



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE  
CONTROLADORES DISCRETOS PID Y DIFUSO  
EVALUADOS EN UN SISTEMA DE TRANSFERENCIA  
DE CALOR”

**TRABAJO DE TITULACIÓN**

Previo a la obtención del Título de:

**MAGISTER EN AUTOMATIZACIÓN Y CONTROL  
INDUSTRIAL**

FABIÁN CELSO GUNSHA MAJI

GUAYAQUIL – ECUADOR

AÑO: 2018

## **AGRADECIMIENTOS**

Mis más sinceros agradecimientos a mis maestros, por compartir su experiencia en cada uno de las cátedras y responder cada una de mis dudas que me permitieron crecer profesionalmente. Un especial agradecimiento al MSc. Efrén Herrera por su constante apoyo en el desarrollo de este trabajo de titulación.

## DEDICATORIA

El presente proyecto lo dedico a mis padres María y Juan pilares fundamentales en mi educación y desarrollo profesional, quienes, con su amor, ejemplo, sabiduría, me han demostrado que en la vida todo es posible, que el sacrificio, perseverancia, siempre tiene su recompensa

## **TRIBUNAL DE EVALUACIÓN**

.....  
**PhD. César Antonio Martín Moreno**  
**SUBDECANO DE LA FIEC**

.....  
**MSc. Efrén Vinicio Herrera Muentes**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

.....  
**MSc. Dennys Dick Cortez Álvarez**  
**MIEMBRO PRINCIPAL DEL TRIBUNAL**



## DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....  
Ing. Fabián Celso Gunsha Maji

## RESUMEN

La presente tesis realiza la comparación de los controladores PID discretos y difuso, evaluados en un sistema de transferencia de calor considerando la ducha eléctrica para este propósito por ser de mayor uso y quizá el de menor costo, mediante la construcción de un sistema embebido con un convertidor AC-AC por ángulo de fase, desarrollada con la tarjeta STM32F4Discovery y las placas: detector de cruce por cero, interfaz de potencia, modulo Bluetooth como principales elementos, mientras la programación es desarrollada en Matlab-Simulink con la librería Waijung.

Con el sistema embebido desarrollado se adquiere las señales de temperatura que se transfieren mediante Bluetooth a la computadora y son almacenadas en Excel mediante la aplicación desarrollada en LabView. Los datos obtenidos permiten identificar la función de transferencia mediante Matlab-Ident que a la vez permiten desarrollar el PID estándar con la aplicación PID Tuner de Matlab y encontrar las constantes proporcional, integral, derivativo que sirve para desarrollar los PID discretos de forma posicional, velocidad y difuso.

Los PID discretos posicional, velocidad y difuso son simulados con o sin ruido en Simulink y también son desarrollados con la librería Waijung para ser programados en el sistema embebido de manera individual, para realizar el control de temperatura de la ducha eléctrica y verificar su resultado.

Con los resultados obtenidos se realiza la comparación entre los PID simulados y reales, donde se establece que el PID difuso como la mejor opción, debido al tiempo de establecimiento menor en la simulación con o sin ruido, mientras en el real alcanza más rápido la temperatura de referencia con una oscilación de  $\pm 1.5^{\circ}\text{C}$  aproximadamente que es imperceptible para el usuario, con una potencia media de consumo de 1806.92 vatios que determina que el sistema embebido ahorra consumo de energía.

## ÍNDICE GENERAL

|   |      |
|---|------|
| AGRADECIMIENTOS.....  | ii   |
| DEDICATORIA .....   | iii  |
| TRIBUNAL DE EVALUACIÓN .....  | iv   |
| DECLARACIÓN EXPRESA.....  | v    |
| RESUMEN.....  | vi   |
| ÍNDICE GENERAL.....   | vii  |
| ÍNDICE DE FIGURAS.....  | x    |
| ÍNDICE DE TABLAS .....  | xiii |
| CAPÍTULO 1.....   | 1    |
| 1. ANTECEDENTES.....  | 1    |
| 1.1 Planteamiento del problema ahorro energético en sistemas de calentamiento de agua. .... | 1    |
| 1.2 Objetivos.....  | 2    |
| 1.2.1 Objetivo General.....   | 2    |
| 1.2.2 Objetivos Específicos. ....   | 2    |
| 1.3 Metodología.....  | 2    |
| 1.3.1 Método.....   | 2    |
| 1.3.2 Técnica. ....   | 3    |
| 1.3.3 Instrumento.....  | 3    |
| 1.4 Estado del Arte.....  | 3    |
| 1.4.1 Sistemas de control de temperatura de ducha eléctrica ....                            | 3    |
| 1.4.2 Comparación de controladores.....   | 5    |
| CAPÍTULO 2.....   | 8    |
| 2. MARCO TEÓRICO.....   | 8    |

|                 |  |    |
|-----------------|--|----|
| 2.1             | Trasferencia de Calor.....   | 8  |
| 2.1.1           | Conducción.....  | 8  |
| 2.1.2           | Convección.....  | 9  |
| 2.1.3           | Radiación.....   | 9  |
| 2.2             | Convertidores AC-AC.....   | 10 |
| 2.2.1           | Control on-off.....  | 10 |
| 2.2.2           | Control por ángulo de fase.....  | 12 |
| 2.3             | Sistemas de Control utilizados en intercambiadores de calor. ....        | 13 |
| 2.3.1           | Controlador PID discreto. ....   | 14 |
| 2.3.2           | Controlador PID difuso. ....   | 15 |
| CAPÍTULO 3..... |  | 20 |
| 3.              | DISEÑO DEL SISTEMA DE CONTROL EMBEBIDO.....                              | 20 |
| 3.1             | Diseño del control por ángulo de fase. ....                              | 21 |
| 3.1.1           | Circuito detector de cruce por cero.....                                 | 21 |
| 3.1.2           | Circuito Interfaz de potencia de corriente alterna. ....                 | 23 |
| 3.1.3           | Circuito de control embebido. ....                                       | 24 |
| 3.2             | Diseño del controlador.....  | 26 |
| 3.2.1           | Identificación de la planta. ....  | 27 |
| 3.2.2           | Algoritmo del PID discreto. ....   | 32 |
| 3.2.3           | Algoritmo del PID difuso. ....   | 37 |
| CAPÍTULO 4..... |  | 41 |
| 4.              | ANÁLISIS COMPARATIVO DE LOS RESULTADOS.....                              | 41 |
| 4.1             | Evaluación del Desempeño del control embebido con PID Discreto.<br>..... | 41 |
| 4.1.1           | Comparación del PID posicional simulado y real.....                      | 42 |
| 4.1.2           | Comparación del PID de velocidad simulado y real.....                    | 45 |
| 4.2             | Evaluación del Desempeño del control embebido con PID Difuso.<br>.....   | 48 |
| 4.2.1           | Comparación del PID difuso simulados y real. ....                        | 48 |

|       |  |    |
|-------|--|----|
| 4.3   | Comparación de los controladores PID discreto y difuso. .... | 52 |
| 4.3.1 | Comparación de PID discretos y difuso sin ruido. ....        | 52 |
| 4.3.2 | Comparación de PID discretos y difuso con ruido. ....        | 55 |
| 4.3.3 | Comparación de PID discretos y difuso reales. ....           | 59 |
|       | CONCLUSIONES Y RECOMENDACIONES .....                         | 63 |
|       | BIBLIOGRAFÍA.....  | 65 |
|       | ANEXOS.....  | 67 |

## ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 2.1: Transferencia de calor. ....  | 8  |
| Figura 2.2: Circuito de convertidor AC-AC.....                                      | 10 |
| Figura 2.3: Control on-off. ....  | 11 |
| Figura 2.4: Control por ángulo de fase.....   | 13 |
| Figura 2.5: Diagrama de bloques control a) PID Posicional, b) PID de velocidad..... | 15 |
| Figura 2.6: PID difuso con tres entradas.....                                       | 16 |
| Figura 2.7: PID difuso con PI + PD. ....  | 16 |
| Figura 2.8: PID difuso con PI + PD con cambio $de(t)/dt$ por $-dy(t)/dt$ ....       | 17 |
| Figura 2.9: Respuesta a una entrada escalón. ....                                   | 18 |
| Figura 3.1: Diagrama de bloques del sistema de control embebido. ....               | 20 |
| Figura 3.2: Circuito detector de cruce por cero.....                                | 21 |
| Figura 3.3: Señales de detector de cruce por cero. ....                             | 22 |
| Figura 3.4: Circuito Interfaz de potencia AC. ....                                  | 23 |
| Figura 3.5: Señales de control y de carga. ....                                     | 24 |
| Figura 3.6: Circuito de control embebido. ....                                      | 25 |
| Figura 3.7: a) Variables; Configuraciones b) tarjeta, c) UART, d) LCD.....          | 27 |
| Figura 3.8: Switch case de la variable contador.....                                | 28 |
| Figura 3.9: Cálculo de variables (tiempo y tiempo en milisegundos). ....            | 28 |
| Figura 3.10: Cálculo de temperatura.....  | 29 |
| Figura 3.11: Configuración serial. ....   | 29 |
| Figura 3.12: Cálculo de VRMS en LabView.....  | 30 |
| Figura 3.13: Datos Capturados. ....   | 31 |
| Figura 3.14: Gráficas de funciones de transferencia.....                            | 32 |
| Figura 3.15: Señales de respuesta generada y ajustada con PID tuner. ....           | 32 |
| Figura 3.16: Subsistema de PID forma a) Posicional y b) Velocidad. ....             | 33 |
| Figura 3.17: Configuración inicial para PID forma Posicional. ....                  | 34 |

|  |    |
|--|----|
| Figura 3.18: Cálculo de la temperatura de referencia. ....                             | 34 |
| Figura 3.19: Cálculo error y conexión con PID posicional y DAC .....                   | 35 |
| Figura 3.20: Envío de datos por puerto UART .....                                      | 35 |
| Figura 3.21: Datos enviados por el LCD .....   | 36 |
| Figura 3.22: Cálculo error y conexión con PID velocidad y DAC .....                    | 36 |
| Figura 3.23: Fuzzy Logic Designer .....  | 38 |
| Figura 3.24: Funciones de Membresía PID difuso. ....                                   | 39 |
| Figura 3.25: Superficie de Control. ....   | 39 |
| Figura 3.26: PID Difuso.....   | 40 |
| Figura 4.1: Respuestas del PID posicional simulado y real. ....                        | 42 |
| Figura 4.2: Voltaje RMS de la planta con PID posicional simulados y real. ..           | 44 |
| Figura 4.3: Corriente RMS de la planta con PID posicional simulados y real.<br>.....   | 44 |
| Figura 4.4: Respuestas del PID de velocidad simulados y real. ....                     | 45 |
| Figura 4.5: Voltaje RMS de la planta con PID de velocidad simulados y real.<br>.....   | 47 |
| Figura 4.6: Corriente RMS de la planta con PID de velocidad simulados y real.<br>..... | 47 |
| Figura 4.7: Respuestas del PID difuso simulados y real. ....                           | 49 |
| Figura 4.8: Voltaje RMS de la planta con PID difuso simulados y real. ....             | 51 |
| Figura 4.9: Corriente RMS de la planta con PID difuso simulados y real.....            | 51 |
| Figura 4.10: Respuestas de los PID simulados sin ruido. ....                           | 53 |
| Figura 4.11: Respuestas de los PID simulados sin ruido al incremento de 3°C.<br>.....  | 54 |
| Figura 4.12: Voltaje RMS Aplicado a la planta por PID simulados sin ruido.             | 55 |
| Figura 4.13: Respuestas de los PID simulados con ruido.....                            | 56 |
| Figura 4.14: Respuestas de los PID simulados con ruido al incremento de 3°C.<br>.....  | 57 |
| Figura 4.15: Voltaje RMS Aplicado a la planta por PID simulados con ruido.<br>.....    | 58 |

|  |    |
|--|----|
| Figura 4.16: Respuestas de los PID reales.....                       | 59 |
| Figura 4.17: Respuestas de los PID reales al incremento de 3°C. .... | 60 |
| Figura 4.18: Voltaje RMS Aplicado a la planta por PID real.....      | 61 |



## ÍNDICE DE TABLAS

|   |    |
|---|----|
| Tabla 1: Matriz de 9 reglas difusas .....   | 18 |
| Tabla 2: Matriz de 25 reglas difusas .....  | 19 |
| Tabla 3: Funciones de transferencia.....  | 31 |
| Tabla 4: Constantes PID y valores de respuesta transitoria. ....  | 33 |
| Tabla 5: Parámetros de las entradas de controlador PID difuso.....  | 37 |
| Tabla 6: Parámetros de salida de controlador PID difuso.....  | 38 |
| Tabla 7: Valores de ganancia para Simulink.....   | 40 |
| Tabla 8: Características de respuestas del PID posicional simulados y real.                                     | 43 |
| Tabla 9: Características de respuestas al incremento 3°C al PID posicional simulados y real. ....               | 43 |
| Tabla 10: Voltaje y Corriente RMS de la planta con PID posicional simulados y real.....                         | 45 |
| Tabla 11: Características de respuestas del PID de velocidad simulados y real. ....                             | 46 |
| Tabla 12: Características de respuestas al incremento 3°C al PID de velocidad simulados y real. ....            | 46 |
| Tabla 13: Voltaje y Corriente RMS de la planta con PID de velocidad simulados y real. ....                      | 48 |
| Tabla 14: Características de respuestas del PID difuso simulados y real.....                                    | 49 |
| Tabla 15: Características de respuestas al incremento 3°C al PID difuso simulados y real. ....                  | 50 |
| Tabla 16: Voltaje y Corriente RMS de la planta con PID difuso simulados y real. ....                            | 52 |
| Tabla 17: Características de respuestas de los PID simulados sin ruido.....                                     | 53 |
| Tabla 18: Características de respuestas de los controladores PID simulados sin ruido al incremento de 3°C. .... | 54 |

|   |    |
|---|----|
| Tabla 19: Voltaje y Corriente RMS de la planta con PID simulados sin ruido.<br>.....                              | 55 |
| Tabla 20: Características de respuestas de los PID simulados con ruido. ...                                       | 56 |
| Tabla 21: Características de respuestas de los controladores PID simulados<br>con ruido al incremento de 3°C..... | 57 |
| Tabla 22: Voltaje y Corriente RMS de la planta con PID simulados con ruido.<br>.....                              | 58 |
| Tabla 23: Características de respuestas de los PID reales. ....   | 60 |
| Tabla 24: Características de respuestas de los controladores PID reales al<br>incremento de 3°C. ....             | 61 |
| Tabla 25: Voltaje y Corriente RMS de la planta con PID reales.....  | 62 |

# CAPÍTULO 1

## 1. ANTECEDENTES.

### 1.1 Planteamiento del problema ahorro energético en sistemas de calentamiento de agua.

Los sistemas de transferencia de calor por conducción, convección y radiación que existen en el mercado en múltiples aplicaciones tales como calentamiento de agua, material o sustancia, aclimatación de habitaciones o lugares de trabajo etc.

