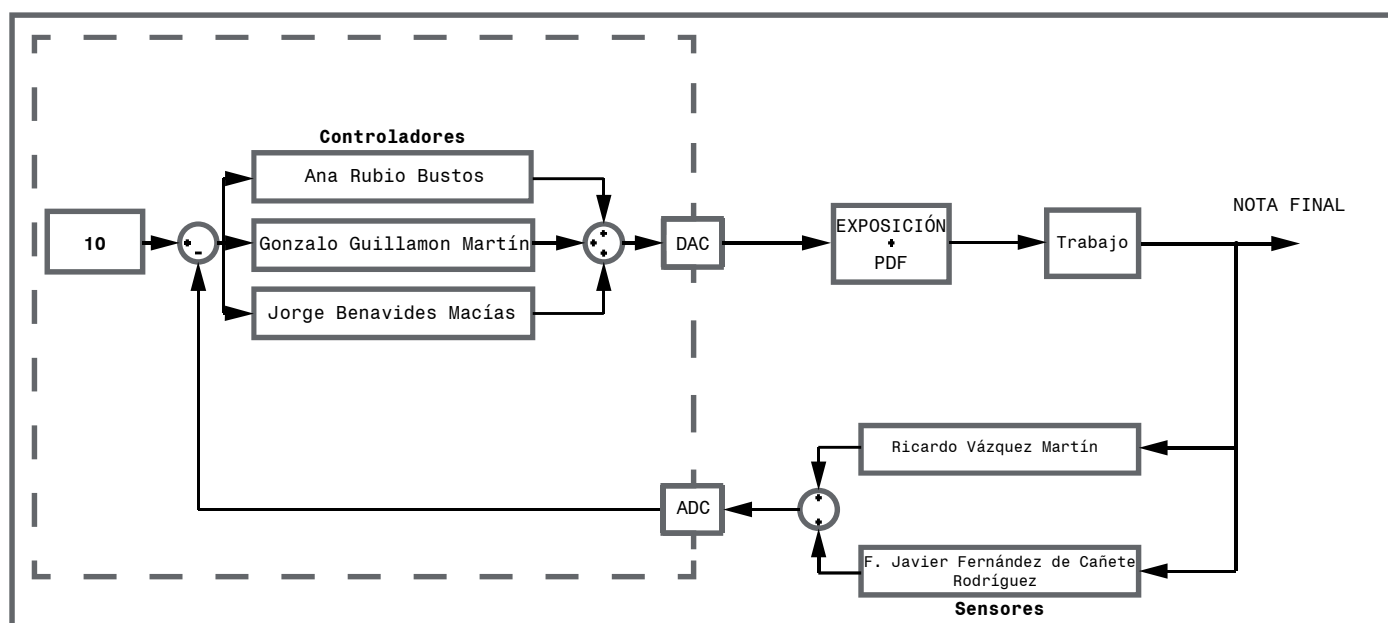




UNIVERSIDAD DE MÁLAGA

CONTROL POR COMPUTADOR

TRABAJO EN GRUPO



Índice

1. Objetivo general	4
2. Planteamiento del problema de control	4
3. Actividades	5
Actividad 1	5
Actividad 2	5
Actividad 3	6
Actividad 4	8
Actividad 5	9
Actividad 6	12
Actividad 7	13
Actividad 8	14
Actividad 9	16
Actividad 10	18
Actividad 11	20
Actividad 12	24
Actividad 13	25
Actividad 14	28
Actividad 15	29
Actividad 16	33

Actividad 17	35
4. Código escrito en MATLAB	36

Índice de figuras

1. Respuesta de funciones de transferencia frente a una rampa.	7
2. Especificaciones de respuesta transitoria en bucle cerrado.	12
3. Polos en bucle cerrado $k_a = 1$	13
4. Polos en bucle cerrado $k_a = 10$	13
5. Rltool - Ganancia crítica.	14
6. Compensador proporcional con ganancia igual a 1.	15
7. Controlador PD.	16
8. Respuesta del sistema frente a un escalón con un controlador PD.	17
9. Controlador PID.	18
10. Respuesta del sistema frente a un escalón con un controlador PID.	19
11. Respuesta del sistema frente a una rampa con un controlador PID.	20
12. Simulink - Diagrama de bloques.	21
13. Simulink - Respuesta a un escalón.	21
14. Simulink - Diagrama de bloques.	22
15. Simulink - Respuesta a un escalón.	22
16. Simulink - Respuesta a una rampa.	23
17. Simulink Resintonización PD.	24
18. Simulink Resintonización PID.	24
19. Respuesta del sistema en bucle abierto frente a un escalón.	25
20. Respuesta continua Ziegler-Nichols	26

21. Respuesta discretizada mediante el método Trapezoidal	27
22. Respuesta del sistema abierto frente a un escalón.	29
23. Respuesta del sistema cerrado frente a un escalón.	30
24. Respuesta del sistema abierto frente a un escalón.	31
25. Respuesta del sistema cerrado frente a un escalón.	32
26. Respuesta del sistema con error nulo.	34
27. Respuesta del sistema con error nulo.	35

1. Objetivo general

El objetivo general del trabajo de la asignatura es el de fomentar la comprensión de los conceptos que se abordan a lo largo del curso, en particular los relativos a la obtención de modelo lineal de una planta, discretización y efecto del muestreo, análisis de la respuesta transitoria y permanente continua y discreta, lugar de las raíces en el plano s y z , efecto del retardo y diseño de controlador analógico y digital PID en sus diferentes métodos, así como el control en espacio de estado.

Para ello se parte de un sistema univariable definido por un diagrama de bloques que representan su dinámica temporal, y se tratará de obtener su comportamiento dinámico y de diseñar un controlador PID o en espacio de estado que cumpla unas especificaciones de funcionamiento.

El trabajo se realizará en formato colaborativo y en grupo de 2 o 3 alumnos bajo un soporte fichero tipo Word, donde se irán detallando las diferentes tareas secuenciales de las que consta el trabajo conforme se vayan realizando. Al final se subirá el trabajo en formato .pdf al Campus Virtual y se realizará una exposición oral del trabajo por parte del grupo completo al profesor y se evaluará junto con el contenido del trabajo como calificación de asignatura.

2. Planteamiento del problema de control

El objetivo del trabajo es controlar la posición de una carga mecánica de acuerdo con una posición de consigna, utilizando para ello un esquema tipo servomecanismo.

El sistema inicialmente está controlado proporcionalmente en base a una ganancia variable k_a sobre un amplificador electrónico que actúa en base a la diferencia de posición entre el ángulo de consigna $\theta_r(t)$ y el ángulo de la carga mecánica $\theta_L(t)$.

El sistema objeto del control es un motor de CC con tren de engranajes reductor que tiene acoplado una carga mecánica, cuya función de transferencia viene indicada en el diagrama de bloques adjunto

con señal de entrada $E_a(s)$ y señal de salida $\theta_L(s)$ con parámetros físicos determinados.

El sistema de control de posición será controlado a través de un controlador PID analógico que reemplazará al controlador proporcional con ganancia k_a , disponiéndose para ello de un sensor de posición (potenciómetro) de ganancia k_R .

El esquema de control propuesto también permite alternativamente controlar la velocidad del servomotor, sustituyendo el sensor de posición por un sensor de velocidad (tacómetro) de ganancia k_w .

Por otra parte, se tratará de realizar un control digital PID sobre el sistema servomotor ya sea en posición o en velocidad, para lo cual se sustituirá el amplificador por un computador añadiendo los dispositivos de muestreo y reconstrucción pertinentes.

Finalmente se tratará también la representación en espacio de estado y el diseño del sistema de control por realimentación del vector de estado.

3. Actividades

Actividad 1

Ejecutar la aplicación `PARÁMETROS_SERVO.M` a través del DNI del miembro de mayor edad del grupo para calcular los parámetros del sistema servomotor.

Parámetros					
k_i	0.0544	b_{eq}	0.0344	k_r	3.8197
k_b	0.0444	R_a	0.3438	k_w	1
J_{eq}	0.0064	L_a	0.0054	n	0.1

Cuadro 1: Tabla de parámetros - DNI de Gonzalo Guilamon Martín

Actividad 2

Obtener la ecuación diferencial del servomotor de posición (en bucle abierto) a partir de la función de transferencia en s . **Con condiciones iniciales nulas.**

$$\begin{aligned}
 \frac{\theta_L(s)}{E_a(s)} &= \frac{nk_i}{s((L_a s + R_a)(J_{eq} s + b_{eq}) + nk_i k_b)} \\
 nk_i E_a(s) &= [s((L_a s + R_a)(J_{eq} s + b_{eq}) + nk_i k_b)] \theta_L(s) \\
 nk_i E_a(s) &= [(L_a s^2 + R_a s)(J_{eq} s + b_{eq}) + nk_i k_b s] \theta_L(s) \\
 nk_i E_a(s) &= [J_{eq} L_a s^3 + b_{eq} L_a s^2 + R_a J_{eq} s^2 + R_a b_{eq} s + nk_i k_b s] \theta_L(s) \\
 nk_i E_a(s) &= s^3 \theta_L(s) J_{eq} L_a + s^2 \theta_L(s) (b_{eq} L_a + (s) R_a J_{eq}) + s \theta_L(s) (R_a b_{eq} + nk_i k_b) \\
 \mathcal{L}^{-1} [nk_i E_a(s)] &= \mathcal{L}^{-1} [s^3 \theta_L(s) J_{eq} L_a + s^2 \theta_L(s) (b_{eq} L_a + R_a J_{eq}) + s \theta_L(s) (R_a b_{eq} + nk_i k_b)] \\
 nk_i E_a(t) &= \theta_L'''(t) J_{eq} L_a + \theta_L''(t) (b_{eq} L_a + R_a J_{eq}) + \theta_L'(t) (R_a b_{eq} + nk_i k_b) \\
 0 &= \theta_L'''(t) J_{eq} L_a + \theta_L''(t) (b_{eq} L_a + R_a J_{eq}) + \theta_L'(t) (R_a b_{eq} + nk_i k_b) - nk_i E_a(t) \\
 0 &= 34,56 \cdot 10^{-6} \theta_L'''(t) + 220,32 \cdot 10^{-6} \theta_L''(t) + 12,068 \cdot 10^{-3} \theta_L'(t) - 5,44 \cdot 10^{-3} E_a(t)
 \end{aligned} \tag{1}$$

Actividad 3

Obtener la función de transferencia en z del sistema servomotor de posición discretizado con $T = 0,05$ (en bucle abierto) a través de MATLAB y obtener la ecuación en diferencias correspondiente.

Existen otros métodos para discretizar la planta pero solo podemos usar el 'zoh' porque da el mejor resultado frente a la reconstrucción del filtro de primer orden ('foh').

