

Taller 6

Análisis de estabilidad de sistemas usando el criterio de Routh-Hurwitz

Jaime Patricio Chiqui Chiqui
Ingeniería en Telecomunicaciones
Universidad de Cuenca
Cuenca, Ecuador
jpatricio.chiquic@ucuenca.edu.ec

Abstract—Este trabajo presenta un análisis exhaustivo de la estabilidad de sistemas dinámicos mediante el criterio de Routh-Hurwitz y la ubicación de polos en el plano complejo. Se implementaron diversos sistemas de primer y segundo orden en MATLAB, evaluando su comportamiento temporal mediante respuestas al escalón y diagramas de polos y ceros. Se analizaron sistemas estables con diferentes factores de amortiguamiento ($\zeta = 0, 0.2, 1, 2$) y sistemas inestables tanto oscilatorios como no oscilatorios. Los resultados demuestran la relación directa entre la posición de los polos en el plano-s y la estabilidad del sistema, validando el criterio de Routh-Hurwitz como herramienta fundamental para el análisis de estabilidad sin necesidad de calcular explícitamente las raíces del polinomio característico. Las simulaciones confirman que sistemas con polos en el semiplano derecho presentan respuestas divergentes, mientras que aquellos con polos en el semiplano izquierdo convergen a valores finitos.

Index Terms—Estabilidad, Routh-Hurwitz, polos y ceros, función de transferencia, sistemas de control, amortiguamiento

I. INTRODUCCIÓN

La estabilidad es una de las propiedades más fundamentales en el análisis y diseño de sistemas de control. Un sistema estable es aquel cuya salida permanece acotada ante entradas acotadas, mientras que un sistema inestable presenta respuestas que crecen indefinidamente, lo cual es inaceptable en aplicaciones prácticas [1]. La determinación de la estabilidad de un sistema es crucial antes de proceder con cualquier diseño de controlador o análisis de desempeño.

El criterio de estabilidad de Routh-Hurwitz, desarrollado independientemente por Edward John Routh en 1877 y Adolf Hurwitz en 1895, proporciona un método algebraico para determinar la estabilidad de un sistema sin necesidad de calcular explícitamente las raíces del polinomio característico [2]. Este criterio establece que un sistema es estable si y solo si todos los elementos de la primera columna de la tabla de Routh tienen el mismo signo (generalmente positivo), **sin cambios de signo**.

Complementariamente, el análisis de la ubicación de polos en el plano complejo ofrece una interpretación geométrica de la estabilidad. Un sistema lineal e invariante en el tiempo (LTI) es estable si todos sus polos se encuentran en el semiplano izquierdo del plano-s (**parte real negativa**). Los polos en el

eje imaginario producen oscilaciones sostenidas, mientras que polos en el semiplano derecho generan respuestas divergentes.

El presente taller tiene como objetivo principal analizar la estabilidad de diversos sistemas dinámicos mediante simulaciones en MATLAB, verificando tanto la respuesta temporal como la ubicación de polos. Se estudiarán sistemas de primer orden, sistemas de segundo orden con diferentes factores de amortiguamiento, sistemas realimentados y sistemas intencionalmente inestables para comprender exhaustivamente el comportamiento dinámico en función de la ubicación de sus polos.

II. MARCO TEÓRICO

A. Función de Transferencia

La función de transferencia de un sistema LTI se define como la relación entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada, asumiendo condiciones iniciales nulas:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_ms^m + b_{m-1}s^{m-1} + \dots + b_1s + b_0}{a_ns^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} \quad (1)$$

donde a y b son constantes y $n \geq m$ para sistemas físicamente realizables. **Los valores de s que hacen que el numerador sea cero se denominan ceros, mientras que los valores que anulan el denominador se llaman polos.**

B. Criterio de Estabilidad BIBO

Un sistema es estable en el sentido BIBO (Bounded-Input Bounded-Output) si para toda entrada acotada, la salida también permanece acotada (visto en la materia de **sistemas y procesos estocásticos**). Matemáticamente, esto se expresa como:

$$|u(t)| < M_u \Rightarrow |y(t)| < M_y \quad (2)$$

Para sistemas LTI, la estabilidad BIBO se garantiza si todos los polos de $G(s)$ tienen parte real estrictamente negativa.

C. Criterio de Routh-Hurwitz

Dada la ecuación 1, este se hace muy compleja resolver, por lo que el criterio de Routh, nos indica si existen polos inestables dado el polinomio característico (denominador de la ec.1):

$$a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0 = 0 \quad (3)$$

La tabla de Routh se construye como se muestra en la Tabla I.

TABLE I: Estructura de la Tabla de Routh-Hurwitz

s^n	a_n	a_{n-2}	a_{n-4}	\dots
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	\dots
s^{n-2}	b_1	b_2	b_3	\dots
s^{n-3}	c_1	c_2	c_3	\dots
\vdots	\vdots	\vdots	\vdots	\vdots
s^0	h_1			

Los elementos de las filas subsecuentes se calculan mediante:

$$b_1 = \frac{a_{n-1} \cdot a_{n-2} - a_n \cdot a_{n-3}}{a_{n-1}} \quad (4)$$

$$b_2 = \frac{a_{n-1} \cdot a_{n-4} - a_n \cdot a_{n-5}}{a_{n-1}}$$

$$b_3 = \frac{a_{n-1} \cdot a_{n-6} - a_n \cdot a_{n-7}}{a_{n-1}}$$

.....

$$c_1 = \frac{b_1 \cdot a_{n-3} - a_{n-1} \cdot b_2}{b_1}$$

$$c_2 = \frac{b_1 \cdot a_{n-5} - a_{n-1} \cdot b_3}{b_1}$$

$$c_3 = \frac{b_1 \cdot a_{n-7} - a_{n-1} \cdot b_4}{b_1}$$

....

Criterio de estabilidad: El número de cambios de signo en la primera columna de la tabla de Routh es igual al número de raíces con parte real positiva (inestables). Un sistema es estable si y solo si todos los elementos de la primera columna de la tabla de Routh tienen el mismo signo (generalmente positivo), **sin cambios de signo**

D. Sistemas de Segundo Orden

Un sistema de segundo orden típico tiene la forma:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5)$$

donde ω_n es la frecuencia natural no amortiguada y ζ es el factor de amortiguamiento. Los polos del sistema son:

$$s_{1,2} = -\zeta\omega_n \pm \omega_n \sqrt{\zeta^2 - 1} \quad (6)$$

Según el valor de ζ , el sistema se clasifica como se muestra en la Tabla II.

TABLE II: Clasificación según el Factor de Amortiguamiento

Tipo	Condición	Naturaleza de Polos
No amortiguado	$\zeta = 0$	2 polos, Imaginarios puros
Subamortiguado	$0 < \zeta < 1$	2 polos, Complejos conjugados
Críticamente amortiguado	$\zeta = 1$	2 polos, Reales iguales
Sobreamortiguado	$\zeta > 1$	2 polos, Reales distintos

E. Sistemas Realimentados

Para un sistema con realimentación negativa, la función de transferencia en lazo cerrado es:

$$G_{cl}(s) = \frac{G(s)}{1 + G(s)H(s)} \quad (7)$$

donde $G(s)$ es la función de transferencia directa y $H(s)$ es la función de transferencia de realimentación.

III. DESARROLLO

A. Ejercicio 1: Sistemas de Primer Orden

Enunciado: Analizar dos sistemas de primer orden con funciones de transferencia:

$$G_1(s) = \frac{1}{2s+1}, \quad G_2(s) = \frac{1}{2s-1} \quad (8)$$

Desarrollo:

Para $G_1(s)$, el polo está en:

$$2s+1=0; \quad 2s=-1; \quad s=-\frac{1}{2}$$

$s = -0.5$ (semiplano izquierdo), lo que indica un sistema estable.

ANÁLISIS DEL TIEMPO DE ASENTAMIENTO (t_s)

- 1) **Constante de Tiempo (τ):** Dado el sistema de primer orden $G_1(s) = \frac{1}{2s+1}$, la constante de tiempo τ es el coeficiente de s en el denominador.

$$\tau = 2 \text{ segundos}$$

- 2) **Tiempo de Asentamiento (t_s):** Para un sistema de primer orden, el tiempo de asentamiento (t_s), que es el tiempo para alcanzar aproximadamente el 98% del valor final (criterio del 2%), se define como:

$$t_s = 4\tau$$

- 3) **Identificación del tiempo $t = 4\tau$:** Calculando el valor para el sistema dado:

$$t_s = 4 \times 2 = 8 \text{ segundos}$$

Este punto temporal (8 segundos) se identifica en la simulación para obtener el valor exacto de la respuesta al escalón $y(8)$. Se observa que en $t = 8$ segundos, la respuesta ha alcanzado aproximadamente el 98% de su valor final ($y_{ss} \approx 1$), confirmando el tiempo de asentamiento.

