

Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

Conexión OPC, identificación y control de la Planta
Multiprocesos en recirculación

Autor: Francisco Bravo Moreno

Tutor: Daniel Limón Marruedo

**Dpto. Departamento de Ingeniería de Sistemas y
Automática**

Escuela Técnica Superior de Ingeniería

Sevilla, 2021



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Conexión OPC, identificación y control de la Planta Multiprocesos en recirculación

Autor:

Francisco Bravo Moreno

Tutor:

Daniel Limón Marruedo

Profesor titular

Dpto. Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Trabajo de Fin de Grado: Conexión OPC, identificación y control de la Planta Multiprocesos en recirculación

Autor: Francisco Bravo Moreno

Tutor: Daniel Limón Marruedo

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mi familia

A mis maestros

A mis amigos

A mi equipo de remo

Agradecimientos

Un grado como el de Ingeniería Industrial ha sido un reto que sin el apoyo de los que han estado cerca de mí durante estos años hubiera sido imposible de superar.

Agradecer a mi familia por su apoyo y por su ayuda en todo lo que estaba en su mano.

También a mis amigos y en especial a Juanma y a Óscar. Jornadas maratonianas de biblioteca hubieran sido insufribles si no hubiera sido por ellos, por su apoyo y por la inspiración que reflejan las personas a las que rodean . Un equipo en el que la suma del conjunto es mayor que la de sus partes por separado y con el que no hay problema imposible de superar. Estos años han sido el principio de lo que nos queda.

En último lugar, mencionar a mi tutor durante este Trabajo de Fin de Grado Daniel Limón que me ha inspirado con su pasión por el aprendizaje y que ha colocado el objetivo de este proyecto en justo eso, la formación, el aprendizaje y en la superación de obstáculos encontrados a lo largo del mismo.

Francisco Bravo

Alumno de la Escuela Técnica Superior de Ingeniería de la universidad de Sevilla

Sevilla 2021

La planta Multiprocesos es un sistema para la implantación y prueba de técnicas y algoritmos de control que en ocasiones se quedan en planteamientos de problemas abstractos con los que cuesta establecer relaciones al mundo real.

En este Proyecto de Fin de Grado se ha realizado una actualización en el sistema de comunicaciones que tiene la planta con el usuario. Para poder implementar las técnicas y estrategias de control automático se requiere un entorno de trabajo adecuado y para ello se ha usado la comunicación OPC UA. Este sistema permite conectar el PLC que gobierna la planta con cualquier dispositivo que se desee dentro de la red Ethernet del laboratorio de Control.

Posteriormente, se ha realizado una descripción detallada de los aspectos a seguir para trabajar en el entorno de Matlab. Este programa junto a su extensión Simulink, ha sido sin duda uno de los más empleados a lo largo del grado por lo que es un lugar adecuado para la implementación de las metodologías de control.

Una vez se ha conseguido un espacio de trabajo adecuado, el proyecto se ha centrado en la tercera configuración de la planta que consiste en un proceso térmico de calentamiento y enfriamiento de una caudal de agua en un circuito cerrado a través del tanque de la planta. En esta configuración todavía no se ha implementado ningún sistema de control por lo que se ha obtenido un modelo lineal en variables de espacio de estados que represente las dinámicas de esta a partir del cual diseñar los parámetros necesarios para poder controlar las variables de la planta.

Dicho diseño ha consistido en el cálculo de las ganancias del controlador y el observador, así como un cálculo en línea de los estados y las señales de control a los que la planta deberá converger.

En último lugar, se ha probado dicha estrategia de control en el sistema real junto con su posterior análisis de los resultados obtenidos.

Abstract

The multiprocesses plant is a great set on where algorithms and techniques can be implemented that occasionally remain as abstracts problems with no connection with the real world.

On this project the communication system of the plant has been upgraded. To be able to implement different techniques and control strategies, a comfort working environment is required and for that purpose, the OPC UA system has been chosen. OPC UA can communicate the plant's PLC with any other device connected to the laboratory Ethernet network.

After that, it has been made a description of the different important aspects of using Matlab Workspace with the plant. This software has been, for sure, one of the mostly used along the degree and that's why it has been chosen as the option for implementing those techniques and algorithms.

Once a comfortable workspace is available, the project has been focused on the third configuration of the plant. No one has never worked on this configuration before, so the first stage has been obtaining a linear space-state variables model to describe its dynamics and design all the parameters needed to control the plant outputs.

In the Design stage have been estimated the controller and the observer gains as well as the online computing of the reference states and control signals that will define the plant response.

Finally, the control strategy has been tested in the real system and its results have been analyzed.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xix
Índice de Figuras	xxi
Notación	xxv
1 Introducción	1
1.1 <i>Motivación y objetivos</i>	1
1.3 <i>Estructura del trabajo</i>	4
2 Implementación de la comunicación a través de OPC UA	5
2.1 <i>OPC UA</i>	5
2.2 <i>Conexión del hardware de la planta a la red Ethernet</i>	6
2.2.1 <i>Cambio dirección IP PLC</i>	6
2.2.2 <i>Cambio dirección IP pantalla Magelis</i>	7
2.3 <i>Configuración y conexión del servidor EcoStruxure OPC UA server Expert</i>	8
2.3.1 <i>Cambio dirección IP del PC</i>	8
2.3.2 <i>Instalación y configuración EcoStruxure OPC UA server Expert</i>	10
2.3.3 <i>Uso de conexión Cliente-Servidor</i>	12
3 Control de la planta desde Matlab	11
3.1 <i>Conexión del servidor a Matlab</i>	11
3.2 <i>Funciones para inicio y apagado de la planta remotamente</i>	12
3.2.1 <i>Función puesta en remoto</i>	14
3.2.2 <i>Función para apagar la planta</i>	16
3.3 <i>Modelos de Simulink</i>	18
4 Puesta en funcionamiento de la planta en configuración 3	21
4.1 <i>Nuevo sensor de nivel</i>	21
4.2 <i>Planta de frío</i>	22
4.3 <i>Configuración 3</i>	23
4.4 <i>Puesta en funcionamiento de la configuración 3</i>	24
5 Identificación de la configuración 3	27
5.1 <i>Punto de funcionamiento</i>	27
5.3 <i>Ensayos de identificación</i>	29
5.3.1 <i>Ensayo VR1</i>	31
5.3.2 <i>Ensayo R</i>	32
5.3.3 <i>Ensayo VR2</i>	33
5.4 <i>Representación en variables de estado de un sistema de control</i>	35
5.4 <i>Modelos obtenidos</i>	35
5.4.1 <i>Modelo del efecto de VR1</i>	36
5.4.2 <i>Modelo del efecto de VR2</i>	36

5.4.3	Modelo del efecto de R	37
6	Diseño del control y simulación del modelo	39
6.1	<i>Modelo lineal en espacio de estados</i>	39
6.1.1	Controlabilidad del sistema	40
6.1.2	Observabilidad del sistema	40
6.1	<i>Estructura del modelo</i>	41
6.1.1	Controlador K	41
6.1.2	Observador de Luenberger	41
6.3	<i>Principio de separación</i>	42
6.4	<i>Cálculo de la ganancia K</i>	42
6.5	<i>Cálculo de ganancia L</i>	44
6.6	<i>Obtención de referencias a través de un SSTO</i>	45
6.6.1	Análisis de estados en cambios de referencia	46
6.6.2	Análisis de la incorporación de TT5 en el SSTO	48
6.7	<i>Mejora de la caracterización de TT5</i>	49
6.8	<i>Resultados de la simulación completa</i>	51
7	Implementación del control en la planta real	55
7.1	<i>Modelo Simulink</i>	55
7.2	<i>Ensayo con señales de control relativas a 0</i>	57
7.3	<i>Ensayo en el punto de funcionamiento</i>	58
7.4	<i>Ensayo ante cambios en la referencia</i>	59
8	Conclusiones y propuestas de mejora	61
8.1	<i>Conclusiones</i>	61
8.2	<i>Propuestas de mejora</i>	61
Anexos		63
	<i>Anexo A</i>	63
	<i>Anexo B</i>	65
	<i>Anexo C</i>	66
	<i>Anexo D</i>	67
Referencias		69

ÍNDICE DE TABLAS

Tabla 1.1 Variables de trabajo para la comunicación	4
Tabla 3.1 Pantallas Magelis	13
Tabla 3.3 Variables puesta en remoto OPC	14

ÍNDICE DE FIGURAS

Figura 1.1 Esquema de la planta	2
Figura 1.2 Esquema configuración 2	3
Figura 2.2.1 Cambio IP Unity 1	6
Figura 2.2 Cambio IP Unity 2	7
Figura 2.3 Cambio IP Unity 3	7
Figura 2.4 Configuración de la IP en el PC 1	8
Figura 2.5 Configuración de la IP en el PC 2	9
Figura 2.6 Configuración de la IP en el PC 3	9
Figura 2.7 Configuración de la IP en el PC 4	10
Figura 2.8 Configuración Configuration Tool 1	10
Figura 2.9 Configuración Configuration Tool 2	11
Figura 2.10 Configuración Configuration Tool 3	11
Figura 2.11 Configuración Configuration Tool 4	12
Figura 2.12 Configuración Configuration Tool 5	12
Figura 2.13 Servidor OPC UA	13
Figura 2.14 Cliente OPC UA	13
Figura 2.15 Acceso a variables en el cliente OPC UA	14
Figura 2.16 Escritura de variables en cliente OPC UA	14
Figura 2.17 Error por IP fuera de dominio	15
Figura 2.18 Asistente de instalación del EcoStruxure	15
Figura 3.1 Pantalla Magelis 1	12
Figura 3.2 Pantalla Magelis 2	12
Figura 3.3 Pantalla Magelis 3	13
Figura 3.4 Pantalla Magelis 4	13
Figura 3.5 Pantalla Magelis 5	13
Figura 3.6 Pantalla Magelis 6	13
Figura 3.7 Dos variables en paralelo que activan la misma transición	14
Figura 3.8 Bloque de selección modificado	14
Figura 3.9 Camino por la red de Petri global para la puesta en remoto	15
Figura 3.10 Camino por la red de Petri del control	16
Figura 3.11 Camino por la red de Petri global para el apagado	17
Figura 3.12 Modelo simplificado de conexión con la planta	18
Figura 3.13 Bloque de comunicación por OPC	19
Figura 3.14 Cuadro de parámetros del bloque de sincronización	19

Figura 4.1 Sensor de nivel instalado	21
Figura 4.2 Ensayo de calibración del sensor de nivel	22
Figura 4.3 Temperatura del agua de la planta de frío	22
Figura 4.4 Esquema configuración 3	23
Figura 4.5 Válvula VS5	24
Figura 4.6 Llave de paso al depósito colector	24
Figura 4.7 Extracción de aire de la bomba	25
Figura 5.1 Ensayo en el punto de funcionamiento	27
Figura 5.2 TT5 durante el ensayo en el punto de funcionamiento	28
Figura 5.3 ensayo en el punto de funcionamiento corregido	29
Figura 5.4 Ensayos en escalón de los actuadores	30
Figura 5.5 Ensayo CHIRP global	31
Figura 5.6 Ensayo CHIRP VR1	31
Figura 5.7 Ensayo CHIRP VR1 comprimido	32
Figura 5.8 Ensayo CHIRP R	33
Figura 5.9 Ensayo con señal seno en VR2	34
Figura 5.10 Ensayo CHIRP VR2 modificado	34
Figura 5.11 Concepto de máquina de estados Fuente [7]	35
Figura 5.12 Diagrama de flujo de un sistema en variables de espacio de estados. Fuente [6]	35
Figura 5.13 Modelo TT2 y TT4 frente a VR1 y TT5	36
Figura 5.14 Modelo TT2 y TT4 frente a VR2 y TT5	37
Figura 5.15 Modelo TT2 y TT4 frente a R y TT5	38
Figura 6.1 Estructura bucle de control	41
Figura 6.2 Bloque del observador	42
Figura 6.3 Bucle de control por estados del sistema	43
Figura 6.4 Respuesta del sistema controlado por estados	43
Figura 6.5 Respuesta del sistema con observador de Luenberger	44
Figura 6.6 Respuesta para distintas referencias del SSTO	46
Figura 6.7 Estados reales planta lineal	46
Figura 6.8 Estados reales planta lineal	47
Figura 6.9 Estados reales planta lineal	47
Figura 6.10 Respuesta del sistema sin introducir TT5 en el SSTO	48
Figura 6.11 Respuesta del sistema introduciendo TT5 en el SSTO	48
Figura 6.12 Forma de onda de TT5	49
Figura 6.13 Señal TT5 generada	49
Figura 6.14 Respuesta del sistema introduciendo TT5 en el SSTO interpolada	50
Figura 6.15 Salida del sistema con la perturbación mejorada	50
Figura 6.16 Modelo del bucle de control completo	51

Figura 6.17 Respuesta del sistema simulado	52
Figura 6.18 Respuesta del sistema simulado	52
Figura 7.1 Modelo Simulink para los ensayos	55
Figura 7.2 Subsistema modelo lineal	56
Figura 7.3 Subsistema modelo modelo real y lineal	56
Figura 7.4 Salidas relativas con señal de control relativa nula	57
Figura 7.5 Salidas en el punto de funcionamiento	58
Figura 7.6 Señales de control relativas	58
Figura 7.7 Salidas relativas ante cambios en la referencia	59
Figura 7.8 Comparación salidas del modelo lineal y de la planta real	59
Figura 7.9 Señales de control calculadas	60

A^T	Traspuesto
e	número e
sen	Función seno
$\partial y \partial x$	Derivada parcial de y respecto
x°	Notación de grado, x grados.
<	Menor o igual
>	Mayor o igual
\	Backslash
\Leftrightarrow	Si y sólo si
OPC	OLE for Process Control
OLE	Object Linking and Embedding
PLC	Programmable Logic Controller
PID	Controlador Proporcional-Integral-Derivativo
SSTO	Steady-State Target Optimization
SISO	Single input single output
MIMO	Multiple input multiple output

1 INTRODUCCIÓN

La planta multiprocesos del Laboratorio de Control es una máquina hidráulica diseñada para la implementación y estudio de controladores automáticos gracias a las amplias posibilidades que ofrece en cuanto variables controlables y controladas.

1.1 Motivación y objetivos

Como se verá a continuación, la planta se comunica con el usuario bien por una pantalla táctil situada en el frontal de la propia planta o bien a través de un protocolo OPC TCP/IP que permite el control de la misma a través de distintos clientes desde el ordenador anexo a la planta. A lo largo de este trabajo se desarrollará el camino seguido para que el control remoto de la planta deje de depender de la pantalla táctil y para, a través de un servidor OPC UA poder acceder a su control desde cualquier dispositivo de la red Ethernet del laboratorio. De esta manera también queda independiente del ordenador anexo a la planta.

Se va realizar la integración del nuevo sensor de nivel dejándolo operativo de ahora en adelante en la planta.

Posteriormente se va a realizar un control sobre la tercera configuración de la planta. Para ello, se desarrollará una identificación dinámica del sistema. Dicha configuración no ha sido todavía usada por lo que se tratará de obtener un modelo en variables de espacio de estados que refleje el comportamiento del sistema real. Con él será posible la implementación de los algoritmos y técnicas de control estudiados a lo largo del grado

Esta implementación del control automático en un sistema real hará que salgan a la luz diversidad de problemas que en el marco teórico no aparecen. Las soluciones a estos inconvenientes quedan reflejadas a lo largo de los distintos capítulos de los que consta este trabajo.



1.2 Descripción general y esquema de la planta

La planta multiprocesos es un equipo de laboratorio de carácter hidráulico orientado a la ilustración y experimentación académica de metodologías de control de procesos con variables fundamentales como el nivel de agua en un tanque, caudal o temperatura a lo largo de diversas fases del sistema.

El control sobre la planta es posible gracias al PLC Modicon M340 de Schneider Electric y a la pantalla Magelis instalados. El código importado en ambos dispositivos y su conexión por OPC al OPC Factory Server del ordenador anexo a la planta fue realizado por José Enrique Alonso en su Trabajo de Fin de Grado en 2013 [1]. Dicho Código ha sido revisado por José Luis Pozo en 2015 [2].

Se le ha instalado un nuevo sensor de nivel cuya calibración se detallará en capítulos posteriores. Su control remoto por comunicación OPC se realizaba a través de una router propio conectado al ordenador instalado anexo a la planta, lo que también será actualizado a lo largo del trabajo.

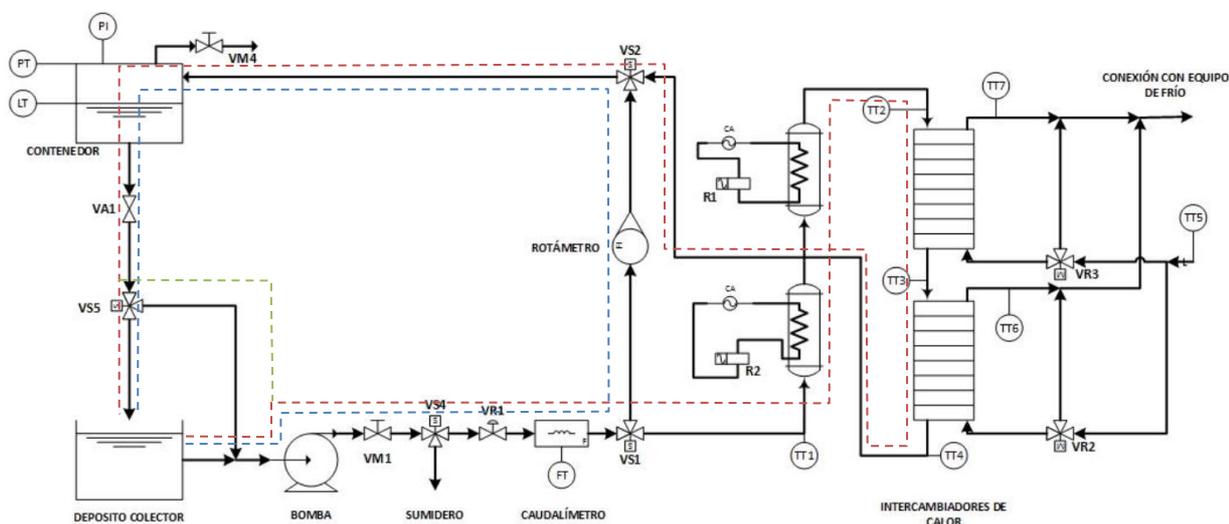


Figura 1.1 Esquema de la planta

Este es el esquema general de los circuitos posibles que ofrece la planta en sus distintas configuraciones. En el quedan representados los caminos por los que se dirige el caudal de agua impulsado por la bomba hidráulica que deja pasar la válvula VR1 recorriendo los distintos sensores de temperatura (desde TT1 hasta TT9) cuyos valores servirán de objeto de análisis en los capítulos posteriores.

El trazo azul de la figura 1.2 corresponde a la configuración 1 de la planta. En ella al colocar la válvula solenoide VS1 en la posición correspondiente, el caudal de agua sale hacia el rotámetro para luego llegar a al tanque. Tras ello, al pasar por la válvula manual VA1 retorna de nuevo al depósito colector del que se alimenta la bomba.

Es sin duda, la configuración más sencilla, en ella la única variable manipulable es la apertura de la válvula tras la bomba hidráulica que propulsa el agua a través del circuito. Las variables para controlar serán el caudal FT proporcionado por el caudalímetro y el nivel de agua en el tanque LT medido por el nuevo sensor del nivel instalado en la planta.

La configuración 2, representada por el trazo naranja, entran en escena las etapas de enfriamiento y calentamiento del caudal llevadas a cabo por las resistencias y los intercambiadores de calor.

Con la válvula de solenoide VS1 cambiada de posición se accede al circuito de resistencias e intercambiadores

de frío de la planta.

En primer lugar, se pasa por las dos resistencias eléctricas instaladas en serie, el salto térmico que el caudal experimenta al paso por ellas se puede medir como la resta entre la temperatura a la salida y a la entrada de estas, TT2 y TT1 respectivamente.

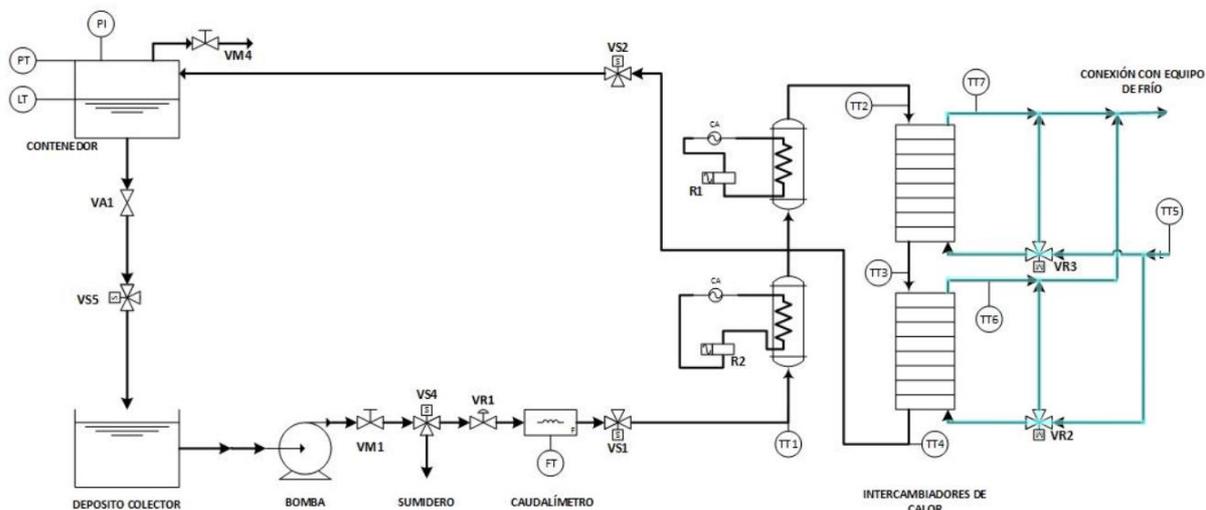


Figura 1.2 Esquema configuración 2

Tras ellas sucede la etapa de los intercambiadores de calor, estos dispositivos están alimentados por una planta de frío la cual aporta un caudal de entre 4 °C y 8 °C. En el esquema se representa este circuito de agua fría, independiente respecto al circuito propio de la planta, con un trazo de color azul. La variable manipulable que controla el funcionamiento de los intercambiadores es la posición de las válvulas de tres vías VR2 y VR3. Para caracterizar esta etapa hay de nuevo, diversos sensores de temperatura en las distintas fases por la que circula el agua: TT3 y TT4. Para evaluar el comportamiento de la planta de frío se dispone también de termómetros a la entrada del caudal frío TT5 y a la salida de los intercambiadores de calor TT6 y TT7.

Tras pasar por los intercambiadores se vuelve al circuito original a través de la válvula solenoide VS2. A partir de este punto se pasan por las mismas etapas que en la configuración 1 con la lectura de nivel LT.

La tercera configuración, como se puede ver en el esquema, es bastante parecida a la configuración 2. Sin embargo, el cierre del circuito por la válvula VS5 hace que el comportamiento del sistema se modifique considerablemente.

Al cerrarse el circuito el caudal estará en constante recirculación. Esto hará que el control de nivel queda inutilizado ya que, en este caso, será fijo independientemente del valor del caudal que fluya por el circuito. Por otro lado, el resto de los actuadores y sensores trabajan de manera igual a como lo hacían en la configuración 2.

1.2.1 Correspondencia de las variables

Para mayor facilidad en la comprensión de los códigos y del resto de capítulos en adelante se presenta una tabla con las variables con las que se intercambian los datos de lectura y escritura de sensores y actuadores.

Apertura bomba VR1	OPC01_r_VR1
Posición VR2	OPC02_r_REF_VR2
Posición VR3	OPC03_r_REF_VR3
Duty cycle de la Resistencia 1	OPC04_r_R1
Duty cycle de la Resistencia 2	OPC05_r_R2
Caudal FT	OPC12_w_FT
Nivel LT en el tanque	OPC44_w_LT_puntos
TT1	OPC13_w_TT1
TT2	OPC14_w_TT2
TT3	OPC15_w_TT3
TT4	OPC16_w_TT4
TT5	OPC17_w_TT5
TT6	OPC18_w_TT6
TT7	OPC19_w_TT7

Tabla 1.1 Variables de trabajo para la comunicación

1.3 Estructura del trabajo

Este trabajo consta de dos partes con objetivos distintos y las cuáles se van basando en los resultados obtenidos en las anteriores.

En la primera, va a tener lugar una actualización en la comunicación de la planta introduciendo el protocolo OPC UA y se expondrá los pasos a seguir para el uso de un servidor de este tipo desde cualquier dispositivo conectada a la red de Ethernet del laboratorio. Tras ello, se diseñará un algoritmo que permita poner la planta en modo remoto y controlada por sus variables OPC sin necesidad de usar la pantalla Magelis instalada. Una vez completada esta fase del trabajo la planta ya no será dependiente de la pantalla Magelis ni del ordenador al que previamente estaba conectado evitando así posibles inconvenientes y problemas derivados de dichos aparatos. Esta parte constará de los capítulos 2 y 3 de este Trabajo.

Tras la primera parte, se realizará una descripción completa de la planta para poder ilustrar el entorno de trabajo de los capítulos que siguen a continuación

La segunda parte será el diseño de un sistema de control que consiga controlar unas temperaturas seleccionadas a través de ciertos actuadores en la configuración 3 de la planta. Comienza en el capítulo quinto y en él se llevará a cabo una identificación del sistema.

En el capítulo sexto se trabajará en el modelo lineal obtenido a través de simulaciones y el séptimo se aplicará la estrategia de control diseñada con las simulaciones en la planta real y se analizarán los datos obtenidos.

2 IMPLEMENTACIÓN DE LA COMUNICACIÓN A TRAVÉS DE OPC UA

La comunicación entre el mundo real y la planta se lleva a cabo del estándar OPC (Object Linking and Embedding for Process Control). El estándar de comunicación OPC es una de las tecnologías de comunicación industrial más populares debido a su facilidad de conexión a software de distinta naturaleza. Es una solución flexible y abierta para el intercambio de información entre los distintos componentes de un sistema industrial en la que se evita el problema de incompatibilidades entre drivers propietarios de los dispositivos.



Figura 2.1 Logo OPC

OPC permite el intercambio de datos e información entre los distintos dispositivos de un sistema gracias a su arquitectura Cliente-Servidor. Cada elemento que desee tener acceso a los datos podrá conectarse al servidor y podrá leer y escribir los valores de las variables que se intercambian.

Hasta ahora se ha usado en la planta el OPC Factory Server, OFS en adelante, el cual es un software de servidor de datos proporcionado por Schneider Electric. El OFS pese a funcionar adecuadamente ligaba el control remoto sobre la planta al ordenador anexo a ella y dificultaba su conexión a otro ordenador ya que requiere Windows 7 para su funcionamiento.

2.1 OPC UA

Como ya se ha expuesto anteriormente, uno de los objetivos de este Trabajo de Fin de Grado era evitar la dependencia del ordenador instalado al lado de la planta. Parte de ese objetivo se conseguirá a través de la implementación de un sistema de comunicación OPC mejorado, el OPC UA (arquitectura unificada).

Si bien el estándar OPC resolvía los problemas de interoperabilidad entre dispositivos, no podía funcionar en todos los sistemas operativos que hay en la actualidad. De la revisión y mejora del OPC nace el OPC UA.



Figura 2.2 Logo OPC UA

OPC UA brinda una Plataforma neutral que funciona desde cualquier sistema operativo y está preparada para comunicar sistemas antiguos y actuales. OPC UA, que se desprende del protocolo DCOM, presenta también mejoras de rendimiento, conectividad y visibilidad lo que le hace prolongar su éxito en la industria y en sistemas de intercambio de datos.

Precisamente la capacidad de poder ser utilizado desde un sistema operativo actual es el motivo principal por el que se ha decidido apostar por esta tecnología de comunicación.

Para ello se usará el EcoStruxure™ OPC UA Server Expert, una plataforma de comunicaciones OPC UA que permite conectar el PLC Modicon a los distintos clientes que el usuario configure. A continuación, se entrará en detalle de cómo instalar y conectar el ordenador desde el que desee trabajar con la planta.

2.2 Conexión del hardware de la planta a la red Ethernet

Tanto el PLC como la pantalla táctil con la que funciona la planta disponían de un router que se conectaba al ordenador anexo. Para conectar dichos dispositivos a la red de Ethernet del laboratorio se han de hacer diversas modificaciones en los códigos exportados de manera que la comunicación sea posible.

2.2.1 Cambio dirección IP PLC

Por un lado, la dirección IP que usaba el PLC de la planta era local del router privado al que estaba conectado. Esto ahora cambia ya que se usará la dirección IP correspondiente al puerto del router de todo el Laboratorio de Control.

Para ello dentro del archivo .stu de Untiy con el que trabaja el PLC debemos hacer las siguientes modificaciones:

Primero, seleccionando la carpeta Redes, se agrega la red Ethernet a través la cual se comunicará el PLC.

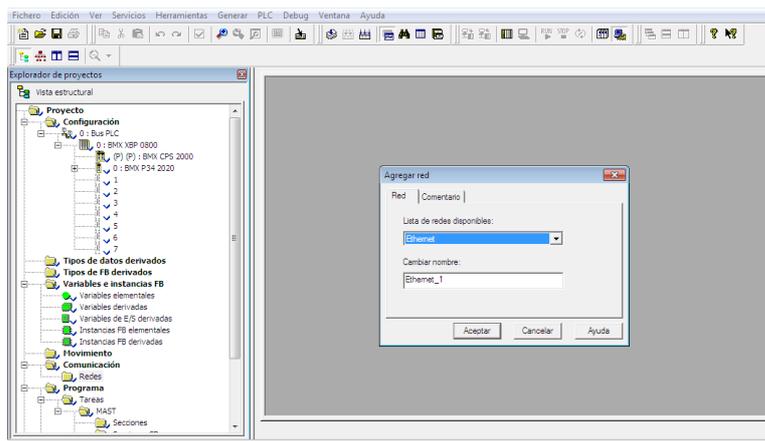


Figura 2.1 Cambio IP Unity 1

Una vez la red esté disponible, se selecciona en la representación de los módulos del PLC el puerto Ethernet.

