Sistemas de Control Automático

9. Diseño de Observadores de Estados

Profesor: **Luis Enrique González Jiménez**

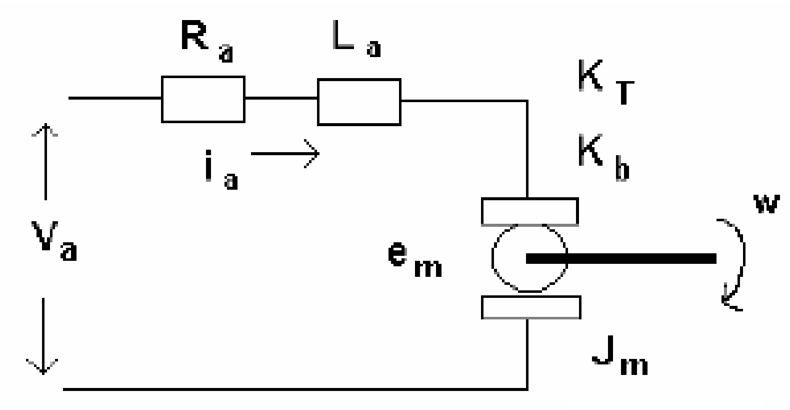
Departamento de Electrónica, Sistemas e Informática (DESI)

Hora: Ma-Vi [16:00 - 18:00]

Aula: T-201/203

¿Porqué es necesario un observador?

Recordemos el modelo en espacio de estados del motor de CD.



$$\Rightarrow \quad \dot{x}(t) = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} \\ \frac{K_T}{J_m} & -\frac{B_m}{J_m} \end{bmatrix} x(t) + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t)$$

Con los estados y la entrada definidos como $x(t) = \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix}, \quad u(t) = V_a(t)$

Las leyes de control que hemos visto en el curso (a excepción del PID) tienen las siguientes estructuras:

$$u(t) = Kx(t) + Fr, \quad u(t) = (CB)^{-1}\left[\dot{y}_{ref} - CAx(t) - K_e e(t)\right]$$

¿Qué característica comparten estos dos algoritmos de control?

Ambas necesitan el conocimiento total del vector de estados $x(t)$

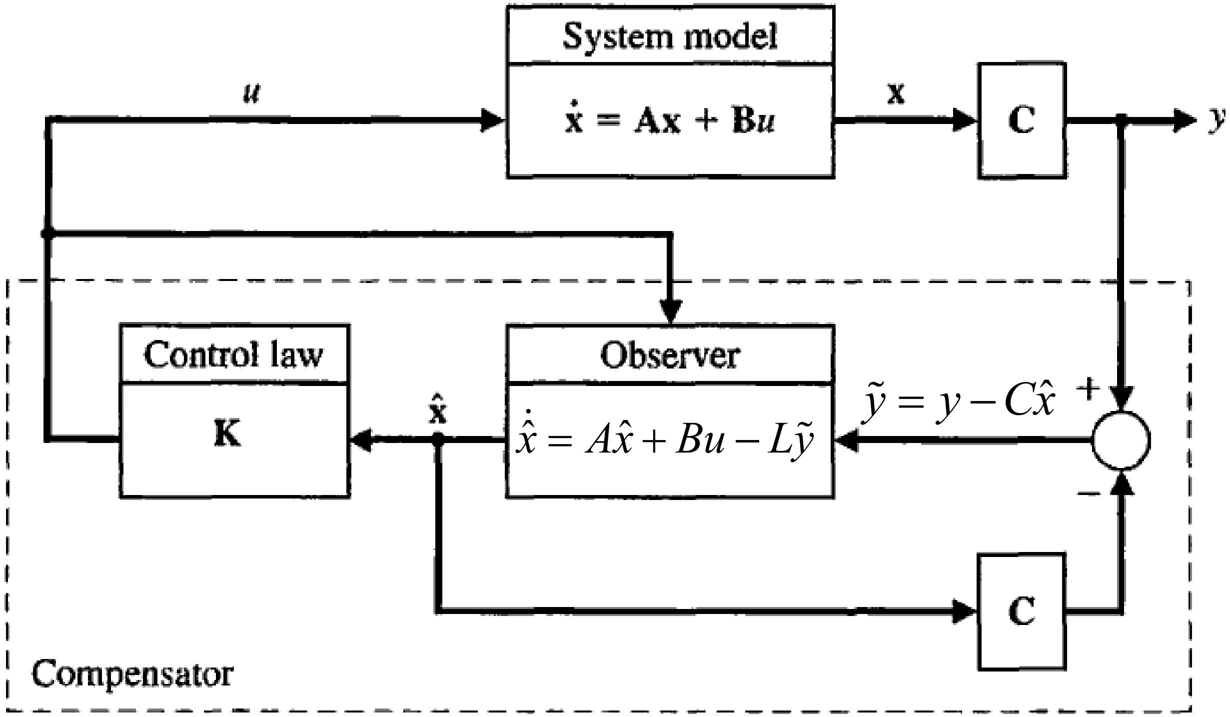
¿Es siempre esto posible? Veamos…

Si seguimos analizando el ejemplo del motor de CD, deberíamos tener sensores para la velocidad angular del eje ($\omega(t)$) y para la corriente de armadura ($i_a(t)$).

Por falta de recursos, espacio, simplicidad en la estructura física, etc. podríamos no tener acceso a los dos sensores a la vez. ¿Entonces como implementar los excelsos controladores vistos en clase?

Si conocemos la estructura del sistema a controlar (en este caso, $A, B, C \text{ y } D$) podemos diseñar un OBSERVADOR DE ESTADOS para estimar los estados que no se estén sensando.

En espacio de estados se puede diseñar un <u>observador de estados</u>, para luego retroalimentar al sistema:



$x =$ Estado

$y =$ Salida

$\hat{x} =$ Estimación del Estado

$C\hat{x} =$ Salida del Observador

La idea detrás del observador es proponer una <u>*copia*</u> del sistema original que utilice las entradas y salidas del mismo, de tal forma que cuando la salida de ambos sistemas sean iguales $y = C\hat{x} \quad \Rightarrow \quad \tilde{y} = 0$**, la dinámica de ambos sistemas también lo sea, y por lo tanto sus estados** $x = \hat{x}$ **.**

Si definimos el error de estimación como $e_o = x - \hat{x}$ **, podemos obtener la dinámica del error como:** $\dot{e}_o = \dot{x} - \dot{\hat{x}}$

$$
\begin{aligned}
&= Ax + Bu - \left(A\hat{x} + Bu - L\tilde{y} \right) \\
&= Ae_0 + L\left(y - C\hat{x} \right) \\
&= Ae_0 + L\left(Cx - C\hat{x} \right) \\
&= \left(A + LC \right) e_0
\end{aligned}
$$

Entonces, podemos diseñar la matriz L **de forma que** $A + LC$ **tenga polos estables. En caso de lograr esto aseguramos que** $e_o \to 0, \quad \hat{x} \to x$ **.**

Pero… ¿Cómo diseñar L**?**

Nosotros ya aprendimos a diseñar K **para que** $A + BK$ **tenga polos deseados (siempre que el sistema fuera controlable).**

Usaremos el mismo método con algunas variantes.