Función de transferencia en z .

```
>>> Gposicionz = c2d(Gt,T,'zoh')
```

```
Gposicionz =
```

```
0.001609 z^2 + 0.003276 z + 0.0003034
-----
z^3 - 1.801 z^2 + 0.8331 z - 0.03168
```

```
Sample time: 0.05 seconds
```

```
Discrete-time transfer function.
```

Considerando el **sistema lineal discreto** descrito por la ecuación en diferencias

$$y(k+n) + a_1 \cdot y(k+n-1) + \dots + a_n \cdot y(k) = b_0 \cdot u(k+n) + \dots b_n \cdot u(k) \quad (2)$$

con función de transferencia

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 \cdot z^n + b_1 \cdot z^{n-1} + \dots + b_n}{z^n + a_1 \cdot z^{n-1} + \dots + a_n} \quad (3)$$

luego podemos usar esto para extraer la función.

$$G(z) = \frac{Y(z)}{U(z)} = \frac{0,016z^2 + 0,03231z + 0,003004}{z^3 - 1,763z^2 + 0,7949z - 0,03168} \quad (4)$$

$$n = 3; \quad b_1 = 0,0016; \quad b_2 = 0,03231; \quad b_3 = 0,003004; \quad a_1 = -1,763; \quad a_2 = 0,7949; \quad a_3 = -0,03168$$

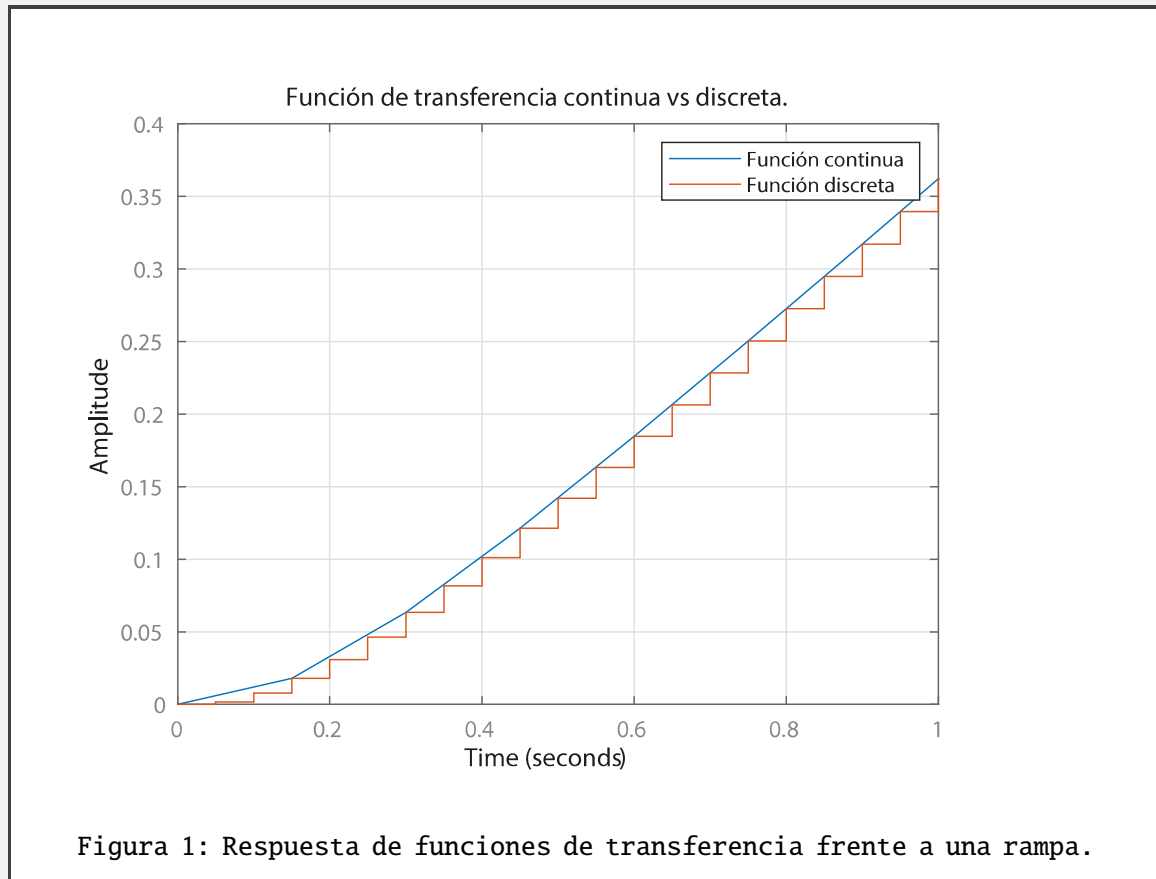
Con los parámetros anteriores podemos calcular la ecuación en diferencias.

$$y(k+3) - 1,763y(k+2) + 0,7949y(k+1) - 0,03168y(k) = 0,016u(k+2) + 0,03231u(k+1) + 0,003004u(k) \quad (5)$$

Para confirmar que la conversión continua-discreta ha sido correcta hemos representado ambas funciones de transferencia en una misma gráfica, con un entrada impulso.

Funciones de transferencia.

```
>>> figure
>>> step(Gposicion)
>>> hold on
>>> step(Gposicionz)
>>> xlim([0 1])
>>> title('Función de transferencia continua vs discreta.')
>>> legend('Función continua', 'Función discreta')
>>> grid on
```



Actividad 4

Obtener la descripción en espacio de estado A , B , C y D del sistema servomotor de velocidad y calcular las matrices G , H , C y D del sistema por discretización.

Espacio de estados continuo.

```
>>> [A,B,C,D] = ssdata(ss(Gt))
```

```
A =  
-69.0417 -21.8248 0  
16.0000 0 0  
0 0.1250 0  
B =  
8  
0  
0  
C =  
0 0 9.8380  
D =  
0
```

Espacio de estados discretos.

```
>>> [G,H,Cz,Dz,T]=ssdata(ss(Gtz))
```

```
G =  
1.8015 -0.8331 0.1267  
1.0000 0 0  
0 0.2500 0  
H =  
0.0625  
0  
0  
Cz =  
0.0257 0.0524 0.0194  
Dz =  
0  
T =  
0.0500
```

Actividad 5

Calcular el error en régimen permanente para el caso de usar controlador proporcional con ganancia k_a , ante entrada escalón y rampa para los casos continuo y discreto con $T = 0,05$ tanto del sistema servomotor de posición como de velocidad.

Posición

Continuo

$$G(s) = \frac{157,41}{s(s + 63,55)(s + 5,495)} \rightarrow \text{Tipo 1} \quad (6)$$

$$kG(s)H(s) = \frac{157,41 * kr * k}{s(s + 63,55)(s + 5,495)} = \frac{N(s)}{D(s)} \quad (7)$$

Mediante el **Teorema del valor final**

$$e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s \cdot E(s) = \lim_{s \rightarrow 0} \frac{sD(s)}{N(s) + D(s)} \cdot R(s) \quad (8)$$

■ Escalón $R(s) = \frac{1}{s}$

$$\lim_{s \rightarrow 0} \frac{sD(s)}{N(s) + D(s)} \cdot \frac{kr}{s} = \lim_{s \rightarrow 0} \frac{kr \cdot s(s + 63,55)(s + 5,495)}{601,2590 \cdot k + s(s + 63,55)(s + 5,495)} = \frac{0}{601,2590 \cdot k} = 0 \quad (9)$$

■ Rampa $R(s) = \frac{1}{s^2}$

$$\lim_{s \rightarrow 0} \frac{sD(s)}{N(s) + D(s)} \cdot \frac{kr}{s^2} = \lim_{s \rightarrow 0} \frac{kr \cdot (s + 63,55)(s + 5,495)}{601,2590 \cdot k + s(s + 63,55)(s + 5,495)} = \frac{63,55 \cdot 5,495 \cdot 3,8197}{601,2590 \cdot k} = \frac{2,2185}{k} \quad (10)$$

Discreto

Mediante el **Teorema del valor final**

■ Escalón $(1 - z^{-1})$

$$e_{ss} = \lim_{z \rightarrow 1} \left[(1 - z^{-1}) \frac{kr}{1 + GH(z)} \frac{1}{1 - z^{-1}} \right] \quad (11)$$

$$e_{ss} = \lim_{z \rightarrow 1} \left[\frac{kr}{1 + k_a \frac{0,0061465(z + 1,939)(z + 0,09724)}{(z - 1)(z - 0,7598)(z - 0,0417)}} \right] \quad (12)$$

$$e_{ss} = 0 \quad (13)$$

- Rampa $\frac{Tz^{-1}}{(1-z^{-1})^2}$

$$\lim_{z \rightarrow 1} \frac{1}{k_a \left(\frac{0,01609(z+1,939)(z+0,0974)}{(z-0,7598)(z-0,0417)} \right)} \quad (14)$$

$$e(\infty) = \frac{1}{k_a(4,6915)} \quad (15)$$

Velocidad

Continuo

$$G(s) = \frac{157,41}{(s+63,55)(s+5,495)} \rightarrow \text{Tipo } 0 \quad (16)$$

$$kG(s)H(s) = \frac{157,41 * k}{(s+63,55)(s+5,495)} = \frac{N(s)}{D(s)} \quad (17)$$

Mediante el **Teorema del valor final**

$$e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s \cdot E(s) = \lim_{s \rightarrow 0} \frac{sD(s)}{N(s) + D(s)} \cdot R(s) \quad (18)$$

- Escalón $R(s) = \frac{1}{s}$

$$\lim_{s \rightarrow 0} \frac{sD(s)}{N(s) + D(s)} \cdot \frac{kr}{s} = \lim_{s \rightarrow 0} \frac{(s+63,55)(s+5,495)}{157,4k + (s+63,55)(s+5,495)} = \frac{349,2072}{157,4k + 349,2072} = \frac{0,2218}{k + 0,2218} \quad (19)$$

- Rampa $R(s) = \frac{1}{s^2}$

$$\lim_{s \rightarrow 0} \frac{sD(s)}{N(s) + D(s)} \cdot \frac{kr}{s^2} = \lim_{s \rightarrow 0} \frac{(s+63,55)(s+5,495)}{157,4k + (s+63,55)(s+5,495)} \cdot \frac{1}{s} = \lim_{s \rightarrow 0} \frac{(s+63,55)(s+5,495)}{157,4k + (s+63,55)(s+5,495)} \cdot \lim_{s \rightarrow 0} \frac{1}{s} = \infty \quad (20)$$

Discreto

Mediante el **Teorema del valor final**

- Escalón $(1-z^{-1})$

$$e_{ss} = \lim_{z \rightarrow 1} \left[(1-z^{-1}) \frac{kr}{1+GH(z)} \frac{1}{1-z^{-1}} \right] \quad (21)$$

$$e_{ss} = \lim_{z \rightarrow 1} \left[\frac{kr}{1 + k_a \frac{0,077655(z+0,3364)}{(z-0,7598)(z-0,0417)}} \right] \quad (22)$$

$$e_{ss} = \frac{1}{1 + k_a(0,4508)} \quad (23)$$

- Rampa $\frac{Tz^{-1}}{(1-z^{-1})^2}$

$$e_{ss} = \lim_{z \rightarrow 1} \left[(1 - z^{-1}) \frac{kr}{1 + GH(z)} \frac{Tz^{-1}}{(1 - z^{-1})^2} \right] \quad (24)$$

$$e_{ss} = \lim_{z \rightarrow 1} \left[\frac{T \cdot kr}{(1 - z^{-1}) \cdot kr \cdot \frac{0,077655(z + 0,336)}{(z - 0,7598)(z - 0,0417)}} \right] \quad (25)$$

$$e_{ss} = \infty \quad (26)$$

Actividad 6

Aplicar `ltiview` al sistema servomotor de posición en bucle cerrado discretizado con ganancia $k_a = 1$ y calcular las especificaciones de respuesta transitoria (t_{subida} , t_{pico} , $t_{establecimiento}$, sobreoscilación) ante entrada escalón unitario.