Muchos de estos sistemas poseen sistemas de control de distinto tipo de acuerdo a su aplicación y necesidad, pero en algunos casos solo cuentan con un control on-off o ningún tipo de control de temperatura, estos sistemas en muchos casos no consideran el aporte inicial de temperatura y provocan un excesivo consumo de corriente.

Dentro de este gran contexto de aplicaciones el de mayor uso y quizá el de menor costo es la ducha eléctrica, que de acuerdo a la marca y tipo poseen un control de temperatura de dos niveles (fría-caliente) respectivamente, mediante la resistencia eléctrica interna encargada de calentar el agua, bajo esta perspectiva el control de temperatura ha sido el problema más común, debido al caudal variable producido por la distribución de la tubería y apertura de llaves dentro de los problemas más comunes.

Pero, la temperatura de entrada del agua podría provocar que la ducha eléctrica alcance una temperatura adecuada en mayor o menor tiempo y provocar que el consumo de energía eléctrica aumente o disminuya respectivamente.

Considerando todo el entorno descrito anteriormente y conociendo que el controlador más utilizado es el PID, se estableció realizar la comparación entre un control PID discreto y difuso en un sistema embebido para el control de temperatura de la ducha eléctrica y conseguir un posible ahorro de energía eléctrica.

## **1.2 Objetivos.**

### **1.2.1 Objetivo General.**

Analizar comparativamente el desempeño de controladores discretos tanto PID como difuso, evaluados en un sistema de transferencia de calor, mediante un sistema embebido con un convertidor AC-AC por ángulo de fase, que permita el control más eficiente de temperatura y un posible ahorro de energía eléctrica.

### **1.2.2 Objetivos Específicos.**

- Determinar el modelo del sistema con datos obtenidos a través de una tarjeta STM32 Discovery y procesados mediante Matlab.
- Implementar el control de disparo de tiristores por ángulo de fase para el control de temperatura del sistema de transferencia de calor (ducha eléctrica).
- Diseñar el sistema embebido para el control de temperatura del sistema de transferencia de calor (ducha eléctrica).
- Programar los algoritmos de control PID digital y difuso en el sistema embebido, para el control de temperatura y la evaluación del desempeño.

## **1.3 Metodología.**

Para el análisis de controladores PID discreto y difuso será a través de las siguientes metodologías:

- Sistemas Discretos de Control. - Para el desarrollo de PID discreto y análisis de la planta.
- Control Inteligente. - Para el desarrollo de PID difuso mediante reglas lingüísticas.

### **1.3.1 Método.**

Inductivo. – A través de un sistema embebido con control PID discreto o difuso, que permitirá controlar la energía suministrada a la resistencia térmica, para el control de temperatura de la ducha eléctrica con un

posible ahorro de energético y podrá tener el mismo resultado en los distintos tipos de ducha eléctrica.

### 1.3.2 Técnica.

Las técnicas que permitirán arribar a los resultados esperados son:

- Controlador PID discreto
- Controlador PID difuso

### 1.3.3 Instrumento.

Los instrumentos que permitirán recolectar, registrar, procesar la información son:

- Prototipo de un sistema embebido con PID discreto o difuso para el control de temperatura de agua de la ducha eléctrica.
- Pruebas de control de temperatura del agua de la ducha eléctrica.

## 1.4 Estado del Arte.

### 1.4.1 Sistemas de control de temperatura de ducha eléctrica

Dentro de los diferentes sistemas de transferencia de calor, quizá el de mayor uso es la ducha eléctrica debido a su costo y tamaño, pero el control de temperatura, consumo de energía eléctrica y agua, ha sido el problema más común debido al caudal variable, presión, apertura de llaves etc.

Debido a este inconveniente se han desarrollado algunos trabajos tales como:

El trabajo de tesis titulado “**Diseño y construcción de un prototipo de ducha eléctrica de alta eficiencia**” consiste en un sistema de recuperación de calor mediante un intercambiador de calor de 90x65 cm, mediante un serpentín de media pulgada de cobre ubicado en el piso, para aprovechar la temperatura del agua producida por la ducha eléctrica. Además, la ducha sufrió modificaciones agregando una reducción de efecto Venturi y la modificación del rociador para lograr menor consumo de agua. Para el control de la temperatura utiliza un circuito electrónico

controlado por un microcontrolador PIC16F877A, un sensor de temperatura LM35, y relé para conmutar la ducha. Bajo todos los parámetros antes descritos, los resultados obtenidos fueron ahorro de agua de un 51.61% comparados con el consumo de agua de la ducha normal, el ingreso de agua con temperatura ambiente de 14°C sufrió una contribución de 5 a 7°C, provocando que la temperatura del agua al ingreso a la ducha varía entre 19 a 21°C, y debido al control ON/OFF el tiempo de encendido sea del 42.37% y apagado sea del 57.63% por cada uso de la ducha, produciendo un ahorro de 745.33 (Kwh) al año [1].

El trabajo de tesis titulado **“Diseño, Modelación, Simulación e Implementación de un Sistema de Control de Temperatura para la Ducha Eléctrica”**, realiza el control de temperatura de la ducha eléctrica a través de un controlador OMEGA CN9400 y un relé estado sólido SSRL660DC100 para controlar la resistencia eléctrica, además incluye el control de caudal de agua mediante un servo motor acoplado a la válvula de la ducha eléctrica controlado por un microcontrolador ATMEGA328P y con un sensor de movimiento para encender y apagar el sistema. Con estos elementos el mejor resultado obtenido a una temperatura de referencia de 35°C y con un control proporcional de 2.5 se obtuvieron una variación de  $\pm 3^\circ\text{C}$  y un ahorro aproximado del 30% de volumen de agua sin control de caudal [2].

El trabajo de tesis titulado **“Sistema de Control Automático de una ducha eléctrica mediante Reconocimiento de la Voz para personas con discapacidad en las Extremidades Superiores”**, permite el control automático de una ducha eléctrica mediante comando de voz para personas con capacidades especiales en las extremidades superiores, estableciendo tres niveles de temperatura (temperada-tibia-caliente) a través de comandos de voz mediante un módulo EasyVR, para mover la válvula de agua se utiliza un servo motor MS-645SG todo controlado a través de un Arduino, mientras que el control por ángulo de fase actúa sobre la resistencia eléctrica mediante un TRIAC de potencia, para

obtener los tres niveles de temperatura establecidos, todo a través del microcontrolador PIC16F628A [3].

Todos estos trabajos desarrollados permiten tener un punto de partida para el desarrollo de este proyecto considerando que ninguno de ellos establece el desarrollo de un PID difuso.

#### 1.4.2 Comparación de controladores.

La comparación entre PID tradicionales y difusos con otros tipos de controladores, ha sido también la tarea de muchos investigadores buscando encontrar diferencias en el tipo de control, tiempo de respuesta y efectividad en el proceso aplicado.

Dentro de este contexto se encuentran trabajos tales como:

El trabajo de tesis titulado **“Diseño y análisis comparativo de un sistema de control de temperatura con lógica difusa vs control proporcional integral derivativo (PID) en un prototipo de una torre de enfriamiento en una planta de procesos”** para controlar la temperatura de la glicerina, utilizo una tarjeta de adquisición de datos DAQ-USB 6009, y mediante LabView se desarrolla un sistema de monitoreo de temperaturas de fluidos y control de bombas, válvulas y ventilador, con los datos de velocidad del ventilador (entrada) y temperatura de la glicerina (salida) con un periodo de muestreo 0.25 segundos, mediante la aplicación Ident de Matlab obtuvieron la función de transferencia con el modelo Box Jenkin con 99% de confianza y 81.71% de similitud a la planta original, desarrollan un control difuso con entradas (error (e) - derivada del error ( $\Delta e$ )) con 25 reglas, estableciendo dos simulaciones en Simulink-Matlab para PID no iterativo y Control difuso, obteniendo un tiempo de estabilización con el control difuso menor en 250 segundos, porcentaje de error máximo mayor en el controlador difuso, la rapidez de cambio es mucho mayor en el PID, error medio y valor medio de la señal de control similares [4].

El trabajo “**Diseño e implementación de un PID profesional y un PID difuso utilizando un microcontrolador PIC18F4550**” contempla el desarrollo de dos controladores, programados en un microcontrolador en lenguaje C para variar la velocidad de un motor de corriente directa, el denominado PID profesional que trabaja en forma paralela que permite analizar las acciones de control por separado para determinar el efecto que ejerce sobre el proceso, con un intervalo de tiempo de 786.2us entre dos acciones sucesivas de control y un tiempo de establecimiento de 0.1094s. Mientras que el PID difuso está constituido por dos controladores difusos un PD (Proporcional-Derivativo) y PI (Proporcional-Integral) cada uno constituido por 25 reglas priorizando la rapidez y el porcentaje de sobreimpulso, además el intervalo de tiempo de 42.3ms entre dos acciones de control sucesivas y un tiempo de establecimiento de 0.8324s [5].

El trabajo “**Comparación del control de velocidad de un motor de DC utilizando lógica difusa vs PID tradicional**” publicada en la revista CINTEX, desarrollo un control difuso en un microcontrolador 6808GP32 de Motorola con dos entradas (set point y error) con nueve reglas para la fuzzyficación mientras que para la salida se estableció tres conjuntos difusos para establecer valores adecuados de PWM (Modulación por ancho de pulso) con aproximación por centroide, los resultados obtenidos con sobreimpulso de 0.49% con velocidades superiores a 806 rpm y comportamiento aproximadamente lineal, a 500 rpm existe un sobreimpulso 4%, a 130 rpm un sobreimpulso de 27.69% y un error crítico, todo esto proceso demora aproximadamente 3.93 milisegundos tiempo aceptable para un proceso industrial y comportamiento similar a un controlador tradicional [6].

El trabajo “**Comparación de desempeño entre controladores fuzzy y controladores proporcionales**” sobre un tanque con descarga gravitatoria con caudal de entrada constante, mediante simulink de Matlab se desarrollaron dos controladores Proporcional y fuzzy con tres y siete reglas, mediante un conmutador se puede activar cualquier controlador,



el set point es variable desde un nivel de 1m a 2m .Las pruebas realizadas a una entrada escalón, determinan que el control proporcional es más rápido para alcanzar el 63% valor final prefijado en la referencia, mientras que el control fuzzy generan menos, oscilaciones, armónicos y sobre picos de menor amplitud. [7].

Con los trabajos y artículos mencionados se pretende realizar un regulador AC, específicamente el control por ángulo de fase que permite controlar la potencia suministrada al sistema de transferencia de calor (ducha eléctrica) y con un algoritmo de PID digital y difuso para controlar la temperatura de manera más eficiente y posible ahorro de energía eléctrica.

## CAPÍTULO 2

### 2. MARCO TEÓRICO.

#### 2.1 Trasterencia de Calor.

La transferencia de calor es la energía calorífica que se transfiere de un sistema de mayor temperatura al de menor temperatura hasta llegara a una condición de equilibrio, la rapidez con la que se realiza es motivo de estudio de los tres tipos de transferencia de calor conocidos como: conducción, convección, radiación.

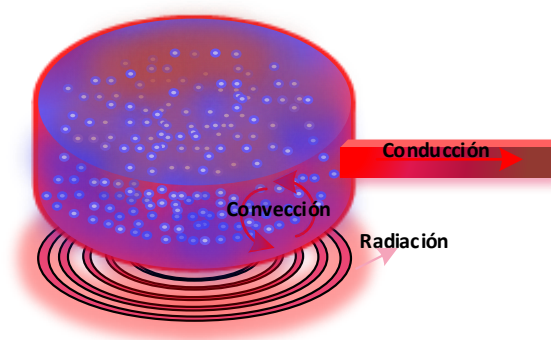


Figura 2.1: Transferencia de calor.

