

Diseño de un controlador PID digital

1. Resumen

Este proyecto consiste en el diseño, implementación, sintonización y prueba de un controlador PID en su versión digital en tiempo real programado en un Arduino.

2. Notas importantes

1. Se debe explicar a la profesora todos los procedimientos utilizados para la solución del problema, incluir todos sus resultados, con sus respectivos análisis, y las conclusiones.
2. Se debe justificar la elección de todos los parámetros utilizados para resolver el problema.
3. Se debe entregar vía Mediación virtual el archivo con el código, si realizaron simulaciones todos aquellos archivos que justifiquen los resultados obtenidos y la presentación (no hay trabajo escrito). Todos los archivos se deben incluir dentro de una sola carpeta.
4. La fecha de entrega es el jueves 22 de setiembre de 2022 a las 12:00 a.m., fecha en la que se realizará la presentación del diseño en clase. Si no se presenta el trabajo, perderá todos los puntos del proyecto.
5. No se aceptan trabajos posteriores a esta fecha.
6. Para el desarrollo de este trabajo puede utilizar MATLAB, Python, Mathematica, y el software y hardware de Arduino®, en todos aquellos puntos que sea necesario.

3. Problema

El sistema que se desea controlar se presenta en la Figura 1. Los parámetros del circuito son $R = 47 \text{ k}$ y $C = 470 \text{ nF}$. El modelo en variables de estado en tiempo continuo vendría dado por:

$$\begin{aligned}\frac{dv_1(t)}{dt} &= \frac{-2}{RC}v_1(t) + \frac{1}{RC}v_2(t) + \frac{1}{RC}v_{in}(t), \\ \frac{dv_2(t)}{dt} &= \frac{1}{RC}v_1(t) - \frac{1}{RC}v_2(t),\end{aligned}$$

y la salida del sistema sería igual a $v_2(t)$ (es decir, la variable controlada es la tensión del capacitor 2).

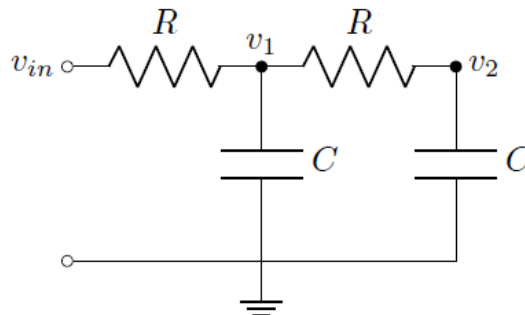


Figura 1: Sistema por controlar.

Los requisitos mínimos del proyecto son los siguientes:

1. Con respecto al proceso
 - a) Deben implementar el circuito en la protoboard (5%).
 - b) Estudiar el circuito dado y su respuesta es lazo abierto. Pueden usar el software de simulación Matlab, Python, Mathematica. Indique el tipo de sistema si es amortiguado, subamortiguado o críticamente amortiguado (5%).

2. Con respecto a la implementación del controlador

- a) El controlador debe implementar un algoritmo PID digital, debe elegir un tiempo de muestreo en relación con las constantes de tiempo del proceso (5 %). Ver anexo para la ecuación del controlador.

b) El controlador (Arduino) recibe su *setpoint* (referencia) a través de una entrada analógica (AI). Considere en el código que esta señal se llama SP (5 %).

c) La variable controlada $v_2(t)$ entra a un pin analógico (AI) del Arduino. Este tiene un convertidor Analógico/Digital CAD de 10 bits (0-1023). Investigue cuál pin puede usar. (5 %).

d) La señal de control debe salir del pin digital del Arduino que sea PWM (la función de un filtro paso bajos para obtener el valor DC lo hacemos con el proceso). El Arduino tiene un CDA de 8 bits (0-255). Investigue cuál pin puede usar. (5 %).

e) El controlador debe ser capaz de pasar del modo manual a automático. Suponga una señal denominada M, conectada a una entrada digital, le indicará al controlador cuando pasará a modo manual (si la entrada digital tiene un valor lógico de 1 (5V) se debe pasar a modo manual). En el modo manual, el usuario es capaz de decidir el valor de salida del controlador, independientemente del error. (5 %).

