



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR DE ALCOY

TRABAJO 1

**IMPLEMENTACIÓN DE REGULADORES PID CON TIA PORTAL
CONTROL DISCRETO DE PROCESOS INDUSTRIALES**

Sergi Navarro Pérez
Rodrigo Carpio Cerda
Julio de la Peñita Mateos

Supervisado por
Hilario Caballero, Adolfo

March 12, 2025

Contents

1	Resumen	3
2	Introducción	4
3	Herramientas de Trabajo	5
4	Funcionamiento de regulador PI - PID	6
4.1	Diagrama de bloques <i>Simulink</i>	6
4.2	Ponderación sobre la referencia	9
4.3	Anti-windup	9
5	Metodología	10
6	Resultados obtenidos	11
6.1	Ensayo lazo abierto	11
6.2	Cálculo de parámetros	12
6.3	Planta controlada - Simulink	14
6.3.1	Método Ziegler–Nichols en lazo cerrado - PI	14
6.3.2	Método Ziegler–Nichols/Cohen-Coon en lazo cerrado - PID	16
6.3.3	Ponderación sobre la referencia - PI-D	17
6.4	Implementación sobre PLC	19
7	Conclusiones y Trabajo Futuro	21
8	Bibliografía	22

List of Figures

3.1	Configuración entorno de programación de PLC - TIA PORTAL	5
4.1	Diagrama Lazo Cerrado - Simulink	7
4.2	Algoritmo PI - Bloque Simulink	8
4.3	Algoritmo PID - Bloque Simulink	8
6.1	Ensayo lazo abierto - Simulink	11
6.2	Entrada del sistema - Lazo abierto	12
6.3	Salida temperatura - Lazo abierto	12
6.4	Calculo Δy en Simulink	13
6.5	Resultado simulación Ziegler-Nichols - PI	15
6.6	Resultado simulación Ziegler-Nichols/Cohen-Coon - PID	16
6.7	Resultado simulación PID con algoritmo anti-windup	17
6.8	Resultado simulación PI-D	18
6.9	Resultado controlador tiempo real - PLC	19
6.10	Resultado comparación regulador PI-D entre Simulink y PLC	20

1 Resumen

El contenido de este informe abarca información sobre cómo programar un regulador PID, con el fin de controlar la temperatura de salida de un producto en un tanque de calentamiento, sin superar esta el límite referenciado. Para solucionar este problema, se han realizado diferentes tipos de reguladores, como PI, PID y PI-D. Primeramente se han llevado a cabo ajustes de los reguladores a través de simulaciones para luego implementarlos en un sistema de control real. Observando los diferentes resultados, se ha concluido que el mejor método para nuestro sistema es el regulador PI-D con ponderación de la referencia, ya que al tener mayor control sobre el sobrepasamiento del sistema, evita que se supere la referencia dada. Por otro lado, aunque el PID tuviera una mayor reacción frente al cambio de referencia, siempre sobrepasaba el límite.

Palabra clave: *Simulink*, PLC, regulador, PID, control del sistema.

2 Introducción

Este trabajo tiene como finalidad programar un regulador PID sobre un PLC de Siemens ,con el propósito de controlar la temperatura de salida de un producto en un tanque calentador. El objetivo que se plantea es supervisar la temperatura saliente del tanque para que esta no supere el límite dado por la referencia, ya que este exceso de temperatura podría suponer daños a futuro en el producto.

El enfoque que se ha propuesto para solucionar este proceso consistirá en realizar ensayos simulados de lazo abierto sobre una planta virtual, y a través de las simulaciones, se obtendrán datos clave, para, posteriormente, poder programar y ajustar reguladores PI y PID, tanto en simulación como en un PLC, haciendo diferentes pruebas para ver cual es el que se adapta mejor a nuestro plan de funcionamiento.

Para alcanzar este objetivo se han utilizado diferentes herramientas de simulación y sistemas de control, se pueden ver detallados en el apartado 3. Seguidamente, en el apartado 4, se ve explicado el funcionamiento de los reguladores PI y PID y sus diferentes algoritmos, para poder entender los resultados finales obtenidos. Posteriormente, se explicará la Metodología 5 a seguir, y en el apartado 6 se mostrarán los resultados de los ensayos practicados a lo largo del proceso, tanto los realizados en simulación como en el PLC.

3 Herramientas de Trabajo

Las herramientas necesarias para lograr el objetivo se van a diferenciar en dos aspectos: simulación y control real de la planta.

Para realizar las simulaciones utilizaremos el entorno de programación visual *Simulink*, en esta simularemos sobre una planta virtual ensayos de lazos abierto y cerrado, con reguladores PI y PID, probando diferentes ajustes en sus parámetros. Podemos ver un ejemplo de programación de bloques de *Simulink* en la Figura 4.1. Este entorno de programación funciona sobre la plataforma *MATLAB*, la cual utilizaremos para poder calcular los parámetros necesarios y comparar las gráficas dadas por los diferentes tipos de ajuste.

Por otro lado, en la Figura 3.1 se puede observar el entorno que utilizaremos para poder ejecutar el regulador en un controlador en tiempo real. Como se puede observar, TIA Portal será nuestro entorno de programación para controlar el PLC Siemens S7-1214C, pero en nuestro caso, estableceremos conexión a través de red. En este caso, la tarjeta NI USB-60001, se utiliza interactuar con el controlador, utilizando el modelo virtual implementado en *Simulink*. Aunque el modelo sea virtualizado, el PLC está funcionando como si estuviera controlado un proceso de planta real, como si tuviéramos el tanque calentador presente.

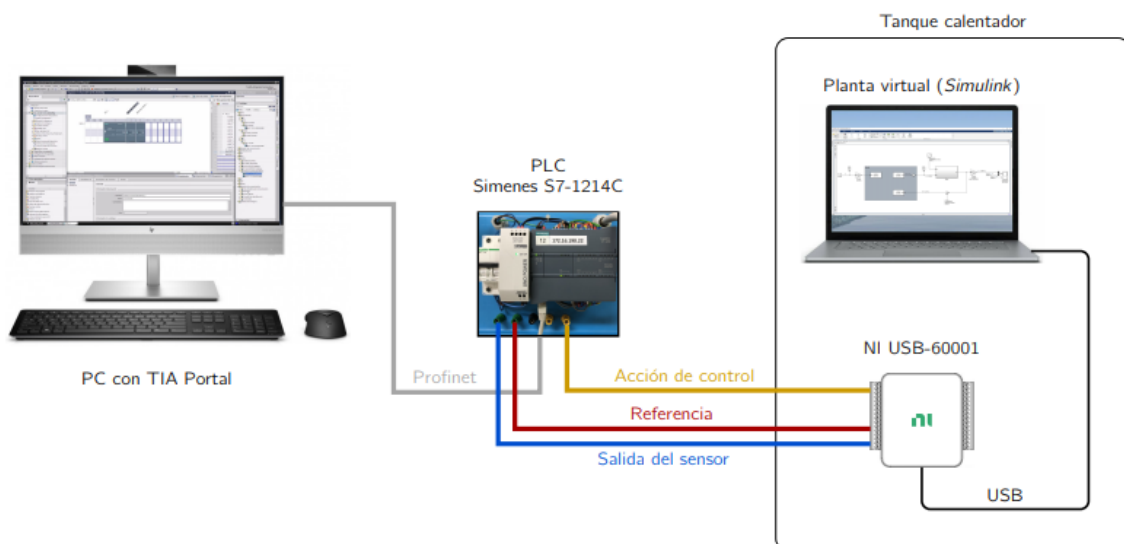


Figure 3.1 – Configuración entorno de programación de PLC - TIA PORTAL

4 Funcionamiento de regulador PI - PID

Para conseguir de manera eficiente el control de la planta, se va a utilizar un proceso de lazo cerrado con reguladores PI y PID. Este ensayo se caracteriza por un sistema de control automático, basado en la realimentación, con el cual supervisamos el funcionamiento del sistema, donde de manera autosuficiente se corregirán las variaciones de nuestra salida respecto al funcionamiento base.

