

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**CONTROL DE SISTEMAS NO LINEALES CON RETARDO DE
TIEMPO VARIABLE**

**DISEÑO E IMPLEMENTACION DE CONTROLADORES EN UN
SISTEMA EMBEBIDO APLICADOS A UN PROCESO QUIMICO
CON RETARDO DE TIEMPO VARIABLE SIMULADO EN MATLAB
SIMULINK BAJO EL CONCEPTO DE “HARWARE IN THE LOOP”**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRONICA Y AUTOMATIZACION**

EDISON JOEL JIMENEZ GAVILANEZ

edison.jimenez01@epn.edu.ec

DIRECTOR: DIEGO PILAQUINGA A, M.SC.

diego.pilaquinga@epn.edu.ec

Quito, febrero 2022

CERTIFICACIONES

Yo, EDISON JOEL JIMÉNEZ GAVILÁNEZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



EDISON JOEL JIMÉNEZ GAVILÁNEZ

Certifico que el presente trabajo de integración curricular fue desarrollado por EDISON JOEL JIMÉNEZ GAVILÁNEZ, bajo mi supervisión.



Firmado electrónicamente por:
**DIEGO IVAN
PILAQUINGA
ABADIANO**

**DIEGO IVAN PILAQUINGA ABADIANO, M.Sc.
DIRECTOR**

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

EDISON JOEL JIMÉNEZ GAVILÁNEZ

DIEGO IVAN PILAQUINGA ABADIANO, M.Sc.

DEDICATORIA

Dedico este trabajo a mis padres Edison y Nancy y a mi hermano Ariel, quienes me han apoyado y me han dado fuerza para seguir adelante durante todos estos años, quienes han sido una guía y un faro de luz por este sendero llamado vida, para ellos mi eterna gratitud y amor.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
ÍNDICE DE CONTENIDO.....	IV
RESUMEN.....	VII
ABSTRACT.....	VIII
1 INTRODUCCIÓN.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teorico	3
1.4.1 Sistema de control	3
1.4.2 Procesos con retardos	3
1.4.3 Controladores PID	5
1.4.3.1 Sintonización del controlador PID.....	5
1.4.4 Controladores por modos deslizantes (SMC)	6
1.4.4.1 Sintonización del controlador SMC.....	7
1.4.5 Controladores por modos deslizantes dinamicos (DSMC)	8
1.4.5.1 Sintonización del controlador DSMC	8
1.4.6 Controladores en cascada.....	9
1.4.7 Implemetacion de controladores en sistemas embebidos.....	9
1.4.8 Hardware in the loop.....	10
1.4.8.1 Arduino.....	11
1.4.8.2 MatLab Simulink.....	12
2 METODOLOGÍA.....	13

2.1	Implementación del proceso con retardo variable	13
2.2	Esquemas de controladores.....	15
2.2.1	Controlador cascada PID-PID.....	15
2.2.2	Controlador cascada SMC-SMC.....	16
2.2.3	Controlador cascada DSMC-PI.....	17
2.3	Discretización de controladores	19
2.3.1	Discretización esquema PID-PID.....	19
2.3.2	Discretización esquema SMC-SMC.....	19
2.3.3	Discretización esquema DSMC-Pi.....	21
2.4	Implementación de los esquemas de control.....	23
2.4.1	Diagrama de flujo general del programa.	23
2.4.2	Diagramas de flujo de los esquemas de control.....	24
2.5	Sintonización de controladores	25
2.5.1	Sintonización del controlador cascada PID-PID.....	26
2.5.1.1	PID de lazo interno de flujo.....	26
2.5.1.2	PID de lazo externo de temperatura	27
2.5.2	Sintonización del controlador cascada DSMC-PI.....	28
2.5.2.1	PI de lazo interno de flujo	28
2.5.2.2	DSMC de lazo externo de temperatura	29
2.5.3	Sintonización del controlador cascada SMC-SMC.....	30
2.5.3.1	SMC de lazo interno de flujo.....	30
2.5.3.2	SMC de lazo externo de temperatura	31
2.6	Comunicación arduino simulink.....	33
2.6.1	Configuración para envío y recepción de datos en arduino.....	33
2.6.1.1	Recepción de datos en arduino	33
2.6.1.2	Envío de datos desde arduino	33
2.6.2	Configuración para envío y recepción de datos en Simulink	34

2.6.2.1	Recepción de datos en Simulink	34
2.6.2.2	Envío de datos desde Simulink	36
2.7	Desarrollo de hardware in the loop.....	37
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	39
3.1	Resultados.....	39
3.1.1	Resultados de implementación hardware in the loop.....	39
3.1.2	Pruebas ante perturbaciones.....	43
3.1.2.1	Respuesta de temperatura y acción de control del esquema PID- PID.....	43
3.1.2.2	Respuesta de temperatura y acción de control del esquema SMC- SMC.....	45
3.1.2.3	Respuesta de temperatura y acción de control del esquema DSMC- PI.....	46
3.1.2.4	Índices de desempeño.	49
3.2	Conclusiones	50
3.3	Recomendaciones	51
4	REFERENCIAS BIBLIOGRÁFICAS.....	52
5	ANEXOS	53
	ANEXO I	54
	ANEXO II	55
	ANEXO III	62

RESUMEN

El presente trabajo de titulación muestra el diseño e implementación de controladores en cascada tipo PID-PID, SMC-SMC y DSMC-PI dentro de una tarjeta Arduino Mega 2560 la cual funciona bajo el concepto de hardware in the loop con un proceso químico de tanque de mezclado que presenta retardo de tiempo. Dicho proceso se encuentra simulado en el software computacional MatLab Simulink mediante el uso de bloques. Además, muestra conceptos de discretización, tiempos muertos, controladores, entre otros, que permitirán entender el desarrollo del trabajo de titulación.

El documento también describe las formas en las que se discretizó cada uno de los controladores y la forma en la que estos fueron sintonizados.

Para la prueba de los esquemas de control se establece la comunicación entre el ordenador que contiene el proceso y el sistema embebido que contiene los controladores, además se añaden perturbaciones de flujo caliente dentro del tanque de mezclado para poder medir el desempeño de cada uno de los controladores mediante índices ISE e ISCO, siendo así el esquema DSMC-PI el controlador que presenta los índices ISE e ISCO más pequeños de los tres controladores.

PALABRAS CLAVE: SMC, DSMC, esquema, cascada, discretización, Hardware in the loop, retardo de tiempo.

ABSTRACT

The present work shows the design and implementation of cascade controllers type PID-PID, SMC-SMC and DSMC-PI in an Arduino Mega 2560 which works under the concept of Hardware in the loop with a chemical process. This process is simulated in the MatLab Simulink using of blocks.

The document also describes the discretization of each controller and the way in which they were tuned.

A communication is established between the computer that contains the process and the embedded system to test the control schemes. Hot water disturbances inside the mixing tank are also included to measure the performance of each controller through ISE and ISCO indices being DSMC-PI the controller that present the smallest index of the three controllers

KEYWORDS: SMC, DSMC, scheme, cascade, discretization, Hardware in the loop, time delay.

1 INTRODUCCIÓN

Muchos de los procesos existentes dentro de las industrias presentan tiempos muertos que son causados por distintos factores y son difíciles de controlar [1]. En la industria se ha explotado el diseño de controladores PID desde diferentes enfoques, pero ante la presencia de retardos demasiado grandes estos controladores no alcanzan el rendimiento deseado. Por estos motivos en los últimos años muchos ingenieros se han visto motivados en dar una solución a este inconveniente dado por los retardos de tiempo.

Una de estas soluciones fue presentada por M. Coronel y R. Orellana [2], la cual consiste en un controlador en cascada que hace uso de un controlador SMC para ambos lazos. Dicho control presenta un buen desempeño ante errores de modelado. Otra variación de este tipo de esquemas de control es el DSMC que fue propuesto por Marco Herrera [3], el cual hace uso de un control SMC y conceptos de compensadores de modelo interno (Linoya y Altpeter), para poder mejorar la estabilidad del sistema, así como eliminar los problemas de chattering que este tipo de esquemas presentan.

El presente trabajo tiene el propósito de diseñar y poner a prueba controladores de modos deslizantes en sistemas cascada aplicándolos en un proceso químico implementado en Simulink, dicho proceso es el tanque de mezclado que presenta un retardo en el tiempo [1]. Para esto el trabajo muestra un desarrollo de tres esquemas de control, siendo el primer esquema un control en cascada PID-PID para observar cómo le afecta los retardos de tiempo, un segundo esquema SMC-SMC basado en lo propuesto por M. Coronel y R. Orellana [2] y un último esquema DSMC-PI para el cual se toman ciertos conceptos propuestos por Marco Herrera [3], donde los resultados de los tres son comparados mediante índices de desempeño ISE e ISCO.

En adición, se implementan los controladores dentro de una tarjeta embebida (Arduino) para poder probarlos bajo el concepto de Hardware in The Loop (HIL), el cual permite probar los algoritmos de programación en una tarjeta embebida haciendo uso de un proceso o entorno simulado. Cabe mencionar que la simulación del proceso se la lleva a cabo de software MatLab Simulink, en donde se encuentran implementadas todas las ecuaciones del modelo matemático del tanque de mezclado y la parte física (Tarjeta Arduino) contiene los tres controladores es manejada mediante una interfaz que permite seleccionar el esquema de control a utilizar.

1.1 OBJETIVO GENERAL

Diseñar e implementar controladores en cascada tipo PID-PID, SMC-SMC y DSMC-PI en un sistema embebido para aplicarlos a un proceso químico con retardo de tiempo variable bajo el concepto de hardware in the loop.

1.2 OBJETIVOS ESPECÍFICOS

1. Realizar una revisión bibliográfica acerca del diseño y sintonización de controladores en cascada del tipo PID-PID, SMC-SMC y DSMC-PI para implementarlos en un sistema embebido.
2. Implementar en MatLab Simulink los modelos matemáticos del proceso químico con retardo de tiempo para realizar pruebas de perturbaciones de flujo de líquido caliente.
3. Diseñar y sintonizar tres esquemas de control en cascada para un proceso químico con retardo de tiempo.
4. Implementar los esquemas de control en una tarjeta embebida para realizar pruebas de funcionamiento junto con el proceso químico implementado en MatLab Simulink bajo el concepto hardware in the loop.
5. Obtener los índices ISE e ISCO de los tres esquemas de control para comparar su desempeño.

1.3 ALCANCE

Se realizará una revisión bibliográfica acerca del diseño y sintonización de controladores en cascada PID-PID, DSMC-PI y SMC-SMC.

Se realizará una revisión bibliográfica acerca de la implementación de controladores en un sistema embebido.

Se diseñará y simulará en MatLab Simulink un proceso químico con retardo de tiempo variable.

Se diseñarán tres controladores de temperatura discretos en cascada: PID-PID, DSMC-PI y SMC-SMC para aplicarlos a un proceso químico con retardo de tiempo variable y perturbaciones.

Se implementarán los tres controladores de temperatura diseñados en un sistema embebido Arduino.

Se verificará y comparará el desempeño de los controladores implementados mediante el concepto “Hardware in the loop” con el modelo matemático de la planta implementado en MatLab Simulink.

Se realizarán pruebas de perturbaciones de flujo de agua caliente en los tres esquemas de control.

1.4 MARCO TEORICO

1.4.1 SISTEMA DE CONTROL

Se considera como sistema de control a aquel cuyas variables de salida tienen un comportamiento según una orden dada por distintas variables de entrada. Estos se caracterizan por ser más complejos y presentar una mayor estabilidad ante perturbaciones y variaciones que tenga un proceso. Este tipo de sistemas se los encuentra en bucle cerrado como se lo observa en la Figura 1.1, dejando así que el proceso se encargue de tomar decisiones sin necesidad de que una persona intervenga.

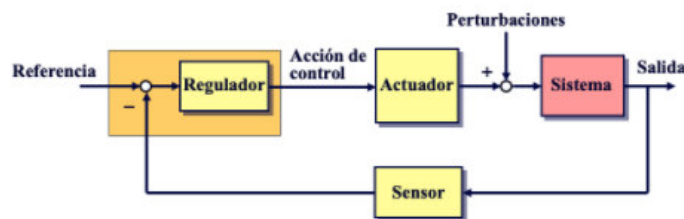


Figura 1.1. Sistema de control en bucle cerrado. [4]

1.4.2 PROCESOS CON RETARDOS

El tiempo muerto o retardo de tiempo se lo puede encontrar en la gran mayoría de procesos dentro de las industrias por lo que la mayoría de los controladores utilizados allí toman en cuenta este parámetro con el fin de realizar un ajuste a estos. Estos retardos se encuentran relacionados con el movimiento de información, masas o energía, pero a su vez pueden ser causados debido a tiempos de procesamiento dentro de sistemas embebidos o a acumulaciones de tiempo de varios sistemas dinámicos conectados uno a continuación de otro.

El efecto que tiene el tiempo muerto sobre un proceso se lo aprecia en la Figura 1.2, donde si se modifica la acción de control en un tiempo determinado, la salida del proceso se verá afectada después de transcurrido el tiempo muerto. Esto provoca que se tenga una función de transferencia no racional de sistema, lo cual lo vuelve más compleja de controlar y analizar.

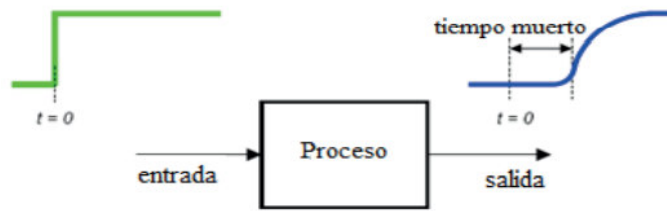


Figura 1.2 Efecto de tiempo muerto en un proceso [1]

Una función desplazada en el tiempo viene dada por:

$$y(t) = x(t - t_0)u(t - t_0) \quad (1.1)$$

Para poder analizar la función mostrada en la Ecuación 1.1, es necesario saber algunas aproximaciones que esta función posee ya que inclusive esto permitirá obtener ciertos modelos que servirán de gran ayuda al momento de sintonizar controladores. Estas aproximaciones se encuentran en forma de función de transferencia por lo que es necesario conocer la función de transferencia del retardo siendo esta la Ecuación 1.2 obtenida al aplicar la transformada de Laplace en la Ecuación 1.1.

$$G(s) = e^{-t_0s} \quad (1.2)$$

Las aproximaciones más comunes para el tiempo muerto son la de Padé y la serie de Taylor, siendo ambas utilizadas para el desarrollo del trabajo de titulación. Para Taylor se considera la aproximación de primer orden la cual se muestra en la Ecuación 1.3:

$$G(s) = e^{-t_0s} \cong \frac{1}{t_0s+1} \quad (1.3)$$

Para el caso de la aproximación de Padé se tiene:

$$G(s) = e^{-t_0s} \cong \frac{1 - \frac{t_0s}{2}}{1 + \frac{t_0s}{2}} \quad (1.4)$$

El tiempo muerto o retardo de tiempo se los suele encontrar en modelos de orden reducido, los cuales cuentan con parámetros de ganancia, tiempo muerto y constantes de tiempo (K, τ y t_0), que pueden ser identificados mediante la respuesta del proceso ante una entrada tipo escalón [7].

Existen una gran variedad de aproximaciones de orden reducido, siendo una de las más utilizadas la aproximación de primer orden con retardo (FOPDT), debido a la simplicidad que esta presenta. El modelo se lo aprecia en la Ecuación 1.5.

$$G(s) = \frac{K}{\tau s + 1} e^{-t_0s} \quad (1.5)$$

Uno de los métodos para hallar los parámetros K , τ y t_0 es el propuesto por Smith el cual consiste en tomar dos puntos de la curva de respuesta, siendo el primero al 23.8% de la respuesta y el segundo al 63.2% [1].

Otro de los métodos existentes es el propuesto por Alfaro donde se calcula los tiempos al 25% y al 75% de la curva de reacción [5].

1.4.3 CONTROLADORES PID

Son los controladores más clásicos y simples que existen y están compuestos por tres tipos de acciones:

- a. Proporcional: se encarga de que la acción de control haga que la salida del sistema en estado estacionario trate de alcanzar en mayor o menor medida el valor de consigna.
- b. Integral: se encarga que la salida del proceso coincida con el punto de consigna de tal manera de que el error entre estos dos sea cero.
- c. Derivativa: esta acción se encarga de mejorar la estabilidad del lazo, la cual puede dar una mayor o menor dinámica al proceso.

Los tres tipos de acción pueden ser combinados, de tal manera que se pueden obtener controles más clásicos tipo P, PD, PI y PID, siendo los dos últimos los aplicados para el desarrollo del trabajo de titulación.

Los controladores PI y PID son presentados en las Ecuaciones 1.6 y 1.7 respectivamente:

$$u_{PI}(t) = k_p e(t) + k_i \int e(t) dt \quad (1.6)$$

$$u_{PID}(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad (1.7)$$

1.4.3.1 Sintonización del controlador PID

La sintonización de estos parámetros puede hacerse mediante distintos métodos. Los propuestos en este caso son el método de sintonización de Dahlin, cuyas ecuaciones para sintonizar los parámetros proporcional, integral y derivativo del controlador son las siguientes [1]:

$$k_p = \frac{0.5 \tau}{K t_0} \quad (1.8)$$

$$k_i = \frac{k_p}{\tau} \quad (1.9)$$

$$k_d = k_p \frac{t_0}{2} \quad (1.10)$$

Otro método utilizado para la sintonización es el propuesto por Chien et al, del cual se describen las ecuaciones para sintonizar un controlador tipo PI con las siguientes ecuaciones [6]:

Para el parámetro proporcional:

$$k_p = \frac{0.6 \tau}{K t_0} \quad (1.11)$$

Para el parámetro integral:

$$k_i = \frac{k_p}{\tau} \quad (1.12)$$

1.4.4 CONTROLADORES POR MODOS DESLIZANTES (SMC)

Este tipo de controladores surgen como ayuda a procesos que presentan demasiadas perturbaciones, las cuales pueden llevar a que un proceso sea inestable. Este tipo de controladores se basa en el uso de modelos por lo que, en procesos completos, como por ejemplo procesos químicos, el desarrollo de estos se vuelve más complejo y a la larga más tedioso. Por tal motivo se plantea una solución la cual consiste en utilizar modelos aproximados como lo es el caso del modelo de primer orden con retardo indicado en el apartado 1.4.2.

La idea principal de este tipo de controladores es definir una superficie en la cual el proceso se deslice hasta poder tener un valor deseado, una interpretación de esto se lo ve en la Figura 1.3. Por lo tanto, antes de empezar con la sintonización de parámetros es necesario definir la superficie de deslizamiento. Dicha superficie puede ser definida por la Ecuación 1.13 [8] la cual actúa sobre el error de seguimiento.

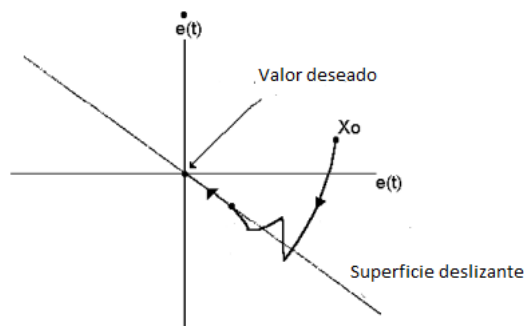


Figura 1.3. Interpretación de un controlador por modos deslizantes.[8]

$$s(t) = \left(\frac{d}{dt} + \lambda \right)^n \int e(t) dt \quad (1.13)$$

Donde:

λ : Es un parámetro de ajuste q ayuda a la definición de la superficie

n : Es el orden del sistema.

