



| Carrera | Clave de la Asignatura | Nombre de la Asignatura | Curso |
|--|------------------------|-------------------------|--------|
| Ingeniería en Computación | 36284 | Sistemas de Control | 2021-2 |
| Alumno: Zavala Román Irvin Eduardo | | | |
| Práctica #3: Sistemas de Segundo orden | | | |

Objetivo:

El alumno conocerá y aplicará bloques básicos para la construcción de modelos en Simulink, además de distinguir distintos tipos de sistemas de segundo orden. Identificar conceptos de respuesta sobre amortiguada, sub amortiguada y críticamente amortiguada, así como el concepto de tiempo muerto.

Introducción:

Los sistemas de segundo orden son todos aquellos que tienen dos polos y están representados típicamente por ecuaciones diferenciales ordinarias de segundo orden.

La fórmula general para sistemas de segundo orden es:

$$\frac{X(s)}{F(s)} = K \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Donde X(s) es la salida, F(s) la entrada, K la ganancia estática del sistema, ω_n frecuencia natural no amortiguada del sistema y ζ factor de amortiguamiento. Dependiendo del valor de ζ sabremos el comportamiento del sistema:

$\zeta = 0$ Sistema oscilatorio (Polos imaginarios $\pm j\omega_n$)

$0 < \zeta < 1$ Sistema subamortiguado (Polos complejos conjugados con parte real negativa $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$)

$\zeta = 1$ Sistema críticamente amortiguado (2 polos iguales en el plano complejo $-\omega_n$)

$\zeta > 1$ Sistema sobre amortiguado (2 polos reales $-\zeta\omega_n \pm \omega_n\sqrt{\zeta^2-1}$)

**Material:**

- Lápiz y papel
- Equipo utilizado Equipo de cómputo con software Matlab

Desarrollo de la Práctica:

1. Construya un diagrama de bloques en Simulink para la implementación de un lazo de control; formado por un proceso y controlador, dados por $G_p(s)$, $G_d(s)$ y $G_c(s)$, como se muestra en la figura 3.1.

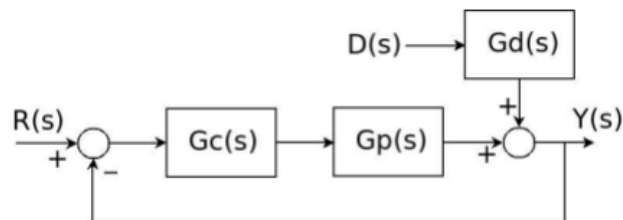


Figura 3.1 Diagrama de Bloques de lazo realimentado

En donde:

$$G_d = \frac{2}{7s + 1}$$

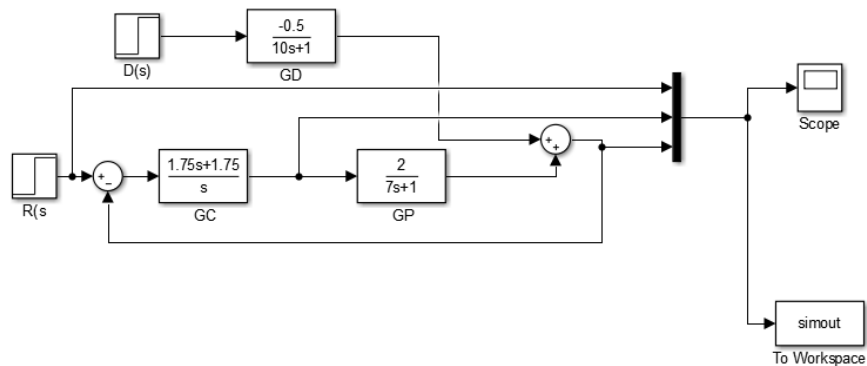
$$G_c = \frac{-0.5}{10s + 1}$$

$$G_p = \frac{1.75s + 1.75}{s}$$

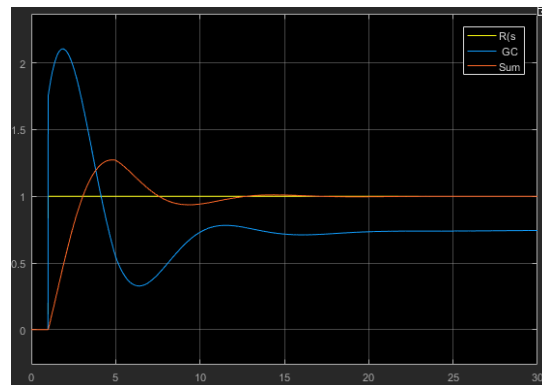


2. Simule un cambio de referencia $R(t)$ de 0 a 1 iniciando en $t=1$, además de la presencia de una perturbación $D(t)$ de 0 a 1 en $t=5$. El tiempo total de simulación del sistema de control deberá ser de 30 segundos. Grafique los resultados de simulación en Matlab.

