Sistemas de Control II

Trabajo práctico Nº4

UNIVERSIDAD NACIONAL DE CÓRDOBA

FCEFyN

- Profesor: Ing. Laboret, Sergio
- Alumno: Valdez Benavidez, Mauricio L.

DESARROLLO

Datos indicados para el desarrollo del trabajo práctico:

m	b	l	G	Delta	p (triple)
1	0.4	1	10	180	-4

Se dispondrá para cada alumno una tabla con valores de masa (m), longitud (1), coeficiente de rozamiento (b), constante gravitatoria g (10) y ángulo de referencia en grados (delta) de un péndulo simple con la ecuación:

$$ml^2\ddot{\theta} + b\dot{\theta} + mglsen(\theta) = T$$

Se desea que el péndulo se estabilice en el ángulo δ dado tomando como estados, entrada y salida respectivamente (nótese que se ha desplazado el punto de equilibrio al origen tomando el error como salida)

$$x_1 = \theta - \delta = e$$

$$x_2 = \dot{\theta}$$

$$u = T$$

$$y = e$$

- Hallar el sistema dinámico en VE

Para hallar el sistema dinámico, se realizan las siguientes asignaciones:

$x_1 = \theta - \delta = e$
$\dot{x_1} = \dot{\theta} = x_2$
$x_2 = \dot{\theta}$
$\dot{x}_2 = \ddot{\theta}$

$$\dot{x}_2 = \ddot{\theta} = \frac{T - b \cdot \dot{\theta} - m \cdot g \cdot l \cdot sen(\theta)}{m \cdot l^2} = \frac{T}{m \cdot l^2} - \frac{b \cdot \dot{\theta}}{m \cdot l^2} - \frac{g \cdot sen(\theta)}{l}$$

$$\dot{x}_2 = \frac{u}{m \cdot l^2} - \frac{b \cdot x_2}{m \cdot l^2} - \frac{g \cdot sen(x_1 + \delta)}{l}$$

Definiendo

$$\alpha = \frac{1}{m \cdot l^2}$$

$$\beta = \frac{g}{l}$$

$$\dot{x_1} = x_2$$
$$\dot{x}_2 = \alpha\, u - \alpha\, b\, x_2 - \beta sen(x_1 + \delta)$$

- Hallar el torque estático necesario u_f para que el sistema tenga como punto de equilibrio el origen, es decir $f(0, u_f) = 0$

Para que el sistema tenga como punto de equilibrio el origen, el torque debe tener una componente estática que compense la fuerza de la gravedad. Por lo tanto la acción de control debe ser:

$$u = T - T_f \ \rightarrow T = u + T_f$$

Para hallar T_f se utiliza la ecuación hallada valuada en el origen, es decir $f\left(o, u_f\right) = 0$

$$\dot{x}_2 = f\left(0, u_f\right) = \alpha\, u_f - \alpha\, b\, 0 - \beta \cdot sen(0 + \delta) = 0$$

$$u_f = \frac{\beta \cdot sen(\delta)}{\alpha} = \frac{\frac{g}{l} \cdot sen(\delta)}{\frac{1}{m \cdot l^2}}$$

$$u_f = mglsen(\delta) = 1 * 10 * 1 * sen(180) = 0$$

Ahora se reemplaza T para obtener el sistema que tiene el punto de equilibrio en el origen

$$\dot{x_1} = f_1(x, u) = \dot{\theta}$$
$$\dot{x}_2 = f_2(x, u) = \ddot{\theta} = \alpha u - \alpha \cdot b \cdot x_2 - \beta \cdot [sen(x_1 + \delta) - sen(\delta)]$$

- Linealizar el sistema mediante la jacobiana

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

Valuando en el punto

$$\begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ u_f \end{bmatrix}$$

Se comienza calculando las matrices

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} \cdot \cos(\delta) & -\frac{b}{m \cdot l^2} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{m \cdot l^2} \end{bmatrix}$$

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

Quedando las matrices valuadas:

```
Aa =

         0    1.0000
   10.0000   -0.4000

Bb =

     0
     1

Cc =

     1     0
```

- Hallar los autovalores de A y determinar estabilidad por el método indirecto de Lyapunov

```
>> eig(Aa)

ans =

    2.9686
   -3.3686
```

Se determina la estabilidad del sistema por el método indirecto de Lyapunov, el cual utiliza los autovalores de la matriz A. Se establece que si los autovalores:

- Resultan todos con <u>parte real negativas</u>, el origen del sistema no lineal es *asintóticamente estable*.
- Existe al menos uno cuya <u>parte real sea positiva</u>, el origen del sistema no lineal es *inestable*.
- Sin parte real positiva ni nulos, pero hay al menos uno *sobre el eje jw*, *el método falla y los términos de orden superior deciden la estabilidad*.

Para este caso, se tiene un autovalor positivo y uno negativo, por lo que el origen del sistema no lineal es ***inestable.***

- Comparar los resultados obtenidos con los de la linealizacion por Matlab y Simulink

Se obtienen las matrices y coinciden con las anteriormente calculadas. Luego se calculan los autovalores y coinciden con los calculados antes también además se calculo la controlabilidad del sistema verificandola, entonces se concluye que los resultados son correctos

Command Window

```
A =

         0    1.0000
   10.0000   -0.4000


B =

     0
     1


C =

     1     0


D =

     0
```

```
autoval_2 =

    2.9686
   -3.3686

La matriz A es controlable
>>
```

- Encontrar las matrices del sistema ampliado:

$$A_A = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}; \quad B_A = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

```
AA =

         0    1.0000         0
   10.0000   -0.4000         0
    1.0000         0         0
```

```
BB =

     0
     1
     0
```

- Verificar autovalores, estabilidad y controlabilidad del nuevo par

```
autoval_AA =

         0
   -3.3686
    2.9686


ans =

     3

La matriz AA es controlable
>>
```

- Diseñar por asignación de polos un controlador con la orden acker() de matlab

$$u = -[k_1 \quad k_2 \quad k_3] \cdot \begin{bmatrix} x_1 \\ x_2 \\ \sigma \end{bmatrix}$$