Una vez la superficie está definida se debe diseñar la ley de control, que lleva al proceso hacia el valor de consigan. Dicha ley consiste en dos partes:

Parte Continua

$$u_C(t) = f(x(t), R(t)) \quad (1.14)$$

Parte discontinua:

$$u_D(t) = K_d \frac{s(t)}{|s(t)| + \delta} \quad (1.15)$$

Donde K_d es un parámetro de ajuste y δ reduce el problema de chattering, la cual es una oscilación de alta frecuencia que se produce alrededor de un punto de equilibrio [8].

1.4.4.1 Sintonización del controlador SMC

Dependiendo del tipo de superficie que se obtenga a partir de la Ecuación 1.13 las ecuaciones para la sintonización pueden cambiar. En caso de que la superficie de deslizamiento obtenida sea tipo PI como lo es el caso de la Ecuación 1.16, la ecuacion necesaria para la sintonización es la 1.17:

$$s(t) = e(t) + \lambda_1 \int e(t)dt \quad (1.16)$$

$$\lambda_1 = \frac{1}{\tau} \quad (1.17)$$

Si la superficie resulta ser tipo PID como la Ecuación 1.18 la sintonización se realizará con las ecuaciones:

$$s(t) = \dot{e}(t) + \lambda_1 e(t) + \lambda_0 \int e(t)dt \quad (1.18)$$

$$\lambda_1 = \frac{t_0 + \tau}{t_0 \tau} \quad (1.19)$$

$$\lambda_0 = \frac{\lambda_1^2}{4} \quad (1.20)$$

Para la obtención de los parámetros K_d y δ de la ley de control discontinua se utiliza las Ecuaciones 1.21 y 1.22 respectivamente:

$$K_d = \frac{0.51}{|K|} \left(\frac{\tau}{t_0} \right)^{0.76} \quad (1.21)$$

$$\delta = 0.68 + 0.12|K|K_d\lambda_1 \quad (1.22)$$

1.4.5 CONTROLADORES POR MODOS DESLIZANTES DINAMICOS (DSMC)

Este tipo de controlador busca mejorar el rendimiento de los controladores SMC de tal forma que elimina el efecto de chattering que se menciona al final del apartado 1.4.5. Una de las ventajas que también presenta este tipo de controlador es que mejora el desempeño ante perturbaciones en procesos que presentan retardos de tiempo.

Existen algunas variantes de este controlador de las cuales el propuesto para el desarrollo del trabajo es el planteado por Marco Herrera [3] que propone el uso de un compensador de Linoya and Alperter mostrado en la Figura 1.4 para el desarrollo del controlador DSMC donde $G_m(s)$ es el modelo aproximado de la planta y $G_p(s)$ es el modelo interno dada por la Ecuación 1.23.

$$G_p(s) = \frac{k \lambda s}{(\tau s + 1)(1 + \frac{\tau_0}{2}s)} \quad (1.23)$$

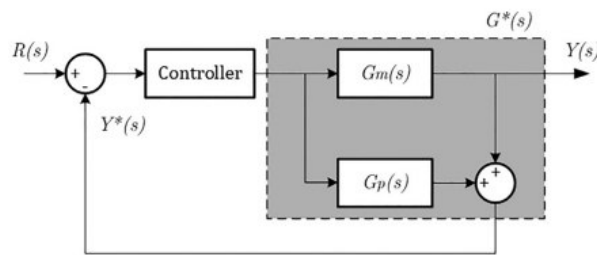


Figura 1.4. Diagrama de bloques del compensador Linoya and Alperter. [3]

Al igual que con el esquema SMC este control también necesita de una superficie que puede ser obtenida a partir de la Ecuación 1.13. En el caso de la ley de control, esta también cuenta con dos partes [3]:

Parte Continua

$$\dot{u}_c(t) = f(x(t), R(t)) \quad (1.24)$$

Parte discontinua:

$$\dot{u}_D(t) = K_d \frac{s(t)}{|s(t)| + \delta} \quad (1.25)$$

Al momento de integrar la parte discontinua del controlador el efecto de chattering se atenúa.

1.4.5.1 Sintonización del controlador DSMC

Para sintonización de este controlador se toma en cuenta una superficie tipo PID como la Ecuación 1.18 donde las ecuaciones necesarias para la sintonización en este caso son:

$$\lambda_1 = \frac{t_0 + 2\tau}{t_0\tau} \quad (1.26)$$

$$\lambda_0 = \frac{\lambda_1^2}{4} \quad (1.27)$$

Para la obtención del parámetro λ del modelo interno se tiene que:

$$\lambda > n \quad (1.28)$$

Para el cálculo de las constantes K_d y δ de la parte discontinua se aplican las ecuaciones 1.21 y 1.22.

1.4.6 CONTROLADORES EN CASCADA

Este tipo de control se utiliza para reducir las perturbaciones que se presenten un proceso a lo mínimo posible. Este control se define como una configuración donde una parte del proceso se la toma como ajuste para otro controlador de realimentación [1]. En la Figura 1.5 se tiene un diagrama de bloques que muestra un control en cascada que consta de dos lazos con sus respectivos controladores.

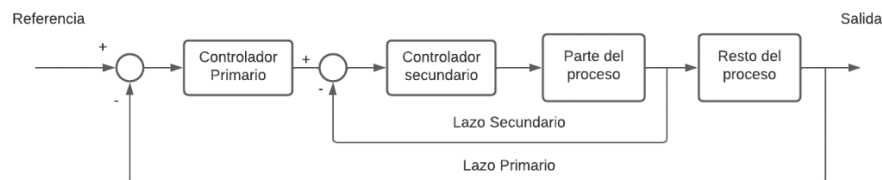


Figura 1.5. Diagrama de bloques de control cascada.

Para la sintonización de este tipo de control, primero se sintoniza el controlador secundario dejando abierto ambos lazos de control. Una vez completada la sintonización del controlador secundario, se conecta el lazo interno y se procede a sintonizar el controlador primario.

1.4.7 IMPLEMETACION DE CONTROLADORES EN SISTEMAS EMBEBIDOS

La mayoría de las señales existentes en el entorno son de tipo analógico por lo que si se desea procesarlas en un sistema digital es necesario discretizarlas, ya que de otro modo un sistema embebido no sabrá como interpretar las señales analógicas de manera adecuada. Por este motivo es necesario el discretizar cada uno de los controladores para poder implementarlos dentro de una tarjeta embebida.

Una señal discreta es una secuencia de muestras de una variable en el dominio del tiempo como se lo observa en la Figura 1.6, donde si se toman suficientes muestras se puede obtener una representación discreta lo más parecida a la señal en el dominio del tiempo.

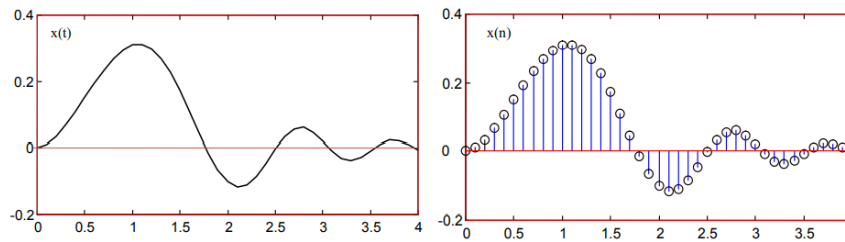


Figura 1.6. Señal analógica y muestreada [9].

Debido a que a nivel industrial la mayoría de los procesos no presentan una respuesta oscilatoria, es difícil determinar el ancho de banda para obtener el tiempo de muestro, por lo que se considera que deben tomarse entre 10 a 30 muestras de tiempo cuando se tienen sistemas que presenten una respuesta de primer orden. Con lo mencionado anteriormente se tiene la Ecuación 1.29 [9], que indica como se obtiene el tiempo de muestreo, el cual es de mucha ayuda cuando se discretizan los distintos tipos de controladores.

$$T = \frac{\tau}{\text{numero de muestras}} \quad (1.29)$$

1.4.8 HARDWARE IN THE LOOP

Es una técnica que permite conectar señales que envía un controlador en físico hacia un proceso que se encuentra simulado, de tal manera que el sistema embebido piensa que se encuentra conectado a un proceso en la vida real [10], un ejemplo de esto se lo aprecia en la Figura 1.7 que muestra una conexión básica de comunicación entre una tarjeta Arduino y un ordenador el cual contiene un proceso. Este tipo de pruebas son simulaciones que se realizan en tiempo real y permiten saber si la programación desarrollada dentro del sistema embebido opera dentro de los rangos requeridos. El desempeño de estas pruebas depende mucho de la complejidad de la dinámica del proceso, así como del hardware que se utiliza.

Para realizar este tipo de pruebas es necesario tener ciertos componentes los cuales son:

- a. Un sistema embebido: Encargado de procesar las señales recibidas desde el software computacional. Para este caso se hace uso de una tarjeta Arduino junto con su software de programación[11].
- b. Un software: Almacena los modelos matemáticos de los procesos. El software utilizado es MatLab Simulink.

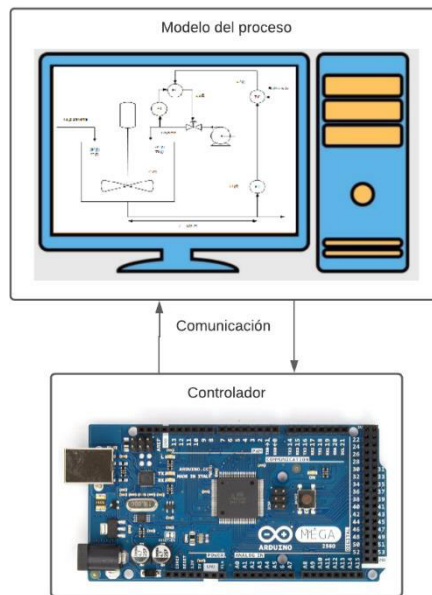


Figura 1.7. Esquema básico de Hardware in the loop

1.4.8.1 Arduino

Arduino es un medio de desarrollo de hardware libre que se encuentra basada en una placa electrónica que incluye un microcontrolador de la marca Atmel con la capacidad de reprogramación, por lo que la mayoría de los modelos comparten librerías y arquitecturas. Estos dispositivos están montados en una placa electrónica llamada PCB (Placa de circuito impreso) [11]. Existe una gran variedad de modelos presentando una gran variedad de características, entre ellas el tamaño de la placa, el modelo de microcontrolador, el número de pines, entre otros.

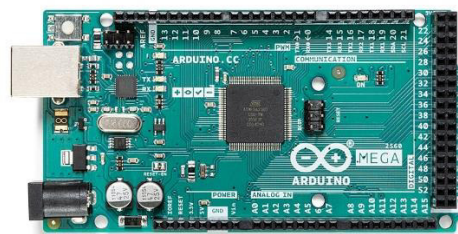


Figura 1.8. Arduino Mega 2560 [12]

Para la implementación de los controladores se hace uso de la tarjeta Arduino Mega 2560, mostrada en la Figura 1.8, que contiene el microcontrolador ATmega2560. Esta placa contiene un total de 54 pines digitales E/S, 4 puertos de comunicación serial, 16 entradas analógicas, una conexión USB, entre otras [12].

1.4.8.2 MatLab Simulink

Simulink es un software de simulación que hace uso de diagramas de bloques para la simulación y modelación de distintos sistemas dinámicos. Este programa permite realizar verificaciones de sistemas, diseño de sistemas, generación de código, adquisición de datos, entre otros [13].

2 METODOLOGÍA

2.1 IMPLEMENTACIÓN DEL PROCESO CON RETARDO VARIABLE

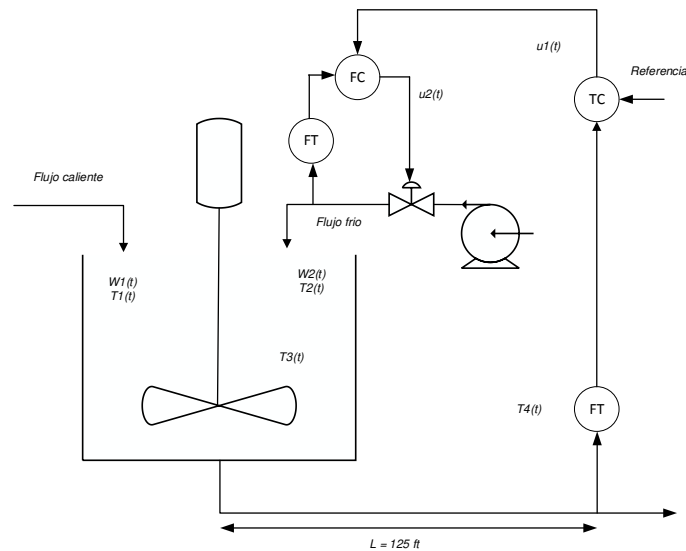


Figura 2.1. Tanque de mezclado con retardo de tiempo.

El proceso que se toma en cuenta para la implementación en Simulink proviene de Camacho y Smith [3], el cual se encuentra compuesto de un tanque que mezcla un fluido caliente ($W_1(t)$) y uno frío ($W_2(t)$). Dicho proceso debe mantener una temperatura entre los 100°F a 200°F para lo cual se agrega un sensor de temperatura, que se encuentra ubicado a 125 ft de distancia del tanque, una válvula y un microcontrolador. Además, al proceso se agrega un sensor de flujo en la tubería que lleva el líquido frío, ya que para el proceso se propone una implementación de controladores en cascada. Todo lo mencionado anteriormente se o puede apreciar en la Figura 2.1 la cual contiene un esquema del sistema.

El tanque cumple con las siguientes condiciones:

- El volumen del líquido en el interior del tanque es constante
- El líquido se encuentra bien mezclado
- Toda la tubería y el tanque están completamente aislados

Las ecuaciones que representan la dinámica del proceso se muestran a continuación:

- a) Balance de energía del tanque de mezclado

$$W_1(t)C_{p1}(t)T_1(t) + W_2(t)C_{p2}(t)T_2(t) - (W_1(t) + W_2(t))C_{p3}(t)T_3(t) = V\rho C v_3 \frac{dT_3(t)}{dt} \quad (2.1)$$

b) Retardo existente entre la ubicación del sensor y el tanque de mezclado

$$T_4(t) = T_3(t - t_0) \quad (2.2)$$

c) Tiempo muerto del proceso

$$t_0 = \frac{LAp}{W_1(t)+W_2(t)} \quad (2.3)$$

d) Ecuación del transmisor de temperatura

$$\frac{dT_O(t)}{t} = \frac{1}{\tau_T} \left[\frac{T_4(t)-100}{100} - T_O(t) \right] \quad (2.4)$$

e) Posicionador de la válvula

$$\frac{dV_p(t)}{dt} = \frac{1}{\tau_{V_p}} [m(t) - V_p(t)] \quad (2.5)$$

f) Ecuación de la válvula

$$W_2(t) = \frac{500}{60} C_{VL} V_p(t) \sqrt{G_f \Delta P_v} \quad (2.6)$$

g) Ecuación del transmisor de Flujo

$$\frac{dFO(t)}{t} = \frac{1}{\tau_F} \left[\frac{W_2(t)-100}{200} - FO(t) \right] \quad (2.7)$$

Donde:

$W_1(t)$: Flujo de líquido caliente [lb/min].

$W_2(t)$: Flujo de líquido frío [lb/min].

$C_p(t)$: Capacidad calorífica de un líquido cuando su presión constante [Btu/lb -°F].

$C_v(t)$: Capacidad calorífica de un líquido cuando su volumen constante [Btu/lb -°F].

$T_1(t)$: Temperatura del flujo caliente [°F].

$T_2(t)$: Temperatura del flujo frío [°F].

$T_3(t)$: Temperatura en tanque de mezclado [°F].

$T_4(t)$: Temperatura del tanque de mezclado más el retardo [°F].

- t_o : Retardo de tiempo [min].
- p : Densidad de la mezcla del tanque [lbm/ft³].
- V : Volumen de la mezcla [ft³].
- $TO(t)$: Salida de transmisor de temperatura en por unidad.
- $V_p(t)$: Posición de la válvula, 0 a 1 (Cerrada o abierta).
- $m(t)$: Salida del controlador en por unidad.
- C_{VL} : Coeficiente de flujo de la válvula [gpm/psi^{1/2}].
- $G_f(t)$: Gravedad específica, adimensional.
- ΔP_v : Caída de presión en la válvula [psi].
- τ_T : Constante de tiempo del sensor de temperatura [min].
- τ_{Vp} : Constante de tiempo del actuador de la válvula [min].
- A : sección transversal de la tubería [ft²].
- L : Longitud de tubería [ft].
- $FO(t)$: Salida de transmisor de flujo en por unidad.
- τ_F : Constante de tiempo de sensor de flujo [min].

2.2 ESQUEMAS DE CONTROLADORES

Para esta sección del documento se explica el desarrollo matemático de los esquemas de control implementados en el proyecto, los cuales se pueden aplicar a sistemas que presentan retardo, como lo es el caso del tanque de mezclado.

2.2.1 CONTROLADOR CASCADA PID-PID

Para este esquema se hace uso de los modelos aproximados FOPDT de las respuestas de flujo y de temperatura, además de un controlador tipo PID para ambos, lazos el cual es determinado a partir de los modelos aproximados.

El controlador tomado en cuenta es un control clásico tipo PID mostrado en la Ecuación 2.8.

$$G_c(s) = \frac{u(t)}{e(t)} = K_p + \frac{K_i}{s} + K_d s \quad (2.8)$$

Para la sintonización se hace uso del método de Dahlin, explicado en el capítulo 1.