Debemos asignar *n* polos estables a la dinámica del observador (que tan rápido el estado estimado convergerá al estado real)

$$\{p_{do1} \quad p_{do2} \quad p_{do3} \quad \cdots \quad p_{don}\}$$

La ganancia del observador será $\boxed{L = -H_o(A)M_o^{-1}[0 \quad 0 \quad \cdots \quad 1]^T}$

$$M_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}, \quad H_o(A) = (A - p_{do1}I)(A - p_{do2}I)\cdots(A - p_{don}I)$$

Entonces, la dinámica del observador queda: $\dot{\hat{x}} = A\hat{x} + Bu - L\tilde{y}$

FÓRMULACIÓN DE ACKERMAN

NOTA: La ubicación de polos del observador solo será posible si se cumple cualquiera de las siguientes condiciones:

$$\operatorname{rank}(M_o) = n, \quad \det(M_o) \neq 0 \quad M_o \text{ es invertible}$$

En cuyo caso se dice que el <u>sistema es observable</u>, y a M_o se le conoce como Matriz de Observabilidad.

Supongamos que tenemos un sensor para la velocidad del motor, $\omega(t)$, y queremos estimar la corriente de armadura, $i_a(t)$.

$$x(t)=\begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix} \quad \Rightarrow \quad \dot{x}(t)=\begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} \\ \frac{K_T}{J_m} & -\frac{B_m}{J_m} \end{bmatrix} x(t)+\begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} u(t)$$

$$u(t)=V_a(t) \qquad y(t)=\begin{bmatrix} 0 & 1 \end{bmatrix} x(t)$$

Con los parámetros:

$$R_a=2.2\ \Omega, \quad L_a=6.3\text{ mH}, \quad B_m=0.015, \quad J_m=0.0236, \quad K_T=1.78$$

El modelo resulta en:

$$A=\begin{bmatrix} -91.53 & -60 \\ 0.97 & -0.12 \end{bmatrix}, B=\begin{bmatrix} 76.9 \\ 0 \end{bmatrix}, C=\begin{bmatrix} 0 & 1 \end{bmatrix}$$

PASO 1: Comprobamos observabilidad del sistema.

$$rank(M_o)=rank\left(\begin{bmatrix} C \\ CA \end{bmatrix}\right)=rank\left(\begin{bmatrix} 0 & 1 \\ 0.97 & -0.12 \end{bmatrix}\right)=2 \qquad \boxed{\text{¡ Sistema Observable !}}$$

PASO 2: **Definimos polos deseados para la dinámica del error de estimación.**

De la misma forma que en el controlador, estos polos deben ser estables para que el error de estimación converja a cero. La recomendación es que deben ser polos más rápidos que los obtenidos para el controlador.

$$P_{dO} = \begin{bmatrix} -4 & -4 \end{bmatrix}$$

PASO 3: **Calculemos la matriz de ganancias del observador.**

$$H_o(A) = (A + 4I)^2$$

$$\boxed{L = -H_o(A) M_o^{-1} \begin{bmatrix} 0 & 1 \end{bmatrix}^T = \begin{bmatrix} -7799.5 \\ 83.7 \end{bmatrix}}$$

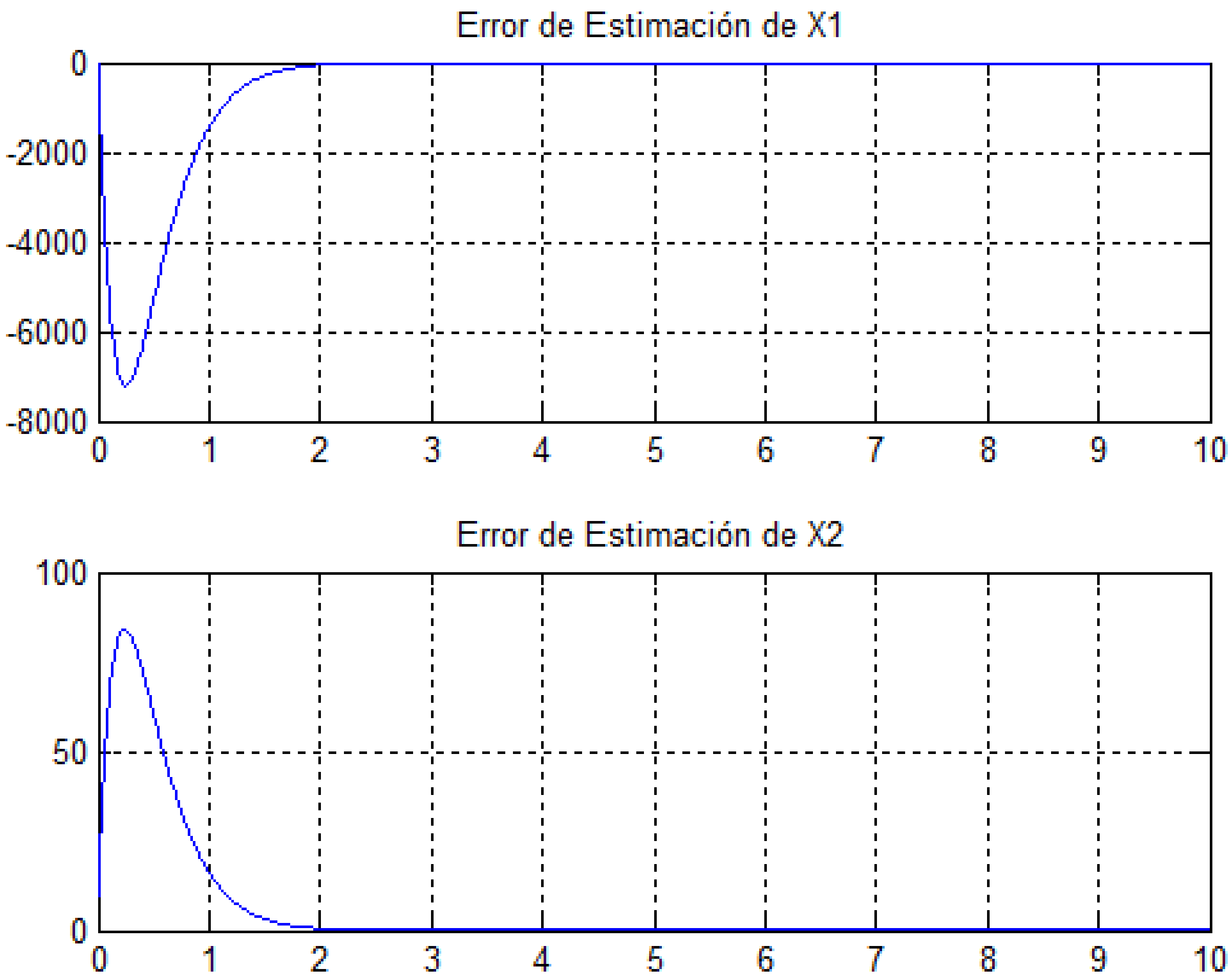
PASO 4: **Implementarlo en Matlab para simular…**

Código .m

```
function MotorCD_OBS_plot( tspan, x0, Pdo)

global A B C L

%Parámetros del motor
Ra = 1.19;       La = 0.013;      Kb = 0.78;
Jm = 0.8;        Bm = 0.1;          Kt = 0.78;

%Matrices del sistema
A = [-Ra/La -Kb/La;Kt/Jm -Bm/Jm]
B = [1/La;0]
C = [0 1];

Mo = [C;C*A]
Rango_de_Mo = rank(Mo)

Ho = (A - Pdo(1)*eye(2))*(A - Pdo(2)*eye(2));
L  = -Ho*inv(Mo)*[0;1]

%RESUELVE LAS ECUACIONES DIFERENCIALES ORDINARIAS (ODE'S)
[t, X] = ode45(@MotorCD_OBS_sys, tspan, [x0 0 0]);

%Graficando
figure;
subplot(2,1,1); plot(t, X(:,1), t, X(:,3), '--r');  title('ESTADO 1 Y ESTIMACIÓN');  grid;
subplot(2,1,2); plot(t, X(:,2), t, X(:,4), '--r');  title('ESTADO 2 Y ESTIMACIÓN');  grid;

figure;
subplot(2,1,1); plot(t, X(:,1) - X(:,3));  title('Error de Estimación de X1');  grid;
subplot(2,1,2); plot(t, X(:,2) - X(:,4));  title('Error de Estimación de X2');  grid;

end
```

```
function dX = MotorCD_OBS_sys( t, X)

global A B C L K

x = X(1:2); %ESTADOS DEL SISTEMA
xo = X(3:4); %ESTADOS DEL OBSERVADOR

ref = 50; dref = 0;

e = ref - C*x;
% Ley de control
U = 0;%(C*B)^-1*(dref - C*A*x + K*e);

%Observador
Y = C*x;        %Salida orifinal del sistema
Ye = Y - C*xo;  %Error de Observación

%ODE's
dx = A*x  + B*U;
dxo = A*xo + B*U - L*Ye;

dX = [dx ; dxo];
```

¡ Los estados estimados convergen a los reales !



