

Prácticas de Laboratorio de Control Avanzado

2022

ÍNDICE GENERAL

1. Práctica 1: Comandos básicos y diagramas de bloques en MATLAB®	1
1.1. Objetivos	1
1.1.1. Objetivo general	1
1.1.2. Objetivos específicos	1
1.2. Ejercicios	1
1.3. Conclusiones y Recomendaciones	3
2. Práctica 2: Transformada Z y sus propiedades	4
2.1. Objetivos	4
2.1.1. Objetivo general	4
2.1.2. Objetivos específicos	4
2.2. Introducción	4
2.3. Ejercicios	5
2.3.1. Discretización de un sistema continuo y su transformada Z	5
2.3.2. Transformada Z de un sistema discreto y transformada Z inversa	6
2.3.3. Superposición	7
2.4. Conclusiones y Recomendaciones	7
3. Práctica 3: Teorema de muestreo	8
3.1. Objetivos	8
3.1.1. Objetivo general	8
3.1.2. Objetivos específicos	8
3.2. Introducción	8
3.3. Especificaciones del experimento	9
3.4. Procedimiento	10
3.5. Conclusiones y Recomendaciones	13
4. Práctica 4: Discretización de controladores PID analógicos-Simulación	14
4.1. Objetivos	14
4.1.1. Objetivo general	14
4.1.2. Objetivos específicos	14
4.2. Introducción	14
4.3. Especificaciones del experimento	16
4.4. Procedimiento	16
4.5. Conclusiones y Recomendaciones	17

5. Práctica 5: Discretización controladores PID analógicos-Planta	21
5.1. Objetivos	21
5.1.1. Objetivo general	21
5.1.2. Objetivos específicos	21
5.2. Conclusiones y Recomendaciones	21
6. Práctica 6: Correspondencia entre plano s y z	22
6.1. Objetivos	22
6.1.1. Objetivo general	22
6.1.2. Objetivos específicos	22
6.2. Introducción	22
6.3. Ejercicios	23
6.3.1. Ejercicio 1	23
6.3.2. Ejercicio 2	24
6.3.3. Ejercicio 3	24
6.3.4. Sisotool	25
6.4. Conclusiones y Recomendaciones	25
7. Práctica 7: Diseño de controladores discretos usando lugar geométrico de las raíces-Simulación	26
7.1. Objetivos	26
7.1.1. Objetivo general	26
7.1.2. Objetivos específicos	26
7.2. Introducción	26
7.3. Especificaciones del experimento	27
7.4. Procedimiento	28
7.5. Conclusiones y Recomendaciones	29
8. Práctica 8: Diseño controlador discreto usando lugar geométrico de las raíces-Planta	30
8.1. Objetivos	30
8.1.1. Objetivo general	30
8.1.2. Objetivos específicos	30
8.2. Especificaciones	30
8.3. Procedimiento	30
8.4. Conclusiones y Recomendaciones	31
9. Práctica 9: Representación de sistemas en variables de estado	32
9.1. Objetivos	32
9.1.1. Objetivo general	32
9.1.2. Objetivos específicos	32
9.2. Ejercicio	32
9.3. Procedimiento	33
9.4. Conclusiones y Recomendaciones	33
10. Práctica 10: Controlabilidad y Observabilidad	34
10.1. Objetivos	34
10.1.1. Objetivo general	34
10.1.2. Objetivos específicos	34
10.2. Procedimiento	34

10.2.1. Controlabilidad	35
10.2.2. Observabilidad	36
10.3. Conclusiones y Recomendaciones	37
11. Práctica 11: Diseño de controlador en variables de estado - Simulación	38
11.1. Objetivos	38
11.1.1. Objetivo general	38
11.1.2. Objetivos específicos	38
11.2. Introducción	38
11.3. Procedimiento	40
11.4. Conclusiones y Recomendaciones	41

Práctica 1: Comandos básicos y diagramas de bloques en MATLAB[®]

1.1. Objetivos

1.1.1. Objetivo general

Realizar ejercicios de comandos básicos y diagramas de bloques en MATLAB[®] aplicados a la ingeniería de Control para reforzar los conocimientos aprendidos en cursos anteriores.

1.1.2. Objetivos específicos

- Practicar el uso de comandos de reducción y representación de sistemas.
- Representar sistemas usando diagramas de bloques en Simulink.
- Reforzar el uso de comandos de representación gráfica y otros comandos de utilidad para el desarrollo del curso.

1.2. Ejercicios

Cree un script en MATLAB[®] y coloque su nombre al inicio del mismo. Recuerde utilizar el símbolo % para comentar las líneas de código que considere necesarias y utilizar %% para separar secciones de código. Resuelva en dicho script los ejercicios que se presentan a continuación y presente sus resultados en el *formato* proporcionado en el **blog** del curso. Recuerde utilizar el comando **help** para obtener ayuda y ejemplos de los comandos que requiera.

