control de temperatura

ISA, proyecto 8 - Joel Sanz Martí, 2ºcfgs

Table of Contents

[1. Enunciado 2](#_Toc157367604)

[2. Documentación 3](#_Toc157367606)

[2.1. Diagrama de Bloques 3](#_Toc157367607)

[2.2. Medida de Temperatura 4](#_Toc157367608)

[2.3. Esquema Eléctrico 4](#_Toc157367609)

[2.4. Cambio de Sensor de Temperatura a un Termopar 16](#_Toc157367610)

[2.5. Programa PLC 18](#_Toc157367611)

[2.6. SCADA 26](#_Toc157367613)

[2.7. Explicación Función PID en TIA Portal 29](#_Toc157367616)

[2.8. Configuración del PID 31](#_Toc157367621)

[2.9. Primer Control Proporcional 33](#_Toc157367625)

[2.10. Segundo Control Proporcional 33](#_Toc157367626)

[2.11. Primer control PI 34](#_Toc157367627)

[2.12. Segundo y Tercer Control PI 34](#_Toc157367628)

[2.13. autotuning 36](#_Toc157367629)

[2.14. Control PID 37](#_Toc157367630)

[2.15. Explicación PID con Salida PWM 37](#_Toc157367631)

[2.16. Control PID con Salida PWM 38](#_Toc157367632)

[2.17. Datalogging 38](#_Toc157367633)

[2.18. Conclusiones y Valoración Final 39](#_Toc157367634)

# 1. Enunciado

Se pretende realizar un sistema de control de temperatura mediante el cual se consiga mantener la temperatura de un recinto en un valor constante, y de forma seleccionable. El procedimiento de control será en bucle cerrado y se realizará una monitorización de forma continua en el PC. Las especificaciones del sistema a realizar son:

**a)** **Magnitud a medir y controlar:** Temperatura.

**b) Márgenes de variación:** Entre 0ºC y 100ºC seleccionable.

**c) Tipo de control:** El proceso podrá trabajar de forma MANUAL (bucle abierto) o AUTOMÁTICA (bucle cerrado). En AUTOMÁTICO el tipo de control será un PID con acciones analógicas y PWM, indistintamente.

**d) Gráficos de tendencia:** Para visualizar en tiempo real la variable de proceso.

**e) Registro de variables en fichero:** Que cree registros de las variables más significativas (temperatura, alarmas, estado de la planta, etc.) con capacidad de generación de archivos portables a entornos Microsoft Excel.

## HMI

La interfaz consistirá en un programa desarrollado con WinCC sobre un Runtime, con el que podremos controlar el funcionamiento del conjunto.

# 2. Documentación

## 2.1. Diagrama de Bloques



**Magnitud regulada (PV):** Magnitud física que se quiere controlar. En nuestro caso será la temperatura del recinto.

**Consigna (SP):** Valor deseado del PV. En nuestro caso será una temperatura seleccionable en el rango [0..100]ºC.

**Instrumento de medida:** Proporciona la señal de realimentación enviada al regulador (PLC). En nuestro caso será una señal [0..10]V proveniente del acondicionador de la Pt100 para un rango de [0..100]ºC.

**Elemento de comparación:** Forma la señal de error comparando PV con SP. En nuestro caso será el PID dentro del PLC.

**Elemento de regulación:** A partir de la señal de error, proporciona una variable correctora que actúa sobre el elemento final de control. En nuestro caso será el PID dentro del PLC.

**Mando del actuador:** Comunica al actuador cómo debe modificarse la magnitud regulada. En nuestro caso serán el SSR y los relés que controlan la resistencia calefactora, que a su vez serán controlados por las salidas analógica y PWM del PID.

**Actuador o elemento final de control:** Circuito de regulación que modifica la magnitud regulada en función de la magnitud manipulada.

**Sistema regulado:** Sistema donde se encuentra la magnitud que se quiere regular. En nuestro caso será el recinto con el horno.

**Perturbaciones:** Es la magnitud que influye de manera no deseada en la magnitud regulada y la aleja de la consigna actual. En nuestro caso será el viento que generará un ventilador.

## 2.2. Medida de Temperatura

La medida de temperatura se realizó con una Pt100 conectada a un acondicionador para conseguir una señal de [0..10]V.

El acondicionador que usamos (INOR APAQ 3LPT100) tiene una serie de jumpers que se deben soldar según la configuración. Primero ajustamos el punto “Zero” a 0ºC soldando las pistas 2 y 3 del jumper 9. A continuación, ajustamos el punto “Span” a 100ºC soldando los jumpers del 2 al 8 de esta forma:



Por último, soldamos las dos últimas pistas del pin 1 para desactivar la detección de resistencia quemada (detectar a valores de tensión >11V o <0V).

Para ajustar el sistema, usamos resistencias patrón que simularan la resistencia de la Pt100 a 0ºC y a 100ºC. Para conseguir estas resistencias usamos la siguiente fórmula:

Siendo la resistencia a la temperatura , la resistencia a la temperatura y el coeficiente de temperatura.

