

# Tarea 4: Profesor Laboret

Trabajo práctico

Autor: Villar, Federico Ignacio  
Profesor: Laboret, Sergio Oscar  
Fecha de entrega: 15 de junio de 2023  
Córdoba, Argentina

# Resumen

En el siguiente informe se detallará la resolución de la cuarta tarea propuesta por el profesor Laboret para la materia de Sistemas de Control II. En el documento se adjuntan imágenes, fragmentos de código y plots de respuestas.

Se pretende ser lo más claro posible, es por ello que se detallan varios procedimientos de cálculo y aplicación. Aunque, la mayoría de los resultados se obtuvieron de simulaciones con Matlab, por lo que no hay tantas expresiones analíticas.

# Índice de Contenidos

<b>1. Consigna</b>	<b>1</b>
<b>2. Obtención del modelo del sistema</b>	<b>4</b>
<b>3. Simulaciones</b>	<b>6</b>
3.1. Sistema con masa $m = 3$ . . . . .	6
3.2. Sistema con masa $m = 3.3$ . . . . .	8
3.3. Sistema con masa $m = 2.7$ . . . . .	10
<b>4. Códigos utilizados</b>	<b>13</b>
<b>5. Resultados obtenidos</b>	<b>18</b>
5.1. Valores numéricos . . . . .	18
5.2. Salida del sistema . . . . .	19
5.3. Plano de fase . . . . .	19
5.4. Torque . . . . .	20
5.5. Acción integral . . . . .	20
<b>6. Conclusiones</b>	<b>21</b>

# Índice de Figuras

1. Control por realimentación de estados y salidas con acción integral. . . . .	2
2. Esquema simulación péndulo con PID. . . . .	3
3. Salida del sistema con masa nominal. . . . .	6
4. Plano de fases para masa nominal. . . . .	7
5. Torque del sistema. . . . .	7
6. Acción integral para el sistema con masa nominal. . . . .	8
7. Salida para el sistema con masa mayor a la nominal. . . . .	8
8. Plano de fases para el sistema con masa mayor a la nominal. . . . .	9
9. Torque para el sistema con masa mayor a la nominal. . . . .	9
10. Acción integral para el sistema con masa mayor a la nominal. . . . .	10
11. Salida para el sistema con masa menor a la nominal. . . . .	10
12. Plano de fases para el sistema con masa menor a la nominal. . . . .	11
13. Torque para el sistema con masa menor a la nominal. . . . .	11
14. Acción integral para el sistema con masa menor a la masa nominal. . . . .	12
15. Comparación de las señales de salida. . . . .	19
16. Comparación de los planos de fase. . . . .	19
17. Comparación de los torques. . . . .	20
18. Comparación de las acciones integrales. . . . .	20

## Índice de Tablas

1. Comparación de respuestas. . . . . 18

## Índice de Códigos

1. Obtención de las matrices en variables de estado a partir de Simulink. . . . . 4
2. Código de simulación completo. . . . . 13

# 1. Consigna

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

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

Se desea que el péndulo se estabilice en el ángulo  $\delta$  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.

$$\dot{x} = f(x, u) \quad y = h(x)$$

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

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

$$y = Cx$$

En el punto

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

- Hallar los autovalores de  $A$  y determinar estabilidad por el método indirecto de Lyapunov.
- Comparar los resultados obtenidos con los de la linealización por Matlab y Simulink.

Se desea diseñar para el péndulo linealizado en el punto de operación dado un controlador con acción integral como se muestra:

$$u = k_1(\theta - \delta) + k_2\dot{\theta} + k_3\sigma$$

$$\dot{\sigma} = \theta - \sigma$$

Que no es otra cosa que un PID en la forma PI+D, ya que el torque vale:

$$T = k_1 e + k_3 \int_0^t e(\tau) d\tau + k_2 \dot{\delta}$$

Donde el error se definió en el sentido tradicional como  $e = \delta - \theta$

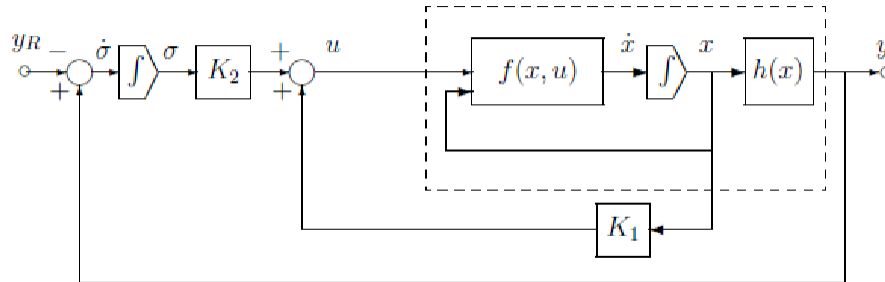


Figura 1: Control por realimentación de estados y salidas con acción integral.

- Encontrar las matrices del sistema ampliado.
- Verificar autovalores, estabilidad y controlabilidad del nuevo par.
- Diseñar por asignación de polos un controlador con la orden `acker()` de Matlab:

$$u = - \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \sigma \end{bmatrix}$$

Para ubicar un polo triple en  $p$  (dato), lo cual daría una respuesta sin sobrepaso (si el sistema fuera lineal y no tuviera ceros de lazo cerrado) y el tiempo 2% sería  $t_{ss} = \frac{7.5}{-p}$

- Simular el péndulo con PID como se muestra en la figura partiendo del origen con velocidad nula y referencia  $\delta$  (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 vs el calculado.
- En todos los casos ajustar las escalas para una visualización correcta.

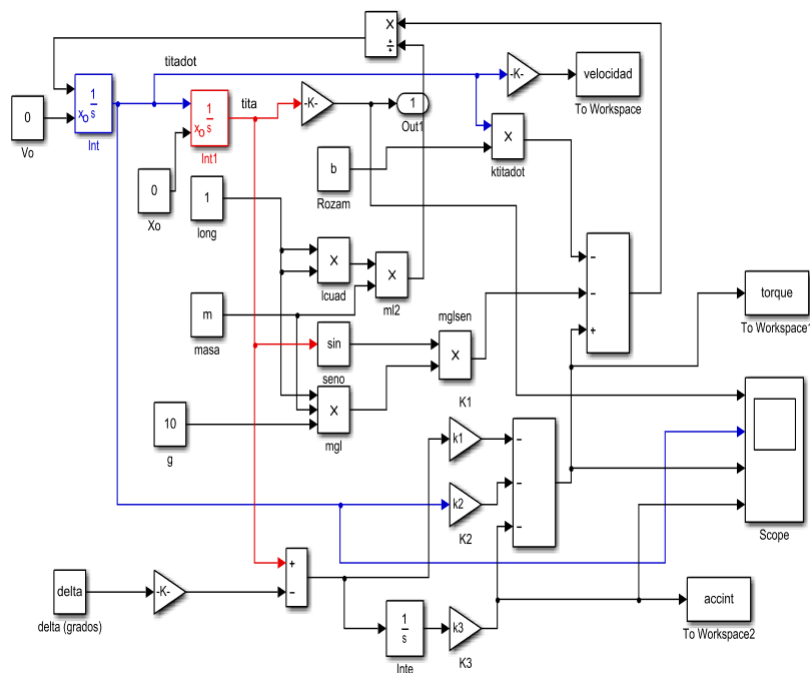


Figura 2: Esquema simulación péndulo con PID.

