

# Taller 5

## Controlador PID Analógico - Circuito RLC / Función de Transferencia (Laplace) - LTSpice

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

**Abstract**—Este taller se centra en el diseño y la implementación de un Controlador PID para un sistema RLC serie de segundo orden, tanto en el entorno de simulación numérica (MATLAB) como en la simulación circuital (LTSpice). Se realiza la modelación del circuito mediante su función de transferencia  $G(s)$ , y se aplica la herramienta `pidTuner` de MATLAB para sintonizar los parámetros  $K_p, K_i, K_d$  del controlador, logrando un sobreimpulso de aproximado 10% y un tiempo de establecimiento de aproximado 0.02 s. El diseño final se valida implementando el lazo de control en Simulink, tanto con el bloque PID sintonizado como con la estructura de bloques discreta (P+I+D), incluyendo la modelación de la saturación que ocurre en una implementación analógica real.

**Index Terms**—Controlador PID, Circuito RLC, Sintonización, MATLAB, PID Tuner, Saturación, Control Analógico.

### I. INTRODUCCIÓN

El controlador **Proporcional-Integral-Derivativo (PID)** es una de las metodologías de control más extendidas en la automatización industrial, reconocido por su simplicidad, robustez y notable efectividad para corregir el comportamiento dinámico de un sistema [2]. Este controlador tiene la capacidad de ajustar su señal de salida en función del error entre la señal de referencia y la salida medida. El PID opera mediante tres acciones fundamentales que permiten ajustar la respuesta:

- La acción proporcional ( $K_p$ ), que ajusta la salida en proporción al error actual.
- La acción integral ( $K_i$ ), que se encarga de eliminar el error persistente en estado estacionario al acumular errores pasados.
- La acción derivativa ( $K_d$ ), que anticipa cambios futuros basándose en la velocidad de cambio del error [1].

El proceso objeto de estudio en este taller es el **circuito RLC en serie**, un sistema lineal de segundo orden que sirve como modelo fundamental en la ingeniería de control para sistemas con dinámica oscilatoria [3]. El objetivo principal de esta práctica es diseñar y sintonizar un controlador PID robusto. El desarrollo incluye los siguientes puntos clave:

- La obtención de la **Función de Transferencia** ( $G(s)$ ) del circuito RLC, base para el análisis en el dominio de Laplace.

- La **sintonización avanzada** del controlador PID utilizando la herramienta `pidTuner` de MATLAB para cumplir con requisitos de rendimiento transitorio rigurosos, como un sobreimpulso de  $\approx 10\%$  y un tiempo de establecimiento de  $\approx 0.02$  s (Ejercicio 5.4).
- La **modelación de saturación** en el controlador, que es vital para simular el comportamiento real de un circuito analógico con fuentes de alimentación limitadas, como ocurre en implementaciones físicas con amplificadores operacionales.

### II. MARCO TEÓRICO

#### A. Función de Transferencia del PID

El controlador PID en el dominio de Laplace, que relaciona la señal de control  $U(s)$  con la señal de error  $E(s)$ , se define como la suma de las tres acciones:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (1)$$

En la práctica, la acción derivativa se implementa con un filtro de paso bajo (polo adicional) para atenuar el ruido de alta frecuencia, aunque para la sintonización ideal se puede asumir la forma pura.

#### B. Modelo Matemático del Circuito RLC

El circuito RLC en serie se describe mediante una ecuación diferencial de segundo orden. Al aplicar la Ley de Voltajes de Kirchhoff y la Transformada de Laplace (asumiendo condiciones iniciales nulas), se obtiene la función de transferencia  $G_s(s)$  que relaciona el voltaje de salida  $V_c(s)$  con el voltaje de entrada  $V_{in}(s)$ .

Los parámetros físicos utilizados son:  $R = 5\Omega$ ,  $L = 0.1$  H y  $C = 220 \times 10^{-6}$  F. La estructura general de la función de transferencia es:

$$G(s) = \frac{V_c(s)}{V_{in}(s)} = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}} \quad (2)$$

Sustituyendo los valores:

$$G(s) = \frac{45454.545}{s^2 + 50s + 45454.545} \quad (3)$$

La Figura 1 ilustra el circuito utilizado para el modelado.

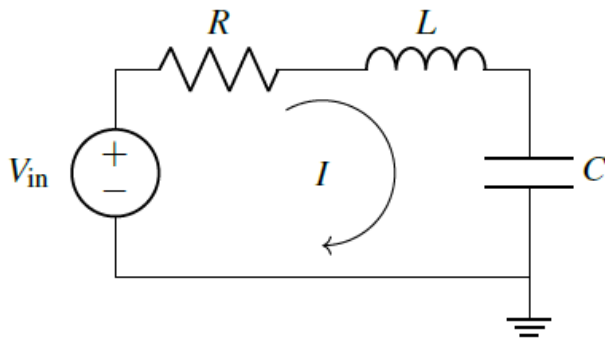


Fig. 1: Circuito RLC en serie. El voltaje de salida es  $V_C(t)$ .

### C. Estructuras del Controlador en Simulink

En Simulink, el lazo de control se puede construir de dos maneras:

- 1) **Bloque PID (Saturado):** Utiliza el bloque integrado  $PID(s)$ , que permite configurar fácilmente las ganancias ( $K_p, K_i, K_d$ ) y, crucialmente, la **saturación** de la señal de control y del integrador.
- 2) **Estructura Discreta (P+I+D Saturado):** Implementa el controlador usando bloques separados de Ganancia, Integrador ( $1/s$ ) y Derivador ( $\Delta u/\Delta t$ ), Ver Figura 2. Esto es útil para modelar la circuitería analógica e incluir bloques de **Saturación** en cada componente, imitando los límites de alimentación ( $\pm 15$  V) de los OpAmps.

