

**DEPARTAMENTO DE ELECTRÓNICA Y AUTOMÁTICA**

**FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE SAN JUAN**

Informe de Control Nº 6

OBSERVADORES DE ESTADOS

**Asignatura:** Control II

**Ing. Electrónica**

***Autor:***

*Avila, Juan Agustin – Registro 26076*

**2º Semestre**

**Año 2019**

Contenido

[1 Introducción 2](#_Toc25689165)

[1.1 Planta a utilizar 2](#_Toc25689166)

[2 Observadores de orden completo 2](#_Toc25689167)

[2.1 Punto 1 2](#_Toc25689168)

[2.1.1 Desarrollo 3](#_Toc25689169)

[2.1.2 Simulacion 5](#_Toc25689170)

[2.2 Punto 2 6](#_Toc25689171)

[2.2.1 Desarrollo 6](#_Toc25689172)

[2.2.2 Simulacion 8](#_Toc25689173)

[2.3 Punto 3 10](#_Toc25689174)

[2.3.1 Desarrollo 10](#_Toc25689175)

[2.3.2 Simulación 11](#_Toc25689176)

[3 Observadores de orden reducido 12](#_Toc25689177)

[3.1 Punto 1 12](#_Toc25689178)

[3.2 Punto 2 13](#_Toc25689179)

[3.2.1 Desarrollo 13](#_Toc25689180)

[3.2.2 Simulación 14](#_Toc25689181)

[3.3 Punto 3 15](#_Toc25689182)

[3.3.1 Desarrollo 15](#_Toc25689183)

[3.3.2 Simulación 16](#_Toc25689184)

[3.4 Punto 4 17](#_Toc25689185)

[3.4.1 Desarrollo 17](#_Toc25689186)

[3.4.2 Simulacion 18](#_Toc25689187)

# Introducción

El objetivo del siguiente trabajo es aplicar los conocimientos teóricos sobre observadores de estados en un ejemplo práctico, primero con un observador de orden completo y luego con un observador de orden reducido.

## Planta a utilizar

Para el siguiente trabajo, se trabaja con la planta N° 4 y con w0= 0.5

# Observadores de orden completo

Para este punto, se debe realizar un controlador servo tipo 1 con acción integral y observador de Chen.

## Punto 1

Diseñar el observador de orden completo para la planta que le corresponda y mostrar la evolución de los estados y sus estimaciones para condiciones iniciales de los estados distintas de cero y entrada nula u(t) = 0

### Desarrollo

Se parte obteniendo el MMEE

G=zpk(-7,[-3 -5 roots([1 10 74])' roots([1 20 81])'],160)

Gss=ss(G)

[A,B,C,D]=ssdata(Gss)

n=length(A);

G =

160 (s+7)

------------------------------------------------

(s+3) (s+5) (s+5.641) (s+14.36) (s^2 + 10s + 74)

Gss =

A =

x1 x2 x3 x4 x5 x6

x1 -3 2 0 0 0 0

x2 0 -5 1 0 0 0

x3 0 0 -5.641 0.378 0 0

x4 0 0 0 -5 7 0

x5 0 0 0 -7 -5 0.378

x6 0 0 0 0 0 -14.36

B =

u1

x1 0

x2 0

x3 0

x4 0

x5 0

x6 16

C =

x1 x2 x3 x4 x5 x6

y1 20 10 0 0 0 0

D =

u1

y1 0

Se elige la matriz F que sea Hurwitz (la parte real de los autovalores son negativas) de 6x6 y que no tenga autovalores en común con los de A. En este caso, está conformada por los 6 lambdas deseados.

Se eligen y se declaran los lambdas deseados del observador:

lambdasObs=[-6.3 -8.4 -10.5 -12.6 -14.7 -16.8];

F=diag(lambdasObs)

Se elige un vector S cualquiera de 6x1 tal que (F, S) sea controlable

S=rand(length(A),1)

Co=ctrb(F,S)

R=rank(Co)

S =

0.0046

0.7749

0.8173

0.8687

0.0844

0.3998

Co =

1.0e+05 \*

0.0000 -0.0000 0.0000 -0.0000 0.0001 -0.0005

0.0000 -0.0001 0.0005 -0.0046 0.0386 -0.3241

0.0000 -0.0001 0.0009 -0.0095 0.0993 -1.0431

0.0000 -0.0001 0.0014 -0.0174 0.2190 -2.7588

0.0000 -0.0000 0.0002 -0.0027 0.0394 -0.5796

0.0000 -0.0001 0.0011 -0.0190 0.3185 -5.3502

R = 6

Se comprueba que el rango de la matriz de controlabilidad sea igual al rango de la matriz A (n)

if R==n

disp('El par (F,S) es controlable')

else

disp('El par (F,S) no es controlable')

end

Se calcula la solución única T, no singular, de la ecuación de Sylvester:

T=sylvester(-F,A,S\*C)

T =

0.0281 -0.0076 0.0115 -0.0001 0.0006 0.0000

2.8700 0.5909 -0.2142 0.0045 -0.0094 -0.0006

2.1795 0.6935 -0.1427 0.0037 -0.0048 -0.0005

1.8098 0.6668 -0.0958 0.0026 -0.0024 -0.0005

0.1443 0.0573 -0.0063 0.0002 -0.0001 0.0001

0.5794 0.2406 -0.0216 0.0005 -0.0003 0.0000

Se arman las matrices A,B,C,D del observador

Ao=inv(T)\*F\*T;

Bo=[B inv(T)\*S];

Co=eye(length(A));

Do=zeros(length(A),2);

Gobs=ss(Ao,Bo,Co,Do)

Gobs =

A =

x1 x2 x3 x4 x5 x6

x1 -5.633 0.6837 -8.882e-16 0 2.776e-17 -1.735e-18

x2 -57.33 -33.67 1 -1.11e-16 0 2.776e-17

x3 -399.2 -199.6 -5.641 0.378 -5.329e-15 1.11e-15

x4 1762 881.1 0 -5 7 7.105e-15

x5 7218 3609 9.095e-13 -7 -5 0.378

x6 73.76 36.88 1.137e-13 -7.105e-15 0 -14.36

B =

u1 u2

x1 0 0.1316

x2 0 2.867

x3 0 19.96

x4 0 -88.11

x5 0 -360.9

x6 16 -3.688

C =

x1 x2 x3 x4 x5 x6

y1 1 0 0 0 0 0

y2 0 1 0 0 0 0

y3 0 0 1 0 0 0

y4 0 0 0 1 0 0

y5 0 0 0 0 1 0

y6 0 0 0 0 0 1

D =

u1 u2

y1 0 0

y2 0 0

y3 0 0

y4 0 0

y5 0 0

y6 0 0

Con esto, queda conformado el modelo del observador en espacio de estados.