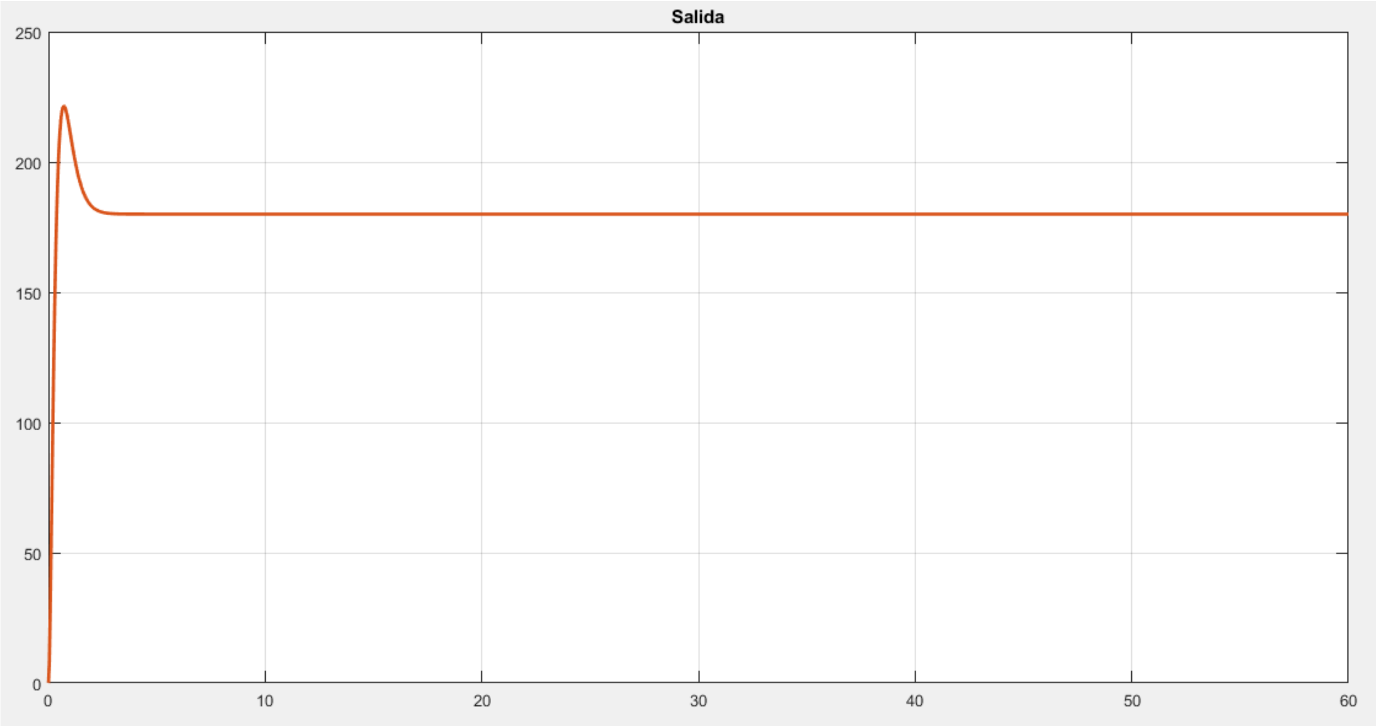
para ubicar un polo triple p, lo cual daría una respuesta sin sobrepaso (si el sistema fuera lineal y no tuviera ceros de lazo cerrado) y el tiempo 2% seria $t_{ss} \backsimeq \frac{7.5}{-p}$.
Mediante la línea de código K = acker(AA,BB,[p p p]), fue posible obtener el siguiente controlador:

```
K =

   58.0000   11.6000   64.0000
```

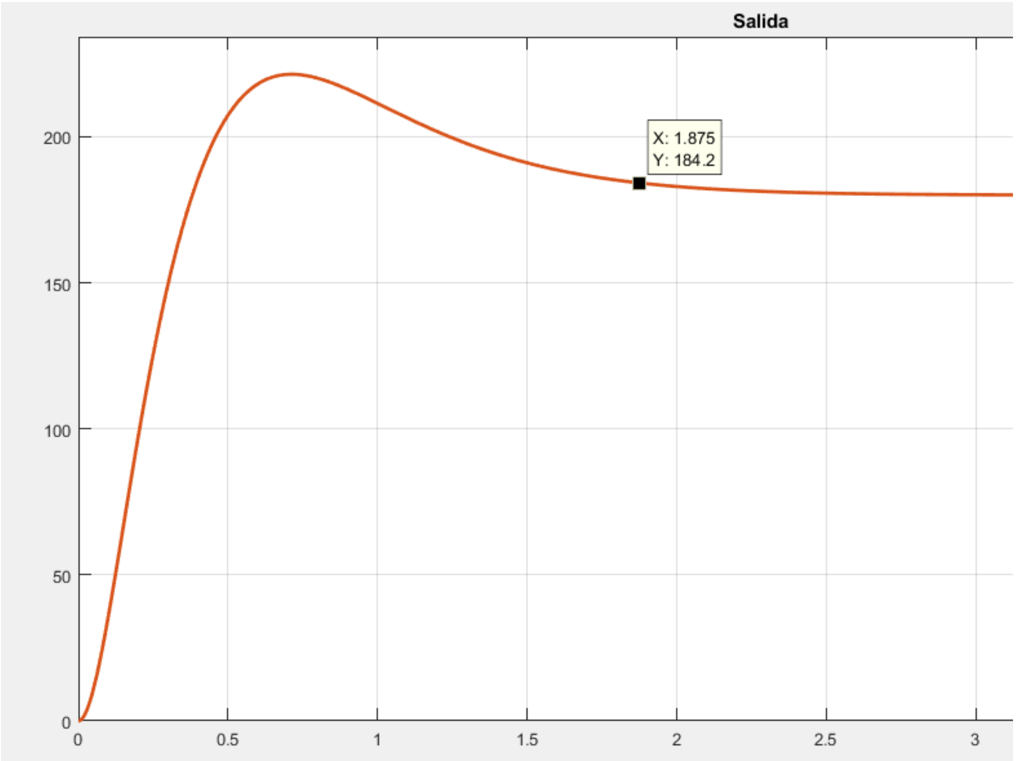
Se corrobora que el polo triple esté en -4 y se calcula el tiempo de respuesta

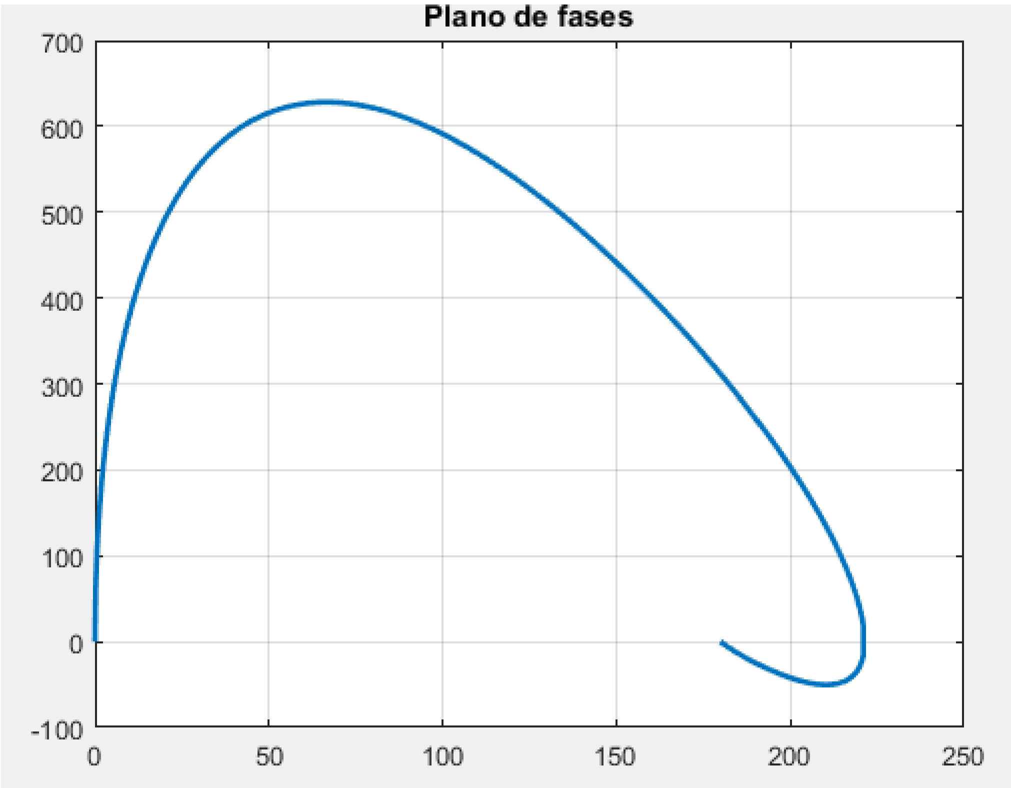
```
ans =

  -4.0000 + 0.0000i
  -4.0000 - 0.0000i
  -4.0000 + 0.0000i


tscalc =

    1.8750

>>
```

