



industriales  
etsii

Escuela Técnica  
Superior  
de Ingeniería  
Industrial

# UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

## DESARROLLO DE UNA INTERFAZ GRÁFICA DE USUARIO PARA PLANTA PILOTO

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL

**Autor:** Julia Navarro Gallardo  
**Director:** José Luis Muñoz Lozano

Cartagena, febrero 2024



Universidad  
Politécnica  
de Cartagena

## ÍNDICE

1.	OBJETIVOS .....	4
2.	INTRODUCCIÓN .....	5
2.1.	Fundamentos de la herramienta AppDesigner .....	5
2.1.1.	Editor de diseño .....	6
2.1.2.	Editor de código .....	7
3.	PLANTA PILOTO .....	8
3.1.	Descripción de la planta piloto.....	8
3.2.	Líneas de agua .....	9
3.2.1.	Línea de agua caliente.....	9
3.2.2.	Línea de agua fría .....	9
3.3.	Equipo .....	9
3.3.1.	Depósitos.....	9
3.3.2.	Reactor .....	10
3.3.3.	Sensores y transmisores.....	11
3.3.4.	Actuadores .....	16
4.	INTERFAZ GRÁFICA: INFORMATIVA.....	24
4.1.	Introducción a la planta piloto .....	24
4.2.	Elementos de la planta piloto .....	27
4.2.2.	Ventana emergente .....	32
1.	ESTUDIO DE LAZOS DE CONTROL.....	41
4.3.	Lazo de nivel del depósito V-10102 .....	43
4.4.	Lazo de caudal en la línea de agua fría.....	44
4.5.	Lazo de temperatura a la salida de la línea de agua fría .....	45
4.6.	Lazo de temperatura a la salida de la línea de agua caliente .....	46
4.7.	Lazo de temperatura a la salida de la línea de agua fría: En cascada .....	47
4.8.	Lazo de temperatura del agua en la camisa del reactor. ....	49
5.	INTERFAZ GRÁFICA: LAZOS DE CONTROL.....	51
6.	MODELADO DE UN LAZO DE CONTROL.....	56
6.1.	Ecuaciones dinámicas del sistema en el dominio del tiempo .....	57
6.1.1.	Sensor-transmisor de temperatura.....	57
6.1.2.	Comparador .....	59
6.1.3.	Controlador .....	60

6.1.4.	Variador de frecuencia .....	60
6.1.5.	Bomba hidráulica.....	61
6.1.6.	Intercambiador de calor .....	62
6.1.7.	Modelo matemático del intercambiador de calor .....	63
6.2.	Linealización de las ecuaciones dinámicas del sistema en torno a un punto de funcionamiento .....	66
6.2.1.	Sensor-transmisor de temperatura.....	66
6.2.2.	Comparador .....	67
6.2.3.	Controlador .....	67
6.2.4.	Variador de frecuencia .....	68
6.2.5.	Bomba hidráulica.....	68
6.2.6.	Intercambiador de calor.....	68
6.3.	Ecuaciones dinámicas del sistema en el dominio de la frecuencia.....	69
6.3.1.	Sensor-transmisor de temperatura.....	69
6.3.2.	Comparador .....	69
6.3.3.	Controlador .....	70
6.3.4.	Variador de frecuencia.....	70
6.3.5.	Bomba hidráulica.....	71
6.3.6.	Intercambiador de calor.....	71
6.4.	Parámetros del sistema y punto de funcionamiento.....	72
6.5.	Construcción del diagrama de bloques en Simulink .....	79
7.	Análisis del sistema resultante.....	82
7.1.	Función de transferencia.....	82
8.	GUI EN APP DESIGNER DEL LAZO DE CONTROL .....	84
8.1.	Puesta en marcha.....	87
8.1.1.	Función valoresDefecto (app) .....	90
8.1.2.	Función puntoFuncionamiento (app).....	93
8.1.3.	Función estabilidadPlanta(app).....	93
8.1.4.	Función muestraValores (app) .....	94
8.1.5.	Función cargaDatosEnModelWorkspacePlanta (app).....	96
8.1.6.	Función cargaDatosEnWorkspace (app) .....	97
8.2.	Callback SwitchValueChanged (app, event) .....	98
8.2.1.	Función borraAnálisisPID (app) .....	100
8.2.2.	Función borraAnálisisPlanta (app) .....	100
8.2.3.	Función borraAnálisis (app).....	101
8.2.4.	Función gestionaGraficaPlanta (app, 0) .....	101

8.2.5.	Función gestionaGraficaPID(app,0).....	102
8.2.6.	Función gestionaGraficasPlantayPID(app) .....	103
8.2.7.	Función calculaGrafica(app, tiempoLimiteMax) .....	103
8.2.8.	Función analizaEstablecimientoDatos (app, tiempos, datos, ps) = [is, ts, ys, yrp] 107	
8.2.9.	Función AnalizaRespuestaDatos (app) .....	108
8.2.10.	Función analizaPicoDatos (tiempo, datos) .....	110
8.2.11.	Función analizaRespuestaPlanta (app).....	111
8.2.12.	Función analizaRespuestaPID (app) .....	112
8.2.13.	DibujaSoloGraficaPlanta (app) .....	112
8.2.14.	Función dibujaSoloGraficaPID (app).....	114
8.2.15.	Función dibujaGraficasPlantayPID .....	114
8.3.	Resultados y observaciones .....	115
9.	Conclusiones.....	121
10.	Trabajos futuros .....	122
11.	Referencias.....	123
12.	Bibliografía .....	124

*Quiero expresar mi profundo agradecimiento a todas las personas que han contribuido de manera significativa a la realización de este Trabajo de Fin de Grado.*

*En primer lugar, quiero agradecer a mi tutor, José Luis Muñoz Lozano, por su orientación y su apoyo constante a lo largo de todo este proceso de formación. Su dedicación, paciencia y la notable rapidez con la que atendía a mis dudas han sido fundamentales para el desarrollo y la conclusión exitosa de este proyecto.*

*Asimismo, deseo expresar mi agradecimiento a los profesores que diseñaron la planta piloto en la que se fundamenta este trabajo, Julio José Ibarrola Lacalle y José Manuel Cano Izquierdo. Ambos compartieron de forma generosa su conocimiento y experiencia, guiándome durante la visita a la planta piloto y respondiendo a todas mis dudas sobre el estado actual de la instalación.*

*Este trabajo no habría sido posible sin la valiosa contribución de cada una de estas personas, y estoy sinceramente agradecida por el apoyo constante que me han brindado a lo largo de este proyecto.*

*Finalmente, quiero expresar mi gratitud a mi familia y amigos por su constante aliento y comprensión a lo largo de este desafiante proceso académico.*

## 1. OBJETIVOS

El objetivo principal de este proyecto es programar una interfaz gráfica de usuario (GUI) didáctica mediante la aplicación *App Designer* de Matlab con los siguientes apartados:

1. Implementación de una guía informativa de la planta piloto situada en la Universidad Politécnica de Cartagena (UPCT).

Esta interfaz trata de proporcionar al usuario información sobre la planta piloto, que engloba una descripción de la propia planta y los distintos elementos comprendidos en ella.

La información es proporcionada mediante ventanas emergentes resultado de interactuar con una serie de componentes en la aplicación.

Así, mediante la distribución de botones (puntos de información) localizados sobre una representación del esquema de la planta, se puede acceder a una descripción de los elementos que la componen junto con fotografías y enlaces a especificaciones adicionales del fabricante.

Este primer apartado permite al usuario una mejor comprensión tanto del funcionamiento de la planta como de los elementos comprendidos en ella. Esta interfaz ayuda a la familiarización con ciertos componentes cruciales en la planta.

2. El estudio de posibles lazos de control que pueden llegar a implantarse en la planta mencionada.

Mediante un desplegable, el usuario podrá seleccionar un lazo del conjunto ofrecido y adquirir información relacionada con cada lazo que se ofrece.

Esta información consiste en una explicación detallada sobre el funcionamiento del lazo de control y un resumen de las señales y elementos más importantes. También se dispone de un esquema representativo de cada lazo concreto, que mediante una serie de botones el usuario podrá interactuar y recibir información específica de cada elemento.

Con esta parte de la interfaz, se consigue que el usuario comprenda los distintos lazos de control disponibles para la planta.

3. Un análisis de un posible lazo de control de la planta mediante un modelo sencillo.

Este análisis consiste en una representación gráfica de la variable controlada en función del tiempo, así como en la distribución de valores obtenidos en la simulación, entre los que se encuentran variables en el punto de funcionamiento y otros parámetros característicos de los elementos comprendidos en el lazo. También se hará un estudio del sistema indicando el tipo y parámetros asociados entre los que se encuentran la situación de polos, el valor resultante del factor de amortiguación, etc.

Para ello, se realiza un modelado simple mediante *Simulink*, manejado mediante *App Designer*, que incluye también un controlador PID para experimentar con el control del lazo.

Las acciones proporcional, integral y derivativa de este controlador, junto con los valores del punto de funcionamiento y otros parámetros asociados a los elementos que componen el lazo, pueden ser modificados por medio de la interfaz.

Así, se pretende que el usuario pueda escoger tanto un punto de funcionamiento como un controlador concreto, permitiéndole ensayar distintas simulaciones.

El diseño del control viene dado por una simulación de la planta en la que se pueden probar distintas configuraciones mediante ensayo y error. De este modo, el usuario puede obtener un mejor entendimiento de las consecuencias de cada modificación mediante la representación gráfica y los resultados obtenidos en las distintas variables y parámetros involucrados.

## 2. INTRODUCCIÓN

### 2.1. Fundamentos de la herramienta AppDesigner

En el ámbito del control y la automatización es esencial el desarrollo de interfaces de usuario intuitivas y eficientes. MATLAB es una buena herramienta para el cálculo numérico y la simulación y ofrece un entorno de desarrollo gráfico, App Designer, para la creación de aplicaciones con interfaces de usuario interactivas y personalizadas.

App Designer es una interfaz gráfica en MATLAB que facilita la creación de aplicaciones con interfaces de usuario personalizadas. Permite el diseño interactivo, la programación de código MATLAB junto con funciones adicionales que pueden interactuar con los modelos Simulink, todo ello mediante la visualización en tiempo real de sistemas de control.

En la estructura de la aplicación se integran componentes gráficos, programación en MATLAB y gestión de eventos a través de callbacks.

## Desarrollo de una interfaz gráfica de usuario para planta piloto



Figura 2.1-Ventana 'App Designer. Design View'

A continuación, se detalla la estructura esencial de una aplicación en App Designer.

### 2.1.1. Editor de diseño

El editor de diseño consta del panel de diseño y la librería de componentes.

El panel de diseño es la zona principal donde se organiza y coloca visualmente cada componente de la interfaz de usuario. Los componentes se encuentran en la librería, desde la cual se arrastran y se sueltan en esta área.

Aquí se define la disposición visual de la interfaz, incluyendo la ubicación y el tamaño de cada elemento.

La librería de componentes comprende gráficos, botones, controles deslizantes, cajas de texto, entre otros.



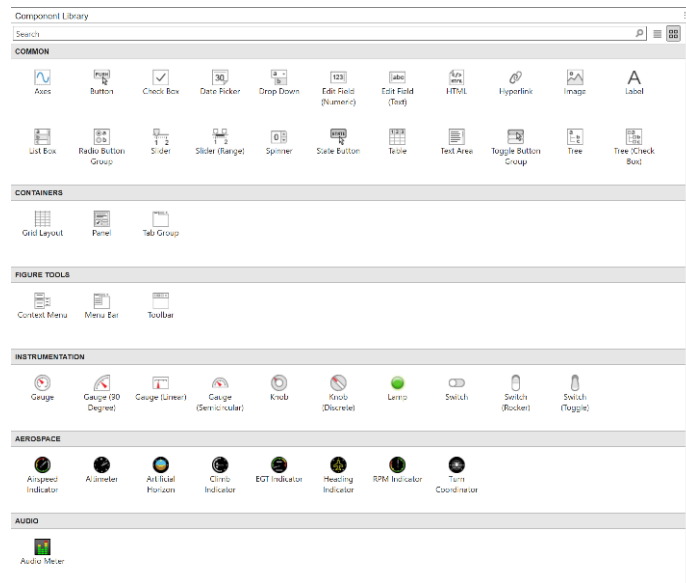


Figura 2.2: Librería 'App Designer. Design View'

### 2.1.2. Editor de código

La pestaña de editor de código permite escribir y editar código MATLAB. Aquí es donde se implementan los callbacks y otras funciones personalizadas que definen el comportamiento de la aplicación.

- Propiedades

Las propiedades son variables globales que almacenan datos o configuraciones específicas de la aplicación. De modo que cada componente de la interfaz tiene propiedades asociadas que definen su apariencia y comportamiento.

Estas propiedades son necesarias para mantener el estado de la aplicación, almacenar datos y ajustar la configuración de los componentes de la interfaz.

- Callbacks

Los callbacks son funciones que se ejecutan en respuesta a eventos específicos, como pulsar un botón, mover un deslizador o cambiar el contenido en un cuadro de texto del panel de diseño.

Se utilizan callbacks para provocar una reacción por parte del programa como respuesta a la interacción del usuario.

- Funciones

Las funciones adicionales son bloques de código que se pueden definir para realizar tareas específicas, como pueden ser realizar cálculos o activar unas condiciones deseadas para unas situaciones concretas.

Pueden ser llamadas desde callbacks o desde otras funciones.

### 3. PLANTA PILOTO

#### 3.1. Descripción de la planta piloto

La planta piloto alberga un reactor protegido por una camisa, a la cual se suministra agua a través de dos líneas: una de agua caliente y otra de agua fría.

El objetivo en esta instalación es regular la temperatura del líquido que se encuentra dentro del reactor.

Ambas líneas de agua tienen un impacto directo en la temperatura de la camisa, afectando, a su vez, la temperatura del líquido presente en el reactor.



Figura 3.1: Planta piloto

## 3.2. Líneas de agua

### 3.2.1. Línea de agua caliente

El proceso de calentamiento de esta línea consta de dos etapas:

- Primera etapa: El agua de red a temperatura ambiente es suministrada a un depósito, por medio de una válvula. Este depósito contiene una resistencia que calienta el agua almacenada hasta una temperatura intermedia.
- Segunda etapa: El agua a la temperatura intermedia, es impulsada por acción de una bomba hacia la camisa del reactor. A su paso por este tramo de línea, el agua se calienta hasta una temperatura final mediante un calentador en línea.

### 3.2.2. Línea de agua fría

El proceso de enfriamiento de esta línea consta de dos etapas:

- Primera etapa: El agua de red a temperatura ambiente es suministrada por medio de una válvula a un depósito, donde se almacena.
- Segunda etapa: Al salir de dicho depósito, el agua es bombeada hacia la camisa del reactor. A su paso por este tramo de línea, el agua se enfría mediante un intercambiador de calor por el que circula un líquido refrigerante.

## 3.3. Equipo

### 3.3.1. Depósitos

La planta cuenta con el depósito V-10101 en la línea de agua caliente, y el depósito V-10102 en la línea de agua fría.

Ambos depósitos almacenan agua a temperatura ambiente proveniente de la red, para después ser bombeada a través de la línea correspondiente. Además, cuentan con un sensor que registra la altura de llenado para poder controlarla. Por esta razón, también cuenta con un sensor de temperatura, para monitorizar y controlar la temperatura que alcanza el agua durante su almacenamiento en el tanque.



Figura 3.2: Depósitos de las líneas fría y caliente

### 3.3.2. Reactor

El reactor R-10105 se trata de un reactor Batch, por lotes, el cual cuenta con un agitador, A-10105.

En realidad, la planta no está diseñada para que ocurra ninguna reacción, de modo que se ha incorporado una resistencia, J-10106, que simula el calor generado por una reacción exotérmica.

El reactor dispone de un sensor de presión, y un sensor de temperatura, para medir el nivel de llenado y la temperatura, respectivamente.



Figura 3.3: Reactor

### 3.3.3. Sensores y transmisores

En el ámbito de regulación automática y control, los sensores son dispositivos que permiten medir variables físicas del sistema y del entorno para proporcionar información en tiempo real. Esta información se utiliza para ajustar y controlar sistemas automáticamente.

A continuación, se describen los sensores utilizados en la planta, así como sus especificaciones.

#### 3.3.3.1. Sensores de Temperatura

En la planta piloto se miden las temperaturas en las líneas de agua fría y caliente, justo antes de su mezclado, así como la temperatura de la mezcla a la entrada de la camisa. También se mide la temperatura del agua en el depósito correspondiente a la línea de agua caliente, y la temperatura de la línea de líquido refrigerante, utilizado para enfriar el agua perteneciente a la línea de agua fría.



Figura 3.4: Sensor de temperatura en el reactor

Estas mediciones se realizan mediante sensores de temperatura Pt100. Estos sensores son del tipo RTD (Detector de temperatura resistivo) y son dispositivos que utilizan la variación de la resistencia eléctrica con la temperatura para medir la temperatura de un entorno.

Están compuesto de platino, de manera que a 0 °C su resistencia eléctrica es de 100 ohmios.

Los RTD pueden medir temperaturas en un rango bastante amplio, concretamente desde -200°C hasta unos 600°C.

La resistencia eléctrica del RTD varía de manera predecible con la temperatura. Esta variación es casi lineal, sobre todo cuando se encuentra en un rango de valores positivos, lo que facilita la conversión de la señal en una lectura de temperatura precisa.

Esta variación viene definida por la siguiente expresión:

$$R_{Pt100} = 100 \cdot (1 + 0.00385 \cdot T)$$

Donde:

- T es la temperatura medida por el sensor, en °C
- $R_{Pt100}$  es el valor de la resistencia a dicha temperatura, en  $\Omega$ .

#### 3.3.3.1.1. Transmisores

Además, la planta cuenta con transmisores Rosemount 248R, que se encargan de transmitir la información sobre la temperatura medida.



**Figura 3.5: Transmisores asociados a los sensores de temperatura**

Este transmisor, que sigue el protocolo HART (Highway Addressable Remote Transducer), convierte la señal resistiva correspondiente a la temperatura medida en una señal eléctrica.

La salida de corriente de estos dispositivos comprende desde los 4 mA hasta 20 mA, con una relación lineal entre la señal transmitida y la intensidad de corriente obtenida.

Un beneficio es que la utilización de 4 mA como referencia de "cero" mejora la relación señal/ruido en niveles bajos. Además, un circuito que no funciona correctamente debido a una conexión abierta provoca una intensidad de corriente fuera del rango válido. Esto facilita la detección de ese funcionamiento incorrecto.

#### 3.3.3.2. *Sensor de nivel*

El nivel de agua almacenada en el depósito correspondiente a la línea de agua fría se mide mediante un sensor de nivel por radar de onda guiada. Este sensor emite pulsos de ondas electromagnéticas de radar hacia el medio cuyo nivel se está midiendo, de modo que los pulsos de radar viajan hasta la superficie del medio y son reflejados de vuelta hacia el sensor por dicha superficie.

La medición se basa en el tiempo que tarda la onda en viajar desde el transmisor hasta la superficie y regresar al receptor. Este tiempo se utiliza para calcular la distancia desde el sensor hasta la superficie del material.

Este sensor es el Rosemount 2051T, que tiene incorporado un transductor para transmitir la señal correspondiente. Al igual que los transmisores encargados de la señal de temperatura, sigue el protocolo HART.



Figura 3.6: Sensor de nivel del depósito V-10102

### 3.3.3.3. Sensor de presión

El nivel de agua almacenada en el depósito correspondiente a la línea de agua caliente se mide mediante un sensor de presión diferencial.

Este tipo de sensor consta de dos cámaras separadas por una membrana o diafragma. Una cámara está expuesta al fluido cuyo nivel se va a medir (en este caso, el agua en el depósito), mientras que la otra está en contacto con la presión atmosférica.

La cámara de referencia está abierta a la atmósfera, por lo que la presión en ella es constante y se utiliza como referencia. La presión atmosférica actúa como un punto fijo, mientras que la presión en la otra cámara varía con el nivel del agua.

La cámara en contacto con el agua en el depósito experimenta la presión hidrostática, que es proporcional a la profundidad del agua en el depósito. A medida que el nivel del agua cambia, la presión en esta cámara también cambia.

La salida del sensor es la diferencia de presión entre las dos cámaras. Cuando el nivel del agua sube o baja, la diferencia de presión varía en consecuencia.

Esta señal de salida se puede convertir en una medida del nivel de agua en el depósito mediante un proceso de calibración. Este proceso establece una relación entre la diferencia de presión y la altura del agua, dado por la siguiente ecuación:

$$\Delta P = P_A - P_B = (\rho \cdot g \cdot \Delta h + P_{atm}) - P_{atm} \rightarrow \Delta P = \rho \cdot g \cdot \Delta h$$

donde:  $P_A$  es la presión hidrostática y  $P_B$ , la presión atmosférica

Así como el sensor de nivel anterior, éste también es el Rosemount 2051T, que tiene incorporado un transductor para transmitir la señal correspondiente.



Figura 3.7: Sensor de presión del reactor



#### 3.3.3.4. *Sensor de caudal*

El caudal másico de las líneas de agua fría y caliente se mide mediante un caudalímetro o sensor de Coriolis.

En el centro del sensor hay un tubo por el que circula el fluido. Este tubo se hace vibrar a una frecuencia específica utilizando una bomba.

Cuando no hay caudal, el fluido está en reposo y no hay fuerzas de Coriolis. Cuando el fluido circula a través del tubo de medición en movimiento surgen las fuerzas de Coriolis. El movimiento generado por la oscilación del tubo de medición se suma al movimiento lineal del fluido. En este momento, la fuerza de Coriolis provoca una torsión en los tubos, y los sensores ubicados en la entrada y la salida registran una diferencia de tiempos en este movimiento, expresada como una diferencia de fase. A medida que aumenta el caudal másico, la diferencia de fase también se incrementa, siendo proporcional al flujo de fluido.

En este sensor, viene incorporado un transmisor HART, capaz de transmitir una señal de intensidad proporcional al caudal másico, con un rango de 4-20 mA.

La principal ventaja de los sensores de Coriolis en comparación con otros tipos de caudalímetros es que miden directamente el caudal másico del fluido en lugar del volumétrico, por lo que son inmunes a variaciones en la densidad.



Figura 3.8: Caudalímetro en la línea de agua fría

### 3.3.4. Actuadores

Un actuador es un dispositivo que se utiliza para generar una acción en respuesta a una señal de control.

Los actuadores son importantes en sistemas de control para traducir las señales emitidas por el controlador en cambios en el proceso que está siendo controlado. Las señales de control (que generalmente son señales eléctricas, neumáticas o hidráulicas), las transforma en movimientos mecánicos, cambios en la posición, o ajustes físicos en el sistema.

#### 3.3.4.1. Válvulas

En la planta se encuentran las siguientes válvulas:

- EV-01: Esta válvula permite el paso de agua a temperatura ambiente desde la red en la línea caliente.
- EV-02: Esta válvula permite el paso de agua a temperatura ambiente desde la red en la línea fría.
- EV-03: Esta válvula permite el paso de agua calentada hacia la camisa del reactor.
- EV-04: Esta válvula permite el paso de agua enfriada hacia la camisa del reactor.



Figura 3.9: Válvulas de la planta piloto

Se trata de válvulas solenoides o electroválvulas “todo-nada”. Se encargan de abrir o cerrar el paso de agua cuando se aplica o retira una corriente eléctrica proveniente del controlador.

Operan en dos estados extremos: completamente abierta, para permitir el paso de flujo o completamente cerrada, para impedirlo.

Están compuestas por:

1. Bobina: La válvula solenoide tiene una bobina de alambre enrollado alrededor de un núcleo. Cuando se aplica una corriente eléctrica a la bobina, crea un campo magnético alrededor del núcleo.
2. Núcleo magnético: El núcleo magnético es atraído hacia la bobina cuando se activa la corriente, lo que puede cambiar la posición de la válvula.
3. Válvula: La válvula en sí misma es una compuerta que controla el flujo del fluido, Cuando la bobina se encuentra desactivada, la válvula se encuentra en una posición cerrada, bloqueando el paso de fluido. Cuando se activa la bobina, el núcleo magnético se mueve y abre la válvula, permitiendo que el fluido fluya a través de la válvula.

#### 3.3.4.2. Bombas

En la instalación se encuentran tres bombas trifásicas cuya capacidad máxima de trabajo son 15 L/min:

- N-10050: Bombea líquido refrigerante desde el intercambiador de calor E-10110 hacia el equipo de frío, en la línea de fluido refrigerante.
- N-10051: Bombea agua desde el depósito V-10101, hacia el calentador en línea G-10108, en la línea de agua caliente.
- N-10052: Bombea agua desde el depósito V-10102, hacia el intercambiador de calor E-10110, en la línea de agua fría.



Figura 3.10: Bombas hidráulicas de la planta piloto

Debido al diseño de tuberías y caudalímetros, el caudal de agua está limitado a alcanzar un máximo de 1.8 L/min, así como una temperatura máxima de 80°C.

Estas bombas tienen la capacidad de ajustar la cantidad de fluido impulsado mediante la modificación de la frecuencia de los variadores de frecuencia, posibilitando así, un control preciso sobre el caudal y, por ende, sobre la temperatura de la camisa del reactor.

En particular, la bomba N-10052 se trata del modelo Prisma 15-3. A una frecuencia de 50 Hz gira a 2900 rpm, y el motor es asíncrono de dos polos, según las especificaciones del fabricante [PRIS1].

Además, en [JMG] se especifican los valores de 50 Hz y 2900 rpm para la bomba de la planta.

#### 3.3.4.2.1. Bote de expansión

La planta cuenta con un bote de expansión, cuya función es almacenar agua procedente de la línea antes de la entrada a la bomba. De esta manera, la bomba siempre se encuentra cebada de agua y se evitan fallos en su funcionamiento.



Figura 3.11: Bote de expansión

#### 3.3.4.3. Variadores de frecuencia

La planta dispone de un armario donde se encuentran los variadores de frecuencia.

Estos tienen como objetivo controlar la velocidad de los motores eléctricos asociados a las bombas de la planta, alterando la frecuencia y potencia suministrados al motor. Al modificar la velocidad del motor eléctrico, varía también el caudal suministrado por la bomba a la que va asociado el motor. Como consecuencia de esto, la temperatura del flujo bombeado también se verá alterado.

El variador toma energía de la red eléctrica, que generalmente es de corriente alterna (CA) a una frecuencia y voltaje constante. La corriente alterna de entrada se rectifica para obtener corriente continua (CC).

La corriente continua se convierte nuevamente en corriente alterna, pero con una frecuencia variable. Este proceso se realiza mediante dispositivos de conmutación electrónica.

El controlador del variador ajusta la frecuencia y potencia de salida para modificar la velocidad del motor. Así, se logra un control preciso y eficiente.

Por otro lado, se conoce que el modelo de variador de frecuencia utilizado en la planta es ATV 11HU12M2E Altivar 11.

En la hoja de especificaciones del fabricante, la frecuencia de salida varía entre 0 y 200 Hz, y la intensidad de corriente de entrada analógica podría ir de 0 a 20 mA, con una resistencia de 250 ohmios o de 4 a 20 mA, sin dicha resistencia. [ATV 11] [MCB]



Figura 3.12: Armario de variadores de frecuencia

#### 3.3.4.4. Resistencias térmicas

En la instalación se pueden observar tres resistencias térmicas:

- J-10110: Se encuentra en el interior del depósito V-10101 en la línea caliente. Esta resistencia calienta el agua contenida en el depósito a una temperatura intermedia, para después ser bombeada.
- J-10106: Esta resistencia se encuentra en el interior del reactor Batch, R-10105, y su función es simular una reacción exotérmica, cediendo calor al sistema.



Figura 3.13: Resistencias del depósito J-10106 y J-10110

- G-10108: Se trata de un calentador en línea situado entre la bomba N-10051 y la válvula que permite el acceso a la camisa del reactor, en la línea caliente, EV-03. Este calentador contiene una resistencia muy potente que es capaz de calentar el agua proveniente del depósito V-10101, de la temperatura intermedia a una temperatura final elevada. Debido al diseño de tuberías de la planta, el calentador se encuentra limitado a calentar el agua a una temperatura máxima de 80°C, tal y como se menciona en puntos anteriores.



Figura 3.14: Calentador en línea

#### 3.3.4.5. Tiristores

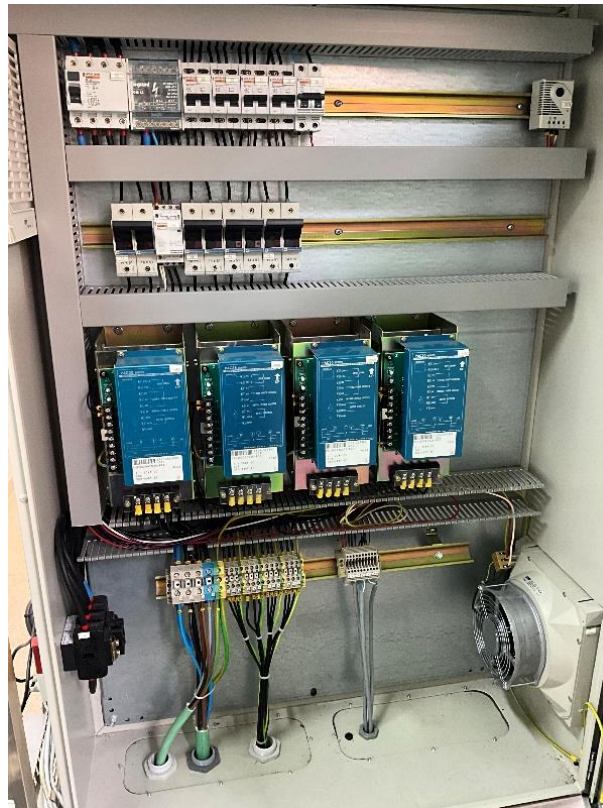
La planta cuenta con un armario donde se encuentra una serie de tiristores que permiten controlar la potencia suministrada a las resistencias térmicas de la planta. La corriente eléctrica que fluye a través de las resistencias genera calor, y el control de los tiristores permite ajustar la cantidad de calor producido.

Los tiristores, específicamente los SCR (Rectificadores Controlados de Silicio) son dispositivos de conmutación unidireccionales. Una vez que se activan mediante un pulso de disparo, conducen corriente en una sola dirección hasta que se corte.

El control de los tiristores se realiza mediante la modulación de la duración de los pulsos de disparo, lo que a su vez modifica el ángulo de conducción. Cuanto mayor sea el ángulo de conducción, mayor será la potencia suministrada a la carga.

Al ajustar la potencia suministrada a las resistencias térmicas, se puede controlar la temperatura en un sistema de calefacción.

La capacidad de modular la potencia eléctrica permite un control más preciso y eficiente del calentamiento. En comparación con métodos tradicionales de encendido y apagado, los tiristores permiten un control más suave, evitando picos de corriente y mejorando la eficiencia energética.



**Figura 3.15: Armario de tiristores**

#### *3.3.4.6. Equipo de frío*

El equipo de frío E-10107, está ubicado en la línea refrigerante, y se encarga de reducir la temperatura del fluido refrigerante. La temperatura que alcanza es crucial para el posterior enfriamiento del agua en la línea fría.

Como la temperatura del refrigerante en este proceso puede reducirse por debajo de 0°C, se opta por una mezcla de agua y glicol, un anticongelante comúnmente utilizado en sistemas de refrigeración.

El fluido refrigerante ya enfriado, circulará por el intercambiador de calor E-10110 con el fin de enfriar el agua en la línea fría.





Figura 3.16: Enfriador E-10107

#### 3.3.4.7. Intercambiador de calor

El intercambiador de calor E-10110 se ubica entre la bomba N-10052 y la válvula responsable del paso de agua a la camisa en la línea fría, EV-04.

Este intercambiador, de tipo carcasa-tubos, se encarga de enfriar el agua de la línea fría mediante el uso de la mezcla de agua con glicol, que constituye el refrigerante de la planta. No se sabe el modelo exacto, pero se conoce el fabricante, John Crane [HXR].



Figura 3.17: Intercambiador de calor E-10110

## 4. INTERFAZ GRÁFICA: INFORMATIVA

### 4.1. Introducción a la planta piloto

Esta introducción consiste en una pantalla informativa en la que se muestran los elementos principales de los que consta, a la que se adjunta una fotografía.



Figura 4.1: Pestaña 'GUI Presentación'

Ninguno de los componentes que conforma esta primera pestaña es interactivo, y muestran un único contenido, de manera fija.

Consta de los siguientes componentes:

- Una etiqueta que muestra el título de la pestaña, 'INTRODUCCIÓN A LA PLANTA PILOTO', para lo que modificamos su propiedad, 'Text'. Adicionalmente se cambia el tipo de letra y su estilo.