El código MATLAB implementado es:

Listing 1: Análisis de Sistemas de Primer Orden en MATLAB

```
1 clc, clear all, close all;
2
3 num = [1];
4 den = [2 1];
5 G_s = tf(num, den);
6 [y, t] = step(G_s, 0:0.01:12); % Forzamos t
7 [~, idx] = min(abs(t - 8)); % obtener y(8)
8 roots(den)
9
10 figure(); hold on;
11 plot(t, y);
12 plot(8, y(idx), '*', 'Color', [1 0 0]);
13 legend('Respuesta al escalon', 'Estabilizacion: $t = 4\tau$');
14 title('Sistema Estable G_1(s)'); grid on;
```

Sistema Inestable:

Para $G_2(s)$ (el cual no conocemos), el polo debe estar en $s = 0.5$ (semiplano derecho), indicando inestabilidad. Proceso inverso de G_1 :

$$s = \frac{1}{2}, \quad 2s = 1; \quad 2s - 1 = 0; \quad G_2(s) = \frac{1}{2s - 1}$$

El código MATLAB implementado es:

Listing 2: Análisis de Sistemas de Primer Orden en MATLAB

```
1 clc, clear all, close all;
2
3 num = [1];
4 den = [2 -1];
5 G_s2 = tf(num, den);
6 [y, t] = step(G_s2, 0:0.01:120); % Forzamos t
7 [~, idx] = min(abs(t - 8)); % obtener y(8)
8 roots(den)
9
10 figure(); hold on;
11 plot(t, y);
12 plot(8, y(idx), '*', 'Color', [1 0 0]);
13 legend('Respuesta al escalon', 'Estabilizacion: $t = 4\tau$');
14 title('Sistema Inestable G_2(s)'); grid on;
```

Resultados y Análisis:

La Figura 1 muestra las respuestas al escalón de ambos sistemas. El sistema $G_1(s)$ presenta una respuesta exponencial convergente típica de un sistema de primer orden estable, alcanzando el valor de estado estacionario en aproximadamente 4τ donde $\tau = 2$ segundos.

Por el contrario, en la Figura 2, $G_2(s)$ exhibe crecimiento exponencial ilimitado, confirmando su inestabilidad debido al polo en el semiplano derecho.

Aplicando el criterio de Routh-Hurwitz a $G_1(s)$: $2s + 1 = 0$, todos los coeficientes son positivos, confirmando estabilidad. Coeficientes: $a_0 = 2$, $a_1 = 1$.

Raíz	Signo	Coeficientes
s^1	(+)	$a_0 = 2 \quad 0$
s^0	(+)	$a_1 = 1 \quad 0$

Para $G_2(s)$: $2s - 1 = 0$, el cambio de signo indica inestabilidad. Coeficientes: $a_0 = 2$, $a_1 = -1$.

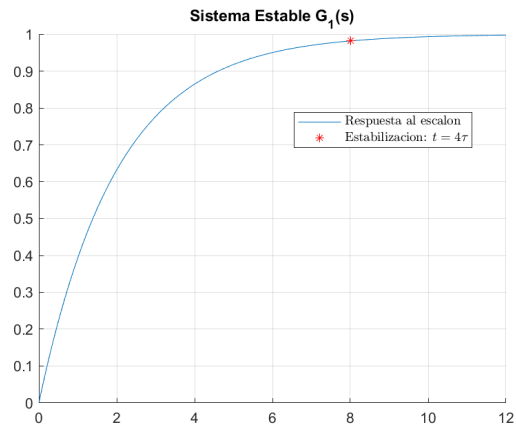


Fig. 1: Respuesta al Escalón del Sistema Estable de $G_1(s)$

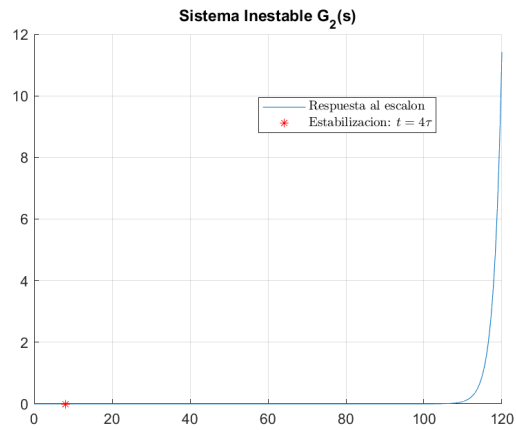


Fig. 2: Respuesta al Escalón del Sistema Inestable de $G_2(s)$

Raíz	Signo	Coeficientes
s^1	(+)	$a_0 = 2 \quad 0$
s^0	(-)	$a_1 = -1 \quad 0$

B. Ejercicio 2: Sistemas de Segundo Orden con Diferentes Amortiguamientos

Enunciado: Analizar sistemas de segundo orden con $\omega_n = 1$ rad/s y factores de amortiguamiento $\zeta = 0, 0.2, 1, 2$.

Desarrollo:

Listing 3: Análisis de Sistemas con Diferentes Amortiguamientos

```
1 clc; clear; close all;
2
3 %% Parametros Generales
4 wn = 1; % Frec natural no amortiguada (rad/s)
5 t = 0:0.01:15;
6
7 %% Figura 1: No Amortiguado y Sub-amortiguado
8 figure;
9 hold on;
10
11 % 1. Sistema No Amortiguado (zeta = 0)
12 graficar_sistema(0, wn, t, 1, 'No Amortiguado');
13
14 % 2. Sistema Sub-amortiguado (zeta = 0.2)
15 graficar_sistema(0.2, wn, t, 3, 'subamortiguado');
```

```

16 %% Figura 2: Críticamente Amortiguado y Sobre-
    amortiguado
17 figure;
18 hold on;
19
20 % 3. Sistema Críticamente Amortiguado (zeta = 1)
21 graficar_sistema(1, wn, t, 1, 'críticamente
    amortiguado');
22
23 % 4. Sistema Sobre-amortiguado (zeta = 2)
24 graficar_sistema(2, wn, t, 3, 'sobre amortiguado');
25
26
27 %% --- FUNCION Graficar ---
28 function graficar_sistema(zeta_val, wn, t,
    plot_index_start, titulo_tipo)
29     num_sys = wn^2;
30     den_sys = [1, 2*zeta_val*wn, wn^2];
31     G_sys = tf(num_sys, den_sys);
32
33     % Calculo de la respuesta al escalon
34     [Y_sys, T_sys] = step(G_sys, t);
35
36     % Gráfica 1: Respuesta en el tiempo
37     subplot(2, 2, plot_index_start);
38     plot(T_sys, Y_sys);
39     title(['Respuesta al escalon (' titulo_tipo ')
        ']); grid on;
40
41     % Gráfica 2: Mapa de Polos y Ceros
42     subplot(2, 2, plot_index_start + 1);
43     pzmap(G_sys);
44     title(['Diagrama de Polos y Ceros (' titulo_tipo
        ' ')]);
45 end

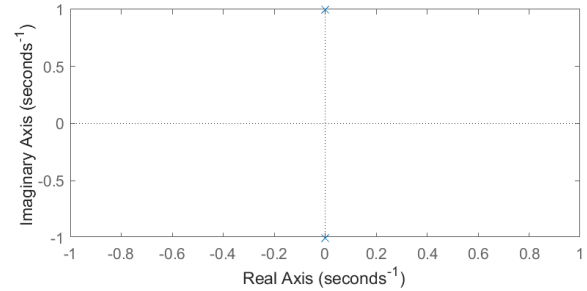
```

Análisis de Resultados:

- **Sistema No Amortiguado** ($\zeta = 0$): Según la ecuación (6), los polos son $s = \pm j\omega_n = \pm j$, ubicados en el eje imaginario. La Figura 3 muestra oscilaciones sostenidas de amplitud constante sin convergencia, característica de sistemas marginalmente estables.
- **Sistema Subamortiguado** ($\zeta = 0.2$): Los polos son complejos conjugados $s = -0.2 \pm j0.98$ en el semiplano izquierdo. La Figura 4, para la respuesta presenta oscilaciones amortiguadas que convergen al valor final (aproximadamente 1), con sobrepico significativo típico de $\zeta < 1$.
- **Sistema Críticamente Amortiguado** ($\zeta = 1$): Polos reales dobles en $s = -1$. Representa el caso límite entre comportamiento oscilatorio y no oscilatorio, con la respuesta más rápida sin sobrepico, como se observa en la Figura 5.
- **Sistema Sobreamortiguado** ($\zeta = 2$): Polos reales distintos en $s = -0.268$ y $s = -3.732$. La Figura 6, la respuesta es lenta y exponencial sin oscilaciones ni sobrepico, pero con mayor tiempo de asentamiento que el caso críticamente amortiguado.