Para realizar los ensayos de lazo cerrado de PI y PID, primero partiremos de unos cálculos realizados mediante el método de lazo abierto (proceso sin realimentación), para obtener los parámetros necesarios para el ajuste de la planta, siguiendo el procedimiento de Ziegler-Nichols y Cohen-Coon. Durante este proceso, abordaremos el cálculo de los parámetros estándares de este modelo:

- **K - Constanta proporcional:** Respuesta del controlador frente al cambio de señal.
- **Ti - Tiempo integral:** Ajusta el tiempo de la acción integral, acumula el error a lo largo del tiempo y corrige estas desviaciones.
- **Td - Tiempo derivativo:** Ajusta la acción de control frente a la tasa de cambio del error.

4.1 Diagrama de bloques *Simulink*

Como he mencionado en el apartado HERRAMIENTAS DE TRABAJO, para realizar las simulaciones de nuestro regulador trabajaremos con el entorno de programación simulink.

En la Figura 4.1 se puede observar el diagrama de bloques de Simulink propuesto para realizar el ensayo de lazo cerrado. También se pueden observar las señales más importantes con las que trabajaremos en este sistema realimentado.

- Tr - **Referencia de temperatura**, en °C
- T - **Señal de temperatura escalada**, en unidades en rango de [0, 250°C]
- U - **Señal de control**, en unidades de salida del regulador
- Te - **Perturbación Temperatura de entrada**, en °C
- Tm - **Salida controlada de la Planta**, unidades en rango de [0, 10V]

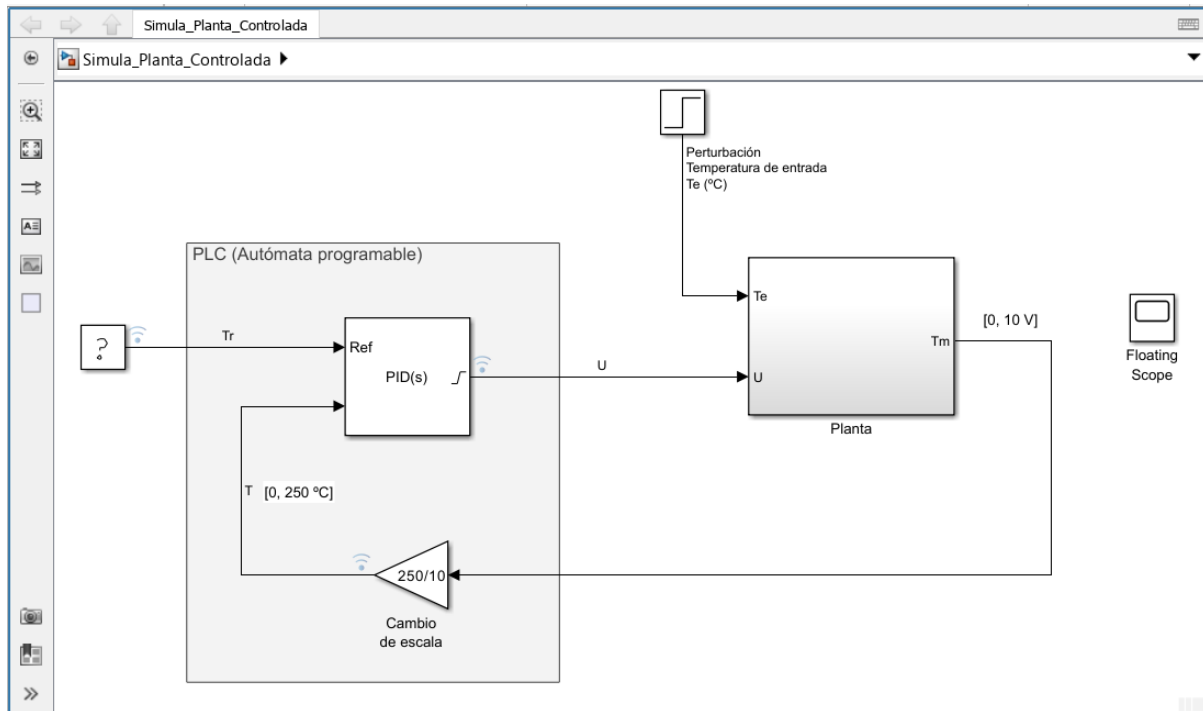


Figure 4.1 – Diagrama Lazo Cerrado - Simulink

Como hemos dicho anteriormente, la Figura 4.1 representa el funcionamiento de un sistema de control automático. En nuestro caso, queremos corregir de manera automática las desviaciones que se puedan producir en la Temperatura y alcanzar nuestra referencia. Esta lógica de control, se lleva a cabo dentro del bloque de PI o PID.

En la Figura 4.2 podemos observar el algoritmo que estaría dentro de un bloque PI de Simulink. En la izquierda de la imagen, bloque de entrada que se realiza para el cálculo del error. Lo obtenemos mediante esta ecuación:

$$e(t) = Tr - T$$

De forma directa, esta se multiplica por la constante proporcional K_p , que tratará de reducir este de forma inmediata. Como K_p no es capaz de solucionar el error a lo largo del tiempo por ella sola, se le aplica la acción integral T_i , que acumula el error a lo largo de la ejecución para poder realizar un mejor ajuste. El sumatorio de ambas señales es la acción de control que se proporciona. Este término lo podemos ver reflejado en la siguiente ecuación:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt$$

Con este método conseguimos eliminar el error de posición, pero este proceso es propenso a oscilantes, como se puede observar en la Figura 6.5. Por ese motivo, para realizar un ajuste óptimo de nuestro sistema debemos añadir la acción integral, T_i , es decir, ajustar un regulador PID, que es el bloque que vemos en nuestro diagrama de simulink en la Figura 4.1. Como podemos ver en la Figura 4.3, donde se puede observar el algoritmo interno del PID, se añade el bloque derivativo. Su función es prevenir al sistema ante futuras variaciones, mostrando una señal de control más estable y sin oscilaciones. Como vemos en la Figura 6.6. La expresión matemática del PID es la siguiente:

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

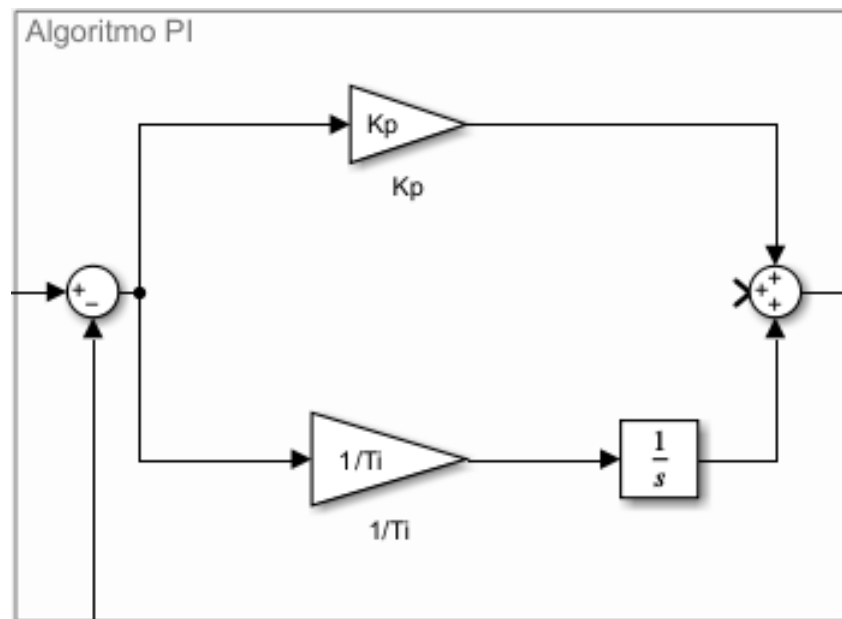


Figure 4.2 – Algoritmo PI - Bloque Simulink

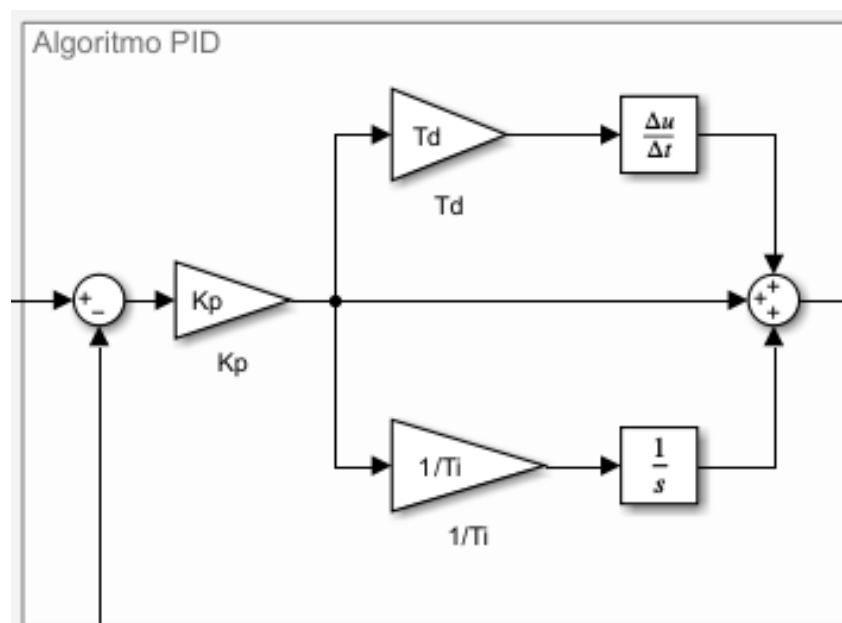


Figure 4.3 – Algoritmo PID - Bloque Simulink

4.2 Ponderación sobre la referencia

Por otro lado, a la ecuación del PID 4.1, se pueden introducir nuevos parámetros, para realizar ajustes ponderando la referencia en la parte proporcional y en la parte derivativa. Son los siguientes:

- β : Ponderación de la referencia en la acción proporcional
- γ : Ponderación de la referencia en la acción derivativa

De esta forma, la ecuación seguiría la siguiente lógica:

$$u = K_p(\beta Tr - T) + K_i \int_0^t e dt + K_d \gamma \left(\frac{dTr}{dt} - \frac{dT}{dt} \right)$$

(1)

Modificando estos parámetros, podemos obtener diferentes ajustes en el regulador.

1. Controlador PI-D:

Este regulador se obtiene con los valores de $\beta = 1$ y $\gamma = 0$ en la expresión 4.2.

$$u = K_p e + K_i \int_0^t e dt - K_d \frac{dy}{dT}$$

De esta forma no se deriva la referencia(Tr) y solo se deriva la salida del sistema(T). Con esta modificación eliminamos fuertes variaciones en la acción de control frente a cambios en la referencia.

2. Control PI-D con ponderación de la referencia:

Este regulador se alcanza con los valores de $0 < \beta < 1$ y $\gamma = 0$ en la ecuación 4.2.

$$u = K_p(\beta Tr - y) + K_i \int_0^t e dt - K_d \frac{Ty}{dt}$$

En este caso, ponderamos la referencia de la acción proporcional en el valor dado de β para tener un mayor control sobre el sobrepasamiento del sistema. Además, se sigue derivando solo la salida(T)..

Ambos métodos los podemos ver reflejados en la Figura 6.8.

4.3 Anti-windup

El efecto windup es un comportamiento que tiene el regulador cuando la acumulación del error en la acción integral es elevada durante mucho tiempo y la acción de control alcanza el límite de saturación. Esto implica que mientras la acción de control esté saturada, el sistema da una mala reacción ya que el regulador no puede actuar sobre la salida más allá de su límite, provocando por ejemplo sobrepasamientos..

Para evitarlo, aplicamos el algoritmo anti-windup. Este trata de prevenir y corregir la acumulación de la acción integral cuando esta está saturada. Podemos ver la comparación de este método en la Figura 6.7.

5 Metodología

Para alcanzar el objetivo del proceso planteado, deberemos seguir los siguientes pasos.

1. Realizar el ensayo de lazo abierto en la planta.
2. Cálculo de los parámetros necesarios mediante el resultado obtenido.
3. Ajustar reguladores mediante los siguientes métodos:
 - (a) Ziegler-Nichols (PI, PID)
 - (b) Cohen-Coon (PID)
4. Realizar simulaciones con cada uno de los métodos mencionados controlando el modelo de la planta. Añadiendo ajuste sobre la ponderación sobre la referencia.
5. Implementar los reguladores diseñados sobre el PLC y realizar ensayos de lazo cerrado experimentales.

6 Resultados obtenidos

En este apartado vamos a ver todos los resultados obtenidos del apartado Metodología 5

6.1 Ensayo lazo abierto

El primer paso para programar un regulador PID es realizar el ensayo de lazo abierto mediante simulación.

Este proceso no utiliza retroalimentación para medir el comportamiento del sistema, es decir, se aplica la acción de control sin tener en cuenta la salida del sistema. Para realizar este algoritmo, se nos ha proporcionado un diagrama de bloques programado en el entorno *Simulink*, para entre otros objetivos, representar y analizar la variación de la salida de temperatura(T) frente al valor de entrada del sistema(U). En la Figura 6.1 vemos el esquema de bloques en Simulink. Como podemos observar, sobre la planta.

Como podemos observar en el programa Simulink, representado en la Figura 6.1, vemos que la planta recibe una señal, representada con la variable U , y al otro lado, vemos la salida de temperatura de la planta, almacenada en T . Ambos lo podemos observar en la Figura 6.2 y 6.3.