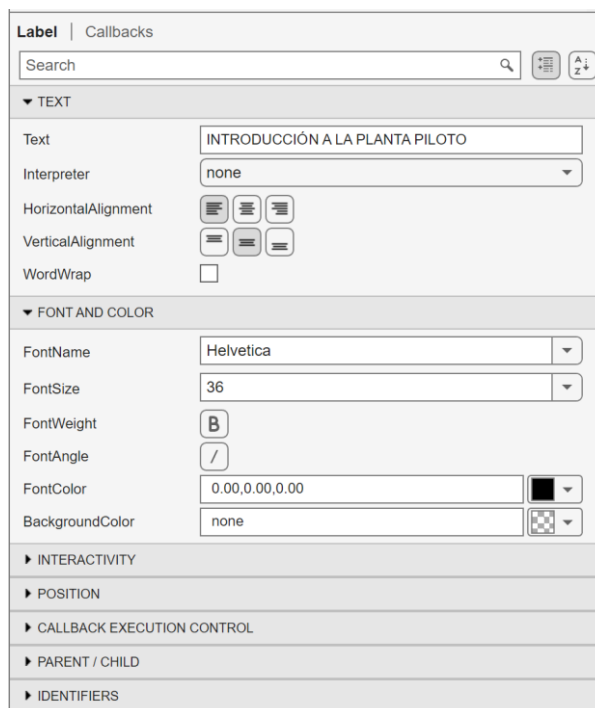


Figura 4.2: Ventana 'Etiqueta Presentación'

- Un texto introducido mediante un “enlace html”, con una descripción general de la planta.
- Primero, se debe crear el archivo html y colocarlo en el panel donde encontramos el programa.
- Se debe utilizar el lenguaje de html para dar estructura al texto y se especifica el formato de codificación de caracteres Unicode, UTF-8.
- Adicionalmente, se debe crear un fichero de estilo, css, que complementa al texto, añadiéndole estructura mediante un estilo determinado.
- Por último, se añade la ruta para encontrar el archivo html en la zona de edición del programa.

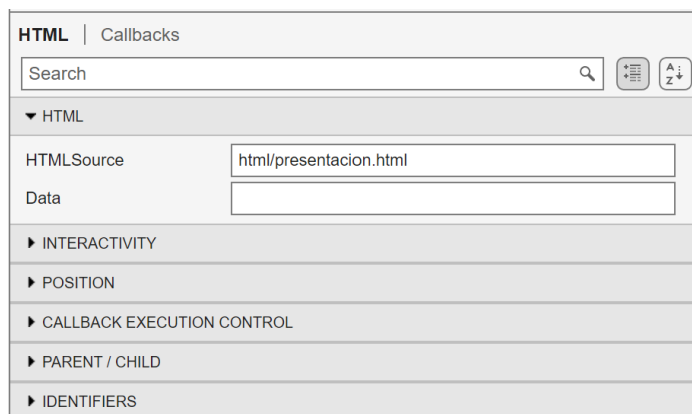
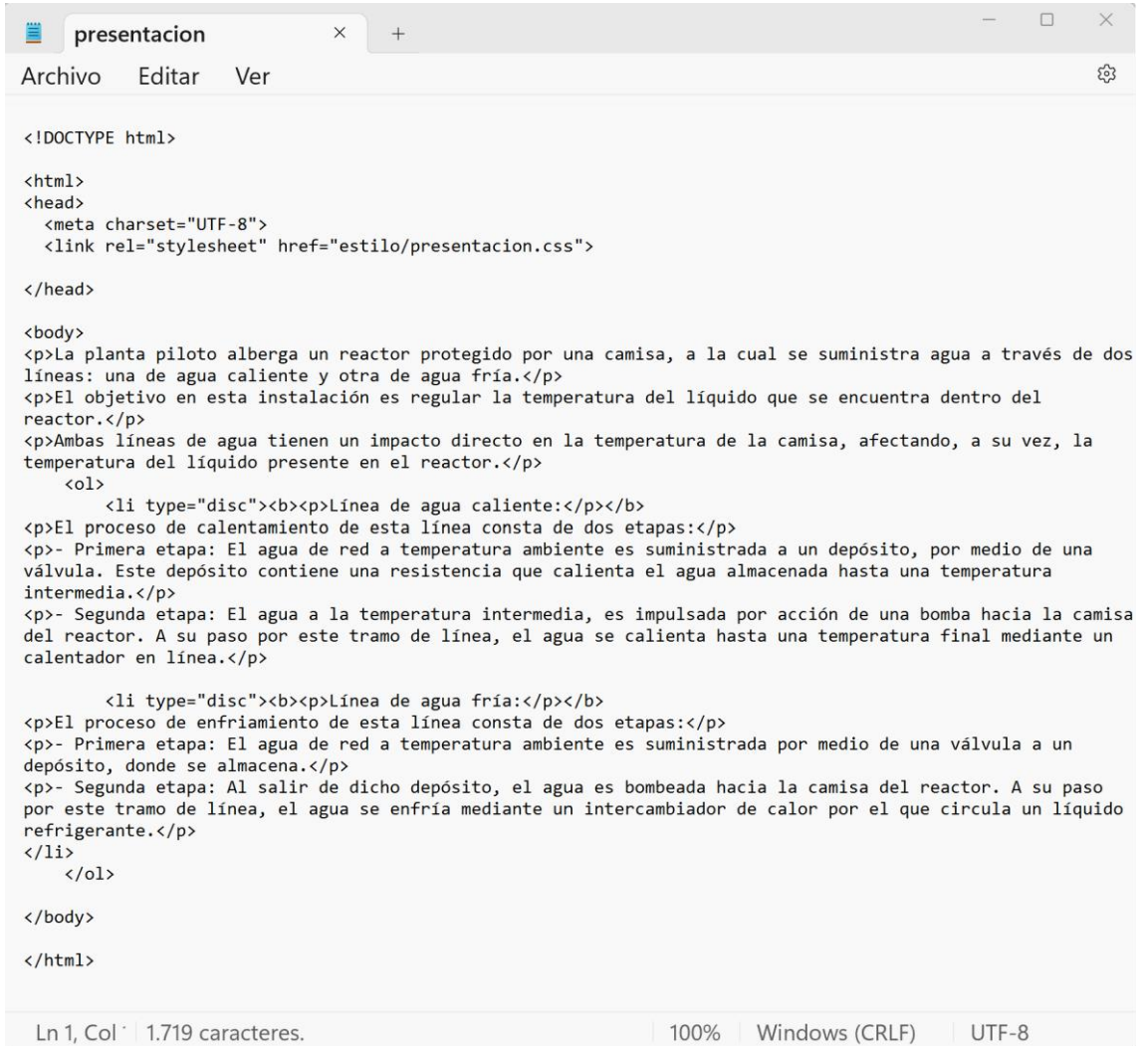


Figura 4.3: 'HTML Presentación'

## Desarrollo de una interfaz gráfica de usuario para planta piloto



```
<!DOCTYPE html>

<html>
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="estilo/presentacion.css">
</head>

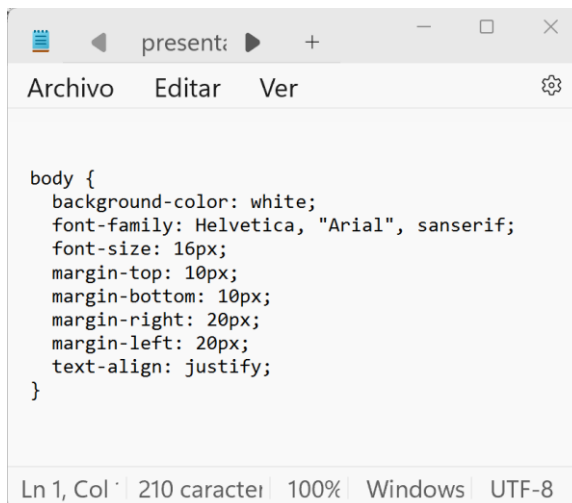
<body>
<p>La planta piloto alberga un reactor protegido por una camisa, a la cual se suministra agua a través de dos líneas: una de agua caliente y otra de agua fría.</p>
<p>El objetivo en esta instalación es regular la temperatura del líquido que se encuentra dentro del reactor.</p>
<p>Ambas líneas de agua tienen un impacto directo en la temperatura de la camisa, afectando, a su vez, la temperatura del líquido presente en el reactor.</p>
  <ol>
    <li type="disc"><b><p>Línea de agua caliente:</p></b>
<p>El proceso de calentamiento de esta línea consta de dos etapas:</p>
<p>- Primera etapa: El agua de red a temperatura ambiente es suministrada a un depósito, por medio de una válvula. Este depósito contiene una resistencia que calienta el agua almacenada hasta una temperatura intermedia.</p>
<p>- Segunda etapa: El agua a la temperatura intermedia, es impulsada por acción de una bomba hacia la camisa del reactor. A su paso por este tramo de línea, el agua se calienta hasta una temperatura final mediante un calentador en línea.</p>
    <li type="disc"><b><p>Línea de agua fría:</p></b>
<p>El proceso de enfriamiento de esta línea consta de dos etapas:</p>
<p>- Primera etapa: El agua de red a temperatura ambiente es suministrada por medio de una válvula a un depósito, donde se almacena.</p>
<p>- Segunda etapa: Al salir de dicho depósito, el agua es bombeada hacia la camisa del reactor. A su paso por este tramo de línea, el agua se enfría mediante un intercambiador de calor por el que circula un líquido refrigerante.</p>
  </li>
  </ol>

</body>

</html>
```

Ln 1, Col 1 | 1.719 caracteres. | 100% | Windows (CRLF) | UTF-8

Figura 4.4: Documento 'html Presentación'



```
body {
  background-color: white;
  font-family: Helvetica, "Arial", sanserif;
  font-size: 16px;
  margin-top: 10px;
  margin-bottom: 10px;
  margin-right: 20px;
  margin-left: 20px;
  text-align: justify;
}
```

Ln 1, Col 1 | 210 caracter | 100% | Windows | UTF-8

Figura 4.5: Documento de estilo 'css Presentación'

- Una fotografía de la planta, en la que se pueden apreciar los elementos que la conforman.

Para ello añadimos la ruta donde se encuentra la fotografía, en la zona de edición del programa.

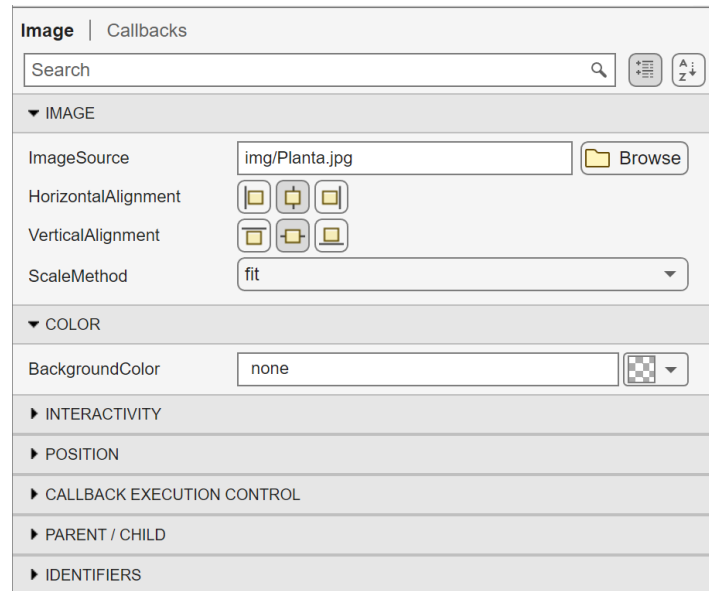


Figura 4.6: 'Image Presentación'

## 4.2. Elementos de la planta piloto

La pestaña 'Esquema' muestra una representación simplificada de la planta, y los elementos comprendidos en ésta.

Se añaden botones junto a cada elemento, de forma que, al pulsarlos, aparece una ventana emergente con información acerca del elemento en cuestión.

Esta información consiste en una fotografía en formato 'jpg', una etiqueta a modo de título, una tabla resumen con especificaciones sobre el elemento guardado en un archivo 'csv', y un resumen de su funcionamiento en la planta guardado en un archivo 'html'. Además, en este último archivo, se añaden 'links' que muestran para cada elemento, información adicional, como el lugar que ocupa ese elemento en la planta, y la página web del fabricante con más información acerca de sus especificaciones.

Finalmente, se ha añadido al esquema un menú contextual, que se activa con el botón derecho del ratón en una parte vacía del esquema. Este menú puede invisibilizar los botones asociados a los elementos o "puntos de información" o revertir su visualización. Además, puede hacer la misma función que la pulsación de los botones. Su código está

muy disperso entre diferentes callbacks y no se va a discutir aquí (se encuentra en el código fuente).

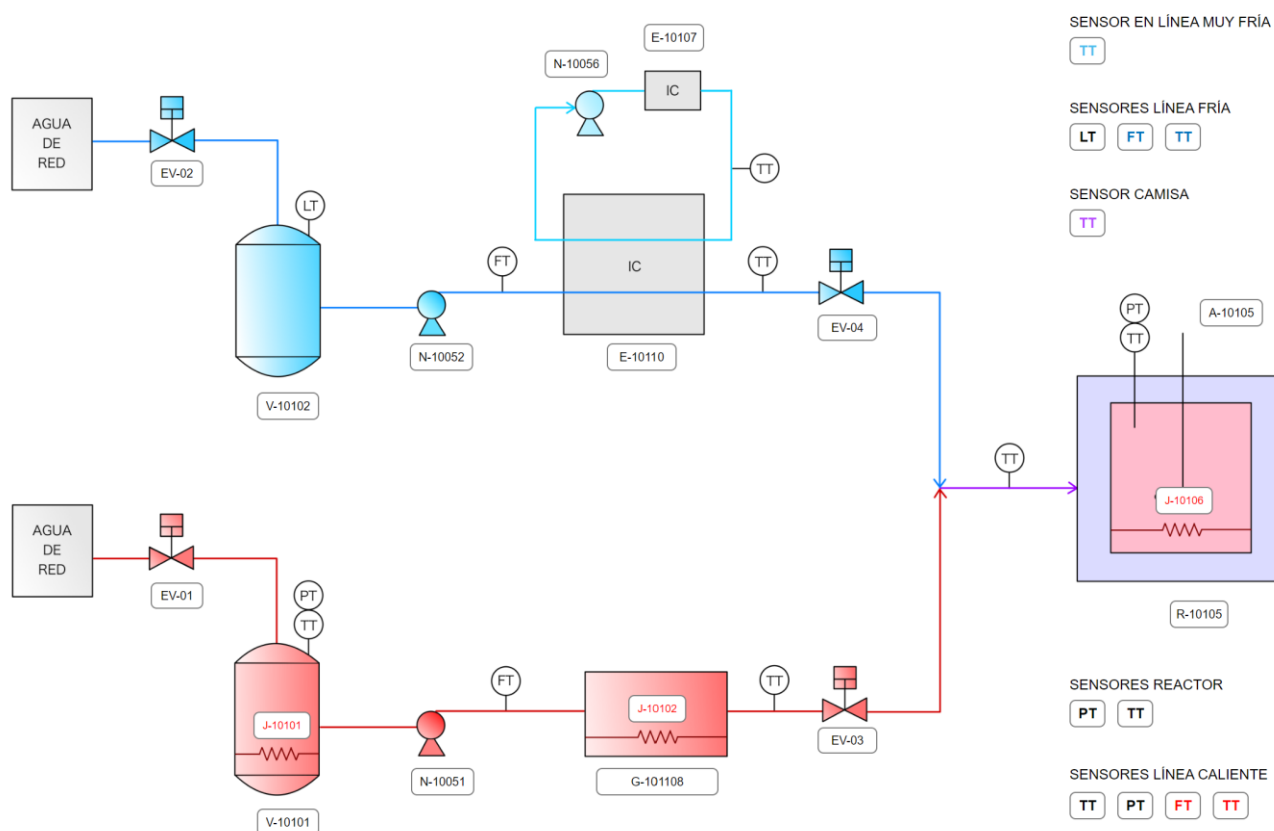


Figura 4.7: Pestaña 'Esquema'

#### 4.2.1.1. Tamaño de la pantalla

Antes de empezar con la programación de los componentes que conforman la GUI informativa, es importante considerar que, dependiendo del monitor en el que se abra el programa, puede haber problemas con la posición de la ventana de programa 'PlantaPiloto.mlapp', así como de las distintas ventanas emergentes que se programarán más adelante al interactuar con ciertos componentes.

Para ello, primero es necesario guardar información acerca del tamaño del monitor en el que se vaya a ejecutar el programa. Esto se consigue definiendo la función screenSize(app).

La información acerca del tamaño de la pantalla se recoge en píxeles, por medio del código set(), y se encuentra en los parámetros pixss(3) y pixss(4).

Como regla general, los índices 1 y 2, guardan información de la posición, y los índices 3 y 4, guardan información del tamaño.

- *pixss(3)*, contiene la posición horizontal de la ventana en pixeles, la cual se guarda en la variable pública definida en propiedades *app.pxHorizMonitorSize*.
- *pixss(4)*, la posición vertical de la ventana en pixeles, la cual se guarda en la variable pública definida en propiedades *app.pxVertMonitorSize*.

```
function screenSize(app)
    % Obtiene el tamaño del monitor en pixels
    % El resultado se almacena en los parámetros públicos:
    % app.pxHorizMonitorSize, app.pxVertMonitorSize
    % Establezco las unidades del monitor en pixels
    set(0, 'units', 'pixels');
    % Obtengo la información en pixels
    pixss = get(0, 'screensize');
    app.pxHorizMonitorSize = pixss(3);
    app.pxVertMonitorSize = pixss(4);
end
```

Figura 4.8: Función screenSize

Las variables públicas mencionadas se definen en el apartado de propiedades o 'properties'.

```
properties (Access = public)
    % MONITOR DEL ORDENADOR
    pxHorizMonitorSize % Tamaño horizontal del monitor (pixels)
    pxVertMonitorSize  % Tamaño vertical del monitor (pixels)
```

Figura 4.9: Propiedades screenSize

#### 4.2.1.2. Ventana de programa centrada

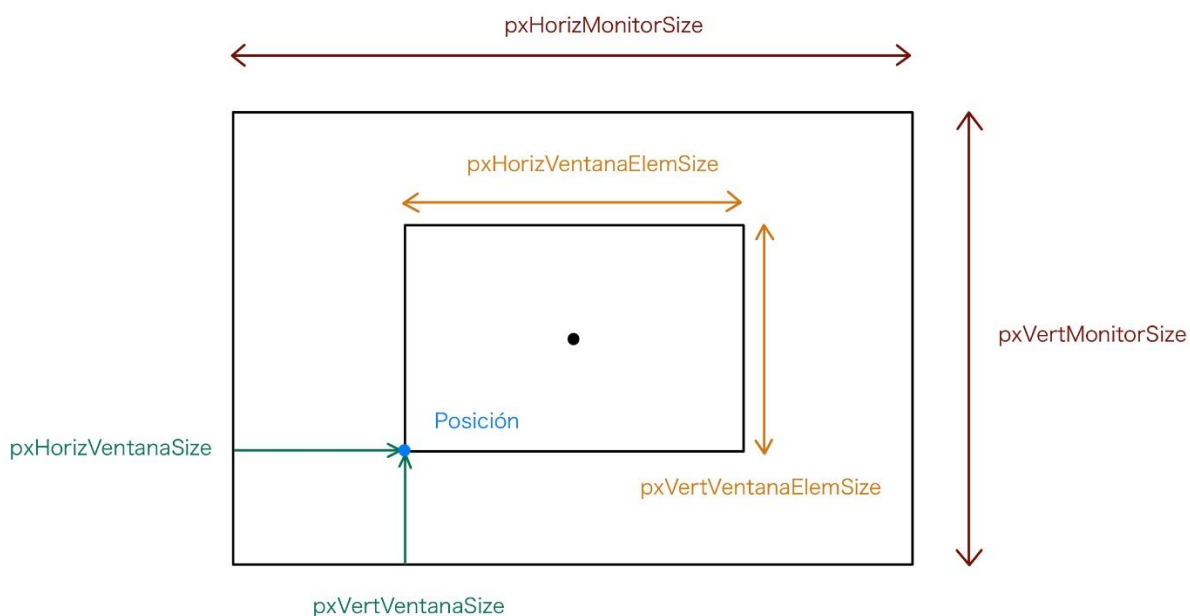
Como se ha mencionado en el apartado anterior, se deben centrar las ventanas del programa '*PlantaPiloto.mlapp*' y las ventanas emergentes fruto de las interacciones con los componentes que se disponen en la interfaz.

En primer lugar, en esta función se menciona la función *screenSize(app)*, descrita en el apartado anterior, de forma que se recoja información acerca del tamaño de la pantalla.

La posición de la ventana se encuentra registrada en los índices 1 y 2 de la variable *app.UIFigure.Position*, y el tamaño en los índices 3 y 4, siendo:

- *app.UIFigure.Position(3)*, el tamaño horizontal de la ventana en pixeles, que se guarda en la variable pública definida en propiedades *app.pxHorizVentanaAppSize*.
- *app.UIFigure.Position(4)*, el tamaño vertical de la ventana en pixeles, que se guarda en la variable pública definida en propiedades *app.pxVertVentanaAppSize*.

Una vez almacenada la información en las correspondientes variables públicas, se procede a centrar la ventana del monitor.



**Figura 4.10: Esquema configuración ventanas**

La posición apropiada para la ventana viene dada por las siguientes coordenadas:

$$\left( \frac{\text{pxHorizMonitorSize} - \text{pxHorizVentanaElemSize}}{2}, \frac{\text{pxVertMonitorSize} - \text{pxVertVentanaElemSize}}{2} \right)$$

De forma que, se almacena la coordenada horizontal,  $\frac{\text{pxHorizMonitorSize} - \text{pxHorizVentanaElemSize}}{2}$ , en la variable pública *app.pxHorizVentanaAppPos*.



Asimismo, se almacena la coordenada vertical,  $\frac{\text{pxVertMonitorSize}-\text{pxVertVentanaElemSize}}{2}$ , en la variable pública `app.pxVertVentanaAppPos`.

El tamaño horizontal de la ventana almacenado en `app.pxHorizVentanaAppPos` se guarda en `app.UIFigure.Position(1)`.

De igual manera, el tamaño vertical de la ventana almacenado en `app.pxVertVentanaAppPos` se guarda en `app.UIFigure.Position(2)`.

A continuación, se muestra el código utilizado para la programación de esta función.

```
function centraVentanaApp(app)
    % y app.inchHorizScrSize y app.inchVertScrSize
    screenSize(app);
    % Obtiene el tamaño original de la ventana de la app
    % No me interesa la posición original de la ventana
    % (índices 1 y 2 de UIFigure.Position)
    app.pxHorizVentanaAppSize = app.UIFigure.Position(3);
    app.pxVertVentanaAppSize = app.UIFigure.Position(4);
    % Centra la ventana en el monitor cambiando la posición
    % de la ventana (al final se restaura).
    app.pxHorizVentanaAppPos = (app.pxHorizMonitorSize - app.pxHorizVentanaAppSize)/2;
    app.pxVertVentanaAppPos = (app.pxVertMonitorSize - app.pxVertVentanaAppSize)/2;
    app.UIFigure.Position(1) = app.pxHorizVentanaAppPos;
    app.UIFigure.Position(2) = app.pxVertVentanaAppPos;
end
```

**Figura 4.11: Función centraVentanaApp**

Siendo las variables públicas definidas en el apartado de propiedades o '*properties*', las que se muestran a continuación.

```
% VENTANA PRINCIPAL DE LA APP
% La posición vertical se cuenta desde la parte inferior del
% monitor.
pxHorizVentanaAppPos % Posición horizontal de la ventana de la app (pixels)
pxVertVentanaAppPos % Posición vertical de la ventana de la app (pixels)
pxHorizVentanaAppSize % Tamaño horizontal de la ventana de la app (pixels)
pxVertVentanaAppSize % Tamaño vertical de la ventana de la app (pixels)

% VENTANA DE INFORMACIÓN DE LOS ELEMENTOS
pxHorizVentanaElemSize = 1050; % Tamaño horizontal de la ventana de información de elementos (pixels)
pxVertVentanaElemSize = 700; % Tamaño vertical de la ventana de información de elementos (pixels)
pxHorizVentanaElemPos; % Posición horizontal de la ventana de información de elementos (pixels)
pxVertVentanaElemPos; % Posición vertical de la ventana de información de elementos (pixels)
```

**Figura 4.12: Propiedades centraVentanaApp**

#### 4.2.2. Ventana emergente

Para ello, primero se define una función pública llamada *creaVentanaDeElemento()*.

Para crear una ventana emergente, según el lenguaje de Matlab, se utiliza el elemento *UIFigure*, al que se le modifican las propiedades:

- *'Name'*: El nombre de la ventana, que se guarda en la variable *'titulo'*, creada por el usuario.
- *'WindowsStyle'*: Referido a la funcionalidad de la ventana. Se elige *'modal'*, lo cual implica que, al abrir una ventana, se debe cerrar para abrir cualquier otra. De este modo se evita una saturación de ventanas abiertas, pudiendo evitar fallos en el programa.

A continuación, calcula la posición para centrar la ventana emergente en función del monitor en el que se abra el programa. Estas medidas se calculan de forma análoga a las correspondientes a la ventana *'plantapiloto.mlapp'*, tal y como se puede observar en el código mostrado.

```
function creaVentanaDeElemento(app, titulo, etiqueta, foto, ficheroTablaCSV, ficheroHTML)
% VENTANA
% MATLAB llama 'figure' a las ventanas
ventana = uifigure('Name',titulo,'WindowStyle','modal');
% Calcula la posición para centrar esta ventana emergente
app.pxHorizVentanaElemPos = (app.pxHorizMonitorSize - app.pxHorizVentanaElemSize)/2;
app.pxVertVentanaElemPos = (app.pxVertMonitorSize - app.pxVertVentanaElemSize)/2;
ventana.Position = [app.pxHorizVentanaElemPos app.pxVertVentanaElemPos
    app.pxHorizVentanaElemSize app.pxVertVentanaElemSize];
```

**Figura 4.13: Función creaVentanaDeElemento (1)**

Las variables públicas definidas en el apartado de propiedades, se muestran a continuación.

En este caso, se ha fijado el tamaño de la ventana emergente deseado, siendo de 1050x800 píxeles. Estas medidas son adecuadas para una correcta visualización en cualquier tamaño de monitor.

```
% VENTANA DE INFORMACIÓN DE LOS ELEMENTOS
pxHorizVentanaElemSize = 1050; % Tamaño horizontal de la ventana de información de elementos (pixels)
pxVertVentanaElemSize = 800; % Tamaño vertical de la ventana de información de elementos (pixels)
pxHorizVentanaElemPos; % Posición horizontal de la ventana de información de elementos (pixels)
pxVertVentanaElemPos; % Posición vertical de la ventana de información de elementos (pixels)
```

**Figura 4.14: Propiedades creaVentanaDeElemento**

Esta función tiene carácter general, y se podrá especificar más adelante para los Callback asociados a cada botón.

#### 4.2.2.1. Rejilla

La ventana emergente no es editable como las ventanas principales en las que se disponen los botones en 'Design View'. Esto es, no se visualiza su diseño mientras se organiza, forzando a programar por tanteo la distribución de la información de la ventana aparente para una vista clara y ordenada.

Se debe entonces, crear una rejilla mediante la que dividir la ventana emergente, de modo que se distribuyan los elementos ordenadamente.

La forma de crear una rejilla en el entorno de Matlab es mediante el código `uigridlayout()`, al que se incorpora la ventana creada donde ira la rejilla y el formato de la rejilla. En este caso, se elige una rejilla compuesta por cuatro filas y dos columnas.

```
% REJILLA (grid)
% Asigna una rejilla (grid) de 4x2 a la ventana
gl = uigridlayout(ventana,[4,2]);
% Alturas fijas de filas
pxAltoFila1 = 40;
pxAnchoCol1 = 350;
% Asigno altura de la fila 2 para que se pueda ver la tabla
pxAltoFila2 = 150;
pxAltoFila4 = 40;
pxAltoFila3 = app.pxVertVentanaElemSize - pxAltoFila1 - pxAltoFila2 - pxAltoFila4;
gl.RowHeight = {pxAltoFila1, pxAltoFila2, pxAltoFila3, pxAltoFila4};
% Anchuras fijas de columnas
pxAnchoCol2 = app.pxHorizVentanaElemSize - pxAnchoCol1 - 30;
gl.ColumnWidth = {pxAnchoCol1, pxAnchoCol2};
```

**Figura 4.15: Rejilla creaVentanaDeElemento**

Se definen las distancias que ocupa cada fila y columna de la rejilla:

- En el caso que nos ocupa, la primera fila tendrá un alto de 40 pixeles y la primera columna, un ancho de 350 pixeles. Esta casilla será ocupada por la imagen del elemento a seleccionar.
- En principio, se asigna un alto para la segunda fila de 150 pixeles, de forma que se pueda apreciar la tabla informativa, y un alto para la cuarta fila de 40 pixeles. Esta última fila servirá de retención para el resto de filas.

La operación `pxAltoFila3` se calcula restándole al alto de la ventana emergente, el alto de las filas 1,2 y 4.

Finalmente, se adjudica a `gl.RowHeight`, que contiene información sobre el alto de cada fila, los valores calculados anteriormente.

- Del mismo modo, se asigna el ancho restante al adjudicado para la primera columna, a la segunda columna, mediante *pxAnchoCol2*.

La operación *pxAnchoCol2* se calcula restándole al ancho de la ventana emergente, el ancho de la primera columna. Adicionalmente, al resultado se le resta 30, debido a pixeles que no tiene en cuenta esta rejilla en Matlab.

Finalmente, se adjudica a *gl.ColumnWidth*, que contiene información sobre el alto de cada fila, los valores calculados anteriormente.

En los siguientes apartados se muestra en qué lugar de la rejilla se encuentran los componentes que conforman la ventana emergente programada para cada botón.

#### 4.2.2.2. Etiqueta

La etiqueta que contiene el título del elemento a describir, se añade mediante el código *uicontrol()*, al que se adjunta la rejilla donde viene colocada, *gl*. Esta etiqueta se almacena en la variable '*lbl*'.

```
% ETIQUETA (fila 1 completa)
lbl = uicontrol(gl);
lbl.Text = etiqueta;
lbl.FontSize = 15;
lbl.FontWeight = 'bold';
lbl.HorizontalAlignment = 'center';
lbl.VerticalAlignment = 'center';
lbl.Layout.Row = 1;
lbl.Layout.Column = [1 2];
```

Figura 4.16: Etiqueta creaVentanaDeElemento

Mediante el código:

- *lbl.Text*, guardaremos en '*etiqueta*' el nombre de la variable. Esta etiqueta tendrá distintos nombres según el elemento que se seleccione.
- *lbl.FontSize*, se selecciona el tamaño de letra para esta etiqueta.
- *Lbl.FontWeight*, se selecciona '*bold*' para que el texto aparezca en negrita.
- *lbl.HorizontalAlignment* y *lbl.VerticalAlignment*, se centra esta etiqueta en la casilla correspondiente de la rejilla, seleccionando '*center*'.
- *lbl.Layout.Row*, se selecciona la fila de la rejilla donde se quiere colocar la etiqueta. En este caso, se situará en la primera fila de la rejilla.
- *lbl.Layout.Column*, se selecciona las columnas de la rejilla donde se quiere colocar la etiqueta. En este caso, se situará en el centro de ambas columnas.

#### 4.2.2.3. Fotografía

La fotografía del elemento a describir, se añade mediante el código `uiimage()`, especificando el lugar donde se coloca, en este caso, la rejilla `gl`. Esta imagen, se almacena en la variable `'im'`.

```
% FOTO
im = uiimage(gl);
pathFoto = strcat('img/', foto);
im.ImageSource = pathFoto;
im.ScaleMethod = 'fit';
im.HorizontalAlignment = 'center';
im.VerticalAlignment = 'top';
im.Layout.Row = [2 3];
im.Layout.Column = 1;
```

**Figura 4.17: Fotografía creaVentanaDeElemento**

Por medio del código `strcat()`, se selecciona la ruta en la que se encuentra la fotografía que se quiere incorporar, en `'img/'`. Se incluye la variable `'foto'` que incluirá para cada caso, la fotografía a seleccionar. Esta ruta se almacena en la variable `'pathPhoto'`.

Mediante el código:

- `im.ImageSource`, se guarda la ruta donde se encuentra la imagen.
- `im.ScaleMethod`, se asegura de que la imagen cabe dentro del lugar seleccionado.
- `im.HorizontalAlignment`, se centra horizontalmente, la imagen en la casilla de la rejilla correspondiente, seleccionando `'center'`.
- `im.VerticalAlignment`, se ajusta en la zona de arriba, la imagen en la casilla de la rejilla correspondiente, seleccionando `'top'`.
- `im.Layout.Row`, se selecciona las filas de la rejilla donde se quiere colocar la imagen. En este caso, se situará en la segunda y tercera fila.
- `im.Layout.Column`, se selecciona la columna de la rejilla donde se quiere colocar la imagen. En este caso, se situará en la primera columna.

#### 4.2.2.4. Tabla

La tabla resumen de las especificaciones del elemento a describir, se añade mediante el código `uitable()`, especificando el lugar donde se coloca, en este caso, la rejilla `gl`.

Esta tabla se almacena en la variable `'uitbl'`.

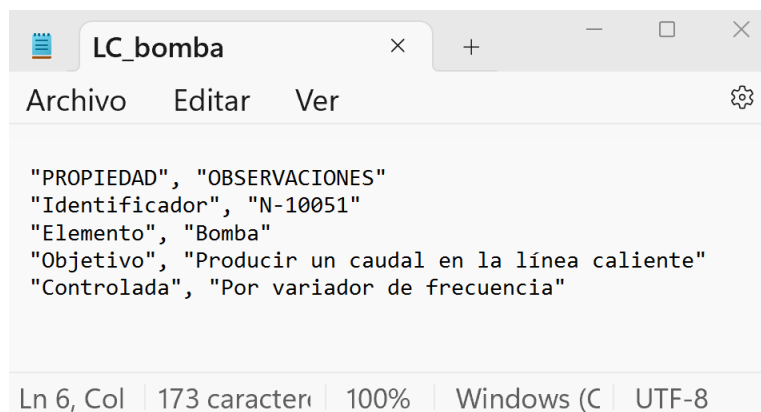
```
% TABLA
uitbl = uitable(gl);
pathTabla = strcat('datos/', ficheroTablaCSV);
uitbl.Data = readtable(pathTabla);
uitbl.Layout.Row = 2;
uitbl.Layout.Column = 2;
```

**Figura 4.18: Tabla creaVentanaDeElemento**

Por medio del código `strcat()`, se selecciona la ruta en la que se encuentra la tabla que se quiere incorporar, en 'datos/'. Se incluye la variable '`ficheroTablaCSV`' que indicará, para cada caso, la tabla seleccionada. Esta ruta se almacena en la variable '`pathTabla`'.