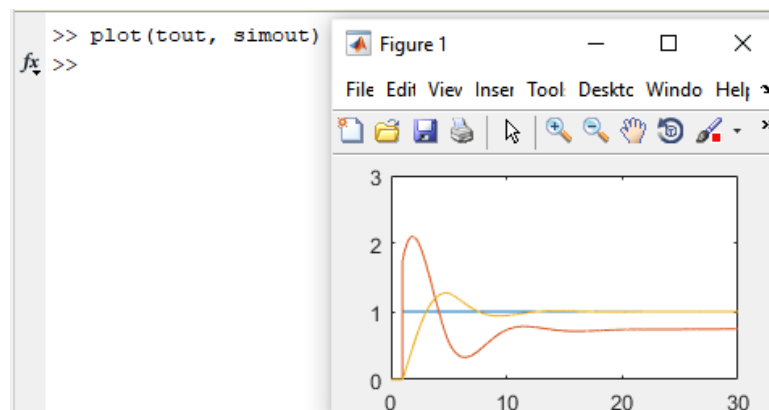
La implementación con las funciones dadas en Matlab queda así:



El resultado de la simulación en el scope es lo siguiente:



Y en Matlab:





3. Construya el modelo descrito en la figura 3.3, en Simulink.

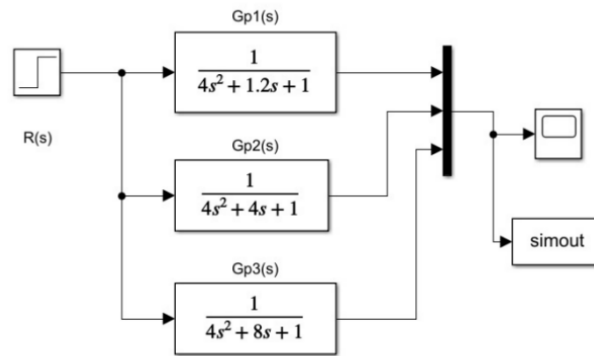
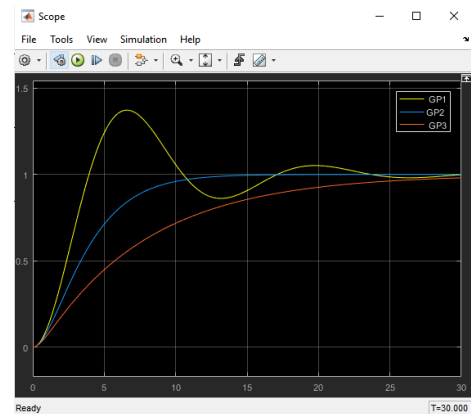
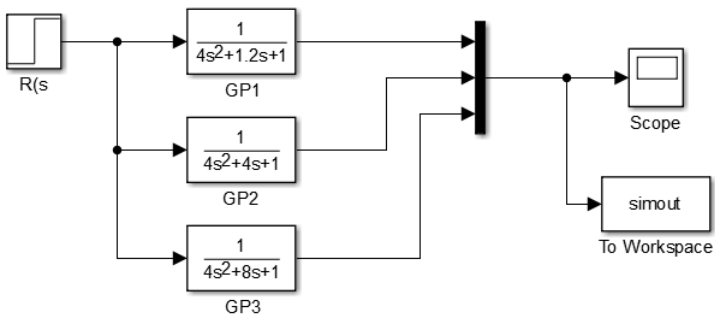


Figura 3.3

4. Utilice el diagrama de Simulink descrito en el punto anterior para comparar la respuesta de los tres tipos de sistemas de segundo orden ante una entrada escalón unitario.

La salida en el scope es la siguiente:





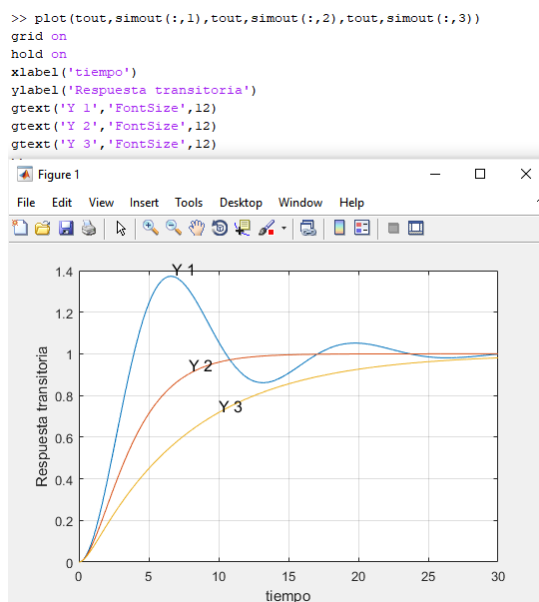
5. Calcule el discriminante (el discriminante se refiere al calculo que se realiza al aplicar la formula general, específicamente al valor que da $\sqrt{b^2 - 4ac}$), e identifique si es que corresponde a una respuesta: sub-amortiguada, sobre-amortiguada o críticamente amortiguada, identificando cada sistema del inciso anterior en la gráfica del SCOPE de Simulink.

