

UNIVERSIDAD NACIONAL DE CÓRDOBA

FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES



SISTEMA DE CONTROL II

ACTIVIDAD PRÁCTICA N 2:
Diseño de controladores en variables
de estado en tiempo continuo

DOCENTE: Pucheta Julián

ALUMNO: Tito Ricardo Clemente

MATRICULA: 35308739

Caso de estudio 1. Sistema de tres variables de estado

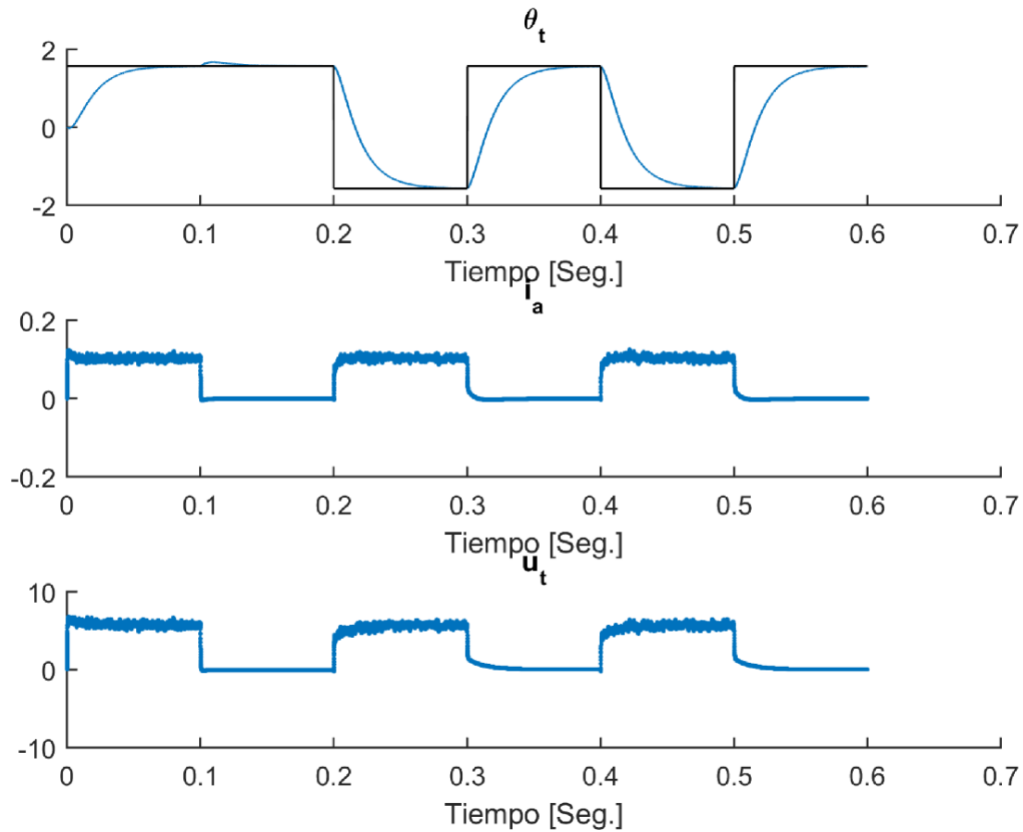


Figura 1. Evolución del ángulo cuando el controlador en variables de estado tiene perturbaciones en su operación.

Dadas las ecuaciones del motor de corriente continua con torque de carga T_L no nulo, con los parámetros $L_{AA} = 5 \cdot 10^{-3}$; $J = 0,004$; $R_A = 0,2$; $B = 0,005$; $K_i = 6,5 \cdot 10^{-5}$; $K_m = 0.055$:

$$\frac{di_a}{dt} = -\frac{R_A}{L_{AA}} i_a - \frac{K_m}{L_{AA}} \omega_r + \frac{1}{L_{AA}} v_a \quad (1)$$

$$\frac{d\omega_r}{dt} = \frac{K_i}{J} i_a - \frac{B_m}{J} \omega_r - \frac{1}{J} T_L \quad (2)$$

$$\frac{d\theta_t}{dt} = \omega_r. \quad (3)$$

Item 1: Implementar un sistema en variables de estado que controle el ángulo del motor, para consignas de $\pi/2$ y $-\pi/2$ cambiando cada 2 segundos y que el T_L de

$1, 15 \cdot 10^{-3}$ aparece sólo para $\pi/2$, para $-\pi/2$ es nulo. Hallar el valor de integración Euler adecuado. El objetivo es mejorar la dinámica del controlador que muestra la Fig 1.

Dado el sistema lineal determinista en tiempo continuo:

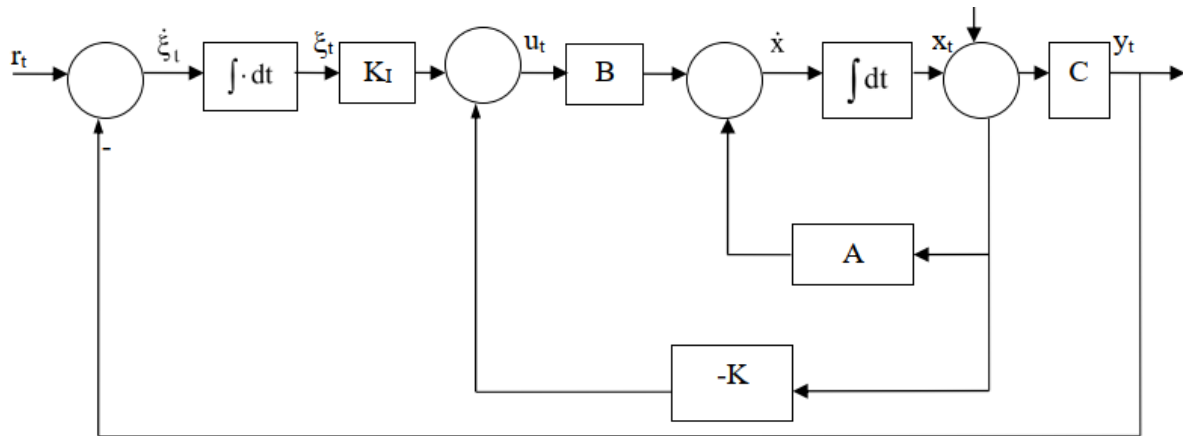
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

la regulación del sistema se logra mediante un controlador lineal, del tipo:

$$\mathbf{u}_t = -\mathbf{K} \cdot \mathbf{x}(t)$$

donde K es la matriz del controlador del siguiente diagrama de bloques:



Donde se tendrá un término extra, un integrador, con el fin de mejorar el rechazo a perturbaciones o incertidumbres en los parámetros empleados para el cálculo del controlador.

Para ello debemos de partir de la definición de las variables de estado a partir de las EDOs del sistema:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{R_A}{L_{AA}} & -\frac{K_m}{L_{AA}} & 0 \\ \frac{K_i}{J} & -\frac{B_m}{L_{AA}} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ \dot{\theta} \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{1}{L_{AA}} \\ 0 \\ 0 \end{bmatrix}$$

Las Matrices resultantes de desarrollar las variables de estado son las siguientes y la función de transferencia que relaciona entrada con salida:

A =

```
-40.0000  -11.0000    0
   0.0162   -1.2500    0
         0    1.0000    0
```

B =

```
200
   0
   0
```

C =

```
0    0    1
```

D =

```
theta      6.5e-05
----- = -----
Va      2e-05 s^3 + 0.000825 s^2 + 0.001004 s
```

Los autovalores de la Matriz A (Matriz de Estados) son los valores correspondientes a las raíces de la ecuación característica:

```
% Calculos los autovalores (son tres)
autovalores=eig(A);
sigma1=autovalores(1); % sigma1=0
sigma2=autovalores(2); % sigma2=-1.2546
sigma3=autovalores(3); % sigma3=-39.9954
```

Se puede observar que $\lambda_2 = -39,9954$ es el de dinámica más rápido y $\lambda_2 = -1,2546$ por lo cual es que usamos para el cálculo del paso de integración de Euler:

```
% Tomo el autovalor con la dinámica más rápida y el más lento

t_I=log(0.95)/sigma3; % tr=0.00128
t_I=6e-04;           % defino un valor dos veces más chico
t_S=log(0.05)/sigma2; % t_s=2.3878
t_S=4;               % defino un valor dos veces más grande
paso=t_S/t_I;        % 8000
```

Para poder seguir con el cálculo de K, necesité de ver que el sistema cumpliera con la condición de controlabilidad.

La Condición de Controlabilidad se cumple si es posible generar una Acción de control u para transferir al sistema desde cualquier condición inicial a cualquier condición final en un intervalo de tiempo finito. Si cada estado es controlable, entonces se dice que el sistema es completamente controlable. Para esto necesite de definir la matriz de control M :

$$[B | AB | \dots A^{n-1}B] = M$$

Solo requiere construir la matriz con los valores de La Matriz de Estados y La Matriz de Entrada y que esta sea del mismo que el rango de la matriz A , ello se calcula con el siguiente código:

```
% Prueba de Condición de Controlabilidad:

M=[B, (A*B), (A^2)*B];
rangoM= rank(M);           % el rango es 3 ;Cumple con la Condición!
```

Donde los resultados muestran que la condición se cumple, por lo tanto se pasa al siguiente paso.

Para el diseño de la matriz K se usó el método de Control Óptimo Cuadrático. La respuesta óptima se obtiene de minimizar el índice de desempeño:

$$J = \int_0^{\infty} L(x, u) dt = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

Donde se debe de determinar de Q y R , los cuales determinan la importancia del error y el gasto de energía, al mismo tiempo, son matrices simétricas definidas positivas. Se usó la función " $\text{lqr}()$ " que nos ofrece Matlab, con lo cual se puede obtener K , la matriz de Control y E , la matriz de los polos deseados correspondientes a la matriz de Control.