Probemos con distintas condiciones iniciales y polos deseados del estimador.

1) **Verificar Controlabilidad** $\operatorname{rank}(M_c) = n$

2) **Diseñar la ganancia del controlador usando Ackerman para los polos deseados** $\{p_{dc1} \quad p_{dc2} \quad \cdots \quad p_{dcn}\}$ **y cumpliendo los objetivos de control:**

$$K = -\begin{bmatrix}0 & 0 & \cdots & 1\end{bmatrix} M_c^{-1} H(A)$$

3) **Verificar Observabilidad** $\operatorname{rank}(M_o) = n$

4) **Diseñar la ganancia del observador usando Ackerman, de modo que los polos deseados del observador sean** $\{p_{do1} \quad p_{do2} \quad \cdots \quad p_{don}\}$ **como:**

$$L = -H_o(A) M_o^{-1} \begin{bmatrix}0 & 0 & \cdots & 1\end{bmatrix}^T$$

5) **Conecte la salida del observador a la entrada del sistema mediante:** $u = K\hat{x}$

Cuando tenemos el observador + controlador en lazo cerrado la dinámica del sistema total queda:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + BK\hat{x} \\ &= Ax(t) + BK\left(x(t) - e_o\right) \\ &= \left(A + BK\right)x(t) - BKe_o\end{aligned}$$

Definiendo el error de observación como una ampliación del estado nos queda:

$$\begin{bmatrix}\dot{x}(t) \\ \dot{e}_o\end{bmatrix} = \begin{bmatrix}A+BK & -BK \\ 0 & A+LC\end{bmatrix}\begin{bmatrix}x(t) \\ e_o\end{bmatrix}$$

Podemos observar que es una matriz triangular superior por bloques, por lo que los polos del sistema global son iguales a los polos del controlador mas los polos del observador.

$$\operatorname{eigs}\left(\begin{bmatrix}A+BK & -BK \\ 0 & A+LC\end{bmatrix}\right) = \operatorname{eigs}\left(A+BK\right) + \operatorname{eigs}\left(A+LC\right)$$

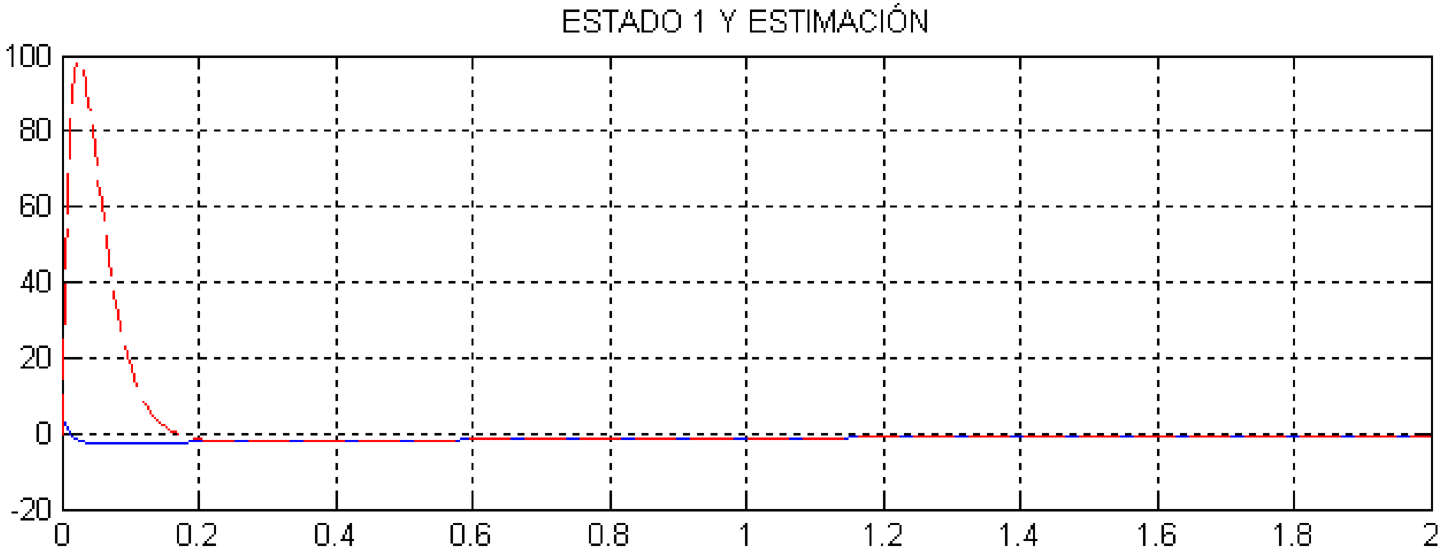
Esto significa que se puede diseñar el controlador y el observador de forma independiente. A esto se le llama <u>*principio de separación*</u> y siempre se cumple para sistemas lineales.