1. Ingrese las siguientes funciones de transferencia usando el comando **tf**. Haga uso del comando **conv** en caso de ser necesario.

a) $G_1(s) = \frac{10}{s^2 + 7s + 10}$

b) $G_2(s) = \frac{5}{s^3 + 15s^4 + 30s}$

c) $G_3(s) = \frac{2(s + 3)}{(s + 1)(s + 6)(s + 8)}$

d) $G_4(s) = \frac{8s^2}{s^3 + 18s}$

2. Ingrese las siguientes funciones de transferencia utilizando el comando **zpk**. Haga uso del comando **roots** en caso de ser necesario.

a) $G_5(s) = \frac{2(s + 3)}{(s + 1)(s + 6)(s + 8)}$

b) $G_6(s) = \frac{5}{(s + 2)(s + 4)}$

c) $G_7(s) = \frac{3s + 6}{(s + 5)(s + 2)}$

d) $G_8(s) = \frac{10s}{s^2 + 7s + 10}$

3. Reduzca el siguiente sistema usando los comandos **series**, **parallel** y **feedback**. Recuerde utilizar el comando **minreal** sobre los resultados obtenidos en cada reducción.

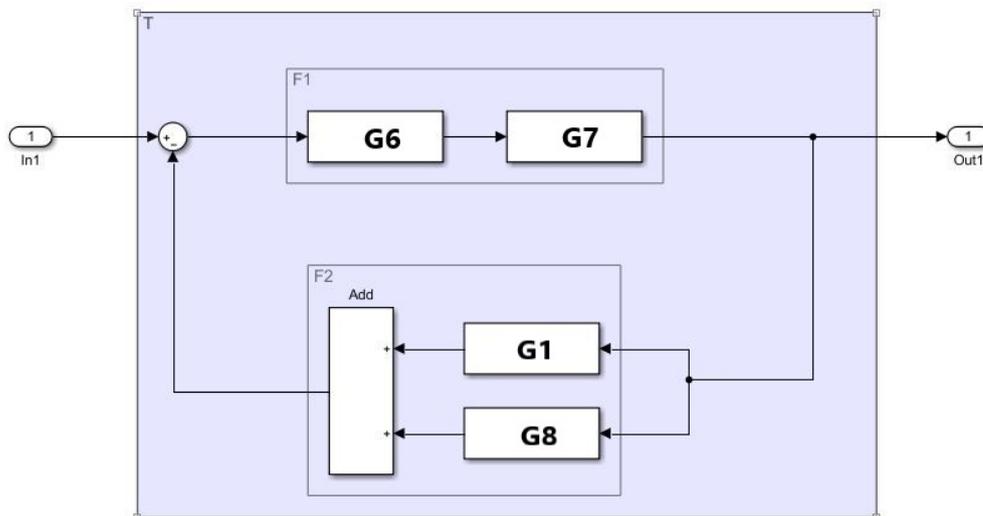


Figura 1.1: Ejercicio 3: Diagrama de bloques

4. Obtenga la respuesta escalón del sistema F1 usando el comando **step**. Recuerde mostrar las características del sistema y activar la cuadrícula. Puede usar el comando **stepinfo** para observar las características de la respuesta escalón.

5. Obtenga los polos y ceros del sistema T usando los comandos **pole** y **zero** respectivamente.

6. Obtenga nuevamente T asumiendo realimentación positiva (T_2).
7. En Simulink, cree un modelo que represente el siguiente grupo de ecuaciones y realice lo que se solicita a continuación.

$$20 \frac{dc(t)}{dt} = -8c(t) + \sqrt{v(t)} + 10 \frac{dv(t)}{dt}$$

$$\frac{d^2v(t)}{dt^2} = r^2(t) - 6v(t) - 5 \frac{dv(t)}{dt}$$

- Arme el diagrama de bloques del sistema de manera que no utilice bloques derivadores y represente a las ecuaciones dadas. Considere como entrada $r(t)$ y como salida $c(t)$. Configure la simulación de manera que el tiempo de muestreo de MATLAB[®] sea igual a T .
 - La entrada del sistema R es excitada con una señal escalón que inicialmente toma un valor de $r_i = 1$ y en $t = 20[s]$ cambia a $r_f = 1.5$. Coloque y configure el bloque **Step** para que cumpla lo solicitado.
 - Coloque el bloque **Mux** para observar de manera simultánea la entrada y salida del sistema.
 - Coloque el bloque **To Workspace** y/o el bloque **Scope** para exportar los datos de la simulación al espacio de trabajo (*Workspace*) y para visualizar las señales generadas. En este modelo asigne el nombre **modelo1** a la variable a utilizar para guardar los datos, ya sea con el bloque To Workspace o con el bloque Scope. Recuerde guardar la variable en formato Array.
8. Realice la simulación del modelo creado a través del comando **sim** con $T = 0.1[s]$ y $T = 5[s]$. Utilice un tiempo de simulación de 50 segundos.
9. Con los datos obtenidos de cada simulación, obtenga las siguientes gráficas. Incluya cuadrícula, título, nombre de los ejes y leyendas en cada gráfica. Recuerde que las gráficas solicitadas deben ser realizadas en función del tiempo.
- Una figura con las señales $c(t)$ obtenidas con cada tiempo de muestreo superpuestas. Puede utilizar representación matricial o el comando **hold on**.
 - Una figura con dos gráficas. En la gráfica de la izquierda muestre $r(t)$ y en la de la derecha muestre $v(t)$. Utilice el comando **subplot** para cumplir esta tarea.