El nombre Pt100 indica que es una RTD de platino (Pt), y por tanto tiene un coeficiente de temperatura de , y que tiene una resistencia de a 0ºC (100). Por tanto:

Para ajustar el convertidor, primero pondremos la resistencia de . Mediremos la salida del convertidor con un voltímetro y ajustaremos el potenciómetro “Zero” hasta que midamos 0V a la salida. A continuación, cambiaremos la resistencia por la de y ajustaremos el potenciómetro “Span” hasta que midamos 10V a la salida. Cuando ajustemos un extremo del rango, el otro se habrá desajustado, por lo que repetiremos el proceso completo hasta que al poner la resistencia de midamos 0V y al poner la resistencia de midamos 10V.

## 2.3. Esquema Eléctrico

## 2.4. Cambio de Sensor de Temperatura a un Termopar

Necesitamos sustituir la Pt100 por un termopar junto con su circuito de acondicionamiento que cumplan con las características del sistema de medición de temperatura actual. Por tanto, el termopar debe tener un rango mínimo de [0..100]ºC, y el circuito de acondicionamiento debe alimentarse a 24Vdc y que su salida sea del rango [0..10]V.

El termopar elegido para sustituir a la Pt100 es el modelo E52-ETJS1-B de OMRON ([Datasheet](https://assets.omron.eu/downloads/latest/datasheet/es/h10e_e52-e_1_temperature_sensors_datasheet_es.pdf?v=8)).

A white sheet with black text

Description automatically generated

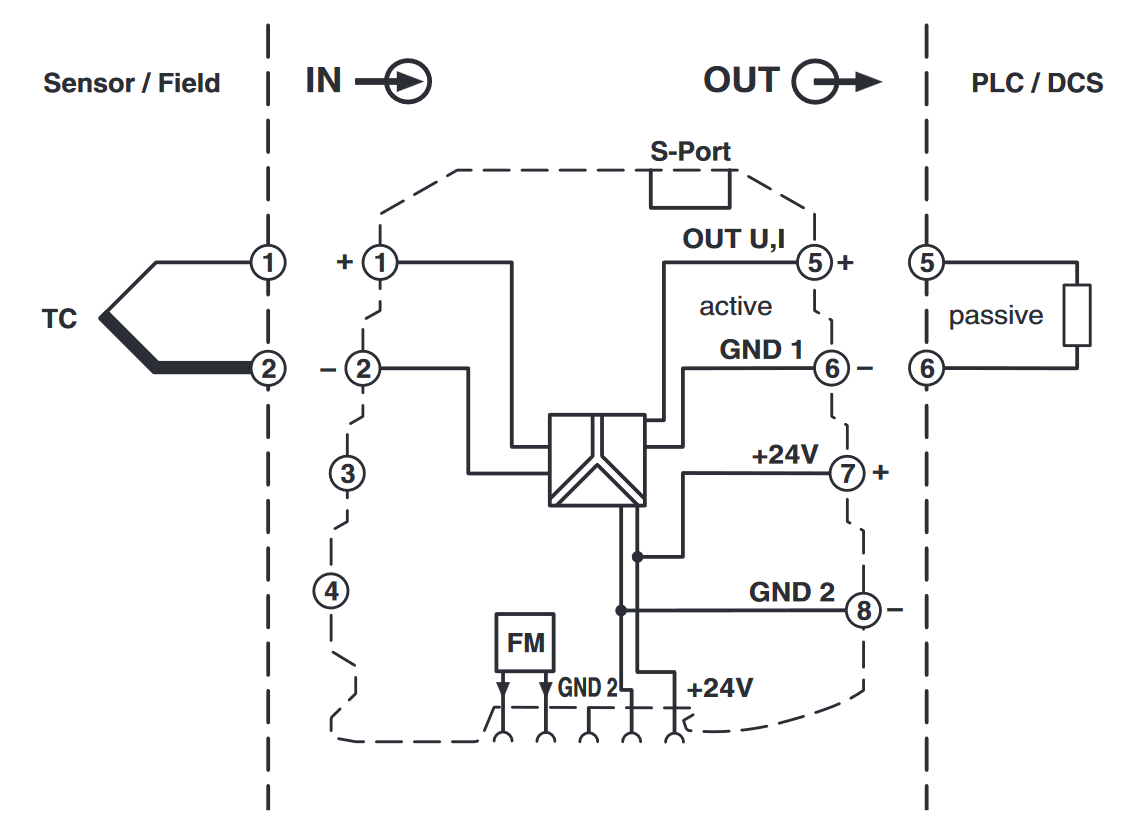
Este termopar estará conectado al conversor MINI MCR-TC-UI-NC de Phoenix Contact ([Datasheet](https://pxc-crisp-production-platform-cr-s3downloadbucket-1rf23da6xdlmt.s3.eu-west-1.amazonaws.com/7572094?response-content-disposition=inline%3B%20filename%3D%22db_en_mini_mcr_tc_ui_nc_105447_en_02.pdf%22&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240128T121155Z&X-Amz-SignedHeaders=host&X-Amz-Expires=10800&X-Amz-Credential=AKIAWWXX4BDRHAMFOGGZ%2F20240128%2Feu-west-1%2Fs3%2Faws4_request&X-Amz-Signature=99bc72455b1326a54ecc50782d9e605778f35fa9576b3a13697a5753771c3f48)).

