



Dirección General de  
**EDUCACIÓN TÉCNICA Y  
FORMACIÓN PROFESIONAL**

Ministerio de  
**EDUCACIÓN**



**CÓRDOBA**  
*Seguimos haciendo*

Materia: Sistemas de Control y Servicios

## **TRABAJO PRÁCTICO #3**

**Docente: Cristian Gonzalo Vera**

**Alumno: José Augusto Orsili**

**Año 2024**

## 1. Simulación y análisis de un control dinámico de lazo abierto

El control de lazo abierto a simular y analizar será un cargador de baterías:

En este caso, simulo la carga de una batería con una corriente constante y verifico cómo la carga acumulada de la batería cambia con el tiempo.

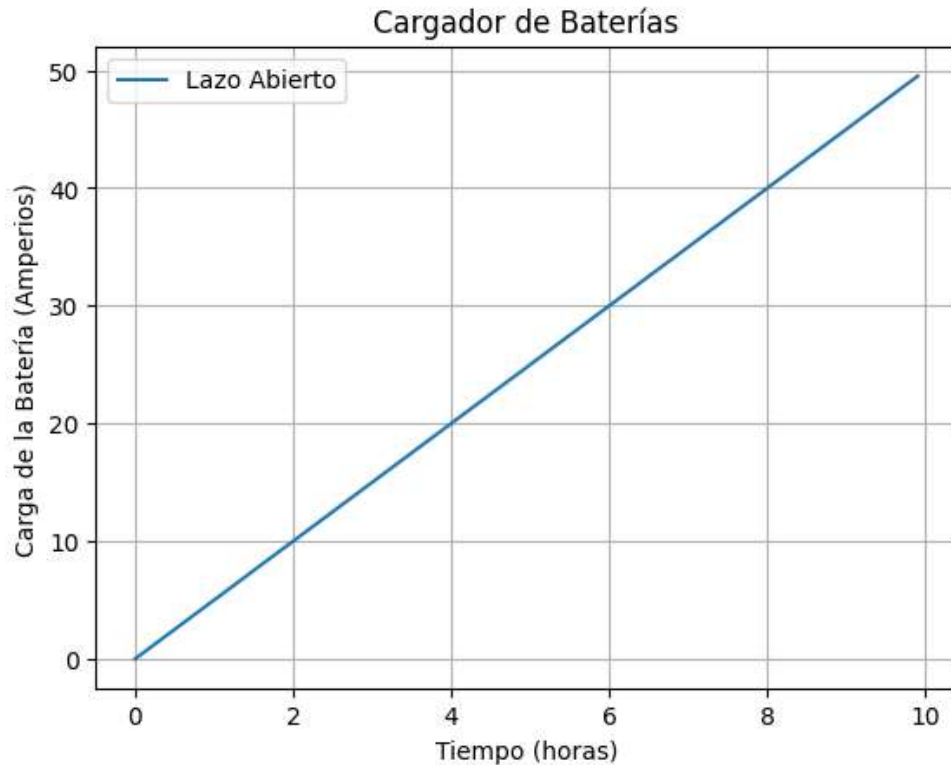
```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros del sistema
tasa_carga_deseada = 5 # Amperios por hora
capacidad_bateria = 7 # Capacidad total de la batería en amperios
tiempo_simulacion = 10 # Tiempo de simulación en horas
dt = 0.1 # Incremento de tiempo en horas

# Inicialización del sistema
tiempo = np.arange(0, tiempo_simulacion, dt) # Array de tiempo
corriente_carga_abierto = np.ones_like(tiempo) * tasa_carga_deseada # Corriente de carga constante

# Simulación del sistema en lazo abierto
carga_bateria_abierto = np.zeros_like(tiempo) # Inicialización de la carga de la batería
for i, t in enumerate(tiempo):
    if i > 0:
        carga_bateria_abierto[i] = carga_bateria_abierto[i - 1] + corriente_carga_abierto[i] * dt

# Visualización de los resultados del lazo abierto
plt.plot(tiempo, carga_bateria_abierto, label="Lazo Abierto")
plt.title('Cargador de Baterías')
plt.xlabel('Tiempo (horas)')
plt.ylabel('Carga de la Batería (Amperios)')
plt.grid(True)
plt.legend()
plt.show()
```



A continuación, simulo el cargador de baterías en lazo cerrado, donde se controla la corriente de carga para mantener una tasa de carga específica y evitar la sobrecarga de la batería. En este caso, controlo la corriente de carga para mantener la tasa de carga deseada y evitar que la batería se sobrecargue.