Para ello es necesario una construcción del sistema ampliado:

```
% Construcción del sistema ampliado
Aa=[A zeros(3,1);-C 0];
Ba=[B;0];
Ma=[Ba Aa*Ba Aa^2*Ba Aa^3*Ba];%Matriz Controlabilidad
rango_Ma=rank(Ma);% el rango es 4 ;Cumple con la Condición!
```

Pero comprobando la controlabilidad de la matriz M ampliada. De la misma manera la matriz W y la matriz T :

```

%-Diseno mediante LQR -----
auto_val=eig(Ma);
% Coeficientes del polinomio característico de A (Lazo Abierto) ai
ai_A=conv(conv(conv([1 -auto_val(1)], [1 -auto_val(2)]), [1 -auto_val(3)]),
[1 -auto_val(4)]);

% Matriz W (3x3)
Wa=[ai_A(4) ai_A(3) ai_A(2) 1 ;
     ai_A(3) ai_A(2) 1 0 ;
     ai_A(1) 1 0 0 ;
     1 0 0 0];

% Defino la Matriz de transformación para obtener una forma canónica
% controlable.
Ta=Ma*Wa;

```

El cálculo del Controlador nos da los siguientes resultados:

```

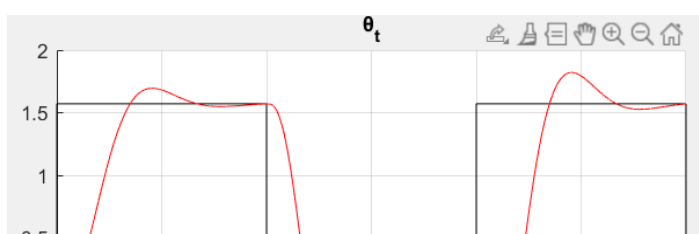
%CALCULO DEL CONTROLADOR K
%para el calculo del mismo se utiliza el metodo LQR para lo cual definimos
Q=diag([.001 1 1 1e5]);
R=0.00001; %
[Ka, Pa, Ea]=lqr(Aa, Ba, Q, R); % Controlador ampliado
K_i= -Ka(4); % Ka=[K -Ki ]
K=Ka(1:3);

```

$$K_a = [9.8509, 6.0307 \times 10^3, 3.689 \times 10^4, -100000]$$

De donde se tenía los valores correspondientes al controlador ampliado, de donde se reconoce a la matriz del controlador, correspondiente a los tres primeros valores y el cuarto al Ki

Dando los siguientes gráficas:





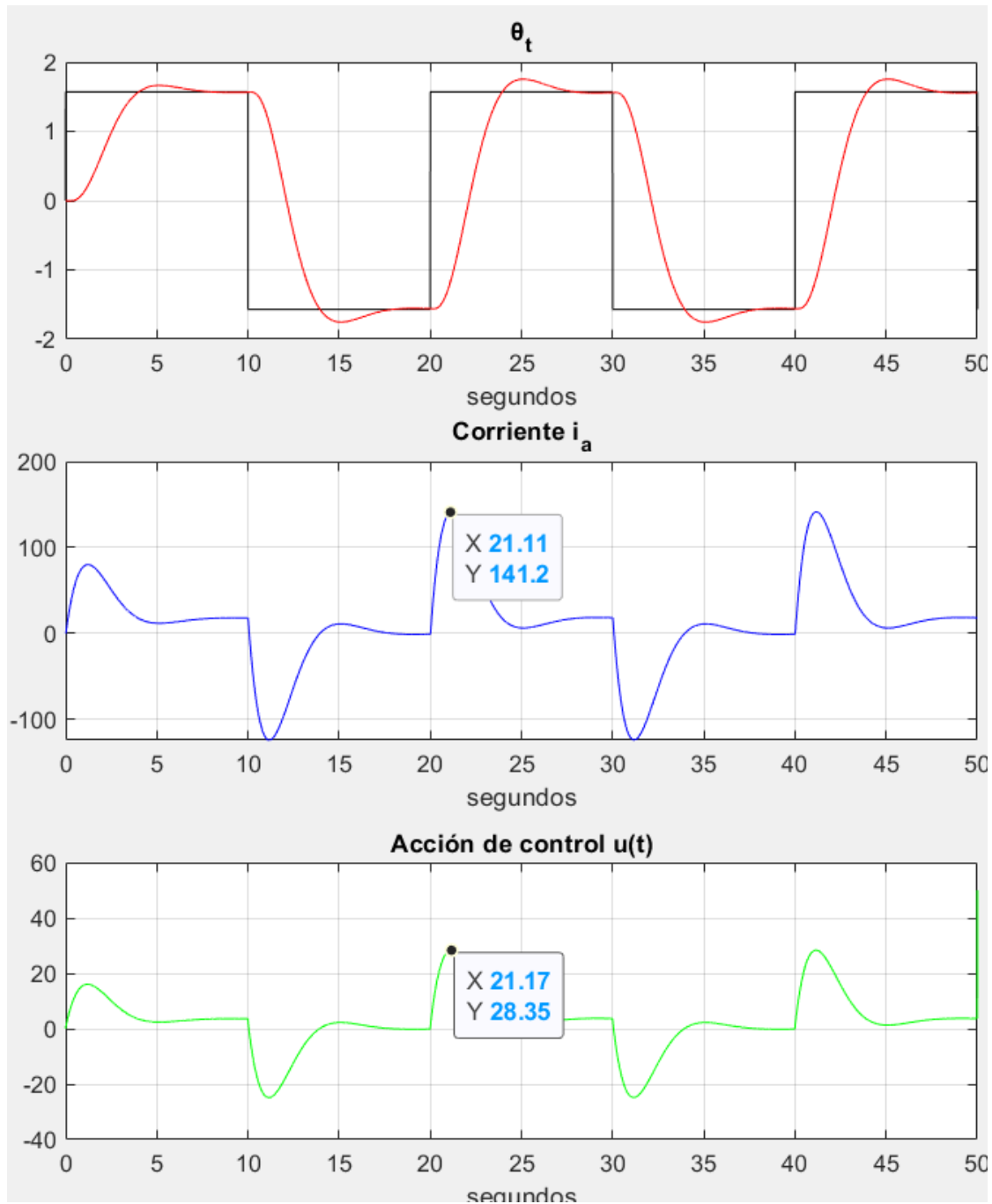
De todo esto podemos concluir que el periodo dado no es suficiente para que el motor alcance el ángulo deseado, sin que implique una gran corriente de armadura para alcanzarla, por lo cual es necesario modificar los datos entregados del principio. La matriz Q y R modificada para este momento arrojan los siguientes valores:

```
Q=diag([.001 1 1 1e5]);
R=0.00001; %
```

Modificando el Periodo del sistema en 20 segundos, se logró obtener valores de corrientes más bajos:

```
% Referencia
ref=(pi/2)*square(2*pi*t/20);
% Torque
TL=(T_L/2)+(T_L/2)*square(2*pi*t/20);
```

Las gráficas resultantes, mostraron los siguientes valores:



Para estos valores la Matriz Q y R, se definen con los siguientes valores:

```
%CALCULO DEL CONTROLADOR K
%para el calculo del mismo se utiliza el metodo LQR para lo cual definimos
Q=diag([1 1 1/1e9 1e4]);
R=9; %
```

Item 2: Considerar que no puede medirse la corriente y sólo pueda medirse el ángulo, por lo que debe implementarse un observador. Obtener la simulación en las mismas condiciones que en el punto anterior, y superponer las gráficas para comparar.

Para poder implementar el observador, se debió agregar en el script anterior el siguiente código:

```
% Calculo del Observador-----
A_o=A';
B_o=C';
C_o=B';
%CALCULO DEL CONTROLADOR Ko
%para el calculo del mismo se utiliza el metodo LQR para lo cual definimos
Qo=diag([1 1 1]);
Ro=0.001; %
[K_o,Po,polos_deseados_o]=lqr(A_o,B_o,Qo,Ro); % Controlador ampliado
mui_o=polos_deseados_o;
Ko=K_o';
disp('Observador en:')
eig(A-Ko*C)
```

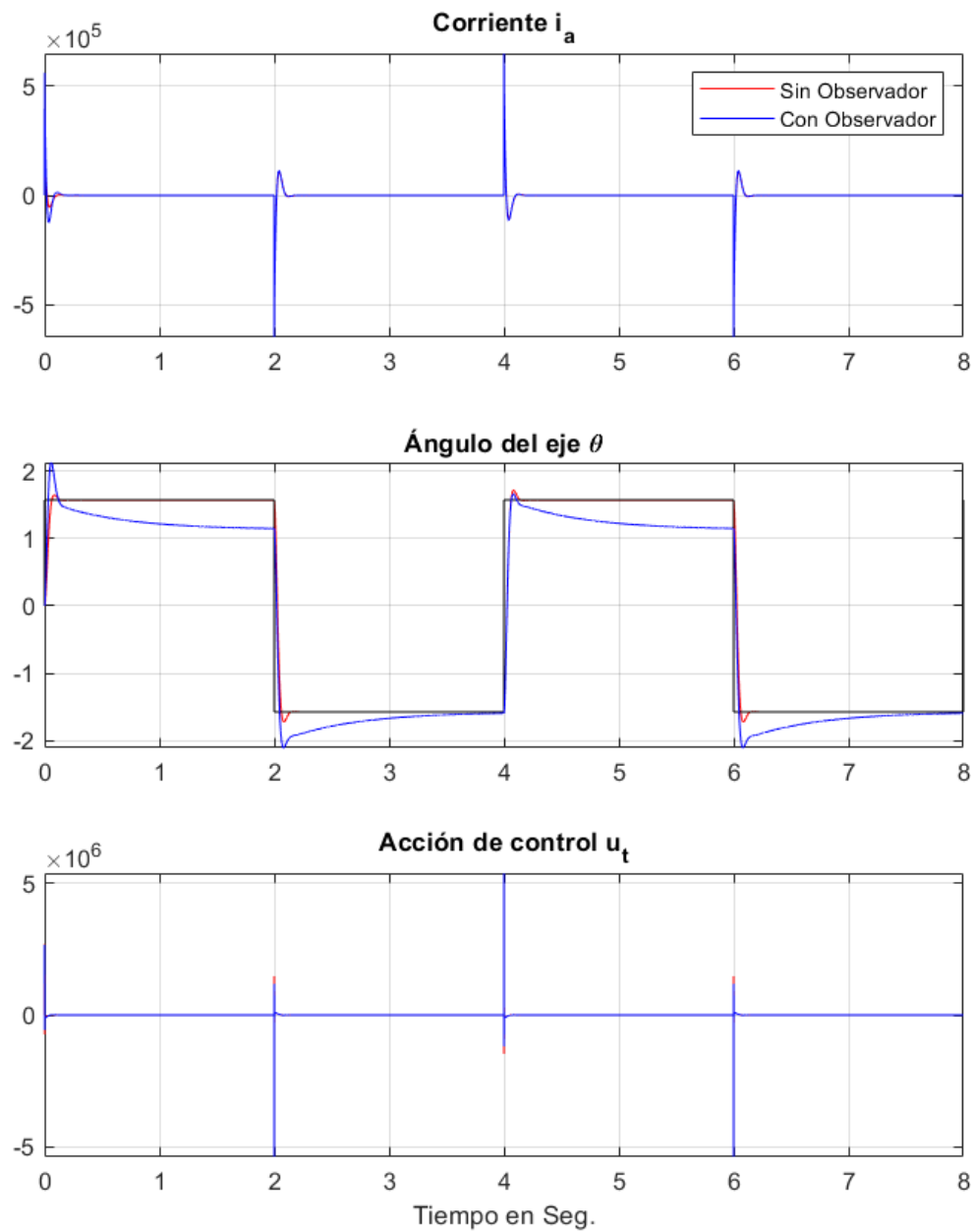
Este permite calcular las nuevas matrices, correspondientes para el diseño del controlador K_o el cual se calculó usando la función 'lqr()' que ofrece Matlab usando las matrices traspuestas de la matriz de estados, la matriz de entrada y la de salida.