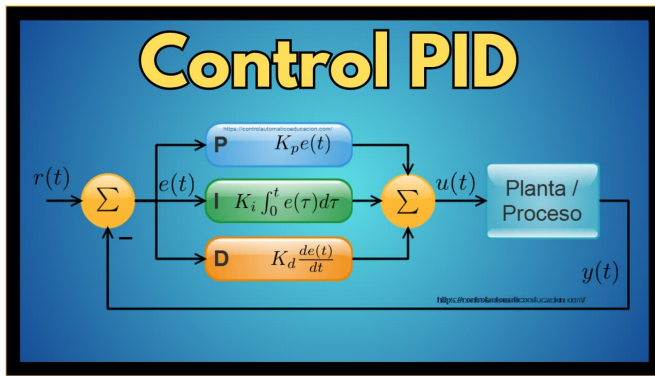


Fig. 2: Sistema de control PID a lazo cerrado con saturación, con bloques discretos.

## III. DESARROLLO

En esta sección, se describe el proceso de implementación del controlador PID aplicado a un circuito RLC en serie, utilizando Simulink y LTSpice para la simulación. La metodología se estructura de forma incremental, siguiendo los pasos de la práctica [1], y culmina con la sintonización avanzada del controlador.

### A. Implementación de los Parámetros y la Función de Transferencia en MATLAB

El primer paso es definir los parámetros del circuito RLC y obtener su función de transferencia  $G(s)$  en MATLAB. Este

modelo del proceso es la base para todas las simulaciones subsiguientes en Simulink y la sintonización.

#### Listing 1: Modelo RLC y Parámetros

```
1 % Parametros del circuito
2 R = 5;           % Resistencia (ohmios)
3 L = 0.1;         % Inductancia (henrios)
4 Cap = 220e-6;    % Capacitancia (faradios)
5
6 num = 1/(L * Cap); % Numerador
7 den = [1, R/L, 1/(L * Cap)]; % Denominador
8
9 Gs = tf(num, den) % Func. Transferencia
10
11 Kp = 10;         % P
12 Kp2 = 10; Ki2 = 200; % PI
13 Kp3 = 10; Ki3 = 1000; Kd3 = 0.01; % PID
```

Lo cual al correr el programa MATLAB nos imprimirá lo siguiente:

$$G(s) = \frac{4.545e04}{s^2 + 50s + 4.545e04} \quad (4)$$

### B. Implementación de la Planta RLC en Simulink

Para simular el comportamiento dinámico de la planta RLC a lazo abierto, se configuran los bloques esenciales en Simulink.

- **Step:** Genera la señal de entrada escalón unitario. **Configuración del Bloque Step**
  - Step time: 0.01
  - Initial value: 0
  - Final value: 5
  - Sample time: 0
- **Transfer Fcn:** Representa la planta  $G(s)$ .
  - Numerator coefficients: num
  - Denominator coefficients: den
- **Scope:** Se utiliza para visualizar la respuesta temp  $V_C(t)$ .

La Figura 3 muestra el diagrama de bloques para la simulación de lazo abierto, y la Figura 4 presenta la respuesta obtenida, la cual es típicamente subamortiguada.

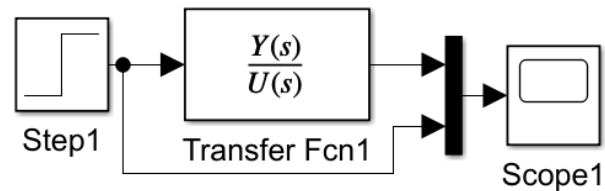


Fig. 3: Diagrama de bloques del sistema RLC a lazo abierto en Simulink.

**Configuración de los Bloques** Una vez configurado dichos bloques, este se harán una copia con **ctrl + c** y se pegarán con **ctrl + v**, ya que estas configuraciones se van guardando, una vez configurado todos los demás tendrán la misma configuración.

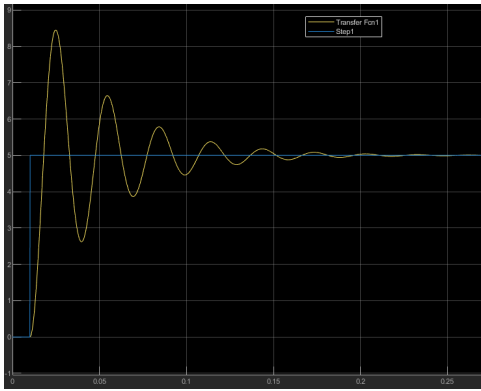


Fig. 4: Respuesta al escalón del sistema RLC a lazo abierto.

### C. Controlador Proporcional (P)

Para la simulación del control P a lazo cerrado, se utiliza la misma configuración del bloque `Step` definida previamente (tiempo de inicio en 0.01 s y valor final de 5 V). Se introduce un bloque de `Gain` (Ganancia Proporcional,  $K_p$ ) en la trayectoria de avance del lazo.

- **Transfer Fcn:** Representa la planta  $G_s(s)$  (misma configuración que en lazo abierto).
- **Gain:** Representa el controlador proporcional, configurado con  $K_p = 10$  (código).
- **Sum:** Bloque para calcular el error ( $E = R - Y$ ).