### Simulacion

Luego, se procede a simular la respuesta de los estados del observador con condiciones iniciales distintas de cero y entrada nula u(t)=0:

t=0:0.01:5;

u=0\*ones(length(t),1); % Entrada nula u(t)=0

x0=[-1 2 -3 1 -0.5 -2]; %condiciones iniciales

[y,t,x]=lsim(Gss,u,t,x0);% Simulamos la planta, se obtiene 'y'

[ye,t,xe]=lsim(Gobs,[u y]',t);%Simulamos los estados del obs

Y se procede a graficar la evolución de los estados y de sus estimadores:

figure('Name','Evolucion de los estados y sus estimaciones');

title('Evolucion de todos los estados y sus estimaciones');

hold on;

% Grafico el estado 1 y sus estimaciones

subplot(2,3,1);

plot(t,x(:,1),'k',t,xe(:,1),'b','LineWidth',1);

title('Estado 1');

grid on;

hold on;

legend('x1(t)','x\_est1(t)');

% Grafico el estado 2 y sus estimaciones

subplot(2,3,2);

plot(t,x(:,2),'k',t,xe(:,2),'b','LineWidth',1);

title('Estado 2');

grid on;

hold on;

legend('x2(t)','x\_est2(t)');

% Grafico el estado 3 y sus estimaciones

subplot(2,3,3);

plot(t,x(:,3),'k',t,xe(:,3),'b','LineWidth',1);

title('Estado 3');

grid on;

hold on;

legend('x3(t)','x\_est3(t)');

% Grafico el estado 4 y sus estimaciones

subplot(2,3,4);

plot(t,x(:,4),'k',t,xe(:,4),'b','LineWidth',1);

title('Estado 4');

grid on;

hold on;

legend('x4(t)','x\_est4(t)');

% Grafico el estado 5 y sus estimaciones

subplot(2,3,5);

plot(t,x(:,5),'k',t,xe(:,5),'b','LineWidth',1);

title('Estado 5');

grid on;

hold on;

legend('x5(t)','x\_est5(t)');

% Grafico el estado 6 y sus estimaciones

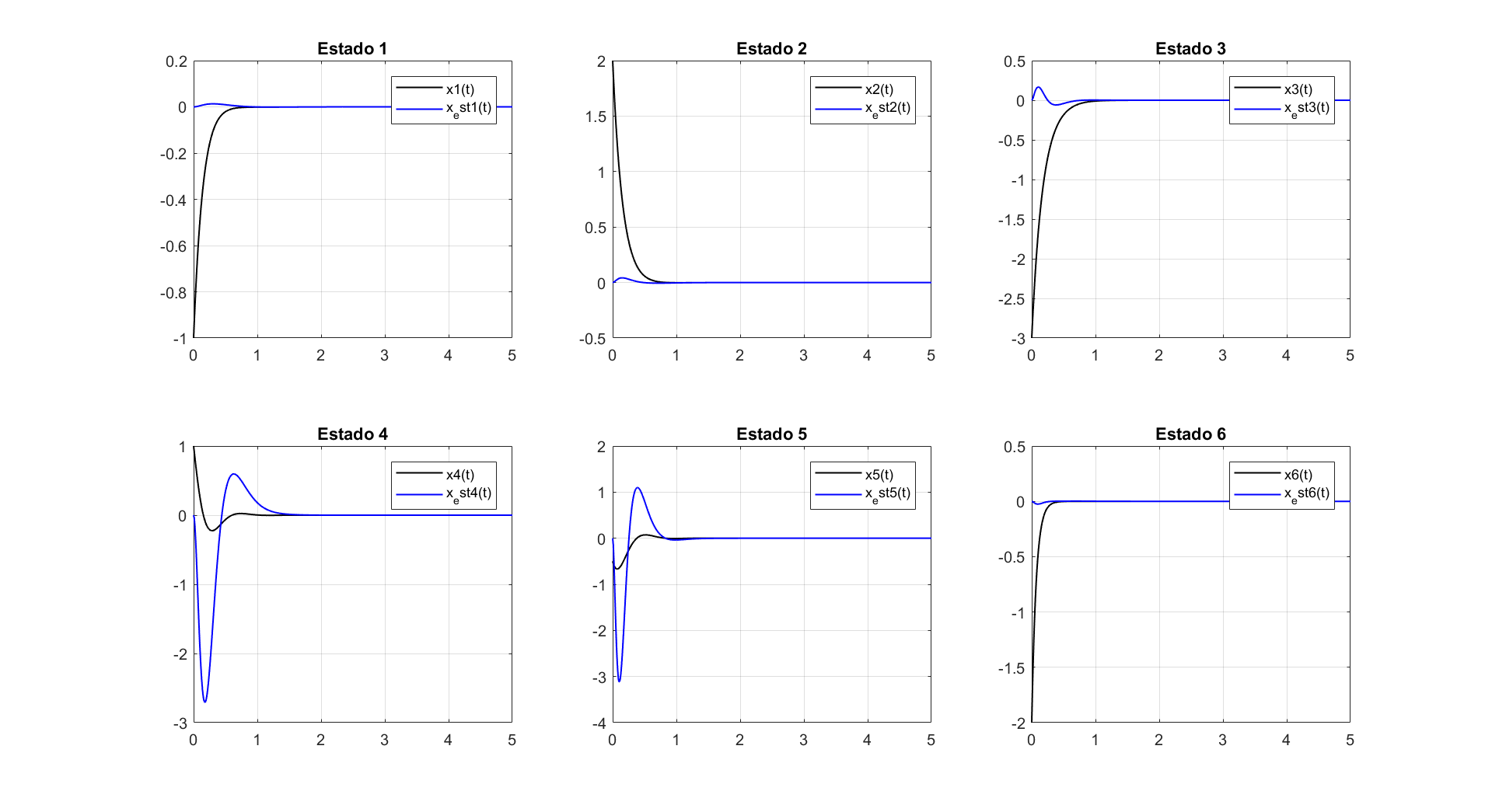
subplot(2,3,6);

plot(t,x(:,6),'k',t,xe(:,6),'b','LineWidth',1);

title('Estado 6');

grid on;

legend('x6(t)','x\_est6(t)');



## Punto 2

Diseñar el controlador mediante realimentación de estado y observador de orden completo que le corresponda. Asignar los autovalores con el criterio visto anteriormente utilizando polinomios ITAE del apendice y el valor w0 asignado a cada planta. Verificar su funcionamiento para condiciones iniciales iguales y distintas de cero y entrada escalon u(t)=1

### Desarrollo

Se comienza cargando la función de transferencia y pasándola a MMEE:

G=zpk(-7,[-3 -5 roots([1 10 74])' roots([1 20 81])'],160)

Gss=ss(G)

[A,B,C,D]=ssdata(Gss)

G =

160 (s+7)

--------------------------------------------------------------

(s+3) (s+5) (s+5.641) (s+14.36) (s^2 + 10s + 74)

Gss =

A =

x1 x2 x3 x4 x5 x6

x1 -3 2 0 0 0 0

x2 0 -5 1 0 0 0

x3 0 0 -5.641 0.378 0 0

x4 0 0 0 -5 7 0

x5 0 0 0 -7 -5 0.378

x6 0 0 0 0 0 -14.36

B =

u1

x1 0

x2 0

x3 0

x4 0

x5 0

x6 16

C =

x1 x2 x3 x4 x5 x6

y1 20 10 0 0 0 0

D =

u1

y1 0

Teniendo las matrices, se forman las matrices ampliadas:

Aa=[A zeros(length(A),1); -C 0];

Ba=[B; 0];

Ca=[C 0];

Da=D;

A partir del polinomio ITAE de orden 6, obtenemos los autovalores deseados en el controlador. Como agregaremos un integrador, asignaremos un polo adicional en -7 para así cancelar el cero que tenemos en -7.

w0=0.5;

P\_ITAE = [1 2.15\*w0 5.63\*(w0)^2 6.93\*(w0)^3 6.79\*(w0)^4 3.74\*(w0)^5 w0^6];

lamb\_deseados=[roots(P\_ITAE)' -7];

Mediante Ackerman, se calcula la matriz K ampliada:

Ka=acker(Aa,Ba,lamb\_deseados);

Con la matriz K ampliada, se forman las matrices del sistema a lazo cerrado con el controlador:

Acl=Aa-Ba\*Ka;

Bcl=[zeros(length(B),1); 1];

Ccl=Ca;

Dcl=Da;

