Proyecto 1: Diseño de PID mediante Lugar Geométrico de las Raíces para la planta Térmica

Departamento de Ingeniería Electrónica
Universidad de Antioquia

6 de junio de 2025

Introducción

En esta actividad usaremos la planta TempLABUdeA, cuyo modelo linealizado para cada zona térmica es:

$$G(s) \;=\; \frac{K}{\tau\, s+1} e^{\theta_d s},$$

con K y τ obtenidos de la linealización. El objetivo es diseñar un controlador PID que cumpla especificaciones de sobreimpulso, tiempo de asentamiento y margen de fase utilizando el Lugar Geométrico de las Raíces (LGR).

Objetivos de la Actividad

1. Aplicar el Lugar Geométrico de las Raíces para ubicar polos del lazo cerrado en posiciones deseadas.
2. Determinar los parámetros K_p, T_i y T_d de un PID que satisfaga especificaciones de desempeño.
3. Verificar mediante simulación en Python o MATLAB la respuesta al escalón y la estabilidad del sistema controlado.

Procedimiento

1. Función de transferencia en lazo abierto

Partiendo del controlador PID:

$$G_{\text{PID}}(s) = K_p\Bigl(1 + \frac{1}{T_i\, s} + T_d\, s\Bigr),$$

la función de transferencia en lazo abierto es:

$$L(s) \;=\; G_{\text{PID}}(s)\; G(s) \;=\; K_p\, \frac{K\,(T_i\, T_d\, s^2 + T_i\, s + 1)}{T_i\, s\,(\tau\, s + 1)}.$$

- Defina K_p, T_i y T_d como símbolos iniciales y exprese $L(s)$.
- Verifique que el numerador de $L(s)$ sea $K_p\,K\,(T_i\,T_d\,s^2 + T_i\,s + 1)$.
- Verifique que el denominador de $L(s)$ sea $T_i\,s\,(\tau\,s + 1)$.

2. Trazado del Lugar Geométrico de las Raíces

1. En Python (módulo `control` y `matplotlib`) o MATLAB, defina valores numéricos para K y τ de la zona térmica seleccionada.
2. Grafique el LGR de
$$L(s) \;=\; K_p\,\frac{K\,(T_i\,T_d\,s^2 + T_i\,s + 1)}{T_i\,s\,(\tau\,s + 1)}$$
variando K_p. Use valores provisionales para T_i y T_d, por ejemplo $T_i = \tau$ y $T_d = \tau/10$.
3. Observe el lugar trazado en el plano s. Identifique el recorrido de los polos al aumentar K_p.

3. Selección de polos deseados

- Defina las especificaciones de desempeño:
  - Porcentaje de sobreimpulso $10\,\%OS$.
  - Tiempo de asentamiento T_s (2) minutos.
- Relacione $\%OS$ con el coeficiente de amortiguamiento ζ mediante:
$$\%OS \approx 100\,e^{-\frac{\pi\,\zeta}{\sqrt{1-\zeta^2}}}.$$
- Relacione T_s con la parte real del polo dominante:
$$T_s \approx \frac{4}{\zeta\,\omega_n}, \quad \omega_n = \text{frecuencia natural.}$$
- Calcule ζ y ω_n a partir de $\%OS$ y T_s.
- Determine la ubicación deseada de los polos dominantes:
$$s_{1,2} = -\zeta\,\omega_n \pm j\,\omega_n\sqrt{1-\zeta^2}.$$

4. Cálculo de K_p, T_i, T_d

1. En el punto deseado s_d (uno de los polos dominantes), imponga la condición de fase:
$$\angle L(s_d) = -180^\circ.$$
Esto equivale a:
$$\angle\big(T_i\,T_d\,s_d^2 + T_i\,s_d + 1\big) \;-\; \angle s_d \;-\; \angle(\tau\,s_d + 1) = -180^\circ - \angle K_p - \angle K.$$

2. Imponga la condición de módulo en s_d:

$$\left|L(s_d)\right| = 1 \quad \Longleftrightarrow \quad K_p\,K\,\frac{\left|T_i\,T_d\,s_d^2 + T_i\,s_d + 1\right|}{T_i\,|s_d|\,|\tau\,s_d + 1|} = 1.$$

3. Resuelva numéricamente el par de ecuaciones (fase y módulo) para T_i y T_d, una vez fijado K_p si así lo desea, o bien resuelva simultáneamente para K_p, T_i, T_d.

4. Ajuste afín de K_p si la condición $\left|L(s_d)\right| = 1$ no se cumple exactamente.

5. Simulación y verificación

- Con los valores obtenidos de K_p, T_i, T_d, simule la respuesta al escalón de la función de transferencia en lazo cerrado:

$$G_{\mathrm{cl}}(s) = \frac{L(s)}{1 + L(s)}.$$

- Compare la %OS, T_s y margen de fase resultantes con las especificaciones.
- Ajuste parámetros si es necesario y vuelva a simular hasta cumplir las metas.

6. Implementación en TempLABUdeA

- En el script Python de control, reemplace la ley PI por la ley PID:

$$Q(t) \;=\; Q_{\mathrm{bias}} + K_p\,e(t) + \frac{K_p}{T_i}\int_0^t e(\tau)\,d\tau + K_p\,T_d\,\frac{de}{dt}.$$

- Aproxime la derivada discretamente:

$$\frac{de}{dt} \approx \frac{e(k) - e(k-1)}{\Delta t}.$$

- Implemente antiwindup restando el error cuando $Q(t)$ se sature:

$$\text{si } Q = 100,\ ierr \mathrel{-}= err; \quad \text{si } Q = 0,\ ierr \mathrel{-}= err.$$

- Ejecute la prueba en TempLABUdeA, registre $T_1(t)$, $Q_1(t)$, compárelos con la simulación.

Entregables y Evaluación

- Documento en LaTeX detallando:
  - Trazado del LGR y selección de polos dominantes.
  - Cálculo paso a paso de K_p, T_i, T_d.
  - Gráficas de LGR y respuesta al escalón simulada.
- Código Python/MATLAB para:

- Generar el Lugar Geométrico del lazo abierto.
- Simular respuesta al escalón de $G_{\mathrm{cl}}(s)$.
- Implementar PID en TempLABUdeA y recolectar datos.

- Archivos de datos experimentales (*.csv) y gráficos que comparen experimento vs. simulación.

Rúbrica de Evaluación (Adaptada)

Criterio	**0**	**1**	**2**	**3**	**4**	**5**
Trazado correcto LGR	Ausente	Incorrecto	Parcial	Funcional	Bueno	Excelente
Ubicación de polos deseados	N/D	Errónea	Incompleta	Aceptable	Buena	Óptima
Cálculo de K_p, T_i, T_d	Ausente	Inexacto	Parcial	Funcional	Bien justificado	Excelente
Comparación experimento vs. simulación	N/D	Muy disperso	>25 %	15–25 %	10–15 %	<10 %
Calidad de reporte NOTEBOOK y código	Ausente	Muy deficiente	Deficiente	Aceptable	Buena	Excelente

Cuadro 1: Rúbrica de evaluación del diseño PID por LGR (0–5).