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros del sistema
tasa_carga_deseada = 1 # Amperios por hora
capacidad_bateria = 7 # Capacidad total de la batería en amperios
tiempo_simulacion = 10 # Tiempo de simulación en horas
dt = 0.1 # Incremento de tiempo en horas

# Inicialización del sistema
tiempo = np.arange(0, tiempo_simulacion, dt) # Array de tiempo

# Simulación del sistema en lazo abierto
corriente_carga_abierto = np.ones_like(tiempo) * tasa_carga_deseada #
Corriente de carga constante en lazo abierto
```

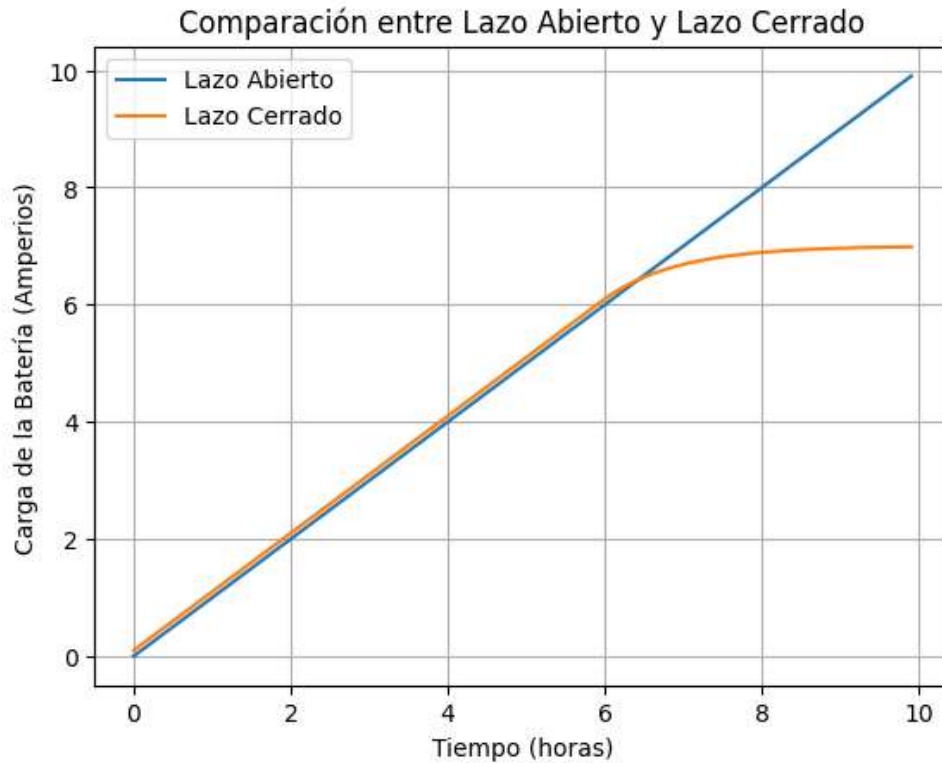
```

carga_bateria_abierto = np.zeros_like(tiempo) # Inicialización de la carga
de la batería en lazo abierto
for i, t in enumerate(tiempo):
    if i > 0:
        carga_bateria_abierto[i] = carga_bateria_abierto[i - 1] +
corriente_carga_abierto[i] * dt

# Simulación del sistema en lazo cerrado
carga_bateria_cerrado = np.zeros_like(tiempo) # Inicialización de la carga
de la batería en lazo cerrado
corriente_carga_cerrado = np.zeros_like(tiempo) # Corriente de carga en
lazo cerrado
for i, t in enumerate(tiempo):
    # Calcular la corriente de carga para mantener la tasa de carga deseada
en lazo cerrado
    if carga_bateria_cerrado[i-1] < capacidad_bateria:
        corriente_carga_cerrado[i] = min(tasa_carga_deseada,
capacidad_bateria - carga_bateria_cerrado[i-1])
    # Actualizar la carga de la batería en lazo cerrado
    carga_bateria_cerrado[i] = carga_bateria_cerrado[i - 1] +
corriente_carga_cerrado[i] * dt if i > 0 else corriente_carga_cerrado[i] *
dt

