



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**PROYECTO DE PUESTA EN MARCHA DE
UN SISTEMA DE CONTROL CONTINUO
EN AUTÓMATAS S7 1200 APLICADO AL
CONTROL EN TIEMPO REAL DE
PROCESOS QUÍMICOS VIRTUALES
IMPLEMENTADOS EN LABVIEW**

AUTOR: DANIEL ROMÁN BOLTA

TUTOR: JOSÉ VICENTE SALCEDO ROMERO DE ÁVILA

COTUTOR: ADOLFO HILARIO CABALLERO

Curso Académico: 2018-19

RESUMEN

En este trabajo se presenta la puesta en marcha de un sistema de control en tiempo real en autómatas S7 1200 aplicado a procesos químicos virtuales simulados en labVIEW. Con esto se persigue desarrollar una plataforma que nos permita simular un proceso real sin necesidad de tener el proceso físico, con lo que no sería necesario realizar grandes inversiones de capital para comprobar la viabilidad del proyecto.

Los procesos químicos estarán en labVIEW el cual presenta una interfaz capaz de mostrar en tiempo real los resultados de la simulación al usuario para su posterior interpretación.

Por otro lado, se diseñará los controladores que vendrá implementados en un PLC Siemens S7 programado mediante la herramienta Tia Portal con la que obtendremos la acción de control necesaria para controlar los procesos.

El flujo de datos entre ambas aplicaciones se gestionará mediante el NI Server OPC disponible dentro del labVIEW con el que se intercambiarán información el autómata y el ordenador.

Para finalizar se facilitará una plantilla para que se pueda efectuar la simulación de cualquier proceso y la implementación de cualquier tipo de controlador utilizando las herramientas facilitadas en este trabajo.

RESUM

En este treball es presenta la posada en marxa d'un sistema de control en temps real en autòmats S7 1200 aplicat a processos químics virtuals simulats en labVIEW. Amb açò es persegueix desenrotllar una plataforma que ens permeta simular un procés real sense necessitat de tindre el procés físic, amb la qual cosa no seria necessari realitzar grans inversions de capital per a comprovar la viabilitat del projecte.

Els processos químics estaran en labVIEW el qual presenta una interfície capaç de mostrar en temps real els resultats de la simulació a l'usuari per a la seua posterior interpretació.

D'altra banda, es dissenyarà els controladors que vindrà implementats en un PLC Siemens S7 programat per mitjà de la ferramenta Tia Portal amb què obtindrem l'acció de control necessària per a controlar els processos.

El flux de dades entre ambdós aplicacions es gestionarà per mitjà del NI Server OPC disponible dins del labVIEW amb el que s'intercanviaran informació l'autòmat i l'ordinador.

Per a finalitzar es facilitarà una plantilla perquè es puga efectuar la simulació de qualsevol procés i la implementació de qualsevol tipus de controlador utilitzant les ferramentes facilitades en este treball.

ABSTRACT

This paper presents the implementation of a real-time control system in S7 1200 automats applied to virtual chemical processes simulated in LabVIEW. This project seeks to develop a platform that allows us to simulate a real process without having to have the physical process, so it would not be necessary to make large capital investments to verify the viability of the project.

The chemical processes will be in LabVIEW which presents an interface capable of displaying in real time the results of the simulation to the user for later interpretation.

On the other hand, the controllers that will be implemented in a Siemens S7 PLC programmed with Tia Portal tool which we will obtain the necessary control action to control the processes.

The data flow between both applications will be managed through the NI Server OPC available within the LabVIEW with which the automaton and the computer will exchange information.

Finally, a template will be provided so that the simulation of any process and the implementation of any type of controller can be carried out using the tools provided in this work.

INDICE GENERAL DEL TRABAJO

MEMORIA	11
PRESUPUESTO	73
PLIEGO DE CONDICIONES	79
ANEXO	83

ÍNDICE DE LA MEMORIA

1. INTRODUCCIÓN	13
1.1 Antecedentes	13
1.2 Motivación	14
1.3 Objetivos	15
2. MODELADO DE LA PLANTA	17
2.1 Descripción de la planta	17
2.2 Modelo Matemático	21
2.3 Discretización del modelo no lineal.....	28
3. DISEÑO DE LOS REGULADORES	33
3.1 Controlador del reactor	33
3.2 Controlador de los actuadores.....	41
4. IMPLEMENTACIÓN DE LOS PROGRAMAS EN TIA PORTAL	47
4.1Válvula Camisa	48
4.2 Válvula refrigeración.....	51
4.3 Reactor.....	53
5. VIRTUALIZAR LA PLANTA	57
5.1 Diseño planta con labVIEW	57
5.2 NI Servidor OPC.....	59
6. PLANTILLAS	61
6.1 Elaboración de la plantilla labVIEW	61
6.2 Elaboración de la plantilla Tia Portal.....	63
7. EVALUACIÓN DE RESULTADOS	65
7.1 Conclusiones.....	65
BIBLIOGRAFIA.....	71

ÍNDICE DEL PRESUPUESTO

1. CUADRO DE PRECIOS N.º 1: MANO DE OBRA	74
2. CUADRO DE PRECIOS N.º 2.....	74
2.1. MATERIALES.....	74
2.2. MAQUINARIA.....	74
3. CUADRO DE PRECIOS N.º 3: PRECIOS PARCIALES.....	75
4. CUADRO DE PRECIOS N.º 4: PRECIOS DESCOMPUESTOS.	76
5. PRECIOS FINALES.....	78

ÍNDICE DEL PLIEGO DE CONDICIONES

Características del material empleado	80
Normativa	81

ÍNDICE DEL ANEXO

LabVIEW	84
Formula Node	85
Variables del sistema	88
TIA PORTAL.....	92
Bloques de programa.....	93
Tabla de Variables	96
NI Servidor OPC.....	97

MEMORIA

INTRODUCCIÓN

En este primer capítulo se explican las líneas generales que se han seguido para la realización del trabajo: antecedentes, motivación y objetivos a cumplir, así como las herramientas utilizadas, tanto en el terreno industrial como puramente matemático para la realización del estudio. Dado que el resto de los capítulos consistirán fundamentalmente en solucionar los problemas técnicos planteados, sirva este capítulo introductorio como justificación y argumentación para la realización de este.

1.1 Antecedentes

Hoy en día la sociedad se rige fundamentalmente por criterios económicos, luego buscar procedimientos que nos garanticen la viabilidad económica de un proyecto antes de su ejecución es una herramienta que, de manera evidente, demanda la sociedad actualmente. Este proyecto pretende formalizar esta idea proveyendo a la industria de una herramienta que le permita visualizar con premeditación los costes y beneficios de la implantación de un nuevo proceso lo cual le proporciona una ventaja fundamental en la toma de decisiones con la consiguiente optimización de recursos tanto económicos como humanos.

1.2 Motivación

La principal motivación de este Trabajo Final de grado será aprender cómo resolver una de las infinitas aplicaciones que nos ofrece la comunicación por servidor OPC. Los proyectos de automatización no solo comparten el objetivo de mejorar la productividad y la eficiencia de las Plantas, otro rasgo en común que no conoce fronteras ni sectores industriales, es el protocolo de comunicación que utilizan para lograrlo.

El control por comunicación OPC trata de un protocolo estándar de comunicación en el campo de control y supervisión de procesos industriales que es usado prácticamente por todas las industrias del mundo. Consiste en un software de conectividad que hace posible que muchas máquinas y sistemas de control puedan hablarse entre sí y compartir información, sin importar la marca, el fabricante o la tecnología.

Por otra parte, aprovecho este trabajo para introducirme en un mundo que me apasiona y en el que me gustaría dedicarme en un futuro como es la automática, dando mis primeros pasos fuera del aula y descubriendo de primera mano la capital importancia que tiene este campo en la industria actual, ya que hoy en día la inmensa mayoría de los proyectos tienen como objetivo automatizar procesos para poder competir con garantías de éxito en el mercado. Esto me ha impulsado a investigar sobre un modo de hacer este proceso más efectivo en un campo tan influenciado por la automática como son los procesos químicos.

1.3 Objetivos

En primer lugar, dada mi etapa de formación en la que me encuentro el principal objetivo es aprender, desarrollarme en todos los campos que me sea posible, aplicando conocimientos estudiados y complementarlos con herramientas nuevas como el LabVIEW el cual me permite simular controlar y gestionar procesos industriales, el hecho de tener en mi monitor un proceso industrial complejo simulado de forma que pueda modificarlo controlarlo y estudiarlo sin necesidad de tener la planta física me parece apasionante.

Por otro lado, como ya se ha comentado en el punto 1.2, el principal objetivo ha sido estudiar y utilizar la comunicación OPC como forma de conectar un autómatas Siemens con un programa de implementación de aplicaciones como es el LabVIEW

Para crear los controladores, simularlos y analizarlos he tenido que profundizar en la herramienta Matlab con la que he descubierto la infinidad de posibilidades que te ofrece este programa, como por ejemplo el Simulink herramienta que me permite comparar resultados, elaborar PID e infinidad de opciones que han sido vitales en el desarrollo de este trabajo.

De este modo el objetivo aquí tratado es simular, en tiempo real, un proceso industrial sin necesidad de tenerlo en físico, este hecho nos da la posibilidad de estudiar el método de control más eficiente, los elementos que compondrá la planta, así como poder hacer todos los cambios que veamos oportunos sin necesidad de hacer más inversión económica que las licencias de los programas que vayamos a utilizar. Para ello recrearemos nuestro sistema en Simulink de manera que el resultado obtenido con LabVIEW y Tia Portal ha de ser equivalente a este, de esta forma determinaremos que sí es posible controlar una planta real con los programas utilizados.

A continuación, se utilizará un esquema que nos muestre el principal flujo de información entre programas con el fin de esbozar a grosso modo el funcionamiento de estos.

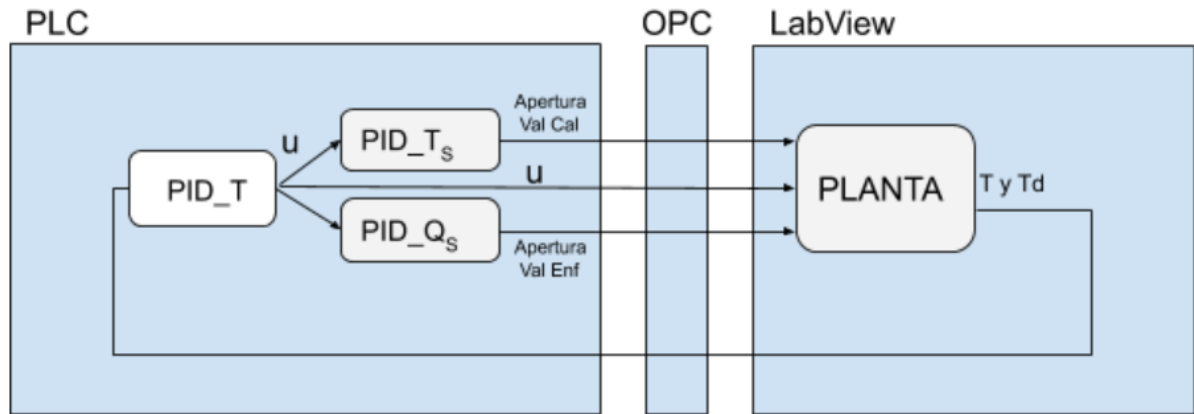


Ilustración 1 elaboración propia

MODELADO DE LA PLANTA

En el presente capítulo se presentará el proceso industrial que se ha elegido, se explicará su funcionamiento y se hallarán las ecuaciones que definen el proceso y las diferentes formas de representar estas ecuaciones. Tendrá un alto contenido de carga matemática.

El objetivo principal de este capítulo será obtener las ecuaciones necesarias para la planta virtualizada con la que realizar diseños y ensayos para obtener el proceso automatizado

2.1 Descripción de la planta

Cada vez con mayor frecuencia, la automatización de los procesos industriales está siendo más necesarios, hasta el punto de que la mayoría de los proyectos en los que están inmersos las empresas es precisamente esto, automatizar procesos, hacerlos más eficientes con el fin de ahorrar costes, por ello nos hemos centrado en un sector muy importante, los procesos químicos, queríamos saber cómo sería automatizar un proceso químico. Se ha decidido virtualizar un reactor en el que se dé una reacción de la especie A a la especie B.

El reactor que se utiliza es un “Batch reactor”, descrito por Jana A. K. (2011) en el libro *Chemical Process Modeling and Computer Simulating*, es muy importante en industrias químicas, su principal objetivo es mantener la temperatura de la cámara a una temperatura deseada durante un tiempo definido para obtener la mayor cantidad de la especie B posible.

Se procederá a explicar cómo funciona este reactor y sacaremos las ecuaciones que nos permitan controlar la temperatura de este, así como controlar en todo momento cuánta cantidad de A y de B se encuentra en el reactor.

Este será nuestro reactor:

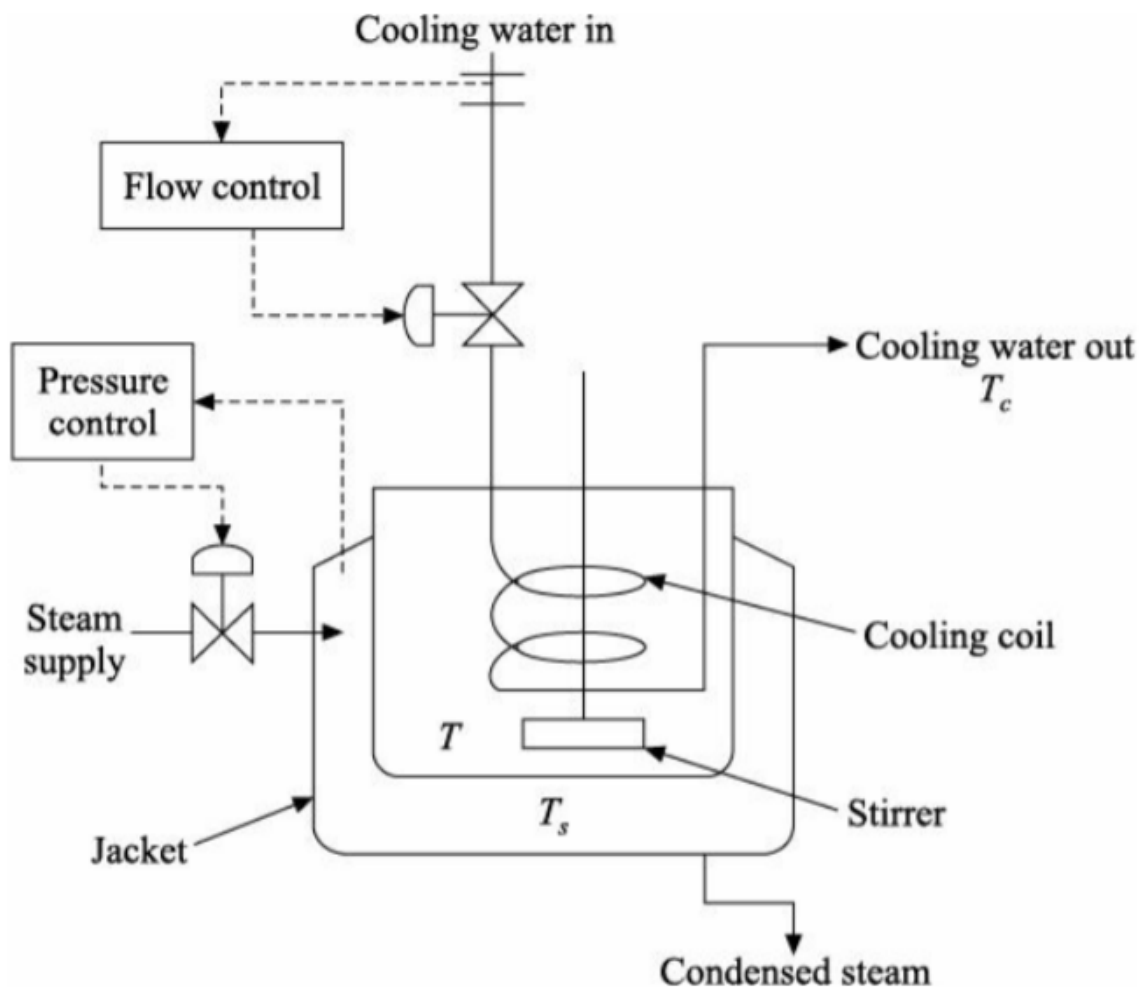


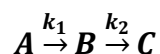
Ilustración 2 imagen obtenida del libro *Chemical Process Modelling and Computer Simulation*/Amiya. , Jana A. K. (2011)

El reactor se llena del reactante y se sella. El contenido de este reactor se calienta con el vapor saturado que circula por la camisa Ts controlada por una válvula que controlará el caudal de vapor de entrada a la camisa. Dentro del reactor hay un agitador a unas rp fijas. El reactor tiene una bobina refrigerada a una temperatura Tc fija, controlando el caudal también por una válvula, para eliminar el calor exotérmico y así dirigir el sistema de temperaturas por una trayectoria Temperatura-tiempo deseadas.

Conviene explicar las diferentes fases de un reactor Batch exotérmico:

1. Fase inicial: Elevar la temperatura del reactor mediante el uso de vapor hasta la temperatura predefinida de reacción. En este caso, en los primeros compases de la reacción, el reactor se encuentra a Temperatura ambiente por lo que nos conviene elevar su temperatura lo antes posible, dentro de los límites de seguridad.
2. Reacción: Mantener la temperatura mediante el uso de agua de refrigeración mientras se produce la reacción y se genera vapor.
3. Digestión: Mantener la temperatura mediante el sistema de refrigeración o vapor y así conseguir en el tiempo deseado la cantidad de reactante óptimo.

Consideraremos la siguiente serie de reacciones



asumiendo que la reacción de A a B es de segundo orden mientras que de B a C es de primer orden. Donde B es el producto que queremos obtener.

Se nos plantea el siguiente problema: en el caso de que la reacción se ejecute durante demasiado tiempo tendremos exceso de excedentes C mientras que si paramos la reacción demasiado pronto la cantidad de B será menor a la deseada. Entonces habrá un tiempo óptimo en el que podamos parar la reacción para conseguir la máxima cantidad de B. Este tiempo al que llegaremos gracias a la Temperatura deseada Td.

Datos que conocer:

- $C_{A0}=1.0 \text{ kmol/m}^3$
- $C_{B0}=1.0 \text{ kmol/m}^3$
- $E_1=2.09 \times 10^4 \text{ KJ/kmol}$
- $E_2=2.09 \times 10^4 \text{ KJ/kmol}$
- $C_p=1.0 \text{ kJ/(kg)(}^\circ\text{C)}$
- $\rho=1000 \text{ kg/m}^3$
- $T_c =25^\circ\text{C}$
- $T_{\text{max}}=125^\circ\text{C}$
- $T_{\text{min}}=25^\circ\text{C}$
- $T_{s,\text{max}}=150^\circ\text{C}$
- $T_{s,\text{min}}=70^\circ\text{C}$
- $U_{c,\text{max}}=4.42 \text{ kJ/(m}^2\text{)(}^\circ\text{C)(s)}$
- $U_{c,\text{min}}=1.39 \text{ kJ/(m}^2\text{)(}^\circ\text{C)(s)}$
- $U_j=1.16 \text{ kJ/(m}^2\text{)(}^\circ\text{C)(s)}$
- $R=8.314 \text{ kJ/(kmol)(K)}$
- $A_c /V = 17.0 \text{ m}^2 /\text{m}^3$
- $A_j /V = 30.0 \text{ m}^2 /\text{m}^3$
- $\alpha_{10}=1.1 \text{ m}^3 /(\text{kmol)(s)}$
- $\alpha_{20}=172.2$
- $\Delta H_1=4.18 \times 10^4 \text{ kJ/kmol}$
- $\Delta H_2=8.36 \times 10^4 \text{ kJ/kmol}$

2.2 Modelo Matemático

Antes de empezar con el modelo matemático se harán una serie de suposiciones:

- Las dos reacciones que hemos nombrado anteriormente son exotérmicas.
- Suponemos el reactor perfectamente aislado de manera que no existe intercambio de calor con el exterior
- Las temperaturas del refrigerante de entrada y salida no varían mucho debido a un caudal de agua suficientemente rápido. de este se utilizará la temperatura media, T_c en el modelo matemático
- El vapor condensado se descarga a su temperatura de saturación (sin subenfriamiento)
- Se considera que el Volumen del reactor es constante V así como el calor específico CP . (Nota, con Volumen nos referimos al volumen de la mezcla)
- El contenido del reactor se supone perfectamente mezclado y sin gradiente de temperaturas.
- El coeficiente de transferencia de calor global para la camisa de calentamiento U_j se asume constante pero el coeficiente de transferencia de la bobina se asume que es función del caudal del refrigerante Q mediante la siguiente función:

$$\frac{1}{U_c} = \frac{1}{4550Q^{0.8}} + \frac{1}{10.8} \quad (2.1)$$

DESARROLLO DEL MODELO

Continuaremos desarrollando las ecuaciones necesarias para virtualizar el proceso. Se empezará describiendo un balance básico:

[Tasa de acumulación dentro del sistema.] = [Ratio de flujo del sistema] - [Ratio de flujo fuera del sistema] + [Ratio de generación en la reacción química en el sistema]

o más simple:

[Ratio de acumulación dentro del sistema] = [flujo neto en el sistema] + [Ratio de generación en la reacción química en el sistema]

En el sistema internacional las ecuaciones anteriores se escriben en kmol/s o kg/s para el balance de masas y kJ/s para el balance de energía.