$polos\ deseados = [-18.59 + 32.19i; -18.59 - 32.19i; -37.19; -2 \times 10^3]$

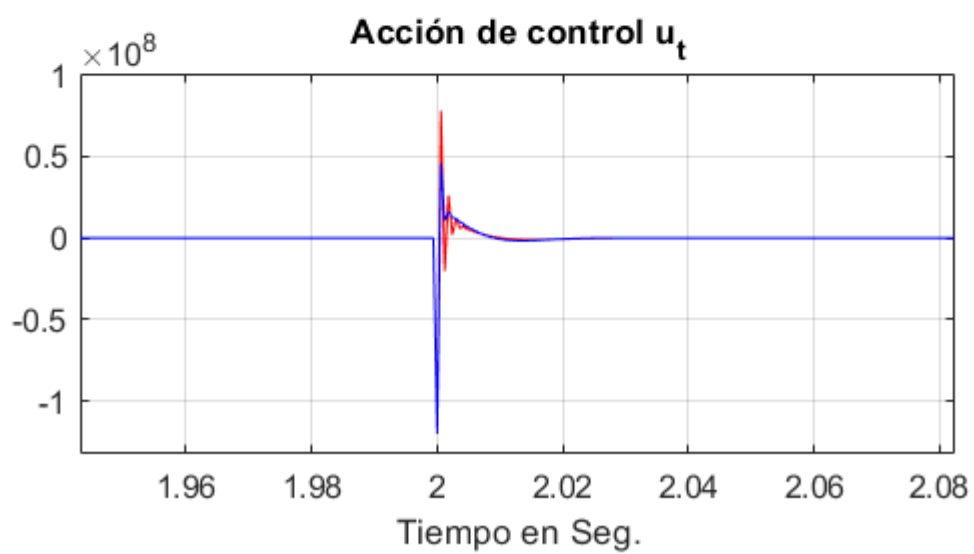
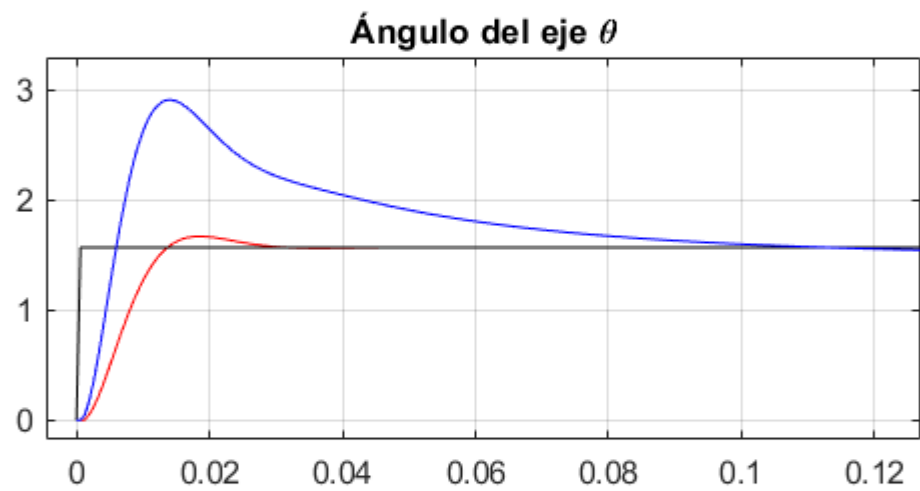
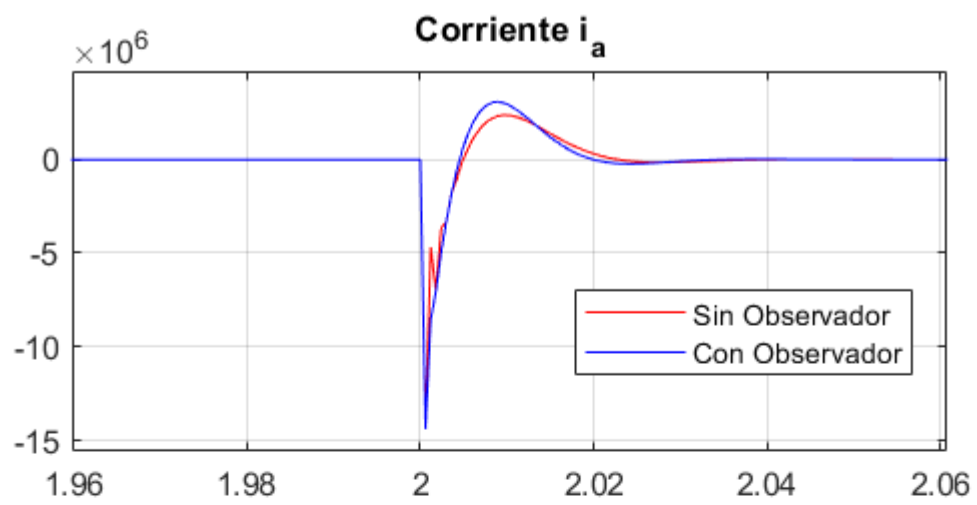
El Controlador sin Observador (usando un controlador ampliado) entrega los polos deseados con parte imaginaria y muy cercanos al eje imaginario. Mientras el sistema con un Controlador con Observador:

$polos\ deseados\ observador = [-1.61; -31.61; -39.96]$

Este nos muestra polos sin parte imaginaria, con valores más bajos, pero siempre del lado izquierdo del eje imaginario, las gráficas se asemejan bastante sobre todo en la señal de corriente, por más que difieran en los polos:



La principal diferencia se observa en lo concerniente a la sobrepasó en que alcanzan al ángulo requerido, vemos que el mismo sin observador es de menor sobrepaso lo cual es muy importante en el diseño de los componentes electrónicos empleados en el diseño, la corriente y la señal de control se asemejan bastante, pero como se usaron los periodos requeridos anteriormente, los valores de corriente son bastante grandes.



Item 3: Calcular un controlador que haga evolucionar al péndulo en el equilibrio inestable, partiendo de una condición inicial nula en el desplazamiento y termine en -10 metros manteniendo la vertical. Determinar el ángulo máximo que puede alejarse de la vertical en $t=0$ para que el sistema cumpla el objetivo de control.

Caso de estudio 2. Sistema no lineal de cuatro variables de estado

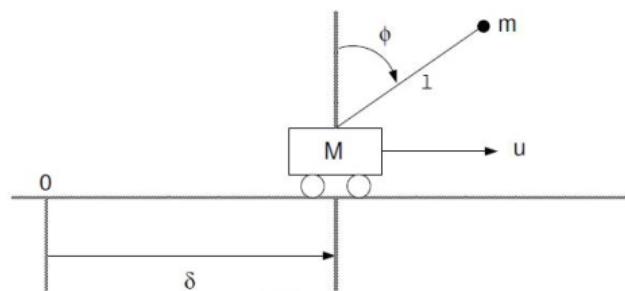


Fig. 2. Sistemas para modelar, extraído de ¹.

Para el caso del esquema del péndulo invertido de la Fig. 2 donde el modelo es,

$$\begin{cases} (M + m)\ddot{\delta} + m l \ddot{\phi} \cos \phi - m l \dot{\phi}^2 \sin \phi + F \dot{\delta} = u \\ l \ddot{\phi} - g \sin \phi + \ddot{\delta} \cos \phi = 0 \end{cases} \quad (4)$$

Con las variables de estado $x = [\delta \quad \dot{\delta} \quad \phi \quad \dot{\phi}]^T$, y los valores de los coeficientes de $m=0,1$; $F=0,1$; $l=1,6$; $g=9,8$; $M=1,5$. Determinar Δt y el tiempo de simulación adecuados.

Considerando las especificaciones dadas en la consigna se desarrolló el algoritmo usando nuevamente el método de Control Óptimo Cuadrático (LQR) donde se definió en primer lugar el tiempo de integración y simulación:

```
% Tomo el autovalor con la dinámica más rápida y el más lento

t_I=log(0.95)/sigma3;    % tr=0.00134
h=5e-3;                  % defino un valor tres veces más chico At=0.005
%t_S=log(0.05)/sigma2;   % t_s=2.3878
t_S=20;                  % defino un valor dos veces más grande
tiempo=round(t_S/h);     % 4000
```

Definiendo el sistema con sus correspondientes matrices a partir de las constantes dadas en el enunciado y definiendo sus autovalores para la definición de los tiempos anteriores:

```

% Valor de las Parámetros:
m=0.1;      % masa del péndulo
Fricc=0.1;  % Fricción del rodado con la superficie
long=1.6;   % largo del péndulo
g=9.8;      % constante de la gravedad
M=1.5;      % mase del carro

% Defino las Matrices del sistema
A=[0 1 0 0; 0 -Fricc/M -(m*g)/M 0; 0 0 0 1; 0 Fricc/(long*M) g*(m+M)/long*M 0];
B=[ 0; 1/M; 0; -1/(long*M)];
C=[1 0 1 0]; % Salida en posición y ángulo
%-Calculo de At-----

% Autovalores de A:
autovalores_A=eig(A);

sigma1=autovalores_A(1); % sigma1=0
sigma2=autovalores_A(2); % sigma2=-0.0648
sigma3=autovalores_A(3); % sigma3=-3.8350

```

De acá se obtiene que la dinámica más veloz es la de $\lambda_3 = -3,84$ con lo cual

se define a $\Delta t = 5 \times 10^{-3}$.