# Visualización de los resultados
plt.plot(tiempo, carga_bateria_abierto, label="Lazo Abierto")
plt.plot(tiempo, carga_bateria_cerrado, label="Lazo Cerrado")
plt.title('Comparación entre Lazo Abierto y Lazo Cerrado')
plt.xlabel('Tiempo (horas)')
plt.ylabel('Carga de la Batería (Amperios)')
plt.grid(True)
plt.legend()
plt.show()

```



### **Conclusiones**

Al comparar los sistemas de lazo abierto y lazo cerrado mediante el gráfico, se puede observar una diferencia notable en el comportamiento de la carga de la batería:

- En el caso del sistema de lazo abierto, la carga de la batería continúa indefinidamente a medida que transcurre el tiempo. Esto se debe a que en este tipo de sistema no existe ningún mecanismo de control que regule la corriente de carga o el nivel máximo de carga de la batería. Por lo tanto, la corriente de carga se mantiene constante y la batería continúa cargándose sin restricciones.
- Por otro lado, en el sistema de lazo cerrado, donde se controla tanto la cantidad de amperios por hora como el nivel máximo de carga de la batería, se observa un comportamiento diferente. Una vez que la batería alcanza su capacidad máxima de carga, el sistema detiene automáticamente el proceso de carga. Esto se debe a que el controlador ajusta la corriente de carga para mantener la tasa de carga deseada y evitar la sobrecarga de la batería. Como resultado, una vez completada la carga, el sistema se detiene y no continúa el proceso de carga, lo que garantiza una gestión más eficiente y segura de la batería.

Resumiendo, mientras que el sistema de lazo abierto no controla ni regula la carga de la batería, lo que puede llevar a una carga excesiva y potencialmente dañina, el sistema de lazo cerrado ofrece un control preciso y automatizado que garantiza una carga óptima y segura de la batería.

## 2. Respuesta al escalón de un sistema de control de control dinámico en lazo cerrado.

En un sistema de control dinámico en lazo cerrado, la respuesta al escalón es la salida del sistema cuando se le aplica una entrada escalón, es decir, una entrada que cambia abruptamente de un valor constante a otro valor constante. La respuesta al escalón es una herramienta importante para analizar el comportamiento dinámico del sistema y evaluar su rendimiento.

### Características de la respuesta al escalón:

**Tiempo de establecimiento:** Es el tiempo que tarda la salida en alcanzar el 95% de su valor final. Un tiempo de establecimiento corto indica una respuesta rápida del sistema.

**Tiempo de pico:** Es el tiempo que tarda la salida en alcanzar su valor máximo (si lo hay) después del cambio de entrada.

**Sobreimpulso:** Es la diferencia entre el valor máximo de la salida y su valor final en estado estable. Un sobreimpulso alto indica una respuesta oscilatoria del sistema.

**Tiempo de sedimentación:** Es el tiempo que tarda la salida en estabilizarse dentro de una banda de error especificada alrededor de su valor final. Un tiempo de sedimentación corto indica una respuesta estable del sistema.

### Factores que afectan la respuesta al escalón:

**Tipo de sistema:** El tipo de sistema, como de primer orden, segundo orden o de orden superior, determina la forma general de la respuesta al escalón.

**Parámetros del sistema:** Los parámetros del sistema, como la constante de tiempo, la ganancia y la amortiguación, afectan los valores específicos del tiempo de establecimiento, el tiempo de pico, el sobreimpulso y el tiempo de sedimentación.

**Controlador:** El tipo de controlador utilizado en el sistema de lazo cerrado también afecta la respuesta al escalón. Los controladores proporcionales, integrales y derivados (PID) son ejemplos comunes de controladores que pueden modificar la forma de la respuesta.

### Análisis de la respuesta al escalón:

El análisis de la respuesta al escalón se utiliza para evaluar el rendimiento del sistema de control en términos de su velocidad, precisión y estabilidad. Al comparar la respuesta al escalón del sistema con las especificaciones de diseño deseadas, se puede determinar si el sistema cumple con los requisitos de rendimiento.

### Aplicaciones de la respuesta al escalón:

**Diseño de controladores:** La respuesta al escalón se utiliza para diseñar controladores que optimicen el rendimiento del sistema de control.

**Sintonización de controladores:** La respuesta al escalón se utiliza para ajustar los parámetros de los controladores existentes para mejorar el rendimiento del sistema.

**Análisis de estabilidad:** La respuesta al escalón se utiliza para analizar la estabilidad del sistema de control y detectar posibles oscilaciones o inestabilidades.

Ejemplo en Python de cómo graficar la respuesta al escalón de un sistema de control dinámico en lazo cerrado utilizando Python y la biblioteca Matplotlib:

En este código, primero se define la función de transferencia del sistema en lazo cerrado. Luego, calcula la respuesta al escalón del sistema utilizando la función de transferencia y la grafica utilizando Matplotlib.

```
import numpy as np
import matplotlib.pyplot as plt

# Definición de la función de transferencia del sistema en lazo cerrado
def sistema_lazo_cerrado(Kp, Tau):
    def tf(s):
        return Kp / (Tau * s + 1)
    return tf

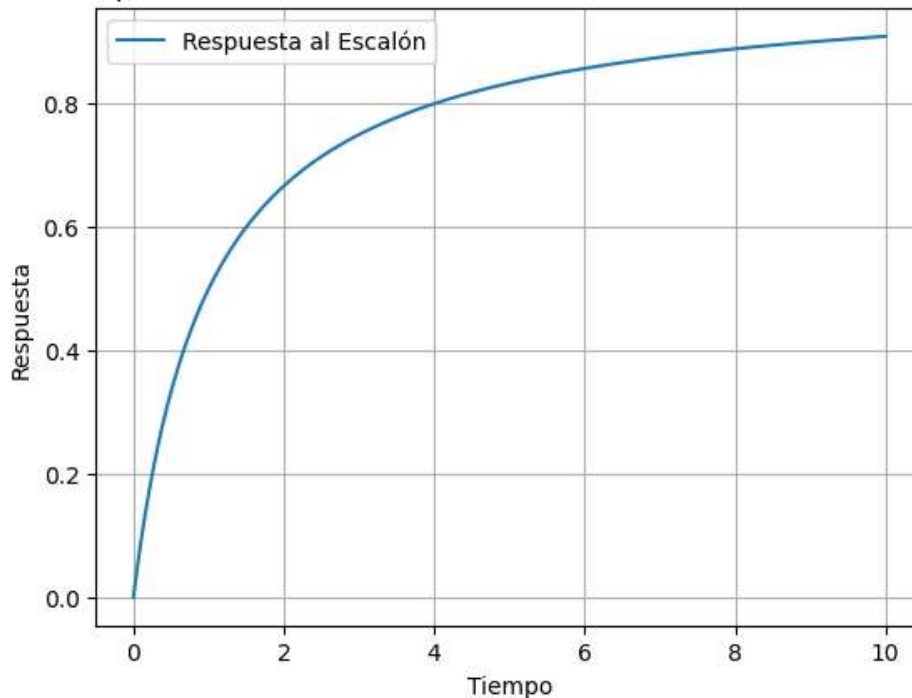
# Parámetros del sistema
Kp = 2.0 # Ganancia proporcional
Tau = 1.0 # Constante de tiempo