2.2.2 CONTROLADOR CASCADA SMC-SMC

Este esquema es más complejo que el esquema en cascada PID-PID, debido a que aquí es necesario definir una superficie, el modelo aproximado del sistema y una ley de control para ambos lazos. Antes de empezar con el proceso de sintonización primeramente se define el tipo de superficie que necesita el lazo de flujo. Para esto es necesario contar con el modelo aproximado de la planta y en el caso de la respuesta de flujo de líquido frío al no presentar un retardo grande se opta por despreciarlo y utilizar un modelo de primer orden como modelo aproximado (Ecuación 2.9).

$$G_f(s) = \frac{x_f(s)}{u_f(s)} = \frac{K_f}{\tau_f s + 1} \quad (2.9)$$

Haciendo uso de la Ecuación (1.X) se tiene como resultado una la superficie tipo PI (Ecuación 2.10).

$$s_f(t) = e_f(t) + \lambda_2 \int e_f(t) dt \quad (2.10)$$

Una vez definida la superficie y sabiendo que este controlador es basado en modelos se obtiene la ley de control, para ello, a partir de la Ecuación 2.9 se obtiene su representación en forma de ecuación diferencial (Ecuación 2.11).

$$\tau_f \dot{x}_f(t) + x_f(t) = K_f u_f(t) \quad (2.11)$$

Sabiendo que:

$$e_f(t) = u_t(t) - x_f(t) \quad (2.12)$$

Y derivando la ecuación de la superficie e igualándola a cero se tiene:

$$s_f \dot{t} = e_f(t) + \lambda_2 e_f(t) = 0 \quad (2.13)$$

Sustituyendo 2.12 y 2.13 en 2.11 al despejar $u_f(t)$ se tiene la ley de control de lazo interno:

$$u_f(t) = \frac{1}{K_f} [\tau_f \widehat{u}_t(t) + u_t(t) + (\tau_f \lambda_2 - 1) e_f(t)] \quad (2.14)$$

Donde $\widehat{u}_t(t)$ es un filtro derivativo que evita el efecto de set-point kick.

$$\tau_u \widehat{u}_t(t) + \widehat{u}_t(t) = \dot{u}_t(t) \quad (2.15)$$

Para la sintonización del lazo de temperatura se tiene un modelo aproximado FOPDT (Ecuación 2.16), que cuenta con dos partes: la primera que es la dinámica de la respuesta y la segunda que es el retardo que presenta el sistema.

$$G_f(s) = \frac{x_t(s)}{u_t(s)} = \frac{K_t}{\tau_t s + 1} e^{-t_0 s} \quad (2.16)$$

Al aplicar la aproximación de Taylor al retardo $e^{-t_0 s}$ se tiene que el modelo del sistema es de segundo orden (Ecuación 2.17) y por ende la superficie necesaria para el controlador es tipo PID (Ecuación 2.18), de acuerdo con la ecuación 1.13 mostrada en el capítulo 1 para la definición de superficies.

$$G_t(s) = \frac{x_t(s)}{u_t(s)} = \frac{K_t}{\tau_t s + 1} \frac{1}{(t_0 s + 1)} \quad (2.17)$$

$$s_t(t) = -\dot{x}_t(t) + \lambda_1 e_t(t) + \lambda_0 \int e_t(t) dt \quad (2.18)$$

Pasando la ecuación 2.13 a forma diferencial da como resultado:

$$\tau_t t_{0t} \ddot{x}_t(t) + (\tau_t + t_0) \dot{x}_t(t) + x_t = K_t u_t(t) \quad (2.19)$$

Para la obtención de la ley de control se procede a derivar la superficie e igualarla a cero quedando:

$$\dot{s}_t(t) = \ddot{e}_t(t) + \lambda_1 \dot{e}_t(t) + \lambda_0 e(t) = 0 \quad (2.20)$$

Sabiendo que:

$$e_t(t) = R(t) - x_t(t) \quad (2.21)$$

Por tanto, al reemplazar las ecuaciones 2.20 y 2.21 en 2.19 se tiene que la ley de control para el lazo de temperatura es:

$$u_t(t) = \frac{\tau_t t_{0t}}{K_t} \left[\frac{\dot{x}_t(t)}{\tau_t t_{0t}} + \lambda_0 e_t(t) \right] + \frac{K_D S(t)}{|S(t)| + \delta} \quad (2.22)$$

2.2.3 CONTROLADOR CASCADA DSMC-PI

Como se menciona anteriormente la respuesta de flujo presenta un retardo muy bajo por lo que el modelo aproximado es el mismo que la Ecuación 2.9 por lo que para este caso simplemente se opta por un control clásico tipo PI que ayude con el seguimiento de la consigna.

La ecuación del controlador es el siguiente:

$$G_c(s) = K_p + \frac{K_i}{s} \quad (2.23)$$

La sintonización del controlador se lo hace mediante las reglas definidas por Chien et al descritas en el capítulo 1.

Para la segunda parte de esquema se propone un control tipo DSMC basado en Herrera debido a que el sistema presenta retardos variables y este controlador es capaz de soportar los retardos e irregularidades presentadas por el tanque de mezclado en comparación con los controladores clásicos tipo PID, que ante varias perturbaciones consecutivas tienden a perder estabilidad. Dicho controlador es una variación del esquema SMC, por lo que la obtención de la ley de control y las superficies se lo hace de manera parecida al esquema ya mencionado.

Para la obtención de la superficie de control se necesita saber el orden del modelo aproximado del sistema, el cual de acuerdo con Herrera hace uso del compensador de Linoya y Alpeter, quedando la Ecuación 2.24 como modelo aproximado.

$$G_t(s) = \frac{x_t^*(s)}{u_t(s)} = \frac{K_t[(\lambda-n)s+1]}{(\tau_t s+1)\left(\frac{t_0}{2}s+1\right)} \quad (2.24)$$

Al pasar el modelo aproximado a su forma diferencial se tiene:

$$\frac{\tau_t t_0}{2} \ddot{x}_t^*(t) + \left(\tau_t + \frac{t_0}{2}\right) \dot{x}_t^*(t) + x_t^* = K_t(\lambda - n)u_t + K_t u_t(t) \quad (2.25)$$

Lo siguiente ahora es obtener la superficie deslizante para lo cual se sabe que el modelo aproximado es de segundo orden siendo:

$$s_t(t) = \dot{x}_t^*(t) - \lambda_1 e_t^*(t) - \lambda_0 \int e_t^*(t) dt \quad (2.26)$$

Si se deriva la Ecuación 2.26 y se iguala a cero:

$$s_t(t) = \ddot{x}_t^*(t) - \lambda_1 \dot{e}_t^*(t) - \lambda_0 e_t^*(t) = 0 \quad (2.27)$$

Partiendo de:

$$e_t^*(t) = R(t) - x_t^*(t) \quad (2.28)$$

Al reemplazar las ecuaciones 2.27 y 2.28 en 2.25 se tiene que la ley de control es:

$$u_t(t) = \int \frac{1}{2k_t(\lambda-n)} [-\lambda_0 \tau_t t_0 e_t^*(t) - 2x_t^*(t) + 2k_t u_t(t)] dt + \int \frac{K_D S(t)}{|s(t)|+\delta} \quad (2.29)$$

2.3 DISCRETIZACIÓN DE CONTROLADORES

La siguiente sección muestra la discretización de los tres esquemas de control desarrollados y presentados en el literal anterior, con el objetivo de poder integrarlos dentro de los algoritmos a ser implementados en la tarjeta embebida

2.3.1 DISCRETIZACIÓN ESQUEMA PID-PID

Para la discretización se hace uso del equivalente discreto de integración trapezoidal mostrada en la Ecuación 2.30. este equivalente se lo aplica en la Ecuación 2.8 para poder obtener la acción de control de forma discreta ($u(k)$).

$$s = \frac{2}{T} \left[\frac{z-1}{z+1} \right] \quad (2.30)$$

$$G_c(z^{-1}) = \frac{u(z^{-1})}{e(z^{-1})} = K_p + \frac{K_i T (1+z^{-1})}{2(1-z^{-1})} + \frac{2K_d}{T} \left[\frac{1-z^{-1}}{1+z^{-1}} \right] \quad (2.31)$$

Obteniendo el parámetro $u(z^{-1})$ de la Ecuación 2.31 y transformándola en su forma discreta ($u(k)$) se obtiene la ley de control para el controlador PID que se puede utilizar para los dos lazos. Esta ecuación se muestra a continuación.

$$u(k) = a_c e(k) + b_c e(k-1) + c_c e(k-2) + u(k-1) \quad (2.32)$$

Donde:

$$a_c = \frac{(2T k_p + k_i T^2 + 2k_d)}{(2T)} \quad (2.33)$$

$$b_c = \frac{(-2T k_p + k_i T^2 - 4k_d)}{(2T)} \quad (2.34)$$

$$c_c = \frac{k_d}{T} \quad (2.35)$$

2.3.2 DISCRETIZACIÓN ESQUEMA SMC-SMC

Para este esquema también se hace uso del equivalente discreto de integración trapezoidal para ambos lazos de control. En el caso de lazo de flujo se empieza con la discretización de la superficie deslizante (Ecuación 2.10) la cual se separa en dos para discretizarla, quedando:

$$s_{f1}(s) = e_f(s) \quad (2.36)$$

$$s_{f2}(s) = \frac{\lambda_2 e(s)}{s} \quad (2.37)$$

Aplicando el equivalente de discretización en las ecuaciones 2.36 y 2.37 se tiene que:

$$s_{f1}(z^{-1}) = e_f(z^{-1}) \quad (2.38)$$

$$s_{f2}(z^{-1}) = \frac{T \lambda_2 (1+z^{-1}) e(z^{-1})}{2(1-z^{-1})} \quad (2.39)$$

Al pasarlas a su forma discreta:

$$s_{f1}(k) = e_f(k) \quad (2.40)$$

$$s_{f2}(k) = \frac{T\lambda_2}{2} (e_f(k) + e_f(k-1)) + s_{f2}(k-1) \quad (2.41)$$

Por tanto, la superficie deslizante discretizada para el lazo de flujo es:

$$s_f(k) = s_{f1}(k) + s_{f2}(k) \quad (2.42)$$

Discretizando la ley de control (Ecuación 2.14) se tiene:

$$u_f(k) = \frac{1}{K_f} [\tau_f \widehat{u}_t(k) + u_t(k) + (\tau_f \lambda_2 - 1) e_f(k)] \quad (2.43)$$

Discretizando el filtro derivativo con el equivalente discreto se tiene:

$$\widehat{u}_t(k) = \frac{1}{2Tu+T} [2(u_t(k) - u_t(k-1)) - (T - 2Tu)\widehat{u}_t(k-1)] \quad (2.44)$$

Para el lazo de flujo se tiene un total de 5 ecuaciones que deben ser implementadas dentro del algoritmo de programación del embebido.

Para el lazo de temperatura se procede a discretizar la superficie deslizante mostrada en la Ecuación 2.18 por lo que se divide la ecuación en 2 partes al igual que se lo hizo con la superficie del lazo interno, resultando en:

$$s_{t1}(s) = \lambda_1 e_t(s) + \frac{\lambda_0 e_t(s)}{s} \quad (2.45)$$

$$s_{t2}(s) = -s x_t(s) \quad (2.46)$$

Si se aplica el equivalente discreto en las Ecuaciones 2.45 y 2.46:

$$s_{t1}(z^{-1}) = \lambda_1 e_t(z^{-1}) + \frac{T\lambda_0(1+z^{-1})e_t(z^{-1})}{2(1-z^{-1})} \quad (2.47)$$

$$s_{t2}(s) = -\frac{2(1-z^{-1})}{T(1+z^{-1})} x_t(z^{-1}) \quad (2.48)$$

Al pasar las dos ecuaciones anteriores a su forma discreta queda:

$$s_{t1}(k) = \frac{2\lambda_1 + \lambda_0 T}{2} e_t(k) + e_t(k-1) + s_{t1}(k-1) \quad (2.49)$$

$$s_{t2}(k) = \frac{1}{T} [-x_t(k) + x_t(k-1)] \quad (2.50)$$

Finalmente, la ecuación total de la superficie deslizante para este lazo es:

$$s_t(k) = s_{t1}(k) + s_{t2}(k) \quad (2.51)$$

Discretizando la ley de control mostrada en la Ecuación (2.22) se tiene:

$$u_t(k) = \frac{\tau_t t_{ot}}{K_t} \left[\frac{x_t(k)}{\tau_t t_{ot}} + \lambda_0 e_t(k) \right] + \frac{K_D S(k)}{|S(k)| + \delta} \quad (2.52)$$

2.3.3 DISCRETIZACIÓN ESQUEMA DSMC-PI

En este esquema se empieza discretizando la Ecuación 2.23 la cual es el control PI para el lazo interno. Para este caso también se aplica el equivalente discreto trapezoidal lo cual queda como:

$$G_c(z^{-1}) = \frac{u_f(s)}{e_f(s)} = K_p + \frac{TK_i(1+z^{-1})}{2(1-z^{-1})} \quad (2.53)$$

Al pasarlo a su forma discreta se tiene que la salida del controlador de lazo de flujo queda:

$$u_f(k) = \frac{2k_{pf} + k_{if}T}{2} e_f(k) + \frac{-2k_{pf} + k_{if}T}{2} e_f(k-1) + u_f(k-1) \quad (2.54)$$

Con las ecuaciones del lazo de flujo discretizadas ahora se procede a discretizar las ecuaciones necesarias para el lazo de temperatura. Primero se empieza con la discretización de la superficie deslizante presentada en la Ecuación 2.26, para lo cual se la divide en dos partes quedando:

$$s_{t1}(s) = -\lambda_1 e_t^*(s) - \frac{\lambda_0}{s} e_t^*(s) \quad (2.55)$$

$$s_{t1}(s) = \dot{x}_t^*(s) \quad (2.56)$$

Al aplicar el equivalente de discretización en 2.55 y 2.56 da como resultado:

$$s_{t1}(z^{-1}) = -\lambda_1 e_t^*(z^{-1}) - \frac{\lambda_0 T(1+z^{-1})}{2(1-z^{-1})} e_t^*(z^{-1}) \quad (2.57)$$

$$s_{t2}(z^{-1}) = \dot{x}_t^*(z^{-1}) \quad (2.58)$$

Al obtener la forma discreta de 2.57 y 2.58 resulta en:

$$s_{t1}(k) = \frac{-\lambda_0 T - 2\lambda_1}{2} e_f(k) + \frac{-\lambda_0 T + 2\lambda_1}{2} e_f(k-1) + s_{t1}(k-1) \quad (2.59)$$

$$s_{t2}(k) = \frac{1}{T} [x_t^*(k) - x_t^*(k-1)] \quad (2.60)$$

La ley de control final es la suma de las ecuaciones 2.59 y 2.60.

$$s_t(k) = s_{t1}(k) + s_{t2}(k) \quad (2.61)$$

La ley de control presentada en la ecuación 2.29 se discretiza de manera similar que la superficie de control, para lo cual se la separa en 4 partes de la siguiente manera:

$$u_{t1}(s) = \frac{-\lambda_0 \tau_t t_0}{2k_t(\lambda-n)s} e_t^*(s) \quad (2.62)$$

$$u_{t2}(s) = \frac{-1}{k_t(\lambda-n)s} x_t^*(s) \quad (2.63)$$

$$u_{t3}(s) = \frac{1}{(\lambda-n)s} u_t(s) \quad (2.64)$$

$$u_{t4}(s) = d_c(s) \quad (2.65)$$

$$d_c(s) = \frac{K_D s(s)}{|s(s)|+\delta} \quad (2.66)$$

Reemplazando el equivalente discreto en las tres ecuaciones anteriores se tiene:

$$u_{t1}(z^{-1}) = \frac{-\lambda_0 \tau_t t_0}{2k_t(\lambda-n)} \frac{T(1+z^{-1})}{2(1-z^{-1})} e_t^*(z^{-1}) \quad (2.67)$$

$$u_{t2}(z^{-1}) = \frac{-1}{k_t(\lambda-n)} \frac{T(1+z^{-1})}{2(1-z^{-1})} x_t^*(z^{-1}) \quad (2.68)$$

$$u_{t3}(z^{-1}) = \frac{1}{(\lambda-n)} \frac{T(1+z^{-1})}{2(1-z^{-1})} u_t(z^{-1}) \quad (2.69)$$

$$u_{t4}(z^{-1}) = d_c(z^{-1}) \quad (2.70)$$

Obteniendo el equivalente discreto de cada parte de la ley de control se tiene:

$$u_{t1}(k) = -\frac{\lambda_0 \tau_t t_0 T}{4k_t(\lambda-n)} [e_t^*(k) + e_t^*(k-1)] + u_{t1}(k-1) \quad (2.71)$$

$$u_{t2}(k) = \frac{-T}{2k_t(\lambda-n)} [x_t^*(k) + x_t^*(k-1)] + u_{t2}(k-1) \quad (2.72)$$

$$u_{t3}(k) = \frac{T}{2(\lambda-n)} [u_t(k) + u_t(k-1)] + u_{t3}(k-1) \quad (2.73)$$

$$u_{t4}(k) = \frac{T}{2} [d_c(k) + d_c(k-1)] + u_{t4}(k-1) \quad (2.74)$$

Como ley de control final se tiene:

$$u_t(k) = u_{t1}(k) + u_{t2}(k) + u_{t3}(k) + u_{t4}(k) \quad (2.75)$$

Finalmente se discretiza el modelo de Linoya y Alperter que completa todo el esquema de control, siendo el modelo:

$$G_m(s) = \frac{x_m(s)}{u_f(s)} = \frac{\lambda k_t s}{(\tau_t s + 1) \left(\frac{t_0}{2} s + 1 \right)} \quad (2.76)$$

Usando el equivalente discreto se tiene:

$$G_m(z^{-1}) = \frac{x_m(z^{-1})}{u_f(z^{-1})} = \frac{\frac{2\lambda k_t (1-z^{-1})}{T(1+z^{-1})}}{\left(\frac{2\tau_t (1-z^{-1})}{T(1+z^{-1})} + 1 \right) \left(\frac{t_0 (1-z^{-1})}{T(1+z^{-1})} + 1 \right)} \quad (2.77)$$

Resolviendo la ecuación anterior y pasando a su forma discreta se tiene:

$$x_m = a_m [u_t(k) - u_t(k-2)] - b_m x_m(k-1) - c_m x_m(k-2) \quad (2.78)$$

Donde:

$$a_m = \frac{2k_t \lambda T}{(2\tau_t + T)(t_0 + T)} \quad (2.79)$$

$$b_m = \frac{2T^2 - 4\tau_t t_0}{(2\tau_t + T)(t_0 + T)} \quad (2.80)$$

$$c_m = \frac{(-2\tau_t + T)(-t_0 + T)}{(2\tau_t + T)(t_0 + T)} \quad (2.81)$$

Con las ecuaciones necesarias ya discretizadas, ya se puede sintonizar los controladores y reemplazar las constantes con el fin de implementarlas dentro del sistema embebido.

2.4 IMPLEMENTACIÓN DE LOS ESQUEMAS DE CONTROL

Este apartado del documento contiene los diagramas de flujo de los esquemas de control por separado y un diagrama de flujo total del programa implementado en la tarjeta embebida. Dichos diagramas de flujo explican el funcionamiento de los algoritmos y como estos están implementados.

2.4.1 DIAGRAMA DE FLUJO GENERAL DEL PROGRAMA.

El diagrama de flujo total se lo puede apreciar en la Figura 2.2. Dicho esquema es considerado como un diagrama de flujo global ya que describe la configuración de una pantalla LCD 2X16, la lectura y escritura de datos desde el puerto serial, y el menú de selección de controladores.