Con estas matrices, formamos el sistema a lazo cerrado:

sys\_cl=minreal(ss(Acl,Bcl,Ccl,Dcl))

sys\_cl =

A =

x1 x2 x3 x4 x5 x6

x1 0.5643 0.892 -1.573 -1.301 1.442 11.65

x2 15.63 -2.253 5.504 1.456 -4.187 -22.56

x3 -15.69 -6.987 6.181 13.07 -12.7 -101.9

x4 12.37 3.156 -8.435 -6.964 4.272 25.91

x5 14.07 3.592 -8.795 -3.621 2.691 53.74

x6 -0.4137 -0.105 0.258 0.175 -0.1316 -1.294

B =

u1

x1 0.05332

x2 0.9088

x3 0.09603

x4 0.3441

x5 0.2039

x6 -0.04602

C =

x1 x2 x3 x4 x5 x6

y1 -19.87 0.848 -0.8044 0.6339 2.923 9.733

D =

u1

y1 0

### Simulacion

Teniendo las matrices del sistema controlado, se lo simula con condiciones iniciales nulas:

t=0:0.001:15;

u=ones(1,length(t));

x0=[-1 -2 3 -3 -0.5 1];

x0=x0\*.001;

[y0,t,x\_salida]=lsim(Gss,u,t);

[y0\_cl,t,xcl\_salida]=lsim(Gss\_cl,u,t);

[y\_itae,t,x\_itae]=lsim(tf(w0^6,P\_ITAE),u,t);

figure(1);

plot(t,y0,'b','LineWidth',2);

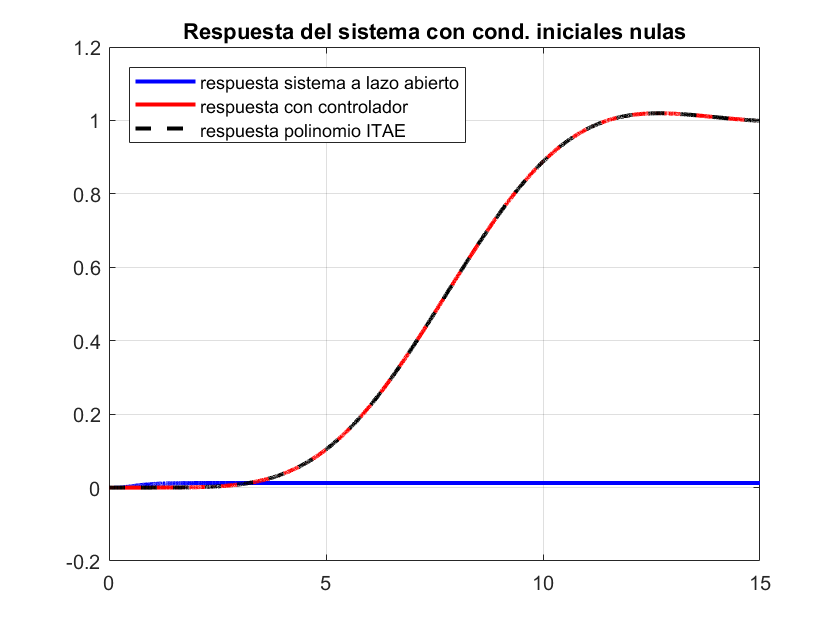
hold on; grid on;

plot(t,y0\_cl,'r','LineWidth',2)

plot(t,y\_itae,'--k','LineWidth',2);

legend('respuesta sistema a lazo abierto','respuesta con controlador','respuesta polinomio ITAE','Location','NorthWest');

title('Respuesta del sistema con cond. iniciales nulas');



Por lo tanto, se observa que para condiciones iniciales nulas, el controlador por realimentación de estados cumple con las especificaciones requeridas y tiene la respuesta del polinomio ITAE de orden 6. Luego, se lo simula para condiciones iniciales distintas de cero:

[y\_x0,t,x\_x0]=lsim(Gss,u,t,x0);

[ycl\_x0,t,xcl\_x0]=lsim(Gss\_cl,u,t,x0);

figure(2);

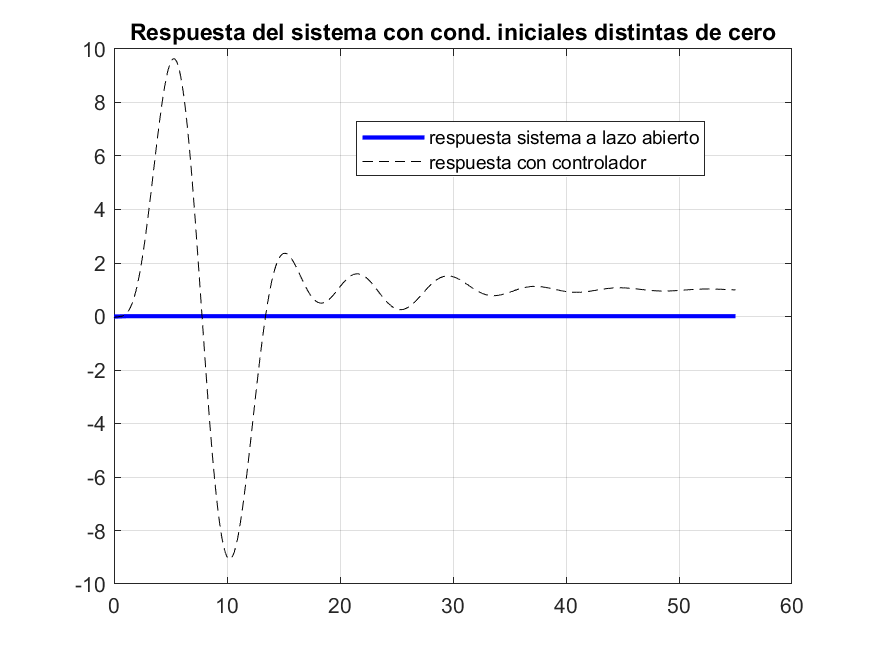
plot(t,y\_x0,'b','LineWidth',2);

hold on; grid on;

plot(t,ycl\_x0,'--k')

legend('respuesta sistema a lazo abierto','respuesta con controlador','Location','NorthWest');

title('Respuesta del sistema con cond. iniciales distintas de cero');



## Punto 3

Combinar los dos subsistemas anteriores y formar un controlador con observador de orden completo. Comparar el funcionamiento con el del controlador sin observador del punto anterior.

### Desarrollo

Partiendo de las bases del apartado 1 y 2, y partiendo de la base de que Gobs es el MMEE del observador de chen, y que Gss\_cl es el modelo de la planta controlado, se forma el controlador con observador de Chen de orden completo:

A partir de la matriz Ka, se obtiene la matriz K y por separado la ganancia Ki del controlador:

K=Ka(1:length(A));

Ki=-Ka(end);

Con estos valores, se forman las matrices que modelan al sistema de lazo cerrado con el controlador y observador de Chen incluido:

Acl\_obs=[A -B\*K\*inv(T) B\*Ki;S\*C F-T\*B\*K\*inv(T) T\*B\*Ki;-C zeros(1,length(C)) 0]

Bcl\_obs=[zeros(2\*length(B),1);1]

Ccl\_obs=[C zeros(1,length(C)) 0]

Dcl\_obs=0

Teniendo las cuatro matrices, se forma el MMEE del sistema total:

Gss\_cl\_obs=minreal(ss(Acl\_obs,Bcl\_obs,Ccl\_obs,Dcl\_obs));

Gss\_cl\_obs =

A =

x1 x2 x3 x4 x5 x6

x1 -7.426 22.14 -5.159 19.12 35.3 63.23

x2 -9.146 9.385 -6.739 17.24 32.23 58.94

x3 2.79 -16.72 1.074 -23.36 -36.02 -69.74

x4 -0.6296 3.935 -1.035 -0.402 9.044 11.38

x5 0.03278 -0.2049 0.083 1.761 4.524 8.019

x6 0.09858 -0.6161 0.251 -0.763 -5.5 -8.23

B =

u1

x1 0.007743

x2 0.07957

x3 -0.0802

x4 -0.4882

x5 -0.702

x6 0.5061

C =

x1 x2 x3 x4 x5 x6

y1 0.6748 -0.17 0.08824 -1.967 4.647 4.579

D =

u1

y1 0

Continuous-time state-space model.