# Definición de la función de transferencia del sistema en lazo cerrado
G = sistema_lazo_cerrado(Kp, Tau)

# Respuesta al escalón del sistema en lazo cerrado
t = np.linspace(0, 10, 1000)
y = 1.0 - 1.0 / (1 + Tau * t) # Respuesta al escalón unitario

# Graficación de la respuesta al escalón
plt.plot(t, y, label='Respuesta al Escalón')
plt.title('Respuesta al Escalón de un Sistema de Control en Lazo Cerrado')
plt.xlabel('Tiempo')
plt.ylabel('Respuesta')
plt.grid(True)
plt.legend()
plt.show()
```

### Respuesta al Escalón de un Sistema de Control en Lazo Cerrado



### Análisis de la respuesta en función de los parámetros del sistema

En este caso, la ganancia proporcional ( $K_p$ ) y la constante de tiempo ( $\tau$ ), permite comprender cómo afectan estos parámetros al comportamiento dinámico del sistema en lazo cerrado.

- **Ganancia Proporcional ( $K_p$ ):** Aumentar la ganancia proporcional ( $K_p$ ) amplifica la respuesta del sistema. Esto significa que la salida del sistema responderá más rápidamente y con una mayor amplitud ante un escalón en la entrada. Una ganancia proporcional demasiado alta puede provocar oscilaciones excesivas o inestabilidad en el sistema.
- **Constante de Tiempo ( $\tau$ ):** La constante de tiempo ( $\tau$ ) determina la rapidez con la que el sistema responde al escalón de entrada. A medida que  $\tau$  aumenta, la respuesta del sistema se vuelve más lenta. Una constante de tiempo pequeña conlleva una rápida respuesta del sistema, mientras que una constante de tiempo grande resulta en una respuesta más suave y gradual.

Visualizar cómo varía la respuesta al escalón del sistema en lazo cerrado al modificar estos parámetros:



```

import numpy as np
import matplotlib.pyplot as plt

# Función de transferencia del sistema en lazo cerrado
def sistema_lazo_cerrado(Kp, Tau):
    def tf(s):
        return Kp / (Tau * s + 1)
    return tf

# Parámetros de análisis
Kp_values = [1.0, 2.0, 3.0] # Valores de ganancia proporcional
Tau_values = [1.0, 2.0, 3.0] # Valores de constante de tiempo
t = np.linspace(0, 10, 1000) # Tiempo

