



Universidad Nacional de Córdoba

Facultad de Ciencias Exactas, Físicas y Naturales

Sistemas de control II

Tarea 2 - Profesor Laboret

Nombre	DNI
Diaz Mateo	41.265.543

Docentes Laboret Sergio

Córdoba, República Argentina
15 de mayo de 2024

Índice

1. Introduccion	2
2. Consignas	3
3. Controladores Digitales	4
3.1. Función de transferencia	4
3.2. Constantes	4
3.3. Lugar de raices	5
3.4. Controlador PI	5
3.5. Controlador PID	7
4. Simulación controladores	10
4.1. Controlador PI	10
4.1.1. Salida controlador	10
4.1.2. Señal error	10
4.1.3. Accion integral	11
4.1.4. Accion proporcional	11
4.2. Controlador PID	11
4.3. Comparación entre controladores	12
5. Conclusión	13

1. Introduccion

En el siguiente informe se desarrolla la resolucion de la segunda tarea propuesta por el profesor Sergio Laboret en la cátedra de Sistemas de Control II. Se detallan procedimientos y se adjuntan códigos utilizados para la resolución. El trabajo además se encuentra en un repositorio de Github personal donde se adjuntan todos los códigos en sus diversos formatos

2. Consignas

De la especificaciones de diseño dadas archivo adjunto (sobrepaso, tiempo de establecimiento 2 % y error de régimen ante escalón=0)

- Obtener los valores de $\zeta, \omega_0, \omega_d$:

$$\zeta = \frac{-\ln(\frac{S}{100})}{\sqrt{\pi^2 + \ln^2(\frac{S}{100})}}; t_R(2\%) = \frac{4}{\zeta\omega_0}; \omega_d = \omega_0\sqrt{1 - \zeta^2}; td = \frac{2\pi}{\omega_d}$$

- Calcular la cantidad de muestras por ciclo de la frecuencia amortiguada ω_d :

$$m = \frac{t_d}{T_m}$$

- Mediante la equivalencia de planos s y z determinar la ubicacion de los polos deseados en el plano z:

$$r = |z_{1,2}| = e^{-\zeta\omega_0 T_m} \Omega = \angle z_{1,2} = \pm\omega_d T_m$$

- Seleccionar y diseñar al menos dos controladores digitales (PI, PD, PID o adelanto) que cumplan (para los polos dominantes) las especificaciones dadas mediante SISOTOOL, en caso de que no se cumpla analizar por qué.
- La condición de error debe complicarse con exactitud.
- Simular mediante un diagrama de Simulink el control que mejores prestaciones haya obtenido según muestra en los diagramas adjunto, graficando la salida, el error, las acciones de control individuales P, I, D o la de adelanto (las que hubiera) y la acción de control total después del retentor de salida

3. Controladores Digitales

3.1. Función de transferencia

Partiendo de los parametros dados como consigna, obtenemos las siguientes funciones de transferencia en tiempo continuo y tiempo discreto

```
%FDT continuo
G = zpkm([-10],[-2 -1],[5])

%FDT discreto
Tm = 0.3;
Gd = c2d(G,Tm, 'zoh')
```

Obteniendo en tiempo continuo: $G(s) = \frac{5(s+10)}{(s+2)(s+1)}$

Y en tiempo discreto: $Gd(z) = \frac{2,6394(z+0,1076)}{(z-0,7408)(z-0,5488)}$

3.2. Constantes

A partir de las especificaciones dadas :

- $t_R = 4$
- $S = 5\%$

```
% Requerimientos
tR=4; %tiempo 2% error
S=5; %sobrepasamiento

%Obtencion parametros
pzita = -log(S/100)/sqrt(pi^2+(log(S/100))^2)
w0=4/(pzita*tR)
wd=w0*sqrt(1-pzita^2)
tD=2*pi/wd
r=exp(-pzita*w0*Tm)
omega = wd*Tm

%muestras por ciclo de la frecuencia amortiguada
m=tD/Tm
```