- 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 para los distintos valores de masa.

## 2. Obtención del modelo del sistema

Partiendo de la ecuación diferencial que modela el péndulo simple:

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

Y, con las variables de estado definidas en la consigna, se tendría lo siguiente:

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

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

$$y = e = x_1$$

Entonces,

$$\dot{x}_1 = \dot{\theta} = x_2$$

$$\dot{x}_2 = \ddot{\theta} = \frac{1}{ml^2}T - \frac{b}{ml^2}\dot{\theta} - \frac{g}{l}\sin(\theta) = \frac{1}{ml^2}u - \frac{b}{ml^2}x_2 - \frac{g}{l}\sin(x_1 + \delta)$$

Ahora, para el torque máximo:

$$u = T - T_f \Rightarrow u + T_f \Rightarrow T_f = mgl\sin(\delta)$$

Haciendo uso ahora de la jacobiana solicitada, se pueden obtener las siguientes matrices, que representan al sistema en variables de estado, tomando como referencia lo obtenido durante los últimos cálculos.

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

$$B = \begin{bmatrix} 0 & \frac{1}{ml^2} \end{bmatrix}$$

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

Evalutando lo obtenido anteriormente, se tiene:

$$T_f = 15\sqrt{2} \approx 21.21$$

$$A = \begin{bmatrix} 0 & 1 \\ 7.071 & -0.033 \end{bmatrix}; B = \begin{bmatrix} 0 & 0.333 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Con el código de Matlab provisto por la consigna, el cual es el siguiente:

Código 1: Obtención de las matrices en variables de estado a partir de Simulink.

```
1 m = 3
2 b = 0.1
3 delta = 135
4 l = 1;
5 G = 10;
6 [A,B,C,D] = linmod('pendulo_mod_tarea',delta*pi/180)
```



se obtienen los mismos valores que se calcularon previamente. Además, el torque estático que se solicitaba calcular coincide con el valor en estado estable de torque, por lo que se comprende que es el valor correcto.

Ahora, para comprobar estabilidad haciendo uso del método indirecto de Lyapunov, se obtienen los autovalores de la matriz A, los cuales son:

$$\lambda_1 = 2.6425; \lambda_2 = -2.6759$$

Con esto, se verifica que el sistema a lazo abierto es inestable, ya que se comprueba la existencia de un autovalor positivo, es decir, un polo con parte real positiva.

Ahora, ampliando las matrices de modo de contar con el integrador del controlador, se obtienen las siguientes:

$$A_{amp} = \begin{bmatrix} 0 & 1 & 0 \\ 7.071 & -0.033 & 0 \\ 1 & 0 & 0 \end{bmatrix}; B_{amp} = \begin{bmatrix} 0 \\ 0.333 \\ 0 \end{bmatrix}$$

Ahora, la matriz de controlabilidad para estas nuevas matrices ampliadas, es:

$$M_c = \begin{bmatrix} 0 & 0.333 & -0.011 \\ 0.333 & -0.011 & 2.357 \\ 0 & 0 & 0.333 \end{bmatrix}$$

Y los autovalores de la matriz a son:

- $\lambda_1 = 0$
- $\lambda_2 = -2.6759$
- $\lambda_3 = 2.6425$

Se corrobora que el sistema sigue siendo inestable, y controlable a su vez, ya que el rango de la matriz de controlabilidad coincide con el orden del sistema.

Con estas matrices ampliadas, ahora se diseña el siguiente controlador por asignación de polos mediante la fórmula de Ackerman:

$$K = [165.2132 \quad 35.9 \quad 192]$$

### 3. Simulaciones

El sistema se simulará en 3 situaciones, primero es con masa nominal, luego se aumenta o disminuye la masa en un décimo.

#### 3.1. Sistema con masa $m = 3$

Con la masa en su valor nominal, se obtiene la siguiente simulación.

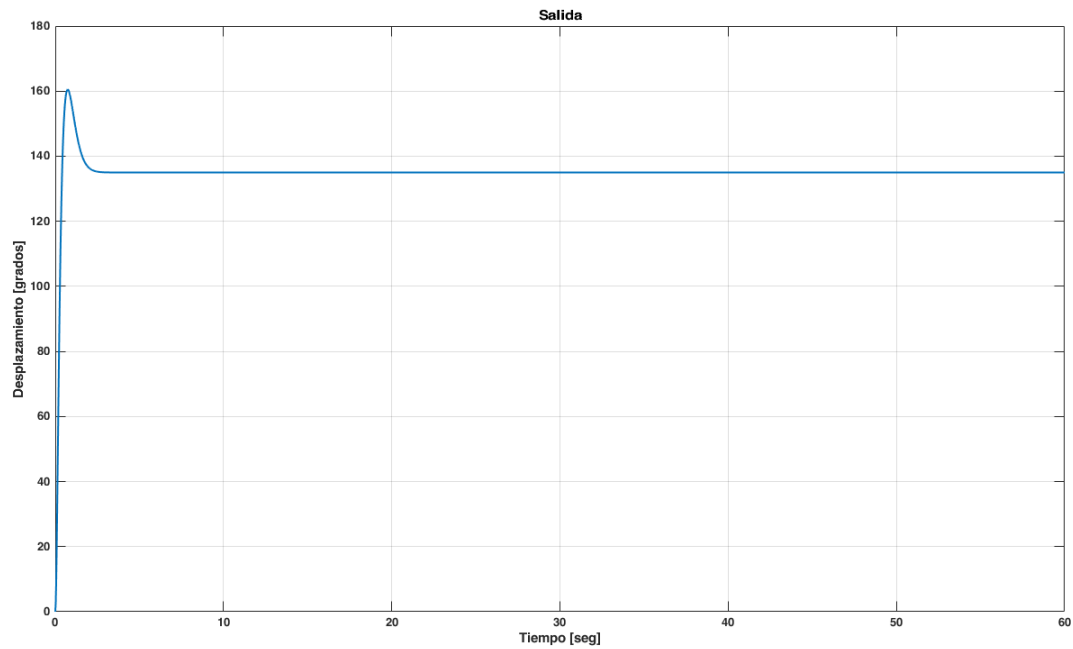


Figura 3: Salida del sistema con masa nominal.