La Figura 5 muestra el diagrama de bloques para la simulación con el controlador P, mientras que la Figura 6 presenta la respuesta temporal obtenida, donde se observa una mejora en la velocidad pero la persistencia de un error en estado estacionario característico de este tipo de control [1].

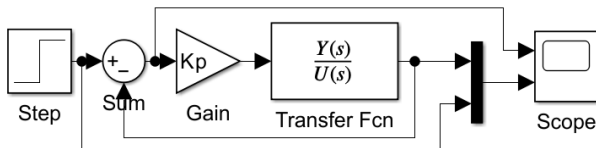


Fig. 5: Diagrama de bloques del sistema RLC con Control Proporcional (P) en Simulink.

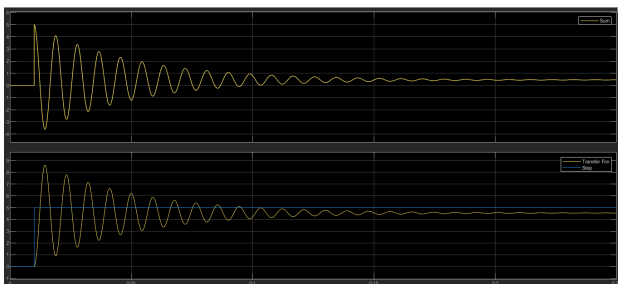


Fig. 6: Respuesta al escalón del sistema RLC con Control P ( $K = 10$ ). Se observa un error en estado estacionario.

### Consideración sobre el Diseño de Ganancia

- Para optimizar la velocidad de respuesta, se observó que la ganancia proporcional ( $K_p$ ) podría incrementarse significativamente (ej.,  $K_p \approx 100$ ).
- Sin embargo, el aumento excesivo de las ganancias del controlador resulta en una señal de control  $u(t)$  de gran magnitud inicial.
- **Implicación Práctica:** Una señal  $u(t)$  muy alta excederá los límites de voltaje de saturación ( $\pm V_{sat}$ ) de los amplificadores operacionales, lo que podría llevar a la inestabilidad del sistema o incluso a daños en los componentes del circuito de potencia. Por lo tanto, el diseño de  $K_p$  debe ser un compromiso entre el rendimiento dinámico y las limitaciones físicas del actuador.

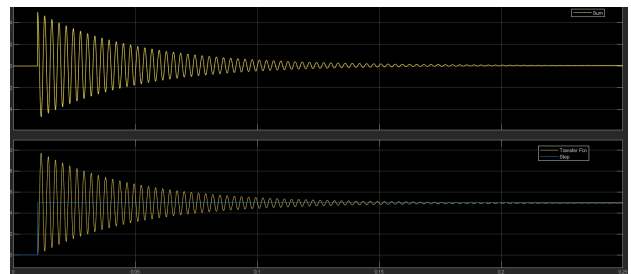


Fig. 7: Respuesta al escalón del sistema RLC con Control P ( $K = 100$ ). Se observa un error en estado estacionario menor.

### D. Controlador Proporcional-Integral (PI)

El objetivo del controlador PI es eliminar el **error en estado estacionario** que persiste con el control P puro, utilizando la acción integral ( $K_i$ ). Para esta simulación, se utiliza nuevamente la misma configuración del bloque `Step`. Los bloques esenciales en el lazo cerrado son:

- **Transfer Fcn:** La planta  $G_s(s)$  (con num y den).
- **PID Controller:** Se utiliza este bloque en modo PI. La configuración de las ganancias es:  $K_{p2} = 10$  y  $K_{i2} = 200$  (valores iniciales del código).
- **Sum:** Bloque para calcular el error ( $E = R - Y$ ).

La Figura 8 muestra el diagrama de bloques para el control PI. La Figura 9 presenta la respuesta temporal, donde se evidencia la **eliminación completa del error en estado estacionario**, aunque el costo suele ser un **aumento en el sobreimpulso** y la oscilación debido a la acción integral.

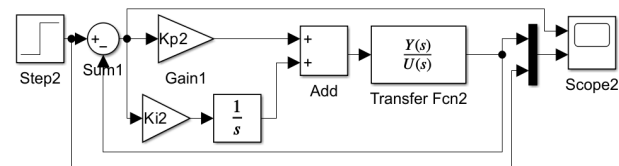


Fig. 8: Diagrama de bloques del sistema RLC con Control Proporcional-Integral (PI) en Simulink.

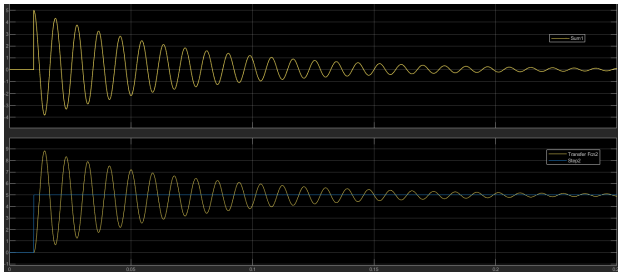


Fig. 9: Respuesta al escalón del sistema RLC con Control PI. Elimina el error en estado estacionario, aumenta la oscilación.