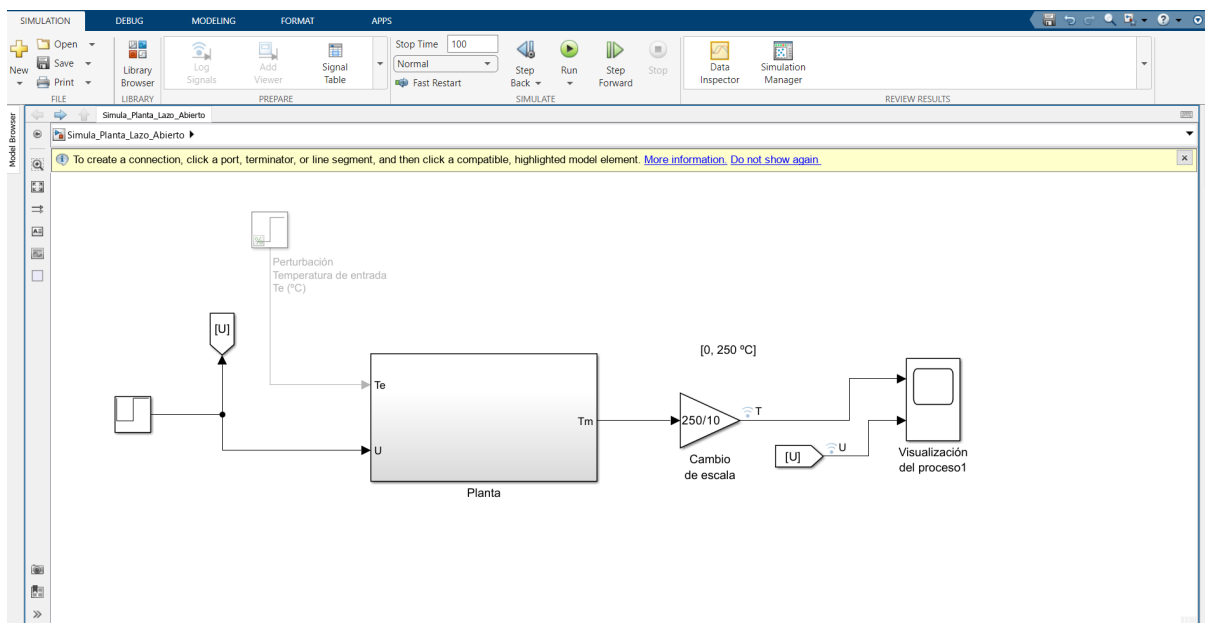


Figure 6.1 – Ensayo lazo abierto - Simulink

Al ejecutar el programa, hemos obtenido las siguientes gráficas. En la gráfica superior, correspondiente a la Figura 6.2, podemos ver la entrada del sistema y la inferior, Figura 6.3, la salida de la planta. En esta última, podemos ver la respuesta progresiva ante el aumento en la entrada del sistema.