1.3. Conclusiones y Recomendaciones

Práctica 2: Transformada Z y sus propiedades

2.1. Objetivos

2.1.1. Objetivo general

Realizar ejercicios referentes a la transformada Z usando MATLAB[®] para la comprobación de algunas de sus propiedades y la simplificación del cálculo de la transformada Z inversa.

2.1.2. Objetivos específicos

- Usar comandos de MATLAB[®] para la discretización y cálculo de la transformada Z de un sistema continuo.
- Usar comandos de MATLAB[®] para el cálculo de la transformada Z inversa de un sistema discreto.
- Comprobar la propiedad de superposición de manera gráfica a través de simulaciones en Simulink.

2.2. Introducción

De manera análoga a la transformada de Laplace, la transformada Z permite convertir una señal definida en el dominio del tiempo discreto en una representación en el dominio de la frecuencia. En ingeniería de control, se utiliza principalmente para obtener la función de transferencia de un sistema a partir de las ecuaciones en diferencia que lo representan. A su vez, a partir de dichas funciones de transferencia se puede hallar la respuesta del sistema ante diferentes entradas de manera más sencilla que al usar las ecuaciones en diferencia del mismo.

Al ser la transformada Z un operador lineal, existen diversas propiedades que facilitan el uso de la misma. Las más representativas para este curso son las siguientes:

- Linealidad

$$\mathcal{Z}\{a_1x_1[n] + a_2x_2[n]\} = a_1X_1(z) + a_2X_2(z)$$

- Desplazamiento en el tiempo

$$\mathcal{Z}\{x[n-k]\} = z^{-k}X(z)$$

- Convolución

$$\mathcal{Z}\{x_1[n] * x_2[n]\} = X_1(z)X_2(z)$$

- Primera diferencia

$$\mathcal{L}\{x[n] - x[n-1]\} = (1 - z^{-1})X(z)$$

- Acumulación

$$\mathcal{L}\left\{\sum_{k=0}^n x[k]\right\} = (1 - z^{-1})^{-1}X(z)$$

En esta práctica, se hará uso de MATLAB[®] para hallar de manera más rápida la transformada Z y la transformada Z inversa de distintas señales y sistemas. Además se verificará la propiedad de linealidad de manera gráfica a través de Simulink.

2.3. Ejercicios

Cree un script en MATLAB[®] y coloque su nombre al inicio del mismo. Recuerde utilizar el símbolo % para comentar las líneas de código que considere necesarias y utilizar %% para separar secciones de código. Resuelva en dicho script los ejercicios que se presentan a continuación y presente sus resultados en el *formato* proporcionado en el **blog** del curso. Recuerde utilizar el comando **help** para obtener ayuda y ejemplos de los comandos que requiera.

2.3.1. Discretización de un sistema continuo y su transformada Z

1. Ingrese las siguientes funciones de transferencia continuas. Para el literal a, ingrese el retardo agregando 'InputDelay' seguido del tiempo de retardo luego de los argumentos correspondientes al comando que utilice. Por ejemplo:

$$tf(num,den,'InputDelay', Retardo)$$

$$a) G_2(s) = \frac{5e^{-0.5s}}{s + 0.01}$$

$$b) G_4(s) = \frac{4s}{(s + 8)(s + 10)}$$

2. Utilice el comando **c2d** para encontrar las funciones de transferencia discretas equivalentes de los sistemas del literal anterior. Utilice los tiempos de muestreo que se especifican a continuación para cada caso y compare sus respuestas escalón.

$$a) T=0.005 \text{ y } T=0.25$$

$$b) T=0.1 \text{ y } T=0.001$$

3. Verifique el cumplimiento de la propiedad de desplazamiento en el tiempo para el sistema del literal a teniendo en cuenta el retenedor de orden cero implícito en la conversión. Conteste las siguientes preguntas.

a) ¿Cómo afecta el tiempo de muestreo al orden de la función de transferencia discretizada del literal a?

b) ¿Qué sucedería si se muestrea con T=0.4? Compruebe a través de una prueba escalón si se cumple lo esperado.

4. Ingrese las siguientes funciones de transferencia discretas usando los tiempos de muestreo indicados.

$$a) G_1z(z) = \frac{5z^{-1}}{z+0.5}; T = 0.2$$

$$b) G_2z(z) = \frac{10}{z^2 + 0.6z + 0.05}; T = 0.5$$

$$c) G_3z(z) = \frac{3z+1}{(z-0.1)(z+0.3)}; T = 1$$

$$d) G_4z(z) = \frac{7z}{(z-1)(z-0.5)}; T = 0.1$$

5. Obtenga la representación continua equivalente de las funciones ingresadas en el literal anterior haciendo uso del comando **d2c**.

2.3.2. Transformada Z de un sistema discreto y transformada Z inversa

1. Usando el comando **syms** declare la variable independiente discreta n, la variable de muestreo T. Adicionalmente declare la variable z.