# Graficación de la respuesta al escalón para diferentes valores de Kp y Tau
plt.figure(figsize=(12, 8))

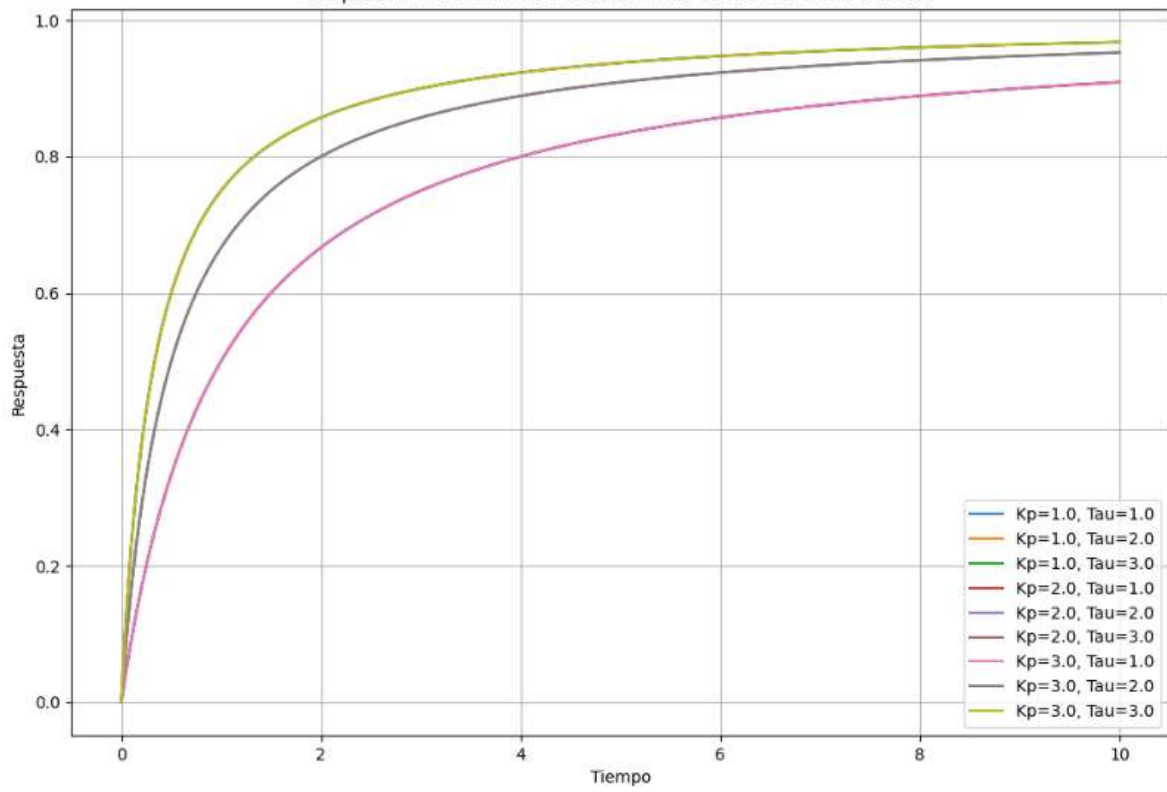
for Kp in Kp_values:
    for Tau in Tau_values:
        G = sistema_lazo_cerrado(Kp, Tau)
        y = 1.0 - 1.0 / (1 + Tau * t)
        plt.plot(t, y, label=f'Kp={Kp}, Tau={Tau}')