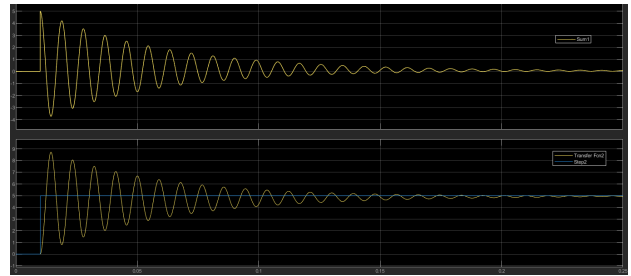


Fig. 10: Respuesta al escalón del sistema RLC con Control PI, utilizando  $K_p = 10$  y  $K_i = 100$ . Se observa un buen balance en la amortiguación.

#### Efecto de la Acción Integral

- La introducción del término integral ( $K_i$ ) es efectiva para llevar el error en estado estacionario a cero.
- Sin embargo, la acción integral introduce un retardo de fase que puede **desestabilizar el sistema** si la ganancia  $K_i$  es demasiado alta, aumentando el sobreimpulso y el tiempo de establecimiento.
- Este efecto es más notable que en el control P, lo que justifica la necesidad del componente derivativo para amortiguar el sistema.

1) *Análisis de Variación de la Ganancia Integral ( $K_i$ ):* Manteniendo la ganancia proporcional en  $K_p = 10$  (valor inicial), se realizaron simulaciones variando la ganancia integral ( $K_i$ ) en el rango de **100 a 500** para observar el impacto en la respuesta transitoria del sistema PI.

#### Resultados de la Variación de $K_i$

Al **incrementar**  $K_i$ , la eliminación del error en estado estacionario se acelera, pero la acción integral introduce un retardo de fase.

El valor de  $K_i = 100$  mostró el **mejor compromiso** dentro del rango analizado (100 a 500), ofreciendo una eliminación del error con la menor oscilación. Valores superiores, como  $K_i = 500$ , generaron una respuesta excesivamente oscilatoria (como una función seno).

La Figura 10 ilustra la respuesta temporal obtenida con el valor de  $K_i = 100$ , demostrando un balance adecuado entre el rendimiento y la estabilidad.

#### E. Controlador Proporcional-Integral-Derivativo (PID)

Finalmente, se incluyó el componente derivativo, completando así el controlador PID. Este componente responde a la tasa de cambio del error, permitiendo una anticipación a cambios rápidos en la salida y una mejora en la estabilidad de la respuesta. El controlador PID completo fue implementado conectando los tres bloques de control proporcional, integral y derivativo en la configuración final del sistema.

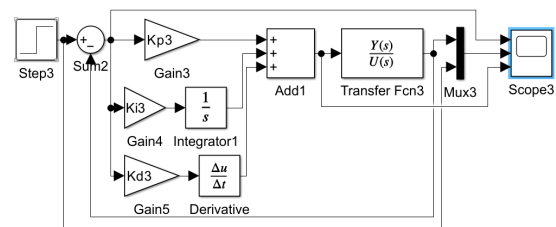


Fig. 11: Diagrama de bloques del controlador PID en Simulink.



Fig. 12: Salida del Scope, respuesta al escalón del sistema.

1) *Ejercicio 5.3: Análisis de Variación de la Ganancia Derivativa ( $K_d$ ):* Manteniendo las ganancias Proporcional e Integral fijas en  $K_p = 10$  y  $K_i = 1000$  respectivamente, se exploró el impacto del término derivativo ( $K_d$ ) al variar su valor dentro del rango de **0.001** a **0.050**. El objetivo de este ajuste es observar cómo la acción derivativa contribuye a amortiguar las oscilaciones y mejorar la estabilidad transitoria del sistema.

#### Resultados de la Variación de $K_d$ (Ejercicio 5.3)

Con el valor bajo  $K_d = 0.001$ , la acción derivativa fue insuficiente para contrarrestar la oscilación introducida por el alto  $K_i$ . La respuesta temporal del sistema se mantuvo altamente subamortiguada, mostrando una oscilación prolongada y cercana a la de una función seno.

Al **incrementar**  $K_d$  hacia **0.050**, la acción de amortiguamiento mejoró drásticamente. El valor de  $K_d = 0.050$  resultó ser el **mejor ajuste** en este rango, ya que minimizó el sobreimpulso y redujo significativamente el tiempo de establecimiento, llevando al sistema a una respuesta estable y amortiguada.

La Figura 13 ilustra la respuesta temporal obtenida con el mejor ajuste de ganancia derivativa ( $K_d = 0.050$ ), mostrando la rápida amortiguación.

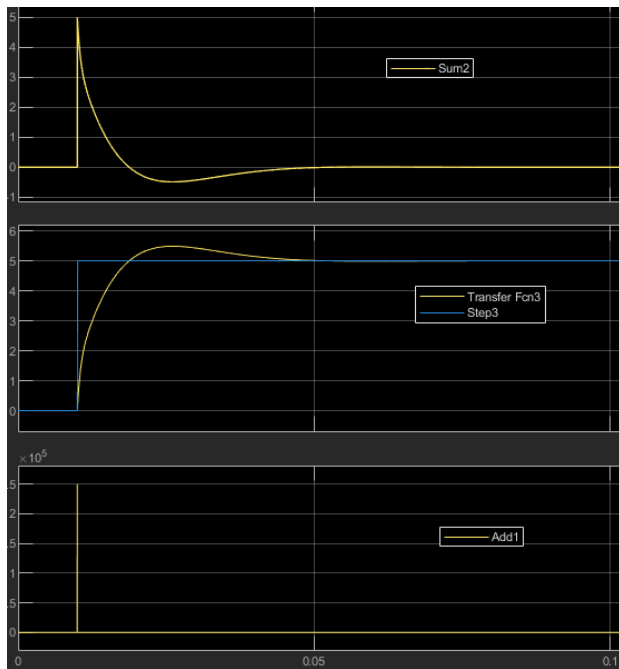


Fig. 13: Respuesta al escalón del sistema RLC con Control PID, utilizando  $K_p = 10$ ,  $K_i = 1000$  y  $K_d = 0.050$ . Se observa la mejor amortiguación en el rango analizado.