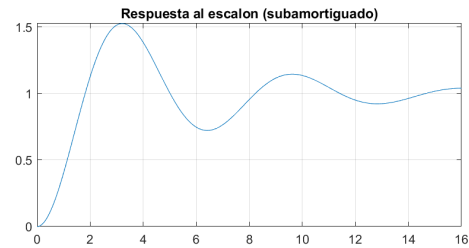


(a) Respuesta al Escalón, step
Diagrama de Polos y Ceros (No Amortiguado)

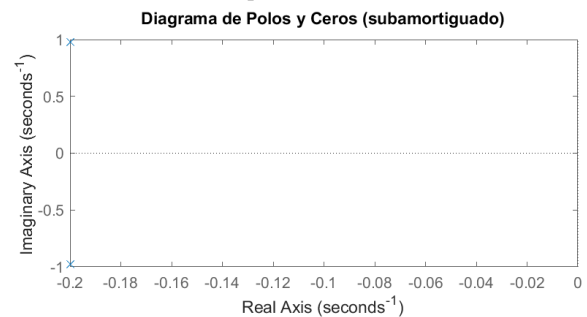


(b) Mapa de Polos ($s = \pm j$), pzmap

Fig. 3: Sistema No Amortiguado ($\zeta = 0$). Se observan oscilaciones sostenidas y polos sobre el eje imaginario.



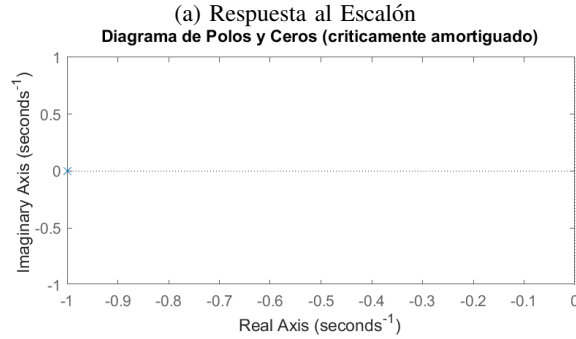
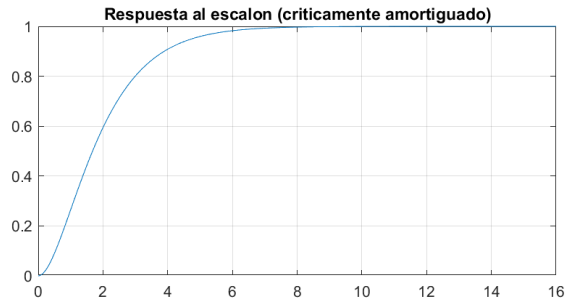
(a) Respuesta al Escalón



(b) Mapa de Polos ($s = -0.2 \pm j0.98$)

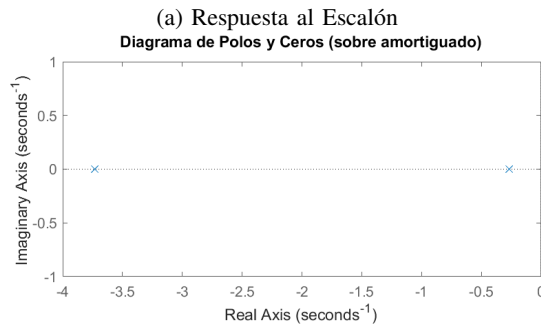
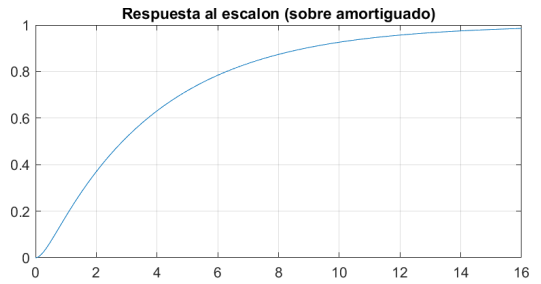
Fig. 4: Sistema Subamortiguado ($\zeta = 0.2$). Oscilaciones amortiguadas con sobrepico significativo.

En todos los casos excepto $\zeta = 0$, los sistemas son estables según Routh-Hurwitz, ya que todos los coeficientes del polinomio característico son positivos y no hay cambios de signo en la primera columna.



(b) Mapa de Polos ($s = -1$ doble)

Fig. 5: Sistema Críticamente Amortiguado ($\zeta = 1$). Respuesta más rápida sin sobrepico (límite oscilatorio).



(b) Mapa de Polos Reales Distintos

Fig. 6: Sistema Sobreamortiguado ($\zeta = 2$). Respuesta lenta y exponencial sin oscilaciones.

C. Ejercicio 3: Sistemas Inestables

Enunciado: Analizar dos sistemas inestables:

$$G_1(s) = \frac{1}{s^2 - 0.5s + 4}, \quad G_2(s) = \frac{1}{25s^2 - 12.5s + 1} \quad (9)$$

Desarrollo:

Listing 4: Análisis de Sistemas Inestables en MATLAB

```
1 clc; clear; close all;
2 t_inestable = 0:0.01:30;
3
4 %% 1. Sistema Inestable Oscilatorio (G1)
5 % G1 = 1 / (s^2 - 0.5s + 4)
6 num_G1 = [1];
7 den_G1 = [1 -0.5 4];
8 G1 = tf(num_G1, den_G1)
9 roots(den_G1)
10 analizar_sistema_inestable(G1, t_inestable, ...
11     'G1: Inestable Oscilatorio (Polos Complejos)');
12
13 %% 2. Sistema Inestable No Oscilatorio (G2)
14 % G2 = 1 / (25s^2 - 12.5s + 1)
15 num_G2 = [1];
16 den_G2 = [25 -12.5 1];
17 G2 = tf(num_G2, den_G2)
18 roots(den_G2)
19 analizar_sistema_inestable(G2, t_inestable, ...
20     'G2: Inestable No Oscilatorio (Polos Reales)');
21
22 % Graficas
23 function analizar_sistema_inestable(G_sys, t,
24     titulo_general)
25     [Y, T] = step(G_sys, t);
26
27     figure(); subplot(1, 2, 1);
28     plot(T, Y); % Respuesta al Escalon
29     title(['Respuesta al Escalon: ' titulo_general]);
30     xlabel('Tiempo (s)'); ylabel('Amplitud');
31     grid on;
32
33     subplot(1, 2, 2);
34     pzmap(G_sys); % Mapa de Polos y Ceros
35     title(['Mapa de Polos y Ceros: ' titulo_general]);
36 end
```

Análisis con Routh-Hurwitz:

Para $G_1(s)$, el polinomio característico es $s^2 - 0.5s + 4 = 0$:

Coefficientes: $a_0 = 1$, $a_1 = -0.5$, $a_2 = 4$.

Raíz	Signo	Columna 1	Columna 2
s^2	(+)	$a_0 = 1$	$a_2 = 4$
s^1	(-)	$a_1 = -0.5$	0
s^0	(+)	$\frac{(-0.5 \times 4) - (1 \times 0)}{-0.5} = \frac{-2}{-0.5} = 4$	0

Conclusión: La primera columna de Routh es $[1, -0.5, 4]^T$. Hay dos cambios de signo ($+ \rightarrow -$ y $- \rightarrow +$). Esto indica que hay **dos polos** en el Semi-Plano Derecho, confirmando la inestabilidad del sistema $G_1(s)$. En MATLAB la función (`roots(den_G1)`), nos confirma que la parte real está en el eje positivo (0.25), inestabilidad:

$$0.2500 + 1.9843i$$

$$0.2500 - 1.9843i$$

Caso 2: $G_2(s)$ Polinomio: $25s^2 - 12.5s + 1 = 0$

Coefficientes: $a_0 = 25$, $a_1 = -12.5$, $a_2 = 1$.