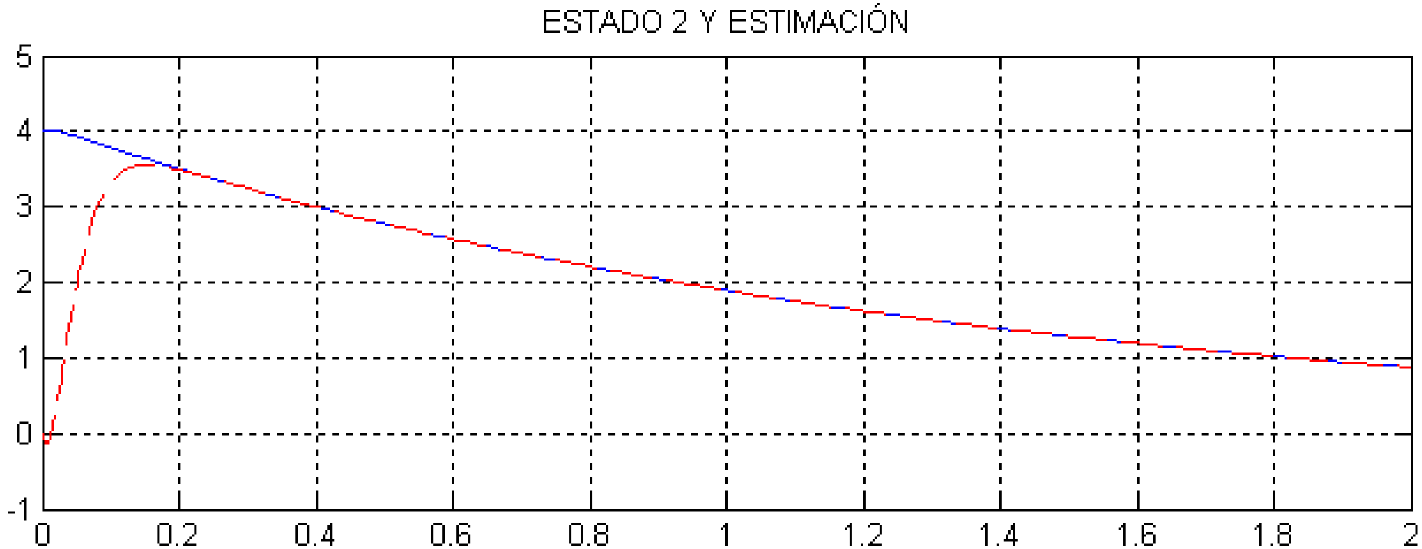
“¡ A como observo, doy !”… OBSERVADOR + CONTROL en Motor de CD

```
function MotorCD_OBS_CONT_plot( tspan, x0, Pdo, Pdc)

global A B C L K F
%Parámetros del motor
Ra = 1.19;       La = 0.013;      Kb = 0.78;
Jm = 0.8;        Bm = 0.1;          Kt = 0.78;
%Matrices del sistema
A = [-Ra/La -Kb/La;Kt/Jm -Bm/Jm];   B = [1/La;0];   C = [0 1];

%DISEÑO DEL OBSERVADOR
Mo = [C;C*A]
Rango_de_Mo = rank(Mo)
Ho = (A - Pdo(1)*eye(2))*(A - Pdo(2)*eye(2));
L  = -Ho*inv(Mo)*[0;1]

%DISEÑO DEL CONTROLADOR
Mc = [B A*B]
Rango_de_Mc = rank(Mc)
Hc = (A - Pdc(1)*eye(2))*(A - Pdc(2)*eye(2));
K  = -[0 1]*inv(Mc)*Hc
F  = 1/(C*inv(-A-B*K)*B)

%RESUELVE LAS ECUACIONES DIFERENCIALES ORDINARIAS (ODE'S)
[t, X] = ode45(@MotorCD_OBS_CONT_sys, tspan, [x0 0 0]);

%Graficando
figure;
subplot(2,1,1); plot(t, X(:,1), t, X(:,3), '--r');  title('ESTADO 1 Y ESTIMACIÓN');  grid;
subplot(2,1,2); plot(t, X(:,2), t, X(:,4), '--r');  title('ESTADO 2 Y ESTIMACIÓN');  grid;

figure;
subplot(2,1,1); plot(t, X(:,1) - X(:,3));  title('Error de Estimación de X1');  grid;
subplot(2,1,2); plot(t, X(:,2) - X(:,4));  title('Error de Estimación de X2');  grid;

end
```

```
function dX = MotorCD_OBS_CONT_sys( t, X)

global A B C L K F

x = X(1:2); %ESTADOS DEL SISTEMA
xo = X(3:4); %ESTADOS DEL OBSERVADOR

ref = 50; dref = 0;

% Ley de control
%U = 0;               %CONDICIONES INICIALES
%U = K*x;             %ESTABILIZACIÓN
U = K*xo + F*ref; %SEG DE REF

%Observador
Y = C*x;        %Salida orifinal del sistema
Ye = Y - C*xo;  %Error de Observación

%ODE's
dx = A*x  + B*U;
dxo = A*xo + B*U - L*Ye;

dX = [dx ; dxo];
```

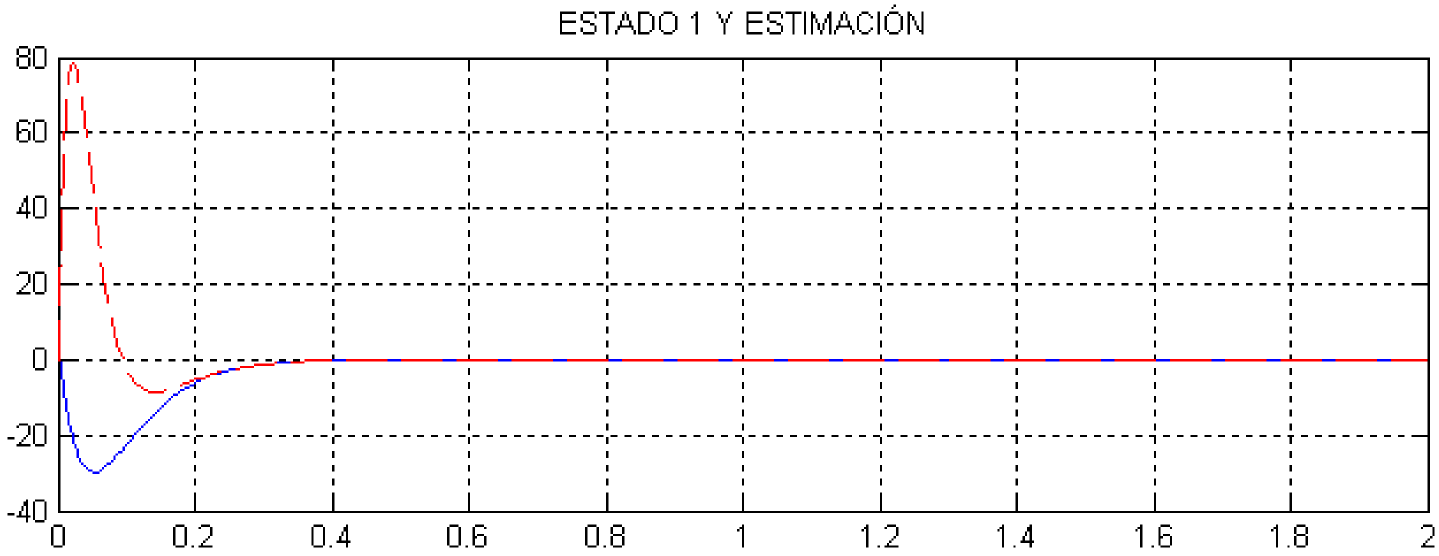
$$x(0) = [5 \quad 4]^T, \quad P_{do} = [-40 \quad -40], \quad u = 0$$