Se puede verificar que el valor del tiempo de establecimiento calculado, de 1.875 *seg* es el que se visualiza en la gráfica.

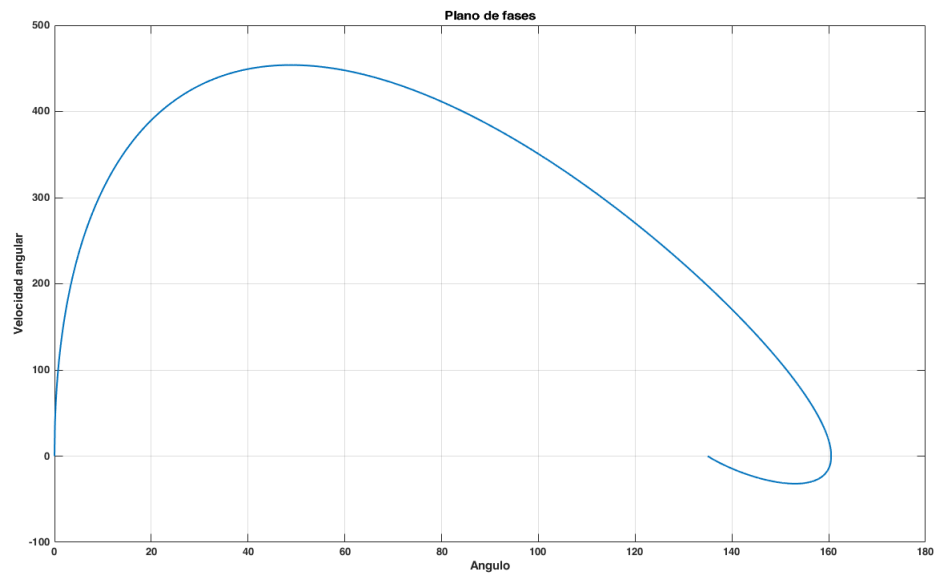


Figura 4: Plano de fases para masa nominal.

El plano de fases resultante es el que se aprecia en la figura anterior. Ahora, el torque resultante es el siguiente:

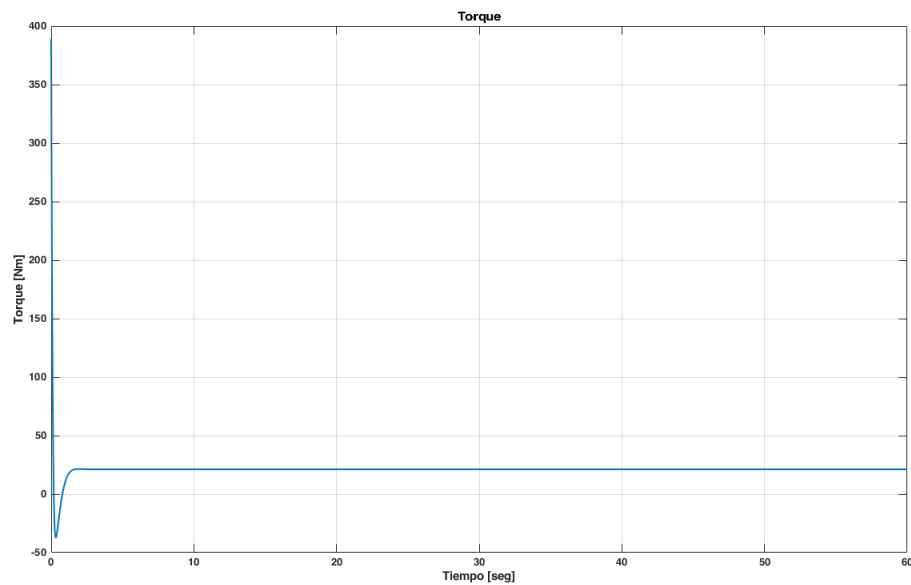


Figura 5: Torque del sistema.

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  $T_f$  realizado en la sección anterior.

Finalmente, la acción integral del controlador arroja la siguiente gráfica.

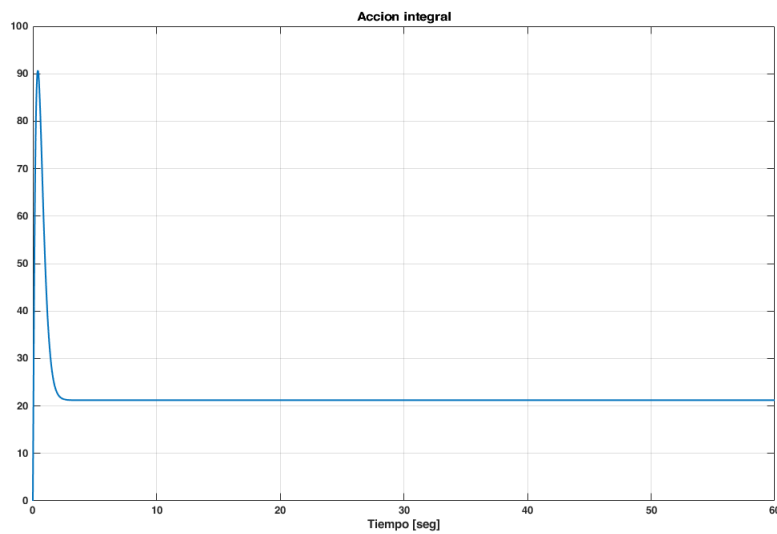


Figura 6: Acción integral para el sistema con masa nominal.

### 3.2. Sistema con masa $m = 3.3$

Al disminuir la masa, se pueden apreciar las siguientes respuestas.

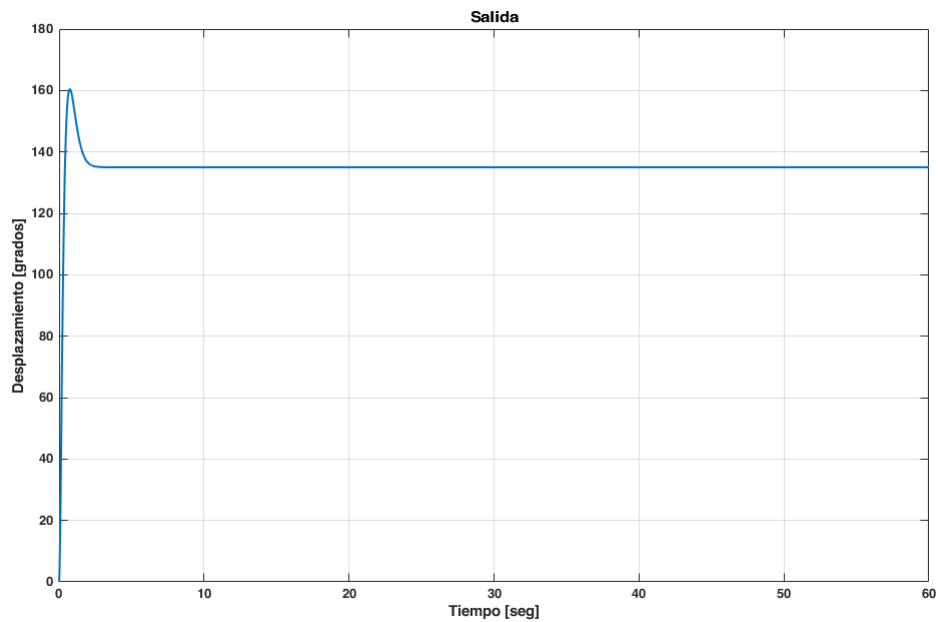


Figura 7: Salida para el sistema con masa mayor a la nominal.