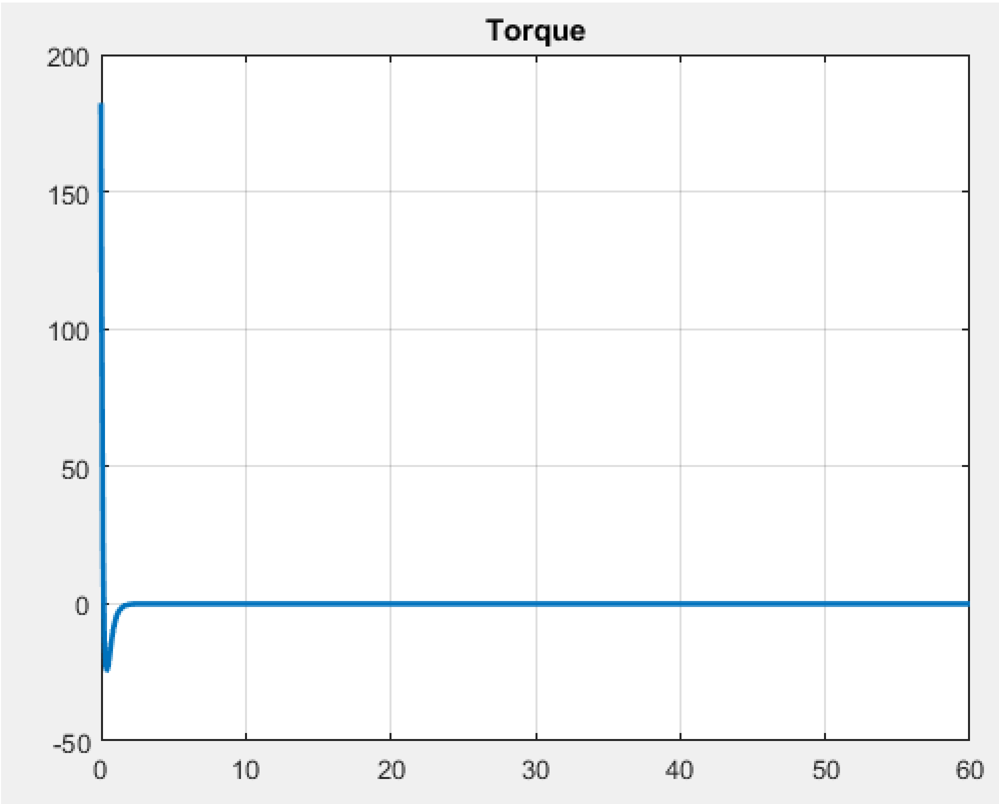
- Simular el péndulo con PID como se muestra en la figura partiendo del origen con velocidad nula y referencia δ (dato)
- Dibujar la salida, el plano de fases, el torque total y la acción integral, comparar con el valor de u_f calculado antes y verificar sobrepaso y tiempo de establecimiento real versus el calculado
- En todos los casos ajustar las escalas para una visualización correcta Se adjunta el archivo de simulación pendulo_PID_tarea.slx



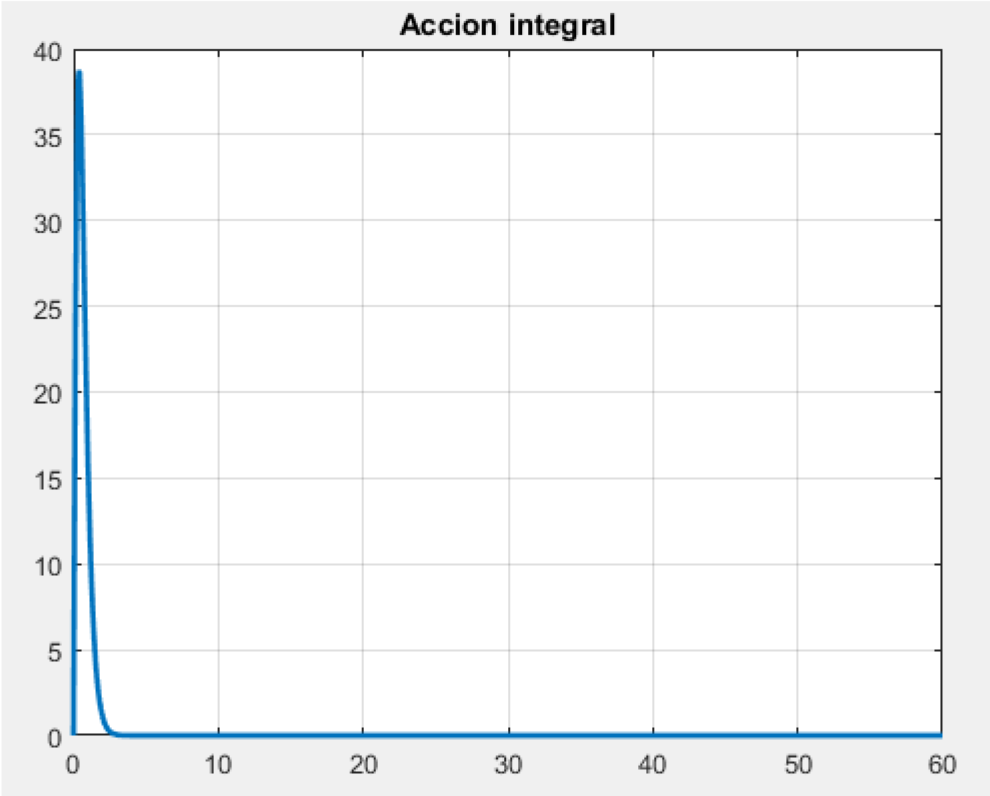
Se verifica en la gráfica que el tiempo de establecimiento calculado de 1.875[s]