Raíz	Signo	Columna 1	Columna 2
s^2	(+)	$a_0 = 25$	$a_2 = 1$
s^1	(-)	$a_1 = -12.5$	0
s^0	(+)	$\frac{(-12.5 \times 1) - (25 \times 0)}{-12.5} = \frac{-12.5}{-12.5} = 1$	0

Conclusión: La primera columna de Routh es $[25, -12.5, 1]^T$. Hay dos cambios de signo ($+ \rightarrow -$ y $- \rightarrow +$). Esto indica que hay dos polos en el Semi-Plano Derecho, confirmando la inestabilidad del sistema $G_2(s)$. En MATLAB la función (`roots(den_G2)`), nos confirma que la parte real está en el eje positivo (0.4 y 0.1), inestabilidad:

```
0.4000
0.1000
```

Resultados:

La Figura 7 muestra que $G_1(s)$ presenta oscilaciones de amplitud creciente (inestabilidad oscilatoria) debido a polos complejos con parte real positiva.

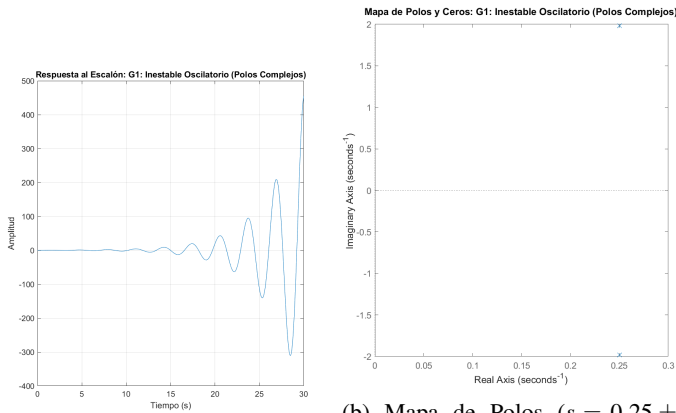


Fig. 7: Sistema Inestable Oscilatorio $G_1(s)$. Los polos están en el Semi-Plano Derecho, lo que causa un crecimiento oscilatorio de la respuesta.

Mientras que $G_2(s)$, la Figura 8 exhibe crecimiento exponencial sin oscilaciones (inestabilidad no oscilatoria) debido a polos reales positivos.

Nota importante: Ambos sistemas divergen rápidamente, confirmando la predicción del criterio de Routh-Hurwitz. Los sistemas inestables requieren tiempos de simulación más cortos ya que sus respuestas divergen rápidamente. Estos sistemas deben ser estabilizados mediante realimentación o compensación antes de su implementación.

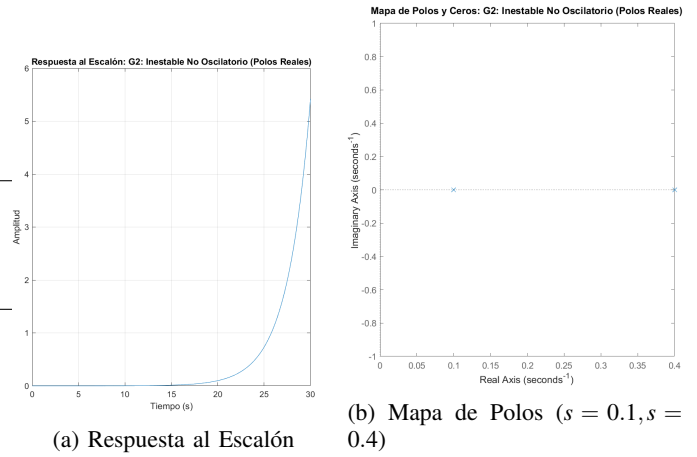


Fig. 8: Sistema Inestable No Oscilatorio $G_2(s)$. Polos reales positivos, lo que resulta en un crecimiento exponencial monotónico.

D. Ejercicio 4: Sistema Realimentado

Enunciado: Analizar un sistema con realimentación compuesto por:

$$G_1(s) = \frac{3}{s}, \quad G_2(s) = \frac{1}{s^2 + 1}, \quad H_1(s) = 3 \quad (10)$$

Donde G_1 y G_2 están en serie, y H_1 está en la realimentación.

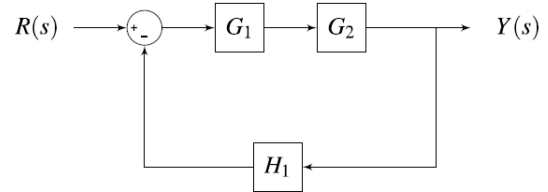


Fig. 9: Diagrama de Bloques del sistema a implementar

Desarrollo en Simulink:

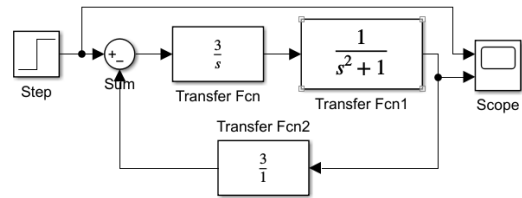


Fig. 10: Diagrama de Bloques del sistema a implementar

- 1) **Modificar el Tiempo de Parada (Stop Time):** Cambiar el valor predeterminado del *Stop Time* del modelo (ubicado en la parte superior de la ventana de Simulink) de 10.0s a 45s.
- 2) **Ejecutar y Abrir el Scope:** Ejecutar la simulación y abrir la ventana del bloque *Scope* para visualizar la salida $y(t)$.

3) **Ajustar el Eje Vertical (Y-limits):** Dado que el sistema es inestable y la señal crece exponencialmente, el auto-escalado del Scope puede "aplanar" las oscilaciones iniciales. Para observarlas, se debe ajustar el eje Y:

- Clic en **Settings** (engranaje) dentro del Scope.
- En la pestaña **Y-limits**, establecer manualmente un rango de visualización más pequeño, por ejemplo, **[-50, 50]**.

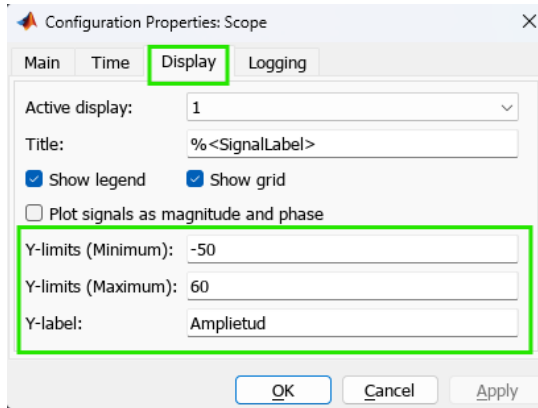


Fig. 11: Configuración simulink para visualizar mejor que el sistema es inestable

Desarrollo en MATLAB:

El código MATLAB implementado calcula la función de transferencia en lazo cerrado:

Listing 5: Sistema Realimentado en MATLAB

```
1 clc, clear all, close all;
2 t = 0:0.01:45;
3
4 num1 = [3]; den1 = [1 0];
5 G1 = tf(num1, den1)
6 num2 = [1]; den2 = [1 0 1];
7 G2 = tf(num2, den2)
8 num3 = [3]; den3 = [1];
9 H1 = tf(num3, den3)
10
11 % Bloques en Serie: Combinar G1 y G2
12 Gol = series(G1, G2) % G1 * G2
13
14 % lazo cerrado (retroalimentado)
15 Gcl = feedback(Gol, H1)
16 [y, t] = step(Gcl, t);
17
18 figure(); subplot(1,2,1);
19 plot(t, y); ylim([-10, 100]);
20 grid("on"); subplot(1,2,2);
21 pzmap(Gcl)
22
23 % DATOS FT
24 [num_cl, den_cl] = tfdata(Gcl, 'v')
25 roots(den_cl) % polos Gcl
```

Análisis Teórico:

La función de transferencia en lazo abierto es:

$$G_{ol}(s) = G_1(s) \cdot G_2(s) = \frac{3}{s(s^2+1)} = \frac{3}{s^3+s}$$

La función de transferencia en lazo cerrado según la ecuación (7):

$$G_{cl}(s) = \frac{\frac{3}{s^3+s}}{1 + \frac{3}{s^3+s} \cdot 3} = \frac{\frac{3}{s^3+s}}{\frac{s^3+s+9}{s^3+s}} = \frac{3}{s^3+s+9}$$

Aplicación del Criterio de Routh-Hurwitz:

Para el polinomio característico $s^3 + 0s^2 + s + 9 = 0$:

Coefficientes iniciales: $a_0 = 1$, $a_1 = 0$, $a_2 = 1$, $a_3 = 9$.

El coeficiente $a_1 = 0$ en la fila s^2 requiere la sustitución por ε para continuar el cálculo.