### Simulación

Luego, se procede a simular la respuesta del sistema:

t=0:0.001:15;

u=ones(1,length(t));

[y,t,x]=lsim(gss,u,t);

[y\_cl,t,xcl]=lsim(Gss\_cl,u,t);

[y\_cl\_obs,t,x\_cl\_obs]=lsim(Gss\_cl\_obs,u,t);

Y se procede a graficar las respuestas:

figure(1);

plot(t,y,'b','LineWidth',2);

hold on; grid on;

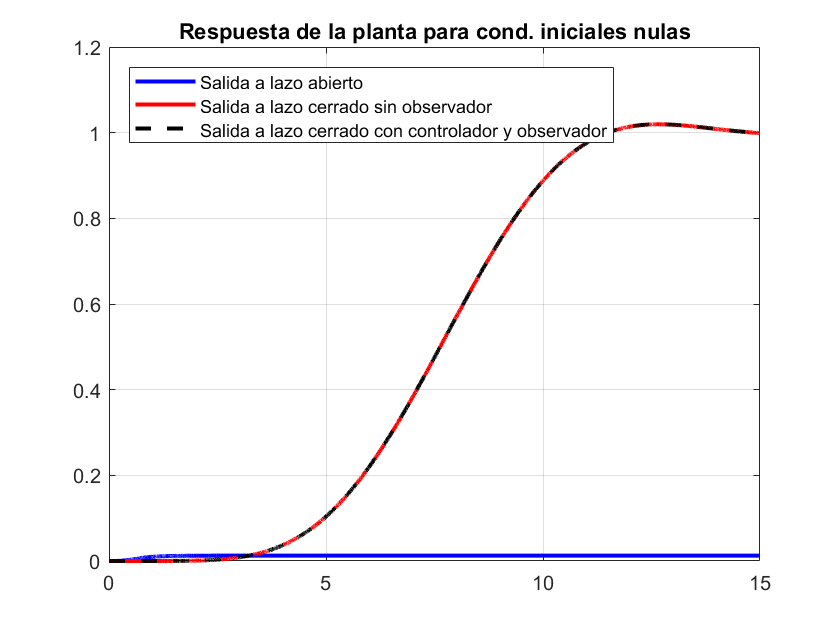
plot(t,y\_cl,'r','LineWidth',2);

plot(t,y\_cl\_obs,'--k','LineWidth',2);

axis([0 15 0 1.2]);

legend('Salida a lazo abierto','Salida a lazo cerrado sin observador','Salida a lazo cerrado con controlador y observador','Location','NorthWest');

title('Respuesta de la planta para cond. iniciales nulas');



Se comprueba que ambos sistemas funcionan de la misma manera para condiciones iniciales iguales a cero. Luego, se procede a simular la respuesta de los distintos sistemas para condiciones iniciales distintas de cero:

x0=[1 -2 3 2 -0.5 -1];%condiciones iniciales

x0=x0\*.0001

t=0:0.001:45;

u=ones(1,length(t));

[y,t,x]=lsim(Gss,u,t,x0);

[y\_cl,t,xcl]=lsim(Gss\_cl,u,t,x0);

[y\_cl\_obs,t,x\_cl\_obs]=lsim(Gss\_cl\_obs,u,t,x0);

figure(2);

plot(t,y,'b','LineWidth',2);

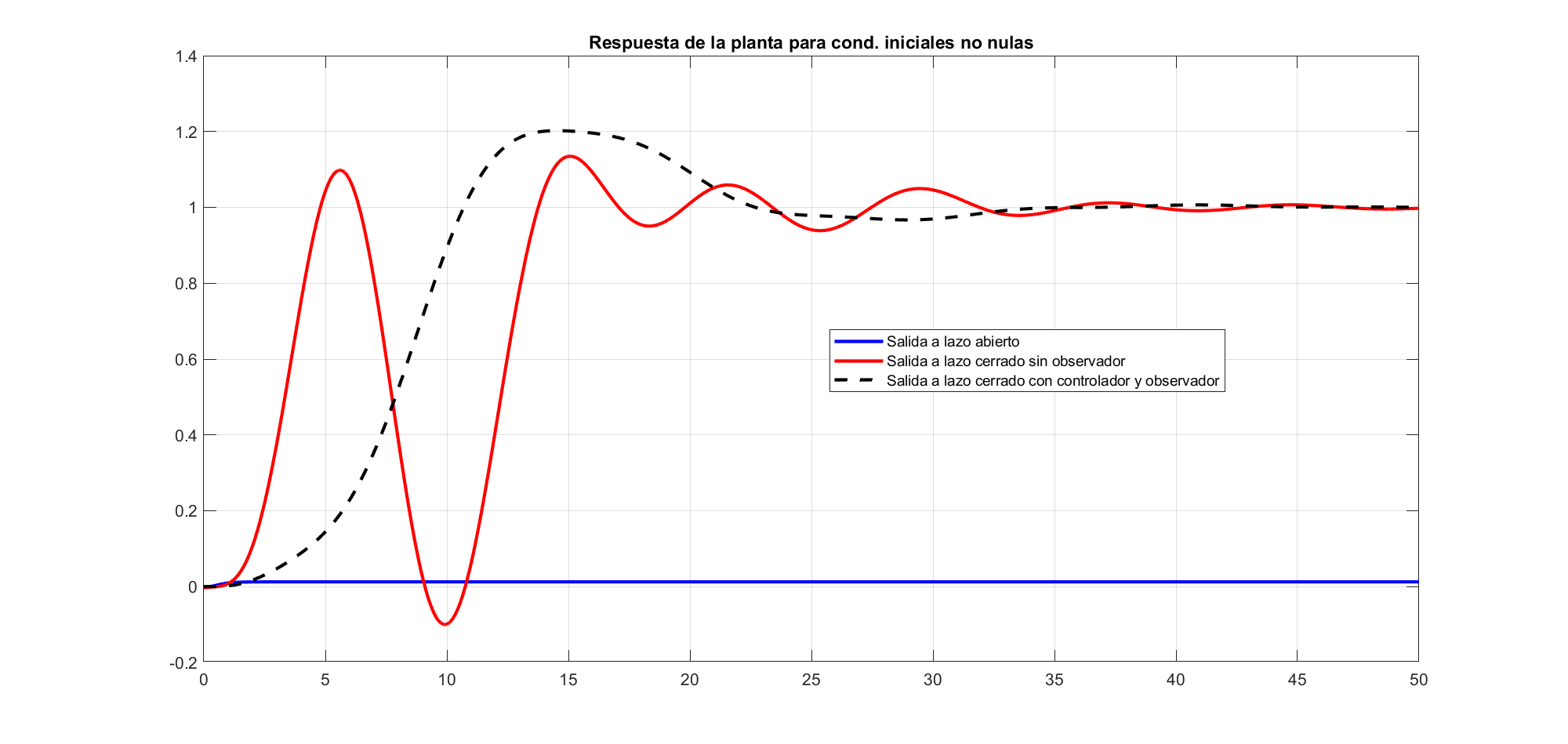
hold on; grid on;

plot(t,y\_cl,'r','LineWidth',2);

plot(t,y\_cl\_obs,'--k','LineWidth',2);

legend('Salida a lazo abierto','Salida a lazo cerrado sin observador','Salida a lazo cerrado con controlador y observador','Location','NorthWest');

title('Respuesta de la planta para cond. iniciales no nulas');



Al comparar el sistema con realimentación de estados con el sistema con controlador y observador, se observa que este ultimo tiene una salida mas cercana a la del sistema con condiciones iniciales nulas.