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

En el datasheet del acondicionador hay un esquema que muestra el conexionado de este mismo:



Según muestra la anterior figura, el termopar va conectado a los bornes 1 y 2, mientras que la entrada analógica del PLC va conectada a los bornes 5 y 6, respetando la polaridad tanto del termopar como de la entrada del PLC (Indicada en la figura).

Este acondicionador tiene dos series de switches para realizar los ajustes de la medida.

La primera serie de switches sirve para configurar el tipo de termopar, la compensación fría, el rango de la señal de salida y el rango inferior de la temperatura.

A screenshot of a computer

Description automatically generated

En nuestro caso, activaremos los switches 1 (Termopar tipo J), 2 (Activar compensación fría) y 5 (Salida [0..10]V). No activando los switches 6, 7 y 8 seleccionamos el rango inferior de la temperatura a 0ºC.

La segunda serie de switches configura el rango superior de la señal de entrada.

A sheet of a graph

Description automatically generated with medium confidence

En nuestro caso, activaremos los switches 2 y 4 para seleccionar un rango superior de 100ºC.

## 2.5. Programa PLC

## 2.6. SCADA

### Pantalla 1: Control PID A screenshot of a computer Description automatically generated

Desde esta pestaña se controlan todas las variables relacionadas con el PID.

El pulsador “Ventilador” está asociado a la variable “Ventilador\_SCADA” y controla la activación y desactivación del ventilador. Actúa como telerruptor.

Los pulsadores “inicio file” y “paro file” inician y terminan la grabación del SP, PV y OUT en un documento de texto.

El campo de entrada numérico “Setpoint” está asociado a la variable “Setpoint\_PID” y desde este se introduce el punto en el que queremos que se estabilice la temperatura.

El visor numérico “Input” está asociado a la variable “Temperatura\_Pt100” y muestra la temperatura que está midiendo la sonda de temperatura en ºC.

El campo de entrada numérico “ManualValue” está asociado a la variable “Manual\_Value\_PID” y sirve para introducir el porcentaje de activación de la salida del PID cuando este está funcionando en modo manual.

El pulsador “Stop” está asociado a la variable “Reset\_PID” y sirve para poner el PID en estado inactivo (modo de funcionamiento 0).

El pulsador “Activate” está asociado a la variable “Mode\_Activate\_PID” y sirve para introducir en el PID el modo de funcionamiento seleccionado desde “Mode”.

El campo de entradas “Mode” está asociado a la variable “Modo\_PID” y muestra el contenido de la lista de textos “Modos\_PID”. Sirve para introducir el modo de funcionamiento en que queremos que funcione el PID.

El visor numérico “Output” está asociado a la variable “Salida\_PID” y muestra el porcentaje de activación de la salida del PID.

El visor numérico “State” está asociado a la variable “State\_PID” y muestra el estado en el que se encuentra actualmente el PID.

El piloto “Error” está asociado a la variable “Error\_PID” e indica si el PID se encuentra en error.

El visor numérico “ErrorBits” está asociado a la variable “ErrorBits\_PID” y muestra el código de error (en hexadecimal) en el que se encuentra el PID.

Los campos de entrada numéricos “k”, “Ti” y “Td” están asociados a las variables “PID.Retain.CtrlParams.Gain”, “PID.Retain.CtrlParams.Ti” y “PID.Retain.CtrlParams.Td” respectivamente, y sirven para ver e introducir la ganancia, tiempo integral y tiempo derivativo del PID.

El interruptor “Salida PWM” está asociado a la variable “Activar\_PWM\_SCADA” y sirve para conmutar entre salida analógica (OFF) y salida PWM (ON).

El campo de entrada numérico “Tiempo Algoritmo” está asociado a la variable “PID.Retain.CtrlParams.Cycle” y sirve para introducir el tiempo que dura un ciclo de la salida PWM, que es la suma del tiempo que se mantiene la salida encendida y el que se mantiene apagada.

Los campos de entrada numéricos “T\_On” y “T\_Off” están asociados a las variables “PID.Config.MinimumOnTime” y “PID.Config.MinimumOffTime” respectivamente, y sirven para introducir el mínimo tiempo que la salida PWM debe estar activa e inactiva cada ciclo.

El pulsador “Gráficos” sirve para cambiar a la pantalla “Gráficos”.

### Pantalla 2: Gráficos

A screenshot of a graph

Description automatically generated

Desde esta pantalla se puede observar una gráfica que muestra la evolución de la temperatura medida, el punto de consigna y las salidas analógica y PWM en el tiempo. Además, pulsar el botón “Control” cambia a la pantalla “Control”.