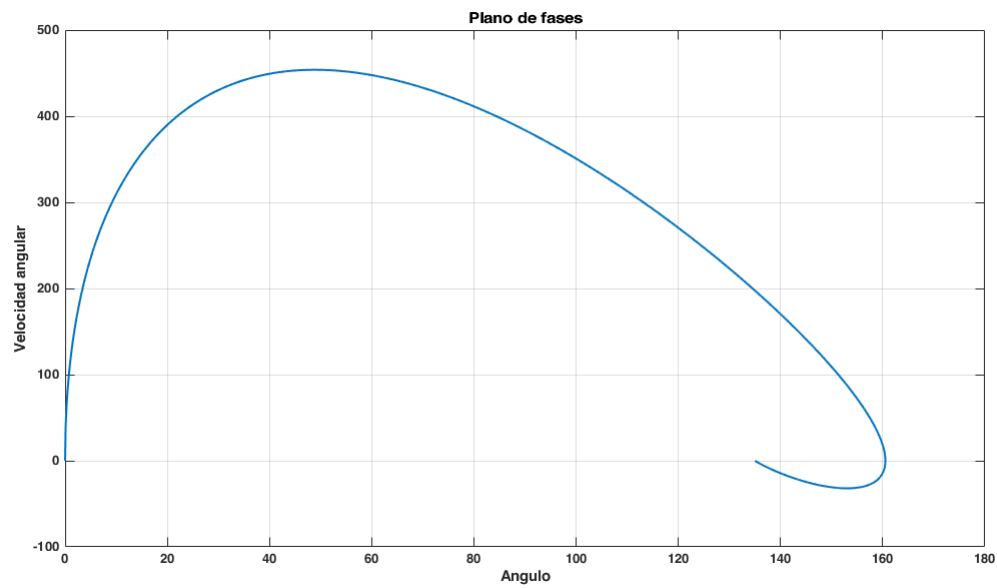


Figura 8: Plano de fases para el sistema con masa mayor a la nominal.

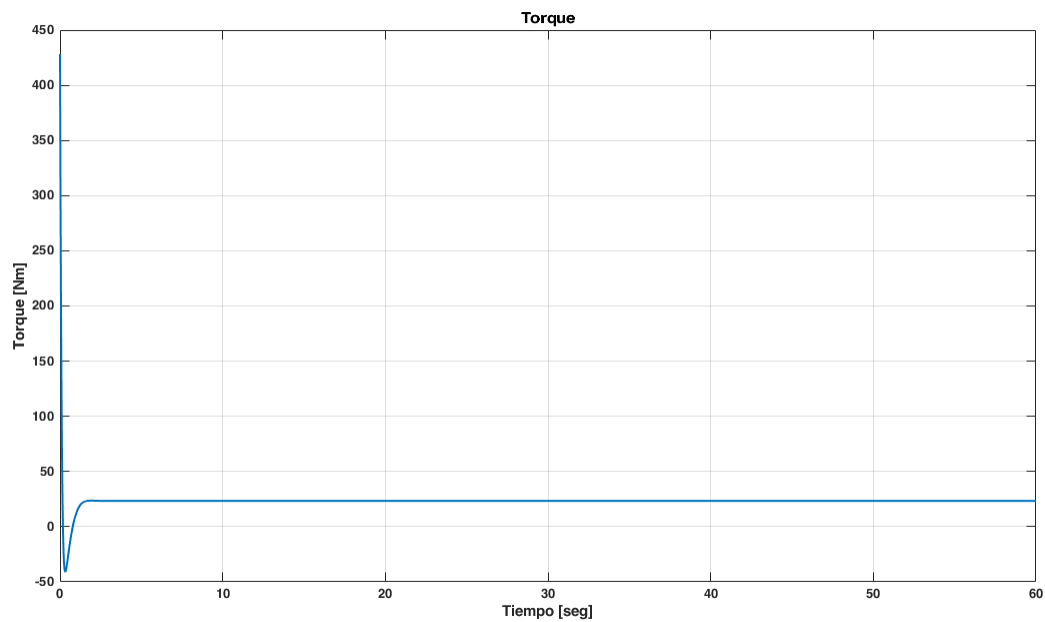


Figura 9: Torque para el sistema con masa mayor a la nominal.

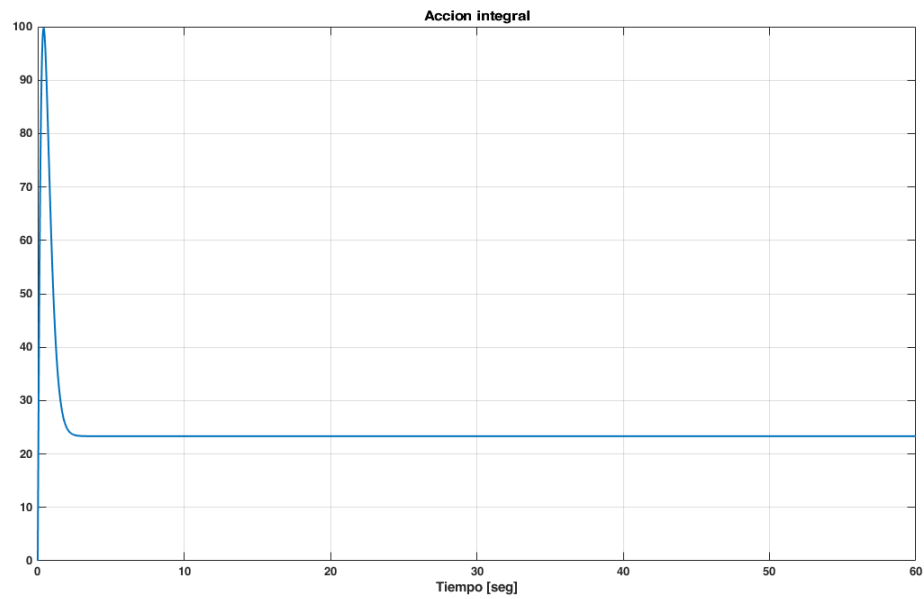


Figura 10: Acción integral para el sistema con masa mayor a la nominal.