Ecuación de la continuidad

Dentro del reactor hemos de saber que el flujo neto es 0, y el Ratio de generación de masa en el reactor también vale 0, mientras que el ratio de acumulación de masa en el reactor es $\frac{d(\rho V)}{dt}$. Siendo ρ la densidad de la mezcla y V el volumen de esta, considerándolos ambos constantes

$$\frac{d(\rho V)}{dt} = 0 \quad (2.2)$$

Ecuaciones de continuidad de los componentes

Para el componente A

En el balance de especies del componente A, el flujo neto dentro del reactor del componente A es 0, ratio de generación de componente A por reacción química será = $-(-r_A)V$ o lo que es lo mismo = $(r_A)V$

Además, el ratio de acumulación del componente A en el reactor = $\frac{dN_A}{dt}$

Por convención, $-r_A$ es el ratio de desaparición de especies A, consecuentemente r_A es el ratio de formación de A, aquí el N_A son el número de moles en el sistema en el tiempo t , con lo cual la ecuación de generación de A se quedaría tal que así:

$$\frac{dN_A}{dt} = -(-r_A)V \quad (2.3)$$

esto es

$$\frac{1}{V} \frac{dN_A}{dt} = \frac{d(N_A/V)}{dt} = \frac{dC_A}{dt} = -(-r_A) \quad (2.4)$$

donde C_A es la concentración de especies en A. Como la reacción $A \xrightarrow{k_1} B$ es una ecuación de segundo grado asíque:

$$(-r_A) = k_1 C_A^2 \quad (2.5)$$

donde k_1 representa el ratio de reacción constante.

$$\frac{dC_A}{dt} = -k_1 C_A^2 \quad (2.6)$$

Esta es la ecuación de continuidad de la especie A.

Para el componente B

En el balance de especies del componente B

El flujo neto dentro del reactor del componente B es 0,

Ratio de generación de componente A por reacción química será $= -(-r_B)V$ o lo que es lo mismo $= (r_B)V$

Además el ratio de acumulación del componente B en el reactor $= \frac{dN_B}{dt}$

Por convención, $-r_B$ es el ratio de desaparición de especies B, consecuentemente r_A es el ratio de formación de B, aquí el N_B son los números de moles en el sistema al tiempo t , con lo cual la ecuación de generación de B se quedaría tal que así:

$$\frac{dN_B}{dt} = -(-r_B)V \quad (2.7)$$

como hemos dicho antes

$$(-r_B) = k_2 C_B - k_1 C_A^2 \quad (2.8)$$

aquí C_B es la concentración de especies en B y k_2 es la constante de la reacción, sustituyendo hallamos que:

$$\frac{dC_B}{dt} = k_1 C_A^2 - k_2 C_B \quad (2.9)$$

Esta es la ecuación de continuidad de la especie B.

La temperatura es importante tenerla en cuenta porque bien es sabido que a mayor temperatura más rápido se hará la reacción, esta relación la obtendremos de la ecuación de Arrhenius la cual comprueba la relación entre la constante de velocidad de la reacción con la Temperatura a la que está expuesta:

$$k = \alpha e^{\left(\frac{-E}{RT}\right)} \quad (2.10)$$

donde E es la energía de activación(J/mol) la R la constante universal de los gases (8,3143 J·K⁻¹·mol⁻¹), y α , factor preexponencial o factor de frecuencia, indica la frecuencia de las colisiones, T temperatura absoluta en Kelvin y k la constante cinética (dependiente de la temperatura)

$$k_1 = \alpha_{10} e^{\left(\frac{-E_1}{RT}\right)} \quad (2.11)$$

$$k_2 = \alpha_{20} e^{\left(\frac{-E_2}{RT}\right)} \quad (2.12)$$

Balance energético:

Ratio de acumulación de energía = Ratio de entrada de energía + Ratio de energía añadida por reacciones químicas exotérmicas

o lo que es lo mismo,

Flujo neto de entrada de energía= Ratio de entrada de calor por la camisa - Ratio de salida de calor del reactor al serpentín de refrigeración

$$\text{Flujo neto de entrada energía} = U_j A_j (T_s - T) - U_c A_c (T - T_c) \quad (2.13)$$

Aquí A_j y A_c son superficies de intercambio de calor de la camisa y de las bobinas. Antes se ha asumido que el reactor está bien aislado con lo que no hay intercambio de calor entre el proceso y el ambiente.

El ratio de energía añadido por reacción exotérmica es:

$$\begin{aligned} \text{Ratio de energía añadido por la reacción} &= \sum_{i=1}^n (-\Delta H_i) (-r_i) V = \\ &= (-\Delta H_1) k_1 C_A^2 V + (-\Delta H_2) k_2 C_B V \end{aligned} \quad (2.14)$$

Donde n=2, número de reacciones que tenemos. En esta ecuación ΔH_1 y 2 son el incremento de entalpía de las dos reacciones.

Ahora el ratio de acumulación de energía es:

$$\frac{d(\rho V h)}{dt} = \rho V C_p \frac{dT}{dt} \quad (2.15)$$

Aquí la densidad ρ V y C_p son constantes, y h es la entalpía ($=C_p T$). Sustituyendo todos los términos en esta ecuación se obtiene que:

$$\rho C_p V \frac{dT}{dt} = (-\Delta H_1) k_1 C_A^2 V + (-\Delta H_2) k_2 C_B V + U_j A_j (T_s - T) - U_c A_c (T - T_c) \quad (2.16)$$

Reordenando términos:

$$\frac{dT}{dt} = \frac{(-\Delta H_1)}{\rho C_p} k_1 C_A^2 + \frac{(-\Delta H_2)}{\rho C_p} k_2 C_B + \frac{U_j A_j}{\rho C_p V} (T_s - T) - \frac{U_c A_c}{\rho C_p V} (T - T_c) \quad (2.17)$$

Esta es la ecuación de la energía para nuestro reactor.

Se pretende obtener una acción de control sobre este proceso. Para empezar, se quiere obtener la máxima cantidad de producto deseado, para ello debemos tener una ecuación que nos permita encontrar la temperatura perfecta.

Vamos a utilizar la ecuación que relaciona T_d (temperatura deseada) con el tiempo t de "KRAVARIS AND CHUNG, 1987" de esta forma conseguiremos en la primera etapa aumentar la temperatura del reactor todo lo posible y a medida que pase el tiempo ir disminuyendo hasta alcanzar la Temperatura de establecimiento 54°C .

$$T_d(t) = 54 + 71e^{(-2.5 \times 10^{-3} t)} \quad (2.18)$$

Ahora se nos plantea un problema, ya que tenemos una variable que queremos controlar, en nuestro caso T la cual se puede manipular tanto con el serpentín de refrigeración U_c como con la temperatura de camisa T_s , si la controláramos con ambas el sistema estaría sobre determinado. Una manera de solventar este problema sería relacionarlas con un parámetro u definido de la siguiente manera:

$$T_s = (T_{s,max} - T_{s,min})u + T_{s,min} \quad (2.19)$$

$$U_c = (U_{c,min} - U_{c,max})u + U_{c,max} \quad (2.20)$$

Donde Tmax y Tmin se elegirán de procesos con límites seguros. Claramente la u=0 representa que el sistema tendría máxima refrigeración o u=1 para máximo calor. Este parámetro se usará como variable manipulada.

Vamos a reescribir la ecuación de la energía:

$$\frac{dT}{dt} = \gamma_1 k_1 C_A^2 + \gamma_2 k_2 C_B + (A_1 + A_2 T) + (B_1 + B_2 T)u \quad (2.21)$$

donde:

$$\gamma_1 = \frac{(-\Delta H_1)}{\rho C_p}$$

$$\gamma_2 = \frac{(-\Delta H_2)}{\rho C_p}$$

$$A_1 = (U_j A_j T_{s,min} + U_{c,max} A_c T_c) / (\rho C_p V)$$

$$A_2 = -(U_j A_j + U_{c,max} A_c) / (\rho C_p V)$$

$$B_1 = (U_j A_j (T_{s,max} - T_{s,min}) - (U_{c,max} - U_{c,min}) A_c T_c) / (\rho C_p V)$$

$$B_2 = ((U_{c,max} - U_{c,min}) A_c) / (\rho C_p V)$$

Modelo matemático no lineal:

- $\frac{dT}{dt} = \gamma_1 k_1 C_A^2 + \gamma_2 k_2 C_B + (A_1 + A_2 T) + (B_1 + B_2 T)u$
- $\frac{dC_B}{dt} = k_1 C_A^2 - k_2 C_B$
- $\frac{dC_A}{dt} = -k_1 C_A^2$

2.3 Discretización del modelo no lineal

Una vez obtenido el modelo matemático se precisa simular la planta en LabView utilizamos esta plataforma ya que nos permite diseñar sistemas, en nuestro caso un reactor, con un lenguaje de programación visual gráfico y atractivo para el consumidor. Con este programa conseguiremos simular el proceso industrial de un reactor para observar su comportamiento, estudiarlo y mejorarlo sin necesidad de hacer una gran inversión de capital.

Para ello se utilizan las ecuaciones no lineales de nuestro sistema y las discretizaremos. El método que utilizaremos será Euler, es un procedimiento de integración numérica para resolver ecuaciones diferenciales ordinarias (EDO) a partir de un valor inicial dado.

Tomaremos los siguientes puntos iniciales:

- La C_A , concentración en A, es el producto inicial con el que se ha partido en la reacción por consiguiente su valor ha de ser 1 y a medida que avance el tiempo de la reacción irá disminuyendo
- El C_B es la concentración de producto B es decir el producto de la reacción, por consiguiente, en el instante inicial valdrá 0, es decir todavía no se ha transformado nada de A en B.
- Para la Temperatura del reactor tomaremos 25°C , es decir, temperatura ambiente.

Condiciones iniciales: $T_0 = 25^\circ\text{C}$, $C_{A0} = 1$, $C_{B0} = 0$.

Euler:

Euler proporciona un método sencillo para resolver problemas de valores iniciales. Básicamente, el método propone una sucesión recurrente de valores para calcular, de forma aproximada el valor de una función en un punto. En general el método se basa en: dado el problema de valores iniciales

$$\begin{aligned} PVI: T' &= f(t, T) \\ T_0 &= T(t_0) \end{aligned} \quad (2.22)$$

Tomando un valor de paso constante “h” y considerando la sucesión de valores:

$$t_i = t_0 + ih; \quad i = 1, 2, 3, \dots \quad (2.23)$$

Se calcula la sucesión recurrente

$$T_i = T(t_i) = T_{i-1} + h(f(t_{i-1}, T_{i-1})); i=1,2,3\dots \quad (2.24)$$

dicha sucesión dará el valor aproximado de la función T(t) solución del PVI propuesto. En este caso y aplicando lo expuesto a las ecuaciones diferenciales obtenidas anteriormente se obtienen las ecuaciones:

Ecuaciones:

$$T(K + 1) = T(K) + tiempo(\gamma_1 \alpha_{10} e^{\frac{-E_1}{RT(K)}} \times C_A(K)^2 + (\gamma_2 \alpha_{20} e^{\frac{-E_2}{RT(K)}} C_B(K)) + (A_1 + A_2 T(K)) + (B_1 + B_2 T(K))U(K))$$

$$C_A(K + 1) = C_A(K) - tiempo(\alpha_{10} e^{\frac{-E_1}{RT(K)}}) C_A(K)^2$$

$$C_B(K + 1) = C_B(K) + tiempo(\alpha_{10} e^{\frac{-E_1}{RT(K)}}) C_A(K)^2 + \alpha_{20} e^{\frac{-E_2}{RT(K)}} C_B(K)$$

Estas ecuaciones son las que se utilizan en el modelo de la planta en Labview por lo tanto procederemos a introducirlas en Simulink para comprobar que los resultados obtenidos tanto en Matlab como en Labview sean los mismos.

Para introducir las ecuaciones discretizadas en Matlab utilizaremos la herramienta Simulink, primero se ha de representar Arrhenius el cual devolverá k dependiendo de la Temperatura.

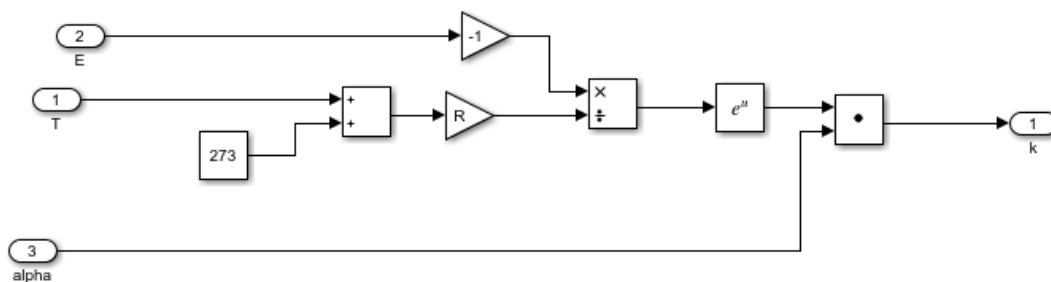


Ilustración 3 ecuación de Arrhenius Simulink, elaboración propia

Donde E será E1 para la k1 y E2 para la k2 lo mismo ocurre con alpha.

Una vez obtenido el sistema de Arrhenius se puede representar las ecuaciones de la C_A y C_B
 Empezaremos por la ecuación de la concentración de especies en A

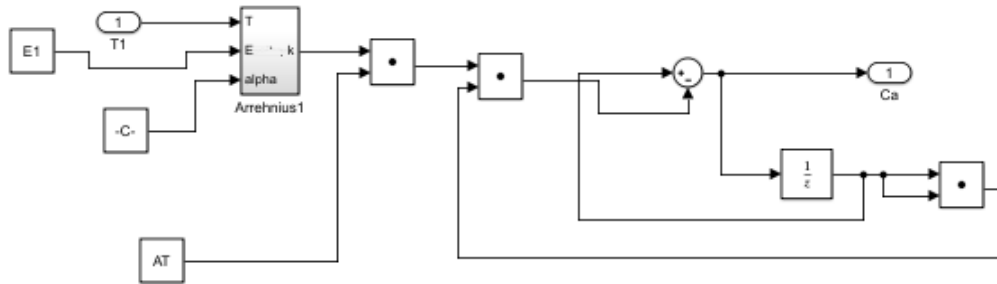


Ilustración 4 ecuación concentración A, elaboración propia

Con el cual se obtiene C_A en función de la temperatura del reactor:

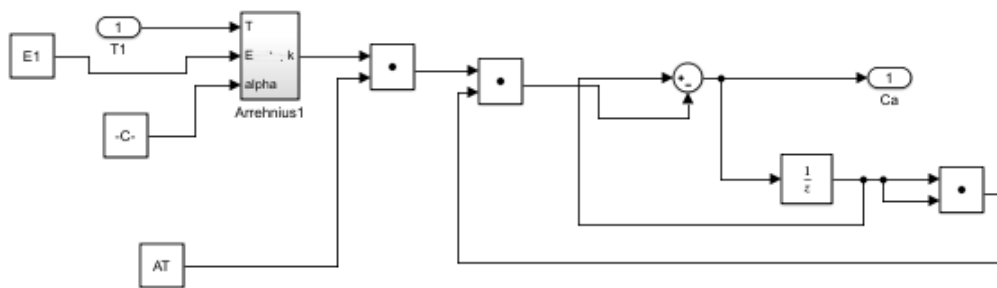


Ilustración 5 ecuación concentración de A en Simulink, elaboración propia

Se creará un subsistema con este esquema para poder utilizarlo en otras ecuaciones



Ilustración 6 Imagen obtenida de Simulink, elaboración propia

Una vez obtenido el subsistema de Ca podemos utilizarlo para crear el subsistema de Cb, volvemos a utilizar Arrhenius, pero esta vez cogemos como datos E₂ y alpha₂₀.

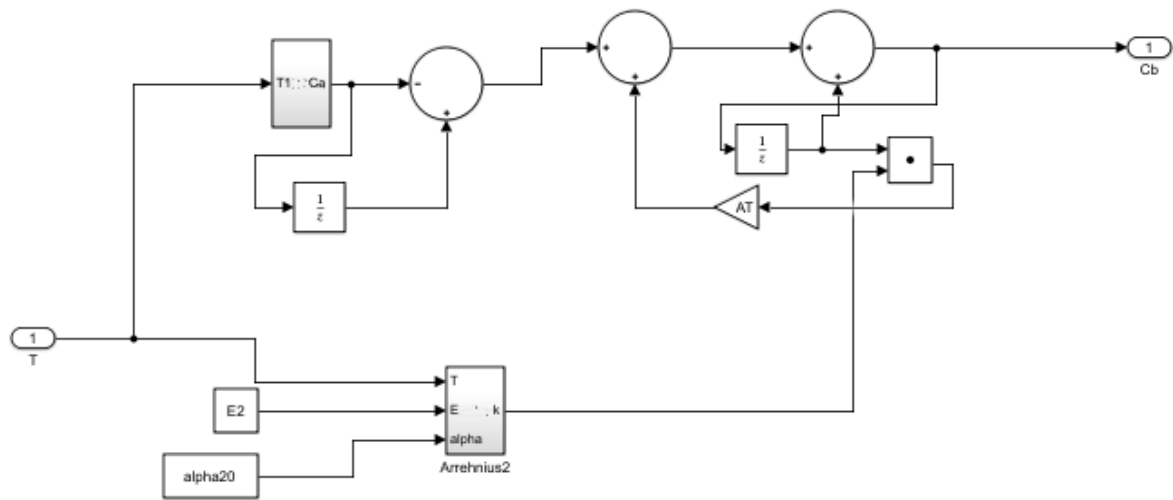


Ilustración 7 Imagen obtenida de Simulink, elaboración propia

Creamos el subsistema de C_B



Ilustración 8 Imagen obtenida de Simulink, elaboración propia

Como valores iniciales se utilizan los citados al principio de este apartado.

$$C_{A0}=1 \text{ kmol/m}^3$$

$$C_{B0}=0 \text{ kmol/m}^3$$

$$u_0=0$$

$$T_0=25^\circ\text{C}$$

Ahora que se dispone de las ecuaciones que relacionan la Temperatura con las concentraciones de A y B, y podemos crear el sistema discretizado que nos relacione la Temperatura del reactor con u:

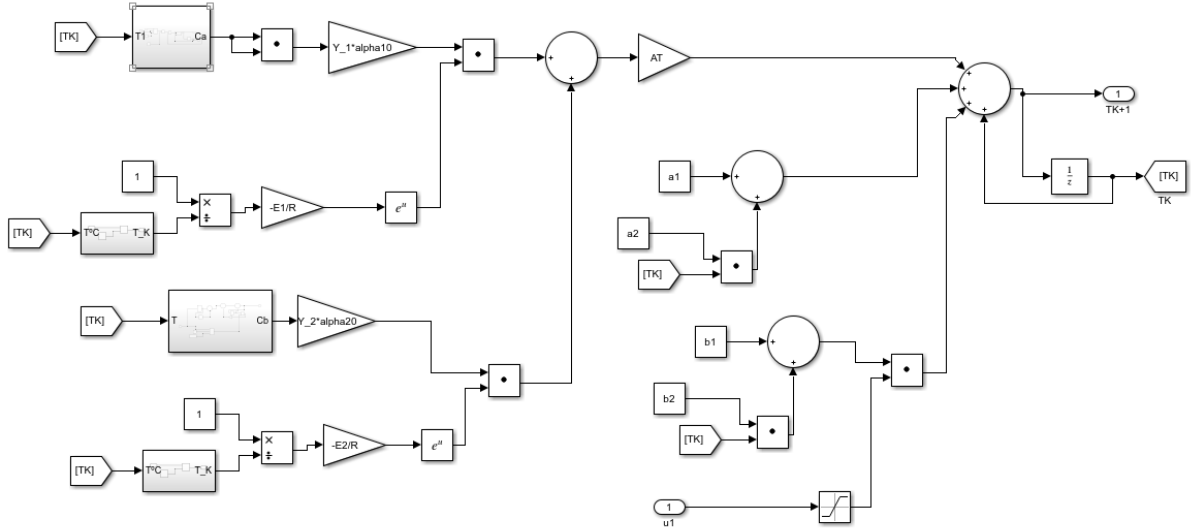


Ilustración 9 ecuación discretizada de la Temperatura en Simulink, elaboración propia

Se ha considerado saturar u ya que no queremos que valga más de 1 o menos de 0 como ya explicamos en el punto 2.2. Para la temperatura inicial se utiliza 25°C. Ya tenemos nuestro modelo discretizado:

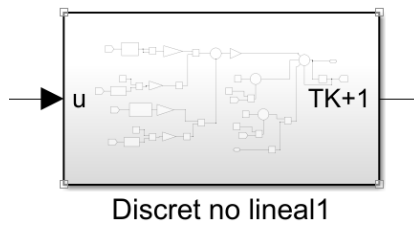


Ilustración 10 Imagen obtenida de Simulink, elaboración propia

DISEÑO DE LOS REGULADORES

En este capítulo se procederá a diseñar los controladores gracias a la herramienta Matlab. Se obtendrán las ecuaciones de transferencia, con el propio programa obtendremos los parámetros del controlador y se comprobará que el comportamiento del sistema es el adecuado.

3.1 Controlador del reactor

En este apartado se diseñará el controlador que utilizaremos en nuestro PLC SIEMENS para controlar la variable “u”, para ello el primer paso será linealizar las ecuaciones diferenciales (Levine 1996; Serban Agachi P. 2006; Wayne Bequette B.1998).

LINEALIZAR EN TORNO AL PUNTO DE FUNCIONAMIENTO

Para aplicar técnicas de control a nuestro sistema tendremos que linealizar las ecuaciones que, aunque se aleja un poco de la realidad, servirá como aproximación a dicha realidad, ya que la mayoría de las técnicas que existen se basan en modelos lineales.

Analizando bien la ecuación principal (2.21), la cual proporciona la derivada de la T del reactor respecto al tiempo:

$$\frac{dT}{dt} = \gamma_1 k_1 C_A^2 + \gamma_2 k_2 C_B + (A_1 + A_2 T) + (B_1 + B_2 T)u$$

Se puede pensar que el calor aportado por la reacción, que en este caso son los dos primeros términos de la ecuación, podrían estar aportándonos variaciones muy pequeñas de la temperatura ya que estamos trabajando con una exponencial negativa, se comprobará si se pueden despreciar estos términos.