Para el cálculo del Controlador usó el siguiente algoritmo:

```

% Ubicación de los polos de lazo cerrado en mui
% Defino la matriz Q como la identidad:
Q=diag([1 1 1 1]);
% Defino la Matriz R como la
R=0.9;
% Aleternativa a la función LQR
% P=care(A,B,Q,R);
% K=inv(R)*B'*P;
[K,P,E]=lqr(A,B,Q,R);

% Polos del sistema
mui1=round(E(1)); % -1
mui2=round(E(2)); % -1
mui3=round(E(3)); % -4
mui4=round(E(4)); % -4

```

Este nos define los siguientes polos deseados:

$$\mu = [-1 \quad -1 \quad -4 \quad -4]$$

Finalmente para simular el comportamiento del sistema ante estos polos y las la ganancia de prealimentación definida en:

```

% Ganancia de prealimentación de la referencia.
Gj=-inv(C*inv(A-B*K)*B); % Gj=-1.0541

```

Haciendo uso de un ciclo while, se simuló el siguiente script:

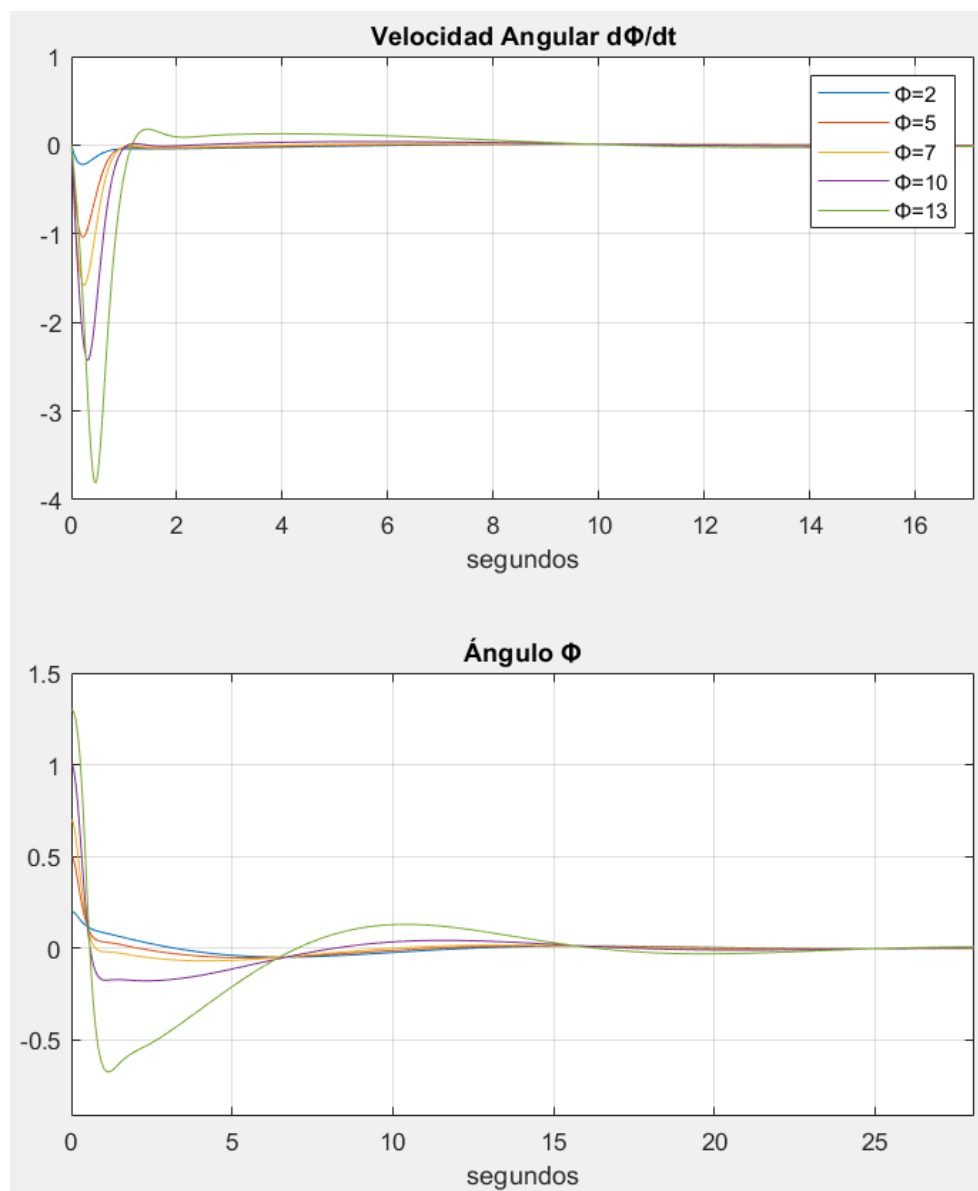
```

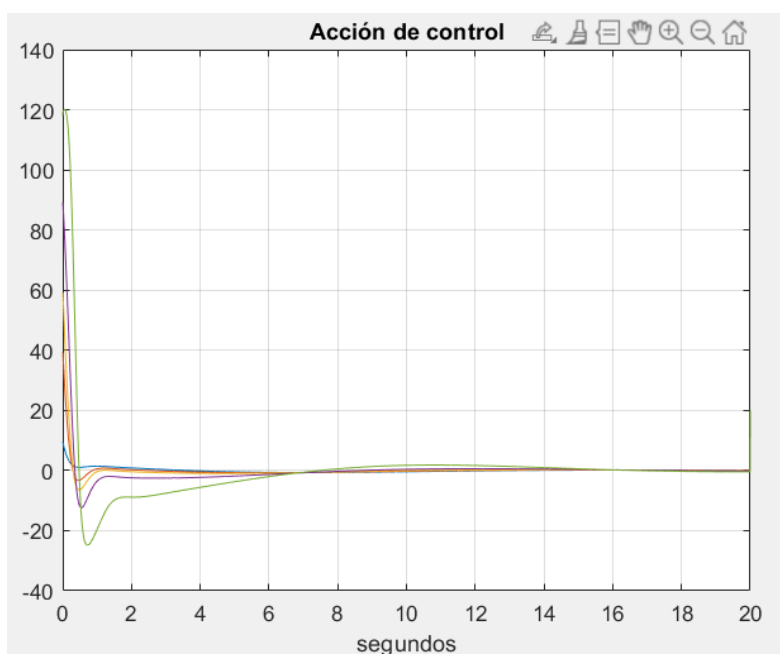
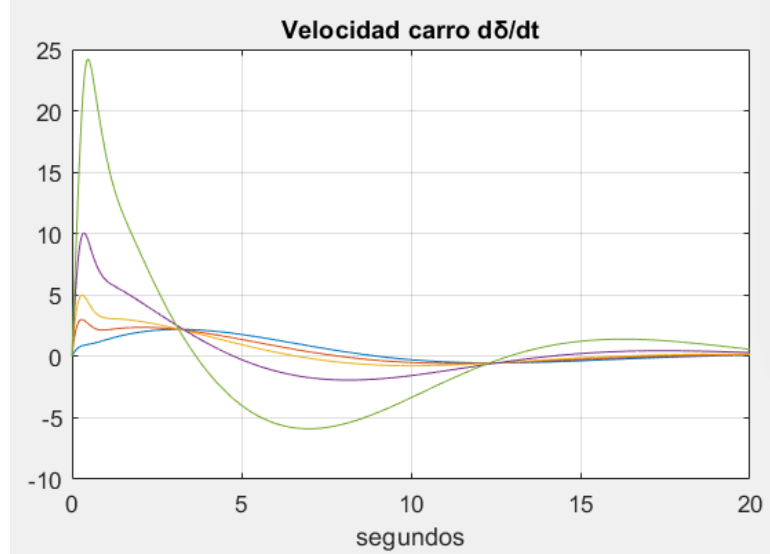
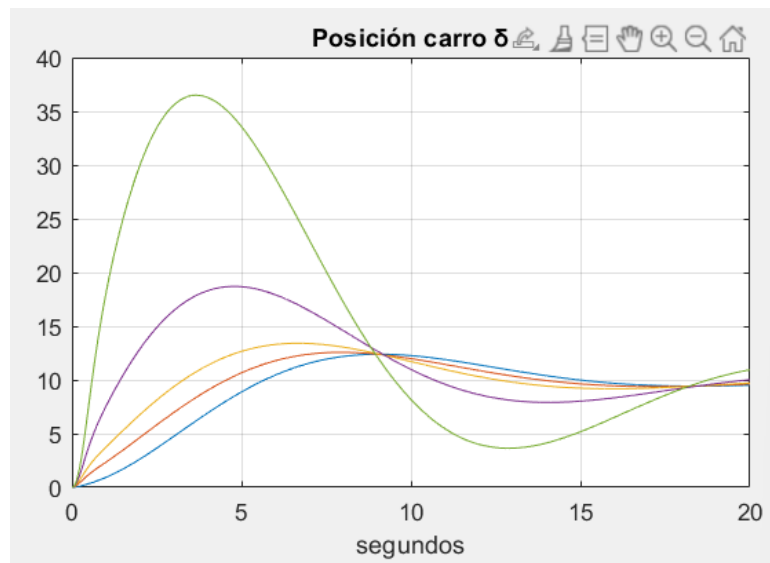
while (ii<tiempo)
    % Definición de Variables de estados
    estado=[p(ii); p_p(ii); alfa(ii); omega(ii) ];
    % Acción de Control
    u(ii)=-K*estado+Gj*ref;
    % Sistema no lineal
    p_pp=(1/(M+m))*(u(ii)-m*long*tita_pp*cos(alfa(ii))+m*long*omega(ii)^2*sin(alfa(ii))-Fricc*p_p(ii));
    tita_pp(ii+1)=(1/long)*(g*sin(alfa(ii))-p_pp(ii)*cos(alfa(ii)));
    p_p(ii+1)=p_p(ii)+h*p_pp(ii);
    p(ii+1)=p(ii)+h*p_p(ii);
    omega(ii+1)=omega(ii)+h*tita_pp(ii);
    alfa(ii+1)=alfa(ii)+h*omega(ii);

    ii=ii+1;
end

```

La gráfica se hace para múltiples valores iniciales de φ , con el fin de obtener el ángulo máximo al cual el sistema puede llegar a corregir la verticalidad del péndulo:





De todo ello se puede concluir que el sistema no responde para valores mayores a 13 radianes, por lo cual este es el valor máximo en que se puede corregir la verticalidad del péndulo, después de este valor el sistema no lo llega a corregir.