En GP1 tenemos 2 raíces imaginarias con parte real negativa, lo que nos da un sistema subamortiguado. En GP2 tenemos un sistema con 2 raíces reales negativas iguales, dando un sistema críticamente amortiguado. En GP3 tenemos 2 raíces reales negativas diferentes, dando un sistema sobre amortiguado. Discriminantes : GP1 = 3.8157i, GP2 = 0, GP3 = 6.928.

6. Grafique los resultados de la simulación en Matlab, utilizando el siguiente código: *Indique que es lo que realiza, si es necesario realice los ajustes pertinentes*

```
plot(tout,simout(:,1),tout,simout(:,2),tout,simout(:,3))
grid on
hold on
xlabel('tiempo')
ylabel('Respuesta transitoria')
gtext('Y 1','fontsize',12)
gtext('Y 2','fontsize',12)
gtext('Y 3','fontsize',12)
```

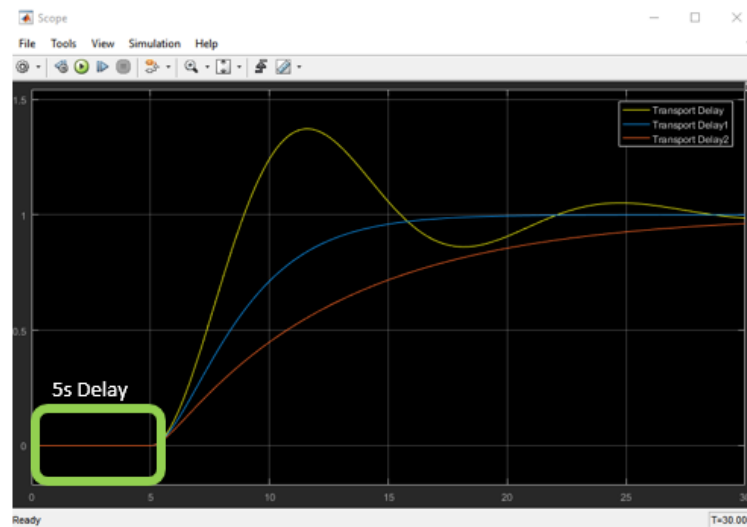
El código dado no funciona en Matlab 2016a, esto porque 'fontsize' debe ser 'FontSize'. Lo que hace el código es graficar el resultado del diagrama 3.3, los gtext colocan texto en la gráfica al dar click izquierdo. El resultado es el siguiente:



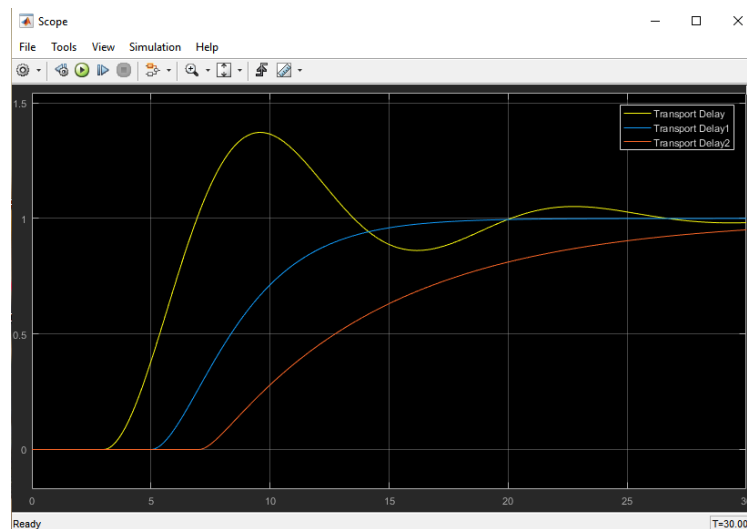


7. El diagrama de Simulink de la figura 3.4, muestra la manera de incluir un tiempo muerto a la respuesta del sistema. Compare la respuesta de los sistemas con tiempo muerto y sin tiempo muerto.

Puedo poner el mismo tiempo muerto en los 3 bloques, por ejemplo, si el delay es de 5 segundos la salida es la siguiente:



También puedo poner tiempos muertos diferentes en los bloques, por ejemplo, en GP1 pongo 3s, en GP2 5s y en GP3 7s.





Conclusiones:

Con Matlab para saber el amortiguamiento de un sistema tenemos bastantes opciones, desde implementarlo en simulink y usar las herramientas de scope/simout para poder manejar las salidas de los diagramas hasta cosas más sencillas, como calcular los polos y dependiendo de los resultados nos dice el comportamiento del sistema, y esto en Matlab es fácil de implementar con pzmap.

Bibliografía:

dademuch. (7 de Mayo de 2020). *Sistemas de segundo orden – Sistemas de control*. Obtenido de <https://dademuch.com/2020/05/07/sistema-de-segundo-orden/>

Giraldo, S. A. (s.f.). *Sistemas de Segundo Orden*. Obtenido de <https://controlautomaticoeducacion.com/control-realimentado/sistemas-de-segundo-orden/>