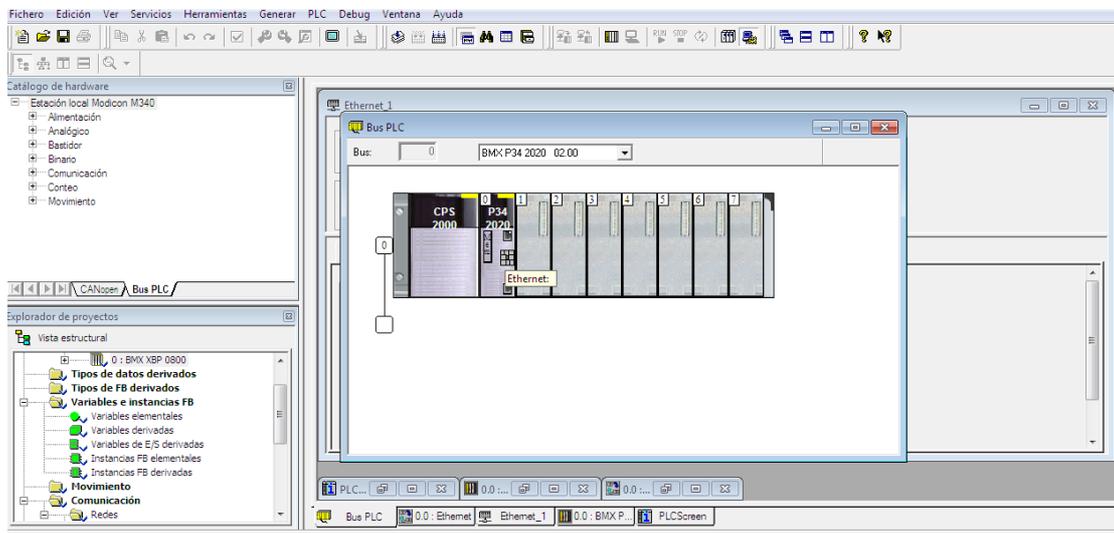


Figura 2.2 Cambio IP Unity 2

Tras ello se abrirá la siguiente ventana de opciones.

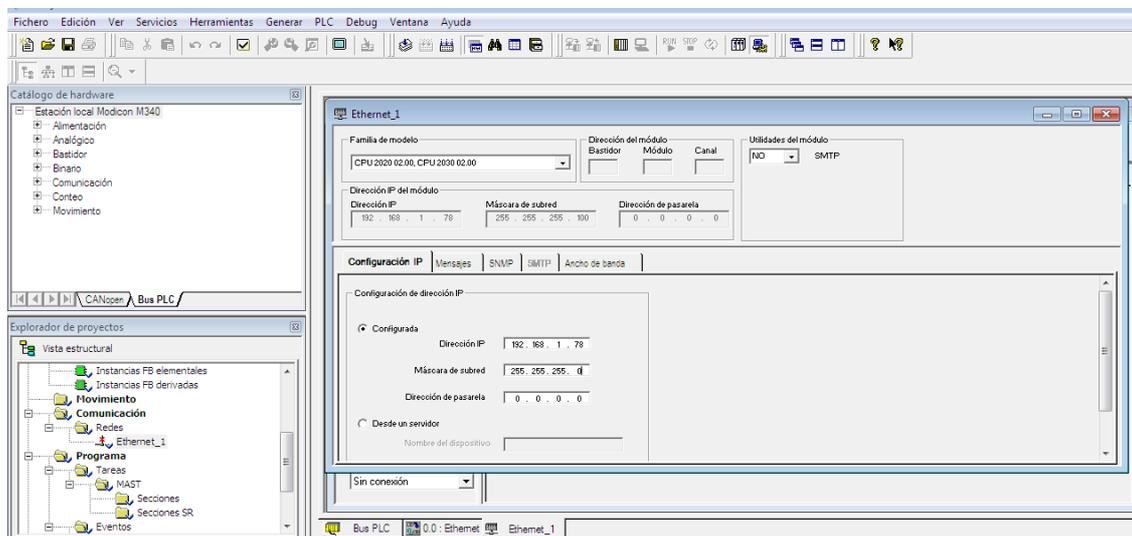


Figura 2.3 Cambio IP Unity 3

En ella se selecciona la familia del modelo, se introducen la dirección IP correspondiente a la nueva ubicación del PLC en la red Ethernet en este caso 192.168.1.78 y la máscara de subred 255.255.255.0

2.2.2 Cambio dirección IP pantalla Magelis

Del mismo modo que el PLC contaba con su dirección local y particular en el router también lo hacía la pantalla táctil instalada. Al igual que en el caso anterior, se modificará el archivo .xvz exportado en el dispositivo.

2.3. Configuración y conexión del servidor EcoStruxure OPC UA server Expert

Al igual que se ha hecho con el PLC y la pantalla Magelis, el ordenador desde el que queremos controlar la planta también ha de ser ubicado dentro de la red Ethernet del laboratorio. Si se trata de uno de los ordenadores instalados en el Laboratorio esta configuración puede no ser necesaria ya que por defecto cuentan con la dirección IP correcta dentro del router global.

2.3.1 Cambio dirección IP del PC

En el caso de utilizar un ordenador externo, es necesario saber a qué dirección IP del router del laboratorio se va a conectar, en el ejemplo con el que se ilustrará este capítulo usará la dirección IP 192.168.1.90 correspondiente a uno de los puertos Ethernet del router.

Para modificar la dirección con la que el ordenador se conecta a la red se deberán seguir los siguientes pasos.

En primer lugar, se accede al panel de control, ver estado y tareas de red. Tras ello se selecciona el campo 'Ethernet' tal y como se ve en la figura 2.4.

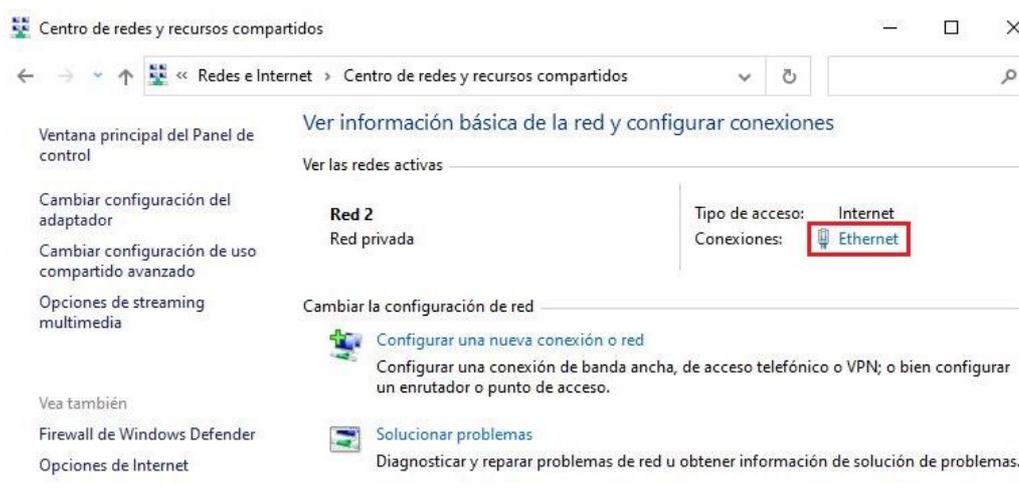


Figura 2.4 Configuración de la IP en el PC 1

Una vez allí vemos las propiedades de la conexión.

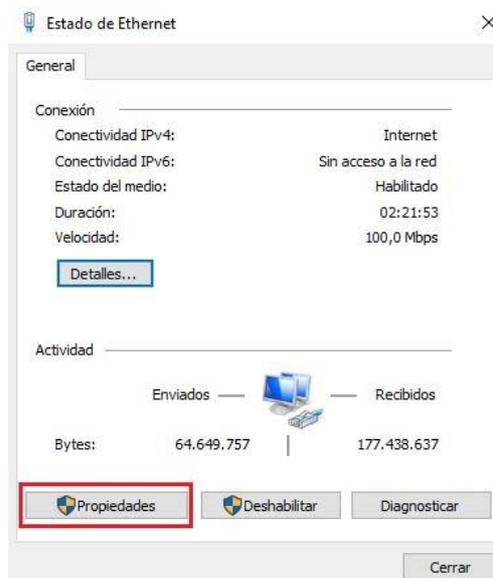


Figura 2.5 Configuración de la IP en el PC 2

Dentro de los elementos que usa la conexión Ethernet hacemos doble clic en el protocolo de internet TCP IPv4 que es el protocolo de conexión que usará el servidor para comunicarse.

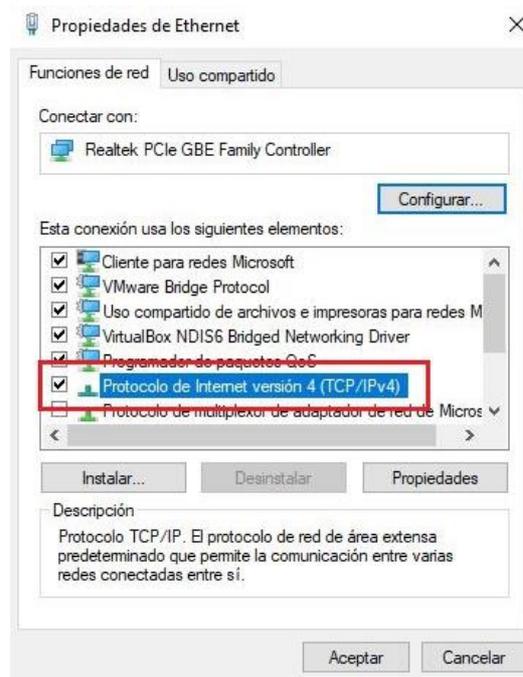


Figura 2.6 Configuración de la IP en el PC 3

Finalmente, en el cuadro de propiedades del protocolo seleccionado se rellenan los elementos resaltados. Por un lado, la dirección IP que se usará, la máscara de subred del router y la puerta de enlace predeterminada.

También se escriben los servidores DNS por preferidos. En el caso de trabajar desde el router del Laboratorio de Control se pueden usar los mismos valores para la máscara de subred, la puerta de enlace y los servidores DNS preferidos y alternativos.

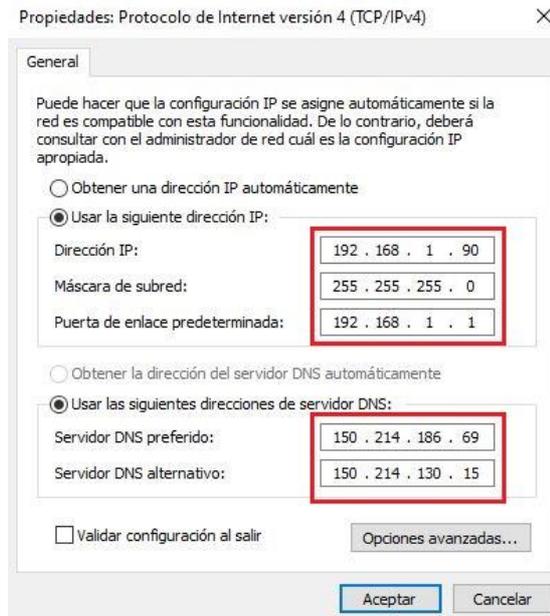


Figura 2.7 Configuración de la IP en el PC 4

2.3.2 Instalación y configuración EcoStruxure OPC UA server Expert

Una vez todas las direcciones IP de los dispositivos implicados sean las correctas pasamos al descargar el software necesario para crear el servidor OPC UA.

Al iniciar el ejecutable e instalarlo por defecto se instalarán tres programas del entorno EcoStruxure: el programa de configuración ‘configuration tool’, el servidor y un cliente.

Se comenzará por los pasos a seguir para configurar el servidor en la aplicación ‘configuration tool’.

Dentro de la aplicación se deberá darle un nombre al dispositivo PLC al que se desea acceder desde el ordenador del usuario.

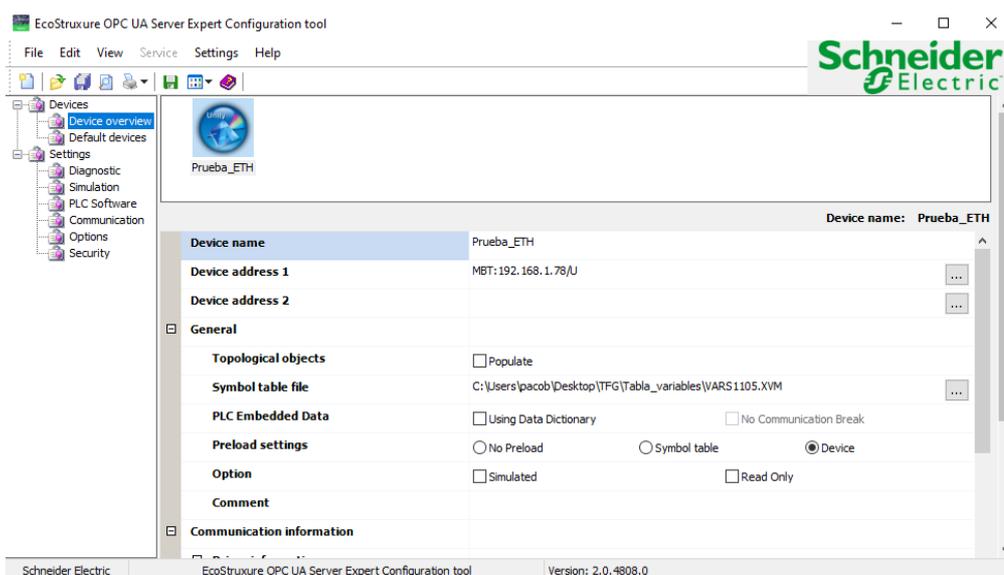


Figura 2.8 Configuración Configuration Tool 1

Tras ello, pulsando los tres puntos suspensivos en el campo ‘Device address1’ se introduce la dirección IP del

PLC que, tal y como colocamos en el archivo del Unity Pro, actualmente es 192.168.1.78.

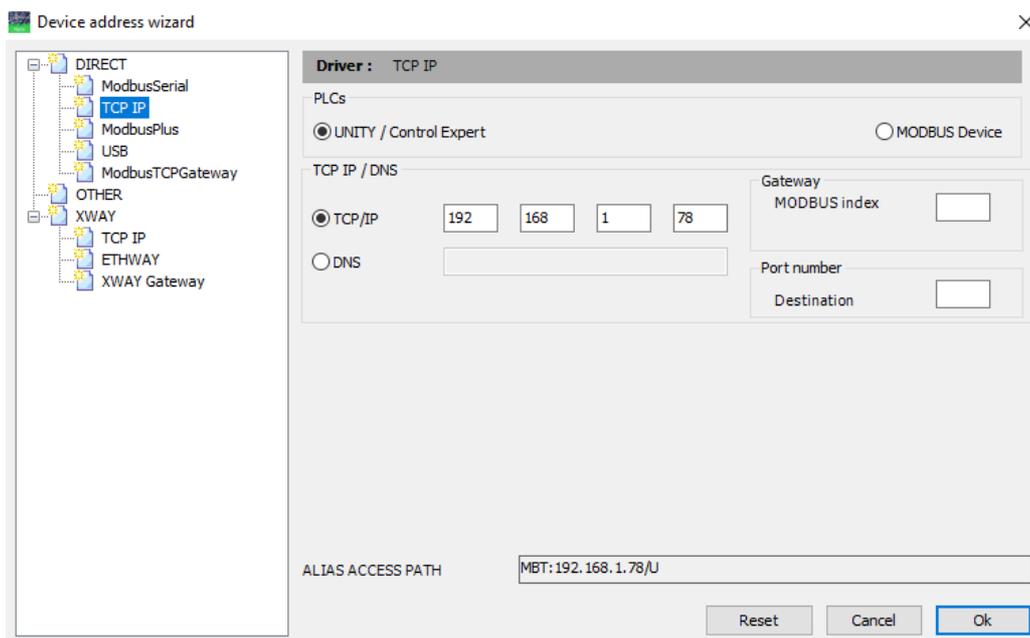


Figura 2.9 Configuración Configuration Tool 2

Se debe direccionar la tabla de variables exportada del proyecto de Unity Pro con las variables que se intercambiarán a través del servidor.

El resto de los parámetros de la primera ventana se colocan tan y como se representa en la figura 2.7

En Settings, PLC Software se deberá introducir el ciclo de chequeo de conexión entre el servidor y el PLC. El resto de los parámetros se dejan por defecto.

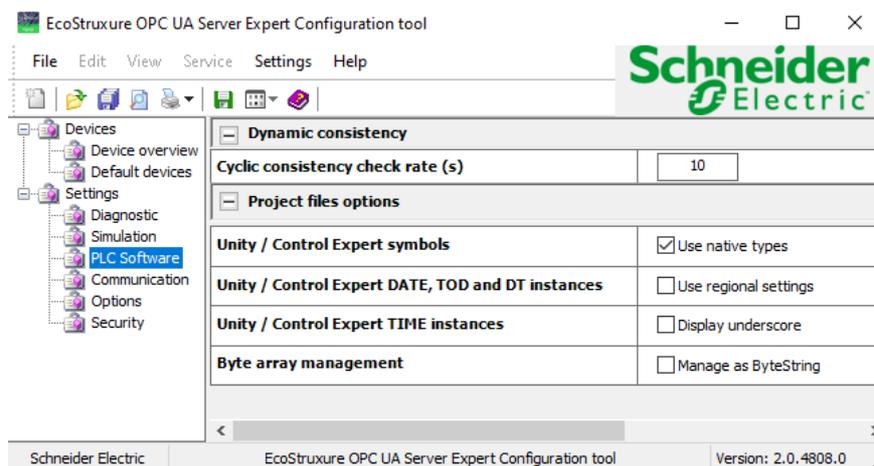


Figura 2.10 Configuración Configuration Tool 3

A continuación, en la ventana de Options, se marca la casilla de DNS scanning TCP/IP y se elige uno de los puertos del ordenador para la recepción de datos TCP. Por defecto viene uno de los puertos del ordenador y se deberá comprobar si está disponible o no, en el caso de no estarlo, bastará con probar con cualquier otro hasta que uno resulte disponible.

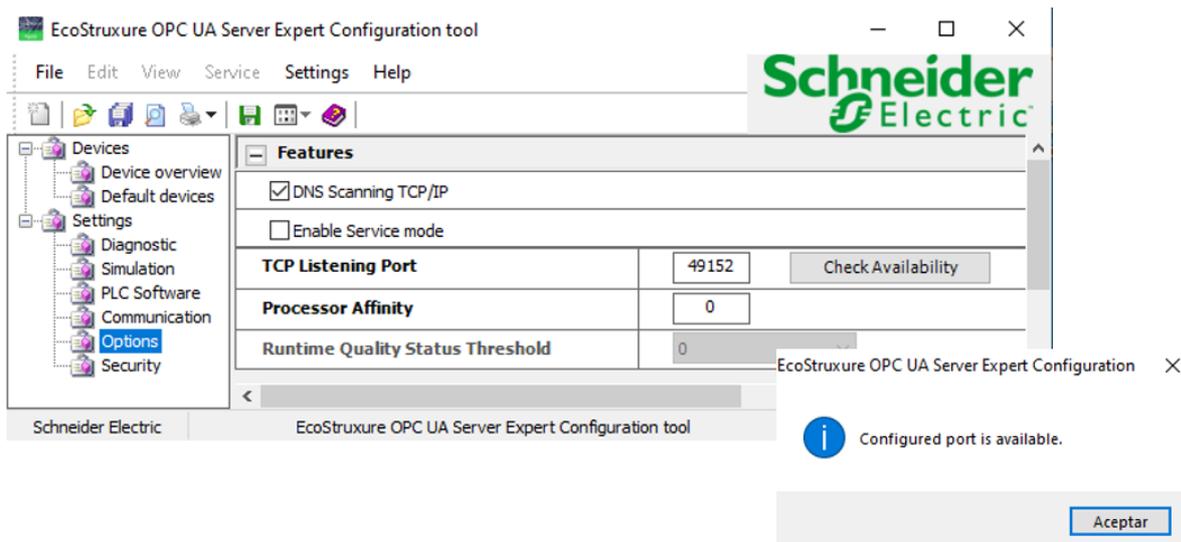


Figura 2.11 Configuración Configuration Tool 4

En último lugar se seleccionan las opciones de seguridad del servidor, como se trabajará en una red conocida y se aceptarán conexiones sin protocolo de seguridad y anónimas.

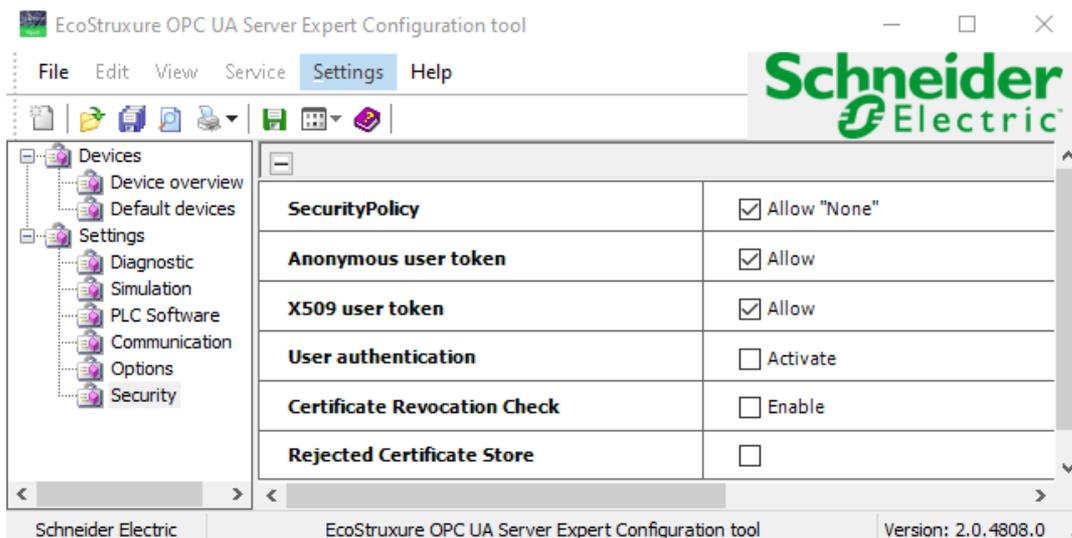


Figura 2.12 Configuración Configuration Tool 5

Tras estos pasos, el servidor está ya configurado para su uso.

2.3.3 Uso de conexión Cliente-Servidor

Arrancar el servidor será tan sencillo como iniciar la aplicación OPC UA Server Expert. Al abrirla, podemos ver la dirección del servidor y los mensajes de diagnóstico de este.

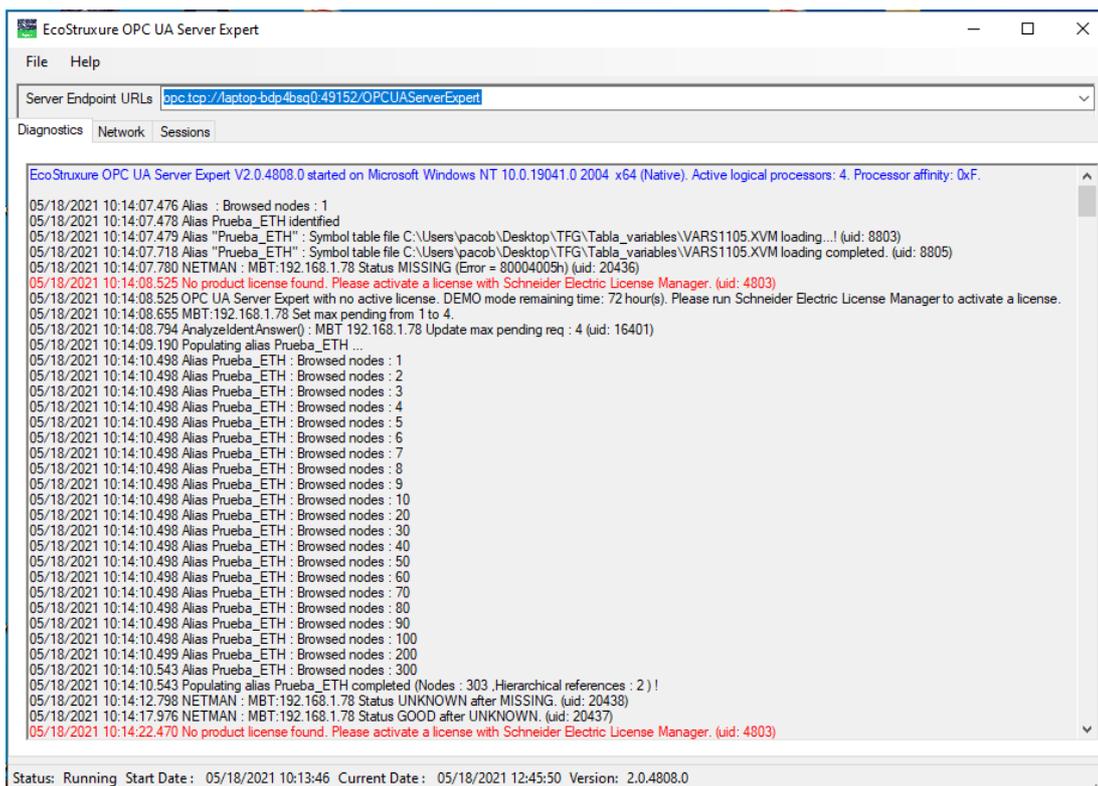


Figura 2.13 Servidor OPC UA

La dirección del servidor será con lo que ubicaremos el servidor desde los clientes. En ella se encuentran el protocolo de internet utilizado, el nombre del dispositivo donde se aloja el servidor y el puerto por el que se comunica.

La figura anterior corresponde a lo que deberá salir una vez se inicialice el servidor de manera correcta. Se ve cómo se identifica el alias “prueba_ETH”, se localiza la tabla de variables y se chequean los nodos.

Si en el diagnóstico recibimos el mensaje de ‘Status GOOD after MISSING’ y vemos como los nodos han sido chequeados, tenemos el servidor listo para su uso.

Para conectar el cliente del entorno EcoStruxure basta con abrirlo y en el único campo que se muestra por pantalla debemos introducir la dirección del servidor. Al escribirla pulsamos ‘connect’ se abrirá una nueva ventana en la que no se seleccionarán ningún modo de encriptamiento y entraremos como ‘Anonymous’.

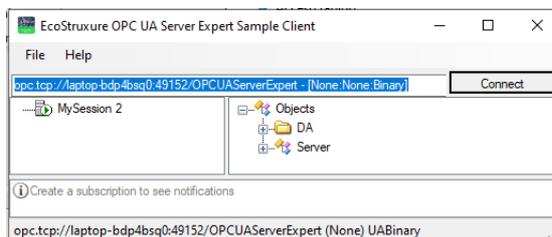


Figura 2.14 Cliente OPC UA

Esta es la imagen que aparece una vez el cliente está conectado.

Desde el cliente tenemos acceso a las variables intercambiadas con permiso de lectura y escritura, para manipularlas, se selecciona unas de las variables y se modifica su valor.

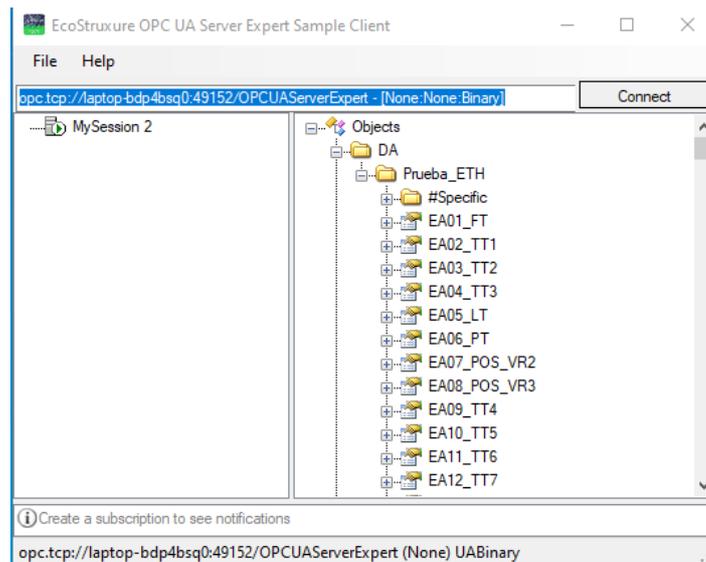


Figura 2.15 Acceso a variables en el cliente OPC UA

Para ello basta con pulsar botón derecho en la variable que se desee y seleccionar ‘Read’ o ‘Write’. Con ello se abrirá la siguiente ventana desde la que tenemos accesos a modificar el valor de la variable.