plt.title('Respuesta al Escalón de un Sistema de Control en Lazo Cerrado')
plt.xlabel('Tiempo')
plt.ylabel('Respuesta')
plt.grid(True)
plt.legend()
plt.show()

```

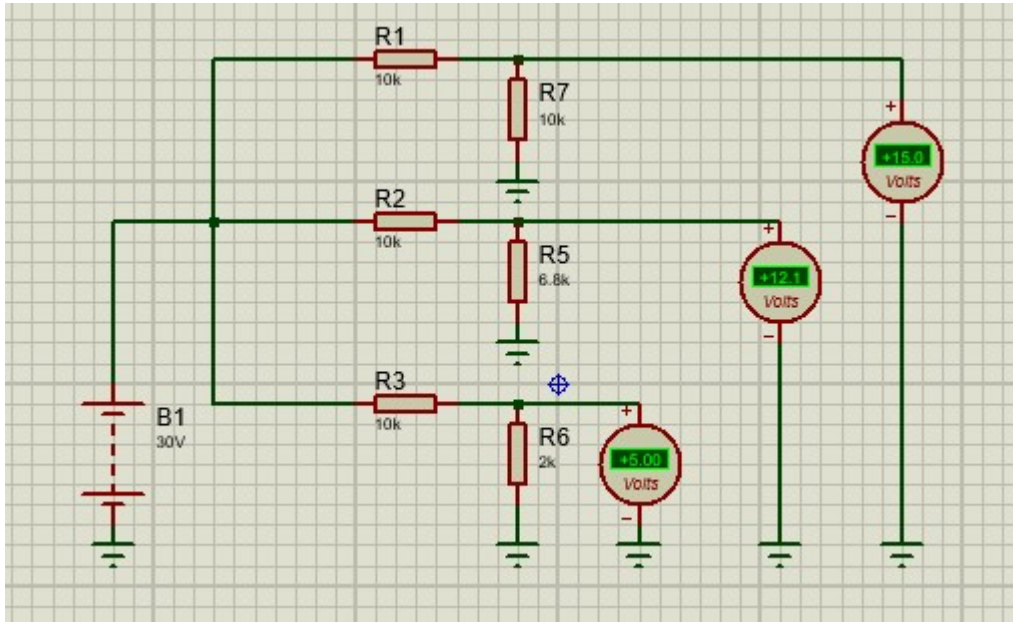
En el código anterior, se grafica la respuesta al escalón del sistema para diferentes combinaciones de valores de ganancia proporcional (Kp) y constante de tiempo (Tau). Se puede observar cómo cambia la forma de la respuesta al escalón a medida que se modifican estos parámetros.

Respuesta al Escalón de un Sistema de Control en Lazo Cerrado



### 3. Sistemas de control estático y dinámico

#### Circuito de control estático:



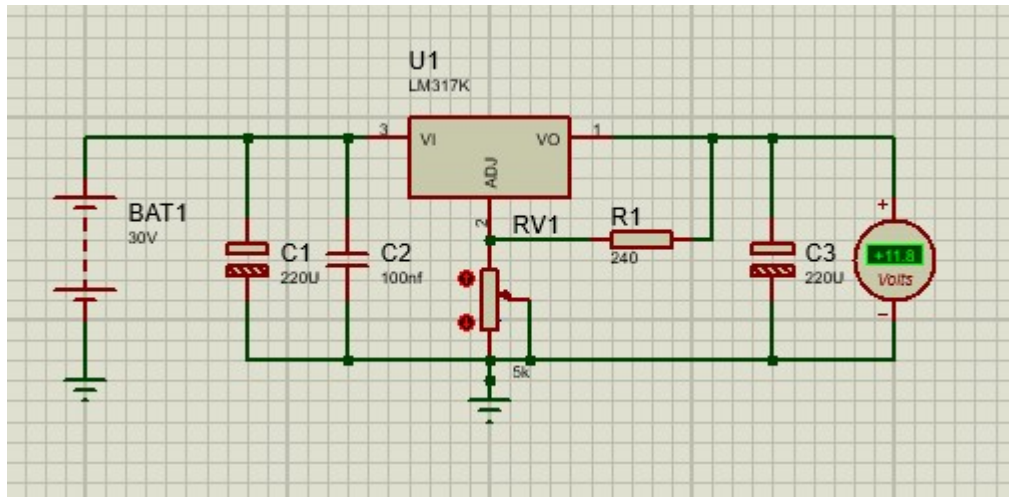
#### **Descripción del circuito:**

El circuito diseñado es un reductor de tensión que reduce una tensión de entrada de 30 voltios a tres tensiones de salida diferentes: 5 voltios, 12 voltios y 15 voltios. Para lograr esto, se utilizan tres divisores de tensión, cada uno compuesto por dos resistencias. Las resistencias están conectadas en serie entre la tensión de entrada de 30V y la tensión de salida deseada. La tensión de salida se toma de la conexión entre las resistencias.

Para obtener las tensiones de salida deseadas, se eligen valores específicos para una de las resistencias en cada divisor de tensión. Luego, se calcula el valor de la otra resistencia utilizando la fórmula del divisor de tensión. Una vez seleccionados los valores de las resistencias, el circuito puede reducir eficazmente la tensión de 30V a las tensiones de salida deseadas de 5V, 12V y 15V.

#### Circuito de control dinámico:

Este circuito utiliza el regulador de voltaje LM317 como una alternativa para regular la tensión de salida de una fuente de alimentación. A diferencia del circuito anterior, donde se utilizaban divisores de tensión con resistencias fijas para obtener tensiones específicas, este circuito permite ajustar la tensión de salida de manera continua mediante un potenciómetro.



Comparativo entre el sistema utilizando divisores de tensión con resistencias fijas y el sistema utilizando el regulador de voltaje LM317 con potenciómetro:

#### **Divisores de Tensión con Resistencias Fijas:**

##### **Ventajas:**

- **Simplicidad:** El circuito es simple y fácil de implementar, ya que solo requiere unas pocas resistencias.
- **Bajo Costo:** Los componentes necesarios son económicos y ampliamente disponibles.