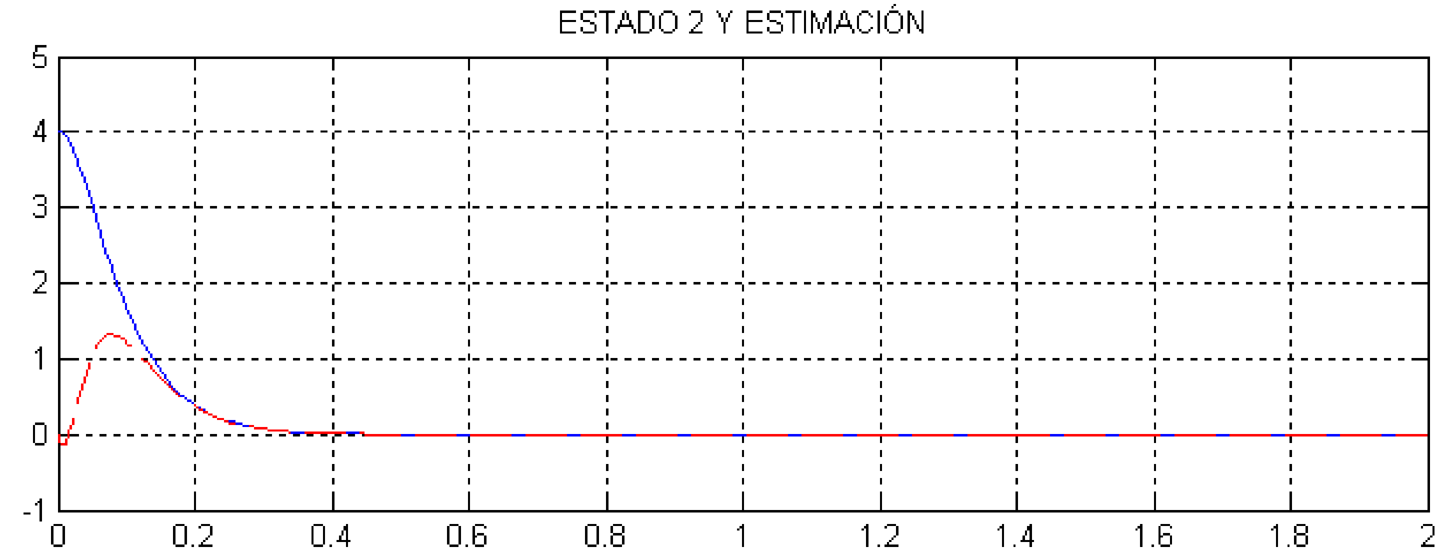




¡ CONDICIONES INICIALES !

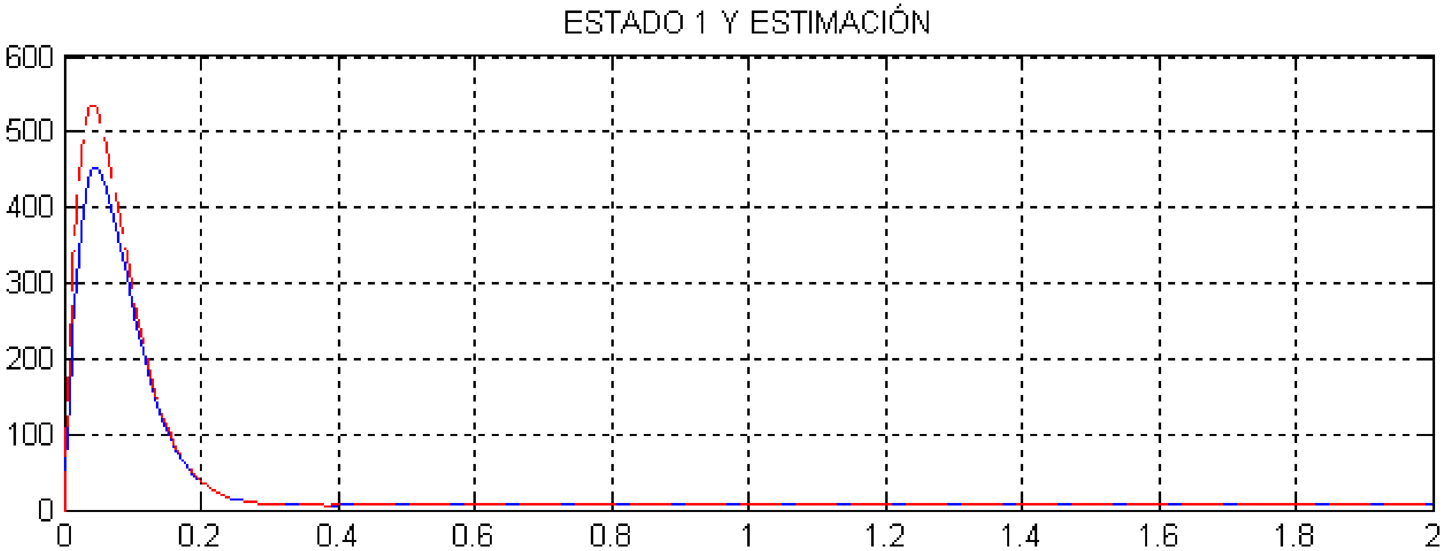
$$x(0)=\begin{bmatrix}5 & 4\end{bmatrix}^T, \quad P_{do}=\begin{bmatrix}-40 & -40\end{bmatrix}, \quad P_{dc}=\begin{bmatrix}-20 & -20\end{bmatrix}, \quad u=K\hat{x}(t)$$

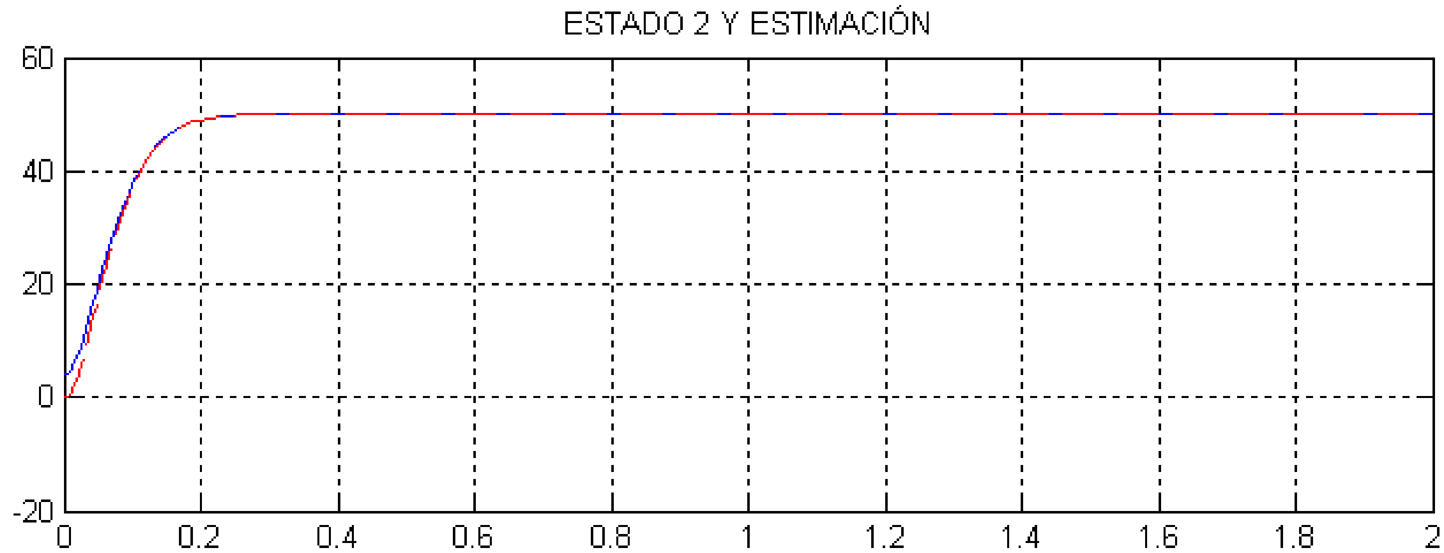




¡ ESTABILIZACIÓN !

$$x(0)=[5 \quad 4]^T, \quad P_{do}=[-40 \quad -40], \quad P_{dc}=[-20 \quad -20], \quad u=K\hat{x}(t)+Fr(t)$$





¡ ESTABILIZACIÓN Y SEGUIMIENTO DE REFERENCIAS CONSTANTES!