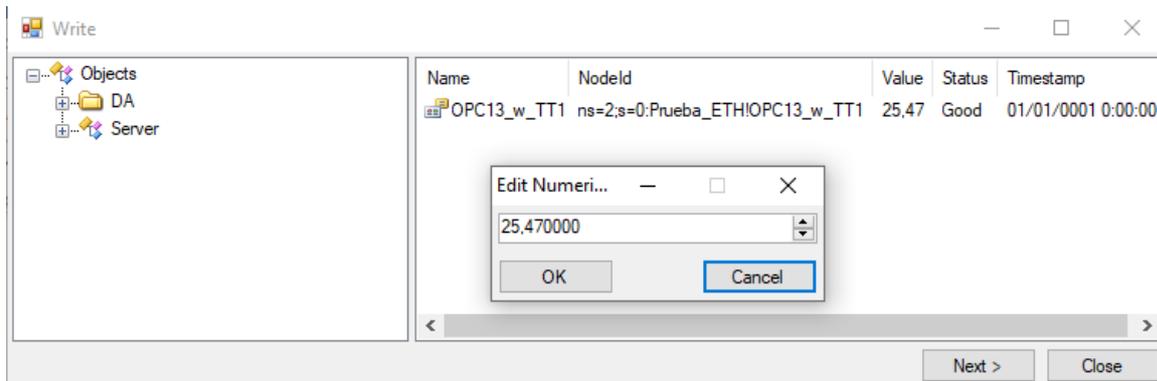


Figura 2.16 Escritura de variables en cliente OPC UA

Con la planta en modo remoto con la pantalla Magelis se podrá chequear que todo está bien conectado modificando uno de los actuadores, si, al cambiarle el valor a OPC_w_32_VR1 a un valor superior a 0 la bomba empieza a propulsar agua, la implementación se ha hecho de manera correcta y se puede avanzar a la siguiente etapa.

Es importante destacar antes de avanzar, que, si se trata de abrir el servidor desde una dirección IP diferente a la que se usó en la última vez que se arrancó el servidor, saldrá el siguiente cuadro de error.



Figura 2.17 Error por IP fuera de dominio

Para solucionarlo, bastará con volver a iniciar el ejecutable con el que se realizó la instalación en un primer lugar. Al avanzar en el asistente de instalación se llega a la siguiente ventana.

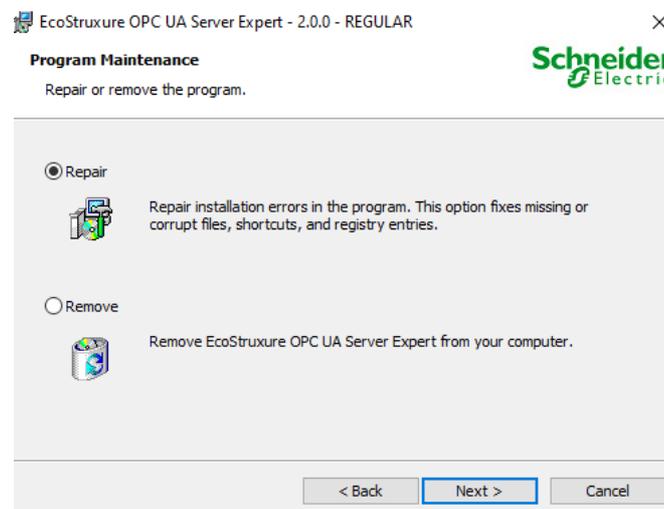


Figura 2.18 Asistente de instalación del EcoStruxure

Con la opción de 'Repair' marcada se pulsa 'Next' y el servidor quedará configurado de nuevo con la nueva dirección en la que está ahora conectada.

3 CONTROL DE LA PLANTA DESDE MATLAB

Si bien ya con el cliente del EcoStruxure tenemos acceso a las variables de la planta necesitamos un entorno en el que manejar fácilmente los actuadores y recopilar cómodamente la información de los sensores para su análisis. Para ello usaremos Matlab en su versión 2020b con su extensión Simulink.

Matlab tiene en su librería de Simulink bloques preparados para su conexión con servidores OPC. Sin embargo, no soportan conexiones por OPC UA. A través de comandos sí que es posible operar con OPC UA, por lo que se hará ahora en adelante, es ejecutar un script con los comandos necesarios para leer y escribir las variables que necesitemos. Las tareas de escritura y la recepción de datos se realizarán a través de los bloques típicos de un modelo Simulink.

3.1 Conexión del servidor a Matlab

Ejecutando la función ‘opcua’ al que se le pasa por argumento la dirección del servidor al que se desea conectarse como cadena de caracteres se crea un objeto en el Workspace categorizado como cliente, ‘Client’. A través de este objeto ‘Client’ quedará definido la conexión entre Matlab y el servidor.

```
uaClient=opcua('opc.tcp://laptop-bdp4bsq0:49152/OPCUAServerExpert');
```

Tras ello debemos configurar la seguridad con la que el cliente se conectará al servidor con la función ‘setSecurityModel’. Como argumento se le pasan el cliente recién creado y la cadena de caracteres que definirán la seguridad del cliente. En este caso, tal y como se configuró anteriormente en el servidor se puede acceder sin ningún modelo de seguridad por lo que es posible conectarlo escribiendo en dicho campo ‘None’.

```
setSecurityModel(uaClient, 'None')
```

Finalmente, con la función ‘connect’ conecta el cliente que se le pasa como argumento.

```
connect(uaClient)
```

Si la seguridad no está correctamente configurada o configurada por defecto, la conexión al servidor fallará no siendo posible la comunicación.

Ya está el servidor configurado, creado y conectado a Matlab. Para poder trabajar con las variables de manera cómoda debemos crear tantas variables globales como variables queramos leer o escribir. Para asociar las variables intercambiadas a través del OPC a las variables globales creadas se usará la función ‘findNodeByName’. A la función se le pasan como argumentos el nombre del espacio del cliente conectado, el nombre de la variable del servidor OPC y una cadena de caracteres que definirán la tipología del nodo que se está creando. En el caso de las variables se usará ‘-once’.

```
input1= findNodeByName(uaClient.Namespace, 'OPC01_r_VR1', '-once')
```

Ya con las variables correctamente inicializadas se tiene acceso a ellas tanto con permiso tanto de escritura como de lectura. Con las funciones readValue, se guardarán en el primer elemento de los tres que devuelve, el

valor de la variable asociada al nodo que se le mete como segundo argumento. Como primer argumento se introduce el cliente conectado al servidor al que pertenece la variable.

De manera parecida se trabaja con `writeValue` en el que se introducen como primer argumento el cliente al que se está haciendo referencia, el nodo correspondiente a la variable cuyo valor se desea modificar y el valor a colocar en la variable.

```
[valor_leido,~,~]=readValue(uaClient, output1);
writeValue(uaClient,input1,valor_escrito)
```

Estas funciones son las piezas esenciales que se ejecutarán en el código en cada iteración de la simulación. El script tipo que se ejecutará se encuentra en el Anexo A.

Se configurará dicho módulo para que tenga las entradas y salidas que se considere oportunos para cada caso. La comunicación por OPC tiene un Watchdog que chequea que la conexión se encuentra en buen estado, la entrada y la salida de la señal de Watchdog es necesaria para cualquier experimento que se realice remotamente. Si no está implementada el código del proyecto ejecutándose en el PLC reportará un fallo leve lo que hará que se pare las actuaciones

3.2 Funciones para inicio y apagado de la planta remotamente

Continuando en el camino de permitir el control completo de la planta de manera remota y ajena a la pantalla Magelis, se necesitan maneras para poder colocar la planta en el estado en el que el PLC permite el intercambio de datos a través de la comunicación OPC. Este corresponde al estado “OPC REMOTO” al que se accede a través de la tarea “G_F1_CONTROL” de la red de Petri general del sistema.

El acceso a estas tareas se realizaba hasta ahora a través de la sucesión de pantallas que activaba ciertas variables que ‘abren’ las transiciones. Las variables que acceden a los distintos paneles se representan en la siguiente tabla.

 <p>Figura 3.1 Pantalla Magelis 1</p>	<p>VJ25_EP_INTERRUPTOR</p>
 <p>Figura 3.2 Pantalla Magelis 2</p>	<p>VJint104_r_SELECCION_COFIG</p>

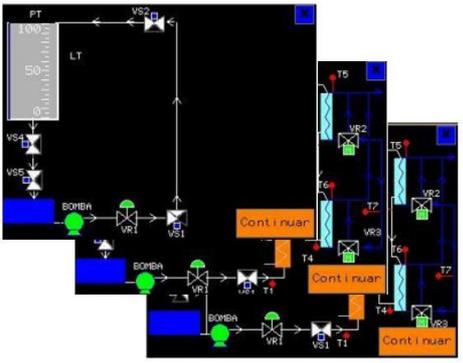
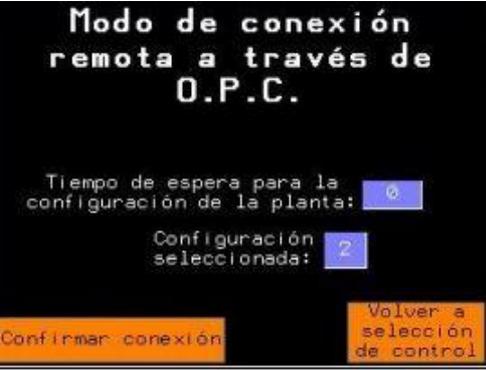
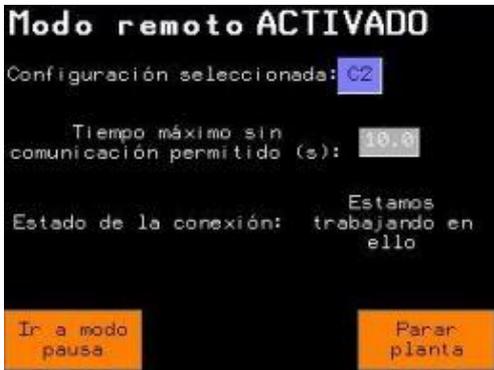
 <p>Figura 3.3 Pantalla Magelis 3</p>	<p>-</p>
 <p>Figura 3.4 Pantalla Magelis 4</p>	<p>VJ37_r_CONFIR_CONF_SELECC</p>
 <p>Figura 3.5 Pantalla Magelis 5</p>	<p>VJ29_r_atajo_profesores</p> <p>VJ38_r_REMOTO_OPC</p>
 <p>Figura 3.6 Pantalla Magelis 6</p>	<p>-</p>

Tabla 3.1 Pantallas Magelis

Para acceder al estado deseado sin la pantalla se han colocado en paralelo nuevas variables que se activarán en el orden adecuado para sustituir las activaciones que producen los toques en la pantalla. Estas adaptaciones del código se han hecho principalmente en la tarea VJEO_M340 (LD) de la guía GEMMA encargada en activar y desactivar transiciones a partir de las variables que manda la pantalla

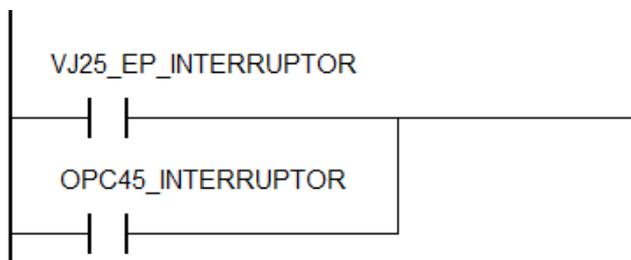


Figura 3.7 Dos variables en paralelo que activan la misma transición

Para la selección de configuración se ha tenido que adaptar la tarea GEM_F2_1_INICIO_PREPARACION en la que en esta ocasión se ha copiado el bloque de selección de configuración introduciendo en lugar de la variable de la pantalla la variable OPC creada.

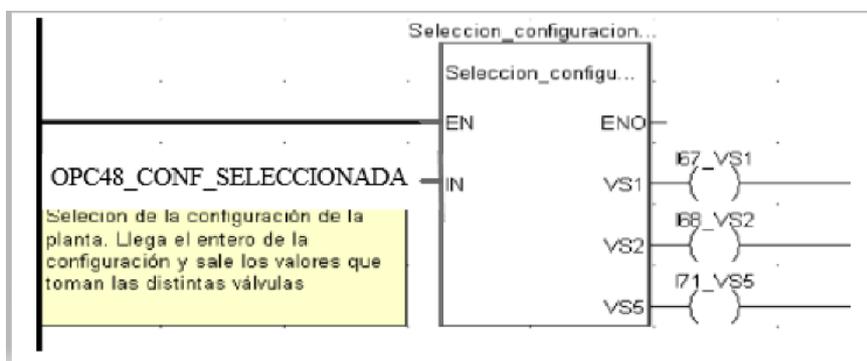


Figura 3.8 Bloque de selección modificado

Con estos cambios el sistema pasa por las etapas correspondientes hasta colocarse en la tarea de “CONTROL” de la guía GEMMA y, dentro de ésta, en la de “OPC_REMOTO”.

3.2.1 Función puesta en remoto

En la siguiente tabla se representan las variables que modifica la pantalla y su sustituta que será activada a través de la función de Matlab.

VJ25_EP_INTERRUPTOR	OPC45_INTERRUPTOR	Booleana
VJint104_r_SELECCION_COFIG	OPC48_CONF_SELECCIONADA	Entera
VJ37_r_CONFIR_CONF_SELECC	OPC46_CONF_COFIRMADA	Booleana
VJ29_r_atajo_profesores	OPC7_ATAJO_PREPARACION	Booleana
VJ38_r_REMOTO_OPC	OPC49_REMOTO_OPC	Booleana

Tabla 3.2 Variables puesta en remoto OPC

Pues bien, teniendo claro las asociaciones entre variables y la función de cada una de ellas se pueden activar las transiciones necesarias para pasar por los estados necesarios.

El camino que se debe seguir a través de los SFC del código de la planta se representa en las siguientes figuras.

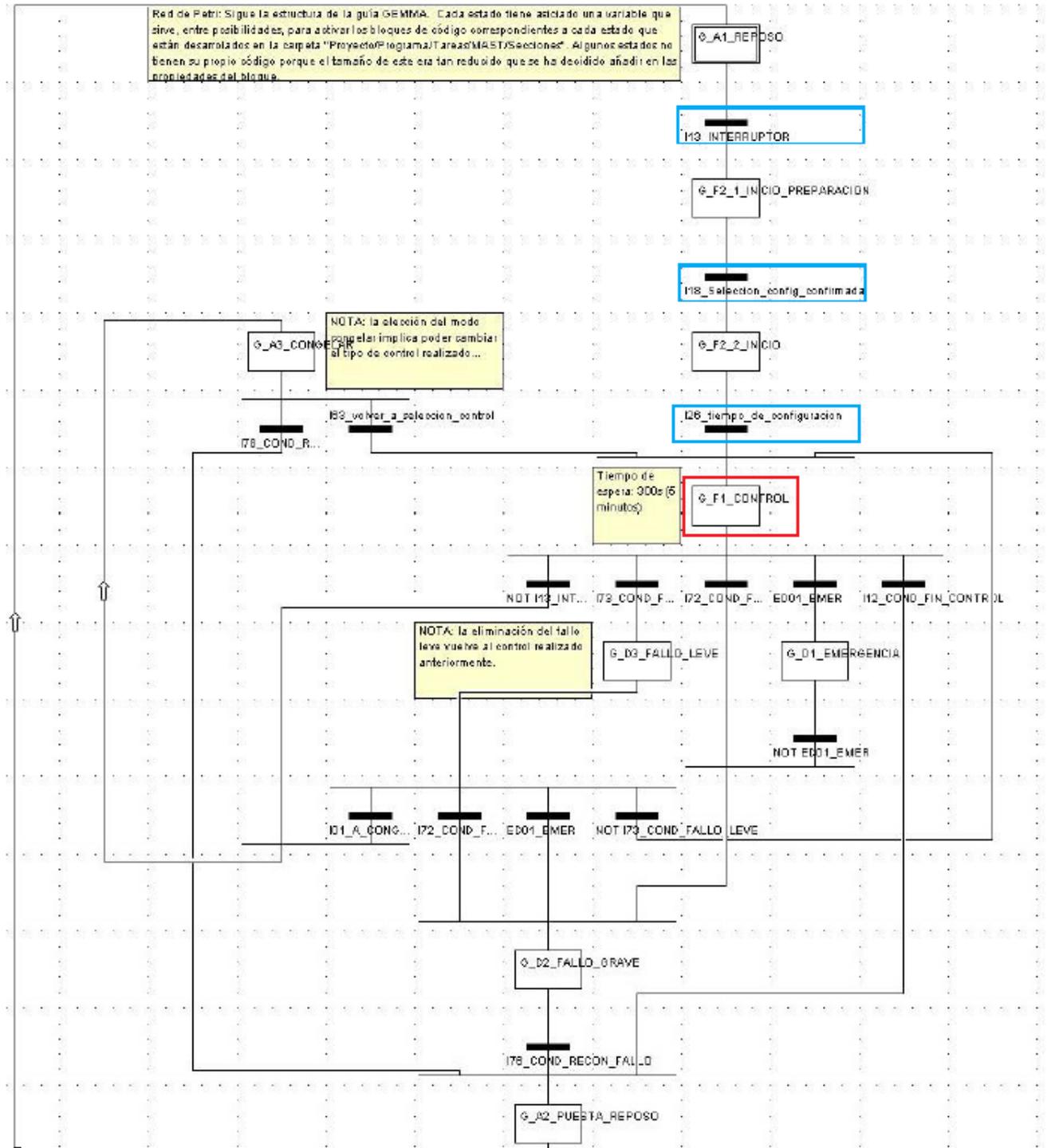


Figura 3.9 Camino por la red de Petri global para la puesta en remoto

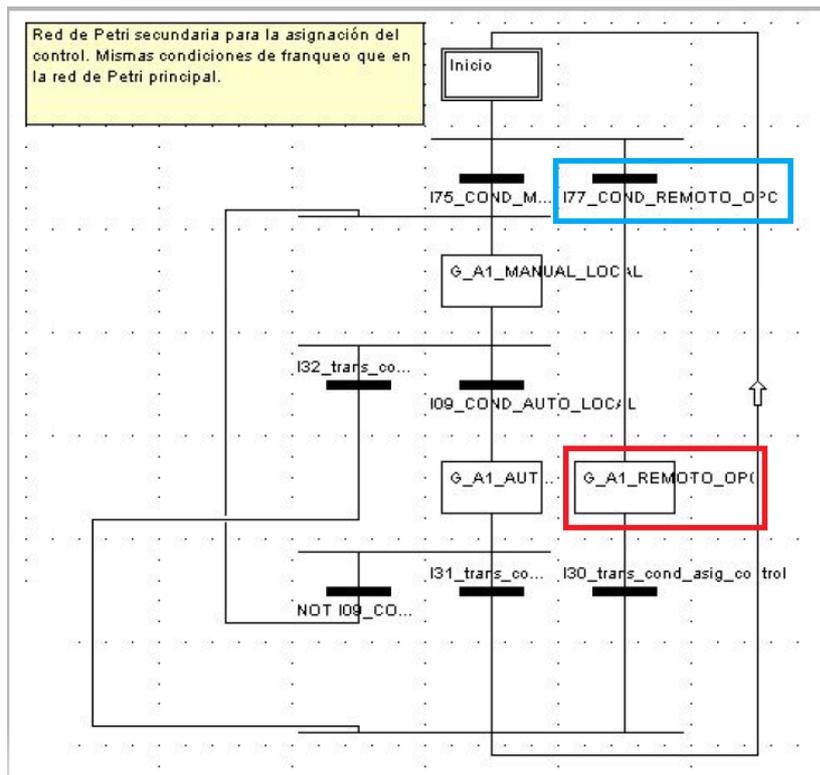


Figura 3.10 Camino por la red de Petri del control

En la función de Matlab “puesta_en_remoto”, disponible en el Anexo B, se opera de igual manera que en el script encargado en la comunicación con Simulink. En el caso que no esté conectado el servidor OPC se conecta y se crean los nodos necesarios para realizar la función que buscamos. Tras ello comienza la activación de variables que gracias a los cambios realizados en el código explicados anteriormente abrirán las transiciones necesarias.

Se empieza escribiendo a 1 la variable “OPC45_INTERRUPTOR” y esperando medio segundo.

Esta espera se hace para evitar posibles errores en la transición de estado y se repetirá en cada paso. Tras ello se escribe el entero que se le pasa como argumento correspondiente a la configuración seleccionada y se confirma la selección con las variables “OPC48_CONF_SELECCIONADA” y “OPC46_CONF_COFIRMADA”.

A continuación, se realiza una espera en este caso, por motivos diferentes. Se requiere de cierto plazo para colocar las válvulas solenoides en la posición correspondiente a la configuración seleccionada. En el proyecto del Unity está configurada para un minuto, pero se le ofrece al usuario ‘saltar’ la espera modificando el tiempo de espera de la línea 47 del código (ver Anexo B). Se debe tener cuidado ya que ese tiempo de espera corresponde al tiempo que tarda la válvula solenoide en permutar de posición y no se debe operar la planta antes de que ésta termine. Al fin de esta espera modificada, se abren las transiciones correspondientes a las variables “OPC7_ATAJO_PREPARACION” y “OPC49_REMOTO_OPC”.

Tras ello la planta se encuentra operativa y en comunicación con el servidor OPC. Conviene, para evitar posibles problemas si se sale de este estado ‘modo remoto’ poner a cero ciertas variables de nuevo.

3.2.2 Función para apagar la planta

De manera parecida a como se ha hecho para la función de puesta en remoto, para apagar la planta y dejarla en su estado de reposo. En este caso bastará con activar la variable correspondiente a ‘VJ40_r_FIN_CONTROL’ y poner a cero ‘OPC45_INTERRUPTOR’ en la tarea VIJEO_M340 (LD) para que desde el estado en reposo no se avance de nuevo.

Para evitar salir del modo de control remoto en una posible vuelta a puesta en remoto de la planta antes de cerrar la función se devuelve el valor de la variable de fin de control a cero. El código de la función

“apagar_planta” se encuentra en el Anexo C.

De igual manera que para poner la planta en remoto se representa el camino a seguir por la función.

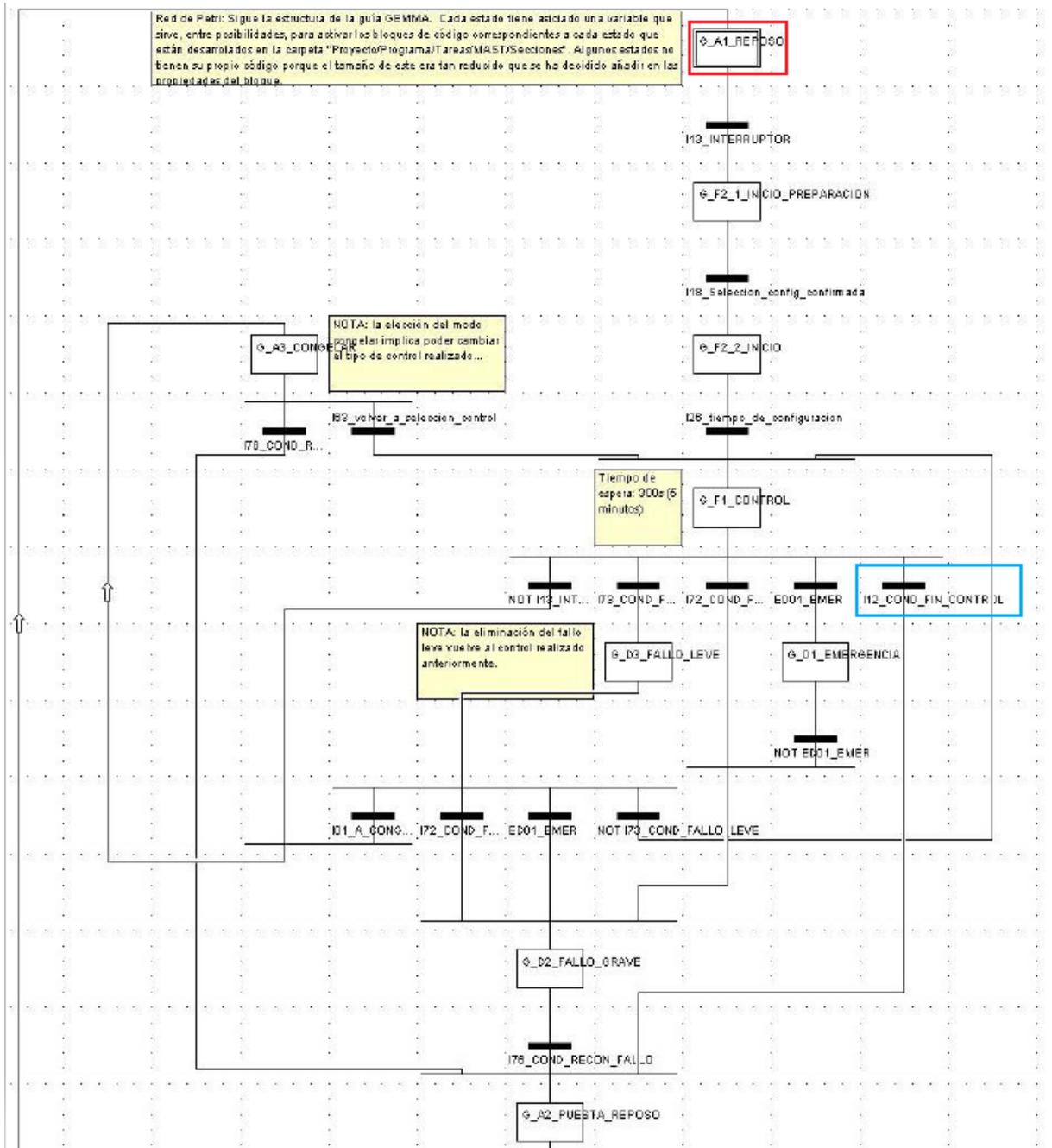


Figura 3.11 Camino por la red de Petri global para el apagado

3.3 Modelos de Simulink

De acuerdo con las distintas configuraciones en la que puede operar la planta y a los distintos experimentos de las fases de identificación e implementación de controladores que se llevarán a cabo en los siguientes capítulos de este Trabajo de Fin de Grado se han usado diferentes modelos de Simulink encargados de comunicarse con el PLC.

En dichos modelos necesitamos que la simulación ocurra en tiempo real, es decir, que el tiempo en el que se ejecuta la simulación vaya al mismo ritmo que el tiempo real. Esto, en la metodología de trabajo por defecto, se realiza a través de los bloques de Simulink propios para el OPC que configura automáticamente el modelo para que ocurra. Sin embargo, al estar en este caso forzados a usar un script con comandos se deberá configurar de manera manual. Esto se hará a través del bloque 'Real Time Sync' que acopla el tiempo real y el de la simulación. La estructura básica de la que partirán el resto de modelos de Simulink será la de la figura XX: envío de señales de control y recepción de datos a través del bloque de comunicación.

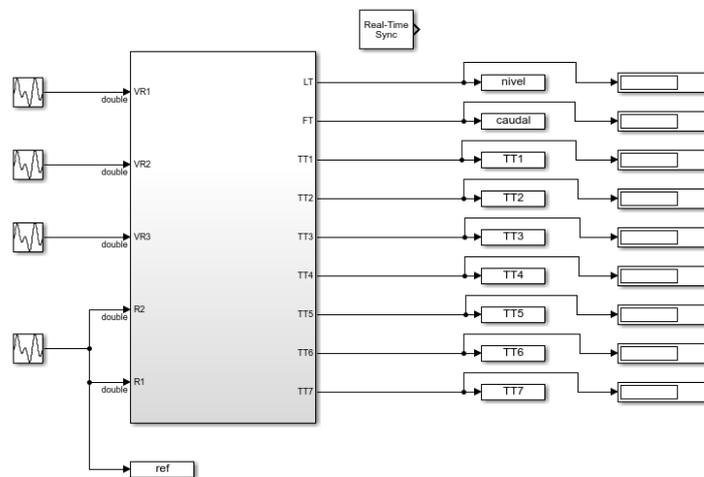


Figura 3.12 Modelo simplificado de conexión con la planta

Hay que tener cuidado con el tiempo de muestreo de la simulación, ya que, la operación de ejecutar un script con funciones de conexión por OPC es relativamente costosa para el ordenador y si se elige un tiempo de muestreo demasiado pequeño, el tiempo de ejecución del código será mayor que el propio tiempo de muestreo de la simulación. Esto provocará que el tiempo de simulación sea mayor que el real desvirtuando los datos.

Con el bloque 'Interpreted Matlab Fcn' mandamos y recibimos la información a través del cliente OPC. Se debe tener cuidado con las dimensiones de las señales de entrada y salida del bloque 'Interpreted Matlab Fcn'. Deberán estar en concordancia con las entradas y salidas del propio script en ejecución. Para construir el vector de entradas y separar el de salidas se ha hecho uso de los bloques 'Mux' y 'Demux' tal y como se muestra en la figura.

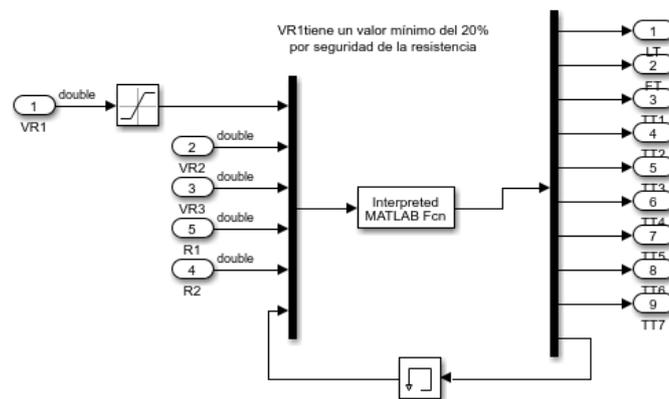


Figura 3.13 Bloque de comunicación por OPC

El script encargado en recibir y mandar información a través del cliente OPC su código se encuentra en el Anexo A. Su funcionamiento es bastante simple. En el caso de no haber un cliente OPC UA disponible trata de crearlo. Enlaza los nodos a variables globales que se han creado previamente y escribe los valores de las entradas y lee los que serán las salidas a través de las funciones descritas previamente en este capítulo.

En ocasiones, a lo largo del transcurso del trabajo el bloque ‘Real Time Sync’ ha dado algunos problemas. Este bloque tiene un número de tics máximo en los que la ejecución de cada iteración de la simulación puede retrasar al tiempo de simulación. Dicho máximo se modifica en el campo ‘Maximun missed ticks’.