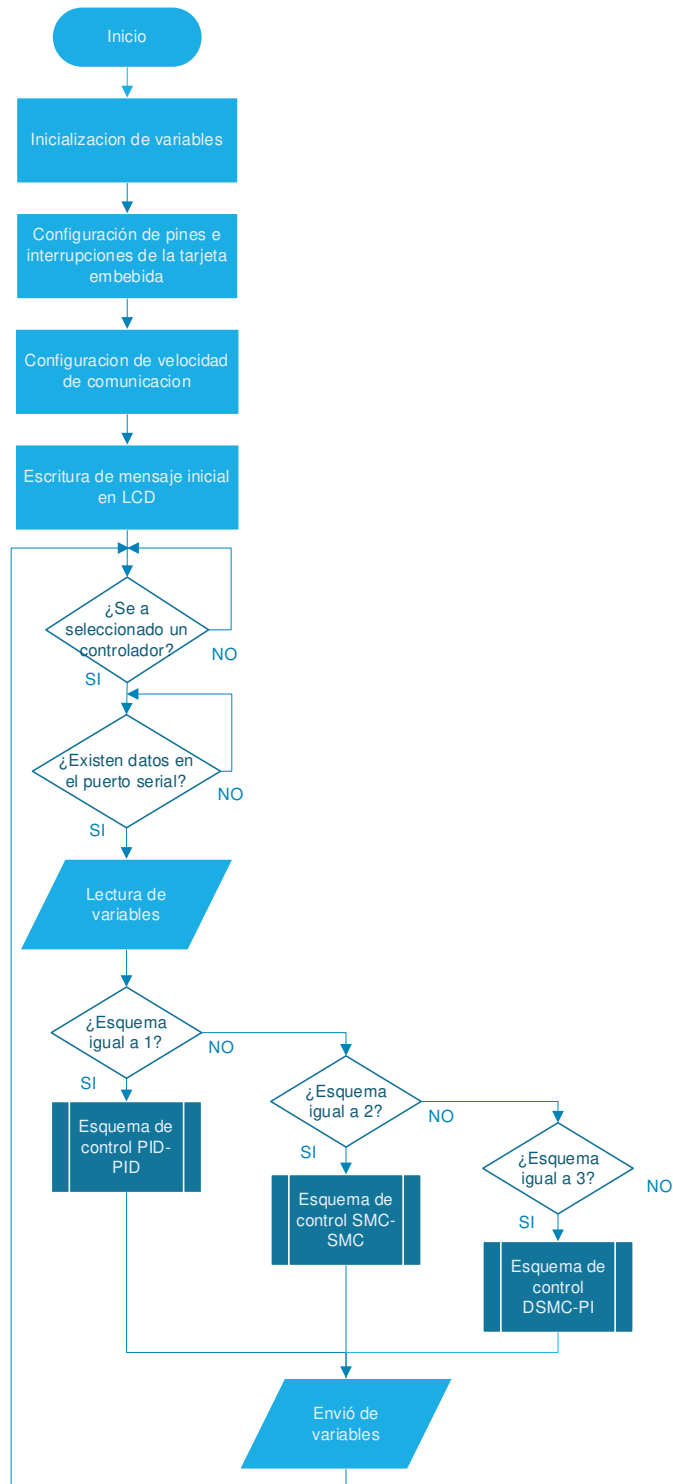


Figura 2.2. Diagrama de flujo global

2.4.2 DIAGRAMAS DE FLUJO DE LOS ESQUEMAS DE CONTROL

En la Figura 2.3 se tiene el diagrama de flujo de los controladores que muestran la manera en la que se van a ejecutar haciendo uso de las leyes de control discretizadas

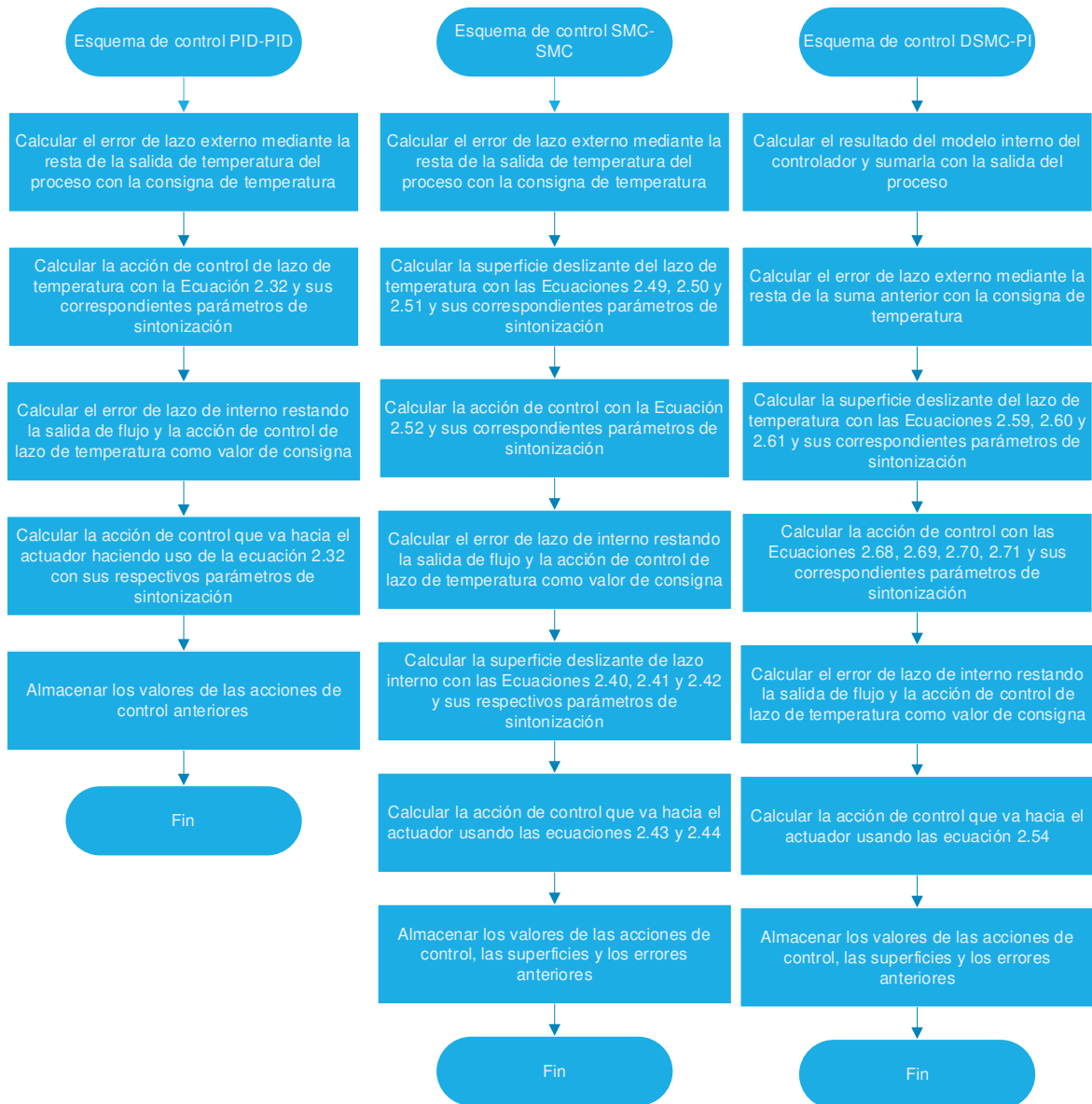


Figura 2.3. Diagrama de flujo de los esquemas PID-PID, SMC-SMC y DSMC-PI

2.5 SINTONIZACION DE CONTROLADORES

Para la sintonización de cada esquema de control se aplica el método de Austin el cual consiste en realizar un cambio escalonado del 10% en el flujo de agua fría, con el fin de obtener la respuesta del transmisor de flujo y con ello los valores necesarios para la sintonización del lazo interno. A continuación, se procede a cerrar el lazo interno y se aplica un cambio escalón de temperatura del 10% y de igual manera que con el lazo de flujo se obtiene la respuesta del transmisor de temperatura junto con todos los datos requeridos para realizar la sintonización.

2.5.1 SINTONIZACION DEL CONTROLADOR CASCADA PID-PID

2.5.1.1 PID de Lazo interno de flujo

Como se observa en la Figura 2.4 se tiene la respuesta del sistema del flujo de líquido frío del tanque de mezclado. Debido a la forma que presenta la respuesta de flujo del sistema, se opta por obtener un modelo FOPDT mediante el método de Ziegler y Nichols, el cual proporciona los parámetros τ , t_0 y K que se pueden apreciar en la Tabla 2.1.

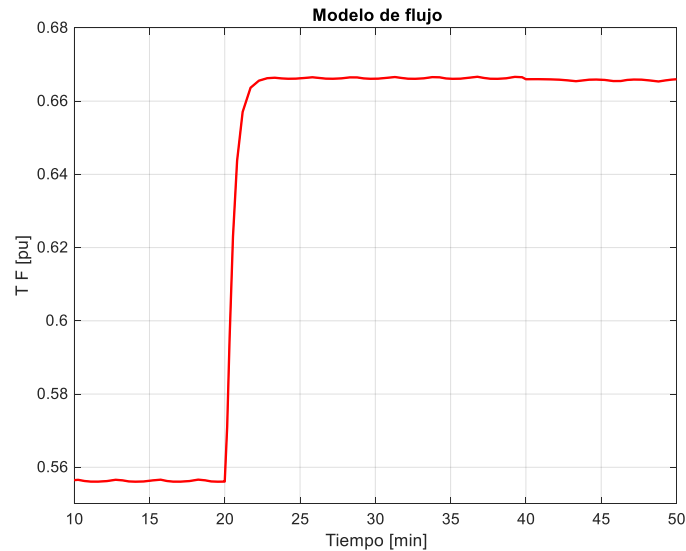


Figura 2.4 Respuesta de flujo de la planta con lazo interno abierto

Tabla 2.1. Valores de constantes de modelo FOPDT de lazo de flujo

Parámetro	Valor
K_f	2
τ_f	0.525
t_{0f}	0.055

Mediante el método de sintonización de Dahlin se calcula los valores de las constantes K_p , K_i y K_d , dando como resultados los valores que se observan en la Tabla 2.2.

Tabla 2.2. Valores de constantes del controlador PID de lazo interno

Parámetro	Valor
K_{p_f}	2.39
K_{i_f}	4.55
K_{d_f}	0.07

2.5.1.2 PID de Lazo externo de temperatura

Para la sintonización de este lazo se cierra el lazo interno de flujo y aplicando un cambio en la entrada de temperatura del sistema se tiene el resultado de la Figura 2.5.

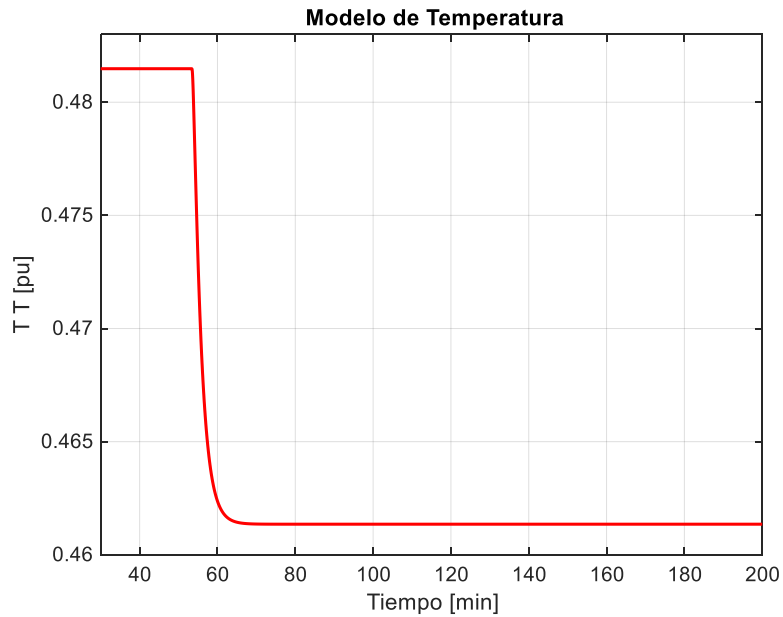


Figura 2.5. Respuesta de temperatura con lazo interno cerrado para esquema PID-PID

De igual manera que con el lazo interno se obtienen los parámetros necesarios para la sintonización los cuales se muestran en la Tabla 2.3.

Tabla 2.3. Valores de constantes de modelo FOPDT de lazo de temperatura

Parámetro	Valor
K_t	-0.78
τ_t	2.17
t_{0t}	4.03

Haciendo uso de las ecuaciones de sintonización de Dahlin se obtienen las constantes del controlador mostrados en la Tabla 2.4.

Tabla 2.4. Valores de constantes del controlador PID de lazo interno

Parámetro	Valor
K_{pt}	-0.34
K_{it}	-0.16
K_{dt}	-0.69

Los valores de las Tabla 2.1 y 2.3 junto con los valores de las ganancias de las Tablas 2.2 y 3.4 son reemplazados en las ecuaciones discretizadas del controlador PID-PID en cascada quedando como ecuación de lazo interno:

$$u_{cf} = 3.45 e_f(k) - 3.96 e_f(k - 1) + 0.88 e_f(k - 2) + u_{cf}(k - 1) \quad (2.82)$$

Para la ley de control de lazo externo se tiene:

$$u_{ct} = -9.07 e_t(k) + 17.77 e_t(k - 1) - 8.72 e_t(k - 2) + u_{ct}(k - 1) \quad (2.83)$$

Siendo así este el único esquema el único en el que necesita implementar dos ecuaciones para lograr su funcionamiento.

2.5.2 SINTONIZACION DEL CONTROLADOR CASCADA DSMC-PI

2.5.2.1 PI de Lazo interno de flujo

Para la sintonizar este lazo se parte de la Figura 2.5 que nos proporciona la respuesta del flujo de agua fría en lazo abierto. Con esta respuesta y con ayuda de las ecuaciones del método de Alfaro para la obtención del modelo FOPDT se tienen los valores de las constantes, mostradas en la Tabla 2.5, necesarias para sintonizar el controlador de manera adecuada.

Para la sintonización de este lazo se hace uso de las ecuaciones propuestas por Chien et al, dando como resultado las constantes necesarias para el controlador mostradas en la Tabla 2.6.

Tabla 2.5. Constantes de modelo FOPDT para la respuesta de flujo

Parámetro	Valor
K_f	2
τ_f	0.45
t_{0f}	0.12

Tabla 2.6. Valores de constantes del controlador PI

Parámetro	Valor
K_{pf}	1.11
K_{if}	2.50

2.5.2.2 DSMC de Lazo externo de temperatura

Al igual que en el esquema de control anterior, se cierra el lazo interno de flujo y se obtiene la respuesta de temperatura con lazo abierto mostrado en la Figura 2.6.

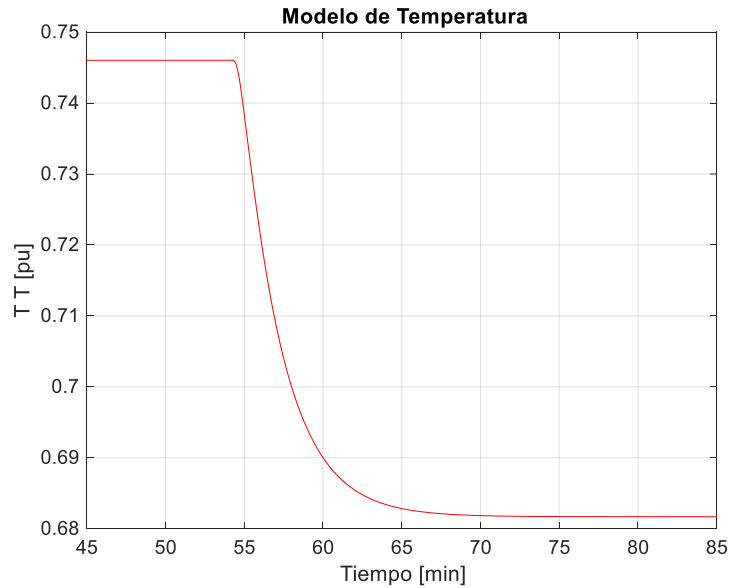


Figura 2.6. Respuesta de temperatura con lazo interno cerrado para esquema DSMC-PI

Con ayuda del método de Alfaro se consigue las constantes que necesita el modelo FOPDT del modelo de temperatura mostradas en la Tabla 2.7.

Tabla 2.7. Valores de constantes para modelo de temperatura

Parámetro	Valor
K_t	-0.58
τ_t	2.06
t_{0t}	3.72

Tabla 2.8. Valores de constantes del controlador PID de lazo interno

Parámetro	Valor
λ_0	0.26
λ_1	1.02
λ	25
K_d	5.60
δ	6.40

Los parámetros de sintonización del controlador del lazo externo se obtienen a partir de los valores mostrados en la tabla anterior. En la tabla 2.8 se indican los resultados de todos los parámetros del controlador DSMC.

Los parámetros de las Tablas 2.5, 2.6, 2,7 y 2,8 son reemplazados en las leyes de control y superficies discretizadas resultando en lo siguiente:

Lazo interno de flujo:

$$u_{cf} = 1.21 e_f(k) - 2.03 e_f(k - 1) + u_{cf}(k - 1) \quad (2.84)$$

Superficie para lazo de temperatura:

$$s_{f1}(k) = -1.03 e_f(k) + 1.01 e_f(k - 1) + S_{f1}(k - 1) \quad (2.85)$$

$$s_{f2}(k) = 12.5 (X_t(k) - X_t(k - 1)) \quad (2.86)$$

$$s_f = S_{f1}(k) + S_{f2}(k) \quad (2.87)$$

Ley de control para lazo de temperatura:

$$u_{c1}(k) = 0.003 (e_t(k) + e_t(k - 1)) + u_{c1}(k - 1) \quad (2.88)$$

$$u_{c2}(k) = 0.003 (X_a(k) + X_a(k - 1)) + u_{c2}(k - 1) \quad (2.89)$$

$$u_{c3}(k) = -0.002 (u_t(k) + u_t(k - 1)) + u_{c3}(k - 1) \quad (2.90)$$

$$u_{c4}(k) = 0.04 (d_c(k) + d_c(k - 1)) + u_{c4}(k - 1) \quad (2.91)$$

$$d_c(k) = \frac{5.6 S(k)}{|S(k)|+6.4} \quad (2.92)$$

$$u_t(k) = u_{c1}(k) + u_{c2}(k) + u_{c3}(k) + u_{c4}(k) \quad (2.93)$$

Modelo de la planta:

$$X_m = -0.14 (u_t(k) - u_t(k - 2)) - 1.92 X_m(k - 1) + 0.92 X_m(k - 2) \quad (2.94)$$

2.5.3 SINTONIZACION DEL CONTROLADOR CASCADA SMC-SMC

2.5.3.1 SMC de Lazo interno de flujo

La sintonización de este controlador se lo hace de la misma manera que el lazo interno del esquema de control anterior, de tal manera que se toma los valores de la Tabla 2.5 para el modelo FOPDT y para el ajuste del control de lazo interno.

Los valores obtenidos de las ecuaciones para sintonizar este controlador se encuentran en la Tabla 2.9.

Tabla 2.9. Valores de constantes del controlador SMC de lazo interno de flujo

Parámetro	Valor
λ_2	2.25
T_u	0.05
Kd	6.9
δ	11

2.5.3.2 SMC de Lazo externo de temperatura

Para encontrar los parámetros de este lazo de control se procede a cerrar el lazo interno de igual manera que en los procesos de sintonización realizados con anterioridad, lo que da como resultado la respuesta de temperatura de la Figura 2.7.

Con ayuda del método de Alfaro para obtención de modelos FOPDT se tiene las constantes necesarias para realizar el ajuste adecuado del control de lazo externo, siendo los valores de la Tabla 2.10 el resultado de cada una de las constantes.

Tabla 2.10. Valores de constantes para modelo de temperatura

Parámetro	Valor
K_t	-0.386
τ_t	2
t_{0t}	3.56

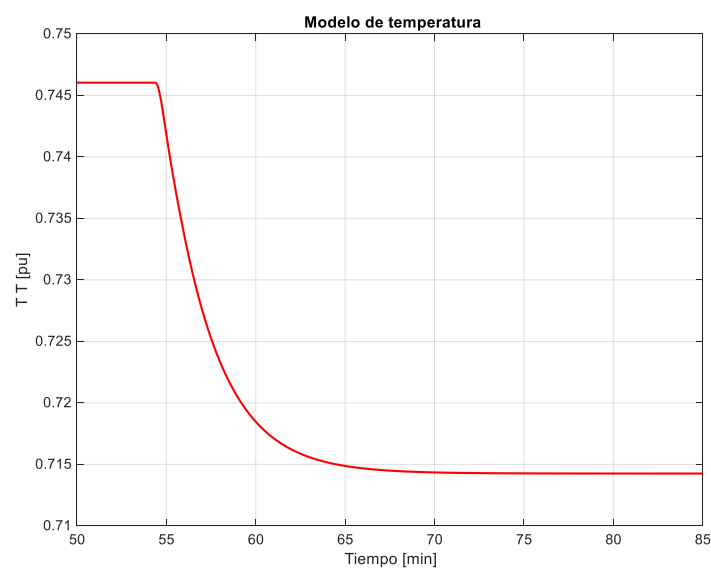


Figura 2.7. Respuesta de temperatura con lazo interno cerrado para esquema SMC-SMC

Los valores del control sintonizado se encuentran en la Tabla 2.11, obtenidos con los datos de la tabla anterior.