Calculamos las constantes:

- Factor de amortiguamiento: $\zeta = 0,6901$
- Frecuencia natural: $w_0 = 1,4491$
- Frecuencia amortiguada: $w_d = 2,7288$
- Cantidad de muestras por ciclo: $m = 19,9715$

Para ubicar los polos en el plano z :

```
[r1,ph1] = pol2cart(omega,r)

[r2,ph2] = pol2cart(-omega,r)

p1=r1+ph1*1i
p2=r2+ph2*1i
```

Obteniendo así un par de polos complejos conjugados :

$$p_{1,2} = 0,7045 \pm 0,2292i$$

3.3. Lugar de raíces

Podemos observar planteando el lugar de raíces de nuestro sistema y los polos basado en los requerimientos que los mismos se encuentran dentro del círculo unitario, por ende es estable, simplemente están desplazados.

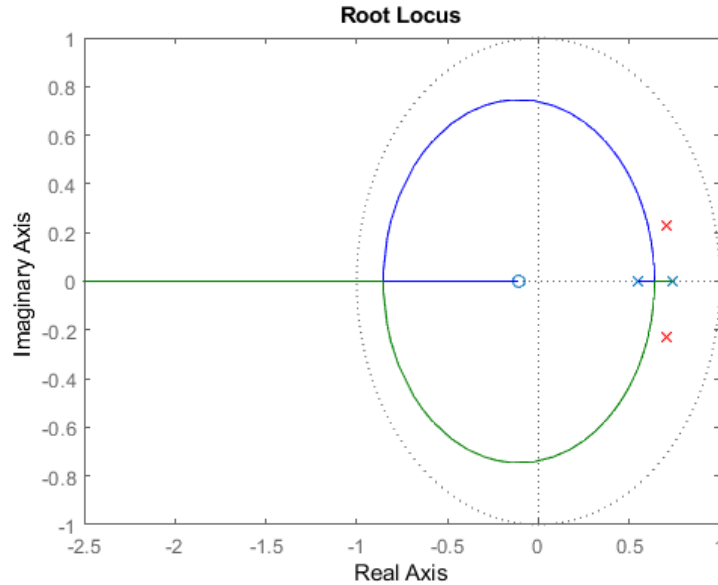


Figura 1: Lugar de raíces y polos deseados

Basándonos en que el sistema es de tipo 0, seguiremos el diseño en cumplir las condiciones de error, sobrepasamiento y tiempo de establecimiento. Al ser de tipo 0, no tenemos un polo integrador, el cual se lo añadimos para hacer que el error de estado estable tienda a 0. Plantearemos dos controladores con acción integral: PI y PID basándonos en el diseño de controladores digitales de Rlocus.

3.4. Controlador PI

Para este controlador fijamos en primer lugar un polo en 1 y luego tendremos un cero móvil para colocarlo de acuerdo con la respuesta esperada. Se plantea la necesidad de cancelar el polo dominante con el centro del controlador, $Z_c = 0,7408$. Utilizando la herramienta **SISOTOOL**, ejecutamos la función de transferencia en el mismo y elegimos sintonizar nuestro controlador basándonos en el método gráfico de lugar de raíces. En primer lugar nuestro sistema genera la siguiente figura

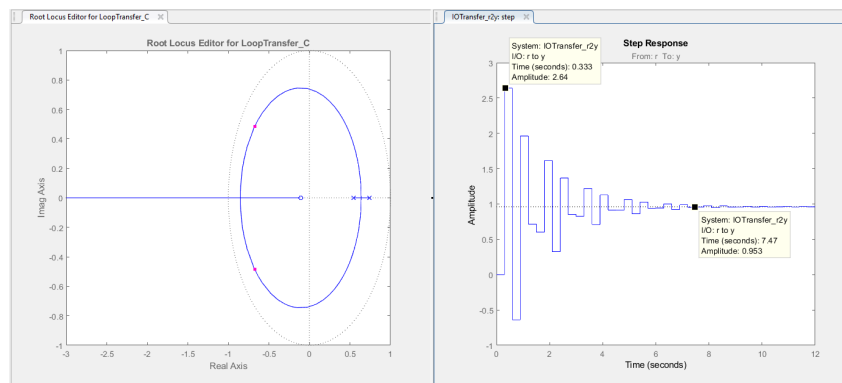


Figura 2: Lugar de raíces previo al controlador

Luego, planteando los dos requerimientos de tiempo de establecimiento y sobrepasamiento:

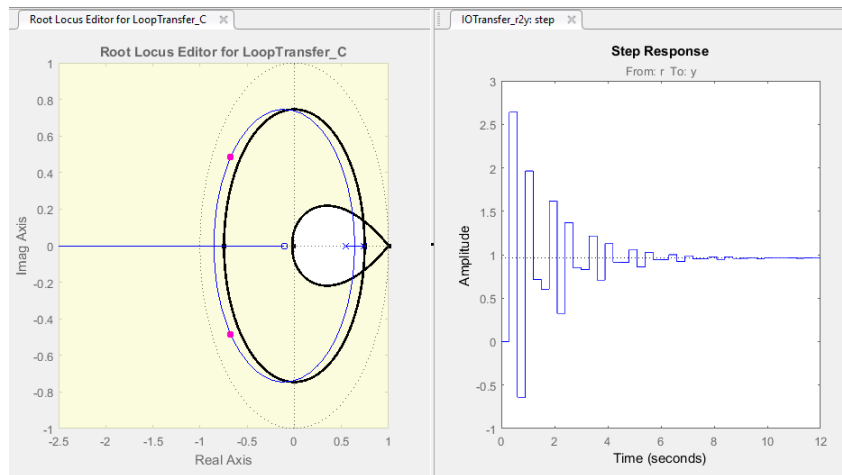


Figura 3: Lugar de raíces con requerimientos

Finalmente ajustando manualmente las ganancias buscamos el cruce entre ambas curvas donde coinciden nuestros puntos de diseño. Luego, planteando los dos requerimientos de tiempo de establecimiento y sobrepasamiento:

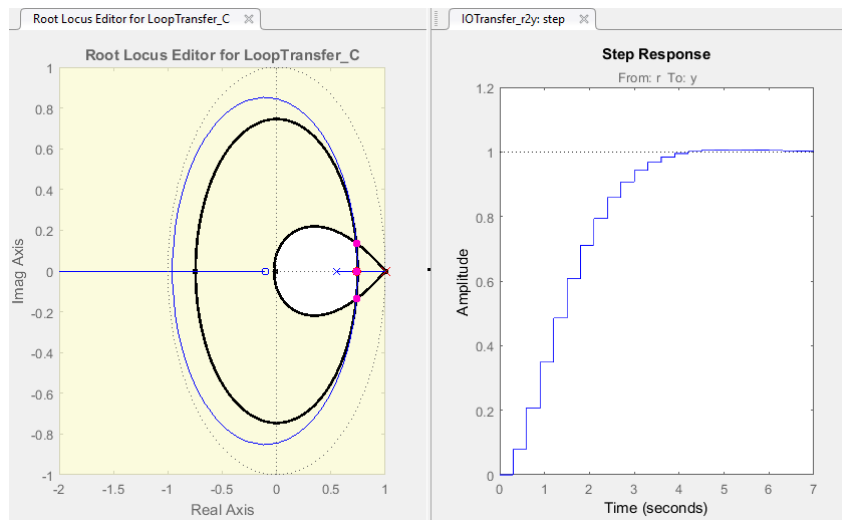


Figura 4: Lugar de raíces y polos deseados

Una vez finalizado el proceso, podemos exportar el controlador hacia nuestro espacio de trabajo de matlab.