Sistema servomotor de posición.

```
>>> transient_ans = kr*feedback(Gposicionz,kr)
>>> ltiview('step',transient_ans)
```

```
transient_ans =

0.006147 z^2 + 0.01251 z + 0.001159
-----
z^3 - 1.795 z^2 + 0.8456 z - 0.03052

Sample time: 0.05 seconds
Discrete-time transfer function.
```

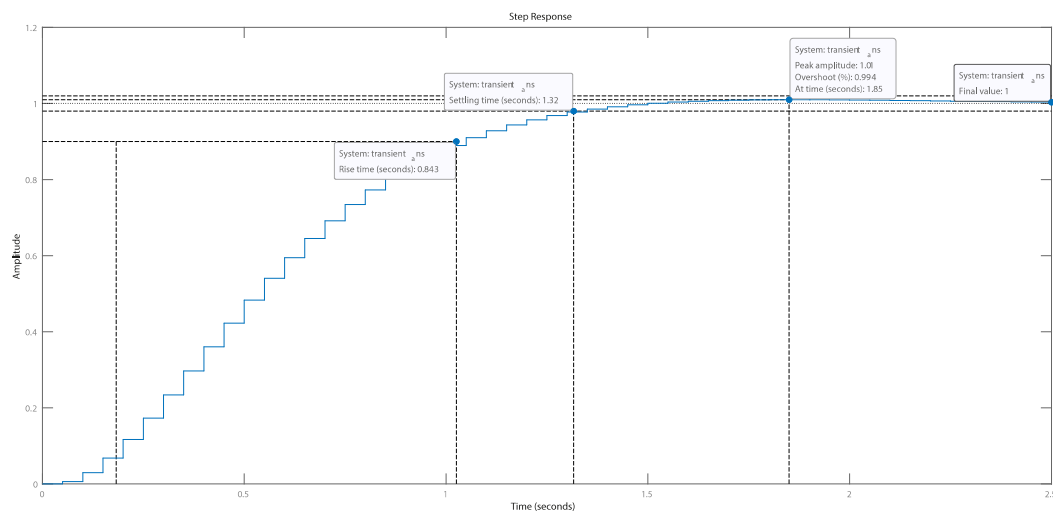


Figura 2: Especificaciones de respuesta transitoria en bucle cerrado.

Actividad 7

Trazar el lugar de las raíces discreto en z del sistema servomotor de posición con `RLTOOL` y situar los polos en bucle cerrado para $k_a = 1$, $k_a = 10$ y calcular los valores de la ganancia crítica para el caso de que el controlador PID discreto sea una ganancia k_a , comentando los resultados.

Polos en bucle cerrado.

```
>>> rltool(Gposicionz)
```

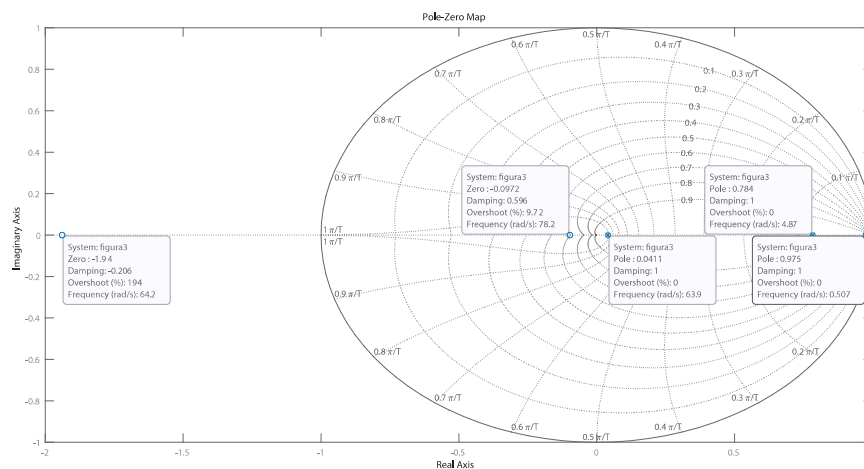


Figura 3: Polos en bucle cerrado $k_a = 1$.

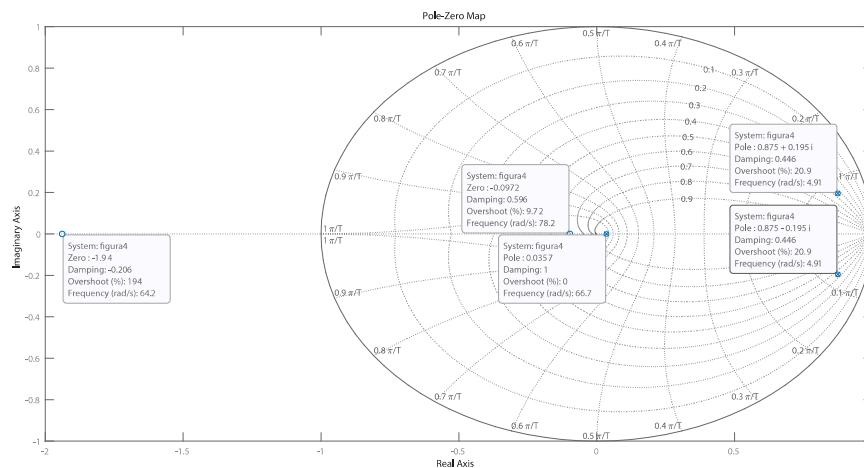


Figura 4: Polos en bucle cerrado $k_a = 10$.

Mediante rltool y la opción NEW PLOT >NEW POLE - ZERO MAP podemos generar una gráfica con los polos según la ganancia que colequemos en el compensador. Los resultados son los siguientes:

- $k_a = 1$ Como se puede observar en la figura 3 los polos son:

Polos		
0.0393	0.878+0.0758i	0.878-0.0758i

- $k_a = 10$ Como se puede observar en la figura 4 los polos son:

Polos		
0.0218	0.859+0.428i	0.859-0.428i

Para el calculo de la ganancia crítica a través de la herramienta rltool, variamos la ganancia proporcional hasta llegar al límite de la estabilidad.

$$k_{a\text{ganancia crítica}} = 15.232$$

Ganancia crítica.

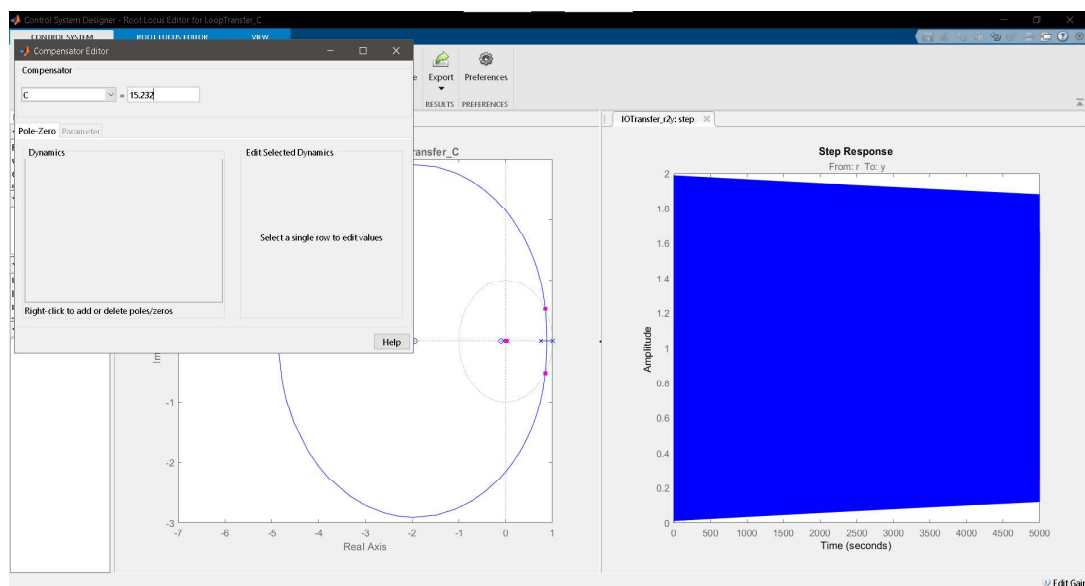


Figura 5: Rltool - Ganancia crítica.

Actividad 8

Verificar si es posible compensar en discreto el sistema servomotor de posición con un controlador proporcional para que el sistema en bucle cerrado cumpla las especificaciones de sobreoscilación $< 5\%$ y tiempo de establecimiento $t_{est} < 0,3s$. y error nulo ante consigna escalón unitaria, graficando la respuesta escalón.

Compensador proporcional en tiempo discreto.

```
>>> rltool(Gposicionz)
```

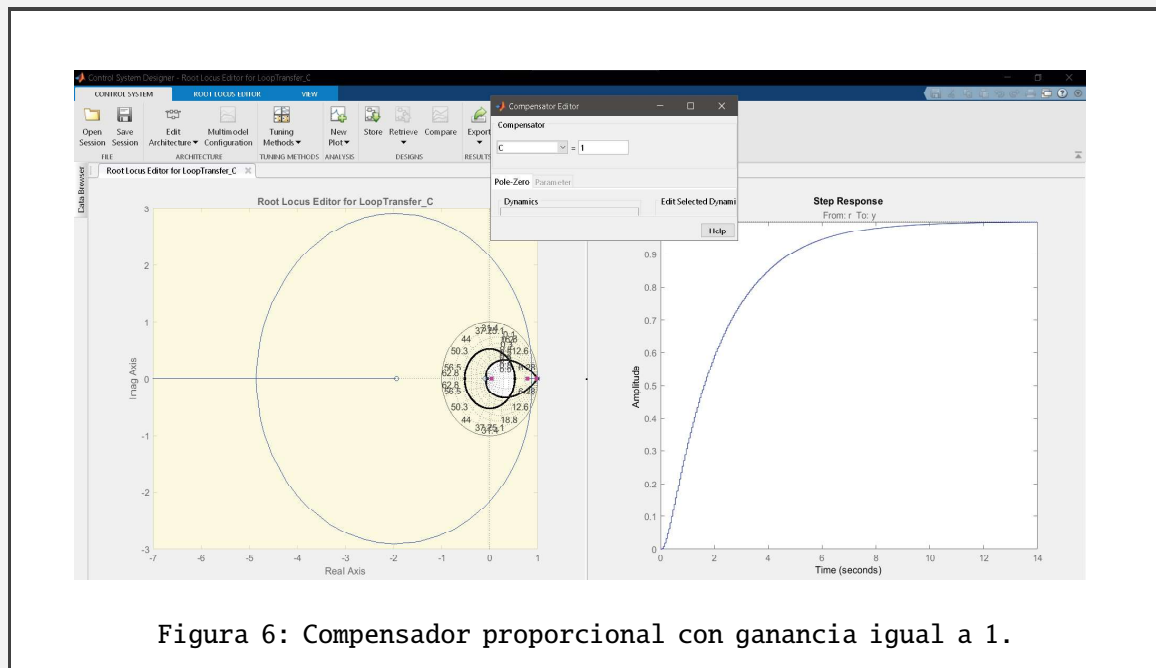


Figura 6: Compensador proporcional con ganancia igual a 1.

Utilizando el comando `RLTOOL` con nuestra planta, en la ventana `ROOT LOCUS EDITOR` añadimos las especificaciones necesarias, se puede observar como los polos en bucle cerrado en nuestro lugar de la raíces nunca llegan a cumplir todas las especificaciones con un control proporcional ya que los polos en bucle cerrado nunca llegan a la zona blanca sin importar la ganancia positiva que pongamos.

Actividad 9

En caso contrario diseñar un controlador tipo PID digital implementable (PD, PI o PID) en el lugar de las raíces en z y ver si es posible cumplir las especificaciones requeridas, expresando el controlador obtenido, graficando el nuevo lugar de las raíces en z , y comprobando a través de **LTIVIEW** que se cumplen ciertamente las especificaciones requeridas.