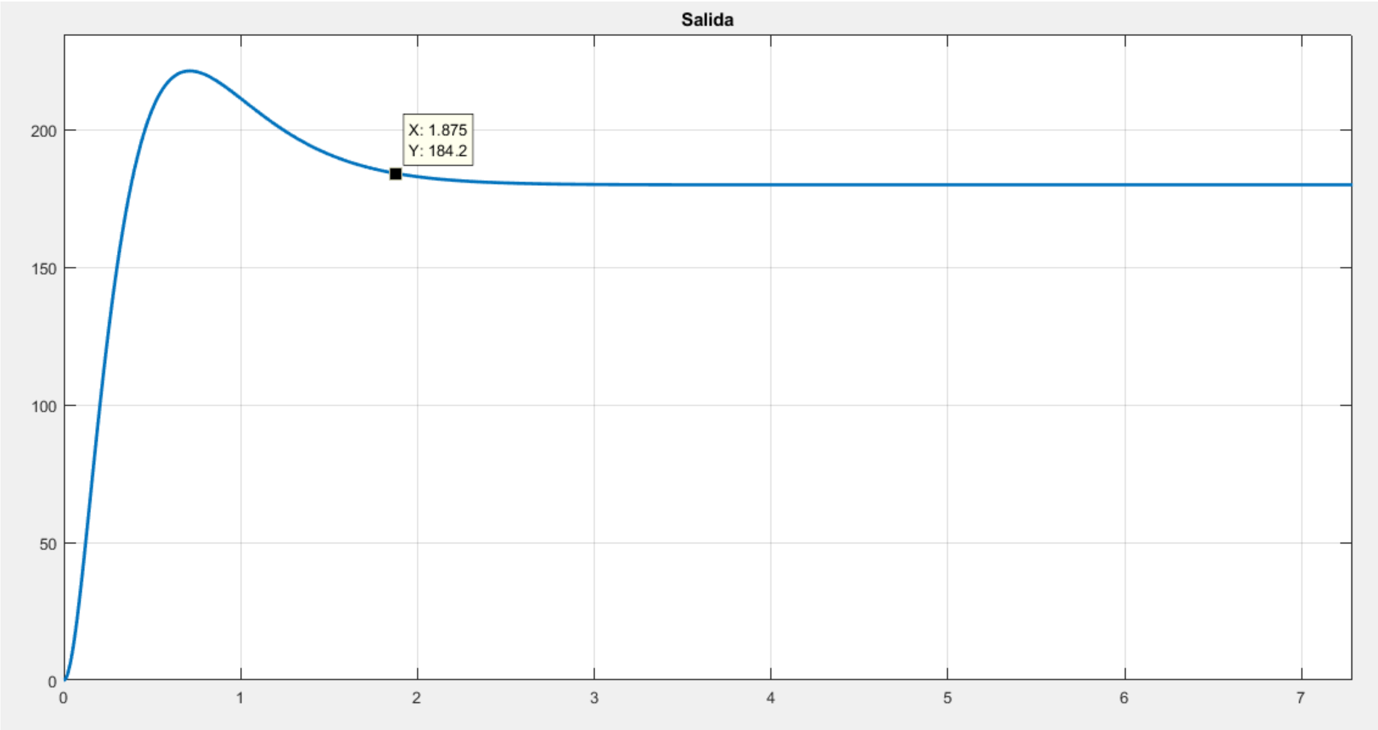
Puede apreciarse que el torque en estado de régimen tiene el mismo valor que el máximo necesario para establecer el sistema en cero, lo cual corrobora la exactitud del cálculo sobre Tf realizado en la sección anterior.

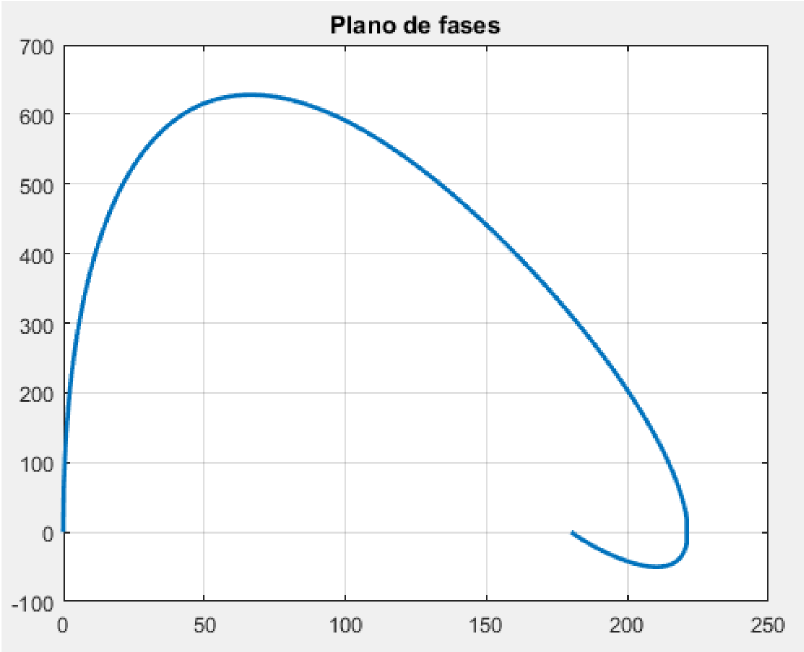


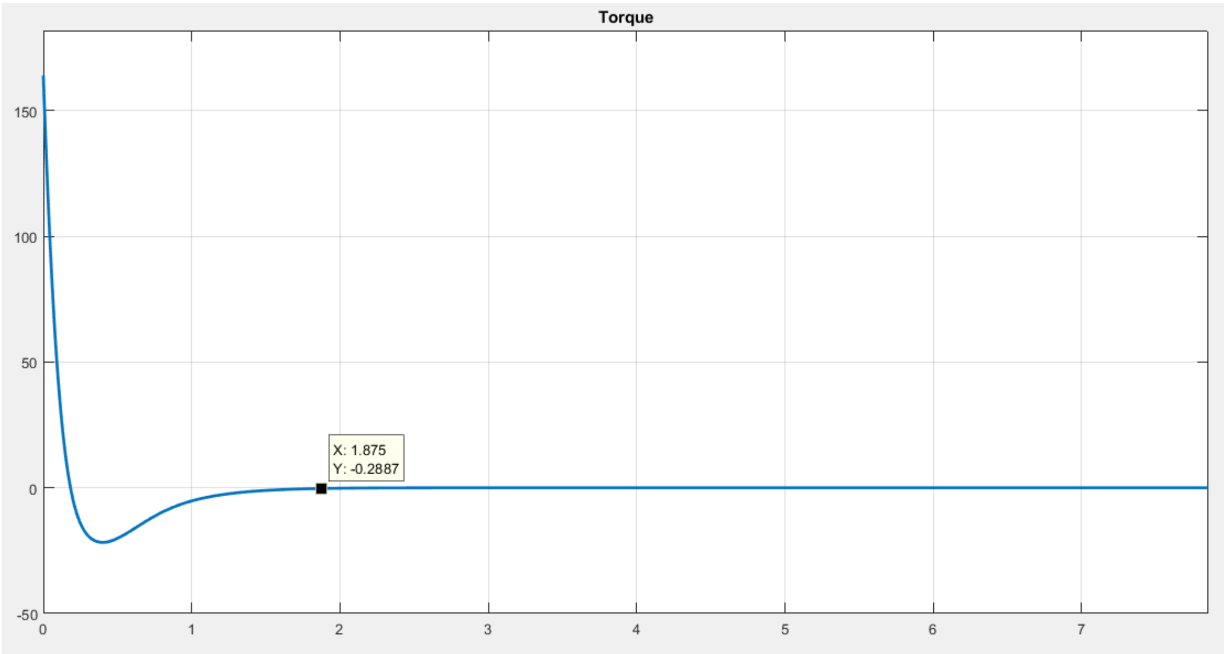
Aquí se ve el comportamiento de la acción integral