Para ello se usó un ciclo for que modificara las condiciones iniciales del ángulo en el que se encuentra el péndulo:

```
for phi=[0.2 0.5 0.7 1 1.3]
    p_p(1)=0; p_pp(1)=0; alfa(1)=phi;
    tita_pp(1)=0; ref=10; omega(1)=0;
```

Item 4: Incorporar un observador para el caso en que sólo puedan medirse el desplazamiento y el ángulo , repetir las simulaciones para las condiciones anteriores y graficar los resultados en gráficas superpuestas.

Considerando los polos anteriormente calculados, el controlador K (sin observador) y la ganancia de prealimentación:

```
% Ubicación de los polos de lazo cerrado en mui.
mui(1)=-1; mui(2)=-1; mui(3)=-4; mui(4)=-4;

% Cálculo del Controlador K.
K=flipplr(alfa_i(2:end)-c_ai(2:end))*inv(T);
% K=[-3.9184,-9.896,-1.0115e+02,-39.6735]
% Ganancia de prealimentación para referencia no nula.
Gj=-inv(C*inv(A-B*K)*B);
% Gj=-3.9184
```

Se procede a calcular el Controlador por Observador, considerando que el mismo tendrá polos mucho más cercanos al eje imaginario, por ende más lentos (5 veces más chico), para ello se probó varias relaciones:

```
% Ubicacion del Observador.
mui_o=real(mui)*0.2;

% Diseño de observador, usando propiedad de dualidad.-----
Mat_A_O=A';
Mat_B_O=C';

% Matriz Controlabilidad.
Mat_M_Dual=[Mat_B_O Mat_A_O*Mat_B_O Mat_A_O^2*Mat_B_O Mat_A_O^3*Mat_B_O];

% Analisis del rango para determinar si es observable.
rango_M_Dual=rank(Mat_M_Dual);
alfaO_i=alfa_i;
```



```

alfaO_i=poly(mui_o);
Mat_T_O=Mat_M_Dual*W;
Ko=(fliplr(alfaO_i(2:end)-c_ai(2:end))*inv(Mat_T_O))';

% Verifica la ubicación de los polos.
eig(Mat_A_O'-Ko*C);
x_hat=[0;0;0;0]; %Inicializo el Observador.

```

Donde se inicializa la matriz “x_hat” correspondiente al observador.
Y se agrega el siguiente código, al ciclo “while”, para el cálculo de la misma valiéndose de las salidas de control, en este caso el ángulo y la posición:

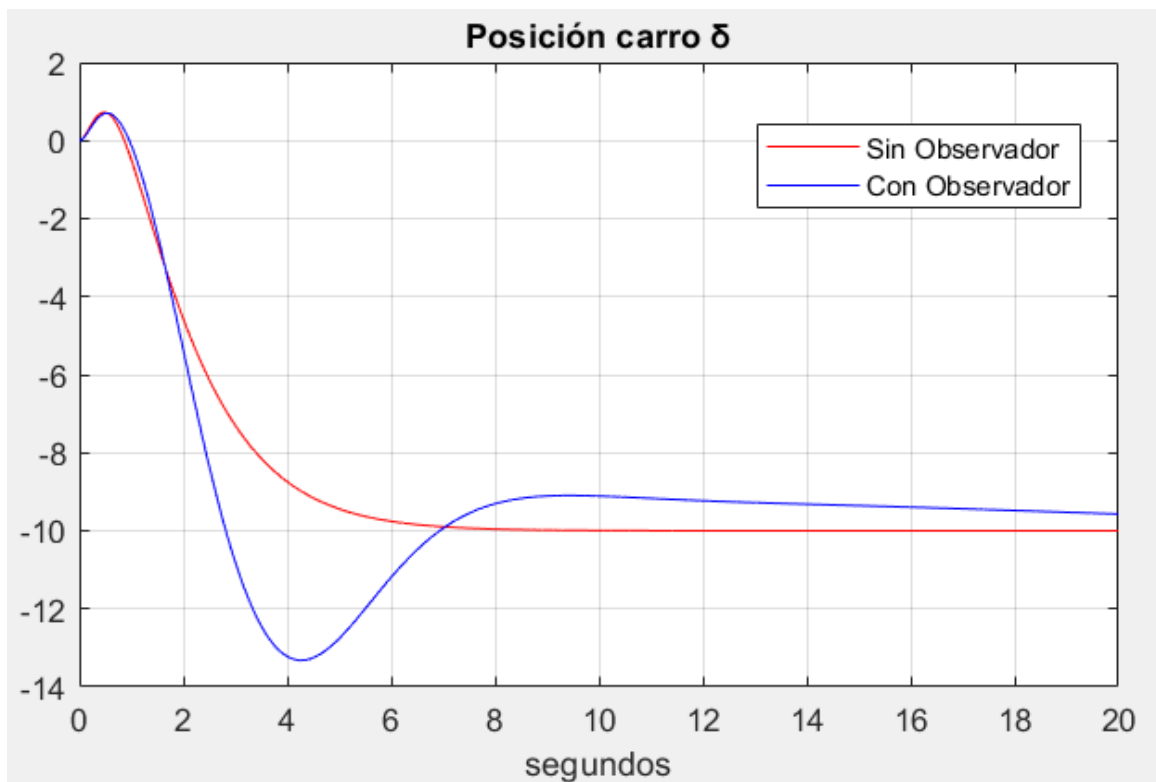
```

% Observador-----
y_sal_O(ii)=C*x_hat;
y_sal(ii)=C*estado;
x_hatp=A*x_hat+B*u(ii)+Ko*(y_sal(ii)-y_sal_O(ii));
x_hat=x_hat+h*x_hatp;

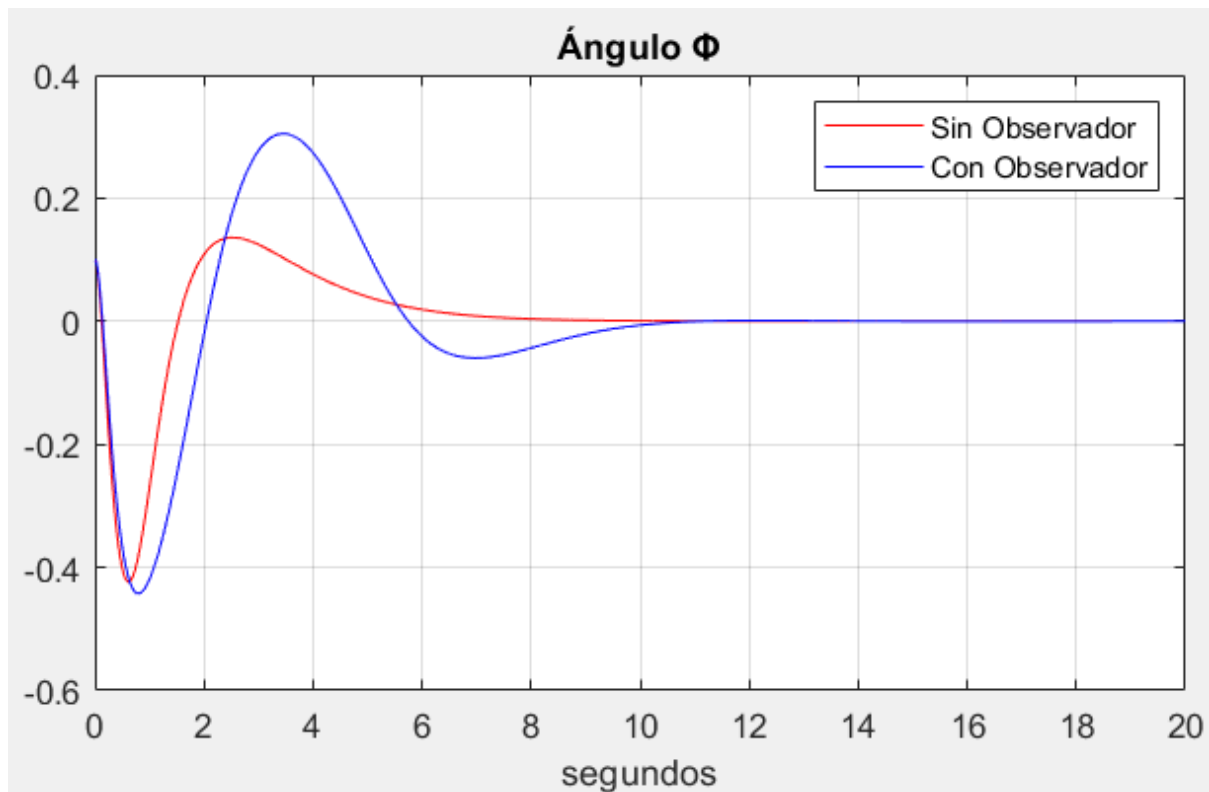
ii=ii+1;

```

Los gráficos muestran en azul al sistema “Con Observador” y con rojo al sistema “Sin Observador”, en el gráfico de posición vemos como se estabiliza en -10 metros y como ambas curvas logran hacerlo pero con diferentes sobrepasos y punto de establecimiento, el sistema sin un observador es más rápido y menos oscilante.