##### 2.1.1 Conducción.

Es la trasterencia de energía entre partículas adyacentes y puede tener lugar en sólidos, líquidos y gaseosos. En líquidos y gases la conducción se produce por choques y difusión de las moléculas, en solidos se produce por la vibración de las moléculas mientras los electrones libres transportan la energía, el modelo que mejor describe todos estos fenómenos es la ecuación 2.1 ley de Fourier [8].

$$\dot{Q} = -kA \frac{dT}{dx} \quad (2.1)$$

Donde:

$\dot{Q}$ : Transferencia de calor por conducción [W]

k: Conductividad térmica [ $\frac{W}{m \cdot ^\circ C}$ ]

$\frac{dT}{dx}$ : Gradiente de temperatura

A: Área de transferencia de calor [ $m^2$ ]

### 2.1.2 Convección.

La transferencia de energía entre dos zonas mediante dos efectos combinados (conducción – movimiento de fluidos), esta puede ser: Convección forzada donde el fluido es forzado a circular mediante medios externos tales como bomba, ventilador etc. Convección natural producida por la diferencia de densidad debido al cambio de temperatura produce el movimiento del fluido. La ecuación 2.2 muestra la ley de Newton del enfriamiento que mejor describe el comportamiento de este fenómeno [8].

$$\dot{Q} = hA_S(T_S - T_\alpha) \quad (2.2)$$

Donde:

$\dot{Q}$ : Transferencia de calor por convección [W]

h: Coeficiente de convección [ $\frac{W}{m^2 \cdot ^\circ C}$ ]

$A_S$ : Área superficial [ $m^2$ ]

$T_S$ : Temperatura de la superficie [ $^\circ C$ ]

$T_\alpha$ : Temperatura alejada de la superficie [ $^\circ C$ ]

### 2.1.3 Radiación.

La radiación térmica es energía en forma de ondas electromagnéticas emitida por un cuerpo debido a la temperatura, producida por cambios en las configuraciones electromagnéticas de átomos consecutivos. La ley de Stefan-Boltzmann que se muestra en la ecuación 2.3 describe cuando un cuerpo absorbe y refleja la radiación [8].

$$\dot{Q} = \varepsilon \sigma A_S T_S^4 \quad (2.3)$$

Donde:

$\dot{Q}$ : Transferencia de calor por radiación [W]

$\varepsilon$ : Emisividad del cuerpo ( $0 \leq \varepsilon \leq 1$ )

$\sigma$ : Constante de Stefan-Boltzmann [ $\frac{W}{m^2 \cdot K^4}$ ]

A: Área superficial radiante [ $m^2$ ]

$T_s$ : Temperatura absoluta en el área del cuerpo [ $^{\circ}K$ ]

## 2.2 Conversores AC-AC.

Los convertidores AC-AC permiten controlar el voltaje, corriente y potencia que la fuente entrega a la carga mediante la conmutación de tiristores tomando como base el circuito de la figura 2.2. Las aplicaciones más comunes son: control de intensidad luminosa, temperatura y velocidad de motores de inducción.

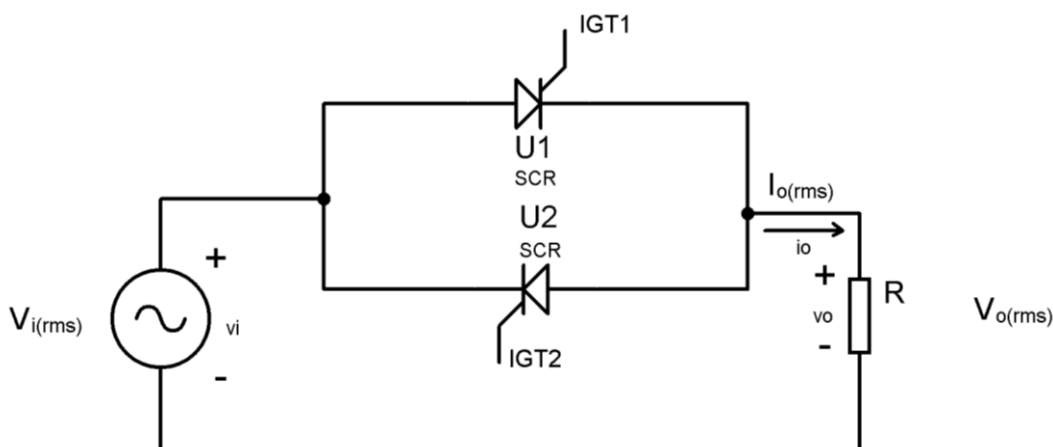


Figura 2.2: Circuito de convertidor AC-AC.

### 2.2.1 Control on-off.

El principio de funcionamiento es la activación por un tiempo  $n$  y desactivación en un tiempo  $m$  de ciclos completos de la señal sinusoidal de entrada, mediante los tiristores U1, U2 con señal  $I_{GT1}$ ,  $I_{GT2}$  respectivamente de forma alternada como se muestra la figura 2.3, el apagado de los tiristores se realiza por conmutación natural [9].

El voltaje de salida en la carga está dado por la ecuación 2.4.

$$V_{o(rms)} = \sqrt{\frac{1}{2\pi} \frac{n}{n+m} \int_0^{2\pi} (\sqrt{2} V_{i(rms)} \sin(\omega t))^2 d(\omega t)} = V_{i(rms)} \sqrt{\frac{n}{n+m}} \quad (2.4)$$

Donde:

$V_{i(rms)}$ : Valor eficaz o rms de voltaje de entrada.

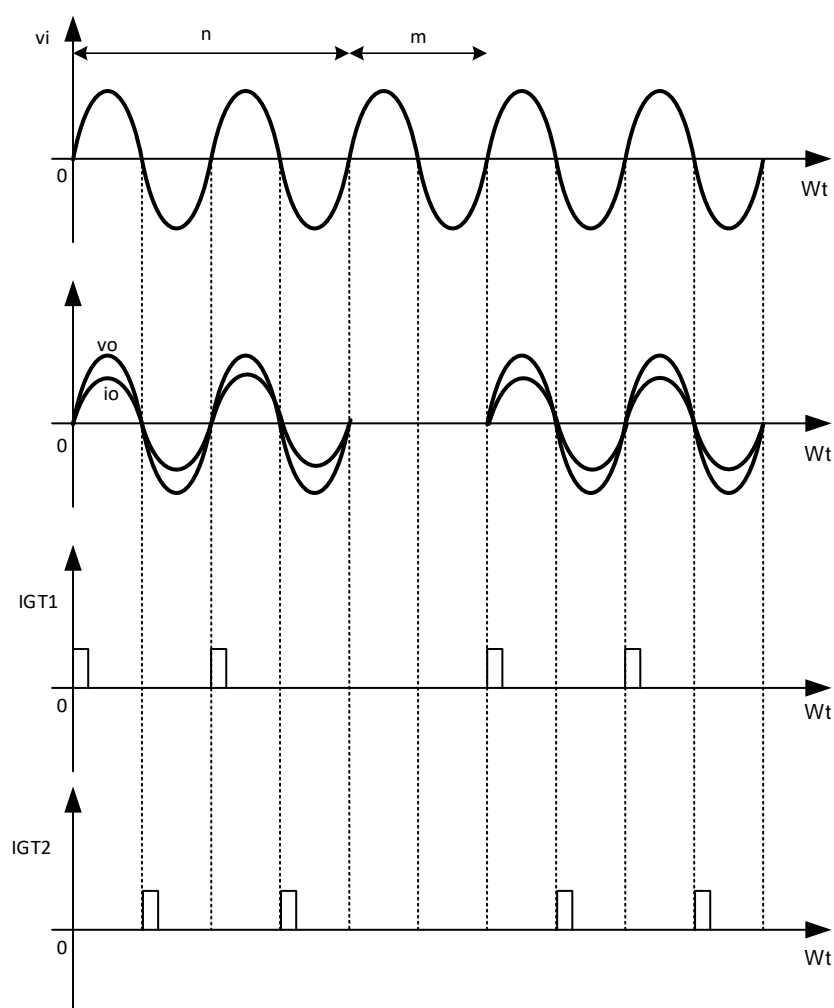
$V_{o(rms)}$ : Valor eficaz o rms de voltaje de salida sobre la carga resistiva

$n$ : tiempo de activación de los tiristores

$m$ : tiempo de desactivación de los tiristores

Además se considera que como ciclo de trabajo a  $C_T$  y puede variar entre 0 y 1 como se muestra en la ecuación 2.5.

$$C_T = \frac{n}{n+m} \quad (2.5)$$



**Figura 2.3: Control on-off.**

Para cargas resistivas el valor eficaz o rms de corriente está dado por la ecuación 2.6:

$$I_{o(rms)} = \frac{V_{o(rms)}}{R} \quad (2.6)$$

Donde:

$I_{o(rms)}$ : Valor eficaz o rms de corriente en la carga

$R$ : Resistencia de la carga

Las aplicaciones más comunes de alta constante térmica e inercia mecánica como son calefacción industrial y control de velocidad de motores respectivamente.

### 2.2.2 Control por ángulo de fase.

El controlador por ángulo de fase permite controlar la potencia en el semiciclo positivo y negativo mediante un ángulo de disparo  $\alpha$ , tomado desde el cruce por cero de cada ciclo, es decir es tiristor U1, se dispara con ángulo  $\omega t = \alpha$  con señal  $I_{GT1}$ , mientras que el tiristor U2 se dispara con ángulo  $\omega t = \pi + \alpha$  con señal  $I_{GT2}$  como se muestra en la figura 2.4 las formas de onda [9].

El voltaje salido sobre una carga resistiva se muestra en la ecuación 2.7.

$$V_{o(rms)} = \sqrt{\frac{1}{2\pi} \int_{\alpha}^{2\pi} (\sqrt{2} V_{i(rms)} \sin(\omega t))^2 d(\omega t)} = V_{i(rms)} \sqrt{\frac{1}{\pi} (\pi - \alpha + \frac{\sin(2\alpha)}{2})} \quad (2.7)$$

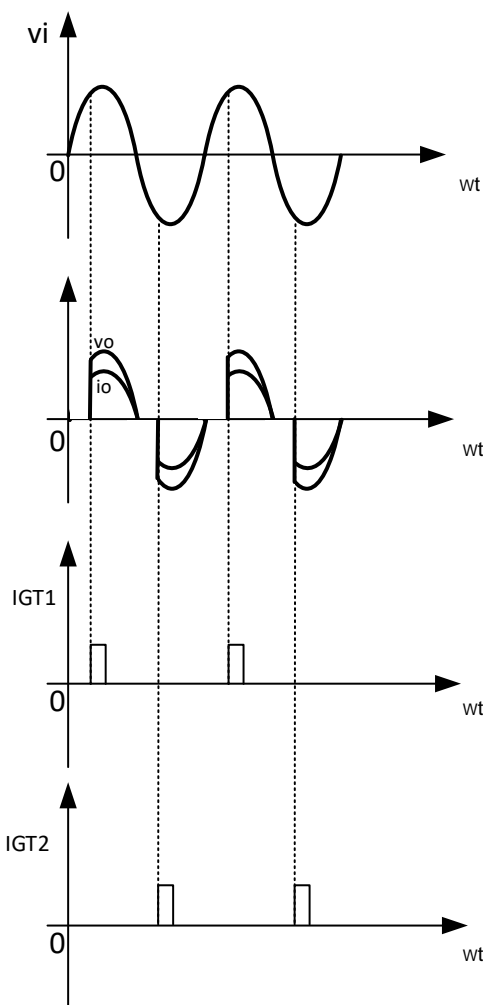
Donde  $\alpha$  esta entre 0 y  $\pi$  y la corriente se puede determinar por la ecuación 2.6, mientras que potencia de entrada ( $P_i$ ) y de salida ( $P_o$ ) se muestra en la Ecuaciones 2.8 y 2.9 respectivamente.

$$P_i = V_{i(rms)} I_{i(rms)} = V_{i(rms)} I_{o(rms)} \quad (2.8)$$

$$P_o = V_{o(rms)} I_{o(rms)} = I_{o(rms)}^2 R \quad (2.9)$$

El factor de potencia (FP) se determina por la ecuación 2.10.

$$FP = \frac{P_o}{P_i} \quad (2.10)$$



**Figura 2.4: Control por ángulo de fase.**

### 2.3 Sistemas de Control utilizados en intercambiadores de calor.

Los distintos sistemas de control utilizados en los intercambiadores de calor el de uso más común son los controladores PID. Dentro de este gran contexto el PID analógico que se muestra en la ecuación 2.11, se ha utilizado durante mucho tiempo y ha permitido el desarrollo de los PID discreto y difuso.

$$m(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt}] \quad (2.11)$$

Donde:

$m(t)$ : Salida del controlador

$e(t)$ : Error

$K$ : Ganancia de controlador

$T_i$ : Tiempo Integral

$T_d$ : Tiempo diferencial

### 2.3.1 Controlador PID discreto.

Los controladores PID discreto o digital se obtiene al digitalizar la ecuación 2.11 aproximando la componente integral a una sumatoria trapezoidal y en diferencia de puntos al término derivativo, obteniendo dos representaciones:

- Forma Posicional
- Forma de Velocidad

La Ecuación 2.12 muestra la ecuación del controlador PID en su forma posicional como muestra en la figura 2.5 (a).

$$M(z) = [K_P + \frac{K_I}{1-z^{-1}} + K_D(1 - z^{-1})]E(z) \quad (2.12)$$

Mientras que la ecuación 2.13 muestra el controlador PID en su forma de velocidad como muestra en la figura 2.5 (b), computacionalmente más eficiente y no necesita especificación del término en estado estable.

$$M(z) = -K_P C(z) + K_I \frac{R(z) - C(z)}{1-z^{-1}} + K_D(1 - z^{-1})C(z) \quad (2.13)$$

Las ganancias que necesitan los PID de posición y velocidad se calculan mediante las ecuaciones  $K_P = K$ ;  $K_I = K \frac{T}{T_i}$ ;  $K_D = K \frac{T_d}{T}$  y se pueden considerar  $K_P = K - \frac{K_I}{2}$  para mayor exactitud [10].