## 2.7. Explicación Función PID en TIA Portal

Esta es la función PID\_Compact en TIA Portal:

A screenshot of a computer

Description automatically generated

### Parámetros de Entrada

**Setpoint**: Valor en el que queremos que se estabilice la magnitud que vamos a controlar, introducida en la unidad de la magnitud.

**Input:** Entrada de la variable de proceso. Esta entrada recibe una variable interna previamente escalada (si es necesario).

**Input\_PER:** Entrada de la variable de proceso. Esta entrada recibe la señal de una entrada analógica física y realiza el escalado internamente.

**ManualEnable:** El PID se mantiene activo en modo manual mientras esta entrada esté en TRUE.

**ManualValue:** Porcentaje de activación de la salida cuando el PID trabaja en modo manual.

**Reset:** Renicia el PID y lo mantiene inactivo mientras esta entrada esté en TRUE.

**ModeActivate:** En el flanco de subida, cambia el modo de funcionamiento del PID al introducido en el parámetro de entrada “Mode”.

**Mode:** Modo del PID deseado (no se introduce hasta el flanco de subida en la variable “ModeActivate”). Los modos pueden ser Inactivo (0), Optimización Inicial (1), Optimización fina manual (2), automático (3), manual (4) y valor de salida sustitutivo con vigilancia de errores (5).

### Parámetros de Salida

**ScaledInput:** Valor de proceso escalado en función del escalado interno del PID cuando estamos utilizando Input\_PER.

**Output:** Salida porcentual del PID de [0..100]% (desde la configuración se puede configurar el rango).

**Output\_PER:** Salida de bits del PID. Recibe una salida analógica física y actúa según el escalado interno del PID.

**Output\_PWM:** Salida PWM del PID. PWM (Pulse Width Modulation) es un tipo de señal digital que simula ser una analógica variando la amplitud de los pulsos de una señal digital pulsativa.

**State:** Modo de funcionamiento actual del PID.

**Error:** Indica si el PID está en error o no.

**ErrorBits:** Indica el identificador del error del PID en hexadecimal.

### Como Cambiar El Modo de Trabajo

Para cambiar el modo de trabajo, primero tenemos que introducirlo en el parámetro “Mode”. Esto no pondrá el PID en el modo que seleccionemos. A continuación, tendremos que activar el parámetro “ModeActivate”. En el flanco de subida, este parámetro introducirá en el PID. Para comprobar que el modo se ha introducido correctamente, podemos comprobar que en el parámetro de salida “State” esté el modo de funcionamiento deseado.

### Introducción de Ti, Td y K

En el árbol de proyecto, podremos encontrar la instancia del PID en la ruta “Bloques de programa -> Bloques de sistema -> Recursos de programa”. Si accedemos a ella, veremos todas las variables del PID. Los parámetros Ti, Td y k tienen cada uno su propia variable:

k -> Static.Retain.CtrlParams.Gain

Ti -> Static.Retain.CtrlParams.Ti

Ti -> Static.Retain.CtrlParams.Td

En nuestro caso, modificamos las variables mediante campos de entrada numéricos desde SCADA. Estas variables son de entrada y salida, y pueden ser escritas tanto por el usuario como por el propio PID (por ejemplo, al realizar un autotuning). Por tanto, los campos de entrada numéricos a los que estén asociados deben ser del tipo “Entrada/Salida”.

## 2.8. Configuración del PID

### Ajustes Básicos

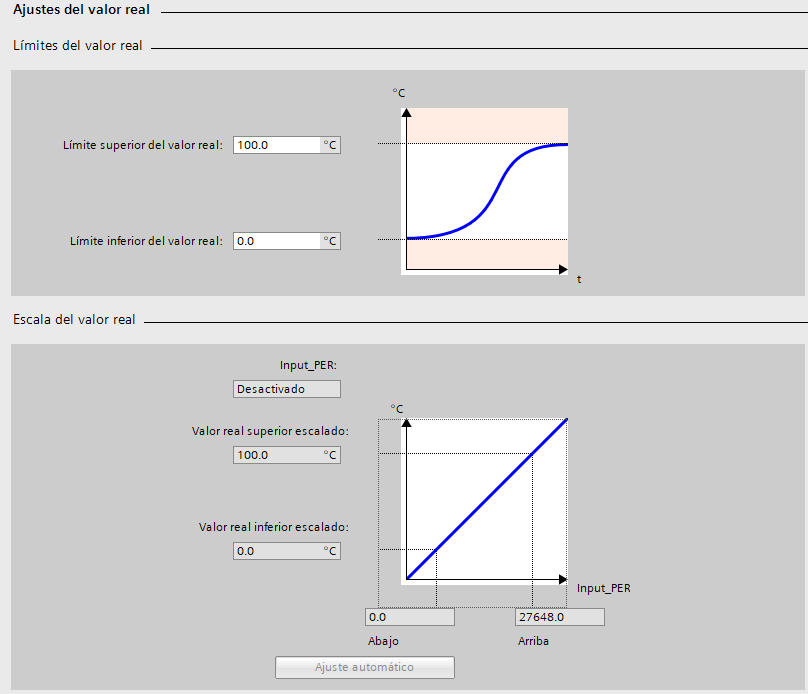
En “Tipo de regulación” indicamos que la regulación será de temperatura en ºC y en “Parámetros de entrada/salida” indicamos que vamos a trabajar con los parámetros “Input” y “Output”, que son los que reciben variables internas.