### 3.3. Sistema con masa $m = 2.7$

Al aumentar la masa, se pueden apreciar las siguientes gráficas.

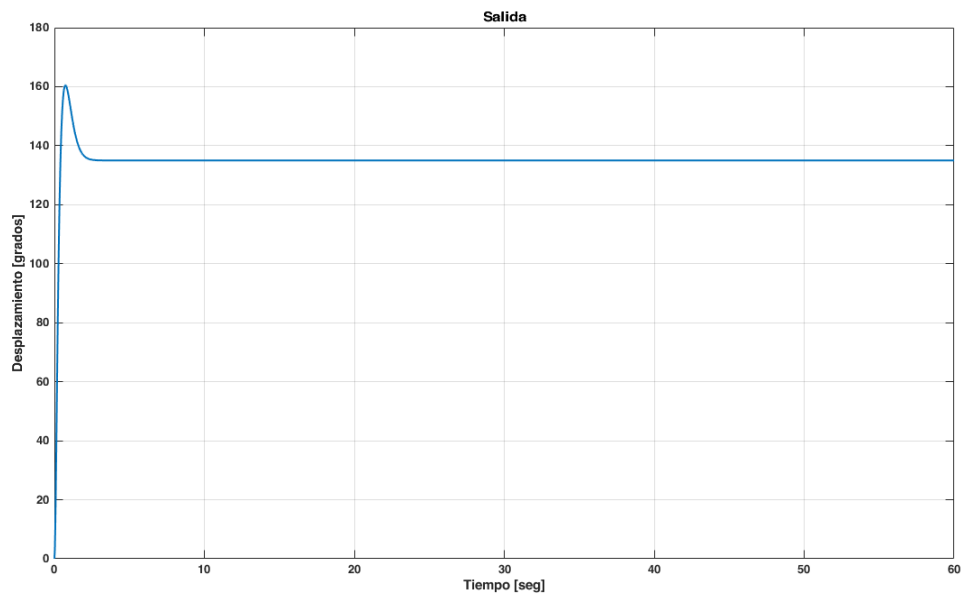


Figura 11: Salida para el sistema con masa menor a la nominal.

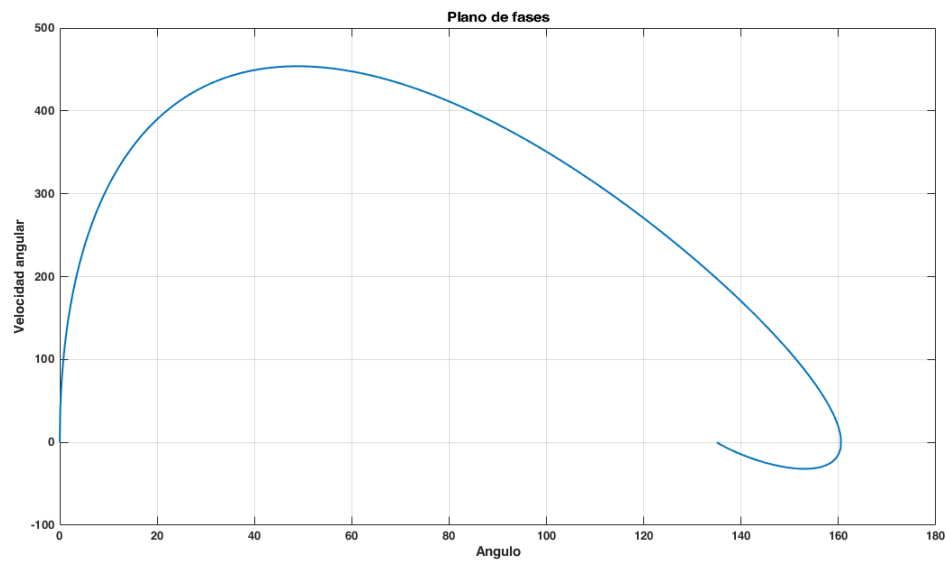


Figura 12: Plano de fases para el sistema con masa menor a la nominal.

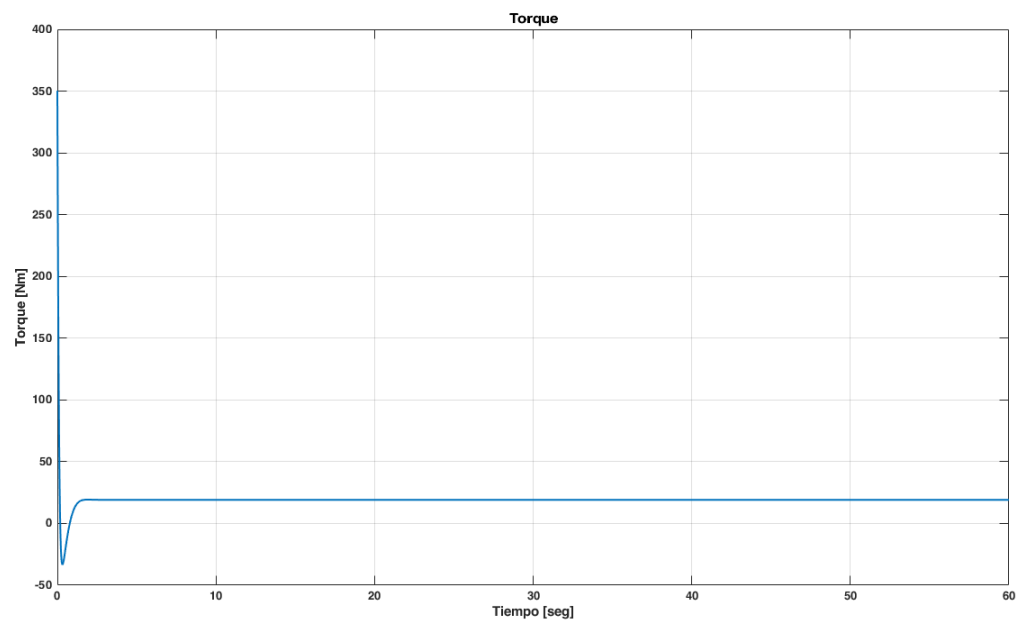


Figura 13: Torque para el sistema con masa menor a la nominal.

