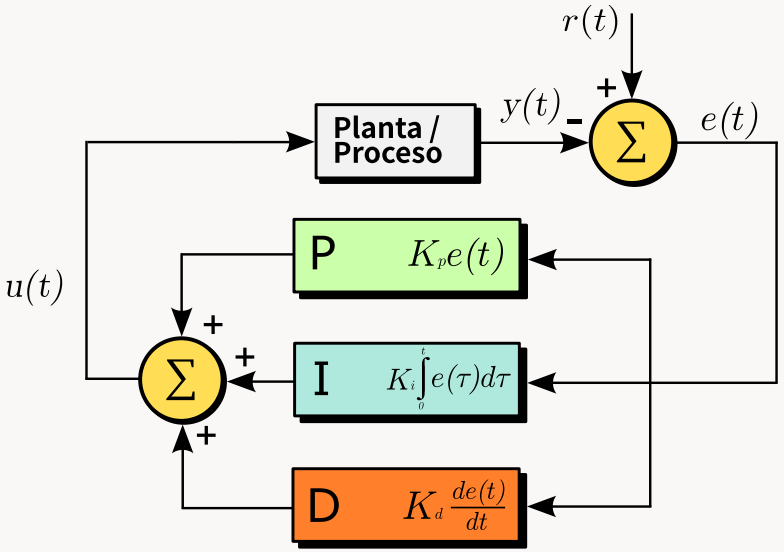


Control clásico

Robótica

Alberto Díaz y Raúl Lara

Curso 2022/2023

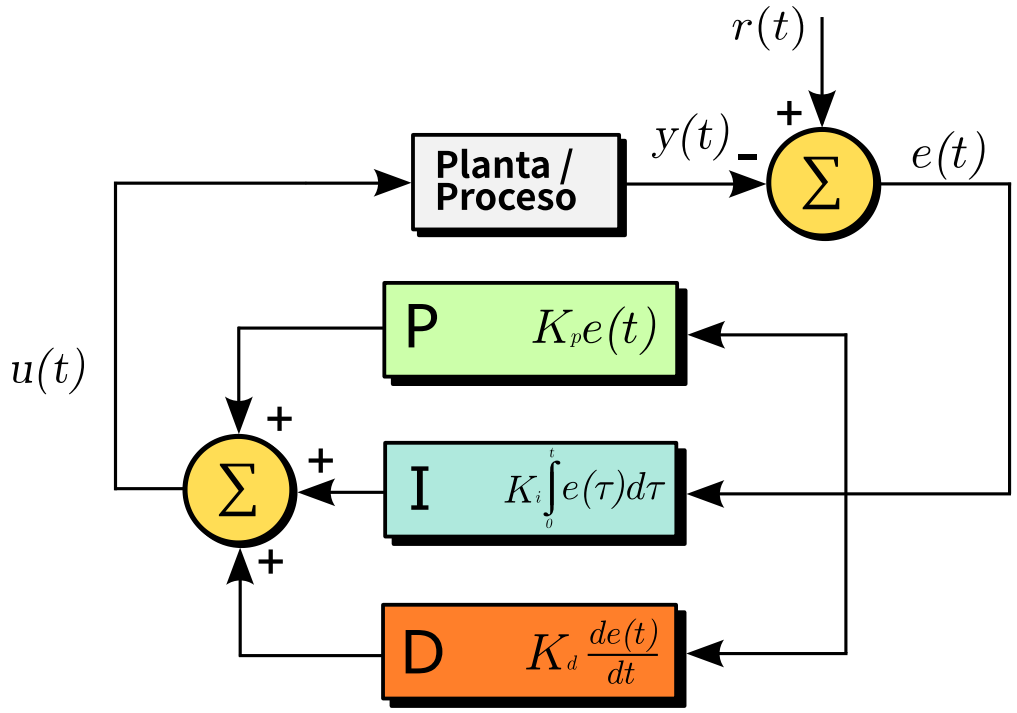
Departamento de Sistemas Informáticos

Control clásico con realimentación (principios 1920)

Se basa en la realimentación de la señal de salida al sistema de control.

Modifica la señal de entrada para que la señal de salida se aproxime a la señal deseada, basándose en un modelo matemático del sistema.

Las componentes P (Proportional), I (Integral) y D (Derivative) se pueden combinar para obtener un control más eficiente.



La señal de salida tiene que ser reescalada en función del actuador que se esté utilizando.

Control proporcional P

En el algoritmo de control proporcional, la salida del controlador es proporcional a la señal de error, que es la diferencia entre el punto objetivo que se desea y la variable de proceso:

$$u(t) = K_p e(t)$$

donde K_p es la ganancia proporcional y $e(t)$ es la señal de error.

El control proporcional es el más simple de los controladores, pero también el más inestable. El control proporcional es adecuado para sistemas que tienen un error pequeño y que no cambian rápidamente.

Control integral I

En el algoritmo de control integral, la salida del controlador es proporcional a la integral de la señal de error, que es la suma de las diferencias entre el punto objetivo que se desea y la variable de proceso:

$$u(t) = K_i \int_0^t e(\tau)d\tau$$

donde K_i es la ganancia integral y $e(t)$ es la señal de error.

Aumenta la acción en relación no sólo con el error sino también con el **tiempo** durante el cual ha persistido. Así, si la fuerza aplicada no es suficiente para llevar el error a cero, esta fuerza se incrementará a medida que pase el tiempo.