A screenshot of a computer

Description automatically generated

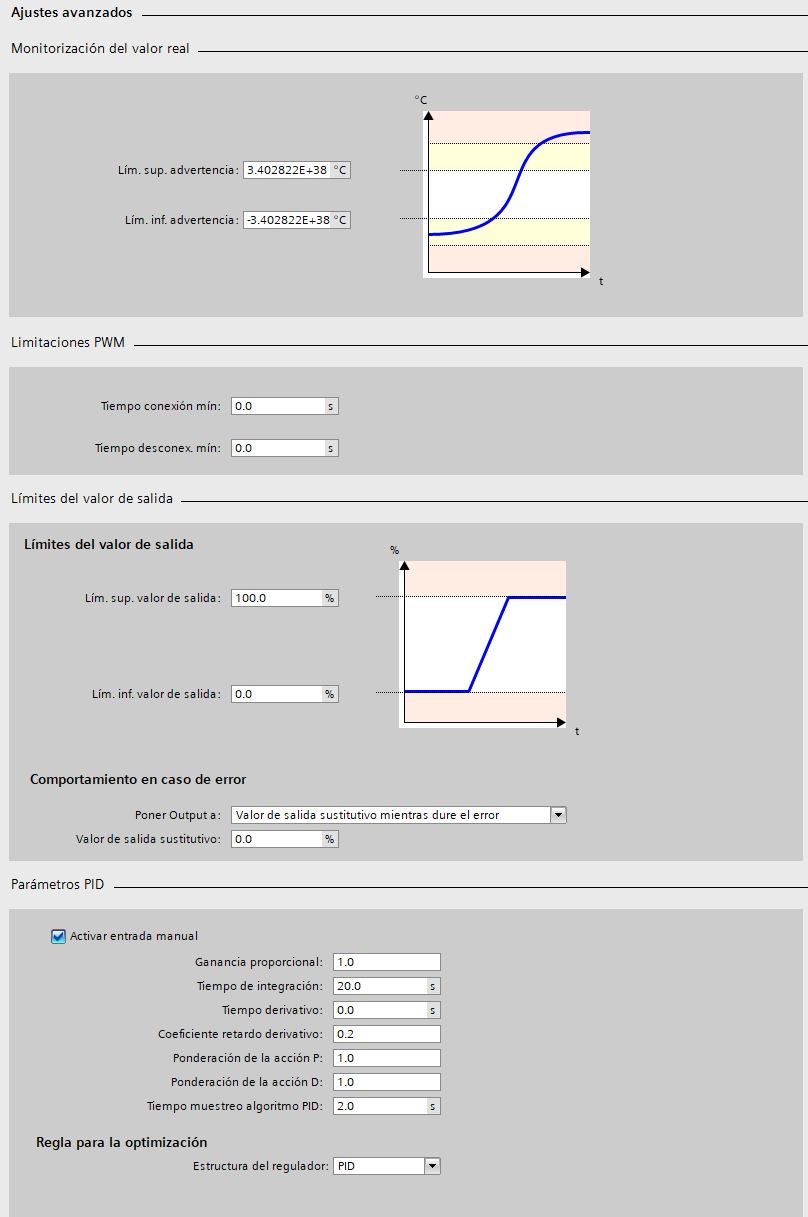
### Ajustes del Valor Real

En “Límites del valor real” dentro indicaremos el rango de la variable de entrada al PID. En “Escala del valor real” indicaríamos el escalado interno si usáramos los parámetros “Input\_PER” y “Output\_PER”, que no es el caso.



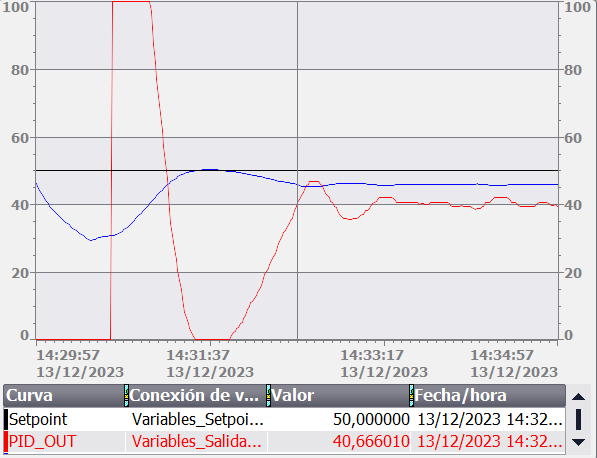
### Ajustes Avanzados

En “Limitaciones PWM” están los parámetros T\_On y T\_Off que modificamos desde SCADA y que controlan el mínimo tiempo de ON y OFF de la salida de PWM por cada ciclo. En “Límites del valor de salida” se puede limitar porcentualmente el rango de la salida (útil cuando se trabaja con Input\_PER y Output\_PER). En “Parámetros PID” están k, Ti, Td y el tiempo de muestreo (entre otros), que son variables que controlamos desde SCADA.