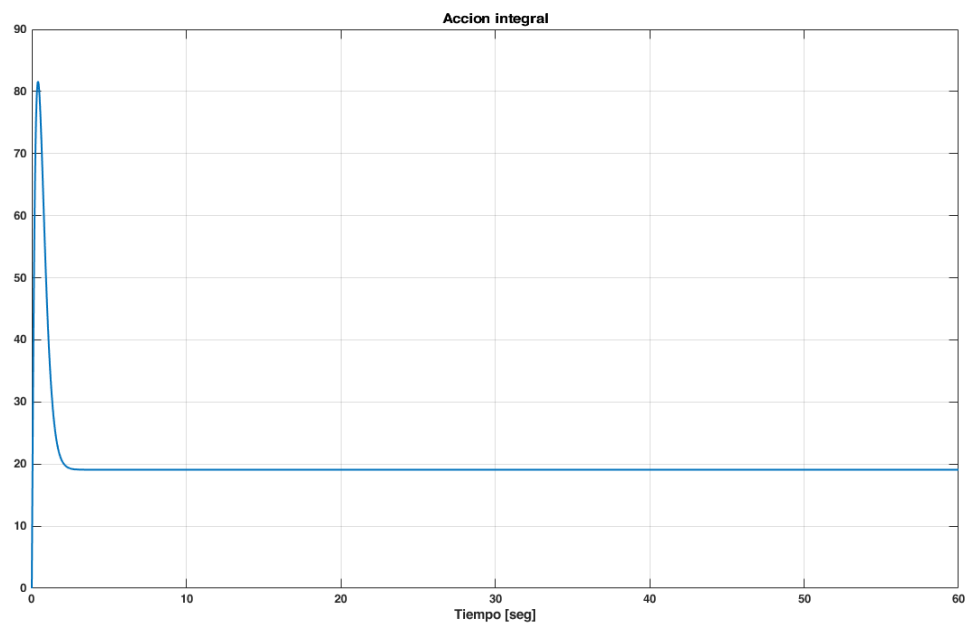


Figura 14: Acción integral para el sistema con masa menor a la masa nominal.



## 4. Códigos utilizados

A continuación se adjunta el código de Matlab completo que se utilizó para simular el comportamiento del sistema bajo distintas masas.

Código 2: Código de simulación completo.

```

1 %% Tarea 4: Profesor Laboret
2 %% Para la tarea se elige la primer fila de la tabla de los valores de masa,
3 %% longitud, coeficiente de rozamiento y ángulo de referencia en grados del sistema
4 %% péndulo simple.
5 clear all; close all; clc
6 %%
7 %% Se comienza definiendo los parámetros a utilizar.
8 m = 3
9 b = 0.1
10 delta = 135
11 l = 1;
12 G = 10;
13 %%
14 %% Se obtiene el modelo en espacio de estado para el sistema del péndulo linealizado
15 %% a partir del modelo de Simulink.
16 [A,B,C,D] = linmod('pendulo_mod_tarea',delta*pi/180)
17 %%
18 %% Los autovalores de la matriz A, que son los polos del sistema, son:
19 eig(A)
20 %%
21 %% El rango de la matriz de controlabilidad es el siguiente:
22 rank(ctrb(A,B))
23 %%
24 %% Se amplían las matrices, de modo de obtener la acción integral del controlador.
25 Aa = [[A;C] zeros(3,1)]
26 Ba = [B;0]
27 eig(Aa)
28 rank(ctrb(Aa,Ba))
29 %%
30 %% Polo triple en -4.
31 p = -4
32 K = acker(Aa,Ba,[p p p])
33 k1 = K(1)
34 k2 = K(2)
35 k3 = K(3)
36 eig(Aa-Ba*K)
37 tscal = 7.5/(-p)
38 sim('pendulo_pid_tarea')
39 ym1 = yout;
40 vel1 = velocidad;
41 tor1 = torque;
42 int1 = accint;
43 plot(tout,yout, 'LineWidth',1.5)
44 grid on

```

```

45 title('Salida')
46 xlabel('Tiempo [seg]');
47 ylabel('Desplazamiento [grados]')
48 plot(yout,velocidad, 'LineWidth',1.5)
49 grid
50 title('Plano de fases');
51 xlabel('Angulo');
52 ylabel('Velocidad angular')
53 plot(tout,torque, 'LineWidth',1.5)
54 grid on,
55 title('Torque');
56 xlabel('Tiempo [seg]');
57 ylabel('Torque [Nm]')
58 plot(tout,-accint, 'LineWidth',1.5)
59 grid
60 title('Accion integral');
61 xlabel('Tiempo [seg]');
62 ymax = max(yout)
63 S = (ymax-delta)/delta*100
64 erel = (delta-yout)/delta;
65 efinal = erel(end)
66 ind = find(abs(erel)>.02);
67 tss = tout(ind(end))
68 yte = yout(ind(end))
69 uf = torque(end)
70 Intf = -accint(end)
71 %%
72 % Para probar la robustez del controlador, se simulará con variaciones de 10 %
73 % en el valor de masa.
74 m = 3*1.1
75 b = 0.1
76 delta = 135
77 l = 1;
78 G = 10;
79 [A,B,C,D] = linmod('pendulo_mod_tarea',delta*pi/180)
80 eig(A)
81 rank(ctrb(A,B))
82 Aa = [[A;C] zeros(3,1)]
83 Ba = [B;0]
84 eig(Aa)
85 rank(ctrb(Aa,Ba))
86 p = -4
87 K = acker(Aa,Ba,[p p p])
88 k1 = K(1)
89 k2 = K(2)
90 k3 = K(3)
91 eig(Aa-Ba*K)
92 tscal = 7.5/(-p)
93 sim('pendulo_pid_tarea')
94 ym2 = yout;
95 vel2 = velocidad;

```

```

96 tor2 = torque;
97 int2 = accint;
98 plot(tout,yout, 'LineWidth',1.5)
99 grid on
100 title('Salida')
101 xlabel('Tiempo [seg]');
102 ylabel('Desplazamiento [grados]')
103 plot(yout,velocidad, 'LineWidth',1.5)
104 grid
105 title('Plano de fases');
106 xlabel('Angulo');
107 ylabel('Velocidad angular')
108 plot(tout,torque, 'LineWidth',1.5)
109 grid on,
110 title('Torque');
111 xlabel('Tiempo [seg]');
112 ylabel('Torque [Nm]')
113 plot(tout,-accint, 'LineWidth',1.5)
114 grid
115 title('Accion integral');
116 xlabel('Tiempo [seg]');
117 ymax = max(yout)
118 S = (ymax-delta)/delta*100
119 erel = (delta-yout)/delta;
120 efinal = erel(end)
121 ind = find(abs(erel)>.02);
122 tss = tout(ind(end))
123 yte = yout(ind(end))
124 uf = torque(end)
125 Intf = -accint(end)
126 m = 3*0.9
127 b = 0.1
128 delta = 135
129 l = 1;
130 G = 10;
131 [A,B,C,D] = linmod('pendulo_mod_tarea',delta*pi/180)
132 eig(A)
133 rank(ctrb(A,B))
134 Aa = [[A;C] zeros(3,1)]
135 Ba = [B;0]
136 eig(Aa)
137 rank(ctrb(Aa,Ba))
138 p = -4
139 K = acker(Aa,Ba,[p p p])
140 k1 = K(1)
141 k2 = K(2)
142 k3 = K(3)
143 eig(Aa-Ba*K)
144 tscal = 7.5/(-p)
145 sim('pendulo_pid_tarea')
146 ym3 = yout;

