



UNIVERSIDAD NACIONAL DE CÓRDOBA
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES
CÁTEDRA DE SISTEMAS DE CONTROL II

TAREA N°3:

“Diseño de Controlador digital para un Péndulo Simple”

Alumno: Monja, Ernesto Joaquín. - DNI: 43.873.728

Docente: Sergio Laboret.

Año 2025

Resolución

Los datos que fueron asignados para la resolución de esta tarea fueron los siguientes:

$m = 1$	$b = 0,4$	$l = 1$	$G = 10$	$\delta = 135$	$p(triple) = -4$
---------	-----------	---------	----------	----------------	------------------

TABLA 1: Datos para la resolución de la Tarea 3.

Con esto en mente se procede a encontrar el sistema dinámico en variables de estado, para lo cual primeramente necesitaremos la ecuación diferencial de un péndulo simple:

$$ml^2 \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} + mgl \sin(\theta) = T$$

De aquí se procede a tomar las variables de estado, las entradas y salidas (nótese que se ha desplazado el punto de equilibrio al origen tomando el error como salida) del sistema como:

$x_1 = \theta - \delta = e$	$x_2 = \frac{d\theta}{dt}$	$u = T$	$y = e$
-----------------------------	----------------------------	---------	---------

TABLA 2: Variables de estado, entrada y salida del sistema.

Luego, de la ecuación del sistema, despejamos la derivada de mayor orden como:

$$\begin{aligned} \frac{d^2\theta}{dt^2} &= -\frac{mgl \sin(\theta)}{ml^2} - \frac{b}{ml^2} \frac{d\theta}{dt} + \frac{1}{ml^2} T \\ \frac{d^2\theta}{dt^2} &= \left(-\frac{g}{l}\right) \sin(\theta) + \left(-\frac{b}{ml^2}\right) \frac{d\theta}{dt} + \left(\frac{1}{ml^2}\right) T \end{aligned}$$

Reemplazando por las definiciones dadas en la tabla 2, resulta que:

$$\frac{dx_2}{dt} = \left(-\frac{g}{l}\right) \sin(x_1 + \delta) + \left(-\frac{b}{ml^2}\right)x_2 + \left(\frac{1}{ml^2}\right)u$$

Ahora buscaremos obtener el torque estático necesario u_f para que el sistema tenga como punto de equilibrio el origen, es decir que $f(0, u_f) = 0$. Para ello sustituiremos las condiciones de equilibrio dadas para estas variables de estado, donde: $x_1 = 0$, $x_2 = 0$ y

$\frac{dx_2}{dt} = 0$ en la última ecuación en variables de estado, tal que:

$$0 = \left(-\frac{g}{l}\right) \sin(\delta) + \left(-\frac{b}{ml^2}\right)(0) + \left(\frac{1}{ml^2}\right) u_f$$

$$\left(\frac{1}{ml^2}\right) u_f = \left(\frac{g}{l}\right) \sin(\delta)$$

$$u_f = mgl \sin(\delta)$$

Sustituyendo en esta expresión los valores dados como datos en la Tabla 1, se tiene que:

$$u_f = (1)(10)(1) \sin(135^\circ) = 5\sqrt{2} \approx 7,071$$

Ahora debemos linealizar el sistema mediante el uso de la Jacobiana, para ello partimos de una representación genérica de un sistema en variables de estados, donde:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dx_1} & \frac{df_1}{dx_2} \\ \frac{df_2}{dx_1} & \frac{df_2}{dx_2} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ ml^2 \end{bmatrix} \cdot T$$

Donde se utilizó el siguiente punto para linealizar el sistema:

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

Resulta entonces que la representación en variables de estado de este sistema es la siguiente:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -g \cos(\delta) & -\frac{b}{ml^2} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ ml^2 \end{bmatrix} \cdot T$$

Y si reemplazamos por los valores dados como datos del sistema de la Tabla 1, resulta que este sistema es igual a:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-(10)\cos(135)}{(1)} & \frac{-(0,4)}{(1)(1)^2} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (1)(1)^2 \end{bmatrix} \cdot T$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 7,071 & -0,4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot T$$

$$[y] = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Luego se propone hallar los autovalores de A para poder determinar la estabilidad mediante el método indirecto de Lyapunov, con lo que planteamos que:

$$\det(\lambda I - A) = \det \left(\begin{bmatrix} \lambda & 1 \\ 7,071 & \lambda + 0,4 \end{bmatrix} \right) = \lambda(\lambda + 0,4) - 7,071$$

$$\lambda^2 + 0,4\lambda - 7,071 = 0$$

Esta última expresión se obtiene que los autovalores de la matriz A son: $\lambda_1 = 2,4666$ y $\lambda_2 = -2,8666$ con lo cual según el método indirecto de Lyapunov se tiene que el sistema es inestable ya que: $\lambda_1 > 0$.