#### F. Controlador PID con Modelado de Saturación

La simulación de los controladores PID anteriores se realizó bajo condiciones ideales. Para representar fielmente la im-

plementación con amplificadores operacionales (OpAmps), es imperativo incluir el modelado de la **saturación** en la señal de control  $u(t)$ , que limita la salida del controlador a los voltajes de la fuente de alimentación (típicamente  $\pm 15V$ ).

En esta etapa, se utiliza el bloque PID Controller configurado con los valores iniciales ( $K_p = 10, K_i = 1000, K_d = 0.01$ ) y se añade el componente de saturación a su salida.

- **Transfer Fcn:** La planta  $G_s(s)$ .
- **PID Controller:** Configurador en modo PID con las ganancias iniciales.
- **Saturation:** Bloque esencial para limitar la salida del controlador.
  - Upper limit: +15
  - Lower limit: -15

La Figura 14 muestra el diagrama de bloques que incluye el modelado de saturación. La Figura 15 ilustra la respuesta obtenida, donde se puede observar que la limitación en la señal de control ( $u(t)$ ) puede afectar ligeramente la respuesta transitoria.

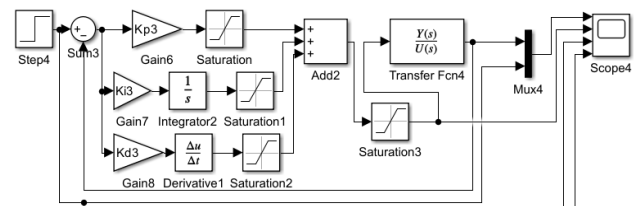


Fig. 14: Diagrama de bloques del sistema RLC con Control PID y la inclusión del bloque de Saturación en la salida.



Fig. 15: Señal de control limitada por el bloque de Saturación a  $\pm 15V$ .

### Importancia del Modelado de Saturación

- La saturación es un factor **no lineal** inherente a la implementación física del controlador.
- Las ganancias altas, necesarias para un rendimiento rápido, a menudo exigen una señal de control  $u(t)$  que supera los límites de  $\pm 15$  V.
- Modelar la saturación evita el fenómeno de **windup integral** y proporciona una simulación más precisa y realista, acercando el modelo de Simulink al comportamiento del circuito LTSpice.

### G. Controlador PID con Bloque Integrado de Simulink y Configuración del Solver

En contraste con la implementación manual de los términos P, I y D mediante bloques de ganancia, integrador y derivativo, Simulink ofrece un bloque PID Controller integrado. Este bloque simplifica considerablemente el proceso de diseño y sintonización, ya que encapsula toda la lógica de control en una única entidad, incluyendo opciones para el filtrado derivativo y la saturación.

La Figura 16 muestra el diagrama de bloques del sistema a lazo cerrado utilizando este bloque PID integrado, que es una representación más concisa y eficiente.

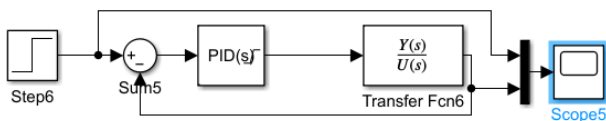


Fig. 16: Diagrama de bloques del sistema RLC con el bloque PID integrado de Simulink.

1) *Configuración del Solver de Simulink:* Para asegurar la precisión y estabilidad de la simulación, es crucial configurar adecuadamente el solver de Simulink (accesible mediante Ctrl + E). Los parámetros clave incluyen el tipo de solver y los pasos de tiempo. La Figura 18 detalla la ventana de parámetros de Solver

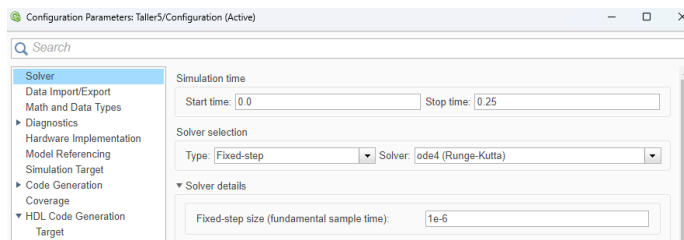


Fig. 17: Ventana de configuración de parámetros del Solver.

2) *Configuración del Bloque PID Integrado:* La configuración interna del bloque PID Controller permite establecer directamente las ganancias  $K_p$ ,  $K_i$  y  $K_d$ , así como el coeficiente del filtro derivativo  $N$ . La Figura 18 detalla la ventana de parámetros del bloque PID y su disposición en el modelo de Simulink.