Tabla 2.11. Valores de constantes del controlador SMC de lazo externo de flujo

Parámetro	Valor
λ_0	0.15
λ_1	0.78
K_d	8.5
δ	6.5

Este esquema presenta un total de 7 ecuaciones discretizadas para la implementación en el sistema embebido, por lo que reemplazando los valores de las Tablas 2.5, 2.9, 2.10 y 2.11 en dichas ecuaciones se tiene:

La superficie del lazo interno como:

$$s_{f1}(k) = 0.09 (e_f(k) + e_f(k - 1)) + S_{f1}(k - 1) \quad (2.95)$$

$$s_{f2}(k) = e_f(k) \quad (2.96)$$

$$s_f = S_{f1}(k) + S_{f2}(k) \quad (2.97)$$

Ley de control de lazo de flujo:

$$u_{cf} = 0.22 u_s(k) + 0.5 u_{ct}(k) + 0.0006 e_f(k) + \frac{6.9 S_f}{|S_f|+11} \quad (2.98)$$

$$u_s(k) = 11.11(u_t(k) - u_t(k)) + 0.11 u_s(k - 1) \quad (2.99)$$

Superficie de lazo externo:

$$s_{t1}(k) = 0.79 e_f(k) - 0.77 e_f(k - 1) + S_{t1}(k - 1) \quad (2.100)$$

$$s_{t2}(k) = 12.5 (x_t(k) - x_t(k - 1)) \quad (2.101)$$

$$s_t = -S_{t1}(k) + S_2(k) \quad (2.102)$$

Ley de control de lazo de temperatura

$$u_{cf} = -2.59 x_t(k) - 2.77 e_t(k) + \frac{8.5 S_t}{|S_t|+6.8} \quad (2.103)$$

2.6 COMUNICACIÓN ARDUINO SIMULINK

En esta sección se indica como se encuentra configurado la comunicación para la conexión entre el programa Simulink y la tarjeta embebida Arduino. Esta es la parte más importante dentro del proyecto ya que si no se logra establecer una comunicación entre el software y hardware el proyecto no puede llevarse a cabo. Por esta razón se toma la decisión de explicar cómo se configura la comunicación en Simulink y en Arduino.

2.6.1 CONFIGURACION PARA ENVIO Y RECEPCIÓN DE DATOS EN ARDUINO

Arduino Cuenta con una gran variedad de funciones que permiten enviar y recibir datos a través del puerto serial.

2.6.1.1 Recepción de datos en Arduino

En este proyecto se optó por utilizar la función “Serial.parseFloat” que permite leer datos tipo flotante. La forma en la que se utiliza dicha función se la puede observar en la Figura 2.8.

```
dato1=Serial.parseFloat();  
dato2=Serial.parseFloat();
```

Figura 2.8. Uso de la función “Serial.parseFloat”.

El escribir la función tal como se muestra en la Figura 2.5, permite recibir dos datos y almacenarlos dentro de una variable. En caso de que se necesiten recibir 3 o más datos simplemente basta con crear una nueva línea de código con una nueva variable.

2.6.1.2 Envío de datos desde Arduino

Para enviar datos se utiliza la función “Serial.print” y “Serial.println”. la forma en la que se los utiliza se observa en la figura 2.9.

```
Serial.print(dato1);  
Serial.print(',');  
Serial.println(dato2);
```

Figura 2.9. Uso de las funciones “Serial.print” y “Serial.println”.

La estructura del envío consta de 3 partes

- a. “Serial.print(dato1)”: envía el dato 1.

- b. "Serial.print(',')": separa el dato 1 del siguiente dato 2.
- c. "Serial.println(dato2)": envía el último dato y pasa a una siguiente línea.

2.6.2 CONFIGURACION PARA ENVIO Y RECEPCIÓN DE DATOS EN SIMULINK

Para realizar el envío y recepción de información en Simulink, es necesario tener instalado la librería "Instrument Control Toolbox", la cual cuenta con una gran variedad de bloques distintos para poder establecer una comunicación.

En este caso para la comunicación se hará uso especialmente de los bloques "Query instrument" y "To Instrument" para la recepción y envío de datos respectivamente.

2.6.2.1 Recepción de datos en Simulink

La estructura con la que se va a recibir datos en Simulink se lo puede observar en la figura 2.10. Dicha estructura permite recibir varios datos a través del puerto serial.

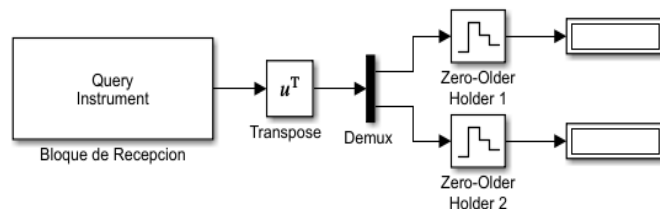


Figura 2.10. Estructura para recibir datos en Simulink

La estructura cuenta de 4 partes:

- a. El bloque "Query instrument": encargado de leer datos desde el puerto serial
- b. El bloque Transpose: transpone los datos recibidos desde el embebido
- c. Bloque Demux: permite separar los datos.
- d. Bloque Zero-Order Holder o retenedor de orden cero: permite mantener el ultimo valor recibido durante cierto tiempo.

Para poder recibir los datos de manera adecuada es necesario configurar algunos parámetros dentro del bloque "Query instrument". Dichos parámetros configurables se los observa en las Figuras 2.11 y 2.12.

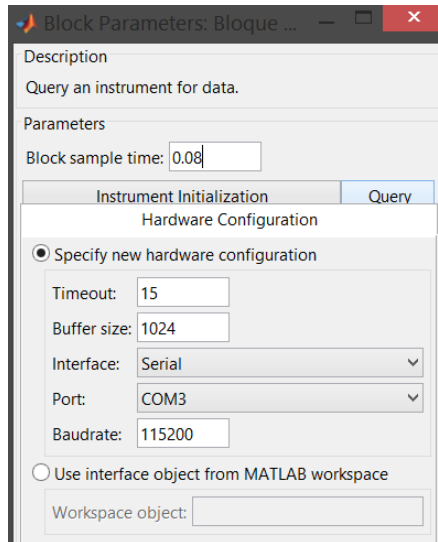


Figura 2.11. Pestaña de configuración de hardware del bloque “Query instrument”.

En la pestaña de configuración de hardware se especifica el tiempo de muestreo (sample time), además de otros parámetros los cuales son:

- a. Time out: tiempo de espera para completar la recepción de datos
- b. Buffer size: el total de bytes almacenables en el buffer.
- c. Interfaz: el tipo de interfaz para la comunicación.
- d. Port: el puesto donde se realiza la comunicación.
- e. Baudrate: la velocidad de transmisión de información.

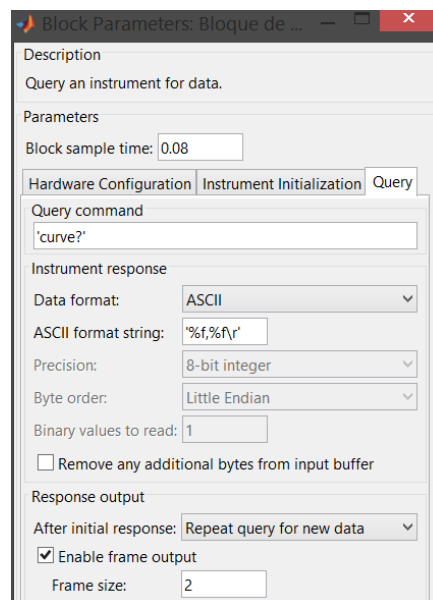


Figura 2.12. Pestaña de configuración de lectura del bloque “Query instrument”.

En la pestaña “Query” se tiene la configuración del tipo de formato en el que se van a recibir los datos, la estructura en que se va a recibir la información, y la cantidad de datos que va a aceptar.

2.6.2.2 Envío de datos desde Simulink

La forma en la que se va a recibir datos dentro del programa se lo aprecia en la Figura 2.13, la cual se compone de 3 partes:

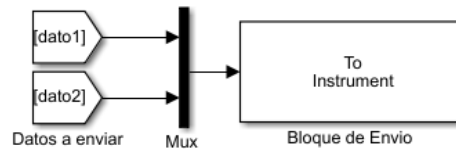


Figura 2.13. Estructura para enviar datos desde Simulink

- a. Datos por enviar
- b. Mux: bloque que es capaz de combinar dos o más señales en una salida única de forma vectorial.
- c. Bloque de “To Instrument”: encargado de enviar la información a través del puerto serial.

De la misma forma que el bloque “Query instrument”, este también debe configurarse de manera correcta para no tener inconvenientes al momento de enviar datos mediante el puerto serial. Las configuraciones de este bloque se las puede observar en las Figuras 2.14 y 2.15.

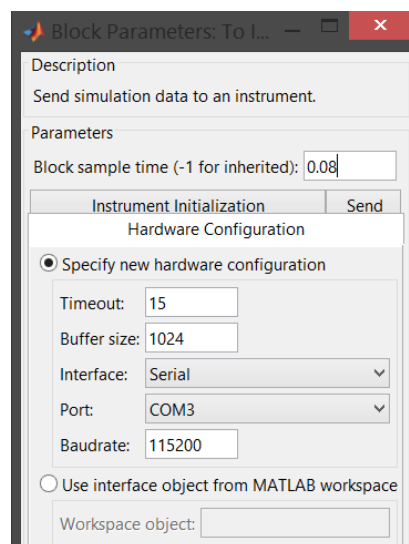


Figura 2.14. Pestaña de configuración de hardware del bloque “To Instrument”.

El bloque “to Instrument” al igual que el bloque “Query instrument” cuentan con la misma pestaña de configuración de hardware, por lo que, si se desea que la comunicación entre el software y el hardware funcione de manera adecuada, los parámetros de ambas pestañas deben ser configuradas con los mismos valores.

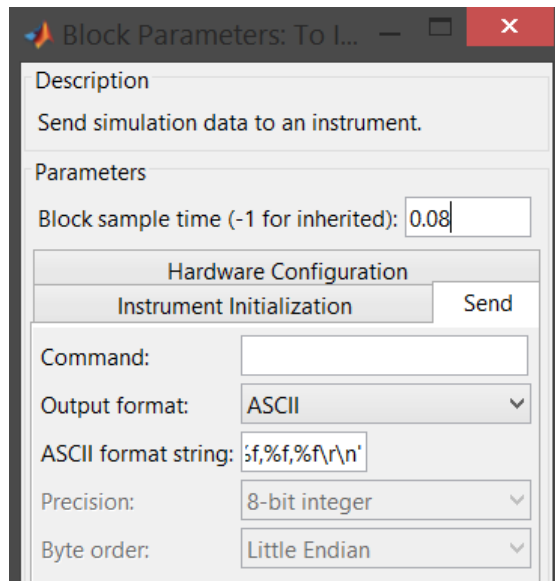


Figura 2.15. Pestaña de configuración de envío del bloque “To Instrument”.

En el caso de la pestaña de envío vista en la Figura 2.15 basta elegir el tipo de estructura que esta posea y el formato de salida que se enviara al embebido, siendo para este caso un formato ASCII.

2.7 DESARROLLO DE HARDWARE IN THE LOOP

Para la implementación bajo el concepto de Hardware in the loop se plantea el uso de la tarjeta Arduino Mega 2560 la cual está acompañada de una interfaz simple que permite la selección de uno de los 3 controladores planteados. El esquema de la interfaz se muestra en la Figura 2.16:



Figura 2.16. Esquema de conexión de interfaz para manejo de la tarjeta embebida.

Donde los dispositivos de salida son un Display 16X2 que permite mostrar mensajes de hasta 16 caracteres por línea y un par de leds que indicaran si la tarjeta esta lista para

enviar y recibir información. Los dispositivos de entrada son una hilera de botones que permiten la navegación en el menú.

La interfaz se encuentra implementada en una placa electrónica PCB (Anexo III) que fue desarrollada con ayuda del software Proteus.

Para completar el concepto de Hardware in the loop se conecta el Arduino mediante un cable USB a un ordenador que contiene implementado las ecuaciones del modelo matemático del tanque de mezclado mostradas en la sección 2.1 en el software Simulink, cerrando de esta manera el lazo de control. En la Figura 2.17 se puede apreciar el esquema básico de la conexión HIL.

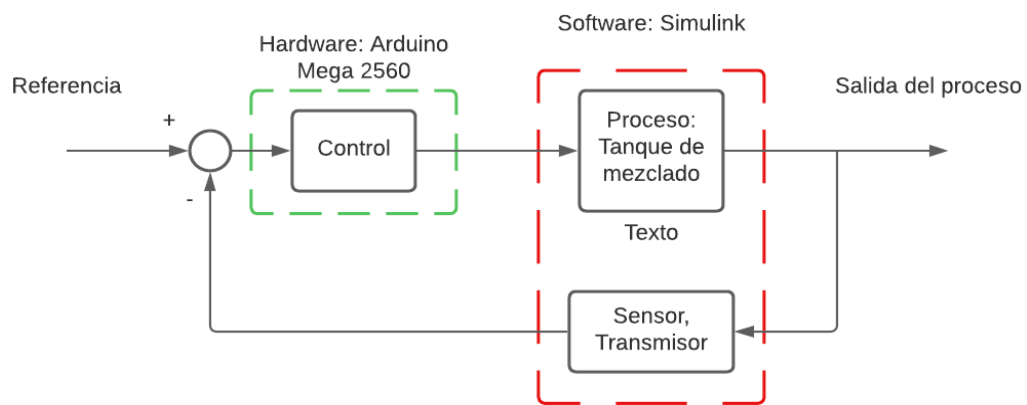


Figura 2.17. Esquema básico de la conexión HIL

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En esta sección del documento se presenta la implementación en físico de un módulo que permite el manejo y selección de tres controladores implementados en la tarjeta embebida. También se muestran las respuestas de temperatura del tanque de mezclado y las acciones de control que proporciona cada uno de los controladores implementados, Además de indicar los valores y resultados de los índices de desempeño ISE e ISCO. Estas respuestas se obtienen a partir de a partir de la implementación de todos los esquemas de control dentro de una tarjeta embebida llamada Arduino Mega 2560.

Para las pruebas de funcionamiento de cada esquema implementado se aplican perturbaciones de flujo de líquido caliente dentro del proceso de mezclado con el fin de poder observar cual es el control que presenta un mejor desempeño.

3.1 RESULTADOS

3.1.1 RESULTADOS DE IMPLEMENTACION HARWARE IN THE LOOP.

La implementación bajo el concepto de Hardware in the loop se lo aprecia en la Figura 3.1, donde se observa la conexión mediante comunicación serial entre el proceso químico implementado en Simulink dentro del ordenador y los controladores en cascada implementados en la tarjeta embebida.

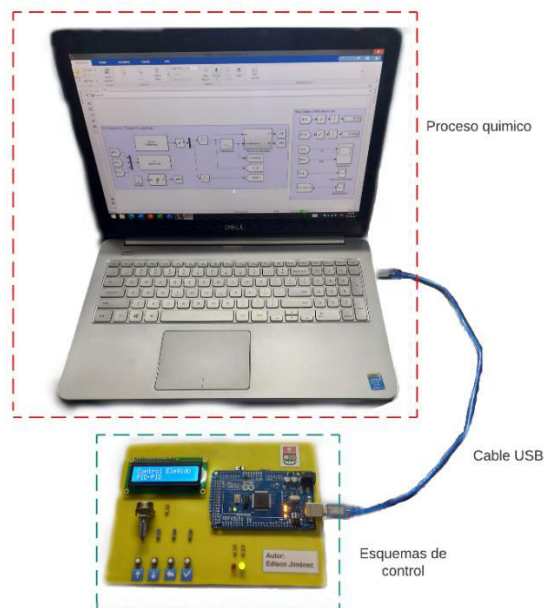


Figura 3.1. Implementación Hardware in the loop

Los esquemas de control pueden ser seleccionados mediante una interfaz montada en una baquelita la cual se observa en mejor detalle en la Figura 3.2.



Figura 3.2. Interfaz para selección de controladores

La interfaz de la Figura 3.2 está compuesta por distintos componentes, entre los cuales se tiene:

- a. Un LCD 16x2 que indica el menú de con los controladores disponibles como se lo observa en la Figura 3.3 la cual muestra la opción 3 que corresponde al controlador tipo DSMC-PI.



Figura 3.3. LCD 16x2

Dicho LCD está acompañado de un potenciómetro que permite el ajuste del contraste de la pantalla.

- b. Una Hilera de cuatro botones que permite seleccionar cualquiera de los tres esquemas de control. Dichos botones se los aprecia de mejor manera en la Figura 3.4.



Figura 3.4. Botones de navegación.

- c. Dos leds indicadores, uno de color verde y otro rojo, los cuales indican si un controlador esta seleccionado o no respectivamente. Los leds se muestran en la Figura 3.5.



Figura 3.5. Leds indicadores

- d. Una tarjeta embebida Arduino que contiene programado los tres esquemas de control, los menús de navegación y la programación del LCD. La tarjeta se observa en la figura 3.6.

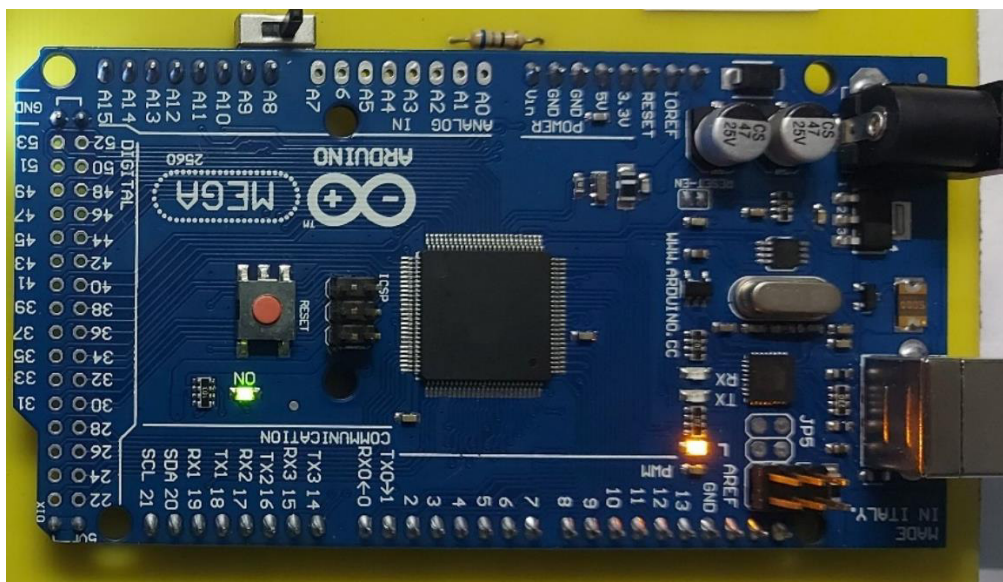


Figura 3.6. Tarjeta Arduino

- e. Un interruptor que permite seleccionar el modo de funcionamiento de la tarjeta embebida, siendo estos el modo de programación o el modo de trabajo. El interruptor se indica en la Figura 3.7.



Figura 3.7. Interruptor para selección de modos de funcionamiento

Para la verificación del funcionamiento del módulo, este se lo conecta al ordenador que contiene el tanque de mezclado con las perturbaciones mediante un cable USB y se escoge el esquema de control PID-PID con ayuda de los botones de navegación.

En el módulo de la Figura 3.1 se puede observar el LCD que indica el controlador elegido, PID-PID para este caso, y mediante la luz verde que el esquema está listo para utilizarse. La tarjeta Arduino indica mediante dos leds de color naranja (RX y TX) que existe comunicación con el software Simulink que contiene el proceso químico.