2. Ingrese las siguientes expresiones discretas.

$$a) y_1 = \sin(5nT)$$

$$b) y_2 = 2 * nT$$

$$c) y_3 = e^{-3nT}$$

3. Encuentre la transformada Z de las expresiones anteriores usando el comando **ztrans**.

4. Encuentre la transformada Z inversa de las siguientes expresiones usando el comando **iztrans**.

$$a) yz_1 = \frac{z}{(z-1)^2}$$

$$b) yz_2 = \frac{z}{z - e^{-5}}$$

2.3.3. Superposición

1. Cree el modelo en Simulink que se muestran a continuación. Siga las configuraciones indicadas.

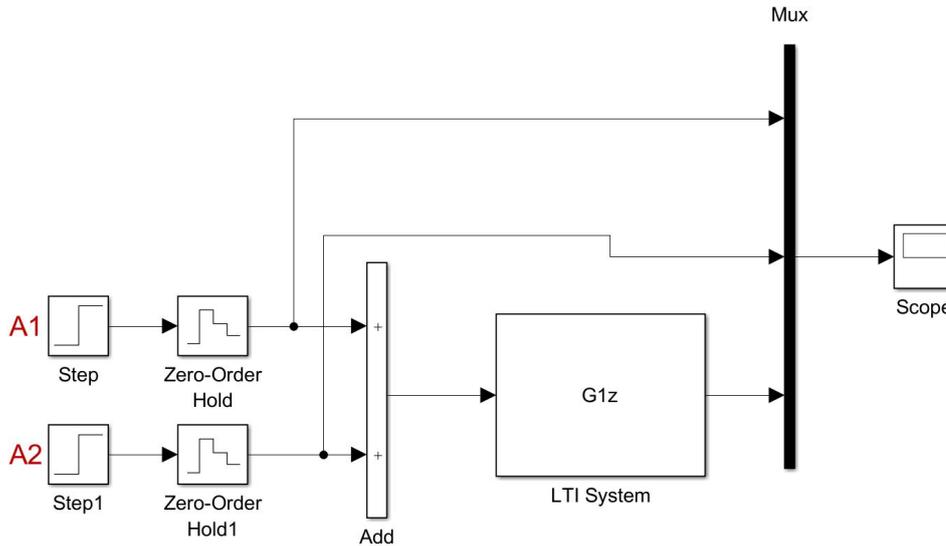


Figura 2.1: Diagrama de bloques a utilizar

- Configure los bloques Zero-Order Hold con tiempo de muestreo igual a 0.2 segundos.
 - Configure el bloque Step A1 con tiempo de paso de 0 segundos, valor inicial igual a 0 y valor final igual a a_1 .
 - Configure el bloque Step A2 con tiempo de paso de 5 segundos, valor inicial igual a 0 y valor final igual a a_2 .
 - Configure el bloque Scope de manera que le permita guardar los datos de la simulación en el espacio de trabajo como un arreglo.
 - El tiempo de simulación del sistema es de 10 segundos.
2. Asigne a $a_1 = 1$ y a $a_2 = 0$; realice la simulación y guarde estos datos como A1G.
 3. Asigne a $a_1 = 0$ y a $a_2 = 1$; realice la simulación y guarde estos datos como A2G.
 4. Asigne a $a_1 = 1$ y a $a_2 = 1$; realice la simulación y guarde estos datos como A12G.
 5. Cree las figuras y gráficas solicitadas a continuación y comente los resultados obtenidos. Debe incluir títulos, cuadrículas, nombre a los ejes y leyendas en todas las gráficas; **todas** las gráficas se realizan en función del tiempo.
 - a) Con los datos A1G muestre en **una** subfigura las dos entradas superpuestas en una gráfica y la salida obtenida en la otra gráfica.
 - b) Repita lo misma para A2G.
 - c) En una nueva subfigura muestre la suma de la salida A1G y de la salida A2G en una gráfica y la salida de A12G en otra gráfica.

2.4. Conclusiones y Recomendaciones

Práctica 3: Teorema de muestreo

Nombre:

Paralelo:

Fecha:

3.1. Objetivos

3.1.1. Objetivo general

Realizar el proceso de muestreo y análisis espectral de las señales obtenidas usando diferentes tiempos de muestreo mediante una simulación en Simulink de MATLAB[®] como demostración de la efectividad del teorema de muestreo.

3.1.2. Objetivos específicos

- Armar en Simulink el diagrama de bloques del proceso de muestreo reconociendo cada elemento a utilizar.
- Comprobar la efectividad del teorema de muestreo bajo distintos escenarios.
- Realizar el análisis espectral para cada uno de los escenarios propuestos.

3.2. Introducción

Todo lo que nos rodea en el mundo es análogo y las señales que percibimos de nuestro entorno varían en el tiempo que es una variable continua cuyo estudio es de suma importancia. Sin embargo, para realizar análisis y cálculos de ciertos modelos se necesita la toma de muestras, ya que realmente las computadoras trabajan de manera discreta. La velocidad o tasa de muestreo de las computadoras utilizadas actualmente hace que las muestras tomadas se asemejen a una señal continua.