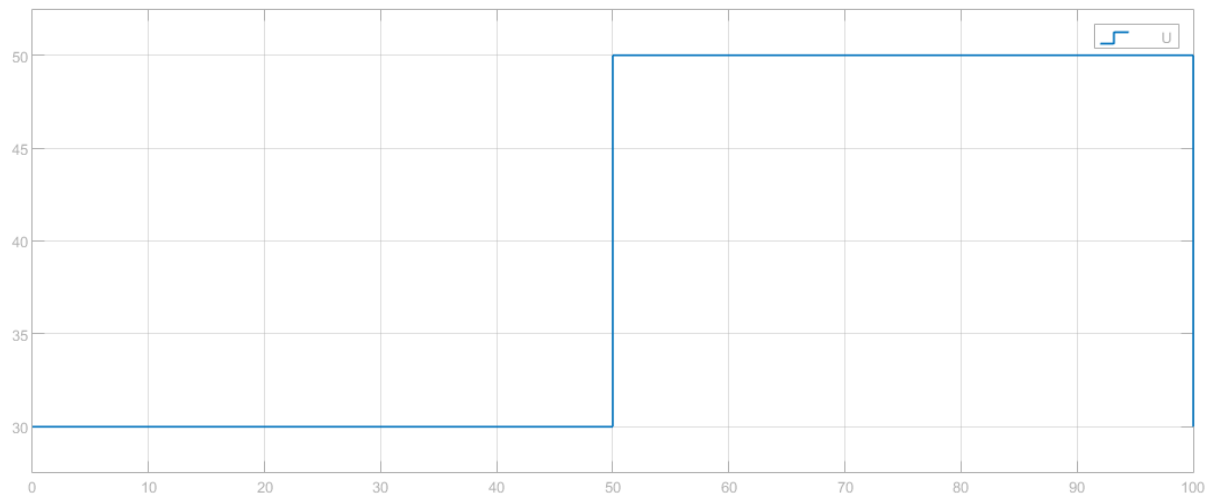


Figure 6.2 – Entrada del sistema - Lazo abierto

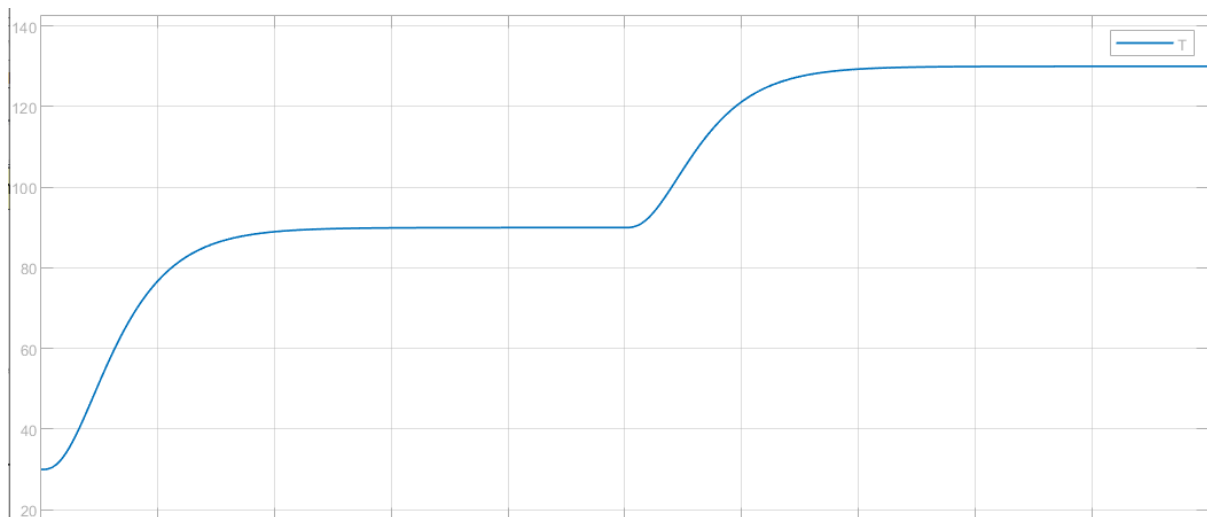


Figure 6.3 – Salida temperatura - Lazo abierto

6.2 Cálculo de parámetros

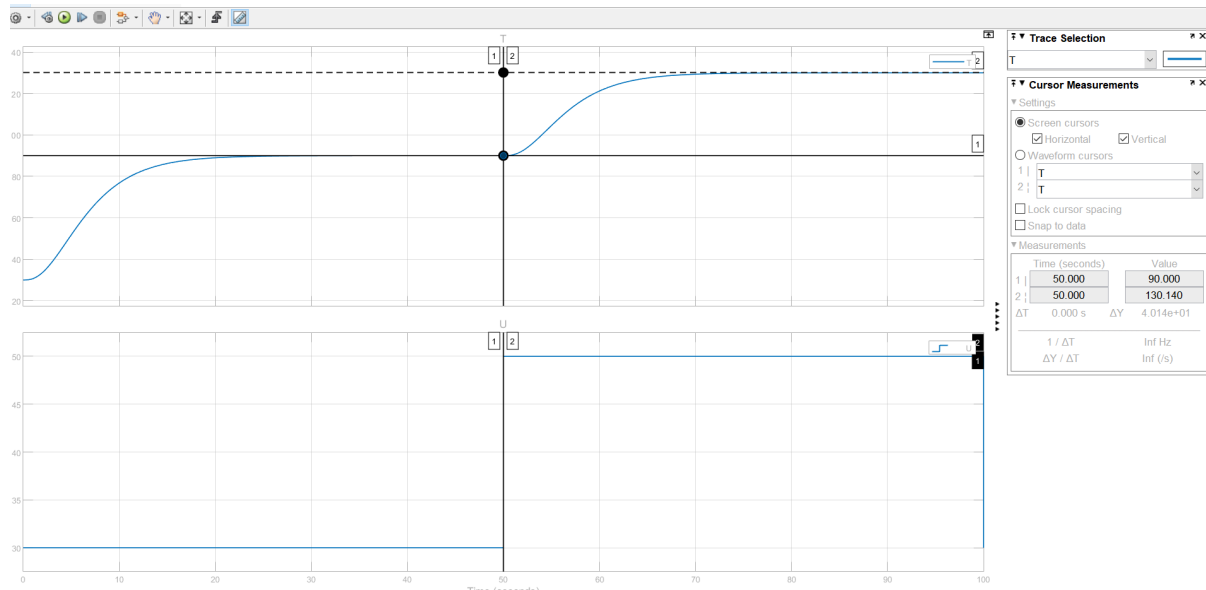
Una vez obtenidas las gráfica de lazo abierto, las analizaremos para extraer los datos necesarios, los cuales serán utilizados para la obtención de parámetros clave, posteriormente aplicados en el ajuste del PID - lazo cerrado.

1. El primer paso a seguir es medir sobre la curva de respuesta el incremento que se produce. En el caso de la entrada, como podemos ver en la Figura 6.2 es de 30 a 50, es decir, una subida de 20.

$$\Delta u = 20$$

Por otro lado, utilizando herramienta, *Cursor Measurements* de *Simulink*, Figura 6.4, como podemos ver en la Figura , el incremento en la salida es:

$$\Delta y = 40,14 = 100,14 - 90$$

Figure 6.4 – Cálculo Δy en Simulink

- Seguidamente, sobre la curva de reacción, medimos cuanto tarda el sistema en alcanzar el 28% y el 63% del valor final desde que se produce el escalón en la señal del sistema. Estos incrementos de tiempo serán nombrados como t_{28} y t_{63} .

Calculamos el incremento de d_{28} y d_{63} con las siguientes ecuaciones:

$$\Delta_{28} = \Delta y \cdot 0,28$$

$$\Delta_{63} = \Delta y \cdot 0,63$$

El resultado es el siguiente: $\Delta_{28} = 11,200$ y $\Delta_{63} = 25.200$. Ahora estos valores los buscamos en la gráfica de la simulación(utilizando la herramienta anterior) el incremento de valor dado para sacar el tiempo respectivo. Esto nos da un valor de $t_{28} = 4,238$ y $t_{63} = 7,500$.

- Una vez obtenidos estos datos de la gráfica, debemos calcular los parámetros de primer orden del modelo. Estos corresponden a la ganancia del sistema como la razón entre el incremento de Δy y Δu , el tiempo de retardo del sistema, tiempo que tarda en responder el sistema al aplicar la señal, y por último, la constante de tiempo, que representa de respuesta del sistema al cambio de entrada. Estas las denominaremos como K_{ep} , L y T , respectivamente. Se obtendrán mediante las siguientes ecuaciones:

$$K_{ep} = \frac{dy}{du}$$

$$T = 1.5 \cdot (t_{63} - t_{28})$$

$$L = t_{63} - T$$

El resultado es el siguiente:

$$K_{ep} = 2 ; \quad L = 2,6070 ; \quad T = 4,893;$$

4. El ultimo paso para poder pasar a realizar el lazo cerrado es obtener las variables estandares del modelo de ajuste de PI y PID de Ziegler–Nichols y PID de Cohen–Coon. Estas las podemos ver referenciadas en la sección 4. Para calcular estos diferentes tipos de ajustes debemos utilizar los datos recabados anteriormente y aplicarlos en las siguientes formulas según que métodos.

(a) PI de Ziegler–Nichols:

$$K_p = \frac{0.9}{K_{ep}} \cdot \left(\frac{T}{L}\right)$$

$$Ti = 3.3 \cdot L$$

Resultado obtenido:

$$K_p = 0.8446 ; \quad Ti = 8.6031$$

(b) PID de Ziegler–Nichols:

$$K_p = \frac{1.2}{K_{ep}} \cdot \left(\frac{T}{L}\right)$$

$$Ti = 2 \cdot L$$

$$Td = 0.5 \cdot L$$

Resultado obtenido:

$$K_p = 1.1261 ; \quad Ti = 5.2140 ; \quad Td = 1.30353;$$

(c) PID de Cohen–Coon:

$$K_{pc} = \frac{1}{K_{ep}} \cdot \frac{T}{L} \left(\frac{4}{3} + \frac{1}{4} \cdot \frac{L}{T} \right)$$

$$T_{ic} = \frac{13T + 8L}{32T + 6L} \cdot L$$

$$T_{dc} = \frac{4T}{11T + 2L} \cdot L$$

Resultado obtenido:

$$K_p = 1.3762 ; \quad Ti = 5.3155 ; \quad Td = 0.8643;$$

6.3 Planta controlada - Simulink

En este apartado vamos a realizar los ensayos de lazo cerrado con los reguladores PI, PID y PI-D sobre la planta de realimentación en Simulink, referenciada en la Figura 4.1.

Para alcanzar esta meta, utilizaremos los datos obtenidos en el apartado 6.2 en el paso 4. Realizaremos simulaciones utilizando con los diferentes métodos calculados, haciendo ajustes en sus parámetros para ver con que regulador obtenemos un resultado que se acerque más a nuestro objetivo.

6.3.1 Método Ziegler–Nichols en lazo cerrado - PI

En la Figura 6.5 vemos representado, en la parte de arriba la referencia y la salida de temperatura del tanque, y en la de abajo, la acción de control. En este caso, el PI-1 se ha realizado con los datos obtenidos, pero debido al sobrepasamiento de la referencia, se ha realizado los siguientes ajuste en los parámetros para conseguir el PI-2.

$$Kp = 0.6 ; \quad Ti = 7 ;$$

De esta se tarda un poco más en llegar a la referencia, pero no excede el límite, debido a un Kp menos agresivo. Por otro lado, al reducir la acción integral, hace que esta actúe mas rapido y no deja que se acumule tanto el error, lo que produce una acción de control más reducida.