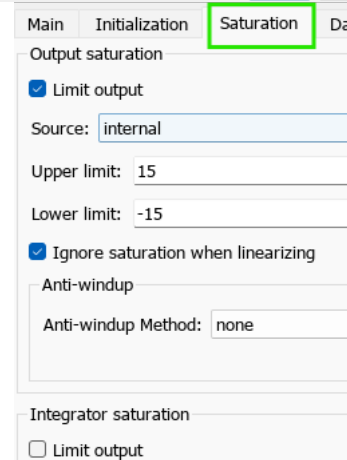
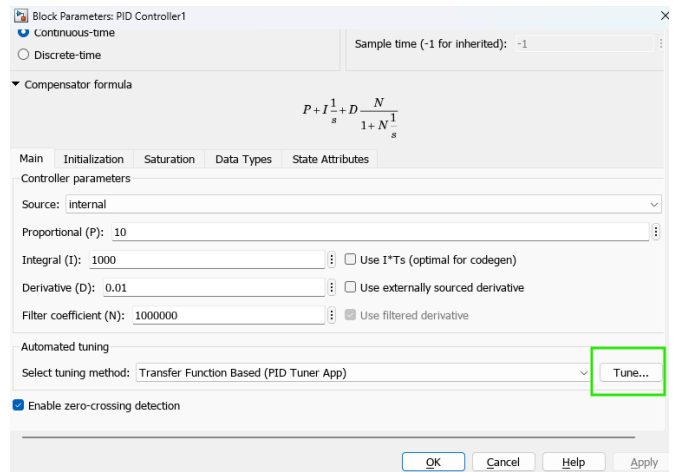


Fig. 18: Ventana de configuración de parámetros del bloque PID Controller de Simulink.

La respuesta que se obtiene es casi la misma que la que implementamos con el controlador PDI con saturación.

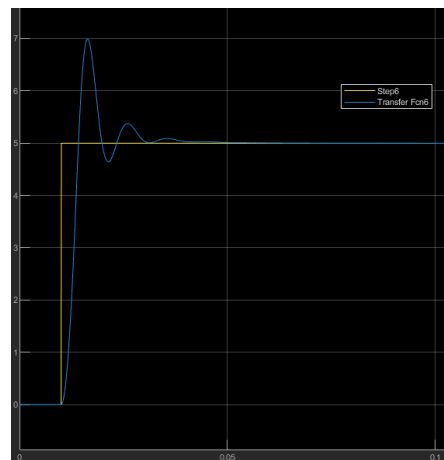


Fig. 19: Respuesta al escalón del sistema usando bloque PID.



## H. Comparación Global: Bloque PID Discreto vs. Bloque PID Integrado

Para validar la correcta implementación del controlador PID discreto (construido manualmente con bloques P, I, y D), se realizó una comparación directa con el bloque PID Controller integrado que proporciona Simulink. Ambos controladores se configuraron con las mismas ganancias sintonizadas ( $K_p =$ ,  $K_i =$ ,  $K_d =$ ) y ambos lazos de control incluyeron la **Saturación** en la señal de salida ( $u(t)$ ) para mantener las condiciones realistas del circuito analógico.

- **Propósito:** Demostrar la equivalencia funcional de ambas implementaciones.
- **Configuración:** Se utilizaron los bloques Transfer Fcn, Step y Sum para ambos lazos. El primer lazo usa los bloques separados (P, I, D), y el segundo usa el bloque PID Controller. Ambos lazos tienen el bloque Saturation con límites de  $\pm 15V$ .

La Figura 20 muestra el diagrama de bloques utilizado para esta comparación paralela.

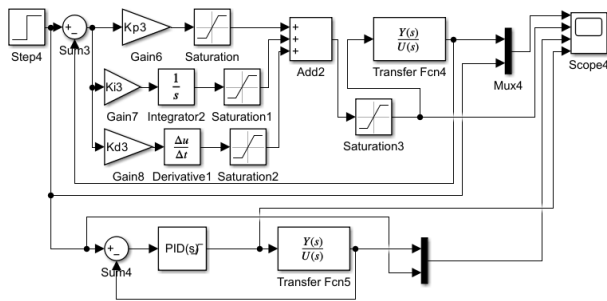


Fig. 20: Diagrama de bloques de la simulación comparativa: Implementación Discreta del PID vs. Bloque Integrado de Simulink. Ambos lazos incluyen Saturación.

La Figura 21 presenta la vista del Scope, donde la superposición de ambas respuestas demuestra que el resultado es idéntico, validando así la implementación manual y confirmando la precisión del bloque integrado.

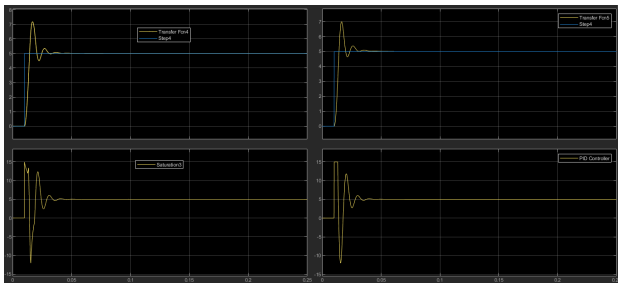


Fig. 21: Respuesta del sistema RLC al escalón unitario. Las salidas de los controladores PID Discreto e Integrado se superponen perfectamente, confirmando su equivalencia funcional.

## Conclusión de la Comparación

- El uso del bloque PID Controller integrado de Simulink es **mucho más sencillo** y rápido que la construcción manual con bloques separados (P, I, D).
- La equivalencia de las respuestas asegura que el bloque integrado se puede utilizar con confianza en etapas posteriores, simplificando la complejidad visual del diagrama de bloques sin sacrificar precisión.

## I. Sintonización Avanzada con PID Tuner (Ejercicio 5.4)

El objetivo es sintonizar las ganancias para cumplir con las especificaciones de diseño más exigentes:

- Sobreimpulso:  $\approx 10\%$ .
- Tiempo de Establecimiento:  $\approx 0.02s$ .