```

```

147 vel3 = velocidad;
148 tor3 = torque;
149 int3 = accint;
150 plot(tout,yout, 'LineWidth',1.5)
151 grid on
152 title('Salida')
153 xlabel('Tiempo [seg]');
154 ylabel('Desplazamiento [grados]')
155 plot(yout,velocidad, 'LineWidth',1.5)
156 grid
157 title('Plano de fases');
158 xlabel('Angulo');
159 ylabel('Velocidad angular')
160 plot(tout,torque, 'LineWidth',1.5)
161 grid on,
162 title('Torque');
163 xlabel('Tiempo [seg]');
164 ylabel('Torque [Nm]')
165 plot(tout,-accint, 'LineWidth',1.5)
166 grid
167 title('Accion integral');
168 xlabel('Tiempo [seg]');
169 ymax = max(yout)
170 S = (ymax-delta)/delta*100
171 erel = (delta-yout)/delta;
172 efinal = erel(end)
173 ind = find(abs(erel)>.02);
174 tss = tout(ind(end))
175 yte = yout(ind(end))
176 uf = torque(end)
177 Intf = -accint(end)
178 %%
179 % Se comparan las señales obtenidas:
180 plot(tout,ym1,"LineWidth",1.5)
181 hold on;
182 plot(tout,ym2,"LineWidth",1.5)
183 plot(tout,ym3,"LineWidth",1.5)
184 grid;
185 hold off;
186 title('Salida del sistema')
187 xlabel('Tiempo [seg]')
188 legend('m = 3','m = 3.3','m = 2.7')
189 plot(ym1,vel1,"LineWidth",1.5)
190 hold on;
191 plot(ym2,vel2,"LineWidth",1.5)
192 plot(ym2,vel2,"LineWidth",1.5)
193 grid;
194 hold off;
195 title('Planos de fases')
196 xlabel('Tiempo [seg]')
197 legend('m = 3','m = 3.3','m = 2.7')

```

```
198 plot(tout,tor1,"LineWidth",1.5)
199 hold on;
200 plot(tout,tor2,"LineWidth",1.5)
201 plot(tout,tor3,"LineWidth",1.5)
202 grid;
203 hold off;
204 title('Torques')
205 xlabel('Tiempo [seg]')
206 legend('m = 3','m = 3.3','m = 2.7')
207 plot(tout,int1,"LineWidth",1.5)
208 hold on;
209 plot(tout,int2,"LineWidth",1.5)
210 plot(tout,int3,"LineWidth",1.5)
211 grid;
212 hold off;
213 title('Acciones integrales')
214 xlabel('Tiempo [seg]')
215 legend('m = 3','m = 3.3','m = 2.7')
```

## 5. Resultados obtenidos

### 5.1. Valores numéricos

Primero que nada, se adjunta 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.

Tabla 1: Comparación de respuestas.

<b>Resultados</b>	$m = 3$	$m = 3.3$	$m = 2.7$
$S$	18.91	18.91	18.91
$e_{final}$	$2.88 \cdot 10^{-14}$	$2.95 \cdot 10^{-14}$	$2.91 \cdot 10^{-14}$
$t_{ss}$	1.80	1.80	1.80
$y_{t_{ss}}$	137.7	137.7	137.7
$u_f$	21.21	23.33	19.09
$I_f$	21.21	23.33	19.09

Los datos que aparecen en la tabla son respectivamente:

- $S$ : sobrepasamiento de la salida con respecto a la referencia (%).
- $e_{final}$ : error con respecto a la referencia en el último instante de muestreo.
- $t_{ss}$ : tiempo de establecimiento del sistema.
- $y_{t_{ss}}$ : salida del sistema al momento de establecerse.
- $u_f$ : torque final.
- $I_f$ : acción integral final.

Puede apreciarse que en la tabla, los únicos valores que varían de forma apreciable son los de torque y acción integral. Si bien el error final varía, el orden de magnitud hace que no sea apreciable en las gráficas. Se muestra a continuación entonces, las gráficas que comparan las tres distintas situaciones de masa.

## 5.2. Salida del sistema

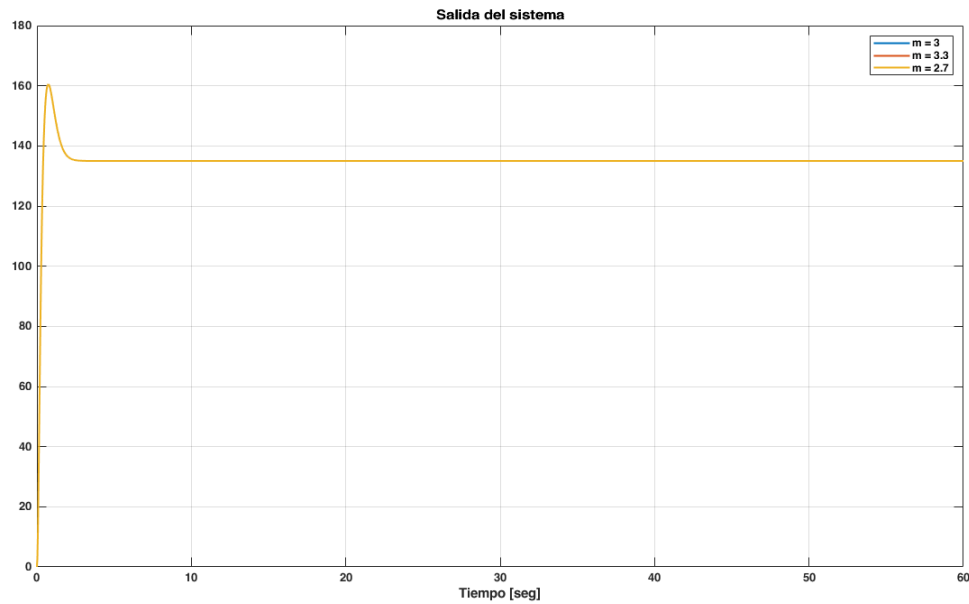


Figura 15: Comparación de las señales de salida.