## 2.9. Primer Control Proporcional

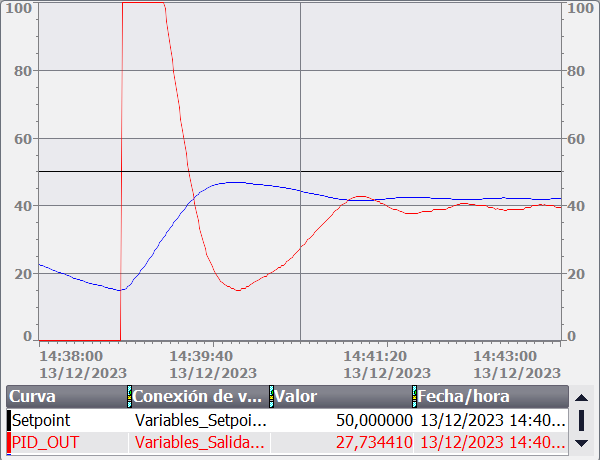
Esta es la gráfica de la evolución de un control proporcional con k = 10 y SP = 50.



Se puede observar que la temperatura se estabiliza en un punto inferior a SP, aproximadamente a 46ºC.

## 2.10. Segundo Control Proporcional

Esta es la gráfica de la evolución de un control proporcional con k = 5 y SP = 50.



Se puede observar que el sistema tarda más en estabilizarse y además hay menos oscilaciones. Esto es debido a que, al disminuir k a la mitad, la banda proporcional aumenta el doble, por lo que la salida de control empieza a bajar del 100% en un punto más alejado de SP.

También se puede observar que la temperatura se estabiliza en un punto más alejado de SP respecto a la anterior gráfica. Esto es porque, en un control proporcional, la temperatura se estabiliza en un punto dentro de la banda proporcional e inferior a SP. Si la banda proporcional es mayor, el punto donde empieza está más alejado de SP.

## 2.11. Primer control PI

Esta es la gráfica de la evolución de un control PI con k = 5, Ti = 20 y SP = 50.

A graph with red and green lines

Description automatically generated

Se puede observar que la temperatura ahora se estabiliza en el SP, a diferencia de con el control P.

## 2.12. Segundo y Tercer Control PI

Esta es la gráfica de la evolución de un control PI con k = 5, Ti = 60 y SP = 50.

A graph with lines and numbers

Description automatically generated with medium confidence

Se puede observar que la temperatura tarda mucho más en estabilizarse respecto a la anterior gráfica, pero no hay oscilaciones. Esto es debido a una acción integral poco pronunciada.

Esta es la gráfica de la evolución de un control PI con k = 5, Ti = 120 y SP = 50.

A graph with red and green lines

Description automatically generated

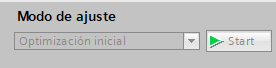
Se puede observar que siguen sin haber oscilaciones, pero la temperatura ya no se estabiliza en SP. Esto es debido a una acción integral muy poco pronunciada, provocando que el sistema sea casi como un control proporcional.

De estas gráficas se puede sacar la conclusión de que la acción proporcional tiene más efecto sobre el sistema cuanto menor es el tiempo integral Ti. Una acción integral demasiado pronunciada provocará un sistema con oscilaciones constantes (error estacionario), mientras que una acción integral demasiado pronunciada provocará un sistema lento, casi siendo un control P en casos extremos.

Por tanto, un buen control PI es aquel en que la temperatura se estabiliza rápido (aunque se pueda generar un error transitorio) sin llegar a generar un error transitorio.

## 2.13. autotuning

Para activar el autotuning del PID, lo haremos desde la pestaña de “Puesta en servicio”. Desde aquí, en “Modo de ajuste” seleccionaremos “Optimización inicial” y pulsaremos “Start”.



Al pulsar “Start” solo empezará el autotuning si cumplimos los siguientes requisitos:

La instrucción “PID\_Compact” se llama en un OB de alarma cíclica.

ManualEnable = FALSE y Reset = FALSE.

PID está en modo manual, automático o inactivo.

La consigna y el real se encuentran dentro de los límites configurados (apartado 2.8).

SP – Valor\_Real > 30% del fondo de escala de entrada.

SP – Valor\_Real > 50% de SP.

Si al pulsar “Start” se cumplen los requisitos, empezará el autotuning. Es importante evitar perturbaciones durante el proceso. Cuando el proceso termine, el PID escribirá en las variables de k, Ti y Td (apartado 2.7) los valores obtenidos.