- **Funcionamiento Pasivo:** No requiere componentes activos adicionales como reguladores de voltaje.

##### **Desventajas:**

- **Tensiones Fijas:** Las tensiones de salida son fijas y no se pueden ajustar sin cambiar las resistencias.
- **Precisión Limitada:** La precisión de las tensiones de salida depende de la precisión de las resistencias utilizadas.
- **Sensibilidad a la Carga:** Los divisores de tensión pueden ser sensibles a cambios en la carga y la corriente de salida.

#### **Regulador de Voltaje LM317 con Potenciómetro:**

##### **Ventajas:**

- **Tensión Ajustable:** Permite ajustar la tensión de salida de manera continua dentro de un rango específico utilizando un potenciómetro.
- **Precisión:** Proporciona una tensión de salida estable y precisa, independientemente de la carga.

- **Flexibilidad:** Puede adaptarse fácilmente a diferentes aplicaciones ajustando la tensión de salida según sea necesario.

**Desventajas:**

- **Mayor Complejidad:** Requiere más componentes y una configuración más elaborada que los divisores de tensión.
- **Costo Relativo:** Los reguladores de voltaje y los potenciómetros pueden ser más costosos que las resistencias simples.
- **Disipación de Calor:** El LM317 puede generar calor cuando disipa la potencia excedente, lo que puede requerir un disipador de calor en aplicaciones de alta potencia

#### 4. Análisis de la respuesta en frecuencia de un sistema lineal invariante en el tiempo.

Para analizar la respuesta en frecuencia de un sistema lineal invariante en el tiempo utilizando Python y la biblioteca SciPy, se puede utilizar la función `scipy.signal.freqz`. Esta función calcula la respuesta en frecuencia de un sistema discreto.

**Código de cómo graficar la respuesta en magnitud y fase:**

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

# Definir los coeficientes del sistema
b = [1.0] # Coeficientes del numerador
a = [1.0, -0.5] # Coeficientes del denominador

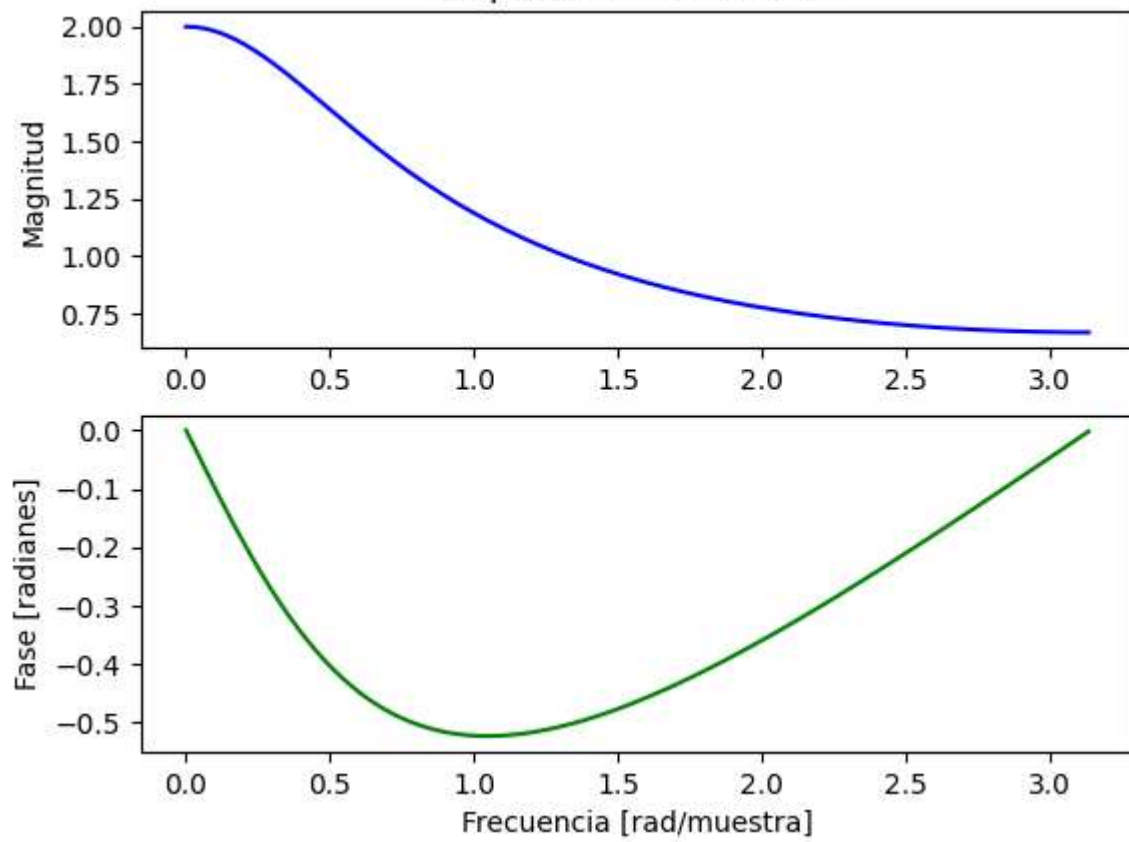
# Calcular la respuesta en frecuencia
w, h = signal.freqz(b, a)

# Graficar la respuesta en magnitud
plt.figure()
plt.subplot(2, 1, 1)
plt.plot(w, np.abs(h), 'b')
plt.title('Respuesta en Frecuencia')
plt.ylabel('Magnitud')

# Graficar la respuesta en fase
plt.subplot(2, 1, 2)
plt.plot(w, np.angle(h), 'g')
plt.xlabel('Frecuencia [rad/muestra]')
plt.ylabel('Fase [radianes]')
plt.show()
```

