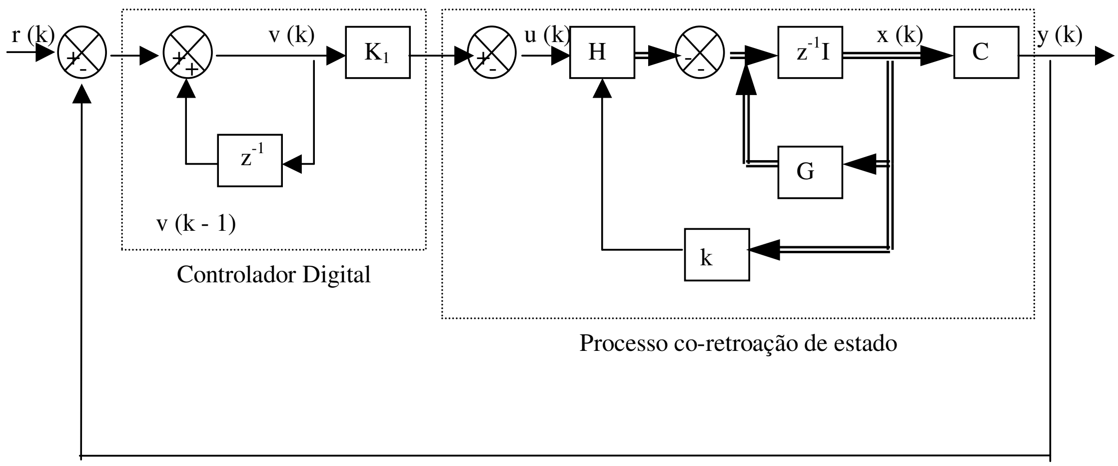


FIGURA 1

Abordagem básica para o projeto de servossistemas

A partir da Fig. 1, a equação de estado do processo a controlar e a equação de saída são:

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}u(k)$$
$$y(k) = \mathbf{C}\mathbf{x}(k)$$

A equação para o integrador é:

$$v(k) = v(k-1) + r(k) - y(k)$$

Como

$$u(k) = -\mathbf{K}\mathbf{x}(k) + K_1 v(k)$$

tem-se

$$\begin{aligned} v(k+1) &= v(k) + r(k+1) - y(k+1) \\ &= v(k) + r(k+1) - C[Gx(k) + Hu(k)] \\ &= (1 - CHK_1)\, v(k) + (-CG + CHK)\, x(k) + r(k+1) \end{aligned}$$

Notando-se que

$$\begin{aligned} x(k+1) &= Gx(k) + H[-Kx(k) + K_1 v(k)] \\ &= (G - HK)\, x(k) + HK_1 v(k) \end{aligned}$$

obtêm-se as seguintes equações de estado e saída para todo o sistema, a malha fechada:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G - HK & HK_1 \\ -CG + CHK & 1 - CHK_1 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k+1)$$

$$y(k) = [C \;\; 0] \begin{bmatrix} x(k) \\ v(k) \end{bmatrix}$$

Para uma entrada em degrau $r(k)=r$,

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G - HK & HK_1 \\ -CG + CHK & 1 - CHK_1 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \quad (1)$$

Para uma entrada em degrau, à medida que k se aproxima do infinito, $x(k)$, $u(k)$ e $v(k)$ se aproximam dos valores constantes $x(\infty)$, $u(\infty)$ e $v(\infty)$, respectivamente. Da mesma forma,

$$v(\infty) = v(\infty) + r(\infty) - y(\infty)$$

ou

$$y(\infty) = r(\infty) = r$$

Não existe erro em regime estacionário. Substituindo-se $k=\infty$ na Eq. (1), obtêm-se

$$\begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} G - HK & HK_1 \\ -CG + CHK & 1 - CHK_1 \end{bmatrix} \begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Seja

$$\begin{aligned} x_e(k) &= x(k) - x(\infty) \\ v_e(k) &= v(k) - v(\infty) \end{aligned}$$

Tem-se, então,

$$\begin{aligned} \begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} &= \begin{bmatrix} G - HK & HK_1 \\ -CG + CHK & 1 - CHK_1 \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \\ &= \begin{bmatrix} G & 0 \\ -CG & 1 \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} [-K \;\; K_1] \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (2) \end{aligned}$$

Definindo-se

$$w(k) = [-K \;\; K_1] \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (3)$$

A Eq. (2), então, se torna

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} G & 0 \\ -CG & 1 \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} w(k) \quad (4)$$

As Eqs. (3) e (4) podem ser reescritas, respectivamente, como

$$\xi(k+1) = \hat{G}\,\xi(k) + \hat{H}\,w(k) \quad (5)$$

$$w(k) = -\hat{K}\,\xi(k) \quad (6)$$

onde

$$\xi(k) = \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix}, \quad \hat{G} = \begin{bmatrix} G & 0 \\ -CG & 1 \end{bmatrix}, \quad \hat{H} = \begin{bmatrix} H \\ -CH \end{bmatrix} \quad e \quad \hat{K} = [K \;\; -K_1]$$

As Eqs. (5) e (6) representam equações padrões no espaço de estados. A matriz **K^** pode, então, ser determinada usando-se a abordagem da alocação de pólos, desde que o sistema representado pelas Eqs. (5) e (6) seja completamente controlável.