El valor de la salida del controlador en modo manual estará indicado por una señal analógica conectada a otra entrada analógica del controlador. Llame a esta señal X y note que esta señal **es diferente** a la señal de setpoint. En el modo automático, el controlador decide el valor de salida del controlador mediante el algoritmo PID independientemente del valor que tenga la señal X. (5 %).

Se le debe indicar al usuario que se pasó del modo manual al modo automático mediante un LED (*built-in* de Arduino) (se debe encender en modo automático) (5 %)

f) Se implementa una etapa de saturación para limitar la señal de salida del controlador (5 %)

3. Con respecto a la sintonización

a) Es válido sintonizar por prueba y error el controlador, así como identificar un modelo y sintonizar con una regla de sintonización de un manual de control de PID. El día de la presentación del proyecto, se realizará una competencia entre todos los equipos. Utilizando un sistema de adquisición de datos en *Labview*, se medirá en línea la integral del valor absoluto del error (IAE), de manera que el equipo con el menor IAE gana.

4. El procedimiento para la prueba será el siguiente:

1. Cada equipo deberá llevar el controlador conectado a la planta.
2. Mediante la interfaz de *LabView* se pondrá el controlador en modo manual.
3. Se llevará la planta **manualmente** a un punto de operación de manera que la salida del sistema sea 50%.
4. Se colocará el *setpoint* en 100%. Se permanecerá 5 segundos en este estado.
5. Se pasará el controlador a modo automático y en ese instante, se empezará a calcular el IAE.
6. La prueba terminará al cumplir los 60 segundos.

5. Sobre los equipos

Los equipos de trabajo se conformarán por un máximo de cuatro personas.

6. Sobre la presentación (50%)

Se evaluará la exposición final del trabajo, por parte de todos los miembros del equipo, y cada miembro será evaluado por aparte según su desempeño. La duración de la presentación es 10 minutos por equipo.

7. Anexo

Controlador *PID* Discreto

■ Algoritmo PID Estándar

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

■ Tiempo de muestreo

$$\Delta t = t_k - t_{k-1} \quad \begin{array}{l} t(k-1) \text{ valor discreto} \\ \text{anterior de tiempo} \end{array}$$

■ Aproximación de la integral

$$\int_0^{t_k} e(\tau) d\tau \approx \sum_{j=1}^k e_j \Delta t$$

■ Aproximación de la derivada

$$\left. \frac{de(t)}{dt} \right|_{t=t_k} \approx \frac{e_k - e_{k-1}}{\Delta t}$$

Aproximación
diferencial finita,
aproximación
rectangular hacia
atrás

■ Algoritmo de posición (absoluto)

$$u_k = K_p \left[e_k + \frac{\Delta t}{T_i} \sum_{j=1}^k e_j + \frac{T_d}{\Delta t} (e_k - e_{k-1}) \right]$$

■ Algoritmo de velocidad (incremental)

$$\Delta u_k = u_k - u_{k-1}$$

■ Evaluamos el algoritmo de posición en k-1:

$$u(k-1) = K_p \left\{ e(k-1) + \frac{\Delta t}{T_i} \sum_{j=1}^k e_j + \frac{T_d}{\Delta t} [e(k-1) - e(k-2)] \right\}$$

- $\Delta u(k) = u(k) - u(k-1)$
- $$\Delta u(k) = K_p \left\{ e(k) + \frac{\Delta t}{T_i} \sum_{j=1}^k e_j + \frac{T_d}{\Delta t} [e(k) - e(k-1)] \right\} - K_p \left\{ e(k-1) + \frac{\Delta t}{T_i} \sum_{j=1}^{k-1} e_j + \frac{T_d}{\Delta t} [e(k-1) - e(k-2)] \right\}$$
- $$\Delta u(k) = K_p \left\{ e(k) - e(k-1) + \frac{\Delta t}{T_i} e(k) + \frac{T_d}{\Delta t} [e(k) - 2e(k-1) + e(k-2)] \right\}$$
- $$\Delta u_k = K_p \left[e_k - e_{k-1} + \frac{\Delta t}{T_i} e_k + \frac{T_d}{\Delta t} (e_k - 2e_{k-1} + e_{k-2}) \right]$$

$$u_k = u_{k-1} + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e_k - \left(1 + \frac{2T_d}{\Delta t} \right) e_{k-1} + \frac{T_d}{\Delta t} e_{k-2} \right]$$

Esta es la ecuación para implementar en el código.