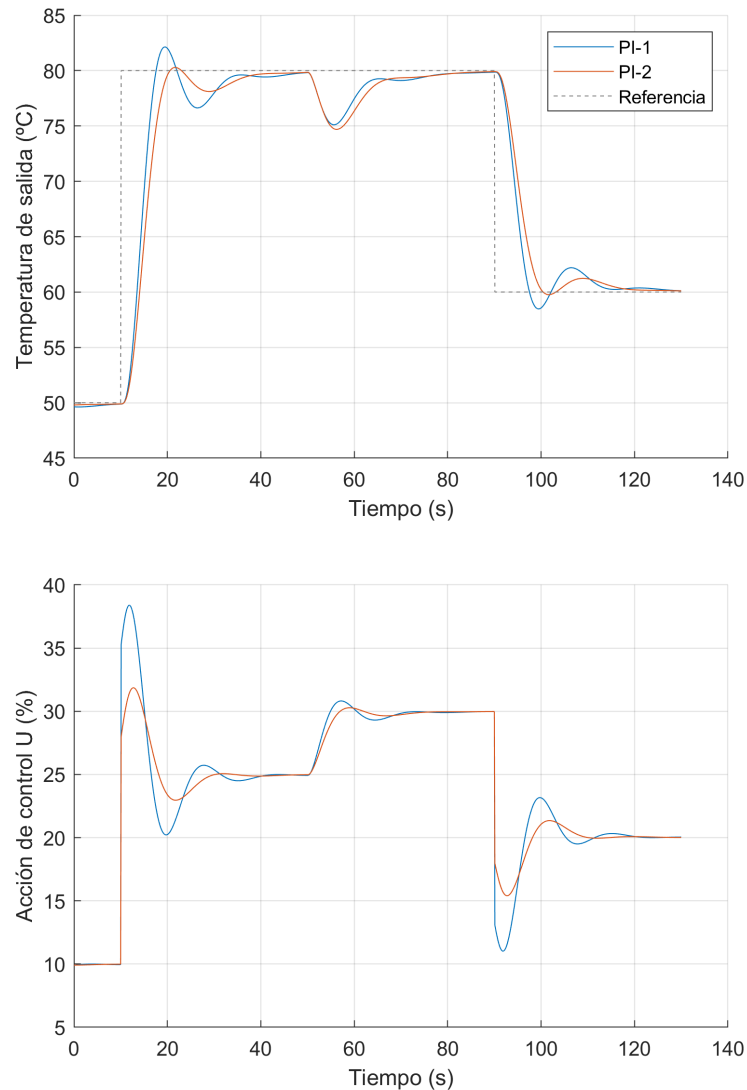


Figure 6.5 – Resultado simulación Ziegler–Nichols - PI

6.3.2 Método Ziegler–Nichols/Cohen-Coon en lazo cerrado - PID

En la Figura 6.6 podemos observar el resultado de nuestro ajuste PID. Primeramente, podemos ver que se han eliminado las oscilaciones del regulador PI de la Figura 6.5 y hemos conseguido un sistema más estable.

Los reguladores PID-1 y PID-2, se han obtenido utilizando los cálculos obtenidos en el apartado 6.2 en el paso 4, con el método Ziegler–Nichols y Cohen-Coon. Pero se puede observar que los dos tienen un gran sobrepasamiento de la referencia, por lo tanto, no se adecuan a nuestro sistema de control. Por otro lado, si realizando los siguientes ajustes sobre las tres acciones de control que se aplican al regulador, conseguimos el PID-3:

$$K_p = 0.85 ; \quad T_i = 6.2 ; \quad T_d = 1.4;$$

De esta forma la salida no supera la referencia adaptándose así a la planificación del sistema. A su vez, también tiene una reacción más estable, previniendo variaciones repentinas en la temperatura de salida.

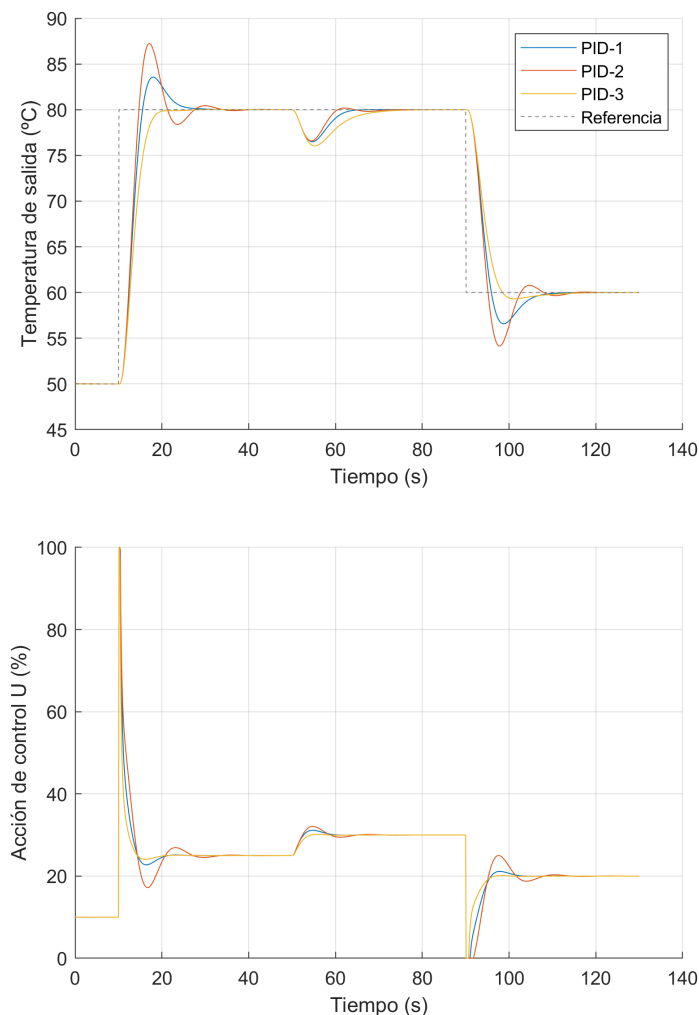


Figure 6.6 – Resultado simulación Ziegler–Nichols/Cohen-Coon - PID

Aunque vemos que el ajuste del PID-3 en la Figura 6.6 es óptimo ya que no supera la referencia, vemos que al valor de la acción de control sufre una saturación debido a la acumulación del error en la integral por el efecto *windup* 4.3. El PLC lleva integrado el algoritmo *anti-windup*, como podemos ver en la Figura 6.9, pero en el *Simulink* no. Para activarlo debemos ir al bloque PID > Saturations > *Anti-windup-Clamping*. Podemos ver el resultado de la simulación con el algoritmo activado en la Figura 6.7.