Y nuestro sistema realimentado

Podemos plantear la verificación de nuestro sistema mediante el siguiente código

```
%verificacion
C %muestra el compensador importado de sisotool
F=feedback(C*Gd,1) % sistema de lazo cerrado
minreal(F)
poles=pole(F)
zeros=zero(F)
pzmap(F)
step(F) % respuesta al escalon
stepinfo(F)
```

La función de transferencia de el controlador C es la siguiente:

$$C = \frac{0,030466(z-0,7408)}{z-1}$$

La función de transferencia de nuestro sistema realimentado F es :

$$F = \frac{0,080412(z+0,1076)}{(z^2-1,468z+0,5575)}$$

Y en consola obteniendo lo siguiente:

```
poles =
    0.7342 + 0.1357i
    0.7342 - 0.1357i

zeros = -0.1076
```

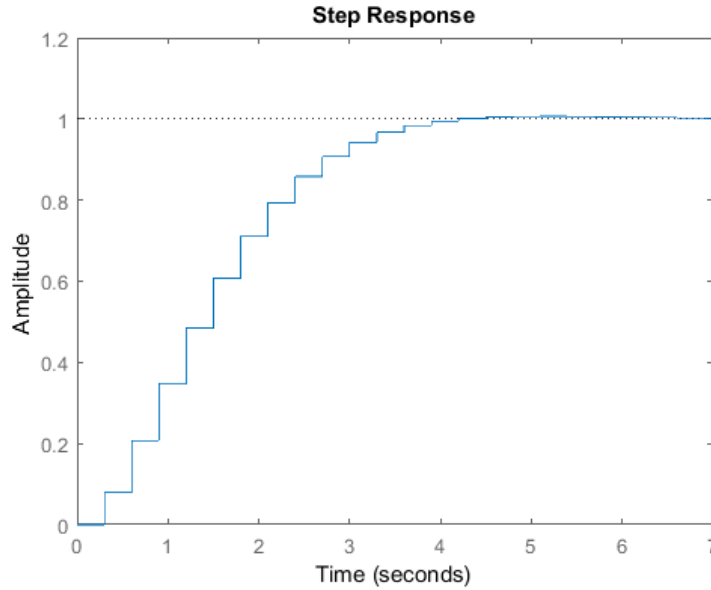


Figura 5: Respuesta al escalón sistema realimentado

3.5. Controlador PID

Para este controlador se añaden dos polos (uno en 0 y otro en 1) y luego similar al anterior se utiliza un cero para cancelar en este caso el polo no dominante y otro que aporta un determinado ángulo en el plano complejo respecto a los polos deseados. Teniendo en cuenta el controlador PID en forma simbólica

$$K_c \frac{(z - c_1)(z - c_2)}{z(z - 1)}$$

Partiendo de el sistema discreto:

$$Gd(z) = \frac{2,6394(z + 0,1076)}{(z - 0,7408)(z - 0,5488)}$$

y recordando que los polos a ubicar son: $p_{1,2} = 0,7045 \pm 0,2292i$

Haciendo que se cancele el polo no dominante , es decir: $c_1 = 0,5488$

Nuestro sistema para lazo abierto quedaría de la forma:

$$C(z)G(z) = \frac{2,6394(z + 0,1076)(z - c_2)}{(z - 0,7408)(z - 1)}$$

Faltaría ubicar el cero restante para hacer pasar el LR por el punto señalado y ajustar la ganancia

$$\theta_0 = \tan^{-1}\left(\frac{0,22}{0,74 + 0,10}\right) = 15,11$$

$$\theta_1 = \tan^{-1}\left(\frac{0,22}{0,7408}\right) = 17,19$$

$$\theta_2 = 90$$

$$\theta_3 = 180 - \tan^{-1}\left(\frac{0,22}{0,3145}\right) = 143,9$$

$$\theta_3 = 180 - \tan^{-1}\left(\frac{0,22}{0,3145}\right) = 143,9$$

$$\theta = \theta_0 - \theta_1 - \theta_2 - \theta_3 = 235$$

El ángulo que debe proporcionar el cero es de 55

La distancia de queda calculada como:

$$d = \frac{0,22932}{\tan(55)} = 0,1476$$

Obteniendo así el polo c2.