Empezaremos por analizar " $\gamma_1 k_1 C_A^2$ ", como podemos observar aparece un cuadrado en la ecuación esto hace que al derivarlo para hacer la aproximada de Taylor.

$$\left. \frac{df(T)}{dt} \right|_{C=0}(C_A-0) = 0$$

Ya que como se ha dicho anteriormente C_A en el punto de funcionamiento vale 0. Por otro lado si se tiene en cuenta que nuestra temperatura deseada (2.18) se encuentra entre los valores 125°C y 54°C, ya que para el tiempo inicial, es decir en $t=0$, la Temperatura tendrá un valor de 125°C (resultado de sumar (54°C+71°C)), mientras que para un tiempo infinito es la obtendrá un valor de 54°C, tomando una temperatura media de 89,5°C se puede ver cómo variará nuestra ecuación con estos términos.

$$\gamma_1 k_1 = 41.8 \times 1.1 \times e^{\frac{-2513.83}{(273+89.5)}} = 0.044$$

$$\gamma_2 k_2 = 83.6 \times 172.2 \times e^{\frac{-5027.66}{(273+89.5)}} = 0.013$$

Como se puede observar sí que influye en la ecuación por lo que lo que se tendrá que linealizar en torno al punto de funcionamiento.

El primer paso será obtener el punto de funcionamiento C_{A0} C_{B0} T_0 u_0 , estudiaremos, pues los puntos en que se estabilizan nuestras ecuaciones a la hora de funcionamiento, es decir 3600s después de que empiece la reacción. Para ello utilizaremos las ecuaciones discretizadas del modelo no lineal que creamos en el apartado 2.3, crearemos un PI sencillo que controle nuestra planta discretizada y observaremos los valores de las variables

PUNTO DE FUNCIONAMIENTO

C_{A0} :

Como se ve en el gráfico de abajo la concentración de especies de A en el reactor a la hora de la operación se estabiliza en 0.3 por lo que este es el valor que se utilizará como punto de funcionamiento.

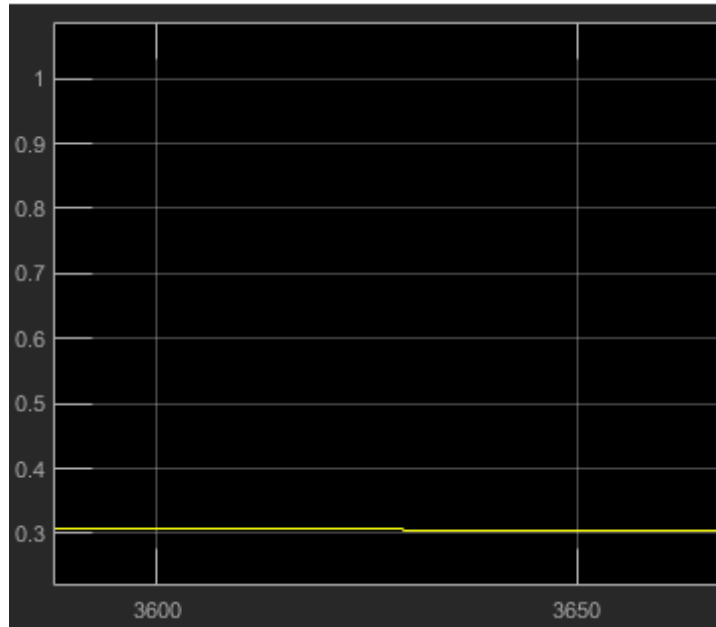


Ilustración 11 Gráfica concentración de A en Simulink, elaboración propia

C_{B0} :

Mismo procedimiento para la concentración de especies de B donde en este caso se estabilizará en 0.8 como se puede ver en la siguiente gráfica



Ilustración 12 Gráfica concentración de B en Simulink, elaboración propia

T₀

Para la T de funcionamiento se utilizará la gráfica de la Temperatura deseada, en esta se observa como a medida que va aumentando el tiempo la ecuación se acerca más a 54°C, por lo que utilizaremos este valor como punto de funcionamiento.

u₀

Por último, se analiza la variable a controlar u, la cual varía entre 1 y 0. Para hallar este valor utilizaremos un scope a la salida del regulador, de esta forma la u que obtenemos a la hora es 0.37, con esto ya tenemos todos los puntos de funcionamiento.

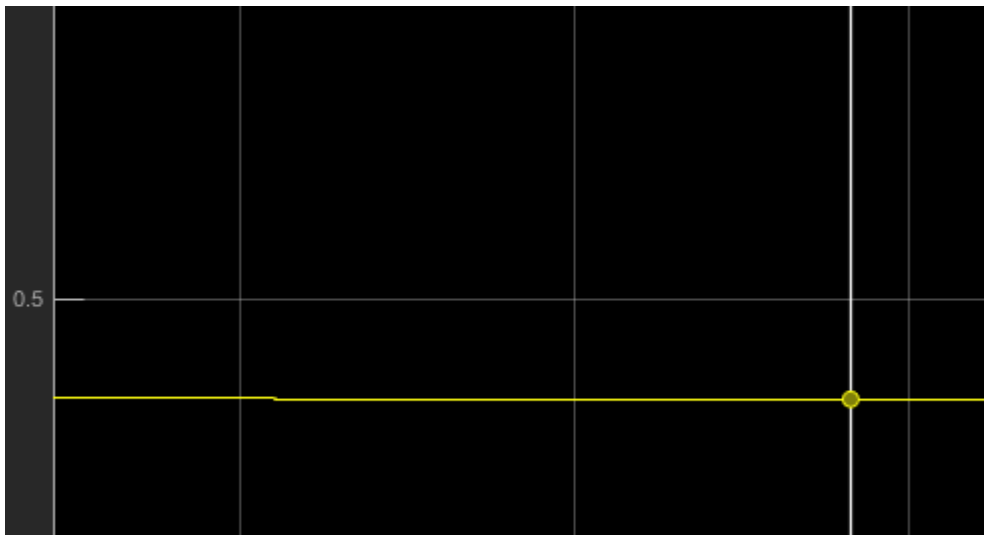


Ilustración 13 Gráfica de la acción de control en Simulink, elaboración propia

Se aplicará el polinomio de Taylor a las ecuaciones discretas con el fin de linealizar los modelos y, substituyendo las constantes del punto de funcionamiento, se obtendrá el valor en régimen nominal. De este modo se obtienen los sistemas necesarios para el diseño de nuestro controlador.

C_A

Primero se linealiza k_1 y k_2 (Arrhenius)

$$k_1 = \alpha_{10} e^{\left(\frac{-E_1}{RT}\right)} \quad k_2 = \alpha_{20} e^{\left(\frac{-E_2}{RT}\right)}$$

Aplicando Taylor obtenemos:

$$k_1 = \alpha_{10} \frac{E_1}{RT} e^{\left(\frac{-E_1}{RT}\right)} \Delta T$$

$$k_2 = \alpha_{20} \frac{E_2}{RT} e^{\left(\frac{-E_2}{RT}\right)} \Delta T$$

De la ecuación de la concentración de A aplicando Taylor obtenemos:

$$\frac{dC_A}{dt} = -k_1 C_A^2$$

(linealizando)

$$\frac{d\Delta C_A}{dt} = -C_{A0}^2 \alpha_{10} \frac{E_1}{RT} e^{\left(\frac{-E_1}{RT}\right)} \Delta T - k_1(T_0) 2C_{A0} \Delta C_A$$
(3.1)

C_B

De igual modo que en con la ecuación anterior se aplicará Taylor para linealizar la ecuación de la concentración de especies de B.

$$\frac{dC_B}{dt} = k_1 C_A^2 - k_2 C_B$$

(linealizando)

$$\frac{d\Delta C_B}{dt} = C_{A0}^2 \alpha_{10} \frac{E_1}{RT} e^{\left(\frac{-E_1}{RT}\right)} \Delta T + k_1(T_0) 2C_{A0} \Delta C_A - k_2(T_0) \Delta C_B$$

$$- C_{B0} \alpha_{20} \frac{E_2}{RT} e^{\left(\frac{-E_2}{RT}\right)} \Delta T$$
(3.2)

T

Por último, linealizaremos la ecuación de la Temperatura del reactor la cual es la que utilizaremos para obtener el regulador

$$\frac{dT}{dt} = \gamma_1 k_1 C_A^2 + \gamma_2 k_2 C_B + (A_1 + A_2 T) + (B_1 + B_2 T)u$$

(linealizando) (3.3)

$$\begin{aligned} \frac{d\Delta T}{dt} = & \gamma_1 \left(C_{A0}^2 \alpha_{10} \frac{E_1}{RT} e^{\left(\frac{-E_1}{RT}\right)} \Delta T + k_1(T_0) 2C_{A0} \Delta C_A \right) \\ & + \gamma_2 \left(k_2(T_0) \Delta C_B - C_{B0} \alpha_{10} \frac{E_2}{RT} e^{\left(\frac{-E_2}{RT}\right)} \Delta T \right) + A_2 \Delta T + (B_1 \\ & + B_2 T_0) \Delta u + B_2 u_0 \Delta T \end{aligned}$$

Una vez obtenidas las ecuaciones obtenemos el regulador. Para ello se utilizará Matlab, mismo procedimiento que en el apartado anterior para introducir las ecuaciones discretizadas en Simulink.

Linealizado C_A

Donde tenemos como entrada ΔT y como salida ΔC_A

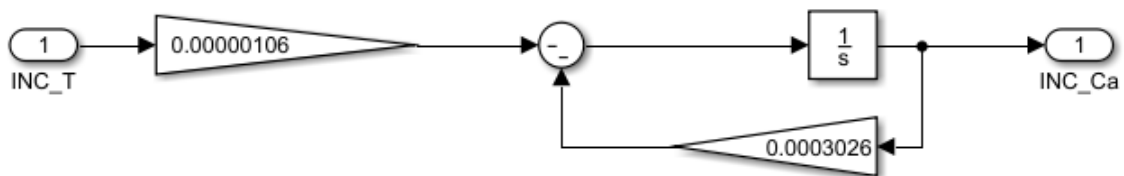


Ilustración 14 Ecuación linealizada de la concentración de A en Simulink, elaboración propia

Linealizado C_B

Donde tenemos como entrada $\Delta C_A \Delta T$ y como salida ΔC_B

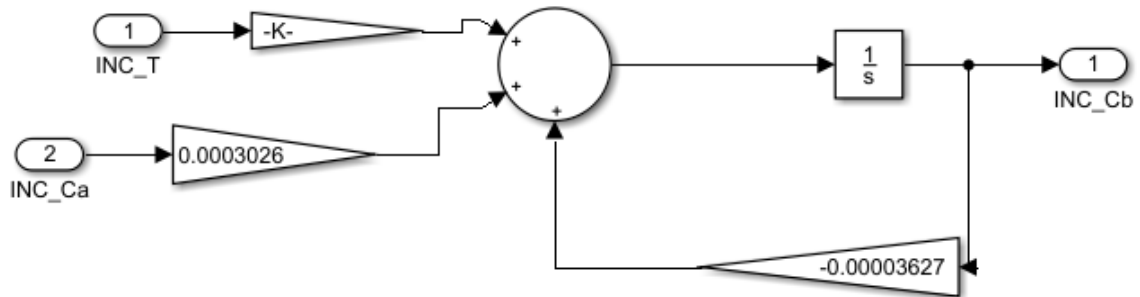


Ilustración 15 Ecuación de la concentración de B linealizada en Simulink, elaboración propia

Linealizado de ΔT

Donde tenemos como entrada $\Delta u \Delta C_A \Delta C_B$ y como salida ΔT

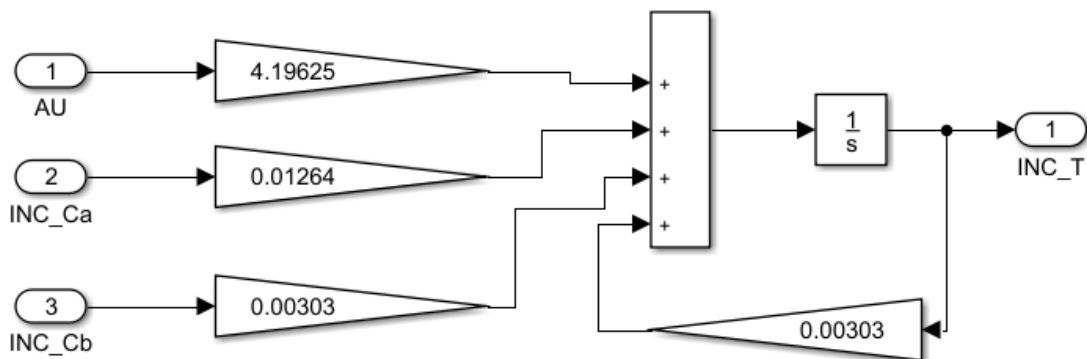


Ilustración 16 Ecuación linealizada de la Temperatura en Simulink, elaboración propia

Con estos tres sistemas podemos montar nuestro sistema linealizado

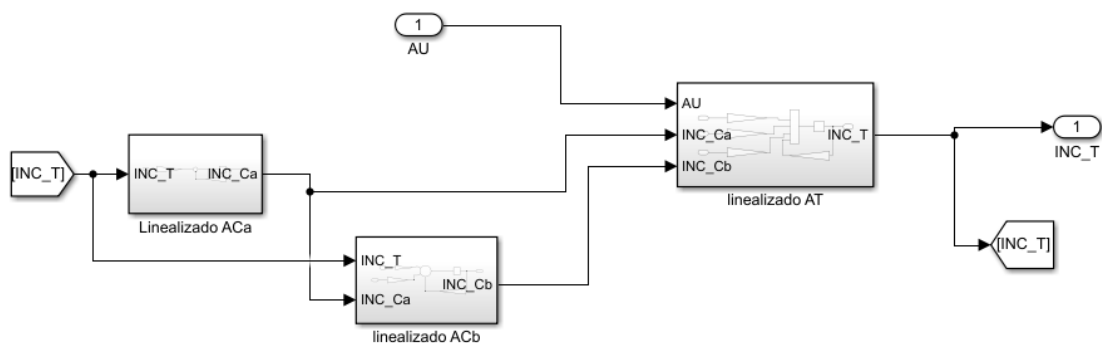


Ilustración 17 Sistema lineal en Simulink, elaboración propia

Importante tener en cuenta que al utilizar incrementos como nuevas variables para adaptar este modelo al nuestro bastará con sumarle a nuestra acción de control su valor en el punto de funcionamiento es decir 0.387.

Se procede pues a crear el controlador que será utilizado en el PLC. Utilizando la herramienta sisotool que nos ofrece Matlab se obtendrá un PI, se ha creado un controlador poco agresivo de manera que se obtenga la Temperatura deseada sin oscilación

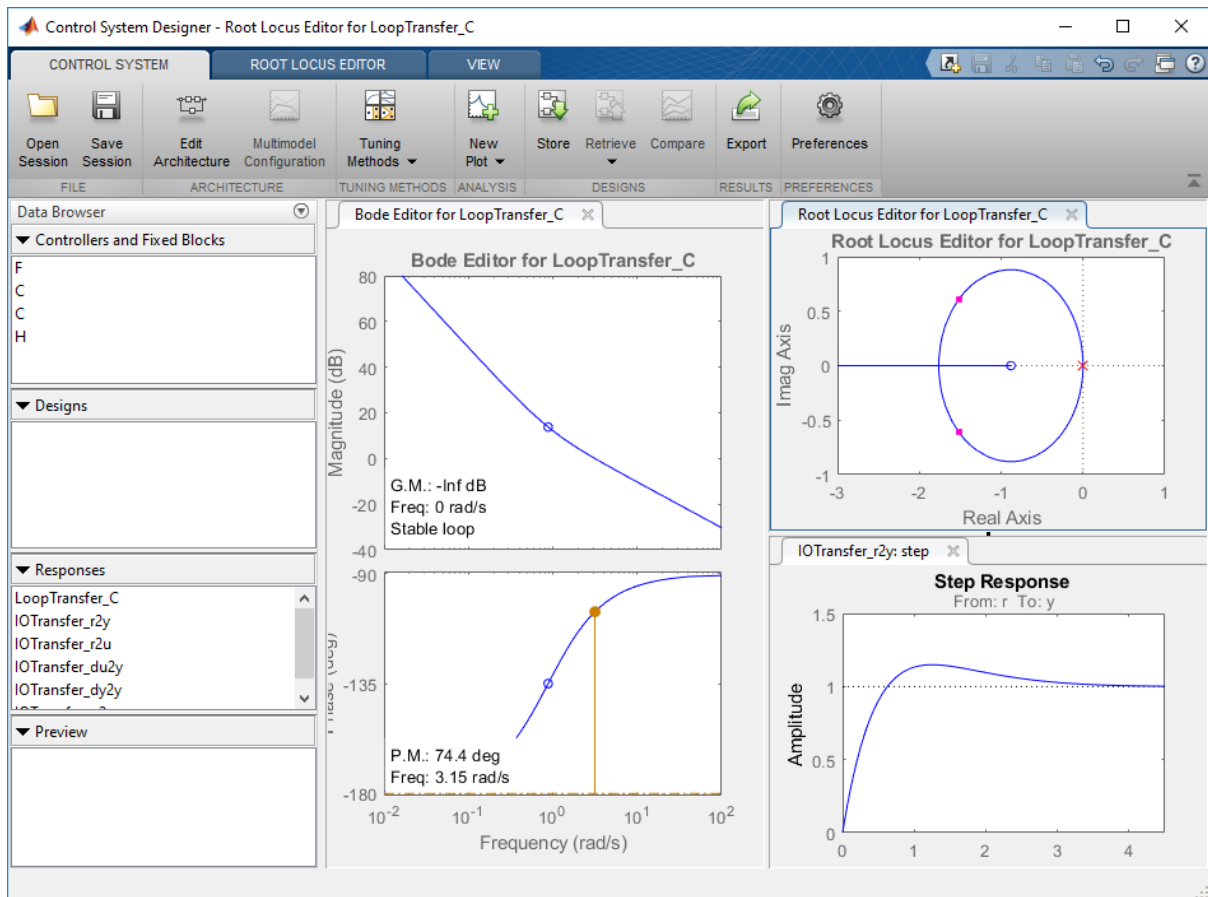


Ilustración 18 Sisotool del controlador en Matlab, elaboración propia

Gracias a la función “pidstdata” obtendremos los parámetros de nuestro controlador:

El controlador contará con los siguientes parámetros:

- $K_p = 0.1714$
- $T_i = 5.193$

3.2 Controlador de los actuadores

Procedamos a crear los reguladores de los actuadores tanto de la camisa como del circuito de refrigeración ya que una vez obtenida la u del regulador de la planta, se necesitará una válvula que regule el caudal de vapor en la camisa y el caudal de agua refrigerante (Louis C.W. 2001).

La mayoría de las válvulas operan por medio de un actuador de posición lineal o alguna modificación de este tipo de actuador.

Una válvula neumática siempre tiene algún retraso dinámico, el cual hace que el movimiento del vapor no responda instantáneamente a la presión aplicada desde el controlador. Se ha encontrado que la relación entre el flujo y la presión para una válvula lineal puede a menudo representarse por una función de transferencia de primer orden, esto es:

$$Gv(s) = \frac{Q(s)}{U(s)} = \frac{Kv}{\tau s + 1}$$

Donde

$Q(s)$ = variable manipulada

$U(s)$ =señal proveniente del controlador

CIRCUITO REFRIGERACIÓN

Para diseñar el regulador de las válvulas del circuito de refrigeración, tomaremos como función de transferencia de la válvula una función de 1º orden de $\tau=0.01$ y k tal que para $t \rightarrow \infty$ es decir $s=0$ la función tienda al Q_{max} ante escalón unitario.

$$Q_{max}=0.002343 \text{ m}^3/\text{s}$$

Para simular la válvula en el proceso del PLC para poder obtener el parámetro de control durante la simulación, en el caso del enfriador:

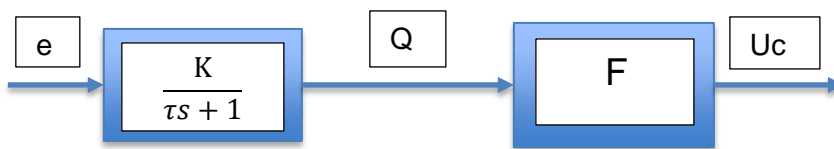


Ilustración 19 esquema del circuito de refrigeración elaboración propia

$$\frac{1}{Uc} = \frac{1}{4550Q^{0.8}} + \frac{1}{10.8} \quad (3.4)$$

$$Uc = \frac{4550 \times 10.8Q^{0.8}}{10.8 + 4550Q^{0.8}}$$

Procederemos pues a crear un subsistema que englobe estas ecuaciones con el fin de obtener los parámetros del controlador

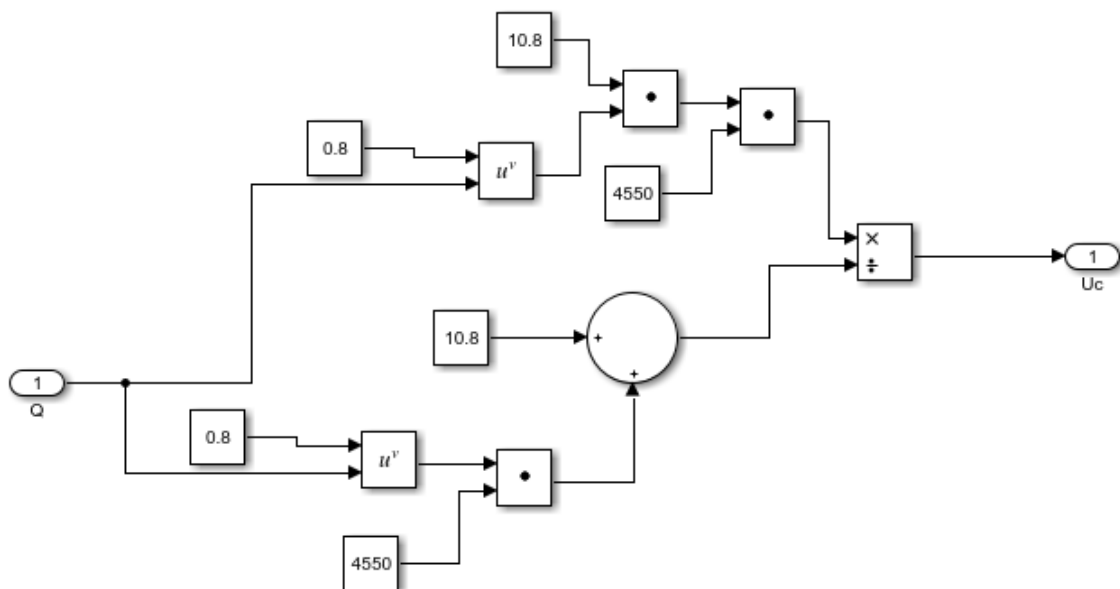


Ilustración 20 Ecuación del controlador en Simulink, elaboración propia

Ahora diseñaremos el regulador con la función `sisotool` buscaremos un control rápido para que no afecte a la dinámica del sistema, y le añadiremos un integrador y un cero cercano al integrador para eliminar el retraso que produce este integrador. Una vez cargue el programa moveremos a lo largo del eje de abscisas para encontrar el control deseado.