Además, aunque se tenga una cantidad limitada de datos luego de muestrear un sistema, se desea tener disponible información del sistema en todo instante de tiempo mientras dure la toma de datos. Para lograr esto se utiliza un retenedor de orden cero (ZOH, por sus siglas en inglés). La estructura de lo mencionado se ilustra en la siguiente figura.

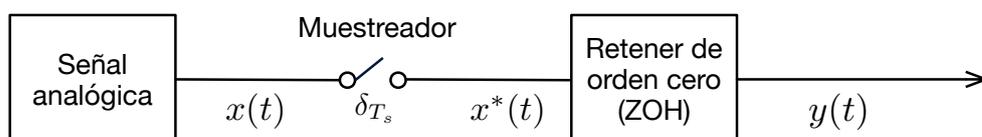


Figura 3.1: Diagrama de un muestreador con retenedor

El retenedor de orden cero ayuda a que el valor de cada muestra se mantenga en la salida hasta recibir la siguiente muestra; actúa como un filtro pasa bajo aunque su respuesta en frecuencia no es la ideal como se observa en la siguiente figura.

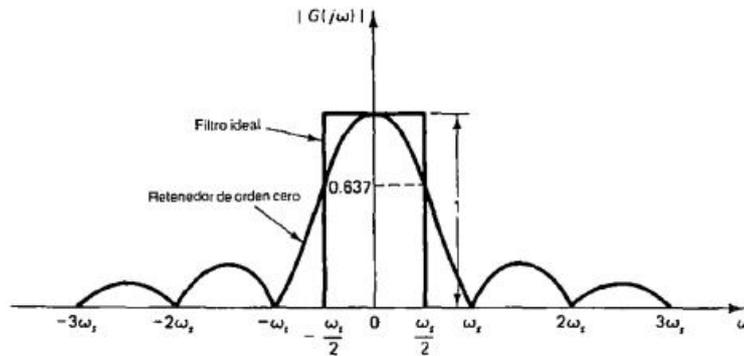


Figura 3.2: Respuesta en frecuencia de la función ZOH

El periodo de muestreo tiene que tener una relación con la frecuencia propia de la señal original. El teorema del muestreo se satisface con la condición:

$$\omega_s > 2\omega_1$$

donde $\omega_s = 2\pi/T_s$, y ω_1 es la máxima frecuencia presente en la señal original $x(t)$.

Al muestrear una señal, en el dominio de la frecuencia ocurren repeticiones de su respuesta en frecuencia desfasadas ω_s de la respuesta en frecuencia original. En caso de no cumplir con el teorema de muestreo y usar $\omega_s < 2\omega_1$, ocurre el efecto de **aliasing**. Aliasing es la superposición de señales en la frecuencia; al tratar de reconstruir la señal original a partir de las muestras obtenidas, este efecto no permite obtener dicha señal correctamente.

3.3. Especificaciones del experimento

- La señal de entrada es

$$x(t) = A_1 \text{Sen}(\omega_1 t)$$

Esta señal tiene excitación únicamente en la frecuencia ω_1

- El periodo de muestreo T_s debe ser variable y manipulado por el usuario.
- El muestreador se debe representar por un tren de impulsos unitario con periodo T_s .

Se debe generar tres escenarios de simulación mediante la selección de valores apropiados para el periodo de muestreo T_s :

- Muestreo suficiente: El teorema del muestreo se satisface de manera amplia ($\omega_s = 10(2\omega_1)$).
- Muestreo ajustado: El teorema del muestreo se satisface de manera exacta o ajustada ($\omega_s \approx 2\omega_1$).
- Muestreo insuficiente: El teorema del muestreo no se satisface.

3.4. Procedimiento

Cree un script en MATLAB[®] y coloque su nombre al inicio del mismo. Recuerde utilizar el símbolo % para comentar las líneas de código que considere necesarias y utilizar %% para separar secciones de código. Realice en dicho script el procedimiento que se muestra a continuación y presente sus resultados en el *formato* proporcionado en el **blog** del curso. Recuerde utilizar el comando **help** para obtener ayuda y ejemplos de los comandos que requiera.

1. Arme el esquema que se muestra en la siguiente figura

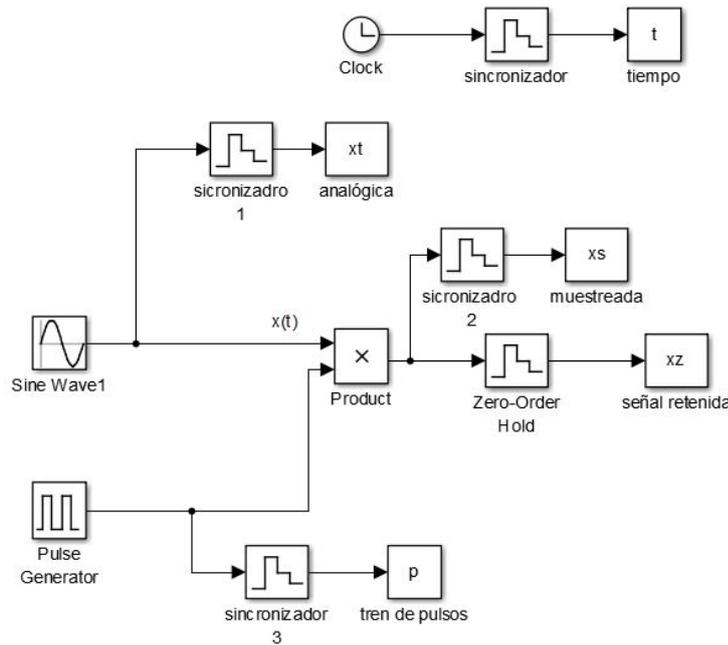


Figura 3.3: Esquema en Simulink

2. Configure Simulink de manera que el tiempo de muestreo para el Solver sea bajo. Para esto, dé click en el botón  o seleccione del menú Simulation la opción Model Configuration Parameters. Configure la ventana que aparece de manera que el tiempo máximo del Solver sea de 0.01 tal como se muestra a continuación.