Controlador PD con sobreoscilamiento $< 5\%$ y $t_{est} < 0,3$.

```
>>> rltool(Gposicionz)
```

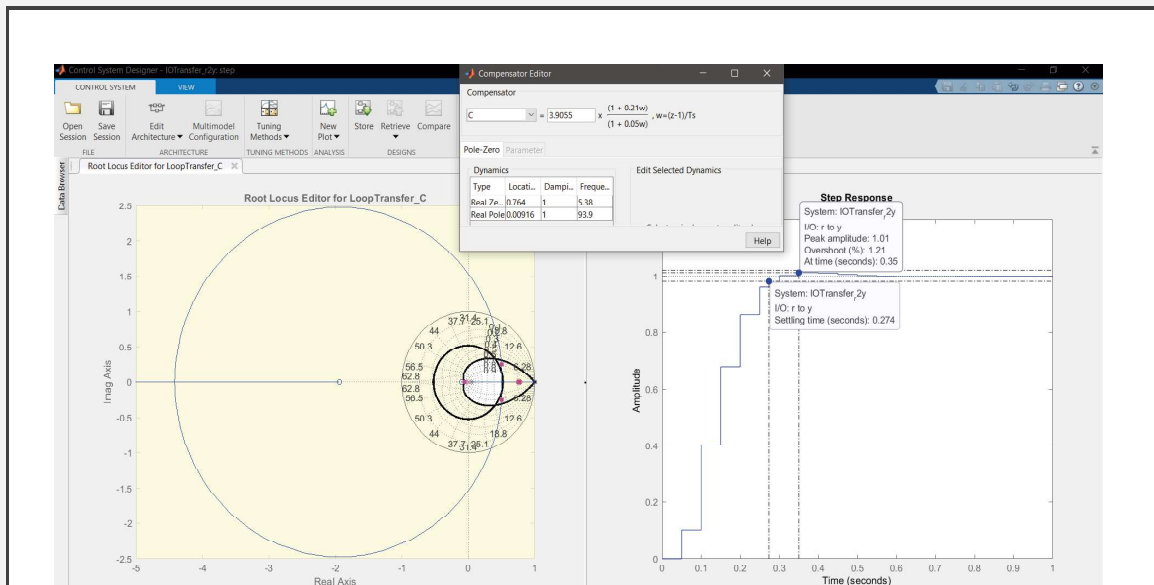


Figura 7: Controlador PD.

Mediante la herramienta **EXPORT>EXPORT TUNED BLOCKS**, podemos enviar el controlador al espacio de trabajo.

```
>>> PD_9
```

```
PD_9 =

    16.404 (z-0.7641)
    -----
    (z-0.009156)

Sample time: 0.05 seconds
Discrete-time zero/pole/gain model.
```

Respuesta del sistema en tiempo discreto con un controlador PD.

```
>>> ltiview('step',kr*feedback(PD_9*Gposicionz,kr))
```

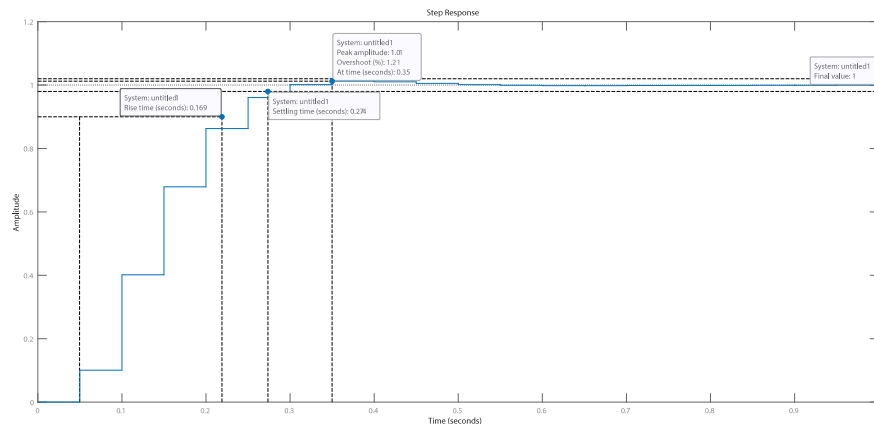


Figura 8: Respuesta del sistema frente a un escalón con un controlador PD.

Actividad 10

Repetir el apartado anterior realizando un diseño de controlador PID discreto implementable (PD, PI o PID) y ver si es posible cumplir las especificaciones de sobreoscilación $< 10\%$, tiempo de establecimiento $t_{est} < 1s$, error nulo ante ángulo de consigna escalón unitario y error nulo ante ángulo de consigna rampa unitaria, graficando la respuesta escalón y rampa, graficando el nuevo lugar de las raíces en z , y comprobando a través de LTRVIEW que se cumplen ciertamente las especificaciones requeridas.

Controlador PID con sobreoscilamiento $< 10\%$ y $t_{est} < 1$.

```
>>> rltool(Gposicionz)
```

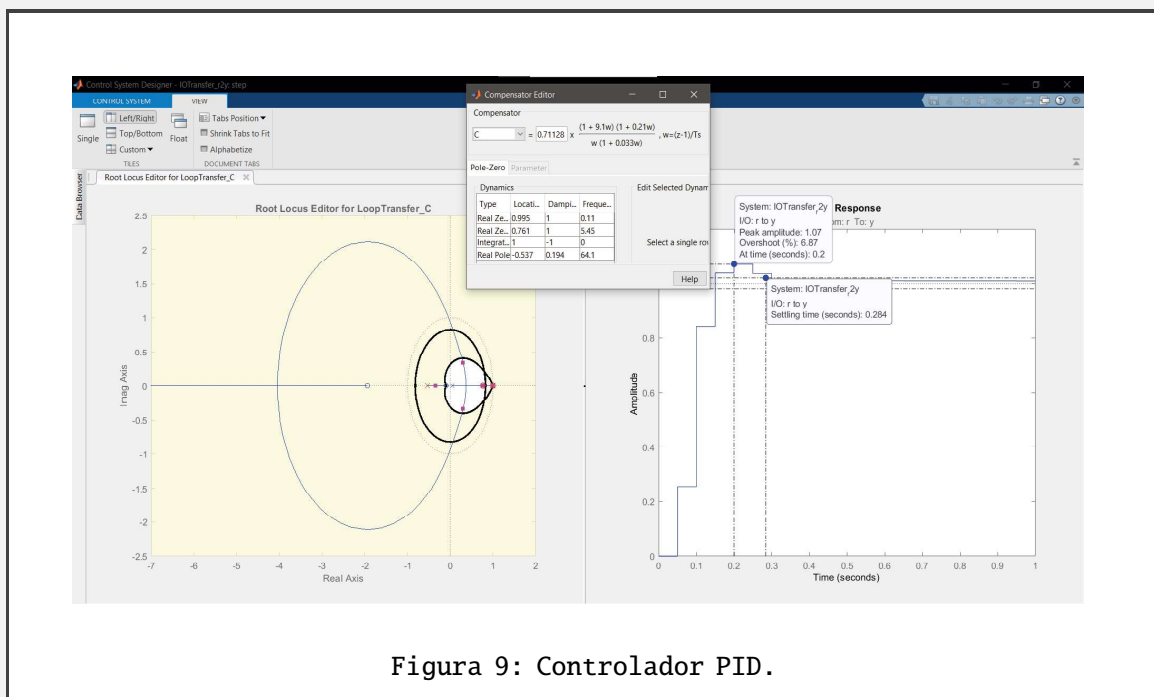


Figura 9: Controlador PID.

Mediante la herramienta EXPORT>EXPORT TUNED BLOCKS. podemos enviar el controlador al espacio de trabajo.

```
>>> PID_10
```

```
PID_10 =

41.625 (z-0.9945) (z-0.7613)
-----
(z-1) (z+0.5366)

Sample time: 0.05 seconds
Discrete-time zero/pole/gain model.
```

Respuesta del sistema en tiempo discreto con un controlador PID.

```
>>> ltiview('step',kr*feedback(PID_10*Gposicionz,kr))
```

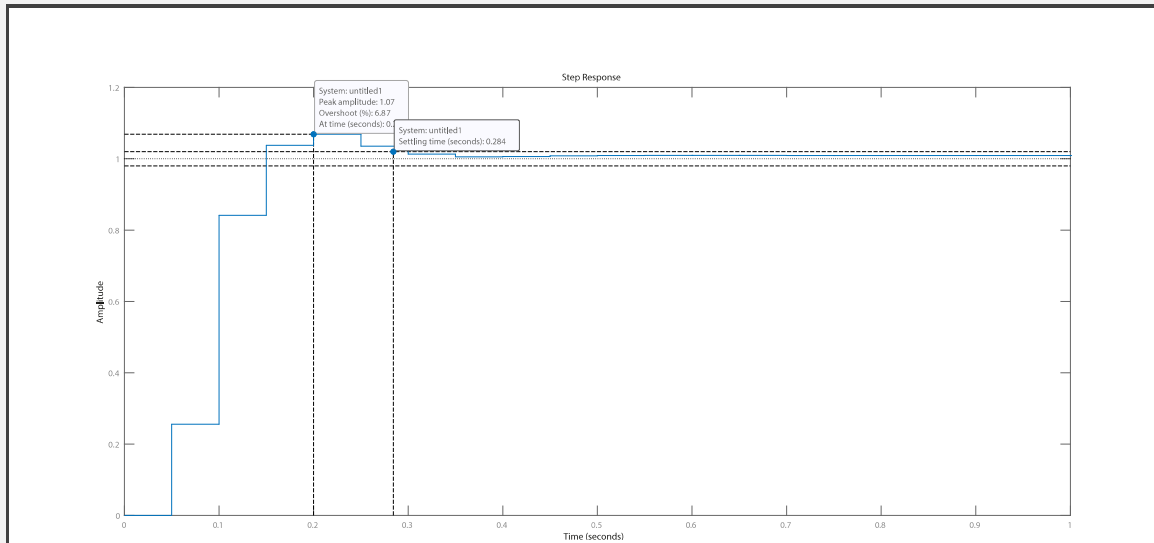


Figura 10: Respuesta del sistema frente a un escalón con un controlador PID.

Respuesta del sistema en tiempo discreto con un controlador PID.

```
>>> ltiview('lsim',kr*feedback(PID_10*Gposicionz,kr))
```

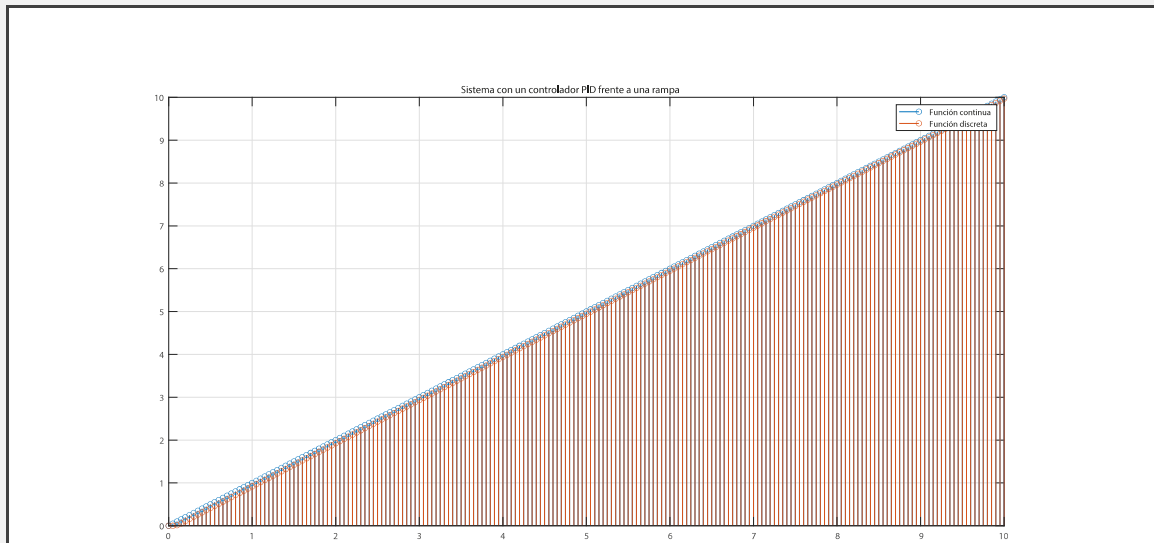


Figura 11: Respuesta del sistema frente a una rampa con un controlador PID.

Actividad 11

Simular en SIMULINK los sistemas de control PID discreto de los apartados 9. y 10. usando los parámetros correspondientes, añadiendo asimismo una limitación de salida del controlador discreto a $(-15,+15)$ añadiendo en su caso el sistema corrector anti-windup y ver la respuesta ante ángulo de consigna escalón unitaria, graficando asimismo la señal de control.

