



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

CONTROL DE TEMPERATURA DE UN PROCESO TÉRMICO SIMULADO EN
UNA PLATAFORMA HARDWARE-IN-THE-LOOP BASADA EN
MATLAB/SIMULINK Y COMUNICACIONES OPC UA

Trabajo de titulación previo a la obtención del
título de Ingeniero en Electrónica

AUTOR: ANTHONY JEAMPIERRE SEVILLA CACERES
PEDRO SEBASTIAN GARCIA JARAMILLO

TUTOR: ING. JULIO CESAR ZAMBRANO ABAD, PhD.

Cuenca – Ecuador

2024-2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Anthony JeanPierre Sevilla Caceres con documento de identificación N° 0704696046 y Pedro Sebastián García Jaramillo con documento de identificación N° 1104787146; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

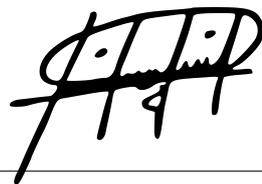
Cuenca, 06 de febrero del 2024

Atentamente,



Anthony JeanPierre Sevilla Caceres

0704696046



Pedro Sebastián García Jaramillo

1104787146

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Anthony JeamPierre Sevilla Caceres con documento de identificación N° 0704696046 y Pedro Sebastián García Jaramillo con documento de identificación N° 1104787146, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Control de temperatura de un proceso térmico simulado en una plataforma hardware-in-the-loop basada en MATLAB/Simulink y comunicaciones OPC UA” el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

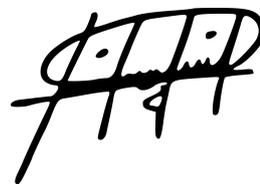
Cuenca, 06 de febrero del 2024

Atentamente,



Anthony JeamPierre Sevilla Caceres

0704696046



Pedro Sebastián García Jaramillo

1104787146

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Julio Cesar Zambrano Abad con documento de identificación N° 0301489696, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: CONTROL DE TEMPERATURA DE UN PROCESO TERMICO SIMULADO EN UNA PLATAFORMA HARDWARE-IN-THE-LOOP BASADA EN MATLAB/SIMULINK Y COMUNICACIONES OPC UA, realizado por Anthony JeanPierre Sevilla Caceres con documento de identificación N° 0704696046 y Pedro Sebastián García Jaramillo con documento de identificación N° 1104787146, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 06 de febrero del 2024

Atentamente,

**JULIO CESAR
ZAMBRANO
ABAD**

Firmado digitalmente por JULIO CESAR
ZAMBRANO ABAD
DN: cn=JULIO CESAR ZAMBRANO ABAD,
o=CEC, ou=BANCO CENTRAL DEL ECUADOR,
ou=ENTIDAD DE CERTIFICACION DE
INFORMACION ECI/CEC,
email=juliozambaroabad@gmail.com
Fecha: 2024.04.03.15:10:42 -05'00'

PhD. Julio Cesar Zambrano Abad

0301489696

AGRADECIMIENTOS

Agradecimientos de Anthony JeanPierre Sevilla Caceres

En primer lugar, quiero agradecer a mi Tutor, ING. JULIO CESAR ZAMBRANO ABAD, PhD, por su orientación experta, apoyo constante y paciencia durante todo el proceso de investigación. Sus consejos sabios y retroalimentación constructiva fueron fundamentales para dar forma a este proyecto y llevarlo a buen término. No puedo dejar de mencionar el apoyo inquebrantable de mis padres y familiares, quienes siempre estuvieron ahí para brindarme su amor, aliento y comprensión, incluso en los momentos más desafiantes. ¡Muchas gracias a todos!

Agradecimientos de Pedro Sebastián García Jaramillo

Quiero expresar mi más sincero agradecimiento a mis padres por su incondicional apoyo brindado durante toda mi vida. De igual manera agradezco a la Universidad Politécnica, a la titulación de Electrónica y automatización por permitirme culminar mi carrera Universitaria con éxito. A los docentes por los conocimientos impartidos durante la formación de mi vida profesional. En especial al Doctor Julio Zambrano, director de tesis por su apoyo y orientación durante el desarrollo de esta investigación. Por último a mis amigos y compañeros por su amistad que han estado presentes durante mis años de estudios.

DEDICATORIAS

Por Anthony JeanPierre Sevilla Caceres

A Dios, fuente de toda sabiduría y fortaleza, que me ha guiado y sostenido en cada paso de este camino académico, le dedico esta tesis como testimonio de mi gratitud y confianza en Su divina providencia. A mi amada familia, cuyo amor incondicional y apoyo inquebrantable han sido mi mayor inspiración y motivación a lo largo de esta travesía, les dedico este trabajo con profundo cariño y agradecimiento. Sus sacrificios, aliento y comprensión han sido fundamentales en mi formación académica y personal. Que este logro sea un reflejo de la fe, la perseverancia y el amor que me han sido inculcados desde mi más tierna infancia. A ustedes, mi familia querida, les dedico este esfuerzo con todo mi corazón.

Por Pedro Sebastián García Jaramillo

Quisiera dedicar este trabajo de Titulación a Dios, cuya guía y fortaleza han sido fundamentales en este camino. Además, quiero expresar un agradecimiento especial a mis padres, a mi hermano y a Emily por su incondicional apoyo, comprensión y ayuda en todos los momentos de mi vida, tanto los buenos como los difíciles. Su presencia ha sido un faro que me ha guiado en la consecución de mis metas, y sin su aliento y orientación, este trabajo no habría sido posible.

Agradezco también a aquellos seres queridos que continúan acompañándome en este viaje, así como a aquellos que ya no están físicamente conmigo, pero cuyo legado y amor perduran en mi corazón. Han sido una parte fundamental de mi vida, enseñándome el valor de la perseverancia en la búsqueda de mis sueños y la importancia de nunca rendirme. Su influencia ha sido invaluable y su recuerdo continúa inspirándome cada día.

Índice general

Agradecimientos	I
Dedicatorias	II
Índice General	IV
Índice de figuras	VII
Resumen	VIII
Abstract	IX
Antecedentes	1
Justificación	4
Objetivos	5
Introducción	6
1. Arquitectura del sistema	8
1.1. Introducción	8
1.2. Comunicación OPC UA	8
1.2.1. ¿Que es OPC UA?	8
1.3. Simulación Hardware in the loop	10
1.3.1. Hardware in the loop	10

2. Componentes de Hardware y Software	12
2.1. Modelo del proceso térmico	12
2.2. Modelo del Proceso en Matlab/Simulink	13
2.2.1. Configuración de OPC UA en controladores SIMATIC S7-1500	14
2.3. Configuración de OPC UA en MATLAB/Simulink	20
3. Implementación del controlador PID y pruebas del sistema	22
3.1. Obtención del modelo del proceso	22
3.2. Control del proceso utilizando la herramienta PID Compact de TIA Portal	29
3.3. Implementación del HMI	33
3.3.1. Pruebas del sistema	38
3.4. Pruebas y Resultados	40
4. Conclusiones y trabajos futuros	44
4.1. Conclusiones	44
4.2. Trabajos futuros	45
Glosario	46
Referencias	48

Índice de figuras

1.1. Topología de comunicación	9
2.1. Implementación del modelo a través de ecuaciones diferenciales	13
2.2. Modelo del proceso industrial y conexión opc ua.	14
2.3. Creación de proyecto en TIA Portal 15.1	15
2.4. Controlador SINAMIC S7-1500 V 2.6	16
2.5. Activación de servidor OPC UA	17
2.6. Licencia para OPC UA	17
2.7. Ajustes del Servidor OPC UA	18
2.8. Ajustes de seguridad del servidor OPC UA	18
2.9. Compilación de Software y Hardware para levantar el servidor OPC UA en el autómeta	19
2.10. Carga en el dispositivo de la configuración realizada	19
2.11. Creación del Data block en TIA Portal	20
2.12. Variables del Data block	20
2.13. Algoritmo de lectura y escritura en servidor OPC UA	21
3.1. Identificación del sistema térmico con un Polo	27
3.2. Identificación del sistema térmico con dos Polos	28
3.3. Identificación del sistema térmico con tres Polos	29
3.4. bloque cycling interrupt	30
3.5. Ubicación de la herramienta PID Compact	30
3.6. Parámetros pid compact	31
3.7. Bloque de PID Compact	31
3.8. Gráfica de la herramienta tunning mode	33
3.9. Gráfica de la herramienta Tracer	33

3.10. Agregar nuevo dispositivo a de proyecto Tia Portal	34
3.11. Selección de dispositivo HMI	35
3.12. Comunicación Profinet de plc con HMI	35
3.13. Imagen Raíz	36
3.14. Carátula en imagen raíz	36
3.15. Gráfica de la entrada de voltaje que ingresa al proceso térmico	37
3.16. gráfica del control PID del proceso térmico	37
3.17. gráfica del control PID del proceso térmico	38
3.18. gráfica de entrada de voltaje al sistema	39
3.19. Gráfica de respuesta del sistema con diferentes valores de voltaje	40
3.20. Gráfica de respuesta del sistema con diferentes valores de voltaje en Matlab	41
3.21. Gráfica de respuesta del sistema con valores de setpoint	41
3.22. Gráfica de respuesta del sistema con valores de setpoint en Matlab	42
3.23. Gráfica de voltaje de ingreso al sistema Tia Portal	43
3.24. Gráfica de voltaje de ingreso al sistema Matlab	43

Resumen

Este trabajo de titulación se centra en la integración entre MATLAB Simulink y el PLC-S7-1500 mediante OPC UA para controlar un proceso térmico. Esta conexión bidireccional permite una comunicación segura y confiable, esencial en entornos industriales. En el entorno de Hardware-in-the-Loop de MATLAB Simulink, se simula el proceso térmico para probar y validar un controlador PID Compact diseñado en TIA Portal. La arquitectura incluye servidores OPC UA en ambas plataformas y un cliente OPC UA para gestionar la comunicación. La implementación de estándares abiertos como OPC UA facilita la interoperabilidad y la conectividad entre sistemas heterogéneos, mejorando la eficiencia y la confiabilidad de los procesos industriales. Este enfoque integral ofrece una solución escalable para el control de procesos industriales avanzados, alineado con las demandas crecientes de la automatización industrial moderna.

Palabras clave: MATLAB Simulink; PLC-S7-1500; OPC UA ; Hardware-in-the-Loop; Controlador PID Compact; TIA Portal.

Abstract

This thesis focuses on the integration between MATLAB Simulink and the PLC-S7-1500 via OPC UA to control a thermal process. This bi-directional connection enables secure and reliable communication, essential in industrial environments. In the MATLAB Simulink Hardware-in-the-Loop environment, the thermal process is simulated to test and validate a PID Compact Controller designed in TIA Portal. The architecture includes OPC UA servers on both platforms and an OPC UA client to manage the communication. The implementation of open standards such as OPC UA facilitates interoperability and connectivity between heterogeneous systems, improving the efficiency and reliability of industrial processes. This comprehensive approach offers a scalable solution for advanced industrial process control, aligned with the growing demands of modern industrial automation.

Keywords: MATLAB Simulink; PLC-S7-1500; OPC U ; Hardware-in-the-Loop; PID Compact Controller; TIA Portal.

Antecedentes

Los controladores lógicos programables (PLC) revolucionaron la industria de la automatización en la década de 1960 y 1970. Hoy en día, los PLC son la base de cualquier proceso de automatización. Su evolución ha sido impulsada por la necesidad de mejorar la eficiencia, la flexibilidad y la confiabilidad en los entornos de producción. En estos dispositivos existe la posibilidad de comunicación o interacción entre ellos y a su vez con varios elementos industriales ya sea como sensores y actuadores, el uso de estos dispositivos lo podemos encontrar en muchas áreas como por ejemplo en la automatización de procesos los cuales permiten automatizar una amplia gama de procesos industriales, desde la fabricación de productos hasta el control de maquinaria y sistemas de gestión de edificios. Esto reduce la necesidad de intervención humana y mejora la eficiencia, a menudo se integran con sistemas de supervisión y control (SCADA) para permitir la monitorización en tiempo real de los procesos industriales. Esto proporciona información crítica para la toma de decisiones y la detección temprana de problemas además de llevar a una reducción de costos a largo plazo al mejorar la eficiencia, reducir los errores humanos y minimizar el desperdicio de recursos. Los controladores PID (Proporcional, Integral y Derivativo) son algoritmos utilizados para el control de procesos en la industria. Estos controladores se pueden implementar en PLCs (Controladores Lógicos Programables) para regular variables como temperatura, presión, velocidad, nivel, y otras, manteniéndolas dentro de un rango deseado, una de las ventajas de usar un PLC para implementar un controlador PID es que se puede programar de manera flexible y adaptarse a diferentes aplicaciones y requisitos de control. Además, los PLCs a menudo se integran con sistemas de supervisión y control (SCADA) para permitir un control y monitoreo centralizado de múltiples procesos.

Esto es fundamental en la industria para garantizar un control preciso y eficiente de los procesos.

Las comunicaciones industriales desempeñan un papel crucial en el funcionamiento de los Controladores Lógicos Programables (PLCs) en entornos industriales. Actualmente, existen diversas tecnologías de comunicación para la industria, como por ejemplo, Profibus, Profinet, Modbus, entre otras. A nivel industrial, las redes cableadas no siempre son factibles, debido a diversos factores, hoy en día existen diversos protocolos de comunicación industrial como el estándar de comunicación OPC (OLE for Process Control) se desarrolló en respuesta a la necesidad de conectar sistemas de control industrial, como sistemas SCADA (Supervisory Control and Data Acquisition) y controladores de automatización, con dispositivos y equipos de diferentes fabricantes, en este protocolo existen varias versiones como, OPC DA, OPC AE, OPC HDA, OPC DX, OPC CML DA y su más reciente Object Linking and Embedding for Process Control Unified Architecture" (OPC UA), Con este trabajo de titulación se pretende realizar un sistema "hardware-in-the-loop" basado en MATLAB/Simulink y OPC UA para la evaluación de controladores PID implementados sobre controladores lógicos programables. Este protocolo de comunicación proporciona una plataforma independiente que permite la comunicación entre dispositivos y sistemas de diferentes fabricantes y plataformas. Esto facilita la integración de equipos y sistemas heterogéneos en un entorno industrial además de haber sido diseñado para ofrecer un alto rendimiento en términos de velocidad de comunicación y eficiencia en el uso de recursos de red, lo que es esencial para aplicaciones en tiempo real y sistemas críticos, otro gran beneficio de esta es proporcionar interoperabilidad y facilitar la integración de sistemas, OPC UA puede ayudar a reducir los costos asociados con la adquisición, implementación y mantenimiento de sistemas de control y automatización industrial.

Hoy en día la simulación es una herramienta poderosa que se utiliza en una variedad de campos para imitar y estudiar el comportamiento de sistemas o procesos en un entorno controlado y virtual, sus beneficios y usos son diversos y abarcan desde la investigación científica hasta la toma de decisiones en la industria gracias a esto permite probar y analizar situaciones de riesgo sin consecuencias

reales. Esto es especialmente valioso en campos como la aviación, la medicina y la ingeniería, donde los errores pueden ser costosos o peligrosos, en la industria, las simulaciones se utilizan para modelar y optimizar procesos de fabricación, logística, cadena de suministro, etc. Con el paso de los años se desarrollo la herramienta de simulación Hardware-in-the-Loop (HIL) basada en Matlab simulink , esta es una técnica utilizada en ingeniería y desarrollo de sistemas para probar y validar hardware, especialmente sistemas electrónicos y de control, en un entorno simulado o virtual antes de implementarlos en el mundo real.

Justificación

Este trabajo de titulación se enfoca en la implementación de un sistema Hardware in the Loop como una herramienta de simulación para evaluar estrategias de control en un Controlador Lógico Programable (PLC) real. Esto supone una contribución significativa al entorno académico de la institución, que se encuentra limitada en términos de recursos, específicamente en la falta de plantas de control físicas debido a restricciones presupuestarias y limitaciones de espacio.

La solución propuesta implica la utilización de un PLC como controlador para supervisar un proceso simulado en un entorno de hardware in the loop (HIL) basado en MATLAB Simulink. Este enfoque brinda la capacidad de evaluar y refinar estrategias de control en un entorno controlado y seguro, al tiempo que proporciona a los estudiantes herramientas prácticas para poner a prueba y perfeccionar sus conocimientos en esquemas de control.

Además, se busca aprovechar el estándar de comunicación OPC UA, que es independiente de sistemas operativos y permite una interconexión sin restricciones significativas entre dispositivos. La novedad radica en la integración de OPC UA con el PLC, una iniciativa que se ha explorado de manera limitada a nivel local. Esto facilitaría una comunicación eficiente entre la simulación y el PLC, mejorando la versatilidad y la funcionalidad del sistema.