Realizamos un autotuning y obtuvimos estos resultados:

k -> 22,78

Ti -> 20,64

Td -> 3,61

## 2.14. Control PID

Esta es la gráfica de la evolución de un control PID con los valores obtenidos en el apartado 2.14 y con SP = 50.

Durante este proceso, primero dejamos que la temperatura se estabilizara en 50ºC, luego activamos el ventilador para generar una perturbación y dejamos que se volviera a estabilizar a 50ºC. A continuación, cambiamos el SP a 65ºC.

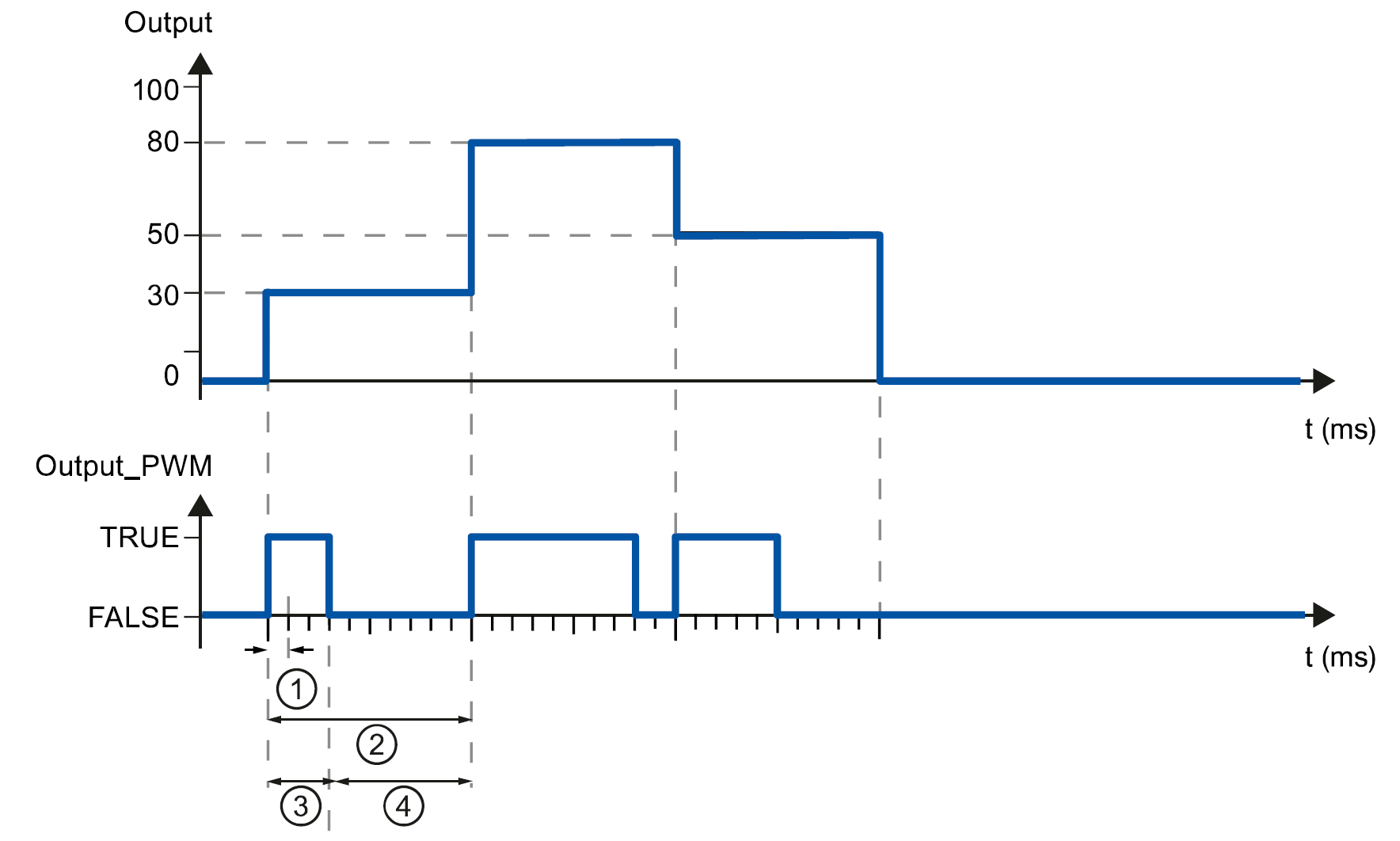
A graph with red and green lines

Description automatically generated

Se puede observar que el sistema se estabiliza en un tiempo mínimo y con un error transitorio mínimo. También se puede observar que la resistencia calefactora no consiguió calentar el horno hasta 65ºC.

## 2.15. Explicación PID con Salida PWM

Una salida PWM (Pulse Width Modulation) permite modular de 0% a 100% la salida del PID con una salida digital formando tiempos de conexión y desconexión variables. La suma de estos dos tiempos es el tiempo de ciclo o periodo de control.



(2) es el tiempo de ciclo, que es la suma del tiempo de conexión (3) y el tiempo de desconexión (4).