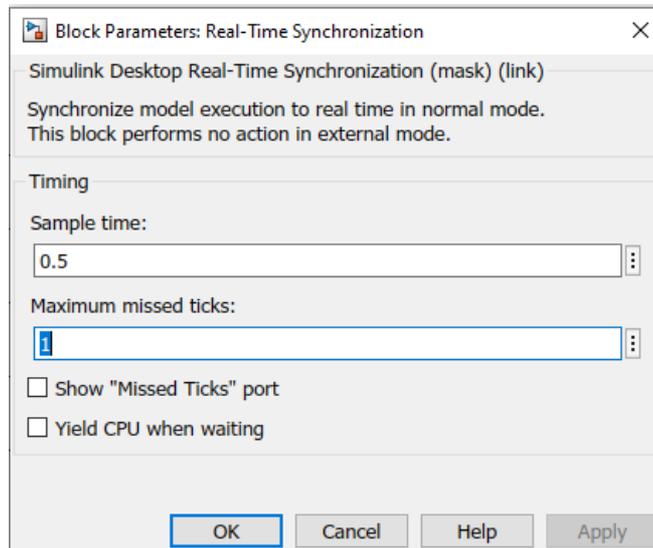


Figura 3.14 Cuadro de parámetros del bloque de sincronización

En el caso en el que la simulación del modelo presente un fallo en el que diga ‘Operation Timed out’ se recomienda poner un número alto de tics permitidos y dar más margen al Matlab para ejecutar el código a costa de una desvirtuación momentánea del tiempo de simulación.

4 PUESTA EN FUNCIONAMIENTO DE LA PLANTA EN CONFIGURACIÓN 3

Con el nuevo sistema de comunicación OPC UA implementado y con los modelos Simulink preparados para el intercambio y análisis de datos con la planta, se está en disposición de poder realizar una identificación de los comportamientos de las dinámicas del sistema. Antes de eso se realizará una descripción y general del funcionamiento de la planta en su tercera configuración.

4.1 Nuevo sensor de nivel

Debido al mal funcionamiento del sensor de nivel instalado antes en la planta, se ha decidido cambiarlo e instalar uno nuevo con una tecnología distinta.

La calibración es un proceso sencillo y gracias a poder contar con un dispositivo nuevo, con realizarla una vez bastará para su uso durante la realización de este Trabajo de Fin de Grado.



Figura 4.1 Sensor de nivel instalado

El sensor de nivel devuelve una señal digital de puntos “OPC44_w_LT_puntos”. La calibración consistirá en calcular la correlación de esos puntos a centímetros. Para ello, se coloca el tanque con un nivel de agua bajo de manera manual. Se recogen los datos obtenidos en puntos y los centímetros a los que correspondería. Se repite el procedimiento para varios niveles y se obtiene la relación deseada.

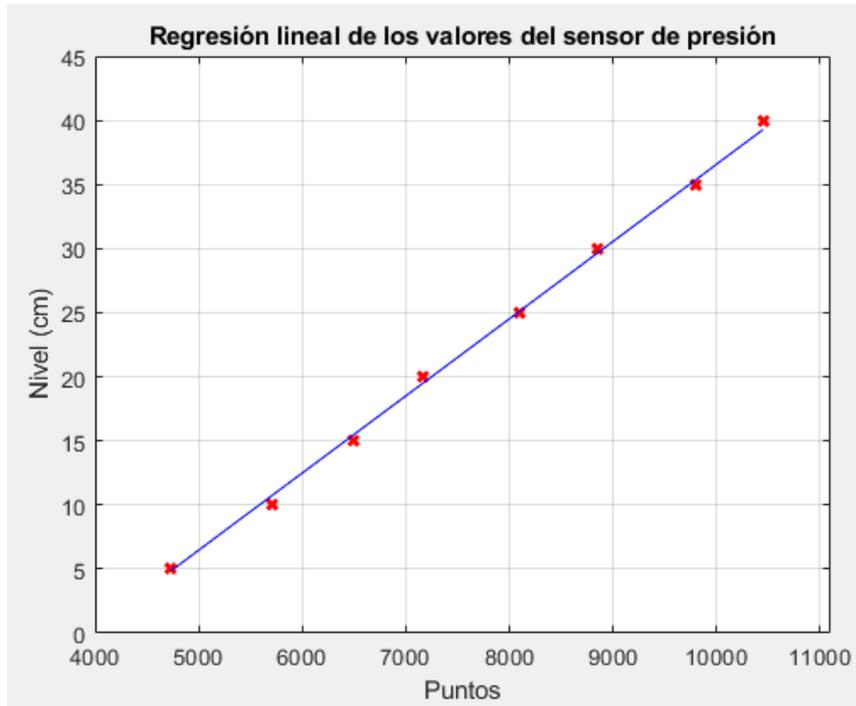


Figura 4.2 Ensayo de calibración del sensor de nivel

La relación obtenida y la que se aplicará a partir de este momento será:

$$\text{nivel en cm} = 0.0060215 * \text{nivel en ptos} - 23.64086$$

4.2 Planta de frío

La planta de frío es la encargada de aportar un caudal de agua fría a los intercambiadores de calor. Este sistema es un circuito cerrado que conectan los dichos intercambiadores con una enfriadora y un tanque del que extrae agua al evaporador de la enfriadora situada en el techo del edificio del laboratorio.

Este sistema de refrigeración mantiene de manera automática el agua entre los 4.5 °C y los 8°C. La temperatura del agua que aporta a los intercambiadores se puede medir a través de la sonda TT5 y esta es una representación de su ciclo habitual.

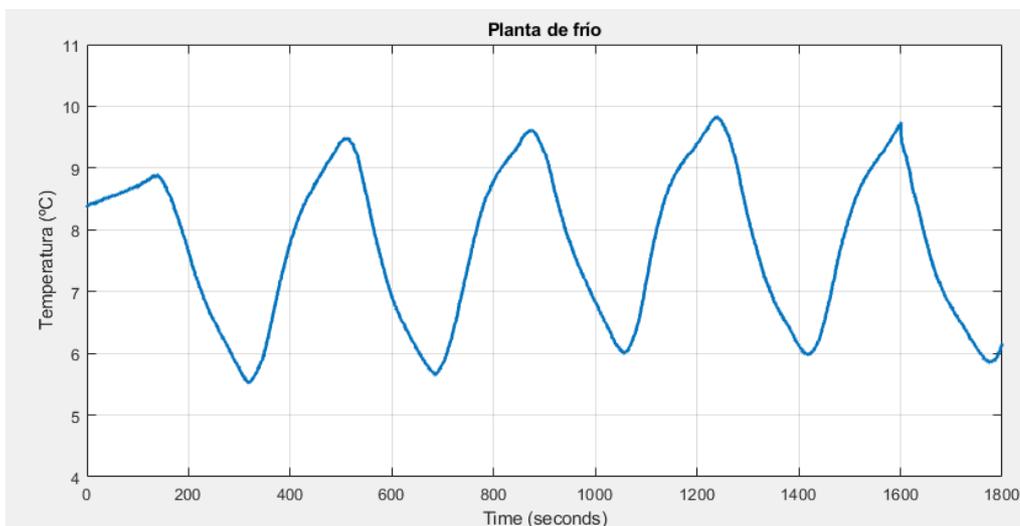


Figura 4.3 Temperatura del agua de la planta de frío

Es importante destacar, que la frecuencia con la que conmuta el controlador del sistema de refrigeración varía en función de la temperatura de retorno de los intercambiadores de calor y por tanto del funcionamiento particular del resto de las partes de la planta.

La planta de frío puede dejar de funcionar si se mantiene de manera prolongada enfriando mucho caudal de agua a través de los intercambiadores. Si se detecta este problema se recomienda apagar su diferencial unos segundos y dejar que vuelva a rango de temperaturas que presenta normalmente.

4.3 Configuración 3

La tercera configuración, como se puede ver en su esquema, prescinde del tanque colector. Sin embargo, el cierre del circuito por la válvula VS5 hace que el comportamiento del sistema se modifique considerablemente.

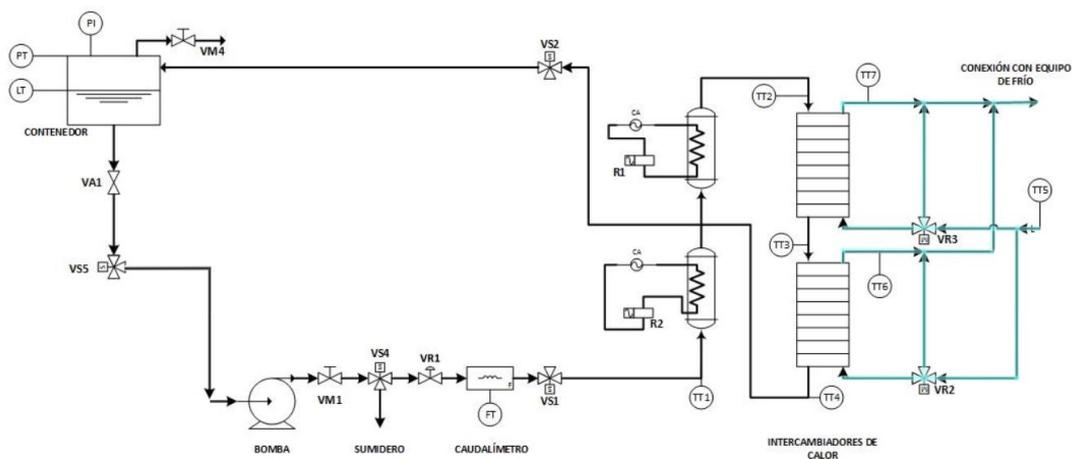


Figura 4.4 Esquema configuración 3

Al considerar la recirculación continua del caudal que circula por el sistema cambia por completo el comportamiento de las salidas. Existe una dinámica integradora dependiente de la diferencia entre los dos saltos de temperatura que ocurren en el circuito. Si el agua sufre un mayor salto térmico en las resistencias que en los intercambiadores de calor, se irá calentando rápidamente todo el volumen de agua en el circuito. Esto hará que en el régimen permanente estos saltos deberán ser equivalentes.

El nivel, debido a la recirculación, se mantendrá constante independientemente del caudal con el que se trabaje.

Las señales manipulables que controlarán los actuadores del sistema en esta configuración serán las siguientes:

- VR1 : la apertura de la válvula VR1 que define el caudal que circula por el circuito.
- R : duty cycle de las resistencias.
- VR2 : apertura de las válvulas de tres vías VR2 y VR3.

Por otro lado, las variables medibles con las que se trabajará en el control serán temperaturas en distintos puntos del circuito:

- TT2 : tras el paso por las resistencias.
- TT4: al salir del segundo intercambiador de calor

4.4 Puesta en funcionamiento de la configuración 3

La configuración 3 de la planta no es accesible desde el PLC ya que las válvulas que la configuran no son actuadores controlables, sino que son válvulas manuales. Para establecer la esta configuración será necesario cerrar la válvula VS5 tal y como se puede ver e la siguiente imagen:



Figura 4.5 Válvula VS5

Y tras ello, colocar la válvula que permite sacar el agua del depósito colector en su posición de cierre:



Figura 4.6 Llave de paso al depósito colector

En este modo, al retornar el agua desde el tanque nivelado a la bomba hidráulica puede ocurrir que pequeñas burbujas producidas por el flujo de entrada al tanque se introduzcan en la propia bomba. Para evitar esto se coloca el nivel del tanque en un nivel alto lo que disminuirá el número de burbujas creadas, así como se mantendrán más lejos de la entrada a la bomba. Cómo se ha indicado, este nivel se mantendrá constante para siempre.

La bomba tendrá aire en su interior si empeora su comportamiento. Si esto ocurre bombeará mucho menos caudal del habitual. Para una apertura de VR1 del 30% el caudal deberá estar en torno a los 380 l/h. De no ser así o escuchar un fuerte silbido en la bomba se debe parar y extraer el aire de su interior.

Para ello se requerirá una llave inglesa para abrir una pequeña apertura en el tornillo que aparece en la figura.



Figura 4.7 Extracción de aire de la bomba

Esta apertura deja salir el aire de la cámara de la bomba por el orificio. Una vez se detecte que no pasa aire sino agua volver a cerrar la apertura con fuerza.

5 IDENTIFICACIÓN DE LA CONFIGURACIÓN 3

La planta ha sido testada y utilizada en años anteriores en las dos primeras configuraciones. Con objeto de completar el conocimiento sobre el funcionamiento de las dinámicas que tienen lugar en la máquina, se va a llevar a cabo durante este capítulo, una identificación de la configuración 3 para obtener un modelo en variables de espacio de estados del sistema.

A diferencia de las configuraciones 1 y 2, en este caso se trabajará con un circuito cerrado de agua en constante circulación por las distintas partes de la planta. Veremos cómo eso afecta a las dinámicas de las variables controlables.

5.1 Punto de funcionamiento

Antes de comenzar cualquier ensayo de identificación es necesario establecer un punto de funcionamiento desde el cual se controlarán las variables controlables.

A partir de los resultados obtenidos en su Trabajo de Fin de Grado Víctor Gracia [3] el punto de operación desde el que se operará ahora en adelante es el denominado como ‘caudal reducido’.

Este punto de funcionamiento se define con los valores de VR1=30%, el Duty cycle de las resistencias 1 y 2 DC=55%; y las válvulas VR2 y VR3 al 30%. Estos valores en los actuadores producirán unos valores de equilibrio en las variables de salida.

Para hallarlo se realiza un ensayo en el que dichos actuadores se mantienen constantes.

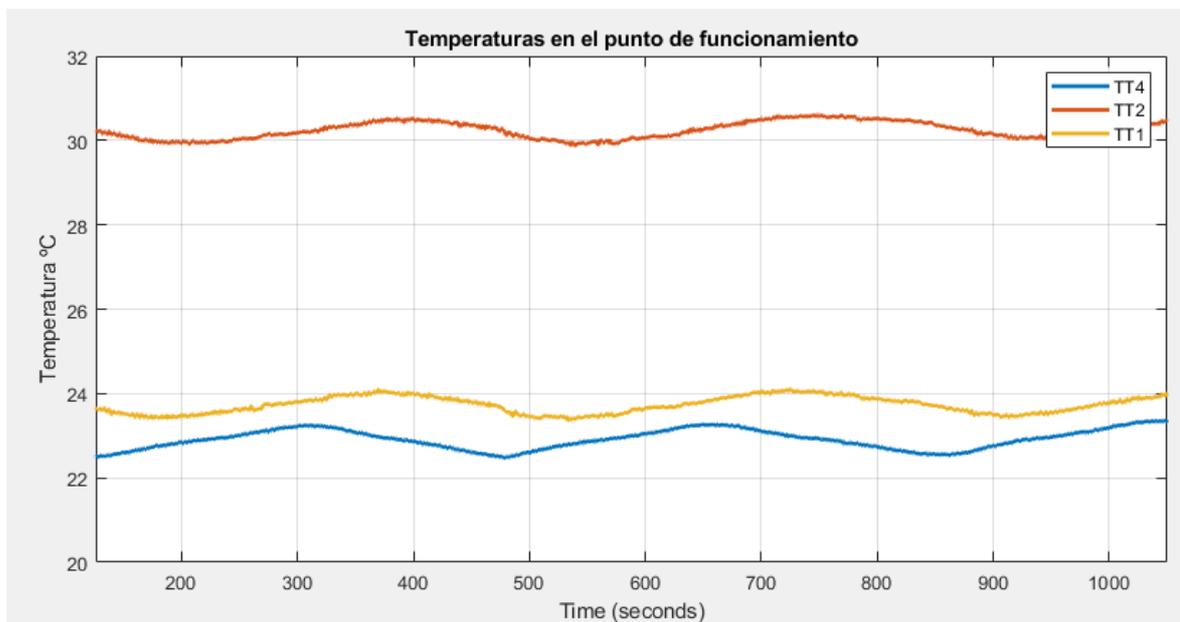


Figura 5.1 Ensayo en el punto de funcionamiento

Se puede apreciar claramente cómo la oscilación que produce TT5 (temperatura de la planta de frío) afecta directamente en el régimen permanente de las temperaturas del sistema.

Para comprobar esta influencia y corregirla, se pueden aplicar a estas señales un procesamiento en el que se resta la influencia de TT5 para cada instante del experimento.

La influencia de TT5 en las temperaturas tiene un retraso que será diferente para cada temperatura. Esto es así debido a tiempo que tarda que llegar el agua saliente del intercambiador de frío a las distintas ubicaciones de los sensores.

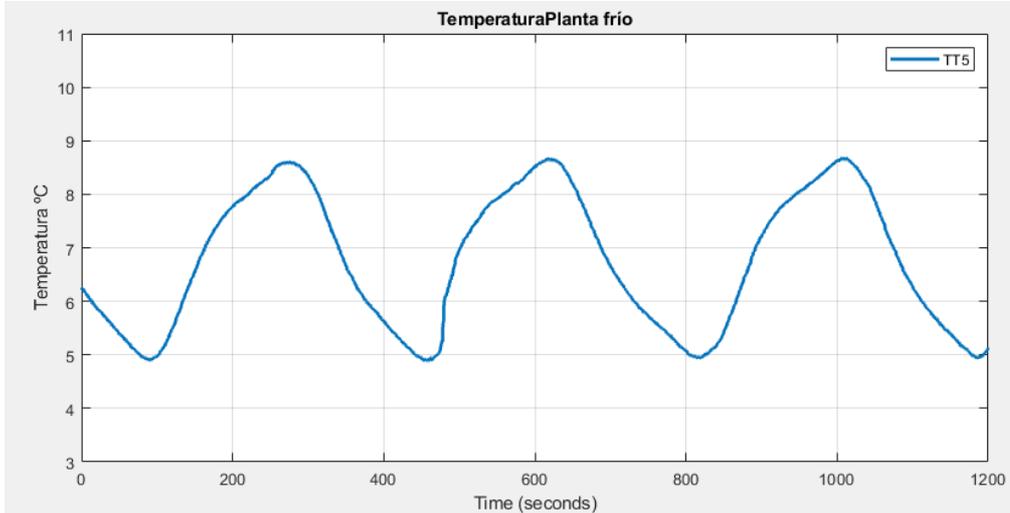


Figura 5.2 TT5 durante el ensayo en el punto de funcionamiento

Para cada temperatura que se desee corregir se deberá medir este retraso 'r' respecto a TT5.

Dicha corrección se ha llevado a cabo con el siguiente bucle:

```

TT5_v=TT5.Data;
TT4_v=TT4.Data;
for i=1:length(TT2.Data)-r
    j=j+1;
    TT4_corregido(i)=TT4_v(i+r)-(G_TT4/A_TT5*(TT5_v(i)-mean(TT5_v)));
end

```

En él se almacenan en vectores los datos obtenidos en el experimento, el vector con las temperaturas corregidas corresponde al del vector original menos el valor normalizado de TT5 (para el instante actual menos el retraso) multiplicada por un factor. Dicho factor se ha calculado mediante pruebas y es diferente para cada temperatura.

Tras este procesamiento las salidas quedan de la siguiente manera.

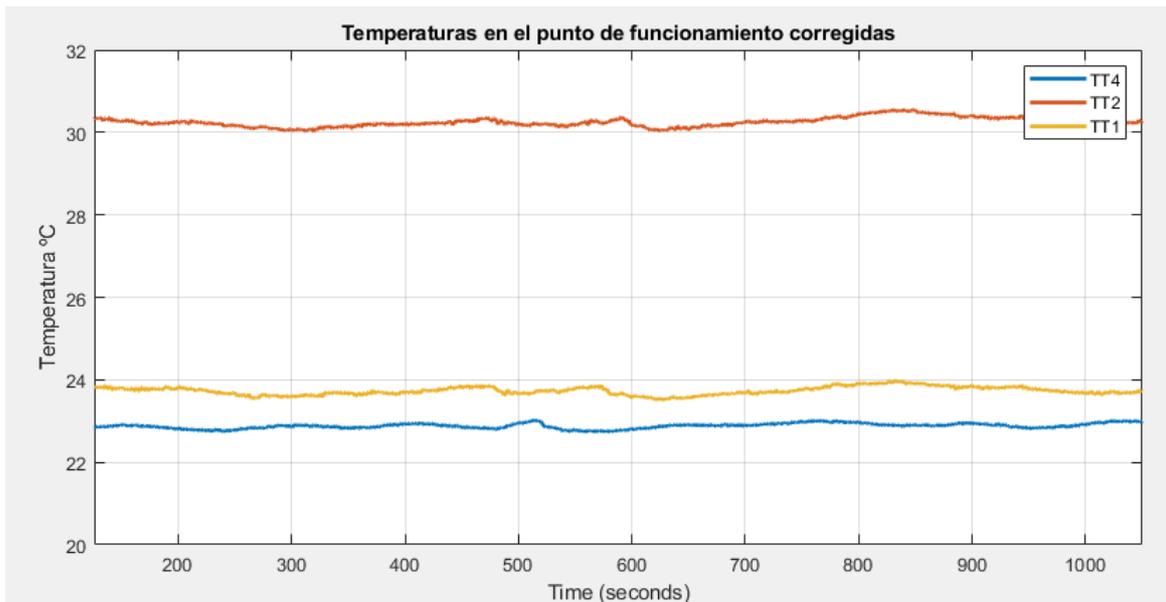


Figura 5.3 ensayo en el punto de funcionamiento corregido

Podemos ahora determinar los valores de los puntos de equilibrio del punto de funcionamiento:

$$TT1 \approx 23.9^{\circ}C \quad TT2 \approx 30.5^{\circ}C \quad TT4 \approx 23^{\circ}C$$

Estas temperaturas pese a todo dependerán de las condiciones ambientales en las que se trabaje con la planta por lo que serán valores orientativos para dimensionar los saltos de temperatura que se producen en las etapas del circuito.

5.3 Ensayos de identificación

Para la realización de la identificación del modelo de la planta se han probado distintas metodologías.

En un primer momento, se trató de realizar ensayos en escalón de los actuadores. Para obtener una mayor información de las dinámicas de las variables de ha decidido realizar un ensayo CHIRP en su lugar. Este ensayo permitirá conocer la respuesta de las salidas en todo su rango de validez aportando una información más interesante para la realización del modelo.

Este ensayo además resuelve el problema de la dinámica integradora que presenta esta configuración. Esta dinámica integradora se refiere al efecto en las salidas del sistema antes cambios en los actuadores no sólo en un primer ciclo del caudal en la planta sino, al consistir en un circuito cerrado, en los ciclos posteriores a éste. El ensayo CHIRP a diferencia del ensayo en escalón, tiene un valor medio de las señales de los actuadores próximo a 0 por lo que esa dinámica integradora se va compensando y manteniéndose las salidas entorno al punto de equilibrio del sistema.

Para determinar el rango de frecuencias en el que se realizará el ensayo CHIRP se han de hacer ciertos cálculos previos. Se ha asumido que el rango de frecuencias de ambas dinámicas es igual para TT2 y TT4.

Por un lado, la frecuencia mínima se calcula a partir de la constante de tiempo de cada sistema

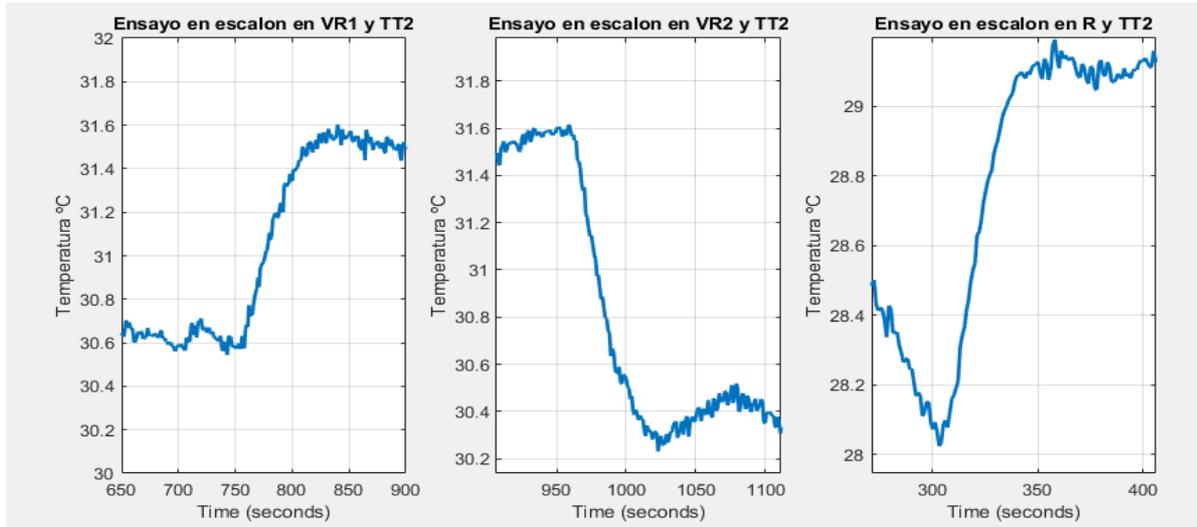


Figura 5.4 Ensayos en escalón de los actuadores

Como son valores orientativos nos vale una determinación cualitativa de la constante de tiempo.

$$\tau_{VR1} \approx 20s \quad \tau_{VR2} \approx 20s \quad \tau_R \approx 15s$$

Siendo

$$\omega_{min} = \frac{1}{10\tau}$$

Quedan las frecuencias:

$$\omega_{minVR1} = 0.005 \text{ rad/s}$$

$$\omega_{minVR2} = 0.005 \text{ rad/s}$$

$$\omega_{minR} = 0.0067 \text{ rad/s}$$

Para las frecuencias máximas se han escogido una 30 veces superior a la mínima.

La amplitud de la señal CHIRP se ha definido:

$$A_{VR1} = 10\%$$

$$A_{VR2} = 15\%$$

$$A_R = 20\%$$

Esta serán los porcentajes de apertura y duty cycle que se desplazarán respecto al valor del punto de funcionamiento respectivamente.

Con estos parámetros se ha procedido a realizar el ensayo en la planta real. Para la creación del modelo existen varias posibilidades. Se puede realizar los tres ensayos uno tras otro en el mismo experimento y tratar de obtener el modelo multivariable directamente con la aplicación “SystemIdentification” de Matlab. Por otro lado, se puede realizar dicho procedimiento tres veces y sacar tres modelos diferentes para cada uno de los actuadores con los que se trabajará, tras ello, colocar las matrices resultantes en un sistema completo multivariable que albergue a los tres modelos obtenidos.

En primer ensayo realizado se trata de obtener el modelo multivariable directamente.

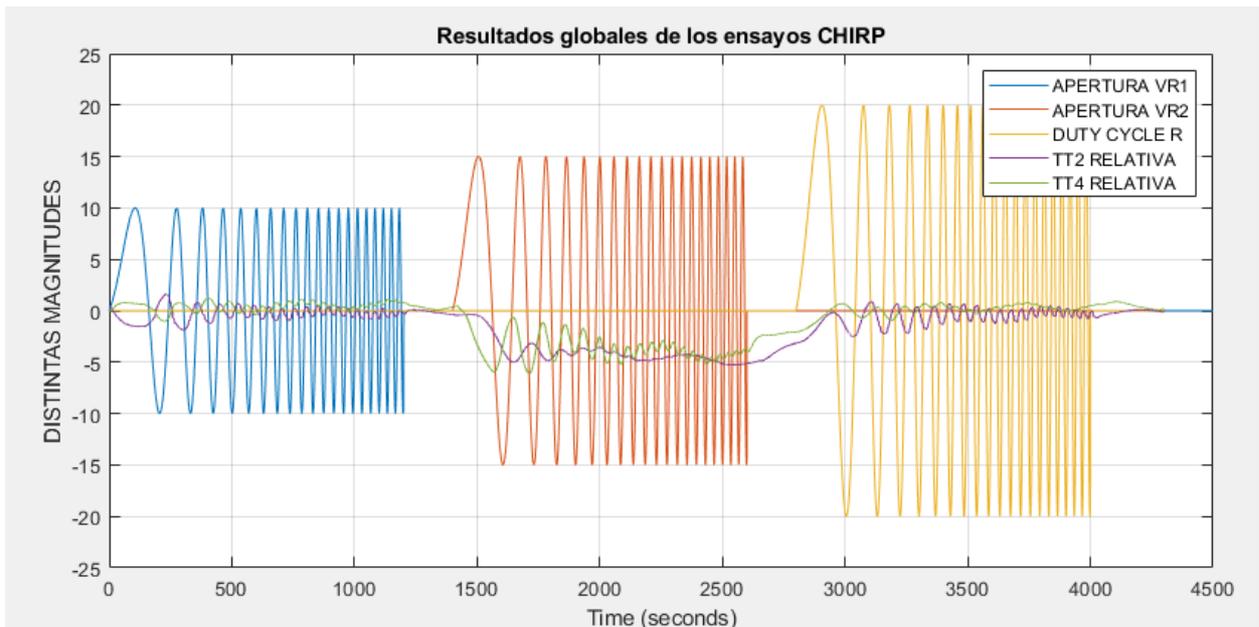


Figura 5.5 Ensayo CHIRP global

Los resultados no son del todo satisfactorios. Pese a que para VR1 y R la respuesta de las temperaturas (relativas) salen como se esperaban, en VR2 presentan una tendencia que enfría el sistema fuertemente. El modelo obtenido tras la introducción de estos datos en el “SystemIdentification” no es bueno por lo que finalmente se ha decidido realizar el ensayo de identificación por separado.

5.3.1 Ensayo VR1

Para modelar la respuesta de TT2 y TT4 frente a cambios en VR1 podemos reutilizar los datos del ensayo general seleccionando el rango que nos interesa.