En resumen, este enfoque responde a la necesidad de abordar los desafíos asociados con la manipulación de procesos térmicos reales, que pueden ser costosos y potencialmente peligrosos. La simulación HIL proporciona a ingenieros y operadores la capacidad de experimentar y optimizar estrategias de control de manera segura, sin exponerse a riesgos ni incurrir en gastos adicionales relacionados con la producción.

Objetivos

Objetivo General

-

Objetivos específicos:

- Construir el modelo matemático del proceso térmico a controlar dentro de la herramienta Simulink Matlab.
- Implementar un sistema de comunicación basado en OPC UA para interconectar el PLC real y el proceso industrial simulado.
- Diseñar e implementar el controlador PID dentro del PLC(S7-1500) para regular la temperatura del proceso térmico simulado.
- Evaluar el desempeño del controlador mediante pruebas de laboratorio registradas a través de un sistema de monitoreo local.

Introducción

En la era actual de la automatización y la digitalización, el control preciso de procesos térmicos es crucial para garantizar la eficiencia operativa y la calidad del producto en una amplia gama de industrias, desde la manufactura hasta la producción de energía. En este contexto, la utilización de plataformas hardware-in-the-loop (HIL) basadas en MATLAB/Simulink y comunicaciones OPC UA se ha convertido en una herramienta fundamental para el diseño, desarrollo y prueba de sistemas de control avanzados.

Ante la creciente demanda de sistemas de control más eficientes y adaptables, surge la necesidad de explorar nuevas metodologías y tecnologías que permitan abordar los desafíos actuales de manera efectiva. En este sentido, el capítulo 1 introduce la arquitectura del sistema, que combina la simulación de procesos térmicos en MATLAB/Simulink con la comunicación mediante el estándar OPC UA. La comunicación OPC UA permite la integración de sistemas heterogéneos y la interoperabilidad entre dispositivos y software de diferentes fabricantes, lo que facilita el desarrollo de sistemas de control distribuido y la supervisión remota de procesos industriales.

La aplicación de esta arquitectura se presenta en el capítulo 2, donde se detallan los componentes de hardware y software utilizados en el sistema. Se describen minuciosamente los detalles del proceso térmico simulado, abarcando desde los elementos físicos involucrados hasta las variables de control. Además, se profundiza en el modelo del proceso industrial, explorando su complejidad y sus características dinámicas, lo que permite comprender mejor su comportamiento y diseñar estrategias de control efectivas. Asimismo, se examina con detalle el modelo implementado en MATLAB/Simulink, destacando su capacidad para simular

y analizar el comportamiento del sistema en diferentes condiciones operativas. Por último, se explora en profundidad cómo se integra OPC UA en los controladores SIMATIC S7-1500, incluyendo aspectos técnicos como la configuración de los nodos de comunicación y la gestión de los datos intercambiados entre los diferentes dispositivos y software involucrados en el sistema.

La implementación del controlador PID y las pruebas del sistema en tiempo real se abordan en el capítulo 3. Aquí, se detalla el proceso de obtención del modelo del proceso, que incluye la recolección de datos experimentales y su análisis para identificar los parámetros del sistema. Posteriormente, se describe en detalle la configuración del controlador PID utilizando la herramienta TIA Portal, destacando los criterios de sintonización utilizados y su impacto en el rendimiento del controlador. Además, se aborda la implementación de la interfaz hombre-máquina (HMI), que permite a los operadores supervisar el proceso en tiempo real y realizar ajustes en el controlador según sea necesario. Por último, se llevan a cabo pruebas exhaustivas del sistema, evaluando su respuesta bajo diferentes condiciones de operación y validando la eficacia del controlador PID en la regulación de la temperatura del proceso.

En el capítulo 4, se presentan las conclusiones a partir de las pruebas realizadas y se discuten posibles áreas de mejora y futuras investigaciones. Se destacan las ventajas y desafíos asociados con la aplicación de la tecnología HIL y OPC UA en el control de procesos térmicos, así como las implicaciones prácticas para la industria. Además, se analiza críticamente el rendimiento del sistema implementado, identificando sus fortalezas y debilidades, y se proponen posibles enfoques para optimizar su desempeño en términos de precisión, robustez y eficiencia energética. Por último, se delinear líneas de investigación futuras, como la aplicación de técnicas de control avanzadas o la integración de sistemas de supervisión y diagnóstico basados en inteligencia artificial, que podrían contribuir significativamente a mejorar la automatización y la eficiencia de los procesos térmicos en la industria.

Capítulo 1

Arquitectura del sistema

1.1. Introducción

La Figura 1.1 muestra una representación visual de la arquitectura del sistema que interconecta un PLC Siemens S7-1500 y MATLAB Simulink, mediante el protocolo OPC UA. En la parte izquierda de la imagen se encuentra el PLC S7-1500, que actúa como el controlador principal del sistema. Este PLC está conectado a través de un switch X2008 a la PC que contiene MATLAB Simulink y que es donde se implementará el proceso a controlar.

La conexión OPC UA permite la comunicación bidireccional entre el PLC y MATLAB Simulink, facilitando la transferencia de datos en tiempo real y el control del proceso desde el entorno de Simulink. Como complemento, en la parte inferior de la imagen se muestra otra conexión, esta vez utilizando el protocolo PROFINET. Esta conexión está establecida entre el PLC y un HMI (Interfaz Hombre-Máquina). El HMI actúa como una interfaz gráfica para el monitoreo y puesta en marcha del sistema.

1.2. Comunicación OPC UA

1.2.1. ¿Que es OPC UA?

OPC UA es el estándar sucesor de OPC Foundation y representa la próxima generación en comunicaciones industriales. A diferencia de OPC clásico, que es bien

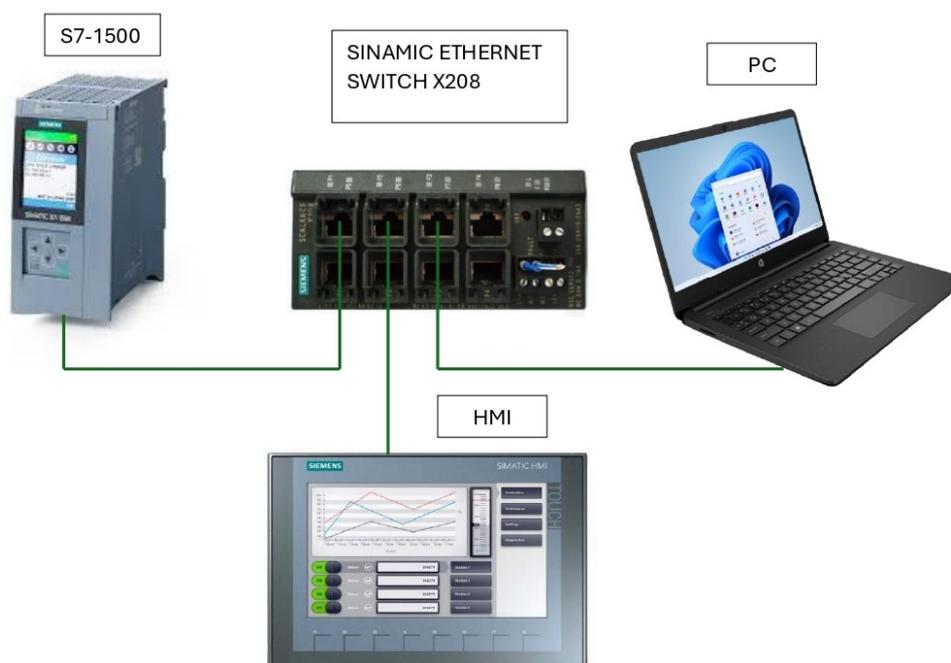


Figura 1.1: Topología de comunicación

conocido en la industria y proporciona una interfaz estándar para la comunicación con Controladores Lógicos Programables, el objetivo de **OPC UA** es expandir esta compatibilidad a dispositivos y empresas. Este protocolo se basa en un modelo cliente/servidor que permite a los clientes conectarse, navegar, leer y escribir en equipos industriales [1]. Lo que distingue a **OPC UA** es su capacidad para definir la comunicación desde la aplicación hasta la capa de transporte, lo que garantiza una alta compatibilidad entre diferentes fabricantes. Además, se destaca por su enfoque en la seguridad, utilizando mensajes firmados y cifrado de transporte bidireccional. [2].

OPC UA proporciona una amplia gama de funcionalidades, lo que lo posiciona como un estándar de comunicación de datos en entornos industriales. Algunas de las características principales de **OPC UA** son:

- Ofrece un conjunto integral de características de seguridad que abarcan autenticación, autorización, confidencialidad e integridad de datos.
- Proporciona un modelo de información flexible y adaptable que permite la definición de tipos de datos personalizados y la representación de información

específica del dominio en aplicaciones industriales. [3].

- Es independiente de la plataforma y se puede implementar en una variedad de sistemas operativos y arquitecturas de hardware, como Windows, Linux, sistemas embebidos y dispositivos móviles. [4].

1.3. Simulación Hardware in the loop

1.3.1. Hardware in the loop

El concepto de "Hardware in the Loop"(HIL) se utiliza como un concepto de simulación donde se emplea hardware físico real para interactuar con modelos de simulación computacional. En el contexto de MATLAB y Simulink, el HIL implica la conexión de un sistema físico o un componente real, como un controlador o un actuador, a un modelo de Simulink que representa el entorno en el que opera dicho sistema [5]. En otras palabras, en un sistema HIL, Simulink se utiliza para simular el comportamiento del sistema completo, mientras que el hardware físico real se utiliza para implementar el controlador.

Cuando se aplica el enfoque de HIL al diseño y control de un horno industrial en MATLAB Simulink, el modelo matemático del horno dentro de Simulink simula el comportamiento térmico del sistema. Este modelo incluye ecuaciones diferenciales que describen cómo la temperatura del horno cambia con el tiempo, considerando parámetros como la potencia del calentador y la capacidad térmica del material [6]. La integración del hardware físico real, como los controladores de temperatura y los actuadores de potencia, permite que el modelo en Simulink interactúe directamente con el sistema físico del horno. Por ejemplo, el modelo puede enviar señales al controlador de temperatura físico, que a su vez ajusta la potencia del calentador del horno para mantener la temperatura deseada. Esta integración de hardware real en la simulación permite probar y validar el controlador en un entorno lo más cercano posible a las condiciones reales de operación antes de su implementación final. Esto ayuda a identificar posibles problemas y optimizar el diseño del controlador de manera eficiente y segura, reduciendo así los costos y el tiempo asociado con el

desarrollo y la depuración de sistemas de control en el mundo real [7].

Para este caso específico, el proceso térmico implementado en Simulink será controlado por un PLC S7-1500. Se plantea la implementación de un controlador PID sobre el PLC utilizando la herramienta PID Compact de TIA Portal. A través del HMI se podrá ingresar el punto de ajuste y se podrá visualizar el desempeño del controlador reflejado en la variable controlada y la variable manipulada.

Capítulo 2

Componentes de Hardware y Software

2.1. Modelo del proceso térmico

Las ecuaciones diferenciales (2.1), (2.2), (2.3) describen el comportamiento del proceso térmico que va a ser controlado. Específicamente se trata de un horno industrial con las siguientes características:

- resistencia eléctrica $R = 10$ ohmios,
- capacidad térmica $C = 372$ faradios,
- diferencia de potencial $V = 220$ voltios,
- área superficial $A = 1$ metros cuadrado,
- coeficiente de transferencia de calor por convección $h_{conv} = 10$,
- temperatura ambiente $T_{amb} = 25$ °C,
- constante de Stefan-Boltzmann $\sigma = 5,67 \times 10^{-8}$,
- emisividad $\epsilon = 1$.

$$RC \frac{dT}{dt} = \frac{V^2}{R} - Ah(T - T_{amb}) - \sigma A(T^4 - T_{amb}^4) \quad (2.1)$$

$$\frac{dT}{dt} = \frac{V^2}{R^2C} - \frac{A \cdot h}{RC}(T - T_{amb}) - \frac{\sigma \cdot A}{RC}(T - T_{amb}) \quad (2.2)$$

$$\frac{dT}{dt} = \frac{V^2}{R^2C} - \frac{A \cdot h \cdot T}{RC} + \frac{A \cdot h \cdot Tamb}{RC} - \frac{\sigma \cdot A \cdot T^4}{RC} + \frac{\sigma \cdot A \cdot Tamb^4}{RC} \quad (2.3)$$

2.2. Modelo del Proceso en Matlab/Simulink

Las ecuaciones y los parámetros descritos en la sección anterior son utilizados para construir el modelo en MATLAB Simulink. La implementación de la ecuación diferencial correspondiente se muestra en la Figura 2.1

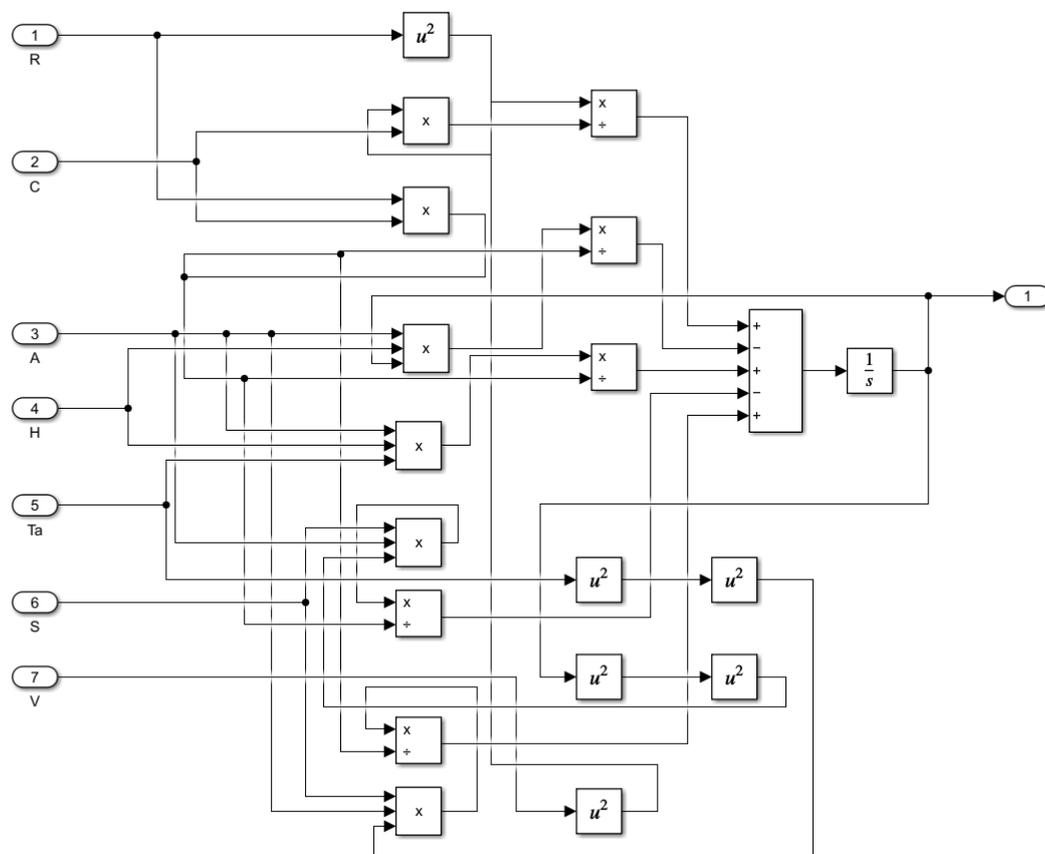


Figura 2.1: Implementación del modelo a través de ecuaciones diferenciales

El modelo completo se muestra en la Figura 2.2. A la izquierda se asignan los valores de las variables que ya fueron explicadas con anterioridad en la Sección 2.1. El Bloque que se denomina Interpreted MATLAB Function, contiene un código de MATLAB que se ejecuta dentro del entorno de Simulink. Dentro de este Bloque,

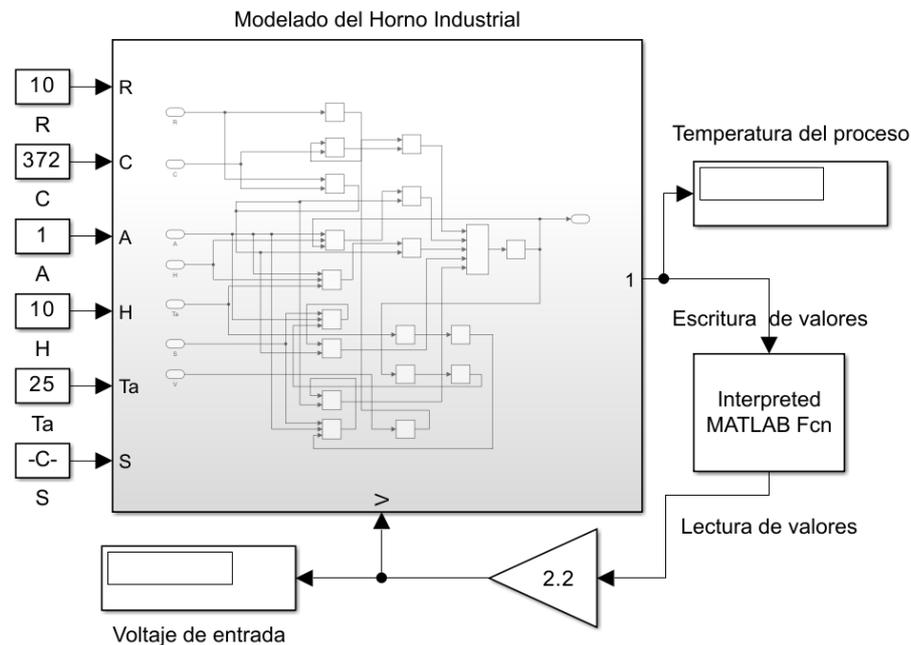


Figura 2.2: Modelo del proceso industrial y conexión opc ua.

se encuentra el código de conexión a un PLC (Controlador Lógico Programable) utilizando la librería OPC UA, la cual permite la comunicación bidireccional entre Simulink y el PLC. La comunicación bidireccional entre Simulink y el PLC es importante en este contexto, ya que permite que el modelo de simulación del horno industrial que se muestra en la Figura 2.1 lea los valores actuales del proceso (como la temperatura y el voltaje de entrada) desde el PLC y también escriba valores de control (como la temperatura deseada y el estado del horno) en el PLC.