Donde:

$T$ : Periodo de muestreo

$K_P$ : Ganancia proporcional

$K_I$ : Ganancia integral

$K_D$ : Ganancia derivativa



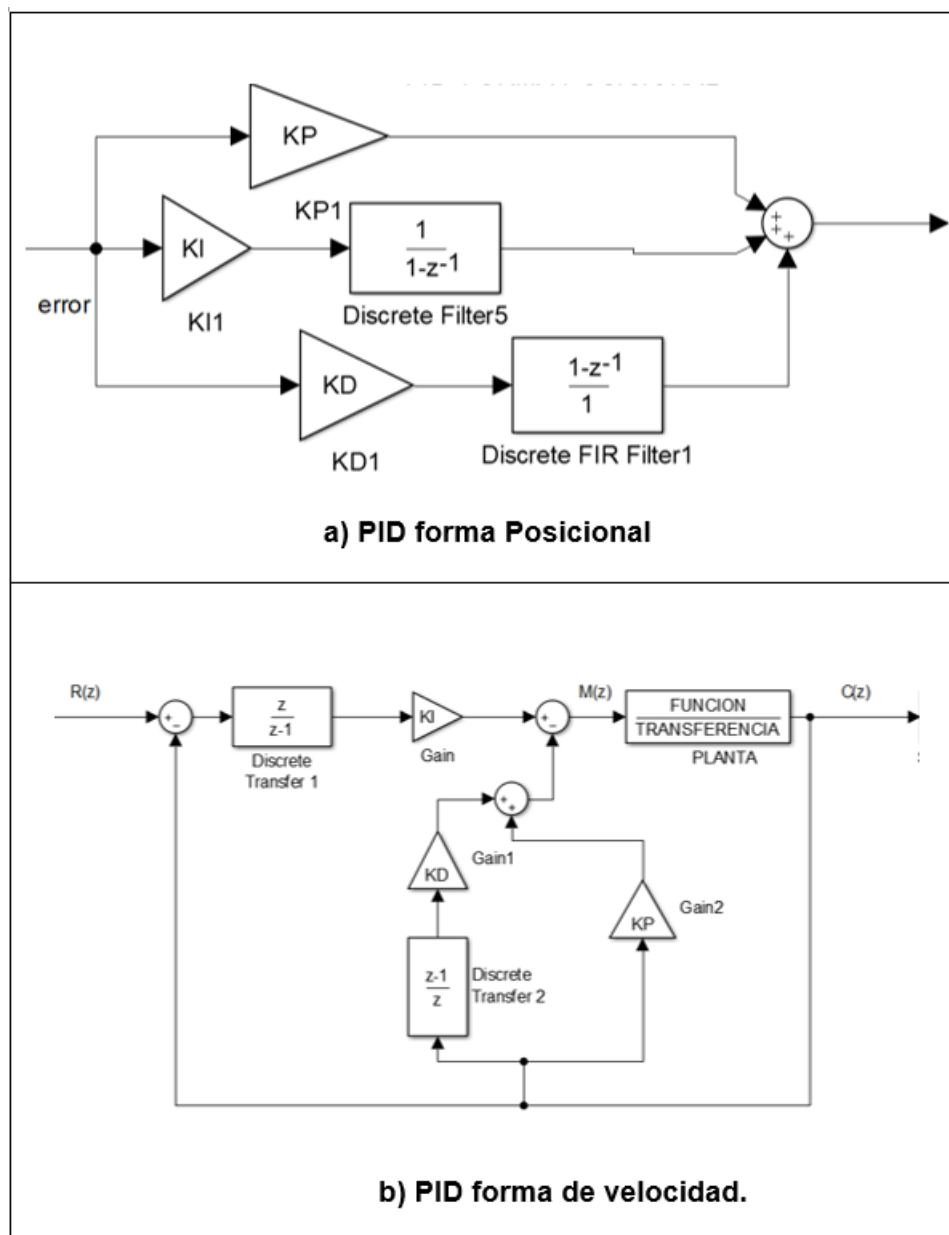
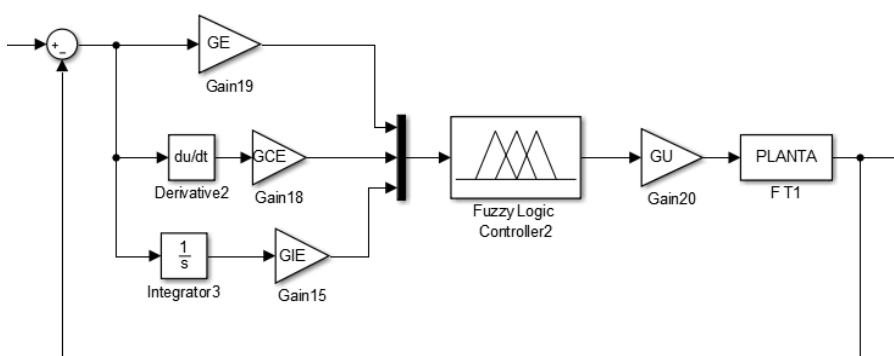


Figura 2.5: Diagrama de bloques control a) PID Posicional, b) PID de velocidad.

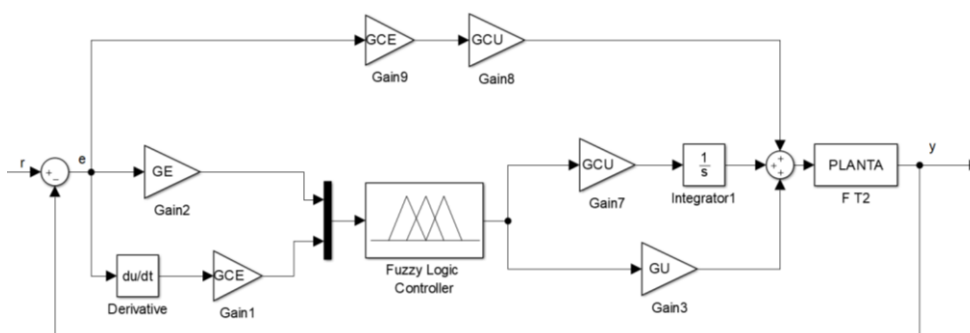
### 2.3.2 Controlador PID difuso.

El diseño de controladores PID difusos siguen los mismos experimentos para ajustar PID lineales sin modelar o utilizando modelos que responden a una entrada escalón, basados en este enfoque cualquier PID puede reproducirse en un sistema difuso.

Los PID difuso se pueden realizar de distintas maneras, uno de ellos es el que se muestra en la figura 2.6 que tiene tres entradas, error, derivada del error y la integral de error, con un sistema de cinco reglas de membresía producirá 125 reglas provocando que el ajuste sea difícil. Por esta razón se propuso dividir al sistema en dos controladores difusos que tiene dos entradas  $e$ =error,  $\Delta e$ =cambio del error, una salida  $u$  en el caso de un controlador que actúa como PD (Proporcional Derivativo), y un  $\Delta u$  en el caso del controlador actúa como PI (Proporcional Integral) y al final consolidar en un controlador PID difuso que se muestra en la figura 2.7 que permite disminuir el número de reglas de control [11].



**Figura 2.6: PID difuso con tres entradas.**



**Figura 2.7: PID difuso con PI + PD.**

Las constantes que se muestran en las figuras 2.6 y 2.7 son: GE: Ganancia de error, GU: Ganancia de salida, GIE: Ganancia integral del error, GCE: Ganancia de control del error, GCU: Ganancia de control de salida. Para mejorar el controlador difuso se realiza el cambio de  $\frac{de(t)}{dt}$  por

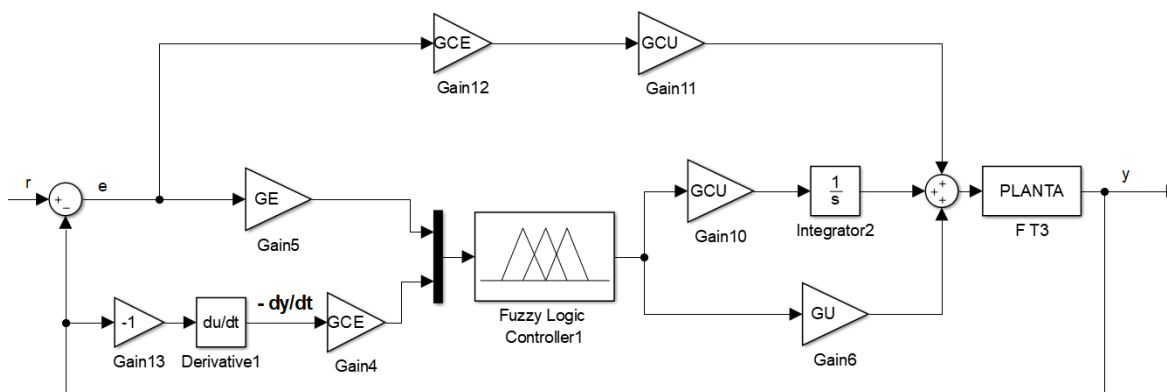
la  $-\frac{dy(t)}{dt}$  como se muestra en la figura 2.8, que permite mejorar la respuesta del sistema de control al permitir que la acción derivativa actúe directamente sobre la salida  $y(t)$  [12].

Para determinar los valores del PID difuso se toma los valores obtenidos del PID tradicional a partir de las siguientes ecuaciones:

$$Kp = GCU * GCE + GU * GE \quad (2.14)$$

$$Ki = GCU * GE \quad (2.15)$$

$$Kd = GU * GCE \quad (2.16)$$



**Figura 2.8: PID difuso con PI + PD con cambio  $\frac{de(t)}{dt}$  por  $-\frac{dy(t)}{dt}$ .**

A partir de las ecuaciones 2.14, 2.15 y 2.16 con un valor máximo de entrada a la planta que permita corregir el error se establece un valor constante para  $GE$  que permite obtener las siguientes ecuaciones:

$$GCE = \frac{GE(Kp - \sqrt{Kp^2 - 4KiKd})}{2Ki} \quad (2.17)$$

$$GCU = \frac{Ki}{GE} \quad (2.18)$$

$$GU = \frac{Kd}{GCE} \quad (2.19)$$

### Reglas difusas.

Para el establecimiento de las reglas difusas, primero se debe establecer **variables lingüísticas** de entrada ( $e, \Delta e$ ) y salida ( $u, \Delta u$ ) necesarias para

el control difuso, en donde se establecen los siguientes **términos lingüísticos**: N= Negativo, Z= Cero, P= Positivo, para la entrada y para la salida se establecieron: MIN= Mínimo, ZER= Cero, MAX= Máximo, los cuales están asociados a **conjuntos difusos** dentro del **universo discurso** de entrada y salida.

Mediante la respuesta a una señal escalón como muestra en la figura 2.9 es posible establecer la matriz de reglas lingüísticas como muestra la tabla 1, con un total de 9 reglas que sirven para un control PD y PI.

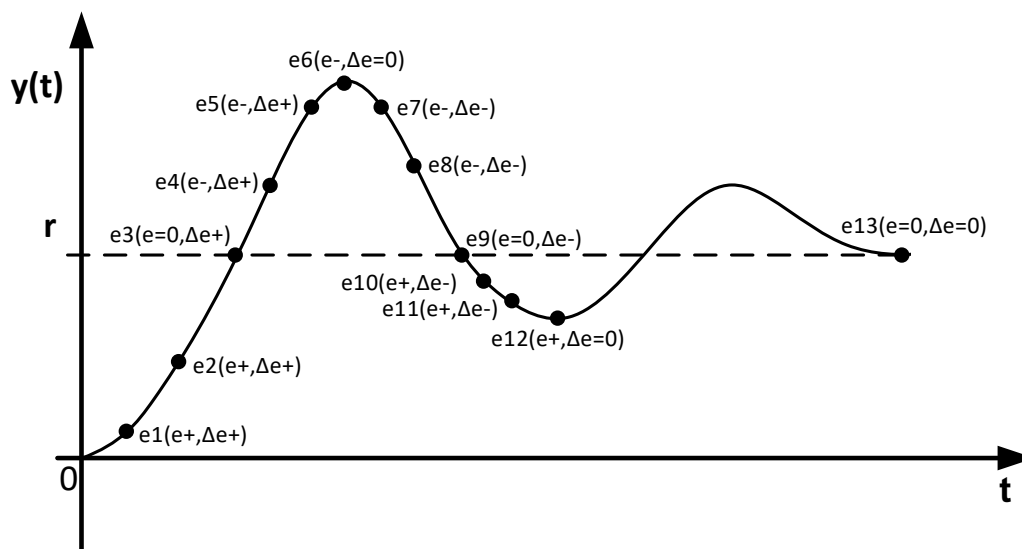


Figura 2.9: Respuesta a una entrada escalón.

|       |   | Δerror |     |     |
|-------|---|--------|-----|-----|
|       |   | N      | Z   | P   |
| error | N | MIN    | MIN | MIN |
|       | Z | MIN    | ZER | MAX |
|       | P | MAX    | MAX | MAX |

Tabla 1: Matriz de 9 reglas difusas

También se pueden establecer 25 reglas lingüísticas con los siguientes términos lingüísticos: NG= Negativo Grande, NM= Negativo Medio, ZE= Cero, PM= Positivo Medio, PG= Positivo grande, para la entrada y la salida como muestra la tabla 2.

|       |    | $\Delta$ error |    |    |    |    |
|-------|----|----------------|----|----|----|----|
|       |    | NG             | NM | ZE | PM | PG |
| error | NG | NG             | NG | NG | NM | NG |
|       | NM | NM             | NM | NM | NM | NM |
|       | ZE | ZE             | ZE | ZE | ZE | ZE |
|       | PM | PM             | PM | PM | PM | PM |
|       | PG | PG             | PG | PG | PG | PG |

**Tabla 2: Matriz de 25 reglas difusas**

## CAPÍTULO 3

### 3. DISEÑO DEL SISTEMA DE CONTROL EMBEBIDO.

El sistema de control embebido está constituido por el subsistema de adquisición de datos y por el subsistema de control de temperatura de agua para la ducha eléctrica. Para el diseño se considera el diagrama de bloques del sistema de control embebido que se muestra en la figura 3.1 y está constituido por:

- STM32F4Discovery
- LM35
- ATmega328P
- Interfaz de potencia
- Detector de cruce por cero
- Modulo Bluetooth
- Ducha Eléctrica

Las características principales de la tarjeta STM32F4Discovery se encuentra en el **anexo 1**, mientras que la programación se desarrolla en Matlab-Simulink con el paquete de librerías de Waijung que se encuentra en el **anexo 2**, mientras la característica técnica de la ducha eléctrica se encuentra en el **anexo 3**.