Figura 3.4: Ventana Solver

3. Configure el bloque **Sine Wave** tal como se muestra en la siguiente figura.

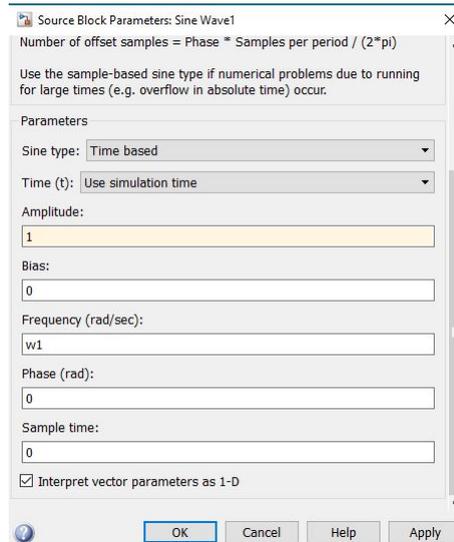


Figura 3.5: Parámetros del bloque Sinusoidal

4. Configure el bloque **Pulse Generator** de acuerdo a la siguiente figura.

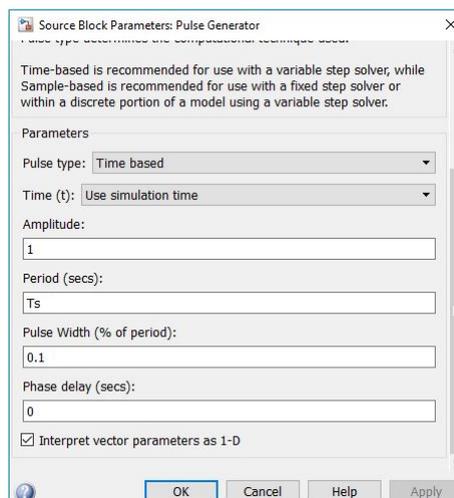


Figura 3.6: Parámetros del bloque Generador de Pulsos

5. Configure los bloques sincronizadores con tiempo de muestreo rTs . De esta manera todos los vectores generados durante la simulación tendrán igual número de elementos. Si coloca diferentes valores en los sincronizadores, corre el riesgo de obtener errores durante la simulación.
6. Configure el bloque **Zero-Order Hold** con la variable Ts como tiempo de muestreo.
7. Tenga en cuenta que debe guardar las variables de los bloques **To Workspace** como Array, ya que por default estos están configurados como estructura.

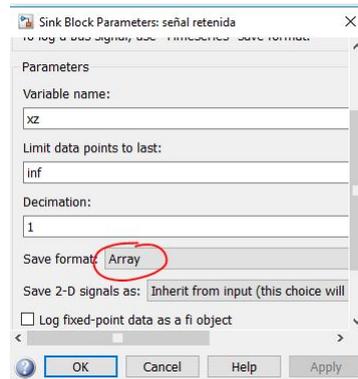


Figura 3.7: Parámetros del bloque To Workspace

8. Cree un archivo .m en MATLAB[®] donde deberá registrar los datos del experimento y el código a utilizar para obtener lo solicitado en la práctica. Los datos del experimento son los siguientes:

- $\omega_1 = 1$: frecuencia de la señal sinusoidal
- $T_{sim} = 200$ [s] : tiempo de simulación del experimento de MATLAB[®]
- $t_{max} = 20$ [s] : tiempo máximo a mostrar en las gráficas a obtener
- $w_{max} = 15 * \omega_1$: frecuencia máxima a visualizar en las gráficas a obtener
- T_s : tiempo de muestreo
 - $T_s = 1$ muestreo ajustado
 - $T_s = 0.5$ muestreo suficiente
 - $T_s = 4$ muestreo insuficiente
 - $T_s = 3$ frecuencia múltiplo (Aliasing)
- $rT_s = 0.01$: tiempo de muestreo de MATLAB[®]
- $\omega_s = \frac{2 * \pi}{T_s}$
- $d\omega_1 = 2 * \omega_1$:Aplicando el teorema de muestreo

9. Use el comando **sim** para simular el modelo creado desde el script especificando el tiempo de simulación T_{sim} .

10. Cree el vector de tiempo entre 0 y T_{sim} con pasos de T_s segundos. Luego interpole los datos de la señal retenida xz usando el siguiente comando:

```
xz2=interp1(tz,xz,t,'previous');
```

11. Grafique las señales obtenidas usando subplot. En la figura superior use **stem** para visualizar la señal del muestreador p . En la figura de la mitad grafique la señal analógica x_t usando plot y la señal muestreada x_s usando stem; recuerde incluir leyenda. En la figura inferior grafique la señal xz_2 usando el comando plot.