Mediante el código:

- `uitbl.Data`, se lee la tabla correspondiente en la ruta donde se encuentra.
- `uitbl.Layout.Row`, se selecciona la fila de la rejilla donde se quiere colocar la tabla. En este caso, se situará en la segunda fila.
- `uitbl.Layout.Column`, se selecciona la columna de la rejilla donde se quiere colocar la tabla. En este caso, se situará en la segunda columna.



**Figura 4.19: Documento 'csv Bomba hidráulica línea caliente'**

#### 4.2.2.5. Descripción

La descripción del elemento se trata de un archivo html, y se añade mediante el código `uihtml()`, especificando el lugar donde se coloca, en este caso, la rejilla `gl`. Este archivo html se almacena en la variable '`h`'.

```

% HTML
h = uihtml(gl);
pathFicheroHTML = strcat('datos/', ficheroHTML);
h.HTMLSource = pathFicheroHTML;
h.Layout.Row = 3;
h.Layout.Column = 2;

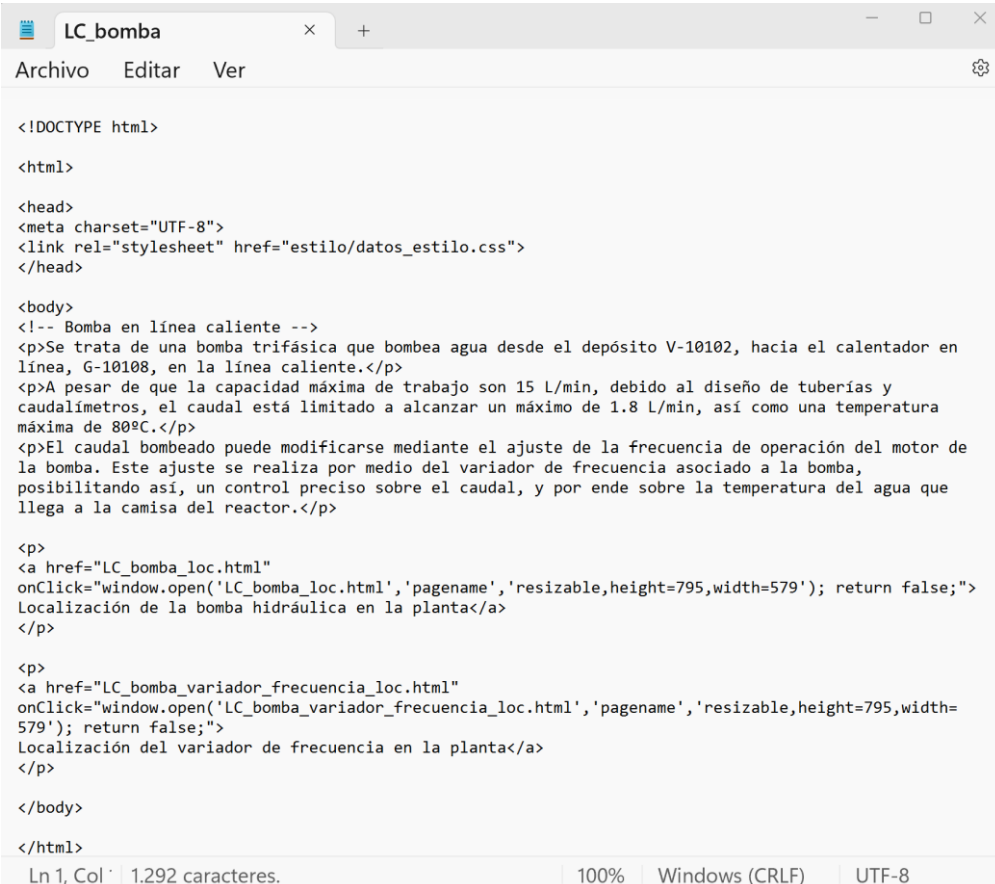
```

Figura 4.20: html 'Ventana de elemento'

Por medio del código *strcat()*, se selecciona la ruta en la que se encuentra el archivo que se quiere incorporar, en 'datos/'. Se incluye la variable '*ficheroHTML*' que incluirá para cada caso, la descripción a seleccionar. Esta ruta se almacena en la variable '*pathFicheroHTML*'.

Mediante el código:

- *h.HTML.Source*, se guarda la ruta donde se encuentra el fichero.
- *h.Layout.Row*, se selecciona la fila de la rejilla donde se quiere colocar la descripción. En este caso, se situará en la tercera fila.
- *h.Layout.Column*, se selecciona la columna de la rejilla donde se quiere colocar la descripción. En este caso, se situará en la segunda columna.



```

<!DOCTYPE html>

<html>

<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="estilo/datos_estilo.css">
</head>

<body>
<!-- Bomba en línea caliente -->
<p>Se trata de una bomba trifásica que bombea agua desde el depósito V-10102, hacia el calentador en línea, G-10108, en la línea caliente.</p>
<p>A pesar de que la capacidad máxima de trabajo son 15 L/min, debido al diseño de tuberías y caudalímetros, el caudal está limitado a alcanzar un máximo de 1.8 L/min, así como una temperatura máxima de 80°C.</p>
<p>El caudal bombeado puede modificarse mediante el ajuste de la frecuencia de operación del motor de la bomba. Este ajuste se realiza por medio del variador de frecuencia asociado a la bomba, posibilitando así, un control preciso sobre el caudal, y por ende sobre la temperatura del agua que llega a la camisa del reactor.</p>

<p>
<a href="LC_bomba_loc.html"
onClick="window.open('LC_bomba_loc.html','pagename','resizable,height=795,width=579'); return false;">
Localización de la bomba hidráulica en la planta</a>
</p>

<p>
<a href="LC_bomba_variador_frecuencia_loc.html"
onClick="window.open('LC_bomba_variador_frecuencia_loc.html','pagename','resizable,height=795,width=579'); return false;">
Localización del variador de frecuencia en la planta</a>
</p>

</body>

</html>

```

Figura 4.21: Documento html 'Bomba línea caliente'

4.2.2.6. *Callbacks*

Se disponen una serie de callbacks asociados a la interacción con los botones distribuidos por la pestaña 'Esquema'.

El callback asociado al botón de la bomba N-10051, encargada de impulsar el fluido a través de la línea caliente, consta del siguiente código:

```
% Callback function: ELELC_BombaMenu, LCBombaButton
function LCBombaButtonPushed(app, event)
    titulo = 'Bomba hidráulica (línea caliente)';
    etiqueta = 'BOMBA HIDRÁULICA (LÍNEA CALIENTE)';
    foto = 'LC_bomba.jpg';
    ficheroTablaCSV = 'LC_bomba.csv';
    ficheroHTML = 'LC_bomba.html';
    creaVentanaDeElemento(app, titulo, etiqueta, foto, ficheroTablaCSV, ficheroHTML);
end
```

**Figura 4.22: callback 'Bomba línea caliente'**

- Se modifica la variable '*titulo*' que muestra el nombre de la ventana emergente.
- Se modifica la variable '*etiqueta*' para que muestre el nombre del elemento correspondiente.
- Se almacena en la variable '*foto*', la fotografía que hace referencia al elemento correspondiente.
- Se almacena en la variable '*ficheroTablaCSV*', la tabla que hace referencia al elemento correspondiente.
- Se almacena en la variable '*ficheroHTML*', el fichero con la descripción del elemento correspondiente.

Finalmente, se llama a la función *creaVentanaDeElemento()*, al que se le adjunta la información anterior.

Para cada Callback, se operará de manera similar, únicamente cambiando los archivos que guardan la información correcta para cada elemento. De este modo, se consigue especificar el tipo de elemento para cada botón en la pestaña 'Esquema'.

A continuación, se muestra la ventana emergente programada tras interactuar con el botón correspondiente a la bomba N-10051.



## Desarrollo de una interfaz gráfica de usuario para planta piloto

Bomba hidráulica (línea caliente)

### BOMBA HIDRÁULICA (LÍNEA CALIENTE)

PROPIEDAD	OBSERVACIONES
Identificador	N-10051
Elemento	Bomba
Objetivo	Producir un caudal en la línea caliente
Controlada	Por variador de frecuencia

Se trata de una bomba trifásica que bombea agua desde el depósito V-10102, hacia el calentador en línea, G-10108, en la línea caliente.

A pesar de que la capacidad máxima de trabajo son 15 L/min, debido al diseño de tuberías y caudalímetros, el caudal está limitado a alcanzar un máximo de 1.8 L/min, así como una temperatura máxima de 80°C.

El caudal bombeado puede modificarse mediante el ajuste de la frecuencia de operación del motor de la bomba. Este ajuste se realiza por medio del variador de frecuencia asociado a la bomba, posibilitando así, un control preciso sobre el caudal, y por ende sobre la temperatura del agua que llega a la camisa del reactor.

[Localización de la bomba hidráulica en la planta](#)

[Localización del variador de frecuencia en la planta](#)

Referencias del variador de frecuencia:

[Página web](#)

[Datos técnicos](#)



Figura 4.23: Ventana de elemento: Bomba hidráulica línea caliente

Al pulsar el enlace 'Localización del variador de frecuencia en la planta', muestra al usuario una fotografía del armario donde se encuentran los variadores de frecuencia.



Figura 4.24: Localización 'Variadores de frecuencia'

Al pulsar el enlace 'Localización de la bomba hidráulica en la planta' muestra al usuario una imagen de la planta donde se encuentra señalada la bomba en color amarillo.

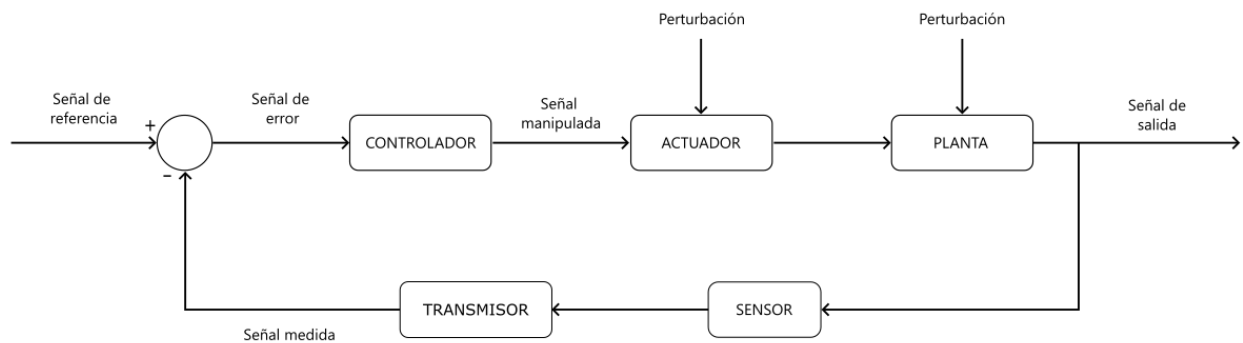


Figura 4.25: Localización 'Bomba línea caliente'

## 1. ESTUDIO DE LAZOS DE CONTROL

En este apartado se explicará el funcionamiento de los lazos de control y se realizará un estudio de los posibles lazos que se pueden observar en la planta. Este estudio tiene el objetivo de desarrollar, más adelante, una interfaz gráfica de usuario explicativa de cada uno de los lazos mencionados.

Un lazo de control cerrado general viene representado por el siguiente esquema:



**Figura 1.1: Lazo de control cerrado general**

Un lazo de control cerrado, opera con el objetivo de mantener una variable de interés en un valor deseado. Este sistema consta de varios elementos clave que colaboran para lograr el control de una variable concreta. A continuación, se describen los elementos principales:

- 1) Sensor/Transductor: Este componente mide la variable a controlar o la salida del sistema y convierte esta información en una señal eléctrica, o de otro tipo, que pueda ser interpretada por el controlador.
- 2) Transmisor: En el caso de que el sensor no lleve integrado un transductor para convertir la señal física en una señal de tipo eléctrico, se hace uso de un transmisor.
- 3) Comparador: Responsable de procesar la señal proveniente del sensor y compararla con la referencia o valor deseado. Basándose en esa comparación, el comparador envía una señal de error al controlador para ajustar el sistema y mantener la variable controlada en el valor objetivo.
- 4) Controlador: Procesa la variable de error proveniente del comparador, devolviendo la señal de control.
- 5) Actuador: Recibe señales del controlador y realiza los ajustes necesarios en la variable manipulada, y a su vez, actúa sobre la planta para modificar el valor de la variable controlada. Puede ser un motor, una válvula o cualquier dispositivo capaz de modificar la variable de interés.

- 6) Proceso o planta: Representa el sistema físico en el que se puede modificar la variable controlada. Puede ser un horno, un sistema de calefacción, etc. En este caso, se trata del reactor.

Este enfoque mejora la precisión y la estabilidad en comparación con un sistema de lazo abierto con perturbaciones, donde no hay realimentación y la regulación depende únicamente de la configuración inicial, siempre y cuando se utilice un regulador adecuado. Por otra parte, un lazo de control en lazo cerrado hace que el sistema sea menos sensible a las posibles perturbaciones.

Las principales señales involucradas en un lazo de control cerrado son:

- 1) Señal de referencia: Es la señal, generalmente eléctrica, que establece el valor al cual se desea mantener la variable controlada. Esta señal es la solicitada por el usuario en el sistema de control y representa el objetivo que se busca alcanzar.
- 2) Señal de error: Se obtiene comparando la señal de referencia con la señal real enviada por el transmisor o transductor, asociado al sensor. La señal de error indica la discrepancia entre el valor deseado y el valor actual de la variable controlada. Un error negativo indica que la variable se encuentra por encima del valor deseado, mientras que un error positivo indica que está por debajo.
- 3) Señal de control: Esta señal se genera en el controlador después de procesar la señal de error. Representa la corrección necesaria para minimizar el error y llevar la variable controlada al valor deseado, por medio de un actuador.
- 4) Señal de manipulada: Esta señal se genera en el actuador para realizar cambios en la variable controlada.
- 5) Señal de salida: Es la salida real del sistema físico, medida por el sensor. Representa el estado actual de la variable controlada y se utiliza para comparar con la señal de referencia, cerrando así el lazo de control.
- 6) Señal medida de realimentación: Corresponde a la información proveniente de la salida, la cual mediante un transductor o transmisor se transforma en una señal eléctrica que se introduce de nuevo al controlador. La realimentación permite ajustes continuos de la señal de control, manteniendo la señal controlada y con un valor cercano al deseado.
- 7) Perturbación: Esta señal representa cualquier influencia, ya sea externa o interna, o cambio no anticipado que afecta al sistema y puede introducir variaciones no deseadas en la variable controlada. Las perturbaciones son eventos imprevistos que pueden surgir de factores externos al sistema controlado, como cambios en las condiciones ambientales, o de factores internos como el cambio de la presión con la que llega un fluido a una válvula.

### 4.3. Lazo de nivel del depósito V-10102

Consiste en el control de nivel de agua en el depósito de la línea fría, V-10102. En este lazo, el nivel únicamente es controlado mediante la abertura de la válvula EV-02.

#### Características importantes:

- Lugar de medición: Depósito V-10102
- Actuador: Electroválvula EV-02 (Todo/nada)
- Sensor: Radar por onda guiada (Rosemount 3300)
- Variable manipulada: Caudal de entrada al depósito
- Variable controlada: Nivel de llenado

#### Descripción del lazo:

- 1) Sensor:** Un sensor de nivel mide el nivel de llenado del depósito correspondiente a la línea fría. Proporciona información en tiempo real sobre el nivel de llenado actual. El transmisor incorporado en el sensor convierte la señal de nivel medida en una señal de corriente y la envía al comparador.
- 2) Comparador:** El comparador compara la medición actual de nivel de llenado con el valor de referencia deseado, de modo que la diferencia entre ellos es la señal de error. Si la señal de referencia es mayor que la señal medida, entonces la señal de error resultante será positiva y se deberá aumentar el nivel de llenado. En cambio, si la señal de referencia es menor, la señal de error resultante será negativa, lo que significa que se necesitará disminuir la altura de llenado. Esta señal se envía al controlador.
- 3) Controlador:** El controlador toma la señal del comparador y genera una histéresis para controlar adecuadamente la válvula todo/nada. La histéresis se genera para evitar que la válvula se abra y cierre constantemente alrededor de un punto de referencia, introduciendo un margen en el controlador. Si el nivel cae por debajo de cierto umbral, el controlador abrirá la válvula. Sin embargo, la válvula no se cerrará hasta que el nivel haya subido por encima de un umbral superior. Esto crea una banda muerta o histéresis que evita oscilaciones innecesarias. Esta señal se envía a la válvula.
- 4) Actuador:** En este caso, el actuador es la válvula todo/nada, la cual recibe la señal del controlador. En función de esta señal, la válvula se abrirá o se cerrará completamente, permitiendo o no el paso de fluido hasta el depósito, lo cual afectará directamente al nivel de llenado del mismo.

#### 4.4. Lazo de caudal en la línea de agua fría

Consiste en el control de caudal de agua en la línea fría mediante la potencia eléctrica suministrada a la bomba N-10052.

##### Características importantes:

- Lugar de medición: Salida del depósito V-10102
- Actuador: Motor de la bomba N-10052
- Sensor: Caudalímetro de Coriolis
- Variable manipulada: Velocidad de giro del motor conectado a la bomba.
- Variable controlada: Caudal

##### Descripción del lazo:

- 1) Sensor:** Un caudalímetro en la línea fría mide en tiempo real el flujo volumétrico. El transductor convierte la señal de nivel medida en una corriente eléctrica, la cual se envía al comparador.
- 2) Comparador:** Este componente compara el caudal actual medido con el valor de referencia deseado. La diferencia entre ambas señales determina el error. Si la señal de referencia supera la medida, el error resultante es positivo, indicando la necesidad de aumentar el caudal. En cambio, si la señal de referencia es menor, el error es negativo, señalando la necesidad de reducir el caudal. Esta información se transmite al controlador.
- 3) Controlador:** El controlador utiliza la información del error proporcionada por el comparador para generar una señal de control. Esta señal se envía al actuador, ajustando la velocidad del motor de la bomba. El objetivo es minimizar el error y mantener el caudal lo más cercano posible al valor de referencia.
- 4) Actuador:** El motor de la bomba actúa como actuador en este caso. Su variador de frecuencia ajusta la frecuencia y, por ende, la potencia suministrada. Esta potencia, directamente proporcional a la velocidad del motor y al caudal de agua, se ajusta conforme a la señal de control generada. Así, si el caudal medido está por debajo del valor deseado, el variador aumenta la frecuencia para incrementar la velocidad del motor y, por consiguiente, el caudal. Por el contrario, si el caudal está por encima del valor deseado, el variador disminuye la frecuencia para reducir la velocidad del motor y, en consecuencia, el caudal.

#### 4.5. Lazo de temperatura a la salida de la línea de agua fría

Consiste en el control de temperatura de agua en la línea fría a la entrada de la camisa, previo al mezclado con la línea caliente. En este lazo, la temperatura únicamente es controlada mediante la potencia eléctrica suministrada a la bomba N-10052, que permite la variación de velocidad del agua.

##### Características importantes:

- Lugar de medición: Línea de agua fría a la salida
- Actuador: Motor de la bomba N-10052
- Sensor: Sensor de temperatura RTD (Pt100)
- Variable Manipulada: Caudal de agua fría
- Variable Controlada: Temperatura

##### Descripción del lazo:

- 1) Sensor:** Un sensor de temperatura ubicado en la salida de la línea fría mide constantemente la temperatura del agua. Proporciona datos en tiempo real sobre la temperatura actual. El transmisor transforma esta información en una señal de corriente proporcional y la transmite al comparador.
- 2) Comparador:** Esta parte del sistema compara la temperatura actual medida con el valor de referencia deseado. La diferencia entre estas señales determina el error. Si la señal de referencia es mayor que la señal medida, el error resultante es positivo, indicando la necesidad de aumentar la temperatura. Si la señal de referencia es menor, el error es negativo, sugiriendo la reducción de la temperatura. Este error se envía al controlador.
- 3) Controlador:** Utilizando la información del error proporcionada por el comparador, el controlador genera una señal de control. La salida del controlador es una señal enviada al actuador para ajustar la velocidad del motor de la bomba, regulando así el caudal. El objetivo del controlador es minimizar el error y mantener la temperatura lo más cercana posible al valor de referencia.
- 4) Actuador:** El actuador en este caso es el motor de la bomba que suministra un determinado caudal. El variador de frecuencia ajusta la frecuencia y suministra una señal de tensión modulada en frecuencia al motor, ajustando la potencia eléctrica entregada a la bomba. Un cambio en la potencia eléctrica supone una variación proporcional en la velocidad del motor, lo que afecta directamente al caudal de agua, y por tanto a la temperatura.  
Esta línea se encarga de la refrigeración del agua, impulsándola hacia un intercambiador de calor mediante una bomba. Al incrementar el caudal de agua que llega al intercambiador, se empeora la transmisión de calor del agua de trabajo al fluido refrigerante, resultando en una temperatura más alta en la salida.

Así, si la temperatura medida está por debajo del valor deseado, el variador aumenta la frecuencia para elevar la velocidad del motor y, por ende, el caudal, elevando la temperatura. Por otro lado, si la temperatura supera el valor deseado, el variador baja la frecuencia para disminuir la velocidad del motor, lo que reduce el caudal y logra una temperatura más baja a la salida.

#### 4.6. Lazo de temperatura a la salida de la línea de agua caliente

Consiste en el control de temperatura de agua en la línea caliente a la entrada de la camisa, previo al mezclado con la línea fría. En este lazo, la temperatura es controlada mediante la potencia térmica suministrada por la resistencia J-10102.

##### Características importantes:

- Lugar de medición: Línea de agua caliente a la salida
- Actuador: Resistencia J-10102
- Sensor: Sensor de temperatura RTD (Pt100)
- Variable Manipulada: Potencia térmica generada por la resistencia
- Variable Controlada: Temperatura

##### Descripción del lazo:

- 1) Sensor:** Un sensor de temperatura mide la temperatura del agua a la salida de la línea caliente, proporcionando información en tiempo real sobre la temperatura actual.  
Un transmisor convierte la señal de temperatura en una señal de corriente proporcional y la transmite al comparador.
- 2) Comparador:** En esta etapa del sistema, el comparador evalúa la medición actual de la temperatura frente al valor de referencia deseado. La discrepancia entre estas señales genera el error. Si la señal de referencia es superior a la señal medida, el error resultante es positivo, indicando la necesidad de aumentar la temperatura. En contraste, si la señal de referencia es menor, el error es negativo, señalando la necesidad de reducir la temperatura. Este error se transmite al controlador.
- 3) Controlador:** El controlador hace uso de la señal de error anterior y genera una señal de control. La salida del controlador es una señal que se envía al actuador para ajustar potencia térmica generada por la resistencia y, por lo tanto, controlar la temperatura de salida. El controlador busca minimizar el error y mantener la temperatura lo más cercana posible al valor de referencia.
- 4) Actuador:** El actuador en este caso es la resistencia del calentador en línea. El tiristor asociado a la resistencia ajusta la intensidad eléctrica que llega a la resistencia para aumentar la potencia eléctrica suministrada por la resistencia.



Así, si la temperatura medida está por debajo del valor deseado, el tiristor incrementa la potencia eléctrica de la resistencia para aumentar la temperatura a la salida. Por otro lado, si la temperatura supera el valor deseado, el tiristor disminuye la potencia térmica de la resistencia para lograr una temperatura más baja en ese punto.

#### 4.7. Lazo de temperatura a la salida de la línea de agua fría: En cascada

Se trata de un lazo de control en cascada para controlar la temperatura a la salida de la línea fría de una forma más rápida y eficiente. Consta de un lazo lento de temperatura llamado primario o maestro y un lazo rápido de caudal llamado secundario o esclavo.

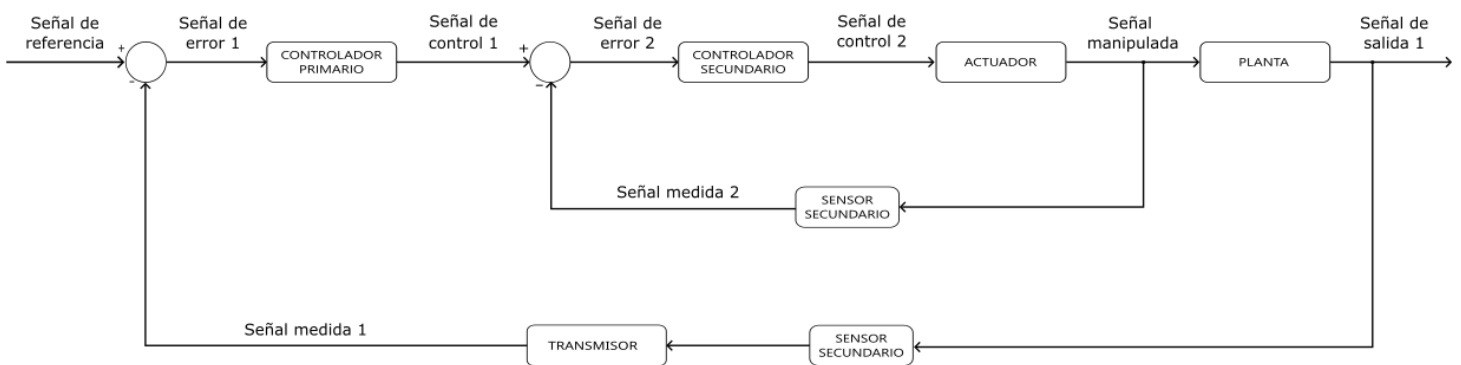


Figura 1.2: Lazo de control en cascada. Temperatura del agua en la línea fría

#### Características importantes:

- Lugar de medición: Línea de agua fría a la salida
- Actuador: Motor de la bomba N-10052
- Sensor primario: Sensor de temperatura RTD (Pt100)
- Sensor secundario: Caudalímetro de Coriolis
- Variable Manipulada: Velocidad de giro de la bomba
- Variable Controlada: Temperatura

#### Descripción del lazo:

##### ➤ Lazo secundario o esclavo:

- 1) **Sensor secundario:** Un caudalímetro mide el flujo volumétrico del agua a la salida de la línea fría, proporcionando información en tiempo real sobre el caudal actual.

Convierte la señal de caudal en una señal de corriente proporcional y la envía al comparador del lazo esclavo.

- 2) **Comparador:** El comparador secundario compara la medición actual de caudal con el valor de referencia deseado proveniente del controlador primario, de modo que la diferencia entre ellos es la señal de error. Si la señal de referencia es mayor que la señal medida, entonces la señal de error resultante será positiva, lo que implica que habría que aumentar el caudal. En cambio, si la señal de referencia es menor, la señal de error resultante será negativa y habría que disminuirlo. Esta señal se envía al controlador.
- 3) **Controlador:** El controlador toma la información del error proporcionada por el comparador y genera una señal de control. La salida del controlador es una señal que se envía al actuador para ajustar la velocidad del motor de la bomba y, por lo tanto, controlar el caudal. El controlador busca minimizar el error y mantener el caudal lo más cercano posible al valor de referencia.
- 4) **Actuador:** El actuador en este caso es la bomba que suministra un determinado caudal. El variador de frecuencia ajusta la frecuencia y, por ende, la potencia suministrada a la bomba. Esta potencia se aplica al motor, afectando al caudal obtenido, de modo que se tratará de modificar de acuerdo con la señal de control generada. Así, si el caudal medido está por debajo del valor deseado, el variador aumenta la frecuencia para aumentar la velocidad del motor y, por lo tanto, el caudal. Por el contrario, si el caudal está por encima del valor deseado, el variador disminuye la frecuencia para reducir la velocidad del motor y, por ende, el caudal.

➤ **Lazo primario o maestro**

- 1) **Sensor primario:** Un sensor de temperatura mide la temperatura del agua a la salida de la línea fría.  
Un transmisor convierte la señal de temperatura en una señal de corriente proporcional y la envía al comparador primario.
- 2) **Comparador primario:** El comparador compara la medición actual de temperatura con el valor de referencia deseado, de modo que la diferencia entre ellos es la señal de error. Si la señal de referencia es mayor que la señal medida, entonces la señal de error resultante será positiva, lo que implica que habría que aumentar la temperatura. En cambio, si la señal de referencia es menor, la señal de error resultante será negativa y habría que disminuir la temperatura. Esta señal se envía al controlador primario.
- 3) **Controlador primario:** El controlador toma la información del error proporcionada por el comparador y genera una señal de control. La salida del controlador constituye la señal de referencia para el comparador secundario. Este controlador busca minimizar el error y mantener la temperatura lo más cercana posible al valor de referencia.

- 4) **Actuador:** Ambos lazos, primario y secundario, comparten el mismo actuador, la bomba N-10052. El caudal de agua suministrado por la bomba afecta directamente a la temperatura de la línea. Al incrementar el caudal de agua que llega al intercambiador, se empeora la transmisión de calor entre el agua que pasa por la línea fría y el fluido refrigerante, resultando en una temperatura más alta en la salida.

Así, si la temperatura medida está por debajo del valor deseado, el variador aumenta la frecuencia para incrementar la velocidad del motor y, por ende, el caudal, elevando la temperatura. Por otro lado, si la temperatura supera el valor deseado, el variador disminuye la frecuencia para reducir la velocidad del motor, lo que reduce el caudal y logra una temperatura más baja a la salida.

#### 4.8. Lazo de temperatura del agua en la camisa del reactor.

Consiste en el control de la temperatura del agua en la camisa mediante los caudales suministrados por las bombas situadas en ambas líneas de agua, manteniendo fijas las temperaturas de las líneas de agua fría y caliente. El flujo volumétrico de agua proveniente de las líneas de agua fría y caliente se podría realizar mediante válvulas todo/nada, a pesar de que un control más gradual podría ser preferible para evitar daños en el sistema. Sin embargo, para este planteamiento de lazo, se considerarán abiertas, puesto que un cierre y abertura abruptos de estas válvulas podrían suponer fallos tanto en las tuberías como en las bombas y en el intercambiador de calor.

El lazo se controla por medio de dos actuadores, de modo que se trata de un control de rango partido.

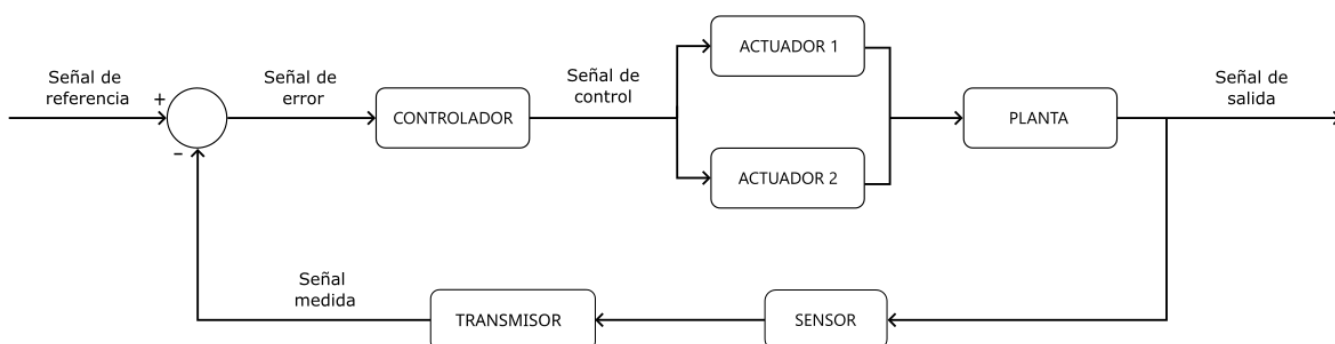


Figura 1.3: Lazo de control en rango partido. Temperatura en la camisa del reactor

Características importantes:

- Lugar de medición: Agua a la entrada de la camisa
- Actuadores: Motores de las bombas N-10051 y N-10052
- Sensor: Sensor de temperatura RTD (Pt100)
- Variables Manipuladas: Caudal de agua fría y caudal de agua caliente
- Variable Controlada: Temperatura

Descripción del lazo:

- 1) **Sensor:** Un dispositivo sensor de temperatura situado en la camisa del reactor realiza mediciones continuas de la temperatura del agua. Un transmisor convierte la señal de temperatura en una señal de corriente proporcional, la cual se transmite al comparador.
- 2) **Comparador:** En esta fase del sistema, el comparador evalúa la medición actual de la temperatura en comparación con el valor de referencia deseado. La discrepancia entre estas señales genera el error. Si la señal de referencia supera a la señal medida, el error resultante es positivo, indicando la necesidad de aumentar la temperatura. En cambio, si la señal de referencia es menor, el error es negativo, señalando la necesidad de reducir la temperatura. Esta señal de error se envía al controlador.
- 3) **Controlador:** El controlador genera dos señales de control por medio de la señal de error proporcionada por el comparador. Las señales se envían a los actuadores de la línea fría y la línea caliente respectivamente, para ajustar la velocidad de las bombas, regulando así el caudal de agua. Con esto se consigue minimizar el error y mantener la temperatura lo más cercana posible al valor de referencia.
- 4) **Actuadores:** El actuador en este caso son las bombas que suministran un determinado caudal de agua proveniente de la línea fría y de agua proveniente de la línea caliente. Los variadores de frecuencia ajustan la frecuencia y suministran una señal eléctrica a los motores, ajustando la potencia eléctrica entregada a cada bomba. Un cambio en la potencia eléctrica supone una variación proporcional en la velocidad del motor, lo que afecta directamente al caudal de agua de ambas líneas, y por tanto a la temperatura de la mezcla. Si la comparación resulta en un valor positivo, se incrementará el caudal suministrado por la bomba que propulsa el agua de la línea caliente y disminuirá en la línea fría. En caso contrario, se aumentará el caudal suministrado en la línea fría y se reducirá en la línea caliente.

