MANUAL DE ESTUDIO · SISTEMAS Y CONTROL EN MECATRÓNICA

Control inteligente, eje electrohidráulico y optimización

Capítulo 06 · En el video: **minuto 52:42** · Duración 10:17

Objetivo. Aplicar técnicas inteligentes y de optimización al control y al diseño de un eje electrohidráulico.

Ideas clave

Lógica difusa

Variables lingüísticas con funciones de pertenencia, reglas SI–ENTONCES e inferencia Mamdani o Takagi-Sugeno. Un PD difuso tiene una superficie de control no lineal; la auto-sintonía difusa ajusta Kp, Ki y Kd según el punto de operación.

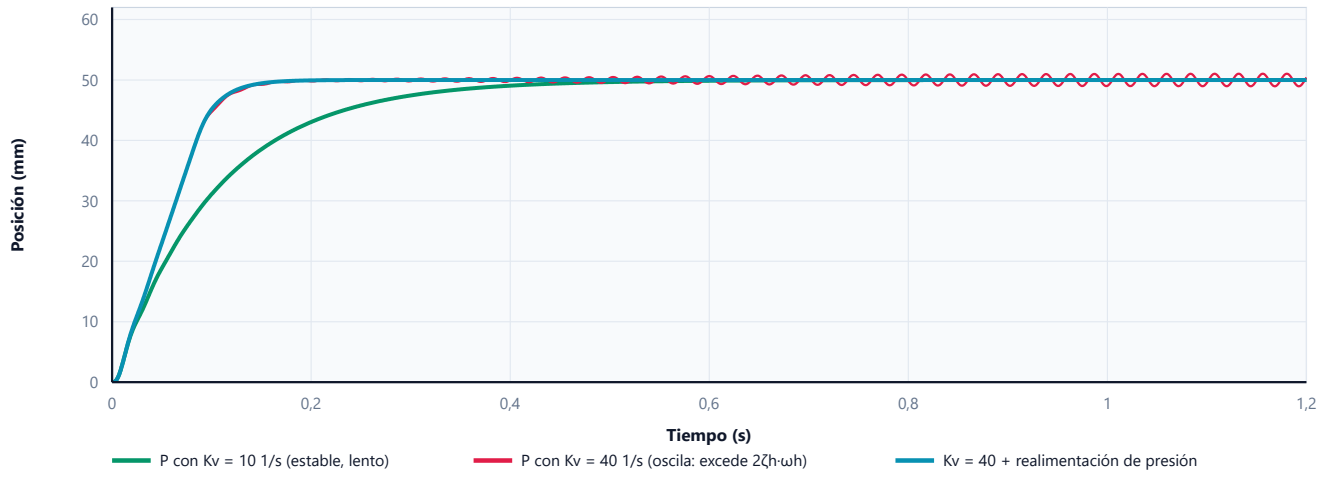
Redes neuronales y ANFIS

Un perceptrón multicapa o una red RBF aproxima cualquier función continua: sirven para identificar plantas (modelos NARX), control por modelo inverso y compensación en línea. ANFIS une la interpretabilidad difusa con el aprendizaje: cinco capas equivalentes a un Takagi-Sugeno.

Eje electrohidráulico

Servoválvula (caudal por orificio, raíz de la diferencia de presión), cilindro y continuidad con compresibilidad del aceite (β_e). La rigidez del aceite y la masa forman una resonancia $\omega_h = \sqrt{(4\beta_e A^2/(V \cdot m))}$ con muy poco amortiguamiento ($\zeta_h \approx 0{,}05–0{,}15$).

Eje electrohidráulico: escalón de 50 mm (ωh = 41 Hz, ζh = 0,055) cursosmecatronica.com



La resonancia hidráulica casi no tiene amortiguamiento: sin ayuda, el lazo de posición debe ser lento. Restar la presión de carga agrega amortiguamiento activo.