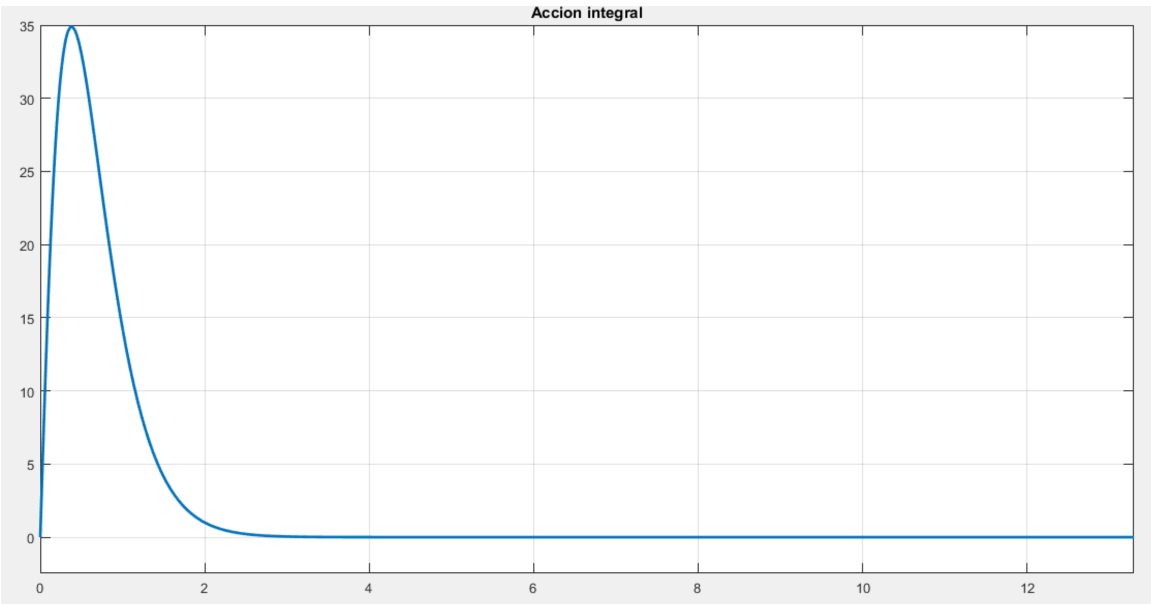
- Analizar la robustez variando la masa del péndulo en más y menos 10% analizar los nuevos valores de sobrepaso, tiempo de establecimiento y acción de control final, elaborar una tabla con los resultados para los distintos valores de masa