2.2.1. Configuración de OPC UA en controladores SIMATIC S7-1500

OPC UA es ampliamente usado en el ámbito industrial debido a la fácil comunicación e intercambio de datos. El controlador Siemens SINAMIC S7-1500 está equipado con un servidor OPC UA y, en versiones más recientes, con un cliente OPC UA. Este servidor admite diferentes sistemas (llamados clientes), estos

pueden ingresar toda la información. Estos tienen la capacidad de leer y escribir datos en el servidor, proporcionando acceso en línea a la información del servidor, incluidos datos de rendimiento y diagnóstico. La configuración del servidor en el PLC SINAMIC S7-1500 incluye la conformación de los datos y métodos disponibles para el cliente. Además, el controlador SINAMIC S7-1500 también puede ser tomado como cliente, de modo que es capaz de leer datos de otros servidores OPC UA. Se cuenta con una herramienta para la programación de estos llamada (TIA Portal) para simplificar la creación de programas que actúan como clientes, incluido un editor de interfaz de cliente y configuración para crear las conexiones OPC UA necesarias. La implementación de esta comienza con la configuración del servidor OPC UA mediante la herramienta TIA Portal 15.1, que brinda una interfaz intuitiva y equipada. Durante la configuración, es importante realizar varios pasos para asegurar el correcto funcionamiento del servidor. Estos pasos incluyen la escritura de los parámetros de comunicación y la atribución de variables. Antes de iniciar la configuración del servidor, debe crear un nuevo proyecto en el TIA Portal del controlador. (véase la Figura 2.3).

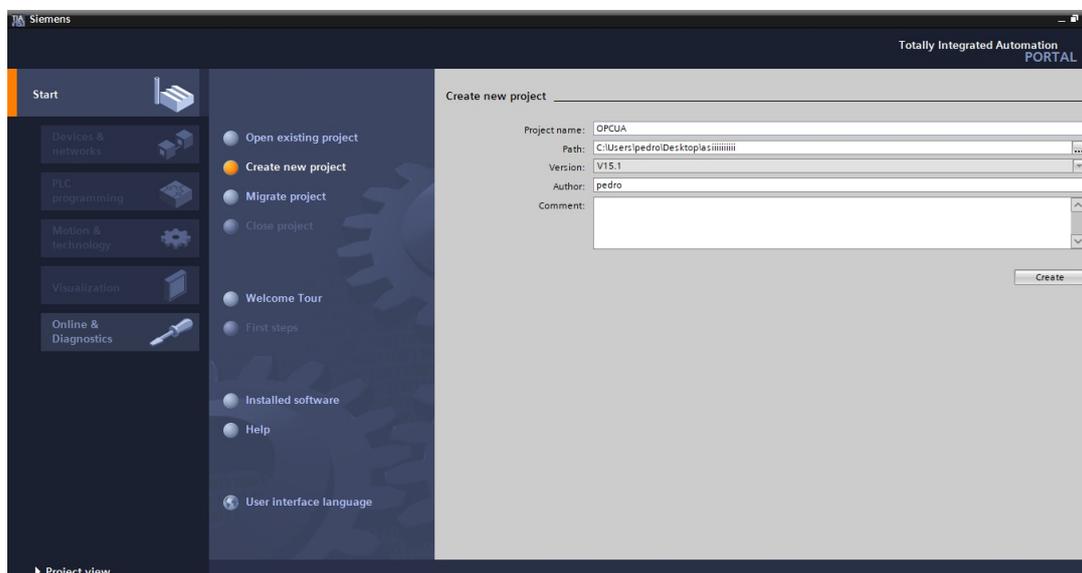


Figura 2.3: Creación de proyecto en TIA Portal 15.1

Además, se selecciona un controlador SINAMIC S7-1500 que tenga un firmware de versión 2.6 o superior. Estos requerimientos son esenciales para garantizar una adecuada configuración y una compatibilidad con el servidor. La

selección de de este controlador se evidencia en la Figura 2.4.

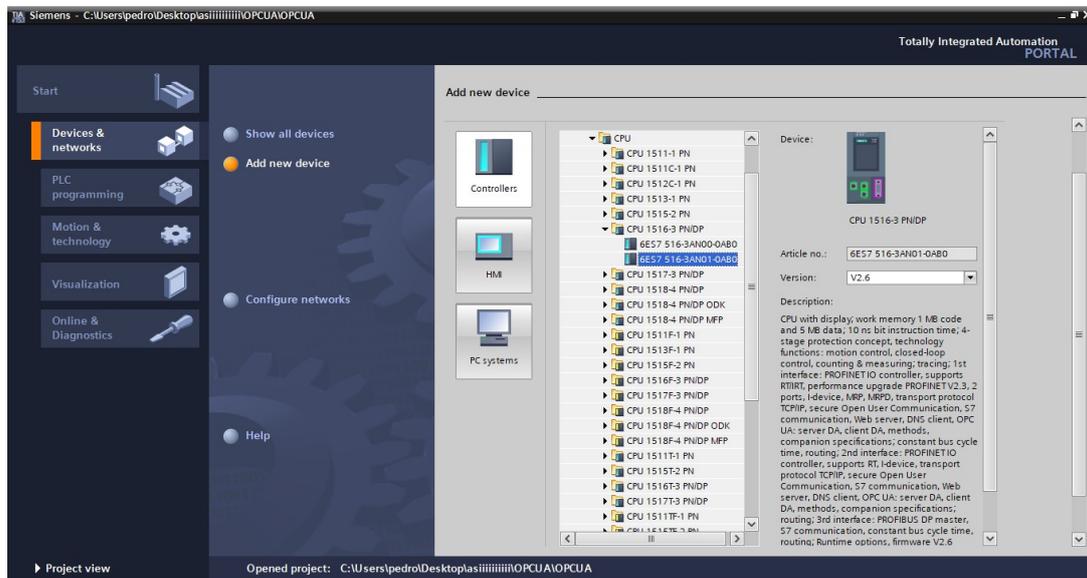


Figura 2.4: Controlador SINAMIC S7-1500 V 2.6

Por defecto, el servidor **OPC UA** del controlador SINAMIC S7-1500 esta desactivado. Seguidamente se muestra los pasos para activar el servidor en el controlador.

- Se accede a las Propiedades de la CPU S7-1500 configurada dentro de TIA Portal. Seguido, en la sección **OPC UA** >Servidor >General, en la ventana de Accesibilidad del servidor se debe marcar la opción Activar **OPC UA** para que quede activado el servidor. Este proceso se puede visualizar en la Figura 2.5. Además, se debe tener en cuenta las Direcciones del servidor. En este caso, la interfaz X1 tiene la dirección 192.168.0.1, mientras que, la interfaz X2 tiene la dirección 192.168.1.1. En el caso que se desee utilizar la interfaz X1 de el PLC, la dirección del servidor sería `opc.tcp://192.168.0.1:4840`.
- En el apartado de propiedades de la CPU S7-1500, es necesario posicionarse en la ventana del inspector de navegación hasta Licencias runtime>OPCUA. Aquí se selecciona la licencia necesaria del servidor. Este procedimiento se puede evidenciar en la Figura 2.6.
- En el apartado de inspección, se selecciona **OPC UA** >Servidor>Ajustes y se selecciona puerto que se quiere utilizar . Se asigna de igual manera al servidor

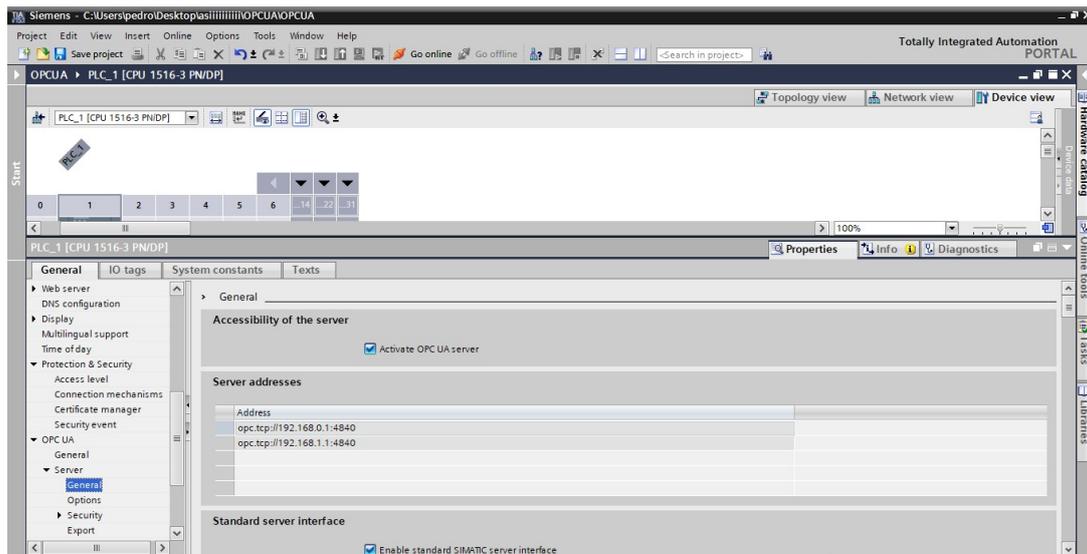


Figura 2.5: Activacion de servidor OPC UA

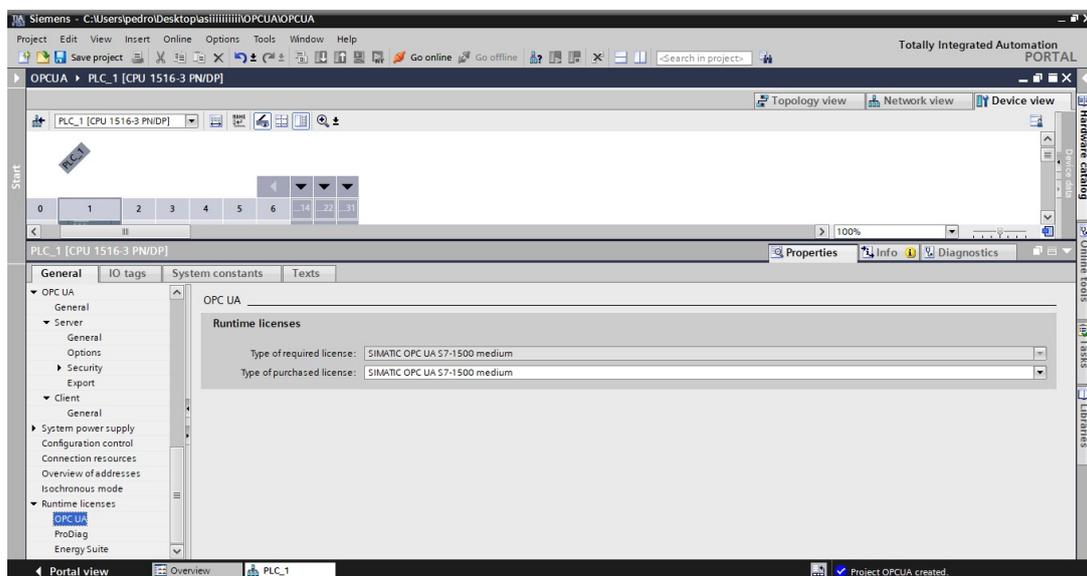


Figura 2.6: Licencia para OPC UA

un intervalo de muestreo y un intervalo de envío. Esto se puede observar en la Figura 2.7.

- En la ventana de inspección, se posiciona **OPC UA**>Security>Secure Channel. A continuación, en Security Policys se asigna Ninguna seguridad. Este proceso se puede observar en la Figura 2.8

Una vez realizados los ajustes que se indicaron anteriormente, se procede a realizar la carga del programa en el Plc, para que todas las funciones del servidor estén activas.

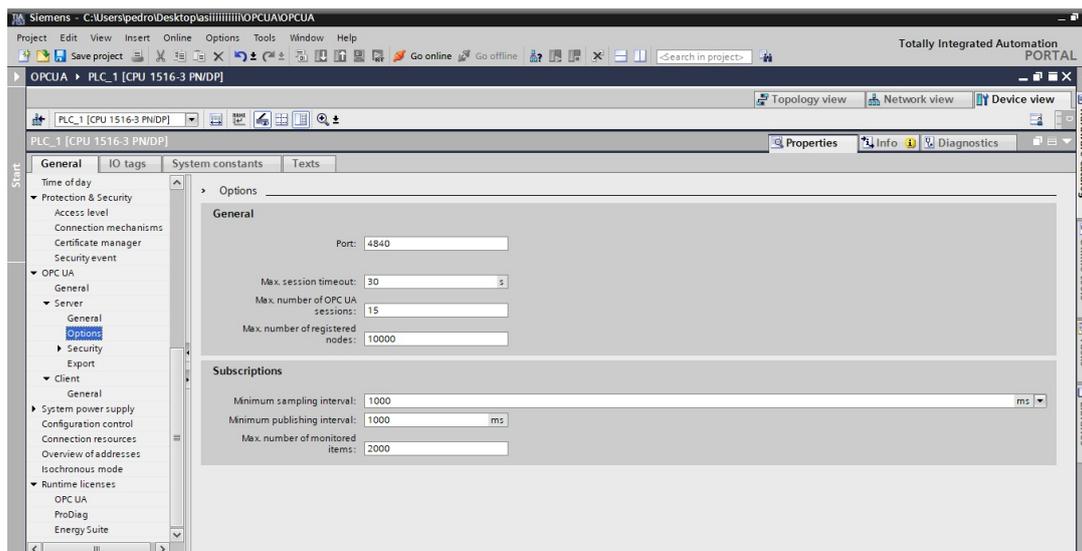


Figura 2.7: Ajustes del Servidor OPC UA

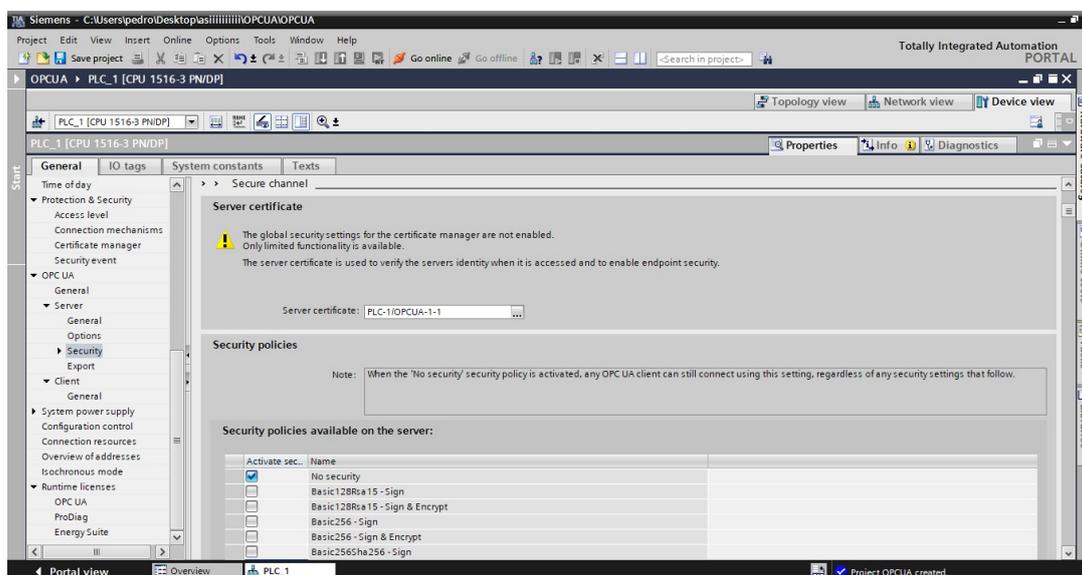


Figura 2.8: Ajustes del seguridad del servidor OPC UA

- En la ventana de Árbol de proyecto, se selecciona el PLC que es llamado OPC-UA y se procede a compilar Software y Hardware, como se muestra en la figura 2.9.
- Nuevamente en el Árbol de proyecto, se selecciona el autómata que tiene como nombre OPC-UA y se procede a cargaren el dispositivo Software y Hardware, como se muestra en la Figura 2.10.