Raíz	Signo	Columna 1	Columna 2
s^3	(+)	$a_0 = 1$	$a_2 = 1$
s^2	(±)	$a_1 \rightarrow \varepsilon$	$a_3 = 9$
s^1	(-)	$\frac{(\varepsilon \times 1) - (1 \times 9)}{\varepsilon} = \frac{\varepsilon - 9}{\varepsilon} \approx -\frac{9}{\varepsilon}$	0
s^0	(+)	$\frac{(-\frac{9}{\varepsilon} \times 9) - (\varepsilon \times 0)}{-\frac{9}{\varepsilon}} = 9$	0

Análisis de Estabilidad

Considerando ε como un número real positivo muy pequeño ($\varepsilon \rightarrow 0^+$), es decir, $0 < \varepsilon < 1$:

- 1) s^3 : Coeficiente es 1 (+).
- 2) s^2 : Coeficiente es ε (+).
- 3) s^1 : Coeficiente es $\frac{\varepsilon - 9}{\varepsilon}$. Como $\varepsilon \rightarrow 0^+$, esta expresión tiende a $-\frac{9}{0^+} \rightarrow -\infty$ (-).
- 4) s^0 : Coeficiente es 9 (+).

Cambios de Signo en la Primera Columna:

- De s^3 (+) a s^2 (+) → **0 cambio**.
- De s^2 (+) a s^1 (-) → **1 cambio**.
- De s^1 (-) a s^0 (+) → **1 cambio (Total: 2 cambios)**.

Conclusión: El Criterio de Routh-Hurwitz revela **dos (2) cambios de signo** en la primera columna (+ → - y - → +). Por lo tanto, el sistema en lazo cerrado tiene **dos polos** ubicados en el Semi-Plano Derecho del plano complejo.

El sistema es **Inestable**.

Resultados de la Consola de MATLAB

La ejecución del código en MATLAB confirmó la función de transferencia en lazo abierto $G_{ol}(s)$ y la función de lazo cerrado $G_{cl}(s)$ y proporcionó los coeficientes del numerador y denominador, así como las raíces del polinomio característico (de $G_{cl}(s)$):

Gol =

$$\frac{3}{s^3 + s}$$

Continuous-time transfer function.

```

Gcl =
      3
-----
    s^3 + s + 9
Continuous-time transfer function.
num_cl =
      0      0      0      3
den_cl =
      1      0      1      9

ans (Raíces de los Polos, lazo cerrado) =
    0.9601 + 1.9404i
    0.9601 - 1.9404i
   -1.9202 + 0.0000i

```

Los polos del sistema son $\text{Re}(s_1) = 0.9601$, $\text{Re}(s_2) = 0.9601$ y $\text{Re}(s_3) = -1.9202$. La presencia de dos polos con parte real positiva (0.9601) confirma la inestabilidad.

E. Gráficas de Simulación

La Figura 14 muestra la respuesta al escalón con una **oscilación de amplitud creciente**, comportamiento que es característico de un sistema inestable con polos complejos en el semiplano derecho.

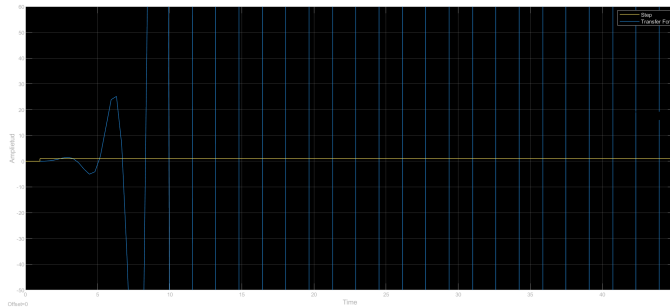
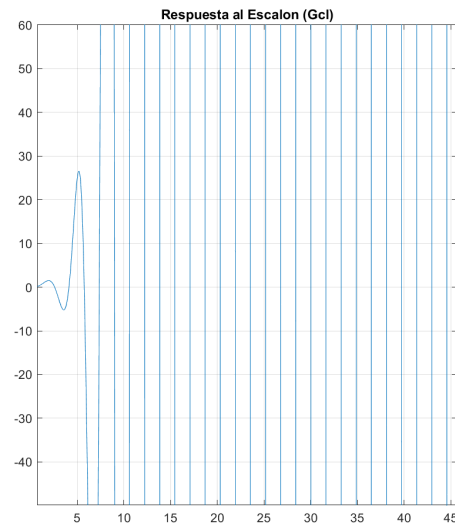


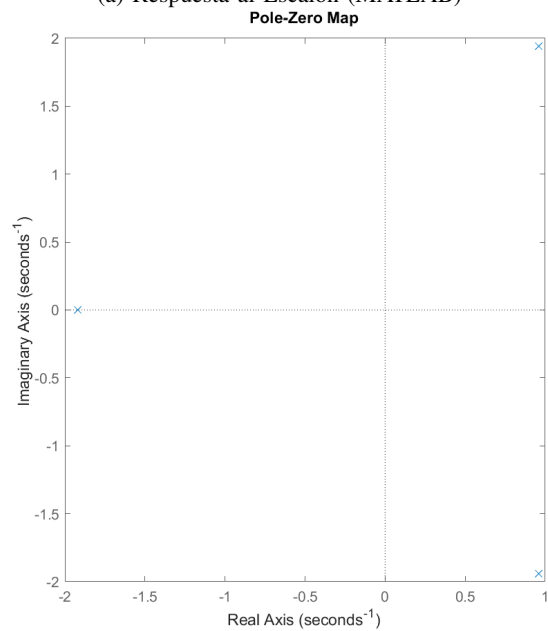
Fig. 12: Respuesta al Escalón obtenida mediante la simulación en Simulink.

La Figura 14 y la Figura 13a, se obtuvieron las mismas gráficas, por lo que se entiende que los resultados obtenidos para graficar la respuesta del escalón en simulink y con base al código en MATLAB están correctos.

Como se observa en el mapa de polos de la Figura 13, la ubicación de los polos en el semiplano derecho (ceranos al eje imaginario), valida el resultado inestable del Criterio de Routh-Hurwitz y el comportamiento de la respuesta temporal simulada, la respuesta al escalón presenta oscilaciones de amplitud creciente sin convergencia a un valor fijo.



(a) Respuesta al Escalón (MATLAB)



(b) Mapa de Polos y Ceros ($s^3 + s + 9 = 0$)

Fig. 13: Resultados de MATLAB. El diagrama de polos confirma la inestabilidad, lo cual se traduce en la respuesta temporal de amplitud creciente.

CONCLUSIÓN DEL ANÁLISIS DE ESTABILIDAD

Aplicación de Pruebas de Estabilidad

A partir del Criterio de Routh-Hurwitz (2 cambios de signo) y la ubicación de los polos (2 polos en el Semi-Plano Derecho), se determina que el sistema en lazo cerrado es:

- **Inestable:** Debido a polos con $\text{Re}(s) > 0$.
- **Oscilante:** Debido a los polos complejos conjugados.
- **Tipo de amortiguamiento:** Su comportamiento es un sistema **NO amortiguado** con un factor de amortiguamiento $\zeta \approx 0$, resultando en **oscilaciones crecientes**.

F. Actividad Reto

1) Implementación en Lazo Cerrado con PID para Estabilizar Circuito RLC.:

Se implementa el circuito RLC analizado en talleres anteriores (Taller 5) con un controlador PID sintonizado de forma específica en un sistema de lazo cerrado. El resultado de este análisis fue documentado para verificar si el sistema es estable o inestable en función de los coeficientes del polinomio característico. Determinar la estabilidad del sistema mediante el criterio de Routh-Hurwitz, siendo el polinomio característico resultante:

$$P(s) = s^4 + 11000s^3 + 1.3 \times 10^8 s^2 + 2.1 \times 10^{13} s + 1 \times 10^{14}$$

Derivación del Polinomio Característico $P(s)$: El polinomio característico de cuarto orden (s^4) se obtiene de la ecuación del sistema en lazo cerrado $1 + G_c(s)G_p(s) = 0$, donde $G_p(s)$ es la función de transferencia de la planta RLC y $G_c(s)$ es un controlador PID con un filtro en la acción derivativa.

Paso 1: Función de Transferencia de la Planta ($G_p(s)$):

Basado en los parámetros del Taller 5 (Taller5_JaimeChiqui.pdf): $R = 5\Omega$, $L = 0.1H$, $C = 220\mu F$. La función de transferencia del circuito RLC (voltaje en el capacitor V_c vs. voltaje de entrada V_{in}) es:

$$G_p(s) = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

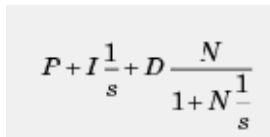
Sustituyendo valores:

$$G_p(s) = \frac{45454.545}{s^2 + 50s + 45454.545}$$

Donde: $\frac{1}{LC} \approx 45454.545$ y $\frac{R}{L} = 50$.