<u>Simulaciones para m disminuida</u>

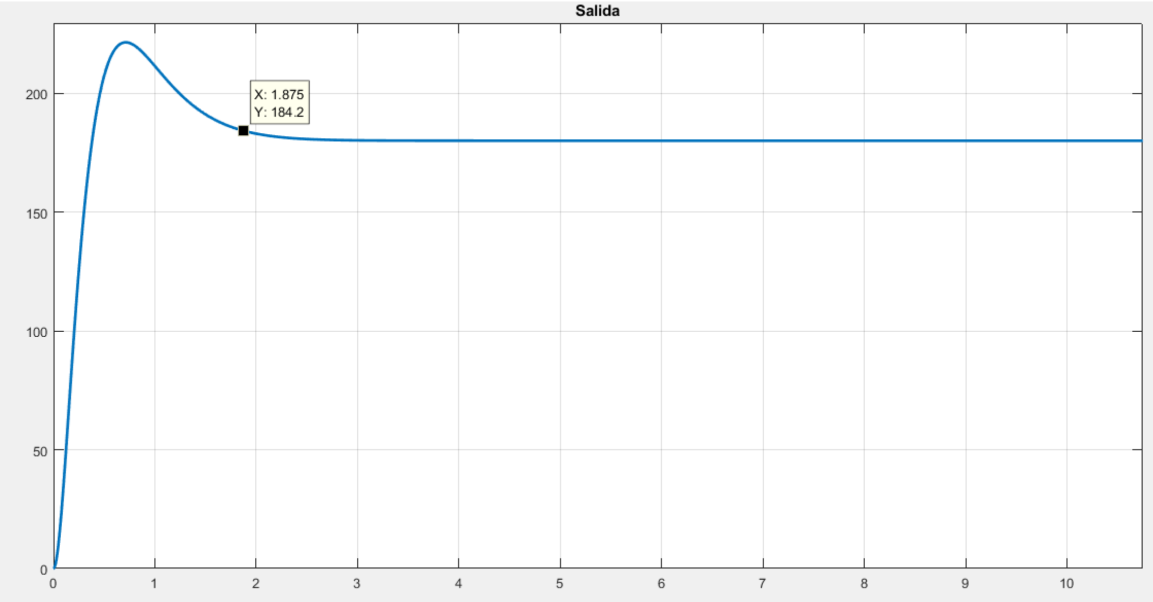


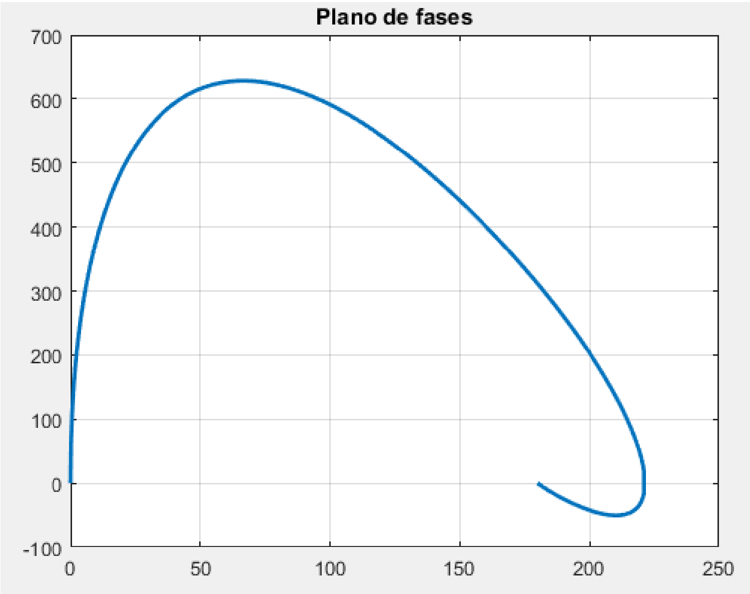

Plano de fases
700
600
500
400
300
200
100
0
-100
0
50
100
150
200
250

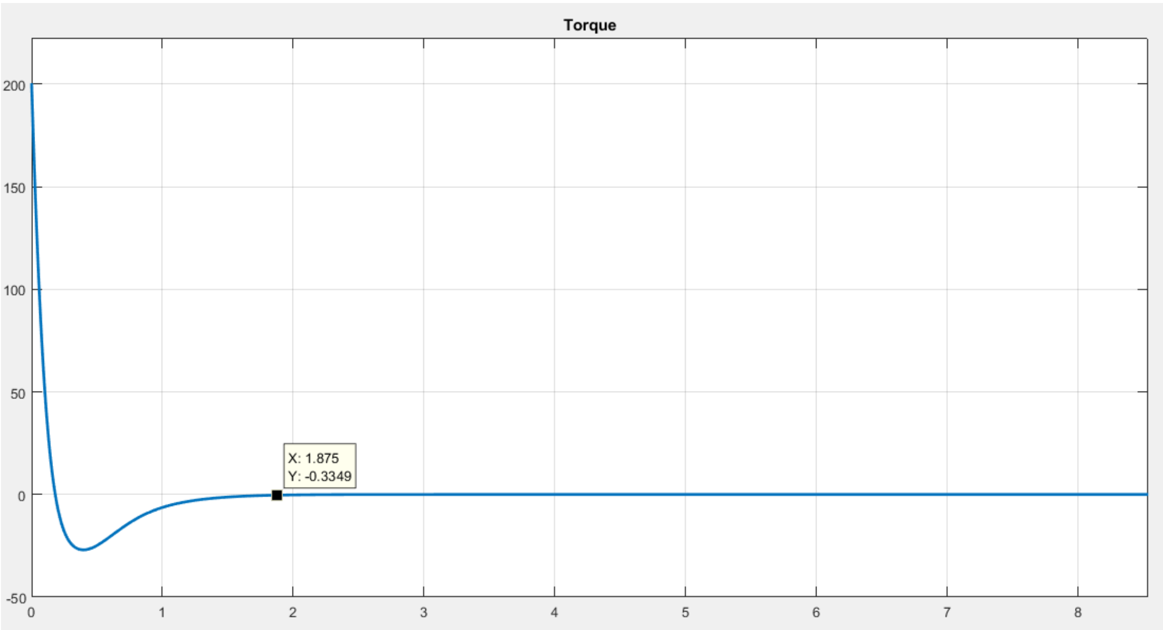

Torque
150
100
50
0
-50
0
1
2
3
4
5
6
7
X: 1.875
Y: -0.2887

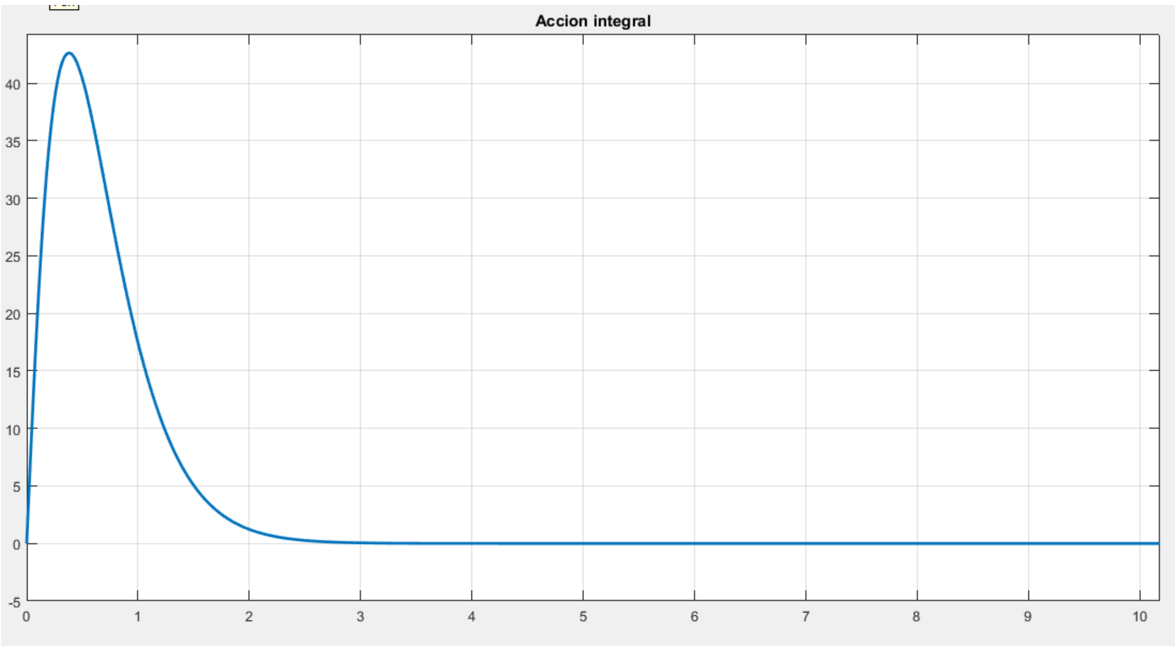

Accion integral
35
30
25
20
15
10
5
0
0
2
4
6
8
10
12

Simulaciones para m aumentada







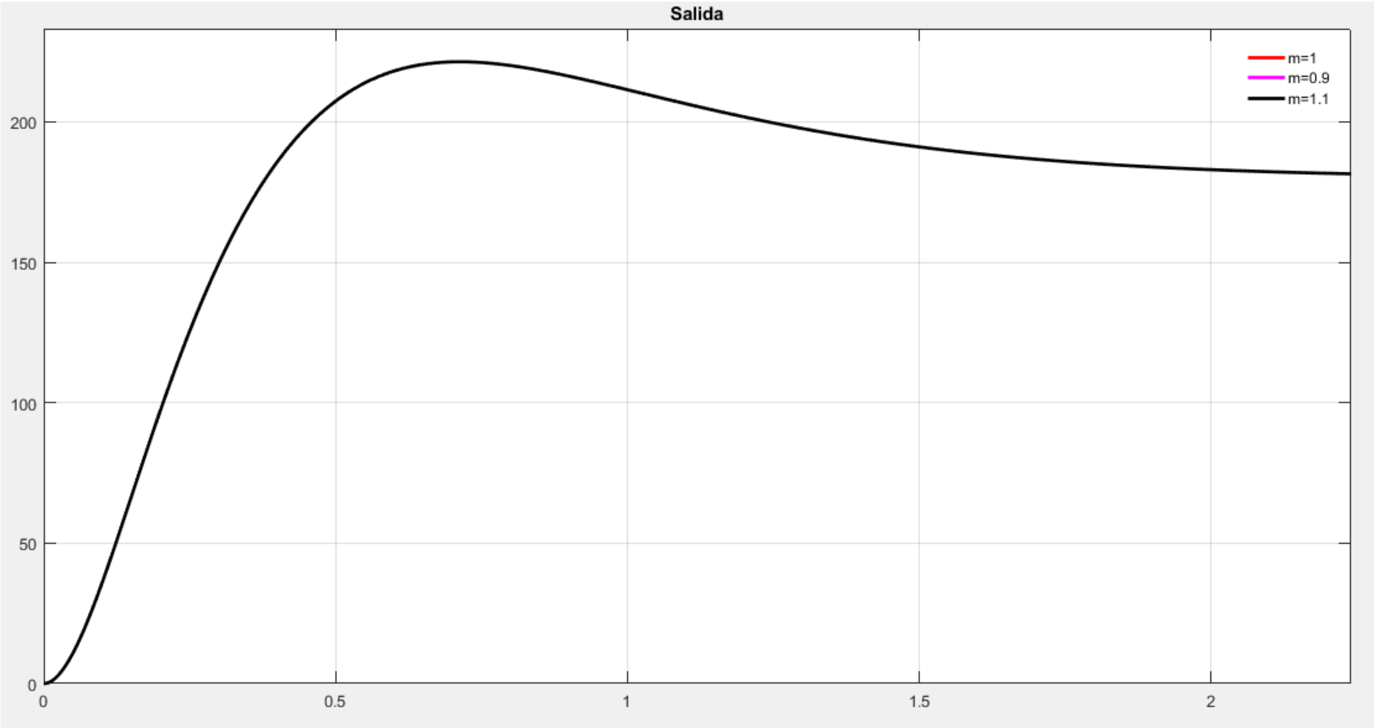


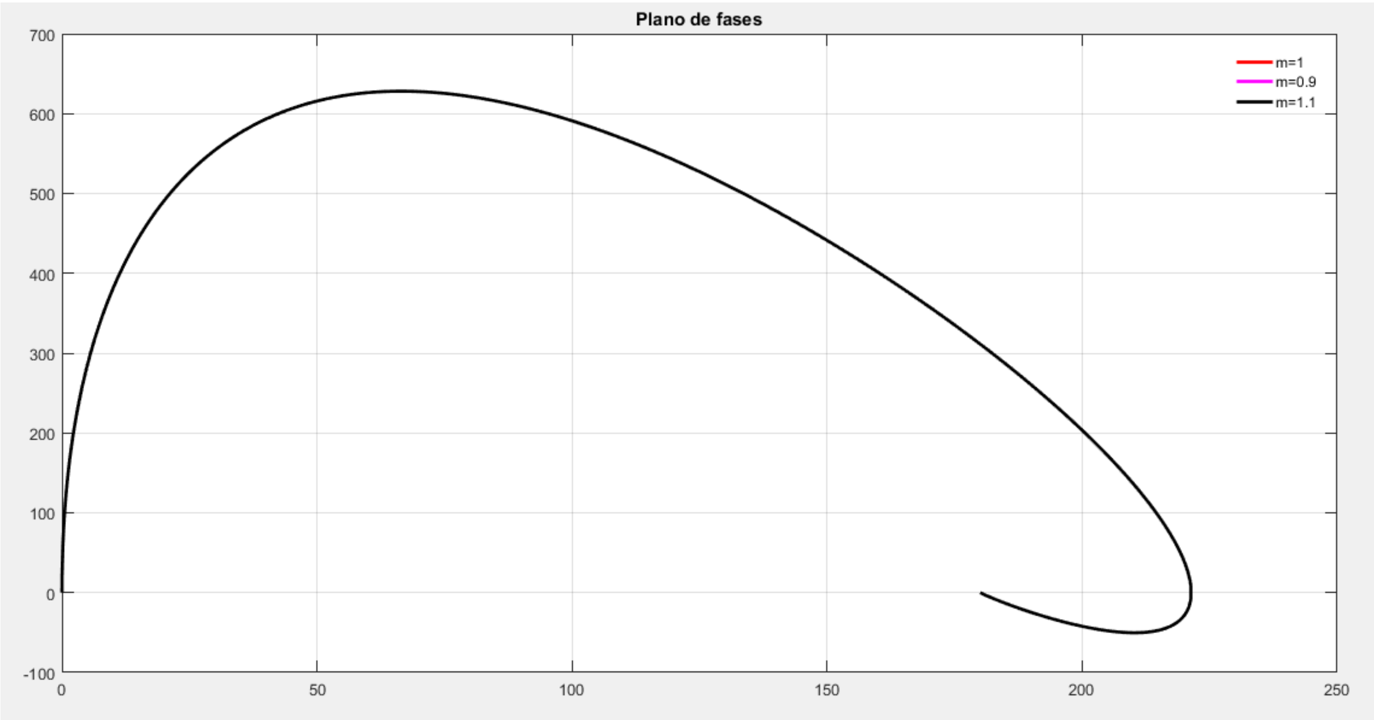
Se configura una tabla que contiene los valores de interés solicitados para la comparación entre cada una de las 3 simulaciones realizadas, de modo de poder analizar la robustez del sistema bajo los diferentes valores de masas a controlar por el sistema de péndulo simple.