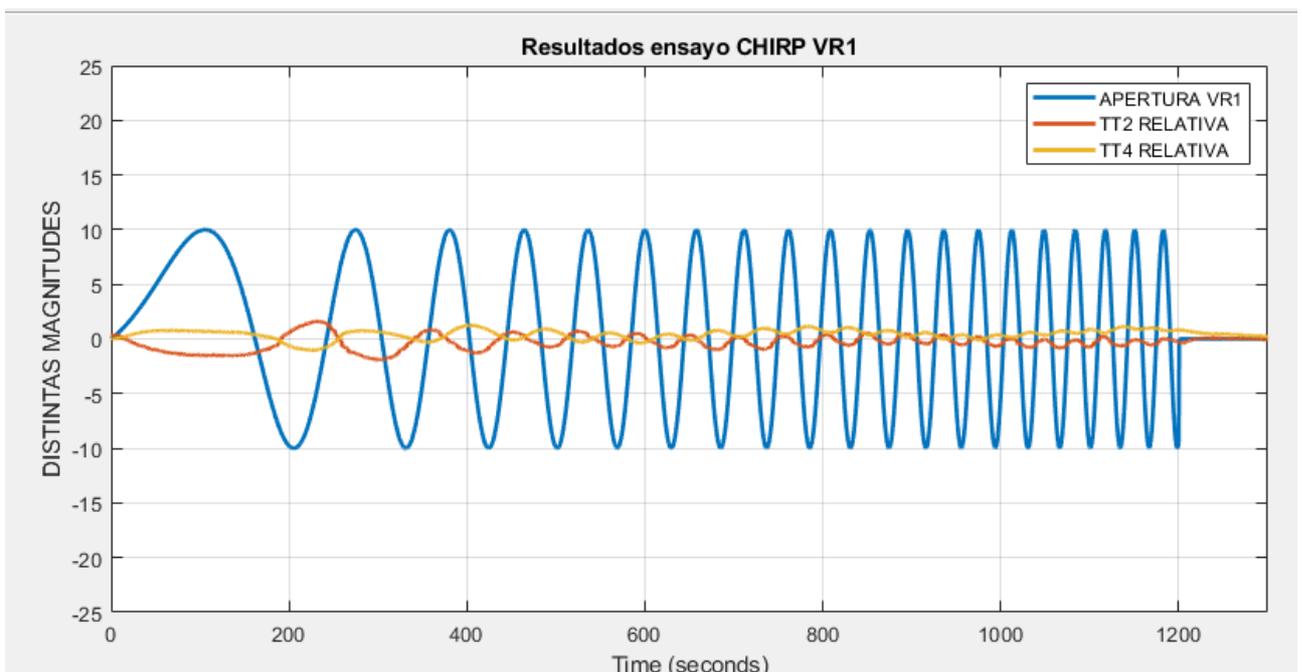


Figura 5.6 Ensayo CHIRP VR1

Se puede observar como la TT2 que se encuentra tras el paso de las resistencias, aumenta cuando el caudal que pasa por éstas disminuye. Esto tiene sentido físico ya que al estar el volumen de agua más tiempo en contacto con las resistencias, éste se calienta más. La potencia calorífica que aportan las resistencias es la misma pero ahora el volumen de agua que la absorbe es menor y por tanto, mayor su temperatura.

Este efecto vuelve a tener lugar tras los intercambiadores en TT4 y como se puede ver, no sólo anula el efecto de las resistencias, sino que se enfría un poco más. Esto puede ser causado por la distinta morfología de los intercambiadores y su distinta eficiencia haciendo que uno de los dos actuadores presente una menor sensibilidad respecto a cambios en el caudal.

Al estirar un poco el eje Y de la figura, se puede ver claramente la influencia de las variaciones de TT5 en las temperaturas.

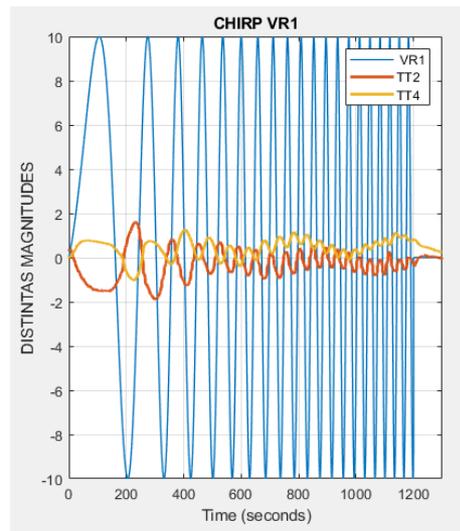


Figura 5.7 Ensayo CHIRP VR1 comprimido

Para solucionar esto, se han probado diversas estrategias. Por un lado, se ha optado por la corrección manual descrita en el apartado 5.1 e introducir como entrada del modelo únicamente VR1 y como salidas TT2 y TT4.

Sin embargo, al realizar esta metodología tanto en este experimento, como en los que le seguirán a continuación, los resultados del modelo no son buenos por lo que se decide optar por otra cuyos resultados ofrezcan un mayor acercamiento al comportamiento real de la planta.

Se ha optado por introducir como datos de estimación los datos obtenidos sin procesar y a su vez, tomar TT5 como una entrada más del modelo. Esta entrada en el futuro modelo multivariable de control será tratada como una perturbación.

5.3.2 Ensayo R

Se repite el procedimiento seguido para VR1 para tratar de modelizar el comportamiento de las temperaturas respecto a cambios en el duty cycle de las resistencias.

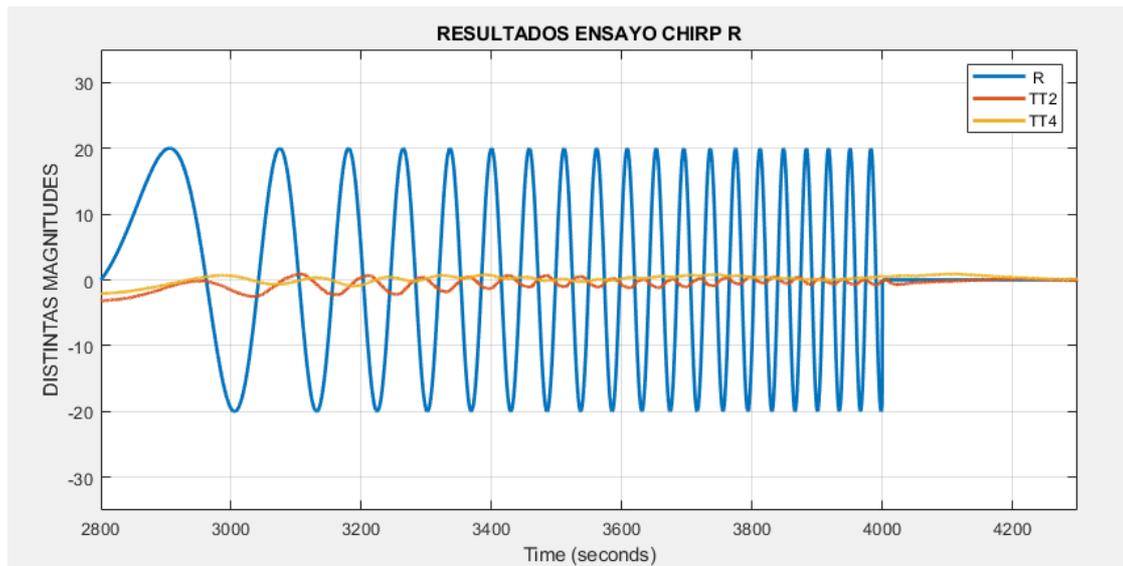


Figura 5.8 Ensayo CHIRP R

Tal y como se expuso anteriormente, los resultados del ensayo CHIRP global son válidos para los cambios en R. Por ello la respuesta de la figura corresponden a esta tercera parte de dicho experimento.

El efecto de las resistencias es claro, tanto TT2 y TT4 son directamente proporcionales al ciclo de trabajo de las resistencias. Cada una de las salidas presenta un retraso en la respuesta debido al tiempo que tarda en caudal en alcanzar el sensor correspondiente. Como no puede ser de otra manera, esta respuesta también está influenciada por las oscilaciones de TT5 y su introducción al modelo será análoga a la de VR1.

5.3.3 Ensayo VR2

El ensayo de identificación de los intercambiadores de frío es sin duda el más problema ha dado a la hora de obtener resultados satisfactorios. Esto es así debido a las distintas dinámicas que presenta el funcionamiento de los intercambiadores y la fuerte influencia que tiene TT5 en los resultados.

Para conseguir un modelo se han hecho los diversos ensayos.

Por un lado, se realizó un ensayo con una señal seno con una frecuencia más baja que en el CHIRP. Como se puede ver, esa tendencia que se presentó en el ensayo CHIRP que provocaba un enfriamiento en las temperaturas que parece que se atenúa bajando la frecuencia de las oscilaciones de la señal. Con este ensayo se ven los efectos de la apertura VR2 en las temperaturas.

Una mayor apertura, deja pasar un mayor caudal de agua del circuito frío en el intercambiador enfriando el caudal del circuito cerrado de la planta. Este enfriamiento tiene cierto retraso como en los ensayos anteriores.

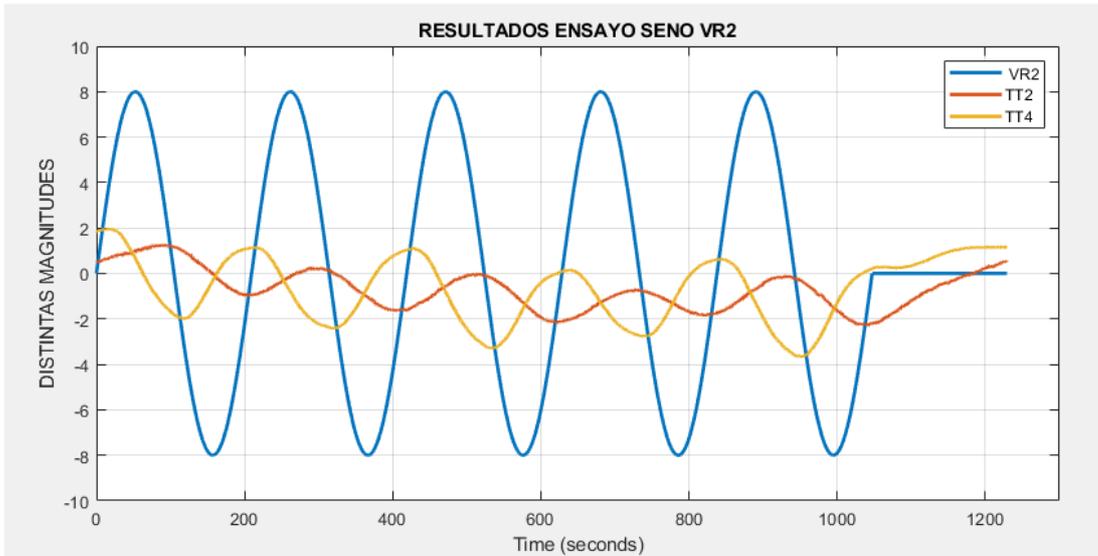


Figura 5.9 Ensayo con señal seno en VR2

Sin embargo, al realizar el ensayo en una sola frecuencia y siendo esta tan baja, no permite obtener mucha información necesaria para la obtención del modelo. Es por ello porque se ha vuelto a intentar un ensayo CHIRP esta vez con la frecuencia de objetivo más baja.

Para ello, se han modificado los parámetros del CHIRP para que se solucionen partes de estos problemas. La amplitud se ha reducido a 7, y las frecuencias se definen:

$$\omega_{minVR2} = 0.01rad/s$$

$$\omega_{maxVR2} = 0.126rad/s$$

Con estas modificaciones el ensayo realizado atenúa ciertamente la tendencia que presentaba en el ensayo global.

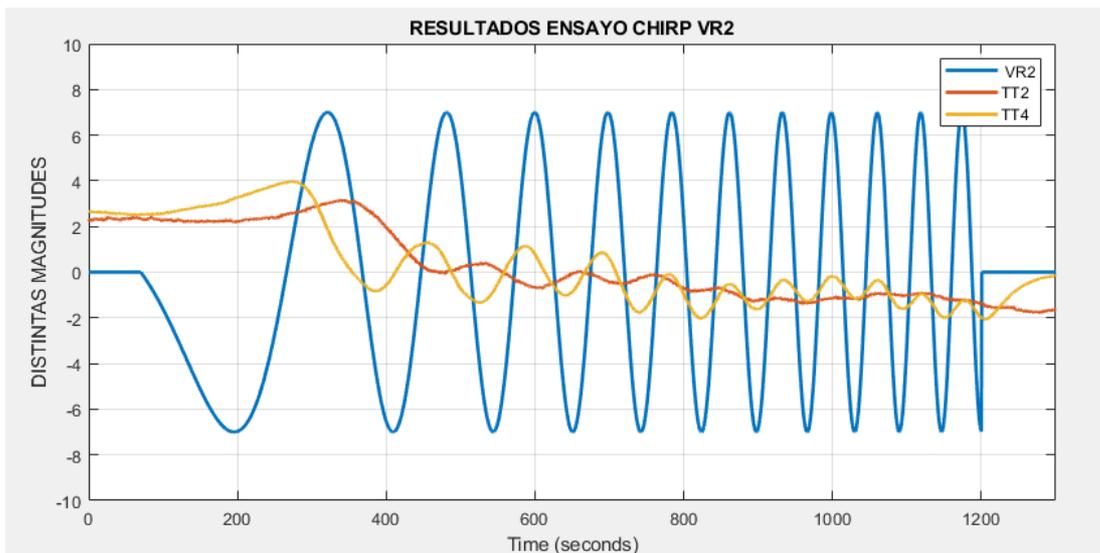


Figura 5.10 Ensayo CHIRP VR2 modificado

Pese a no ser un ensayo exacto es el que se usará para hallar el modelo en variables de espacio de estados esperando que la realimentación en el lazo de control compense las inexactitudes del modelo diseñado.

5.4 Representación en variables de estado de un sistema de control

El modelo que se pretende obtener es un modelo lineal en variables de espacio de estado del sistema.

La representación en espacio de estados o variables de estado trata de describir el comportamiento de sistemas dinámicos. Dicho comportamiento queda definido por n ecuaciones cuyas variables son las propias variables de estado.

El modelo en espacio de estados no se ofrece una simple relación entre las entradas y las salidas del sistema, sino que define la salida de un momento dado a partir de la evolución dinámica que haya experimentado el sistema en el tiempo.

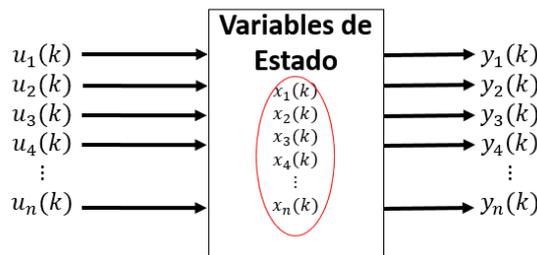


Figura 5.11 Concepto de máquina de estados Fuente [7]

Es una técnica que se puede aplicar tanto a sistemas SISO (una salida y una entrada) como al caso que ocupa el estudio en este Trabajo, sistemas MIMO (varias entradas y varias salidas)

Las ecuaciones que representan un sistema dinámico se representan en forma matricial. Para tiempo discreto se define el sistema como

$$x = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

Estas matrices serán las que definan su comportamiento, así como su observabilidad, controlabilidad y estabilidad. En diagramas de flujo como el que se diseñará en Simulink tiene siguiente estructura:

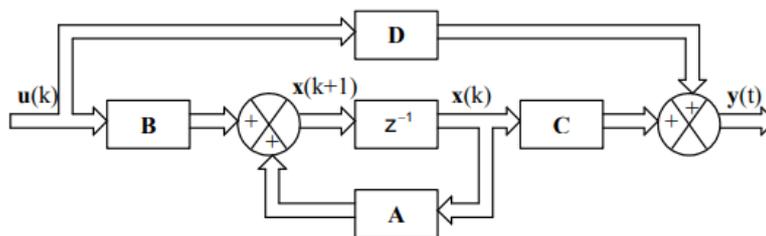


Figura 5.12 Diagrama de flujo de un sistema en variables de espacio de estados. Fuente [6]

5.4 Modelos obtenidos

Una vez definida la forma del modelo que se desea obtener y con los ensayos de identificación completados se van a calcular los parámetros que definirán los submodelos desde los que se formarán en el siguiente capítulo

el modelo lineal completo del sistema.

La obtención del modelo será usando la aplicación “SystemIdentification”. En ella se meten los datos de estimación y validación para estimar un sistema de espacio de estados que modele las respuestas. Es importante destacar que a los modelos siempre se le introducirán datos relativos respecto al punto de funcionamiento y éste calculará las temperaturas relativas también respecto al punto de equilibrio obtenido.

Los modelos elegidos serán modelos de 3 estados discretos con el tiempo de muestreo $T=0.5s$. Esta estructura es la que mejor se adapta a los datos y la que mejores resultados de aproximación a la planta real obtenido.

5.4.1 Modelo del efecto de VR1

Cómo se ha indicado se han introducido los datos de cada variable por separado y para el caso de VR1 este es el resultado. Como se puede ver en éste y en los experimentos que le seguirán, el modelo pese a no ser perfecto presente una clara correlación entre las salidas reales y las modeladas.

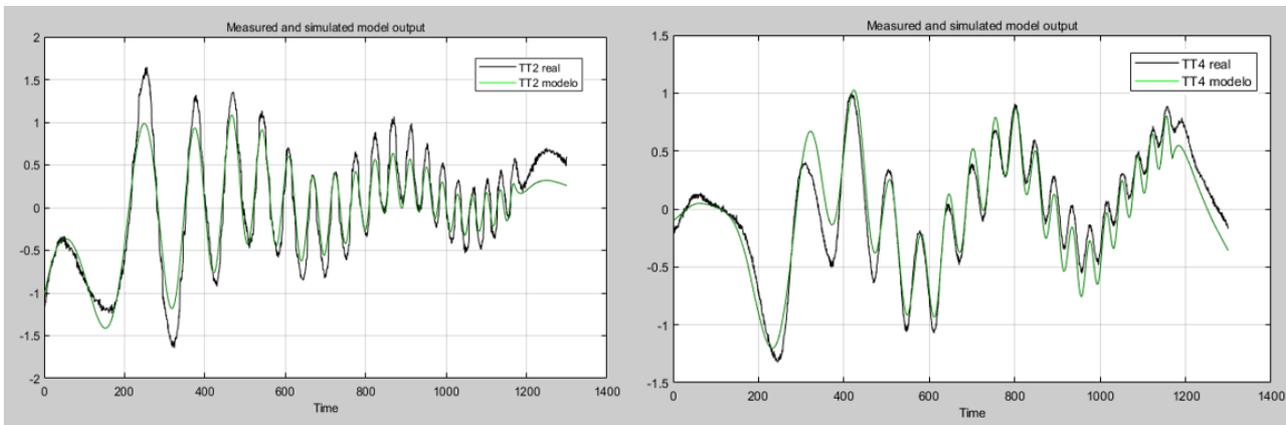


Figura 5.13 Modelo TT2 y TT4 frente a VR1 y TT5

El modelo obtenido por Matlab es el siguiente:

$$A = \begin{bmatrix} 0.9951 & -0.0055 & 0.0268 \\ 0.0011 & 0.9952 & -0.0023 \\ -0.0462 & -0.0180 & 0.9434 \end{bmatrix}$$

$$B = 10^{-3} \begin{bmatrix} 0.0908 & -0.3055 \\ -0.0440 & 0.1481 \\ -0.413 & -0.0014 \end{bmatrix}$$

$$C = \begin{bmatrix} 20.2426 & 25.7708 & 0.568 \\ -15.7624 & 3.1830 & 0.1465 \end{bmatrix}$$

$$D = \emptyset$$

5.4.2 Modelo del efecto de VR2

Como se indicó anteriormente, el modelo de VR2 es el que más complicado a resultado de realizar. Se puede observar en la figura, que la aproximación a TT2 es mejorable, aunque para TT4 presenta un comportamiento

un poco más aceptable. Esto es lógico debido a que la actuación en TT4 se encuentra justo después de la sonda TT2 lo que hace que su respuesta se vea afectada por múltiples factores en ocasiones impredecibles durante su recorrido por el circuito. Pese a la discreta respuesta del modelo se ha decidido continuar con los resultados obtenidos.

Los resultados obtenidos son tras la estimación del modelo se representan en la siguiente figura.

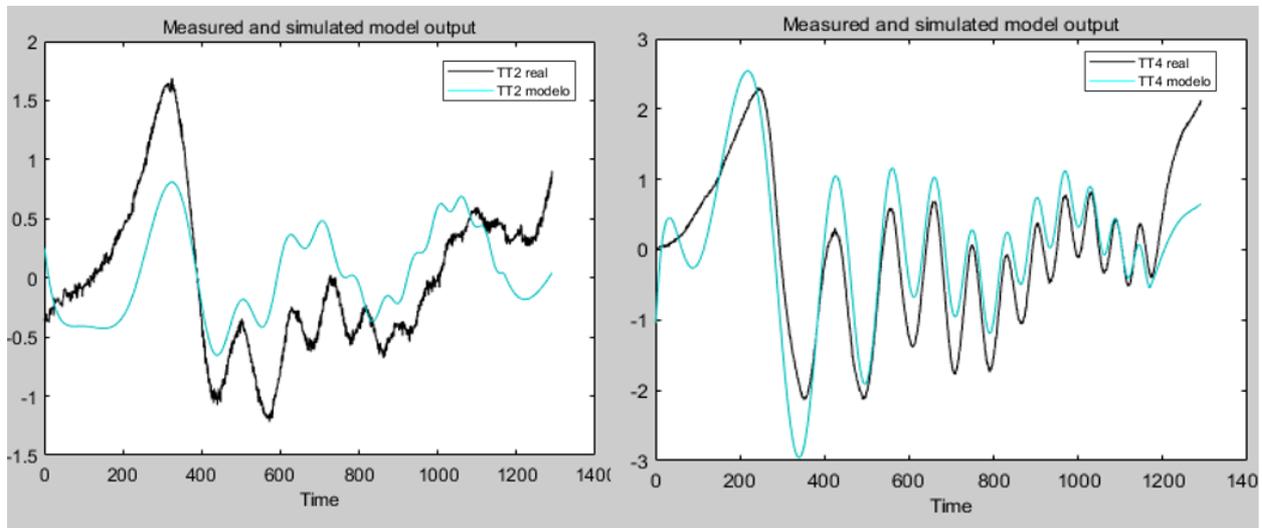


Figura 5.14 Modelo TT2 y TT4 frente a VR2 y TT5

Los valores para las matrices del modelo resultan:

$$A = \begin{bmatrix} 0.9981 & 0.0010 & 0.5678 \\ -0.0099 & 0.9929 & 0.0264 \\ -0.0034 & 0.0024 & 0.9540 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0001 & -0.0004 \\ 0.0001 & -0.0025 \\ 0.0005 & 0.0042 \end{bmatrix}$$

$$C = \begin{bmatrix} -21.3220 & 10.4764 & -0.933 \\ -24.9664 & -4.7048 & 0.2090 \end{bmatrix}$$

$$D = \emptyset$$

5.4.3 Modelo del efecto de R

Para R se vuelve a unos ensayos de estimación más sencillos y vuelven a ser válidos los resultados del ensayo CHIRP global. También ocurre algo parecido a lo que pasaba con VR2. El modelo en TT2 resulta un poco más acertado ya que el sensor de TT2 se encuentra justo tras la etapa donde tiene lugar la actuación de R. En TT4 la respuesta conlleva ciertas dinámicas añadidas en el transcurso del caudal por los intercambiadores de frío que hay tras TT2 y previos a TT4

Los resultados obtenidos son para este último submodelo resultan:

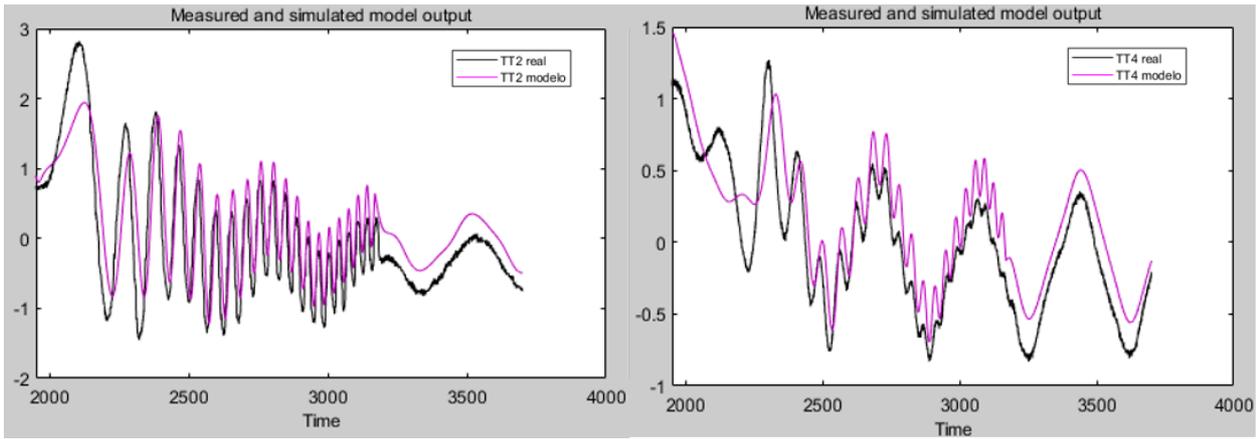


Figura 5.15 Modelo TT2 y TT4 frente a R y TT5

Y el último de los tres modelos resulta:

$$A = \begin{bmatrix} 0.9992 & -0.0072 & 0.0050 \\ 0.0077 & 0.9960 & 0.0479 \\ -0.0028 & -0.0270 & 0.9604 \end{bmatrix}$$

$$B = 10^{-3} \begin{bmatrix} -0.0118 & -0.1355 \\ -0.0726 & 0.0276 \\ 0.0811 & -0.3107 \end{bmatrix}$$

$$C = \begin{bmatrix} -352690 & 32.597 & -0.2303 \\ -23.0021 & -4.4147 & .3583 \end{bmatrix}$$

$$D = \emptyset$$

Se puede ver como los modelos obtenidos se adaptan a las desviaciones producidos por TT5 al meter la propia TT5 como una entrada del sistema. En el siguiente capítulo se desarrollará los cambios realizados al modelo obtenido para tomar esta TT5 como una perturbación en lugar de una variable controlada.

6 DISEÑO DEL CONTROL Y SIMULACIÓN DEL MODELO

En este capítulo se llevará a cabo una explicación detallada de la estrategia elegida para el control de las salidas seleccionadas. Se realizará un modelo lineal en variables de espacio de estados multivariable y se simularán las salidas tras ir implementando las etapas que formarán el modelo de control completo.

Primero se obtendrá el modelo multivariable a partir de los submodelos obtenidos en la identificación, tras ello se diseñarán las ganancias tanto del controlador como del observador; y se calcularán los estados y las señales de control correspondientes a una referencia dada a través de un SSTO (Steady-State Target Optimization).

6.1 Modelo lineal en espacio de estados

Es el momento de particularizar el problema al caso de estudio. El modelo que se buscará tendrá tres variables manipuladas (VR1, VR2, R) que tratarán de controlar a dos variables de salidas (TT2, TT4) y contará con la perturbación de TT5. La forma del modelo será:

$$\begin{aligned}x(k+1) &= Ax(k) + Bext\ u(k) \\y(k) &= Cx(k) + Du(k)\end{aligned}$$

El vector x y u tendrán la siguiente estructura:

$$x = \begin{bmatrix} x_{VR1} \\ x_{VR2} \\ x_R \end{bmatrix} \quad u = \begin{bmatrix} u_{VR1} \\ u_{VR2} \\ u_R \\ d_{TT5} \end{bmatrix}$$

Donde x_{VR1} , x_{VR2} y x_R son los vectores de estado de los submodelos obtenidos en el capítulo 5. Las componentes de u serán las actuaciones relativas de cada actuador. De esta manera el vector de estados contará con 9 estados y el de actuaciones con 3 componentes.

Por su parte la matriz A será una 9x9 con la siguiente estructura:

$$A = \begin{bmatrix} A_{VR1} & \emptyset & \emptyset \\ \emptyset & A_{VR2} & \emptyset \\ \emptyset & \emptyset & A_R \end{bmatrix}$$

La matriz B se compondrá de dos matrices en serie. Esto es así para separar las señales de control de los actuadores de la perturbación de TT5 que hasta ahora se tomaba como entrada.

Se define $Bext$:

$$Bext = [B|E]$$

En B estarán las componentes de cada una de las submatrices B correspondientes a las relaciones entre cada actuador (VR1, VR2, R) y las salidas (TT2, TT4) quedando una matriz 9x3.

$$B = \begin{bmatrix} B_{VR1} & \emptyset & \emptyset \\ \emptyset & B_{VR2} & \emptyset \\ \emptyset & \emptyset & B_R \end{bmatrix}$$

Y en el E las componentes referidas a TT5 formando una 9x1.

$$E = \begin{bmatrix} B_{VR1,TT5} \\ B_{VR2,TT5} \\ B_{R,TT5} \end{bmatrix}$$

Por otro lado, la matriz C queda formada por las submatrices de cada modelo en columnas formando una matriz 2x9.

$$C = [C_{VR1} \quad C_{VR2} \quad C_R]$$

Finalmente, D será una matriz de ceros de las dimensiones correspondientes.

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

De esta manera, el modelo se podría reescribir como:

$$x(k+1) = Ax(k) + Bu(k) + Ed_{TT5}(k)$$

$$y(k) = Cx(k) + Du(k)$$

6.1.1 Controlabilidad del sistema

La controlabilidad de un sistema indica la capacidad del sistema de llegar a cualquier estado del espacio de estado partiendo de un estado x_0 mediante un vector de señales de control u .