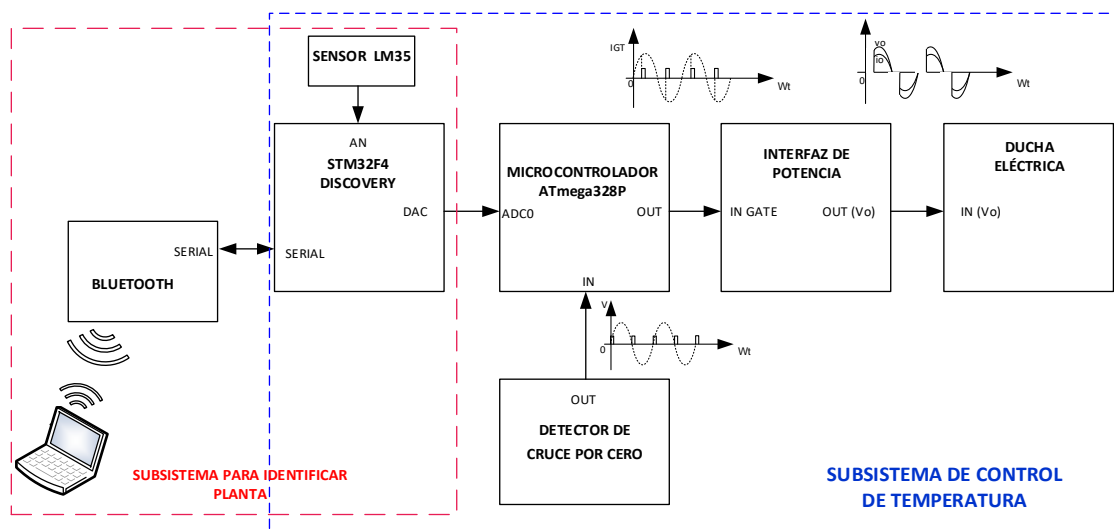


Figura 3.1: Diagrama de bloques del sistema de control embebido.

El circuito total del sistema embebido con las conexiones entre los distintos dispositivos se muestra en el **anexo 4**.

### 3.1 Diseño del control por ángulo de fase.

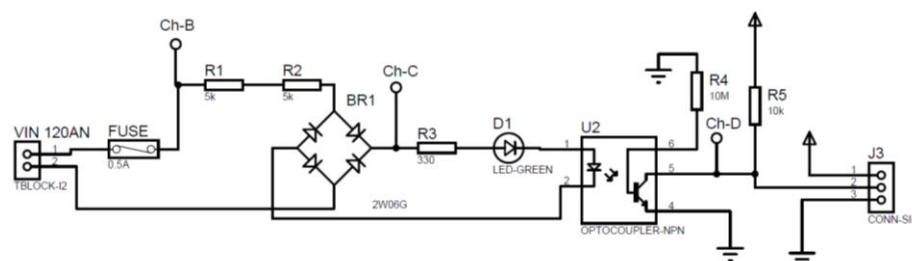
El control por ángulo de fase permite controlar la potencia de la resistencia eléctrica que calienta el agua variando el voltaje de corriente alterna y está constituida por los siguientes circuitos:

- Detector de cruce por cero
- Interfaz de potencia de corriente alterna
- Control embebido

#### 3.1.1 Circuito detector de cruce por cero.

El circuito detector de cruce por cero que muestra la figura 3.2, envía una señal pulso cuando detecta que la señal sinusoidal de la red eléctrica cruza por cero y tiene los siguientes elementos:

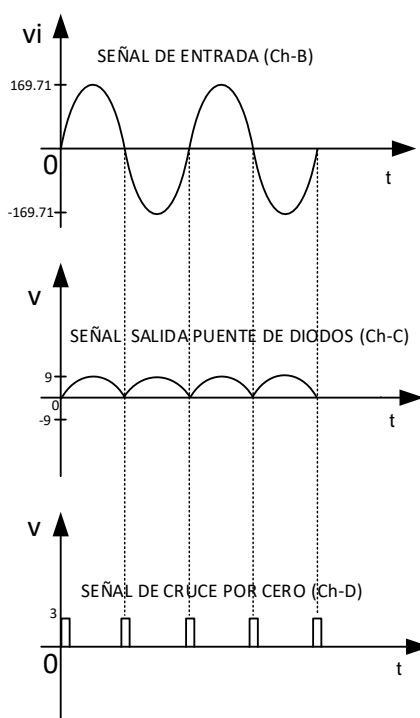
- 1 Fusible de 0.5A
- 2 Resistencias (R1-R2) de  $5K\Omega$  a 5W
- 1 Resistencia (R5) de  $10K\Omega$  a  $\frac{1}{2}W$
- 1 Resistencia (R3) de  $330\Omega$  a  $\frac{1}{2}W$
- 1 Resistencia (R4) de  $1M\Omega$  a  $\frac{1}{2}W$
- 1 Puente de Diodos 2W06G
- 1 Led (D1) de 5mm Rojo
- 1 Optoacoplador- NPN (4N25)
- 1 Bornera de 2
- 1 Molex de 3



**Figura 3.2: Circuito detector de cruce por cero.**

### Funcionamiento.

El circuito de la figura 3.2, toma un voltaje de corriente alterna de 120Vrms o voltaje pico de 169.71V (Ch-B), mediante las resistencias R1, R2 de 5K $\Omega$  se reduce el voltaje y a través del puente de diodos se rectifica la señal, obteniendo aproximadamente un voltaje pico de 9V (Ch-C), para que ingrese al pin-1 del Optoacoplador a través de resistencia de R3 (330 $\Omega$ ) que limita la corriente aproximadamente a 15.7mA como se demuestra en la ecuación 3.1 y 3.2, con los datos técnicos de cada dispositivo electrónico. A la salida del optoacoplador en el pin-5 se conecta una resistencia de R5 de 10K que permite entrar en corte y saturación el fototransistor interno del optoacoplador para obtener una señal cuadrada entre 0 y 5 V<sub>DC</sub> (Ch-D), que ingresa al pin de interrupción externa (INT0) del microcontrolador ATmega328P encargado de accionar el interfaz de potencia. Las señales del cruce por cero se observan en la figura 3.3 y el circuito impreso se muestra en el **anexo 5**.



**Figura 3.3: Señales de detector de cruce por cero.**



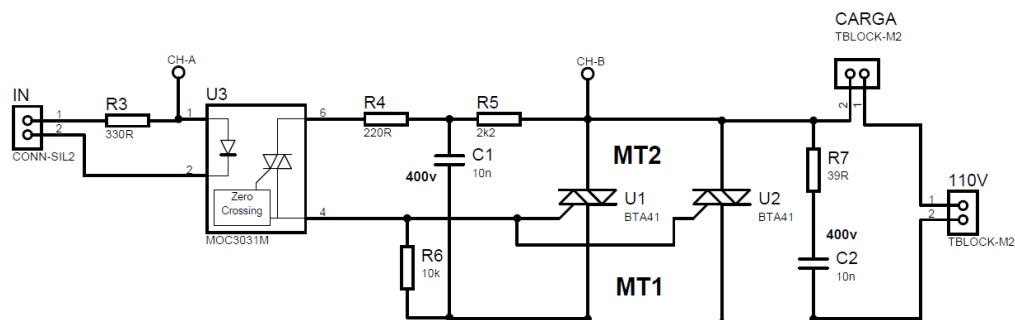
$$I_{Fmax} = \frac{V_{AC} - V_{D1} - V_{LED} - V_{Fmax} - V_{D2}}{R_1 + R_2 + R_3} \quad (3.1)$$

$$I_{Fmax} = \frac{169.71V - 0.7V - 2.2V - 1.5V - 0.7V}{5K\Omega + 5K\Omega + 330\Omega} = 15.93mA \quad (3.2)$$

### 3.1.2 Circuito Interfaz de potencia de corriente alterna.

El circuito interfaz de potencia corriente alterna que se muestra en la figura 3.4 encargada de recibir la señal de activación del ATmega328P que permite controlar el voltaje aplicado a la resistencia eléctrica tiene los siguientes elementos:

- 1 Resistencia (R3) de 330Ω a ½ w
- 1 Resistencia (R4) de 220Ω a ½ w
- 1 Resistencia (R5) de 2.2KΩ a ½ w
- 1 Resistencia (R6) de 10KΩ a ½ w
- 1 Resistencia (R7) de 36Ω a ½ w
- 2 Capacitor de 10nF a 400V
- 2 BTA41 a 40A/600V
- 1 MOC3021
- 1 Molex de 2

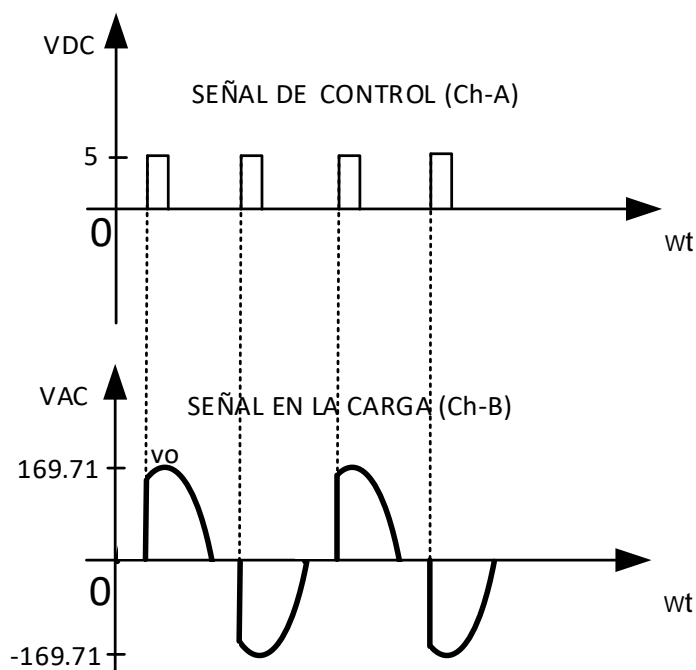


**Figura 3.4: Circuito Interfaz de potencia AC.**

#### Funcionamiento.

El circuito de la figura 3.4 permite recibir una señal cuadrada (Ch-A), enviada por el ATmega328P que a través de la resistencia R3 (220Ω) activa el fotodiodo interno que a su vez activa el triac interno del MOC3021, y mediante las resistencias R4, R5, R6 que activan los Triacs

permiten controlar el voltaje de la carga (Ch-B). El circuito RC (red snubber) conformado por la resistencia R7 ( $39\Omega$ ) y C2 ( $10\text{nF}$ ) permite evitar variaciones bruscas de voltaje en los extremos del triac. Las señales del Ch-A y Ch-B se observa en la figura 3.5, y los circuitos impresos se muestran en el **anexo 5**.



**Figura 3.5: Señales de control y de carga.**

### 3.1.3 Circuito de control embebido.

Este circuito tiene como principal componente el STM32F4Discovery (STM32F407VGT6) que puede ser configurado para funcionar como “adquisición de datos” y “control de potencia de la carga”, que se muestra en la figura 3.6 y tiene los siguientes elementos.

- 1 Tarjeta STM32F4-Discovery
- 1 sensor de temperatura LM35
- 1 Potenciómetro de  $10\text{K}\Omega$
- 1 LCD 16X2



potencia para controlar el voltaje aplicado a la resistencia eléctrica, mientras que el LCD muestra la temperatura de referencia y del agua.

Cabe señalar que el circuito cruce por cero está conectada al pin D2/PD2/INT0 del ATmega328P para sincronizar el pulso de disparo que sale por el pin D10/PB2 del ATmega328p al circuito de potencia, para controlar el voltaje aplicado a la carga a través de los triacs.

### **Descripción del programa para el disparo de los triacs con el ATmega328P.**

El diagrama de flujo del programa del ATmega328P que se muestra en el anexo 6 indica la configuración de los siguientes parámetros:

- PB2/D10 como salida que activa el tiristor.
- El ADC0 con voltaje de referencia igual al voltaje de alimentación del microcontrolador, justificación a la derecha de los registros ADH, ADL, y frecuencia de trabajo de 125 KHz.
- Interrupción externa INT0 con flanco de subida.