Recuerde incluir título y cuadrícula en cada subplot; además limite el eje x según la variable **tmax**.

12. Obtenga el rango de frecuencias a utilizar para el análisis espectral de las señales obtenidas. Las frecuencias a utilizar dependen de la cantidad de muestras N y del tiempo de muestreo rT_s . Se utiliza la siguiente ecuación, donde k varía entre 1 y N .

$$\omega_k = \frac{2 * \pi * (k - 1)}{(rT_s * N)};$$

13. Obtenga la transformada de Fourier de la señal analógica x_t , de la señal muestreada x_s y de la señal retenida e interpolada x_{z2} usando el comando **fft**.
14. Grafique las magnitudes de lo obtenido en el punto anterior utilizando subplot. Para obtener las magnitudes use el comando **abs**. En la imagen superior, muestre el espectro de frecuencias obtenido para la señal analógica, en la imagen de la mitad muestre el espectro de frecuencias de la señal muestreada y en la imagen inferior muestre el espectro de frecuencias de la señal retenida e interpolada.
- Recuerde incluir título y cuadrícula en cada subplot; además limite el eje x según la variable **wmax**.
15. Repita el procedimiento desde el paso 9 para los otros tiempos de muestreo sugeridos.

3.5. Conclusiones y Recomendaciones

Práctica 4: Discretización de controladores PID analógicos-Simulación

4.1. Objetivos

4.1.1. Objetivo general

Analizar el desempeño de un controlador PID discretizado con diferentes tiempos de muestreo mediante simulaciones en Simulink de MATLAB[®] para el estudio del efecto del muestreo en un sistema de control.

4.1.2. Objetivos específicos

- Discretizar un controlador PID utilizando diferentes tiempos de muestreo.
- Obtener los nuevos parámetros del controlador PID digital a partir del analógico.
- Simular el sistema en estudio con cada controlador digital y analizar el desempeño de estos.

4.2. Introducción

Un controlador PID analógico puede ser descrito a través de la ecuación 4.1. El término proporcional corresponde a K , el término integral corresponde a $\frac{K}{T_i}$ y el término derivativo corresponde a KT_d . El controlador PID digital puede ser obtenido al discretizar dicha ecuación.

$$m(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (4.1)$$

El término diferencial discretizado se puede representar de manera aproximada como se observa en la ecuación 4.2.

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{T} \quad (4.2)$$

El término integral discretizado puede ser representado de manera aproximada de dos formas. En la **forma regular** se utiliza la ecuación 4.3 mientras que en la **forma trapezoidal** se utiliza la relación de la ecuación 4.4. La forma trapezoidal sigue una aproximación del término integral más exacta.

Forma regular:

$$\int_0^t e(t') dt' \approx \sum_{h=1}^k e_h T \quad (4.3)$$

Forma trapezoidal:

$$\int_0^t e(t') dt' \approx T \sum_{h=1}^k \frac{e_h + e_{h-1}}{2} \tag{4.4}$$

Finalmente, existen dos representaciones para el controlador PID discretizado: la forma posicional y la forma de velocidad. Sin embargo, se revisará para esta práctica únicamente la forma posicional.

- **Controlador PID discreto en forma posicional:** está representado por la ecuación 4.5 en el caso de la forma regular y por la ecuación 4.6 en el caso de la forma trapezoidal.

$$m(kT) = m_k = \tilde{m} + K \left[e_k + \frac{T}{T_i} \sum_{h=1}^k e_h + \frac{T_d}{T} (e_k - e_{k-1}) \right] \tag{4.5}$$

$$m(kT) = m_k = \tilde{m} + K \left[e_k + \frac{T}{2T_i} \sum_{h=1}^k e_h + \frac{T}{2T_i} \sum_{h=0}^{k-1} e_h + \frac{T_d}{T} (e_k - e_{k-1}) \right] \tag{4.6}$$

La transformada Z de dichas expresiones se muestra en la ecuación 4.7, donde el término integral $K_{ip} = \frac{KT}{T_i}$ y el término diferencial $K_{dp} = \frac{KT_d}{T}$ son válidos para cualquiera de las dos formas. El término proporcional $K_{pp} = K$ para la forma regular y $K_{pp} = K - \frac{K_{ip}}{2}$ para la forma trapezoidal.

$$M(z) = \left[K_{pp} + \frac{K_{ip}}{1 - z^{-1}} + K_{dp}(1 - z^{-1}) \right] E(z) \tag{4.7}$$

La arquitectura de control utilizada comúnmente para controladores PID discretos o digitales se muestra en la figura 4.1.