Para el caso del ángulo se repite la misma situación, se ve una marcada diferencia en el sobrepaso de ambos y el tiempo de estabilización, pero ambos se estabilizan en el ángulo 0 (cero) deseado.



Item 5: Calcular un controlador que haga evolucionar al péndulo en el equilibrio estable, partiendo de una condición inicial nula en el desplazamiento y el ángulo en que termine en 2 metros evitando las oscilaciones de la masa m , considerando que es una grúa. Una vez que $m=2$ modificar a m a un valor 10 veces mayor y volver al origen evitando oscilaciones.

Se definen los parámetros del problema en un comienzo:

```
% Condiciones iniciales.
omega(1)=0; d_p(1)=0; d(1)=0;
u(1)=0; ii=1; alfa(1)=pi;
referencia=2;
```

Donde defino el ángulo inicial en π y la correspondiente posición en 2 metros.

Para el cálculo del controlador se usó el LQR, mediante la construcción del Hamiltoniano y el cálculo de la matriz P . Este se usó para este caso, a razón de que la función "lqr()" no alcanzaba a entregar un controlador K que sea capaz de reducir la oscilación ni las condiciones pedidas por el enunciado.

```

% Cálculo del controlador LQR-----
Q=diag([0.3 13 5e-3 1e-4]);
R=0.05;

% Contrucción del Hamiltoniano
H=[A -B*inv(R)*B'; -Q -A'];
[V,D]=eig(H);MX1X2=[];
for(jj=1:8)
    if real(D(jj,jj))<0
        MX1X2=[MX1X2 V(:,jj)];
    end
end
MX1=MX1X2(1:4,:); MX2=MX1X2(5:8,:);
P=real(MX2*inv(MX1));
K=inv(R)*B'*P;

```

Dando los polos en la siguiente ubicación:

```

% Verifica la ubicación de los polos.
eig(A-B*K)

-10.7126 + 0.0000i
-0.1519 + 0.0000i
-0.0191 + 2.4790i
-0.0191 - 2.4790i

```

Todos del lado izquierdo del eje imaginario, con lo cuál cumplen con la condición de estabilidad.

La Matriz Q y R, quedan definidas de la siguiente manera:

```

% Cálculo del controlador LQR-----
Q=diag([0.9 15 6e-3 9e-6]);
R=0.01;

```

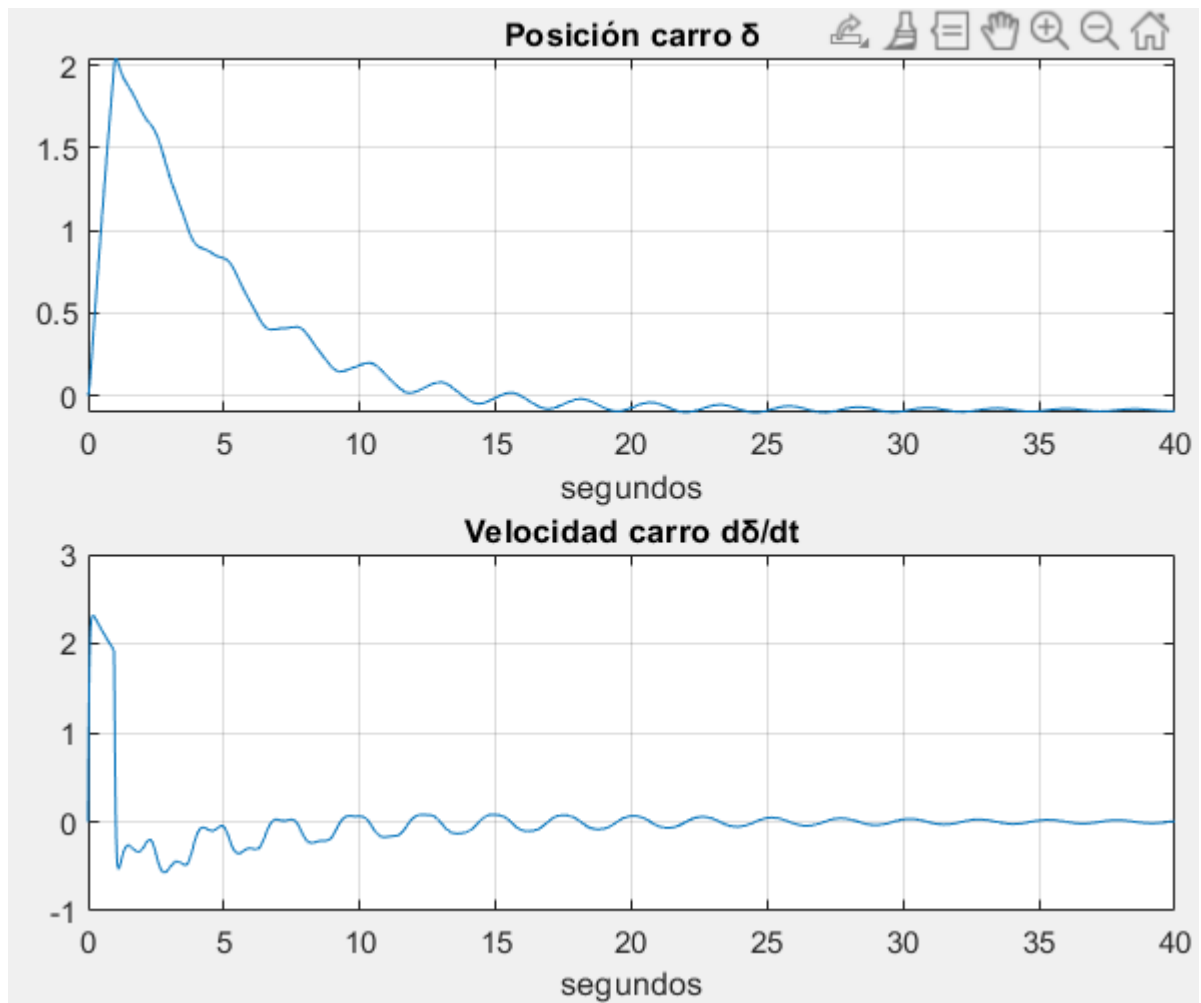
Para poder variar el valor de posición final deseada se usó un “if” para que esta cambiara al alcanzar el valor de 2 metros, al mismo tiempo se cambia el valor de la masa del péndulo a 1 kg:

```

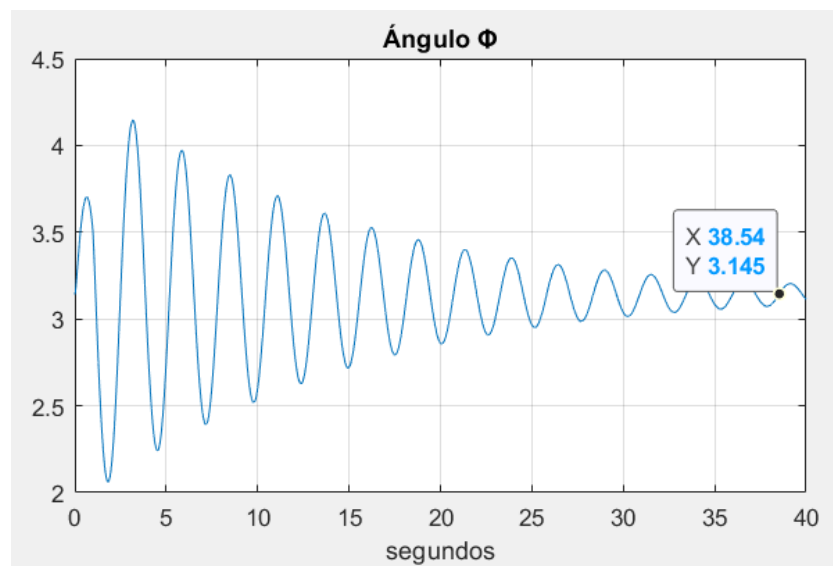
if d(ii) > 2
    referencia=0;
    m=1;
end

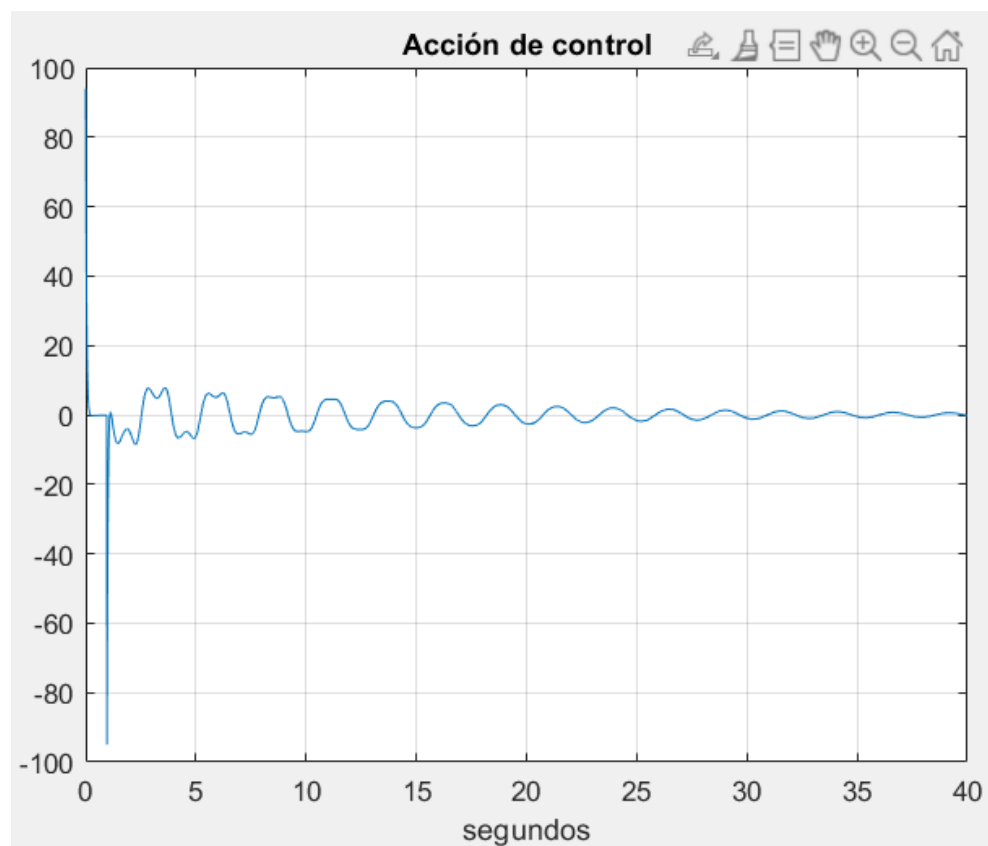
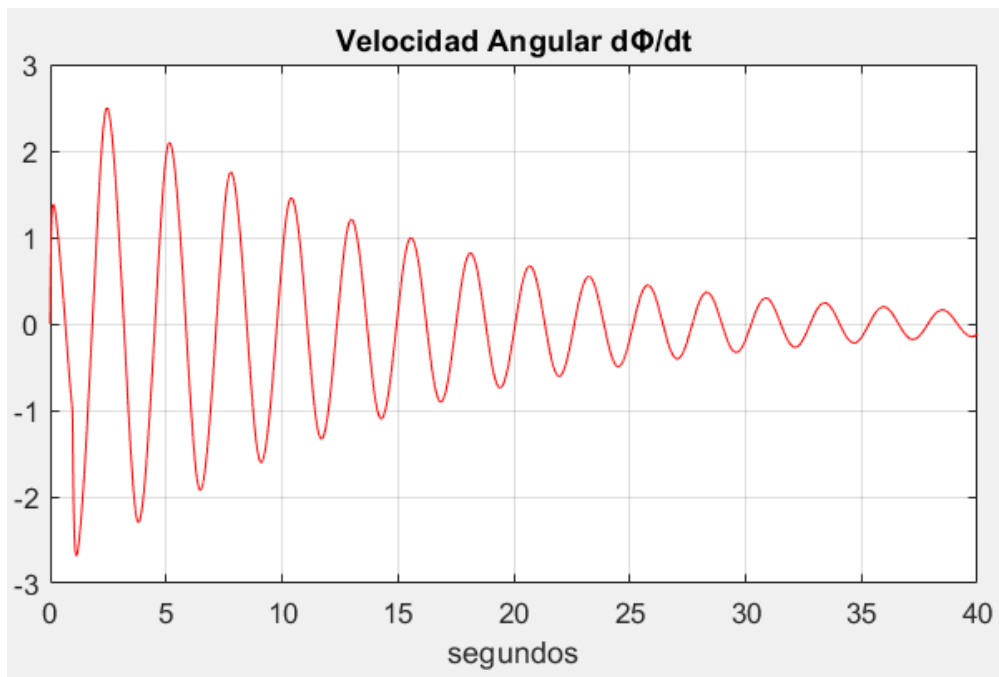
```