Con los parámetros establecidos el programa realiza la lectura del registro ADC asignando el valor a la variable (*valorADC*) para realizar el cálculo del ángulo de disparo a través de la ecuación 3.3.

$$\mathit{angulodisparo} = 8333 - 12 * \mathit{valorADC} \quad (3.3)$$

Cuando se produce la interrupción INT0 se ejecuta una demora en microsegundos con el valor del ángulo de disparo (*angulodisparo*) y posteriormente se envía un pulso de 20 microsegundos por el pin PB2/D10, que permite activar el tiristor. El diagrama de flujo y programa desarrollado en Atmel Studio 7.0 se muestra en el **anexo 6**.

### **3.2 Diseño del controlador.**

Para realizar el diseño del controlador, primero se identifica la función de transferencia mediante Matlab–Ident con los datos guardados en Excel que se obtiene a través del programa de adquisición de datos desarrollado en LabView, para posteriormente desarrollar el PID discreto y difuso.

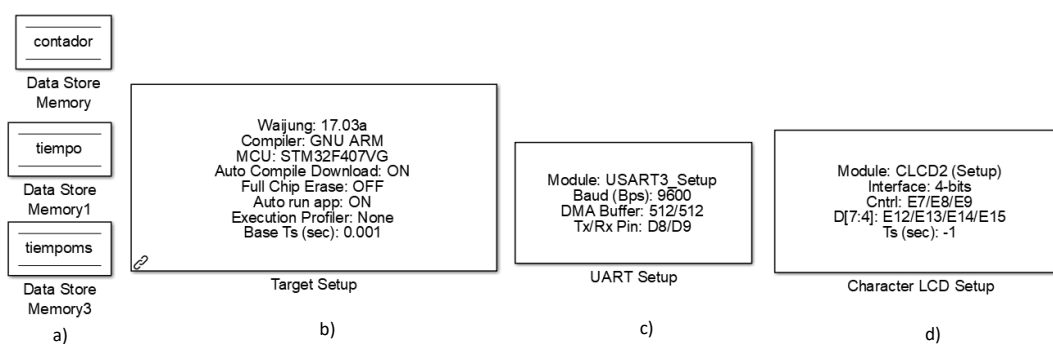
### 3.2.1 Identificación de la planta.

Para identificar la planta se realiza a través del programa captura de datos desarrollado en Simulink, grabado en la tarjeta STM32F4Discovery y otra aplicación desarrollada en LabView que permite guardar los datos en un archivo de Excel.

#### Descripción del programa para la adquisición de datos en Simulink.

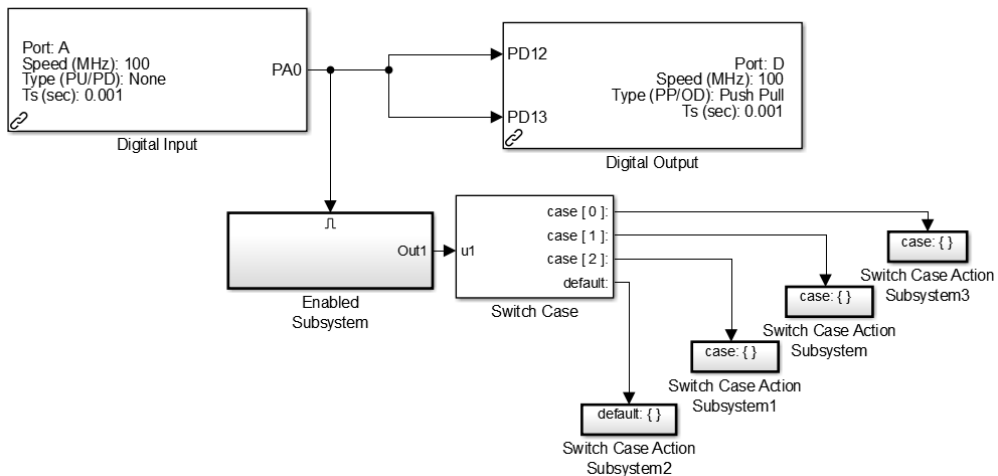
El diagrama de flujo y programa que se muestra en el **anexo 7**, establece una descripción del programa para la adquisición de datos desarrollado en Matlab-Simulink mediante la librería Waijung Blockset.

El programa inicia configurando el bloque “Target Setup” con el compilador GNU ARM, el microcontrolador STM32F407VG, frecuencia de reloj de 8Mhz, ciclo de trabajo de 0.001 segundos para los bloques considerados parámetros importantes, el resto de componentes no se modifica, además “UART Setup” y “Character LCD Setup” se configura como muestra la figura 3.7 y también se declaran en memoria las variables (contador, tiempo, tiempoms).



**Figura 3.7: a) Variables; Configuraciones b) tarjeta, c) UART, d) LCD.**

La variable (contador) permite elegir distintas opciones mediante el “switch case” como muestra la figura 3.8, para realizar las siguientes opciones:



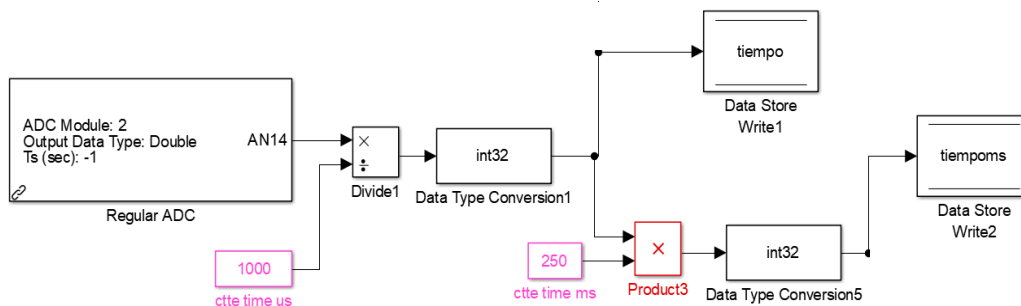
**Figura 3.8: Switch case de la variable contador.**

Caso 0: En este caso el mensaje “CAPTURA DATOS” que se muestra en el LCD.

Caso 1: La señal analógica en AN14 permite establecer el orden mediante la variable (tiempo) con valores enteros de 0, 1, 2, 3, 4 calculados mediante la ecuación 3.4 y establecer valores enteros para el período de muestreo entre 100, 250, 500, 750, 1000 milisegundos mediante la ecuación 3.5 y almacenarla en la variable (tiempoms). El diagrama de bloques se muestra en la figura 3.9.

$$tiempo = \frac{ADC}{1000} \quad (3.4)$$

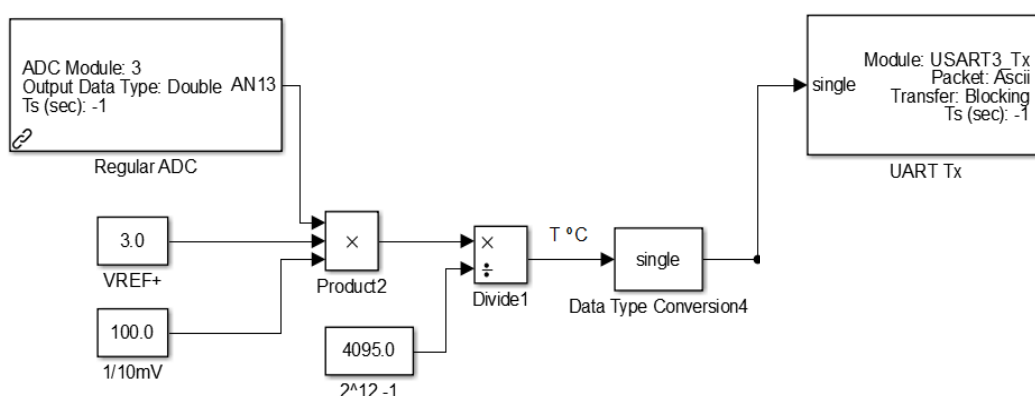
$$tiempoms = \frac{ADC * 250}{100} \quad (3.5)$$



**Figura 3.9: Cálculo de variables (tiempo y tiempoms).**

Caso 2: Permite utilizar el orden y tiempo de muestreo establecido anteriormente mediante un “switch case” para poder recolectar valores entre 0 - 4095 a través AN13 para transformarlo en valores de temperatura mediante la ecuación 3.6, para ser enviados a la computadora a través del puerto serial PD8/TX que está conectado al módulo bluetooth como muestra la figura 3.10.

$$TEMP = \frac{ADC\_LM35 * 3 * 100}{4095} \quad (3.6)$$

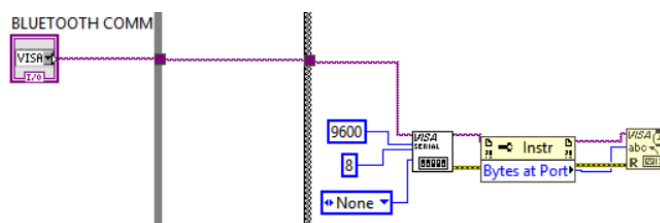


**Figura 3.10: Cálculo de temperatura.**

Default: Aquí se saca el mensaje en el LCD “FIN CAPTURA DATOS” y la variable (contador) se le asigna cero, para permitir que el programa pueda iniciar otra captura de datos.

### Descripción del programa para la adquisición de datos en LabView.

El programa que se muestra en el **anexo 8** permite recibir los datos enviados por el sistema embebido mediante la conexión Bluetooth-serial configurado a 9600 baudios como se puede observar en la figura 3.11.

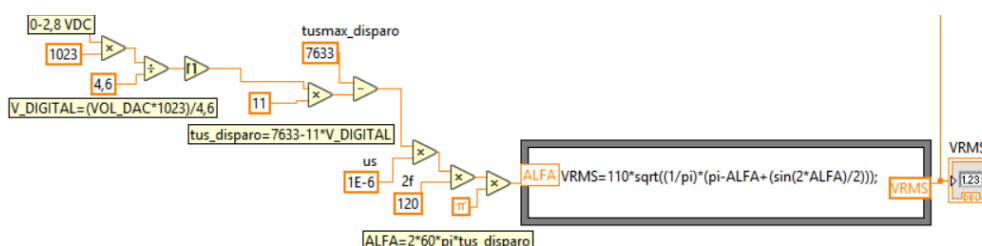


**Figura 3.11: Configuración serial.**

Mediante la siguiente trama (**R%1.3fT%1.3fV%1.3fE%1.3f**) recibida, se establece una letra que determina el tipo de dato al que pertenece (R=valor de referencia, T=valor de temperatura, V=voltaje, E=Error) los que permiten al programa separarlos y establecer sus valores, pero el valor de referencia y error es cero (R=0, E=0) debido a que el sistema embebido está programado en modo captura de datos, mientras que el valor de voltaje recibido entre  $0-2.8V_{DC}$  es transformada a ángulo de disparo, mediante las ecuaciones desarrolladas en LabView como muestra la figura 3.12 considerando las ecuaciones 3.7, 3.8 y la ecuación 2.7 mencionado anteriormente que permite determinar el valor de voltaje VRMS para ser guardados en el archivo de Excel junto con los otros datos.

$$t_{us} = \left( 7633 - 11 * \frac{VOL_{DAC} * 1023}{4.6} \right) * 10^{-6} \quad (3.7)$$

$$\alpha = 2 * 60 * \pi * t_{us} \quad (3.8)$$



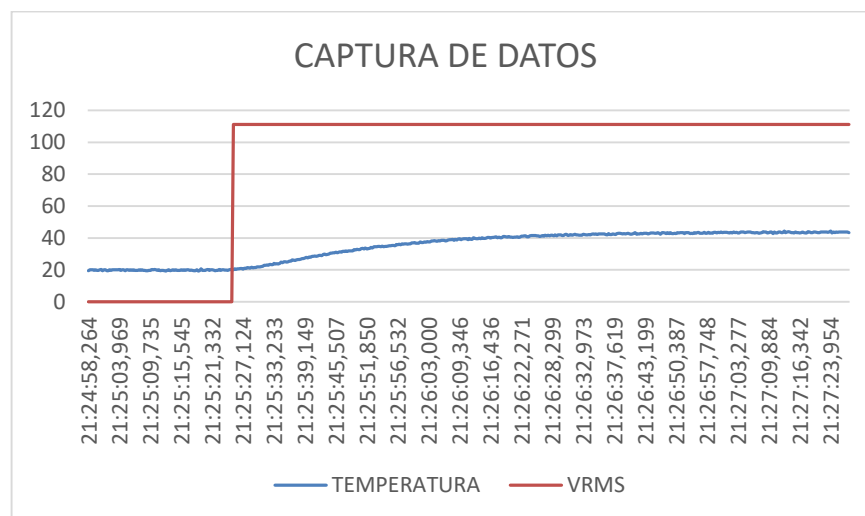
**Figura 3.12: Cálculo de VRMS en LabView.**

### Captura de datos.

Para la captura de datos se realiza la conexión del sistema embebido como se observa en el **anexo 9**, así como una serie de pasos que también se incluye en el anexo.

Los resultados obtenidos se observan en la figura 3.13, donde el voltaje aplicado a la ducha eléctrica es de 113.7 Vrms (señal escalón), con una temperatura inicial de la ducha de 20°C y con período de muestreo de 250ms alcanza una temperatura máxima aproximada de 43°C en 2 minutos con 25 segundos aproximadamente.