Finalmente, el autómata está configurado y operando como Servidor . Ahora, cualquier variable que se cree en el servidor puede ser leída o controlada a través

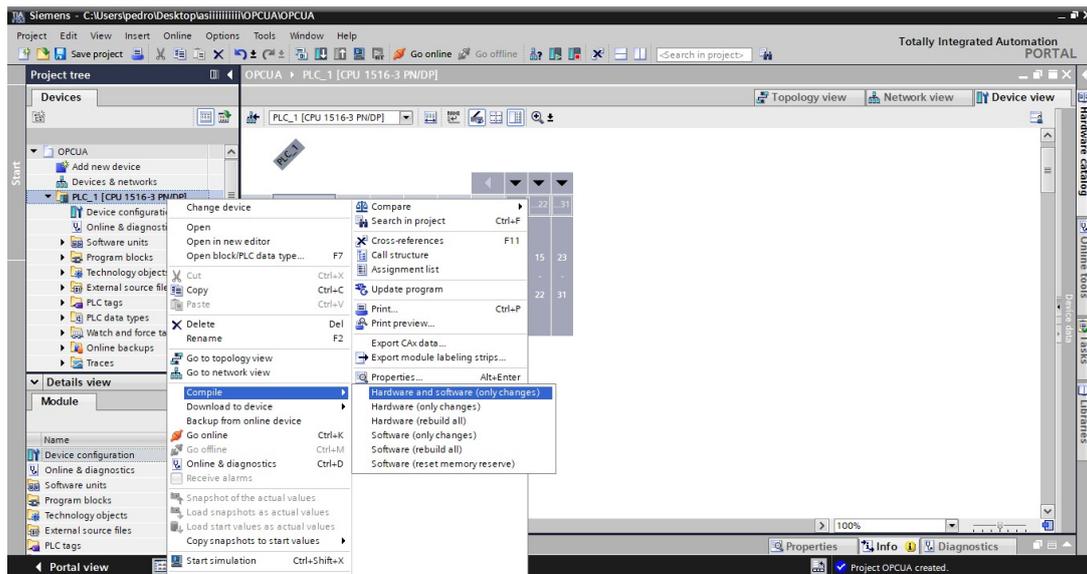


Figura 2.9: Compilación de Software y Hardware para levantar el servidor OPC UA en el autómeta

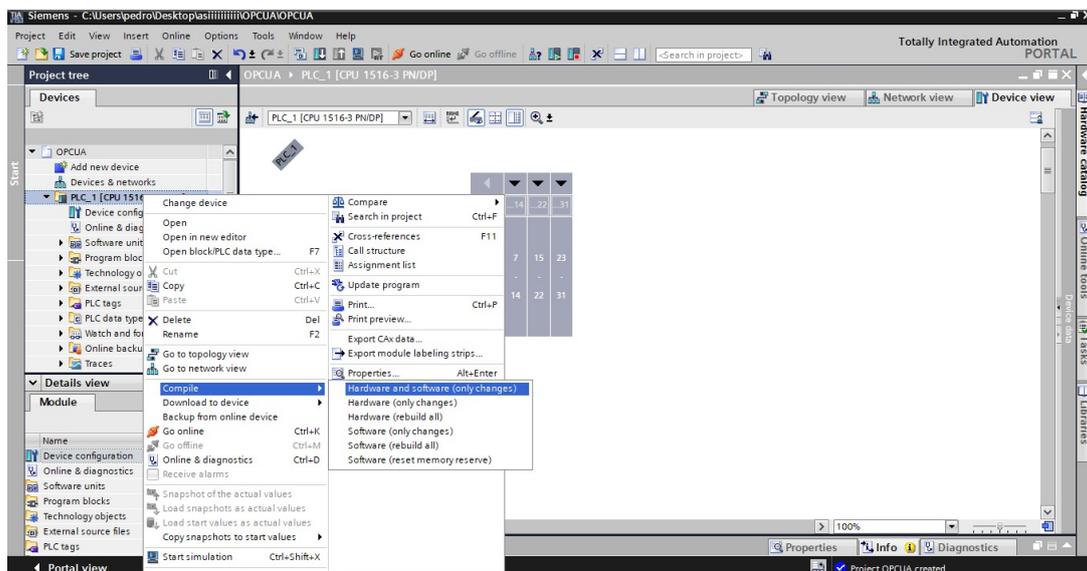


Figura 2.10: Carga en el dispositivo de la configuración realizada

de un cliente.

Culminada la configuración se crea las variables necesarias para el proceso térmico, Para esto se debe acceder al Árbol de proyecto, aquí se debe seleccionar Bloques de programa y agrega un nuevo bloque, se elige un Data block, el cual se llamará OPC. Este proceso se ilustra en la Figura 2.11.

Las variables creadas son intuitivas, estas variables son utilizadas para interactuar con la información del proceso industrial, como por ejemplo: voltaje y temperatura. Esto se puede evidenciar en la Figura 2.12.

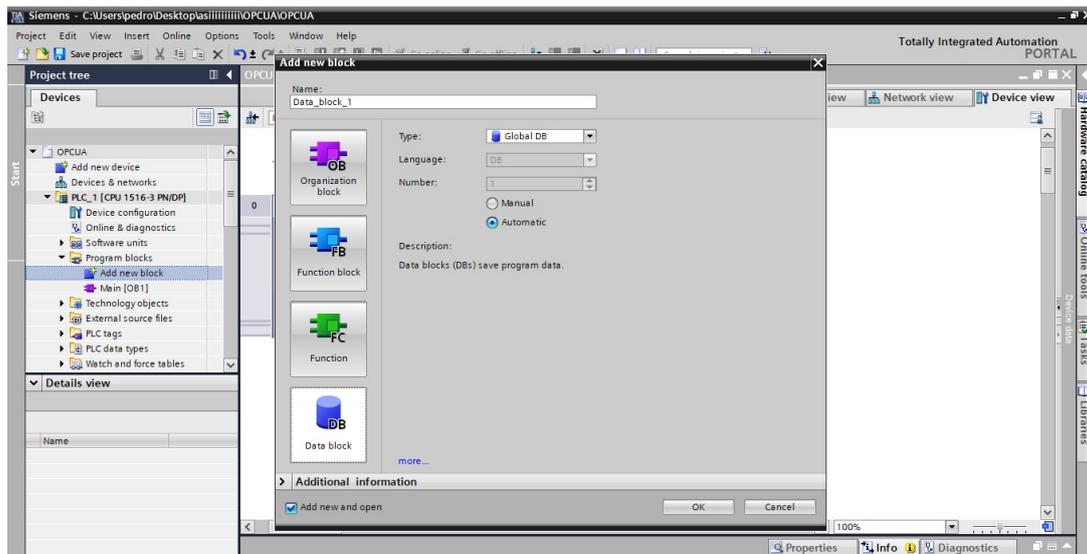


Figura 2.11: Creación del Data block en TIA Portal

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
Static									
voltaje	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
temperatura	Real	25.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
setpoint	Real	25.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
input	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
modoactivate	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
modo	Int	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
manualvalue	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
output	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Figura 2.12: Variables del Data block

2.3. Configuración de OPC UA en MATLAB/Simulink

Una vez realizada la configuración del servidor OPC UA en el PLC SIMATIC S7-1500, se realiza la programación del cliente en Matlab. Para esto, se crea una función Read-OPC-Func2.m. Aquí se programa la función encargada de comunicarse con el servidor OPC UA.

Las líneas necesarias para poder leer o modificar los valores del servidor son las siguientes: `writeValue(uaClient,Var Node OUT,y)` y `readValue(uaClient,Var Node in)`. Estas líneas modifican y leen los valores del servidor.

Algoritmo 1 Lectura y escritura en un servidor OPC UA

Data: Valor de entrada y **Result:** Valor leído y escrito en el servidor OPC UA**Function** *Read OPC Func2(y)*

```

// variables persistentes
if isempty(init_Server) then
| testVal ← 0 init_Server ← 0 init_Nodes ← 0
end
// inicializar el servidor OPC UA
if init_Server == 0 then
| init_Server ← 1 uaClient = opcua('192.168.0.1',4840) connect(uaClient)
end
// definir los nodos de las variables en el servidor
if uaClient.isConnected == 1 and init_Nodes == 0 then
| init_Nodes ← 1 DB_Node = findNodeByName(uaClient.Namespace,'OPC',...
| '-once')
| Var_Node_in = findNodeByName(DB_Node,'voltaje', '-once')
| Var_Node_Out = findNodeByName(DB_Node,'temperatura', '-once')
end
// leer y escribir en el servidor OPC UA
if uaClient.isConnected == 1 and init_Nodes == 1 then
| [val, , ] = readValue(uaClient, Var_Node_in) writeValue(uaClient,
| Var_Node_Out, y) testVal = val
end
x ← double(testVal) return x

```

Figura 2.13: Algoritmo de lectura y escritura en servidor OPC UA

Capítulo 3

Implementación del controlador PID y pruebas del sistema

3.1. Obtención del modelo del proceso

El proceso de identificación del modelo del sistema es esencial para comprender y caracterizar su comportamiento dinámico. En este contexto, la herramienta System Identification Toolbox de MATLAB desempeña un papel crucial al permitir estimar modelos de sistemas dinámicos a partir de datos experimentales. En el código Proporcionado en la parte inferior, se utiliza esta herramienta para identificar un modelo del sistema térmico a partir de los datos de entrada-salida obtenidos del sistema. El resultado de la estimación proporciona información sobre la dinámica del sistema, como el orden y los parámetros del modelo. En este caso, se estiman tres modelos diferentes: de primer, segundo y tercer orden. Cada uno de estos modelos captura diferentes aspectos del comportamiento dinámico del sistema térmico. Las Ecuaciones 3.1, 3.2, 3.3 muestran los modelos identificados para el sistema térmico de primer, segundo y tercer orden, respectivamente. Estas ecuaciones proporcionan visualizaciones de la respuesta del modelo en comparación con los datos experimentales, lo que permite evaluar la calidad del ajuste y la adecuación del modelo para describir el comportamiento del sistema.

Listing 3.1: Ejemplo de código de MATLAB con comentarios

```

% Definir constantes
R = 10;           % Resistencia electrica (Ohm)
C = 372;         % Capacidad termica (Faradios)
A = 1;           % Area de superficie (m^2)
h_conv = 10;     % Coeficiente de transferencia
                 % de calor por conveccion (W/(m^2*K))
T_amb = 25;      % Temperatura ambiente (C)
sigma = 5.67e-8; % Constante de Stefan-Boltzmann (W/(m^2*K^4))
epsilon = 1;     % Emisividad

% Condiciones iniciales
T0 = 25;         % Temperatura inicial (C)
tspan = -1:1:700; % Intervalo de tiempo de simulacion

% Definir escalon de entrada de voltaje
V_step = 220;    % Cambio en el voltaje (V)
t_step = 400;    % Momento del escalon (s)
unitstep(tspan >= 0 & tspan < 100) = 0;
unitstep(tspan >= 100 & tspan < 400) = 1;
unitstep(tspan >= 400 ) = 1.5;
Escalon=100 *unitstep;
Escalon1= Escalon';

% Definir la ecuacion diferencial incluyendo el escalon de entrada
dTdt = @(t, T) ((Volt_step(t, t_step))^2/R - A * h_conv *...
(T - T_amb) - sigma * epsilon * (T^4 - T_amb^4)) / C;

% Definir el intervalo de tiempo
% De 0 a 1000 segundos

% Resolver la ecuacion diferencial
[t, T] = ode45(dTdt, tspan, T0);

```

```
% Preparacion de datos para la identificacion del sistema
Ts=0.1;
u_ide=Escalon1(300:end);
y_ide=T(300:end);
t_ide=0:Ts:40.3-Ts;

% Graficar la solucion
figure
plot(t_ide, u_ide);
xlabel('Tiempo (s)');
ylabel('Temperatura (C)');
title('Temperatura con escalon de entrada de voltaje');
grid on;

% Grafico de la respuesta al escalon
plot(t_ide, y_ide);

% Normalizacion de datos
u_ide=u_ide-u_ide(1);
y_ide=y_ide-mean(y_ide(1:50));
figure
subplot(211);plot(t_ide,u_ide,'b','LineWidth',2);
xlabel('Tiempo(s)');ylabel('Amplitud')
subplot(212);plot(t_ide,y_ide,'r','LineWidth',2);
xlabel('Tiempo(s)');ylabel('Amplitud')

% Identificacion del modelo del sistema
data=iddata(y_ide,u_ide,Ts);
sys1=tfest(data,1,0,0);
figure
compare(data,sys1);title('Un polo')
sys2=tfest(data,2,0,0);
```

```

figure
compare(data,sys2);title('Dos polos')
sys3=tfest(data,3,1,0);
figure
compare(data,sys3);title('Tres polos')

% Analisis de respuesta al escalon
figure
subplot(211);step(sys1)
subplot(212);pzmap(sys1)
S = stepinfo(sys2)

% Funcion para el escalon de entrada de voltaje
function V = Volt_step(t, t_step)
    if t <= t_step
        V = 110;
    else
        V = 220;
    end
end
end

```

- **Definición de constantes:** Se definen varias constantes como la resistencia eléctrica (R), la capacidad térmica (C), el área de superficie (A), el coeficiente de transferencia de calor por convección (h_{conv}), la temperatura ambiente (T_{amb}), la constante de Stefan-Boltzmann (σ), la emisividad (ϵ) y la temperatura inicial (T_0).
- **Definición del intervalo de tiempo:** Se establece el intervalo de tiempo sobre el cual se resolverá la ecuación diferencial.
- **Definición del escalón de entrada de voltaje:** Se define una función `Volt_step` que simula el cambio de voltaje en el sistema. Esta función devuelve un voltaje de 110V antes del tiempo del escalón y 220V después del tiempo del escalón.

- **Definición de la ecuación diferencial:** Se define la ecuación diferencial que describe la evolución de la temperatura en el sistema en función del tiempo y la temperatura actual. La ecuación incluye el efecto del cambio de voltaje, la transferencia de calor por convección y la radiación térmica.
- **Resolución de la ecuación diferencial:** Se utiliza la función `ode45` para resolver la ecuación diferencial y obtener la evolución de la temperatura en el tiempo.
- **Preparación de los datos para la identificación del sistema:** Se preparan los datos eliminando los primeros 300 puntos para evitar el transitorio inicial.
- **Identificación del modelo del sistema:** Se utiliza la función `tfest` para estimar un modelo de transferencia de función de transferencia a partir de los datos de entrada-salida.
- **Análisis del modelo identificado:** Se realizan varios análisis del modelo identificado, incluyendo comparaciones entre los datos reales y la respuesta del modelo, así como visualizaciones de los polos y ceros del sistema.
- **Análisis adicional del sistema:** Se realiza un análisis adicional del sistema, incluyendo el cálculo de los parámetros del sistema y la respuesta al escalón del sistema identificado.

$$\frac{2,867}{s + 0,5949} \quad (3.1)$$

$$\frac{1,182e^{0,5}}{s^2 + 4,122e^{0,4}s + 2,452e^{0,4}} \quad (3.2)$$

$$\frac{40,19s + 139,6}{s^3 + 7,157s^2 + 57,84s + 28,91} \quad (3.3)$$

A través del código de Matlab antes evidenciado, se obtiene las tres ecuaciones las cuales son de primer (3.1), segundo (3.2) y tercer (3.3) orden, respectivamente. De estas tres ecuaciones se elige la ecuación 3.1, que corresponde a un sistema de un polo ya que el aumento de precisión de las ecuaciones de 2 y 3 polos son insignificantes en contraste del aumento de los polos obteniendo un modelo mas complejo y una sintonización mas complicada del controlador. En la figura 3.1 se puede evidenciar la gráfica de respuesta de un sistema con un polo. A su vez en la figura 3.2 se muestra la gráfica de respuesta de un sistema con dos polos y en la Figura 3.3 la gráfica de respuesta de un sistema con tres polos.