El gráfico resultante muestra como la posición varía hacia 2 metros, pero ante el cambio de posición deseada, decrece tratando de alcanzar la posición cero.



Así mismo vemos como el ángulo inicial se mantiene oscilando en el mismo valor:





El gráfico de velocidad y el de control muestran así mismo como generan la señal necesaria para mantener el ángulo y la posición en los parámetros deseados, especialmente la de control que en un momento cambia abruptamente ante el cambio de posición a 0 metros.

Item 6: Incorporar un observador para el caso en que sólo puedan medirse el desplazamiento y el ángulo, repetir las simulaciones para las condiciones anteriores y graficar los resultados en gráficas superpuestas para el equilibrio estable.

Para este ítem, se debió diseñar un nuevo controlador mediante la incorporación de un observador, para ello se usó el mismo método LQR:

```
% Cálculo del controlador LQR para sistema con observador-----
Ao=A';
Bo=C';
Co=B';

Qo=diag([5 400 1e-3 1e-10]);
Ro=0.001;

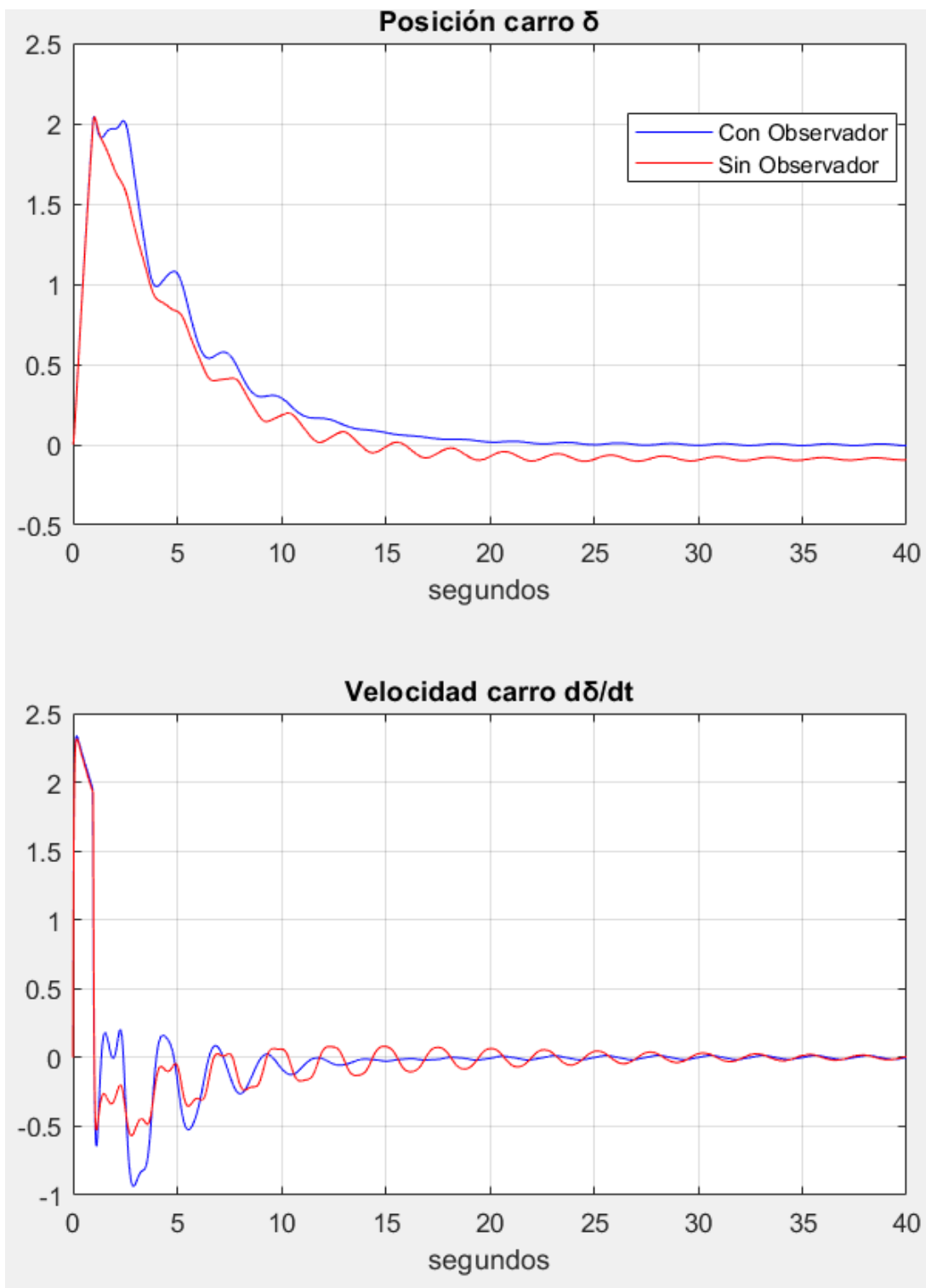
% Contrucción del Hamiltoniano
Ho=[Ao -Bo*inv(Ro)*Bo'; -Qo -Ao'];
[no,vao]=size(Ho);
[V,D]=eig(Ho);MX1X2=[];
for(jj=1:8)
    if real(D(jj,jj))<0
        MX1X2=[MX1X2 V(:,jj)];
    end
end
MX1=MX1X2(1:no/2,:); MX2=MX1X2((no/2)+1:end,:);
Po=real(MX2*inv(MX1));
Ko=inv(Ro)*Bo'*Po;
% Fin Cálculo de Controlador Observable-----
```

El controlador anteriormente calculado, no cambia, mientras este nos da unos nuevos polos del sistema:

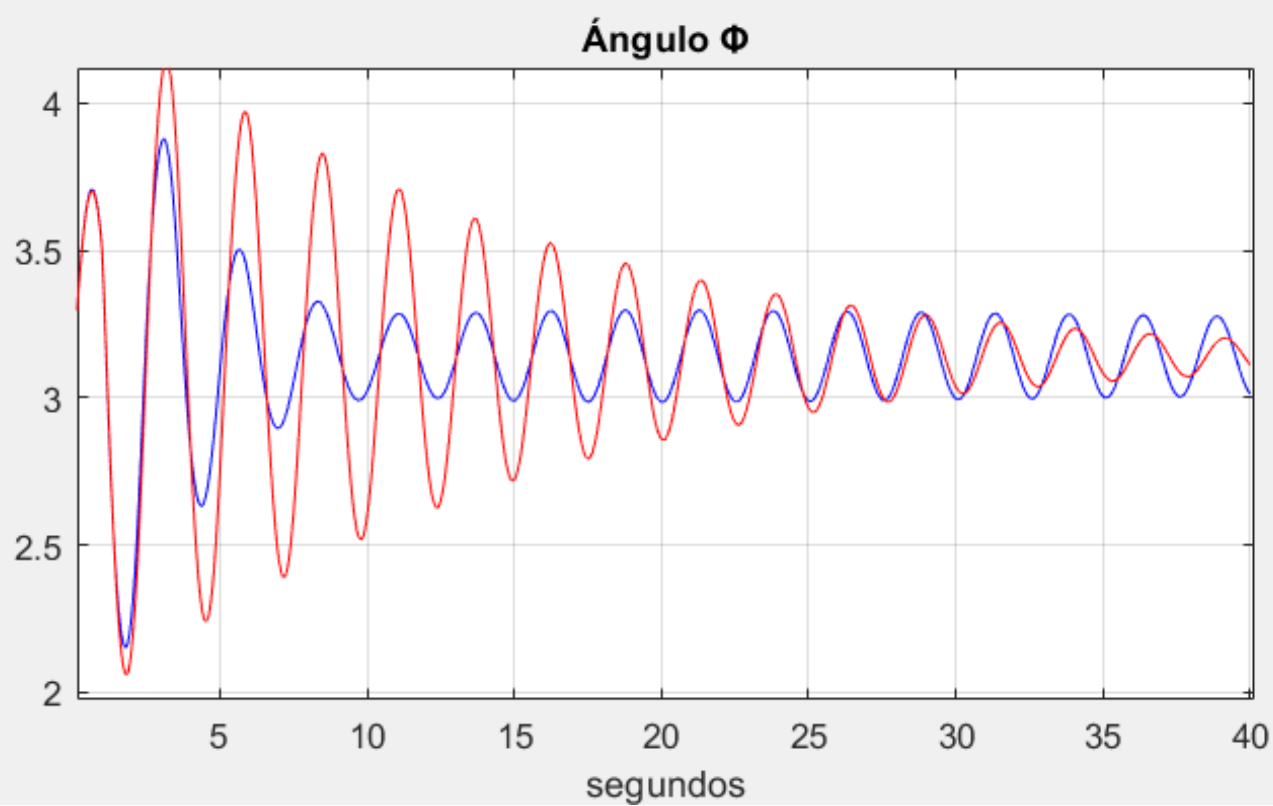
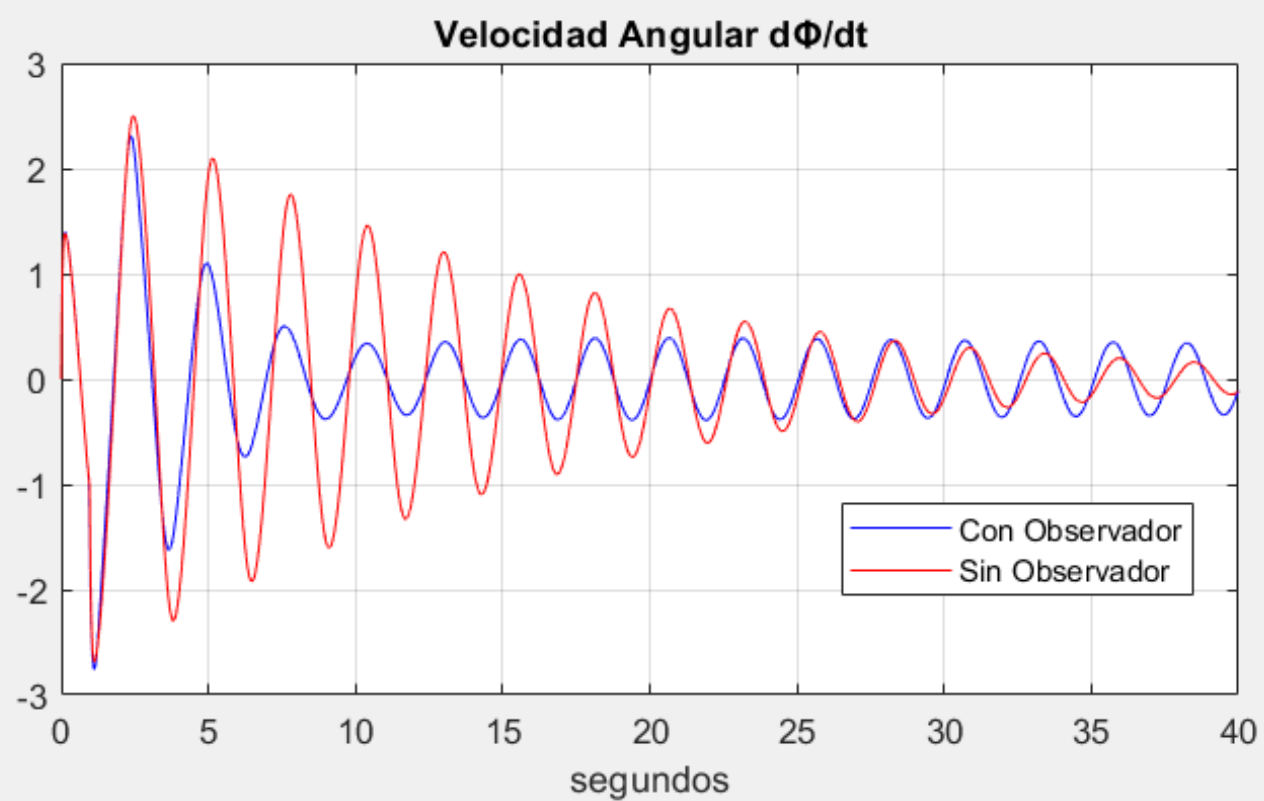
```
-70.1333 + 0.0000i
-9.0182 + 0.0000i
-0.0008 + 2.5560i
-0.0008 - 2.5560i
```

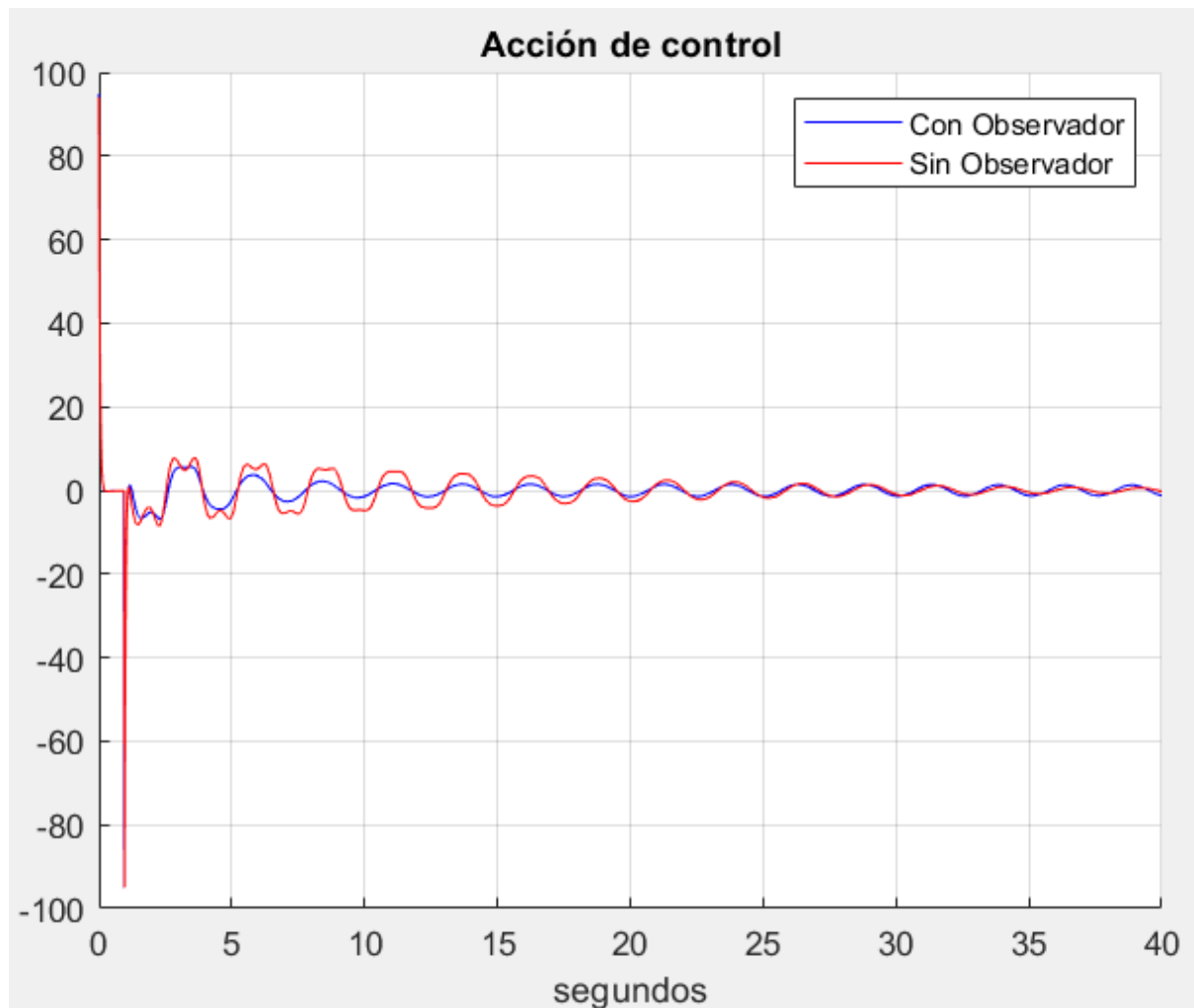
Todos los valores del lado izquierdo del eje real por lo cual cumplen con la condición de estabilidad. Y al igual que en los otros ítems que implican el uso de un observador se sumó al ciclo while el siguiente código:

```
% Observador-----
y_sal_O(ii)=C*x_hat;
y_sal(ii)=C*estado;
x_hatp=A*x_hat+B*u(ii)+Ko'*(y_sal(ii)-y_sal_O(ii));
x_hat=x_hat+h*x_hatp;
ii=ii+1;
```



Para este caso, se ve como el sistema con el observador sigue muy de cerca al sistema sin observador, pero con un pequeño retraso. Mientras en la acción de control el seguimiento es mucho más cercano.





El ángulo en sí mismo, se nota una mayor oscilación en los valores correspondientes al sistema sin observador, al principio, pero después vemos que el seguimiento es mucho más cercano, pasados los 25 segundos.

Esta superposición de gráficas se logra, al igual que en los anteriores ítems, modificando el siguiente código:

```
% Acción de control.
u(ii)=-K*estado+Gj*referencia; color='r';% Sin Observador.
%u(ii)=-K*x_hat+Gj*referencia; color='b';% Con Observador.

%clear all; close all; clc;
```