Considere-se o sistema pêndulo invertido mostrado na Fig. 2, onde um pêndulo invertido é montado sobre um pequeno carro movido a motor. Aqui, se considera somente o problema bidimensional, no qual o pêndulo se move no plano do papel. O sistema pêndulo invertido é instável na medida em que pode cair a qualquer instante, a menos que uma força de controle adequada seja aplicada. Admita-se que a massa do pêndulo está concentrada no final da haste, como mostrado na figura (Considera-se que a haste não tenha massa). A força de controle $\mathbf{u}$ é aplicada ao carrinho.

No diagrama, θ seja é o ângulo da haste com a vertical. Admita-se que o ângulo θ seja pequeno, de forma que se possa aproximar $\mathbf{sen}\ \theta$ por $\mathbf{0}$ e $\mathbf{cos}\ \theta$ por $\mathbf{1}$, e também que $\dot{\theta}$ seja pequeno, de forma q $\theta\dot{\theta}^2 = 0$ (Sob essas condições, as equações não-lineares do sistema podem ser linearizadas).

É desejável manter o pêndulo aprumado na vertical em resposta a mudança em degrau na posição do carro (A força de controle $\mathbf{u}$ é a força aplicada ao carro). Portanto, iremos projetar um controlador digital para o sistema pêndulo invertido.

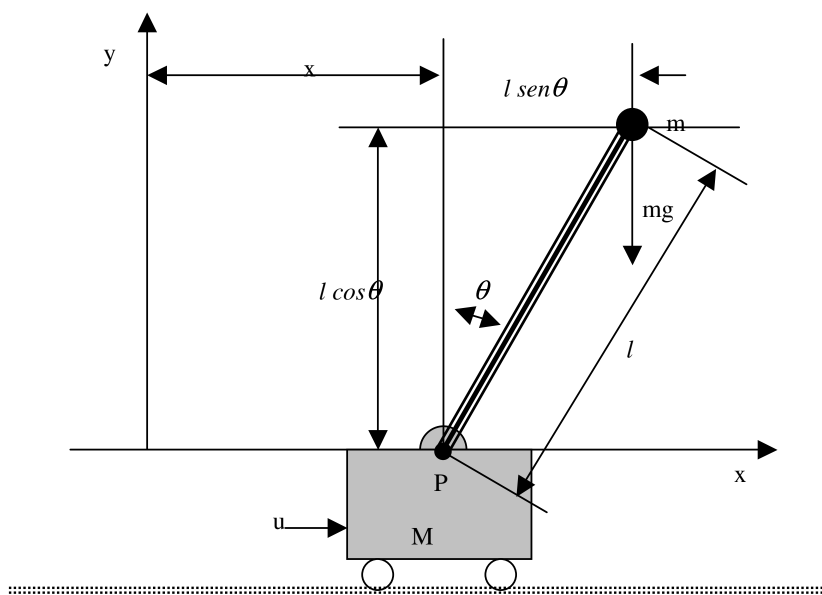


FIGURA 2

Como o processo (pêndulo invertido) não envolve um integrador, há necessidade de incluir um controlador integral. Se este sistema for implementado como um servossistema, seu diagrama de blocos pode tomar a forma mostrada na Fig. 1 (Isto significa que o controlador a ser projetado irá incluir uma retroação de estado e um integrador em malha fechada).

Define-se as variáveis de estado, x_1, x_2, x_3 e x_4, por

$$x_1 = \theta$$
$$x_1 = \dot{\theta}$$
$$x_1 = x$$
$$x_1 = \dot{x}$$

Admitem-se os mesmos valores numéricos para M, m e l. Isto é

$$M = 2\ Kg, \qquad m = 0{,}1\ Kg, \qquad l = 0{,}5\ m$$

Neste sistema, deseja-se manter o ângulo θ tão pequeno quanto possível, na medida em que o carrinho seja movido "em saltos". Considera-se o deslocamento do carrinho como sendo a saída do sistema. Por seguinte, a equação de saída se torna

$$y = [0\ \ 0\ \ 1\ \ 0]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Consequentemente, as equações de estado e de saída para o pêndulo invertido com o carrinho vêm a ser como se segue:

$$\dot{x} = Ax + Bu \quad (7)$$

$$y = Cx + Du \quad (8)$$

onde

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 20{,}601 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0{,}4905 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0{,}5 \end{bmatrix}, \quad C = [0\ \ 0\ \ 1\ \ 0] \text{ e } D = 0$$

Será utilizado o esquema de controle por retroação de estado

$$u = -Kx$$

Como se deve projetar um sistema de controle digital, o primeiro passo é discretizar a equação de estado (7). A discretização pode ser realizada usando-se o seguinte comando do MATLAB:

```
[G, H]=c2d(A,B,T)
```

onde `T` é o período de amostragem envolvido no sistema de controle discreto. Neste problema, admite-se que `T = 0,1 s`. O comando

```
[G, H]=c2d(A,B,0.1)
```

irá transformar as equações de estado do sistema contínuo em equações de estado discretas no tempo.