$$c2 = 0,5932$$

Ajustando la ganancia con la herramienta de SISOTOOL obtenemos el siguiente controlador

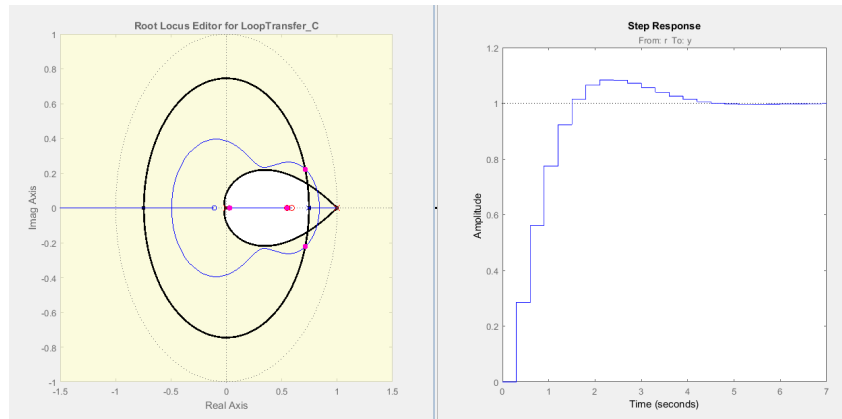


Figura 6: Rlocus Controlador PID

Podemos plantear la verificación de nuestro sistema mediante el siguiente código

```
%verificacion pid
C %muestra el compensador importado de sisotool
FPID=minreal(feedback(C*Gd,1)) % sistema de lazo cerrado
poles=pole(FPID)
zeros=zero(FPID)
pzmap(FPID)
step(FPID) % respuesta al escalon
stepinfo(FPID)
```

La función de transferencia de el controlador C es la siguiente:

$$C = \frac{0,1078(z-0,5932)(z-0,5488)}{(z(z-1))}$$

La función de transferencia de nuestro sistema realimentado F es :

$$F = \frac{0,28453(z-0,5932)(z+0,1076)}{(z-0,03266)(z^2-1,424z+0,5562)}$$

Y en consola obteniendo lo siguiente:

```
poles =
    0.0327 + 0.0000i
    0.7118 + 0.2225i
    0.7118 - 0.2225i
```

```
zeros = 0.5932  
       -0.1076
```

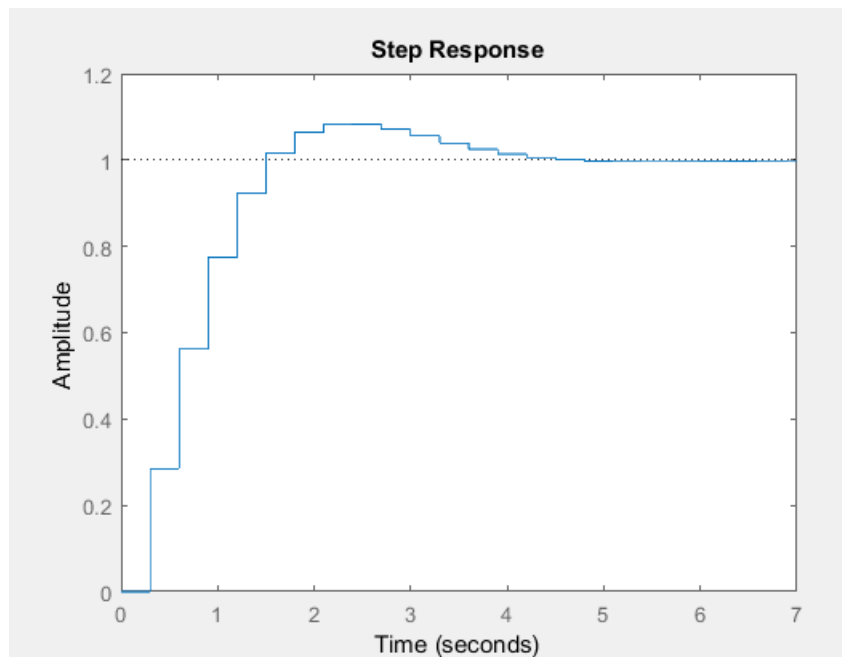


Figura 7: Respuesta al escalón sistema realimentado PID

Nuestro mapa de polos y ceros será el siguiente:

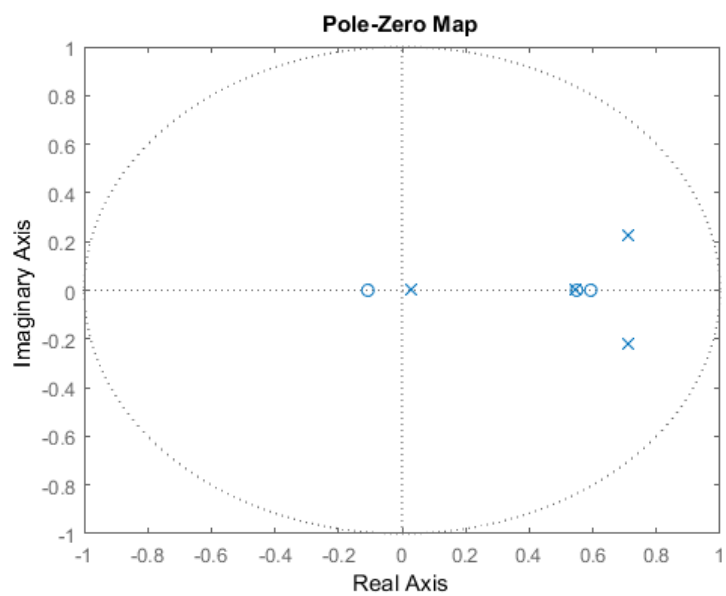


Figura 8: Polos y ceros PID

4. Simulación controladores

Se probaran los controladores desarrollados en **SIMULINK**

4.1. Controlador PI

4.1.1. Salida controlador

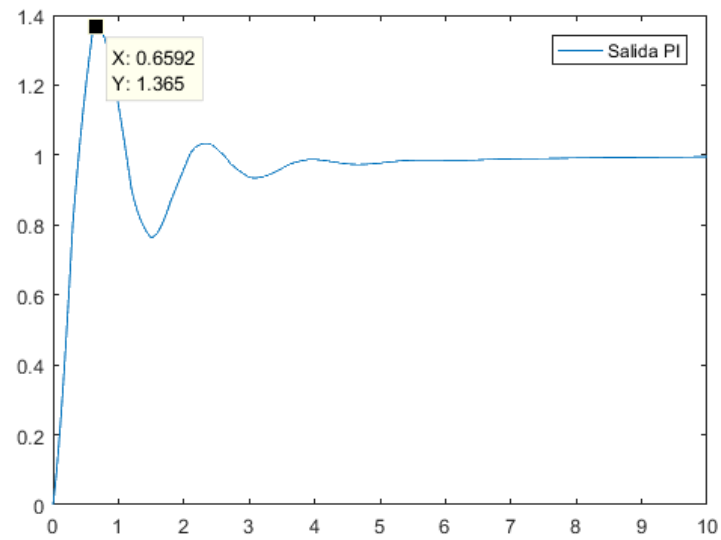


Figura 9: Salida controlador

4.1.2. Señal error

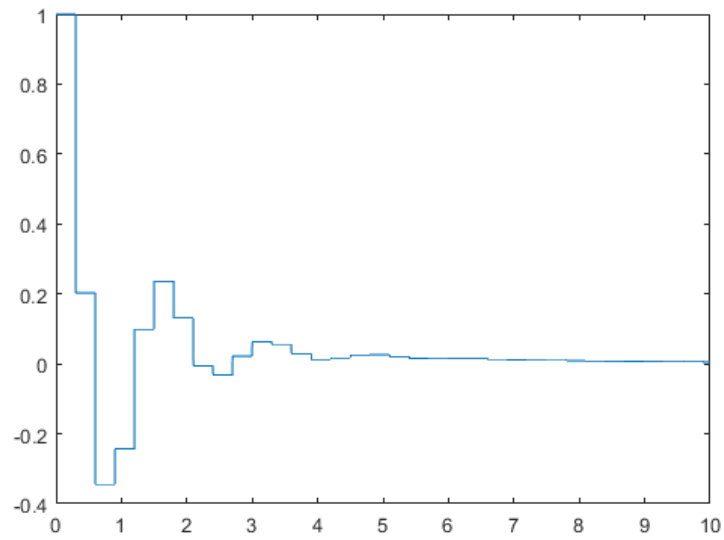


Figura 10: Señal error

4.1.3. Accion integral

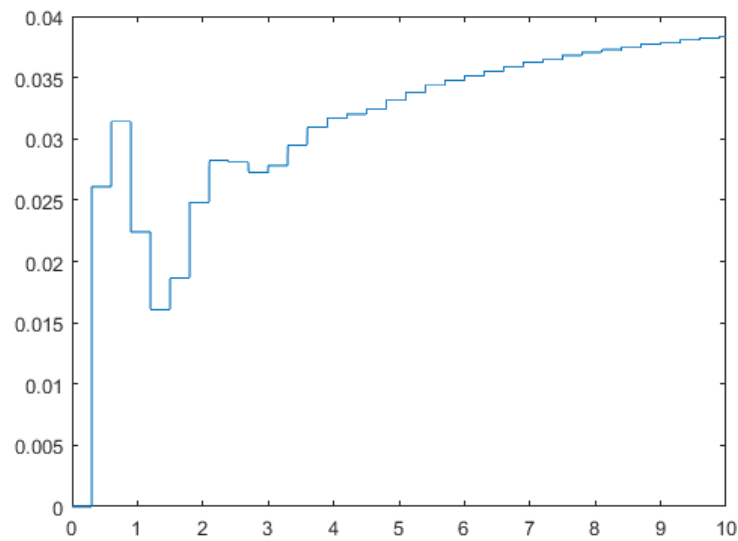