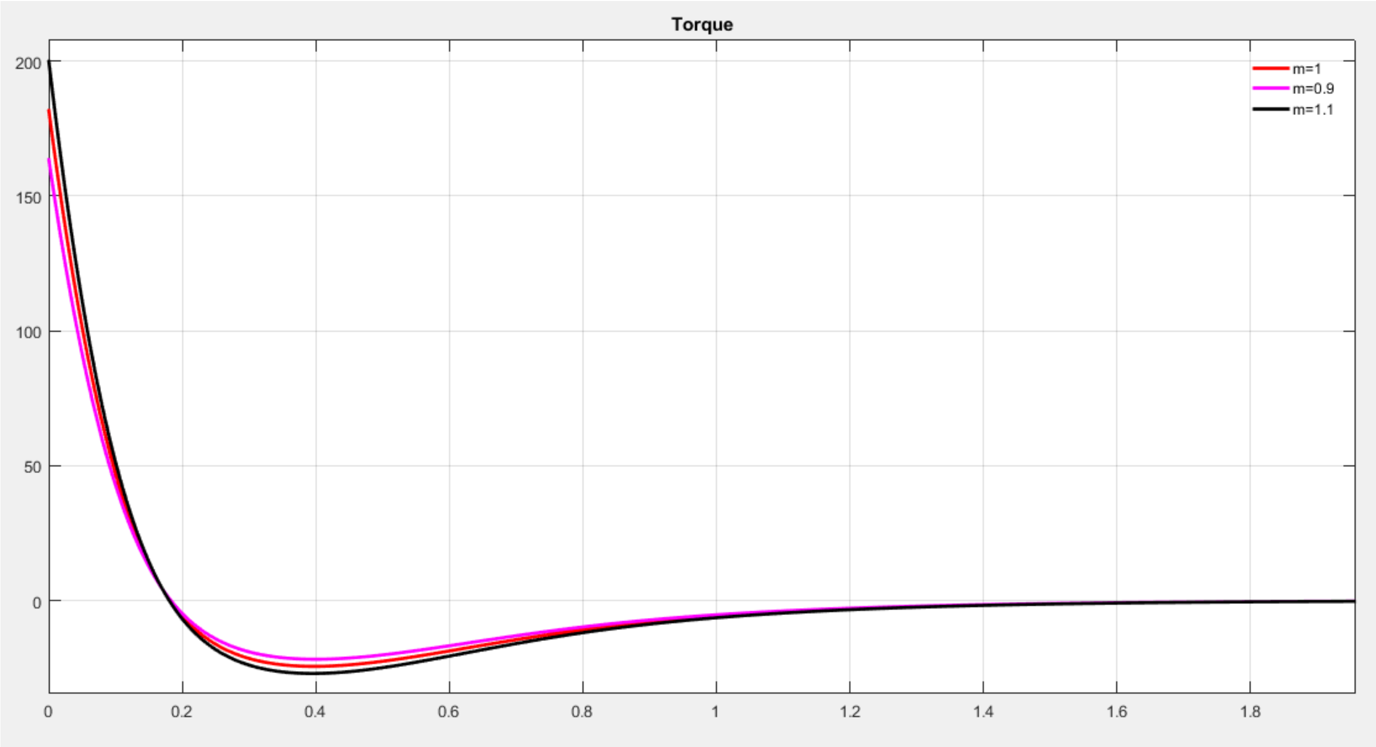
Resultados	**Masa = 0.9**	**Masa = 1**	**Masa = 1.1**
Y_{max} (Max valor salida)	221.38	221.38	221.38
S [%] (Sobrepaso)	22.9889%	22.9889%	22.9889%
E_f (Error final)	0	0	0
T_{ss} tiempo de establecimiento)	1.9296[s]	1.9296[s]	1.9296[s]
Y_{tss} (Salida al tiempo Tss)	183.6002	183.6002	183.6002
U_f (Torque final)	$-8.8644\ 10^{-13}$	$-8.8632\ 10^{-13}$	$-8.8620\ 10^{-13}$
Int_f (Acción integral final)	$-2.39\ 10^{-11}$	$-2.6625\ 10^{-11}$	$-2.9282\ 10^{-11}$

Puede apreciarse que en la tabla, los únicos valores que varían de forma apreciable son los de torque y acción integral. Por lo tanto se puede considerar que a pesar que se varió la masa del sistema, el controlador cumplió con su objetivo, por lo tanto se concluye diciendo que el controlador es Robusto.