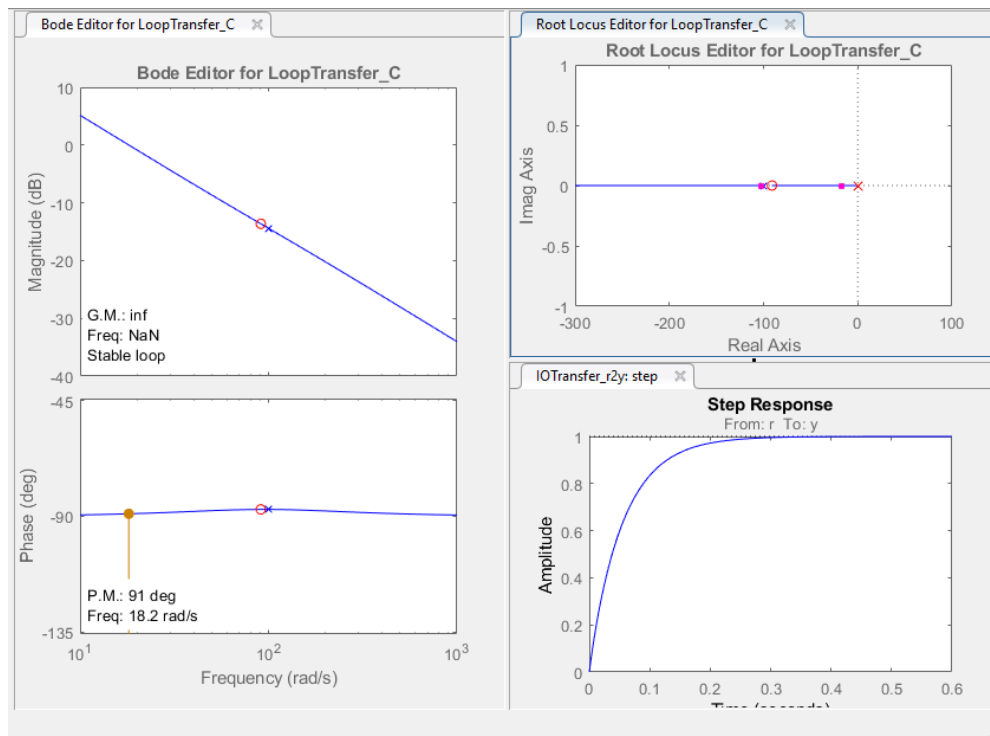


Ilustración 21 Sisotool controlador de la válvula de refrigeración en Matlab, elaboración propia

Ya obtenido el controlador PI usaremos la función `pidstddata` para sacar los valores de K_p y T_i los cuales se utilizarán para programar el controlador en el PLC.

```
>> [kpl,Til,Tdl,N]=pidstddata(PI_Enf)
```

```
kpl =  
  
0.0450
```

```
Til =  
  
0.0110
```

Ilustración 22 Parámetros controlador, elaboración propia

Nuestro controlador tendrá los siguientes valores:

- $K_p=0.045$
- $T_i=0.0110$

VÁLVULA DE LA CAMISA

Para hallar la ecuación de la válvula se supondrá que el Calor aportado será el Calor que adquiere el fluido ya que se supone que no hay pérdidas en la camisa.

$$Q_{aportado} = Q_{calentado} \quad (3.5)$$

$$U_j A_j (T_s - T) = Q_{cal} C_{pl} (T_f - T_i)$$

Para resolver este problema tomaremos como T_i la temperatura ambiente, es decir, 20°C y T_f 150°C.

Sustituiremos la T_s por la ecuación que la relacione con u

$$T_s = (T_{s,max} - T_{s,min})u + T_{s,min}$$

Al final quedará la siguiente ecuación, dado que todos los restantes valores son conocidos:

$$Q_{cal} = 0.0363(80u + 50) \quad (3.6)$$

para sacar la función de transferencia utilizaremos una nueva variable $Q_{cal1} = Q_{cal} - 1.815$ por lo que nuestra función de transferencia será:

$$\frac{Q_{cal1}}{u} = 2.904$$

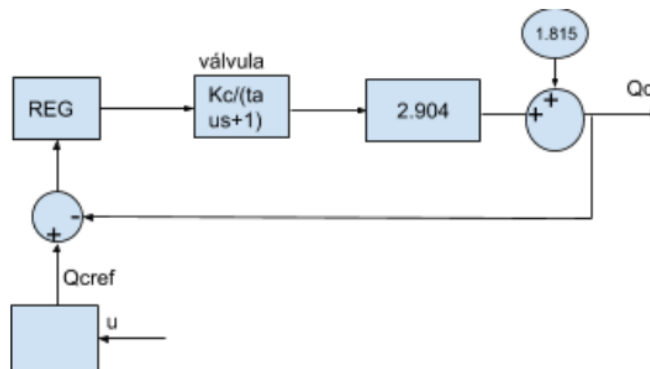


Ilustración 23 Imagen obtenida por elaboración propia

Para crear el controlador de la camisa se usará el mismo procedimiento que en el circuito de refrigeración, utilizando la herramienta Matlab para que nos genere los parámetros del controlador, para ello se introducirá la función de transferencia, siendo K_c el Caudal máximo del circuito ($1.5\text{m}^3/\text{s}$) y tau 0.01.

```
>> Kc=1.5;
>> Ku=2.904;
>> FDT_Cal=Kc*Ku/(0.01*s+1)
```

$$\text{FDT_Cal} = \frac{4.356}{0.01 s + 1}$$

Ilustración 24 Función de transferencia en Matlab, elaboración propia

Utilizaremos de nuevo la herramienta sisotool para configurar nuestro regulador y para que nos controle la válvula.

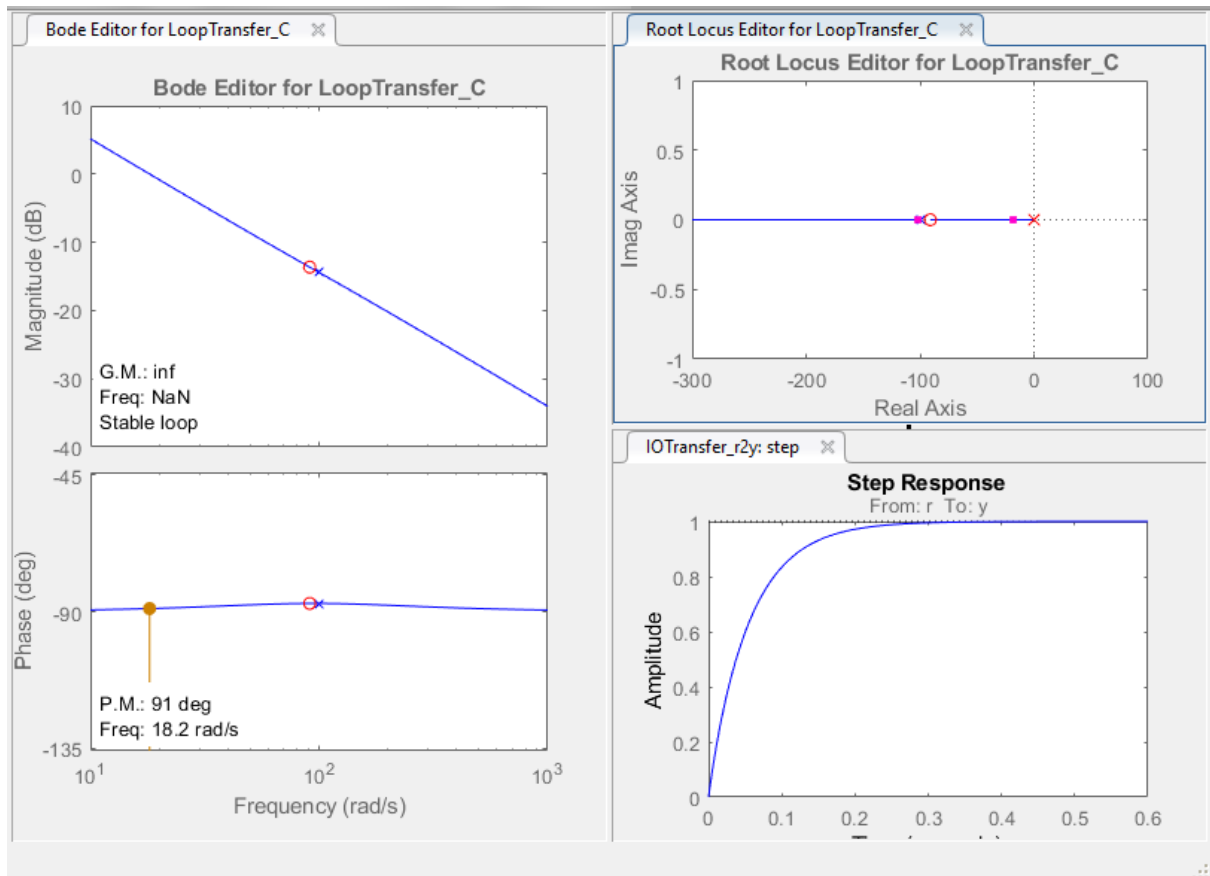


Ilustración 25 Sisotool controlador válvula de calentamiento en Matlab elaboración propia

Por último, se obtendrán los parámetros de nuestro controlador con la función `pidstddata`.

```
>> [kpl,Til,Tdl,N]=pidstddata(PI_Cal)
```

```
kpl =
```

```
0.0456
```

```
Til =
```

```
0.0110
```

Nuestro segundo controlador tendrá los siguientes valores:

$K_p=0.0456$

$T_i=0.0110$

Capítulo 4

IMPLEMENTACIÓN DE LOS PROGRAMAS EN TIA PORTAL

En este capítulo se procederá a diseñar controlador mediante el programa Tia Portal así como crear las funciones que lo componen de forma que controle las Temperaturas provenientes del LabView.

Para la elaboración de este apartado se ha decidido programar las funciones de los controladores en SCL aprendidas gracias al punto 7 de la Bibliografía las cuales son llamadas desde un Cyclic Interrupt que se ejecutará cada 100 ms, por otro lado, utilizaremos el Main para inicializar las variables.

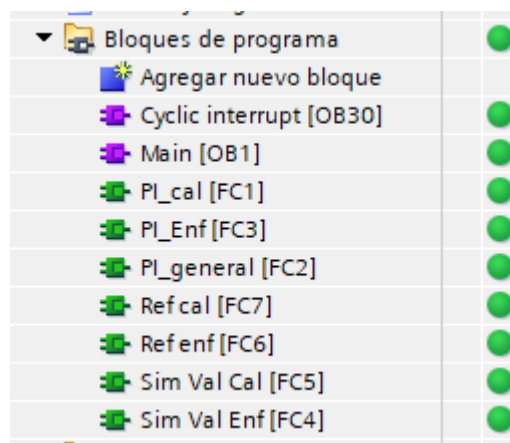


Ilustración 26 Bloques de programa en Tia Portal, elaboración propia

4.1 Válvula Camisa

El primer paso para este capítulo será obtener las ecuaciones discretizadas de los controladores para ello utilizaremos los parámetros que hemos obtenido en el 3.2.

En el caso de la camisa se obtuvieron los siguientes parámetros:

$$K_p=0.0456$$

$$T_i=0.011$$

Partiremos de la ecuación de un PI

$$PI = K_p \left(1 + \frac{1}{T_i s} \right)$$

Sustituyendo nuestros valores obtenemos que:

$$\frac{u}{Q_c} = 0.0456 \left(1 + \frac{1}{0.011s} \right) \quad (4.1)$$

Aplicaremos:

$$Q_{cs} = \frac{qk - qk_1}{\Delta T}$$
$$us = \frac{uk - uk_1}{\Delta T}$$

Reordenando términos y despejando uk obtenemos que:

$$uk = 0.0456(qk - qk_1 + qk\Delta T) + uk_1 \quad (4.2)$$
$$uk = 0.0456(qk(1 + \Delta T) - qk_1) + uk_1$$

En el Tia Portal lo primero que programaremos será la Qref del regulador del calentador para ello aplicaremos la siguiente ecuación

$$\#Qref := 0.0363 * (80 * \#u + 50);$$

Con la cual se obtendrá nuestra Q de referencia a partir de la u procedente del regulador del reactor.

Con este valor podemos calcular la Apertura de la Válvula de la camisa: A_Val_Cal a partir de Qck y Qck_1

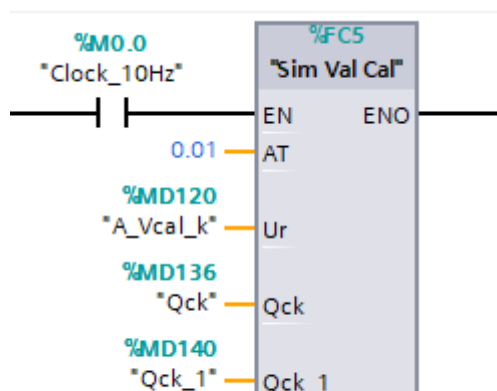


Ilustración 27 Función del controlador válvula Calentamiento en Tia Portal, elaboración propia

Una vez obtenido este valor calcularemos el Error de la Q del calentador para ello restaremos la Q de referencia calculada previamente y la Qck.

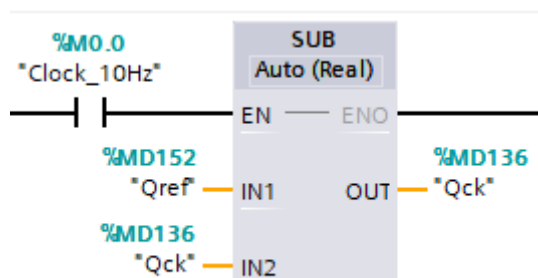


Ilustración 28 Imagen obtenida de Tia Portal, elaboración propia

Con nuestra nueva Qck, la cual ahora es el error, podemos introducirla en el regulador para que aplique la acción de control sobre la apertura de la válvula de enfriamiento.

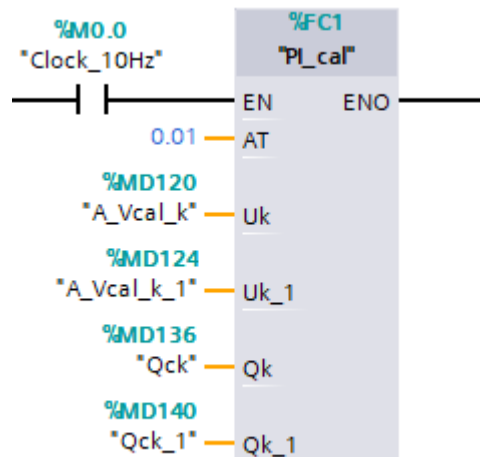


Ilustración 29 Imagen obtenida de Tia Portal, elaboración propia

```

1 #Uk := 0.0456 * (#Qk * (1 + #AT) - #Qk_1) + #Uk_1;
2 IF #Uk > 1.0 THEN
3     #Uk := 1.0;
4 END_IF;
5 IF #Uk < 0.0 THEN
6     #Uk := 0.0;
7 END_IF;
8 #Uk_1 := #Uk;
9 #Qk_1 := #Qk;

```

Ilustración 30 Imagen obtenida de Tia Portal, elaboración propia

(YOUTUBE, “TIA PORTAL 20/20: Programar en SCL S7 1200 Parte2 AUDIO CORREGIDO”)

(YOUTUBE, “TIA PORTAL 8/20: Plantillas de Funciones FC”)

Limitaremos el valor de apertura de la válvula ya que es porcentual entre 1, abierto al máximo y 0 completamente cerrado.

4.2 Válvula refrigeración

Ahora de la misma forma calcularemos la ecuación del controlador de la válvula del circuito de refrigeración cuyos parámetros son:

$K_p=0.045$

$T_i=0.0110$

Aplicando los mismos pasos que para el controlador anterior se obtiene la siguiente ecuación:

$$Q_k = 0.0456(uk - uk_1 + uk\Delta T) + uk_1 \quad (4.3)$$

$$Q_k = 0.045 * (Uck * (1 + AT) - Uck_1) + Uck_1;$$

Antes de aplicar estas ecuaciones tendremos que obtener los errores de la U_c , de este modo necesitamos U_c de referencia calculada a partir de la u proveniente del regulador del reactor.

```
1 #Ucref := (1.39-4.42) * #U + 4.42;
```

Ilustración 31 Imagen obtenida de Tia Portal, elaboración propia

Ahora calcularemos la apertura de la válvula de enfriamiento a partir de este valor U_c

```
1 #Uck := (1 / (1 + (0.01 / #AT))) * (4.42 * #Ur + 0.01 / #AT * #Uck_1);
```

Ilustración 32 Imagen obtenida de Tia Portal, elaboración propia

Este valor es el que pasaremos a Labview para que lo muestre en el panel frontal. Calculamos el error de Q_{enf} restando el valor de U_c de referencia, calculado con la u proveniente del regulador del reactor, y el valor de U_c

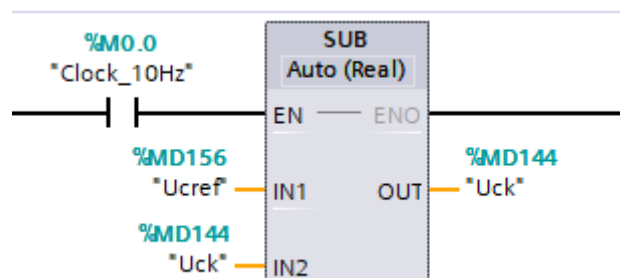


Ilustración 33 Imagen obtenida de Tia Portal, elaboración propia

El último paso será aplicar el regulador que nos controle la apertura de la válvula de refrigeración con nuestro nuevo valor Uck.

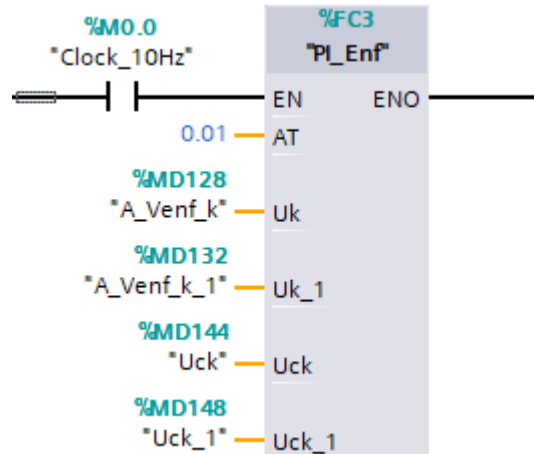


Ilustración 34 Regulador del controlador enfriamiento en Tia Portal, elaboración propia

```

1 #Uk := 0.045 * (#Uck * (1 + #AT) - #Uck_1) + #Uck_1;
2 IF #Uk > 1.0 THEN
3     #Uk := 1.0;
4 END_IF;
5 IF #Uk < 0.0 THEN
6     #Uk := 0.0;
7 END_IF;
8 #Uk_1 := #Uk;
9 #Uck_1 := #Uck;

```

Ilustración 35 Función circuito refrigeración, elaboración propia

Limitaremos el valor de apertura de la válvula ya que es porcentual entre 1, abierto al máximo y 0 completamente cerrado.

4.3 Reactor

Por último, programaremos el regulador del reactor, para ello discretizaremos la ecuación del PI, partiendo de los parámetros obtenidos:

$$K_p = 0.1714$$

$$T_i = 5.193$$

Partimos de la ecuación del PI

$$PI = K_p \left(1 + \frac{1}{T_i s} \right)$$

Sustituyendo con nuestros datos y sabiendo que $PI = u/ET$ donde u es la variable que nos modifica tanto la Temperatura de la camisa como del sistema de refrigeración y ET es el error de Temperatura obtenido previamente en el LabVIEW mediante la resta de la T_d , Temperatura deseada, y la propia Temperatura del reactor en la iteración anterior.

$$\frac{u}{ET} = 0.1714 \left(1 + \frac{1}{5.193s} \right) \quad (4.4)$$

Utilizaremos la conversión de:

$$us = \frac{(uk - uk_1)}{\Delta T}$$
$$ETs = \frac{(ETk - ETk_1)}{\Delta T}$$

Substituyendo todos los valores y despejando uk se obtiene la ecuación del PI discretizada:

$$uk = 0.1714(ETk - ETk_1) + 0.033Tk + uk_1 \quad (4.5)$$

Por último, y como ya se indicó en el cálculo del controlador, para que sea válido habrá que sumarle a la acción de control su valor en el punto de establecimiento es decir 0.387, de esta forma se obtiene la siguiente ecuación:

$$uk = 0.1714(ETk - ETk_1) + 0.033Tk + uk_1 + 0.387 \quad (4.6)$$

Programamos como ha de ser nuestro controlador, tendremos que saturar el valor de u de manera que si su valor sea mayor de 1 se mantenga en 1 y si su valor es menor de 0 se sature en 0, con esto simulamos nuestro sistema sin superar las Temperaturas máximas de la camisa y del refrigerador.