A representação no espaço de estados para o processo a controlar é agora dada por

$$x(k+1) = Gx(k) + Hu(k)$$
$$y(k) = Cx(k) + Du(k)$$
$$v(k) = v(k-1) + r(k) - y(k)$$
$$u(k) = -Kx(k) + K_1 v(k)$$

Para todo o sistema de controle, q equação de estado e a de saída podem ser dadas por

$$\xi(k+1) = \hat{G}\,\xi(k) + \hat{H}\,w(k)$$
$$w(k) = -\hat{K}\,\xi(k)$$

onde

$$\xi(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix}, \quad \hat{G} = G1 = \begin{bmatrix} G & 0 \\ -CG & 1 \end{bmatrix}, \quad \hat{H} = H1 = \begin{bmatrix} H \\ -CH \end{bmatrix},$$

$$\hat{K} = [K \;\; -K_1] = [k_1 \; k_2 \; k_3 \; k_4 \; -K_1], \quad x_5(k) = v(k)$$

O problema aqui é determinar a matriz **K^**. Admita-se que os pólos desejados em malha fechada estejam localizados como a seguir

$$\mu_1 = 0,9 + j0,25, \quad \mu_2 = 0,9 - j0,25, \quad \mu_3 = 0, \quad \mu_4 = 0, \quad \mu5_1 = 0$$

Note-se que as matrizes **G1** e **H1** são, respectivamente, 5x5 e 5x1, como mostrado em seguida no MATLAB

```
G1=[ G zeros(4,1) ; -C*G 1 ]
G1 =
    1.1048    0.1035         0         0         0
    2.1316    1.1048         0         0         0
   -0.0025   -0.0001    1.0000    0.1000         0
   -0.0508   -0.0025         0    1.0000         0
    0.0025    0.0001   -1.0000   -0.1000    1.0000

H1=[ H ; -C*H]
H1 =
   -0.0051
   -0.1035
    0.0025
    0.0501
   -0.0025
```

A matriz de controlabilidade para todo o sistema é

M=[H1   G1*H1   G1^2*H1   G1^3*H1   G1^4*H1]

O posto de **M** é igual a **5**. Por conseguinte, a alocação arbitrária de pólos é possível.

Em seguida, será apresentado o Programa "Controle.m" para determinação da matriz **K^**, usando a fórmula de Ackerman:

Programa CONTROLE.M - PARTE INICIAL - Projeto de um Controlador Digital

```
%**** Projeto de um controlador digital para o sistema de pêndulo
%     invertido com carro. Este é o programa para calcular KK (a
%     matriz K cincunflexo (K^) ). Esta é uma matriz 1x5 onde as
%     primeiras quatro colunas são de ganho de retroação de estado e
%     a quinta coluna é a constante de ganho integral. A fórmula de
%     Ackermann é utilizada na resolução para a matriz KK
%**** Entrada das variaveis
A=[ 0 1 0 0 ; 20.601 0 0 0 ; 0 0 0 1 ; -0.4905 0 0 0 ]
B=[ 0 ; -1 ; 0 ; 0.5 ]
C=[ 0 0 1 0 ]
D=0

%**** Discretizar a equacao de estado
[G,H]=c2d( A , B , 0.1 )

%**** Entrar com as matrizes
G1=[ G zeros(4,1) ; -C*G 1 ]
H1=[ H ; -C*H]

%**** Entrar com a matriz de controlabilidade M
M=[ H1 G1*H1 G1^2*H1 G1^3*H1 G1^4*H1 ]

%**** Verificar o posto da matriz
rank(M)

%**** Como o posto de M=5, a alocação arbitrária de pólos é possível
%**** Determinar o polinômio característico desejado,
%     entrando com a matriz J e calculando poly(J)

J=[ 0.9+0.25*i 0 0 0 0 ; 0 0.9-0.25*i 0 0 0 ; 0 0 0 0 0; 0 0 0 0 0 ;
0 0 0 0 0 ]
JJ=poly(J)

%**** Entrar com o polinômio característico Phi
Phi=polyvalm(poly(J),G1)

%**** A matriz KK é dada pela equação seguinte
KK=[ 0 0 0 0 1]*(inv(M))*Phi
K1=KK(1)
K2=KK(2)
K2=KK(3)
K4=KK(4)
Kl=-KK(5)
```

Em seguida, será obtida a resposta em degrau do sistema projetado. Usando-se a Eq. (1) como referência, a resposta em degrau unitário é obtida a partir das seguintes equações:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G-HK & HK_1 \\ -CG+CHK & 1-CHK_1 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (9)$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \quad (10)$$

Seja a definição

$$GG = \begin{bmatrix} G - HK & HK_1 \\ -CG + CHK & 1 - CHK_1 \end{bmatrix}, \quad HH = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad CC = [C \;\; 0], \quad DD = [D]$$