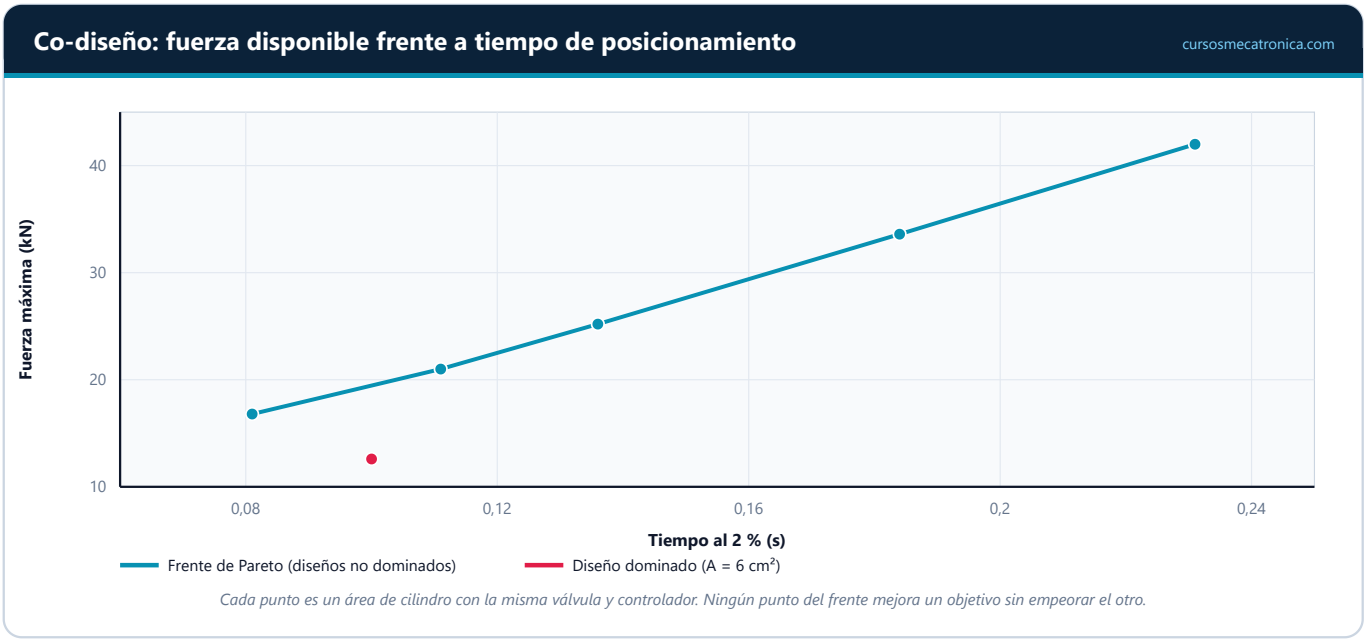
Respuestas simuladas con eje_electrohidraulico.py.

Control del eje

Con un P de posición la ganancia de velocidad debe ser menor que $2\zeta_h\omega_h$. Restar la presión de carga agrega amortiguamiento activo; además se compensa la zona muerta del carrete y se pueden usar backstepping, SMC o control adaptativo.

Optimización y co-diseño

Se buscan las variables de diseño p (mecánicas, eléctricas y del control) que minimizan objetivos J(p) con restricciones g(p) ≤ 0. Con varios objetivos en conflicto (masa, energía, ancho de banda) se obtiene un frente de Pareto. Métodos: SQP, algoritmos genéticos (NSGA-II) y enjambre de partículas (PSO).



Optimización multiobjetivo del eje.

Fórmulas y relaciones

Relación	Expresión	Cómo leerla
Caudal de la servoválvula	$Q_L = K_q \cdot x_v \cdot \sqrt{((P_s - \mathrm{sgn}(x_v) P_L)/P_s)}$	No lineal.
Continuidad	$(V/4\beta_e) \cdot \dot{P}_L = Q_L - A \cdot v - C_{tl} \cdot P_L$	Compresibilidad y fugas.
Frecuencia hidráulica	$\omega_h = \sqrt{(4\beta_e A^2/(V \cdot m))}$	41 Hz en el ejemplo.
Límite del lazo P	$K_v = K_p K_q / A < 2\zeta_h \omega_h$	Estabilidad sin amortiguamiento activo.
Takagi-Sugeno	$u = \Sigma\, w_i \cdot u_i \,/\, \Sigma\, w_i$	Promedio ponderado de las reglas.
Dominancia de Pareto	a domina a b si es mejor o igual en todo y mejor en algo	Define el frente.

Ejemplo resuelto: límite de ganancia de un eje hidráulico

1. A = 12 cm², m = 200 kg, V = 6·10⁻⁴ m³, β_e = 1,4 GPa: $\omega_h = 259$ rad/s (41 Hz), $\zeta_h \approx 0{,}055$.
2. Límite: $K_v < 2 \cdot 0{,}055 \cdot 259 = 28{,}7$ 1/s.
3. K_v = 10: estable, se asienta en 0,39 s. K_v = 40: oscila y no se asienta.
4. K_v = 40 restando la presión de carga: estable y se asienta en 0,14 s.

eje_electrohidraulico.py simula el modelo no lineal, compara los controladores y obtiene el frente de Pareto fuerza–rapidez; eje_hidraulico_plc.st muestra la implementación en un PLC con supervisión.