Para un sistema de control donde exista una saturación en la variable de control como es nuestro caso, puede suceder que la variable de control alcance los límites prefijados del actuador. Cuando esto pasa, el bucle realimentado permanece en su límite independientemente de la salida del proceso. Si se usa un controlador con acción integral, el error continuará siendo integrado, incrementando aún más su valor. Esto significa que el término integral puede volverse muy grande y producirse el efecto llamado “windup”. Para evitar que ocurra este fenómeno hay dos formas:

1. Introducir limitadores en las variaciones de la referencia tal que la salida del controlador nunca alcance los límites del actuador. Esto, a menudo, produce límites en el funcionamiento del controlador y no evita el windup causado por las perturbaciones.
2. Otra forma es el re-cálculo de la integral: cuando la salida se satura, la integral es recalculada tal que su nuevo valor proporciona una salida en el límite de la saturación.

Esta última será la que utilicemos en nuestro trabajo

Existen varios tipos de antiwindup como por ejemplo: Back calculation o Claping. En este trabajo se optará por utilizar Claping, en este tipo de controladores la acción integral solo se tendrá en cuenta cuando la acción de control sature.

De esta forma tendremos dos formas de controlar nuestro sistema:

1. La primera opción será sin antiwindup.

```
1 #ETk := "Td" - "T";
2 #Uk := 0.1714 * (#ETk-#ETk_1) + 0.033 * #ETk_1 * #AT+#Uk_1;
3 #Uk_1 := #Uk;
4 #Uk := #Uk + 0.387;
5
6 IF (#Uk > 1.0) THEN
7     #Uk := 1.0;
8     END_IF;
9 IF (#Uk < 0) THEN
10    #Uk := 0.0;
11    END_IF;
12
13
14    #ETk_1 := #ETk;
```

Ilustración 36 Controlador Temperatura sin antiwindup, elaboración propia

2. La segunda opción será utilizando el método antiwindup Claping de esta forma solo sumaremos la acción integral $(0.033 * \#Etk_1 * AT)$ cuando la $\#Uk$ sea saturada.

```
1 #ETk := "Td" - "T";
2 #Uk := 0.1714 * (#ETk-#ETk_1) + #Uk_1;
3 IF (#Uk < 1.0) & (#Uk > 0) THEN
4     #Uk := #Uk + 0.033 * #ETk_1 * #AT;
5 END_IF;
6 #Uk_1 := #Uk;
7 #Uk := #Uk + 0.387;
8
9 IF (#Uk > 1.0) THEN
10     #Uk := 1.0;
11 END_IF;
12 IF (#Uk < 0) THEN
13     #Uk := 0.0;
14 END_IF;
15
16
17
18     #ETk_1 := #ETk;
```

Ilustración 37 Función controlador Temperatura con antiwindup, elaboración propia

En el cyclic interrupt se utilizará las funciones que hemos creado introduciendo los valores de Td, T y el valor de AT tiempo de muestreo que será el mismo que en el LabVIEW.

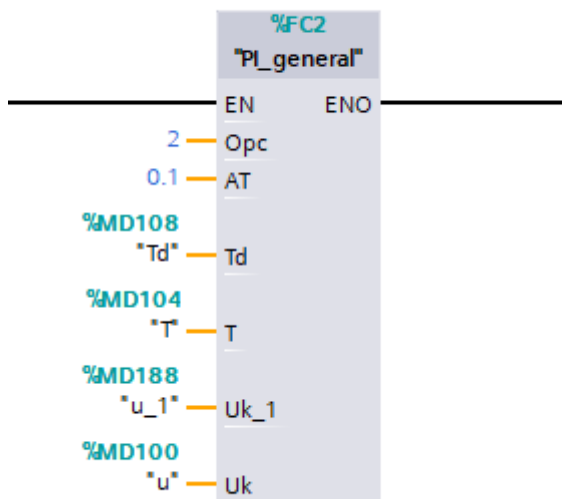


Ilustración 38 Imagen obtenida de Tia Portal, elaboración propia

Y obtendremos el valor de u que enviaremos a la planta simulada en LabVIEW mediante el servidor OPC para que calcule el valor de la Temperatura del reactor.

Capítulo 5

VIRTUALIZAR LA PLANTA

En este capítulo se procederá a diseñar el reactor con la herramienta LabVIEW implementando las ecuaciones para que se comporten como el proceso real con el fin de poder controlarla con el PLC. Contaremos con las variables que vendrán desde el autómata mediante la comunicación OPC de manera que el flujo de información sea constante y nos genere una gráfica con el desarrollo de la reacción.

5.1 Diseño planta con labVIEW

LabVIEW es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico pensado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido.

Esta aplicación creada por National instruments nos ofrecerá la oportunidad de simular nuestro proceso y a su vez de crear una interfaz para que el usuario pueda interpretar resultados y evaluar la viabilidad del proyecto, en este caso, del reactor batch.

(YOUTUBE, “001 - Basics of Labview - LabView with DMZ”)

Así pues, a continuación, se muestra el panel frontal parte que ve el usuario y donde se representan los resultados de la simulación:

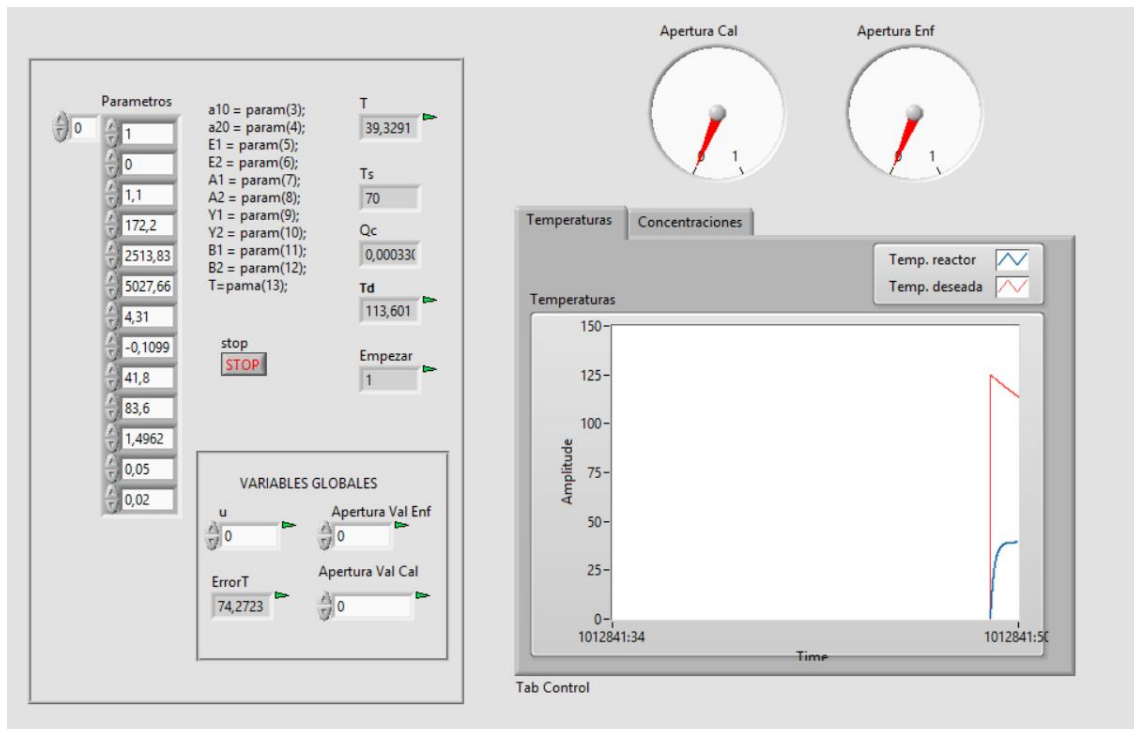


Ilustración 39 Front Panel de labVIEW, elaboración propia

En este panel se puede modificar los parámetros del sistema junto a una tabla explicativa con cada uno de los parámetros.

Se han introducidos unos indicadores que mostrarán en pantalla las variables más importantes como son la acción de control u el error de temperatura la propia temperatura del reactor y la apertura de las válvulas de enfriamiento y calentamiento.

A la derecha disponemos de un cuadro que contiene dos gráficas, la gráfica principal que muestra a tiempo real la evolución de la Temperatura del reactor junto a la temperatura de referencia, y la gráfica de las concentraciones donde se podrá seguir la evolución de las concentraciones de especies de A y de B para que se detenga la reacción cuando se halla alcanzado las cantidades óptimas.

Por otro lado, las válvulas de nuestro sistema se han representado en la parte superior donde podremos observar a tiempo real la apertura de las mismas siendo su valor máximo 1 y su valor mínimo 0.

5.2 NI Servidor OPC

El servidor OPC nos permite el flujo de información entre el PLC y el Labview para configurarlo primero hemos tenido que agregar las variables en el programa Labview.

En nuestro proyecto hemos creado una librería a la cual llamamos librería.lvlib donde introduciremos nuestras variables:

- **A_Val_Cal**: el grado de apertura de la válvula de Calentamiento
- **A_Val_Enf**: el grado de apertura de la válvula de Enfriamiento
- **ErrorT**: Resultado de aplicar la resta entre la Temperatura deseada y la Temperatura del reactor
- **u**: variable a controlar que recibiremos del controlador y la cual nos permitirá calcula la Temperatura del reactor
- **T**: Temperatura del reactor

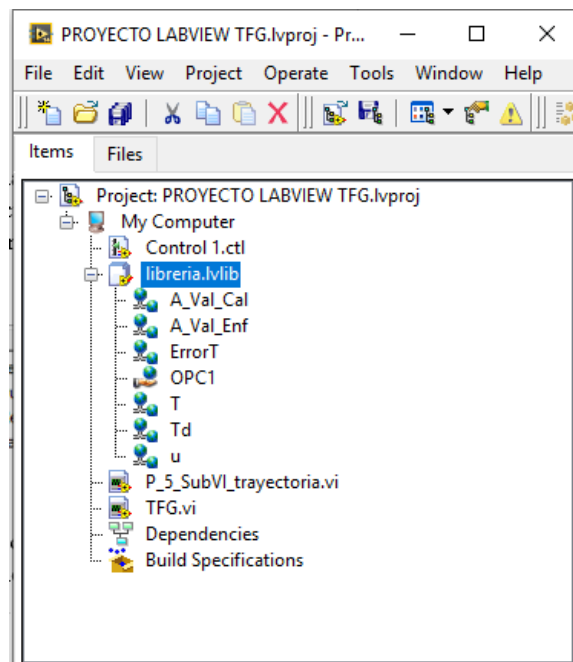


Ilustración 40 Imagen obtenida de labVIEW, elaboración propia

Una vez creadas procederemos a crear nuestro servidor OPC el cual vinculará nuestro autómatas con el Labview para ello tendremos que crear un nuevo canal al cual hemos llamado “SIEMENS”, introducir la IP de nuestro PLC e introducir las variables con la dirección que les hallamos asignado en la tabla de variables del Tia Portal.

Utilizaremos el servidor OPC que nos ofrece National Instruments, es decir ,el NI Server OPC en el anexo explicaremos como se ha de configurar.

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
A_Val_Cal	MD120	Float	100	None	Apertura valvula del Calentador
A_Val_enf	MD128	Float	100	None	apertura de la valvula del enfriador
T	MD104	Float	100	None	Temperatura del Reactor
Td	MD108	Float	100	None	Temperatura deseada
u	MD100	Float	100	None	u

Ilustración 41 Tags de las variables incluidas en el servidor OPC, elaboración propia

Les hemos asignado un tiempo de muestreo de 100ms para que coincida con ambos programas y gracias a la opción de OPC Quick cliente que aparece en la izquierda de la barra de herramientas podremos comprobar los valores de nuestras variables en el momento que se ejecutan.

Capítulo 6

PLANTILLAS

En este capítulo se creará dos plantillas, tanto en LabVIEW como en Tia Portal, de forma que este trabajo no solo abarque un tipo de reactor, sino que pueda ser utilizado para la creación de la simulación de otros procesos.

6.1 Elaboración de la plantilla labVIEW

Para que este trabajo se pueda utilizar en futuros proyectos, se ha creado una plantilla con el fin de que, no solo se pueda simular el reactor batch sino cualquier tipo de proceso, para ello se tendrá que disponer de las ecuaciones del proceso que se quiera simular. De este modo se obtiene la siguiente interfaz:

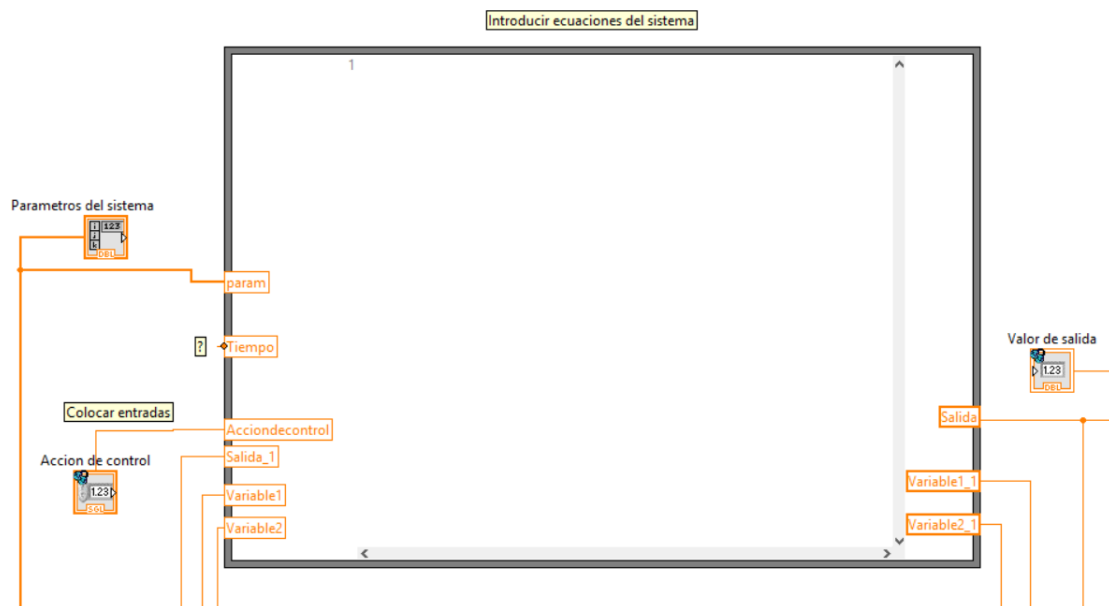


Ilustración 42 | Imagen de la plantilla obtenida en labVIEW, elaboración propia

Se utilizará la función fórmula node para darle la opción al usuario de introducir sus ecuaciones junto con sus variables de entrada y de salida.

También le permitirá introducir sus valores de referencia dando la opción de hacerlo mediante un subVi como es el caso de este trabajo o de introducir una referencia escalón.

Se ha creado una interfaz en el front panel.

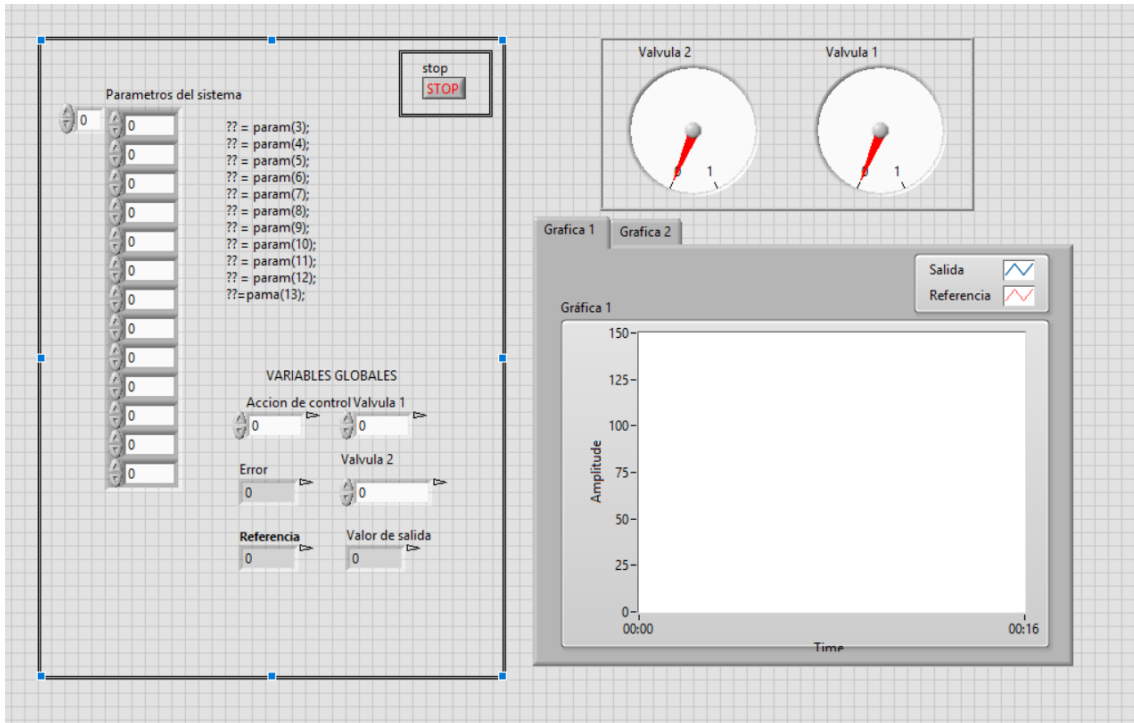


Ilustración 43 Panel frontal del labVIEW, elaboración propia

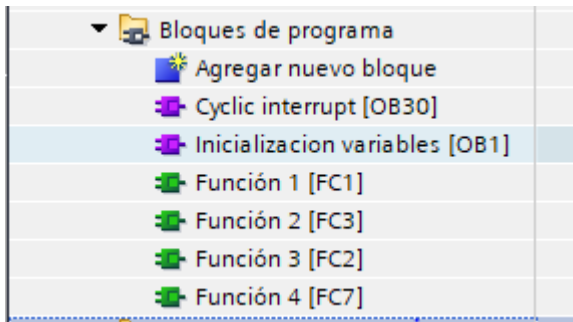
Como se puede observar se le permite la opción al usuario de introducir los parámetros que posteriormente serán utilizados en las ecuaciones del sistema.

En pantalla se mostrará el valor de la acción de control, así como el valor de la referencia del error y el valor de salida de sistema. Para poder monitorizarlos se ha incorporado una gráfica en la que se representará el valor de referencia junto al valor de salida. Además, en caso de que se precise mostrar más valores se ha incorporado otra gráfica.

Todas las variables son globales de forma que el usuario pueda elegir las variables que serán transferidas al PLC para ser controladas, así como las que serán transferidas al Labview.

6.2 Elaboración de la plantilla Tia Portal

Después de crear la plantilla de LabVIEW se ha de crear una plantilla del controlador para ello utilizaremos el programa que hemos diseñado en Tia Portal.

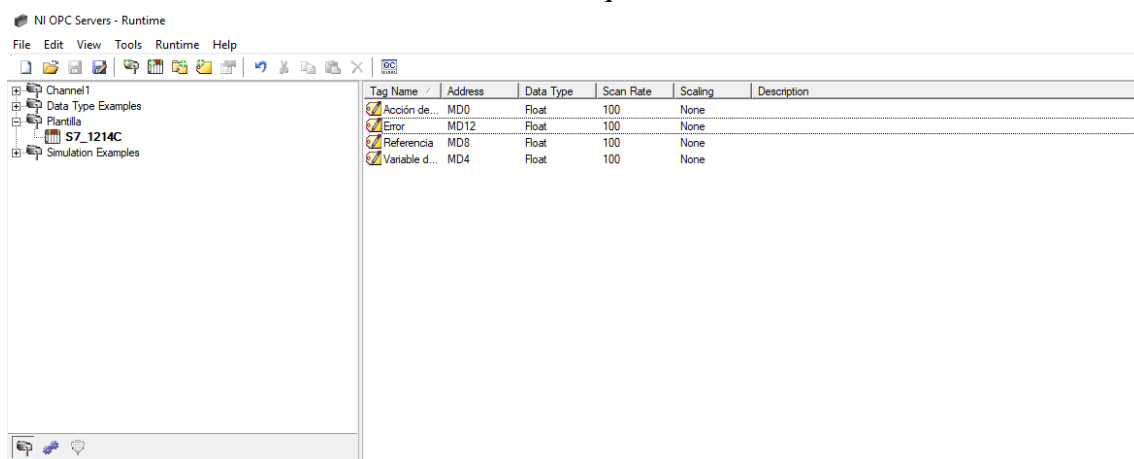


Dejaremos 4 funciones en las que el usuario programará las ecuaciones de su controlador con lenguaje SCL.

Se dejará que el usuario introduzca sus variables en la tabla de variables.

Variables PLC							
	Nombre	Tabla de variables	Tipo de datos ▲	Dirección	Rema...	Visibl...	Acces...
1	Acción de control	Tabla de variables e..	Real	%MD0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Variable de entrada	Tabla de variables e..	Real	%MD4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Referencia	Tabla de variables e..	Real	%MD8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Error	Tabla de variabl...	Real	%MD12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	<Agregar>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Se introduce variables provisionales ya que se necesita que en el NI Server OPC las direcciones de las variables coincidan con las que se introduce.



Se ha creado un canal llamado Plantilla que recoja las variables que se van a utilizar aunque el usuario puede modificarlas con las instrucciones del anexo página 97.

EVALUACIÓN DE RESULTADOS

En este capítulo se procederá a observar resultados obtenidos con nuestro trabajo comparándola con una simulación realizada mediante Simulink la cual nos permite incluir todo nuestro proyecto y comparar resultados.

7.1 Conclusiones

Para determinar que es posible controlar la planta simulada, se procederá a comparar los resultados obtenidos con los programas Tia y Portal y LabVIEW con Matlab. Para ello es fundamental que en ambos se haya realizado el mismo procedimiento. Así pues, en LabVIEW se utiliza el sistema del batch reactor discretizado de modo que en Matlab se utilizará el mismo. Respecto al controlador será necesario que en ambas partes sean iguales de forma que la comparación sea válida por lo que para la primera comparación utilizaremos el controlador sin antiwindup.