# Observadores de orden reducido

Para este punto, se debe utilizar un servo tipo 1 con esquema de Ogata para plantas tipo 1 y observador reducido de Luemberger.

## Punto 1

Obtener el modelo de espacio de estado en la forma canonica controlable 1b (FCC1b) para que el primer estado coincida con la salida de la planta, x1(t)=y(t)

G=zpk(-7,[-3 -5 roots([1 10 74])' roots([1 20 81])'],160)

Gss=ss(G)

[A,B,C,D]=ssdata(Gss);

[Acc,Bcc,Ccc,Dcc]= ss2ss(A,B,C,D,obsv(A,C))

Gcc1b=ss(Acc,Bcc,Ccc,Dcc)

Gcc1b =

A =

x1 x2 x3 x4 x5 x6

x1 0 1 0 0 0 0

x2 0 0 1 0 0 0

x3 0 0 0 1 0 0

x4 0 0 0 0 1 0

x5 0 0 0 0 0 1

x6 -8.991e+04 -8.23e+4 -2.964e+04 -5580 -610 -38

B =

u1

x1 0

x2 0

x3 0

x4 0

x5 160

x6 -4960

C =

x1 x2 x3 x4 x5 x6

y1 1 1.478e-14 5.097e-15 8.71e-16 7.774e-17 3.031e-18

D =

u1

y1 0

Continuous-time state-space model.

## Punto 2

Diseñar el observador de orden reducido que le corresponda a cada uno según la tabla(clásico o Chen) y mostrar la evolución de los estados y sus estimaciones para condiciones iniciales de los estados distintas de cero y entrada nula u(t)=0

### Desarrollo

En este caso, se debe utilizar un observador clásico (de Luemberger). Partiendo del desarrollo del punto 1, se particiona el vector de estados en xa y xb:

Aaa=Acc(1,1);

Aab=Acc(1,2:end);

Aba=Acc(2:end,1);

Abb=Acc(2:end,2:end);

Ba=Bcc(1);

Bb=Bcc(2:end);

Se eligen los autovalores que se utilizaran para diseñar el observador. En este caso, debido a que es un observador de orden reducido, asignaremos sólo 5 valores, y luego formamos la matriz L mediante Ackerman:

lambdas\_obs2=[-14 -16 -10 -19 -17];

L=acker(Abb',Aab',lambdas\_obs2)';

Luego, se calculan las matrices Ahat, Bhat, Chat, Dhat y Fhat, las matrices del observador de estados que genera una estimación asintótica de los estados verdaderos de este sistema.

Ahat=Abb-L\*Aab;

Bhat=Ahat\*L+Aba-L\*Aaa;

Chat=zeros(1,length(A)-1);eye(length(A)-1)];

Dhat=[1;L];

Fhat=Bb-L\*Ba;

Gobs\_r=ss(Ahat,Bhat Fhat],eye(length(A)-1),zeros(length(A)-1,2));

### Simulación

Teniendo el modelo del observador, procedemos a simular

t=[0:0.01:5];

u=zeros(length(t),1);

x0=[-1 2 -3 1 -0.5 -2]; %condiciones iniciales

x0=x0\*.1

y,t,x]=lsim(Gcc1b,u,t,x0);

[z,t]=lsim(Gobs\_r,[y u]',t);

z=z';

for i=1:length(t);

xe(i,:)=Chat\*z(:,i)+Dhat\*y(i);

end;

Y luego se procede a graficar los estados con sus respectivas estimaciones:

% Grafico el estado 1 y sus estimaciones

subplot(2,3,1); plot(t,x(:,1),'k',t,xe(:,1),'b','LineWidth',1);

grid on;

legend('x1(t)','x\_e1(t)');

% Grafico el estado 2 y sus estimaciones

subplot(2,3,2); plot(t,x(:,2),'k',t,xe(:,2),'b','LineWidth',1);

grid on;

legend('x2(t)','x\_e2(t)');

% Grafico el estado 3 y sus estimaciones

subplot(2,3,3); plot(t,x(:,3),'k',t,xe(:,3),'b','LineWidth',1);

grid on;

legend('x3(t)','x\_e3(t)');

% Grafico el estado 4 y sus estimaciones

subplot(2,3,4); plot(t,x(:,4),'k',t,xe(:,4),'b','LineWidth',1);

grid on;

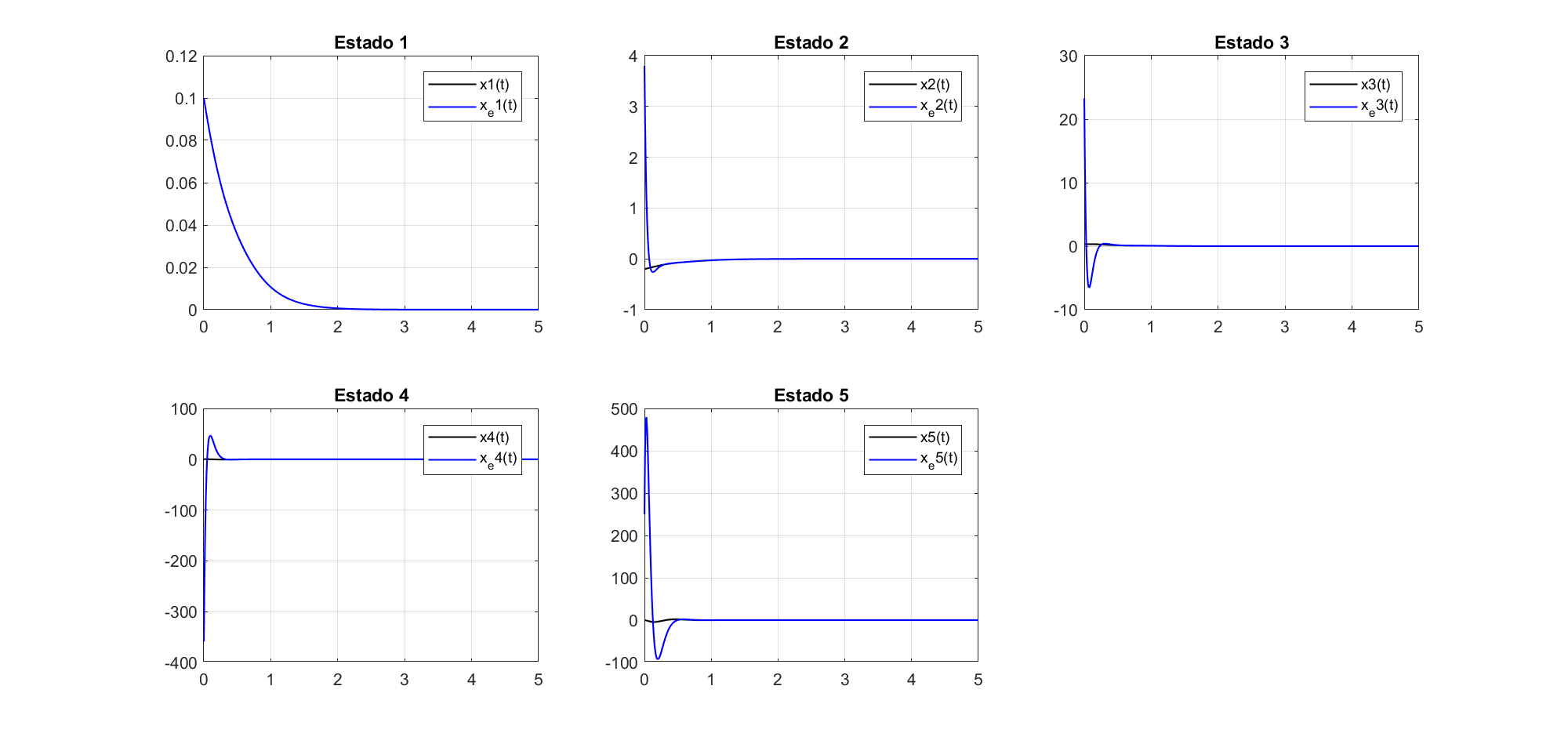
legend('x4(t)','x\_e4(t)');

% Grafico el estado 5 y sus estimaciones

subplot(2,3,5); plot(t,x(:,5),'k',t,xe(:,5),'b','LineWidth',1);

grid on;

legend('x5(t)','x\_e5(t)');



## Punto 3

Diseñar un sistema controlador de estados-observador que les corresponda. Asignar los autovalores como hicimos anteriormente utilizando los polinomios ITAE del apéndice y el valor w0 asignado a cada planta.

