

Estimación Parámetros de un controlador PID

El grupo debe realizar un notebook donde plantee el problema (función objetivo) y la técnica de optimización Heurística a partir del enunciado dado y encuentre la solución optima.

Descripción

Se requiere estimar los parámetros de un sistema de control PID. La función objetivo consistirá en evaluar el desempeño de un conjunto de parámetros proporcional (K_p), integral (K_i) y derivativo (K_d), en el control de un sistema basado en resistores. Las ecuaciones diferenciales que describen el sistema se encuentran detalladas en la pagina <https://apmonitor.com/pdc/index.php/Main/ArduinoModeling2>. Estas ecuaciones diferenciales nos sirven para realizar las simulaciones del sistema, y a la vez poder evaluar el desempeño del controlador. El controlador se puede definir como se muestra en la figura 1.

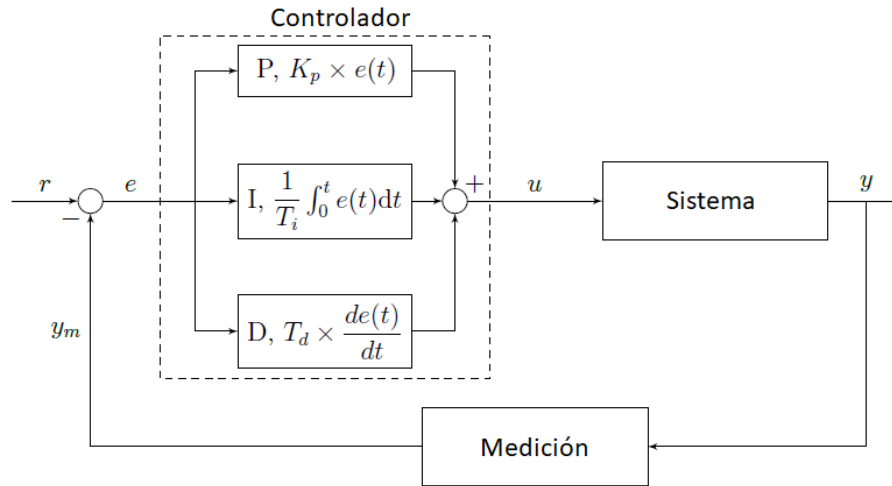


Figura 1: Esquema general de un controlador PID.

Matemáticamente, la señal de control se calcula

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

donde $e(t)$ es el error el cual significa la diferencia entre la referencia $r(t)$ y la salida $y(t)$. En la literatura existe diferentes esquemas para evaluar el desempeño de los controladores y así poder sintonizar o seleccionar las constantes del controlador PID. Se adopta como función objetivo una medida de desempeño para controladores conocida como la integral del error absoluto (IAE), la cual se define

$$IAE = \sum_{n=0}^N |e[n]|.$$

El IAE depende directamente de los parámetros del PID empleados.

Para el algoritmo heurístico se debe implementar en una clase de Python el *Whale Optimization Algorithm* (WOA).

Procedimiento

1. Descripción matemática y conceptual del problema de optimización y del sistema dinámico.
2. Definir y programar la función objetivo de minimización del error del controlador por medio del IAE (Integral Absolute Error) para el sistema suministrado.
3. Descripción matemática del algoritmo Heurístico asignado, como hace exploración y explotación. Recordar que las constantes del controlador son todas positivas. El algoritmo se debe diseñar con una clase de Python, con inicialización de los parámetros del algoritmo (límites, cantidad de iteraciones); con método que se encargue del proceso de optimización, y adicionalmente atributos que almacenen el desempeño de algoritmo (mejor solución por iteración).
4. Realizar el proceso de optimización 10 veces para 50 iteraciones y almacenar los resultados en una matriz.
5. Analizar la convergencia del algoritmo graficando las 10 curvas de la función objetivo vs el número de iteraciones. Graficar adicionalmente la media de las 10 curvas.
6. Comparar el desempeño del algoritmo contra el algoritmo de Enjambre de Partículas visto en clase.

1. Informe

Desarrollar un notebook en Python, que incluya las siguientes secciones:

1. Introducción al problema.
2. Código y desarrollo de la solución.
3. Análisis de resultados.
4. Conclusiones.
5. Bibliografía.

Referencias

APMonitor site. https://apmonitor.com/do/uploads/Main/Lab_B_MIMO_Model.pdf.

EERO HEINÄNEN, A Method for Automatic Tuning of PID Controller following Luus-Jaakola Optimization. Master thesis, 2018.

Whale Optimization Algorithm - blog. <https://www.baeldung.com/cs/whale-optimization-algorithm>