## 5. INTERFAZ GRÁFICA: LAZOS DE CONTROL

La pestaña 'Lazos de control' es una pantalla interactiva donde se pueden estudiar distintos lazos de control de la planta.

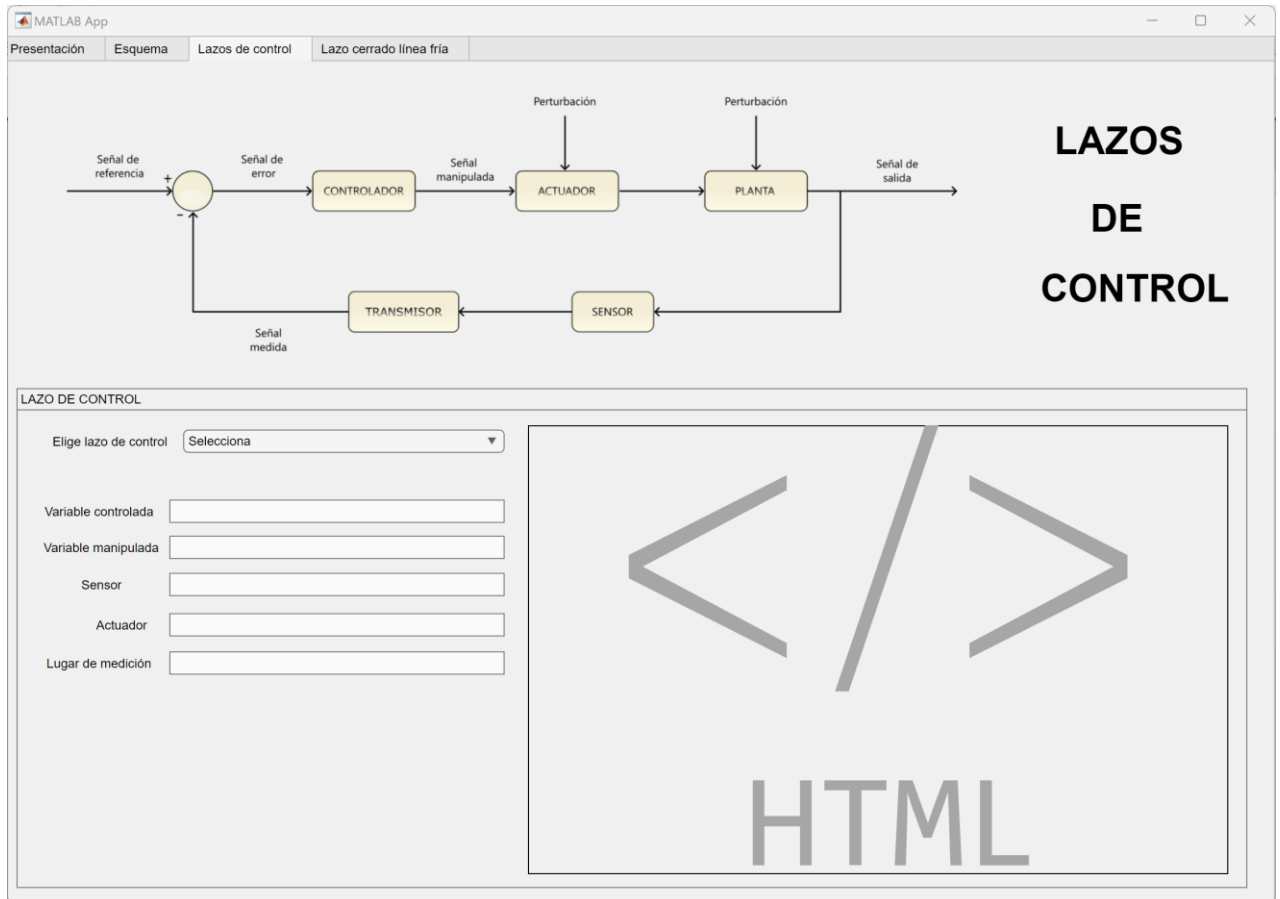


Figura 5.1': Pestaña 'Lazos de control'

Consta de los siguientes componentes:

- Una imagen que muestra un lazo de control cerrado concreto para cada opción seleccionada.
- Un panel que contiene la información aplicada a cada lazo de control específico. El lazo de control se elige por medio de un desplegable o "Drop Down" y muestra información puntual del lazo mediante campos de edición de texto o "EditFields" y una descripción del lazo más detallada, mediante un enlace "html".
- Una serie de botones numerados del 1 al 8 que se activan en función de los componentes que forman el lazo de control cerrado, sustituyendo los bloques correspondientes al controlador, actuador, etc. De esta manera, al pulsar cada uno de los botones muestra una ventana emergente con información acerca de estos componentes.

Se requiere programación en la interfaz para lograr la interacción con el desplegable de forma que muestre cada uno de los lazos.

Primero, se debe añadir cada una de las opciones disponibles para el mismo, en la zona de edición del programa. Cada opción tendrá asociada una numeración.

La primera opción presenta un texto donde se muestra 'Selecciona', cuya función es simplemente incitar al usuario a elegir uno de los disponibles.

Las opciones disponibles de lazos de control, numeradas del número dos al número 8, son las siguientes:

- Lazo de nivel en la línea fría.
- Lazo de nivel en la línea caliente.
- Lazo de caudal en la línea fría.
- Lazo de caudal en la línea caliente.
- Lazo de temperatura a la salida de la línea fría. Es decir, la entrada de la camisa en la línea fría, previamente al mezclado de ambas líneas.
- Lazo de temperatura a la salida de la línea caliente. Es decir, a la entrada de la camisa en la línea caliente, previamente al mezclado de ambas líneas.
- Lazo de control en cascada de temperatura a la salida de la línea fría.
- Lazo de control en rango partido de temperatura en la camisa del reactor.

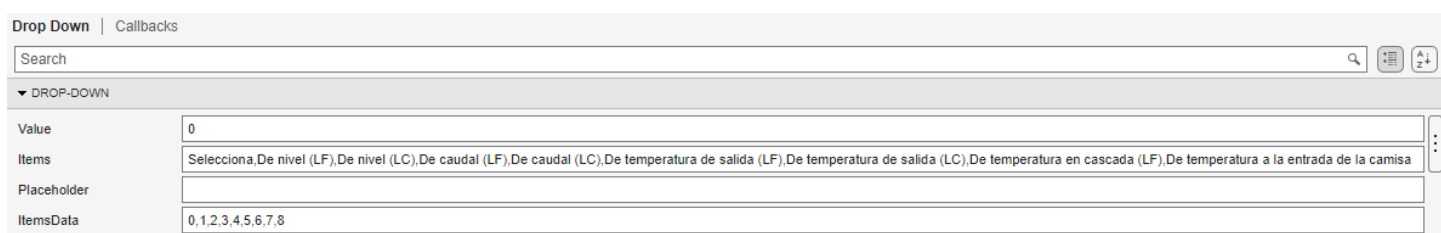


Figura 5.2: Configuración 'Drop Down Lazos de control'

A continuación, se muestra el código necesario para que la interactividad anteriormente mencionada sea posible.

En primer lugar se crea el 'callback' `EligelazoDropDownValueChanged()` que relacionará la selección del desplegable con el resto de componentes de las pestaña.

Creamos un condicional "if" que muestra los escenarios posibles que se pueden encontrar, es decir, los distintos lazos disponibles.

Como hemos mencionado, la primera opción no se trata de ningún lazo y únicamente muestra un esquema general de lazo de control.

```

% Value changed function: EligelazoDropDown
function EligelazoDropDownValueChanged(app, event)
    value = app.EligelazoDropDown.Value;
    app.lazoSeleccionado = value;

    if value == 1
        invisibilizaBotonesLazos(app);
        app.Image.ImageSource = 'img/Lazo_Generico_Color.png';
        app.HTML2.HTMLSource = '';
        app.LugardemedicionEditField.Value = '';
        app.ActuadorEditField.Value = '';
        app.SensorEditField.Value = '';
        app.VariablemanipuladaEditField.Value = '';
        app.VariablecontroladaEditField.Value = '';
        app.LazosButton3.FontSize = 12;

```

**Figura 5.3: Callback EligelazoDropDown (1/2)**

El condicional realmente empieza a partir de la segunda opción. Para programar el 'callback' correspondiente, se crea una serie de variables que contienen la información que ocupan los campos de edición de texto. Por ejemplo, en *app.ActuadorEditField.Value* se almacena cuál es el actuador del lazo de control.

La variable '*descripcion*' constituye el archivo 'html' que se verá incluido en la descripción de la planta. Esta descripción se almacena en *app.HTML2.Data*.

En el componente que contiene la imagen, se incorpora una imagen específica para el lazo elegido.

```

elseif value == 2
    invisibilizaBotonesLazos(app);
    app.Image.ImageSource = 'img/lazo_nivel_LF.png';
    app.HTML2.HTMLSource = 'lazos/descripcionlazo2.html';
    app.LugardemedicionEditField.Value = 'Depósito V-10102';
    app.ActuadorEditField.Value = 'Electroválvula EV-02';
    app.SensorEditField.Value = 'Radar por onda guiada (Rosemount 3300)';
    app.VariablemanipuladaEditField.Value = 'Abertura de la válvula';
    app.VariablecontroladaEditField.Value = 'Nivel de llenado';
    app.LazosButton1.Text = 'Controlador';
    app.LazosButton1.Position = [293 650 100 44];
    app.LazosButton2.Text = 'Actuador';
    app.LazosButton2.Position = [485 650 100 44];
    app.LazosButton3.Text = 'Planta';
    app.LazosButton3.FontSize = 12;
    app.LazosButton3.Position = [664 650 99 43];
    app.LazosButton4.Text = 'Sensor/Tansductor';
    app.LazosButton4.Position = [421 537 153 44];
    app.LazosButton1.Visible = "on";
    app.LazosButton2.Visible = "on";
    app.LazosButton3.Visible = "on";
    app.LazosButton4.Visible = "on";

```

**Figura 5.4: Callback EligelazoDropDown (2/2)**

Para la opción número dos, se visibilizan los botones del 1 al 4, ya que el lazo se compone del controlador, el actuador, la planta o proceso y el sensor/transductor.

Estos botones se posicionan manualmente en los lugares ocupados por los bloques en la imagen correspondiente, y se nombran de la misma forma que los componentes que sustituyen.

**LAZOS DE CONTROL**

**LAZO DE CONTROL**

Elige lazo de control: De nivel (LF)

Variable controlada: Nivel de llenado

Variable manipulada: Caudal de entrada al depósito

Sensor: Radar por onda guiada (Rosemount 3300)

Actuador: Electroválvula EV-02

Lugar de medición: Depósito V-10102

**DESCRIPCIÓN DEL LAZO**

Consiste en el control de nivel de agua en el depósito de la línea fría, V-10102. En este lazo, el nivel únicamente es controlado mediante la abertura de la válvula EV-02.

1. Sensor: Un sensor de nivel mide el nivel de llenado del depósito correspondiente a la línea fría. Proporciona información en tiempo real sobre el nivel de llenado actual. El transmisor incorporado en el sensor convierte la señal de nivel medida en una señal de corriente y la envía al comparador.
2. Comparador: El comparador compara la medición actual de nivel de llenado con el valor de referencia deseado, de modo que la diferencia entre ellos es la señal de error. Si la señal de referencia es mayor que la señal medida, entonces la señal de error resultante será positiva y se deberá aumentar el nivel de llenado. En cambio, si la señal de referencia es menor, la señal de error resultante será negativa, lo que significa que se necesitará disminuir la altura de llenado. Esta señal se envía al controlador.
3. Controlador: El controlador toma la señal del comparador y genera una histéresis para controlar adecuadamente la válvula todo/nada. La histéresis se genera para evitar que la válvula se abra y cierre constantemente alrededor de un punto de referencia, introduciendo un margen en el controlador. Si el nivel

Figura 5.5: Pestaña 'Lazos de control'. Lazo de nivel en la línea fría

Al pulsar cada uno de estos botones, a excepción de 'Planta', muestra una ventana emergente con información acerca de estos componentes.

Estas ventanas emergentes se crean mediante una serie 'callback' llamados 'LazosButton1Pushed(app,event)', 'LazosButton2Pushed(app,event)', etc.

Por ejemplo, para la selección de lazo 'número 2' (lazo de nivel de la línea fría), el 'botón 2' corresponde con el actuador del lazo de nivel. Se utiliza la función creada en la interfaz anterior, *creaVentanaDeElemento(app)*, y se incorporan los datos apropiados para explicar la válvula que controla el nivel de llenado en la línea fría.



```
% Button pushed function: LazosButton2
function LazosButton2Pushed(app, event)
    switch app.lazoSeleccionado
    case 2
        titulo = 'Electroválvula EV-02';
        etiqueta = 'ELECTROVÁLVULA EV-02';
        foto = '../img/LF_valvulaD2.jpg';
        ficheroTablaCSV = 'LF_valvulaD2.csv';
        ficheroHTML = 'LF_valvulaD2.html';
        creaVentanaDeElemento(app, titulo, etiqueta, foto, ficheroTablaCSV, ficheroHTML);
```

Figura 5.6: Callback 'LazosButton2Pushed'

La ventana emergente al pinchar sobre el botón 'Actuador' en esta opción de lazo será la siguiente:



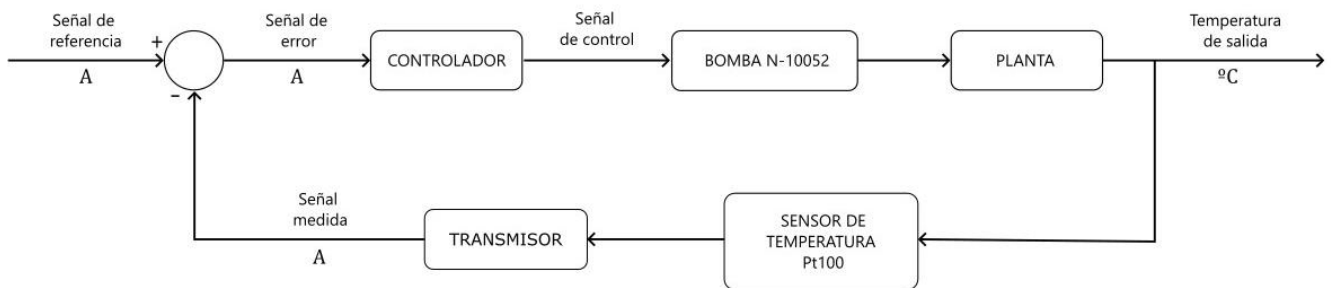
Figura 5.7: Ventana emergente 'LazosButton2Pushed'

## 6. MODELADO DE UN LAZO DE CONTROL

En este apartado se realizará el modelado de un lazo de control, con el fin de realizar un análisis de su control en una interfaz gráfica de usuario.

El lazo escogido consiste en el control de la temperatura de la línea de agua fría a la entrada de la camisa, previo al mezclado de ambas líneas. El control de la temperatura se realizará mediante la potencia eléctrica suministrada a la bomba N-10052, la cual modificará la velocidad del agua impulsada, y con ello su temperatura.

Para una definición más exhaustiva de este lazo, se dispone de toda la información en el punto 4.3 de esta memoria.



**Figura 6.1: Lazo de control de temperatura en la línea fría**

A continuación, se muestran los pasos a seguir para el modelado del sistema.

### 1. Modelo dinámico en el dominio del tiempo

Se comienza con un modelo matemático que describe las relaciones entre las variables dinámicas del sistema en el dominio del tiempo. Este modelo puede ser una ecuación diferencial o en diferencias, y representa cómo las variables evolucionan con el tiempo.

### 2. Identificación del punto de funcionamiento

Se elige un punto de operación alrededor del cual se realizará la linealización. Este punto debe ser un equilibrio del sistema, donde las derivadas de las variables son cero.

### 3. Linealización

Consiste en aproximar un sistema no lineal mediante un sistema lineal en las proximidades de un punto de funcionamiento. Esto se hace para trabajar en el dominio de Laplace, usando las funciones de transferencia, que sólo están definidas para sistemas lineales, o como en este caso, linealizados.

Para realizar la linealización, se deben calcular las derivadas parciales de las ecuaciones respecto a las variables del sistema y luego evaluarlas en el punto de operación.

#### 4. Transformada de Laplace

En este apartado se aplica la transformada de Laplace a las ecuaciones linealizadas en el dominio del tiempo. De este modo, se consigue transformar estas ecuaciones al dominio de la frecuencia, donde las ecuaciones son representadas en términos de la variable compleja 's'.

#### 5. Construcción del diagrama de bloques

A partir de las ecuaciones en el dominio de la frecuencia, se calcula la función de transferencia  $M(s)$  que describe la relación entre la entrada y la salida del sistema. La función de transferencia se expresa como  $M(s)=U(s)/Y(s)$ , donde  $Y(s)$  es la transformada de Laplace de la salida y  $U(s)$  es la transformada de Laplace de la entrada.

### 6.1. Ecuaciones dinámicas del sistema en el dominio del tiempo

#### 6.1.1. Sensor-transmisor de temperatura

Las variables y parámetros que caracterizan el sensor de temperatura son:

- $T_c(t)$ : Temperatura medida a la salida de la línea fría. Es la variable controlada del proceso, en °C.
- $I_{Tc}(t)$ : Señal resultante de la medición de la variable controlada. Es la señal de corriente resultante de medir la temperatura que se está controlando. Esta señal se compara con la señal de referencia para determinar el error, en A.
- $K_{s1}$ : Valor de la intensidad que se corresponde con la temperatura mínima medida.
- $K_{s2}$ : Ganancia del sensor de temperatura.

La relación entre la temperatura medida y la señal de intensidad proporcionada por el sensor-transmisor, en el dominio del tiempo es:

$$I_{Tc}(t) = K_{s1} + K_{s2} \cdot T_c(t)$$

En el siguiente apartado, se detallarán los valores de estas constantes, que están relacionadas con el protocolo Hart que utiliza el transmisor de temperatura.

#### 6.1.1.1. Modelo matemático del sensor-transmisor

Como se ha comentado anteriormente, el sensor de temperatura es un modelo Pt100 de tipo resistivo, RTD. Estos sensores consisten en una resistencia a la cual se le hace pasar una intensidad de corriente pequeña, alrededor de 1mA, dando como resultado una señal de voltaje proporcional a la temperatura.

Según la Ley de Ohm:

$$V = I \cdot R$$

Suponiendo una intensidad de 1mA, se obtiene que:

$$V = R$$

Es, decir, el valor de la resistencia en  $\Omega$ , corresponde al valor de voltaje en mV.

Se conoce que la resistencia resultante de los sensores Pt100 sigue una relación lineal con la temperatura medida, sobre todo cuando esta toma valores positivos.

$$R = 100 \cdot (1 + 0.00385 \cdot T) \quad \text{donde } R \text{ se mide en } \Omega, \text{ y } T \text{ en } ^\circ\text{C}.$$

En esta simulación, el rango de medición de temperatura comprende desde 0°C hasta 100°C, que es la temperatura en la que el agua cambia de estado a la presión atmosférica.

De esta relación, se deduce:

- La resistencia mínima,  $R_{\min}$ , resultante de medir la temperatura mínima,  $T_{\min} = 0^\circ\text{C}$ .

$$R_{\min} = 100 \cdot (1 + 0.00385 \cdot 0)$$

$$R_{\min} = 100 \Omega$$

- La resistencia máxima,  $R_{\max}$ , resultante de medir la temperatura máxima,  $T_{\max} = 100^\circ\text{C}$ .

$$R_{\max} = 100 \cdot (1 + 0.00385 \cdot 100)$$

$$R_{\max} = 138.5 \Omega$$

El transmisor se encarga de transformar la señal de voltaje resultante del sensor en una señal de corriente. Éste cumple el protocolo HART, que entrega una señal comprendida entre un rango de 4 a 20 mA, proporcional a la temperatura medida.

Recordando la relación entre la resistencia obtenida y la temperatura para este sensor:

V (mV)	R ( $\Omega$ )	T <sub>c</sub> (°C)	I (mA)
100	R <sub>min</sub> = 100	T <sub>cmin</sub> = 0 °C	I <sub>min</sub> = 4
V	R	T <sub>c</sub>	I
138.5	R <sub>máx</sub> = 138.5	T <sub>cmáx</sub> = 100 °C	I <sub>máx</sub> = 20

**Tabla 6.1: Interpolación 'Modelado matemático sensor-transmisor'**

Interpolando, se obtiene la siguiente señal de corriente enviada por el transmisor en función de la temperatura medida por el sensor.

$$I_{Tc}(t) = 4 + 0.16 \cdot T_c(t)$$

En esta expresión, I<sub>Tc</sub> se mide en mA y T<sub>c</sub> se mide en °C.

La pendiente de esta relación, K<sub>S2</sub>, es 0.16 mA/°C, y K<sub>S1</sub> es 4 mA.

### 6.1.2. Comparador

Las variables y parámetros que caracterizan al comparador son:

- I<sub>Tcref</sub>(t): Señal de referencia. Es la señal de corriente proporcional al cambio de temperatura que se obtiene en la salida, en A.
- e(t): Señal de error. El comparador genera una señal de error que consiste en la diferencia entre la señal de referencia y la señal de proceso, en A.
- I<sub>Tc</sub>(t): Señal resultante de la medición de la variable controlada. Es la señal de corriente resultante de medir la temperatura que se está controlando. Esta señal se compara con la señal de referencia para determinar el error, en A.

La relación entre las señales correspondientes al comparador, en el dominio del tiempo:

$$I_{Tcref}(t) - I_{Tc}(t) = e(t)$$

### 6.1.3. Controlador

Las variables y parámetros que caracterizan al controlador son:

- $i_c(t)$ : Señal de control obtenida por el controlador, en A.
- $e(t)$ : Señal de error.

La relación entre la señal de control y la señal de error en la planta sin controlador PID o con una acción proporcional,  $K_p=1$ .

$$i_c(t) = e(t)$$

Si se usara un controlador PID, es:

$$i_c(t) = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$$

Esta ecuación es aplicable incluso sin controlador, pues entonces  $K_p = 1$  y  $1/T_i = T_d = 0$ .

### 6.1.4. Variador de frecuencia

Las variables y parámetros que caracterizan al variador de frecuencia son:

- $f(t)$ : Frecuencia de la señal modulada que envía el variador de frecuencia al motor de la bomba, en Hz.
- $i_c(t)$ : Señal de control obtenida por el controlador, en A.
- $K_f$ : Ganancia del variador de frecuencia, en Hz/A.

La relación entre la frecuencia enviada a la bomba y la señal de control, en el dominio del tiempo:

$$f(t) = K_f \cdot i_c(t)$$

En esta relación la constante de proporcionalidad,  $K_f$ , junto con el resto de los parámetros del modelo, se determinarán más adelante al proponer el punto de funcionamiento.

### 6.1.5. Bomba hidráulica

Las variables y parámetros que caracterizan a la bomba hidráulica son:

- $w(t)$ : Velocidad angular del motor de la bomba N-10052, en rad/s.
- $f(t)$ : Frecuencia de la señal modulada que recibe el motor de la bomba, en Hz.
- $K_{b1}$ : Ganancia de la bomba, en (rad/s) /Hz.
- $K_{b2}$ : Ganancia de la bomba, en m<sup>3</sup>/rad.
- $q_c(t)$ : Caudal de agua impulsado por la bomba N-10052, en la línea de agua fría, en m<sup>3</sup>/s.
- $\tau_b$ : Constante de tiempo de la bomba, en s.

La relación entre la frecuencia enviada a la bomba y la velocidad angular del motor, en el dominio del tiempo:

$$w(t) = K_{b1} \cdot f(t)$$

La relación entre la velocidad angular del motor de la bomba y el caudal propulsado, en el dominio del tiempo:

$$q_c(t) = K_{b2} \cdot w(t)$$

#### 6.1.5.1. Modelo matemático de la bomba

El variador de frecuencia suministra una potencia eléctrica al motor de la bomba en forma de pulso, 'PWM'. Esta es una señal de voltaje utilizada para modificar la cantidad de energía enviada a la bomba. Así, se modifica la velocidad angular de la bomba,  $w(t)$ .

La relación entre la velocidad de rotación y el flujo que la bomba impulsa está determinada por la curva característica de la bomba.

En este análisis, se asume la operación alrededor de un punto de funcionamiento específico, y por tanto el sistema se simplifica mediante una linealización. En esta aproximación lineal, consideramos que el cambio de caudal impulsado por la bomba es proporcional al cambio en la velocidad de rotación. La constante de proporcionalidad, o ganancia de la bomba, se deriva de la curva de la bomba.

Al desconocer la curva característica de la bomba que impulsa el fluido a través de la línea fría, se aproxima su valor a la constante  $K_b$ , de valor reducido.

#### 6.1.6. Intercambiador de calor

Las variables y parámetros que caracterizan al intercambiador de calor son:

- $T_c(t)$ : Temperatura medida a la salida de la línea fría. Es la variable controlada del proceso, en  $^{\circ}\text{C}$ .
- $q_c(t)$ : Caudal de agua impulsado por la bomba N-10052, en la línea de agua fría, en  $\text{m}^3/\text{s}$ .
- $\rho$  : Densidad del agua en la línea fría, en  $\text{kg}/\text{m}^3$ .
- $C_p$  : Capacidad calorífica del agua en la línea fría, en  $\text{J}/\text{kg}\cdot^{\circ}\text{C}$ .
- $T_a$  : Temperatura del agua como fluido de trabajo al inicio del intercambio de calor con el refrigerante, en  $^{\circ}\text{C}$ .
- $T_L$  : Temperatura del refrigerante durante toda la operación de intercambio de calor, en  $^{\circ}\text{C}$ . Su temperatura no varía con en el intercambio ya que se encuentra en estado de líquido saturado, y el calor recibido por este fluido únicamente interviene en el cambio de estado de este.
- $U_c A_c$  : Inversa de la resistencia global a la transmisión del calor en el intercambiador, en  $\text{W}/^{\circ}\text{C}$ . Es el producto entre el coeficiente global de la transmisión de calor y el área total de intercambio de calor.
- $m_{vc}$  : Masa de agua en el volumen de control correspondiente al intercambio de calor con el refrigerante, en el intercambiador, en  $\text{kg}$ .

La relación entre la temperatura alcanzada por el fluido y las características del intercambiador de calor, en el dominio del tiempo:

$$m_{vc} \cdot C_p \cdot \frac{dT_c(t)}{dt} = \rho \cdot C_p \cdot q_c(t) \cdot (T_a - T_c(t)) - U_c \cdot A_c \cdot (T_c(t) - T_L)$$



### 6.1.7. Modelo matemático del intercambiador de calor

Se considera que el líquido refrigerante se encuentra saturado. De esta manera, la transmisión de calor en el intercambiador es mayor.

Cuando se cede calor a este líquido refrigerante, al encontrarse saturado, únicamente sufre un cambio de estado, de modo que su temperatura se mantiene constante. Debido a esto, no es necesario realizar un balance de energía para el refrigerante en el intercambiador de calor, y este balance se limita al volumen de control ocupado por la línea de agua.

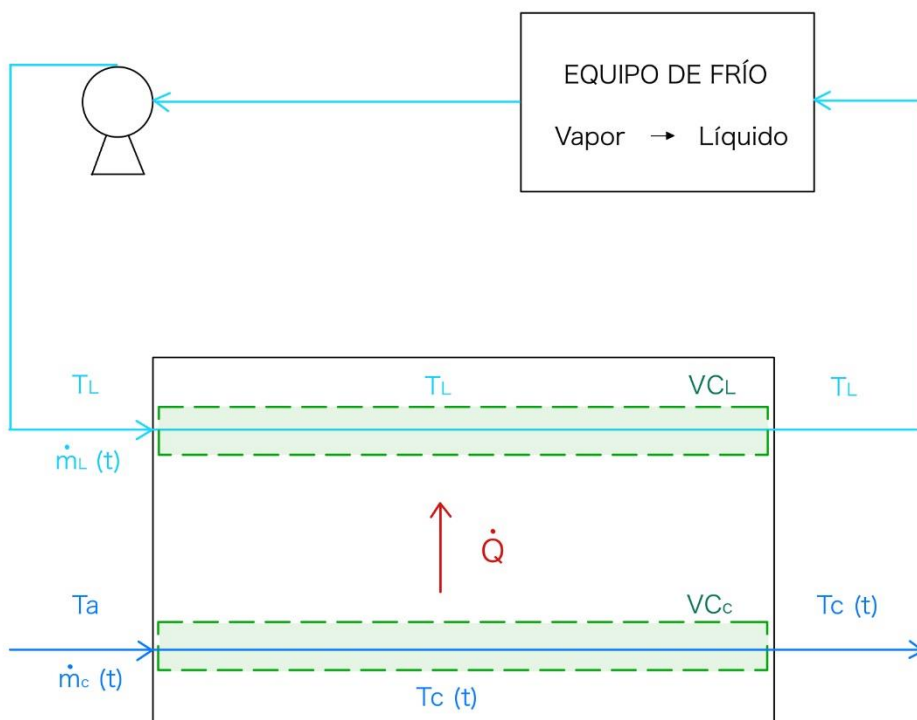


Figura 6.2: Esquema balance de energía en el intercambiador de calor

$VC_c$ : Volumen de control en el intercambiador de calor de la línea fría (fluido de trabajo a enfriar).

$VC_l$ : Volumen de control en el intercambiador de calor de la línea refrigerante.

$T_L$ : Temperatura del líquido refrigerante (constante durante todo el intercambio).

$T_a$ : Temperatura del agua en la línea fría, al inicio del intercambiador de calor.

$T_c(t)$ : Temperatura del agua en la línea fría en función del tiempo, que se considera la misma en todo el volumen de control.

A continuación, se desarrolla el balance de energía en el fluido de trabajo:

1. La energía del fluido de trabajo en el volumen de control seleccionado:

Esta energía depende de la energía interna,  $U_c(t)$ , de la energía cinética,  $E_k(t)$  y de la potencial,  $E_p(t)$ . Se desprecian los dos últimos términos con respecto al primero. Además, para líquidos incompresibles, se puede aproximar la energía interna por la entalpía. Se ha expresado la entalpía por el producto de la masa de fluido de trabajo en el volumen de control,  $m_{VC}$ , que es constante, por la entalpía específica de este a la temperatura  $T_c(t)$ .

$$U_c(t) + E_k(t) + E_p(t) \approx U_c(t) \approx H_c(t) = m_{VC} \cdot h_c(t)$$

La entalpía específica se puede expresar en función del calor específico a presión constante del fluido de trabajo con respecto a una cierta temperatura de referencia.

$$E_c(t) = m_{VC} \cdot \int_{T_{ref}}^{T_c(t)} C_p(t) dT = m_{VC} \cdot C_p \cdot (T_c(t) - T_{ref})$$

Si se considera un calor específico promedio constante, queda la expresión:

$$E_c(t) = m_{VC} \cdot C_p \cdot (T_c(t) - T_{ref})$$

2. La variación de energía del fluido de trabajo con respecto al tiempo en el volumen de control seleccionado:

$$\frac{dE_c(t)}{dt} = m_{VC} \cdot C_p \cdot \frac{d(T_c(t) - T_{ref})}{dt}$$

$$\boxed{\frac{dE_c(t)}{dt} = m_{VC} \cdot C_p \cdot \frac{dT_c(t)}{dt}}$$

3. Primer principio de la Termodinámica aplicado al sistema abierto que incluye el volumen de control, junto con la entrada y salida de fluido:

$$\frac{dE_c(t)}{dt} = \dot{Q}_c - \dot{W}_c + (\dot{m}_c)_e(t) \cdot \left( (h_c)_e(t) + \frac{(c_c)_e^2(t)}{2} + g \cdot (z_c)_e \right) - (\dot{m}_c)_s(t) \cdot \left( (h_c)_s(t) + \frac{(c_c)_s^2(t)}{2} + g \cdot (z_c)_s \right)$$

donde  $\dot{Q}_c$  y  $\dot{W}_c$  son, respectivamente, el flujo de calor ganado por el fluido en el volumen de control y el trabajo por unidad de tiempo que realiza el fluido (este término se desprecia). Las magnitudes  $(\dot{m}_c)_e(t)$  y  $(\dot{m}_c)_s(t)$  representan el flujo másico de fluido que entra al volumen de control y el que sale de él, respectivamente. La entalpía específica del fluido que entra y la del que sale son  $(h_c)_e(t)$  y  $(h_c)_s(t)$ . Además,  $(c_c)_e^2(t)/2$  y  $(c_c)_s^2(t)/2$  son las energías cinéticas específicas de entrada y salida,  $g \cdot (z_c)_e$  y  $g \cdot (z_c)_s$ , son las respectivas energías potenciales. Despreciando también los cambios de energía cinética y potencial, queda la siguiente expresión:

$$\frac{dE_c(t)}{dt} = (\dot{m}_c)_e(t) \cdot (h_c)_e(t) - (\dot{m}_c)_s(t) \cdot (h_c)_s(t) + \dot{Q}_c$$

Además, no se acumula fluido en el volumen de control:

$$\dot{m}_c(t) = (\dot{m}_c)_e(t) = (\dot{m}_c)_s(t)$$

$$\frac{dE_c(t)}{dt} = \dot{m}_c(t) \cdot [(h_c)_e(t) - (h_c)_s(t)] + \dot{Q}_c$$

El fluido de trabajo transfiere calor al refrigerante, por lo que se produce un flujo de calor  $\dot{Q}_c < 0$ .