### Desarrollo

Se comienza como en el punto anterior, planteando el sistema en la forma FCC1b:

G=zpk(-7,[-3 -5 roots([1 10 74])' roots([1 20 81])'],160)

Gss=ss(G)

[A,B,C,D]=ssdata(Gss);

[Acc,Bcc,Ccc,Dcc]= ss2ss(A,B,C,D,obsv(A,C))

Gcc1b=ss(Acc,Bcc,Ccc,Dcc)

Se eligen los polos deseados para el controlador. Para ello, se utiliza el polinomio ITAE. Se adopta un polinomio ITAE de orden 5, ya que debido a que se diseña un Servo tipo 1 con esquema de Ogata para plantas Tipo 1 y observador reducido de Luemberger, se utiliza un estado menos. Ademas, a los lambdas del polinomio ITAE se agrega uno en -7 para cancelar el cero de la función original:

w0=0.5;

Pcd\_itae5=[1 2.07\*w0 4.5\*w0^2 4.68\*w0^3 3.26\*w0^4 w0^5];

lambdas\_des=[roots(Pcd\_itae5)' -7];

Con esto, se calcula la matriz K para la forma canonica controlable:

Kcc=place(Acc,Bcc,lambdas\_des);

Teniendo la matriz Kcc, se calculan las matrices equivalentes del sistema con el controlador:

Abar=Acc-Bcc\*Kcc;

Bbar=Kcc(1)\*Bcc;

Cbar=Ccc;

Dbar=Dcc;

Y se calcula el MMEE del sistema controlado:

Gss\_cl=minreal(ss(Abar,Bbar,Cbar,Dbar));

Y se procede a simular el sistema:

t=0:0.001:15;

u=ones(1,length(t));

x0=[-1 -2 3 -3 -0.5];

x0=x0\*.01; %luego se usaron cond iniciales mas pequeñas

### Simulación

Se definen solo 5 estados iniciales, para poder simular el sistema reducido. Para simular el estado original, luego se le agrega un estado.

[y0,t,x\_salida]=lsim(Gcc1b,u,t);

[y0\_cl,t,xcl\_salida]=lsim(Gss\_cl,u,t);

[y\_itae,t,x\_itae]=lsim(tf(w0^5,P\_ITAE5),u,t);

Primero, se simula el sistema para condiciones iniciales nulas:

figure(1);

plot(t,y0,'b','LineWidth',2);

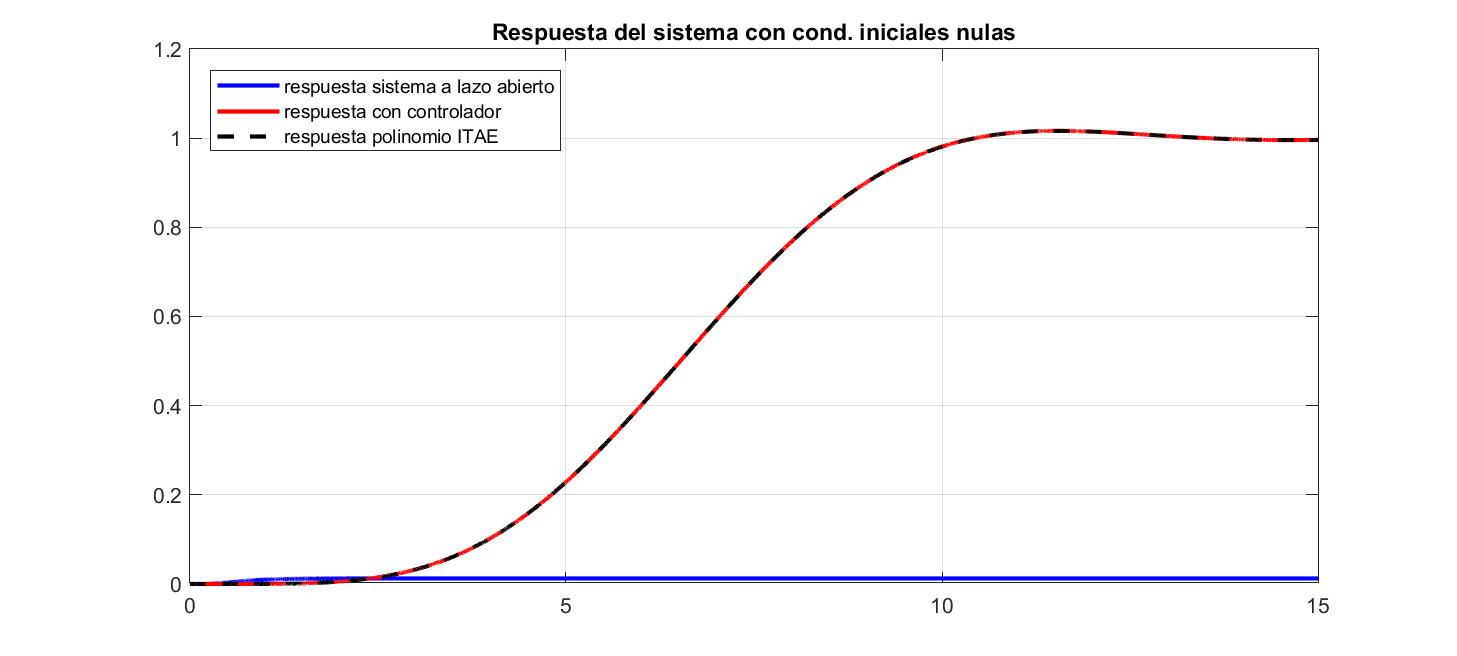
hold on; grid on;

plot(t,y0\_cl,'r','LineWidth',2)

plot(t,y\_itae,'--k','LineWidth',2);

legend('respuesta sistema a lazo abierto','respuesta con controlador','respuesta polinomio ITAE','Location','NorthWest');

title('Respuesta del sistema con cond. iniciales nulas');



Y luego se lo simula para condiciones iniciales no nulas:

[y\_x0,t,x\_x0]=lsim(Gcc1b,u,t,[x0 -3]);

[ycl\_x0,t,xcl\_x0]=lsim(Gss\_cl,u,t,x0);

figure(2);