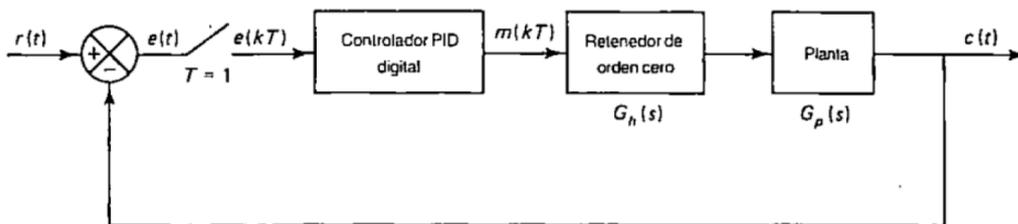


Figura 4.1: Arquitectura de sistema de control a utilizar

4.3. Especificaciones del experimento

Para este experimento, se conoce la función de transferencia de una planta y de un controlador PID analógico diseñado para la misma. Se requiere encontrar el equivalente discreto de dicho controlador para distintos tiempos de muestreo tanto para la forma posicional como para forma de velocidad. Luego se requiere simular el sistema de control discreto con cada controlador obtenido y analizar el desempeño obtenido en cada caso.

Los parámetros conocidos del experimento son los siguientes:

- La función de transferencia de la planta es:

$$G(s) = \frac{2}{(s+2)(s+5)}$$

- La función de transferencia del controlador analógico es:

$$G_c(s) = \frac{0.5833(s+2)(s+30)}{s}$$

- Los tiempos de muestreo a utilizar son $T = 0.02$, $T = 0.05$ y $T = 0.2$ segundos.
- El tiempo de simulación (T_{sim}) a utilizar es de 10 segundos.
- Parametrizar los bloques que representen a los controladores PID y los bloques retenedores o muestreadores.

4.4. Procedimiento

Cree un script en MATLAB[®] y coloque su nombre al inicio del mismo. Recuerde utilizar el símbolo % para comentar las líneas de código que considere necesarias y utilizar %% para separar secciones de código. Escriba en dicho script los comandos necesarios para cumplir el procedimiento detallado a continuación. Presente sus resultados en el *formato* proporcionado en el **blog** del curso. Recuerde utilizar el comando **help** para obtener ayuda y ejemplos de los comandos que requiera.

1. Ingrese la función de transferencia de la planta en MATLAB[®] y los parámetros K , T_i y T_d del controlador PID dado.
2. A partir de dichos valores, halle las constantes proporcional, integral y derivativa de los controladores discretos para cada tiempo de muestreo y cada representación (forma posicional regular, forma posicional trapezoidal y forma de velocidad).

Para la forma posicional considere:
$$G_c(z) = \frac{(K_{pp} + K_{dp} + K_{ip})z^2 - (K_{pp} + 2K_{dp})z + K_{dp}}{z(z-1)}$$

3. En Simulink, cree un modelo (modelo1) que le permita simular el sistema de control continuo con realimentación unitaria negativa ante una entrada **escalón unitario**. Use el bloque y el bloque **Transfer Fcn, Zero-Pole o LTI System** para representar a la planta. Incluya los bloques **Mux** y **Scope** para visualizar y guardar los datos de la entrada, salida y señal de control del sistema.

4. En Simulink, cree otro modelo (modelo2) que le permita simular el sistema de control discreto en la forma posicional de la figura 4.1 ante una entrada **escalón unitario**. Use el bloque **Discrete Transfer Fcn** para representar al controlador PID discreto, el bloque **Zero-Order Hold** para representar el muestreador y el bloque **Transfer Fcn, Zero-Pole o LTI System** para representar a la planta. Incluya los bloques **Mux** y **Scope** para visualizar y guardar los datos de la entrada, salida y señal de control del sistema.
5. Simule el modelo1 una única vez desde el script. Almacene estos datos como cont.
6. Simule el modelo2 cambiando el parámetro del tiempo de muestreo y constantes del controlador para cada tiempo de muestreo y forma (regular o trapezoidal). Use como guía el código mostrado en la figura 4.2. Al finalizar las simulaciones debe tener 6 grupos de datos generados por este modelo.

```
T=  
Kip=  
Kpp=  
Kdp=  
sim('modelo2',Tsim)  
DatosT1PR  
Kpp=  
sim('modelo2',Tsim)  
DatosT1PT
```

Figura 4.2: Diagrama de bloques para controlador en forma de velocidad

7. Obtenga las figuras y gráficas solicitadas a continuación, y comente los resultados obtenidos en cada caso. Las gráficas deben incluir títulos, cuadrículas, nombre a los ejes y leyendas; **todas** las gráficas se realizan en función del tiempo.
 - a) Figura 1: Entrada y salida del modelo analógico (modelo1) vs. tiempo.
 - b) Figura 2: Entrada y superposición de salidas del modelo discreto en forma posicional regular (modelo2) para cada tiempo de muestreo vs. tiempo.
 - c) Figura 3: Entrada y superposición de salidas del modelo discreto en forma posicional trapezoidal (modelo2) para cada tiempo de muestreo vs. tiempo.
 - d) Figura 5: Superposición de señales de control del modelo analógico (modelo1), del modelo discreto en forma posicional regular con mejor desempeño (modelo2) y del modelo discreto en forma posicional trapezoidal con mejor desempeño (modelo2) vs. tiempo.

4.5. Conclusiones y Recomendaciones