**Figura 3.13: Datos Capturados.**

Con los datos obtenidos en Excel el voltaje RMS máximo de 113.6 V es remplazado por el valor digital 4095 que corresponde a  $3V_{DC}$  que controla el ángulo disparo de los triacs, y mediante Matlab-Ident, permite identificar la función de transferencia mediante los pasos establecidos en el **anexo 10**, donde se establece tres casos para la identificación de la función de transferencia. La tabla 3 muestra tres funciones transferencia dos continuas y una discreta además el porcentaje de estimación.

| Casos | Nombre | Tiempo           | Polos | Ceros | Función de transferencia   | % Estimación |
|-------|--------|------------------|-------|-------|--|--------------|
| 1     | tf1    | Continuo         | 2     | 1     | $\frac{5,163 \times 10^{-5} s + 4,172 \times 10^{-5}}{s^2 + 0,198 s + 0,007157}$ | 96,46        |
| 2     | tf2    | Continuo         | 1     | 0     | $\frac{0,0002183}{s + 0,03633}$  | 92,26        |
| 3     | tf3    | Discreto (0,25s) | 2     | 1     | $\frac{2,699 \times 10^{-6} z^{-1}}{1 - 1,949 z^{-1} + 0,9492 z^{-2}}$           | 96,47        |

**Tabla 3: Funciones de transferencia.**

Con el análisis del porcentaje de estimación y la comparación de gráficas de la figura 3.14 se establece que la función de transferencia (tf3), posee el porcentaje de estimación aceptable del 96.47% y además se encuentra en tiempo discreto que permite desarrollar y simular los controladores PID

de manera más adecuada, debido a que los controladores serán programados en el sistema embebido.

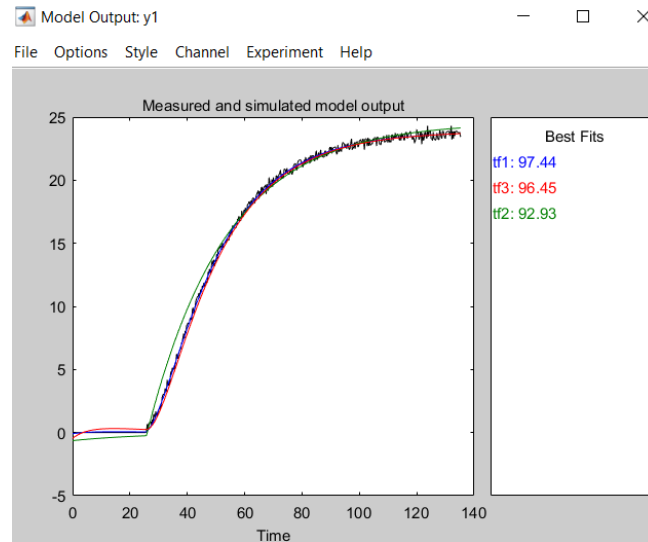


Figura 3.14: Gráficas de funciones de transferencia.

### 3.2.2 Algoritmo del PID discreto.

Para el diseño del PID discreto se realiza mediante Matlab-Simulink con la aplicación “PID Tuner” considerando la ecuación 2.12 y una serie de pasos que se muestra en el **anexo 11**, cuyos resultados se observa en la figura 3.15 y los valores en la tabla 4.

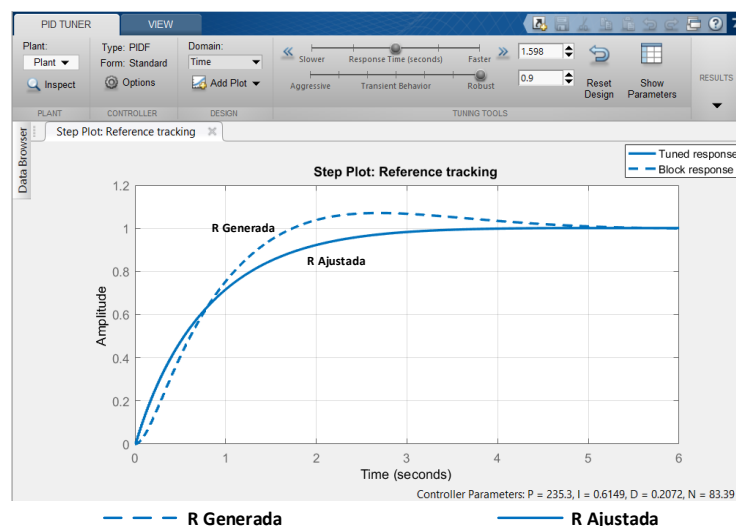


Figura 3.15: Señales de respuesta generada y ajustada con PID tuner.

| CONSTANTES PID                   |             |             |
|----------------------------------|-------------|-------------|
|                                  | R. GENERADA | R. AJUSTADA |
| P                                | 196,0421    | 1812,3529   |
| I                                | 0,041268    | 0,037316    |
| D                                | 2,78        | 6.3002      |
| VALORES DE RESPUESTA TRANSITORIA |             |             |
|                                  | R. GENERADA | R. AJUSTADA |
| t Subida                         | 29          | 5           |
| t Asentamiento                   | 112         | 11,3        |
| % Sobreimpulso                   | 6,9         | 1,84        |
| Pico máximo                      | 1,07        | 1,02        |

Tabla 4: Constates PID y valores de respuesta transitoria.

Considerando el tiempo de asentamiento y porcentaje de sobreimpulso se establecen los valores de  $K_p=1812.3529$ ,  $K_i=0.037316$ ,  $K_d=6.3002$  que se muestra en tabla 4 y mediante el programa del **anexo 12** con período de muestreo de 0.25 segundos ( $T=0.25$ ), permite determinar los valores de  $K_P$ ,  $K_I$ ,  $K_D$  aplicando las siguientes ecuaciones.

$$T_i = \frac{1}{K_i} = \frac{1}{0.037316} = 26.7978 \quad (3.9)$$

$$T_d = K_d = 6.3002 \quad (3.10)$$

$$K_I = \frac{K_p \cdot T}{T_i} = \frac{1812.3529 \cdot 0.25}{26.7978} = 16.9077 \quad (3.11)$$

$$K_D = \frac{K_p \cdot T_d}{T} = \frac{1812.3529 \cdot 6.3002}{0.25} = 45672.74 \quad (3.12)$$

$$K_P = K_p - \frac{K_I}{2} = 1812.4 - \frac{16.9077}{2} = 1803.9 \quad (3.13)$$

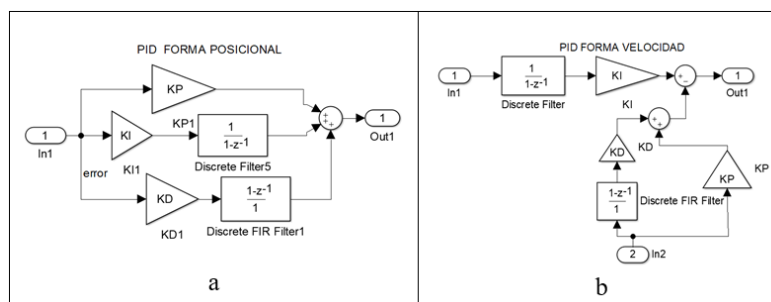


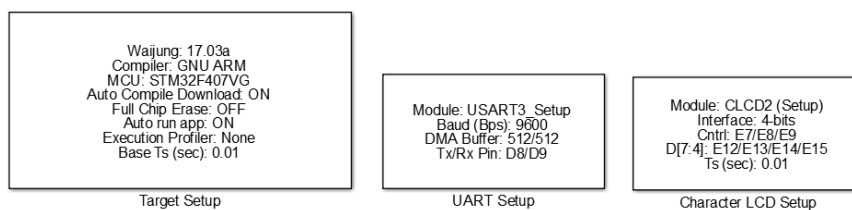
Figura 3.16: Subsistema de PID forma a) Posicional y b) Velocidad.

Con los valores establecidos de KP, KI, KD permite desarrollar los subsistemas de PID forma posicional y velocidad respectivamente como muestra en la figura 3.16.

### Programa del PID forma posicional.

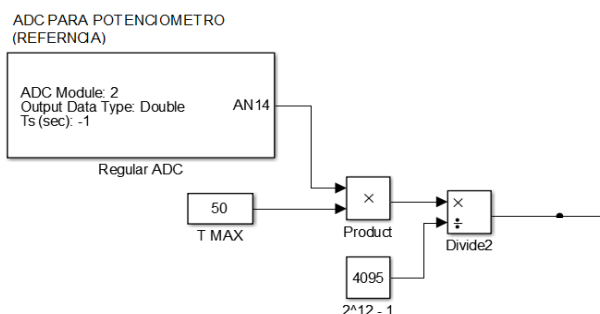
El diagrama de flujo y programa del PID posicional desarrollado en Simulink se muestra en el **anexo 13**. El programa inicia configurando el bloque "Target Setup" se configura con el compilador GNU ARM, el microcontrolador STM32F407VG, frecuencia de reloj de 8Mhz, ciclo de trabajo 0.01 segundos, considerados parámetros importantes, el resto de componentes no se modifica.

El bloque "UART Setup" y "Character LCD Setup" se configura como muestra la figura 3.17.



**Figura 3.17: Configuración inicial para PID forma Posicional.**

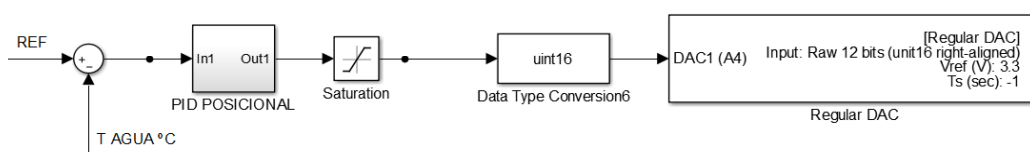
La señal analógica en AN14 permite establecer el valor de la temperatura de referencia entre 0-50°C, mediante la ecuación 3.14 como se muestra en la figura 3.18, mientras que la entrada analógica AN13 realizara las mismas operaciones de la ecuación 3.6 y figura 3.10 mencionados anteriormente y permiten obtener la temperatura del agua.



**Figura 3.18: Cálculo de la temperatura de referencia.**

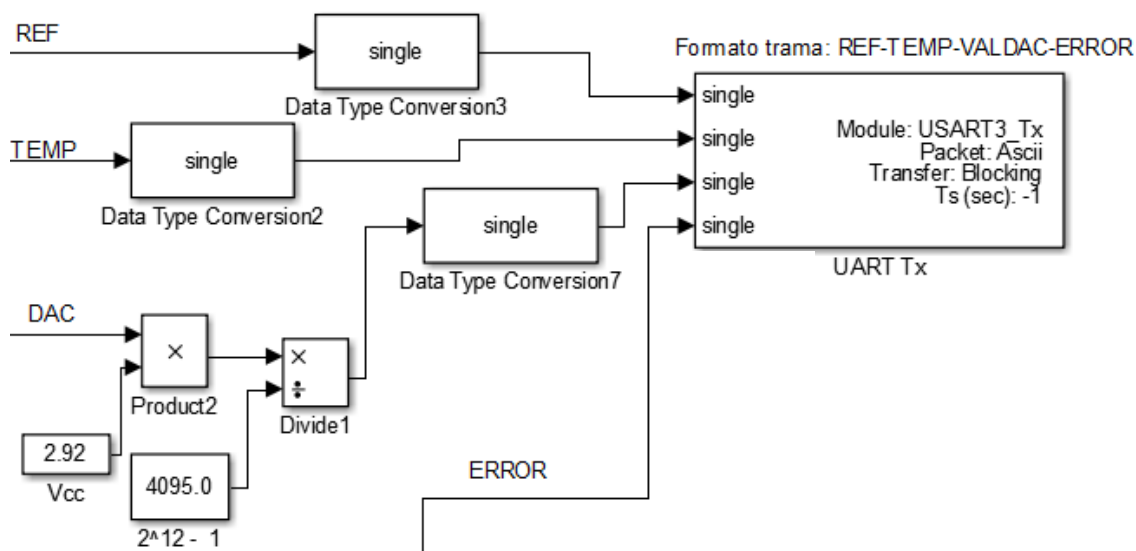
$$REF = \frac{ADC_{REF} * 50}{4095} \quad (3.14)$$

Las señales analógicas que ingresan por AN13 y AN14 se restan para obtener el error que ingresa al subsistema del PID de forma posicional que se muestra en la figura 3.16-a y el resultado hacia la DAC(A4) como muestra la figura 3.19, para obtener niveles de voltaje entre (0 – 2.8V<sub>DC</sub>), que permiten controlar el ángulo de disparo a través del microcontrolador ATmega328P.