Técnicas inteligentes

Técnica	Fortaleza	Limitación
Lógica difusa	Usa el conocimiento del operador; no lineal y legible	Ajuste manual de reglas
Red neuronal (MLP, RBF)	Aprende de datos; aproxima cualquier función	Caja negra; necesita datos
ANFIS	Aprende y sigue siendo interpretable	Crece rápido con el número de entradas
Algoritmo genético / PSO	Optimiza sin derivadas, problemas no convexos	Muchas evaluaciones del modelo

Código del capítulo

Los archivos completos vienen en el paquete ZIP del capítulo (carpeta `codigo/`).

eje_electrohidraulico.py — modelo no lineal de servoválvula y cilindro, amortiguamiento por presión y frente de Pareto. *(extracto: primeras 60 de 77 líneas)*

```
"""
eje_electrohidraulico.py
Eje electrohidráulico con servoválvula y cilindro (capítulo 06).

Modelo no lineal:
    caudal:      QL = Kq·xv·raíz((Ps − sign(xv)·PL)/Ps)
    presión:     (V/(4·beta))·PL' = QL − A·v − Ctl·PL
    movimiento:  m·v' = A·PL − b·v
Se calcula la frecuencia natural hidráulica wh = raíz(4·beta·A²/(V·m)) y su
amortiguamiento, y se comparan dos controladores de posición:
  a) proporcional en posición: con zeta_h ≈ 0,1 la ganancia debe ser baja;
  b) proporcional + realimentación de presión de carga (amortiguamiento activo).
Al final se hace una optimización multiobjetivo sencilla: barrido del área del
cilindro y búsqueda del frente de Pareto (tiempo de respuesta frente a caudal).

Uso:  python eje_electrohidraulico.py
"""
import math

PS, BETA = 210e5, 1.4e9          # Pa
A, M, B = 1.2e-3, 200.0, 2000.0  # m², kg, N·s/m
V = 2 * A * 0.25                 # volumen total para una carrera de 0,5 m
KQ, CTL = 0.6, 2e-12              # (m³/s) por metro de carrete (≈ 36 L/min con 1 mm), fuga (m³/s)/Pa
```

```
def simular(kp, kpl=0.0, a=A, t_fin=1.0, dt=2e-5, ref=0.05):
    x = v = pl = 0.0
    t, osc_max, t98 = 0.0, 0.0, None
    vol = 2 * a * 0.25
    q_max = 0.0
    while t < t_fin:
        xv = kp * (ref - x) - kpl * pl
        xv = max(-1e-3, min(1e-3, xv))           # carrera máxima del carrete
        signo = 1 if xv >= 0 else -1
        ql = KQ * xv * math.sqrt(max(PS - signo * pl, 0) / PS)
        dpl = (4 * BETA / vol) * (ql - a * v - CTL * pl)
        dv = (a * pl - B * v) / M
        pl += dpl * dt
        v += dv * dt
        x += v * dt
        t += dt
        q_max = max(q_max, abs(ql))
        osc_max = max(osc_max, x - ref)
        if t98 is None and abs(x - ref) < 0.02 * ref and t > 0.01:
            t98 = t
        elif t98 is not None and abs(x - ref) >= 0.02 * ref:
            t98 = None
    return osc_max / ref * 100, t98, q_max


if __name__ == '__main__':
    wh = math.sqrt(4 * BETA * A * A / (V * M))
    zh = B / (4 * A) * math.sqrt(V / (BETA * M)) + CTL / A * math.sqrt(BETA * M / V)
    print(f'Frecuencia natural hidráulica wh = {wh:.1f} rad/s ({wh / (2 * math.pi):.1f} Hz), zeta_h ≈
{zh:.3f}')
    print(f'Ganancia de velocidad Kv = Kp·Kq/A; límite de estabilidad sin presión: Kv < 2·zeta_h·wh = {2 * zh
* wh:.1f} 1/s')
    for kp in (0.02, 0.08):
        print(f'   Kp = {kp}: Kv = {kp * KQ / A:.1f} 1/s')
    print('\n   controlador                    sobreimpulso   t al 2 %')
    casos = [('P, Kp = 0,02', 0.02, 0.0), ('P, Kp = 0,08', 0.08, 0.0),
             ('P + presión, Kp = 0,08', 0.08, 3e-11)]
```

eje_hidraulico_plc.st — bloque de función con rampa, realimentación de presión, zona muerta y supervisión.

difuso_y_optimizacion.py — PD difuso Takagi-Sugeno, auto-sintonía y optimización por enjambre de partículas.

Errores frecuentes

- Subir la ganancia de un eje hidráulico sin considerar ω_h.
- Dar escalones de referencia que excitan la resonancia: usar rampas.
- Entrenar una red con datos que no cubren todo el rango de operación.
- Optimizar un solo objetivo y descubrir después que otro quedó inaceptable.

Autoevaluación

1. ¿Qué limita la ganancia de un eje hidráulico?
2. ¿Qué aporta realimentar la presión de carga?
3. ¿Qué tipo de inferencia usa un promedio ponderado de consecuentes?
4. ¿Qué es un frente de Pareto?

Respuestas. 1. La resonancia hidráulica poco amortiguada. **2.** Amortiguamiento activo. **3.** Takagi-Sugeno. **4.** El conjunto de diseños que no pueden mejorar un objetivo sin empeorar otro.