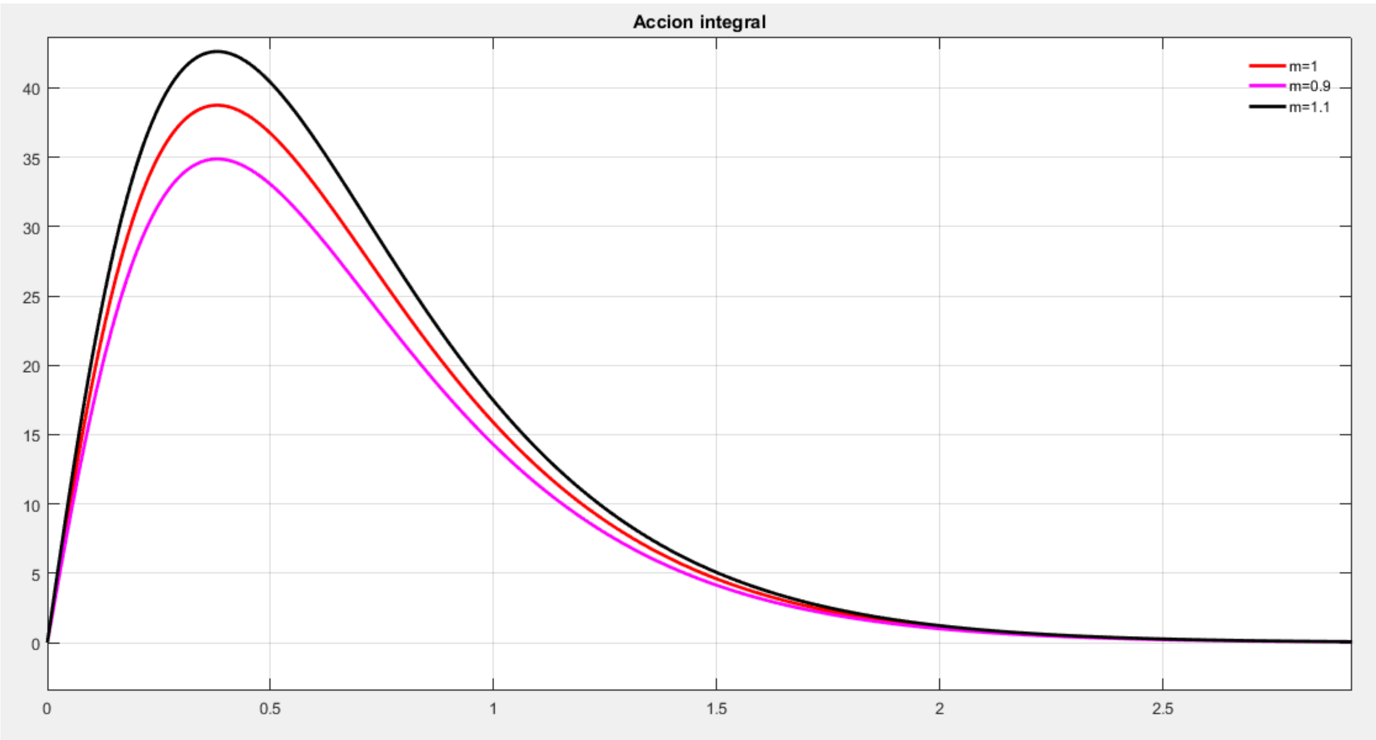
A continuación se grafican las tres respuestas superpuestas







Torque
m=1
m=0.9
m=1.1
200
150
100
50
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8



Accion integral
m=1
m=0.9
m=1.1
40
35
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5

Código utilizado

```
% close all; clear all; clc;
%Se declaran las variables

mnom = 1;% masa nominal
m=mnom;color='r';% corro programa con masa nominal
% m=mnom*0.90;color='m'; % correr de nuevo el código de simulación,
dibujo y analisis
% m=mnom*1.1;color='k'; % correr de nuevo el código de simulación,
dibujo y analisis

b = 0.4;
l = 1;
g = 10;
delta = 180;        %En grados


%linealizacion del sistema
Aa = [          0               1;
      (-g/l)*cosd(delta)   -b/(m/l^2)]

Bb = [    0;
    1/(m*l^2)]

Cc = [1 0]

%autovalores
autoval=eig(Aa)

%comparacion de resultados
[A,B,C,D]=linmod('pendulo_mod_tarea',delta*pi/180)
autoval_2 = eig(A)

%Se verifica la controlabilidad de la matriz A
if (length(A)==rank(ctrb(A,B)))

    disp('La matriz A es controlable')

else

    disp('La matriz A NO es controlable')

end


%matrices del sistema ampliado
AA=[[A;C] zeros(3,1)]
BB=[B;0]

%autovalores, estabilidad y controlabilidad del sistema ampliado
autoval_AA = eig(AA)
rank(ctrb(AA,BB))

%Se verifica la controlabilidad de la matriz ampliada AA
```

```
if (length(AA)==rank(ctrb(AA,BB)))

    disp('La matriz AA es controlable')

else

    disp('La matriz AA NO es controlable')

end

%diseño de un controlador
%asignacion de polos

p = -4;             %Polo triple
K = acker(AA,BB,[p p p])
k1 = K(1)
k2 = K(2)
k3 = K(3)
eig(AA-BB*K)        %Polos lazo cerrado
tscalc = 7.5/(-p)   %Tiempo de respuesta calculado

%simulacion PID
sim('pendulo_pid_tarea');
figure(1);
plot(tout,yout,color,'LineWidth',2); grid on; title('Salida');hold on;
legend('m=1','m=0.9','m=1.1');legend('boxoff');


%Plano de fase
figure(2);
plot(yout,velocidad,color,'LineWidth',2); grid on; title('Plano de
fases'); hold on;
legend('m=1','m=0.9','m=1.1');legend('boxoff');
%Torque total
figure(3);
plot(tout,torque,color,'LineWidth',2); grid on;title('Torque');hold
on;
legend('m=1','m=0.9','m=1.1');legend('boxoff');
%Accion integral
figure(4);
plot(tout,-accint,color,'LineWidth',2); grid on;title('Accion
integral');hold on;
legend('m=1','m=0.9','m=1.1');legend('boxoff');

disp('Maximo valor de salida:')
ymax=max(yout)

disp('Sobrepaso en %:')
S=(ymax-delta)/delta*100

disp('Error relativo:')
erel=(delta-yout)/delta

disp('Error final, debe ser cero:')
```

```
efinal=erel(end)

disp('Indice elementos con error relativo absoluto menor a 2:')
ind=find(abs(erel)>.02)

disp('Tiempo de establecimiento (ultimo valor del vector):')
tss=tout(ind(end))

disp('Salida al tiempo ts:')
yte=yout(ind(end))

disp('Torque final:')
uf=torque(end)

disp('Accion integral final:')
Intf=-accint(end)
```