Aquí se define un sistema con coeficientes `b` y `a`, que representan los coeficientes del numerador y del denominador de la función de transferencia, respectivamente. Luego, se utiliza `scipy.signal.freqz` para calcular la respuesta en frecuencia del sistema. Finalmente, se grafica la respuesta en magnitud y fase utilizando Matplotlib. Se pueden ajustar los coeficientes del sistema `b` y `a` según las necesidades específicas para analizar la respuesta en frecuencia de cualquier sistema lineal invariante en el tiempo.

Respuesta en Frecuencia



## 5. Controlador proporcional, integral y derivativo (PID) para un sistema lineal invariante en el tiempo.

Conceptos clave para diseñar y analizar un controlador PID para un sistema lineal invariante en el tiempo:

**Sistemas Lineales Invariantes en el Tiempo (LTI):** Estos sistemas cumplen con las propiedades de linealidad e invarianza en el tiempo. En otras palabras, su comportamiento no cambia con el tiempo<sup>1</sup>. Por ejemplo, un circuito RC con valores constantes de resistencia y capacitancia es un sistema LTI.

### Controlador PID:

**El controlador PID combina tres acciones:** proporcional (P), integral (I) y derivativa (D) para controlar un sistema.

$$[ u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} ]$$

### Donde:

- $u(t)$  es la señal de control.
- $e(t)$  es el error entre la referencia y la salida actual.
- $(K_p)$ ,  $(K_i)$  y  $(K_d)$  son los coeficientes del controlador.
- 

En este código, se ha creado un sistema de segundo orden y diseñado un controlador PID. Se pueden ajustar los coeficientes  $(K_p)$ ,  $(K_i)$  y  $(K_d)$  según necesidades específicas.

```
import numpy as np
import matplotlib.pyplot as plt
import control as ctrl

# Sistema lineal invariante en el tiempo (por ejemplo, un sistema de segundo
orden)
numerator = [1] # Coeficientes del numerador
denominator = [1, 2, 1] # Coeficientes del denominador (s^2 + 2s + 1)

system = ctrl.TransferFunction(numerator, denominator)

# Diseñar un controlador PID
Kp = 1.0 # Coeficiente proporcional
Ki = 0.5 # Coeficiente integral
Kd = 0.1 # Coeficiente derivativo

controller = ctrl.TransferFunction([Kd, Kp, Ki], [1, 0]) # Controlador PID
```

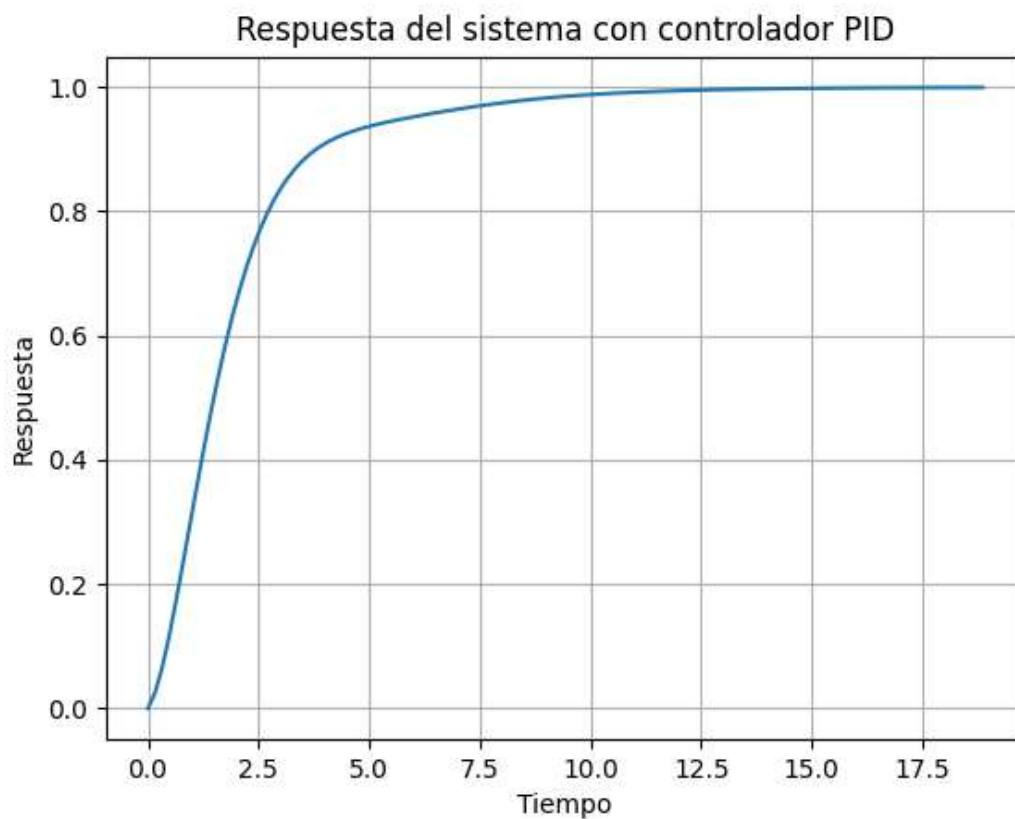


```
# Conectar el controlador al sistema
closed_loop_system = ctrl.feedback(system * controller)

# Generar una respuesta a una entrada escalón
time, response = ctrl.step_response(closed_loop_system)

# Graficar la respuesta
plt.plot(time, response)
plt.xlabel('Tiempo')
plt.ylabel('Respuesta')
plt.title('Respuesta del sistema con controlador PID')
plt.grid(True)
plt.show()
```

Figure 1



## **BIBLIOGRAFIA**

<https://chat.openai.com/>

<https://electronicavm.wordpress.com/2010/10/30/fuente-de-alimentacion-regulable-con-lm317/>

<https://pdf1.alldatasheet.com/datasheet-pdf/download/1147586/ARTSCHIP/LM317.html>

<https://colegiorobles.edu.ar/site/f/605db97e5121bdfc1c37b14a5b2aa4b7.pdf>

<https://www.youtube.com/playlist?list=PLf8NtEL4v5w-dCDmIRo6PcpjMBvcMAEvr>

<http://www0.unsl.edu.ar/~control1/apuntes/C6.2.pdf>