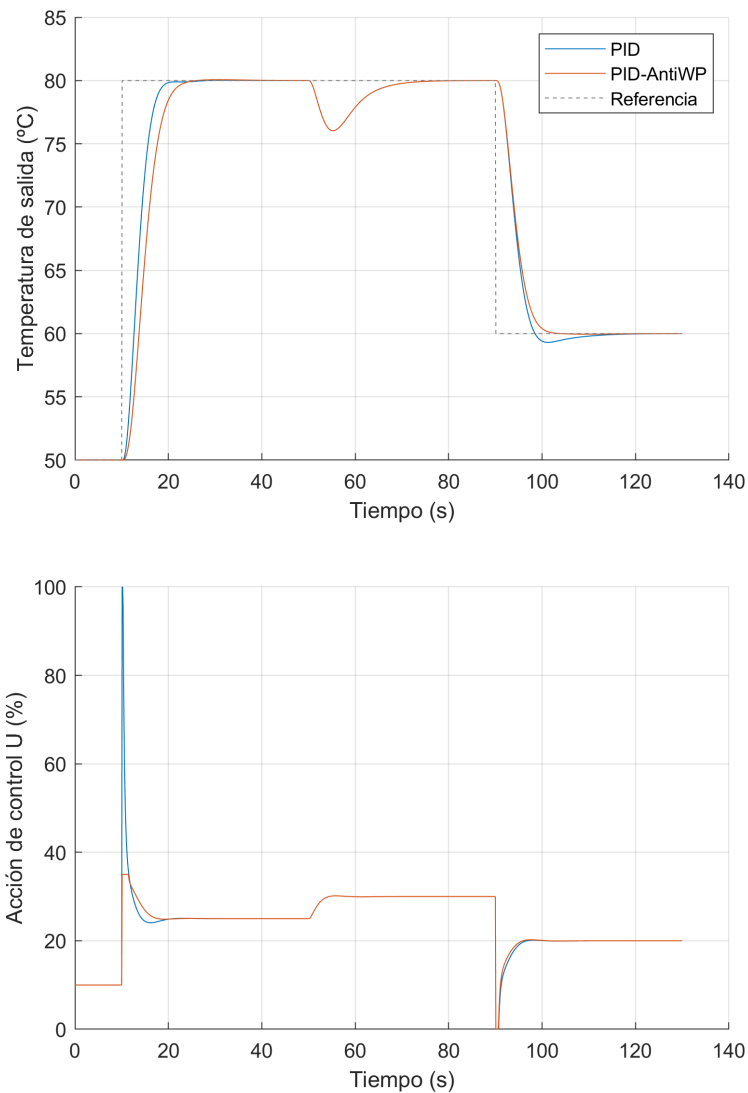


Figure 6.7 – Resultado simulación PID con algoritmo anti-windup

6.3.3 Ponderación sobre la referencia - PI-D

En la Figura 6.8 vemos los resultados obtenidos al realizar ensayos ajustando los valores de β y γ , ponderando sobre referencia en la parte proporcional y derivativa 4.2. En el caso PI-D, se ha realizado utilizando los valores $\beta = 1$ y $\gamma = 0$, pero como vemos hay un sobrepasamiento de la referencia, por lo que hemos realizado un ajuste PI-D con ponderación de la referencia, para tener un mayor control de este. Se ha obtenido con $\beta = 0.2$, $\gamma = 0$ y $K_p = 2$.

En la gráfica inferior de la Figura 6.8, correspondiente a la acción de control, se puede ver que tanto el PI-D como el PI-D con ponderación de la referencia tiene una acción mucho más estable y menos brusca que el PID (mismo que el PID-1 de la Figura 6.6), debido a que los dos primeros solo se deriva la salida del sistema, y en el segundo el error y la salida, por eso tiene una acción de control más agresiva.

Por otro lado, también se puede ver que el sobrepasamiento de PI-D es mayor que el de PID, porque como la derivada no actúa sobre el error, esto hace que la corrección de la salida frente a cambios de referencia sea menos efectiva que en el regulador PID.

Por último vemos que el regulador PI-D con ponderación de la referencia tiene una menor perturbación que los otros dos, debido a que β tiene un valor de 0.2, lo que hace que dependa del error en dicha medida, con lo cual, le hace más estable frente a perturbaciones.

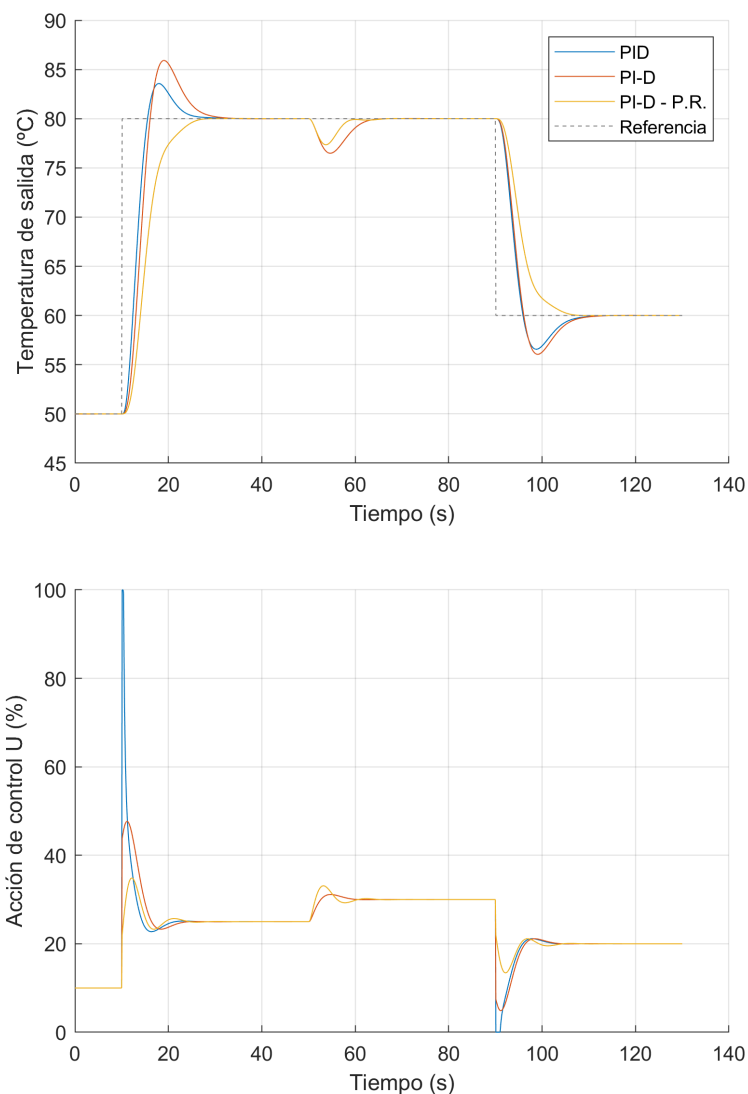


Figure 6.8 – Resultado simulación PI-D

6.4 Implementación sobre PLC

Por último, se van a mostrar las gráficas finales del informe, donde se reflejarán los ensayos realizados sobre el PLC S7-1214C de Siemens utilizando el entorno de programación TIA PORTAL.

Los resultados obtenidos de este ensayo se pueden observar en el Figura 6.9. Los datos del PID corresponden con los datos obtenidos por el método Ziegler-Nichols en el apartado 6.2 en el paso 4 y el PI-D con ponderación de la referencia es el implementado en la Figura 4.2.