Simulink PD

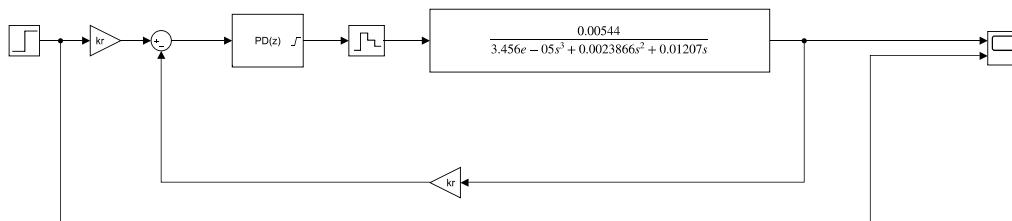


Figura 12: Simulink - Diagrama de bloques.

Use filtered derivative
 Filter coefficient (N): 45
 Upper limit: 15
 Lower limit: -15

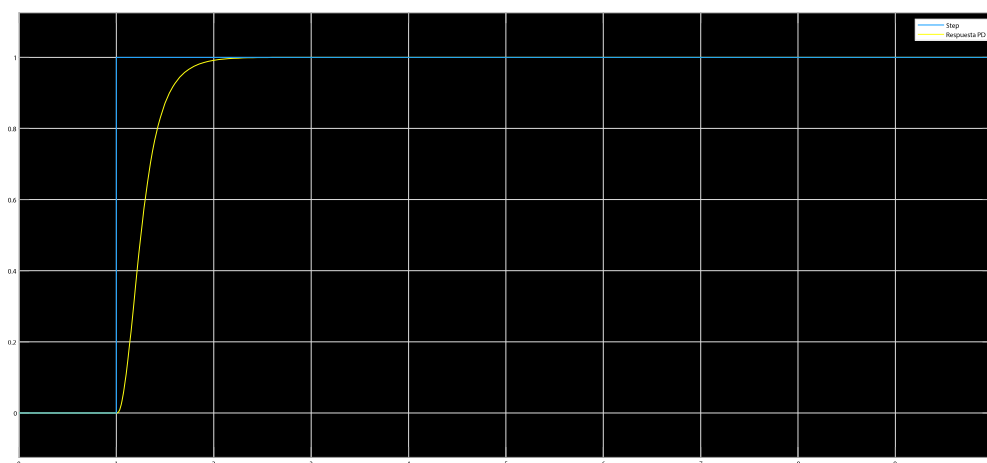


Figura 13: Simulink - Respuesta a un escalón.

Simulink PID

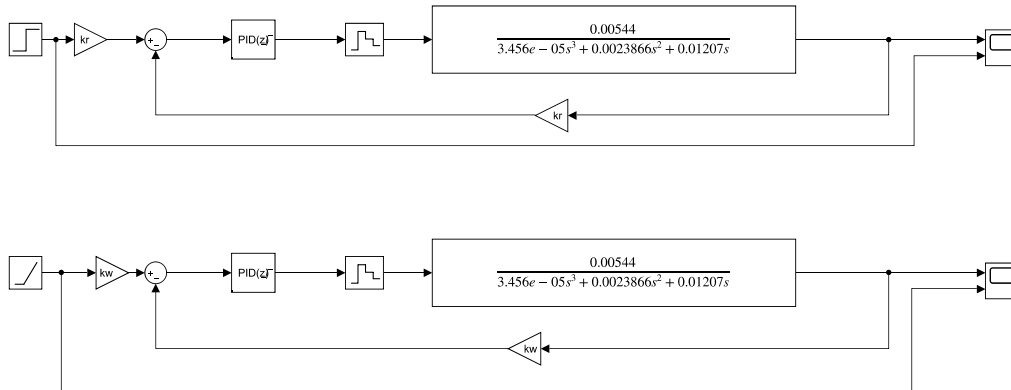


Figura 14: Simulink - Diagrama de bloques.

Use filtered derivative

Filter coefficient (N): 45

Upper limit: 15

Lower limit: -15

Anti-windup Method: Campling

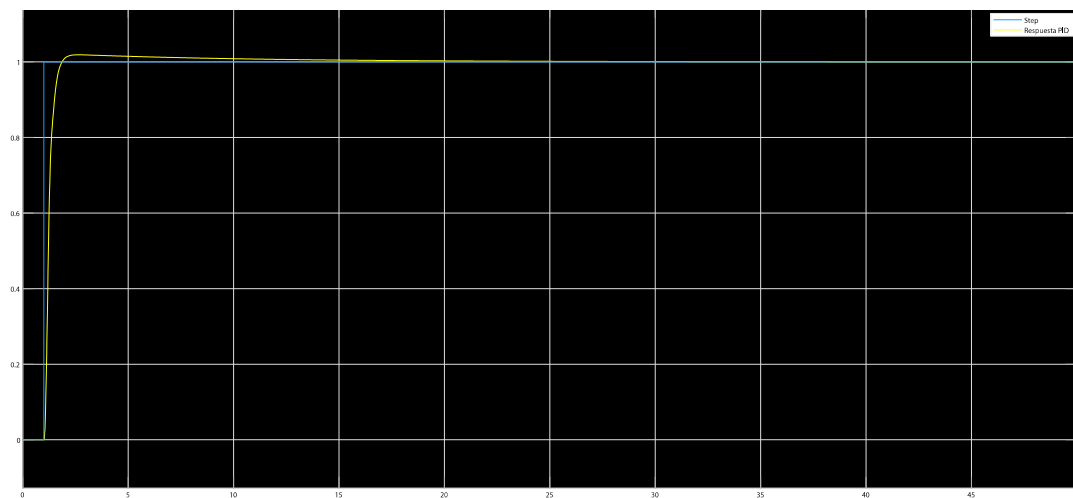


Figura 15: Simulink - Respuesta a un escalón.

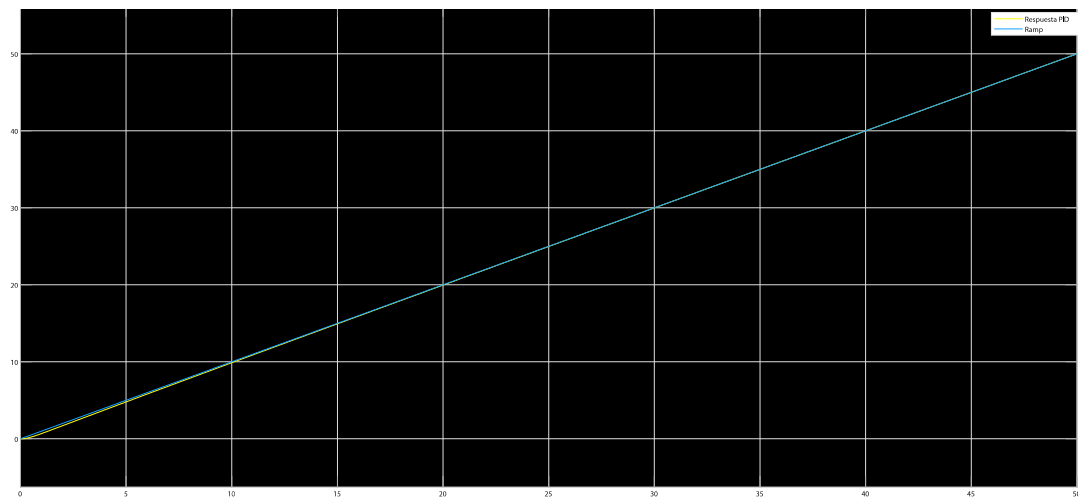


Figura 16: Simulink - Respuesta a una rampa.

Actividad 12

Resintonizar el controlador digital respectivamente en SIMULINK a través de las herramientas de diseño automático, obteniendo una mejora en la respuesta posible ante seguimiento de consigna ángulo de amplitud unitaria.

Simulink

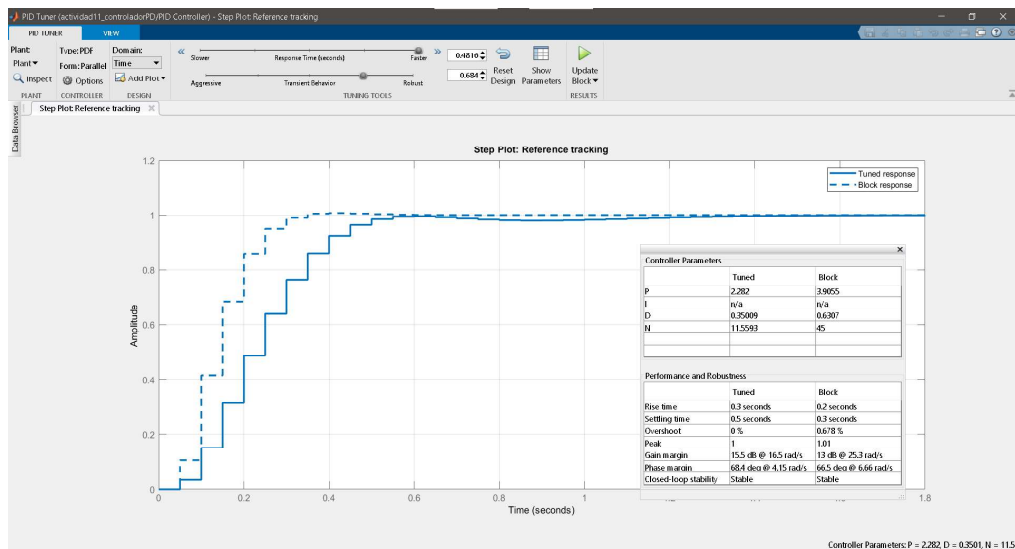


Figura 17: Simulink Resintonización PD.

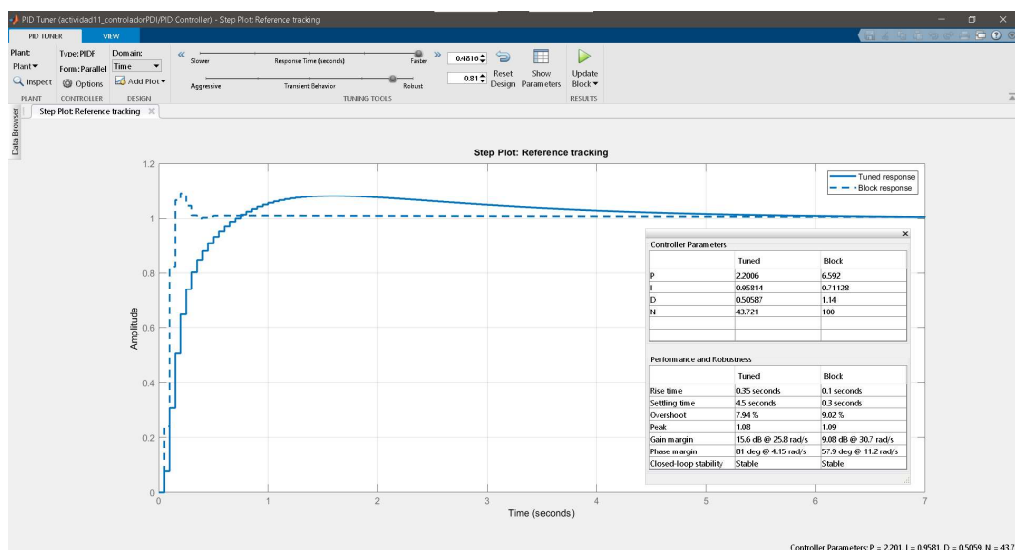


Figura 18: Simulink Resintonización PID.

Actividad 13

Diseñar un controlador analógico tipo PID (PD, PI, PID) para el sistema servomotor de posición por el MÉTODO DE ZIEGLER-NICHOLS en abierto y cerrado (si son posibles) y realizar su discretización con $T = 0,05$ aplicando el método trapezoidal y graficar la respuesta escalón unitaria del sistema de control en bucle cerrado del servomotor de posición a través de la herramienta RLTOOL.