As saídas MATLAB para **GG**, **HH**, **CC** e **DD** são mostradas em seguida:

```
GG=[ G-H*K   H*Kl ; -C*G+C*H*K   1-C*H*Kl ]
GG =
   -0.7832   -0.4243   -1.2041   -0.8571    0.3695
  -36.2743   -9.6303  -24.4946  -17.4357    7.5169
    0.9262    0.2595    1.5923    0.5216   -0.1818
   18.5390    5.1937   11.8562    9.4395   -3.6384
   -0.9262   -0.2595   -1.5923   -0.5216    1.1818

HH=[ 0 ; 0 ; 0 ; 0 ; 1 ]
HH =
     0
     0
     0
     0
     1

CC=[ C 0 ]
CC =
     0     0     1     0     0

DD=[ 0 ]
DD =
     0
```

Para obter a resposta em degrau unitário **y(k) versus k**, deve-se primeiro transformar as equações no espaço de estados [Eqs. (9) e (10)] na função de transferência discreta **Y(z)/R(z)** através do uso do seguinte comando:

[num,den]=ss2tf(GG, HH, CC, DD)

A saída do computador segue abaixo

```
[num,den]=ss2tf(GG,HH,CC,DD)
num =
         0    0.0000   -0.1818    0.2180    0.2180   -0.1818

den =
    1.0000   -1.8000    0.8725    0.0000    0.0000    0.0000
```

Seja usado, em seguida, o comando de filtro, como abaixo:

y=filter(num, den, u)

onde **u** é a entrada em degrau unitário.

Para obter a resposta $x_1(k)$, será usada a seguinte equação fictícia de saída:

$$x_1(k) = [1\ 0\ 0\ 0\ 0]\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} = FF\begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \qquad (11)$$

onde

FF=[1 0 0 0 0]

As equações no espaço de estados [Eqs. (9) e (11)] são convertidas, então, na função de transferência discreta $X_1(z)/R(z)$ através do comando

```
[num1, den1]=ss2tf( GG, HH, FF, DD )
```

e usando o comando de filtro

```
x1=filter( num1, den1, u )
```

Similarmente, para se obter as respostas $x_2(k)$, $x_4(k)$ e $x_5(k)$, são usados os seguintes comandos:

```
[num2, den2]=ss2tf( GG, HH, JJ, DD )
x2=filter( num2, den2, u )
[num4, den4]=ss2tf( GG, HH, LL, DD )
x4=filter( num4, den4, u )
```

e

```
[num5, den5]=ss2tf( GG, HH, MM, DD )
x5=filter( num5, den5, u )
```

onde

```
JJ=[0 1 0 0 0]
LL=[0 0 0 1 0]
MM=[0 0 0 0 1]
```

Usando o Programa "Controle.m", é possível obter as respostas **y(k)**, x_1**(k)**, x_2**(k)**, x_4**(k)** e x_5**(k)** para entrada em degrau unitário, **r=1**.

Programa CONTROLE.M - PARTE FINAL - Projeto de um Controlador Digital

```
%******   Controle de Processos - Pêndulo Invertido   ******

%**** Em seguida, será obtida a resposta em degrau do sistema
%     projetado
K=[ KK(1)  KK(2)  KK(3)  KK(4) ]
GG=[ G-H*K H*Kl ; -C*G+C*H*K 1-C*H*Kl ]
HH=[ 0 ; 0 ; 0 ; 0 ; 1 ]
CC=[ C 0 ]
DD=[ 0 ]

%**** Para obter a resposta em degrau unitário, deve-se primeiro
%     transformar as equações no espaço de estados na função de
%     transferência discreta
[num,den]=ss2tf(GG,HH,CC,DD)

%**** Entrada das variaveis
FF=[ 1 0 0 0 0 ]
LL=[ 0 0 0 1 0 ]
MM=[ 0 0 0 0 1 ]
JJ=[ 0 1 0 0 0 ]

%**** Comandos para obter a resposta em degrau unitário
r=ones(1,101);
axis([ 0 100 -0.5 2 ]);
k=0:100;
y=filter(num,den,r);
plot(k,y,'bo',k,y,'r-')
grid
whitebg
title('Posição do carro y(k)=x3(k)')
xlabel('k')
ylabel('y(k)=x3(k)')

%**** Para obter x1(k), converter as equações no espaço de estados na
%     função de transferência discreta X1(z)/R(z)
[num1,den1]=ss2tf(GG,HH,FF,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -1 1 ])
x1=filter(num1,den1,r);
plot(k,x1,'bo',k,x1,'-')
grid
title('Deslocamento angular Teta: x1(k)')
xlabel('k')
ylabel('x1(k)')
```

Programa CONTROLE.M - PARTE FINAL - Projeto de um Controlador Digital

```
%**** Para obter x2(k), converter as equações no espaço de estados na
%     função de transferência discreta X2(z)/R(z)
[num2,den2]=ss2tf(GG,HH,JJ,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -5 10 ])
x2=filter(num2,den2,r);
plot(k,x2,'bo',k,x2,'-')
grid
title('Deslocamento angular Teta: x2(k)')
xlabel('k')
ylabel('x2(k)')

%**** Para obter x4(k), converter as equações no espaço de estados na
%     função de transferência discreta X4(z)/R(z)
[num4,den4]=ss2tf(GG,HH,LL,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -5 5 ])
x4=filter(num4,den4,r);
plot(k,x4,'bo',k,x4,'-')
grid
title('Velocidade do Carro: x4(k)')
xlabel('k')
ylabel('x4(k)')

%**** Para obter x5(k), converter as equações no espaço de estados na
%     função de transferência discreta X5(z)/R(z)
[num5,den5]=ss2tf(GG,HH,MM,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -2 8 ])
x5=filter(num5,den5,r);
plot(k,x5,'bo',k,x5,'-')
grid
title('Saída do Integrador: x5(k)=v(k)')
xlabel('k')
ylabel('x5(k)=v(k)')
```