Un sistema es controlable si y sólo si su matriz de controlabilidad R presenta rango n:

$$R = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$$

6.1.2 Observabilidad del sistema

Para la realimentación de estados del sistema se requiere estimar los estados en los que se encuentra el sistema ya que en ocasiones no resultan accesibles mediante un observador de estados. La capacidad de un sistema de determinar un estado dado a partir de las salidas de este se denomina observabilidad. Dicho de otro modo, que a través de las salidas del sistema sea posible conocer el comportamiento de todo el sistema.

Un sistema será observable si la matriz de observabilidad es de rango n:

$$O = [C \quad CA \quad CA^2 \quad \dots \quad CA^{n-1}]^T$$

La estimación de los estados del sistema se hará a través del observador de Luenberger que se detallará más adelante.

6.1 Estructura del modelo

Introduciendo el diagrama del modelo en el bloque que aporta Matlab, se puede representar el camino de las señales de control y salida por el observador y el controlador. Esto en Simulink queda de la forma:

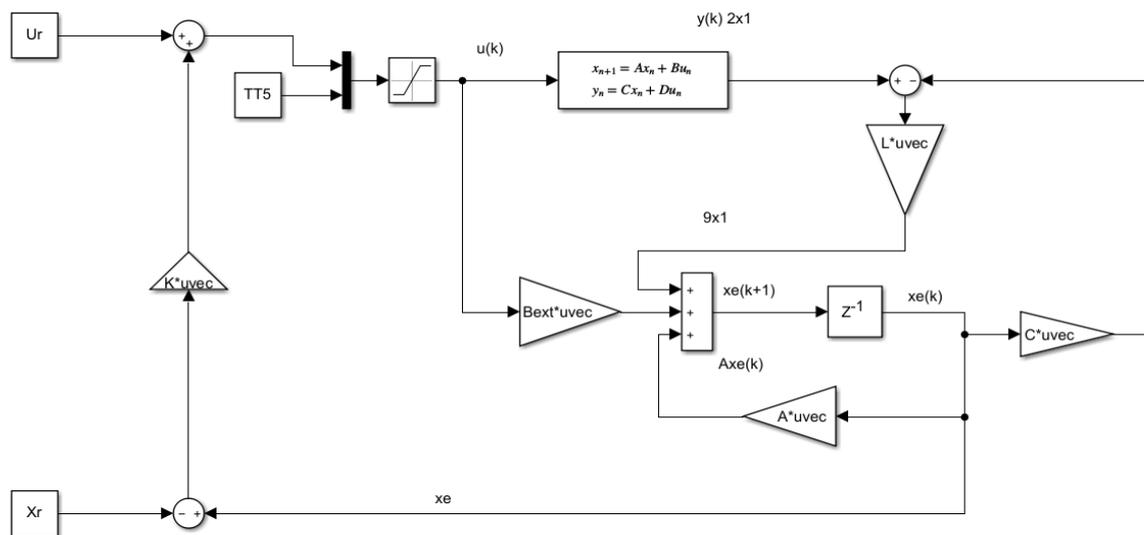


Figura 6.1 Estructura bucle de control

A la hora de controlar el sistema real, se colocará en lugar del modelo lineal el módulo de conexión con la planta real por OPC UA. El cambio los valores absolutos a relativos y viceversa de las señales para su uso en cada etapa del algoritmo se detallará en el siguiente capítulo.

Cómo última indicación sobre la estructura del modelo sobre el que se va a trabajar, recordar que debido al cómo se han determinado las señales que determinan el funcionamiento de los actuadores (porcentajes de apertura y ciclo de trabajo), será necesario saturar dichas señales de control en valores máximos y mínimos presentando unas restricciones en el sistema.

6.1.1 Controlador K

El controlador del sistema tiene una ganancia K que responde a la siguiente expresión.

$$u(k) = K(xe(k) - xr) + ur$$

Hay que tener en cuenta que se trabaja en un sistema multivariable. Por lo que la ganancia tendrá que ser una matriz de tres filas (una por actuador) y 9 columnas (número de estados).

La variable xe representa el estado estimado por el observador.

6.1.2 Observador de Luenberger

El observador de Luenberger se puede ver como un bloque cerrado que, a partir de los estados estimados anteriores, las entradas y las salidas del sistema es capaz de calcular una estimación para el instante $k+1$.

Esta estimación del estado del sistema será la que se realimentará, comparándose con la referencia a través del controlador.

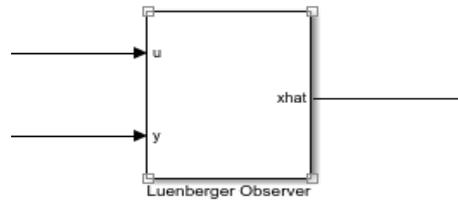


Figura 6.2 Bloque del observador

Ese cálculo se realiza a través de las ecuaciones propias del modelo de espacio de estados

$$x(k+1) = Ax(k) + Bu(k)$$

Y a una corrección de la estimación ponderada por una ganancia L que si está debidamente diseñada converge al estado real del sistema. Con todo ello la expresión del observador queda:

$$xe(k+1) = Axe(k) + Bu(k) + L(y(k) - C(xe(k)))$$

6.3 Principio de separación

Definiendo los errores de seguimiento y estimación como:

$$e_t(k) = x(k) - x_r$$

$$e(k) = x(k) - xe(k)$$

Se puede describir el sistema en bucle cerrado en los siguientes términos:

$$\begin{bmatrix} e_t(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} e_t(k) \\ e(k) \end{bmatrix}$$

Esto indica que para sistemas lineales la sintonización del controlador y del observador que se encargan de los errores de seguimiento y estimación respectivamente, son independientes.

Por un lado, la dinámica del error de seguimiento será dependiente de A, B, y la matriz ganancia del controlador K. El cuánto a la dinámica del error de estimación su dinámica vendrá definida por A, C y la matriz ganancia del observador L.

Se verá como la sintonización de estas dos ganancias se pueden resolver de la misma manera usando un LQR (Linear Quadratic Regulator).

6.4 Cálculo de la ganancia K

Para el cálculo de la ganancia K del controlador se ha decidido por usar un LQR. El LQR obtiene una ganancia K tal que minimiza la siguiente función de coste:

$$\min_{u(k)} \sum (x(k) - xr)^T Q (x(k) - xr) + (u(k) - ur)^T R (u(k) - ur)$$

Donde las matrices Q y R son matrices de ponderación con las que se puede dar más peso a cada una de las variables que nos interese minimizar: la diferencia al estado de referencia o el esfuerzo de control. En un primer caso se optará por normalizar ambas diferencias.

Haciendo uso de la expresión:

$$Q = C^T Q_y C$$

Se puede operar el problema en términos de salida en lugar de estados cuya representación es ficticia y en ocasiones poco intuitiva.

$$\min_{u(k)} \sum (y(k) - yr)^T C^T Q_y C (x(k) - xr) + (u(k) - ur)^T R (u(k) - ur)$$

Los valores para las matrices han sido los siguientes.

$$Q_y = C^T \frac{100}{6} \begin{bmatrix} 50 & 0 \\ 0 & 200 \end{bmatrix} C$$

$$R = \frac{1}{20} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Como se ha visto en el apartado 6.4, la dinámica del error de seguimiento en bucle cerrado viene definido por:

$$A + BK$$

Por lo que el problema se resuelve con las siguientes entradas para la función dlqr de Matlab.

$$K = -dlqr(A, B, Q, R);$$

Para comprobar el funcionamiento del controlador, se intenta controlar el sistema lineal a través de sus estados. Para ello se sustituye en el sistema C por una matriz de identidad de manera que la salida correspondan a sus estados.

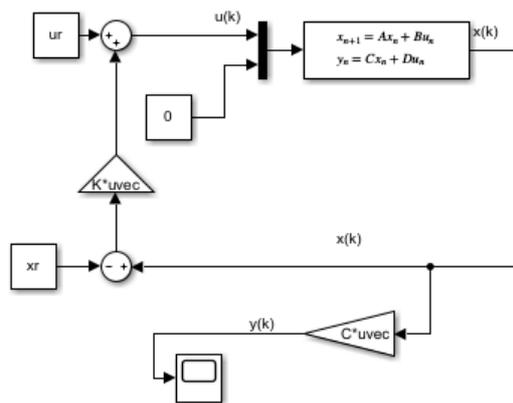


Figura 6.3 Bucle de control por estados del sistema

Se simula el sistema para una referencia arbitraria para los estados y las señales de control y se comprueban los resultados. Para obtener las salidas reales del sistema, se multiplica por C el vector de estados.

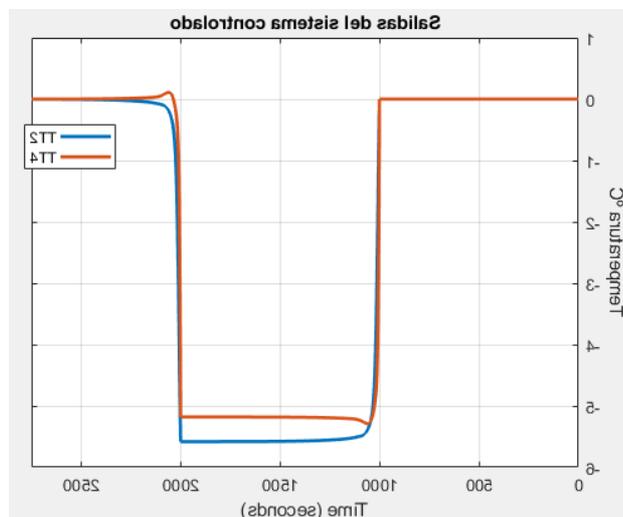


Figura 6.4 Respuesta del sistema controlado por estados

La validez del controlador se observa al ver como desde un estado arbitrario cualquiera vuelve correctamente al origen cuando se fija la referencia a 0.

6.5 Cálculo de ganancia L

Para el cálculo de la ganancia del observador se va a usar la misma metodología. Esto es posible debido a la dualidad del problema que resulta de calcular la ganancia que minimice cierta expresión de errores cuadráticos.

Si la dinámica del error de seguimiento venía definida por:

$$A + BK$$

La del error de estimación era de la forma:

$$A - LC$$

Si se opera un poco y se hace la traspuesta de la expresión

$$(A - LC)^T = A^T - C^T L^T$$

Se puede ver cómo el problema adopta una forma similar, lo que permite repetir el procedimiento anterior.

De esta manera, las entradas para calcular la ganancia L serán A traspuesta, C traspuesta y las matrices de ponderación.

Se definen las matrices de ponderación:

$$Q_0 = \frac{1}{1000} I_{9 \times 9}$$

$$R_0 = 1000 \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$$

Con todo lo necesario para el cálculo de L se ejecuta la función dlqr.

$$L = \text{dlqr}(A', C', Q_0, R_0)';$$

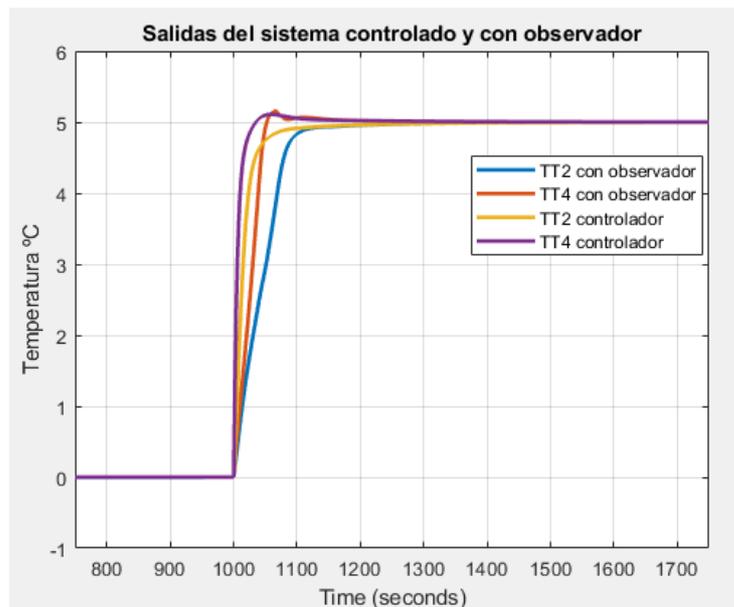


Figura 6.5 Respuesta del sistema con observador de Luenberger

En esta etapa de la sintonización, se busca que la respuesta del sistema controlado a través del observador se parezca lo más posible la que se obtiene al controlarlo a través de los espacios reales del modelo. Como se puede ver en la figura las respuestas se aproximan dándole validez al resultado-

6.6 Obtención de referencias a través de un SSTO

Para la realización de esta estrategia de control que se desea llevar a cabo se requieren sacar los estados del sistema y las señales de control de referencia que corresponden a una referencia en términos de salida dada.

Estos serán, los valores que en régimen permanente satisfagan la siguiente expresión.

$$\begin{aligned}x_s &= Ax_s + Bu_s + Ed_s \\r &= Cx_s\end{aligned}$$

Como se puede ver, la obtención de los valores de los estados y de las señales de control en el equilibrio para una referencia dada, es posible a través de la resolución de un problema puramente algebraico en el que también aparece la perturbación TT5. Reformulando el problema:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} -Ed_s \\ r \end{bmatrix}$$

Para este problema podría no haber solución o por contrario haber infinita. Por lo que habrá que caracterizar el punto de equilibrio.

Analizando el problema por filas se puede obtener la siguiente expresión:

$$[A - I \ B] \begin{bmatrix} x_s \\ u_s \end{bmatrix} = -Ed_s$$

Desde la que se puede llegar a una dependencia del punto de equilibrio lineal.

$$\begin{bmatrix} x_s \\ u_s \end{bmatrix} = M_1 v + M_2 d_s$$

Donde M_1 y M_2 son aquellas matrices que satisfacen:

$$\begin{aligned}[A - I \ B]M_1 &= \emptyset \\ [A - I \ B]M_2 &= \begin{bmatrix} -Ed_s \\ \emptyset \end{bmatrix}\end{aligned}$$

Analizando ahora el régimen permanente en términos de salida.

$$r = cx_s$$

Y aplicando la expresión:

$$r = CM_{1x}v + C_{2x}d_s$$

Observando las expresiones anteriores, M_1 es de tamaño 13×9 (mxn), el vector v de 9×1 y CM_{1x} de 2×9 (pxm). Al ser $p < m$ existen infinitas soluciones para el problema. Por ello el cálculo se transforma en un problema de optimización en el que se volverá a intentar minimizar una función de costes.

$$\begin{aligned}\min_{x_s | u_s} \quad & u_s^T R u_s + x_s^T Q x_s \\ x_s &= Ax_s + BU_s + Ed_s \\ r &= Cx_s \\ u_{min} &< u_s < u_{max}\end{aligned}$$

La compensación de la perturbación queda definida por la matriz E, calculada en la obtención del modelo multivariable a partir de las matrices B de los submodelos y tomando TT5 como una perturbación en lugar de entrada del sistema.

El resultado de dicho problema se puede obtener a través de la función 'quadprog' de Matlab.

Con este problema solucionado, el sistema es capaz de calcular los valores de los estados y las señales de equilibrio que corresponden a una referencia dentro de los límites de funcionamiento.

A esta etapa de optimización para la obtención de unos estados y señales de control de referencias óptimos para una situación del sistema dada se denomina SSTO (steady-state target optimización).

La implementación de dicho algoritmo se ha hecho a través de una función interpretada en Matlab la cual inicializa los datos necesarios con los datos del sistema y realiza los cálculos pertinentes. Su código está disponible en el Anexo D. Para probar su funcionamiento se ha realizado un ensayo con distintos escalones en la referencia viendo cómo el sistema es capaz de generar los estados y las señales de control de equilibrio necesarias.

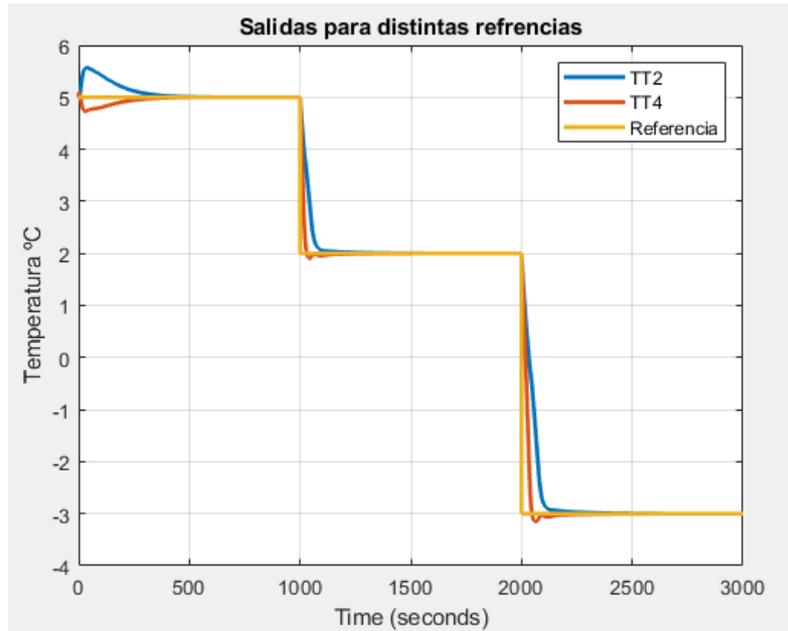


Figura 6.6 Respuesta para distintas referencias del SSTO

6.6.1 Análisis de estados en cambios de referencia

Es interesante ver como se estiman los estados por el observador. Para ello se van a representar los estados estimados a lo largo de la simulación y los estados por los que pasa la planta (modelo lineal).

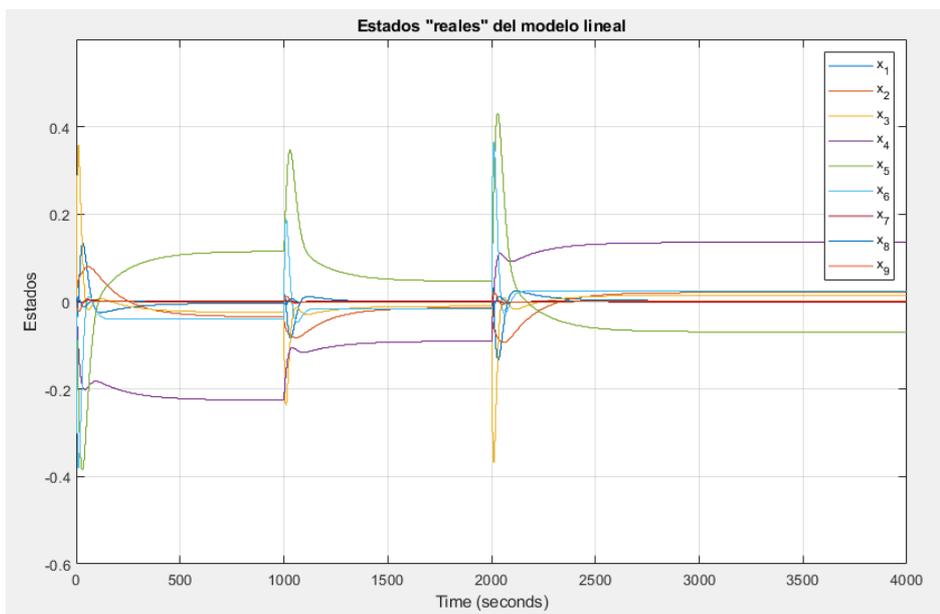


Figura 6.7 Estados reales planta lineal

Se puede ver que la mayoría de los estados permanecen entorno al 0 y que pese a haber diversidad en cuánto a las dinámicas de cada uno de ellos, todos llegan a un régimen permanente.

El observador, por otro lado, produce la siguiente estimación de los estados.

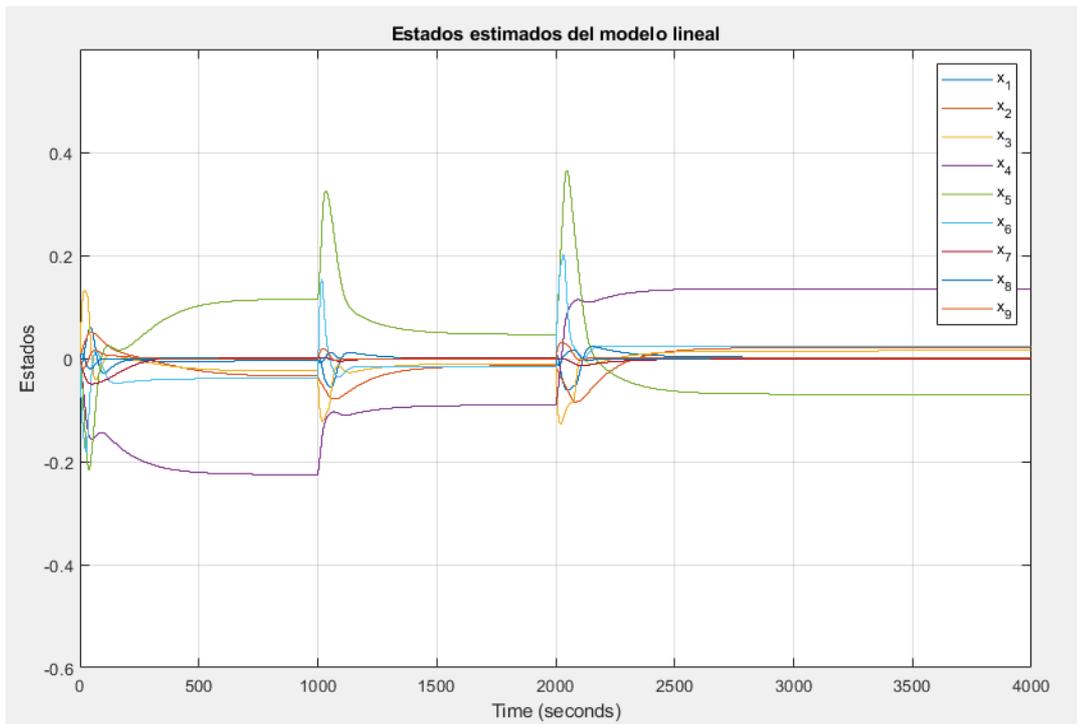


Figura 6.8 Estados reales planta lineal

Se observa una clara similitud entre los valores reales y los estimados. Para calcular la exactitud de las estimaciones se puede representar el error de estimación a lo largo de la simulación.

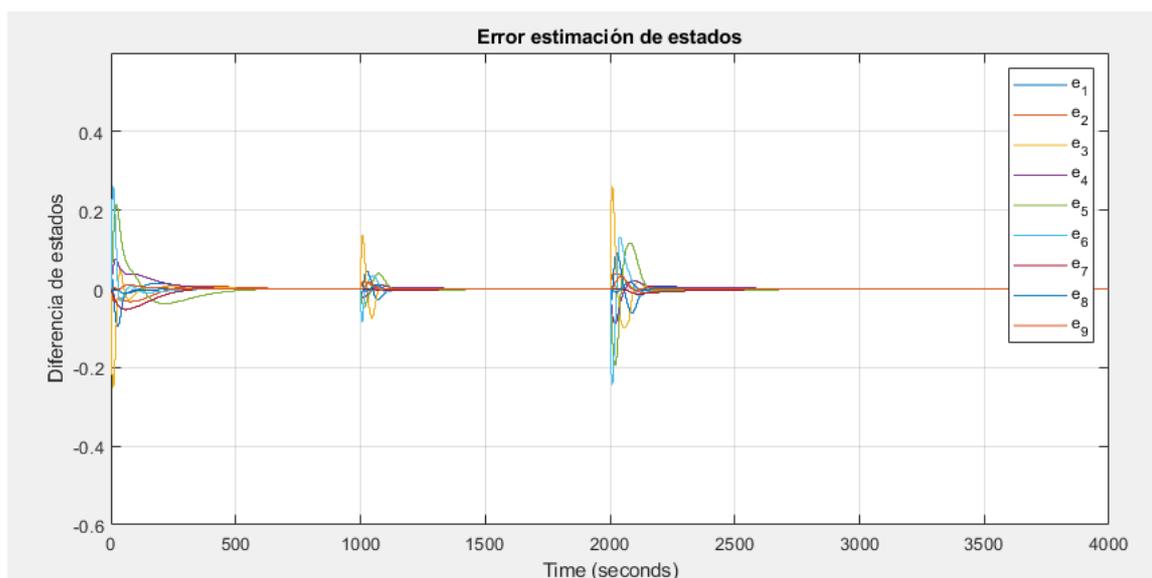


Figura 6.9 Estados reales planta lineal

Como era de esperar, los máximos en los errores de estimación se producen en los instantes tras los cambios en la referencia. Rápidamente se solucionan pasando a un error de estimación casi nulo en un tiempo

relativamente corto.

6.6.2 Análisis de la incorporación de TT5 en el SSTO

El cálculo SSTO tiene como entrada la propia TT5 por lo que se deberá hacer ‘online’, es decir, en cada iteración se calculará para obtener nuevas referencias. Hasta ahora, los ensayos se han realizado con esta perturbación nula, pero es conveniente analizar cómo afecta TT5 y que se represente lo importante que resulta incorporarla al cálculo del SSTO. Modelizando la TT5 como un seno de 2 de amplitud y 1/350 Hz se puede comprobar la validez de la compensación.

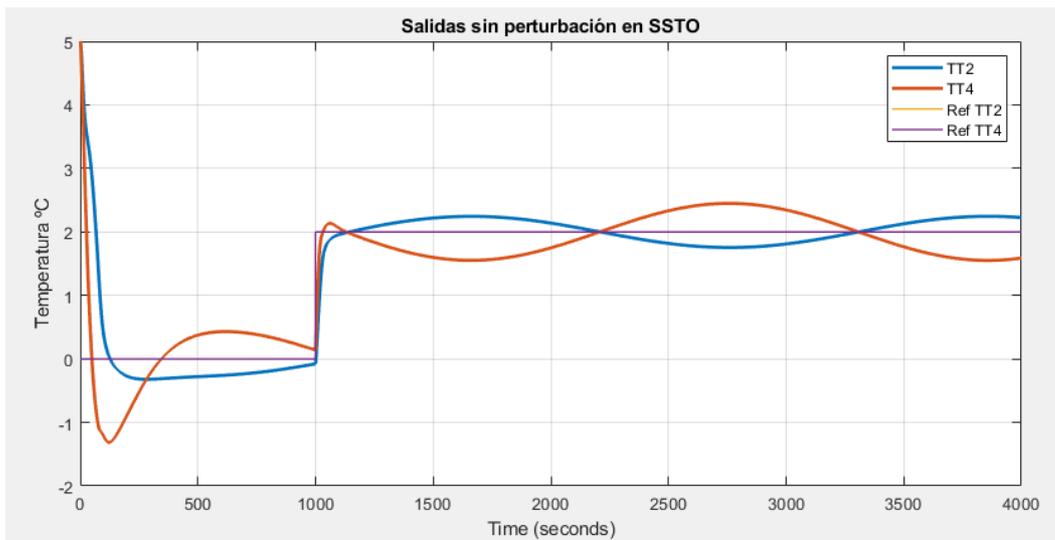


Figura 6.10 Respuesta del sistema sin introducir TT5 en el SSTO

Como vemos el sistema no es capaz de anular por completo a los cambios en TT5. Con la perturbación como entrada en el SSTO el comportamiento cambia radicalmente.

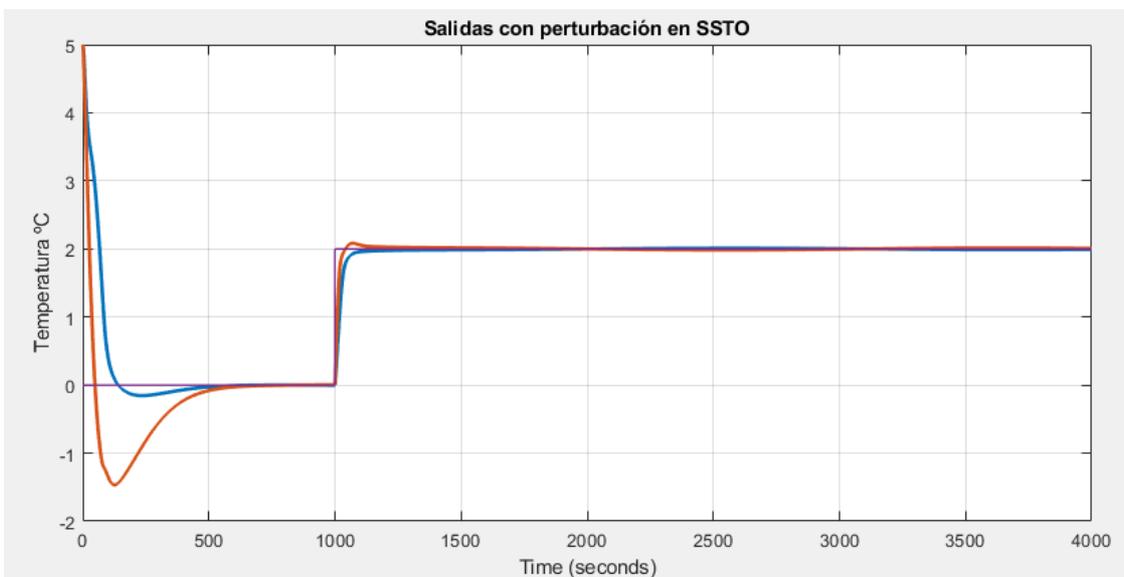


Figura 6.11 Respuesta del sistema introduciendo TT5 en el SSTO

6.7 Mejora de la caracterización de TT5

Sin embargo, experimentalmente se aprecia que la señal de las variaciones de TT5, si bien son claramente periódicas, no se asemeja del todo a la forma de un seno.