Se ha utilizado este esquema para simular la planta en Simulink

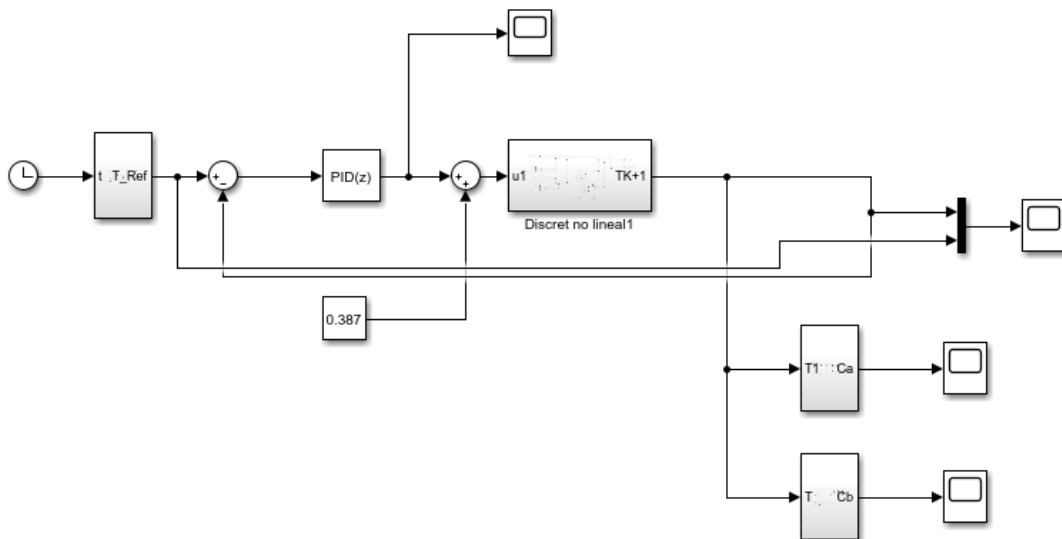


Ilustración 44 Simulación proceso químico Simulink, elaboración propia

En la cual cada 100 ms se ejecutará el sistema y los resultados serán recogidos en un scope de forma que se le dará mayor importancia a la acción de control u y a la Temperatura del sistema.

Para comparar los datos es necesario pasar los resultados obtenidos del Tia Portal a Matlab para ello se ha utilizado la herramienta Trace la cual graba los resultados obtenidos a medida que se ejecute el proceso.

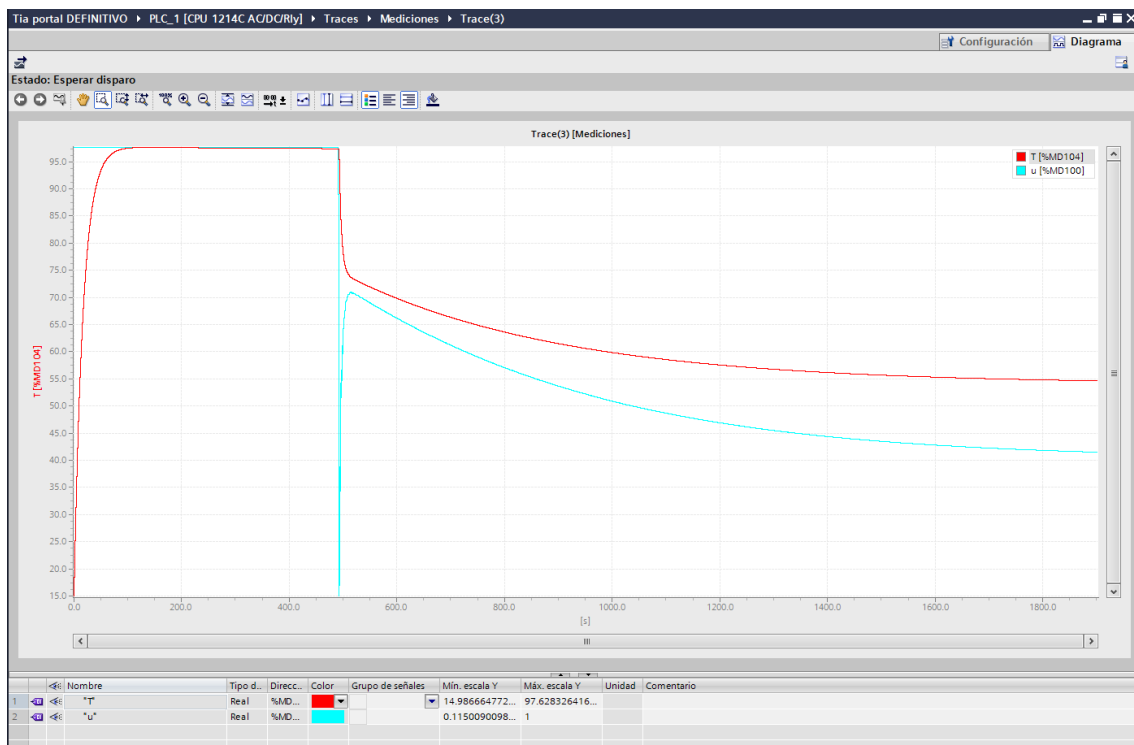


Ilustración 45 Trace de la temperatura del reactor y acción de control, elaboración propia

Se han representado los valores de la acción de control y de la temperatura del reactor, en este caso se han grabado 1800 segundos de forma que se pueda observar de forma clara la etapa de aumento de temperatura y el momento en el que se alcance la temperatura de referencia.

Tia Portal da la opción de guardar las mediciones como .csv lo cual nos facilitará mucho pasar los datos a Matlab.

Una vez introducimos los valores en Matlab utilizaremos un scope para que nos compare los valores de las temperaturas con el valor de referencia y la acción de control.

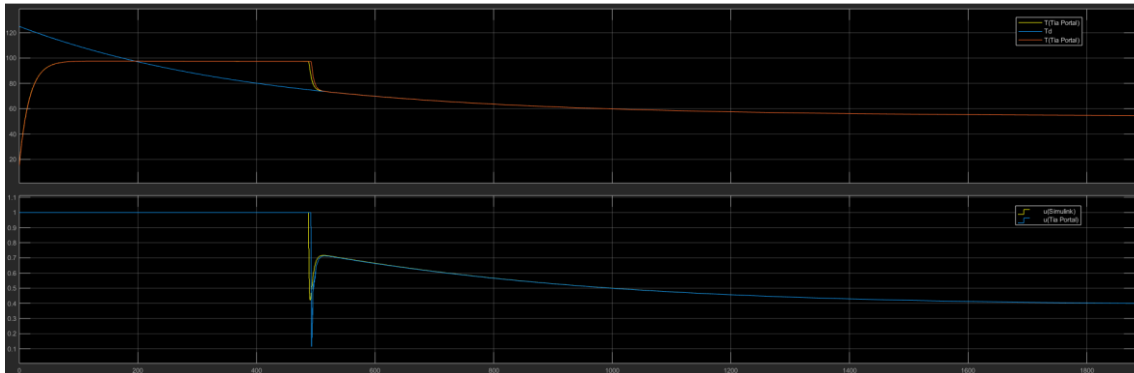


Ilustración 46 Gráficas comparativa de la Temperatura y acción de control

Como se puede observar los resultados son satisfactorios la evolución tanto de la acción de control como de la temperatura del reactor son prácticamente idénticas por lo que podemos concluir que sí es posible realizar una simulación de un proceso industrial mediante los programas utilizados.

Para comprobar que la evolución de las concentraciones de especies es la buscada se ha colocado un scope que mida a medida que se produce la reacción tanto la cantidad de especies de A como de B.

Concentraciones de A

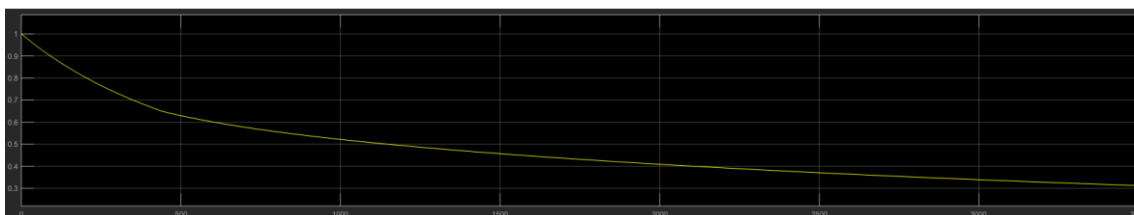


Ilustración Gráfica de la Concentración especies de A obtenida con scope elaboración propia

Al principio de la reacción es decir en el instante t_0 la cantidad de especies de A es 1 y a medida que avanza la reacción su valor va disminuyendo hasta que al transcurrir una hora su valor alcanza 0.3

Concentración de B

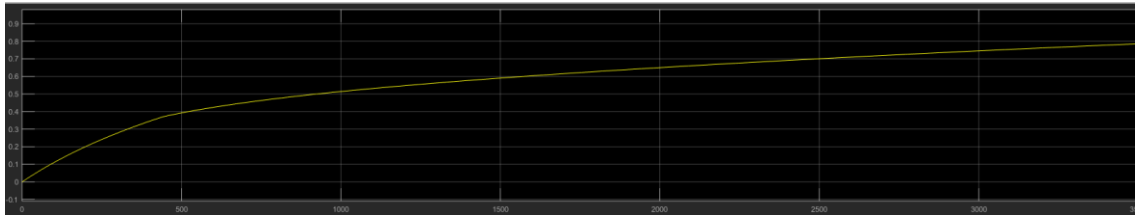


Ilustración 47 Gráfica de la Concentración especies de B obtenida con scope elaboración propia

La concentración de especies de B, es decir la sustancia resultante de la reacción, en el instante inicial ha de ser 0 es decir el componente A todavía no se ha transformado y a medida que avanza el tiempo su valor va aumentando hasta que sobre la hora de la reacción alcanza el valor de 0.8 por lo que podemos concluir que la evolución de las concentraciones en el tiempo es la correcta.

Una vez comprobado que la simulación es posible procederemos a comparar los resultados obtenidos con el control antiwindup el cual nos reducirá el error integral acumulado en las iteraciones y conseguiremos un mejor control sobre el sistema.

Para ello utilizaremos e mismo método que para el control sin antiwindup ejecutaremos la simulación y guardaremos los resultados en un trace para poder compararlos en Matlab. Así pues, una vez guardado los resultados en las mediciones obtenemos la siguiente gráfica:

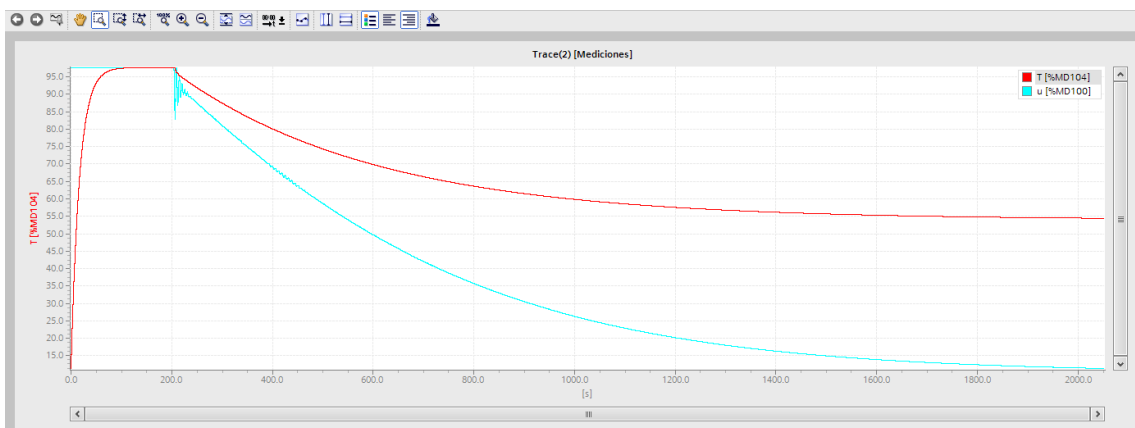


Ilustración 48 Trace de la acción de control, Tia Portal, elaboración propia

Se guarda el archivo en .csv y se modifica el Matlab para poder realizar las comparaciones.

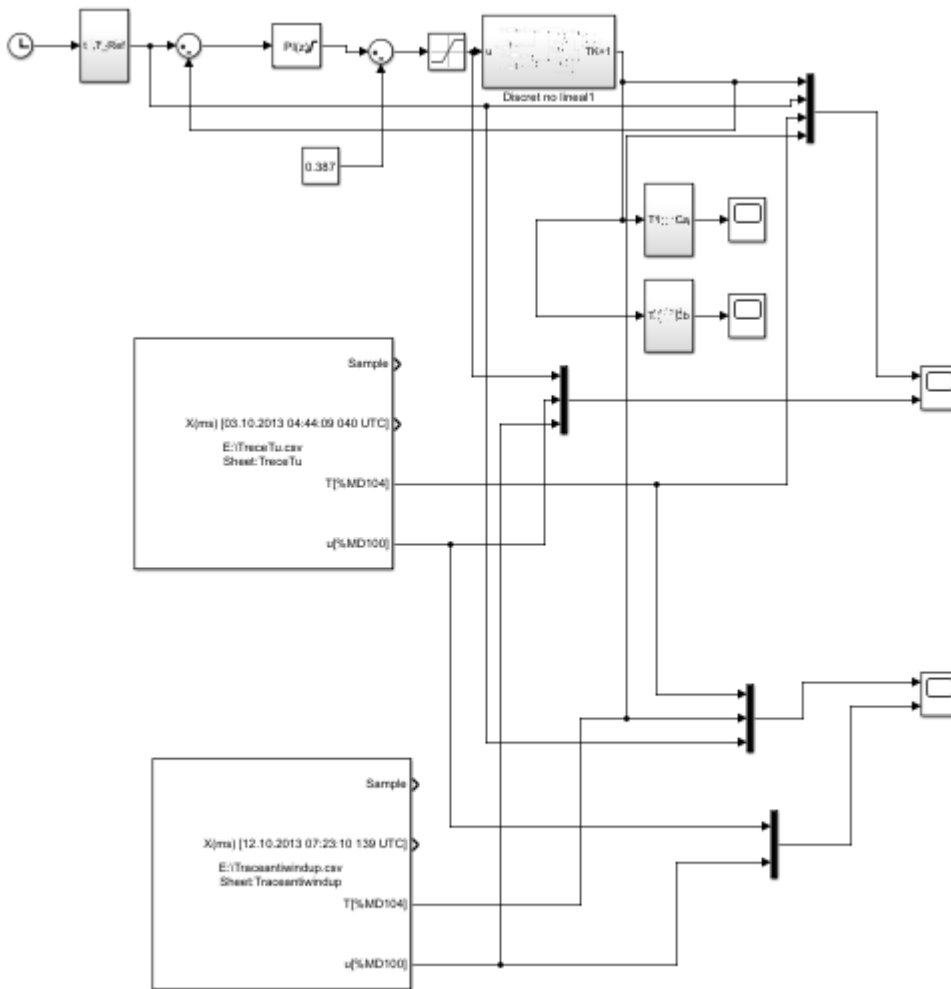


Ilustración 49 Imagen obtenida de Simulink, elaboración propia

Así pues, obtendremos de nuevo dos scopes que nos permita comparar tanto la temperatura de la reacción como la acción de control.

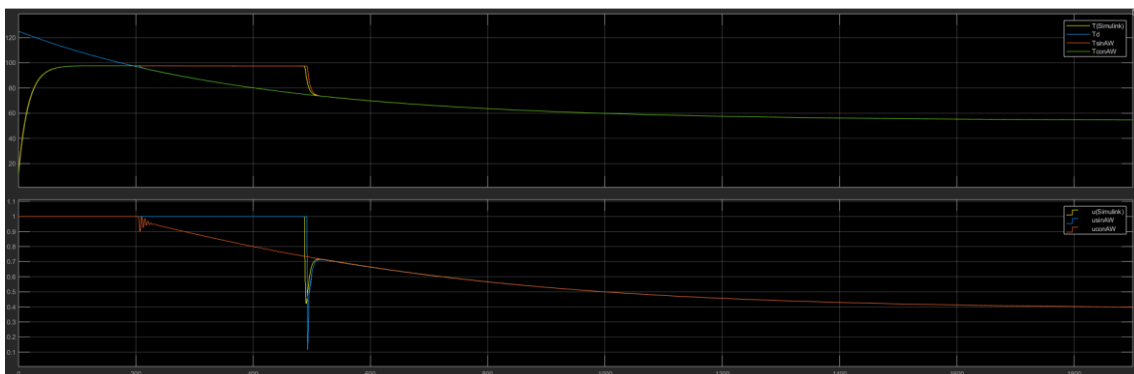


Ilustración 50 Imagen obtenida de Simulink, elaboración propia

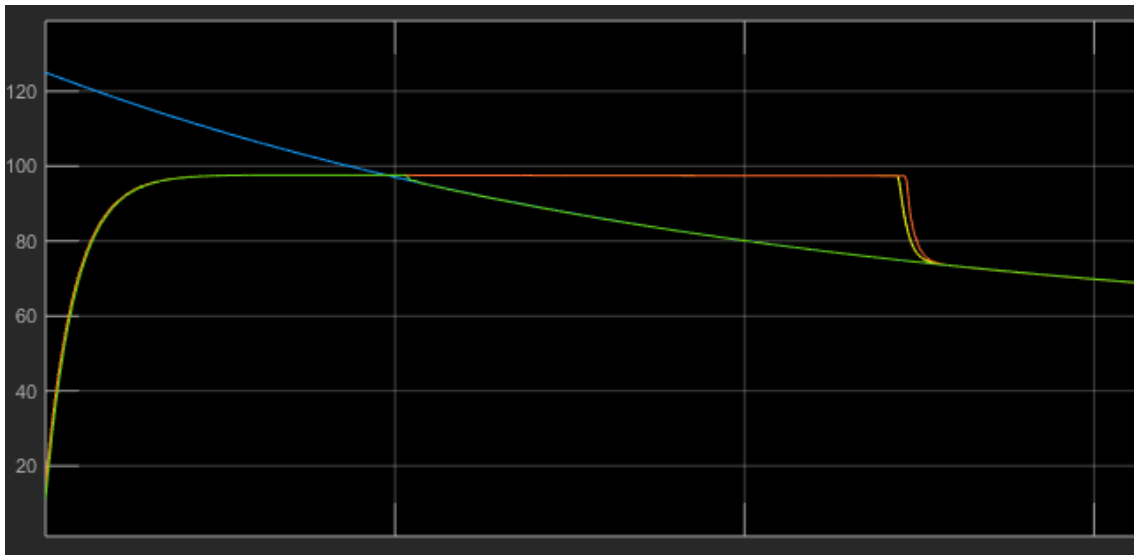


Ilustración 51 Gráfica temperatura del reactor Simulink, elaboración propia

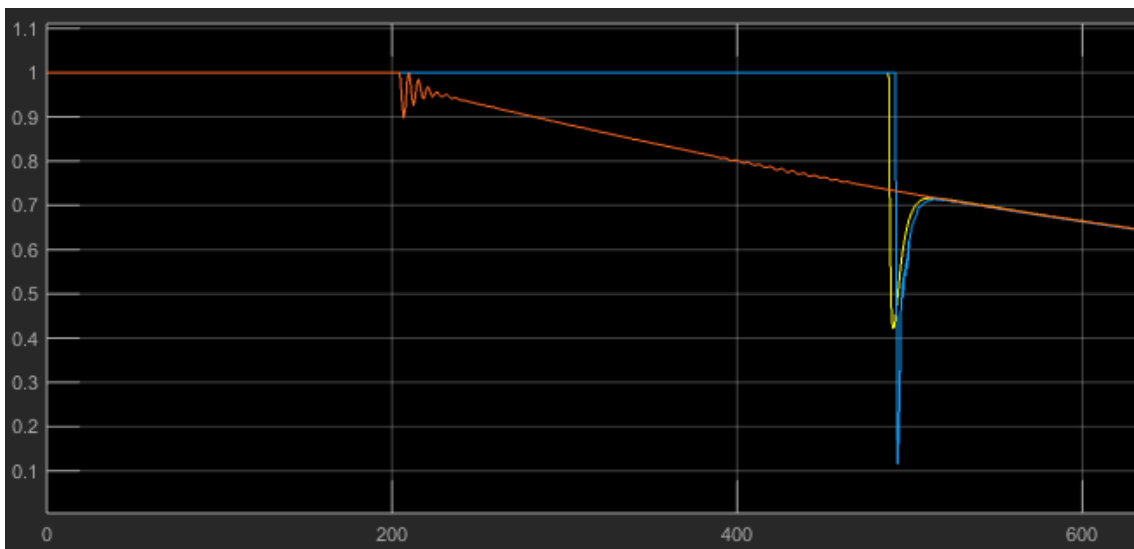


Ilustración 52 Gráfica acciones de control, elaboración propia

En la gráfica de la temperatura se puede observar cómo se ha solucionado el problema del retraso del control de forma que cuando la temperatura deseada baja por debajo de la temperatura máxima del reactor el sistema rápidamente ajusta la temperatura de este. Por lo tanto, se puede afirmar que se ha mejorado el control.

Esto demuestra que gracias a realizar una simulación de un proyecto antes de tener el proceso físico se ha conseguido estudiar el comportamiento del sistema y determinar cual es la forma óptima de controlarlo, en esta gráfica se refleja la importancia de la simulación en proyectos industriales.