Control derivativo D

En el algoritmo de control derivativo, la salida del controlador es proporcional a la derivada de la señal de error, que es la velocidad de cambio de la señal de error:

$$u(t) = K_d \frac{de(t)}{dt}$$

donde K_d es la ganancia derivativa y $e(t)$ es la señal de error.

Aumenta la acción en relación no sólo con el error sino también con la **velocidad** a la que cambia el error. Así, si la fuerza aplicada no es suficiente para llevar el error a cero, esta fuerza se incrementará a medida que el error cambie de signo.

Control proporcional-integral-derivativo PID

El control PID es un controlador que combina las tres componentes básicas de control: proporcional, integral y derivativa. El control PID es el más utilizado en la industria.

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

donde K_p, K_i y K_d son las ganancias proporcional, integral y derivativa, respectivamente.

Hay que optimizar los parámetros K_p, K_i y K_d para obtener un control eficiente.

Simulador

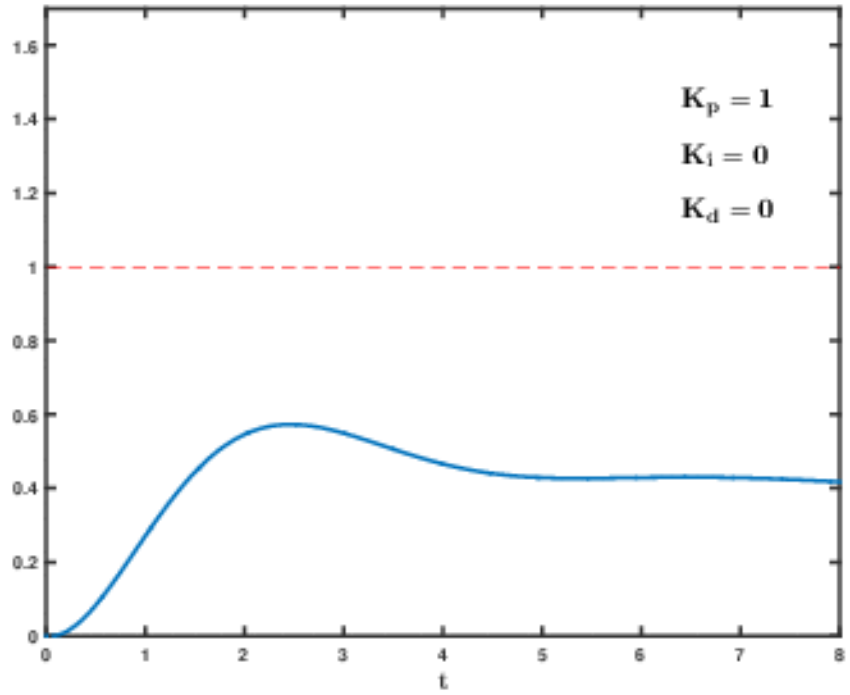
Pseudocódigo del control PID

```
def PID(Kp, Ki, Kd, dt):
    integral = 0
    last_error = 0
    while True:
        error = setpoint - actual_position
        integral += error * dt
        derivative = (error - last_error) / dt
        output = Kp * error + Ki * integral + Kd * derivative
        last_error = error
        yield output
```

`yield` devuelve el valor de la salida del controlador y se suspende hasta la siguiente iteración. Así se define un generador que se puede utilizar en un bucle `for`.

Optimización manual de los parámetros

1. Empezamos con los parámetros a cero.
2. Aumentar K_p hasta que la salida del bucle oscile; entonces, fijar K_p a aproximadamente la mitad de ese valor.
3. Aumentar K_i hasta que cualquier desviación se corrija en tiempo suficiente para el proceso sin causar inestabilidad.
4. Aumentar K_d, si es necesario, hasta que el bucle sea aceptablemente rápido para alcanzar su referencia después de una perturbación.



Optimización Ziegler-Nichols

1. Empezamos con los parámetros a cero.
2. Aumentar K_p hasta obtener una oscilación estable en la salida. Este será el valor K_u y el período de la oscilación será T_u.

   3 Asignar valores a los parámetros K_p, K_i y K_d según la siguiente tabla:

Tipo	K_p	K_i	K_d
PID	$0.6K_u$	$1.2K_u/T_u$	$3K_uT_u/40$
PI	$0.45K_u$	$0.54K_u/T_u$	
P	$0.5K_u$		

Control en cascada

El control en cascada es un método de control en el que se utilizan dos o más controladores para controlar un proceso.

La salida del controlador externo se utiliza como entrada del controlador interno, y representa el punto de referencia del controlador interno.

Se utiliza para controlar procesos que tienen un tiempo de respuesta muy lento, como la temperatura de un horno, una habitación, etc.

Figure 2 – Cascade control system

> Fuente de la imagen: Watlow

Ventajas y desventajas

Ventajas:

1. Fácil de implementar (sólo una simple ecuación)
2. Utiliza pocos recursos
3. Resistente a los desajustes de optimización
4. Fácil de optimizar
5. Buena respuesta a las perturbaciones

Desventajas:

1. Bajo rendimiento en procesos con largos tiempos de espera
2. Bajo rendimiento para tratar fuertes no linealidades
3. Dificultad para manejar múltiples variables con fuerte interrelación
4. Dificultad para manejar múltiples restricciones

¡GRACIAS!