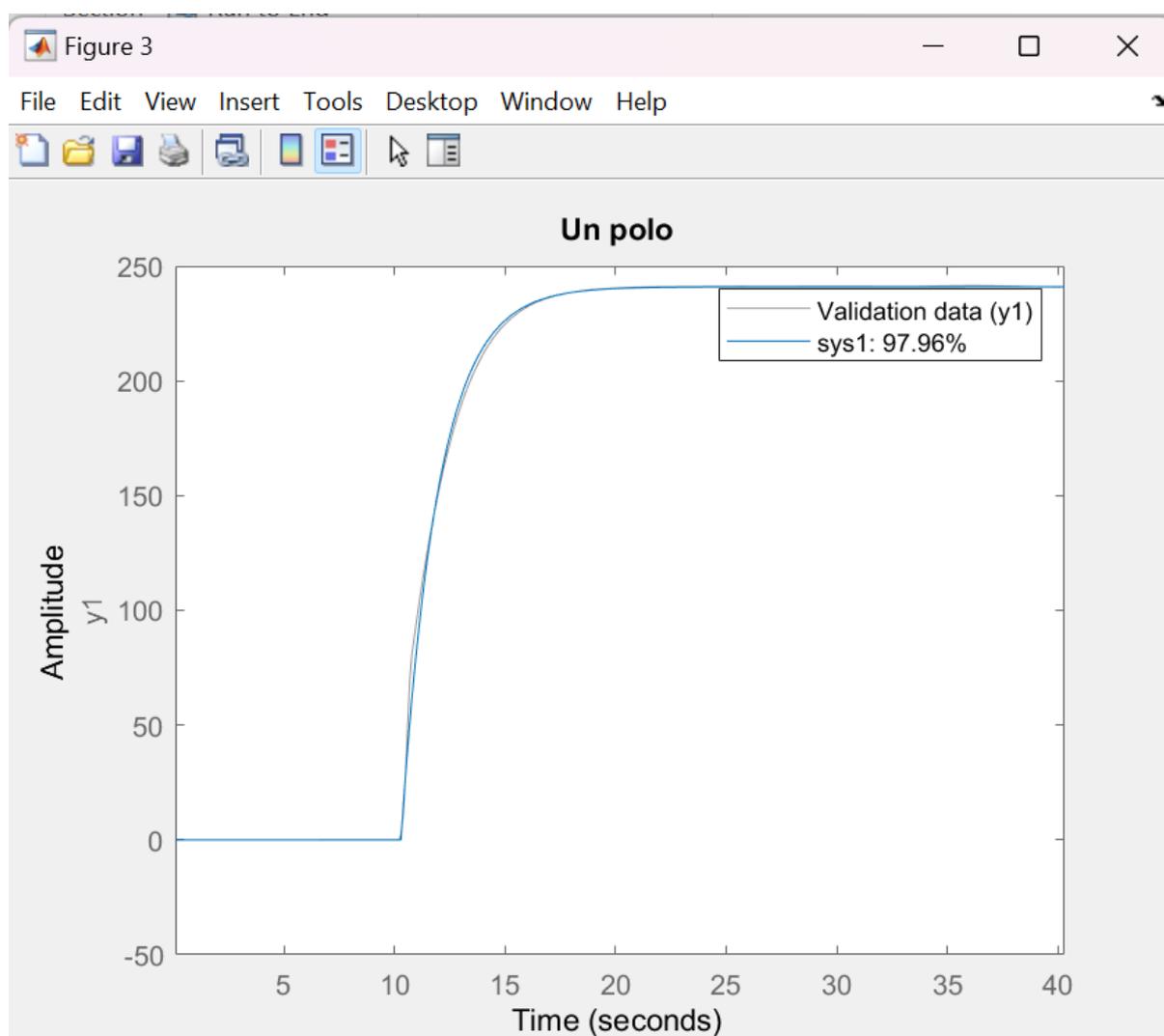


Figura 3.1: Identificación del sistema térmico con un Polo

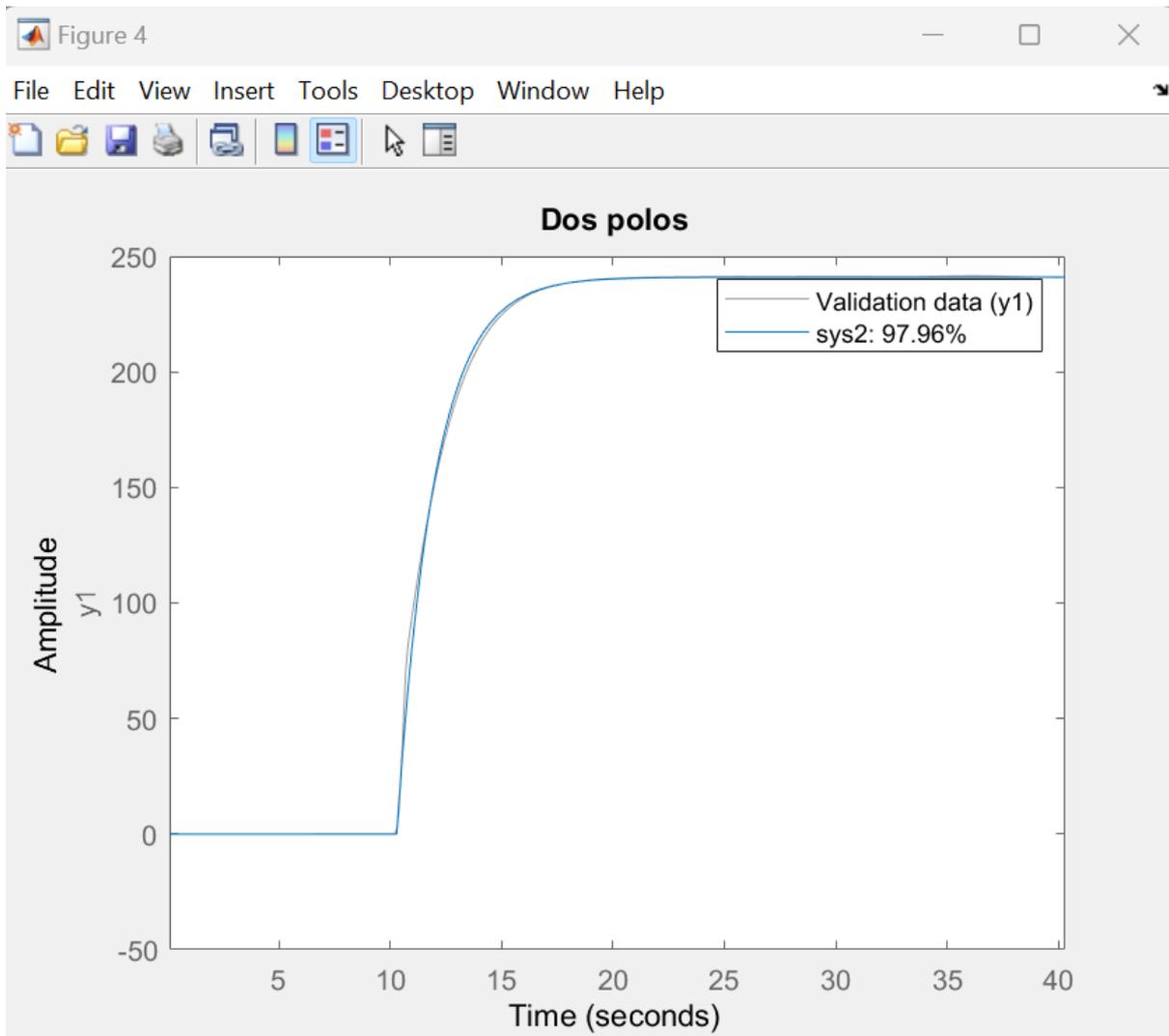


Figura 3.2: Identificación del sistema térmico con dos Polos

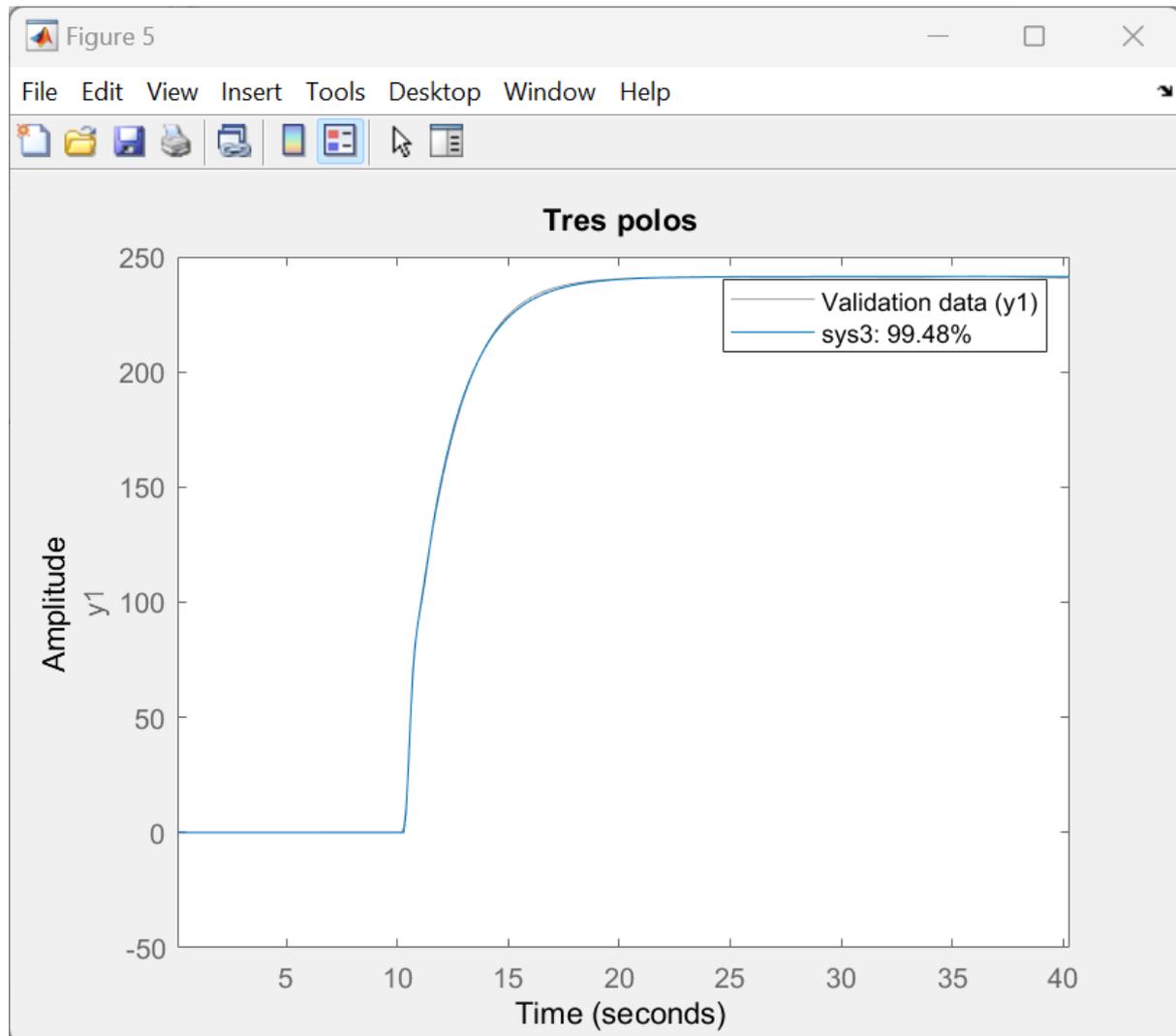


Figura 3.3: Identificación del sistema térmico con tres Polos

3.2. Control del proceso utilizando la herramienta PID Compact de TIA Portal

Una vez que se ha creado el servidor y el cliente OPC UA con conexión bidireccional, se puede poner en marcha el controlador PID. A continuación se presentan los pasos a seguir para la implementación del controlador [8].

- En TIA Portal se crea un nuevo bloque de cyclic Interrupt, el cual servirá para la comunicación OPC UA y la ejecución del PID (véase la Figura 3.4).
- Dentro de Tecnología en TIA Portal, se selecciona el PID Compact, el cual se inserta en el bloque cyclic Interrupt (véase la Figura 3.5).

3.2. CONTROL DEL PROCESO UTILIZANDO LA HERRAMIENTA PID COMPACT DE TIA PORT

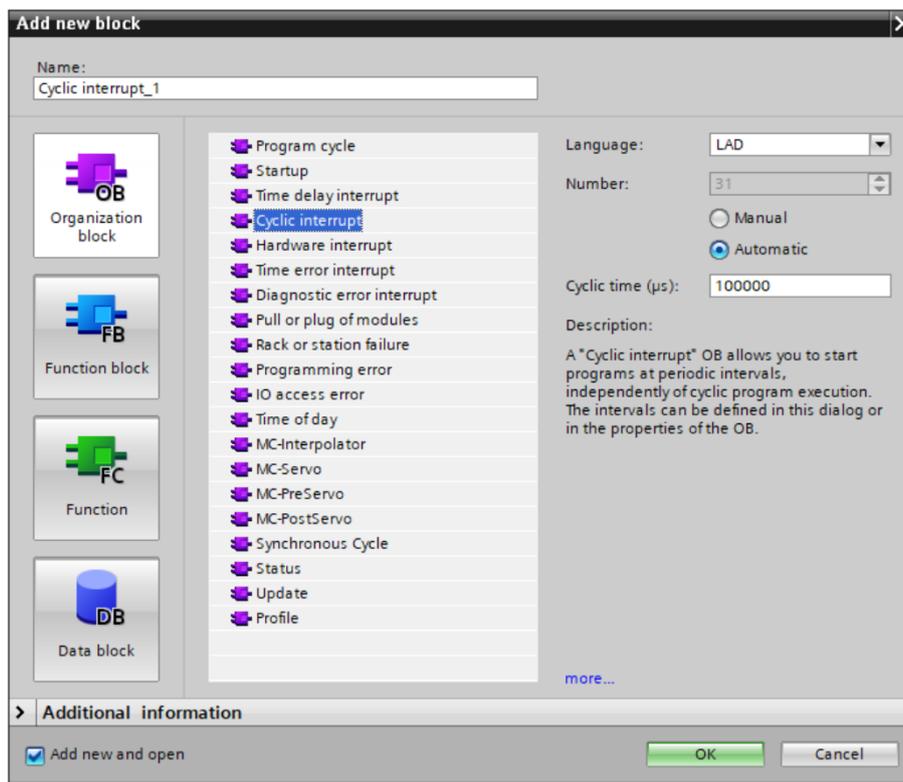


Figura 3.4: bloque cycling interrupt

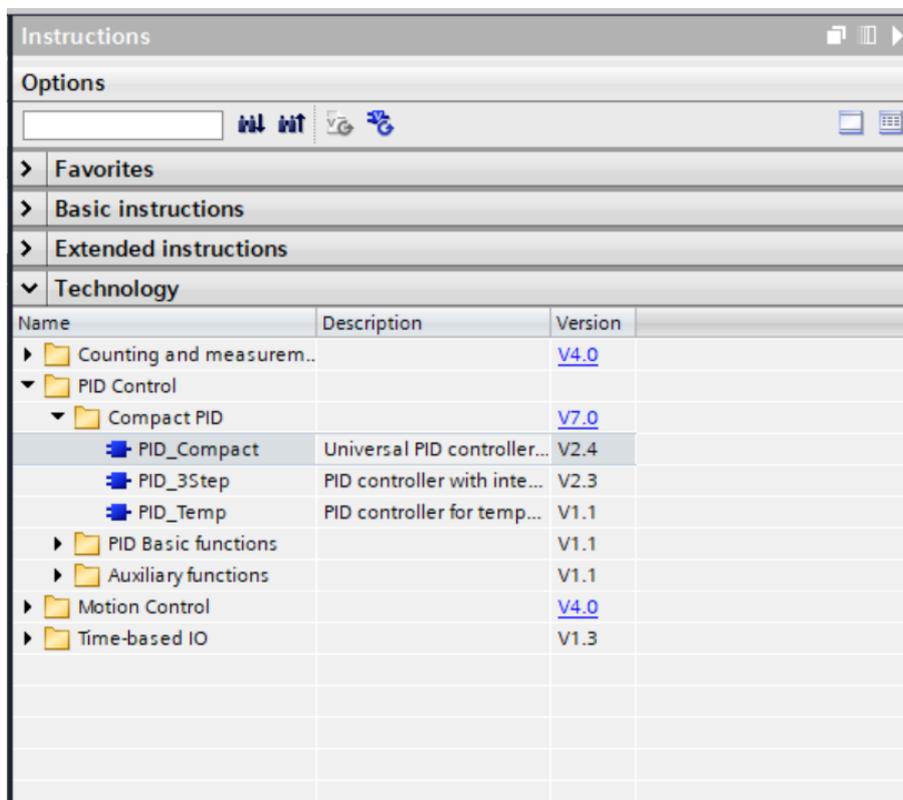


Figura 3.5: Ubicación de la herramienta PID Compact

3.2. CONTROL DEL PROCESO UTILIZANDO LA HERRAMIENTA PID COMPACT DE TIA PORT

- A continuación, se define los parámetros del control PID, como las constantes de proporcionalidad, integral y derivativa, así como los límites de salida y los rangos de operación (véase la Figura 3.6)

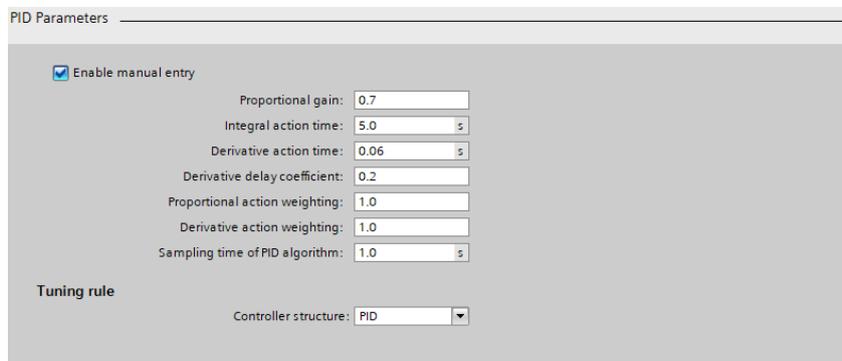


Figura 3.6: Parámetros pid compact

Una vez configurado se puede poner en marcha el controlador PID, a través del bloque que se evidencia en la Figura 3.7.