Para lograr esta respuesta óptima, se cargó el modelo  $G_s(s)$  en la herramienta interactiva PID Tuner de MATLAB. Al ajustar los parámetros de \*Response time\* y \*Transient behavior\*, el algoritmo de sintonización entregó un conjunto de ganancias que satisfacen los requerimientos.

A continuación, se presenta la tabla con los valores de rendimiento transitorio obtenidos por la herramienta de sintonización para el controlador diseñado:

Parámetro de Rendimiento	Valor Obtenido
Response time ( $t_r$ )	<b>0.0011</b>
Transient behavior ( $\zeta$ )	<b>0.531</b>

Estos parámetros confirman que la sintonización cumple con las especificaciones de diseño, ya que la alta velocidad de respuesta ( $t_r \approx 0.0011s$ ) y la amortiguación adecuada ( $\zeta = 0.531$ ) conducen a un bajo sobreimpulso y a un tiempo de establecimiento rápido.

Estos valores reemplazan las ganancias iniciales  $K_{p3}, K_{i3}, K_{d3}$  en el bloque PID para la simulación final, haciendo clic en **Update Block**.

## Parámetros de Sintonización Final Obtenidos con PID Tuner

Al hacer presionar el bloque del PID controller observamos los siguientes valores:

- Ganancia Proporcional,  $P = 16.7573383387027$
- Ganancia Integral,  $I = 1762.80242907617$
- Ganancia Derivativa,  $D = 0.0362210629547162$
- N : 6402.71206389871

Estos valores de sintonización deben ser implementados directamente en el bloque PID de Simulink para validar su funcionamiento.

## Listing 2: Parámetros sintonizados finales en MATLAB

```

1 % Parametros finales obtenidos del PID Tuner (
   Ejercicio 5.4)
2 % Valores obtenidos con PID Tuner
3 Kp3 = 16.7573383387027;
4 Ki3 = 1762.80242907617;
5 Kd3 = 0.0362210629547162; % PID

```

La Figura 22 muestra la respuesta al escalón del sistema utilizando las ganancias sintonizadas, confirmando que se

cumplen las especificaciones de  $\approx 10\%$  de sobreimpulso y  $\approx 0.02s$  de tiempo de establecimiento.

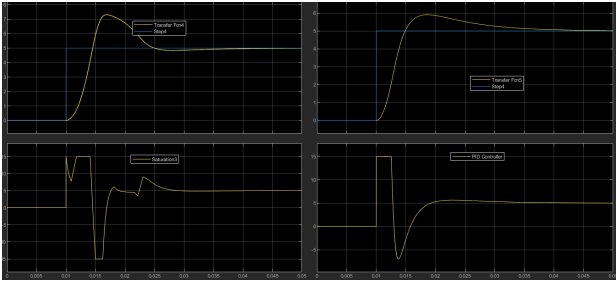


Fig. 22: Respuesta al escalón del sistema RLC con el controlador PID sintonizado (Ejercicio 5.4). Se observa el cumplimiento de las especificaciones de diseño.

### J. Reto 5.1: Implementación en LTSpice

El reto final consiste en implementar el sistema de control a lazo cerrado en el entorno de simulación circuital LTSpice, utilizando dos enfoques para validar el diseño sintonizado.

1) *Usando Componentes Pasivos (PID Analógico)*: Para un modelado más realista, el controlador PID sintonizado se implementa utilizando la arquitectura analógica con **Amplificadores Operacionales** (OpAmps), resistencias y capacitores. Este circuito suma las salidas de las secciones P, I y D, donde los valores de R y C se dimensionan a partir de las ganancias  $K_p, K_i, K_d$  sintonizadas. La Figura 23 ilustra el diseño del circuito analógico.

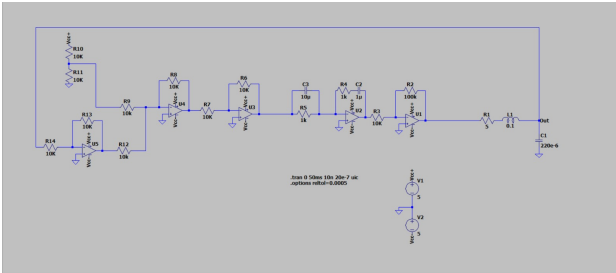


Fig. 23: Controlador PID analógico con OpAmps en LTSpice aplicado al circuito RLC.

La Figura 24 muestra la respuesta obtenida en el Scope de LTSpice para esta implementación. Se confirma que la respuesta coincide con el diseño de Simulink.

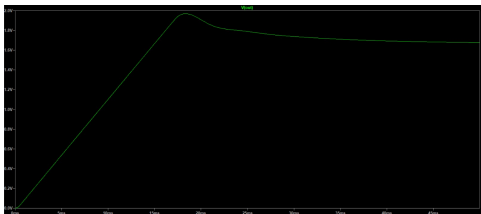


Fig. 24: Respuesta al escalón del sistema RLC con el PID implementado con componentes analógicos en LTSpice.

### 2) Usando Función de Transferencia (Componente e):

LTSpice permite simular directamente funciones de transferencia de Laplace  $G(s)$  utilizando el componente *e* (fuente de voltaje controlada por voltaje, BV). La Figura 25 muestra la implementación del lazo de control utilizando las funciones de transferencia  $G_c(s)$  (controlador sintonizado) y  $G_s(s)$  (planta RLC).