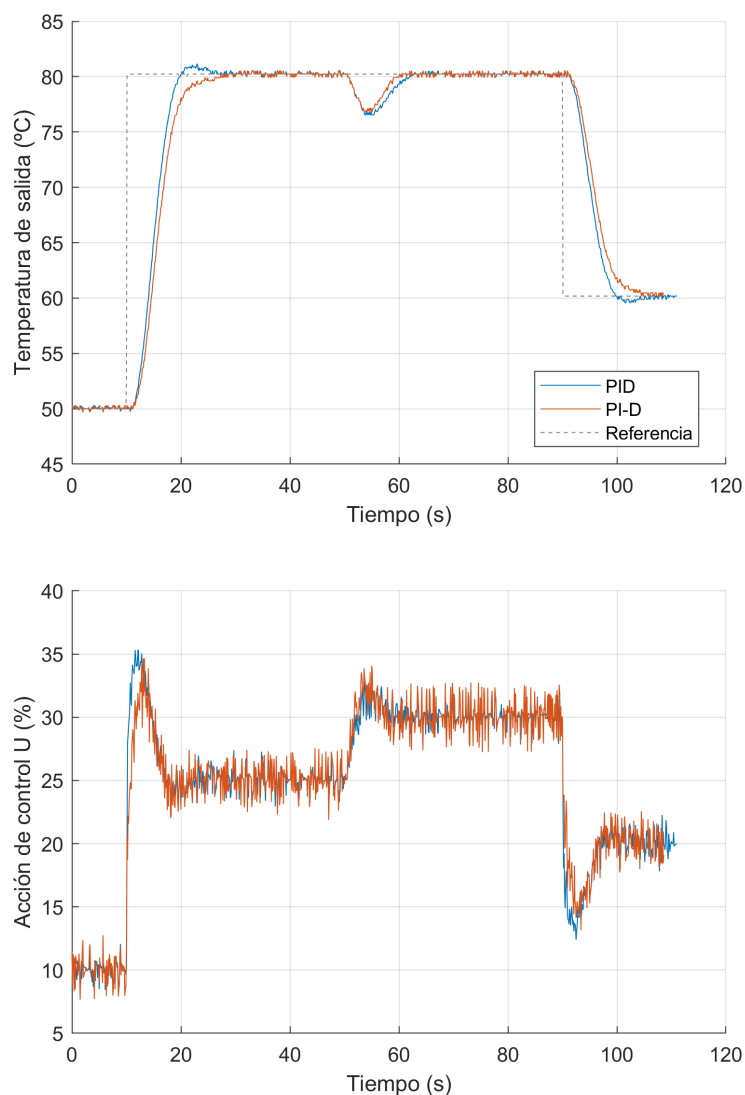


Figure 6.9 – Resultado controlador tiempo real - PLC

Como ya se ha visto anteriormente en la Figura 6.9, no cambia el resultado, el PID, aunque llega antes a la referencia, supera ligeramente esta, por lo cual, el regulador PI-D sería más óptimo para nuestro plan de funcionamiento.

Por otro lado, el PLC tiene el algoritmo anti-windup integrado en el sistema, lo que hace que la acción de control no sea tan agresiva ni sature el sistema como en la Figura 6.6.

Pero, a pesar de todo, como vemos en la Figura 6.10, si comparamos los resultados del regulador de

PI-D con ponderación en Simulink y PLC, la principal diferencia es el ruido. Este factor suele ser habitual en sistemas que se ejecutan en tiempo real y puede darse por diferentes motivos, como interferencias en sensores o limitaciones en la salida de señal del convertidor.

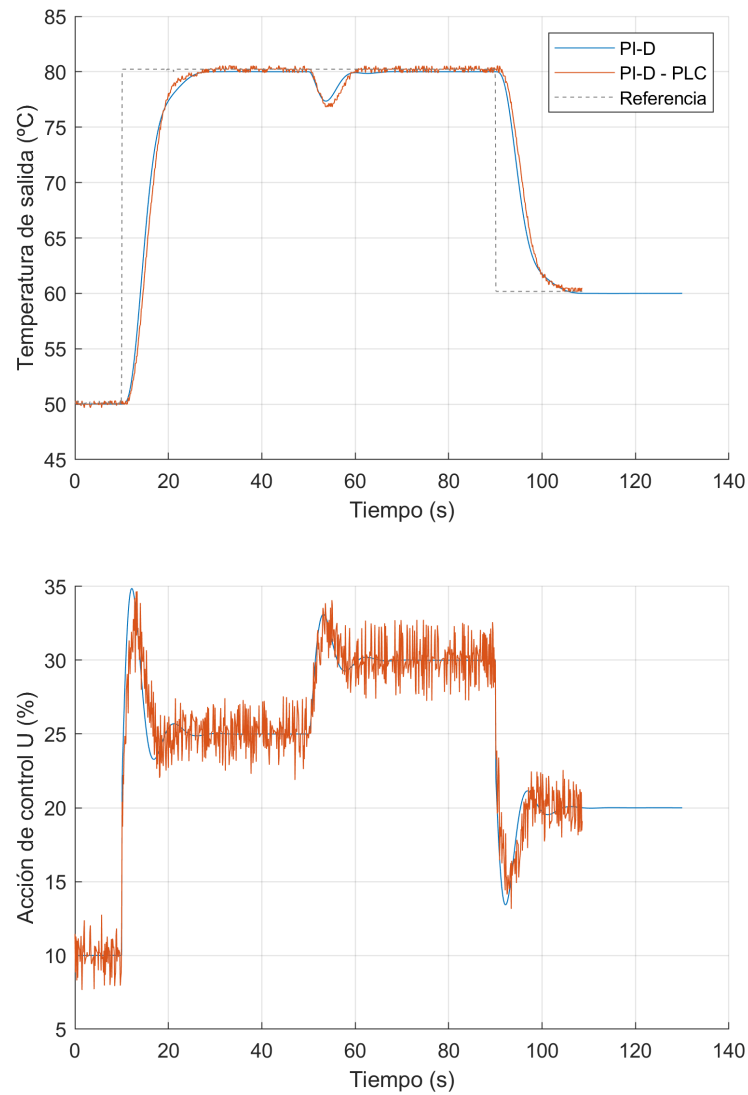


Figure 6.10 – Resultado comparación regulador PI-D entre Simulink y PLC

7 Conclusiones y Trabajo Futuro

Como hemos visto en el apartado Resultados Obtenidos 6, mejor solución para nuestra planta de control ha sido el regulador PI-D con ponderación de la referencia debido a que se adecuaba al límite de esta y no realizaba bruscas variaciones en la acción de control que implicaran la saturación de esta.

Por otro lado, posibles trabajos a futuros para la planta podría ser optimizar los ajustes para mejorar mas el rendimiento del sistema y también implementar técnicas avanzadas de supervisión y monitorización en tiempo real para obtener mas datos del sistema y saber como influyen la modificación de parámetros en este.

8 Bibliografia

A. Hilario Caballero, *Apuntes sobre Control Automático: Control Discreto de Procesos Industriales*, , Universitat Politècnica de València, 2025.