Por comodidad, se desarrolla por separado la diferencia  $(h_c)_e(t) - (h_c)_s(t)$ :

$$(h_c)_e(t) - (h_c)_s(t) = C_p \cdot [(T_c)_e(t) - T_{ref}] - C_p \cdot [(T_c)_s(t) - T_{ref}]$$

$$(h_c)_e(t) - (h_c)_s(t) = C_p \cdot [(T_c)_e(t) - (T_c)_s(t)] = C_p \cdot (T_a - T_c(t))$$

Por otro lado, el flujo de calor que pierde el fluido en el volumen de control se puede escribir como  $\dot{Q}_c = -|\dot{Q}_c| < 0$ . Sustituyendo lo que se ha encontrado hasta ahora, el balance de energía en el VC queda:

$$m_{vc} \cdot C_p \cdot \frac{dT_c(t)}{dt} = \dot{m}_c(t) \cdot C_p \cdot (T_a - T_c(t)) - |\dot{Q}_c|$$

El flujo de calor se produce debido a la diferencia de temperatura, y como hemos supuesto que, el fluido de trabajo se encuentra a la misma temperatura en todo el volumen, y como  $T_c(t) > T_L$ :

$$|\dot{Q}_c| = U_c \cdot A_c \cdot (T_c(t) - T_L)$$

Donde  $U_c$  es el coeficiente global de transferencia de calor en el intercambiador y  $A_c$  es el área de transferencia de calor:

$$m_{vc} \cdot C_p \cdot \frac{dT_c(t)}{dt} = \dot{m}_c(t) \cdot C_{p_c} \cdot (T_a - T_c(t)) - U_c \cdot A_c \cdot (T_c(t) - T_L)$$

$$\frac{dT_c(t)}{dt} = \frac{\dot{m}_c(t) \cdot C_{p_c}}{m_{vc} \cdot C_p} \cdot (T_a - T_c(t)) - \frac{U_c \cdot A_c}{m_{vc} \cdot C_p} \cdot (T_c(t) - T_L)$$

Como el flujo másico  $\dot{m}_c(t) = \rho \cdot q_c(t)$ , siendo  $\rho$  la densidad del fluido supuesto incompresible, y  $q_c(t)$  su caudal volumétrico.

$$\frac{dT_c(t)}{dt} = \frac{\rho \cdot C_{p_c}}{m_{vc} \cdot C_p} \cdot q_c(t) \cdot (T_a - T_c(t)) - \frac{U_c \cdot A_c}{m_{vc} \cdot C_p} \cdot (T_c(t) - T_L)$$

## 6.2. Linealización de las ecuaciones dinámicas del sistema en torno a un punto de funcionamiento

### 6.2.1. Sensor-transmisor de temperatura

Partiendo de la ecuación resultante en el dominio del tiempo:

$$I_{T_c}(t) = K_{s1} + K_{s2} \cdot T_c(t)$$

Se procede a su linealización en torno a un punto de funcionamiento:

$$\Delta I_{Tc}(t) = \left[ \frac{d}{dT_c(t)} (0.004 + K_s \cdot T_c(t)) \right]_0 \cdot \Delta T_c(t)$$

$$\Delta I_{Tc}(t) = K_s \cdot \Delta T_c(t)$$

### 6.2.2. Comparador

Partiendo de la ecuación resultante en el dominio del tiempo:

$$I_{Tcref}(t) - I_{Tc}(t) = e(t)$$

Se procede a su linealización en torno a un punto de funcionamiento:

$$\Delta I_{Tcref}(t) - \Delta I_{Tc}(t) = \Delta e(t)$$

### 6.2.3. Controlador

Se linealiza la ecuación resultante en el dominio del tiempo:

$$i_c(t) = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$$

Si no se tiene en cuenta el controlador  $K_p = 1$  y  $1/T_i = T_d = 0$ , de manera que:

$$\Delta i_c(t) = \Delta e(t)$$

En caso contrario, se tiene:

$$\Delta i_c(t) = K_p \left( \Delta e(t) + \frac{1}{T_i} \int \Delta e(t) dt + T_d \frac{d\Delta e(t)}{dt} \right)$$

#### 6.2.4. Variador de frecuencia

Partiendo de la ecuación resultante en el dominio del tiempo:

$$f(t) = K_f \cdot i_c(t)$$

Se procede a su linealización en torno a un punto de funcionamiento:

$$\Delta f(t) = K_f \cdot \Delta i_c(t)$$

#### 6.2.5. Bomba hidráulica

Partiendo de las ecuaciones resultantes en el dominio del tiempo:

$$w(t) = K_{b1} \cdot f(t)$$

$$q_c(t) = K_{b2} \cdot w(t)$$

Se linealizan en torno a un punto de funcionamiento:

$$\Delta w(t) = K_{b1} \cdot \Delta f(t)$$

$$\Delta q_c(t) = K_b \cdot \Delta w(t)$$

#### 6.2.6. Intercambiador de calor

Partiendo de la ecuación resultante en el dominio del tiempo:

$$m_{VC} \cdot C_p \cdot \frac{dT_c(t)}{dt} = \rho \cdot C_p \cdot q_c(t) \cdot (T_a - T_c(t)) - U_c \cdot A_c \cdot (T_c(t) - T_L)$$

Se procede a su linealización en torno a un punto de funcionamiento:

$$m_{VC} \cdot C_p \cdot \frac{d\Delta T_c(t)}{dt} = \rho \cdot C_p \cdot \left\{ \frac{d}{dq_c(t)} [q_c(t) \cdot (T_a - T_c(t))] \right\}_0 \cdot \Delta q_c(t) + \\ + \rho \cdot C_p \cdot \left\{ \frac{d}{dT_c(t)} [q_c(t) \cdot (T_a - T_c(t))] \right\}_0 \cdot \Delta T_c(t) - U_c \cdot A_c \cdot \Delta T_c(t)$$

$$m_{VC} \cdot C_p \cdot \frac{d\Delta T_c(t)}{dt} = \rho \cdot C_p \cdot \{T_a - T_c(t)\}_0 \cdot \Delta q_c(t) + \\ + \rho \cdot C_p \cdot \{-q_c(t)\}_0 \cdot \Delta T_c(t) - U_c \cdot A_c \cdot \Delta T_c(t)$$

$$m_{VC} \cdot C_p \cdot \frac{d\Delta T_c(t)}{dt} = \rho \cdot C_p \cdot (T_a - T_{c0}) \cdot \Delta q_c(t) - \rho \cdot C_p \cdot q_{c0} \cdot \Delta T_c(t) - \\ - U_c \cdot A_c \cdot \Delta T_c(t)$$

### 6.3. Ecuaciones dinámicas del sistema en el dominio de la frecuencia

#### 6.3.1. Sensor-transmisor de temperatura

Partiendo de la ecuación linealizada alrededor del punto de funcionamiento:

$$\Delta I_{Tc}(t) = K_s \cdot \Delta T_c(t)$$

Se realiza la transformada de Laplace:

$$\Delta I_{Tc}(s) = K_s \cdot \Delta T_c(s)$$

#### 6.3.2. Comparador

Partiendo de la ecuación linealizada alrededor del punto de funcionamiento:

$$\Delta I_{Tcref}(t) - \Delta I_{Tc}(t) = \Delta e(t)$$

Se realiza la transformada de Laplace:

$$\Delta I_{Tcref}(s) - \Delta I_{Tc}(s) = \Delta e(s)$$

### 6.3.3. Controlador

Se parte de la ecuación linealizada alrededor del punto de funcionamiento, considerando el proceso sin regulador y con un controlador PID:

$$\Delta i_c(t) = \Delta e(t)$$

$$\Delta i_c(t) = K_p \left( \Delta e(t) + \frac{1}{T_i} \int \Delta e(t) dt + T_d \frac{d\Delta e(t)}{dt} \right)$$

Y se realiza la transformada de Laplace:

$$\Delta i_c(s) = \Delta e(s)$$

o bien:

$$\Delta i_c(s) = \Delta e(s) \cdot K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

En esta última expresión se puede tomar  $K_p = 1$  y  $1/T_i = T_d = 0$ , en caso de no utilizar controlador PID.

### 6.3.4. Variador de frecuencia

Partiendo de la ecuación linealizada alrededor del punto de funcionamiento:

$$f(t) = K_f \cdot i_c(t)$$

Se realiza la transformada de Laplace:

$$\Delta f(s) = K_f \cdot \Delta i_c(s)$$



### 6.3.5. Bomba hidráulica

Se parte de las ecuaciones linealizadas alrededor del punto de funcionamiento:

$$\Delta w(t) = K_{b1} \cdot \Delta f(t)$$

$$\Delta q_c(t) = K_b \cdot \Delta w(t)$$

Y se realiza la transformada de Laplace:

$$\Delta w(s) = K_{b1} \cdot \Delta f(s)$$

$$\Delta q_c(s) = K_b \cdot \Delta w(s)$$

### 6.3.6. Intercambiador de calor

Partiendo de la ecuación linealizada alrededor del punto de funcionamiento:

$$m_{VC} \cdot C_p \cdot \frac{d\Delta T_c(t)}{dt} = \rho \cdot C_p \cdot (T_a - T_{c0}) \cdot \Delta q_c(t) - \rho \cdot C_p \cdot q_{c0} \cdot \Delta T_c(t) - U_c \cdot A_c \cdot \Delta T_c(t)$$

Se realiza la transformada de Laplace:

$$m_{VC} \cdot C_p \cdot s \cdot T_c(s) = \rho \cdot C_p \cdot (T_a - T_{c0}) \cdot q_c(s) - \rho \cdot C_p \cdot q_{c0} \cdot T_c(s) - U_c \cdot A_c \cdot T_c(s)$$

$$(m_{VC} \cdot C_p \cdot s + \rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c) \cdot T_c(s) = \rho \cdot C_p \cdot (T_a - T_{c0}) \cdot q_c(s)$$

$$[m_{VC} \cdot C_p \cdot s + (\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c)] \cdot T_c(s) = \rho \cdot C_p \cdot (T_a - T_{c0}) \cdot q_c(s)$$

$$\left( \frac{m_{VC} \cdot C_p}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} \cdot s + 1 \right) \cdot (\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c) \cdot T_c(s) = \rho \cdot C_p \cdot (T_a - T_{c0}) \cdot q_c(s)$$

$$T_c(s) = \frac{\rho \cdot C_p \cdot (T_a - T_{c0}) \cdot q_c(s)}{\left( \frac{m_{VC} \cdot C_p}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} \cdot s + 1 \right) \cdot (\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c)}$$

$$T_c(s) = \frac{\rho \cdot C_p \cdot (T_a - T_{c0})}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} \cdot \frac{q_c(s)}{\frac{m_{VC} \cdot C_p}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} \cdot s + 1}$$

$$T_c(s) = \frac{K_c \cdot q_c(s)}{\tau_c \cdot s + 1}$$

En esta expresión, los parámetros  $K_c$  y  $\tau_c$  son:

$$K_c = \frac{\rho \cdot C_p \cdot (T_a - T_{c0})}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} \equiv \text{Ganancia del intercambiador de calor}$$

$$\tau_c = \frac{m_{VC} \cdot C_p}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} \equiv \text{Constante de tiempo en el intercambiador de calor}$$

#### 6.4. Parámetros del sistema y punto de funcionamiento.

Se reescriben las ecuaciones en el dominio del tiempo para un punto de funcionamiento en la planta sin controlador PID, por tanto, en la ecuación 3 no aparece la acción proporcional:

$$1) I_{Tc0} = K_{s1} + K_{s2} \cdot T_{c0}$$

$$2) I_{Tcref0} - I_{Tc0} = e_0$$

$$3) i_{c0} = e_0$$

$$4) f_0 = K_f \cdot i_{c0}$$

$$5) w_0 = K_{b1} \cdot f_0$$

$$6) q_{c0} = K_{b2} \cdot w_0$$

$$7) \rho \cdot C_p \cdot q_{c0} \cdot (T_a - T_{c0}) - U_c \cdot A_c \cdot (T_{c0} - T_L) = 0$$

Se conoce el valor de una serie de parámetros:

- Propiedades físicas

El fluido de trabajo que circula a través de la línea fría asociada al lazo de control seleccionado es agua. Por consiguiente, se considerarán los valores de densidad y capacidad calorífica específica del agua promedio. Esto es, una densidad,  $\rho$ , de 1000 kg/m<sup>3</sup>, y calor específico,  $C_p$ , de 4181 J/kg·°C.

- Sensor/Transmisor de temperatura

Tal y como se desarrolló en el apartado 6.1.1.1, se trata de un sensor de tipo resistivo y el transmisor asociado sigue el protocolo Hart.

Dado que la temperatura del agua podría variar en el rango de 0 a 100°C, generando una señal de intensidad proporcional de 4 a 20 mA, se determinan las constantes  $K_{s1}$ , de 4 mA y  $K_{s2}$ , de 0.16 mA/°C.

- Temperaturas

Tanto la temperatura del fluido refrigerante y la temperatura del agua de trabajo inicialmente se suponen conocidas.

En el intercambiador de calor, el fluido refrigerante se encuentra a una temperatura constante,  $T_L$ , de -2°C y el agua se encuentra inicialmente a una temperatura,  $T_a$ , de 20°C. El refrigerante utilizado en la planta real es una mezcla de agua con glicol, que en este caso actúa como anticongelante. En este modelado se supone un fluido refrigerante como el R-134a, que a una temperatura de -2°C y una presión de 2.7278 bar, se encuentra en estado de líquido saturado [SHA].

En principio, se trata de un sistema de 7 ecuaciones y 12 incógnitas.

Con el fin de poder resolver el sistema de ecuaciones descrito, se suponen conocidos los siguientes valores en el punto de funcionamiento:

- Temperatura,  $T_{c0}$ :

$$T_{c0} = 10^{\circ}\text{C}$$

La temperatura en el punto de funcionamiento de la línea fría debe encontrarse dentro del rango de:  $T_L < T_{c0} < T_a$ .

Puesto que el agua inicialmente se encuentra a una temperatura,  $T_a$ , y sufre un enfriamiento tras su paso por la bomba y el intercambiador de calor, la temperatura en el punto de funcionamiento debe ser inferior a esta. Por otro lado, el fluido refrigerante se encuentra a una temperatura,  $T_L$ , que ha de ser menor que la temperatura que alcance el agua a refrigerar.

- Caudal volumétrico,  $q_{c0}$ :

$$q_{c0} = 3 \frac{\text{L}}{\text{min}} = 5 \cdot 10^{-5} \frac{\text{m}^3}{\text{s}}$$

Se considera un valor apropiado para unas tuberías de sección pequeña, como las que se encuentran en la planta.

Como el objetivo de las líneas de agua fría y caliente es enfriar el reactor tras la reacción exotérmica simulada por la resistencia, y la línea caliente está limitada a un máximo de 1.8 L/min, como se describió en el apartado 3.3.4.2, se considera adecuado un caudal de agua en la línea fría en el punto de funcionamiento de 3 L/min.

- Frecuencia enviada a la bomba,  $f_0$ :

$$f_0 = 50 \text{ Hz}$$

- Velocidad angular del motor de la bomba,  $w_0$ :

$$w_0 = 2900 \text{ rpm} = \frac{290\pi}{3} \frac{\text{rad}}{\text{s}} = 303.6873 \frac{\text{rad}}{\text{s}}$$

Tanto la frecuencia y la velocidad angular seleccionadas para el punto de funcionamiento coinciden con las detalladas en las especificaciones del fabricante, referenciadas en la sección 3.3.4.2.

- Ganancia del variador de frecuencia,  $K_f$ :

$$K_f = 10 \frac{\text{Hz}}{\text{mA}} = 10000 \frac{\text{Hz}}{\text{A}}$$

Para hallar la ganancia, se divide el máximo valor de frecuencia que puede generar el variador entre la máxima corriente que puede recibir.

$$K_f = \frac{200 \text{ Hz}}{20 \text{ mA}}$$

Estos datos de frecuencia y corriente se encuentran referenciados en la sección 3.3.4.3.

Todos estos valores supuestos son razonables para establecer un punto de funcionamiento según las características de la planta.

Una vez establecidos estos valores, el sistema de ecuaciones queda:

$$1) I_{Tc0} = 0.004 + 0.00016 \cdot 10$$

$$2) I_{Tcref0} - I_{Tc0} = e_0$$

$$3) i_{c0} = e_0$$

$$4) 50 = 10000 \cdot i_{c0}$$

$$5) \frac{290\pi}{3} = K_{b1} \cdot 50$$

$$6) 5 \cdot 10^{-5} = K_{b2} \cdot \frac{290\pi}{3}$$

$$7) 1000 \cdot 4181 \cdot 5 \cdot 10^{-5} \cdot (15 - 10) - U_c \cdot A_c \cdot (10 - (-2)) = 0$$

Se despejan las incógnitas restantes del sistema de ecuaciones anterior:

$$1) I_{Tc0} = 0.004 + 0.00016 \cdot 10 \rightarrow I_{Tc0} = 0.0056 \text{ A} = 5.6 \text{ mA}$$

$$4) 50 = 10000 \cdot i_{c0} \rightarrow i_{c0} = 0.005 \text{ A} = 5 \text{ mA}$$

$$3) e_0 = 0.005 \text{ A} = 5 \text{ mA}$$

$$2) I_{Tcref0} - 0.0056 = 0.005 \rightarrow I_{Tcref0} = 0.0106 \text{ A} = 10.6 \text{ mA}$$

$$5) \frac{290\pi}{3} = K_{b1} \cdot 50 \rightarrow K_{b1} = \frac{29\pi}{15} \frac{\text{rad/s}}{\text{Hz}} = 6.0737 \frac{\text{rad/s}}{\text{Hz}} = 58 \frac{\text{rpm}}{\text{Hz}}$$

Este valor de  $K_{b1}$ , coincide aproximadamente con el de la fórmula general que relaciona la velocidad de giro de la bomba con la frecuencia que recibe y el número de polos de la bomba. Esta fórmula general es:

$$w = \frac{120 \cdot f}{P}, \text{ donde:}$$

$w$ : Velocidad de giro del motor, en rpm.

$f$ : Frecuencia recibida por el motor, en Hz.

$P$ : Número de polos del motor.

La bomba N-10052 tiene dos polos según las especificaciones que se encuentran en la sección 3.3.4.2.

$$w_0 = \frac{120 \cdot f_0}{P} = \frac{120 \cdot f_0}{2} = 60 \cdot f_0$$

Se aprecia como la ganancia resultante,  $K_{b1} = 58 \frac{\text{rpm}}{\text{Hz}} \approx 60 \frac{\text{rpm}}{\text{Hz}}$

Al ser esta una ecuación general para cualquier motor de este tipo, es más fiable el valor obtenido a partir de los datos del fabricante.

$$6) 5 \cdot 10^{-5} = K_{b2} \cdot \frac{290\pi}{3} \rightarrow K_{b2} = 164.6430 \cdot 10^{-9} \frac{\text{m}^3}{\text{rad}} = 1.03448 \frac{\text{mL}}{\text{rev}}$$

La relación entre el flujo volumétrico proporcionado por la bomba y la velocidad de giro del motor viene dada por la siguiente expresión:

$$Q = \frac{D \cdot n}{1000}, \text{ donde:}$$

$Q$ : Se expresa en L/min.

$n$ : Se expresa en rpm.

$D$ : Se expresa en mL/rev.

Se despeja  $D$ :

$$D = \frac{1000 \cdot Q}{n} = \frac{1000 \cdot 3}{2900} \rightarrow D = 1.03448 \frac{\text{mL}}{\text{rev}}$$

Este resultado, coincide con el valor de  $K_{b2}$ , calculado anteriormente. Esto verifica la veracidad de los resultados obtenidos. [TET]

$$7) U_c \cdot A_c = \frac{1000 \cdot 4181 \cdot 5 \cdot 10^{-5} \cdot (15-10)}{10-(-2)} \rightarrow U_c \cdot A_c = 87.1042 \frac{W}{^{\circ}C}$$

Asimismo, establecemos valores para el resto de los parámetros involucrados en las ecuaciones en el dominio de la frecuencia descritas anteriormente:

- Intercambiador de calor

Con el fin de representar con mayor precisión la respuesta temporal del intercambiador de calor, se introduce una constante de tiempo,  $\tau_c$ , de 10 segundos.

Esta constante de tiempo se incluye a pesar de no tener ninguna referencia experimental sobre su valor. Como se trata de un intercambiador de calor de dimensiones reducidas, no se espera una masa de agua de trabajo intercambiando calor con el refrigerante muy elevada. Con este valor de  $\tau_c$ , se obtiene una masa de:

$$m_{VC} = \frac{\tau_c \cdot (\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c)}{C_p} = \frac{10 \cdot (1000 \cdot 4181 \cdot 5 \cdot 10^{-5} + 87.1042)}{4181} \rightarrow m_{VC} = 0.7083 \text{ kg}$$

Por otro lado, la ganancia en el intercambiador de calor queda:

$$K_c = \frac{\rho \cdot C_p \cdot (T_a - T_{c0})}{\rho \cdot C_p \cdot q_{c0} + U_c \cdot A_c} = \frac{1000 \cdot 4181 \cdot (15 - 10)}{1000 \cdot 4181 \cdot 5 \cdot 10^{-5} + 87.1042} \rightarrow$$

$$K_c = 70588.2274 \frac{^{\circ}C}{m^3/s} = 1.17647 \frac{^{\circ}C}{L/min}$$

Estas ecuaciones se encuentran descritas en el apartado 6.3.6.

Adicionalmente, se establece un valor apropiado para el área de intercambio de calor en el intercambiador,  $A_c$ , con el único objetivo de comprobar si realmente se obtienen valores lógicos para el coeficiente de intercambio de calor,  $U_c$ . Esta suposición es viable puesto que se conoce las especificaciones de área para distintos modelos de intercambio de calor del fabricante en el documento 'HXR Water-cooled heat exchanger', incluido en la sección 3.3.4.7.

Se desconoce el modelo exacto, pero dado que se trata de un modelo de pequeñas dimensiones, se considera para este modelado un área de intercambio de calor,  $A_c$ , de 3.9 sq ft, equivalente a 0.362322 m<sup>2</sup>. Esta área seleccionada corresponde con el modelo LHXR-0625-SCS.

$$3.9 \text{ ft}^2 \cdot \frac{12^2 \text{ in}^2}{1 \text{ ft}^2} \cdot \frac{2.54^2 \text{ cm}^2}{1 \text{ in}^2} \cdot \frac{1 \text{ m}^2}{10^4 \text{ cm}^2} = 0.362322 \text{ m}^2$$

De este modo, el coeficiente global transferencia de calor en el intercambiador resultante,  $U_c$ , es de  $223.8456 \text{ W/m}^2 \cdot ^\circ\text{C}$ . Este coeficiente implica un valor plausible para el intercambiador de calor que se encuentra en la planta.

A continuación, se muestran los valores tabulados de las variables en el punto de funcionamiento y de los parámetros implicados en el lazo de control.

Variable en el PF	Valor	Unidades
$I_{T_{\text{ref}0}}$	0.0106	A
$I_{T_{c0}}$	0.0056	A
$e_0$	0.005	A
$i_{c0}$	0.005	A
$T_{c0}$	10	$^\circ\text{C}$
$q_{c0}$	$5 \cdot 10^{-5}$	$\text{m}^3/\text{s}$
$f_0$	50	Hz
$w_0$	303.6873	rad/s

Tabla 6.2: Variables en el punto de funcionamiento

Parámetro	Valor	Unidades
$\rho$	1000	$\text{Kg}/\text{m}^3$
$C_p$	1481	$\text{J}/\text{kg} \cdot ^\circ\text{C}$
$T_a$	15	$^\circ\text{C}$
$T_L$	-2	$^\circ\text{C}$
$U_c A_c$	87.1042	$\text{W}/^\circ\text{C}$
$m_{VC}$	0.7083	kg
$K_{s1}$	0.004	A
$K_{s2}$	0.00016	$\text{A}/^\circ\text{C}$
$K_f$	10000	$\text{Hz}/\text{A}$
$K_{b1}$	6.0737	$(\text{rad}/\text{s})/\text{Hz}$
$K_{b2}$	$164.6430 \cdot 10^{-9}$	$\text{m}^3/\text{rad}$
$K_c$	70588.2274	$^\circ\text{C} \cdot \text{s}/\text{m}^3$
$\tau_c$	10	s

Tabla 6.3: Parámetros



### 6.5. Construcción del diagrama de bloques en Simulink

Finalmente, a partir de las ecuaciones obtenidas en el dominio de la frecuencia, se construye el diagrama de bloques en Simulink.

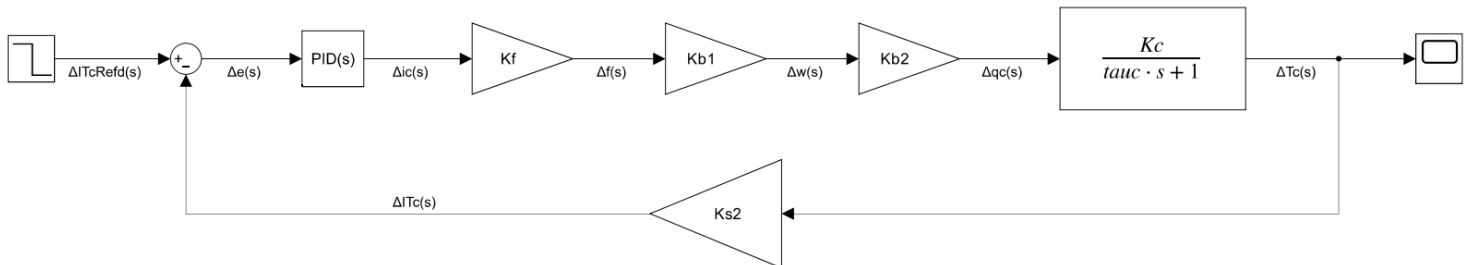


Figura 6.3: Lazo de control Simulink

Las variables que aparecen en el modelado del dominio de Laplace son incrementales, lo que significa que se consideran condiciones iniciales nulas. De modo que, la salida de este modelo refleja cambios con respecto al punto de funcionamiento.

Suponiendo que se quiere cambiar ligeramente la temperatura a un valor,  $T_{cd}$ , de  $6^{\circ}\text{C}$ , se debe aplicar un escalón correspondiente a este 'setpoint',  $\Delta I_{Tcrefd}(s)$ .

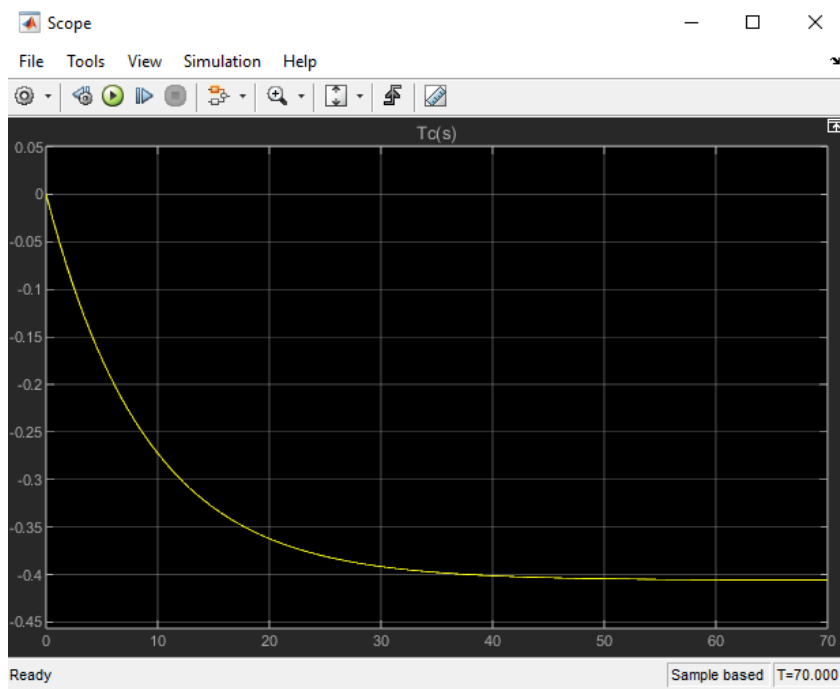
$$\Delta I_{Tcrefd}(s) = K_{s2} \cdot (T_{cd} - T_{c0})$$

$$\Delta I_{Tcrefd}(s) = 0.00016 \cdot (6 - 10)$$

$$\Delta I_{Tcrefd}(s) = -0.00064 \text{ A} = -0.64 \text{ mA}$$

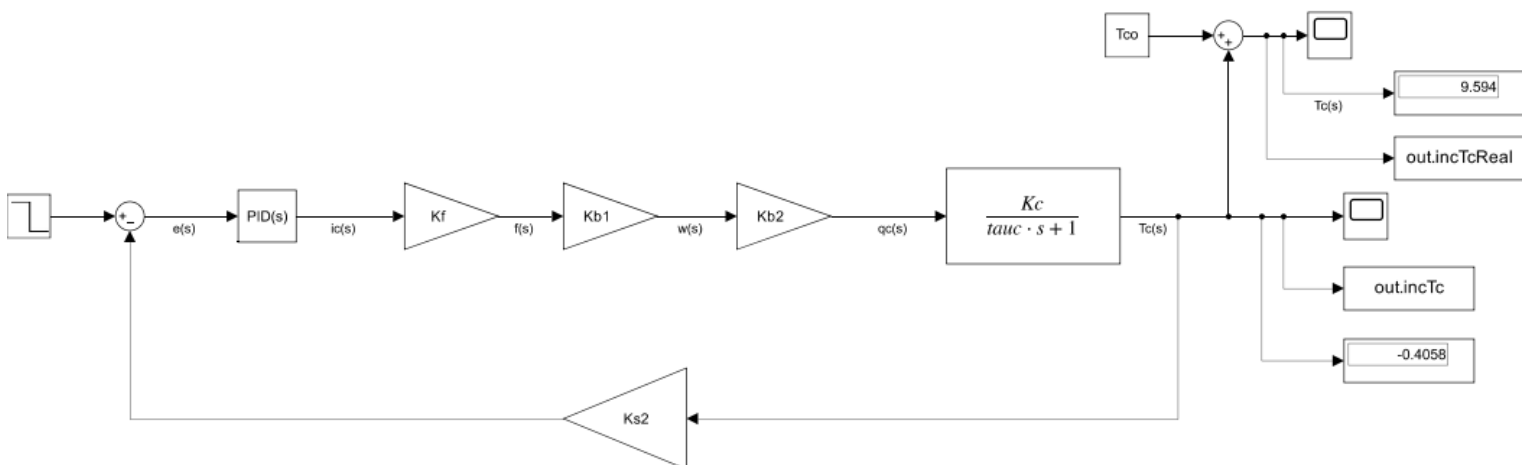
El resultado de la temperatura de salida tras aplicar este escalón es:

## Desarrollo de una interfaz gráfica de usuario para planta piloto

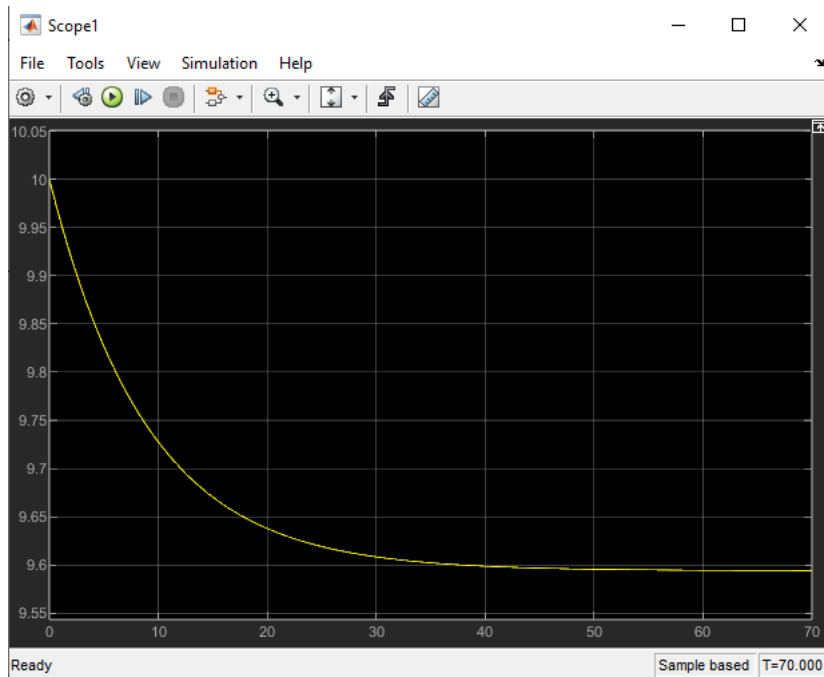


**Gráfica 1: Salida de incremento de temperatura**

Con el fin de visualizar la temperatura final que alcanza realmente el agua en este sistema, se añade un punto de suma al que se incorpora la temperatura en el punto de funcionamiento,  $T_{c0}$ . Así, se observa cómo la temperatura del agua a la salida del lazo de control es de aproximadamente  $9.59\text{ }^{\circ}\text{C}$ , la cual se encuentra por encima del valor deseado propuesto,  $T_{cd}$ , de  $6^{\circ}\text{C}$ .



