

# LABORATORIO No 2: Métodos de Ziegler Nichols para sintonizar un control PID.

## (Noviembre de 2023)

Laura Valentina Buitrago Diaz  
[laurav.buitragod@uqvirtual.edu.co](mailto:laurav.buitragod@uqvirtual.edu.co)  
Universidad del Quindío

### Resumen

Este laboratorio se enfoca en poner en práctica los métodos de Ziegler Nichols, utilizando herramientas de simulación para emular y analizar los datos de un controlador PID, esto se realizó con la función de transferencia de tercer orden asignada en la primera práctica, con el fin de determinar y comprobar los parámetros necesarios para el control de sistemas. Se determinó la ganancia crítica y los parámetros de un PID.

### I. INTRODUCCIÓN

En el presente informe se expone el diseño de un controlador el cual permite eliminar el error de posición estático, esto gracias a los dos métodos de Ziegler Nichols determinando los parámetros del controlador proporcional, integral y derivativo (PID), obteniendo así un valor de variable deseada.

Esta práctica se realizó haciendo uso de la función de transferencia de tercer orden realizada en la primera práctica de los laboratorios, observando el comportamiento de cada uno de los métodos implementados y analizando cuál de los dos mejora la respuesta del sistema. Se determinaron los valores de  $K_p$ ,  $T_i$  y  $T_d$  para calcular en función del punto crítico y la frecuencia de oscilación su ganancia crítica y unitaria, estos proporcionarán una configuración inicial para el controlador PID que debería ser cercana al sistema.

Se partió de la emulación del sistema hallando el sobreimpulso, la ganancia y el tiempo de establecimiento. Posteriormente se implementó el control de mejor respuesta obtenida de los dos métodos de Ziegler Nichols analizando así su comportamiento.

Al finalizar esta práctica se habrán afianzados los conocimientos sobre la implementación de controladores que permiten mejorar la respuesta de un sistema de tercer orden, haciendo uso de las herramientas necesarias, como los software LabView, Matlab y Arduino.

### II. MÉTODOS E INSTRUMENTOS

#### A. Instrumentos

Software:

LabVIEW, Matlab y Arduino.

Tabla 1. Lista de componentes.

| Cantidad | Elemento         | Referencia o valor |
|----------|------------------|--------------------|
| 1        | Microcontrolador | Arduino Uno        |
| 1        | Microcontrolador | Esp32              |
| 1        | Capacitor        | 10 $\mu$ F         |
| 1        | Resistencia      | 1k $\Omega$        |

#### B. Métodos.

La metodología de esta práctica partió de la implementación del controlador PID usando los dos métodos de Ziegler Nichols teniendo en cuenta las tablas 2 y 3 las cuales contienen las reglas de cada controlador, teniendo en cuenta los parámetros necesarios obtenidos del análisis de la gráfica de la función de transferencia de tercer orden asignada en la primera práctica con un escalón determinado para obtener una estabilidad en el punto de operación de 1.5.

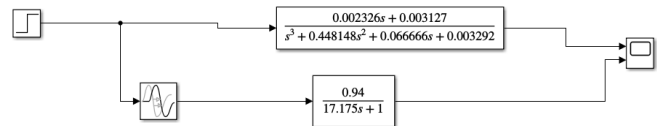


Figura 1. Diagrama de bloques desde simulink.

Tabla 2. Primer método.

| Tipo de controlador | $K_p$              | $t_i$    | $t_d$ |
|---------------------|--------------------|----------|-------|
| P                   | $\frac{T}{LK}$     | $\infty$ | 0     |
| PI                  | $0.9 \frac{T}{LK}$ | 3.33L    | 0     |
| PID                 | $1.2 \frac{T}{LK}$ | 1.2L     | 0.5L  |

Tabla 3. Segundo método.

| Tipo de controlador | Kp           | ti           | td            |
|---------------------|--------------|--------------|---------------|
| P                   | $0.5K_{cr}$  | $\infty$     | 0             |
| PI                  | $0.45K_{cr}$ | $0.83P_{cr}$ | 0             |
| PID                 | $0.6K_{cr}$  | $0.5P_{cr}$  | $0.125P_{cr}$ |

Ya obtenidos los controladores por ambos métodos se realiza la simulación utilizando simulink para visualizar y analizar las respuestas de cada uno. Para el primer método se utilizó el diagrama de la figura 2.

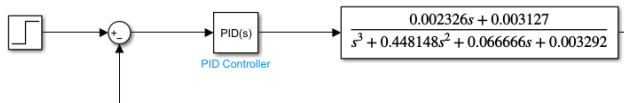


Figura 2. Diagrama de bloques desde simulink primer método.

Para el segundo método, se partió de la ganancia crítica la cual se obtiene al variar una ganancia hasta tener unas oscilaciones sostenidas sin variación alguna, para esto, se implementó el diagrama de la figura 3.