Elija una salida y diseñe un controlador IMPLEMENTABLE para que esa salida siga la referencia $y_{ref} = 20$ y los demás estados se estabilicen.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 \\ 2 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} u(t)$$

Tenemos el siguiente sistema con tres estados, dos entradas y dos salidas:

$$\dot{X}(t) = \underbrace{\begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & -1 \\ 1 & -1 & 0 \end{bmatrix}}_{A} X(t) + \underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & -3 \end{bmatrix}}_{B} U(t)$$

$$\boxed{X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}, \quad U(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}}$$

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{C} X(t)$$

Diseñe un controlador que logre que las salidas del sistema sigan las siguientes referencias

$$r(t) = \begin{bmatrix} y_{1,ref}(t) \\ y_{2,ref}(t) \end{bmatrix} = \begin{bmatrix} 2+\sin(t) \\ 2+\cos(t) \end{bmatrix}$$

Solución: Definamos la variable de error como: $e(t) = r(t) - y(t)$

Derivando, obtenemos la dinámica de error como:

$$\begin{aligned} \dot{e}(t) &= \dot{r}(t) - \dot{y}(t) \\ &= \dot{r}(t) - C\left[AX(t) + BU(t)\right] \end{aligned}$$

El controlador por retro de error queda definido como:

$$U(t)=[CB]^{-1}\left(-CAX(t)+\dot{r}(t)-Ke(t)\right)$$

Sin embargo, para poder implementar el controlador diseñado, es necesario conocer los estados del sistema. Pero, ¿Cómo hacerlo con dos salidas?

Elegimos una salida (la que permita diseñar el observador) como: $C_1=\begin{bmatrix}1 & 0 & 0\end{bmatrix}$

Entonces, la matriz de observabilidad queda como:

$$M_O=\begin{bmatrix}C_1\\ C_1A\\ C_1A^2\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 1 & 0 & 2\\ 3 & -2 & 2\end{bmatrix}\Rightarrow rank(M_O)=3$$

Ya que el sistema es observable, obtenemos la ganancia del observador como:

$$H_o(A)=(A+4I)^3,\quad L=-H_o(A)M_o^{-1}\begin{bmatrix}0 & 0 & \cdots & 1\end{bmatrix}^T=\begin{bmatrix}-14\\ -71.77\\ 11.11\end{bmatrix}$$

Y el observador queda definido como:

$$\boxed{\dot{\hat{x}}=A\hat{x}+Bu-L\tilde{y}}\ \text{donde}\quad \tilde{y}=C_1x-C_1\hat{x}$$

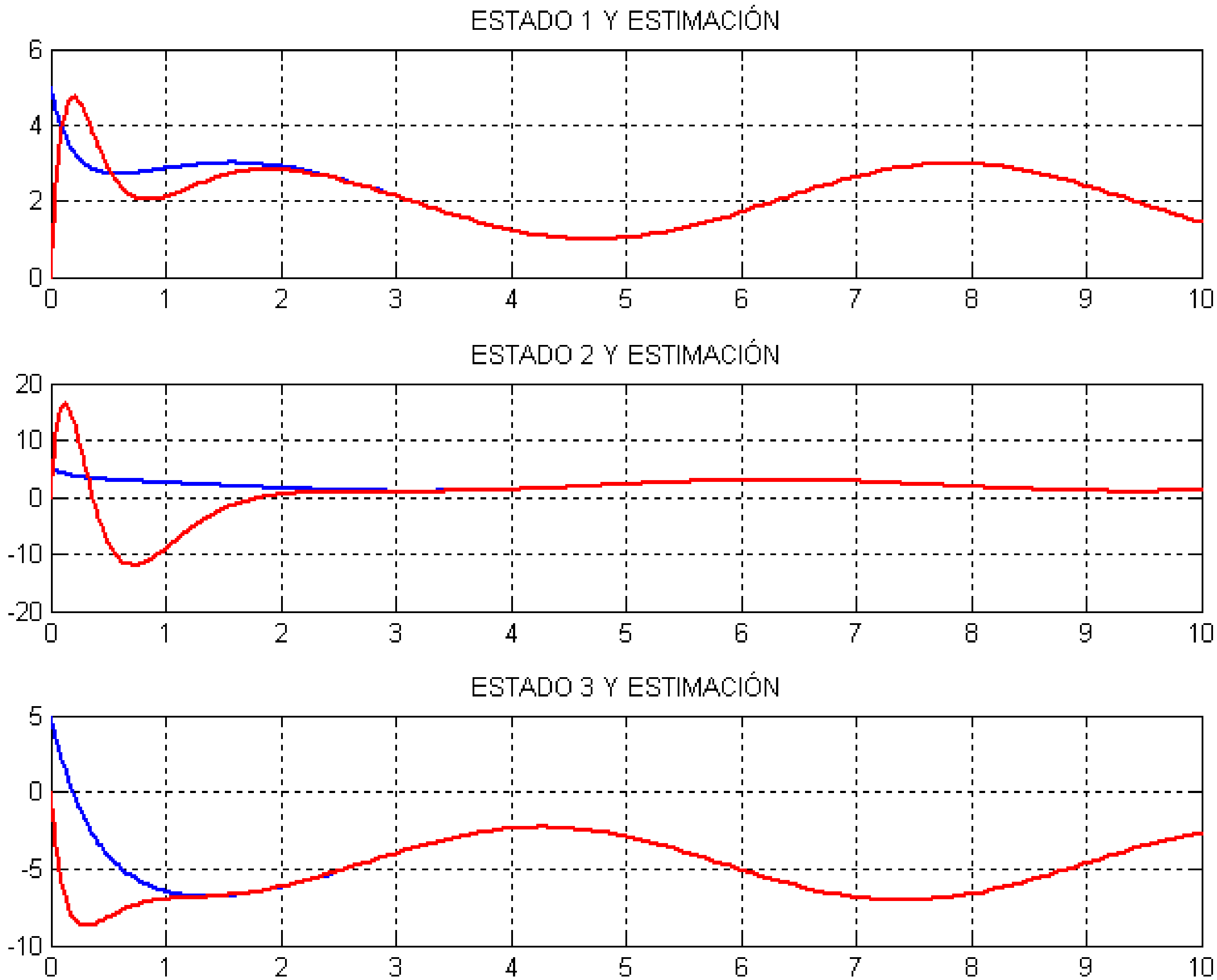
Código .m

```
function MULTIVARIABLE_OBS_CONT_plot( tspan, x0, Pdo, Pdc)
global A B C L Ke C1

A = [-1 1 -2;2 3 -1;1 -1 0];    B = [1 2;3 1;2 -3];    C = [1 0 0;0 1 0];

%DISEÑO DEL OBSERVADOR
C1 = C(1,:);
Mo = [C1;C1*A;C1*A^2];
Ho = (A - Pdo(1)*eye(3))*(A - Pdo(2)*eye(3))*(A - Pdo(3)*eye(3));
L  = -Ho*inv(Mo)*[0;0;1]

%DISEÑO DEL CONTROLADOR
Ke = Pdc;

%RESUELVE LAS ECUACIONES DIFERENCIALES ORDINARIAS (ODE'S)
[t, X] = ode45(@MULTIVARIABLE_OBS_CONT_sys, tspan, [x0 0 0 0]);

%Graficando
figure;
subplot(3,1,1); plot(t, X(:,1), t, X(:,4), '--r');  title('ESTADO 1 Y ESTIMACIÓN');  grid;
subplot(3,1,2); plot(t, X(:,2), t, X(:,5), '--r');  title('ESTADO 2 Y ESTIMACIÓN');  grid;
subplot(3,1,3); plot(t, X(:,3), t, X(:,6), '--r');  title('ESTADO 3 Y ESTIMACIÓN');  grid;

figure;
subplot(3,1,1); plot(t, X(:,1) - X(:,4));  title('Error de Estimación de X1');  grid;
subplot(3,1,2); plot(t, X(:,2) - X(:,5));  title('Error de Estimación de X2');  grid;
subplot(3,1,3); plot(t, X(:,3) - X(:,6));  title('Error de Estimación de X1');  grid;

figure;     Y = X(:,1:3)*C';        ref = [2 + sin(t), 2 + cos(t)];
subplot(2,1,1); plot(t, Y(:,1), t, ref(:,1), '--k');  title('SALIDA 1 Y REFERENCIA 1');  grid;
subplot(2,1,2); plot(t, Y(:,2), t, ref(:,2), '--k');  title('SALIDA 2 Y REFERENCIA 2');  grid;

end
```

```
function dX = MULTIVARIABLE_OBS_CONT_sys( t, X)

global A B C C1 L Ke

x = X(1:3); %ESTADOS DEL SISTEMA
xo = X(4:6); %ESTADOS DEL OBSERVADOR

ref = [2 + sin(t); 2 + cos(t)]; dref = [cos(t); -sin(t)];

e = ref - C*x;

% Ley de control
U = (C*B)^-1*(dref-C*A*x-Ke*e);  %SEG DE REF

%Observador
Y = C1*x;        %Salida original del sistema
Ye = Y - C1*xo; %Error de Observación

%ODE's
dx = A*x  + B*U;
dxo = A*xo + B*U - L*Ye;

dX = [dx ; dxo];
```