A continuación, se indica las pruebas realizadas de los controladores haciendo el uso de la tarjeta embebida y el tanque de mezclado de líquidos en Simulink. Si bien la planta trabaja entre 100°F a 200 °F todos los resultados de las pruebas se encuentran en por unidad (0 a 1), esto debido a que la válvula que regula el flujo de agua fría en el proceso trabaja con este tipo de valores.

3.1.2 PRUEBAS ANTE PERTURBACIONES.

Para esta prueba se realiza perturbaciones dentro del flujo de entrada de agua caliente de tal manera que se hace una serie de cambios tipo escalonados empezando desde la condición inicial que es 250 [lb/min] hasta llegar a las 100 [lb/min]. Estos cambios se los puede apreciar de mejor manera en la Figura 3.8 donde se observa claramente como descende este valor de flujo caliente de forma escalonada hasta llegar al valor mencionado anteriormente.

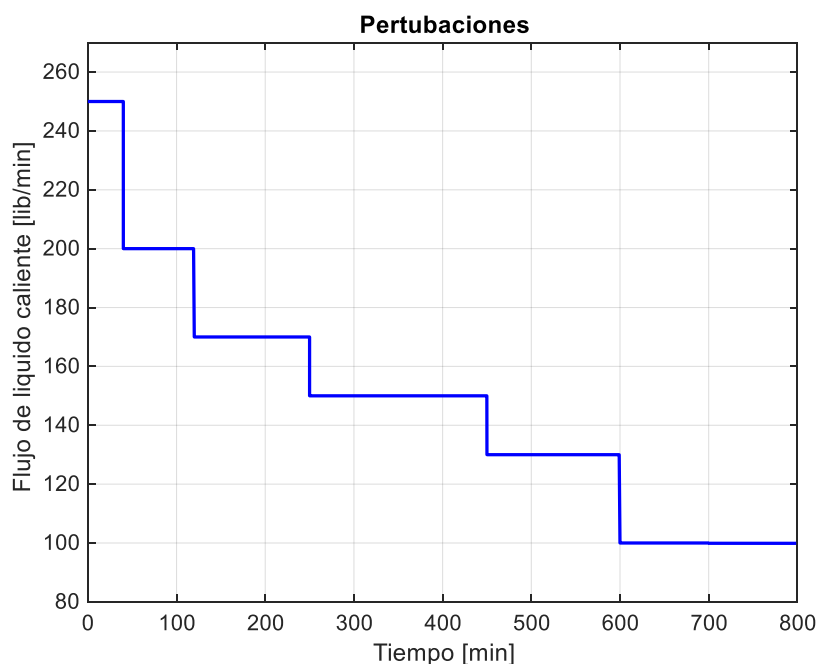


Figura 3.8. Perturbaciones de flujo caliente para el tanque de mezclado

3.1.2.1 Respuesta de temperatura y acción de control del esquema PID-PID

Con las perturbaciones ya definidas la respuesta del controlador PID-PID se lo aprecia en la Figura 3.9 donde se nota este logra alcanzar el valor de consigna de temperatura pese a que el sistema presente una variación en el retardo debido a los cambios de flujo de líquido caliente.

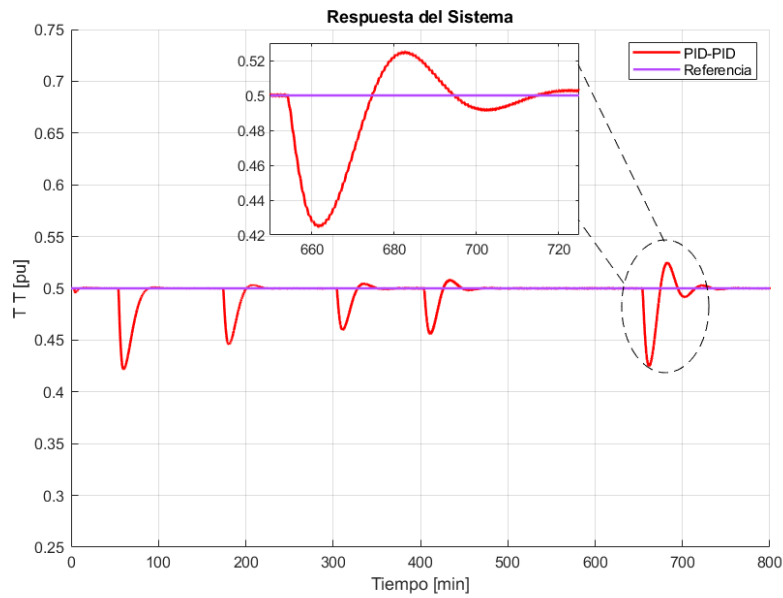


Figura 3.9. Respuesta de temperatura control PID-PID

Observando en detenimiento la respuesta de la Figura 3.9 se puede notar que esta presenta sobre picos de temperatura al momento que el proceso alcanza el valor de consigna, aunque estos se encuentran por debajo del 10%, siendo el más grande de un valor de 4.8% presentado en la última perturbación. El tiempo de establecimiento que presenta la respuesta ante la última perturbación es de 702.2 minutos.

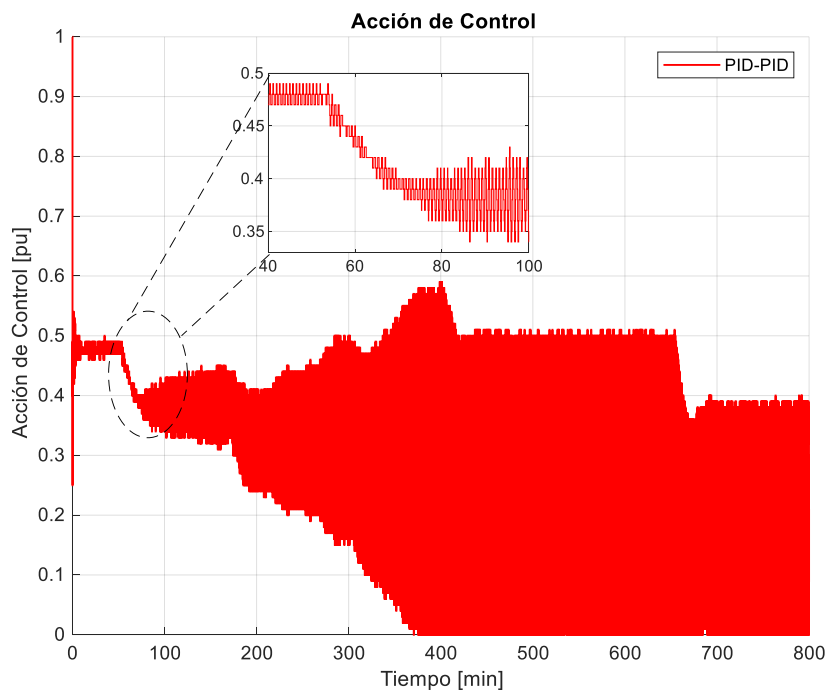


Figura 3.10. Acción de control del control PID-PID

Al observar la Figura 3.10 de la acción de control, se puede notar que a partir de aproximadamente el minuto 76 el esquema PID-PID empieza a presentar una respuesta oscilatoria que se vuelve cada vez más grande a medida que aparecen las perturbaciones de flujo de agua caliente. Estas oscilaciones pueden terminar dañando el accionamiento, en este caso la válvula.

3.1.2.2 Respuesta de temperatura y acción de control del esquema SMC-SMC

En la figura 3.11 se observa la respuesta de temperatura que se obtiene al utilizar el control SMC-SMC, donde se aprecia que el proceso es capaz de llegar al valor de consigna (0.5 [pu]) después de aplicarse cada una de las perturbaciones. El proceso no llega a presentar ningún sobre pico al alcanzar el valor de referencia y el tiempo de establecimiento es de 670,1 en la última perturbación.

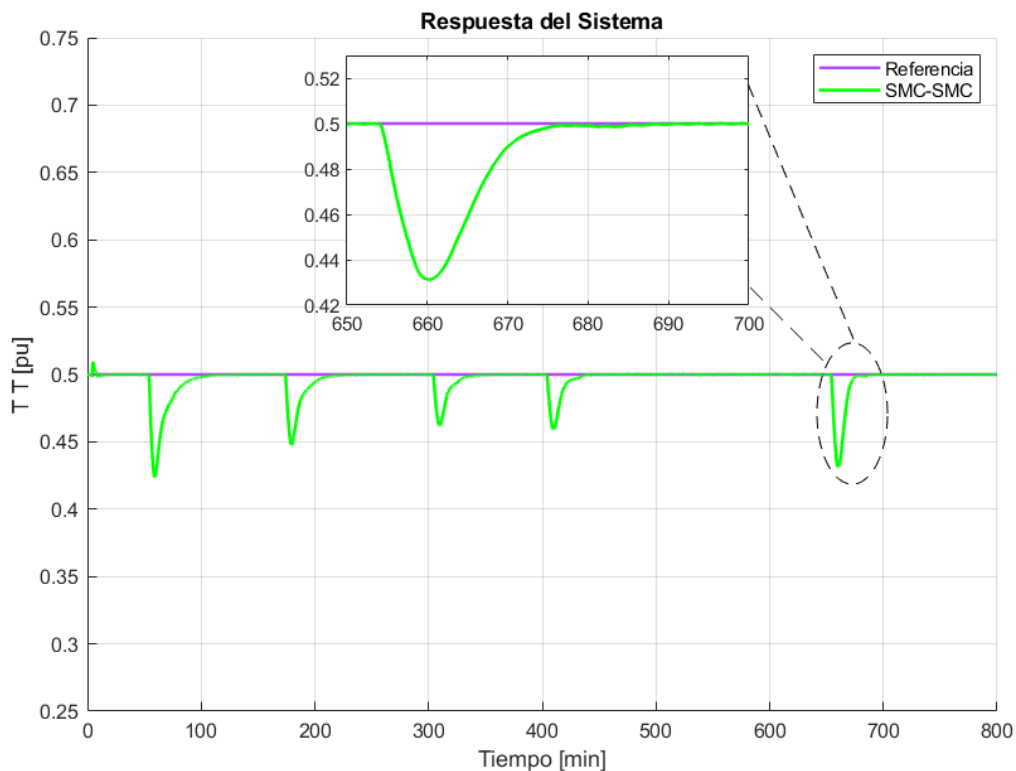


Figura 3.11. Respuesta de temperatura control SMC-SMC

La acción del control se observa en la Figura 3.12. la cual presenta un pequeño efecto de chattering debido a la naturaleza del controlador, lo cual no afecta mucho a los resultados de la planta. Pese a que la acción de control obtenida presenta un movimiento suave ante cambios de perturbaciones, en los primeros segundos esta presenta una acción de control fuerte la cual abre y cierra la válvula de manera brusca, pudiendo llegar a dañar el accionamiento.

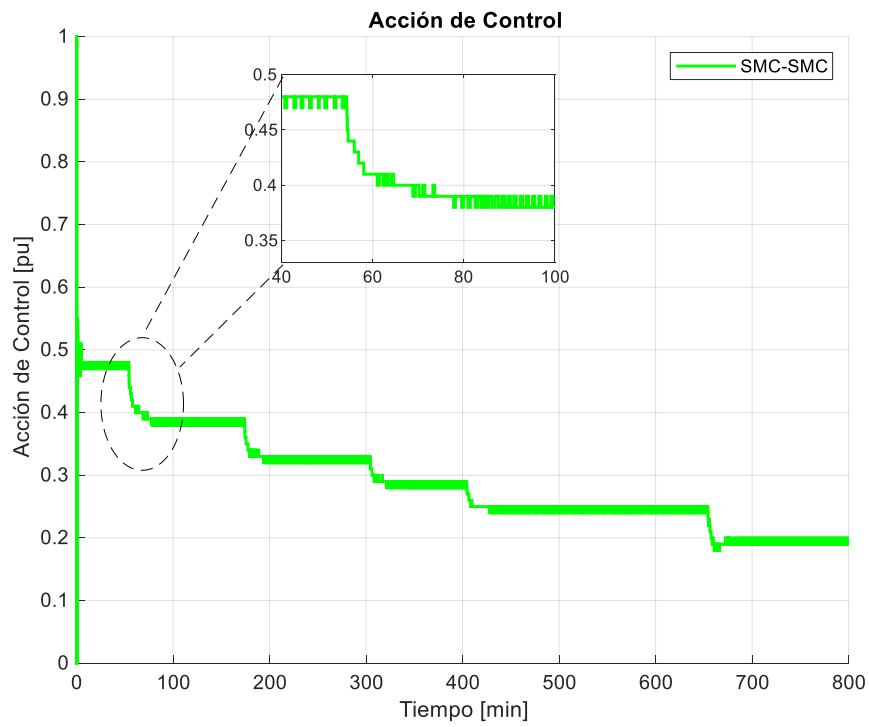


Figura 3.12. Acción de control del control SMC-SMC

3.1.2.3 Respuesta de temperatura y acción de control del esquema DSMC-PI

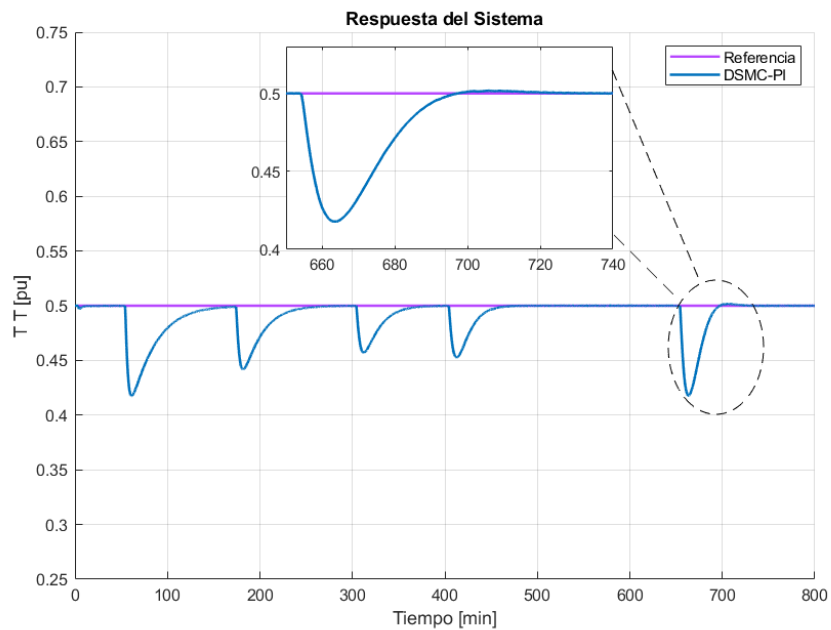


Figura 3.13. Respuesta de temperatura control DSMC-PI

En la Figura 3.13 se logra apreciar la respuesta de temperatura al aplicar el esquema DSMC-PI, dicha respuesta al igual que la proporcionada por los dos controladores vistos

anteriormente, logra alcanzar el valor de consigna después de cada perturbación, además, esta presenta un pequeño sobre pico con un valor de 0.33% al recuperarse de la última perturbación. El tiempo de establecimiento del proceso es 705.8 minutos después de aplicarse la perturbación de flujo caliente de 100 [lb/min].

La acción de control se presenta en la Figura 3.14, dicha acción presenta un cambio suave ante todas las perturbaciones y un efecto de chattering pequeño que no afecta al proceso. Al momento en que el controlador entra en funcionamiento la acción de control es agresiva, pero es menos agresiva que la señal del controlador SMC-SMC y que el controlador PID-PID. Además, se puede ver como se comportará la válvula, llegando a cerrarse hasta el 20% a medida que el flujo de líquido caliente va reduciendo.

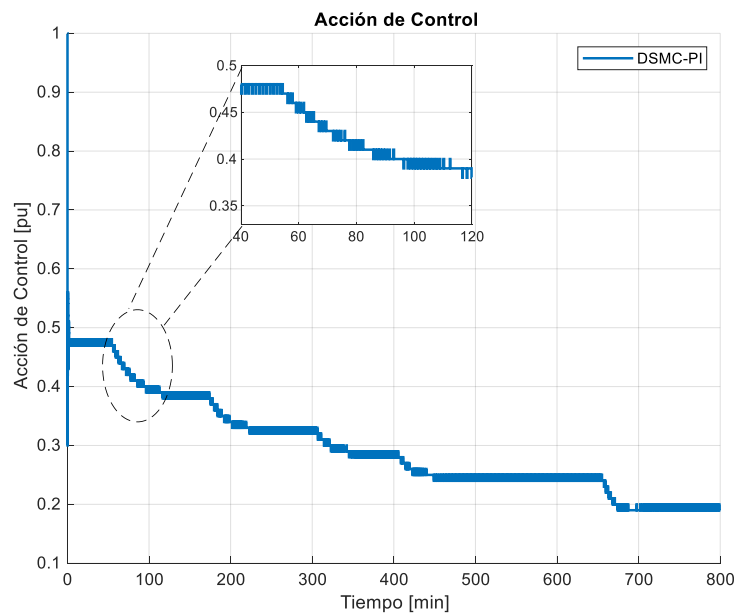


Figura 3.14. Acción de control del control DSMC-PI

Los parámetros de máximo sobre pico y tiempo de establecimiento se los muestra en la Tabla 3.1, estos valores se los presenta mediante unas gráficas de barras mostradas en las Figuras 3.15 y 3.16 para el máximo sobre pico y el tiempo de establecimiento respectivamente.

Tabla 3.1. Máximo sobre pico (Mp) y tiempo de establecimiento (ts)

Esquema	Mp [%]	Ts [min]
PID-PID	4.8	702.2
SMC-SMC	0	670.1
DSMC-PI	0.33	705.8

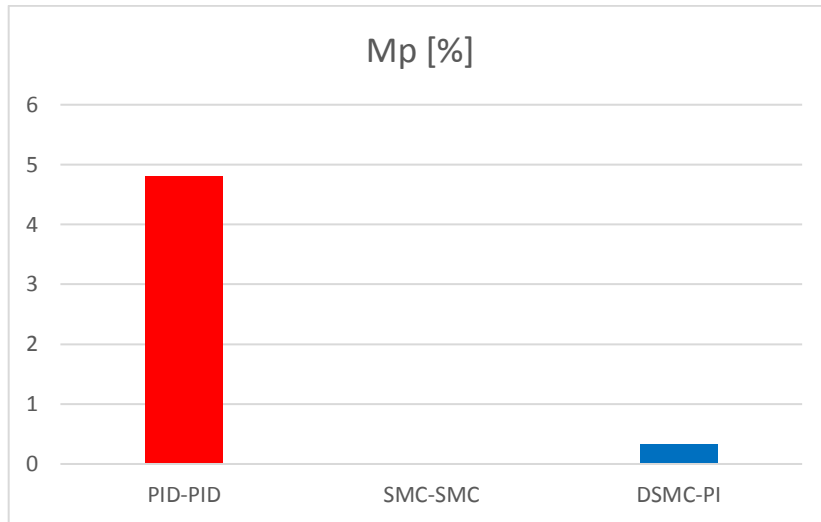


Figura 3.15. Máximo sobre pico

Al comparar un poco los resultados obtenidos de las Figuras 3.15 y 3.16 se aprecia que el controlador DSMC-PI es el más lento al tratar de recuperarse ante varias perturbaciones y el más rápido es el control SMC-SMC, además, ninguno de los esquemas sobrepasa el 10% de sobre pico.