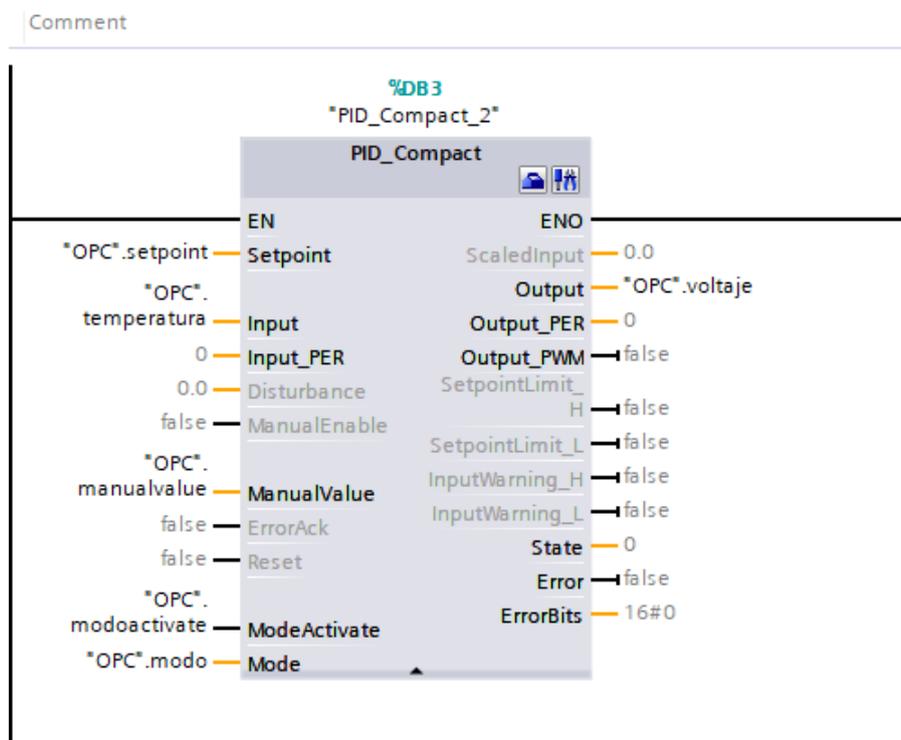


Figura 3.7: Bloque de PID Compact

3.2. CONTROL DEL PROCESO UTILIZANDO LA HERRAMIENTA PID COMPACT DE TIA PORTAL

Como se puede observar en el bloque PID Compact, se tiene varias entradas y salidas, de las cuales se utilizará solo algunas de ellas, como son: Setpoint, input, ManualValue, ModeActivate, Mode y Output. A continuación se explicará para que sirve cada entrada y salida utilizada en el bloque.

1. Setpoint: es el valor deseado, al cual la temperatura deberá llegar en un lapso de tiempo.
2. Input: es el dato de temperatura que se guarda en el servidor OPC UA, escrito por el cliente.
3. ManualValue: Permite poner al controlador en modo manual.
4. ModeActivate: Esta entrada recibe una señal booleana la cual cambia de estado para poder afirmar el cambio de modo ya sea a manual o automático.
5. Mode: Este dato debe ser un entero y puede tomar el valor de 4 (manual) o 3 (automático).
6. Output: Esta es la salida del controlador. Para este caso, esta salida se convertirá en el voltaje que se guarda en una variable del servidor y es enviada a al modelo de Simulink para que varíe la temperatura.
7. Measurement: tanto la salida Representada con color rojo, la entrada de color verde y el setpoint de color negro podrán ser visualizados gracias a la herramienta del Tuning Mode que brinda el PID compact (véase la Figura 3.8)
8. Tracer: esta aplicación esta embebida en TIA portal. Traza gráficas de las variables que se seleccionen. Para este caso se configura para visualizar el setpoint y la variable controlada (véase la Figura 3.9).

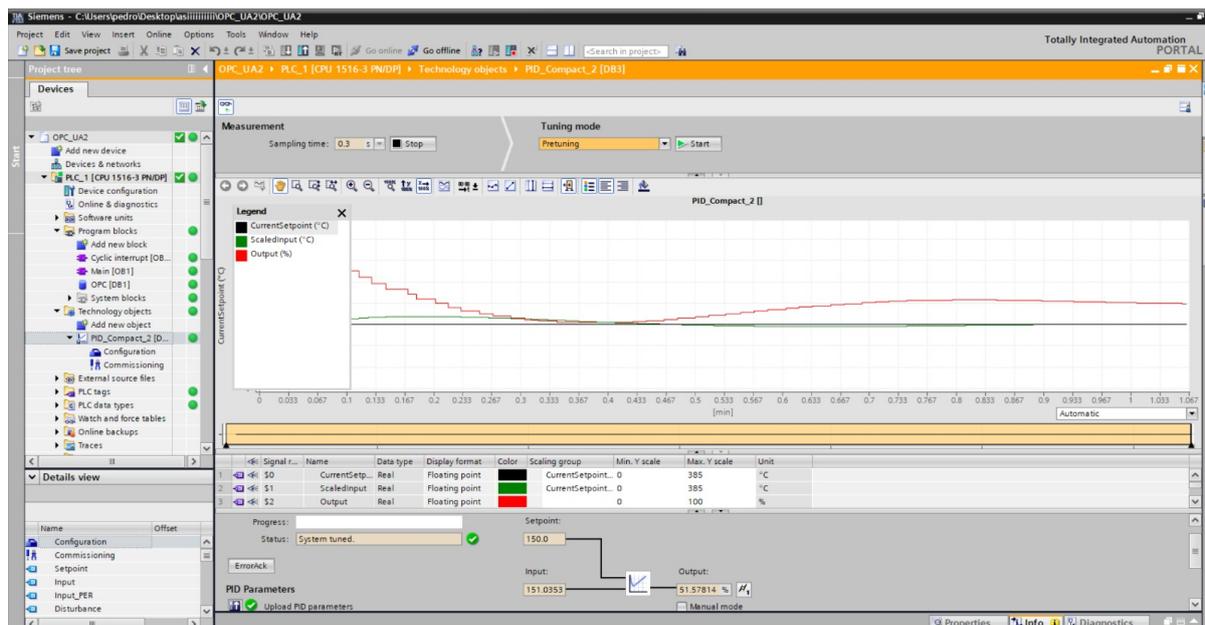


Figura 3.8: Gráfica de la herramienta tuning mode

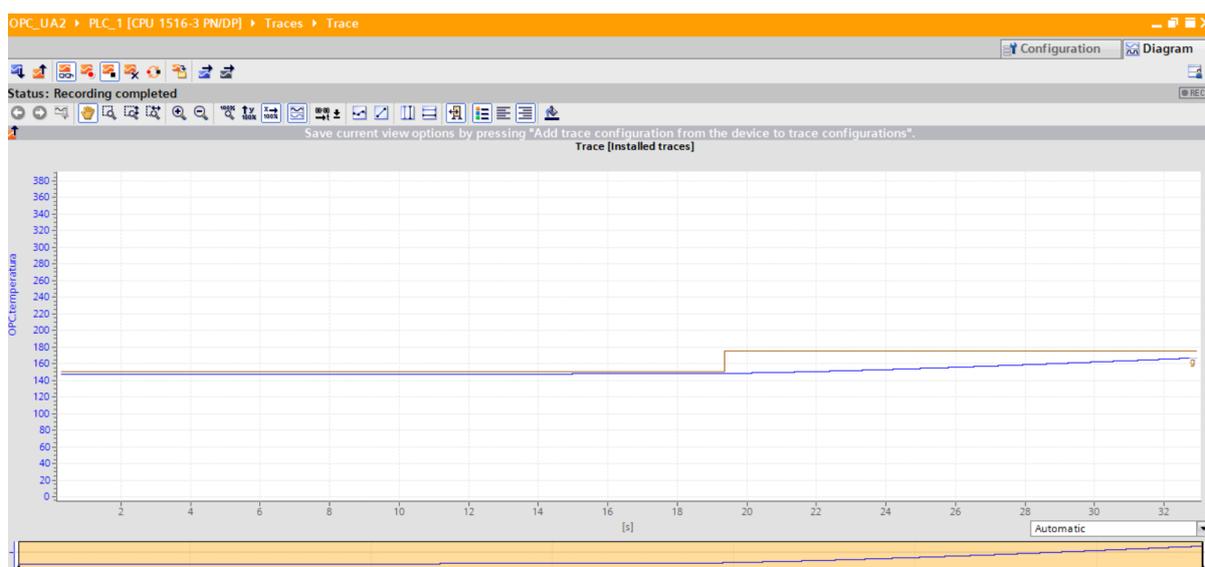


Figura 3.9: Gráfica de la herramienta Tracer

3.3. Implementación del HMI

Dentro de la herramienta Tia Portal el desarrollo de las interfaces hombre-maquina(HMI) son muy necesarias por que permiten a los operadores interactuar con los sistemas de control de una manera fácil e intuitiva, como ya se sabe tenemos varias aplicaciones donde una de gran interés es el control y visualización de datos en tiempo real,esto es indispensable en cualquier proceso industrial.[9]

- Como primer paso se debe acceder al Árbol de proyecto y seleccionar Agregar un nuevo dispositivo donde se pueden elegir ya sean controladores o demás dispositivos. Esto se puede evidenciar en la Figura 3.10

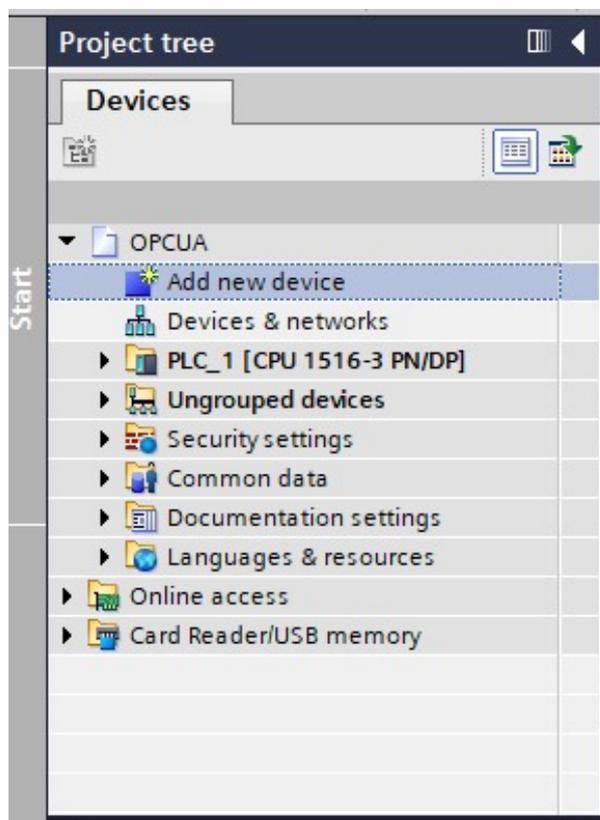


Figura 3.10: Agregar nuevo dispositivo a de proyecto Tia Portal

- En ese apartado se selecciona la opción de HMI donde se podrá elegir el modelo que convenga para la aplicación de cualquier proceso a seguir. Esto se puede evidenciar en la Figura 3.11

Una vez agregado el dispositivo HMI en el proyecto de el servidor OPC AU, lo siguiente es realizar la conexión de el PLC con e HMI atravésé de comunicación profinet, esto se puede evidenciar en la Figura 3.12

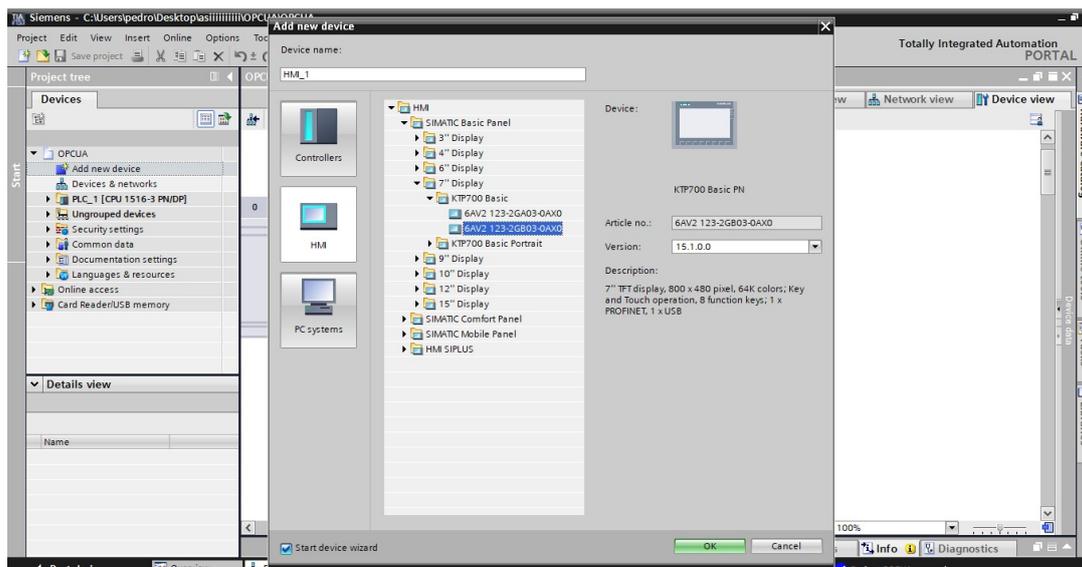


Figura 3.11: Selección de dispositivo HMI

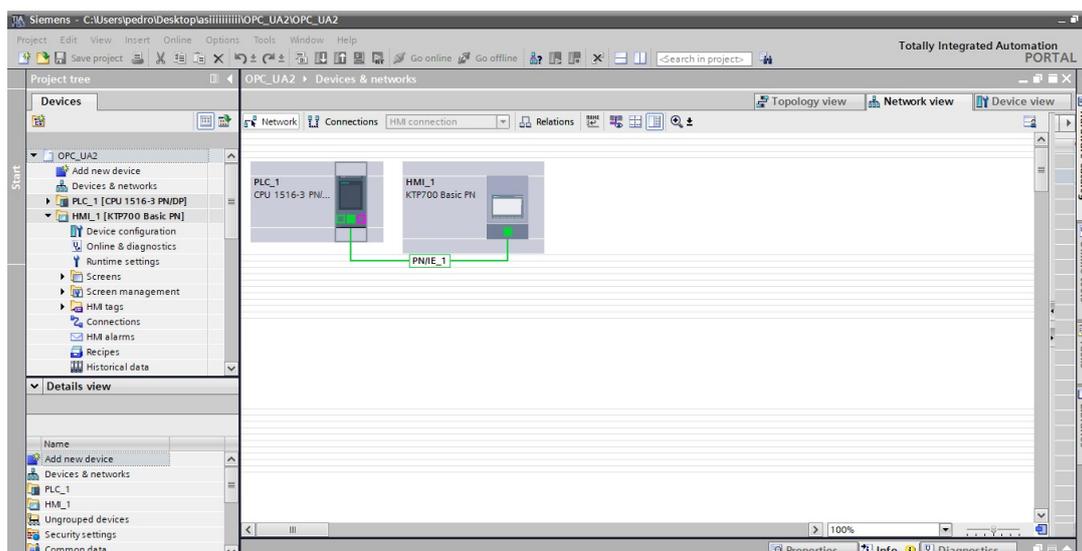


Figura 3.12: Comunicación Profinet de plc con HMI

Una vez conectado el HMI se procede a la creación de la interfaz gráfica, para poder hacerlo nos situamos en el Árbol de proyecto del dispositivo HMI donde nos situaremos en el apartado de Imagen Raíz la cual es la imagen principal de la interfaz.[10]

- En la imagen raíz del HMI se crea una carátula, la cual sera una presentación del proyecto y de los autores del mismo, como se puede observar en la Figura 3.14
- Una vez creada la imagen Raíz podemos crear nuevas imágenes donde se