A Fig. 3 mostra a posição do carro $y(k)$ quando o sistema é submetido a uma entrada em degrau unitário. Dessa figura pode ser visto que, quando uma entrada é fornecida, o carro se move na direção oposta àquela da entrada por um pequeno período (aproximadamente 0,4s ou isso) e, em seguida, move-se na direção da entrada. Tal resposta é um comportamento típico de sistemas de fase não-mínima. O fato de este sistema ser de fase não-mínima pode ser observado a partir da análise seguinte.

Anteriormente, foram obtidos o numerador e o denominador da função de transferência como se segue:

$$\frac{num}{den} = \frac{-0,1818\,z^3 + 0,2180\,z^2 + 0,2180\,z - 0,1818}{z^5 - 1,8\,z^4 + 0,8725\,z^3}$$
$$= \frac{-0,1818\,z^{-2} + 0,2180\,z^{-3} + 0,2180\,z^{-4} - 0,1818\,z^{-5}}{1 - 1,8\,z^{-1} + 0,8725\,z^{-2}}$$

Como o coeficiente de termo z^{-2} no numerador é negativo, a resposta em degrau unitário de 0 até -0,1818 em três passos, como se segue

$$y(0)=0$$
$$y(1)=0$$
$$y(2)=-0,1818$$

A saída $y(x)$ na resposta em degrau unitário se aproxima de um, porque o valor final de $y(\infty)$ pode ser obtido como

$$y(\infty) = \lim_{z\to 1}\left(1 - z^{-1}\right)\frac{-0,1818\,z^{-2} + 0,2180\,z^{-3} + 0,2180\,z^{-4} - 0,1818\,z^{-5}}{(1 - 1,8\,z^{-1} + 0,8725\,z^{-2})(1 - z^{-1})}$$
$$= 1$$

Por conseguinte, apesar de a resposta começar na direção negativa e se manter na região negativa por um curto período, a resposta vem de volta para a região positiva e eventualmente se aproxima da unidade. Esta é uma espécie típica de sistema de fase não-mínima.

A Fig. 4 mostra o deslocamento angular do pêndulo [$x_1(k)$ versus k]. A Fig. 5 retrata a velocidade angular do pêndulo [$x_2(k)$ versus k]. A Fig. 6 mostra a velocidade do carro [$x_4(k)$ versus k]. A saída do integrador [$v(k)$ versus k] é mostrada na Fig. 7.

A partir das Figs. 3 - 7, vê-se que as variáveis de estado quase atingem o regime estacionário para $k > 70$. Como o período de amostragem T é $0,1s$, o regime estacionário é atingido em aproximadamente sete segundos.