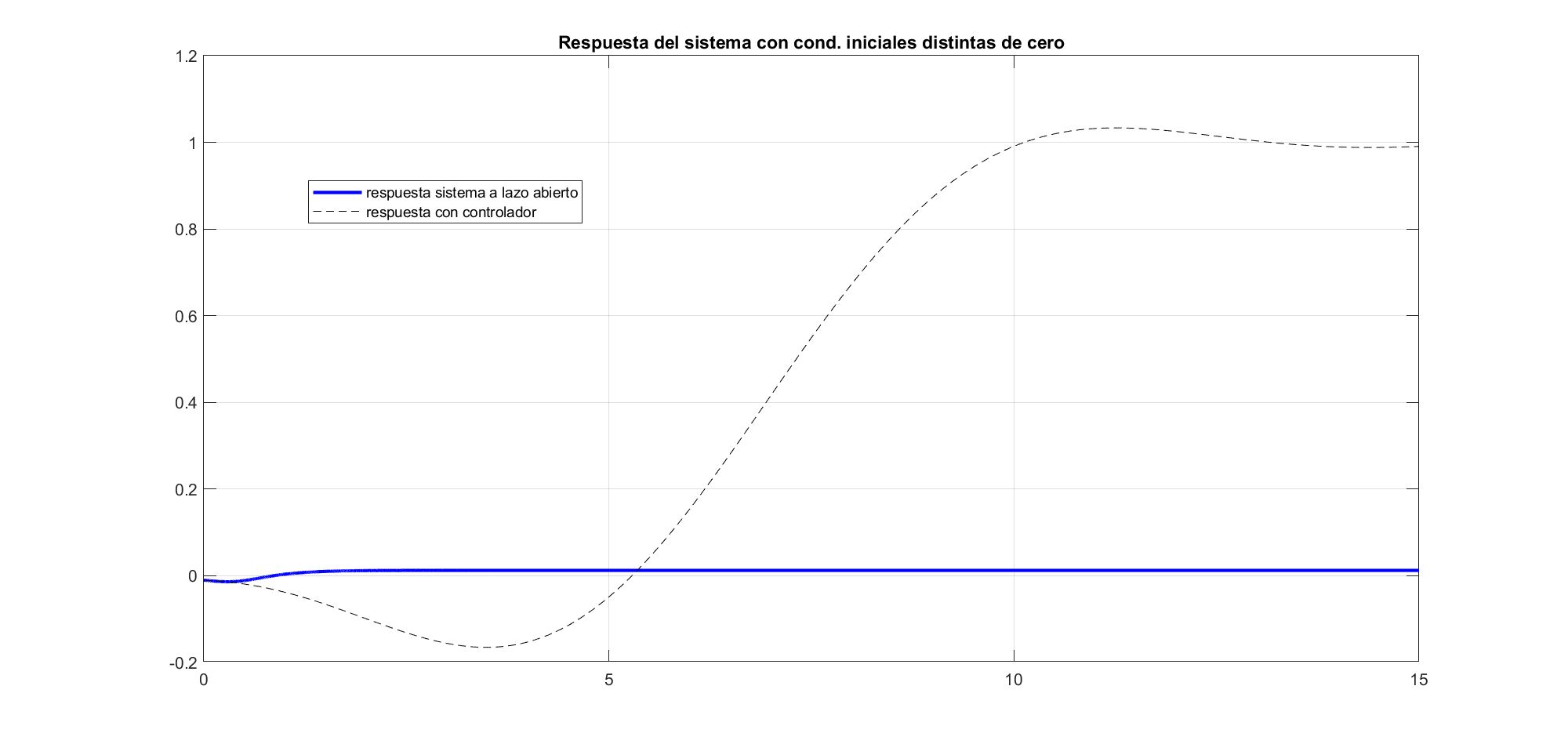
plot(t,y\_x0,'b','LineWidth',2);

hold on; grid on;

plot(t,ycl\_x0,'--k')

legend('respuesta sistema a lazo abierto','respuesta con controlador','Location','NorthWest');

title('Respuesta del sistema con cond. iniciales distintas de cero');



## Punto 4

Combinar los dos subsistemas anteriores y formar un controlador con observador de orden reducido. Comparar el funcionamiento con el del controlador sin observador del punto anterior.

### Desarrollo

Se parte del desarrollo de los puntos anteriores:

G=zpk(-7,[-3 -5 roots([1 10 74])' roots([1 20 81])'],160)

Gss=ss(G)

[A,B,C,D]=ssdata(Gss);

[Acc,Bcc,Ccc,Dcc]= ss2ss(A,B,C,D,obsv(A,C))

Gcc1b=ss(Acc,Bcc,Ccc,Dcc)

Aaa=Acc(1,1);

Aab=Acc(1,2:end);

Aba=Acc(2:end,1);

Abb=Acc(2:end,2:end);

Ba=Bcc(1);

Bb=Bcc(2:end);

lambdas\_obs2=[-14 -16 -10 -19 -17];

L=acker(Abb',Aab',lambdas\_obs2)';

Ahat=Abb-L\*Aab;

Bhat=Ahat\*L+Aba-L\*Aaa;

Chat=[zeros(1,length(A)-1);eye(length(A)-1)];

Dhat=[1;L];

Fhat=Bb-L\*Ba;

Gobs\_r=ss(Ahat,[Bhat Fhat],eye(length(A)-1),zeros(length(A)-1,2))

w0=.5;

Pcd\_itae5=[1 2.07\*w0 4.5\*w0^2 4.68\*w0^3 3.26\*w0^4 w0^5];

lambdas\_des=[roots(Pcd\_itae5)' -7];

Kcc=place(Acc,Bcc,lambdas\_des);

Abar=Acc-Bcc\*Kcc;

Bbar=Kcc(1)\*Bcc;

Cbar=Ccc;

Dbar=Dcc;

G\_cl=minreal(ss(Abar,Bbar,Cbar,Dbar))

Acl2=[Acc-Bcc\*Kcc\*Dhat\*Ccc -Bcc\*Kcc\*Chat; Bhat\*Ccc-Fhat\*Kcc\*Dhat\*Ccc Ahat-Fhat\*Kcc\*Chat];

Bcl2=[Bcc\*Kcc(1); Fhat\*Kcc(1)];

Ccl2=[Ccc zeros(1,n-1)];

Dcl2=0;

G\_cl\_obs=minreal(ss(Acl2,Bcl2,Ccl2,Dcl2))

x0\_ss\_cl=[x0, zeros(n-1,1)'];

### Simulacion

Luego, se procede a simular los sistemas con condiciones iniciales nulas:

t=0:0.001:15;

u=ones(1,length(t));

x0=[1 -2 3 -3 -0.5];%condiciones iniciales

x0=x0\*.1;

[y,t,x]=lsim(Gcc1b,u,t);

[y\_cl,t,xcl]=lsim(G\_cl,u,t);

[y\_cl\_obs,t,x\_cl\_obs]=lsim(G\_cl\_obs,u,t);

figure(1);

plot(t,y,'b','LineWidth',2);

hold on; grid on;