Para implementar el método Ziegler-Nichols en bucle abierto, necesitamos que la respuesta ante entrada escalón sea aproximadamente la de un sistema de primer orden con retardo, por lo que no podemos aplicar este método. Por el contrario sí podemos implementar el método Ziegler-Nichols en bucle cerrado.

Método de Ziegler-Nichols

```
>>> step(Gposicion)
```

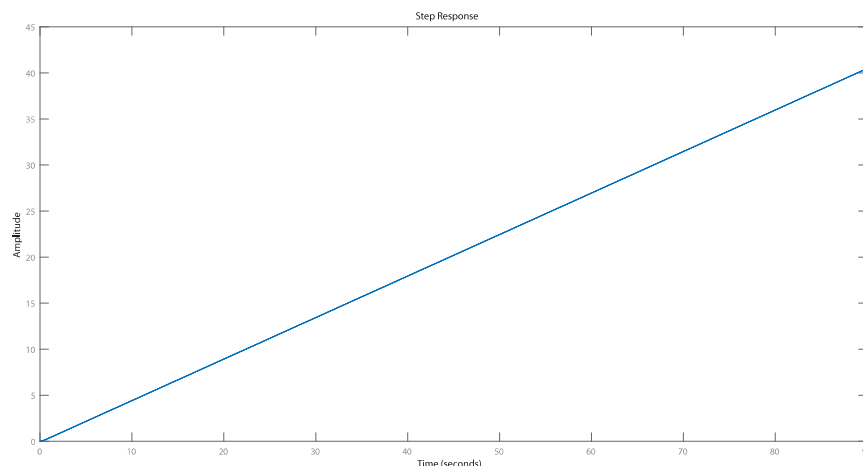


Figura 19: Respuesta del sistema en bucle abierto frente a un escalón.

Como es inestable no se puede hacer el PID Ziegler-Nichols en bucle abierto, por lo tanto lo haremos en bucle cerrado.

```
>>> [Kc,Pm,Wg]=margin(kr*Gposicion);
>>> Tc = 2*pi/Wg;
>>> Kp = 0.75*Kc;
>>> Ti = Tc/1.6;
>>> Ki = Kp/Ti;
>>> Td = Tc/10;
>>> Kd = Td*Kp;
>>> tf = 0.01;
```

```
>>> pidZN = pid(Kp,Ki,Kd,tf);
>>> rltool(Gposicion,pidZN);
>>> pidZNz = c2d(pidZN,T,'trapezoidal');
>>> rltool(Gposicionz,pidZNz);
```

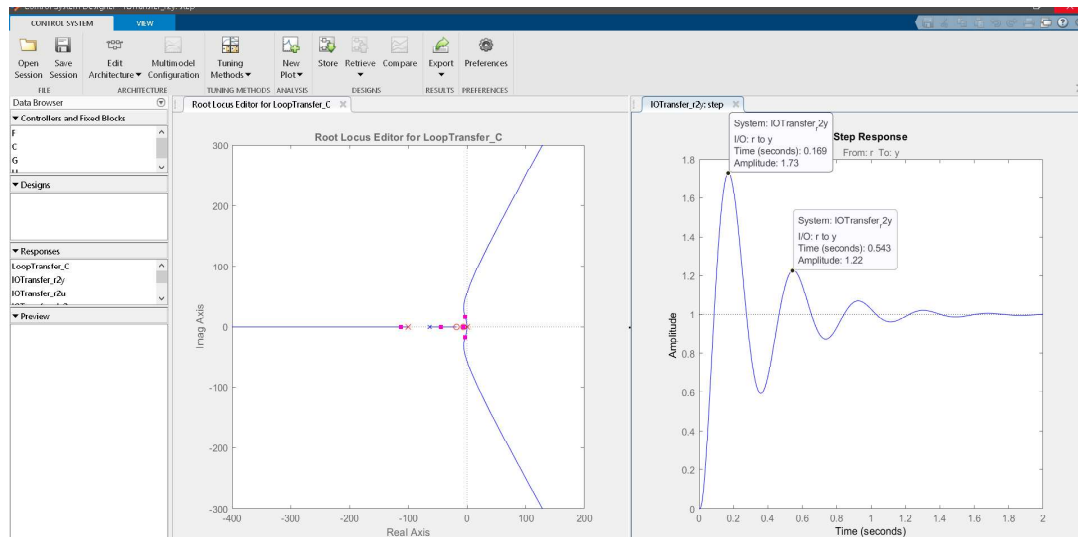


Figura 20: Respuesta continua Ziegler-Nichols

Calculamos la relación de sobreoscilación:

$$0.22/0.73 = 0.30 \rightarrow 30\%$$

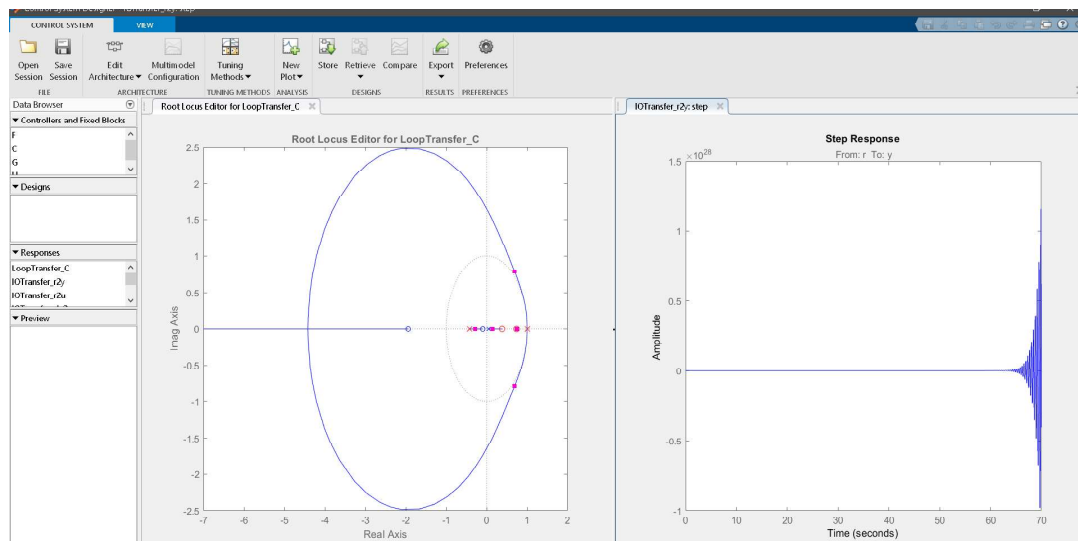


Figura 21: Respuesta discretizada mediante el método Trapezoidal

Mediante el método trapezoidal la discretización no es perfecta como podemos observar, luego el método de discretización influye en la respuesta lo que provoca que el sistema se vuelva inestable y que el PID no cumpla su función.

Actividad 14

Diseñar un controlador analógico tipo PID (PD, PI, PID) para el sistema servomotor de velocidad por el MÉTODO DE ZIEGLER-NICHOLS en abierto y cerrado (si son posibles) y realizar su discretización con $T = 0,05$ aplicando el método trapezoidal y graficar la respuesta escalon de posición unitario del sistema de control en bucle cerrado del servomotor de velocidad a través de la herramienta RLTOOL.

Para implementar el método Ziegler-Nichols en bucle abierto, necesitamos que la respuesta ante entrada escalón sea la respuesta aproximada de un sistema de primer orden con retardo y que cumpla $0,15 < d/\tau < 0,6$. Tras calcular los parámetros de nuestro sistema obtenemos $d/\tau = 3 > 0,6$ por lo que no es válido aplicar este método. En caso de implementar el método de Ziegler-Nichols en bucle cerrado, no se puede implementar debido a que nuestra planta es estable independientemente del valor de la ganancia, imposibilitando el cálculo de los parámetros en el límite de la estabilidad, ya que nunca se alcanza.

Método de Ziegler-Nichols

```
>>> [Kc,Pm,Wcg] = margin(Gvelocidad)
>>> velocidadz = c2d(Gvelocidad,0.05,'zoh')
>>> step(Gvelocidadz)
>>> T1 = 0.2;
>>> T2 = 0.1;
>>> d = (3*(T1-T2))/2;
>>> tau = T1-d;
>>> d/tau
```

```
Kc =
    Inf

Pm =
    Inf

Wcg =
    Inf

ans =
    3.0000
```

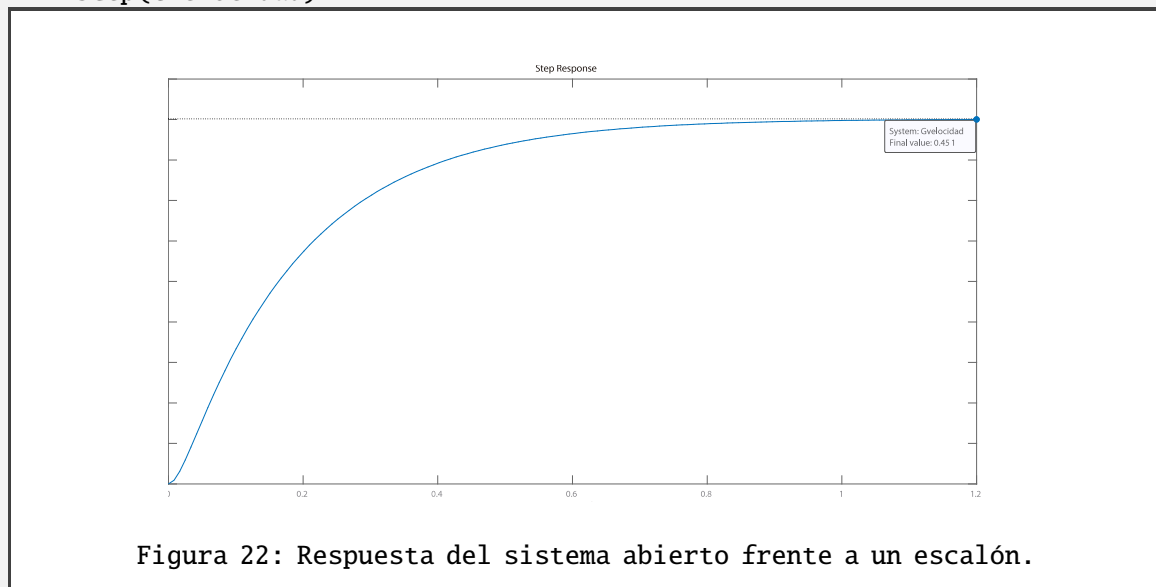
Actividad 15

Diseñar un controlador analógico tipo PID (PD, PI, PID) para el sistema servomotor de velocidad por el MÉTODO DE RIVERA-MORARI y de síntesis directa y realizar su discretización con $T = 0,05$ aplicando el método de emparejamiento polos-ceros y graficar la respuesta escalón unitaria del sistema de control en bucle cerrado del servomotor de velocidad a través de la herramienta RLTOOL.

Controlador síntesis directa

Primero comprobamos si se puede controlar a través de este método. Para ello, el sistema debe ser estable en bucle abierto.

```
>>> step(Gvelocidad)
```



Según la respuesta del sistema en bucle abierto, es estable por lo que se puede aplicar. Nuestra planta es una planta de segundo orden sobreamortiguada.

Calculamos los polos del sistema en bucle abierto

```
>>> [polo1,polo2]= pole(Gvelocidad);
```

Se calculan los parámetros:

k Es el valor final en bucle abierto, por lo tanto, $k = 0,451$.

```
>>> tau_1 = 1/63.5465
```

```
>>> tau_2 = 1/5.4951
```

Nuestra λ debe ser mayor que $0,2 * \tau_{max} = 0,2 * \tau_2 = 5,4951$.