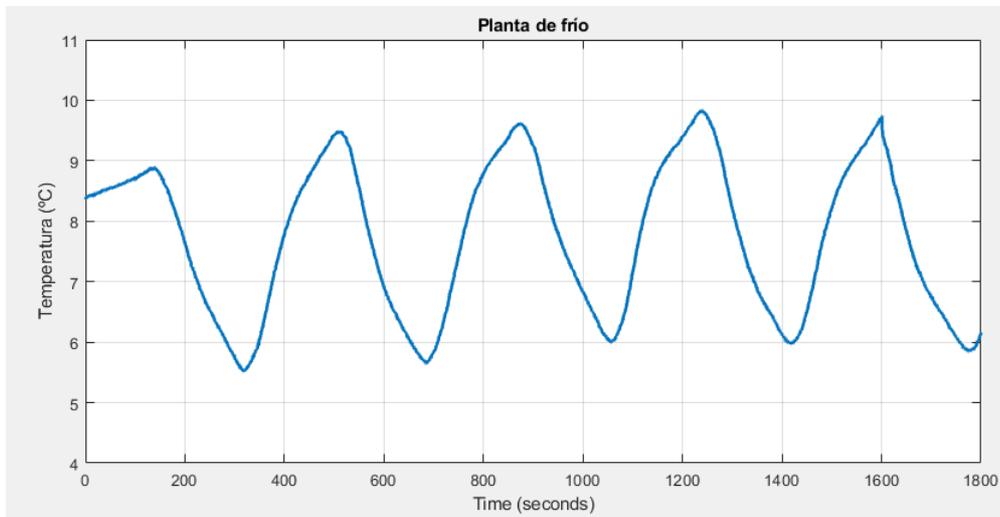


Figura 6.12 Forma de onda de TT5

Para modelar de una forma más parecida a la realidad esta señal se ha decidido calcular, a partir de uno de los periodos, una señal que repita la misma secuencia. Para implementar dicho cálculo en Simulink se puede tratar la señal como la salida de un modelo lineal cuya salida repita una secuencia inicial representada en la condición inicial.

$$xr^+ = Agxr_0$$

Siendo La matriz Ag de la forma:

$$Ag = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Y en xr0 se almacena la secuencia que se desea repetir. De cualquiera de los ensayos realizados a lo largo del trabajo se anotan 15 valores que describan un período de la señal y se simula.

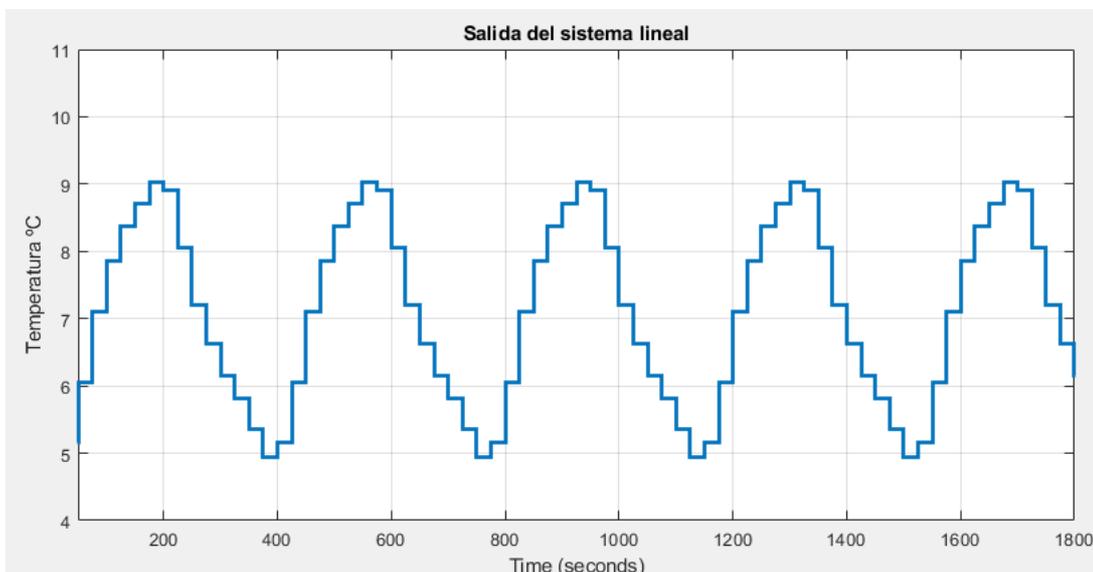


Figura 6.13 Señal TT5 generada

Para evitar ese escalonamiento, se realizan todos los cálculos previamente a la simulación en un script y se exporta la secuencia de datos a través del bloque 'To Workspace'. Este bloque aporta los mismos valores que el sistema lineal, pero interpola entre los dos datos más próximos para aquellas iteraciones para los que no hay valor de la señal.

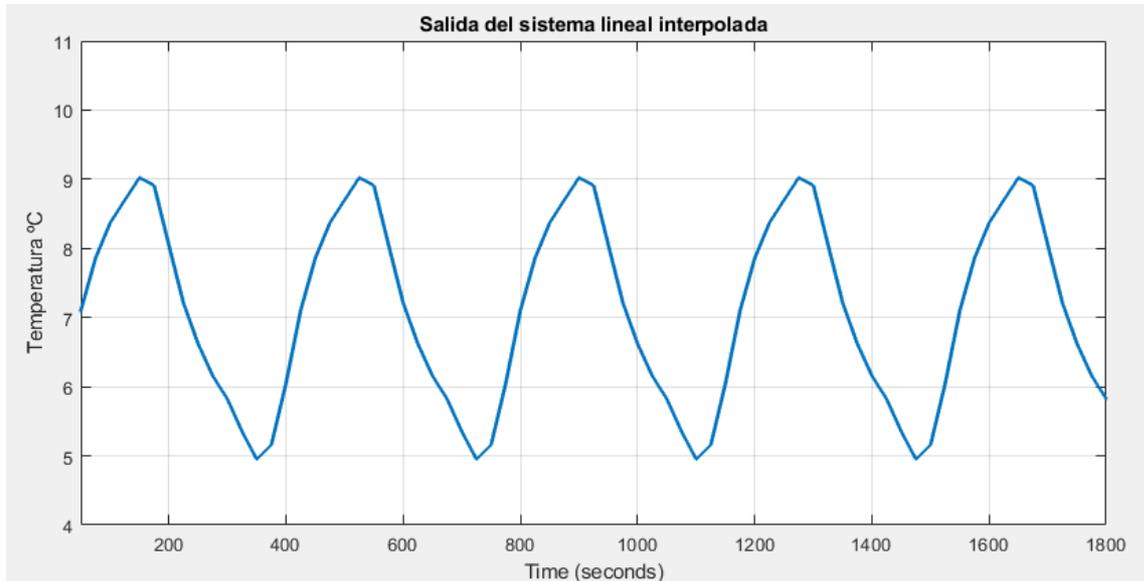


Figura 6.14 Respuesta del sistema introduciendo TT5 en el SSTO interpolada

Se tiene ya una muy buena aproximación de TT5 ya se va a comprobar si el sistema simulado, al igual que anulaba la perturbación con forma de seno, es capaz de anular esta forma de onda. La implementamos en el modelo y se simula.

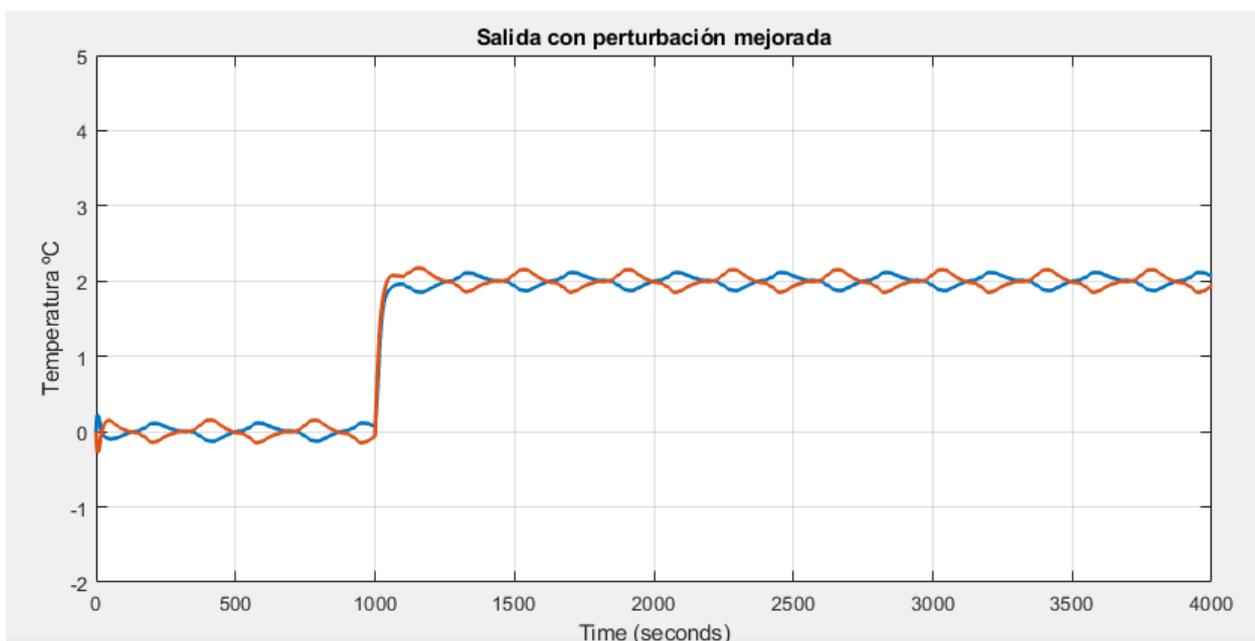


Figura 6.15 Salida del sistema con la perturbación mejorada

Se puede ver que pese a presentar un comportamiento lejos del ideal, las salidas se quedan en un rango cercano a la referencia. El cálculo del SSTO tiene limitaciones pues realiza los cálculos suponiendo una perturbación constante. En este caso, la perturbación es distinta para cada iteración por lo que los cálculos en cada tiempo de muestreo inhabilitan los de la iteración anterior. Existen metodologías de control para anular el error en régimen que solucionarían la oscilación que tiene la señal que quedan fuera de los objetivos de este trabajo

6.8 Resultados de la simulación completa

Una vez completado este capítulo se tienen todas las etapas, cálculos y parámetros necesarios para la implementación completa del modelo de la planta multiprocesos del laboratorio. Haciendo un pequeño resumen, se ha conseguido un modelo lineal del sistema a partir de los ensayos de identificación, se ha estructurado el modelo de control con el observador y el controlador, se han diseñado las ganancias de éstos últimos y se ha diseñado un SSTO capaz de obtener las referencias de los estados y las señales de control necesarios para el problema de seguimiento del sistema.

Todo ello se ha conectado en el modelo de Simulink debidamente para cumplir con todos los requisitos del compilador.

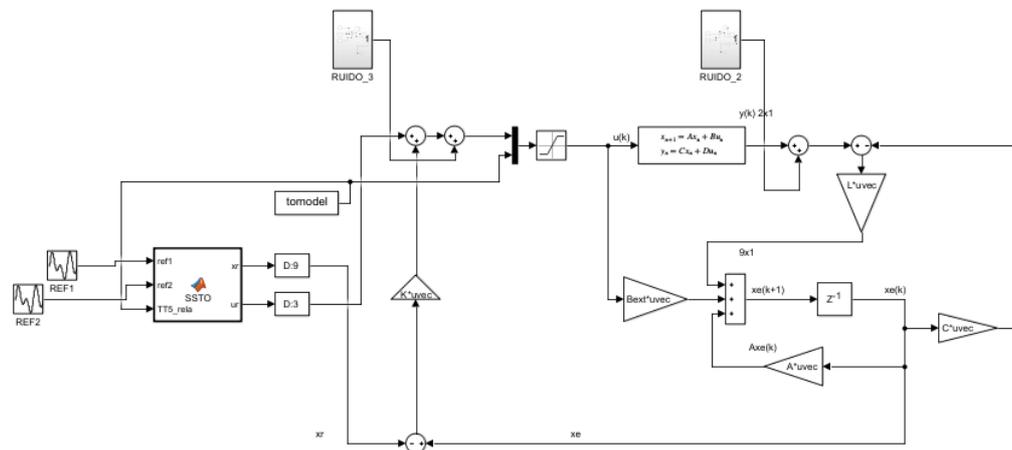


Figura 6.16 Modelo del bucle de control completo

Se le han añadido al modelo dos fuentes de ruido a la entrada a la salida y a la entrada del sistema para acercarlo a la realidad. Hay que comprobar el funcionamiento del ruido a la salida ya que la ganancia del observador la amplifica y puede dar problemas en el funcionamiento del sistema.

TT5 quedará modelizada en la simulación con el bloque "From Workspace" de Matlab a través del vector de datos "tomodel"

Se realiza un ensayo completo para comprobar el funcionamiento de todo el conjunto y así validar el modelo para su posterior ensayo en la planta multiprocesos.

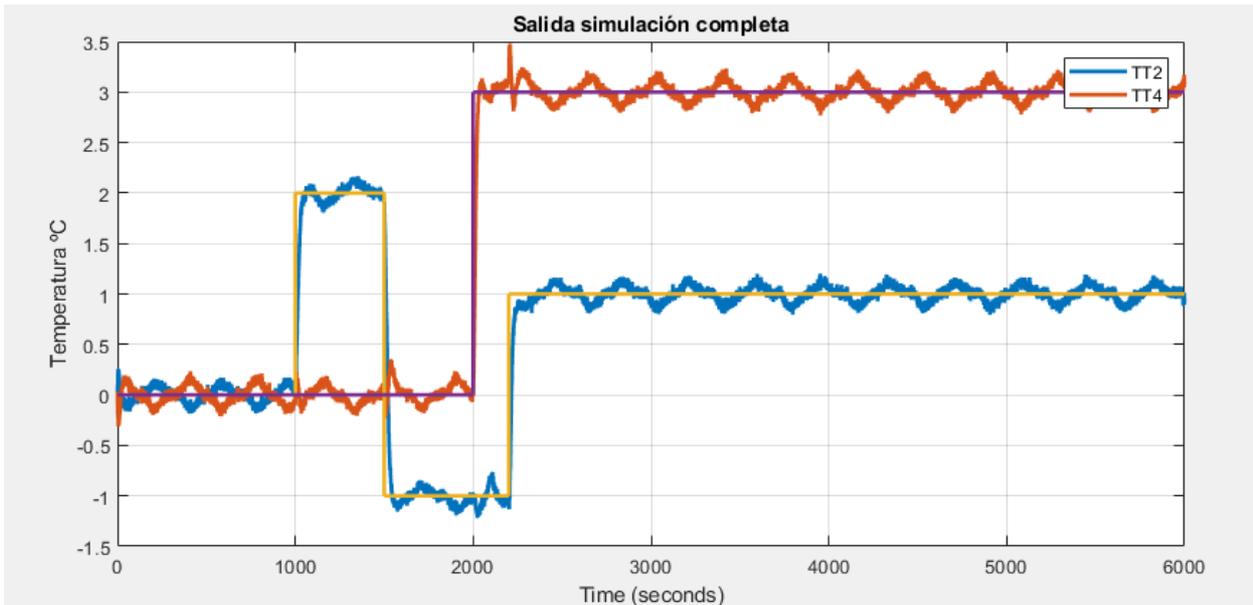


Figura 6.17 Respuesta del sistema simulado

Se puede observar el gran comportamiento que presenta el sistema. Cada salida tiene una dinámica bastante independiente de la otra en la que los cambios en la referencia de una de ellas son rápidamente compensados en la otra como buen sistema multivariable.

Las señales de control serán de la forma:

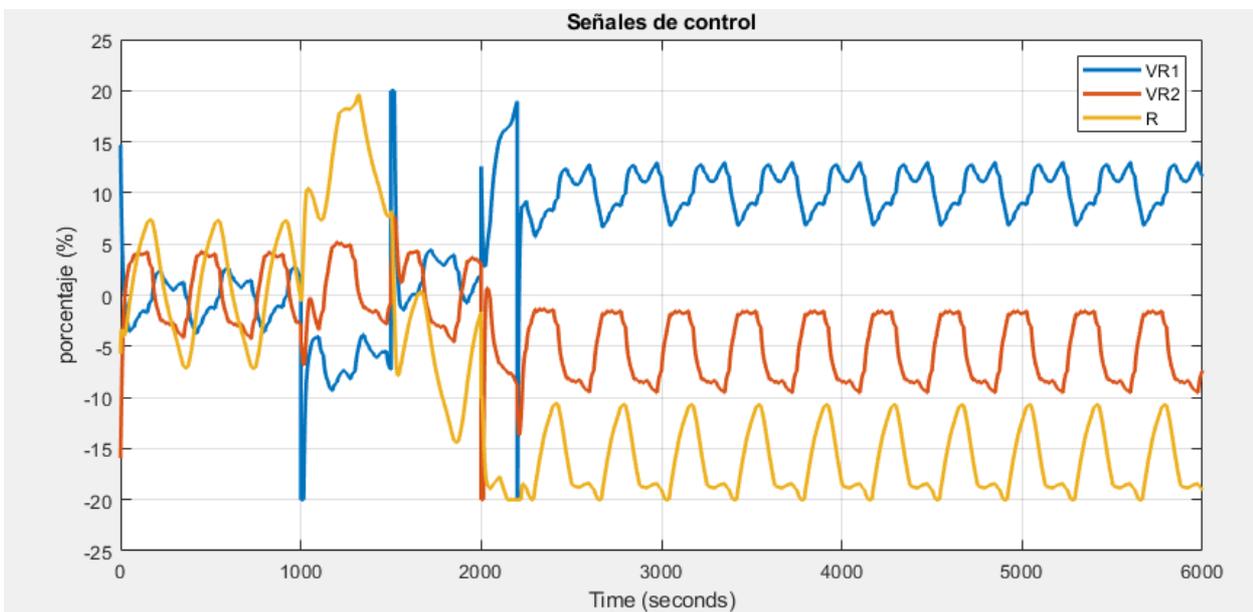


Figura 6.18 Respuesta del sistema simulado

Como se puede ver, en ocasiones son señales muy bruscas, ello se puede solucionar, si es necesario, con un filtro en la etapa de implementación. Es importante destacar, que este será un problema de control periódico. El régimen permanente al que aspira el sistema no presentará unas señales de control constantes, sino que oscilarán con igual período que la oscilación de la perturbación periódica.

Modificando la matriz de ponderación R del cálculo de las referencias del SSTO se pueden priorizar las actuaciones de unos actuadores determinados por encima de otros. Esto puede ser interesante desde un punto de visto económico en el que un gran esfuerzo de control de determinadas entradas suponga un gran coste de recursos.

Con este resultado se da por completada la etapa de simulación de la planta y se da paso a la de implementación de la estrategia descrita a lo largo del capítulo en la planta real multiprocesos.

7 IMPLEMENTACIÓN DEL CONTROL EN LA PLANTA REAL

Ya se ha conseguido trabajar tanto con la planta real como con un modelo lineal en variables de espacio de estados. Para poder realizar indistintamente experimentos con ambos sistemas se ha diseñado un modelo Simulink con un subsistema variante el cual permite permutar entre ambas configuraciones. Esto permitirá experimentar en los dos sistemas a igualdad de condiciones y se podrán identificar la exactitud que resultan las salidas simuladas respecto a las reales.

7.1 Modelo Simulink

Tal y como se ha introducido, el bloque principal del modelo será el subsistema variante. Este es un bloque con el que cuenta Simulink, que tiene la capacidad de poder elegir cuál de los subsistemas que alberga en su interior se ejecuta. Esta decisión se hace a través de una variable externa inicializada antes del experimento.

Para tener también homogeneidad en cuánto a la elección de los puntos de funcionamiento (pueden variar en función de las condiciones externas) se han sustituidos estos valores por variables globales (TT50, TT20 y TT40) de manera que se inicializarán antes de los experimentos y se aplicarán en aquellos puntos que sean necesarios.

Hay que tener en cuenta que las entradas y las salidas que se comunican con la planta a través del OPC UA son absolutas y que para su uso en el controlador y el observador habrá que sumar o restar según corresponda el punto de funcionamiento.

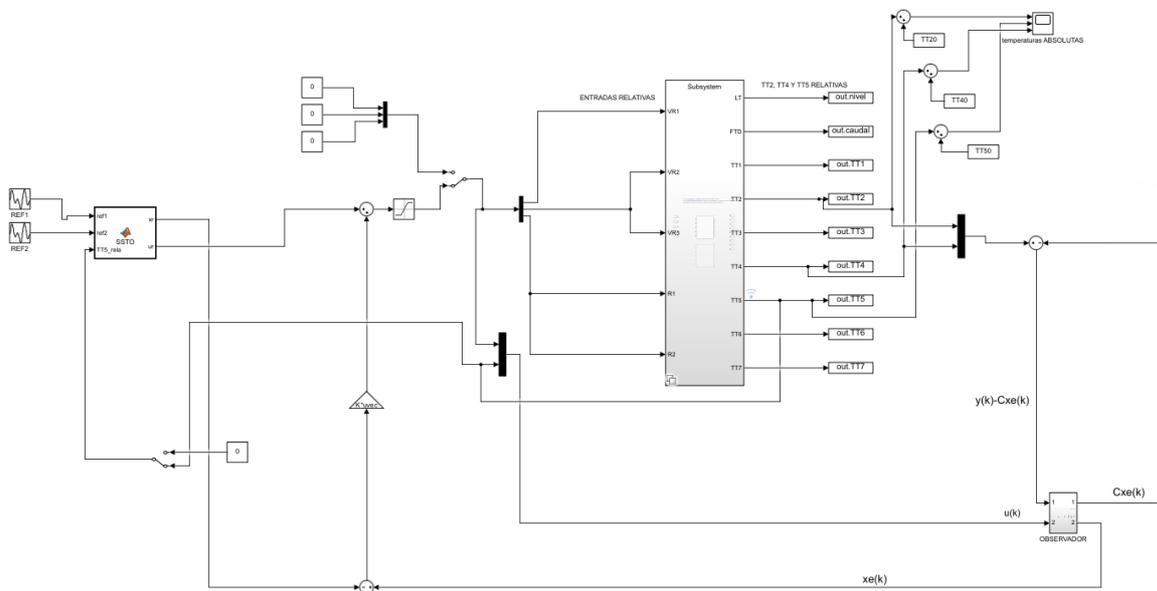


Figura 7.1 Modelo Simulink para los ensayos

Se han quitado los bloques de representación y almacenamiento de datos para una mayor claridad en los caminos de las señales en el modelo.

Se han añadido unos bloques que pueden dar valores a los actuadores a través de un interruptor con los que se podrá colocar la máquina en el punto de funcionamiento en caso de ser necesario. Así como dar tiempo para

que la diferencia entre el estado inicial del modelo lineal y del sistema real se compense.

Para el bloque del modelo lineal:

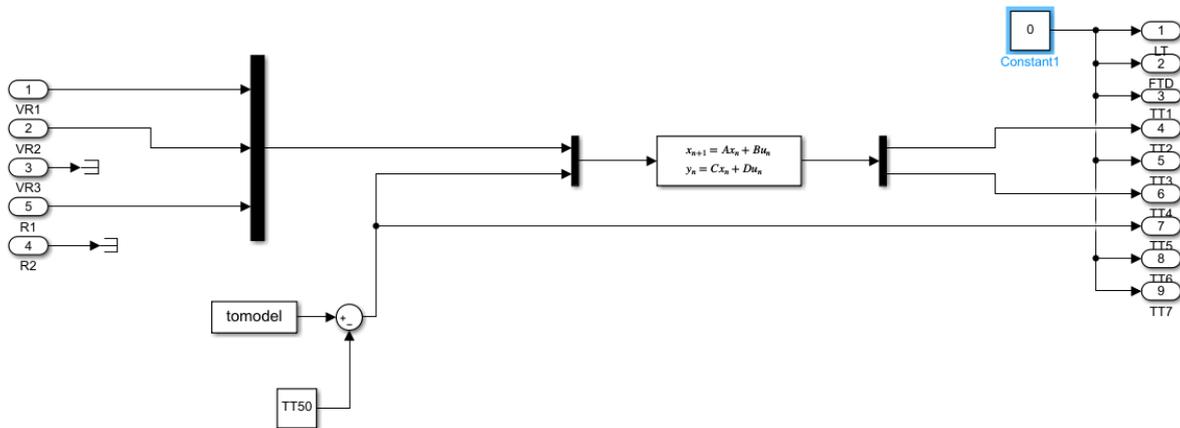


Figura 7.2 Subsistema modelo lineal

Para la realización de experimentos con el modelo lineal se hace uso de la generación artificial de la señal TT5 y se realimentará fuera del subsistema a través de la salida correspondiente a TT5 del bloque. Al estar sólo modelada las salidas TT2 y TT4 el resto se ponen a cero y en este caso no aportaran información alguna.

El bloque de comunicación con el OPC UA queda de la forma:

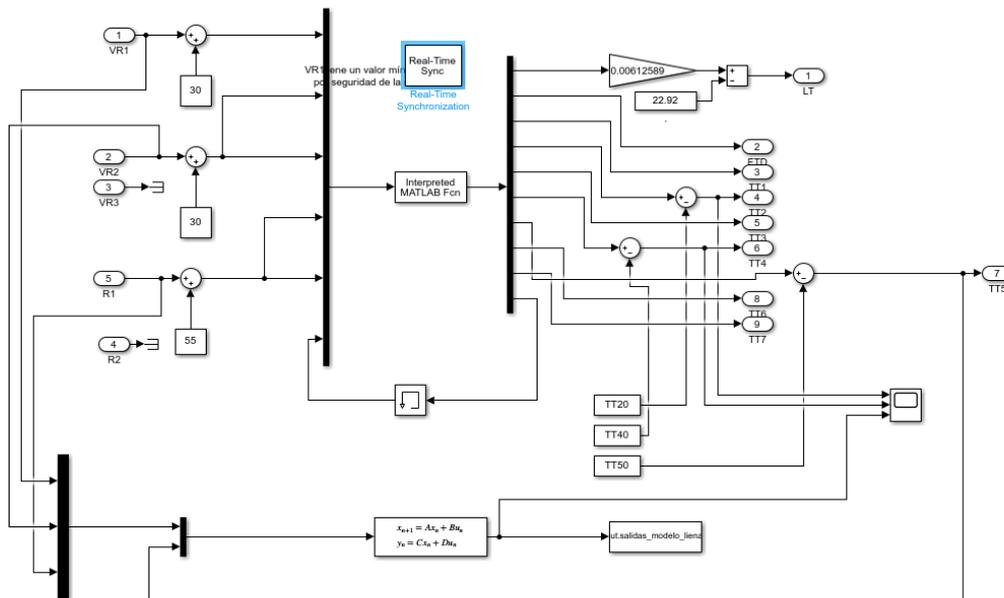


Figura 7.3 Subsistema modelo modelo real y lineal

Se hacen los cambios a variables absolutas y relativas en los puntos que corresponden, por lo demás el funcionamiento es el mismo. Importante introducir el bloque de “Real Time Sync” dentro de este subsistema ya que el único caso en el que nos interesa la simulación en tiempo real es en el de conexión de la planta real.

En este bloque se aprovechan las entradas generadas por la realimentación de las salidas de la planta real y se

introducen también en el modelo lineal. De esta manera se pueden ver las diferencias que presentan las salidas del modelo lineal y de la planta para unas mismas entradas.

7.2 Ensayo con señales de control relativas a 0

En primer lugar, se ha hecho un experimente bastante sencillo. Se han puesto las señales de control relativas a 0 para ver si el sistema se mantiene en el punto de equilibrio.

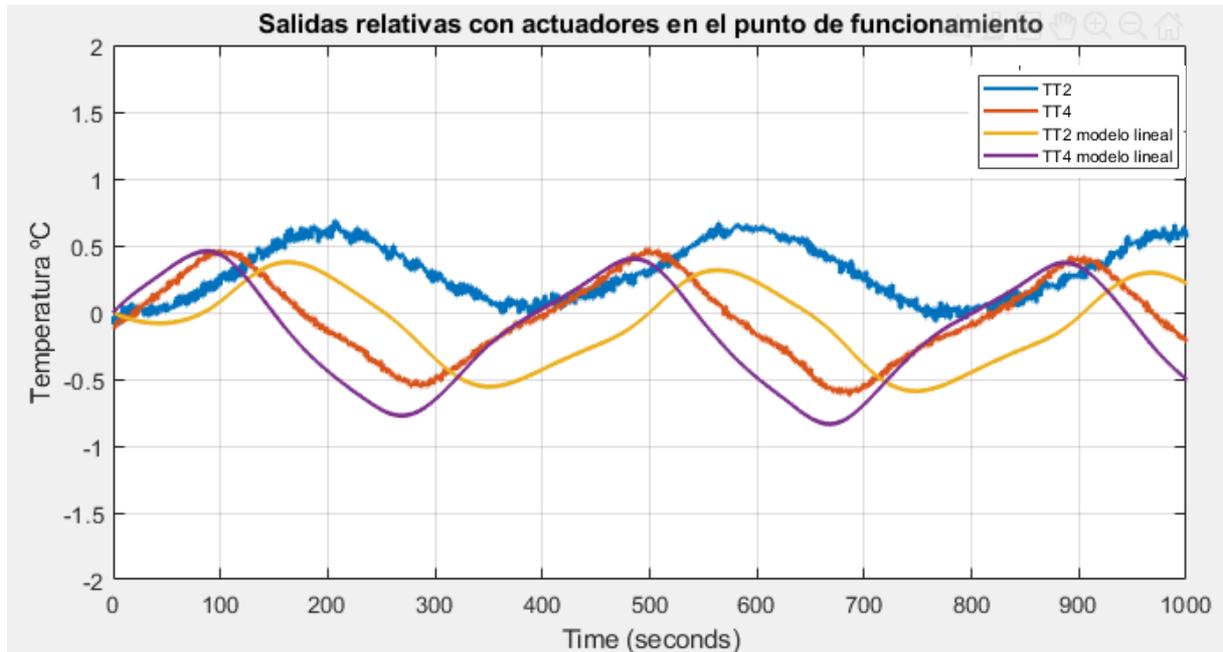


Figura 7.4 Salidas relativas con señal de control relativa nula

Se puede ver como sí que se mantiene el punto de equilibrio. Comparando las señales obtenidas, se puede ver que el modelo lineal al menos sí que es capaz de imitar la influencia de TT5 en las señales de salida de la planta. Esto se complicará con señales del control variantes en el tiempo, pero es un buen punto de comienzo.