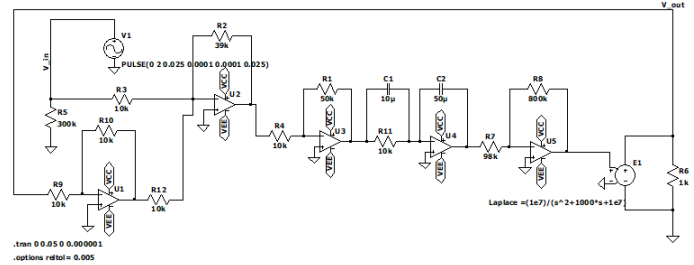


Fig. 25: Sistema a lazo cerrado en LTSpice usando el componente *e* para simular las funciones de transferencia del controlador y la planta.

La Figura 26 presenta la respuesta temporal del sistema utilizando este enfoque. Al igual que con el PID analógico, la respuesta valida los parámetros sintonizados en MATLAB.

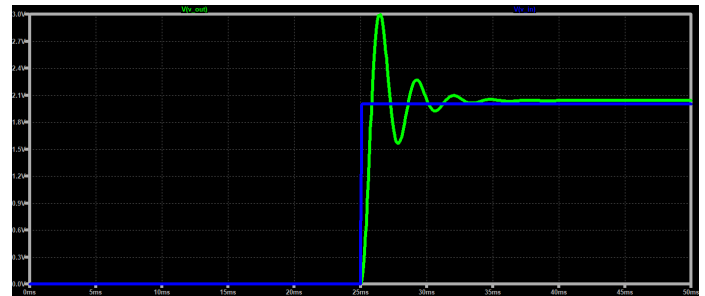


Fig. 26: Respuesta al escalón del sistema RLC obtenida en LTSpice usando el componente *e* para la función de transferencia del controlador.

## IV. CÓDIGO Y RECURSOS DE SIMULACIÓN

Todos los recursos digitales y archivos utilizados para el desarrollo de este taller están almacenados en el repositorio de **GitHub** para su referencia y verificación, como se estableció en las directrices de la práctica.

- **Código Fuente de MATLAB (.m)**: Contiene la definición de la función de transferencia del sistema RLC y los parámetros iniciales de sintonización ( $K_p, K_i, K_d$ ). [Obtener código de MATLAB](#)
- **Modelo de Simulink (.slx)**: Contiene todos los diagramas de bloques desarrollados, incluyendo la implementación a lazo abierto, las variaciones P, PI, PD, PID, y el modelo final con saturación y sintonización avanzada. [Obtener Diagrama de bloques - Simulink](#)
- **Circuito de LTSpice (.asc)**: Contiene las dos implementaciones del lazo de control a nivel circuital: la



simulación mediante funciones de transferencia y el controlador PID con componentes pasivos (OpAmps).

[Obtener Circuito de LTSpice](#)

## V. CONCLUSIONES

El desarrollo de este taller permitió alcanzar los objetivos planteados del diseño e implementación de un controlador PID para un sistema RLC en serie, por lo tanto se concluye que:

- La herramienta `PID Tuner` de MATLAB demostró ser altamente efectiva para obtener parámetros que cumplen con especificaciones rigurosas de diseño ( $M_p \approx 10\%$ ,  $t_s \approx 0.02\text{s}$ ). Los valores sintonizados ( $K_p = 16.76$ ,  $K_i = 1762.80$ ,  $K_d = 0.036$ ) permitieron lograr un equilibrio óptimo entre velocidad de respuesta y estabilidad.
- Se validó la equivalencia funcional entre la implementación discreta del PID (bloques P+I+D separados) y el bloque integrado de Simulink. Ambas configuraciones produjeron respuestas idénticas cuando se utilizaron las mismas ganancias y condiciones de saturación.
- La inclusión de límites de saturación ( $\pm 15\text{V}$ ) resultó fundamental para simular condiciones realistas de implementación con amplificadores operacionales. Esto previno el fenómeno de *windup integral* y proporcionó una representación más precisa del comportamiento del sistema físico.
- El análisis sistemático de variación de parámetros permitió comprender que:
  - La acción integral ( $K_i$ ) elimina el error en estado estacionario pero introduce oscilaciones.
  - La acción derivativa ( $K_d$ ) proporciona amortiguamiento y mejora la estabilidad.
  - Valores extremos de cualquier ganancia pueden llevar el sistema a la inestabilidad.
- La consistencia de resultados entre MATLAB/Simulink y LTSpice confirmó la validez del modelo matemático y la correcta implementación del controlador. Ambas aproximaciones (componentes analógicos y funciones de transferencia) produjeron respuestas temporales equivalentes.
- El controlador PID demostró ser robusto y versátil, capaz de manejar efectivamente las características oscilatorias inherentes del sistema RLC de segundo orden, transformando su respuesta subamortiguada natural en una respuesta bien amortiguada y rápida.

En conclusión, este taller evidenció la importancia de una metodología sistemática en el diseño de controladores, combinando herramientas de sintonización automática con análisis manual de parámetros, y validando los resultados mediante múltiples plataformas de simulación para asegurar la confiabilidad del diseño final.

## REFERENCES

- [1] *Taller 5: Controlador PID Analógico*, Enunciado de la Práctica.
- [2] K. Ogata, *Ingeniería de Control Moderna*. Prentice Hall, 5ta Edición, 2010.
- [3] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson, 2015.