Vamos a tomar

```
>>> lambda = 6
```

```
>>> k = 0.451; kp=(tau_1+tau_2)/(k*lambda);
```

```
>>> ti = tau_1 + tau_2;
```

```
>>> td = (tau_1*tau_2)/(tau_1+tau_2);
```

```
>>> PID = pidstd(kp,ti,td)
```

PID =

$$K_p * \left(1 + \frac{1}{T_i} * \frac{1}{s} + T_d * s \right)$$

with Kp = 0.0731, Ti = 0.198, Td = 0.0145

Continuous-time PID controller in standard form

```
>>> C = c2d(PID,0.05,'matched')
```

C =

$$K_p * \left(1 + \frac{T_s}{T_i} * \frac{z-1}{z-1} + T_d * \frac{z-1}{T_s} \right)$$

with Kp = 0.0961, Ti = 0.26, Td = 0.0417, Ts = 0.05

Sample time: 0.05 seconds

Discrete-time PID controller in standard form

```
>>> step(kw*feedback(Gvelocidadz*C,kw))
```

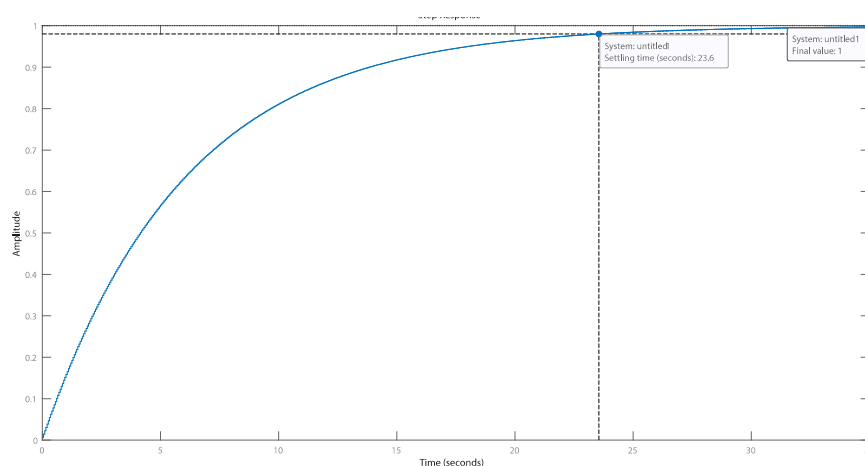


Figura 23: Respuesta del sistema cerrado frente a un escalón.

Controlador Rivera-Morari

Primero comprobamos si se puede controlar a través de este método. Para ello, el sistema debe ser estable en bucle abierto.

```
>>> step(Gvelocidad)
```

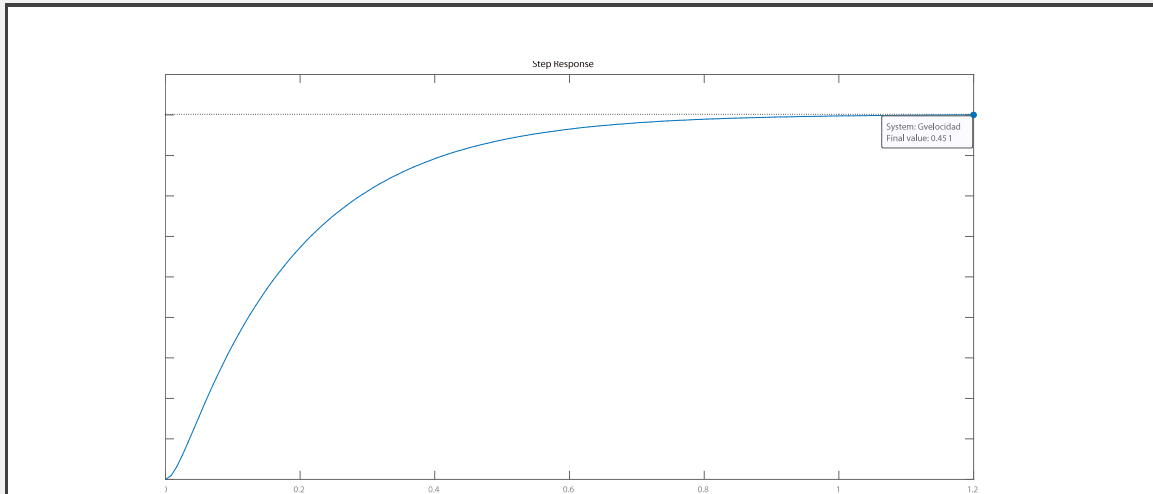


Figura 24: Respuesta del sistema abierto frente a un escalón.

A partir de la respuesta de nuestro sistema obtenemos:

```
>>> k = 0.451
```

```
>>> t632=0.119;t283=0.0773;
```

Obtenemos los parámetros para calcular nuestro PID:

```
>>> d=3*(t632-t283)/2;tau=t632-d;
```

Es recomendable que $\lambda > 0,2 * \tau = 0,0113$

Y que cumpla que $\lambda/d > 1,7$, es decir, $\lambda > d * 1,7 = 0,1063$

Elegiremos nuestro lamda

```
>>> lamda = 1;
```

Llendo a la tabla de PI

```
>>> kp = tau/(k*lamda);ti=tau;
```

Nuestro PI es:

```
>>> PI = pidstd(kp,ti)
```


PI =

$$K_p * \left(1 + \frac{1}{T_i} * \frac{1}{s} \right)$$

with $K_p = 0.125$, $T_i = 0.0565$

Continuous-time PI controller in standard form

```
>>> C = c2d(PID,0.05,'matched')
```

C =

$$K_p * \left(1 + \frac{T_s}{T_i} * \frac{1}{z-1} \right)$$

with $K_p = 0.189$, $T_i = 0.0851$, $T_s = 0.05$

Sample time: 0.05 seconds

Discrete-time PI controller in standard form

```
>>> step(kw*feedback(Gvelocidadz*C,kw))
```

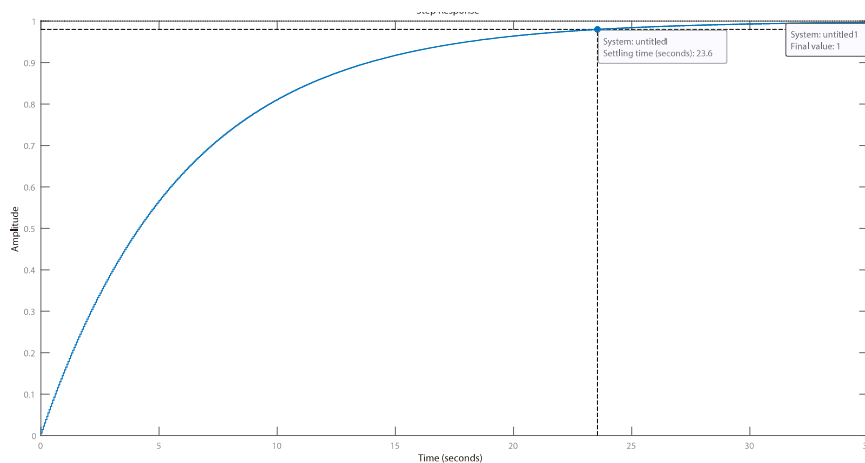


Figura 25: Respuesta del sistema cerrado frente a un escalón.

Actividad 16

Diseñar un controlador vector de ganancias en espacio de estado continuo para que el sistema en bucle cerrado en velocidad presente un par de polos deseados con comportamiento críticamente amortiguado y exhiba una frecuencia natural de ω_n de 10, asegurando error nulo ante consigna velocidad con amplitud unitaria, partiendo de condiciones iniciales nulas y especificando el resto de polos deseados a suficiente distancia de los anteriores. Asumir que el estado es observable. Posteriormente, graficar la respuesta del sistema de control en velocidad en MATLAB.

Aunque MATLAB nos proporciona un modelo en el espacio de estados para el sistema de velocidad hemos preferido hacerlo a mano para seleccionar un vector de estado lógico (respecto a las variables físicas que queremos controlar).

$$\begin{aligned}\frac{V_s(s)}{V_e(s)} &= \frac{0,00544}{3,456 \cdot 10^{-5}s^2 + 0,002386s + 0,01207} \\ 0,00544V_e(s) &= 3,456 \cdot 10^{-5}V_s(s)s^2 + 0,002386V_s(s)s + 0,01207V_s(s) \\ 0,00544V_e(t) &= 3,456 \cdot 10^{-5}V_s''(t) + 0,002386V_s'(t) + 0,01207V_s(t)\end{aligned}\quad (27)$$

Seleccionamos como vectores de estado la velocidad y la aceleración.

$$\begin{aligned}x_1 &= V_s & \dot{x}_1 &= V_s' = x_2 \\ x_2 &= V_s' & \dot{x}_2 &= V_s''\end{aligned}\quad (28)$$

$$\begin{aligned}0,00544u &= 3,456 \cdot 10^{-5}\dot{x}_2 + 0,002386x_2 + 0,01207x_1 \\ \dot{x}_2 &= 157,4074u - 69,0394x_2 - 349,2477x_1\end{aligned}\quad (29)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -349,2477 & -69,0394 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 157,4074 \end{bmatrix} [u]\quad (30)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\quad (31)$$

```

1 a = 0.00544; b = 3.456e-5; c = 0.002386; d=0.01207;
2 Avelocidad = [0 1; -d/b -c/b]; Bvelocidad = [0;a/b]; Dvelocidad_discreto = 0;
   Cvelocidad_discreto = [1 0];
3 A_hat = [Avelocidad zeros(2,1); -Cvelocidad_discreto 0];
4 B_hat = [Bvelocidad; 0];
5 C_hat = [Cvelocidad_discreto 0];
6 P = [Avelocidad Bvelocidad; -Cvelocidad_discreto 0]
7 rank(P)
8 polos = [-10 -10 -40]
9 k_hat = acker(A_hat,B_hat,polos)
10 A_buclecerrado = A_hat-B_hat*k_hat
11 B_buclecerrado = [0 0 1]'
12 Gvelocidad_controlado = ss(A_buclecerrado,B_buclecerrado,C_hat,0)
13 eig(A_buclecerrado)
14 step(Gvelocidad_controlado)

```

Respuesta del sistema en tiempo continuo

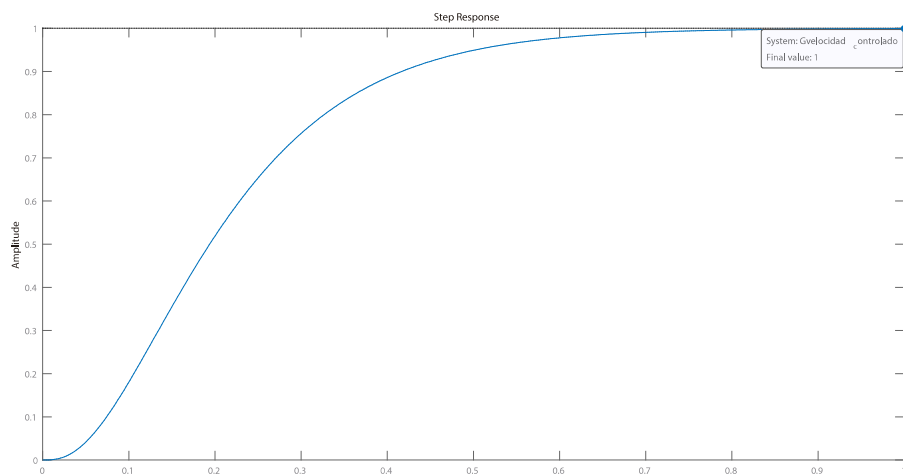


Figura 26: Respuesta del sistema con error nulo.