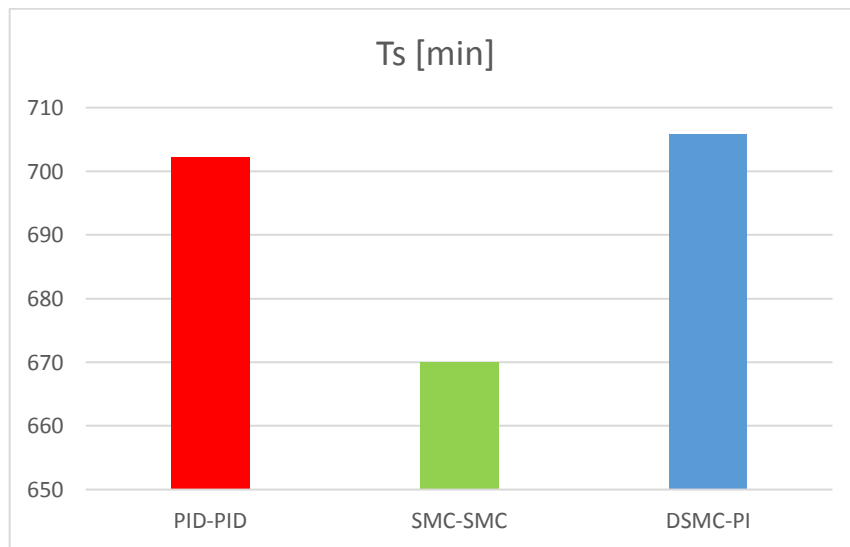


Figura 3.16. Tiempo de establecimiento.

Si bien las respuestas que proporcionaban las plantas con los distintos controladores son bastante aceptables, al momento de analizar las señales de control, la acción del esquema DSMC-PI sobresale en comparación a los otros controladores, puesto que no presenta oscilaciones, como lo es el caso del controlador PID-PID y la acción de control es mejor en comparación a la señal proporcionada por el esquema SMC-SMC, puesto que la acción de control es menos agresiva al inicio de la simulación.

Tomando en cuenta lo que se menciona hasta el momento se puede decir que el mejor esquema de los 3 controladores es el DSMC-PI, pero esto no es suficiente aun para poder definir cual controlador sobresale, por lo que se propone realizar un análisis de los índices de desempeño ISE e ISCO de cada control.

3.1.2.4 Índices de desempeño.

Si bien los valores de los índices de desempeño se muestran en la Tabla 3.2, estos se los presenta mediante una gráfica de barras mostrada en la Figura 3.17 y 3.18 para los índices ISE e ISCO respectivamente. Dichas graficas permiten ver en mejor detalle el desempeño de cada uno de los controladores.

Tabla 3.2. Índices de desempeño

Esquema	ISE	ISCO
PID-PID	0.19	89.17
SMC-SMC	0.12	73.57
DSMC-PI	0.007	76.13

En la Figura 3.17 se puede apreciar el valor de los índices ISE ordenados de mayor a menor, siendo el controlador DSMC-PI el que presenta el índice más pequeño (0.007) y el que mejor desempeño presenta si se toma en cuenta solo este índice. En el caso del controlador PID-PID, el índice es demasiado grande en comparación al esquema DSMC-PI, lo que da a entender que el sistema tiene una estabilidad pobre. Al analizar el valor obtenido por el controlador SMC-SMC también se puede decir que este tiene un desempeño menor que el esquema DSMC-PI.

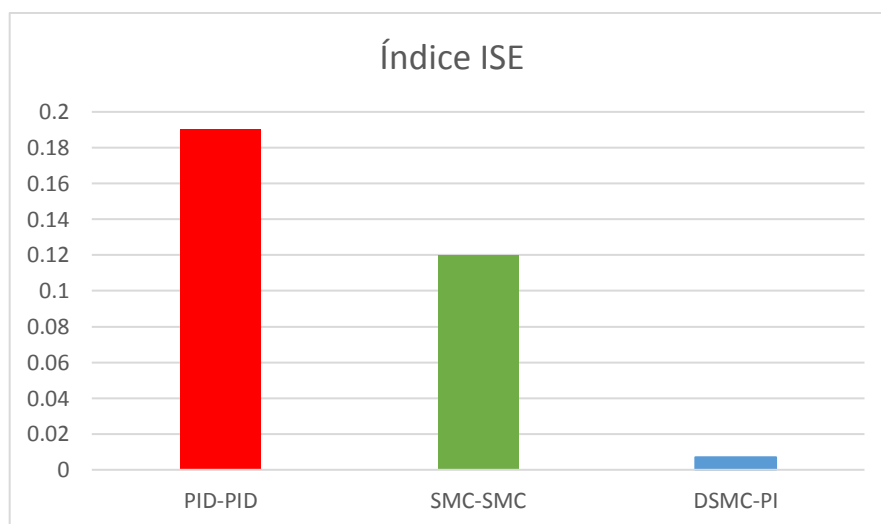


Figura 3.17. Índice de desempeño ISE

En la Figura 3.8 se aprecia que el índice ISCO de los controladores SMC-SMC y DSMC-PI son bastante similares, siendo un poco más pequeño el índice del esquema SMC-SMC. Esto da a entender que las acciones de control de los dos esquemas mencionados.

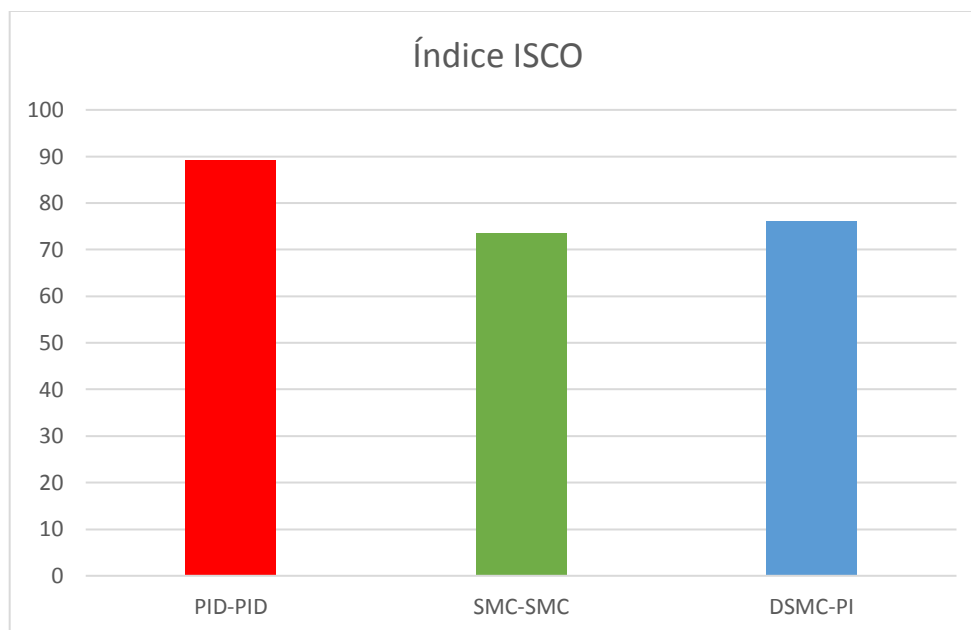


Figura 3.18. Índice de desempeño ISCO

anteriormente gastan casi la misma cantidad de energía, y este consumo es menor que la energía que gasta el esquema de control PID-PID, dejando a este último como el esquema con más bajo desempeño de los tres controladores.

Tomando en cuenta lo mencionado anteriormente sobre los índices ISE e ISCO se puede decir que el esquema de control que presenta el mejor desempeño es el controlador DSMC-PI y si se toma en cuenta lo mencionado respecto a las acciones de control y las respuestas de temperatura el controlador que presenta más ventajas de igual manera el esquema DSMC-PI.

3.2 CONCLUSIONES

- Se diseñaron esquemas en cascada PID-PID, SMC-SMC y DSMC-PI y se comprobó el funcionamiento de cada uno de ellos en un proceso simulado en MatLab Simulink de un tanque de mezclado, dando como resultado que el esquema DSMC-PI presenta un mejor desempeño en comparación a los esquemas SMC-SMC y PID-PID.

- De los resultados obtenidos de los tres esquemas, el desempeño más bajo de los tres esquemas es el presentado por el control PID-PID, debido a retardos de tiempo del proceso muy grandes y variables.
- El concepto de “hardware in the loop” permite realizar pruebas de manera sencilla, ya que se puede cambiar los parámetros de un proceso o agregar perturbaciones y ver el efecto sobre la respuesta, además permite comprobar el funcionamiento de controladores físicos y conocer si éstos se encuentran funcionando dentro de un rango establecido.
- El tiempo de muestreo es un parámetro muy importante a la hora de implementar los controladores en un sistema embebido y si éste no es el adecuado, la acción de control resultante hará que el proceso presente problemas de seguimiento del valor de consigna.
- La acción de control que proporciona el sistema embebido es un parámetro muy importante a analizar, debido a que permite observar cómo se comportará el accionamiento conectado al controlador, y si la acción no es buena o presenta movimientos bruscos, implica que el accionamiento podría sufrir sobre esfuerzos mecánicos e inclusive provocar su avería.

3.3 RECOMENDACIONES

- Si se quiere replicar el proyecto, se recomienda implementar el modelo matemático del proceso mediante bloques, puesto que es mucho más fácil implementar condiciones iniciales del proceso.
- Para obtener los parámetros de los modelos de primer orden con retardo de tiempo, se recomienda utilizar el método de Alfaro, ya que este brinda una aproximación mucho más cercana que cualquier otro método.
- Para desarrollo de futuros trabajos relacionados con Hardware in the loop se puede tomar en cuenta el presente trabajo de titulación como base para la implementación de controladores en un sistema embebido.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] Camacho Oscar, Rosales Andrés, and Rivas Franklin, “Control de procesos”, Primera Edición. Quito: EPN editorial, 2020.
- [2] Coronel María and Mora Luis, “Estrategias de control por modos deslizantes para sistemas en cascada,” Dic. 2019.
- [3] Herrera Marco, Camacho Oscar, Leiva Hugo, and Smith Carlos, “Una aproximación del control de modos deslizantes dinámicos para procesos químicos”, Nov. 2019.
- [4] Brunete Alberto, “1.5 Sistemas de control | Introducción a la Automatización Industrial.” https://bookdown.org/alberto_brunete/intro_automatica/sistemas-de-control.html (accedido Ene. 20, 2022).
- [5] Alfaro Víctor, “Identificación de modelos de orden reducido a partir de la curva de reacción del proceso,” Sep. 2006.
- [6] Smith Carlos and Corripio Armando, “Control automático de procesos”, Primera Edición. México: Limusa, 1991.
- [7] Sabogal Oscar, “Diseño, construcción y validación de un sistema de control de posición auto sintonizable para un cilindro neumático de doble efecto,” 2016.
- [8] Camacho Oscar and Smith Carlos, “Control por modos deslizantes: una aproximación para regular procesos químicos no lineales”.
- [9] Escobar Larry, “Conceptos básicos de procesamiento digital de señales”. 2008.
- [10] NI, “¿Qué es el hardware-in-the-loop?” <https://www.ni.com/es-cr/innovations/white-papers/17/what-is-hardware-in-the-loop-.html#section-1885360111> (accedido Ene. 22, 2022).
- [11] Ruiz José, “Manual de Programación Arduino”, Primera Edición.
- [12] Arduino, “Arduino Mega 2560” <https://www.arduino.cc/en/Main/arduinoBoardMega2560> (accedido Ene. 23, 2022).
- [13] MathWorks, “Simulink” <https://la.mathworks.com/help/simulink/> (accedido Ene. 23, 2022).

5 ANEXOS

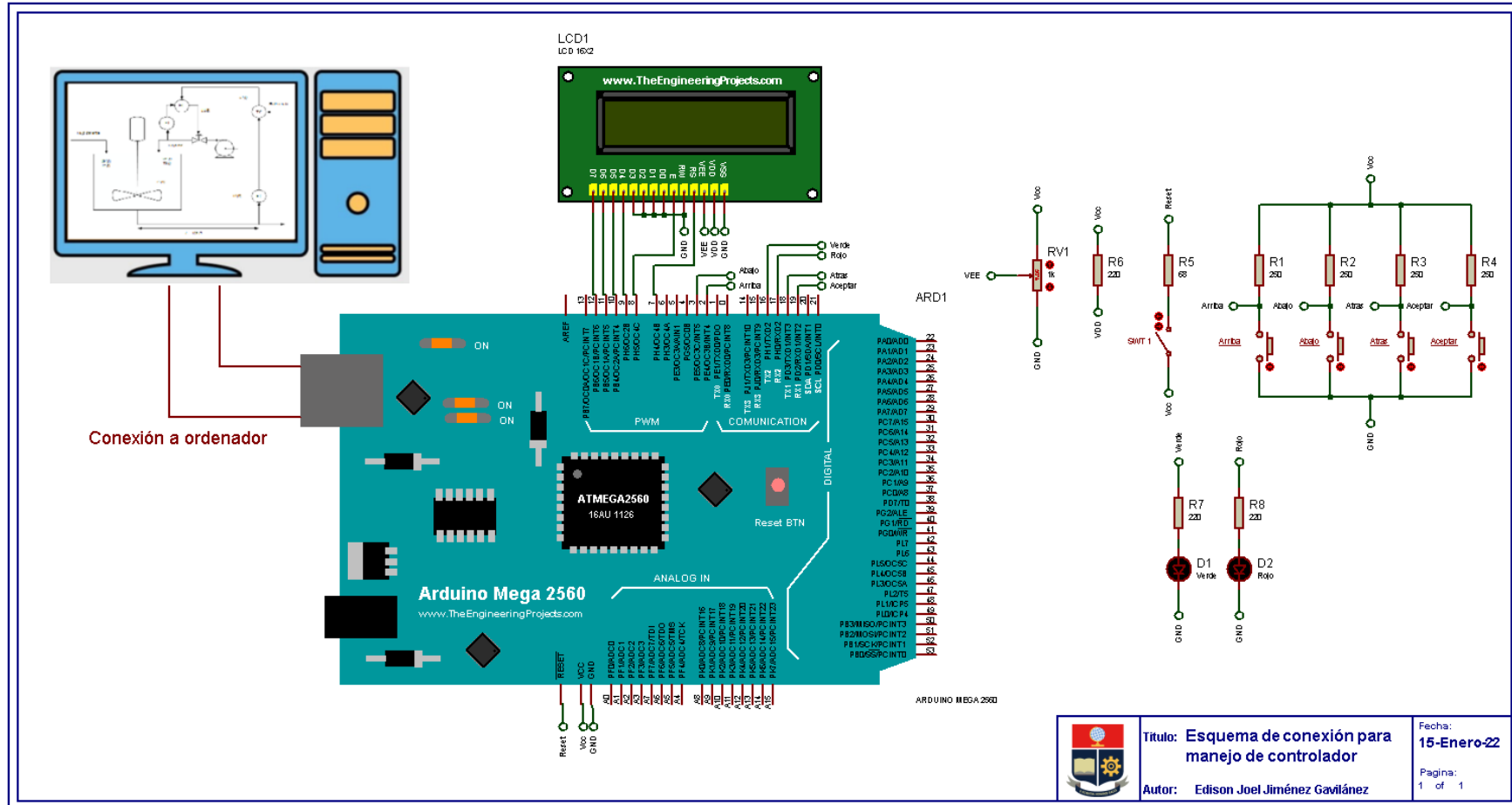
Anexo I: Esquema de conexión.

Anexo II: Manual de usuario.

Anexo III: Placa de circuito impreso

6 ANEXO I

ESQUEMA DE CONEXION



7 ANEXO II

MANUAL DE USUARIO

El siguiente manual explica el manejo del trabajo de titulación. Para esto necesario el contar con una versión de MatLab 2020b o superior, debido a que en versiones inferiores no funcionara por la compatibilidad.

MANEJO DEL MODULO DEL CONTROLADOR

Primero el módulo debe energizarse y conectarse con el ordenador mediante un cable USB como se muestra en la Figura 5.1.

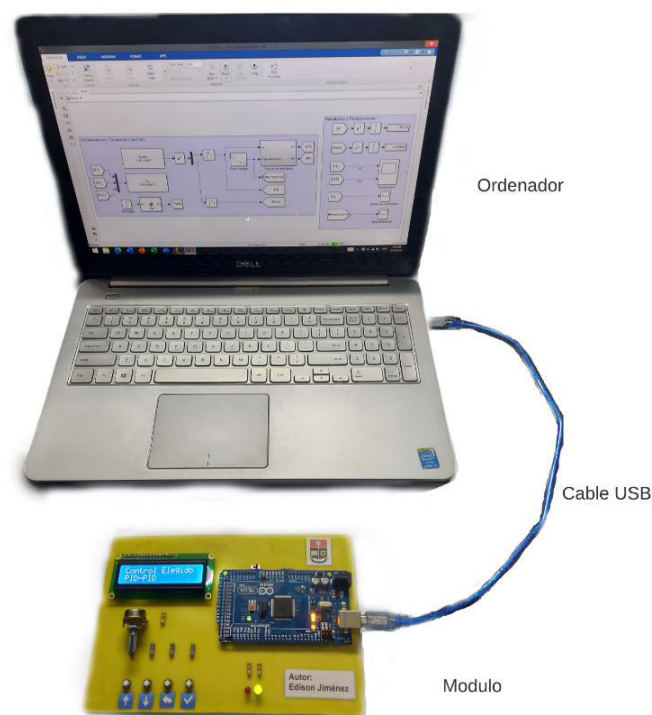


Figura 5.1. Conexión ordenador-modulo

Una vez realizada la conexión el LCD del módulo mostrara dos mensajes. El primero que indica el nombre de la universidad y el nombre del autor del proyecto y el segundo mensaje que menciona que se puede seleccionar el controlador (Figura 5.2).

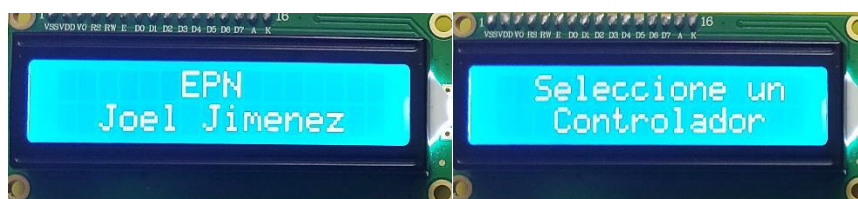


Figura 5.2. Primer y segundo mensaje

Una vez mostrado el segundo mensaje el módulo se vera de la siguiente forma:

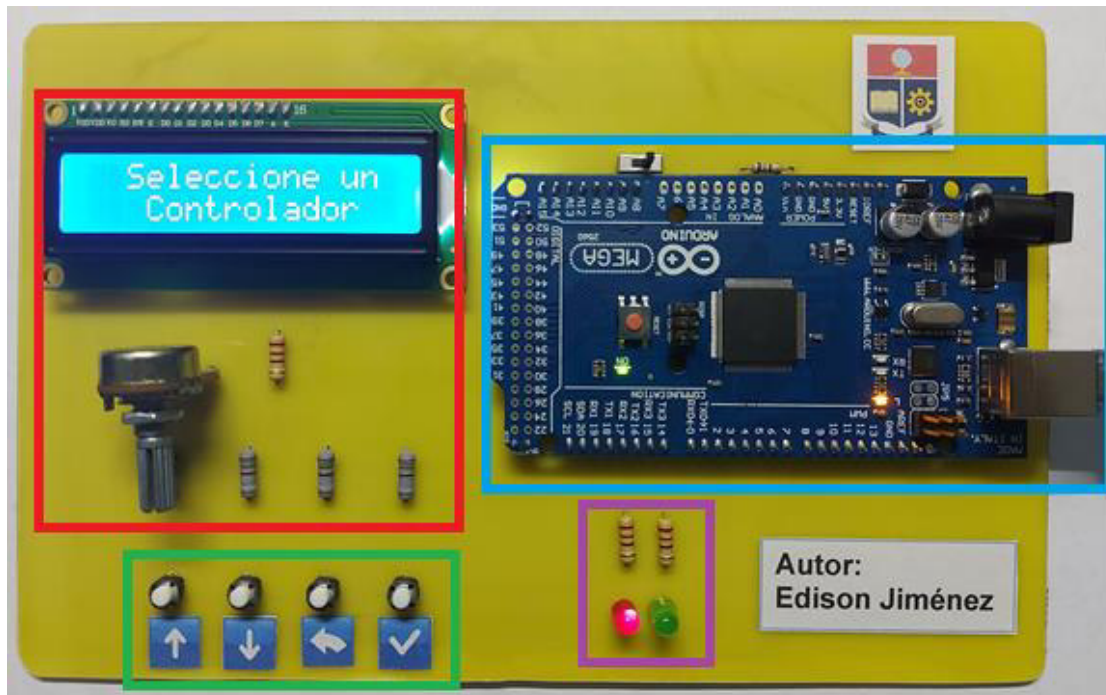


Figura 5.3. Modulo.