**Figura 6.4: Lazo de control Simulink utilizado en la interfaz**



Gráfica 2: Salida de temperatura

El error de posición en régimen permanente obtenido es muy elevado. Calculando este error a partir de los datos obtenidos en la gráfica, resulta aproximadamente:

$$e_{prp} = \frac{\text{error cometido (}^{\circ}\text{C)}}{\text{cambio de temperatura deseado (}^{\circ}\text{C)}} \cdot 100 \approx \frac{-4 - (-0.4059)}{-4} \cdot 100$$

$$e_{prp} \approx 89.85\%$$

Ahora, calculando este error a partir de la función de transferencia en lazo abierto:

$$e_{prp} = \frac{1}{1 + \lim_{s \rightarrow 0} G(s) \cdot H(s)} = \frac{1}{1 + K_p}$$

$$G(s) \cdot H(s) = \frac{K_{s2} \cdot K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c}{\tau_c \cdot s + 1}$$

$$K_p = \lim_{s \rightarrow 0} G(s) \cdot H(s) = K_{s2} \cdot K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c$$

$$K_p = 0.00016 \cdot 10000 \cdot 6.0737 \cdot 164.643 \cdot 10^{-9} \cdot 70588.2274$$

$$K_p = 0.11294$$

$$e_{prp} = \frac{1}{1 + K_p} \cdot 100 = \frac{1}{1 + 0.11294} \cdot 100$$

$$e_{prp} = 89.85\%$$

Adicionalmente, para corregir los resultados de temperatura en la planta, se añade un controlador PID. Se plantea realizar el control de la planta más adelante desde la interfaz que se programará en los siguientes apartados.

En este modelo Simulink, se establece una acción proporcional,  $P=1$ , una integral,  $I=0$ , y una derivativa,  $D=0$ , con el fin de no tener en cuenta este elemento.

## 7. Análisis del sistema resultante

### 7.1. Función de transferencia

En este apartado se obtendrá una expresión para la función de transferencia del sistema en lazo cerrado y se analizarán los parámetros que la componen.

En primer lugar, se simplifica el diagrama de bloques anterior para analizar el sistema.

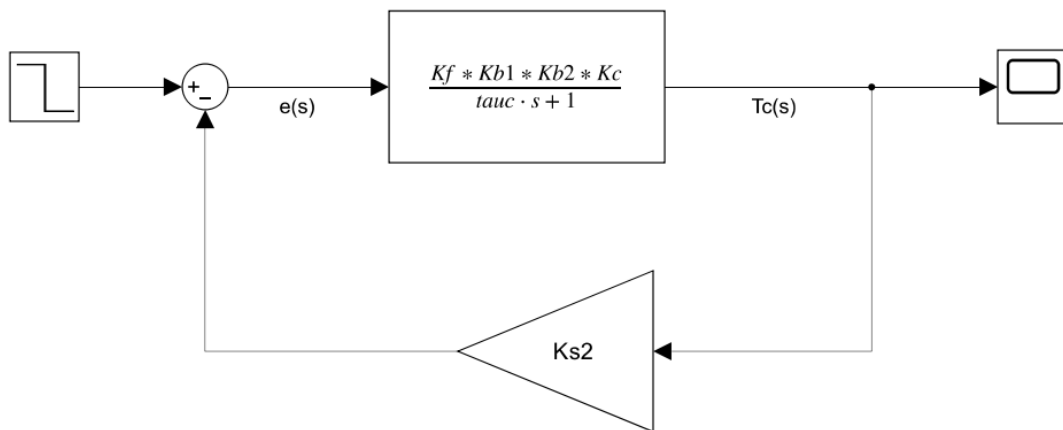


Figura 7.1: Lazo de control Simulink simplificado

$$M(s) = \frac{\frac{K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c}{\tau_c \cdot s + 1}}{1 + \frac{K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c \cdot K_{s2}}{\tau_c \cdot s + 1}}$$

$$M(s) = \frac{\frac{K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c}{\tau_c \cdot s + 1}}{\frac{\tau_c \cdot s + 1 + K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c \cdot K_{s2}}{\tau_c \cdot s + 1}}$$

Así, se obtiene función de transferencia siguiente:

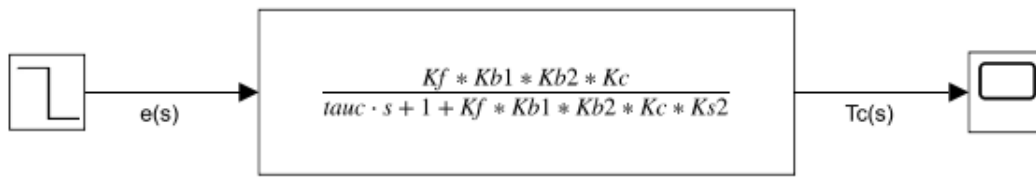


Figura 7.2: Función de transferencia del sistema,  $M(s)$

$$M(s) = \frac{K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c}{\tau_c \cdot s + 1 + K_f \cdot K_{b1} \cdot K_{b2} \cdot K_c \cdot K_{s2}}$$

$$M(s) = \frac{705.8768}{10 \cdot s + 1.1129}$$

Como se trata de una función de transferencia de primer orden, la expresión puede tomar la forma:

$$M(s) = \frac{705.8768}{10 \cdot \left( s + \frac{1.1129}{10} \right)}$$

$$M(s) = \frac{70.58768}{s + 0.11129}$$

La ganancia del sistema,  $K_M$ :

$$K_M = \frac{70.58768}{0.11129} \rightarrow K_M = 634.2679$$

Este sistema tiene un polo en  $s = -0.11129$

Al encontrarse en el semieje real negativo, se trata de un sistema estable.

## 8. GUI EN APP DESIGNER DEL LAZO DE CONTROL

En la cuarta pestaña de la interfaz, se desarrolla un modelo de un posible lazo de control en la planta piloto.

El lazo elegido para este análisis es el de control de temperatura en la línea fría partiendo de la salida del depósito. La descripción de este lazo de control se encuentra desarrollado en el apartado 4.3.

En esta interfaz se encuentran distintos paneles de información:

### ➤ Panel 'ENTRADAS':

En este panel se encuentran los siguientes componentes:

- Campo de edición que muestra la temperatura en el punto de funcionamiento elegido.
- Campo de edición que muestra el 'setpoint' correspondiente a la temperatura deseada.
- Campo de texto, no editable, que muestra la entrada escalón para que la temperatura en el punto de funcionamiento cambie a la del 'setpoint', en mA.
- Campo de edición que muestra la sobreoscilación máxima solicitada por el usuario. Si la simulación sobrepasa este valor, responde cambiando el color de fondo del campo de edición donde se encuentra la sobreoscilación real a rojo. En caso contrario, el color de fondo torna a verde.
- Campo de edición que muestra el tiempo máximo de simulación sugerido por el usuario. Si el usuario introduce un '0' implica que no se elige ningún tiempo de simulación concreto. Será el programa el que incorpore un tiempo aceptable, si se puede.
- Campo de edición que muestra el tiempo límite automático. Este tiempo límite se tendrá en cuenta siempre y cuando el usuario no determine un tiempo de simulación mediante el campo de edición anterior.
- Una casilla de verificación que permite transferir los valores de los parámetros implicados en el sistema al espacio de trabajo, 'Base Workspace', en Matlab.
- Un conjunto de botones proporciona al usuario la capacidad de especificar las condiciones bajo las cuales se desea generar un gráfico de simulación. Si la opción 'Planta' es seleccionada, se generará un gráfico que representa el sistema sin controlador. En cambio, al elegir la opción 'PID', se presentará un gráfico que incorpora un controlador PID al sistema simulado. Finalmente, al pulsar el botón 'Planta+PID', se mostrará una superposición de ambas representaciones gráficas. Para estos dos últimos escenarios, se habilitarán campos de edición adicionales que permitirán al usuario ajustar los diversos parámetros del controlador.
- Un interruptor se utiliza para activar la representación gráfica y análisis de los datos de la simulación después de que el usuario haya seleccionado los valores iniciales mediante los campos de edición en la interfaz.

- Para informar sobre el estado de la representación gráfica, se puede observar una lámpara que tomará los siguientes colores:
  - Rojo: no se ha realizado la representación gráfica de la simulación.
  - Verde: se ha realizado con éxito la representación gráfica de la simulación.
  - Ámbar: la representación gráfica de la simulación va a ser representada en breve.
  - Negro: se ha excedido el tiempo límite sugerido por el usuario y no se ha podido llevar a cabo una representación óptima de la simulación.

#### ➤ **Panel 'OTROS PARÁMETROS'**

En esta sección, se presentarán campos de edición destinados a los parámetros conocidos como las temperaturas del refrigerante y temperatura inicial del agua de trabajo, la constante de tiempo en el intercambiador de calor, entre otros valores elegidos para el punto de funcionamiento.

#### ➤ **Panel 'SALIDA TEMPERATURA LÍNEA FRÍA'**

En este panel se encuentra una serie de campos de texto no editables que muestran los parámetros calculados tras analizar la representación gráfica de la simulación.

Si se opta por la opción 'Planta', solo se presentarán los valores de los parámetros simulados sin controlador. En cambio, al seleccionar 'PID', se mostrarán los valores obtenidos mediante la simulación con el controlador PID establecido, junto con los valores resultantes de la simulación sin controlador. De esta forma, permite al usuario realizar pruebas en base a los valores obtenidos y elegir la mejor alternativa de controlador. En el caso de elegir 'Planta+PID', también se realizará un análisis de los datos de ambas simulaciones.

Además, se incorpora un desplegable o "Drop down" con algunos posibles valores para detectar el tiempo de establecimiento de la señal.

#### ➤ **Tabla 'CONSTANTES'**

Se presentan en forma tabular todas las constantes características de cada elemento del lazo de control. Además, se encuentran parámetros físicos del sistema, específicamente relacionados con el agua, que sirve como fluido de trabajo.

De igual manera, se incluyen valores editables en paneles anteriores, como las temperaturas iniciales del agua de trabajo y del refrigerante.

También se muestra la inversa de la resistencia global de intercambio de calor, la cual influirá en la masa de agua dentro del volumen de control del intercambiador de calor.

➤ **Tabla 'PUNTO DE FUNCIONAMIENTO'**

Se encuentran tabulados todos los valores de las variables en el punto de funcionamiento escogido.

➤ **Tabla 'ESTABILIDAD PLANTA'**

Realiza un análisis de las características de la planta sin controlador PID. Muestra el tipo de sistema, su estabilidad y su orden. En caso de que el sistema sea de segundo orden, se informa acerca del subtipo: amortiguado, subamortiguado o críticamente amortiguado.

Calcula la localización de los polos del sistema, así como de otros parámetros entre los que se encuentran el factor de amortiguamiento, el ángulo de elevación de los polos, etc.

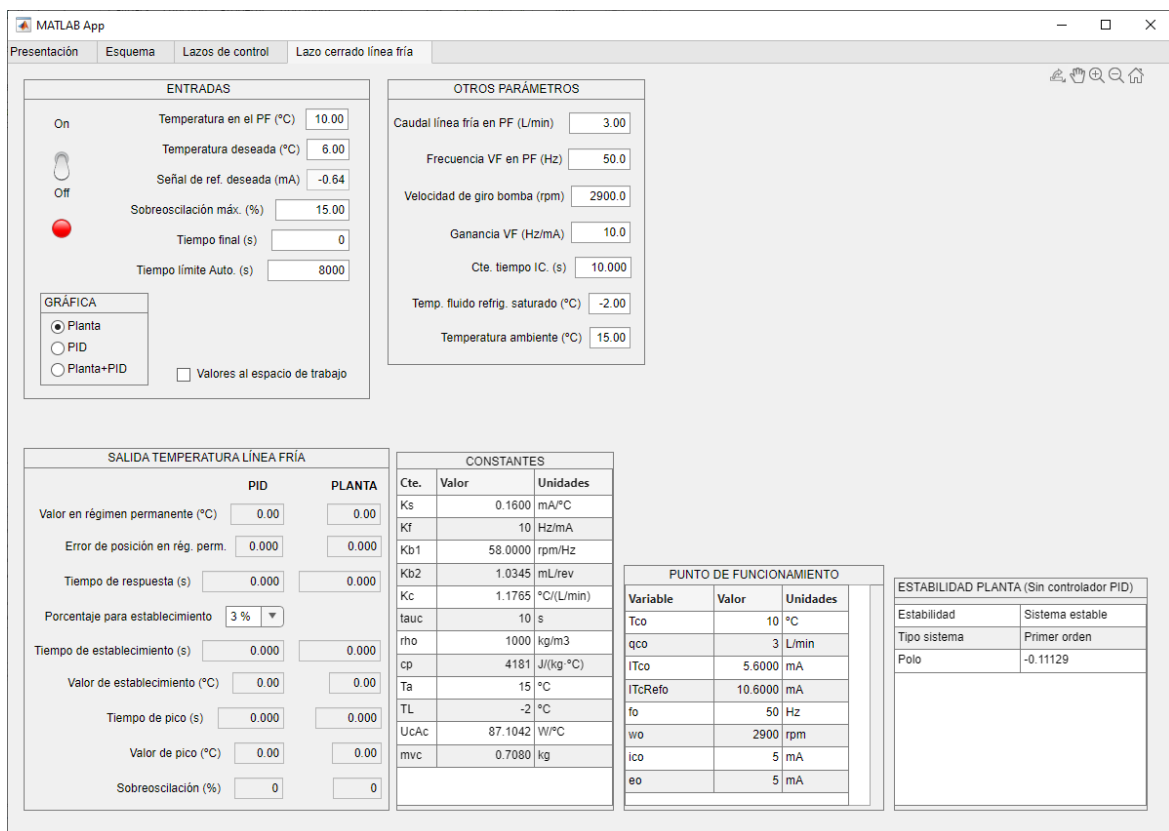


Figura 8.1: Pestaña 'Lazo de control línea fría'

En una primera instancia, todos los valores editables en esta interfaz se configurarán automáticamente con los valores predeterminados incorporados en el programa, definidos en la sección 6.4 de la memoria. Las modificaciones realizadas en la interfaz



no se guardarán para simulaciones posteriores. Al iniciar una nueva simulación, el programa tomará los valores de partida o valores por defecto.

En los siguientes apartados se van a explicar las funciones más importantes para la programación de esta interfaz. Además, se han programado una serie de callbacks o funciones que se ejecutan al interactuar con la GUI, que son numerosas pero su funcionalidad es auxiliar, aunque importante.

Una de las primeras tareas que se deben realizar en esta interfaz es la definición de una serie de propiedades o variables globales que se utilizarán en las funciones. Además, se seleccionan unos valores por defecto para las variables en el punto de funcionamiento y constantes asociadas.

Se puede acceder a estas variables globales añadiendo a su nombre el prefijo 'app.'.

### 8.1. Puesta en marcha

En primer lugar, si se ha seleccionado la pestaña 'Lazo cerrado línea fría' por primera vez, el programa comienza a realizar una serie de acciones encaminadas a poner en marcha la planta.

1. Mediante el código 'Clear out', se borra la posible existencia en el espacio de trabajo de Matlab de la variable 'out' que contiene los arrays de tiempo, temperatura obtenidos en simulaciones anteriores.
2. La lámpara, *app.Lamp*, se encuentra de color rojo. Esto es indicativo de que la planta no está en funcionamiento.  
`app.Lamp.Color = "red"`
3. A su vez, el interruptor, *app.Switch*, se encuentra deshabilitado hasta que la planta se encuentre lista para activarse.  
`App.Switch.Enable = "off"`
4. La gráfica que muestra la temperatura controlada a la salida del lazo, *app.UIAxes*, se encuentra invisible.  
`App.UIAxes.Visible = "off"`
5. Al seleccionar esta pestaña, se debe cargar automáticamente el fichero del modelo Simulink, y al salir debe guardarlo y cerrar. Para ello:
  - Mediante el manejador definido, *app.hsistemaSL*, se carga en memoria el fichero Simulink que contiene el modelo del lazo.  
`app.hsistemaSL = load_system(app.ficheroSL)`

- Una vez cargado el fichero, establecemos mediante la variable correspondiente, que efectivamente se encuentra cargado en la memoria.  
app.simulink\_cargado = true
  - Mediante el manejador definido, *app.hws*, se permitirá acceder a los parámetros del Model Workspace, del modelo Simulink cargado.  
app.hws = get\_param(bdroot, 'modelworkspace')
6. Pone a “cero” el valor de unas variables que recuerdan las gráficas que se han calculado y/o dibujado, mencionando la función *borraRecuerdoGraficas(app)*.

```
function borraRecuerdoGraficas(app)
    app.plaGRA_calculada = false;
    app.pidGRA_calculada = false;
    app.plaGRA_dibujada = false;
    app.pidGRA_dibujada = false;
    app.GRA_dibujada = false;
end
```

```
% Selection change function: TabGroup
function TabGroupSelectionChanged(app, event)
    selectedTab = app.TabGroup.SelectedTab;
    if strcmpi(selectedTab.Title, 'Lazo cerrado línea fría')
        clear out;
        app.error = false;
        % LAZO CERRADO LÍNEA FRÍA
        % Lazo apagado
        app.Lamp.Color = "red";
        app.UIAxes.Visible = "off";
        app.Switch.Enable = "off";
        drawnow

        % MODELO SIMULINK
        % Carga en memoria el fichero del modelo SIMULINK
        % y permite el acceso a las variables del 'Model Workspace'
        if app.simulink_cargado == false
            app.hsistemaSL = load_system(app.ficheroSL);
            app.hws = get_param(bdroot, 'modelworkspace');
            app.simulink_cargado = true;
        end

        % BORRA MEMORIA GRÁFICAS
        % Al arrancar el programa no se ha realizado nada (aún)
        % Borra los datos de análisis de las gráficas (planta y pid)
        % y pone a 'false' las variables que recuerdan si se han
        % calculado y/o dibujado las gráficas
        borraRecuerdoGraficas(app);
```

Figura 8.2: Callback TabGroupSelectionChanged (1/3)

```

% Copia valores por defecto
valoresDefecto(app);
% Calcula el punto de funcionamiento
puntoFuncionamiento(app);
% Estabilidad de la planta (sin PID)
estabilidadPlanta(app);

% Muestra los valores
muestraValores(app);
drawnow
% Envía datos al espacio de trabajo (WS) si está activada la
% opción
if app.valoresWS == 1
    cargaDatosEnWorkspace(app);
end

app.Switch.Enable = "on";
drawnow
else
    desactivaSwitchLamp(app);
end
end

```

Figura 8.3: Callback TabGroupSelectionChanged (2/3)

7. Adjudica los valores por defecto a las variables necesarias, los cuales se encuentran definidos en propiedades. Adicionalmente, invisibiliza los campos de edición referentes a los parámetros del controlador PID mediante la función *invisibilizaValoresPID(app)* ya que el tipo de gráfica por defecto no contiene el controlador. Para ello, se menciona la función *valoresDefecto(app)*.
8. Calcula el punto de funcionamiento para las variables mediante la función *puntoFuncionamiento(app)*.
9. Realiza el análisis de estabilidad de la planta, indicando también el tipo de sistema y subsistema, etc. Esta acción viene dada por la función *estabilidadPlanta(app)*.
10. Expone el valor de las variables y constates calculadas en las correspondientes tablas o campos de edición. Adicionalmente se actualizan los valores tabulados por los resultantes en la simulación reciente mediante las funciones *actualizaTablaConstantes(app)* y *actualizaTablaPF(app)*. Para ello, se menciona la función *muestraValores(app)*.
11. Comprueba si el usuario ha rellenado la casilla de verificación o "Check Box" referente a 'Pasar valores al espacio de trabajo' (representado cuando la variable *app.valoresWS* toma el valor 1), entonces debe cargar los valores de las variables obtenidas en Base Workspace.

Se logra con la función `cargaDatosEnWorkspace(app)` y permite visualizar y realizar cálculos con los valores obtenidos por medio de App Designer en Matlab, de forma independiente a esta aplicación.

12. Por último, se habilita el interruptor `app.Switch`, de forma que se pueda comenzar una simulación asistida por el usuario.

El código encargado de esta operación es `app.Switch.Enable = "on"`.

```
% Copia valores por defecto
valoresDefecto(app);
% Calcula el punto de funcionamiento
puntoFuncionamiento(app);
% Estabilidad de la planta (sin PID)
estabilidadPlanta(app);

% Muestra los valores
muestraValores(app);
drawnow
% Envía datos al espacio de trabajo (WS) si está activada la
% opción
if app.valoresWS == 1
    cargaDatosEnWorkspace(app);
end

app.Switch.Enable = "on";
drawnow
end
end
```

Figura 8.4: Callback TabGroupSelectionChanged (3/3)

#### 8.1.1. Función `valoresDefecto (app)`

Esta función permite guardar en cada una de las variables definidas, sus valores por defecto. Primero, establece el valor por defecto de la gráfica que se va a representar, mediante:

```
app.GRA_tipo = app.GRA_tipo_defecto
```

Como ya se ha mencionado en “propiedades”, el tipo de gráfica por defecto es referida a la planta sin controlador PID. Consecuentemente, el botón `app.PlantaButton` debe encontrarse marcado, tal y como observamos en la imagen. Para lo cual aplicamos el siguiente código:

```
app.PlantaButton.Value = true
```

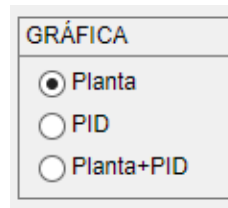


Figura 8.5: Grupo de botones 'Tipo de gráfica'

Las variables definidas que hacen alusión al controlador PID, es decir, sus parámetros P, I y D deben almacenar los valores por defecto. Estos valores deben estar contenidos en los campos de edición o “EditFields” correspondientes.

Por último, invisibiliza los valores del controlador PID, ya que la gráfica por defecto es la de la planta sin controlador. Esta acción se realiza mediante otra función llamada invisibilizaValoresPID(app). Una vez se elija la opción PID mediante el botón correspondiente, se mostrarán los valores guardados, pero en primera instancia deben ocultarse. Por otro lado, se guardan en las correspondientes variables definidas, los valores por defecto de los parámetros del sistema, así como de las variables en su punto de funcionamiento.

```
function valoresDefecto(app)
    % Tipo de gráfico 1 -> planta
    app.GRA_tipo = app.GRA_tipo_defecto;
    app.PlantaButton.Value = true;

    app.pidPID_P = app.pidPID_P_defecto;
    app.pidPID_I = app.pidPID_I_defecto;
    app.pidPID_D = app.pidPID_D_defecto;
    app.pidPID_N = app.pidPID_N_defecto;
    app.PEditField.Value = app.pidPID_P;
    app.IEditField.Value = app.pidPID_I;
    app.DEditField.Value = app.pidPID_D;
    app.NEditField.Value = app.pidPID_N;
    invisibilizaValoresPID(app);
```

Figura 8.6: Función valoresDefecto (1/2)

```
% Completa parámetros por defecto
app.tiempoFinal = app.tiempoFinal_defecto;
app.tiempoLimiteAuto = app.tiempoLimiteAuto_defecto;
app.valoresWS = app.valoresWS_defecto;
app.ps = app.ps_defecto;
app.Mpmax = app.Mpmax_defecto;

app.qco = app.qco_defecto;
app.Tco = app.Tco_defecto;
app.fo = app.fo_defecto;
app.wo = app.wo_defecto;

app.Kf = app.Kf_defecto;
app.Tcd = app.Tcd_defecto;
app.Ta = app.Ta_defecto;
app.TL = app.TL_defecto;
app.tauc = app.tauc_defecto;
end
```

**Figura 8.7: Función valoresDefecto (2/2)**

#### 8.1.1.1. Función invisibilizaValoresPID(app)

Esta función hace invisibles los campos de edición o “Edit Fields” de texto referidos a los parámetros proporcional, integral y diferencial. Esto se consigue mediante el siguiente código:

```
function invisibilizaValoresPID(app)
    app.PEditFieldLabel.Visible = "off";
    app.IEditFieldLabel.Visible = "off";
    app.DEditFieldLabel.Visible = "off";
    app.NEditFieldLabel.Visible = "off";
    app.PEditField.Visible = "off";
    app.IEditField.Visible = "off";
    app.DEditField.Visible = "off";
    app.NEditField.Visible = "off";
end
```

**Figura 8.8: Función invisibilizaValoresPID**

Así, cuando esté seleccionado el botón que visualiza datos de la gráfica teniendo en cuenta sólo la planta, sin controlador PID, estas opciones desaparecen.

Únicamente se deben mostrar si se están mostrando datos de la planta controlada por el PID seleccionado.

## 8.1.2. Función puntoFuncionamiento (app)

Esta función hace los cálculos necesarios para cada una de las variables en el punto de funcionamiento, y las guarda en las variables correspondientes definidas en “propiedades”.

```
function puntoFuncionamiento(app)
    app.UcAc = (app.rho*app.cp*app.qco*(app.Ta-app.Tco)/(app.Tco-app.TL));
    app.Kb2 = app.qco/app.wo;

    app.Kb1 = app.wo / app.fo;
    app.ITco = app.Ks1 + app.Ks2*app.Tco;
    app.eo = app.fo/app.Kf;
    app.ico = app.eo * app.pidPID_P;
    app.ITcRefo = app.eo + app.ITco;

    app.Kc = app.rho*app.cp*(app.Ta-app.Tco)/(app.rho*app.cp*app.qco+app.UcAc);
    app.mvc = app.tauc*(app.rho*app.cp*app.qco+app.UcAc)/app.cp;
end
```

Figura 8.9: Función puntoFuncionamiento

## 8.1.3. Función estabilidadPlanta(app)

Detecta si la planta es estable o no, y realiza una serie de cálculos para comprobar el tipo y subtipo de sistema, además de parámetros asociados.

```
function estabilidadPlanta(app)
    app.controlador = 'No';
    % Coeficientes
    b = app.tauc;
    c = 1 + app.Ks2*app.Kf*app.Kb1*app.Kb2*app.Kc;
    app.orden = 1;
    app.tipoSistema = 'Primer orden';
    app.s1Real = -c/b;
    if app.s1Real > 0
        app.estableTxt = 'Sistema inestable';
    elseif app.s1Real < 0
        app.estableTxt = 'Sistema estable';
    else
        app.estableTxt = 'Sistema inestable';
    end
end
```

Figura 8.10: Función estabilidadPlanta

## 8.1.4. Función muestraValores (app)

Esta función permite que se muestren los valores calculados del punto de funcionamiento, entre otros parámetros, en los correspondientes campos de edición o “Edit Fields”.

El porcentaje de estabilización, `app.ps`, se guarda en el desplegable o “Drop Down” correspondiente. Los porcentajes de estabilización posibles son 1, 3 y 5 %, que, aunque sean números, se leen por este componente como cadena de texto. Por ello, se debe utilizar la función `num2str()`.

El valor de la variable de la casilla de verificación o “Check Box” se guarda en `app.ValoresWSCheckBox.Value`.

Por último, se actualizan las tablas para las constantes y el punto de funcionamiento, así como la tabla de estabilidad. Esta acción se realiza mediante las funciones `actualizaTablaConstantes(app)`, `actualizaTablaPF(app)` y `actualizaTablaEstabilidad(app)`, respectivamente.

Como se editan los distintos campos de edición, es recomendable utilizar la línea de código ‘`drawnow`’ que fuerza a que se manifiesten estos cambios en la interfaz.

```
function muestraValores(app)
    % Valores editables
    app.MpmaxEditField.Value = app.Mpmax;
    app.psDropDown.Value = num2str(app.ps);
    app.TiempoFinalEditField.Value = app.tiempoFinal;
    app.TiempoLimiteAutoEditField.Value = app.tiempoLimiteAuto;
    app.ValoresWSCheckBox.Value = app.valoresWS;
    app.TcoEditField.Value = app.Tco;
    app.TcdEditField.Value = app.Tcd;
    app.ITcRefdEditField.Value = app.ITcRefd * 1000; % mA

    app.qcoEditField.Value = app.qco * 60000; % Paso de m3/s a L/min
    app.foEditField.Value = app.fo; % Hz
    app.woEditField.Value = app.wo * 60/(2*pi); % rpm/Hz
    app.KfEditField.Value = app.Kf/1000; % Hz/mA
    app.taucEditField.Value = app.tauc;
    app.TaEditField.Value = app.Ta;
    app.TLEditField.Value = app.TL;

    % Se actualizan las tablas
    actualizaTablaConstantes(app);
    actualizaTablaPF(app);
    actualizaTablaEstabilidad(app);
    drawnow
end
```

Figura 8.11: Función muestraValores



8.1.4.1. *Funciones actualizaTablaConstantes (app) u actualizaTablaPF (app)*

Estas funciones actualizan los valores de las constantes y de las variables en el punto de funcionamiento en las tablas correspondientes.

Ambas tablas mantienen el mismo orden, de forma que la primera columna está ocupada por la nomenclatura elegida para la variable o constante, la segunda muestra el valor de la misma y la tercera sus unidades.

```
function actualizaTablaConstantes(app)
    app.UITableKs.Data = [{'Ks' app.Ks*1000 'mA/°C'};
                        {'Kf' app.Kf/1000 'Hz/mA'};
                        {'Kb1' app.Kb1*(60/(2*pi)) 'rpm/Hz'};
                        {'Kb2' app.Kb2*2*pi*1e6 'mL/rev'};
                        {'Kc' app.Kc/60000 '°C/(L/min)'};
                        {'tauc' app.tauc 's'};
                        {'rho' app.rho 'kg/m3'};
                        {'cp' app.cp 'J/(kg·°C)'};
                        {'Ta' round(app.Ta, 2) '°C'};
                        {'TL' round(app.TL, 2) '°C'};
                        {'UcAc' app.UcAc 'W/°C'};
                        {'mvc' round(app.mvc,3) 'kg'}];
end
```

Figura 8.12: Función actualizaTablaConstantes

```
function actualizaTablaPF(app)
    app.UITablePF.Data = [{'Tco' app.Tco '°C'};
                        {'qco' app.qco*60000 'L/min'};
                        {'ITco' app.ITco*1000 'mA'};
                        {'ITcRefo' app.ITcRefo*1000 'mA'};
                        {'fo' app.fo 'Hz'};
                        {'wo' app.wo*60/(2*pi) 'rpm'};
                        {'ico' app.ico*1000, 'mA'};
                        {'eo' app.eo*1000 'mA'}];
end
```

Figura 8.13: Función actualizaTablaPF

La tabla de estabilidad de la planta se actualiza mediante el código mostrado a continuación:

```
function actualizaTablaEstabilidad(app)
    actualizaTablaEstabilidadPlanta(app);
end
```

Figura 8.14: Función actualizaTablaEstabilidad

```
function actualizaTablaEstabilidadPlanta(app)
    app.UITableEstabilidad.Data = [
        {'Estabilidad' app.estableTxt}
        {'Tipo sistema' app.tipoSistema}
        {'Polo' num2str(app.s1Real)}];
end
```

Figura 8.15: Función actualizaTablaEstabilidadPlanta

#### 8.1.5. Función cargaDatosEnModelWorkspacePlanta (app)

Esta función asigna los valores de las variables obtenidas mediante la simulación de la planta en App Designer a Model Workspace.

```
function cargaDatosEnModelWorkspacePlanta(app)
    % Se carga en memoria el modelo de SIMULINK de la planta
    % para poder modificar sus variables desde esta app
    % La variable hws permitirá el acceso a las variables del
    % 'Model Workspace' que hay en el modelo de SIMULINK
    % ficheroSL = 'LineaFriaSL';
    % Abre el modelo diseñado en SIMULINK y permite el acceso
    % al 'Model Workspace'.
    % La variable hwsSL tiene acceso a las variables del
    % 'Model Workspace' que hay en el modelo de SIMULINK de la
    % planta.
    % Se asignan valores del modelo (entre comillas simples)
    % utilizando los que hay en la aplicación (app).
    app.hws.assignin('ITcRefd', app.ITcRefd);
    app.hws.assignin('Ks1', app.Ks1);
    app.hws.assignin('Ks2', app.Ks2);
    app.hws.assignin('Kf', app.Kf);
    app.hws.assignin('Kb1', app.Kb1);
    app.hws.assignin('Kb2', app.Kb2);
    app.hws.assignin('Kc', app.Kc);
    app.hws.assignin('tauc', app.tauc);
```

Figura 8.16: Función cargaDatosEnModelWorkspacePlanta (1/2)

```
app.hws.assignin('Ta', app.Ta);
app.hws.assignin('TL', app.TL);
app.hws.assignin('qco', app.qco);
app.hws.assignin('Tco', app.Tco);
app.hws.assignin('fo', app.fo);
app.hws.assignin('wo', app.wo);
app.hws.assignin('eo', app.eo);

app.hws.assignin('rho', app.rho);
app.hws.assignin('cp', app.cp);
app.hws.assignin('UcAc', app.UcAc);
app.hws.assignin('mvc', app.mvc);

app.hws.assignin('P', app.plaPID_P);
app.hws.assignin('I', app.plaPID_I);
app.hws.assignin('D', app.plaPID_D);
app.hws.assignin('N', app.plaPID_N);
end
```

Figura 8.17: Función cargaDatosEnModelWorkspacePlanta (2/2)

Hay otra función similar, `cargaDatosEnModelWorkspacePID(app)` que se encarga de realizar la carga de datos en el modelo, cuando se elige un controlador PID concreto. Esta última solo se diferencia de la anterior en las últimas cuatro líneas.