Actividad 17

Repetir el diseño en formato digital asumiendo un $T = 0,05$, partiendo de la función de transferencia discretizada del sistema y polos deseados obtenidos a partir de la relación $z = e^{sT}$.

```

1 a = 0.00544; b = 3.456e-5; c = 0.002386; d=0.01207;
2 Avelocidad = [0 1; -d/b -c/b]; Bvelocidad = [0;a/b]; Dvelocidad_discreto = 0;
   Cvelocidad_discreto = [1 0]; T = 0.05;
3 sys_discreto = c2d(ss(Avelocidad,Bvelocidad,Cvelocidad_discreto,Dvelocidad_discreto),T
   , 'zoh')
4 [Gvelocidad_discreto,Hvelocidad_discreto,Cvelocidad_discreto,Dvelocidad_discreto] =
   ssdata(sys_discreto)
5 G = [Gvelocidad_discreto [0 0]'; -Cvelocidad_discreto*Gvelocidad_discreto 1]
6 H = [Hvelocidad_discreto; -Cvelocidad_discreto*Hvelocidad_discreto];
7 P_discreto = [exp(-10*T) exp(-10*T) exp(-40*T)]
8 k_discreto = acker (G,H,P_discreto);
9 kd = [k_discreto(1) k_discreto(2)]; kid = -k_discreto(3);
10 GG = [Gvelocidad_discreto-Hvelocidad_discreto*kd Hvelocidad_discreto*kid; -
   Cvelocidad_discreto*Gvelocidad_discreto+Cvelocidad_discreto*Hvelocidad_discreto*kd
   1-Cvelocidad_discreto*Hvelocidad_discreto*kid]
11 HH = [0 0 1]';
12 CC = [Cvelocidad_discreto 0];
13 servo = ss(GG,HH,CC,0,T)
14 zpk(servo)
15 step(servo)

```

Respuesta del sistema en tiempo discreto

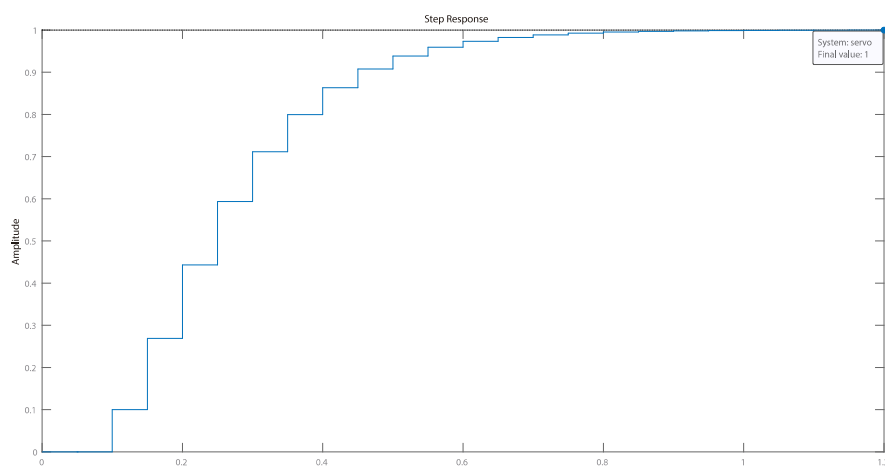


Figura 27: Respuesta del sistema con error nulo.

4. Código escrito en MATLAB

```

1  %%% Init
2  clc
3  clear all
4  clc
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  % Trabajo de control por computador
7  %%% Integrantes %%%
8  %%% Ana Rubio Bustos
9  %%% Gonzalo Guillamón Martín
10 %%% Jorge Benavides Macías
11 % Definimos las variables DNI Gonzalo
12 ki = 0.0544;
13 kb = 0.0444;
14 jep = 0.0064;
15 beq = 0.0344;
16 ra = 0.3438;
17 la = 0.0054;
18 kr = 3.8197;
19 kw = 1;
20 n = 0.1;
21 T = 0.05;
22 ka = 1;
23 %%% Actividad 2 %%%
24 % Función de transferencia en tiempo continuo posición y velocidad
25 Gposicion = tf([n*ki],[la*jep (la*beq)+(ra*jep) (ra*beq)+(n*ki*kb) 0]); %tipo 1
26 Gvelocidad = tf([n*ki],[la*jep (la*beq)+(ra*jep) (ra*beq)+(n*ki*kb)]); %tipo 0
27 %%% Actividad 3 %%%
28 % Función de transferencia en tiempo discreto para con tiempo de muestreo
29 % T = 0.05s.
30 Gposicionz = c2d(Gposicion,T,'zoh');
31 % Representación funciones de transferencia Continua - Discreta
32 figure
33 step(Gposicion)
34 hold on
35 step(Gposicionz)
36 xlim([0 1])
37 title('Función de transferencia continua vs discreta.')
38 legend('Función continua','Función discreta')
39 grid on
40 %%% Actividad 4 %%%
41 % Genera el modelo de espacios A,B,C,D para tiempo continuo
42 [A,B,C,D] = ssdata(ss(Gposicion));
43 % Genera el modelo de espacios A,B,Cz,Dz para tiempo discreto
44 [G,H,Cz,Dz,T]=ssdata(ss(Gposicionz));
45 %%% Actividad 5 %%%
46 Gposicion_discreto = c2d(Gposicion*kr,0.05,'zoh')
47 Gvelocidad_discreto = c2d(Gvelocidad*kw,0.05,'zoh')
48 %%% Actividad 6 %%%
49 % Representación en bucle cerrado para calcular las especificaciones de
50 % respuesta transitoria
51 transient_ans = kr*feedback(Gposicionz,kr)

```

```

52 %ltiview('step',transient_ans)
53 %%%%Actividad 7 %%%%
54 %rltool(Gposicionz);
55 k_ganancia_critica = margin(Gposicionz)
56 %%%%Actividad 8 %%%%
57 %rltool(Gposicionz);
58 %%%%Actividad 9 %%%%
59 %rltool(Gposicionz);
60 %Obtenemos el siguiente controlador:
61 PD_9 = zpk([0.7641],[0.009156],16.404,T)
62 ltiview('step',kr*feedback(PD_9*Gposicionz,kr))
63 %%%%Actividad 10 %%%%
64 PID_10 = zpk([0.9945 0.7613],[1 -0.5366],41.625,T)
65 %step
66 %ltiview('step',kr*feedback(PDI_10*Gposicionz,kr))
67 %rampa
68 t=0:0.05:25;
69 alpha=1;
70 rampa=alpha*t;
71 %ltiview('lsim',kr*feedback(PDI_10*Gposicionz,kr))
72 figure
73 stem(t,rampa);
74 hold on
75 [y,t]=lsim(kr*feedback(PID_10*Gposicionz,kr),rampa,t);
76 stem(t,y)
77 grid on
78 title('Funcionamiento del controlador PID')
79 legend('Función continua','Función discreta')
80 %%%%Actividad 11 %%%%
81 %Controlador PD - Actividad 9
82 PD_11 = pid(PD_9)
83 %Controlador PDI - Actividad 10
84 PID_11 = pid(PID_10)
85 %%%%Actividad 12 %%%%
86 %%%%Actividad 13 %%%%
87 [Kc,Pm,Wg]=margin(kr*Gposicion);
88 Tc=2*pi/Wg;
89 Kp=0.75*Kc;
90 Ti=Tc/1.6;
91 Ki=Kp/Ti;
92 Td=Tc/10;
93 Kd=Td*Kp;
94 tf=0.01;
95 pidZN=pid(Kp,Ki,Kd,tf);
96 rltool(Gposicion,pidZN);
97 pidZNz=c2d(pidZN,T,'trapezoidal');
98 rltool(Gposicionz,pidZNz);
99 %ya tenemos el pidZN que queremos con el criterio del 25%
100 %%%%Actividad 14 %%%%
101 %no se puede hacer el ziegler nichols en cerrado pero vamos a probar en
102 %abierto
103 [Kc,Pm,Wcg] = margin(Gvelocidad)

```

```

104 Gvelocidadz = c2d(Gvelocidad,0.05,'zoh')
105 step(Gvelocidadz)%Medimos t0.632 y T0.284 que corresponderan con T1 y T2
    respectivamente
106 T1 = 0.2;
107 T2 = 0.1;
108 d = (3*(T1-T2))/2;
109 tau = T1-d;
110 d/tau
111 %No podemos hacer en bucle abierto, ya que d/tau no esta en el intervalo
112 %0.15<d/tau<0.6
113 %%%%Actividad 15 %%%%
114 %%% Sintesis directa
115 step(Gvelocidad)
116 pole(Gvelocidad)
117 tau1 = 1/63.5465
118 tau2 = 1/5.4951
119 lamda = 6
120 k = 0.451;kp=(tau1+tau2)/(k*lamda);
121 ti = tau1+tau2;
122 td = (tau1*tau2)/(tau1+tau2);
123 PID = pidstd(kp,ti,td)
124 C = c2d(PID,0.05,'matched')
125 PID_ANA = pid(0.0731,0.37,0.00106,45)
126 %%% Controlador Rivera Morari
127 k=0.451
128 t632=0.119;t283=0.0773;
129 d=3*(t632-t283)/2;tau=t632-d;
130 0.2*tau
131 lamda=1;
132 kp=tau/(k*lamda);ti=tau;
133 PI=pidstd(kp,ti)
134 C=c2d(PI,0.05,'matched')
135 %%%%Actividad 16 %%%%
136 % Insertar las matrices obtenidas a mano
137 a = 0.00544; b = 3.456e-5;c = 0.002386;d=0.01207;
138 Avelocidad = [0 1;-d/b -c/b]; Bvelocidad = [0;a/b]; Dvelocidad_discreto = 0;
    Cvelocidad_discreto = [1 0];
139 A_hat = [Avelocidad zeros(2,1);-Cvelocidad_discreto 0];
140 B_hat = [Bvelocidad;0];
141 C_hat = [Cvelocidad_discreto 0];
142 P = [Avelocidad Bvelocidad;-Cvelocidad_discreto 0]
143 rank(P)
144 polos = [-10 -10 -40]
145 k_hat = acker(A_hat,B_hat,polos)
146 A_buclecerrado = A_hat-B_hat*k_hat;
147 B_buclecerrado = [0 0 1]';
148 Gvelocidad_controlado = ss(A_buclecerrado,B_buclecerrado,C_hat,0)
149 eig(A_buclecerrado)
150 step(Gvelocidad_controlado)
151 %%%%Actividad 17 %%%%
152 a = 0.00544; b = 3.456e-5;c = 0.002386;d=0.01207;

```

```

153 Avelocidad = [0 1;-d/b -c/b]; Bvelocidad = [0;a/b]; Dvelocidad_discreto = 0;
    Cvelocidad_discreto = [1 0]; T = 0.05;
154 sys_discreto = c2d(ss(Avelocidad,Bvelocidad,Cvelocidad_discreto,Dvelocidad_discreto),T
    , 'zoh')
155 [Gvelocidad_discreto,Hvelocidad_discreto,Cvelocidad_discreto,Dvelocidad_discreto] =
    ssdata(sys_discreto)
156 G = [Gvelocidad_discreto [0 0]';-Cvelocidad_discreto*Gvelocidad_discreto 1]
157 H = [Hvelocidad_discreto;-Cvelocidad_discreto*Hvelocidad_discreto];
158 P_discreto = [exp(-10*T) exp(-10*T) exp(-40*T)]
159 k_discreto = acker (G,H,P_discreto);
160 kd = [k_discreto(1) k_discreto(2)]; kid =-k_discreto(3);
161 GG = [Gvelocidad_discreto-Hvelocidad_discreto*kd Hvelocidad_discreto*kid;-
    Cvelocidad_discreto*Gvelocidad_discreto+Cvelocidad_discreto*Hvelocidad_discreto*kd
    1-Cvelocidad_discreto*Hvelocidad_discreto*kid]
162 HH = [0 0 1]';
163 CC = [Cvelocidad_discreto 0];
164 servo = ss(GG,HH,CC,0,T)
165 zpk(servo)
166 step(servo)

```