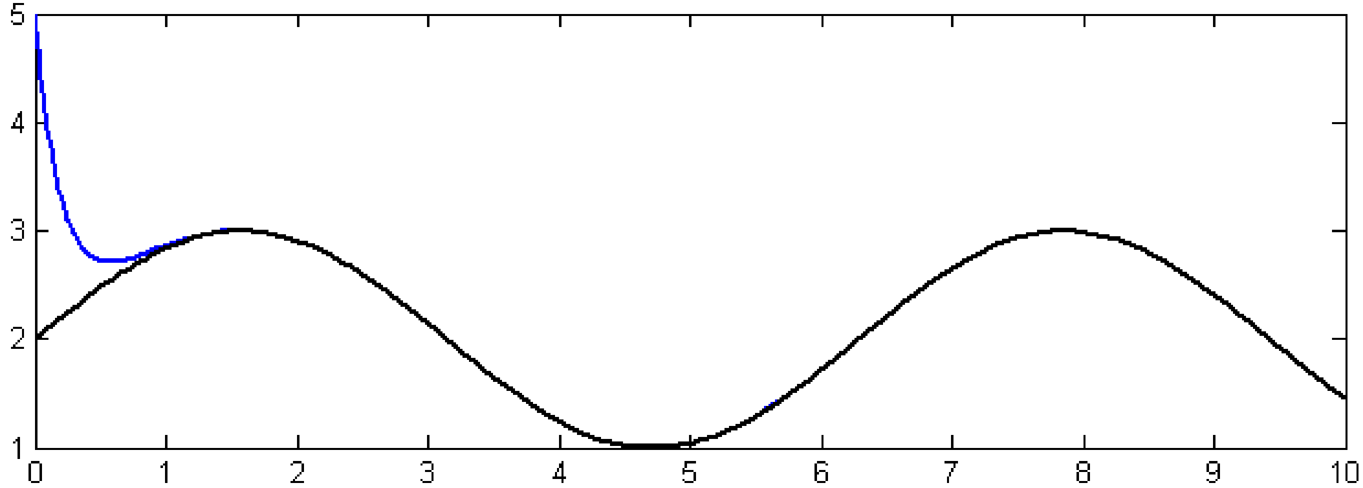
Ojo: En este código se usaron los estados originales del sistema **x** en la ley de control para fines de comparación.
Para que el controlador sea implementable, deberán cambiarse por los estados observados **xo**.

USANDO ESTADOS DEL SISTEMA…

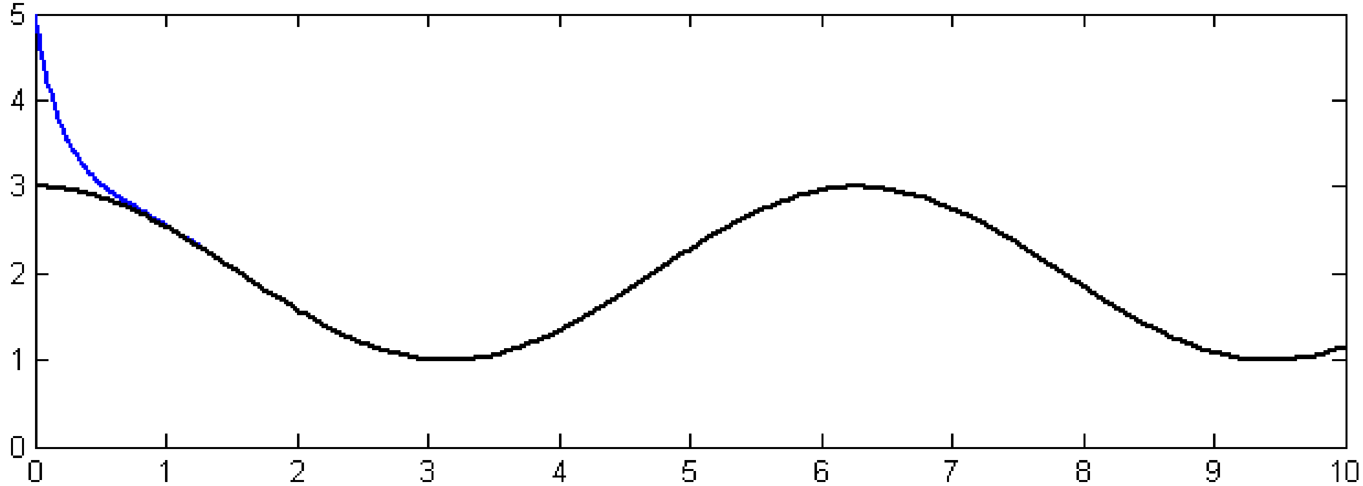


$$x(0)=\begin{bmatrix}5 & 5 & 5\end{bmatrix}^T, \quad P_{dc}=-5, \quad P_{do}=\begin{bmatrix}-4 & -4 & -4\end{bmatrix},$$