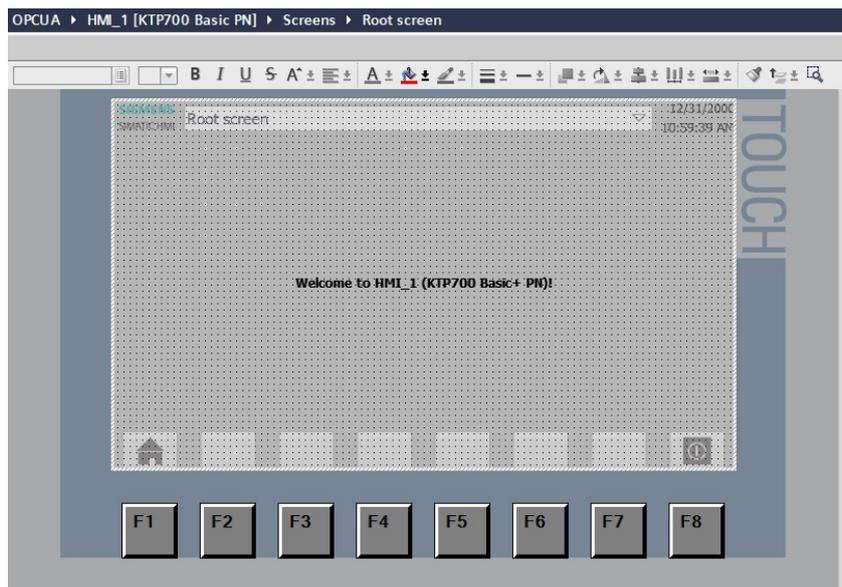


Figura 3.13: Imagen Raíz

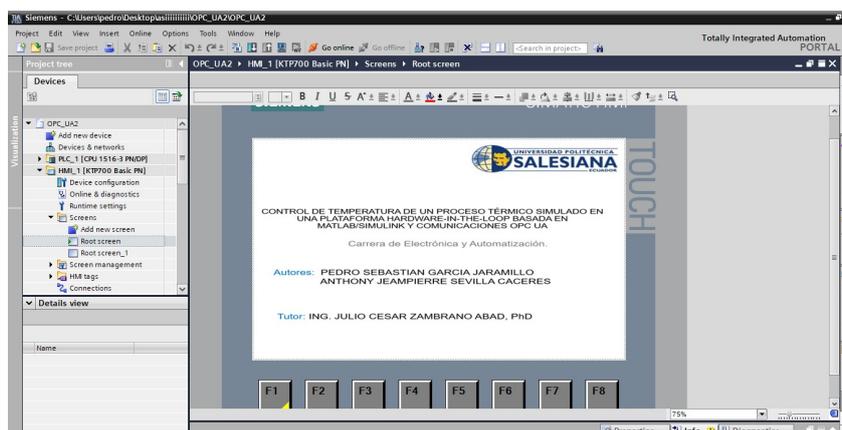


Figura 3.14: Carátula en imagen raíz

colocara información relevante del proceso. la nueva imagen creada sera la encargada de mostrar la entrada de voltaje que ingresa al proceso térmico. En esta podemos observar la gráfica a tiempo real del voltaje además de valores numéricos tanto del propio voltaje así como el de la temperatura actual del proceso y demás datos no tan relevantes en ese momento. Todo esto se puede evidenciar en la Figura 3.15

- Por ultimo se crea una imagen mas, la cual mostrara una gráfica del control PID del proceso térmico, esta mostrara en tiempo real la gráfica de la temperatura controlada por el controlador PID además de poseer la visualización y modificación de ciertos datos ya sean como, la visualización de la temperatura

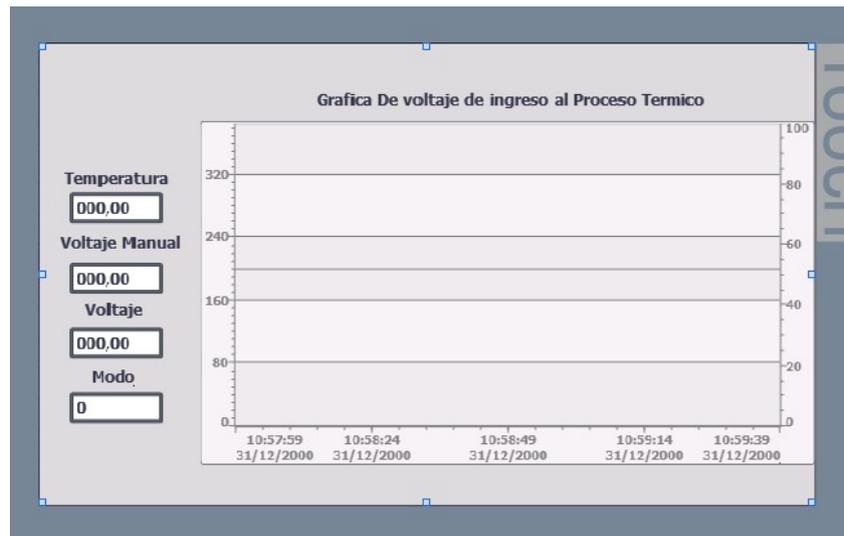


Figura 3.15: Gráfica de la entrada de voltaje que ingresa al proceso térmico

del proceso y el voltaje que ingresa al proceso. Los datos que se modifican son el setpoint el cual ya vimos que es el valor al que se desea llegar gracias al controlador y también se puede modificar el modo de operación del controlador ya sea en 3 o 4, esto se puede evidenciar de mejor manera en el Capítulo ?? Todo lo mencionado se evidencia en la Figura 3.16

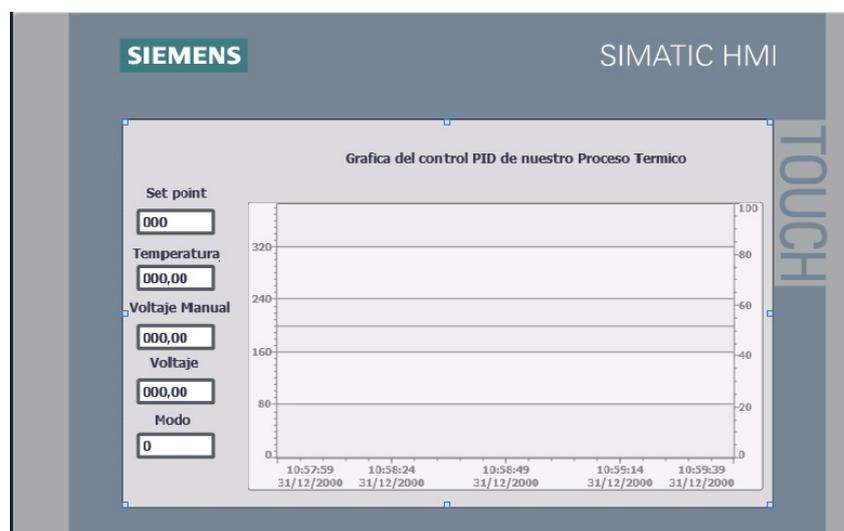


Figura 3.16: gráfica del control PID del proceso térmico

3.3.1. Pruebas del sistema

En este apartado se comprueba el funcionamiento del controlador PID sobre el proceso industrial, Este se vera reflejado a través de las siguientes gráficas.

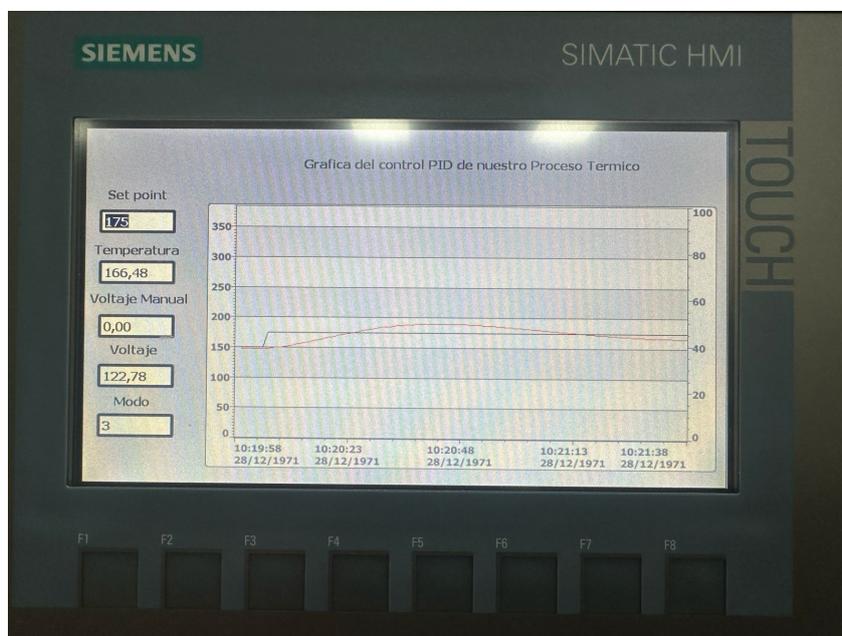


Figura 3.17: gráfica del control PID del proceso térmico

En la figura 3.17 se puede evidenciar de forma clara como es el cambio de temperatura dado por el controlador. En esta se observa algunos valores importantes como lo es el setpoint , la temperatura y el voltaje. En esta gráfica se tiene una línea negra la cual es la representación del valor de setpoint impuesto, este anteriormente poseía el valor de 150 y posteriormente fue modificado a un valor de 175 grados. la línea roja que se puede observar es la representación del valor de temperatura, la cual va cambiando gracias al controlador, esta estaba con un valor de 150 la cual va aumentando ya que le voltaje de ingreso al sistema también lo hace, la temperatura eleva rápidamente hasta sobrepasar el setpoint, llegado a este punto el valor de voltaje pasa a 0 para que la temperatura se pueda estabilizar de una manera mas rápida, cuando la temperatura se aproxima al valor de setpoint el voltaje comienza a subir para poder estabilizar la temperatura en el setpoint establecido.

En la figura 3.18 se puede evidenciar los cambios que tiene la señal de voltaje a través de todo el proceso de control de temperatura. El voltaje es presentado en la gráfica con una línea de color negro, la cual sube y baja. Esto representa todo el

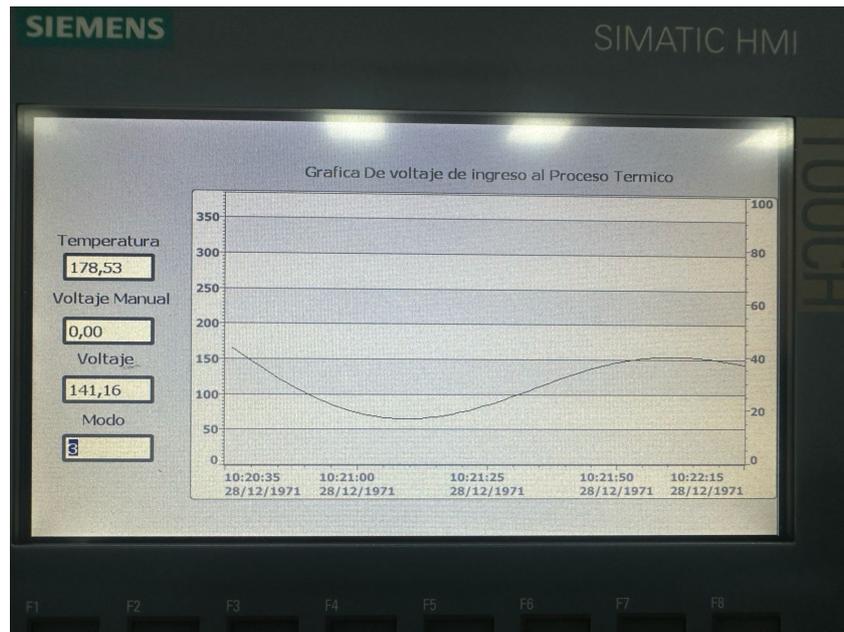


Figura 3.18: gráfica de entrada de voltaje al sistema

proceso que se explico anteriormente, el valor de voltaje fue desde que tiene una temperatura inicial, cuando el setpoint fue de 150 y por ultimo cuando el valor de setpoint se modifica a 175 y se logra evidenciar de igual manera que se estabiliza el voltaje una ves que este es el adecuado para la estabilización de la temperatura con el valor deseado de setpoint.

3.4. Pruebas y Resultados

Se realiza una prueba del funcionamiento del sistema en lazo abierto, en la cual se ingresa diferentes valores de voltaje, con esto se puede observar el comportamiento de la temperatura hasta llegar a un punto de estabilidad. Se ingresaron valores de 100, 150 y 220 voltios respectivamente y así poder observar cual es el punto de estabilidad del sistema con el valor de voltaje ingresado. Para el valor de 100V el sistema llega a una temperatura de 140 grados, con 150V el sistema entrega 231 grados y finalmente con el máximo de voltaje de 220V entrega una temperatura máxima de 384 grados. Esto se puede observar en la Figura 3.19. En la figura 3.19 se observa la

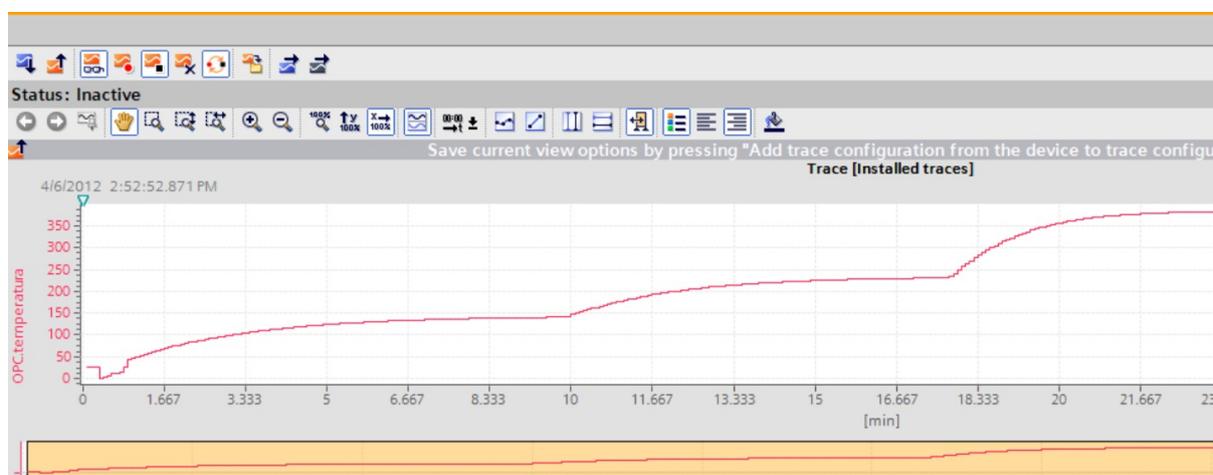


Figura 3.19: Gráfica de respuesta del sistema con diferentes valores de voltaje

gráfica de respuesta del sistema en el trace de Tia Portal, la cual no se aprecia de la mejor manera, para una mejor visualización de esta, se extraen los valores del trace como un archivo csv el cual es ingresado en Matlab para poder obtener la gráfica de respuesta del sistema. Esto se puede observar en la Figura 3.20

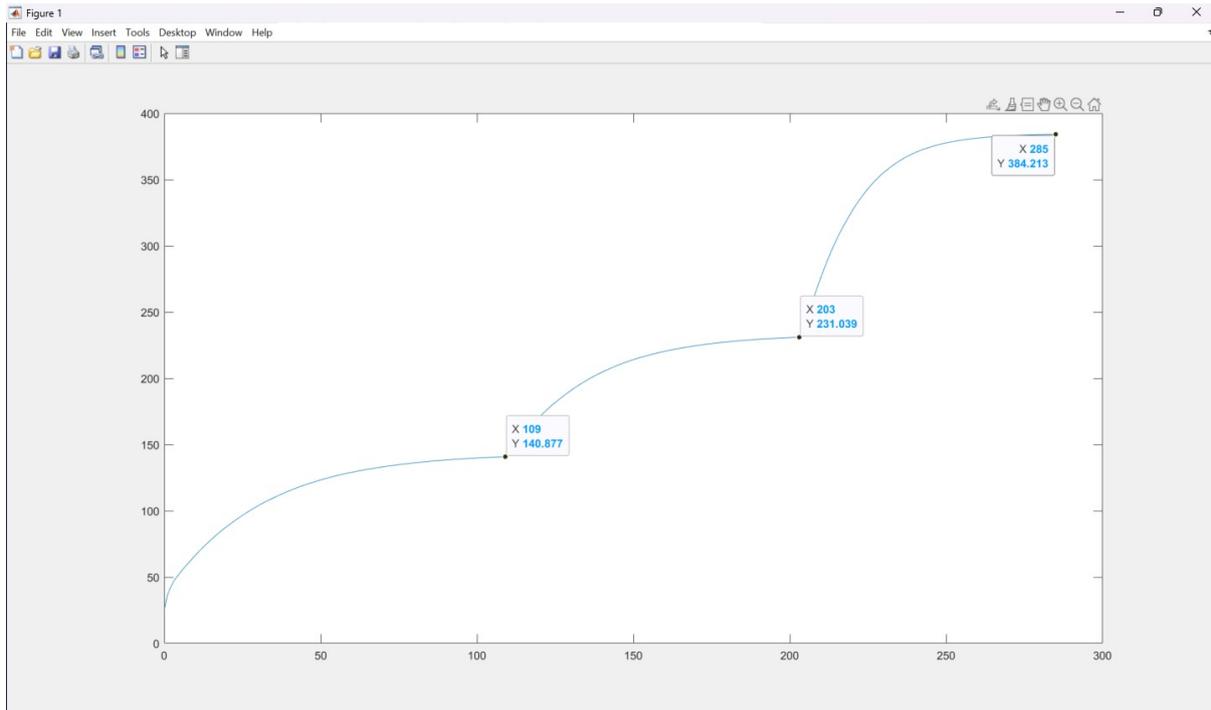


Figura 3.20: Gráfica de respuesta del sistema con diferentes valores de voltaje en Matlab

Una vez realizada la prueba del sistema con los diferentes valores de voltaje, se procede a comparar esa respuesta junto a cada valor de setpoint para ver de mejor manera el comportamiento de la temperatura esto de igual manera se observa en el trace de Tia Portal. Esto se observa en la Figura 3.21.