#### 8.1.6. Función `cargaDatosEnWorkspace (app)`

Tal y como se puede leer en la captura, esta función permite pasar los valores obtenidos en App Designer a variables del BaseWorkspace.

De esta manera, se pueden realizar operaciones y visualizar variables, definidas en App Designer, en Matlab.

```
function cargaDatosEnWorkspace(app)
    % Se cargan las variables principales de esta simulación
    % en el espacio de trabajo principal de Matlab
    % para poder modificar y usar sus variables desde aquél.
    assignin('base','ITcRefo', app.ITcRefo);
    assignin('base','ITco', app.ITco);
    assignin('base','Ks1', app.Ks1);
    assignin('base','Ks2', app.Ks2);
    assignin('base','Kf', app.Kf);
    assignin('base','Kb1', app.Kb1);
    assignin('base','Kb2', app.Kb2);
    assignin('base','Kc', app.Kc);
    assignin('base','tauc', app.tauc);
    assignin('base','rho', app.rho);
    assignin('base','cp', app.cp);
    assignin('base','UcAc', app.UcAc);
    assignin('base','Tco', app.Tco);
    assignin('base','qco', app.qco);
    assignin('base','fo', app.fo);
```

Figura 8.18: Función `cargaDatosEnWorkspace (1/2)`

```
assignin('base', 'eo', app.eo);
assignin('base', 'Ta', app.Ta);
assignin('base', 'TL', app.TL);
assignin('base', 'Tcd', app.Tcd);
assignin('base', 'ITcRefd', app.ITcRefd);
assignin('base', 'mvc', app.mvc);
assignin('base', 'out', app.out);
app.hws.assignin('P', 1);
app.hws.assignin('I', 0);
app.hws.assignin('D', 0);
app.hws.assignin('N', 100);

end
```

Figura 8.19: Función cargaDatosEnWorkspace (2/2)

## 8.2. Callback SwitchValueChanged (app, event)

Este 'callback' determina el comportamiento de la interfaz al interactuar con el interruptor *app.Switch*.

Se utiliza el "condicional case", que provoca que el programa responda de distinta forma en función de si el interruptor está encendido o no.

- Si interactuamos con el interruptor para encenderlo, el color de la lámpara cambiará provisionalmente a un tono ámbar. Este color será indicativo de que la simulación aún no se ha calculado.

La función `evalin('base', 'clear out')` ejecuta la orden 'clear out' en el espacio de trabajo de Matlab desde App Designer. Con esto se consigue borrar la variable 'out' que pudiera existir en el espacio de trabajo. Esta variable contiene los arrays de tiempos, temperaturas y otras variables, de gráficas anteriores.

- Dentro de este caso, se comprueba:

Si el usuario ha seleccionado una simulación de la planta sin uso de controlador PID mediante el botón *app.PlantaButton*.

Se llama a la función `gestionaGraficaPlanta(app,0)`, que se encarga de calcular la gráfica de la planta.

Una vez gestionada la gráfica, se realiza el dibujo de la gráfica de la simulación en la planta y se incorpora en el componente *app.UIAxes* de la interfaz, mediante la función `dibujaSoloGraficaPlanta(app)`.

Por último, analiza los datos resultantes de la simulación mediante la función `analizaRespuestaPlanta(app)`.

Una vez se ha calculado y dibujado la gráfica de la planta, además de analizar los datos obtenidos de realizar la simulación, se modifica el color de la lámpara *app.Lamp* pasando de ámbar a verde. El color verde es indicativo de que ya se ha terminado la simulación y se han procesado y mostrado todos los datos relacionados con ésta.

Además, se utiliza la función `save_system(app.ficheroSL)` para guardar el fichero Simulink.

- Si por el contrario, el usuario elige pulsar el botón *app.PIDButton* para realizar la simulación con un controlador PID, entonces se nombran las siguientes funciones.

Se nombra la función *gestionaGraficaPID(app,0)*, *analizaRespuestaPID(app)*, *analizaRespuestaPlanta(app)* y *dibujaSoloGraficaPID(app)*.

```
% Value changed function: Switch
function SwitchValueChanged(app, event)
    value = app.Switch.Value;
    switch value
        case 'On'
            app.error = false;
            % Cambia color de Lamp a un tono anaranjado
            % Al final se pondrá verde
            app.Lamp.Color = '#FFA500';
            drawnow;
            evalin('base', 'clear out');
            if app.GRA_tipo == 1
                gestionaGraficaPlanta(app, 0);
                % Dibuja la gráfica
                dibujaSoloGraficaPlanta(app);
            elseif app.GRA_tipo == 2
                gestionaGraficaPID(app, 0);
                dibujaSoloGraficaPID(app);
            elseif app.GRA_tipo == 3
                gestionaGraficasPlantayPID(app, 0);
                dibujaGraficasPlantayPID(app);
            elseif app.GRA_tipo == 4
                gestionaGraficasPlantayReal(app,0);
                dibujaGraficasPlantayReal(app);
            end
            save_system(app.ficheroSL);
            app.Lamp.Color = "green";
            drawnow
        case 'Off'
            desactivaSwitchLamp(app);
            borraAnalisis(app);
    end
end
```

Figura 8.20: Callback SwitchValueChanged

### 8.2.1. Función borraAnálisisPID (app)

Esta función adjudicará a todos los valores de las variables y constantes calculadas en la simulación PID, un valor igual a cero.

Se utiliza cuando se desea la gráfica de la planta sin controlador PID, para que no muestre valores de una posible simulación anterior con este controlador, dejando esta columna de datos vacía.

Adicionalmente se añade un color gris al campo de edición donde el programa incorpora el valor de la sobreoscilación, pidGRA\_MpTc, tras simular el modelo de la planta con controlador PID, para que desaparezca el posible color verde o rojo del fondo del campo que registra la sobreoscilación.

A continuación, se muestran las líneas de código necesarias.

```
function borraAnálisisPID(app)
    app.TcrpRealPIDEditField.Value = 0;
    app.GananciaPIDEditField.Value = 0;
    app.trPIDEditField.Value = 0;
    app.tsPIDEditField.Value = 0;
    app.TcsRealPIDEditField.Value = 0;
    app.tpPIDEditField.Value = 0;
    app.TcpRealPIDEditField.Value = 0;
    app.MpTcPIDEditField.Value = 0;
    app.MpTcPIDEditField.BackgroundColor = [0.94 0.94 0.94];
end
```

**Figura 8.21: Función borraAnálisisPID**

### 8.2.2. Función borraAnálisisPlanta (app)

Esta función adjudicará a todos los valores de las variables y constantes calculadas en la simulación de la planta sin controlador, un valor igual a cero.

Se utiliza cuando el interruptor se apaga, bien por el usuario o al cambiar cualquier valor de una variable en el punto de funcionamiento o un parámetro que afecte al análisis.

Adicionalmente se añade un color gris al campo de edición donde el programa incorpora el valor de la sobreoscilación, plaGRA\_MpTc, tras simular el modelo de la planta sin controlador PID. Así, deja ver al usuario que este apartado no se encuentra disponible.

El código es similar al anterior, por lo que no se lista aquí.

### 8.2.3. Función borraAnálisis (app)

Esta función se encarga de borrar tanto el análisis derivado de la simulación de la planta sin controlador PID, como el derivado de la simulación controlada con PID. Esto ocurre normalmente cuando se desactiva la planta.

```
function borraAnálisis(app)
    borraAnálisisPlanta(app);
    borraAnálisisPID(app)
end
```

**Figura 8.22: Función borraAnálisis**

### 8.2.4. Función gestionaGraficaPlanta (app, 0)

Para representar la gráfica de la planta, se crea la función *gestionaGraficaPlanta(app, tmax)*.

En el callback encargado de la interactividad del interruptor *app.Switch*, se ha mencionado esta función incorporando para el valor de este tiempo un cero, de forma que el usuario no dé orden de seleccionar un tiempo de simulación concreto a través de esta línea de código.

En primer lugar, se llama a la función *borraAnálisisPID(app)* por si se hubiera calculado anteriormente una gráfica de simulación con controlador.

Se comprueba si la gráfica de la planta no está calculada, en cuyo caso, comienza con su cálculo mediante la función *calculaGrafica(app)*. Calcular una gráfica significa obtener en una variable 'out', los arrays de tiempo y temperaturas que han de ser representadas más tarde. Por ejemplo, el array de tiempo está en 'out.tout', el de temperaturas en 'out.incTcReal'. La variable que guarda información acerca de si se encuentra o no calculada una gráfica para el proceso (sin controlador PID) se encuentra definida en la propiedad *app.plaGRA\_calculada*. En el caso de una gráfica para la planta a la que se le añade un controlador PID es *app.pidGRA\_calculada*.

Una vez calculada la gráfica, si el parámetro enviado a esta función, contiene un tiempo máximo igual a cero y se hubiera sobrepasado el tiempo límite, entonces se devuelve un error mediante la variable *app.error*. En ese caso, se vuelve a calcular la gráfica con el tiempo límite máximo, para que el usuario, cambiando el valor de tiempo final en el panel 'ENTRADAS', pueda decidir que tiempo final se debe representar para la gráfica.

```

function gestionaGraficaPlanta(app, tmax)
    if app.pidGRA_calculada == true
        borraAnálisisPID(app);
    end
    cargaDatosEnModelWorkspacePlanta(app);
    if app.plaGRA_calculada == false
        [tiempos, incTcReal, incTc, tmax, calculada] = calculaGrafica(app, tmax);
        if app.error == true
            [tiempos, incTcReal, incTc, tmax, calculada] = calculaGrafica(app, app.tiempoLimiteAuto);
        end
        app.plaGRA_tiempos = tiempos;
        app.plaGRA_incTcReal = incTcReal;
        app.plaGRA_incTc = incTc;
        app.plaGRA_tmax = tmax;
        app.plaGRA_calculada = calculada;
    end
    if app.error == false
        analizaRespuestaPlanta(app);
    end

    % Envía datos al espacio de trabajo (WS) si está activada la
    % opción
    if app.valoresWS == 1
        cargaDatosEnWorkspace(app);
    end
end

```

Figura 8.23: Función gestionaGraficaPlanta

#### 8.2.5. Función gestionaGraficaPID(app,0)

Esta función se programa de forma análoga a la anterior, con la diferencia de que se encarga de gestionar previamente la gráfica de la planta si no se hubiera hecho.

```

function gestionaGraficaPID(app, tmax)
    % PLANTA
    gestionaGraficaPlanta(app, tmax);
    % Calcula, dibuja y analiza PID
    cargaDatosEnModelWorkspacePID(app);
    if app.pidGRA_calculada == false
        [tiempos, incTcReal, incTc, tmax, calculada] = calculaGrafica(app, tmax);
        if app.error == true
            [tiempos, incTcReal, incTc, tmax, calculada] = calculaGrafica(app, app.tiempoLimiteAuto);
        end
        app.pidGRA_tiempos = tiempos;
        app.pidGRA_incTcReal = incTcReal;
        app.pidGRA_incTc = incTc;
        app.pidGRA_tmax = tmax;
        app.pidGRA_calculada = calculada;
        % Analiza respuesta de datos
        analizaRespuestaPID(app);
    end
    if app.error == false
        analizaRespuestaPID(app);
    end

    % Envía datos al espacio de trabajo (WS) si está activada la
    % opción
    if app.valoresWS == 1
        cargaDatosEnWorkspace(app);
    end
end

```

Figura 8.24: Función gestionaGraficaPID



### 8.2.6. Función `gestionaGraficasPlantayPID(app)`

Esta función gestiona ambas gráficas, de la planta y de la planta controlada con PID.

Además, se comparan los tiempos máximos asignados a ellas y se elige el mayor de ellos. Entonces, se gestiona de nuevo la gráfica que hubiera utilizado un tiempo menor, asignándole este nuevo tiempo común con el objetivo de que ambas gráficas superpuestas utilicen el mismo intervalo de tiempo.

```
function gestionaGraficasPlantayPID(app, tmax)
    % PLANTA
    gestionaGraficaPlanta(app, tmax);
    % PID
    gestionaGraficaPID(app, tmax);
    if app.plaGRA_tmax > app.pidGRA_tmax
        app.pidGRA_calculada = false;
        gestionaGraficaPID(app, app.plaGRA_tmax);
    elseif app.plaGRA_tmax < app.pidGRA_tmax
        app.plaGRA_calculada = false;
        gestionaGraficaPlanta(app, app.pidGRA_tmax);
    end
    analizaRespuestaPID(app);
end
```

**Figura 8.25: Función `gestionaGraficasPlantayPID`**

### 8.2.7. Función `calculaGrafica(app, tiempoLimiteMax)`

Se ordena que la gráfica empiece a calcularse cuando el tiempo es cero segundos. Para ello, se nombra una variable haciendo referencia al tiempo donde comienza la gráfica, llamada '*tmin*', y se hace uso del manejador `app.hsistemaSL` para adjudicar este parámetro a '*StartTime*' en Simulink.

Se consideran dos fases: una preliminar y la segunda, definitiva. La preliminar se ejecuta cuando el sistema debe buscar un tiempo máximo apropiado para la gráfica. Esto únicamente se hace cuando el usuario deja el valor 'tiempo final' en el panel 'ENTRADAS' a cero. Entonces, el tiempo que recibe la función en el parámetro `tiempoLimiteMax` es cero.

#### 8.2.7.1. *Primera fase*

Esta primera fase de tanteo se basa en calcular un tiempo adecuado para representar la gráfica, el cual se guardará en la variable correspondiente, `app.plaGRA_tmax`. Esto es, que este tiempo no sea escaso de forma que no se pueda apreciar la evolución de la variable a controlar con el tiempo, y tampoco sea excesivo para poder visualizar bien el gráfico.

Esto se consigue determinando el tiempo de estabilización, de forma que se interrumpa la representación poco después de llegar a dicho punto. Así, se conseguirá una buena visualización del contenido en la gráfica.

1. Mediante el manejador *app.hsistemaSL*, se elige como solucionador o “SolverType” el tipo “Variable-step”. Este tipo de solucionador elige un número de pasos o intervalos variable en función de la pendiente de la gráfica. Así, si un tramo de la gráfica tiene una pendiente muy elevada, el programa elige incrementos de tiempo más cortos. Únicamente se sugiere el intervalo de tiempo máximo que no se debe superar, para que la gráfica no conste de segmentos rectos, en la medida de lo posible, utilizando de nuevo el manejador del sistema. Este intervalo límite máximo será de una décima de segundo.
2. Se asigna un tiempo máximo igual al tiempo límite automático especificado en la interfaz, en *app.tiempoLimiteAuto*, para asegurarnos de que el sistema se ha estabilizado. Cuando no ocurra, dará un error. Este parámetro se adjudica a ‘StopTime’ de Simulink.
3. Se ejecuta Simulink.
4. Los resultados de la simulación del modelo Simulink guardado en *app.ficheroSL*, se guarda en la variable *app.out*. Estos resultados son los que se envían a través de los elementos ‘To Workspace’.  
Los tiempos resultantes de la simulación, *app.out.tout*, se almacenan en una variable privada llamada ‘*tiemposPrueba*’.  
Asimismo, los resultados de temperatura, *app.out.Tc*, se almacenan en una variable privada llamada ‘*datosPrueba*’.
5. Llama a la función *analizaEstabilizamientoDatos(app, tiemposPrueba, datosPrueba, 1) = [ts]*, que devuelve como resultado, el valor de *app.ts*.  
A esta función se le incorporan las variables *tiemposPrueba*, *datosPrueba*, y el porcentaje de estabilización, que corresponde con el valor “1”. Se fija este valor para el porcentaje de estabilización ya que, para esta fase de tanteo, el resultado es más preciso y cercano al punto de régimen permanente, donde se pretende terminar la gráfica. Únicamente se fija este porcentaje para los cálculos que se dan dentro de la función *analizaEstabilizamientoDatos()* y no tiene por qué coincidir con el valor seleccionado por el usuario en la interfaz.
6. El tiempo máximo que se utilizará en la gráfica de la planta, *tmax*, se calcula multiplicando este tiempo de estabilización resultante por un número igual a 3.
7. Para redondear a un número entero múltiplo de 10, se llama a la función *tiempoMaxFormatoEstandar(app, tmax)*.

- Ahora, si esta función recibe un tiempo, *tiempoLimiteMax*, distinto de cero, entonces se guardará este tiempo en la variable *tmax*.
- Por otro lado, si el usuario ha modificado el valor del campo de edición *app.TiempoFinalEditField*, entonces éste se guardará en la variable *app.tiempoFinal*.

```
function [tiempos, incTcReal, incTc, tmax, graficaCalculada] = calculaGrafica(app, tiempoLimiteMax)
% Tiempo inicial para la simulación (SIMULINK)
tmin = 0;
set_param(app.hsistemaSL, 'StartTime', num2str(tmin));
% PRIMERA FASE
if tiempoLimiteMax == 0 && app.tiempoFinal == 0
    % Se debe elegir un tiempo máximo
    % Debemos dejar que la app encuentre el "mejor" tiempo máximo posible
    set_param(app.hsistemaSL, 'SolverType', 'Variable-step');
    set_param(app.hsistemaSL, 'MaxStep', '0.1');
    % Tanteo un tiempo de establecimiento probable para la
    % primera fase (fase de tanteo)
    tmax = app.tiempoLimiteAuto;
    set_param(app.hsistemaSL, 'StopTime', num2str(tmax));
    % Se lanza una simulación de prueba con los parámetros anteriores
    app.out = sim(app.ficheroSL);
    tiemposPrueba = app.out.tout;
    datosPrueba = app.out.incTc;
    % Se utiliza el 1% para detectar la estabilización de la señal
    [~,ts,~,~] = analizaEstablecimientoDatos(app, tiemposPrueba, datosPrueba, 1);
    tmax = 1.4*ts;
    % Como se ha un tiempo máximo en la GUI, el sistema
    % ha elegido uno que quizás no sea conveniente, como
    % 234.73, por tanto debemos normalizarlo.
    tmax = tiempoMaxFormatoEstandar(app, tmax);
elseif tiempoLimiteMax ~= 0
    tmax = tiempoLimiteMax;
else
    tmax = app.tiempoFinal;
end
```

Figura 8.26: Función calculaGrafica (1/3)

#### 8.2.7.2. Segunda fase

Mediante el manejador *app.hsistemaSL*, se adjudica el valor resultante de *tmax* al tiempo de parada *'StopTime'* en Simulink.

- Si *app.tiempoFinal* tiene valor cero (su valor por defecto) y el tiempo *tmax* es mayor que el tiempo adjudicado como límite para la simulación, *app.tiempoLimiteAuto*, entonces se mostrará un error.

La variable que guarda información sobre si ocurre un error, *app.error*, adquiere el valor *'true'*, la lámpara pasa a verse de color negro y el interruptor se inhabilita.

Además, en Matlab escribe el siguiente mensaje:

“Error: En modo de tiempo auto se ha superado el tiempo límite: (valor guardado en *tmax*)”.

- Si no ocurre la anterior condición, se incorpora un incremento de tiempo de una centésima, y se añade en Simulink mediante el manejador `app.hsistemaSL`. Se lanza la simulación definitiva y sus resultados se guardan en la variable `app.out`.

Si el usuario ha marcado la casilla ‘Pasar valores al espacio de trabajo’, entonces envía los resultados anteriores a `BaseWorkspace`.

El array de tiempos de la simulación guardado en la variable `app.out.tout` se almacena en la variable *tiempos*. De igual manera, se almacenan los arrays referidos a la temperatura y caudal en las variables correspondientes.

Por último, se habilita el interruptor y se da la gráfica por calculada.

```
% SEGUNDA FASE
set_param(app.hsistemaSL, 'StopTime', num2str(tmax));
set_param(app.hsistemaSL, 'SolverType', 'Variable-step');
if (app.tiempoFinal == 0 && tmax > app.tiempoLimiteAuto)
    maxIncr = 10;
    set_param(app.hsistemaSL, 'MaxStep', num2str(maxIncr));
    app.error = true;
    %app.Lamp.Color = "black";
    %drawnow
    fprintf('Error: En modo de tiempo automático se ha superado el tiempo límite: %f\n', tmax);
else
    app.error = false;
    maxIncr = 0.1;
    set_param(app.hsistemaSL, 'MaxStep', num2str(maxIncr));
    % Se lanza la simulación definitiva
    app.out = sim(app.ficheroSL);
    if app.valoresWS == 1
        assignin('base', "out", app.out);
    end
    tiempos = app.out.tout;
    incTcReal = app.out.incTcReal;
    incTc = app.out.incTc;

    app.Switch.Enable = "on";
    drawnow
    graficaCalculada = true;
end
end
```

Figura 8.27: Función `calculaGrafica` (2/3)

A continuación, se presenta la función que normaliza el tiempo máximo con un valor redondeado para poder ser utilizado como tiempo final en una gráfica.

```
function results = tiempoMaxFormatoEstandar(~, tiempo)
    exponente = fix(log10(tiempo));
    tFinEstandar = 10^exponente * (1+int32(tiempo/10^exponente));

    results = tFinEstandar;
end
```

**Figura 8.28: Función calculaGrafica (3/3)**

#### 8.2.8. Función analizaEstablecimientoDatos (app, tiempos, datos, ps) = [is, ts, ys, yrp]

Esta función devuelve unos resultados, contenidos en el corchete que contiene la respuesta de la función. Estos son, el índice en los arrays de tiempos y datos donde se ha considerado estabilizada la señal, is, el lugar de estabilización del array de tiempo, ys, el tiempo de estabilización, ts, y el valor de régimen permanente en el array de datos, yrp.

Como estos valores de establecimiento no son conocidos de momento, se crean las correspondientes variables privadas a las que se les incorpora el código NaN (Not a Number).

El valor de régimen permanente será el último valor del array de datos. Esto no significa necesariamente que el régimen permanente empiece en este último lugar del array, sino que con seguridad este valor será el valor de temperatura régimen permanente.

Para calcular el establecimiento de la señal, se realiza una serie de cálculos dependiendo de si el valor de temperatura en régimen permanente,  $y_{rp}$  es positivo o negativo.

Se definen las variables privadas ysinf e yssup, que indican las temperaturas superior e inferior dentro del rango de tolerancia admisible, teniendo en cuenta el porcentaje de estabilización.

Estos cálculos se pueden observar en la imagen de código adjunta, dentro del bucle mencionado.

Una vez se conocen estos límites, se recorre el array de datos que contiene las temperaturas en todos los puntos, y se comprueba dónde coinciden estos límites.

Para ello, el recorrido debe ser en orden inverso. Esto es, del último valor al primer valor del array de datos, haciendo la comprobación de dato en dato, tal y como se puede observar en el “bucle for” en la imagen que muestra el código.

La comprobación consiste en comparar el dato de temperatura en la posición ‘i’ a lo largo del array de datos, con los valores inferior y superior anteriormente mencionados. De forma que, si este valor es inferior al límite inferior y superior al límite superior, este punto conformará el valor de estabilización, y se rompe el bucle. Al realizar la comprobación de dato en dato de temperatura, esta comprobación resulta bastante precisa.

Por último, se guardan los valores de punto de establecimiento en las respectivas variables locales. Se recuerda que esta función, dará como resultado el valor de estas variables.

```
function [is,ts,ys,yrp] = analizaEstablecimientoDatos(~, tiempos, datos, ps)
    % RÉGIMEN PERMANENTE
    yrp = datos(length(datos));

    % ESTABLECIMIENTO DE LA SEÑAL
    is = NaN;
    ts = NaN;
    ys = NaN;
    if yrp > 0
        ysinf = yrp - yrp*ps/100;
        yssup = yrp + yrp*ps/100;
    else
        ysinf = yrp + yrp*ps/100;
        yssup = yrp - yrp*ps/100;
    end
    % Recorremos las matrices en orden inverso
    for i=length(tiempos):-1:1
        y = datos(i);
        if any([y < ysinf,y > yssup])
            is = i;
            ts = tiempos(i);
            ys = y;
            break;
        end
    end
end
```

**Figura 8.29:** Función `analizaEstablecimientoDatos`

#### 8.2.9. Función `AnalizaRespuestaDatos` (app)

Esta función calcula la velocidad de respuesta a una entrada escalón. Existen varias maneras de calcularla. En este caso, vamos a utilizar el tiempo de subida o crecimiento, que aquí se denotará como tiempo de respuesta,  $t_r$ . Este es el tiempo requerido para que la respuesta aumente o disminuya del 10 al 90% o del 0 al 100% del valor en régimen permanente. Si el sistema es subamortiguado, se suele utilizar los límites de 0 a 100%, mientras que en el resto de los casos se utilizaremos los límites del 10 al 90%.

Si hay sobreoscilación, se entiende que se trata de un sistema subamortiguado y utilizaremos  $f_{min}=0$  y  $f_{max}=1$ , indicando los límites de 0 y 100% del régimen permanente.

En caso contrario, se utilizarán los límites  $f_{min}=0.1$  y  $f_{max}=0.9$ , correspondientes al 10 y al 90%, respectivamente.

Primero ajustamos los valores de temperatura restándoles su valor inicial para que partan de cero y le aplicamos el valor absoluto. Esto provoca que la función sea creciente al comienzo del transitorio y empiece a partir de 0°C (es decir, se evalúa la simulación considerando valor inicial nulo).

Se calculan los valores de temperatura mínimo,  $y_{min}$  e  $y_{max}$ , multiplicando  $f_{min}$  y  $f_{max}$  por el valor en régimen permanente suponiendo valor inicial nulo y en valor absoluto, respectivamente.

Se asigna a  $ir_{min}$  y  $tr_{min}$  los valores NaN, indicando que aún se ha encontrado ese primer punto mínimo.

Se recorre el array de tiempo y de datos desde la posición  $i=1$  hasta el final. Para cada valor de 'i' en los arrays, se obtiene el valor de temperatura que también se encontrará en forma normalizada.

Ahora, si no se ha encontrado aún el valor mínimo  $ir_{min}$ , entonces se pregunta si el valor de temperatura en el punto 'i' coincide o supera  $yr_{min}$ . En ese caso, se ha encontrado el tiempo de respuesta mínimo,  $tr_{min}$ .

Análogamente, se opera de la misma forma entrando en otro bucle desde el valor de  $ir_{min}+1$  encontrado hasta el final del array. Tras esto, se obtiene el valor de  $tr_{max}$ .

Finalmente obtenemos el tiempo de respuesta,  $tr$ , restando  $tr_{max}-tr_{min}$ .

```
function [tr, irmin, irmax, trmin, trmax] = analizaRespuestaDatos(tiempos, datos, Mp)
    if Mp > 1
        % Si hay sobreoscilación 0% y 100%
        fmin = 0;
        fmax = 1;
    else
        % En caso contrario 10% y 90%
        fmin = 0.1;
        fmax = 0.9;
    end

    % VALOR INICIAL
    yini = datos(1);

    % RÉGIMEN PERMANENTE
    yrp = abs(datos(length(datos))-yini);
    % 0% o 10% de la señal de respuesta
    yrmin = fmin * yrp;
    % 100% o 90% de la señal de respuesta
    yrmax = fmax * yrp;
```

Figura 8.30: Función analizaRespuestaDatos (1/2)

```

% Tiempos de respuesta DE LA SEÑAL (tr, irmin, trmin, yrmin, irmax, trmax, yrmax)
% trmin
irmin = NaN;
trmin = NaN;
% Recorre los tiempos y datos para encontrar yrmin y trmin
for i = 1:length(tiempos)
    y = abs(datos(i)-yini);
    if y >= yrmin
        trmin = tiempos(i);
        irmin = i;
        break;
    end
end
% trmax
irmax = NaN;
trmax = NaN;
for i = (irmin + 1):length(tiempos)
    y = abs(datos(i)-yini);
    if y >= yrmax
        irmax = i;
        trmax = tiempos(i);
        break;
    end
end
tr = trmax-trmin;
end

```

Figura 8.31: Función analizaRespuestaDatos (2/2)

#### 8.2.10. Función analizaPicoDatos (tiempo, datos)

Esta función analiza los datos de pico resultantes de la simulación, es decir, el tiempo de pico,  $t_p$ , y el valor de pico,  $y_p$ , y la sobreoscilación,  $M_p$ .

El sistema utilizado es similar al anterior. Obliga a que las condiciones iniciales sean nulas y las comparaciones las realiza en valor absoluta para no tener que distinguir si el pico se encuentra en un valor más alto o más bajo del valor en régimen permanente.

Empieza la comparación una vez el valor del array datos supere el valor en régimen permanente, y a partir de este punto, registra el mayor valor alcanzado. Así, se encuentra el mayor valor que corresponde al valor de pico.

```

% Esta función recibe los parámetros (aparte de 'app'):
% time: array de tiempos en segundos.
% data: altura del depósito en cada tiempo.
% app.ps: porcentaje considerado para el establecimiento de la
%       señal.
% Retorna un arreglo con:
% ip: Posición de la señal de pico en los arrays time y data
% tp: Tiempo de pico
% yp: Valor de la señal de pico
% Mp: Sobreoscilación

```

Figura 8.32: Función analizaPicoDatos (1/2)



```

function [ip,tp,yp,Mp] = analizaPicoDatos(~, tiempos, datos)
    % VALOR INICIAL
    yini = datos(1);
    yrp = datos(length(datos));
    yrp_incr = yrp - yini;
    % RÉGIMEN PERMANENTE
    % PICO DE LA SEÑAL (ip, tp, yp)
    % Señal más "baja" posible
    % Supongo que no hay pico
    yp_incr = 0;
    ip = 0;
    tp = 0;
    yp = 0;
    Mp = 0;
    % Recorro las matrices de tiempo y de datos
    for i = 1:length(tiempos)
        y = datos(i);
        y_incr = y - yini;
        if ((abs(y_incr) > abs(yrp_incr)) && abs(y_incr) > abs(yp_incr))
            yp_incr = y_incr;
            ip = i;
            tp = tiempos(i);
        end
    end
    if ip ~= 0
        yp = yp_incr + yini;
        Mp = ((yp-yrp)/(yrp-yini))*100;
    end
end

```

Figura 8.33: Función analizaPicoDatos (2/2)

## 8.2.11. Función analizaRespuestaPlanta (app)

Esta función analiza la respuesta de la simulación lanzada de la planta sin controlador PID. Su código se divide en tres partes: el análisis de los valores de pico, del establecimiento de datos y valor en régimen permanente y por último, del tiempo de respuesta en el transitorio.

```

function analizaRespuestaPlanta(app)
    % Analiza pico de datos
    [ip,tp,yp,Mp] = analizaPicoDatos(app, app.plaGRA_tiempos, app.plaGRA_incTcReal);
    app.plaPIC_ip = ip;
    app.plaPIC_tp = tp;
    app.plaPIC_incTc = yp;
    app.plaPIC_MpTc = Mp;
    if ~isnan(app.plaPIC_tp)
        app.tpPlantaEditField.Value = app.plaPIC_tp;
        app.TcpRealPlantaEditField.Value = app.plaPIC_incTc;
        app.MpTcPlantaEditField.Value = app.plaPIC_MpTc;
    else
        app.tpPlantaEditField.Value = 0;
        app.TcpRealPlantaEditField.Value = 0;
        app.MpTcPlantaEditField.Value = 0;
    end
end

```

Figura 8.34: Función analizaRespuestaPlanta (1/2)

```

if (Mp < app.Mpmax || isnan(Mp))
    app.MpTcPlantaEditField.BackgroundColor = "#cbf0cb";
else
    app.MpTcPlantaEditField.BackgroundColor = "#f0cbcb";
end
% Analiza establecimiento y régimen permanente
[is,ts,ys,yrp] = analizaEstablecimientoDatos(app, app.plaGRA_tiempos, app.plaGRA_incTcReal, app.ps);
app.plaEST_is = is;
app.plaEST_ts = ts;
app.plaEST_incTcsReal = ys;
app.plaEST_incTcs = ys - app.Tco;
app.plaRP_incTcrpReal = yrp;
app.plaRP_Tceprp = (app.plaRP_incTcrpReal-app.Tcd)/(app.Tco-app.Tcd);

if ~isnan(app.plaEST_ts)
    app.tsPlantaEditField.Value = app.plaEST_ts;
    app.TcsRealPlantaEditField.Value = app.plaEST_incTcs;
    app.TcrpRealPlantaEditField.Value = app.plaRP_incTcrpReal;
    app.eprpPlantaEditField.Value = app.plaRP_Tceprp;
else
    app.tsPlantaEditField.Value = 0;
    app.TcsRealPlantaEditField.Value = 0;
    app.TcrpRealPlantaEditField.Value = app.plaRP_incTcrpReal;
    app.eprpPlantaEditField.Value = app.plaRP_Tceprp;
end
% Analiza respuesta de datos
[tr,~,~,~,~] = analizaRespuestaDatos(app, app.plaGRA_tiempos, app.plaGRA_incTcReal, Mp);
app.plaRES_tr = tr;
if ~isnan(tr)
    app.trPlantaEditField.Value = tr;
else
    app.trPlantaEditField.Value = 0;
end
drawnow
end

```

Figura 8.35: Función analizaRespuestaPlanta (2/2)

#### 8.2.12. Función analizaRespuestaPID (app)

Esta función es similar a la anterior. Solo se diferencia en que los valores que obtiene los guarda en las variables y en los campos de edición relacionados con el proceso con controlador. Por ejemplo, el valor de tiempo de establecimiento obtenido se almacena en la variable `app.pidEST_ts` en lugar de en `app.plaEST_ts` y se visualiza en el campo de edición `app.tsPIDEditField`, en lugar de en `app.tsPlantaEditField`.