Paso 2: Función de Transferencia del Controlador ($G_c(s)$)

Para alcanzar un orden $n = 4$, se utiliza un PID con filtro en el término derivativo, $K_d \frac{Ns}{s+N}$:



$$P = I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}}$$

Fig. 14: Ecuación obtenida de MATLAB.

$$G_c(s) = K_p + \frac{K_i}{s} + K_d \left(\frac{Ns}{s+N} \right) = \frac{K_d N s^2 + K_p s(s+N) + K_i(s+N)}{s(s+N)}$$

Paso 3: Ecuación Característica del Lazo Cerrado ($P(s)$)

La ecuación característica $P(s)$ es el denominador de la función de transferencia en lazo cerrado ($G_{CL}(s)$):

$$P(s) = \text{Den}(G_{CL}(s))$$

$$P(s) = \text{Den}(G_p(s)) \cdot \text{Den}(G_c(s)) + \text{Num}(G_p(s)) \cdot \text{Num}(G_c(s))$$

$$P(s) = (s^2 + 50s + 45454.55)(s^2 + Ns) + (45454.55)[K_d N s^2 + K_p s(s+N) + K_i(s+N)] = 0$$

Paso 4: Sustitución de Valores Sintonizados y Cálculo del Polinomio Característico $P(s)$

Se utilizan los valores sintonizados obtenidos con la herramienta PID Tuner de MATLAB para el circuito RLC específico del Taller 5 (Taller5_JaimeChiqui.pdf):

- Ganancia Proporcional, $K_p = 16.7573$
- Ganancia Integral, $K_i = 1762.8024$
- Ganancia Derivativa, $K_d = 0.03622$
- Polo del Filtro Derivativo, $N = 6402.7121$
- Constante de la Planta, $A = \frac{1}{LC} \approx 45454.545$

El polinomio característico $P(s)$ se obtiene al expandir la Ecuación anterior y llevar a la forma:

$$P(s) = s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0$$

Paso 1: Expandir el producto de los denominadores (Término 1)

El producto del denominador de la planta (Den_p) y el denominador del controlador (Den_c) es:

$$\begin{aligned} T_1 &= (s^2 + 50s + A)(s^2 + Ns) \\ &= s^2(s^2 + Ns) + 50s(s^2 + Ns) + A(s^2 + Ns) \\ &= s^4 + Ns^3 + 50s^3 + 50Ns^2 + As^2 + ANs \end{aligned}$$

Reordenando por potencias de s :

$$T_1 = s^4 + (N + 50)s^3 + (50N + A)s^2 + (AN)s$$

Paso 2: Expandir el producto del Término 2

El producto del numerador de la planta (A) y el numerador del controlador (Num_c) es:

$$T_2 = A \cdot [K_d N s^2 + K_p s(s+N) + K_i(s+N)]$$

Primero, expandimos el término dentro del corchete:

$$\begin{aligned} &= K_d N s^2 + K_p(s^2 + Ns) + K_i s + K_i N \\ &= K_d N s^2 + K_p s^2 + K_p N s + K_i s + K_i N \\ &= (K_d N + K_p)s^2 + (K_p N + K_i)s + (K_i N) \end{aligned}$$

Ahora, multiplicamos por A :

$$T_2 = A(K_d N + K_p)s^2 + A(K_p N + K_i)s + A K_i N$$

Paso 3: Sumar T_1 y T_2 y Agrupar Coeficientes

Sumamos $P(s) = T_1 + T_2$ e identificamos los coeficientes c_n :

$$\begin{aligned} P(s) &= (s^4 + (N + 50)s^3 + (50N + A)s^2 + (AN)s) \\ &\quad + (A(K_d N + K_p)s^2 + A(K_p N + K_i)s + A K_i N) \end{aligned}$$

Agrupamos los términos con la misma potencia de s :

$$\begin{aligned} P(s) &= s^4 + (N + 50)s^3 + [(50N + A) + A(K_d N + K_p)]s^2 \\ &\quad + [(AN) + A(K_p N + K_i)]s + [A K_i N] \end{aligned}$$

Resultado Final del Polinomio Expandido:

Reemplazando $A = 45454.55$ y simplificando los coeficientes finales (c_3, c_2, c_1, c_0):

$$P(s) = s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0 = 0$$

Donde los coeficientes son:

$$\begin{aligned}c_3 &= N + 50 \\c_2 &= 50N + A + AK_dN + AK_p \\c_1 &= AN + AK_pN + AK_i \\c_0 &= AK_iN\end{aligned}$$

Cálculo de Coeficientes:

1) **Coeficiente c_3 (de s^3):**

$$c_3 = 50 + N = 50 + 6402.7121 \approx \mathbf{6452.71}$$

2) **Coeficiente c_0 (de s^0):**

$$\begin{aligned}c_0 &= A \cdot K_i \cdot N \\c_0 &= (45454.545) \cdot (1762.8024) \cdot (6402.7121) \approx \mathbf{5.13033 \times 10^{11}}\end{aligned}$$

3) **Coeficiente c_2 (de s^2):**

$$c_2 = 50N + A + AK_dN + AK_p \approx \mathbf{1.1669 \times 10^7}$$

4) **Coeficiente c_1 (de s^1):**

$$c_1 = AN + AK_pN + AK_i \approx \mathbf{5.249 \times 10^9}$$

Polinomio Característico Resultante:

Sustituyendo los coeficientes calculados, el polinomio característico del sistema en lazo cerrado sintonizado es:

$$P(s) = s^4 + 6453s^3 + 1.167 \times 10^7 s^2 + 5.249 \times 10^9 s + 5.13 \times 10^{11}$$

Funcion de lazo cerrado es:

$$\begin{aligned}Gcl &= \frac{Gc(s) * Gp(s)}{(1 + Gc(s) * Gp(s))} \\Gcl &= \frac{1.13e7s^2 + 4.957e9s + 5.13e11}{s^4 + 6453s^3 + 1.167e7s^2 + 5.248e9s + 5.13e11}\end{aligned}$$

Análisis de Estabilidad según Routh-Hurwitz

Para el polinomio característico obtenido:

$$P(s) = s^4 + 6453s^3 + 1.1669 \times 10^7 s^2 + 5.249 \times 10^9 s + 5.13 \times 10^{11}$$

Se construye el arreglo de Routh. Definimos los coeficientes como:

- $a_4 = 1$
- $a_3 = 6453$
- $a_2 = 1.1669 \times 10^7$
- $a_1 = 5.249 \times 10^9$
- $a_0 = 5.1303 \times 10^{11}$

Raíz	Columna 1	Columna 2	Columna 3
s^4	1	1.1669×10^7	5.1303×10^{11}
s^3	6453	5.249×10^9	0
s^2	b_1	b_2	0
s^1	c_1	0	0
s^0	d_1	0	0

Cálculos Auxiliares:

- $b_1 = \frac{(6453)(1.1669 \times 10^7) - (1)(5.249 \times 10^9)}{6453} \approx 1.0855 \times 10^7 \quad (+)$
- $b_2 = \frac{(6453)(5.1303 \times 10^{11}) - (1)(0)}{6453} = 5.1303 \times 10^{11} \quad (+)$
- $c_1 = \frac{(b_1)(5.249 \times 10^9) - (6453)(b_2)}{b_1} \approx 4.944 \times 10^9 \quad (+)$
- $d_1 = b_2 = 5.1303 \times 10^{11} \quad (+)$

TABLE III: Tabla de Routh-Hurwitz con Valores Reemplazados

Raíz	Signo	Columna 1	Columna 2	Columna 3
s^4	+	1	1.1669×10^7	5.1303×10^{11}
s^3	+	6453	5.249×10^9	0
s^2	+	1.0855×10^7	5.1303×10^{11}	0
s^1	+	4.944×10^9	0	0
s^0	+	5.1303×10^{11}	0	0

Análisis de la Primera Columna: Los elementos de la primera columna son:

$$[1, \quad 6453, \quad 1.0855 \times 10^7, \quad 4.944 \times 10^9, \quad 5.1303 \times 10^{11}]$$

Todos los coeficientes son **positivos**. No existen cambios de signo, lo que indica que **no hay polos en el semiplano derecho**. El sistema es teóricamente **estable**.