7.3 Ensayo en el punto de funcionamiento

Para el siguiente paso en el control de la planta. Se conecta el SSTO al sistema y se realimenta a través de la estimación de estados calculada por el observador. Los resultados son los siguientes.

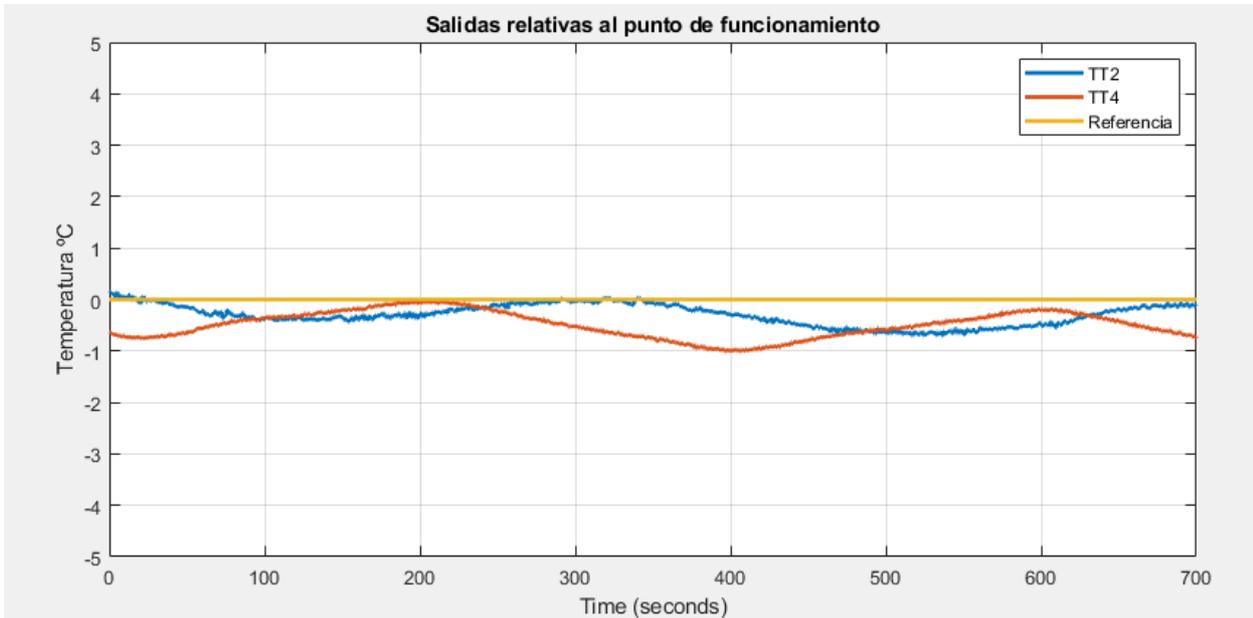


Figura 7.5 Salidas en el punto de funcionamiento

Si bien existe un error en régimen permanente lejos de ser nulo, la verdad que el sistema mantiene las temperaturas dentro de un rango relativamente acotado. La perturbación producida por TT5 no se compensa del todo pese a los esfuerzos de las señales de control.

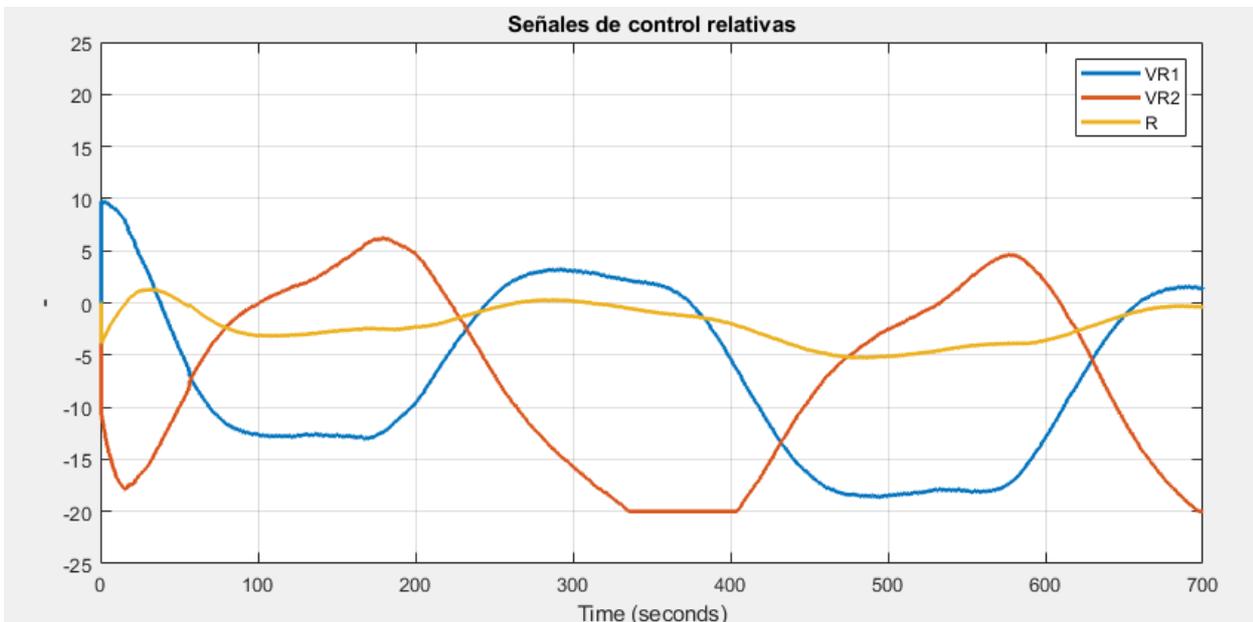


Figura 7.6 Señales de control relativas

Ya se ha comentado que este problema en su régimen permanente no presentará variables con valores constantes. Eso provocará una actuación periódica mantenida en el tiempo y por tanto las señales de salida tampoco serán del todo constantes en el tiempo.

7.4 Ensayo ante cambios en la referencia

En último lugar se realiza el ensayo con cambio de referencias y con el SSTO conectado. Repasando un poco las etapas de las que consta el control tienen lugar los siguientes cálculos. Se le introducirán al SSTO las referencias elegidas (figura 7.7) con las que calculará los estados y las señales de control de referencia oportunas de cada instante de tiempo. A través del error en entre el estado estimado a través del observador y el de referencia se calcula con el controlador la señal de control correspondiente.

Con todas las etapas por fin conectadas se han obtenidos los datos de las salidas controladas cuya representación es la siguiente:

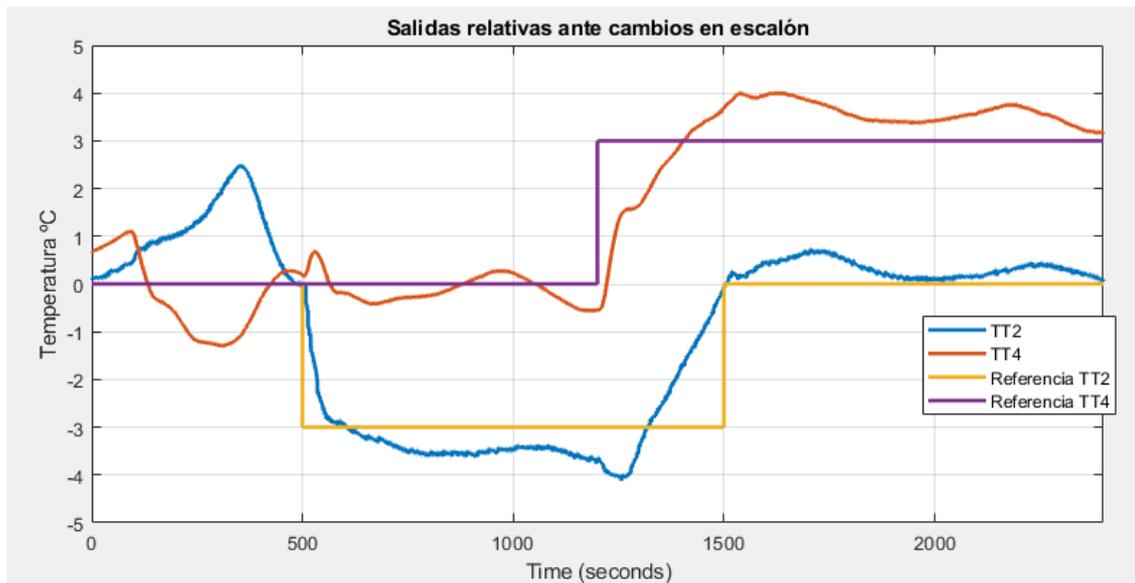


Figura 7.7 Salidas relativas ante cambios en la referenci

Pese a presentar un comportamiento claramente mejorable el sistema es capaz de seguir algunas de las referencias. Quizás un comportamiento peor en torno a 0 que cuando las salidas intentan establecerse en la referencia.

Se analiza a continuación las diferencias entre las salidas del modelo y de la planta real.

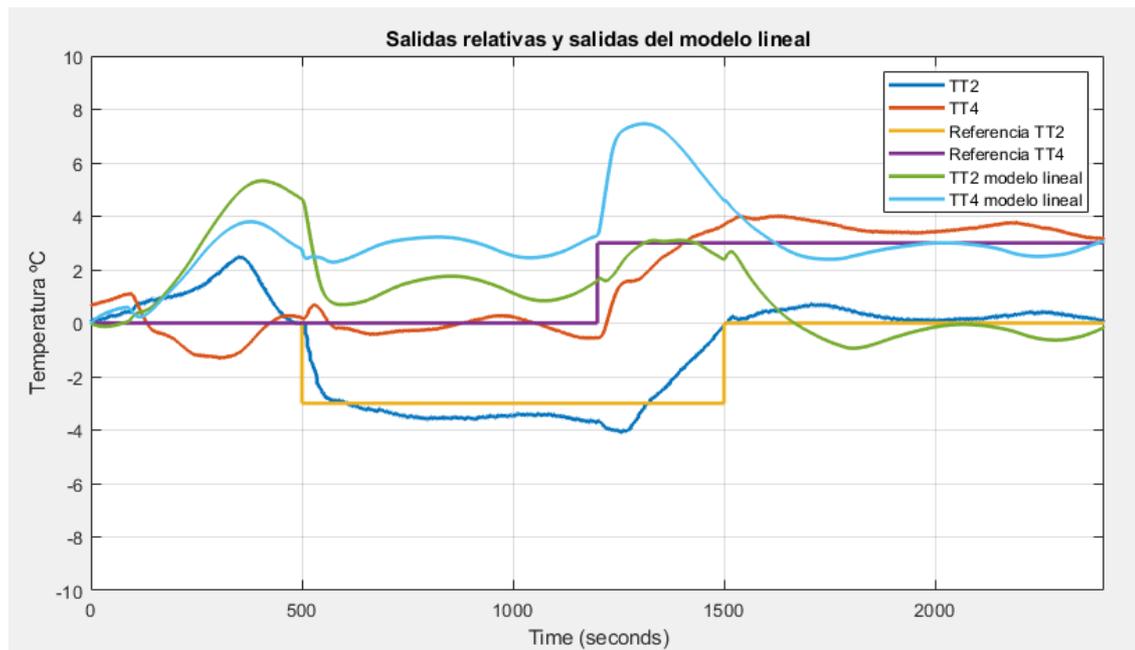


Figura 7.8 Comparación salidas del modelo lineal y de la planta real

Pese al comportamiento aceptable del control, se puede observar que el comportamiento del modelo se aleja de comportamiento que presenta sistema real. Parece que es suficientemente bueno para producir estimaciones del estado coherentes, pero no como para que la salida resulte similar a la del sistema real. Estas diferencias parecen que se alivian un poco en el régimen permanente de puntos de operación distintos a 0. Esta es la prueba más clara de lo mejorable que resulta el modelo lineal. Con un modelo más exacto y que reflejara mejor el comportamiento del sistema real tanto las respuestas de las salidas como la diferencias entre estas y las salidas simuladas mejorarían considerablemente.

En el experimento, los 3 actuadores trabajan simultáneamente para conseguir el control sobre las dos salidas del sistema.

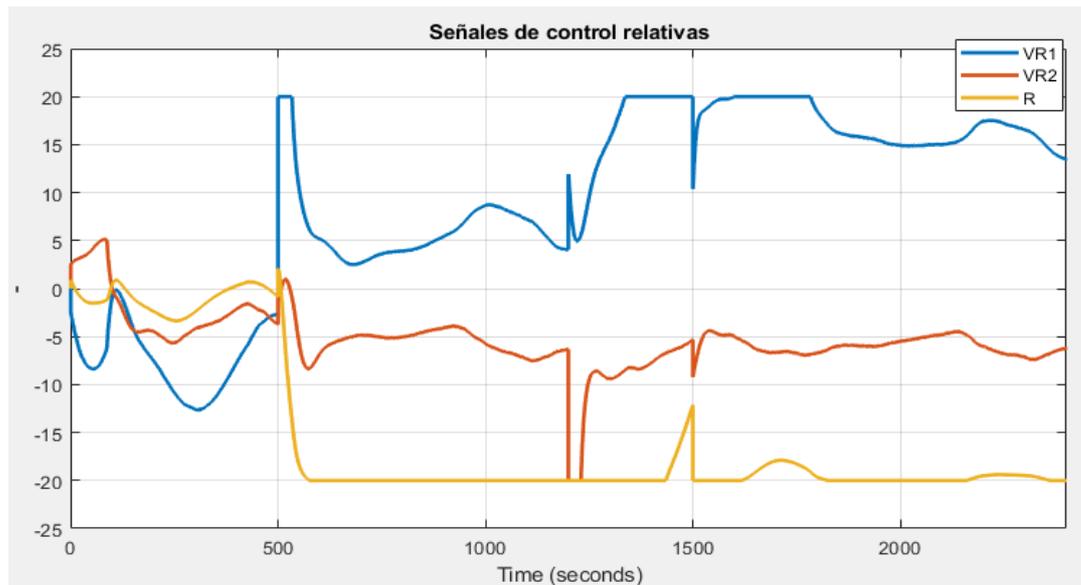


Figura 7.9 Señales de control calculadas

Se puede ver cómo para ciertos valores en la referencia las señales de control se saturan en niveles máximos y mínimos relativos. Para solucionar esto es recomendable un estudio de la región a la que las salidas del sistema pueden acceder a través de las actuaciones posibles dentro de los límites de funcionamiento. Esta región quedará delimitada por las limitaciones en los actuadores y su efecto en las salidas.

En el régimen permanente se puede ver cómo se estabilizan las actuaciones en torno a unos valores concretos a partir de los cuales realizarán las oscilaciones oportunas producidas por la variación de TT5.

8 CONCLUSIONES Y PROPUESTAS DE MEJORA

Con todos los experimentos concluidos se va a realizar una reflexión sobre los resultados obtenidos a lo largo del trabajo, así como propuestas de mejora y continuación para el estudio de las dinámicas tratadas en el mismo.

8.1 Conclusiones

No son pocos los problemas que han surgido a lo largo de este Trabajo de Fin de Grado. La primera parte centrada en la implementación de un sistema de comunicación nuevo está provocada precisamente por la imposibilidad de trabajar con el hardware antes instalado. Tras varios intentos y pruebas de distinta índole se decidió dejar de arreglar los problemas ocasionados por el ordenador antiguo y la pantalla y diseñar un sistema que independice el funcionamiento de la planta de estos.

Este objetivo ha resultado un gran éxito. Por un lado, la inclusión del PLC en la red Ethernet del Laboratorio de y su accesibilidad desde cualquier ordenador conectado a éste permite un control de la planta mucho más cómodo y con la posibilidad de usar softwares más modernos y actuales. Incluso sería posible las lecturas de las variables desde varios puntos simultáneamente para su uso didáctico.

Se ha tratado de guiar a un futuro usuario de la planta por las metodologías más adecuadas para el trabajo en la planta con Matlab como los modelos de Simulink, los scripts con las funciones OPC UA o las funciones de cálculos de parámetros.

La segunda parte del trabajo quizá no se haya sido capaz de obtener resultados completamente satisfactorios. Por un lado, la obtención del modelo lineal resultó realmente difícil y en ocasiones incluso confusa. Pese a ello el algoritmo implementado ha conseguido mantener las salidas escogidas dentro de unos márgenes de error. Sin duda el resultado es mejorable pero no es un mal intento de control de la planta en esta configuración.

Con todo ello, se ha realizado un buen repaso a la metodología de identificación de sistemas dinámicos con resultados muy relevantes de cara a un futuro estudio de la configuración. De igual manera se ha descrito los pasos a seguir para el diseño de una estrategia de control en variables de espacio de estados cuyo diseño son válidos para cualquier otro modelo que se obtenga.

Al final, el objetivo de un Trabajo de Laboratorio como este no es más que la superación de los obstáculos encontrados a lo largo de la implementación de las distintas etapas de las que consta y su reflejo en la memoria para resultar de utilidad a otro posible usuario de la planta. Tras la realización del trabajo se han adquirido conocimientos acerca de la implantación de un sistema como el OPC UA, una metodología que como se ha explicado anteriormente, está ampliamente extendida en la industria. Desde el punto de vista de control, se han experimentado los problemas derivados de trabajar con un sistema físico en lugar de con un enunciado preparado para los cálculos. Esto ha requerido una revisión completa de los conceptos y una ampliación de estos para tratar de solventar los diversos desafíos que el control expuesto ha supuesto y con ello una interiorización de estos términos.

8.2 Propuestas de mejora

Puede ser de gran interés de cara a un estudio nuevo partir de los puntos en los que termina este trabajo para poder implementar metodologías avanzadas que resuelvan las complejas dinámicas que presenta la planta.

Con la planta conectada al entorno de Matlab y familiarizado con las etapas con las que cuenta la configuración tres se puede comenzar directamente a realizar una identificación con una metodología distinta a la que se presenta en este proyecto. También es interesante analizar las matrices de ponderación de las ganancias del controlador y del observador para su diseño particularizado para cada actuador y cada señal de

salida. Un estudio de coste de las actuaciones y de relaciones para cada entrada-salida. De esta manera también se podrá aportar al SSTO información a partir de los cálculos realizados en la parte de diseño de control para que éste sea capaz de priorizar ciertas acciones respecto a otras.

En cuanto a la planta multiprocesos, el sensor de nivel que en ocasiones anteriores ha dado problemas ha sido sustituido por uno nuevo que con una única calibración ha sido suficiente para todo el tiempo de trabajo con la planta. Aun así, si se vuelve a trabajar con él no estaría de más chequear que los datos que obtiene del nivel siguen vigentes.

La planta de frío ha dado algunos problemas cuando se ha requerido un enfriamiento de un cierto caudal durante un tiempo prolongado. Esto puede resultar un problema cuando los ensayos que tienen lugar en la planta pueden rondar los 70 minutos. También puede interesante que la histéresis en la que funciona la planta de frío tuviera un rango un poco más reducido. Esto aliviaría los principales problemas a la hora de identificar y controlar la planta en esta configuración 3.

Anexo A

Código del script ejecutable para la comunicación entre el modelo de Simulink y el PLC de la planta a través de la conexión OPC UA.

```
%% Script para inicio, preparación y comunicación con OPC UA
function [y]=comunicacion OPC_conf3 (VR1,VR2,VR3,R1,R2,WD)
clc
% variables
global init_server_conf2
global init_nodes_conf2
global uaClient
global input1
global input2
global input3
global input4
global input5
global input6
global output1
global output2
global output3
global output4
global output5
global output6
global output7
global output8
global output9
global output10

%Se comprueba la primera vez que no está inicializado
if (isempty(init_server_conf2))
    valor=zeros(1,10);
    init_server_conf2=0;
    init_nodes_conf2=0;
end

%Identificación del servidor y conexión del cliente (Simulink)
if init_server_conf2==0
    uaClient=opcua('opc.tcp://laptop-bdp4bsq0:49152/OPCUAServerExpert');
    setSecurityModel(uaClient,'None');
    connect(uaClient);
    init_server_conf2=1;
end

%Enlace de las variables creadas a las variables OPC
if uaClient.isConnected==1 && init_nodes_conf2==0
    init_nodes_conf2=1;
    input1= findNodeByName(uaClient.Namespace,'OPC01_r_VR1','-once');
    input2= findNodeByName(uaClient.Namespace,'OPC02_r_REF_VR2','-once');
    input3= findNodeByName(uaClient.Namespace,'OPC03_r_REF_VR3','-once');
    input4= findNodeByName(uaClient.Namespace,'OPC04_r_R1','-once');
    input5= findNodeByName(uaClient.Namespace,'OPC05_r_R2','-once');
    input6= findNodeByName(uaClient.Namespace,'OPC08_r_SINEWAVE_WD','-once');

    output1= findNodeByName(uaClient.Namespace,'OPC44_w_LT_puntos','-once');
    output2= findNodeByName(uaClient.Namespace,'OPC12_w_FT','-once');
    output3= findNodeByName(uaClient.Namespace,'OPC13_w_TT1','-once');
    output4= findNodeByName(uaClient.Namespace,'OPC14_w_TT2','-once');
    output5= findNodeByName(uaClient.Namespace,'OPC15_w_TT3','-once');
    output6= findNodeByName(uaClient.Namespace,'OPC16_w_TT4','-once');
    output7= findNodeByName(uaClient.Namespace,'OPC17_w_TT5','-once');
    output8= findNodeByName(uaClient.Namespace,'OPC18_w_TT6','-once');
    output9= findNodeByName(uaClient.Namespace,'OPC19_w_TT7','-once');
    output10= findNodeByName(uaClient.Namespace,'OPC30_w_SINE_WAVE_SENT','-once');
end
```

```
%Envío y recepción de información a través de las variables OPC

if uaClient.isConnected==1 && init_nodes_conf2==1

    [valor(1),~,~]=readValue(uaClient, output1);    %OPC44_w_LT_puntos
    [valor(2),~,~]=readValue(uaClient, output2);    %OPC12_w_FT
    [valor(3),~,~]=readValue(uaClient, output3);    %OPC13_w_TT1
    [valor(4),~,~]=readValue(uaClient, output4);    %OPC14_w_TT2
    [valor(5),~,~]=readValue(uaClient, output5);    %OPC15_w_TT3
    [valor(6),~,~]=readValue(uaClient, output6);    %OPC16_w_TT4
    [valor(7),~,~]=readValue(uaClient, output7);    %OPC17_w_TT5
    [valor(8),~,~]=readValue(uaClient, output8);    %OPC18_w_TT6
    [valor(9),~,~]=readValue(uaClient, output9);    %OPC19_w_TT7
    [valor(10),~,~]=readValue(uaClient, output10);  %WD

    writeValue(uaClient, input1, VR1);
    writeValue(uaClient, input2, VR2);
    writeValue(uaClient, input3, VR3);
    writeValue(uaClient, input4, R1);
    writeValue(uaClient, input5, R2);
    writeValue(uaClient, input6, WD);

end

[y]=double(valor);
% pause(0)
end
```

Anexo B

Código de la función en Matlab para poner el aplanta en modo remote desde el servidor OPC UA.

```

%% Script para poner la planta en modo remoto
function [y]=puesta_en_remoto(conf)
global init_server_rem
global init_nodes_rem
global uaClient
global input_rem_1
global input_rem_2
global input_rem_3
global input_rem_4
global input_rem_5

%Se comprueba la primera vez que no está inicializado
if (isempty(init_server_rem))
    valor=zeros(1,5);
    init_server_rem=0;
    init_nodes_rem=0;
end

%Identificación del servidor y conexión del cliente
if init_server_rem==0
    uaClient=opcua('opc.tcp://laptop-bdp4bsq0:49152/OPCUAServerExpert');
    setSecurityModel(uaClient,'None')
    connect(uaClient)
    init_server_rem=1;
end

%Enlace de las variables creadas a las variables OPC
if uaClient.isConnected==1 && init_nodes_rem==0
    init_nodes_rem=1;
    input_rem_1= findNodeByName(uaClient.Namespace,'OPC45_INTERRUPTOR','-once');
    input_rem_2= findNodeByName(uaClient.Namespace,'OPC46_CONF_CONFIRMADA','-once');
    input_rem_3= findNodeByName(uaClient.Namespace,'OPC47_ATAJO_PREPARACION','-once');
    input_rem_4= findNodeByName(uaClient.Namespace,'OPC48_CONF_SELECCIONADA','-once');
    input_rem_5= findNodeByName(uaClient.Namespace,'OPC49_REMOTO_OPC','-once');

%Envío de las activaciones a través de las variables
if uaClient.isConnected==1 &&init_nodes_rem==1

    writeValue(uaClient,input_rem_1,1)
    pause(0.5)
    writeValue(uaClient,input_rem_4,conf)
    pause(0.5)
    writeValue(uaClient,input_rem_2,1)
    pause(15)
    writeValue(uaClient,input_rem_3,1)
    pause(0.5)
    writeValue(uaClient,input_rem_5,1)

end

writeValue(uaClient,input_rem_2,0)
writeValue(uaClient,input_rem_3,0)
writeValue(uaClient,input_rem_5,0)
end

```

Anexo C

Código de la función en Matlab para apagar la planta desde el servidor OPC UA.

```

%Script para parar la planta y volver a inicio
function [y]=apagar_planta()

global init_server_rem_ap
global init_nodos_rem_ap
global uaClient
global input_rem_ap_1
global input_rem_ap_2

init_nodos_rem_ap=0;
%Se comprueba la primera vez que no está inicializado
if (isempty(init_server_rem_ap))
    valor=zeros(1,5);
    init_server_rem_ap=0;
    init_nodos_rem_ap=0;
end

%Identificación del servidor y conexión del cliente (Simulink)
if init_server_rem_ap==0
    uaClient=opcua('opc.tcp://laptop-bdp4bsq0:49152/OPCUAServerExpert');
    setSecurityModel(uaClient,'None')
    connect(uaClient)
    init_server_rem_ap=1;
end

%Enlace de las variables creadas a las variables OPC
if uaClient.isConnected==1 && init_nodos_rem_ap==0
    init_nodos_rem_ap=1;
    input_rem_ap_1= findNodeByName(uaClient.Namespace,'VJ40_r_FIN_CONTROL','-once');
    input_rem_ap_2= findNodeByName(uaClient.Namespace,'OPC45_INTERRUPTOR','-once');
end

%Envío de las activaciones a través de las variables
if uaClient.isConnected==1 && init_nodos_rem_ap==1

    writeValue(uaClient,input_rem_ap_1,1)
    writeValue(uaClient,input_rem_ap_2,0)
    pause(3)
end

writeValue(uaClient,input_rem_ap_1,0)
end

```

Anexo D

Código de la función encargada en el cálculo online de los parámetros y resultados de las referencias por el SSTO.

```
function [xr,ur]=SSTO(ref1, A, B, C, E, ref2, TT5_rela)
%Inicialización de las salidas, necesario para la compilación del modelo en
%Simulink
xr=zeros(9,1)';
ur=zeros(3,1)';

%Definición matrices y parámetros necesarios
R=1/20*eye(3);
Q=C'*1/6*diag([50,200])*C*100;
H=[Q zeros(9,3);zeros(3,9) R];
f=zeros(12,1);

lb=[-1000*ones(9,1);-20*ones(3,1)];
ub=[+1000*ones(9,1);+20*ones(3,1)];

%Se mete en beq las referencias deseadas y la perturbación TT5
beq=[-E*TT5_rela;ref1;ref2];
Aeq=[A-eye(9) B;C zeros(2,3)];

%Se definen la configuración del solver
options = optimoptions(@quadprog,'Algorithm','active-set','MaxIterations',1500);

%Resolución del problema de optimización
x = quadprog(H,f,zeros(12,12),zeros(12,1),Aeq,beq,lb,ub,zeros(12,1)',options);

%Escritura de los estados y señales de control de referencia obtenidos en
%el cálculo
xr=[x(1:1:9)];
ur=[x(10:1:12)];

xr=double(xr);
ur=double(ur);
end
```


REFERENCIAS

- [1] Alonso Alfaya, J E Desarrollo e implementación del sistema de control de la planta multiprocesos. *Trabajo de fin de grado, Universidad de Sevilla*, 2013.
- [2] J. L. P. A. José Enrique Alonso Alfaya, Planta Multiprocesos: Guía de Usuario, *Trabajo de fin de grado*, Universidad de Sevilla, 2015.
- [3] Gracia Villegas, V M, Laboratorio Virtual para el control de la Planta Multiprocesos basado en LabView *Trabajo de fin de grado*. 2015.
- [4] Rodríguez Ramírez D y Bordóns Alba C, ANALISIS Y CONTROL DE SISTEMAS EN ESPACIO DE ESTADO, IDENTIFICACION´ DE SISTEMAS, *apuntes de Ingeniería de control*, Universidad de Sevilla 2005
- [5] Varela Fernández A, MODELADO Y CONTROL EN EL ESPACIO ESTADOS. Universitat Politècnica de Valencia.. 2016.
- [6] Sirgo Blanco J A, Diseño de sistemas de control por computador, *apuntes de Ingeniería de sistemas*, Universidad de Oviedo, 2007
- [7] Castaño Giraldo A, Variables de estado- Espacio de estados, controlautomatico.com, 2018