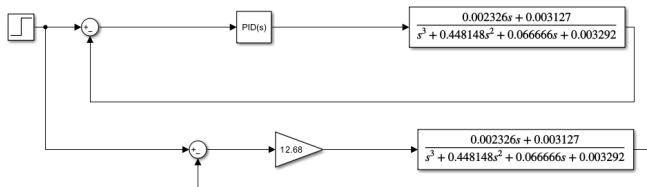


Figura 3. Diagrama de bloques desde simulink segundo método.

Posteriormente, ya obtenida la discretización de la función de transferencia de tercer orden, la cual fue asignada en la primera práctica, se realizó la misma conexión de la tarjeta Arduino UNO y la tarjeta ESP32 para la adquisición de los datos como se muestra en la figura 4.

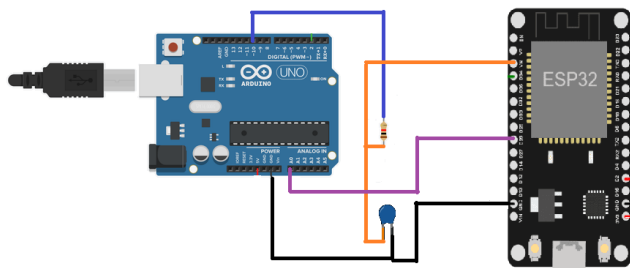


Figura 4. Conexiones del Arduino UNO a la tarjeta ESP32.

Ya obtenidos los resultados desde simulink se procedió a la implementación desde la herramienta LabVIEW, observando la respuesta simulada y emulada del sistema, se partió del

diagrama de la figura 6.

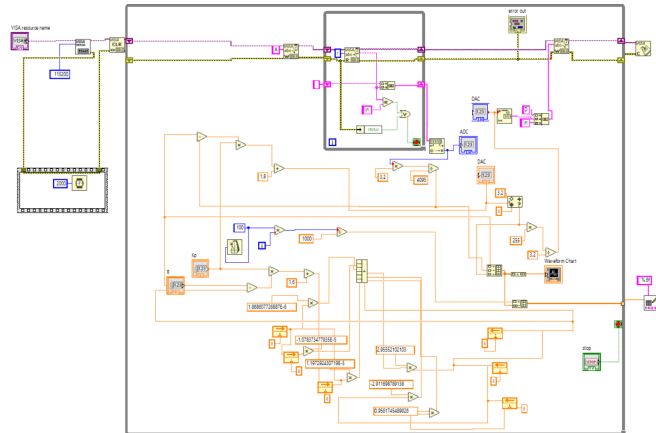


Figura 5. Diagrama desde LabVIEW.

### III. RESULTADOS Y DISCUSIÓN

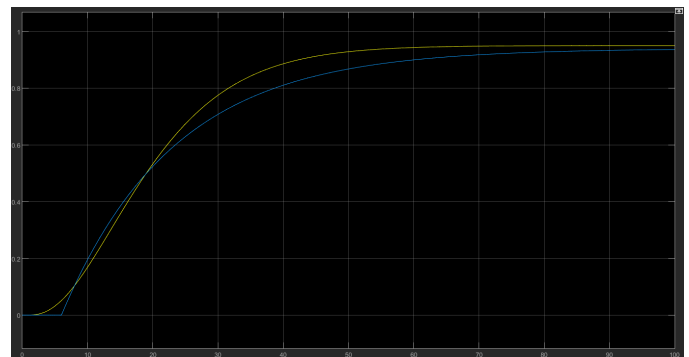


Figura 6. Respuesta de la función desde simulink.

#### Primer método

$$\frac{y(s)}{u(s)} = \frac{Ke^{-Ts}}{Ts^{-1}}$$

Observando la gráfica de la figura se obtuvieron los siguientes parámetros

$$k = 0.94$$

$$L = 5$$

$$T = 17.175$$

Con lo anterior se obtuvieron los parámetros para el diseño del controlador PID

$$k_p = 1.2 \frac{T}{LK} = 4.3851$$

$$t_i = 2L = 10 \quad t_d = 0.5L = 2.5$$

$$k_i = \frac{K_p}{t_i} = 0.43851$$

$$k_d = k_p t_d = 10.96275$$

#### Segundo método

Se parte de la ganancia crítica, la cual se obtuvo al variar una

ganancia a la entrada de la función de transferencia hasta no encontrar oscilaciones en la respuesta como se muestra en la figura 7, esta fue de 12.68.