BIBLIOGRAFIA

1. JANA, A. K.(2011).”Batch reactor” en *Chemical Process Modelling and Computer Simulation* . PHI Learning. Paginas en las que aparece en este proyecto:
2. LEVINE, W. S.(1996). *Control system applications*. CRC Press; Edición: 1 (27 de diciembre de 1999).
3. LOUIS C. W.(2001). *Handbook of control systems engineering*. Springer.
4. SERBAN AGACHI P.(2006). *Model Based Control: Case Studies in Process Engineering* . Wiley-VCH Verlag GmbH.
5. WAYNE BEQUETTE B.(1998). *Process dynamics : modeling, analysis and simulation*. Prentice Hall; Edición: 01.
6. YOUTUBE, “TIA PORTAL 20/20: Programar en SCL S7 1200 Parte2 AUDIO CORREGIDO” <<https://www.youtube.com/watch?v=f3tWXUWoxiE&t=202s>> Consultado en[03/05/2019].
7. YOUTUBE, “TIA PORTAL 8/20: Plantillas de Funciones FC” < <https://www.youtube.com/watch?v=ssa8udhAgag>> Consultado en[19/04/2019].
8. YOUTUBE, “TIA PORTAL 1/20: Configurar Equipos y Redes” < <https://www.youtube.com/watch?v=xFA YA0Q-N9A&t=270s>> Consultado en[10/04/2019].
9. YOUTUBE, “001 - Basics of Labview - LabView with DMZ” < <https://www.youtube.com/watch?v=QxoJljThkKk>> Consultado en[03/04/2019].
10. Procesos de control. *Función de transferencia del elemento final de control (válvula)* < http://plantscontrol.blogspot.com/2012/02/6_13.html>. Pag [38]

PRESUPUESTO

1. CUADRO DE PRECIOS N.º 1: MANO DE OBRA

Código	Empleado	Salario mensual (€/mes)	Salario (€/h)
MO.GITI	Graduado en Tecnologías Industriales	3200	20

Para el cálculo de este cuadro de precios se ha tenido en cuenta que una jornada laboral contemplada se estima en 221 días/año, que trabajando durante 8h/día hacen un total de 1768 h/año.

2. CUADRO DE PRECIOS N.º 2.

2.1. MATERIALES.

Código	Ud.	Denominación	Precio (€/año)	Precio (€/h)
MT.LAB	u	LabVIEW Base	399	0,226
MT.MAT	u	Matlab & Simulink	1200	0,679
MT.TP	u	Tia Portal	2447,83	1,3845

2.2. MAQUINARIA.

Código	Unidades	Denominación	Precio (€)	Precio (€/año)	Precio (€/h)
MQ ORM	u	Ordenador de sobremesa	1300	162,50	0,735
MQ.PLC	u	PLC Siemens S7 1200	308,47	-	-
MQ.SB	u	Signal Board AQ	97,87	-	-

Para la valoración de la amortización del ordenador de sobremesa se ha tenido que estimar su vida útil. La vida útil estimada ha sido la siguiente:

Ordenador de sobremesa: 8 años.

Para el cálculo de la amortización de la maquinaria se ha supuesto un periodo de amortización de 1 año, en el cual la vida útil es de 1800h/año, equivalente a 225 días laborales con jornadas de trabajo de 8 horas.

3. CUADRO DE PRECIOS N.º 3: PRECIOS PARCIALES.

Nº orden	Unidad	Descripción Medición	Precio	Medición	Precio	Importe
Capítulo 1: Implementación del LabVIEW						
UO 1.01	u	Diseño de la interfaz labVIEW		1	317,245	317,245
UO 1.02	u	Programación de la interfaz labVIEW		1	676,79	676,79
Total Capítulo 1						994,035 €
Capítulo 2: Implementación del Tia Portal						
UO 2.01	u	Programación de autómatas con Tia Portal		1	1.120,934	1.120,934
Total Capítulo 2						1.120,934 €
Capítulo 3: Diseño de los procesos						
UO 3.01	u	Pruebas de los procesos químicos industriales		1	655,26	655,26
UO 3.02	u	Programación de los procesos en Matlab		1	1.496,614	1.496,614
Total Capítulo 3						2.151,874 €
Capítulo 4: Recopilación y tratamiento de información						
U= 4.01	u	Redacción de la documentación		1	1300,000	1300,000
Total Capítulo 4						1.300,000 €

4. CUADRO DE PRECIOS N.º 4: PRECIOS DESCOMPUESTOS.

UO 1.01		Diseño de la interfaz labVIEW			
Código	Ud.	Descripción	Precio (€)	Rend.	Importe (€)
		Diseño de la apariencia y el estilo de la interfaz labVIEW			
MO.GITI	h	Graduado en Tecnologías Industriales	20	15	300,000
MQ ORM	h	Ordenador de sobremesa	0,735	15	11,025
	%	Costes directos complementarios		2	6,22
Total unidad de obra					317,245 €
UO 1.02		Programación de la interfaz labVIEW			
Código	Ud.	Descripción	Precio (€)	Rend.	Importe (€)
		Programación de las distintas funciones de interacción de la página web			
MO.GITI	h	Graduado en Tecnologías Industriales	20	32	640,000
MQ ORM	h	Ordenador de sobremesa	0,735	32	23,52
	%	Costes directos complementarios		2	13,27
Total unidad de obra					676,79 €
UO 2.01		Programación del autómata con Tia Portal			
Código	Ud.	Descripción	Precio (€)	Rend.	Importe (€)
		Programación de las distintas funciones del autómata			
MO.GITI	h	Graduado en Tecnologías Industriales	20	53	1060,000
MQ ORM	h	Ordenador de sobremesa	0,735	53	38,955
	%	Costes directos complementarios		2	21,979
Total unidad de obra					1.120,934 €

UO 3.01		Pruebas de los procesos químico industrial			
Código	Ud.	Descripción	Precio (€)	Rend.	Importe (€)
		Pruebas de los diferentes procesos industriales multivariantes programados			
MO.GITI	h	Graduado en Tecnologías Industriales	20	30	600,000
MQ ORM	h	Ordenador de sobremesa	0,735	30	22,05
MT.MAT		Matlab & Simulink	0,679	30	20,362
	%	Costes directos complementarios		2	12,848
Total unidad de obra					655,26 €

UO 3.02		Programación en labVIEW del proceso			
Código	Ud.	Descripción	Precio (€)	Rend.	Importe (€)
		implementación de las funciones en labVIEW			
MO.GITI	h	Graduado en Tecnologías Industriales	20	70	1400,000
MQ ORM	h	Ordenador de sobremesa	0,735	70	51,471
MT.LAB	h	LabVIEW Base	0,226	70	15,798
	%	Costes directos complementarios		2	29,34
Total unidad de obra					1.496,614 €

UO 4.01	Ud	Redacción de la documentación	Precio	Rend.	Importe (€)
MO.GITI	h	Graduado en Tecnologías Industriales	20	65	1300
Total unidad de obra					1.300,000 €

5. PRECIOS FINALES

Código	Descripción	Subtotal (€)	Importe (€)
1	Implementación del LabVIEW	994,035	
2	Implementación del Tia Portal	1.120,934	
3	Diseño de los procesos	2.151,874	
4	Recopilación y tratamiento de información	1300,000	
	Presupuesto de ejecución material		4.266,843
	Gastos generales 13%		554,689
	Beneficio industrial 6%		256,01
			5.077,54
	IVA 21%		1.066,28
	Total presupuesto		6143,82
			€

PLIEGO DE CONDICIONES

Pliego de condiciones

Características del material empleado

En este trabajo se ha simulado una planta real de procesos químicos con programas que simularán el sistema elegido, en este apartado se citarán las herramientas que se han utilizado a lo largo del trabajo.

Software

El software utilizado a lo largo del trabajo ha sido: Tia Portal, Matlab, Simulink (Herramienta interna de Matlab) y LabVIEW, OPC Server.

Tia Portal

Programa diseñado por la empresa SIEMENS el cual se utiliza para programar el diseño del controlador que se ha utilizado.

Matlab

Herramienta que se utilizarán para realizar las simulaciones y obtener las gráficas con las que se presentarán los resultados y conclusiones.

En el ámbito de la industria 4.0, es muy importante tener acceso a programas de estas características, ya que será el que nos permita desarrollar las simulaciones y los cálculos matemáticos complejos que requieren este tipo de trabajos.

Dentro del propio Matlab se utilizará la herramienta Simulink con la cual se implementan los diagramas de bloques necesarios para obtener las ecuaciones linealizadas, con las que se puede obtener las funciones de transferencia y así controlar el sistema.

LabVIEW

LabVIEW es un software que, en este caso, se utiliza para realizar la virtualización de la planta. Es de gran utilidad, ya que te permite tener acceso a dicha planta desde el exterior, de manera que, mediante tarjetas de adquisición de datos, puedes introducir información dentro de la planta. Esta información que se proporciona puede ser, desde sensores, hasta el mismo autómatas que se va a utilizar en este trabajo, que será el que actúe como controlador.

NI Servidor OPC

Por último, el OPC server será el encargado de comunicar el PLC con la planta simulada. Para este trabajo se ha utilizado el servidor OPC que te ofrece el propio LabVIEW es decir el NI Server OPC en el anexo se incluirá una guía de su utilización.

Hardware

PLC

En cuanto al hardware utilizado en este trabajo, se utilizará un PLC (Programmed Logic Controller) Siemens S7-1214C junto a una Signal Board AQ.

Este autómatas fue seleccionado por la gran potencia que presenta, y motivado por la curiosidad y hambre de conocimientos sobre este campo, la automática, como quiero dirigir mi futuro hacia esta rama me pareció buena idea empezar a programar autómatas más complejos y ver su aplicación real fuera de las aulas.

Este PLC estará conectado mediante un servidor OPC al LabVIEW y viceversa de manera que permite el flujo de información entre ambos.

Monitor

Se necesitará un monitor para mostrar los datos en pantalla y así ver el funcionamiento de esta.

Normativa

Se procede a citar todas las normas y legislaciones que el trabajo ha de cumplir:

1. Norma IEC-1131, en la que se estandarizan todos los lenguajes de programación de autómatas para evitar incompatibilidades entre ellos.
2. Reglamento Electrotécnico de Baja Tensión (REBT)
3. Estándar internacional IEEE 802.3, en el que se estandarizan las redes basadas en Ethernet.
4. Manual de TIA Portal V13
5. Manual de Matlab y Simulink R2018a
6. Manual LabVIEW 2014

ANEXO

LabVIEW

Con este anexo vamos a explicar de forma más detallada como se ha introducido el reactor batch en labVIEW para poder controlarlo. Así pues, explicaremos las principales herramientas utilizadas para implementar el sistema y de las partes que lo compone.

Para empezar se mostrará el programa completo de labVIEW el cual esta compuesto por dos partes:

Front Panel:

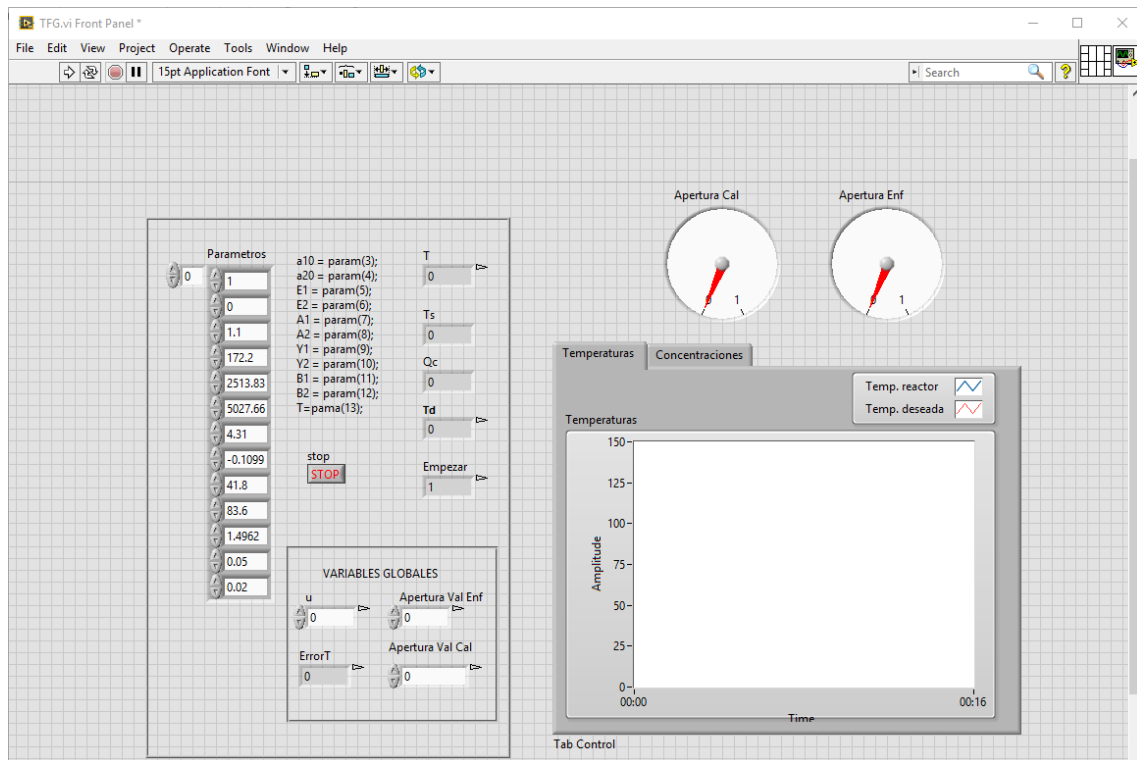
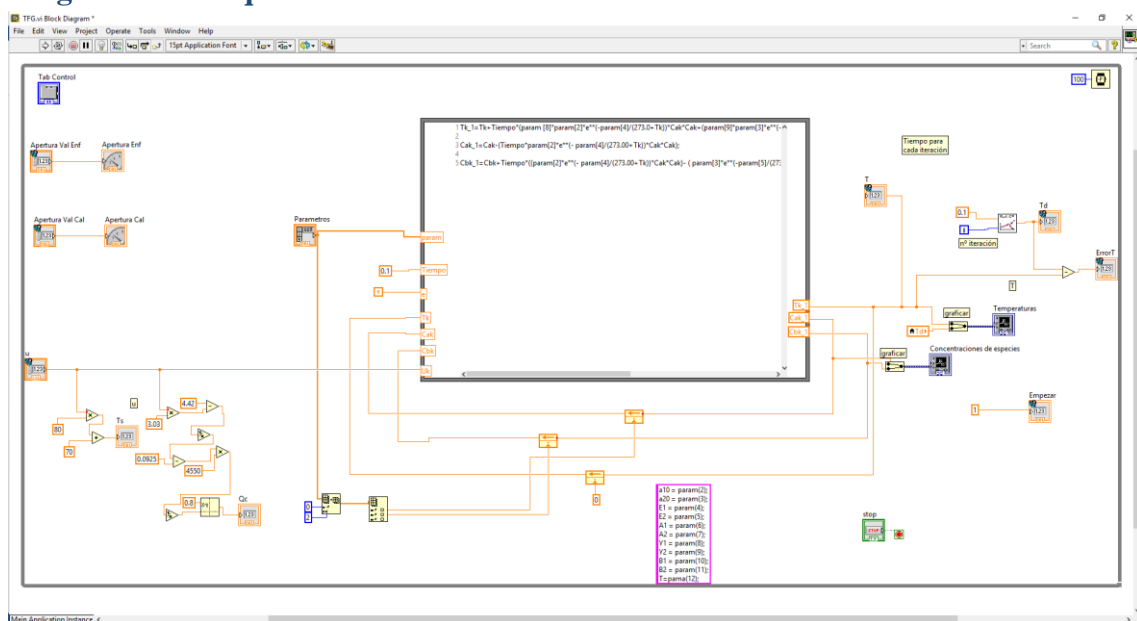


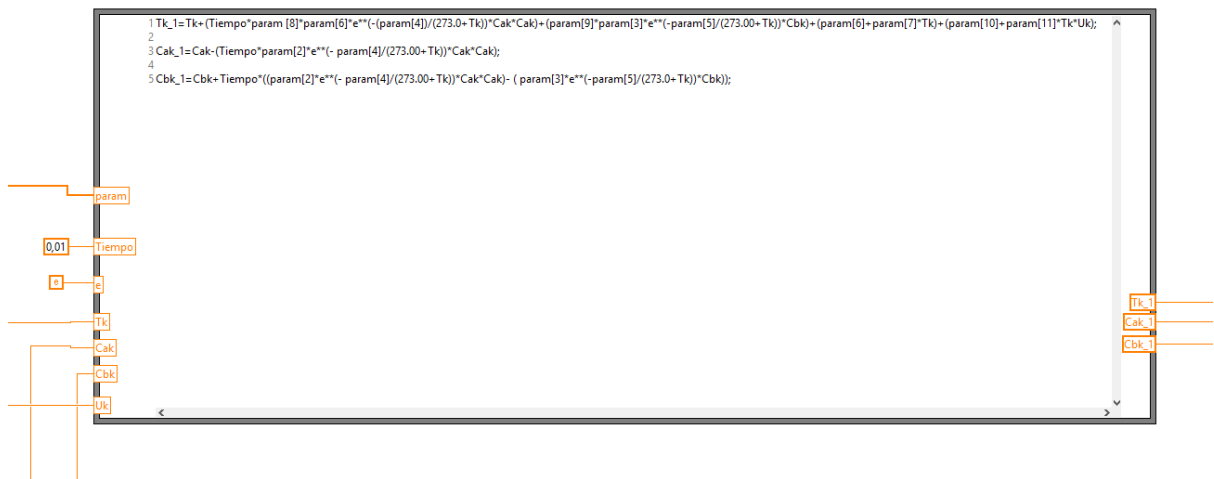
Diagrama de bloques:



Formula Node

Se procede a explicar el proceso seguido para la implementación de la planta virtualizada en el programa LabVIEW. Este software tiene la peculiaridad de ser programación gráfica, pero en este caso y ya que el programa lo permite utilizaremos la programación mixta, parte gráfica y parte escrita. Para ello utilizaremos la herramienta fórmula node donde introduciremos nuestras ecuaciones discretizadas, previamente descritas en el punto 2.3 de la memoria.

La implementación de las ecuaciones ha sido de la siguiente manera:



El formula node nos permite introducir ecuaciones que resolverá sin necesidad de poner cada operación de forma gráfica, de esta forma se nos presenta un cuadro donde hemos de añadir las fórmulas halladas por Euler.

Al tratarse de ecuaciones discretizadas tenemos el problema que necesitamos el valor de la anterior iteración para calcular los nuevos valores para ello hemos utilizado “Feedback Node” el cual nos permite guardar el valor de la anterior iteración y utilizarlo como variable de entrada para la siguiente.



Donde el valor de la salida se guardará de nuevo a la variable que nosotros queramos en la izquierda. La parte inferior se utiliza para inicializar la variable para que pueda calcular la primera iteración, en caso de que no se ponga nada su valor será 0.

En los bordes del cuadro el programa nos permite introducir las variables, donde a la izquierda se introduzcan las variables de entrada y a la derecha colocaremos las variables de salida.

De esta forma tendremos:

Variables de entrada:

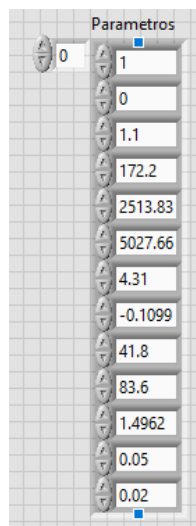
- param
- Tiempo=0.01s
- e (ya que nos facilitará la introducción de este valor en nuestras ecuaciones)
- Tk
- Cak
- Cbk
- uk

Variables de salida:

- Tk_1
- Cak_1
- Cbk_1

Una vez obtenido la Temperatura la mandaremos mediante el servidor OPC a nuestro PLC.

La variable de entrada param es un vector de parámetros donde hemos introducido nuestras constantes de manera que se puedan cambiar fácilmente desde el panel frontal en caso de que queramos modificarlos.

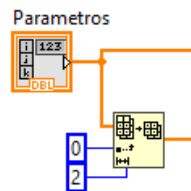


Cada parámetro representa una constante siguiendo la siguiente tabla.

```
a10 = param(3);  
a20 = param(4);  
E1 = param(5);  
E2 = param(6);  
A1 = param(7);  
A2 = param(8);  
Y1 = param(9);  
Y2 = param(10);  
B1 = param(11);  
B2 = param(12);  
T = param(13);
```

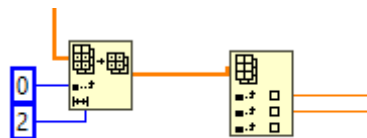
Los parámetros 1 y 2 son respectivamente los valores iniciales de C_A y C_B que no están incluidas en las ecuaciones pero que las necesitamos en la primera iteración por lo que serán los valores que se introducirán en la línea de debajo del Feedback node como hemos indicado anteriormente.

Para ello lo que se hará será, primero, crear un nuevo vector en el que se incluya los dos primeros valores de nuestro vector param, lo haremos utilizando Array Subset Function



en el que introduciremos el vector param, que será el vector del cual extraemos los valores, seguido del número en el que queremos empezar a coger valores, en este caso 0, y el número de parámetros que cogeremos, en nuestro caso 2.

Ahora necesitamos dividir este vector en dos para poner el C_{A0} en el FeedBack Node de la C_A al igual que pasa con el FeedBack Node de C_B para ello utilizaremos el Index Array Function



el cual nos permite separar nuestro vector en sus distintos parámetros.

Con esto se obtendrán los valores de la temperatura del reactor en función de la u , valor que nos vendrá del PLC

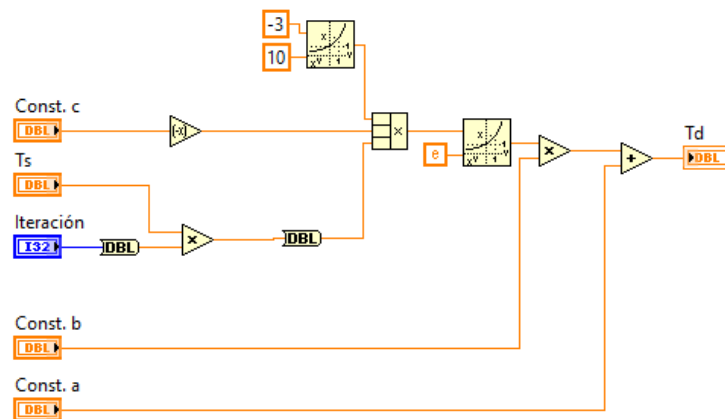
Variables del sistema

A continuación, se explicarán las variables que forman parte de nuestro sistema y cómo las obtendremos.