Se procedió a verificar estos resultados obtenidos mediante un código en MATLAB¹ donde se obtuvieron los siguientes resultados:

```
A =
      0      1.0000
  7.0711  -0.4000

B =
      0
      1

C =
      1      0

D =
      0

ans =
      2.4667
     -2.8667

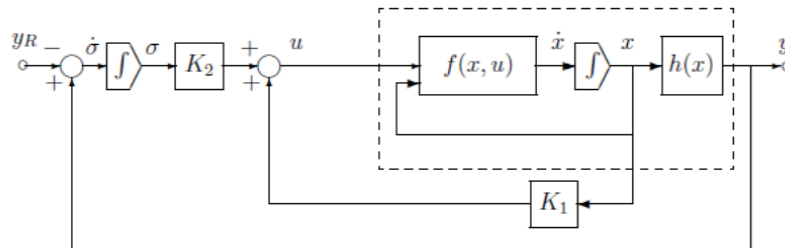
ans =
      2
```

Mediante estos comandos en MATLAB, se verifica el correcto cálculo de las matrices A, B, C y D, donde también coincide el cálculo de los autovalores de la matriz A y el rango

¹ Véase el Anexo 1, donde se muestra el código para resolución de esta tarea.

de la matriz de controlabilidad es igual al número de filas, y por lo tanto, resulta que el sistema es controlable.

El objetivo de este análisis, se debe a que se busca implementar un controlador PID en la forma PI+D el cual sigue el siguiente diagrama en bloques:



Para calcular tal controlador, primero era necesario ampliar el sistema y para ello, se calcularon las matrices ampliadas las cuales se definen como:

$$A_a = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} ; B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

Estas se obtuvieron mediante el código de MATLAB el cual arrojó que:

$$A_a = \begin{bmatrix} 0 & 1 & 0 \\ 7,071 & -0,4 & 0 \\ 1 & 0 & 0 \end{bmatrix} ; B_a = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Donde ahora, a este sistema de matrices ampliadas, se calcularon sus autovalores, la estabilidad por método indirecto de lyapunov y su controlabilidad y se obtuvo que sus autovalores están en: $\lambda_{1A} = 0$, $\lambda_{2A} = 2,4666$ y $\lambda_{3A} = -2,8666$ lo que implica que el sistema sigue siendo inestable sin embargo este es controlable tal como se muestra:

```
ans =  
  
      0  
 -2.8667  
  2.4667  
  
ans =  
  
      3
```

Se deduce que dado que el rango es 3, el sistema será controlable porque este rango coincide con el número de filas y esa es condición suficiente de controlabilidad.

Una vez obtenido el sistema ampliado, se procede a diseñar el controlador por asignación de polos siguiendo la fórmula de Ackerman. Para ello se utilizó el comando *acker()* de MATLAB el cual retorna las ganancias de realimentaciones de estado para ubicar los polos de lazo cerrado del sistema. Estas ganancias son iguales a:

$$K_1 = 55,0711$$

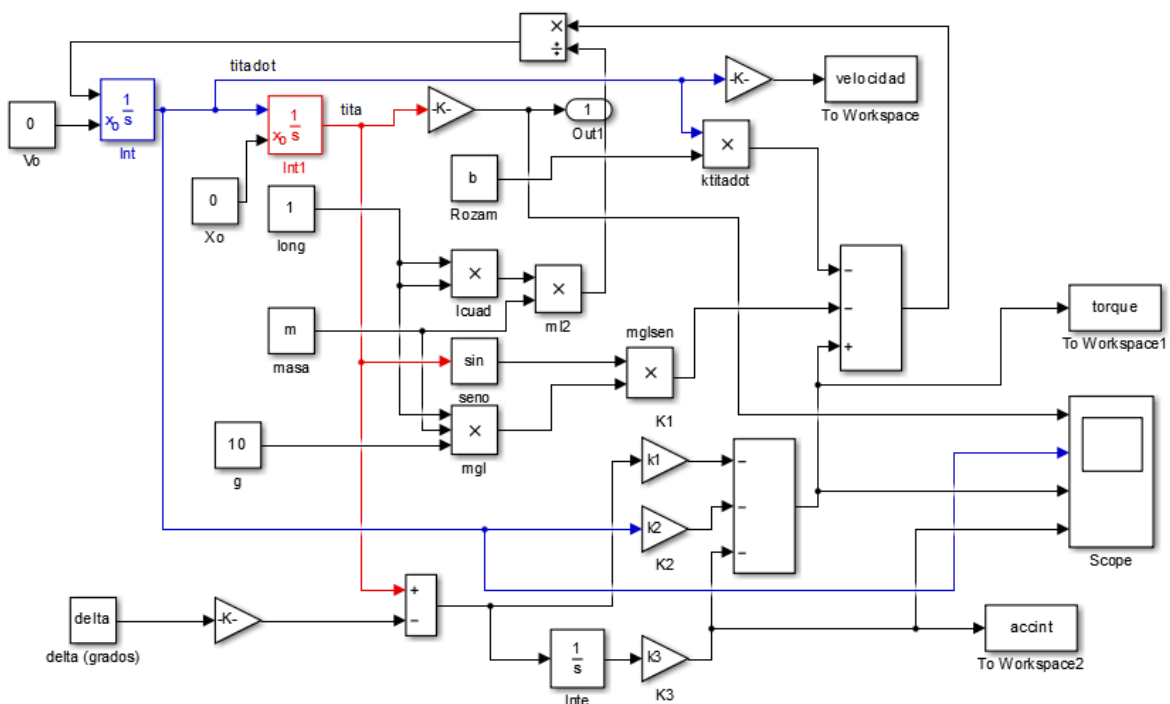
$$K_2 = 11,6$$

$$K_3 = 64$$

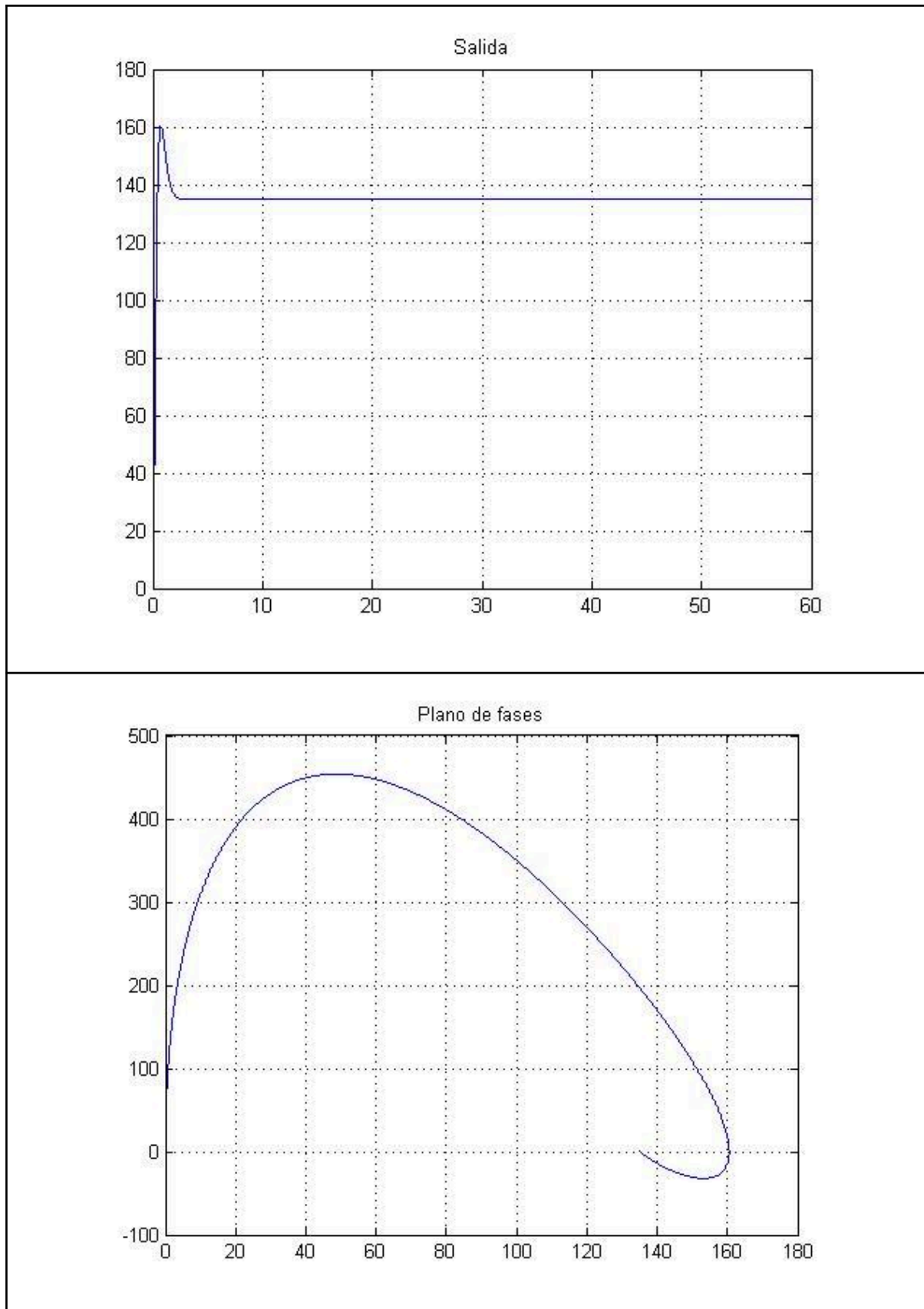
Estas ganancias nos dan un polo triple en -4 (Véase la Tabla 1) lo cual dará una respuesta sin sobrepaso (si el sistema fuera lineal y no tuviera ceros de lazo cerrado) y el tiempo 2% será igual a:

$$t_{ss} \approx \frac{7,5}{-p} = 1.8750 [s]$$

Una vez calculadas las ganancias de los controladores, se utilizó el siguiente sistema para realizar las siguientes simulaciones



A continuación se muestran las gráficas correspondientes a la salida, el plano de fases, el torque total y la acción integral del sistema:



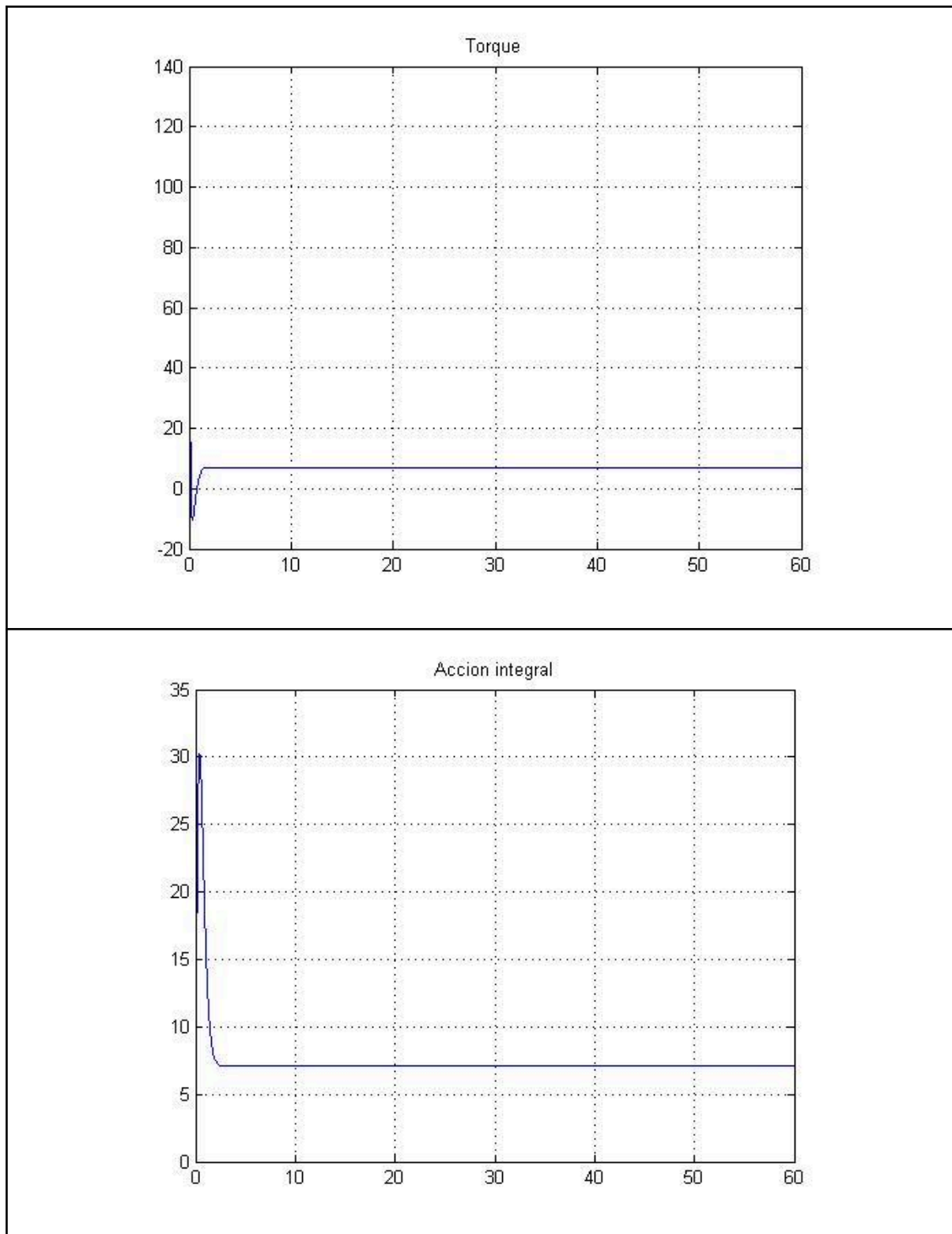


TABLA 3: Gráficas de interés del sistema.

Mirando a la primera gráfica, se observa que el sobrepasamiento máximo fue del $y_{max} = 160.5$ mientras que el valor de régimen de la salida fue de $y_{ss} = 135$, por lo tanto se tiene que:

$$S = \frac{y_{max} - y_{ss}}{y_{ss}} \times 100\% \Rightarrow S \simeq 18,88 [\%]$$

Donde el tiempo de establecimiento fue de $t_{2\%} = 1,805$ [s]. Se tiene también que en la gráfica del torque, se observa que este alcanza el valor de régimen igual a: $T_{ss} = u_f = 7,07$ lo cual coincide con el valor de régimen calculado inicialmente para este controlador.

Estos resultados reales del sistema son muy próximos a los obtenidos con la simulación del sistema, lo que nos indica que la linealización del controlador junto al cálculo numérico de estos valores fue correcta.

Por último, analizaremos la robustez del sistema, donde para ello se varió la masa del péndulo en $\pm 10\%$ y se analizó los nuevos valores de sobrepaso, tiempo de establecimiento y acción de control final. Las gráficas son las siguientes:

- Masa de 0,9:

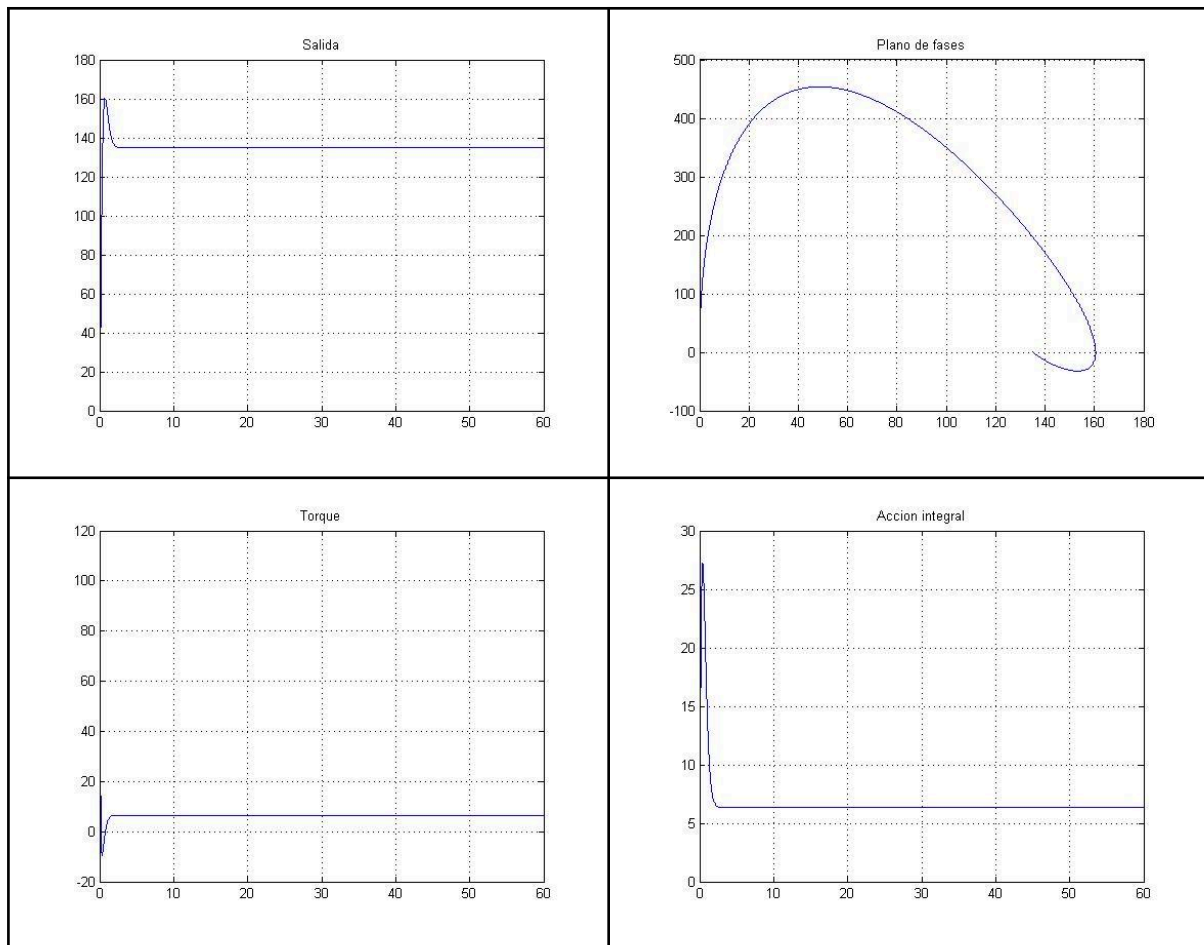


TABLA 4: Gráficas de interés del sistema con $m = 0,9$.

- Masa de 1,1:

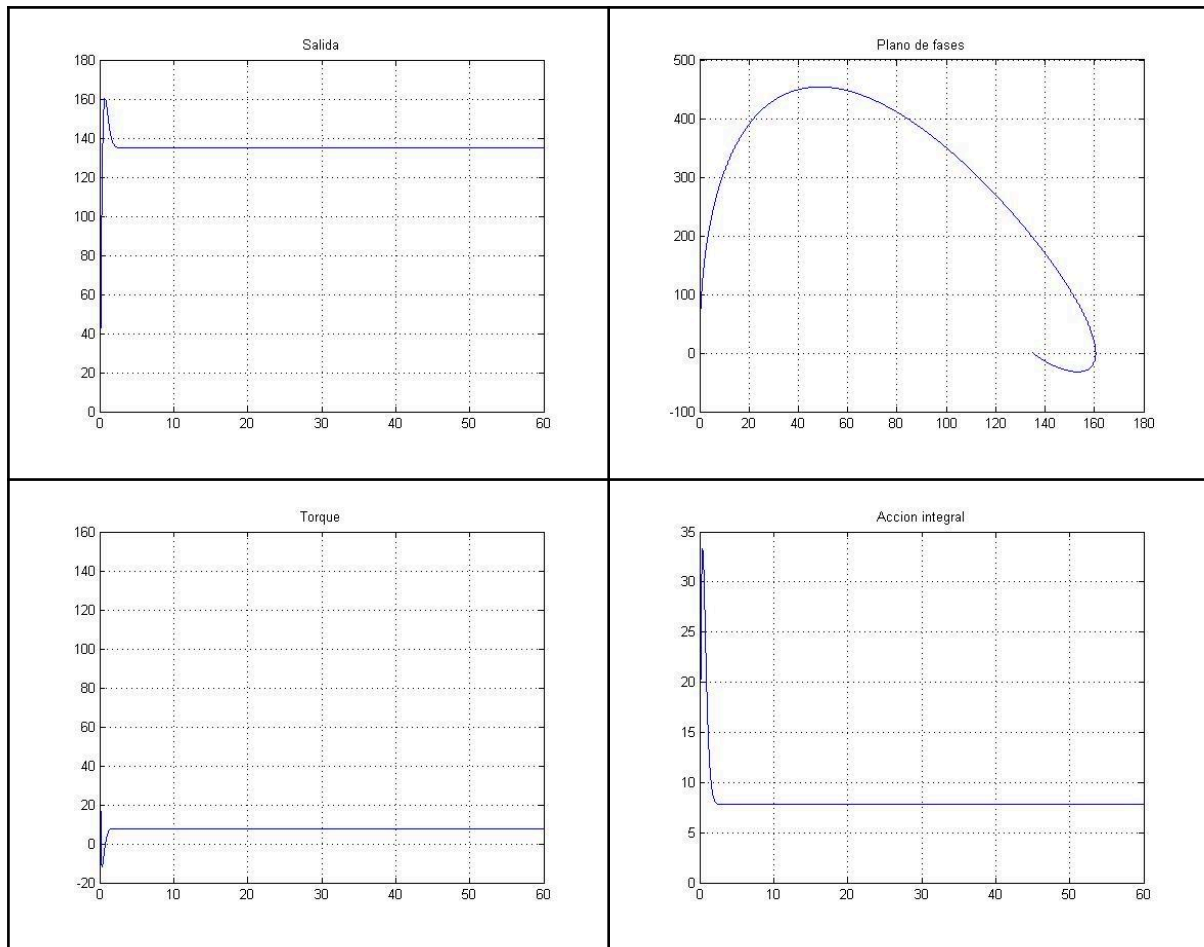


TABLA 4: Gráficas de interés del sistema con $m = 1, 1$.

Se tiene entonces que mediante la siguiente tabla comparativa, los resultados de la prueba de robustez son los siguientes:

	y_{max}	S	e_{final}	t_{ss}	y_{te}	u_f	I_f
$m = 0,9$	160.5228	18.9058	$2.9053e^{-14}$	1.8043	137.7007	6.3640	6.3640
$m = 1$	160.5228	18.9058	$2,8843e^{-14}$	1.8043	137.7007	7.0711	7.0711
$m = 1,1$	160.5228	18.9058	$2.7580e^{-14}$	1.8043	137.7007	7.7782	7.7782

TABLA 5: Comparativa de robustez del sistema con distintas masas.

Esta tabla verifica que el controlador es muy robusto, ya que este mantiene sus valores de régimen en un valor prácticamente igual al de la masa original de $m = 1$ si es que esta masa varía un 10%.

Anexo

El código utilizado para resolver esta tarea fue realizado en MATLAB y se anexa a continuación:

```
clear all; close all; clc;

% Datos dados por la tabla:
m = 1.1;           % Se cambia la masa de 0,9 a 1 y de 1 a 1,1.
b = 0.4;
delta = 135;
l = 1;
G = 10;

% Comparar los resultados obtenidos con los de la linealización por Matlab y Simulink
[A,B,C,D] = linmod('pendulo_mod_tarea',delta*pi/180)
eig(A)
rank(ctrb(A,B))
% Encontrar las matrices del sistema ampliado
Aa = [[A;C] zeros(3,1)]
Ba = [B;0]
eig(Aa)
rank(ctrb(Aa,Ba))
% Diseñar por asignación de polos un controlador con la orden acker() de matlab
p = -4
K = acker(Aa,Ba,[p p p])
k1 = K(1)
k2 = K(2)
k3 = K(3)
eig(Aa-Ba*K)           % polos lazo cerrado
tscal = 7.5/(-p)       % tiempo de respuesta calculado
% SIMULACION
sim('pendulo_PID_tarea')
figure(1), plot(tout,yout)
grid on, title('Salida')
figure(2), plot(yout,velocidad) % plano de fase
grid on, title('Plano de fases')
figure(3), plot(tout,torque) % torque total
grid on, title('Torque')
figure(4), plot(tout,-accint) % acción integral
grid on, title('Accion integral')
ymax = max(yout) % máximo valor de salida
S = (ymax-delta)/delta*100 % sobrepaso en %
erel = (delta-yout)/delta; % error relativo
efinal = erel(end) % error final, debe ser cero
ind = find(abs(erel)>.02); % índice elementos con error relativo absoluto menor a 2%
tss = tout(ind(end)) % tiempo de establecimiento (ultimo valor del vector)
yte = yout(ind(end)) % salida al tiempo ts
uf = torque(end) % torque final
Intf = -accint(end) % acción integral final
```