Implementación en MATLAB

A continuación se presenta el código utilizado para verificar los cálculos teóricos y visualizar la respuesta del sistema.

Listing 6: Código MATLAB para análisis de estabilidad del RLC con PID

```
1 clc; clear; close all;
2 % --- Parametros del Circuito RLC (Taller 5) ---
3 R = 5; % Resistencia (Ohmios)
4 L = 0.1; % Inductancia (Henrios)
5 C = 220e-6; % Capacitancia (Faradios)
6 % Parametros PID Sintonizados (PID Tuner) ---
7 Kp = 16.7573;
8 Ki = 1762.8024;
9 Kd = 0.03622;
10 N = 6402.7121;
11
12 % FT de la Planta (G_p(s)) ---
13 % Gp(s) = A / (s^2 + (R/L)s + A), donde A = 1/(L*C)
14 A = 1 / (L * C);
15 Num_p = A;
16 Den_p = [1, R/L, A];
17 disp('Funcion de Transfere de la Planta G_p(s):');
18 G_RLC = tf(Num_p, Den_p)
19
20 % FT del Controlador (G_c(s))
21 % Gc(s) = Kp + Ki/s + Kd * (N*s / (s + N))
22 Num_c = [ (Kd*N + Kp), (Kp*N + Ki), (Ki*N) ];
23 Den_c = [1, N, 0]; % s*(s+N) = s^2 + Ns + 0
24 disp('Funcion de Transferencia del Controlador
    Filtrado G_c(s):');
25 CPID = tf(Num_c, Den_c)
```

```

26 % Sistema en Lazo Cerrado (Gcl) ---
27 % Gcl = Gc(s) * Gp(s) / (1 + Gc(s) * Gp(s))
28 Serie = series(CPID, G_RLC);
29 disp('Funcion de Transferencia del Sistema en Lazo
    Cerrado G_CL(s):');
30 Gcl = feedback(Serie, 1) % Realimenta unitaria H=1
31
32 %% Analisis de Estabilidad ---
33 [Num_cl, Den_cl] = tfdata(Gcl, 'v');
34
35 % Hallar los Polos (raices de P(s))
36 disp('Ubicacion de los Polos (Raices del Polinomio
    Caracteristico):');
37 p = pole(Gcl)
38
39 % Verificar estabilidad
40 Real_Part = real(p);
41 if all(Real_Part < 0)
42     disp('=> El sistema es ESTABLE (Todos los polos
        estan en el semiplano izquierdo).');
43 else
44     disp('=> El sistema es INESTABLE. ');
45 end
46
47 % --- 7. Graficas de Analisis ---
48 figure(); step(Gcl); % Respuesta al escalon
49 title('Respuesta al Escalon del Sistema RLC con PID
    Filtrado'); grid on;
50
51 figure(); pzmap(Gcl); % Diagrama de polos y ceros
52 title('Diagrama de Polos y Ceros del Sistema en Lazo
    Cerrado'); grid on;

```

Resultados de Simulación

Al ejecutar el script anterior, se obtuvieron los siguientes resultados en el cmd, confirmando la función de transferencia derivada teóricamente y la ubicación de los polos:

Funcion de Transf. de la Planta $G_p(s)$:
 $G_{RLC} =$

$$\frac{4.545e04}{s^2 + 50s + 4.545e04}$$

F. Transf del Controlador Filtrado $G_c(s)$:
 $CPID =$

$$\frac{248.7s^2 + 1.091e05s + 1.129e07}{s^2 + 6403s}$$

F. Transf del Sist. en Lazo Cerrado $G_{CL}(s)$:
 $G_{cl} =$

$$\frac{1.13e07s^2 + 4.957e09s + 5.13e1}{s^4 + 6453s^3 + 1.167e07s^2 + 5.248e09s}$$

Ubicacion de los Polos (Raices del Polinomio
 $p =$

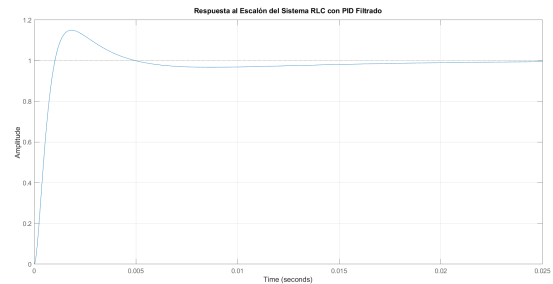
```

1.0e+03 *
-3.6170
-2.2316
-0.4685
-0.1357

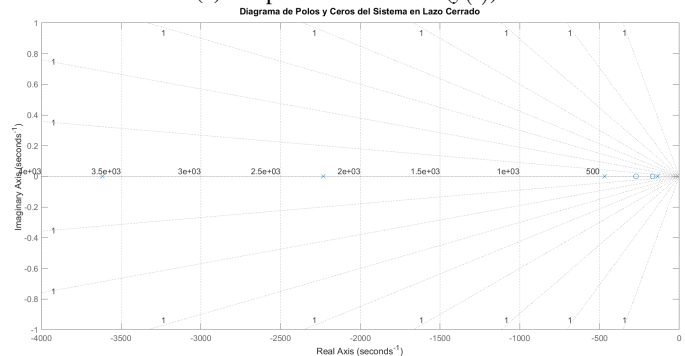
```

=> El sistema es ESTABLE (Todos los polos estan en

Gráficas de Respuesta y Estabilidad: Las figuras generadas por MATLAB ilustran el comportamiento dinámico y la ubicación de los polos.



(a) Respuesta al Escalón ($y(t)$)



(b) Diagrama de Polos y Ceros

Fig. 15: Análisis gráfico del sistema RLC en lazo cerrado.

Como se observa en la Figura 15, todos los polos del sistema se encuentran en el semiplano izquierdo y son **reales**, lo que garantiza la estabilidad del sistema. El sistema presenta un comportamiento **subamortiguado**. El sobreimpulso en la respuesta al escalón se explica por la influencia de los ceros del sistema y la dinámica introducida por el controlador PID, sin comprometer la estabilidad ni generar oscilaciones sostenidas.

CONCLUSIÓN DEL ANÁLISIS DE ESTABILIDAD

Aplicación de Pruebas de Estabilidad

A partir del Criterio de Routh-Hurwitz y la ubicación de los polos, se determina que el sistema en lazo cerrado es:

- **Estable:** Debido a que todos los polos tienen $\text{Re}(s) < 0$, lo que indica que se encuentran en el semiplano izquierdo del plano complejo.
- **No oscilante:** Debido a que todos los polos del sistema son **reales y negativos**,
- **Tipo de amortiguamiento:** El sistema presenta un comportamiento **subamortiguado**, ya que la respuesta al escalón exhibe un sobreimpulso inicial y una ligera oscilación antes de estabilizarse.

G. Replicando la actividad anterior mediante Python

Con el objetivo de validar los resultados obtenidos mediante MATLAB, se replica el análisis completo del sistema RLC con controlador PID utilizando el lenguaje **Python**, haciendo uso de las librerías **control**, **numpy** y **matplotlib**. Esta implementación permite verificar la función de transferencia en lazo cerrado, la ubicación de los polos y la respuesta dinámica del sistema.