## 5.3. Plano de fase

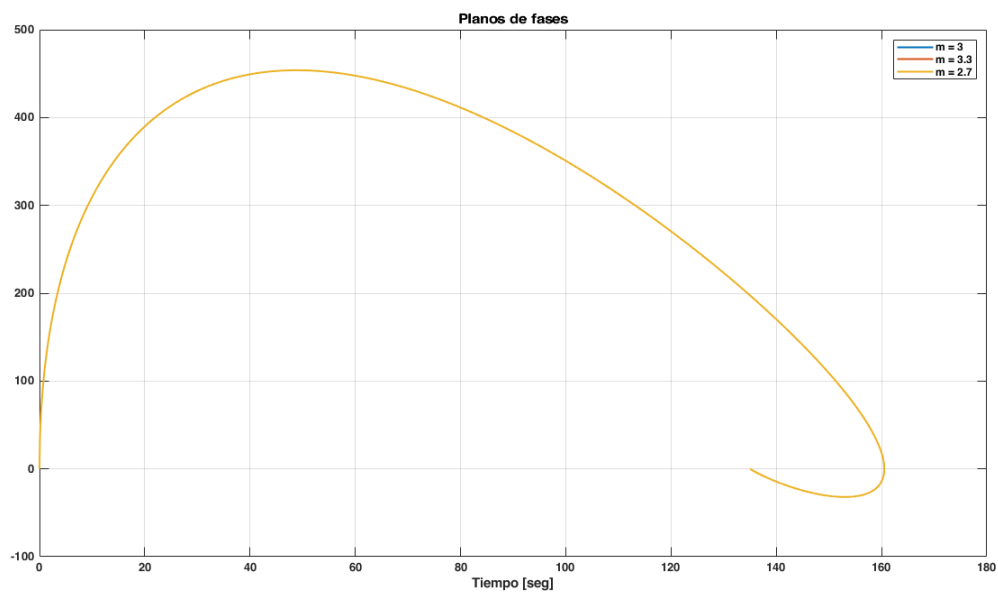


Figura 16: Comparación de los planos de fase.

## 5.4. Torque

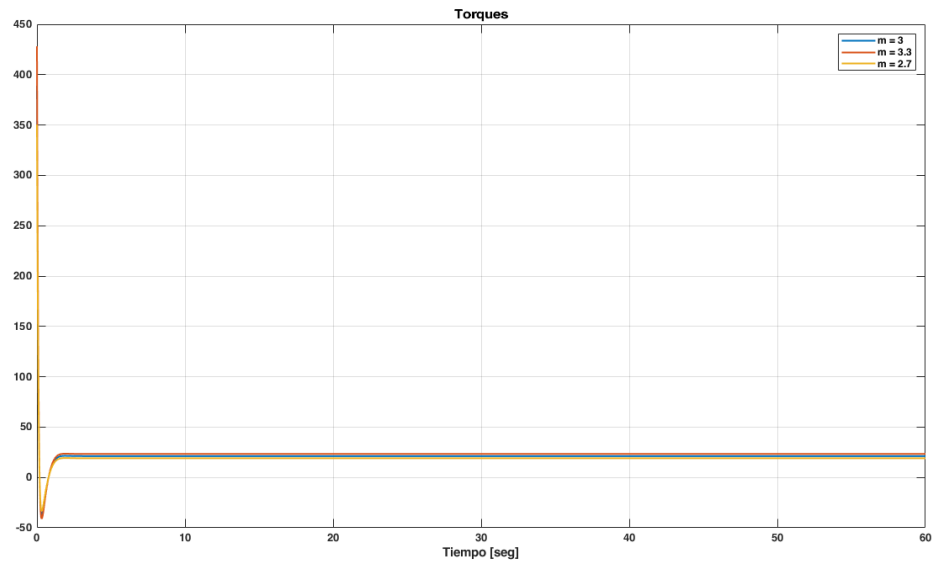


Figura 17: Comparación de los torques.

## 5.5. Acción integral

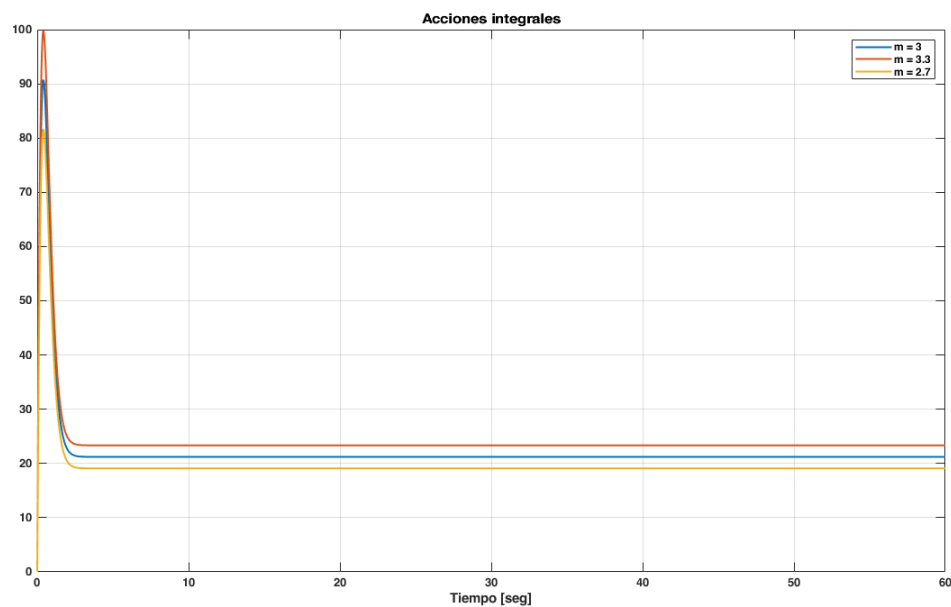


Figura 18: Comparación de las acciones integrales.



## 6. Conclusiones

Puede apreciarse en las gráficas obtenidas de las diferentes simulaciones, que los planos de fase y las salidas del sistema no varían. Esto habla de la robustez del sistema, así como de la forma en que fue planteado el controlador. Para ello se usó la asignación de polos mediante la fórmula de Ackerman, con los mismos polos en cada una de las diferentes respuestas. Esto significará que los polos a lazo cerrado del sistema serán siempre los mismos, por lo que el transitorio en el establecimiento del sistema será único, aún para distintas variaciones de la masa.

Puede comprobarse también, que este sistema diseñado, puede hacer referencia a un sistema en la vida real en donde se tiene un controlador que se reajusta a la masa del sistema a cargar. En este caso, este sistema estaría realizando un switch para diferentes controladores a medida que se varía el valor de la masa a cargar.

Finalmente, pudo obtenerse de forma analítica las diferentes representaciones en variables de estado para las matrices del sistema, así como las ampliadas, corroborando controlabilidad, y comprobando que los resultados calculados de forma manual son los mismos que se obtienen de la linealización que realiza el software. Con ello se puede terminar de relacionar los conceptos vistos en las diferentes clases de la cátedra.