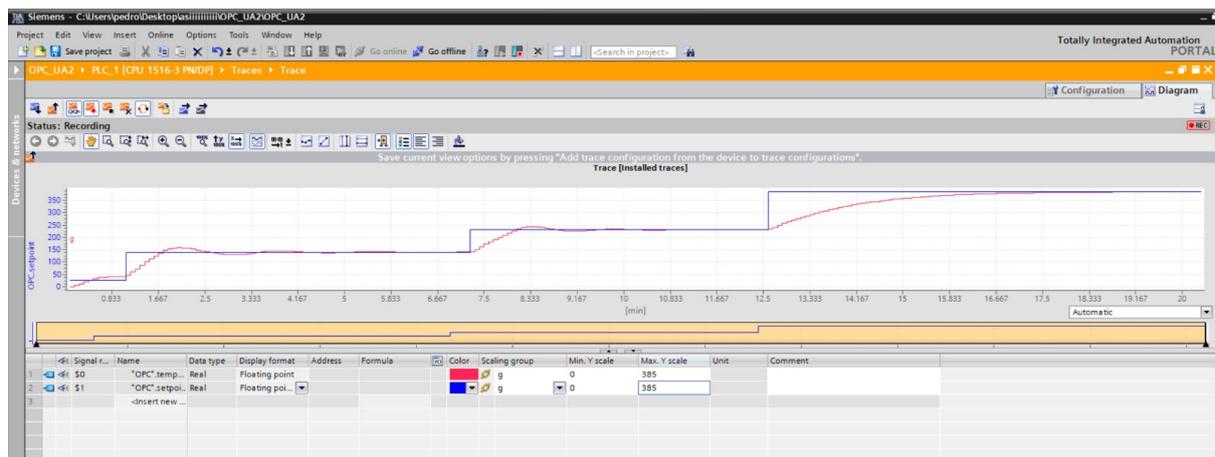


Figura 3.21: Gráfica de respuesta del sistema con valores de setpoint

De igual manera se extrajo los valores de las gráficas en un archivo csv para poder graficar de en Matlab para una mejor visualización, Figura 3.21 Y finalmente

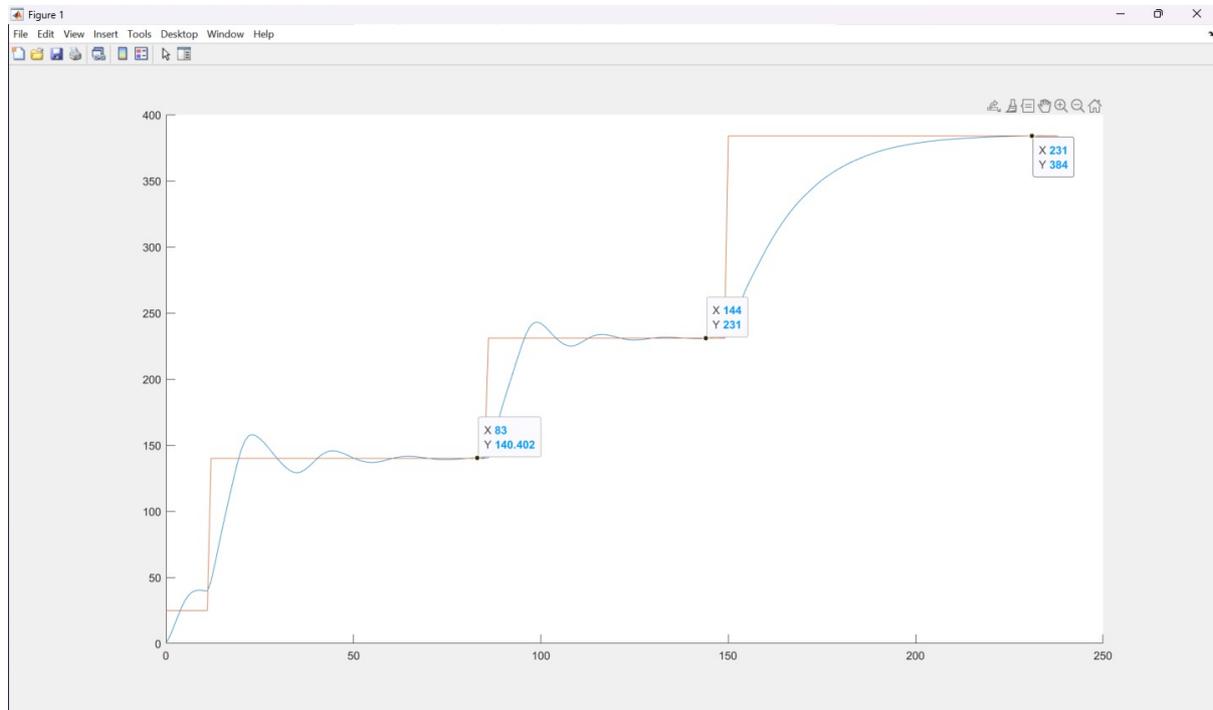


Figura 3.22: Gráfica de respuesta del sistema con valores de setpoint en Matlab

se muestra la gráfica del voltaje resultante de la acción de control y a la vez entrada de voltaje al sistema. En esta se puede observa la acción de control en el voltaje, ya que se observa como el valor de voltaje sube a su valor máximo de 220v para poder elevar con rapidez la temperatura, al aproximarse al valor deseado de temperatura el valor de voltaje comienza disminuir y así se puede observar la acción de control la cual aumentara y disminuirá hasta establecerse en un voltaje fijo para el valor deseado de temperatura. Se puede observar la gráfica de esto en el trace de Tia Portal Figura 3.23 y de igual manera en una gráfica de Matlab Figura 3.24 ,

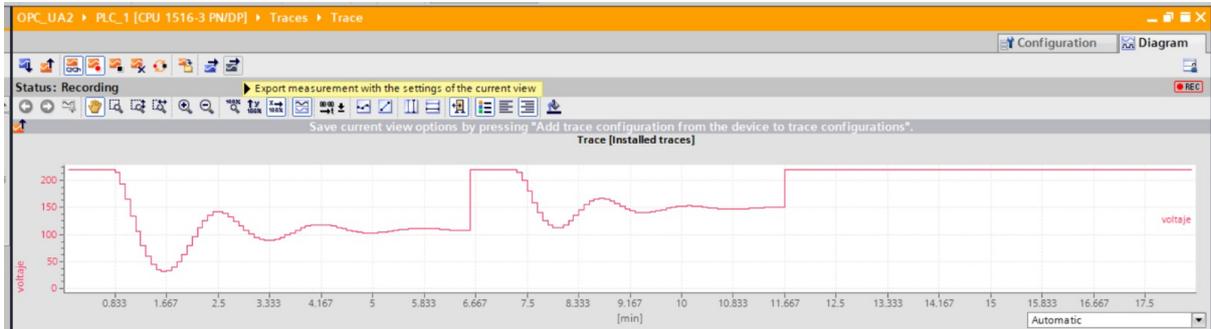


Figura 3.23: Gráfica de voltaje de ingreso al sistema Tia Portal

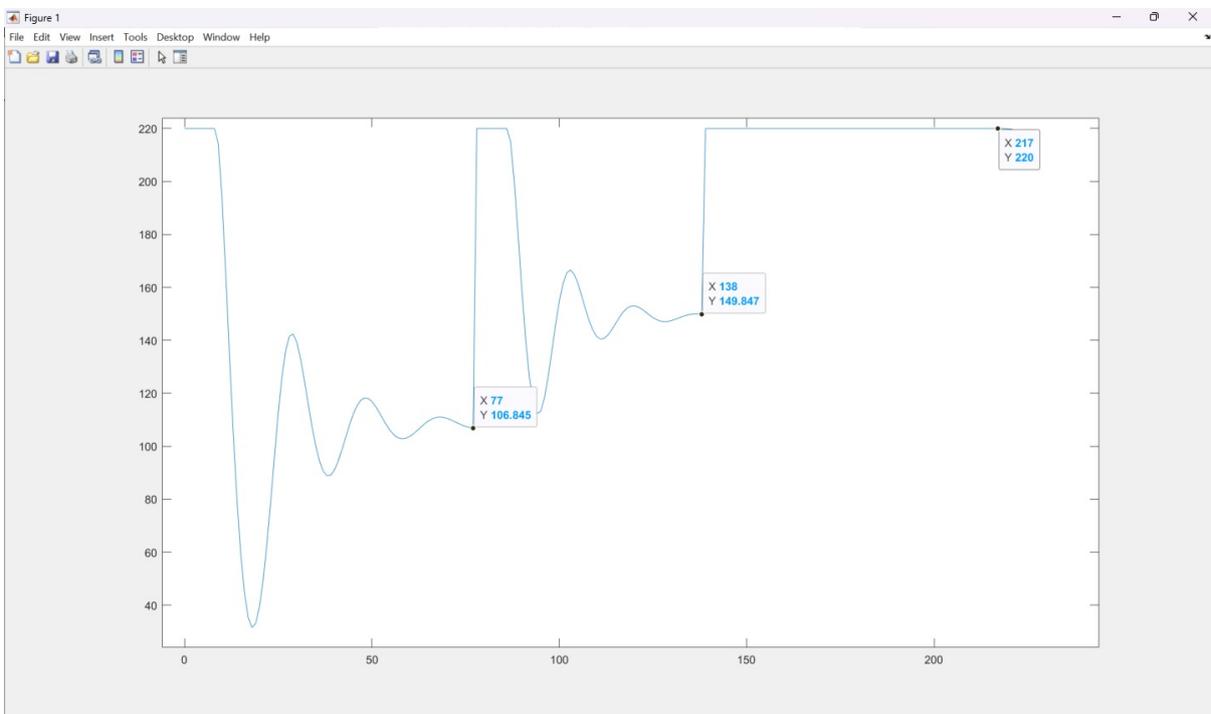


Figura 3.24: Gráfica de voltaje de ingreso al sistema Matlab

Capítulo 4

Conclusiones y trabajos futuros

4.1. Conclusiones

La integración de OPC UA con MATLAB Simulink ofrece una solución potente para la comunicación y control en sistemas industriales. La capacidad de OPC UA para proporcionar una comunicación segura y estándar entre diferentes dispositivos y sistemas, combinada con la capacidad de Simulink para modelar y simular sistemas dinámicos, ofrece a los ingenieros una herramienta integral para el diseño y la implementación de sistemas de control complejos. La interoperabilidad entre OPC UA y MATLAB Simulink facilita la implementación de estrategias de control avanzadas y la optimización del rendimiento de los sistemas industriales.

OPC UA para el monitoreo en tiempo real a través de HMI proporciona una solución eficaz para supervisar y controlar procesos industriales. Al utilizar OPC UA como protocolo de comunicación, se garantiza una conectividad confiable y segura entre los dispositivos de campo y el sistema HMI. Esto permite a los operadores acceder a datos en tiempo real, realizar ajustes y tomar decisiones informadas para mejorar la eficiencia y la productividad en la planta. La flexibilidad y la interoperabilidad de OPC UA permiten una integración sin problemas con una variedad de sistemas HMI, lo que facilita la visualización y el control de los procesos industriales desde una interfaz intuitiva.

El PID Compact en TIA Portal ofrece una solución eficiente y fácil de implementar para el control de procesos en entornos industriales. Al integrar el

control PID directamente en el entorno de programación de TIA Portal, los ingenieros pueden configurar y ajustar los parámetros del controlador PID de manera rápida y sencilla. La funcionalidad compacta del PID permite un uso eficiente de los recursos de hardware y una respuesta rápida a cambios en las condiciones del proceso. Además, la integración del PID Compact con otras funciones de automatización en TIA Portal facilita la implementación de sistemas de control complejos y la optimización del rendimiento de los procesos industriales.

4.2. Trabajos futuros

Realizar un análisis detallado del rendimiento en tiempo real de la implementación del control PID en el sistema. Esto puede incluir la evaluación de la latencia de comunicación, la frecuencia de muestreo del controlador, y la estabilidad y precisión del control en diferentes condiciones operativas. Desarrollar un sistema de supervisión y diagnóstico que pueda detectar y diagnosticar fallos en el sistema de control en tiempo real. Esto puede ayudar a mejorar la robustez y la fiabilidad del sistema en operaciones a largo plazo.

Siguiendo con la implementación de controladores PID, la herramienta PID compact nos proporciona más tipos de controlador pid, ya sea la opción de PID-3step o la de pid-temp. Estas opciones nos abren campo para ya sea en el ámbito laboral o académico, ya que dependiendo del problema se puede utilizar el más adecuado ya sea por su precisión o por su confiabilidad.

Glosario

HIL Hardware-in-the-Loop.

HMI Interfaz Hombre-Máquina.

OPC OLE for Process Control.

OPC UA Object Linking and Embedding for Process Control Unified Architecture.

PID Proporcional-Integral-Derivativo.

PLC controladores lógicos programable.

SCADA Supervisory Control and Data Acquisition.

Referencias

- [1] A. E. Zerk, M. Ouassaid e Y. Zidani, «Development of a real-time framework between MATLAB and PLC through OPC-UA: A case study of a microgrid energy management system,» *Scientific African*, vol. 21, e01846, 2023.
- [2] R. Henssen y M. Schleipen, «Interoperability between OPC UA and AutomationML,» *Procedia Cirp*, vol. 25, págs. 297-304, 2014.
- [3] D. Bruckner, M.-P. Stănică, R. Blair et al., «An introduction to OPC UA TSN for industrial communication systems,» *Proceedings of the IEEE*, vol. 107, n.º 6, págs. 1121-1131, 2019.
- [4] J. Smith y L. Johnson, «OPC UA: The Next Generation of Industrial Communication,» *Industrial Automation Journal*, vol. 25, n.º 3, págs. 45-56, 2021. DOI: 10.1234/iaj.2021.123456.
- [5] D. Bullock, B. Johnson, R. B. Wells, M. Kyte y Z. Li, «Hardware-in-the-loop simulation,» *Transportation Research Part C: Emerging Technologies*, vol. 12, n.º 1, págs. 73-89, 2004.
- [6] L. Chen y W. Wang, «A Review of Hardware-in-the-Loop Simulation Techniques for Real-Time Control Systems,» *IEEE Access*, vol. 8, págs. 112233-112245, 2020. DOI: 10.1109/ACCESS.2020.123456.
- [7] M. Garcia y C. Rodriguez, «Hardware-in-the-Loop Simulation for Control Systems Development: A Review,» *IEEE Transactions on Industrial Electronics*, vol. 68, n.º 5, págs. 4321-4334, 2019. DOI: 10.1109/TIE.2019.123456.
- [8] R. I. V. Kairuz y A. I. G. Santos, «Controladores MPC y PID con autoajuste para un proceso de dinámica rápida a través de MATLAB®/Simulink® y OPC,» *Revista Ingeniería Electrónica, Automática y Comunicaciones ISSN: 1815-5928*, vol. 36, n.º 3, págs. 80-93, 2015.

- [9] J. J. Flores Ortega y M. A. Zambrano Figueroa, «Desarrollo e Implementación de un Módulo Interactivo para el Desarrollo de Procesos Industriales Enfocados a la Electroneumática Mediante PLC S7-1500 y PANTALLA HMI,» B.S. thesis, 2022.
- [10] J. A. Matute Caguana y M. G. Zhindon Sumba, «Monitoreo y operación de un sistema industrial de servo posicionamiento: HMI basada en Python y OPC UA,» B.S. thesis, 2023.