Td

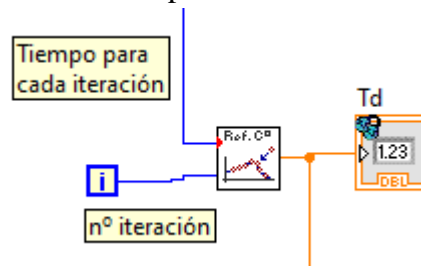
Temperatura deseada, este valor nos permite conocer la Temperatura que debe seguir nuestro reactor para que la reacción se lleve a cabo de forma óptima.

Para conseguir la Temperatura deseada crearemos un sub vi, es decir una función a la que llamaremos para que nos calcule la Td en función del tiempo. Sigue la ecuación (2.18)



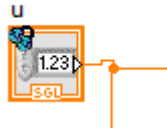
En el cual introduciremos el valor del tiempo (Ts) y la iteración en la que se encuentra el while de nuestro programa. Nos devolverá Td, Temperatura deseada la cual enviaremos a nuestro PLC mediante el servidor OPC.

Haremos un subvi del esquema anterior para añadirlo a nuestro programa principal.



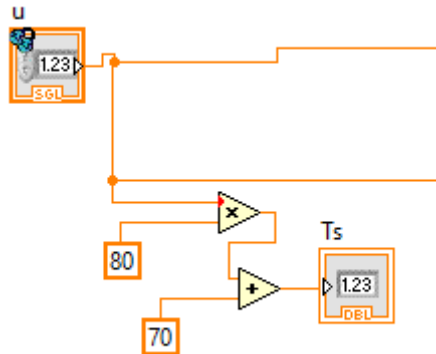
U

Este parámetro es el más importante de nuestro sistema ya que con él, variamos tanto la Temperatura de la Camisa como el caudal de refrigeración con el fin de conseguir que la Temperatura del reactor se asemeje lo máximo posible a nuestra Temperatura deseada. Esta variable la obtendremos mediante el servidor OPC desde el PLC ya que es la variable a controlar.



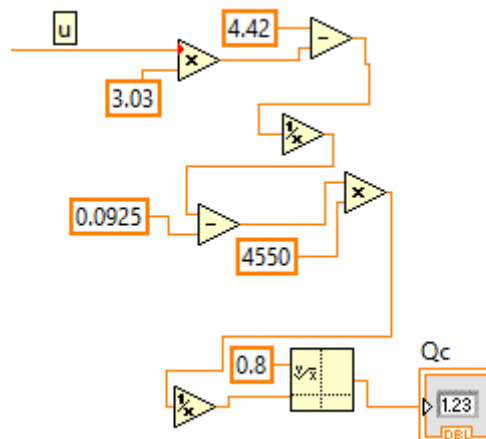
Ts

Esta temperatura no nos vendrá directamente del PLC para conocer esta Temperatura aplicaremos la fórmula que nos relaciona la u con Temperatura de la camisa (Ts) (2.19)



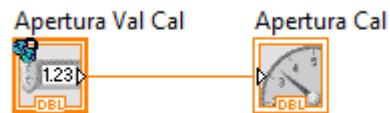
Qc

Para el circuito de refrigeración de nuestro sistema hemos supuesto la Tc constante ya que se trata de Agua a temperatura ambiente 25°C lo que conseguirá regular la Temperatura del reactor será el Caudal que ofrezca la válvula para calcularlo aplicaremos la fórmula que nos relaciona la u con el Caudal Q descrita al principio del Capítulo 2 (2.1)



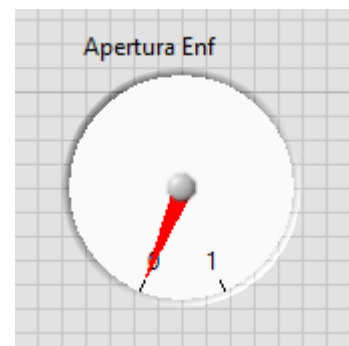
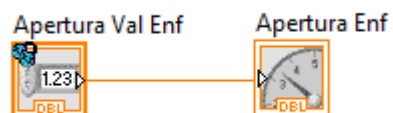
Apertura Válvula Calefactor

Como se ha mencionado anteriormente el Regulador PLC también hará un control sobre las válvulas, las cuales regulan el caudal de vapor que entra en la camisa, por lo que recibiremos por servidor OPC el valor de la apertura que tendrá la válvula siendo 0 completamente cerrada y 1 completamente abierta.



Apertura Válvula Calefactor

Al igual que la válvula anterior, tenemos otra válvula que regula el caudal de agua que entra en el circuito refrigerante la cual será regulada por nuestro PLC y enviada mediante servidor OPC a el LabView

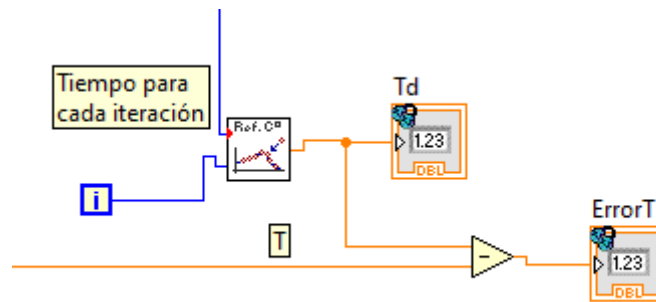


Error Temperatura

Para poder controlar la diferencia de Temperatura entre el reactor y la temperatura se ha decidido introducir esta variable en el sistema de forma que en nuestro panel frontal aparezca su progresión en el tiempo.

Con esto se pretende conocer la efectividad del controlador de forma que cuanto menor sea el valor más nos estamos acercando a los valores deseados.

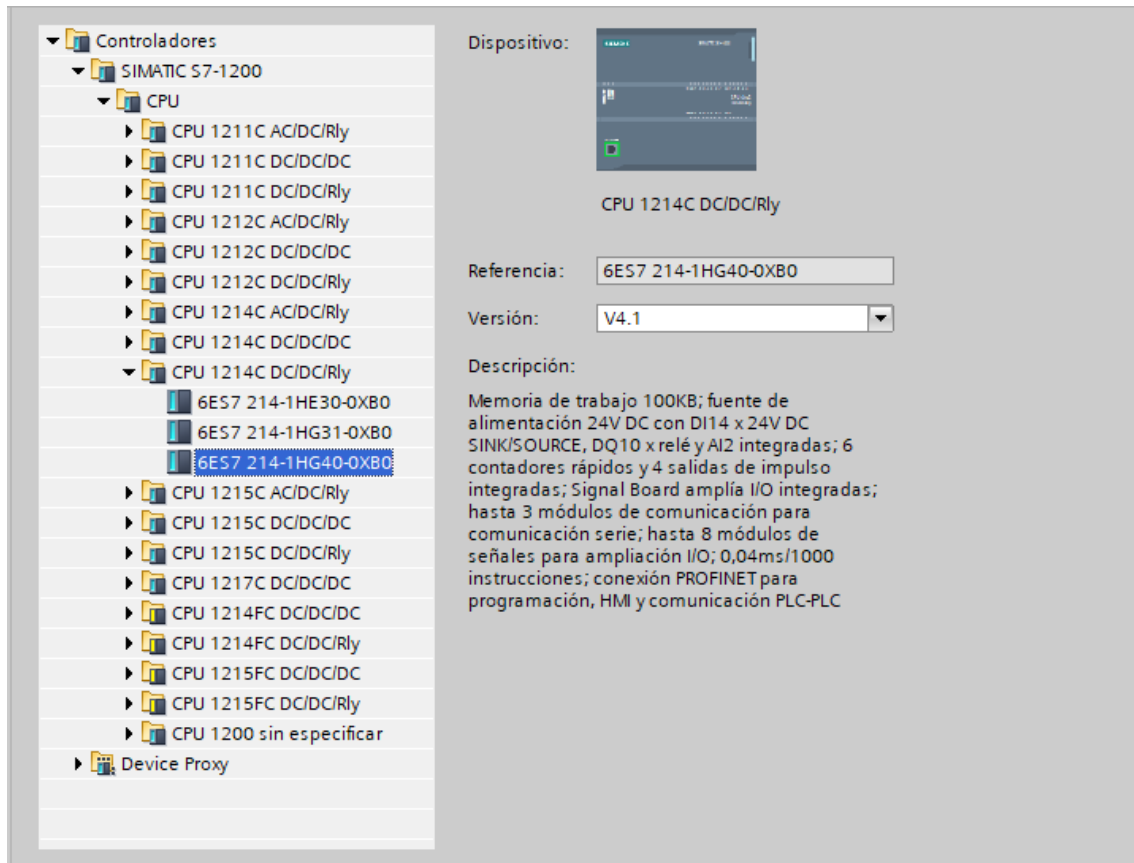
Para obtenerlo se ha realizado una resta entre la Temperatura deseada proveniente de su Subvi correspondiente y de la Temperatura del reactor obtenida del formula node.



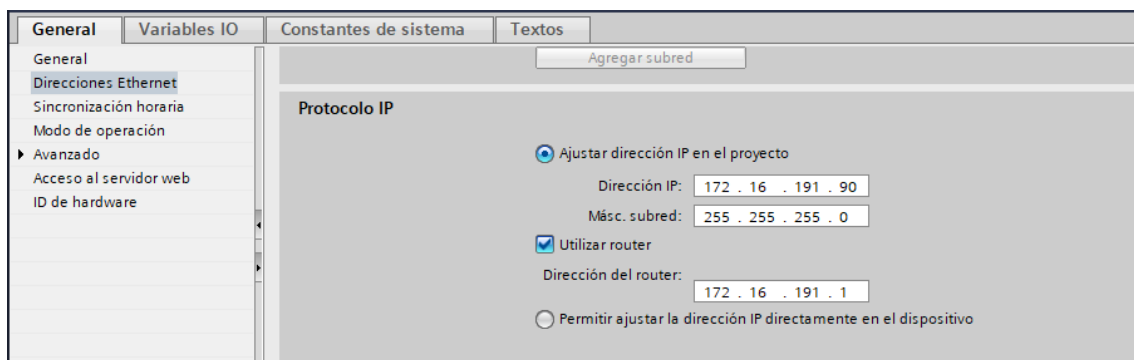
TIA PORTAL

En este apartado del anexo se explicará las partes que componen el programa de Tia Portal utilizado para controlar la planta:

1. El primer paso será introducir el PLC que se utiliza en nuestro caso es un S7 1200 elegiremos nuestro CPU y su versión



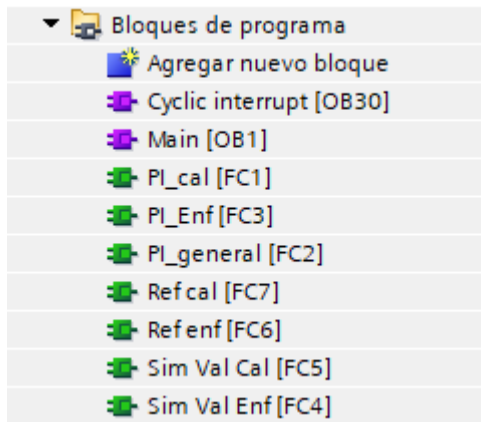
Una vez elegido el autómatas tendremos que introducirle su dirección IP, esta opción se encuentra en Direcciones Ethernet.



Una vez configurado el PLC se podrá subir el programa diseñado para controlar el sistema.

Vamos a mostrar todas las partes que componen nuestro programa:

Bloques de programa



Este apartado lo compone entre otros el Main[OB1], donde hemos inicializado nuestras variables gracias a firstScan el cual solo se activa al ejecutarse el programa:



El Cyclic interrupt [OB30] el cual se ejecuta cada 100ms y donde se ejecutaran las funciones de los controladores

- ▶ **Segmento 1:** Calculo error Temperatura ETk
- ▶ **Segmento 2:** Regulamos la Temperatura del reactor
- ▶ **Segmento 3:** Calculo error Qcalef
- ▶ **Segmento 4:** Calculo parametro de regulacion
- ▶ **Segmento 5:** Calcular la Uc referencia enfriador
- ▶ **Segmento 6:** Calculamos Uc
- ▶ **Segmento 7:** Calculo error Qenf
- ▶ **Segmento 8:** Regulador del circuito de refrigeración

Funciones:

PI_Cal:

```
1 #Uk := 0.0456 * (#Qk * (1 + #AT) - #Qk_1) + #Uk_1;  
2 IF #Uk > 1.0 THEN  
3   #Uk := 1.0;  
4 END_IF;  
5 IF #Uk < 0.0 THEN  
6   #Uk := 0.0;  
7 END_IF;  
8 #Uk_1 := #Uk;  
9 #Qk_1 := #Qk;
```

PI_Enf:

```
1 #Uk := 0.045 * (#Uck * (1 + #AT) - #Uck_1) + #Uck_1;  
2 IF #Uk > 1.0 THEN  
3   #Uk := 1.0;  
4 END_IF;  
5 IF #Uk < 0.0 THEN  
6   #Uk := 0.0;  
7 END_IF;  
8 #Uk_1 := #Uk;  
9 #Uck_1 := #Uck;
```

PI_General:

```
1 #ETk := "Td" - "T";
2 #Uk := 0.1714 * (#ETk-#ETk_1) + 0.033 * #ETk_1 * #AT+#Uk_1;
3 #Uk_1 := #Uk;
4 #Uk := #Uk + 0.387;
5
6 IF (#Uk > 1.0) THEN
7     #Uk := 1.0;
8 END_IF;
9 IF (#Uk < 0) THEN
10    #Uk := 0.0;
11 END_IF;
12
13
14 #ETk_1 := #ETk;
```

Ref Cal

```
1 #Qref := 0.0363 * (80 * #u + 50);
```

Ref Enf

```
1 #Ucref := (1.39-4.42) * #U + 4.42;
```

Sim Val Cal

```
1 #Qck := (1 / (1 + 0.01 / #AT)) * (#Ur * 4.356 + (0.01 / #AT) * #Qck_1) + 1.815;
```

Sim Val Enf

```
1 #Uck := (1 / (1 + (0.01 / #AT))) * (4.42 * #Ur + 0.01 / #AT * #Uck_1);
```

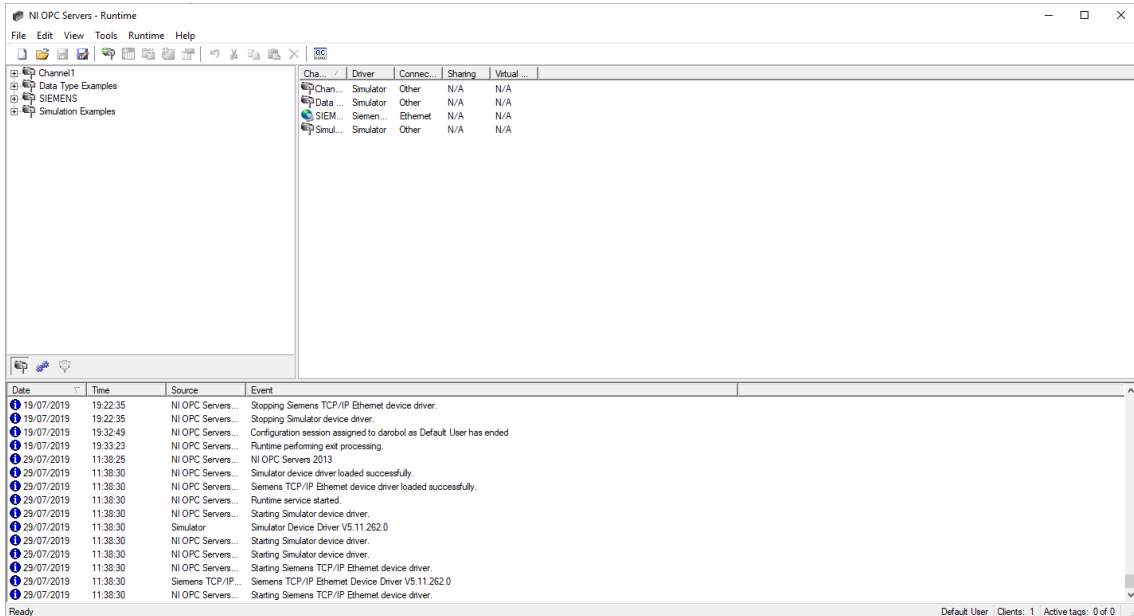
Tabla de Variables

A continuación, se adjuntarán las variables utilizado en nuestro programa:

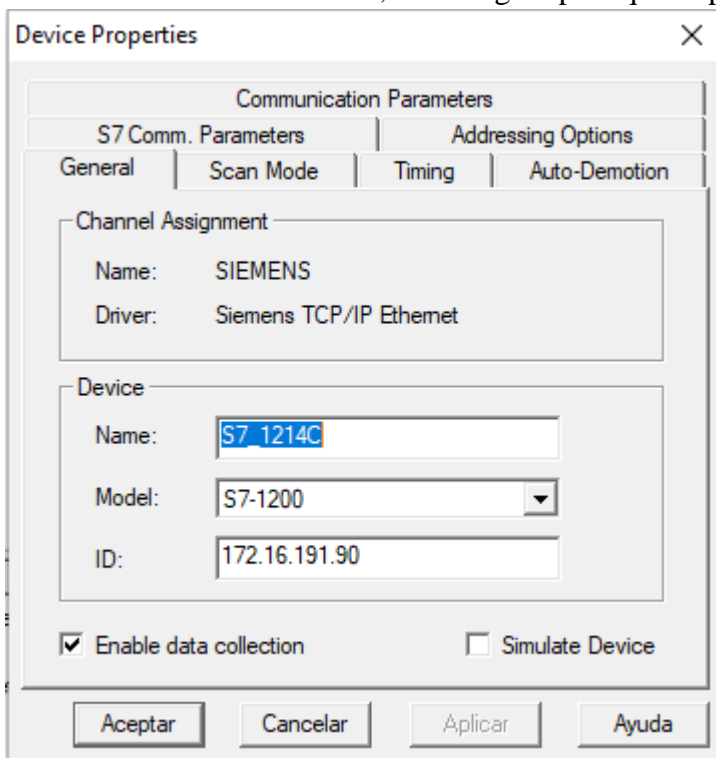
Variables PLC								
	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	
1	u	Tabla de variabl...	Real	%MD100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	T	Tabla de variables e..	Real	%MD104	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Td	Tabla de variables e..	Real	%MD108	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	A_Vcal_k	Tabla de variables e..	Real	%MD120	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	A_Vcal_k_1	Tabla de variables e..	Real	%MD124	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	A_Venf_k	Tabla de variables e..	Real	%MD128	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	A_Venf_k_1	Tabla de variables e..	Real	%MD132	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Qck	Tabla de variables e..	Real	%MD136	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Qck_1	Tabla de variables e..	Real	%MD140	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Uck	Tabla de variables e..	Real	%MD144	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Uck_1	Tabla de variables e..	Real	%MD148	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Clock_10Hz	Tabla de variables e..	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Clock_5Hz	Tabla de variables e..	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Clock_2.5Hz	Tabla de variables e..	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Clock_2Hz	Tabla de variables e..	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Clock_1.25Hz	Tabla de variables e..	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Clock_1Hz	Tabla de variables e..	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Clock_0.625Hz	Tabla de variables e..	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	Clock_0.5Hz	Tabla de variables e..	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	System_Byte	Tabla de variables e..	Byte	%MB1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	FirstScan	Tabla de variables e..	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	DiagStatusUpdate	Tabla de variables e..	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	AlwaysTRUE	Tabla de variables e..	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	AlwaysFALSE	Tabla de variables e..	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	Qref	Tabla de variables e..	Real	%MD152	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	Ucref	Tabla de variables e..	Real	%MD156	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	Ur_enf	Tabla de variables e..	Real	%MD160	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	Ur_cal	Tabla de variables e..	Real	%MD164	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	Tiempo	Tabla de variables e..	Time	%MD168	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	EUc	Tabla de variables e..	Real	%MD172	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	EQc	Tabla de variables e..	Real	%MD176	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	ETk	Tabla de variables e..	Real	%MD180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	ETk_1	Tabla de variables e..	Real	%MD184	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	u_1	Tabla de variables e..	Real	%MD188	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	Clock_Byte(1)	Tabla de variables e..	Byte	%MBO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	Empezar	Tabla de variables e..	Real	%MD200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

NI Servidor OPC

El Servidor OPC utilizado en este trabajo ha sido el que viene incorporado en National Instruments a continuación se explicarán los pasos que se han seguido para su utilización: El primer paso será buscar la carpeta de National Instruments y haremos click en OPC Server Configuration, se abrirá la siguiente pestaña:



Se ha de crear un canal por el cual se efectúe el flujo de información en nuestro caso se ha creado el canal SIEMENS, se configura para que se pueda conectar con el controlador



Se añade el modelo y la IP del controlador.

El siguiente paso será crear las etiquetas (Tags) las cuáles serán las variables que queramos compartir

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
A_Val_Cal	MD120	Float	100	None	Apertura valvula del Calentador
A_Val_enf	MD128	Float	100	None	apertura de la valvula del enfriador
Empezar	MD200	Float	100	None	boton comienzo
T	MD104	Float	100	None	Temperatura del Reactor
Td	MD108	Float	100	None	Temperatura deseada
u	MD100	Float	100	None	u

Se le introduce un nombre, la dirección de memoria donde se ha guardado la variable en el Tia Portal, una descripción, el tipo de dato en nuestro caso float, si queremos que sea de lectura escritura o ambos y el ratio de escaneo en nuestro ha de ser 100 ms para que coincida con el Tia portal y el LabVIEW.

The screenshot shows the 'Tag Properties' dialog box with the following details:

- General Tab:**
 - Identification:**
 - Name: A_Val_Cal
 - Address: MD120
 - Description: Apertura valvula del Calentador
 - Data properties:**
 - Data type: Float
 - Client access: Read/Write
 - Scan rate: 100 milliseconds
- Note:** The scan rate is only used for client applications that do not specify a rate when referencing this tag (e.g., non-OPC clients)
- Buttons:** Aceptar, Cancelar, Aplicar, Ayuda