La Figura 5.3 se encuentra dividida en 4 secciones que permitirán seleccionar un controlador y verificar si se encuentra en funcionamiento. Para acceder al menú de selección de controladores se utiliza la sección verde encerrada en la Figura 5.3, donde se presiona el pulsador con el símbolo de flecha hacia arriba. Una vez presionado el LCD mostrara en un mensaje la primera opción de control disponible, todos los controles disponibles se observan en la Figura 5.4.



Figura 5.4. Opciones disponibles de control

Para navegar en este menú basta con presionar los pulsadores con las flechas arriba y abajo de la sección encerrada en verde de la Figura 5.3.

Para seleccionar una de las tres opciones proporcionadas, se presiona el pulsador con el símbolo de visto, el LCD mostrara uno de los tres mensajes que se muestra en la Figura 5.5 dependiendo de que controlador se seleccione.



Figura 5.5. Opciones seleccionadas de control

Después de seleccionar un controlador un led de color verde de la sección encerrada por el recuadro de color morado de la Figura 5.3 se encenderá indicando se encuentra listo para comunicarse con Simulink.

En caso de que el control seleccionado no sea el deseado se presiona el botón con el símbolo de flecha hacia atrás permitiendo volver de nuevo al menú de controladores. Si no se encuentra seleccionado ningún controlador un led de color rojo se encenderá, tal como se ve en la Figura 5.3, indicando que el módulo no está listo para ejecutarse junto con Simulink.

El módulo conectado y listo para empezar la simulación en conjunto con Simulink se muestra en la Figura 5.6.

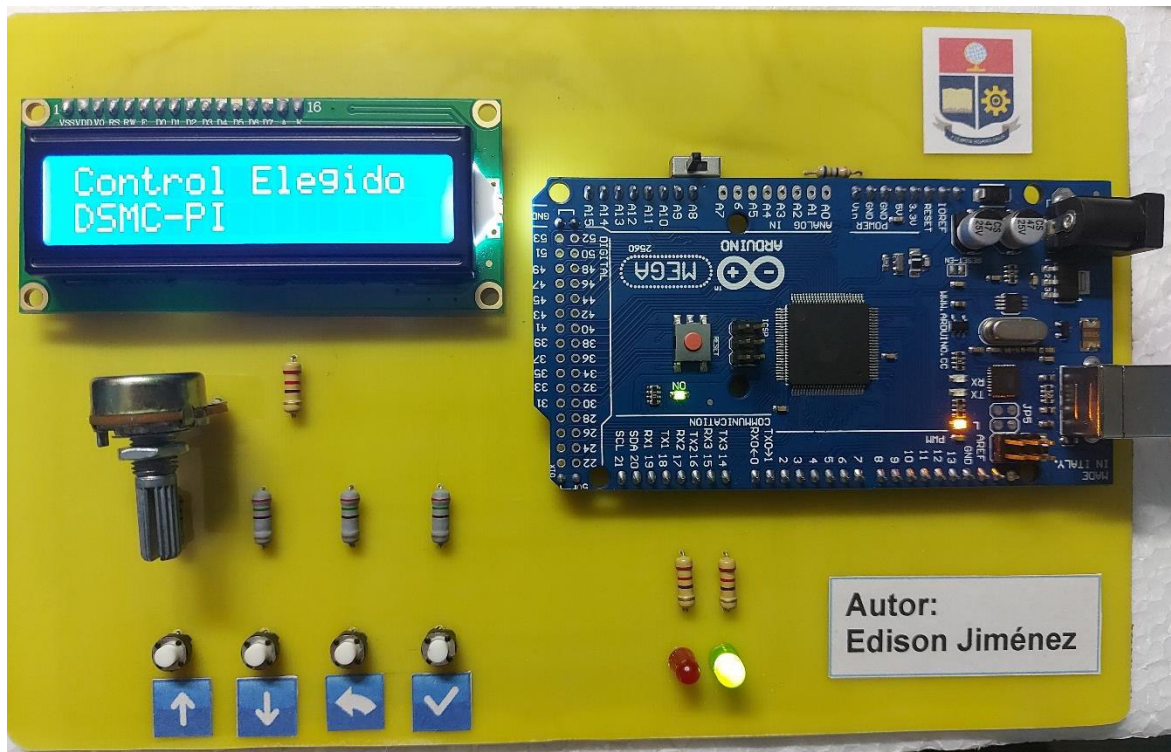


Figura 5.6. Módulo listo para ejecución.

INICIO DE SIMULINK

Para iniciar el programa en Simulink primero se debe cargar el archivo con extensión .slx en la parte izquierda como se indica en la Figura 5.7. Una vez el archivo este cargado se escribe la palabra “Planta” en la ventana de comandos de MatLab y presionando a continuación la tecla Enter.

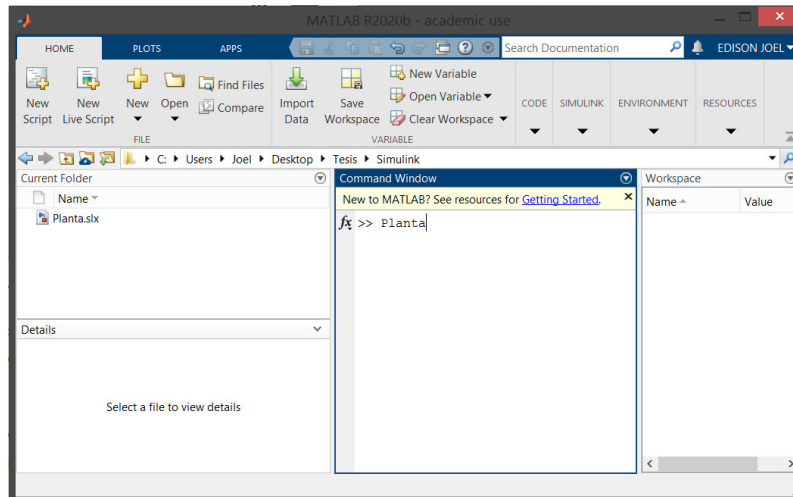


Figura 5.7. Carga de archivo en MatLab

PANTALLA DE SIMULINK

Después de presionar Enter se abrirá una pantalla como se muestra en la Figura 5.8. Esta pantalla contiene dos secciones. La primera donde se encuentra la parte de “Comunicación y Tanque de mezclado” y la segunda que contiene las “Respuestas”, dichas secciones se las puede apreciar de mejor manera en las Figuras 5.9 y 5.10 respectivamente.

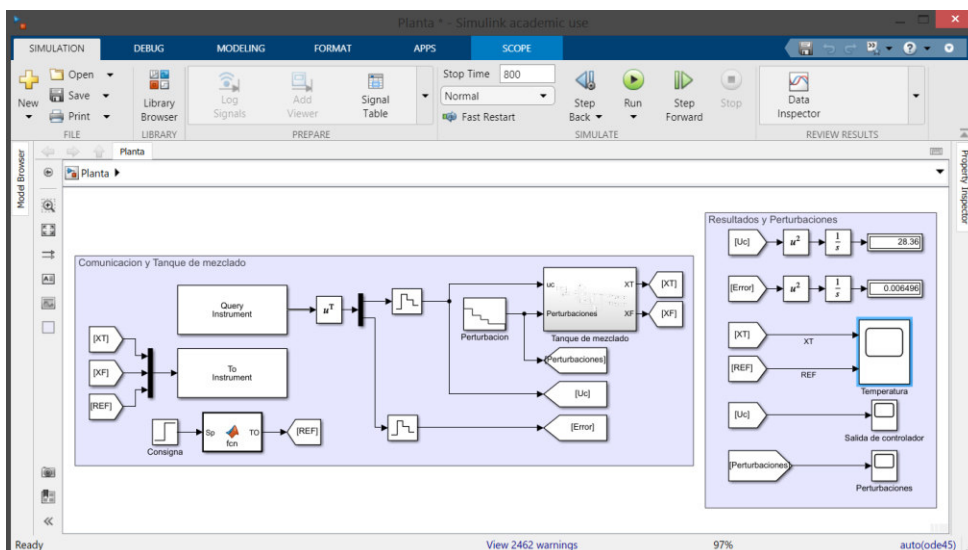


Figura 5.8. Pantalla de Simulink.

En la sección de “Comunicación y tanque de mezclado”, se tiene los bloques que permiten realizar la comunicación entre Simulink y la tarjeta Arduino, también se tiene un bloque que contiene el modelo matemático del tanque de mezclado con retardo.

En caso de querer agregar más perturbaciones simplemente se abre el bloque “Perturbacion” y se modifica los valores de tiempos y perturbaciones.

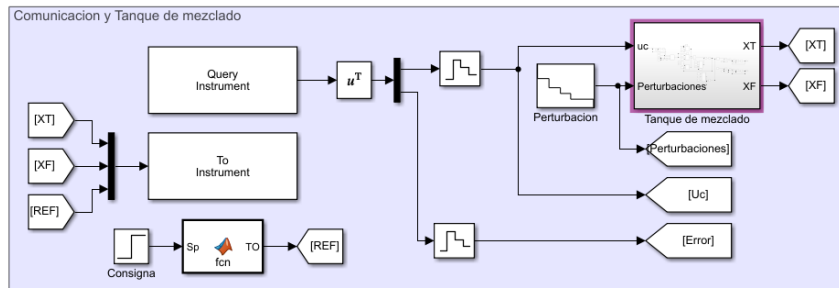


Figura 5.9. Comunicación y tanque de mezclado.

En la sección de respuestas se tiene:

- Visualizador para el índice ISCO.
- Visualizador para el índice ISE.
- Visualizador para respuesta de temperatura.
- Visualizador para la señal de control que proporciona el embebido.
- Visualizador para la señal de perturbaciones.

Los visualizadores de temperatura, acción de control y perturbaciones se los abre haciendo doble clic doble ellos.

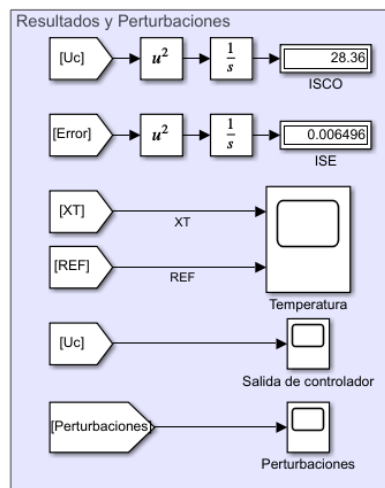


Figura 5.10. “Respuestas”

Para inicial la simulación junto con el módulo primero se configura un tiempo de 800 en la sección encerrada en rojo como se observa en la Figura 5.11. y se arranca la simulación dándole clic en el botón encerrado por el recuadro verde.

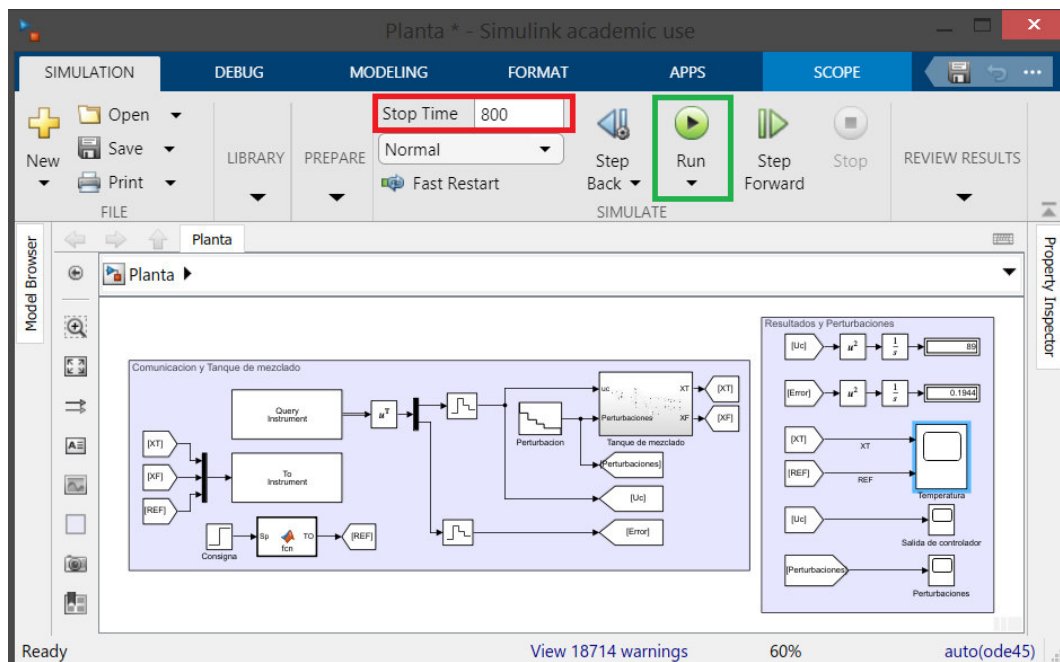


Figura 5.11. Configuración de tiempo y arranque de simulación.

Una vez el programa Simulink se esté ejecutando, unos leds de color amarillo (RX y TX) de la tarjeta Arduino montada en el módulo empezaran a parpadear (Figura 5.12) rápidamente confirmado que existe una transferencia de datos entre el módulo y el software.

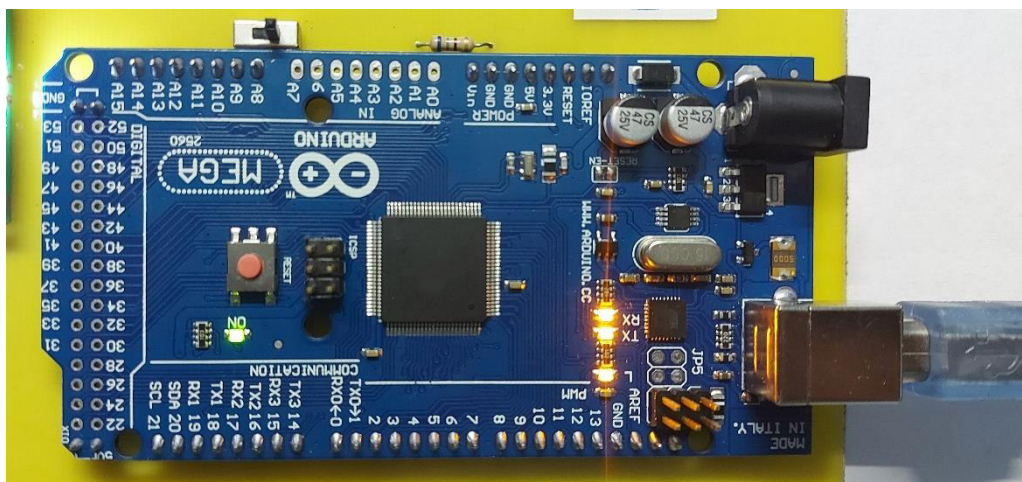


Figura 5.12. Leds indicadores de transferencia de datos

Terminado el tiempo de simulación ya se podrán observar los valores ISE e ISCO proporcionados por el control elegido en el módulo en la pantalla de Simulink y puede acceder a los resultados de temperatura, acción de control y perturbaciones como se mencionó con anterioridad.

8 ANEXO III

PLACA DE CIRCUITO IMPRESO (PCB)

En la figura 5.13 se aprecia el circuito impreso donde se montan los componentes para el manejo de la tarjeta embebida. Dicho circuito impreso fue desarrollado con ayuda del software Proteus

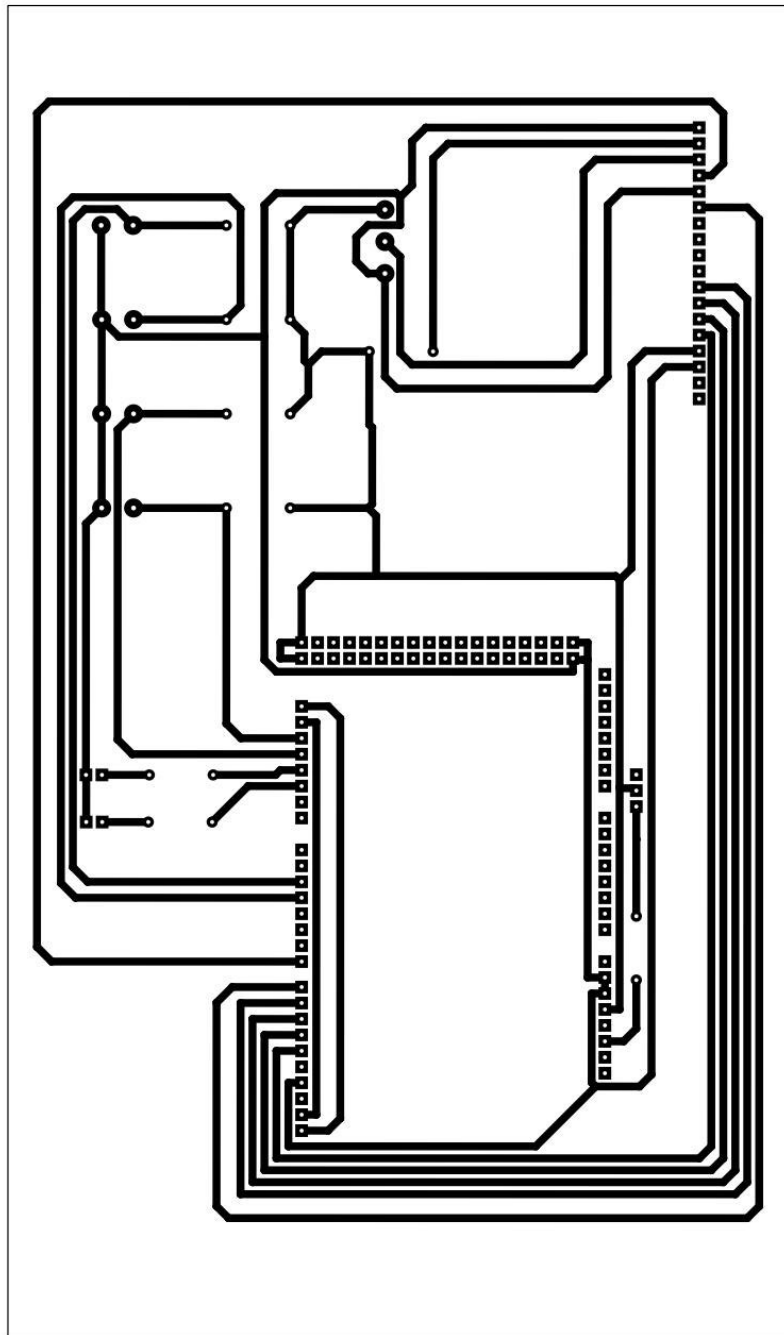


Figura 5.13. Circuito impreso para implementación de la interfaz de manejo de la tarjeta embebida