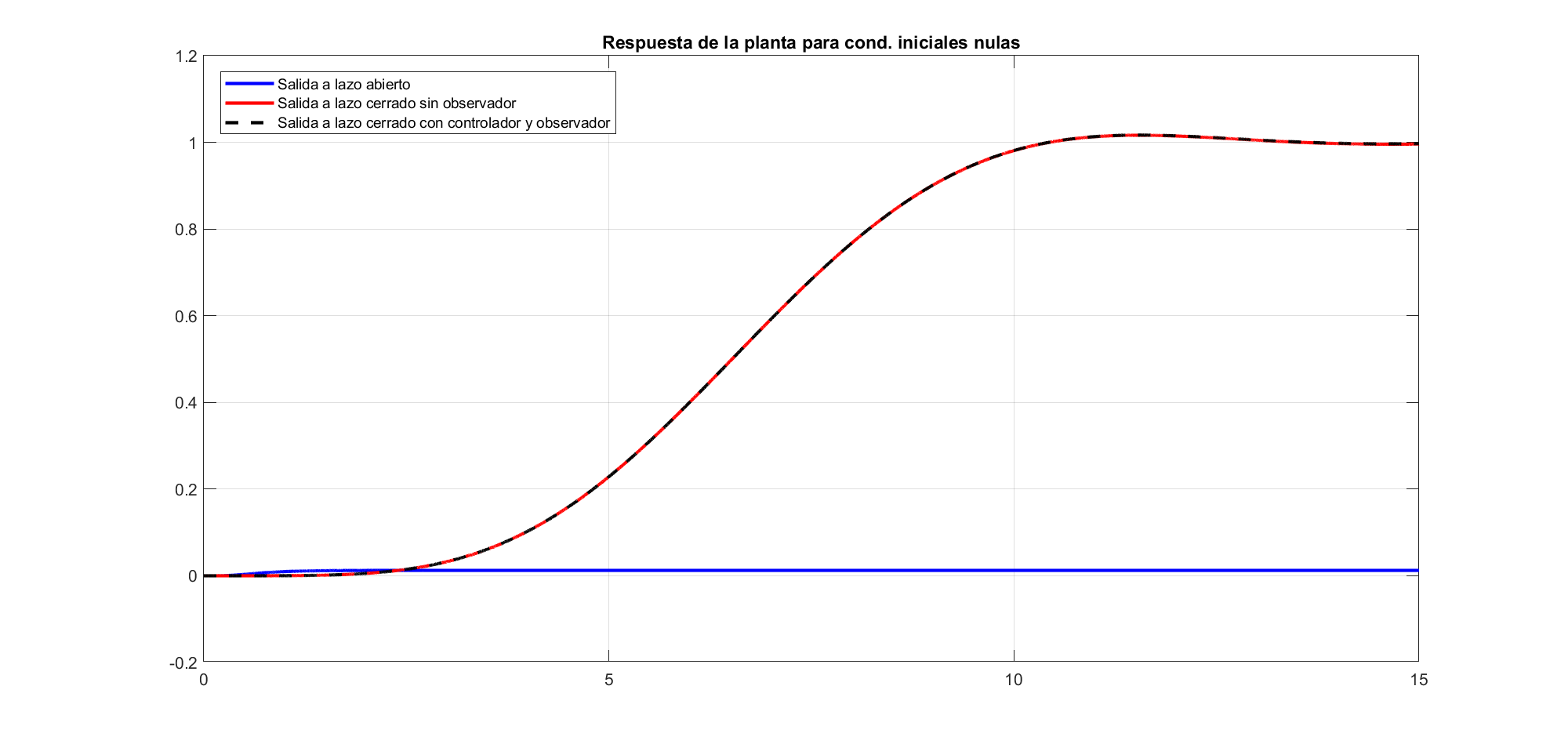
plot(t,y\_cl,'r','LineWidth',2);

plot(t,y\_cl\_obs,'--k','LineWidth',2);

%axis([0 15 0 1.2]);

legend('Salida a lazo abierto','Salida a lazo cerrado sin observador','Salida a lazo cerrado con controlador y observador','Location','NorthWest');

title('Respuesta de la planta para cond. iniciales nulas');



Luego, se procede a simular el sistema para condiciones iniciales distintas de cero:

t=0:0.001:35;

u=ones(1,length(t));

[y,t,x]=lsim(Gcc1b,u,t,[x0 -3]);

[y\_cl,t,xcl]=lsim(G\_cl,u,t,x0);

[y\_cl\_obs,t,x\_cl\_obs]=lsim(G\_cl\_obs,u,t,x0);

figure(2);

plot(t,y,'b','LineWidth',2);

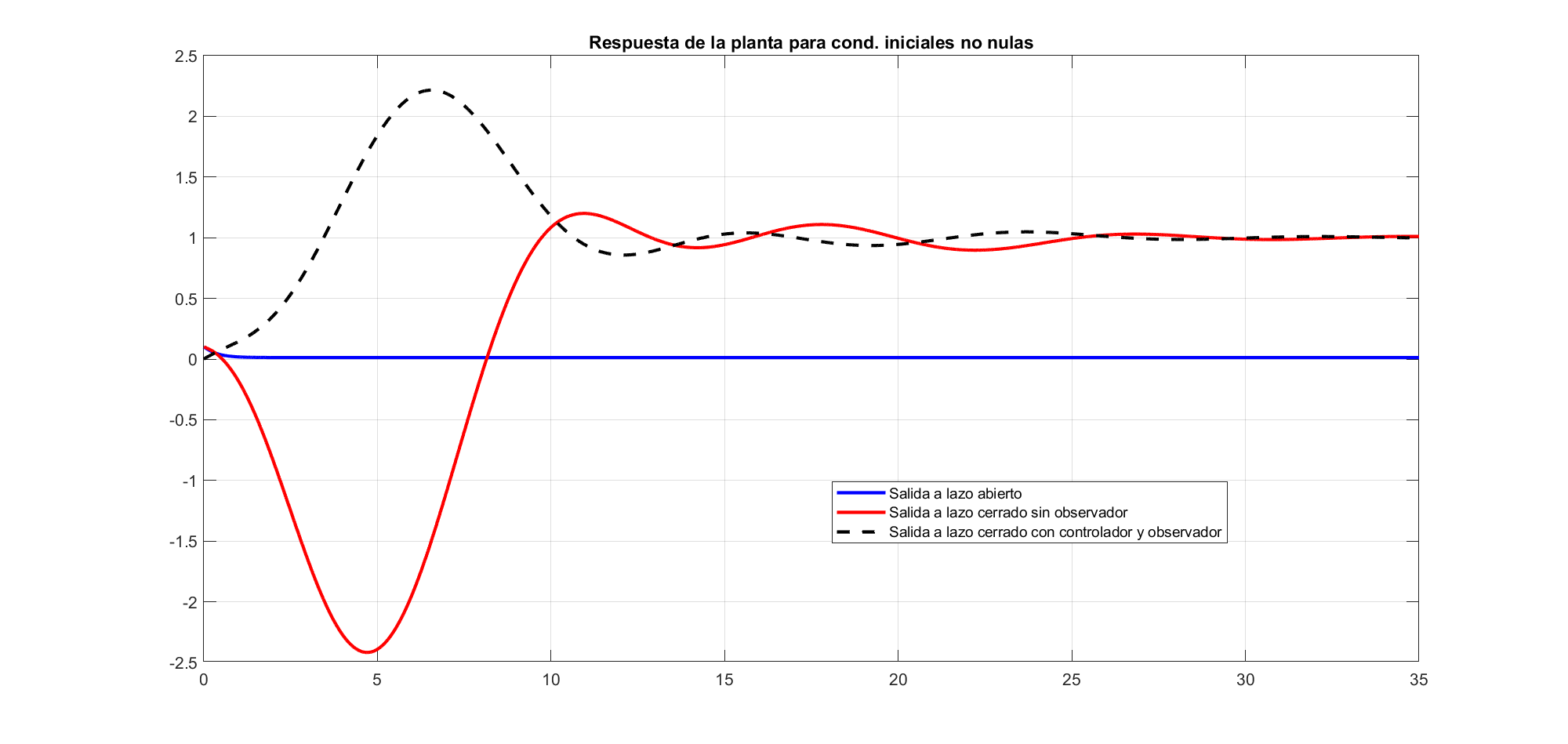
hold on; grid on;

plot(t,y\_cl,'r','LineWidth',2);

plot(t,y\_cl\_obs,'--k','LineWidth',2);

legend('Salida a lazo abierto','Salida a lazo cerrado sin observador','Salida a lazo cerrado con controlador y observador','Location','NorthWest');

title('Respuesta de la planta para cond. iniciales no nulas');



Se observa que el sistema con controlador y observador se comporta mejor que el sistema solo con controlador.