Figura 11: Accion integral

4.1.4. Accion proporcional

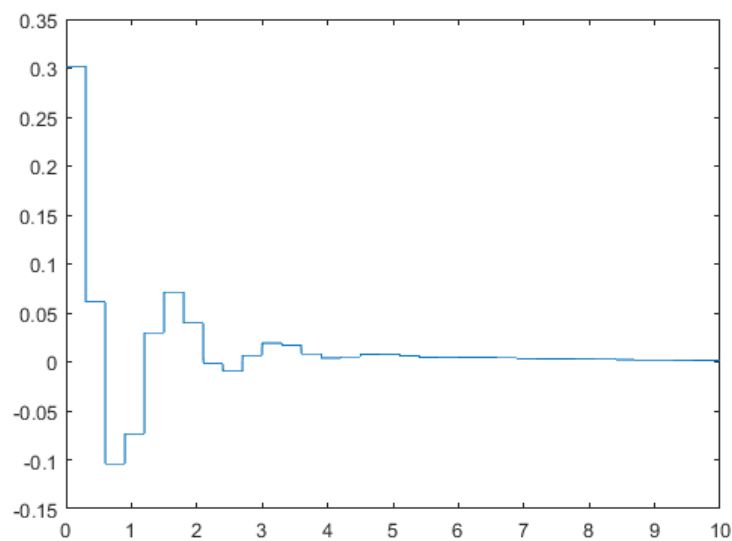


Figura 12: Accion proporcional

4.2. Controlador PID

Se presentará los reesultados obtenidos a partir de la simulacion de Simulink. Para obtener los valores a cargar en el mismo se uso el siguiente commando:

```
pid(C) %%valores de constantes K PID
```

El circuito de simulacion es el siguiente:

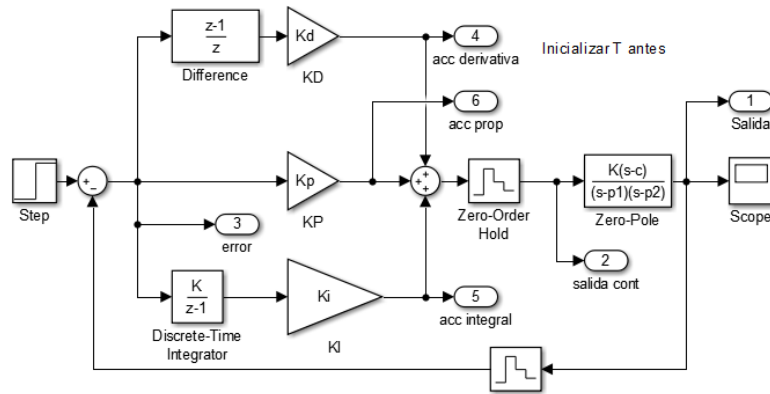


Figura 13: Archivo simulink

Obteniendo como resultado:

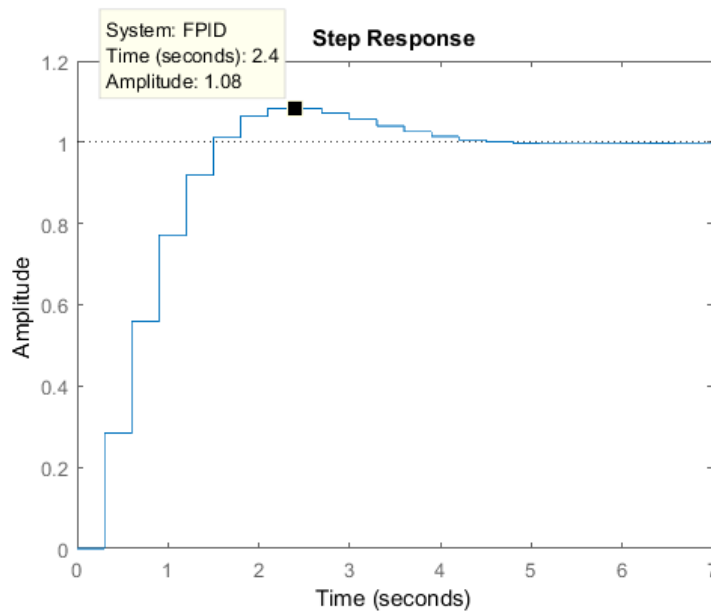


Figura 14: Archivo simulink

4.3. Comparación entre controladores

Se puede observar que ambos controladores con la aparición del polo integrador permite que el **error de estado estable** sea casi nulo o que la referencia en el infinito se establezca. En ese aspecto ambos tienen un comportamiento similar

Con respecto al **tiempo de establecimiento** la consigna pedía un tiempo de establecimiento de 4 segundo, donde podemos ver que se aproxima con bastante exactitud al valor deseado con algún tipo de erro basado en el redondeo de los valores introducidos en polos y ceros.

Finalmente con respecto al **sobrepaso** se requería un sobrepaso del 5 % . En ambos casos no cumple este requisito con satisfacción. En el caso del PI tenemos un sobrepaso del 30 % y en el caso del PID se acerca un poco mas llegando al 8 % del sobrepaso pero se escapa de nusetro . Con algunos ajustes en los polos y ceros , est ultimo podria mejorarse llegando al valor deseado

5. Conclusión

Se pudo apreciar el comportamiento de controladores PI Y pid para compensar un sistema de tiempo continuo discretizado por un ZOH. Se tuvo en cuenta las propiedades de las transformadas para hacer equivalencia entre planos s y z. El uso de la herramienta SISOTOOL permitio el diseño mediantet una interfaz grafica. Vemos que los resultados no son del todo exacto pero las respuestas son aceptables para los requerimientos que se nos pedia.