FIGURA 3

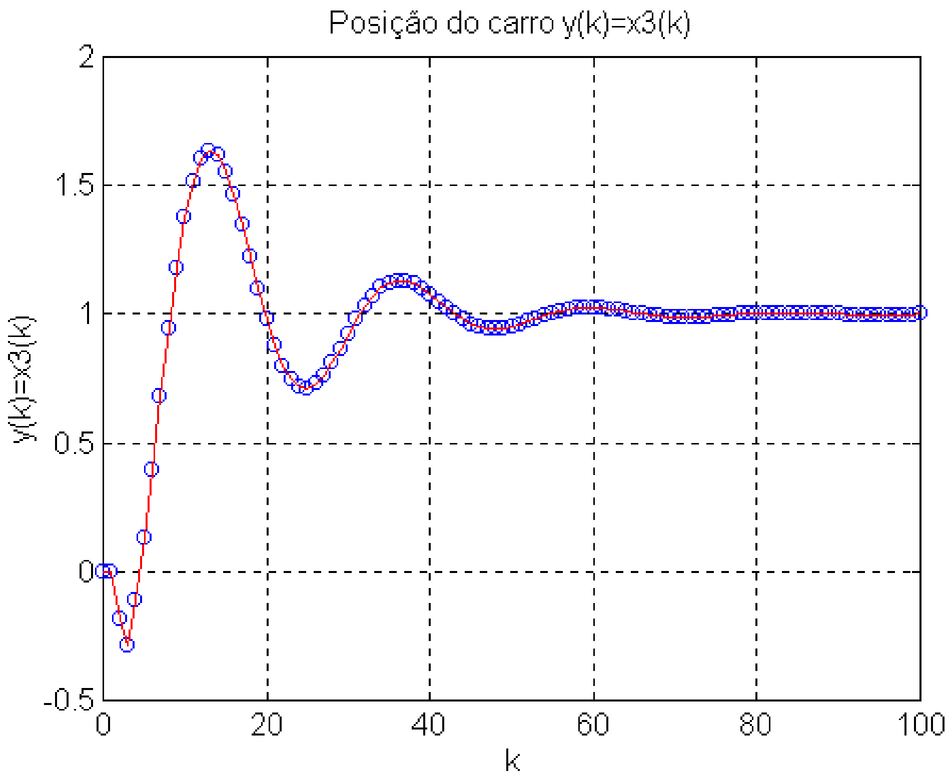


FIGURA 4

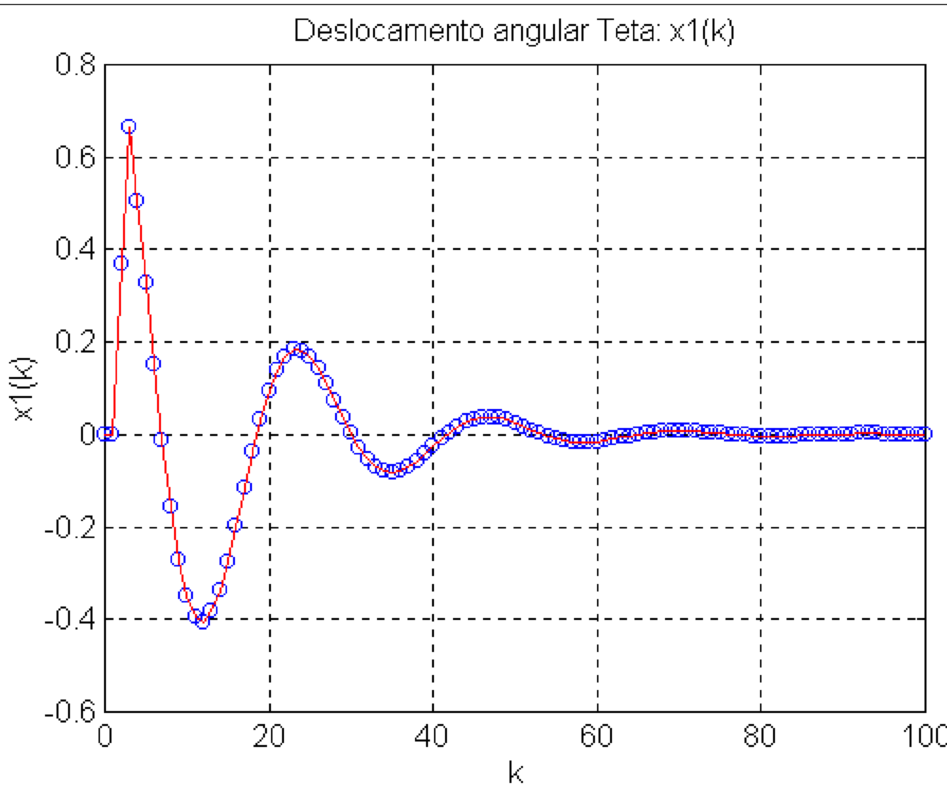


FIGURA 5