## 2.16. Control PID con Salida PWM

Esta es la gráfica de la evolución de un control PID con salida PWM con los valores obtenidos en el apartado 2.14 y con SP = 50.

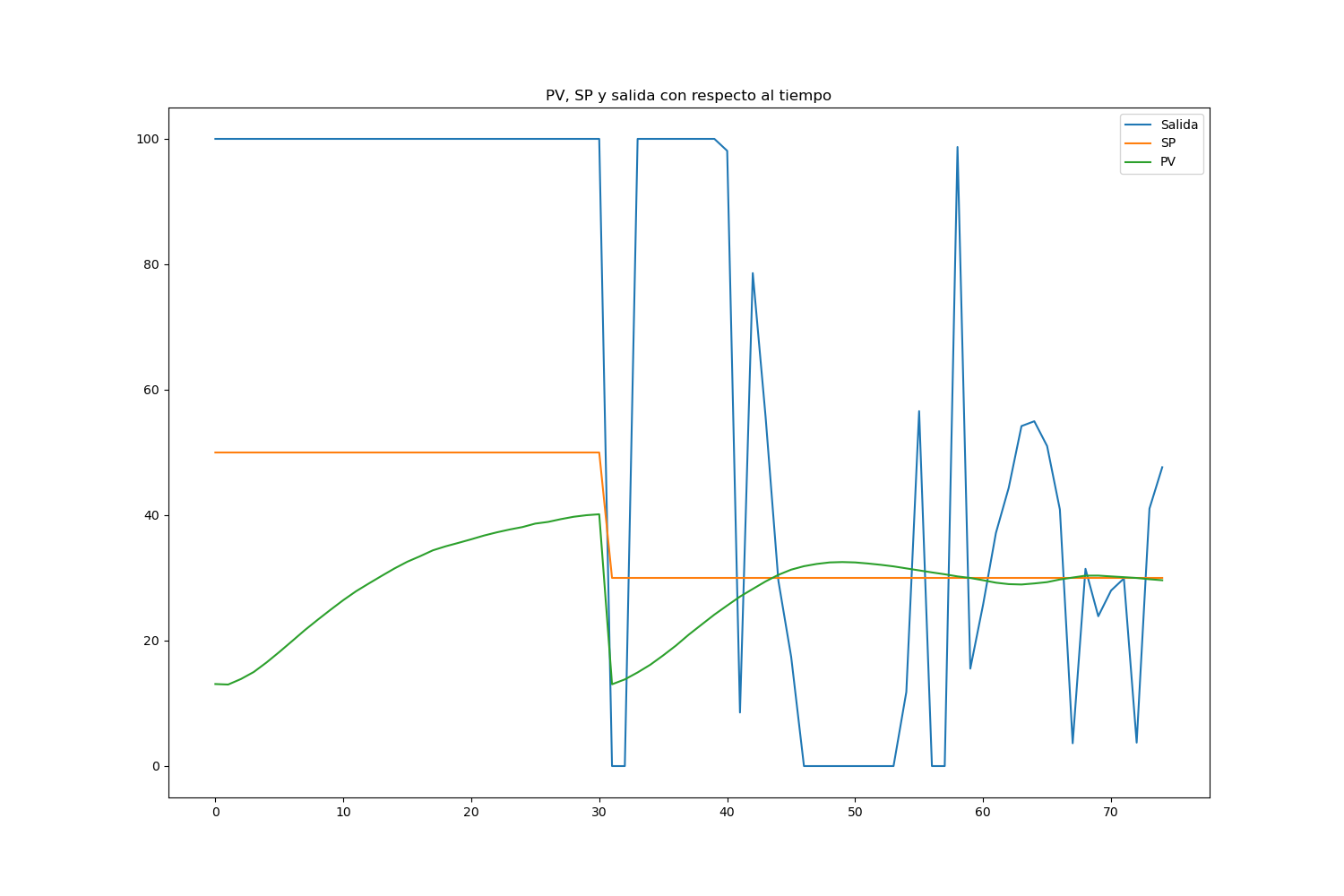
A screen shot of a graph

Description automatically generated

En verde se muestra la salida PWM, en rojo la salida analógica, en negro el SP y en azul la temperatura.

## 2.17. Datalogging

Esta es una gráfica realizada con Jupyter Notebooks a partir de los datos de un archivo generado por el programa.



## 2.18. Conclusiones y Valoración Final

Hemos podido observar la importancia de tener un sistema bien calibrado, que se estabilice rápido (aunque se genere un error transitorio) sin generar un error estacionario. Un sistema mal calibrado también puede estabilizar la magnitud a controlar en el valor deseado, pero lo hará de forma más lenta y generando errores estacionarios.

También hemos observado diferentes soluciones para controlar al actuador, que son la salida analógica (tanto la porcentual como la Output\_PER) o por PWM. Mediante estas opciones, hemos podido obtener el mismo resultado actuando sobre elementos digitales y analógicos.