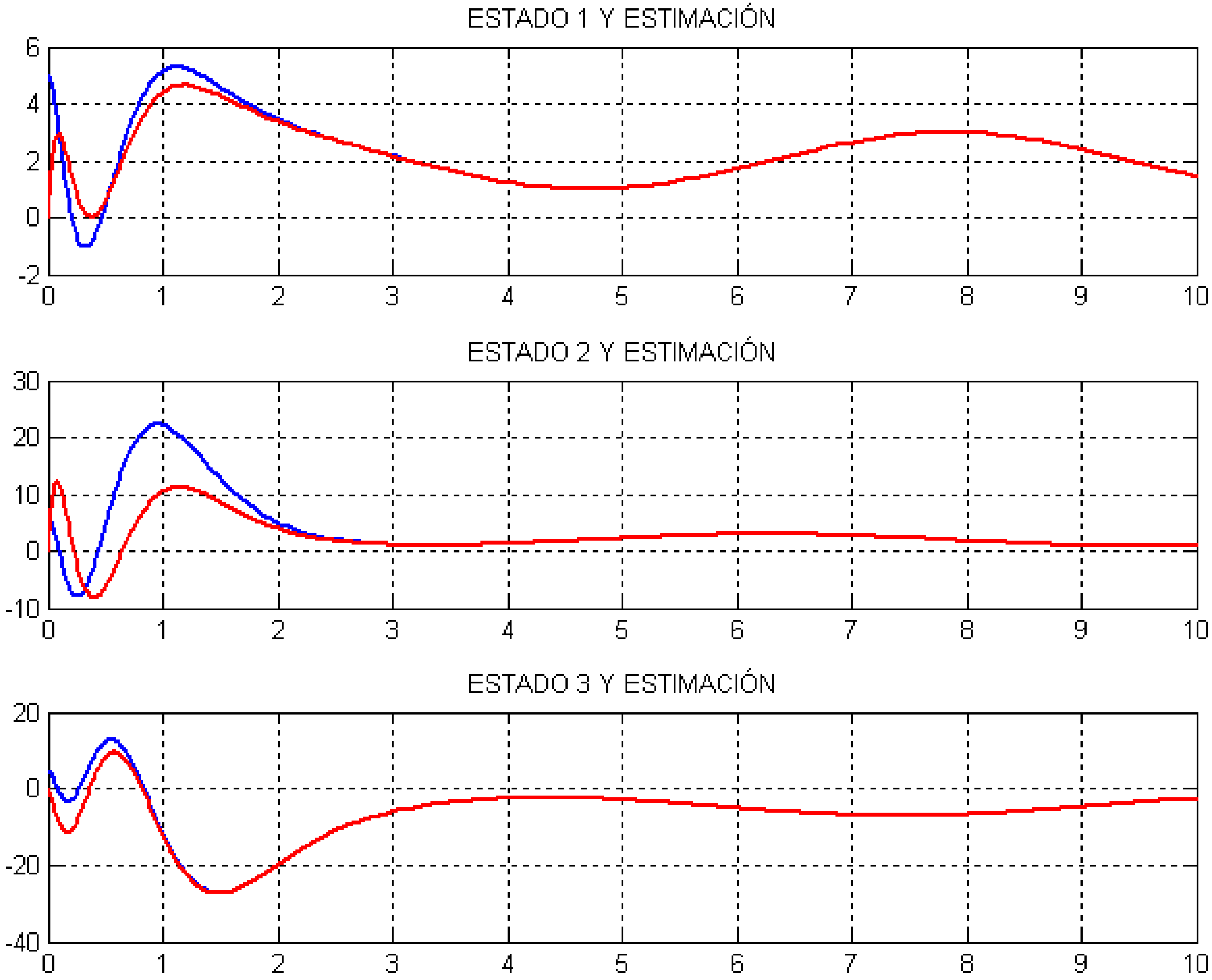
SALIDA 1 Y REFERENCIA 1



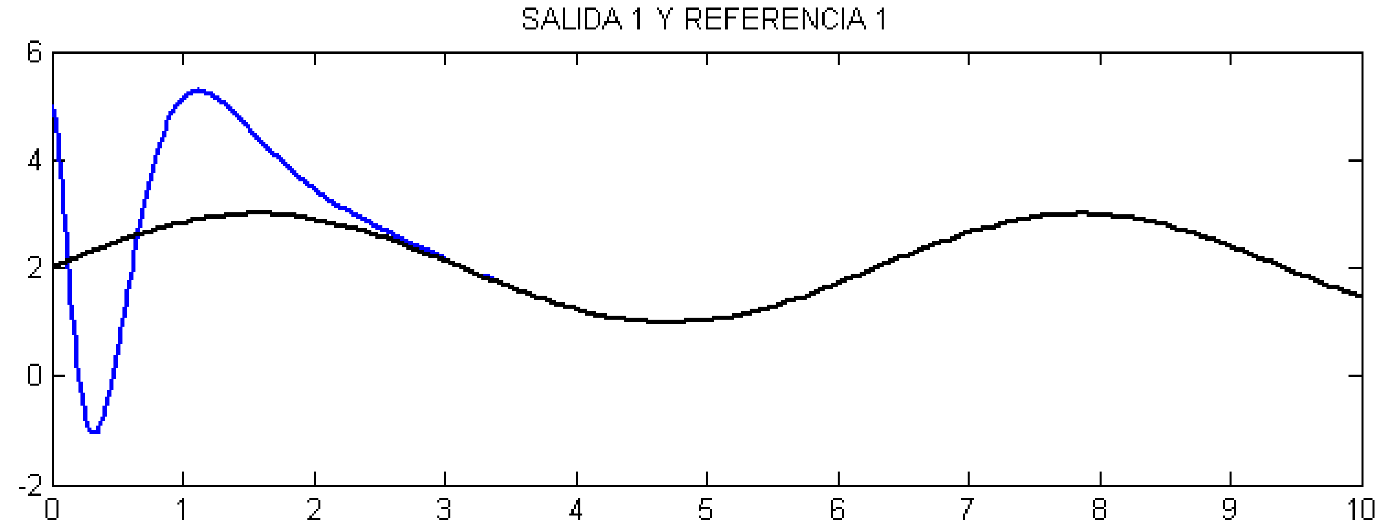
SALIDA 2 Y REFERENCIA 2

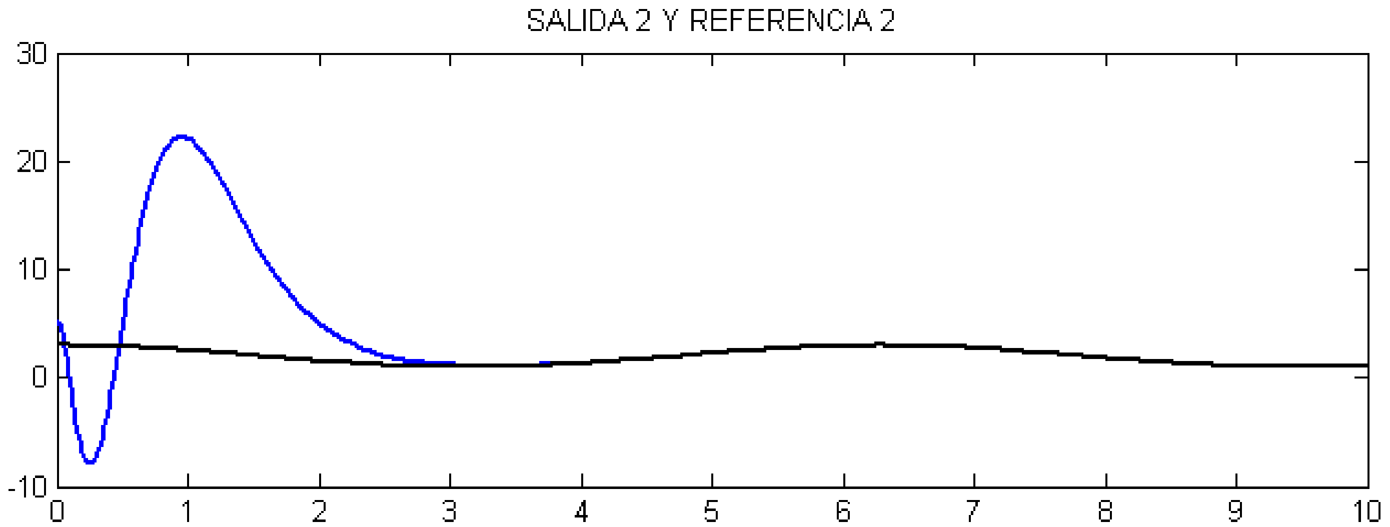


USANDO ESTADOS DEL OBSERVADOR…



$$x(0)=\begin{bmatrix}5 & 5 & 5\end{bmatrix}^T, \quad P_{dc}=-5, \quad P_{do}=\begin{bmatrix}-4 & -4 & -4\end{bmatrix},$$


SALIDA 1 Y REFERENCIA 1
6
4
2
0
-2
0
1
2
3
4
5
6
7
8
9
10


SALIDA 2 Y REFERENCIA 2
30
20
10
0
-10
0
1
2
3
4
5
6
7
8
9
10