Listing 7: Código Python para análisis del sistema RLC con PID

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import control as ctrl
4 # Parametros del Circuito RLC (Taller 5)
5 R = 5 # Resistencia (Ohmios)
6 L = 0.1 # Inductancia (Henrios)
7 C = 220e-6 # Capacitancia (Faradios)
8 # Parametros PID Sintonizados
9 Kp = 16.7573
10 Ki = 1762.8024
11 Kd = 0.03622
12 N = 6402.7121
13
14 # Funcion de Transferencia de la Planta
15 A = 1 / (L * C)
16 Num_p = [A]
17 Den_p = [1, R/L, A]
18 G_RLC = ctrl.TransferFunction(Num_p, Den_p)
19 print("Func. de Transferencia de la Planta G_p(s):")
20 print(G_RLC)
21
22 # Funcion de Transferencia del Controlador
23 Num_c = [(Kd*N + Kp), (Kp*N + Ki), (Ki*N)]
24 Den_c = [1, N, 0] # s*(s+N)
25 CPID = ctrl.TransferFunction(Num_c, Den_c)
26 print("Func. de Transf. del Controlador G_c(s):")
27 print(CPID)
28
29 # Sistema en Lazo Cerrado
30 Gcl = ctrl.feedback(CPID * G_RLC, 1)
31 print("Func. de Transf. en Lazo Cerrado G_CL(s):")
32 print(Gcl)
33
34 # Analisis de Estabilidad
35 p = ctrl.poles(Gcl)
36 print("Ubicacion de los Polos:")
37 print(p)
38
39 if np.all(np.real(p) < 0):
40     print("=> El sistema es ESTABLE")
41 else:
42     print("=> El sistema es INESTABLE")
43
44 # Respuesta al Escalon
45 t, y = ctrl.step_response(Gcl)
46 plt.figure()
47 plt.plot(t, y)
48 plt.title("Respuesta al Escalon del Sistema RLC con PID Filtrado")
49 plt.xlabel('Tiempo (s)')
50 plt.ylabel('Amplitud')
51 plt.grid(True)
52
53 # Diagrama de Polos y Ceros
54 plt.figure()
55 ctrl.pzmap(Gcl)
56 plt.title("Diagrama de Polos y Ceros del Sistema")
57 plt.grid(True)
58 plt.show()
```

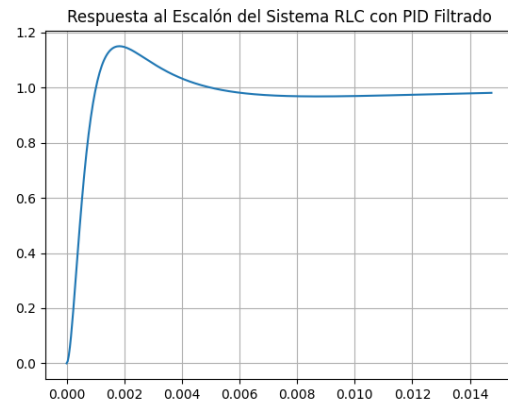
Resultados de la Consola de Python: La ejecución del script en Python generó los siguientes resultados:

```
248.7 s^2 + 1.091e+05 s + 1.129e+07
-----
s^2 + 6403 s
Función de Transferencia del Sistema en Lazo Cerrado G_CL(s):
<TransferFunction>: sys[4]
Inputs (1): ['u[0]']
Outputs (1): ['y[0]']

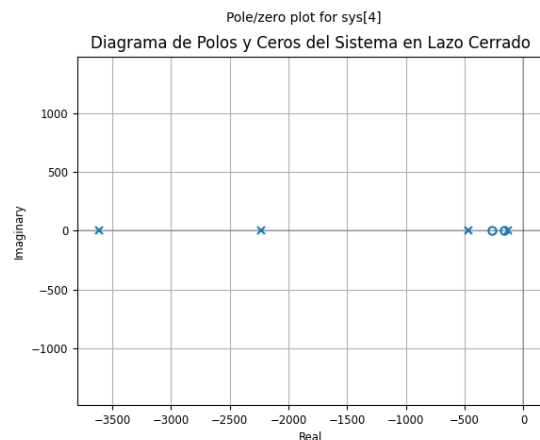
1.13e+07 s^2 + 4.957e+09 s + 5.13e+11
-----
s^4 + 6453 s^3 + 1.167e+07 s^2 + 5.248e+09 s + 5.13e+11
Ubicación de los Polos (Raíces del Polinomio Característico):
[-3617.02187643+0.j -2231.55313358+0.j -468.45669075+0.j
-135.68039924+0.j]
=> El sistema es ESTABLE (Todos los polos están en el semiplano izquierdo).
```

Fig. 16: Resultados obtenidos en Python: función de transferencia en lazo cerrado y ubicación de los polos.

Como se observa en la Figura 16, la función de transferencia del sistema en lazo cerrado y la ubicación de los polos coinciden con los valores obtenidos mediante MATLAB.



(a) Respuesta al escalón del sistema RLC con PID (Python)



(b) Diagrama de polos y ceros del sistema en lazo cerrado (Python)

Fig. 17: Análisis dinámico del sistema RLC con PID obtenido mediante Python.

Verificación Cruzada MATLAB vs Python

El análisis del sistema RLC con controlador PID fue implementado tanto en **MATLAB** como en **Python** utilizando la librería `control`. En ambos entornos se obtuvieron:

- La misma función de transferencia en lazo cerrado.
- Idéntica ubicación de polos (Re y negativos).
- Respuestas al escalón equivalentes.

Esto confirma la correcta formulación del modelo matemático y válida el uso de Python como una alternativa confiable a MATLAB para el análisis de sistemas de control.

IV. CÓDIGO

Todos los archivos de código utilizados en este taller están disponibles en el repositorio de GitHub. A continuación se describen los archivos principales:

- **Código Principal - Taller 6 (.m)**: Contiene los análisis de sistemas de primer orden estables e inestables, así como el sistema realimentado. [Obtener código Taller6.m](#)
- **Sistemas de Segundo Orden (.m)**: Implementación del análisis de sistemas con diferentes factores de amortiguamiento ($\zeta = 0, 0.2, 1, 2$).
[Obtener código de segundo orden](#)
- **Modelo de Simulink (.slx)**: Diagrama de bloques del sistema realimentado con $G_1(s)$, $G_2(s)$ y $H_1(s)$ implementado en Simulink.
[Obtener Diagrama de bloques - Simulink](#)
- **Análisis del Reto - Sistema RLC con PID (.m)**: Código MATLAB para el análisis de estabilidad del circuito RLC en lazo cerrado con controlador PID sintonizado.
[Obtener código del RETO en MATLAB](#)
- **Implementación en Python (.py)**: Réplica del análisis del sistema RLC con PID utilizando Python y la librería `control`. [Obtener código del RETO en Python](#)
- **Todos los códigos**: Todos los códigos se encuentran en: [Todos los codigos usados en este taller](#)

V. CONCLUSIONES

Este taller validó experimentalmente el criterio de Routh-Hurwitz como herramienta fundamental para el análisis de estabilidad de sistemas dinámicos lineales. Los resultados demostraron concordancia entre las predicciones teóricas del criterio, la ubicación de polos en el plano complejo y las respuestas temporales simuladas en MATLAB.

Los sistemas de primer y segundo orden confirmaron que la estabilidad está directamente determinada por la posición de los polos: aquellos con parte real negativa convergen a valores finitos, mientras que los ubicados en el semiplano derecho exhiben respuestas divergentes. El factor de amortiguamiento ζ demostró ser crucial en sistemas de segundo orden, definiendo comportamientos que van desde oscilaciones sostenidas ($\zeta = 0$, estabilidad marginal) hasta respuestas sobreamortiguadas ($\zeta > 1$).

El análisis del sistema realimentado evidenció cómo configuraciones inadecuadas pueden desestabilizar sistemas originalmente estables. La aplicación práctica más relevante fue la estabilización de un circuito RLC mediante un controlador PID sintonizado, cuyo polinomio característico $P(s) = s^4 + 6453s^3 + 1.167 \times 10^7 s^2 + 5.249 \times 10^9 s + 5.13 \times 10^{11}$ mostró ausencia de cambios de signo en la tabla de Routh, confirmando estabilidad teórica y práctica. El sistema presenta polos reales negativos (comportamiento subamortiguado), aunque exhibe un sobreimpulso en la respuesta al escalón debido a la influencia de los ceros introducidos por el controlador PID.

La verificación cruzada entre MATLAB y Python produjo resultados idénticos en funciones de transferencia, ubicación de polos y respuestas temporales, validando el uso de herramientas de código abierto para análisis profesionales de sistemas de control.

Las limitaciones principales del criterio de Routh-Hurwitz incluyen que solo proporciona información sobre estabilidad absoluta (estable/inestable) y no proporciona información sobre el comportamiento transitorio del sistema (tiempo de asentamiento, sobrepico, oscilaciones).

En conclusión, este taller proporcionó comprensión profunda de fundamentos de estabilidad en sistemas de control, validando experimentalmente conceptos teóricos mediante simulación numérica. Los conocimientos adquiridos constituyen la base esencial para diseño, análisis y optimización de sistemas de control en aplicaciones prácticas de ingeniería, y podrían extenderse mediante técnicas complementarias como el criterio de Nyquist, diagramas de Bode, o métodos en el espacio de estados.

REFERENCES

- [1] K. Ogata, *Ingeniería de Control Moderna*, 5ta ed. Madrid: Pearson Educación, 2010.
- [2] B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, 9th ed. Hoboken, NJ: John Wiley & Sons, 2010.
- [3] N. S. Nise, *Control Systems Engineering*, 7th ed. Hoboken, NJ: John Wiley & Sons, 2015.
- [4] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 13th ed. Upper Saddle River, NJ: Pearson Education, 2017.
- [5] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th ed. Upper Saddle River, NJ: Pearson Education, 2015.