#### 8.2.13. DibujaSoloGraficaPlanta (app)

La planta se dibuja por medio del código `'plot()'`, al cual se le incorpora el lugar donde se dibuja, `app.UIAxes`, `app.plaGRA_tiempos`, `app.plaGRA_TcReal`, el color azul, y es grosor de la línea.

De esta manera, se dibujará la gráfica de la evolución de la temperatura de la línea fría con el tiempo, de color azul.

Se da la orden `hold(app.UIAxes, 'on')` para que no dibuje la gráfica aún, ya que se le deben dar más indicaciones.

Se crea una variable local llamada *'intervalo\_t'*, en el que se almacena el intervalo de tiempo que va a utilizarse para llevar a cabo el dibujo de la gráfica. Esto es, desde cero segundos hasta el valor en segundos guardado en la variable `app.plaGRA_tmax`.

De igual manera, se crea la variable local *'intervalo\_TcReal'* y se almacena en ella el valor deseado de temperatura.

Ahora, se dibuja por medio del código *'plot()'*, al cual se le incorpora el lugar donde se dibuja, `app.UIAxes`, *intervalo\_t*, *intervalo\_TcReal*, el color rojo, y es grosor de la línea.

De esta manera, se dibujará el valor deseado de temperatura en la gráfica, de color rojo.

Se incorpora una leyenda para ambas líneas en el gráfico. También se hace el gráfico visible y se añade una rejilla mediante el comando *grid()*.

Por último, se dibuja un punto ficticio en el lugar  $t=0$  y coordenada  $y=0.95*app.T_{co}$  para que la línea que representa la temperatura deseada se visualice correctamente (no se encuentre en el eje de abscisas).

Se da la orden *hold(app.UIAxes, 'off')*, para que dibuje la gráfica con las especificaciones anteriormente descritas y se visibilizan los ejes y las rejillas.

Adicionalmente, se almacena en las variables correspondientes, información acerca de la gráfica que se encuentra dibujada.

```
function dibujaSoloGraficaPlanta(app)
    % Evolución de la temperatura Tc(t)
    plot(app.UIAxes, app.plaGRA_tiempos, app.plaGRA_TcReal, 'Color', 'blue', 'LineWidth', 1.9);
    hold(app.UIAxes, 'on');
    % Valor deseado (Tco) en rojo
    intervalo_t = [0 app.plaGRA_tmax];
    intervalo_TcReal = [app.Tco app.Tco];
    plot(app.UIAxes, intervalo_t, intervalo_TcReal, 'Color','red', 'LineWidth', 0.6, 'LineStyle','-');
    % Me aseguro de que se verá bien la temperatura deseada
    plot(app.UIAxes, 0, app.Tco*0.95, 'HandleVisibility', 'off');
    title(app.UIAxes, 'TEMPERATURA DE SALIDA EN LA LÍNEA FRÍA');
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Temperatura planta (°C)');
    legend(app.UIAxes, 'Temp. planta (°C)', 'Temp. deseada (°C)');

    app.UIAxes.Visible = "on";
    grid(app.UIAxes, 'on');
    hold(app.UIAxes, 'off');
    drawnow;

    app.plaGRA_dibujada = true;
    app.pidGRA_dibujada = false;
    app.GRA_dibujada = true;

end
```

Figura 8.36: Función `dibujaGraficaPlanta`

8.2.14. Función `dibujaSoloGraficaPID` (app)

Esta función se programa de forma análoga a `dibujaSoloGraficaPlanta(app)`, haciendo uso de las variables correspondientes a la simulación de la planta controlada con PID.

A continuación, se muestra el código utilizado para crear esta función.

```
function dibujaSoloGraficaPID(app)
    % Evolución de la temperatura Tc(t)
    plot(app.UIAxes, app.pidGRA_tiempos, app.pidGRA_TcReal, 'Color', 'blue', 'LineWidth', 1.9);
    hold(app.UIAxes, 'on');
    % Valor deseado en rojo
    intervalo_t = [0 app.pidGRA_tmax];
    intervalo_TcReal = [app.Tco app.Tco];
    plot(app.UIAxes, intervalo_t, intervalo_TcReal, 'Color', 'red', 'LineWidth', 0.6, 'LineStyle', '-');
    % Me aseguro de que se verá bien la temperatura deseada
    plot(app.UIAxes, 0, app.Tco*0.95, 'HandleVisibility', 'off');
    title(app.UIAxes, 'TEMPERATURA DE SALIDA EN LA LÍNEA FRÍA');
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Temperatura PID (°C)');
    legend(app.UIAxes, 'Temp. planta (°C)', 'Temp. deseada (°C)');
    app.UIAxes.Visible = "on";
    grid(app.UIAxes, 'on');
    hold(app.UIAxes, 'off');
    drawnow

    app.plaGRA_dibujada = false;
    app.pidGRA_dibujada = true;
    app.GRA_dibujada = true;
end
```

Figura 8.37: Función `dibujaSoloGraficaPID`

8.2.15. Función `dibujaGraficasPlantayPID`

En esta función se representa las gráficas de la planta sin controlador en líneas discontinuas de color negro y la planta controlada con PID en línea continua azul, además de la temperatura deseada, en línea continua en rojo. Todo ello en una misma gráfica.

```

function dibujaGraficasPlantayPID(app)
    % Al entrar aquí: app.plaGRA_tmax = app.pidGRA_tmax
    % Valor deseado en rojo (línea trazos y puntos)
    intervalo_t = [0 app.pidGRA_tmax];
    intervalo_TcReal = [app.Tcd app.Tcd];
    pValorDeseado = plot(app.UIAxes, intervalo_t, intervalo_TcReal, 'Color','red', 'LineWidth', 0.6,'LineStyle', '-');
    hold(app.UIAxes, 'on');
    % Me aseguro de que se verá bien la temperatura deseada
    plot(app.UIAxes, 0, app.Tcd*0.95, 'HandleVisibility', 'off');
    % Planta en negro (línea a trazos)
    pPlanta = plot(app.UIAxes, app.plaGRA_tiempos, app.plaGRA_incTcReal, 'Color', 'k', 'LineWidth', 1.6, 'LineStyle', '--');
    % PID en azul (línea continua)
    pPID = plot(app.UIAxes, app.pidGRA_tiempos, app.pidGRA_incTcReal, 'Color', 'blue', 'LineWidth', 1.9, 'LineStyle', '-');
    grid(app.UIAxes, 'on');
    title(app.UIAxes, 'TEMPERATURA DE SALIDA EN LA LÍNEA FRÍA');
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Temperatura (°C)');
    legend(app.UIAxes, [pPlanta pPID pValorDeseado], {'Temp. Planta (°C)', 'Temp. PID (°C)', 'Temp. deseada (°C)'});
    legend(app.UIAxes, "show")
    hold (app.UIAxes, 'off');
    app.UIAxes.Visible = "on";
    drawnow;

    app.plaGRA_dibujada = true;
    app.pidGRA_dibujada = true;
    app.plarealGRA_dibujada = false;
    app.GRA_dibujada = true;
end

```

Figura 8.38: Función dibujaGraficasPlantayPID

### 8.3. Resultados y observaciones

En este apartado se va a explicar la forma de operar con el programa. Se supone que, tal como se indica en los campos de edición 'Temp. Fluido refrig. saturado' y 'Temperatura ambiente', estas son -2 °C y 15°C, respectivamente. Esta última es la temperatura del agua de red. También se supone que en la línea fría se ha conseguido llegar al punto de funcionamiento indicado en el campo de edición 'Temperatura línea fría' del panel 'ENTRADAS' de 10°C, junto con los otros datos indicados en la tabla 'PUNTO DE FUNCIONAMIENTO'.

- 1) Ahora, se busca bajar cuatro grados Celsius la temperatura de la línea fría, debido a un sobrecalentamiento en el reactor o en la línea de agua caliente, esto es, se quiere bajar hasta los 6°C, como se indica en el campo de edición 'Temperatura deseada' del panel 'ENTRADAS'. Si se quisiera cambiar esta temperatura, habría que modificar este campo de edición, tecleando su valor y pulsando la tecla 'ENTER', pues en caso contrario no se lograría cambiarlo (esto se debe hacer cuando se desee cambiar cualquier otro valor de los campos de edición)

Desarrollo de una interfaz gráfica de usuario para planta piloto

- 2) Se enciende el interruptor, de forma que la bombilla cambia de color rojo a ámbar mientras esté generándose la gráfica.
- 3) Una vez generada esta gráfica, la lámpara cambia de color ámbar a verde y aparece la gráfica de la temperatura alcanzada en función del tiempo y el análisis de los datos simulados en la columna planta del panel 'SALIDA TEMPERATURA LÍNEA FRÍA'.

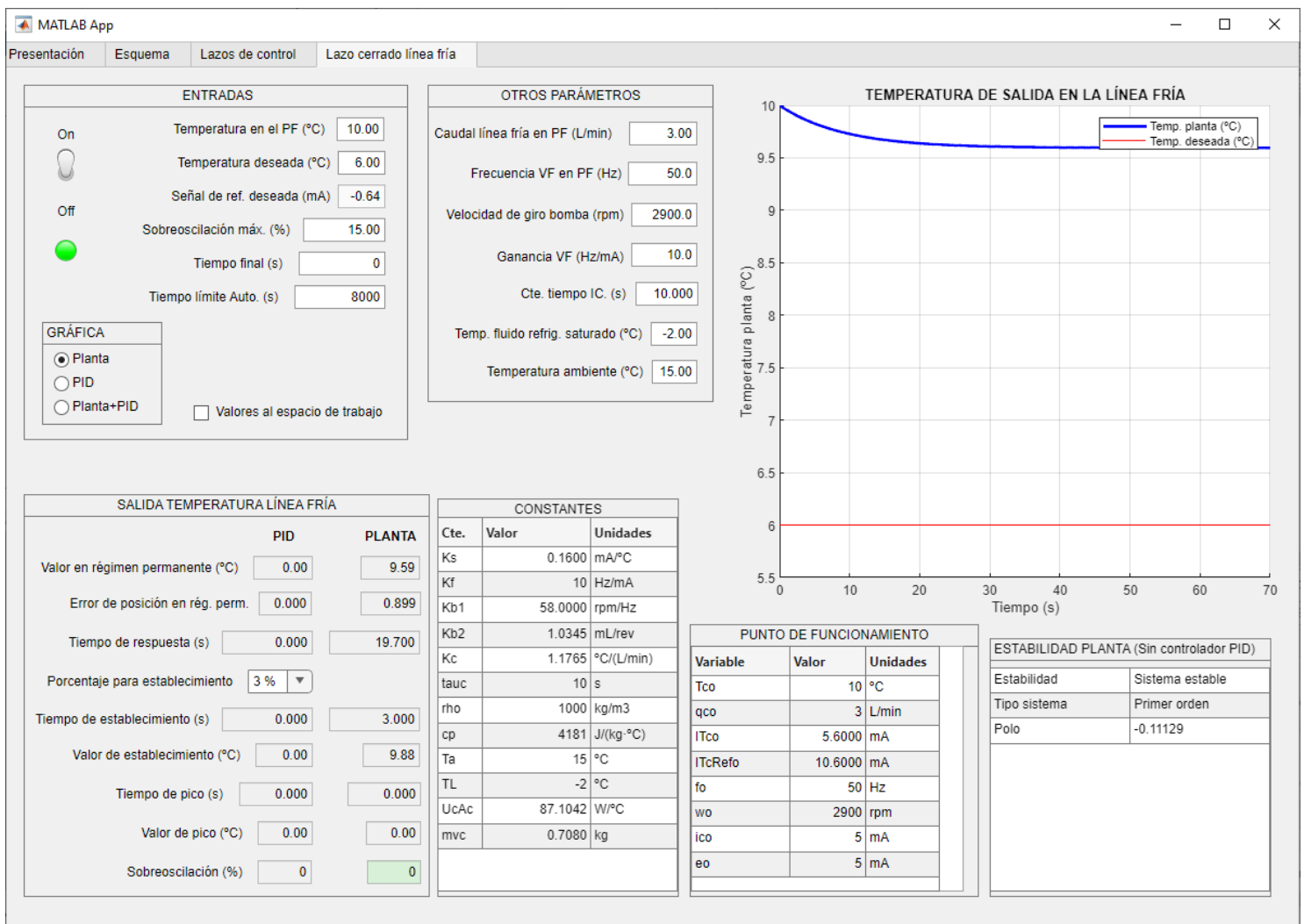


Figura 8.39: Lazo cerrado línea fría 'Planta sin controlador PID'

4) Resultados de la simulación de la planta sin controlador:

➤ Panel ESTABILIDAD PLANTA

Este panel indica que el sistema es estable y de primer orden y que, por tanto, consta de un único polo en el eje real.

➤ Panel SALIDA TEMPERATURA LÍNEA FRÍA

Se observa que no se ha conseguido el valor deseado de 6°C.

En su lugar, se alcanza una temperatura de 9.59°C, con un error de posición en régimen permanente del 89.9%.

El tiempo de establecimiento es de 3 segundos.

5) Una vez analizados estos resultados, se procede a incorporar un controlador PI, porque así, se conseguirá un error de posición en régimen permanente nulo.

En principio, no interesa modificar el valor de la acción proporcional, P, porque no existe sobreoscilación al tratarse de un sistema de primer orden.

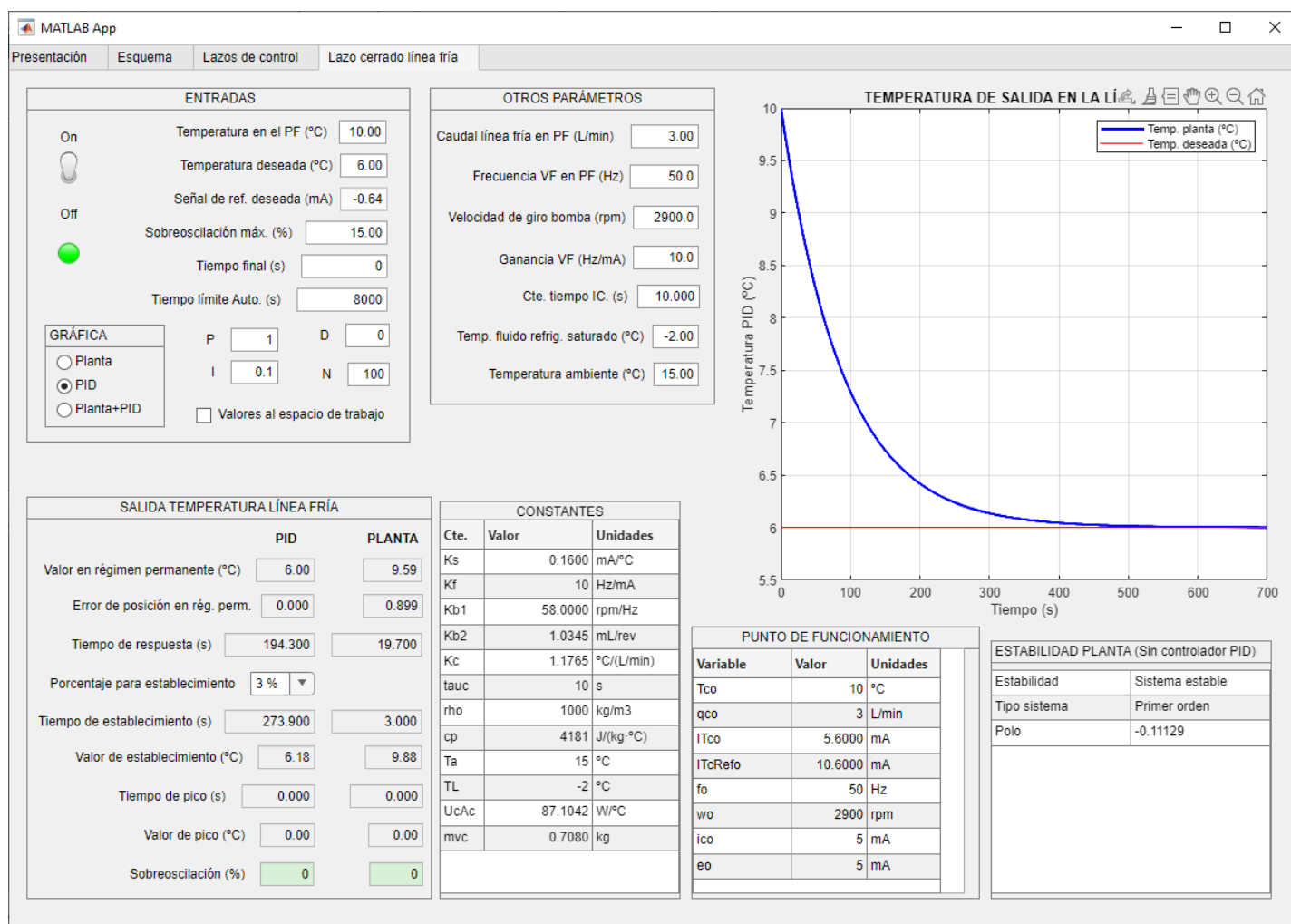
La acción proporcional integral, PI, aumentará la pendiente de la respuesta al escalón en el régimen transitorio.

Se escoge entonces con un controlador PI, en el cual se mantiene la acción proporcional con el valor '1', y se experimenta con el valor de la acción integral, I.

Finalmente, se opta por un controlador PI con acción integral, I = 0.01, definido por defecto, que equivale a un tiempo de integración:

$$T_i = \frac{K_p}{I} = \frac{1}{0.01} \rightarrow T_i = 100 \text{ s}$$

## Desarrollo de una interfaz gráfica de usuario para planta piloto



**Figura 8.40: Lazo cerrado línea fría 'Planta con controlador PI (P=1, I=0.1)'**

### 6) Resultados de la simulación de la planta con controlador PI:

#### ➤ Panel SALIDA TEMPERATURA LÍNEA FRÍA

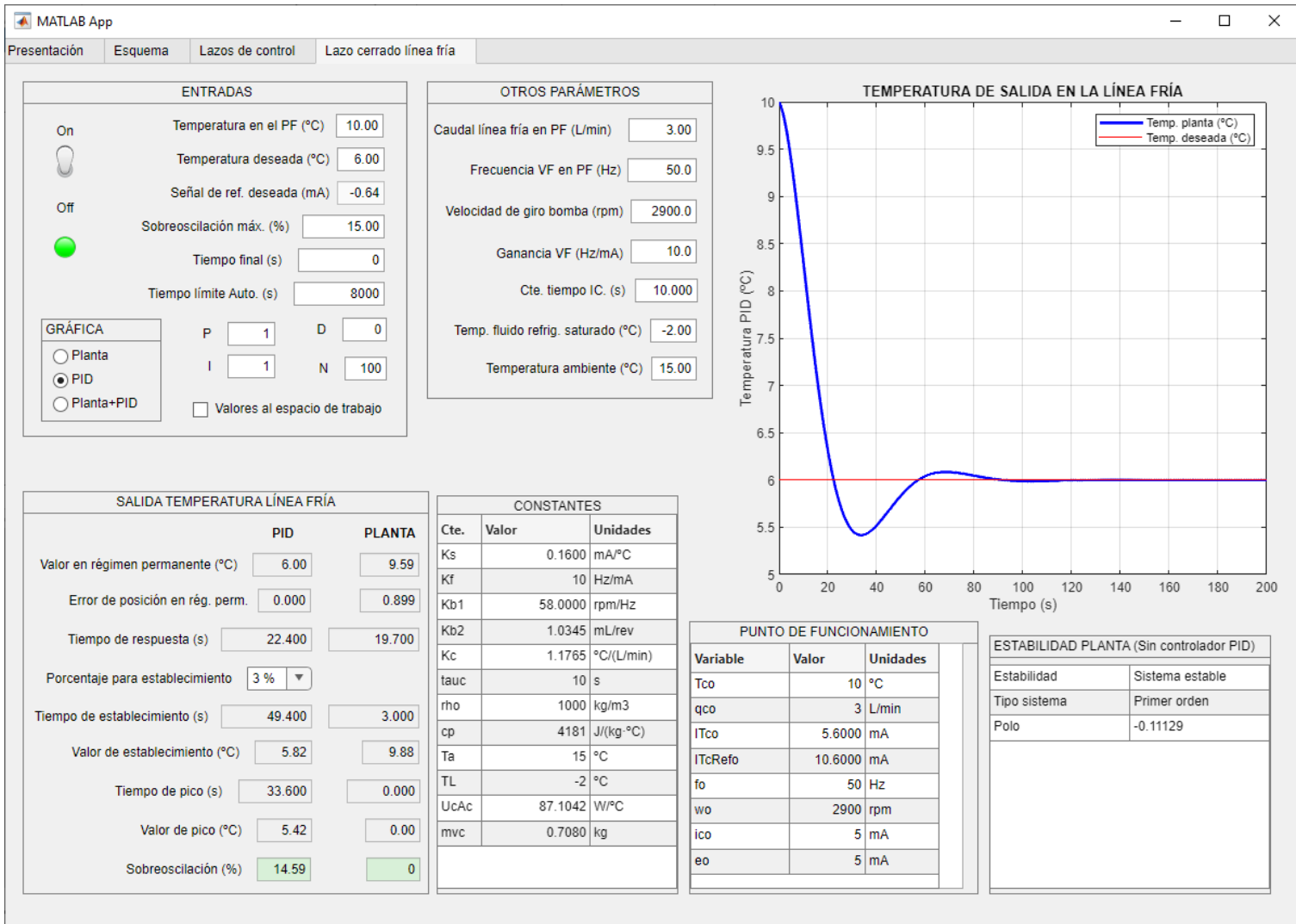
Se observa que se ha logrado alcanzar el valor en régimen permanente deseado de 6°C. Por lo tanto, el error de posición en régimen permanente es nulo. Tampoco hay sobreoscilación, aunque el tiempo de establecimiento ha subido a 273.9 s, que equivale a unos 4.6 min.

### 7) Para tratar de disminuir el tiempo de establecimiento obtenido, se opta por aumentar la acción integral hasta un valor I=1. Con este aumento, se pretende que la pendiente sea mayor y disminuya así el tiempo de establecimiento.

$$T_i = \frac{K_p}{I} = \frac{1}{1} \rightarrow T_i = 1 \text{ s}$$



## Desarrollo de una interfaz gráfica de usuario para planta piloto



**Figura 8.41: Lazo cerrado línea fría 'Planta con controlador PI (P=1, I=1)'**

### 8) Resultados de la simulación de la planta con controlador PI:

Finalmente, se obtiene un tiempo de establecimiento de 49.4 segundos con una sobreoscilación del 14.59%, que se encuentra por debajo del máximo valor aceptado por el usuario en el panel 'ENTRADAS'.

### 9) Para resumir, se selecciona el botón 'Planta+PID' en el panel 'ENTRADAS', y encendiendo de nuevo el interruptor, obtenemos la gráfica que combina la respuesta de la planta sin controlador, con él, y el valor deseado.

## Desarrollo de una interfaz gráfica de usuario para planta piloto

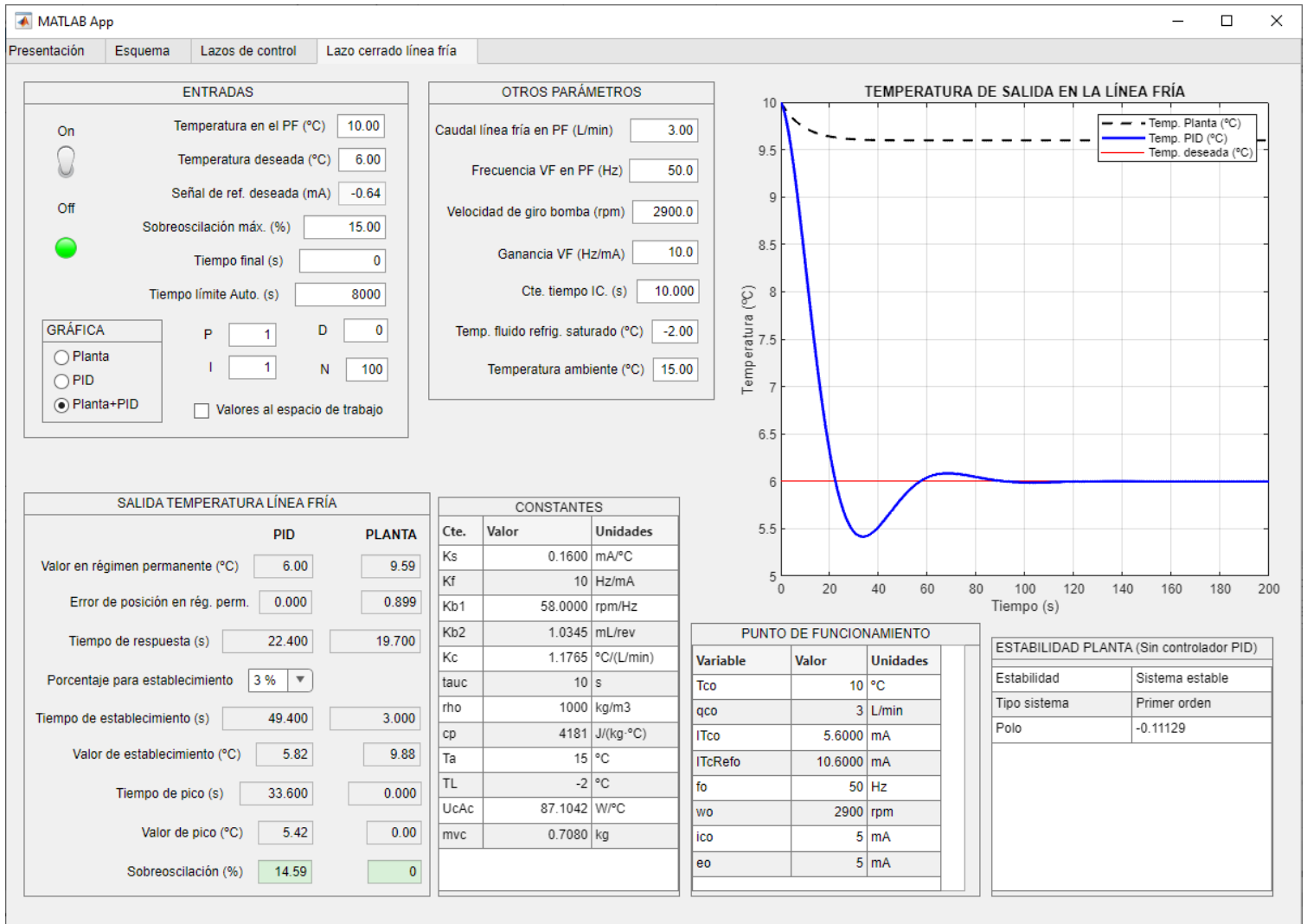


Figura 8.42: Lazo cerrado línea fría. Gráfica Planta sin controlador + Planta con controlador PID (P=1, I=1), combinadas.

Observación: Se recuerda de nuevo que, al modificar cualquier campo de edición, se debe pulsar la tecla 'Enter' tras introducir el valor. De lo contrario, no tendrá en cuenta este cambio, aunque el valor aparezca dentro del campo de edición.

## 9. Conclusiones

La programación de la interfaz gráfica de usuario mediante *App Designer* ha supuesto un refuerzo de los conocimientos adquiridos en el estudio de las asignaturas de 'Regulación Automática' y 'Control e Instrumentación'.

Se ha estudiado el funcionamiento de una planta real, así como los elementos que la componen, favoreciendo una mejor comprensión de una situación práctica.

Se ha logrado entender la relación entre todos los elementos de la planta y su papel en cada uno de los lazos de control propuestos.

Se ha conseguido simplificar el modelo de uno de los lazos de control mencionados para implementarlo en la interfaz, de forma que se pueda estudiar su desempeño para distintas configuraciones. Los resultados obtenidos han mejorado la comprensión de cómo afecta la elección de un punto de funcionamiento y un controlador PID determinado.

En general, se ha logrado el objetivo didáctico en el que se basa el proyecto: crear una interfaz capaz de proporcionar información al usuario, fomentar su interés mediante componentes interactivos y, por último, la experimentación con los valores de los distintos parámetros en el lazo de control para obtener resultados óptimos en la planta. Con todo esto, se favorece que el usuario aprenda de una forma visual y clara todos los aspectos tratados en este proyecto.

Personalmente, la programación de la interfaz ha supuesto un reto dada la complejidad del programa resultante, pero tras mucho esfuerzo y estudio, se ha logrado obtener un buen resultado. Incorporar todo lo aprendido en la creación de un programa ha hecho que profundice mucho más en cada detalle de las asignaturas estudiadas. También se han utilizado conceptos aprendidos en otras asignaturas como 'Termodinámica Aplicada' y 'Transmisión de Calor' para conseguir el modelado del lazo de control propuesto.

Gracias a este proyecto, he logrado comunicar a un futuro usuario todo lo aprendido mediante una interfaz gráfica, lo cual me ha aportado una mejor comprensión de los conceptos tratados en este TFG.

En definitiva, ha sido un acierto aprender a utilizar esta herramienta de apoyo, *App Designer*, junto con *Matlab* y *Simulink*.

## 10. Trabajos futuros

- Desarrollar una aplicación en la cual se utilice un modelo matemático de la línea caliente, análogamente al de la línea fría. Asimismo, se puede incorporar el control de la temperatura en la entrada a la camisa y la entrada al reactor. Cada uno de estos lazos pueden constituir un subproyecto, o bien, englobar uno completo.
- Se podría incorporar un análisis de la estabilidad, polos, etc, cuando se utiliza un controlador PID, ya que este proyecto no lo contempla.
- Incorporar a la guía informativa, por ejemplo, en la pestaña 'Esquema' algún componente que muestre cada elemento mediante un vídeo corto. De esta forma el usuario puede tener una mejor visualización de la planta piloto.
- Añadir a la memoria del proyecto más estudios de posibles lazos de control en la planta piloto.

## 11. Referencias

- [SHA]:

Fundamentos de Termodinámica Técnica, 2º edición. Michael J. Moran y Howard N.Shapiro.

- [PRIS 1]:

URL: <https://www.espa.com/corp-es/prisma-25/?recursos>

y descargando 'catálogo-espa-corporativo-2024-50hz-esp.pdf', se encuentran las especificaciones de la bomba Prisma 15.

(05/02/2024)

- [JMG]:

TFG: 'Control de la temperatura del reactor de la planta piloto'. Septiembre 2008. José Mora González. Página 19.

- [MCB]

TFG: 'Instalación de instrumentación en planta piloto. Control de nivel de los tanques de alimentación'. Octubre 2006. Manuel Crespo Ballesteros. Página 23.

- [TET]

URL: The Engineering ToolBox (2010). Hydraulic Pump Motor Sizing online.

[https://www.engineeringtoolbox.com/hydraulic-pumps-d\\_1628.html](https://www.engineeringtoolbox.com/hydraulic-pumps-d_1628.html)

(05/02/2024)

- [HXR]

URL: [https://www.johncrane.com/-/media/j/johncrane\\_com/files/products/technical-specification/seal-systems/td-hxr-2pg-bw-oct2015.pdf](https://www.johncrane.com/-/media/j/johncrane_com/files/products/technical-specification/seal-systems/td-hxr-2pg-bw-oct2015.pdf)

(05/02/2024)

## 12. Bibliografía

- Ingeniería de control moderna. Katsuhiko Ogata. Pearson
- Apuntes sobre “representación matemática de sistemas” NEU-REG-0015R0. José Luis Muños Lozano.
- Apuntes de identificación NEU-REG-013R0. José Luis Muñoz Lozano.
- Apuntes de sensores y actuadores NEU-REG-012R0. José Luis Muñoz Lozano.
- Apuntes sobre diseño de PID NEU-REG-010R0. José Luis Muñoz Lozano.
- TFG: ‘Control de la temperatura del reactor de la planta piloto’. Septiembre 2008. José Mora González.
- TFG: ‘Instalación de instrumentación en planta piloto. Control de nivel de los tanques de alimentación’. Octubre 2006. Manuel Crespo Ballesteros.
- Fundamentos de Termodinámica Técnica, 2º edición. Michael J.Moran y Howard N.Shapiro.
- URL:  
<https://es.mathworks.com>  
(01/07/2023)
- URL:  
<https://controlautomaticoeducacion.com>  
(01/07/2023)
- URL  
<https://www.youtube.com/@MATLABenEspanol>  
(01/07/2023)
- URL:  
[https://www.youtube.com/channel/UCdzSnI03LpBI\\_8gXJseIDuw](https://www.youtube.com/channel/UCdzSnI03LpBI_8gXJseIDuw)  
(01/07/2023)

- URL:  
[https://www.johncrane.com/-/media/j/johncrane\\_com/files/products/technical-specification/seal-systems/td-hxr-2pg-bw-oct2015.pdf](https://www.johncrane.com/-/media/j/johncrane_com/files/products/technical-specification/seal-systems/td-hxr-2pg-bw-oct2015.pdf)  
(05/02/2024)
- URL: The Engineering ToolBox (2010). Hydraulic Pump Motor Sizing online.  
[https://www.engineeringtoolbox.com/hydraulic-pumps-d\\_1628.html](https://www.engineeringtoolbox.com/hydraulic-pumps-d_1628.html)  
(05/02/2024)
- URL:  
<https://www.espa.com/corp-es/prisma-25/?recursos>  
(05/02/2024)