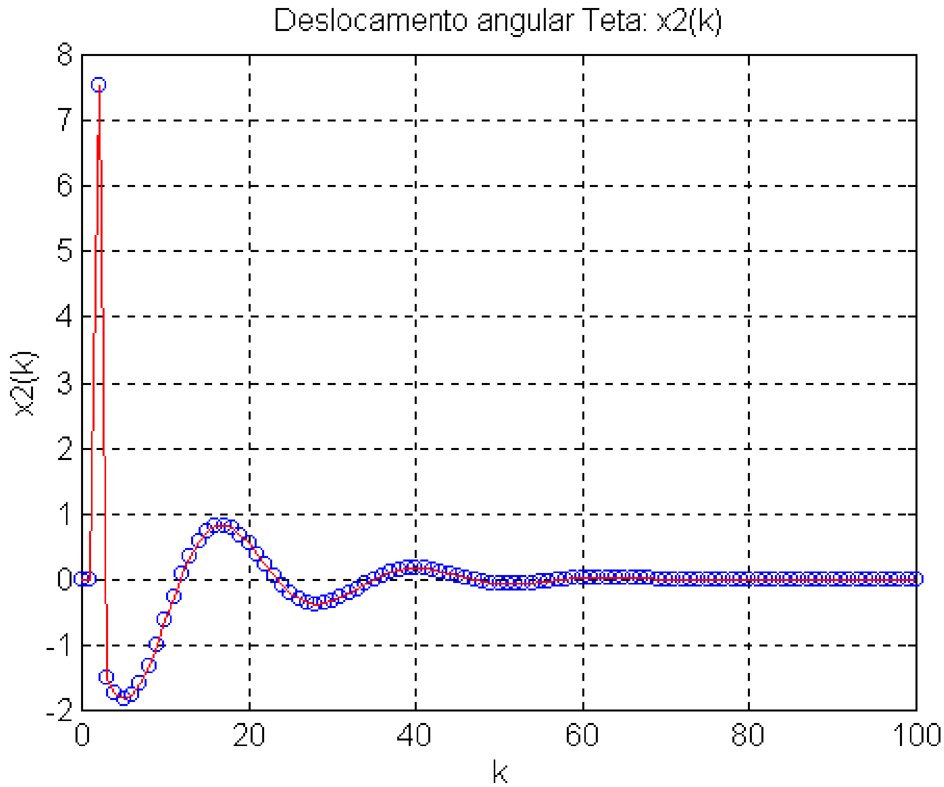


FIGURA 6

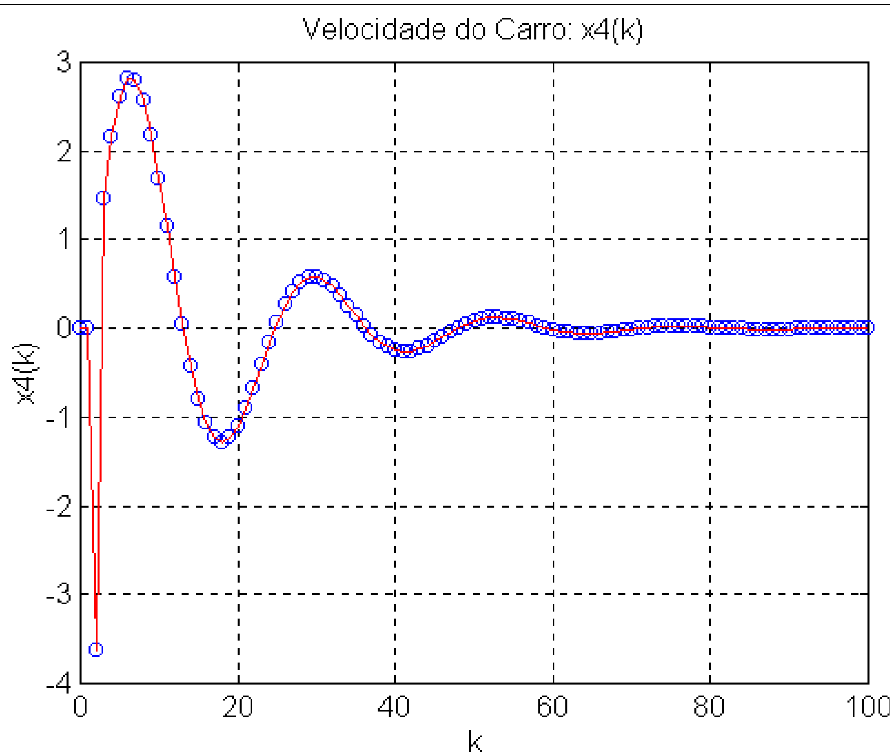
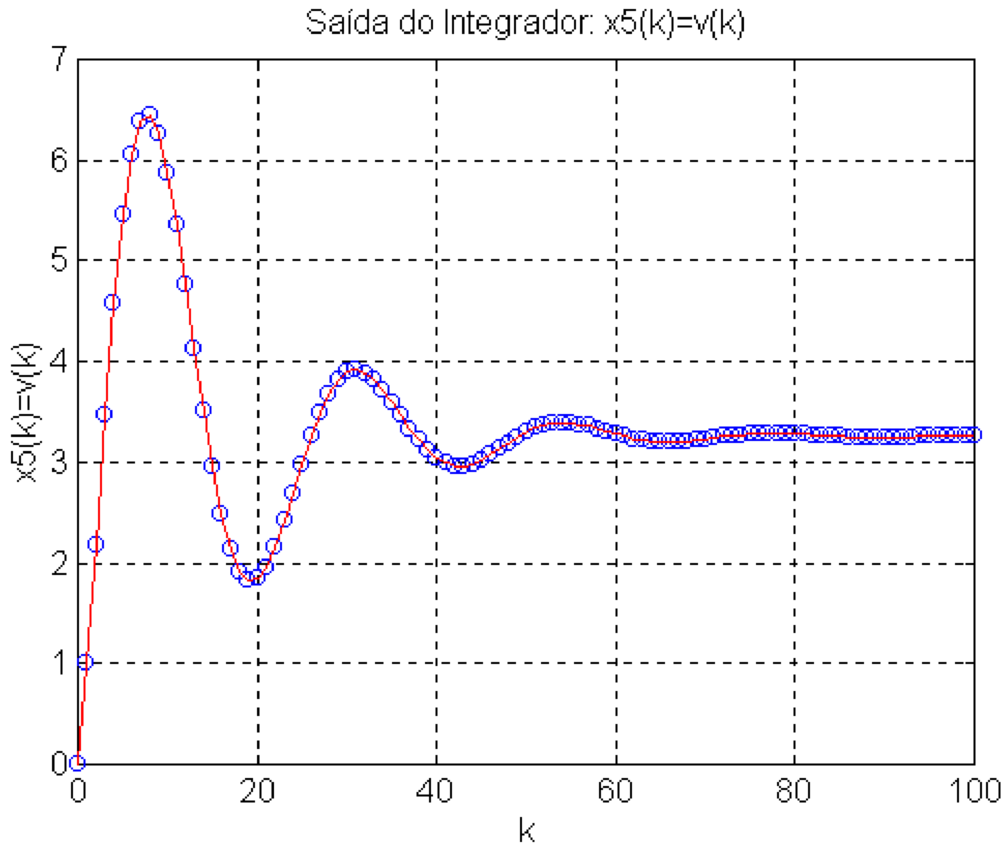