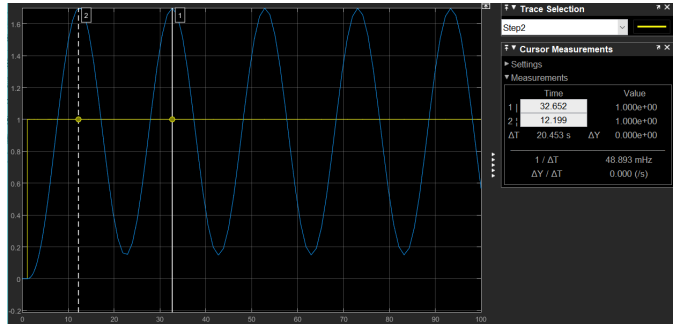


Figura 7. Ganancia critica.

Teniendo  $K_{cr} = 12.68$  se obtuvo los siguientes resultados para el controlador PID

$$K_p = 0.6K_{cr} = 7.608$$

$$P_{cr} = 32.652 - 12.199 = 20.453$$

$$t_i = 0.1P_{cr} = 10.2265 \quad t_d = 0.125P_{cr} = 2.556625$$

$$K_i = \frac{K_p}{t_i} = 0.7439495$$

$$K_d = K_p t_d = 19.4508$$

Con los resultados obtenidos según los dos métodos, se visualiza el comportamiento de cada uno como se muestra en la figura 8.

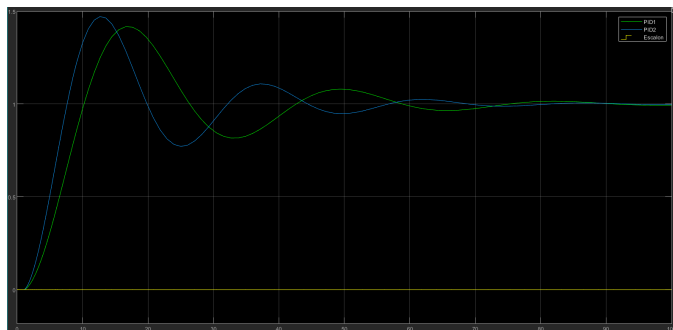
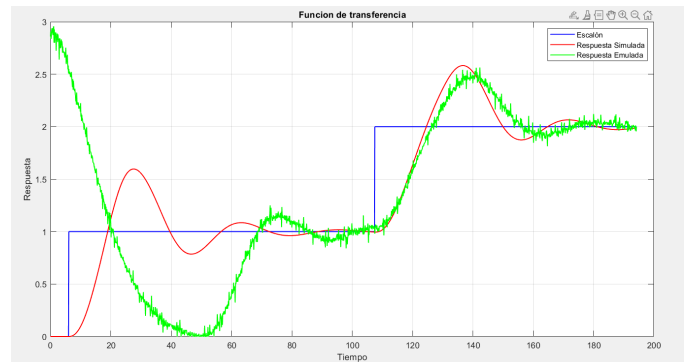
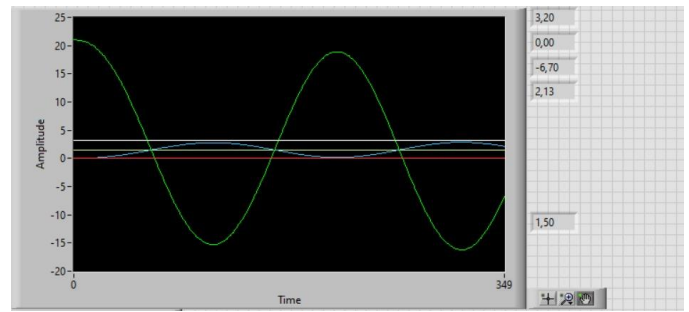


Figura 8. Controles PID por los dos métodos.

Al analizar los resultados en la gráfica según los cálculos obtenidos, se implementa el controlador PID dado del método 1, puesto que tiene mejor respuesta al tener menos porcentaje de sobrepaso. Se realizó un código en el IDE de arduino para simular el sistema, esto se logró obteniendo en tiempo discreto los datos del controlador PID.



#### IV. CONCLUSIONES

Es un método rápido y sencillo para determinar los parámetros iniciales del controlador PID. Este método puede causar oscilaciones iniciales debido a la respuesta rápida al cambio del escalón.

Este método diseñado funciona mejor en sistemas estables ya que en sistemas inestables puede generar respuestas no deseadas o inestabilidades.

#### V. REFERENCIAS

[1]. Buitrago, Laura (2023) Lab1\_Control 2.Emulación de un sistema de tercer orden. Tomado de: [https://github.com/ValentinaB2023/Laboratorios/blob/main/Lab1\\_Control2.docx.pdf](https://github.com/ValentinaB2023/Laboratorios/blob/main/Lab1_Control2.docx.pdf)

[2]. Buitrago, Laura (2023) Figura 4.Conexiones del Arduino UNO a la tarjeta ESP32.