FIGURA 7



Aqui apresenta-se o programa completo do MATLAB, "Controle.m"

CONTROLE.M - Projeto de um Controlador Digital para o sistema de Pêndulo Invertido

```
%**** Projeto de um controlador digital para o sistema de pêndulo
%     invertido com carro. Este é o programa para calcular KK (a
%     matriz K cincunflexo (K^) ). Esta é uma matriz 1x5 onde as
%     primeiras quatro colunas são de ganho de retroação de estado e
%     a quinta coluna é a constante de ganho integral. A fórmula de
%     Ackermann é utilizada na resolução para a matriz KK

%**** Entrada das variaveis
A=[ 0 1 0 0 ; 20.601 0 0 0 ; 0 0 0 1 ; -0.4905 0 0 0 ]
B=[ 0 ; -1 ; 0 ; 0.5 ]
C=[ 0 0 1 0 ]
D=0

%**** Discretizar a equacao de estado
[G,H]=c2d( A , B , 0.1 )

%**** Entrar com as matrizes
G1=[ G zeros(4,1) ; -C*G 1 ]
H1=[ H ; -C*H]

%**** Entrar com a matriz de controlabilidade M
M=[ H1 G1*H1 G1^2*H1 G1^3*H1 G1^4*H1 ]

%**** Verificar o posto da matriz
rank(M)

%**** Como o posto de M=5, a alocação arbitrária de pólos é possível
%**** Determinar o polinômio característico desejado,
%     entrando com a matriz J e calculando poly(J)

J=[ 0.9+0.25*i 0 0 0 0 ; 0 0.9-0.25*i 0 0 0 ; 0 0 0 0 0; 0 0 0 0 0 ;
0 0 0 0 0 ]
JJ=poly(J)

%**** Entrar com o polinômio característico Phi
Phi=polyvalm(poly(J),G1)

%**** A matriz KK é dada pela equação seguinte
KK=[ 0 0 0 0 1]*(inv(M))*Phi
K1=KK(1)
K2=KK(2)
K2=KK(3)
K4=KK(4)
Kl=-KK(5)

%**** Em seguida, será obtida a resposta em degrau do sistema
%     projetado
K=[ KK(1)  KK(2)  KK(3)  KK(4) ]
GG=[ G-H*K H*Kl ; -C*G+C*H*K 1-C*H*Kl ]
HH=[ 0 ; 0 ; 0 ; 0 ; 1 ]
CC=[ C 0 ]
DD=[ 0 ]
```

CONTROLE.M - Projeto de um Controlador Digital para o sistema de Pêndulo Invertido

```
%**** Para obter a resposta em degrau unitário, deve-se primeiro
%     transformar as equações no espaço de estados na função de
%     transferência discreta
[num,den]=ss2tf(GG,HH,CC,DD)

%**** Entrada das variaveis
FF=[ 1 0 0 0 0 ]
LL=[ 0 0 0 1 0 ]
MM=[ 0 0 0 0 1 ]
JJ=[ 0 1 0 0 0 ]

%**** Comandos para obter a resposta em degrau unitário
r=ones(1,101);
axis([ 0 100 -0.5 2 ]);
k=0:100;
y=filter(num,den,r);
plot(k,y,'bo',k,y,'r-')
grid
whitebg
title('Posição do carro y(k)=x3(k)')
xlabel('k')
ylabel('y(k)=x3(k)')

pause

%**** Para obter x1(k), converter as equações no espaço de estados na
%     função de transferência discreta X1(z)/R(z)
[num1,den1]=ss2tf(GG,HH,FF,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -1 1 ])
x1=filter(num1,den1,r);
plot(k,x1,'bo',k,x1,'-')
grid
title('Deslocamento angular Teta: x1(k)')
xlabel('k')
ylabel('x1(k)')

pause

%**** Para obter x2(k), converter as equações no espaço de estados na
%     função de transferência discreta X2(z)/R(z)
[num2,den2]=ss2tf(GG,HH,JJ,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -5 10 ])
x2=filter(num2,den2,r);
plot(k,x2,'bo',k,x2,'-')
grid
title('Deslocamento angular Teta: x2(k)')
xlabel('k')
ylabel('x2(k)')
```

CONTROLE.M - Projeto de um Controlador Digital para o sistema de Pêndulo Invertido

```
%**** Para obter x4(k), converter as equações no espaço de estados na
%     função de transferência discreta X4(z)/R(z)
[num4,den4]=ss2tf(GG,HH,LL,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -5 5 ])
x4=filter(num4,den4,r);
plot(k,x4,'bo',k,x4,'-')
grid
title('Velocidade do Carro: x4(k)')
xlabel('k')
ylabel('x4(k)')

pause

%**** Para obter x5(k), converter as equações no espaço de estados na
%     função de transferência discreta X5(z)/R(z)
[num5,den5]=ss2tf(GG,HH,MM,DD)

%**** Comandos para obter a resposta em degrau unitário
axis([ 0 100 -2 8 ])
x5=filter(num5,den5,r);
plot(k,x5,'bo',k,x5,'-')
grid
title('Saída do Integrador: x5(k)=v(k)')
xlabel('k')
ylabel('x5(k)=v(k)')

pause
```