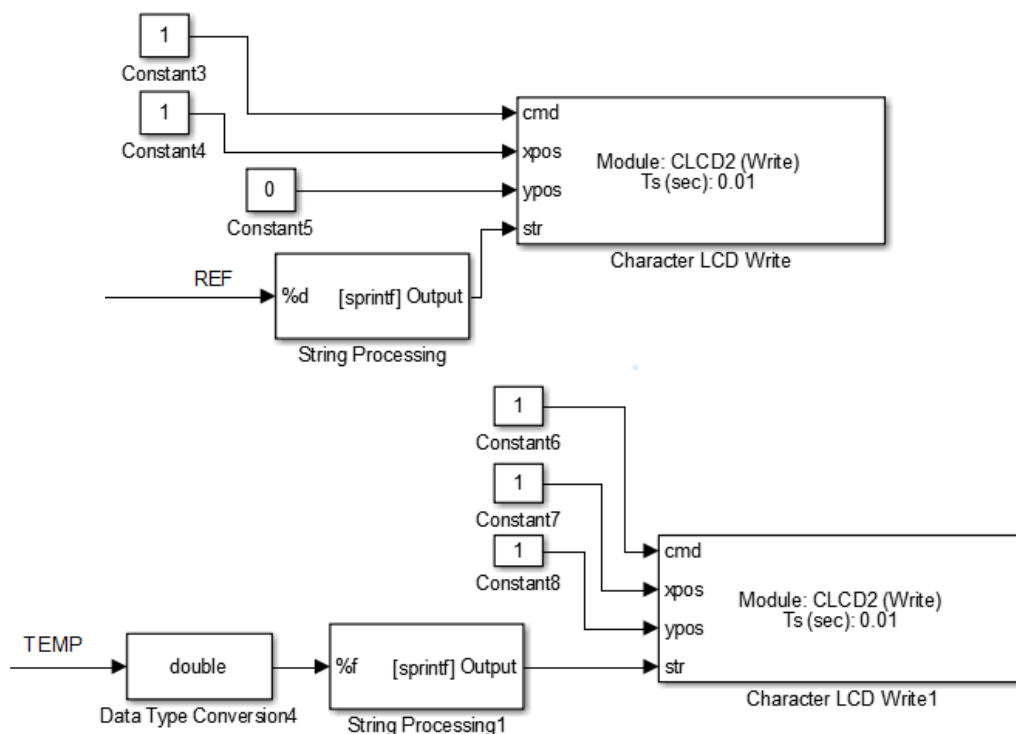
**Figura 3.19: Cálculo error y conexión con PID posicional y DAC**

El programa también envía información a través del módulo Bluetooth-serial mediante el bloque UART como muestra la figura 3.20, la trama **(R%1.3fT%1.3fV%1.3fE%1.3f)** establece una letra para determinar el tipo de dato al que pertenece (R=valor de referencia, T=valor de temperatura, V=voltaje, E=Error), y es recibida por el programa de adquisición de datos desarrollada en LabView descrita anteriormente.



**Figura 3.20: Envío de datos por puerto UART**

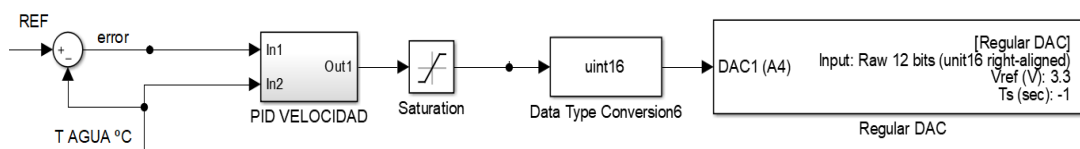
El LCD muestra la temperatura de referencia y temperatura del agua de la ducha eléctrica con los bloques que se observa la figura 3.21.



**Figura 3.21: Datos enviados por el LCD**

### Programa del PID forma de velocidad.

El diagrama de flujo y programa del PID de forma velocidad se muestra en el **anexo 14**, tiene los mismos bloques y parámetros que el PID de forma posicional y solo cambia el subsistema de PID por el subsistema del PID de velocidad de la figura 3.16-b considerando dos entradas, la primera entrada es el error producido entre la referencia y la temperatura del agua, la segunda es exclusivamente la temperatura del agua como se muestra en la figura 3.22.



**Figura 3.22: Cálculo error y conexión con PID velocidad y DAC**

### 3.2.3 Algoritmo del PID difuso.

Para el Desarrollo del PID difuso se establecen los parámetros de entrada y salida que se muestra en las tablas 5 y 6, basados en los siguientes criterios.

- La respuesta a una señal escalón de la función de transferencia de la planta (ducha eléctrica).
- El valor máximo de error de temperatura que se establece con la temperatura inicial del agua entre 15 a 20°C y la temperatura adecuada para el baño entre 30 a 35°C que determina una diferencia máxima aproximada 20 °C que determina un universo discurso de [-20,20] °C.
- La variable de salida establece valores para el DAC entre 0 a 3926 estableciendo el valor máximo para el universo discurso de [-4000, 4000] que permite controlar la ducha eléctrica.

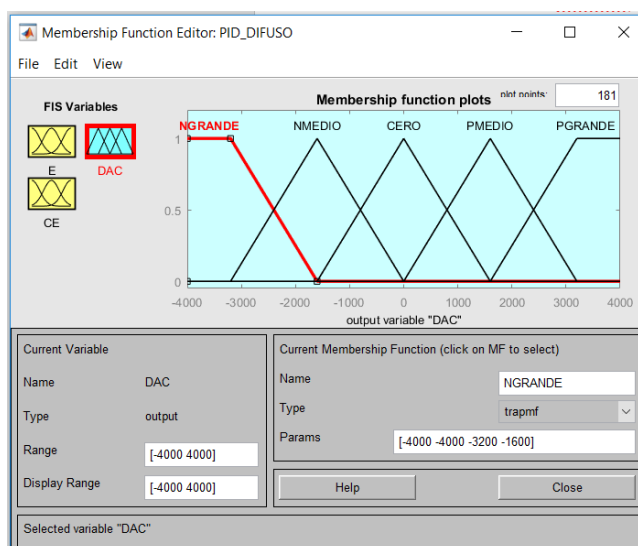
| ENTRADAS               |   |   |
|------------------------|---|---|
| Variable lingüística   | E=Error                                   | CE=Cambio de Error                        |
| Universo discurso      | [-20,20]                                  | [-20,20]                                  |
| Términos Lingüísticos  | NGRANDE                                   | NGRANDE                                   |
|                        | NMEDIO                                    | NMEDIO                                    |
|                        | ZERO                                      | ZERO                                      |
|                        | PMEDIO                                    | PMEDIO                                    |
|                        | PGRANDE                                   | PGRANDE                                   |
| Funciones de Membresía | Trapez. abierto Izq.<br>[-20,-20,-20,-10] | Trapez. abierto Izq.<br>[-20,-20,-20,-10] |
|                        | Triangular<br>[-20,-10,0]                 | Triangular<br>[-20,-10,0]                 |
|                        | Triangular<br>[-10,0,10]                  | Triangular<br>[-10,0,10]                  |
|                        | Triangular [0,10,20]                      | Triangular<br>[0,10,20]                   |
|                        | Trapez. abierto der.<br>[10,20,20,20]     | Trapez. abierto der.<br>[10,20,20,20]     |

Tabla 5: Parámetros de las entradas de controlador PID difuso.

| SALIDA                        |   |
|-------------------------------|---|
| <b>Variable lingüística</b>   | DAC=Valores del DAC                               |
| <b>Universo discurso</b>      | [-4000,4000]                                      |
| <b>Términos Lingüísticos</b>  | NGRANDE   |
|                               | NMEDIO  |
|                               | ZERO  |
|                               | PMEDIO  |
|                               | PGRANDE   |
| <b>Funciones de Membresía</b> | Trapez. abierto lzq.<br>[-4000,-4000,-3200,-1600] |
|                               | Triangular<br>[-3200,-1600,0]                     |
|                               | Triangular<br>[-1600,0,1600]                      |
|                               | Triangular<br>[0,1600,3200]                       |
|                               | Trapez. abierto der.<br>[1600,3200,4000,4000]     |

**Tabla 6: Parámetros de salida de controlador PID difuso**

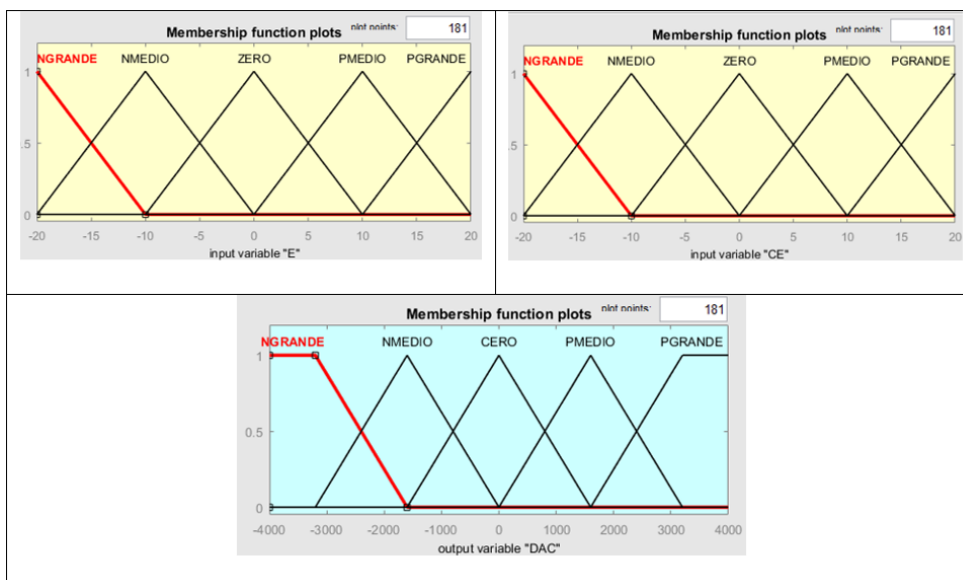
Con los parámetros determinados se realiza el controlador PID difuso con base “Mamdani” con “Fuzzy Logic Designer” de Matlab que se observa en la figura 3.23, mediante los pasos que se muestra en el **anexo 15**.



**Figura 3.23: Fuzzy Logic Designer**

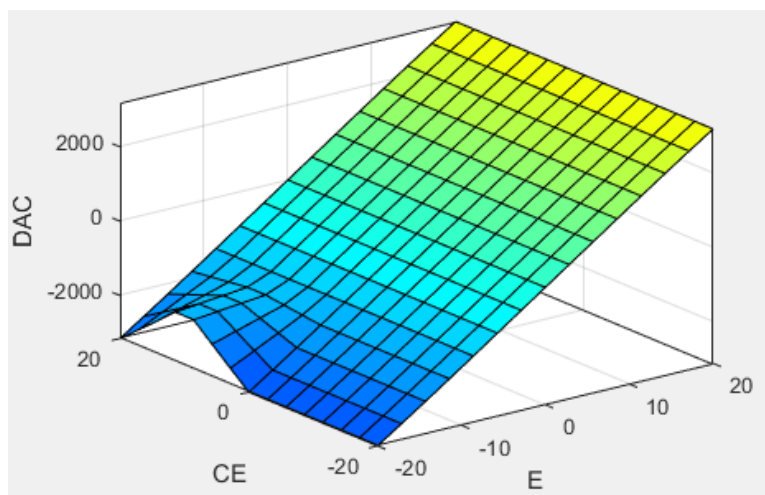


Las gráficas de las funciones de membresía de entradas y salida consideran los valores establecidos en las tablas 5 y 6, y se observa en la figura 3.24.



**Figura 3.24: Funciones de Membresía PID difuso.**

La tabla 2 mencionada anteriormente establece 25 reglas lingüísticas que forma la superficie de control de la figura 3.25.



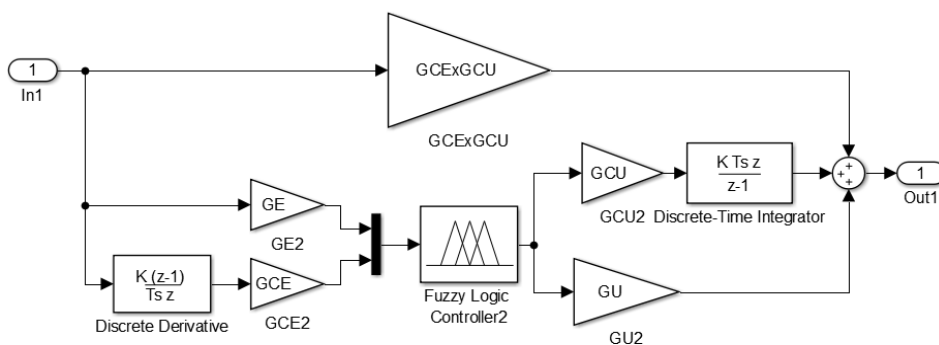
**Figura 3.25: Superficie de Control.**

También se calculan las ganancias, GCE, GU, GCU y GCExGCU utilizando las ecuaciones 2.17, 2.18, 2.19, considerando el valor  $GE=1$  y se obtiene los valores que se muestra en la tabla 7

| GANANCIAS PID DIFUSO |        |
|----------------------|--------|
| GANACIA              | VALOR  |
| GE                   | 1      |
| GCE                  | 0.0035 |
| GCU                  | 0.0373 |
| GU                   | 1812.4 |

**Tabla 7: Valores de ganancia para Simulink.**

Finalmente, el diagrama de flujo y programa del PID difuso se muestra en el **anexo 16**, tiene los mismos bloques y parámetros que el PID de forma posicional y velocidad, solo se reemplaza el subsistema de PID por el PID Difuso que se muestra en la figura 3.26.



**Figura 3.26: PID Difuso**

## CAPÍTULO 4

### 4. ANÁLISIS COMPARATIVO DE LOS RESULTADOS.

Para el análisis comparativo de los resultados de los PID posicional, velocidad, difuso se realiza las simulaciones en Matlab-Simulink con los diagramas de bloques de los **anexos 17,18,19** y se exportan los datos reales de Excel a Matlab, cuyas gráficas se muestra en el **anexo 20** obtenidos mediante la aplicación desarrollada en LabView y además se establece los siguientes parámetros:

- Tiempo total de 450 segundos de análisis comparativo con periodo de muestreo de 0.25 segundos.
- Análisis de los controladores desde la temperatura inicial de 20°C hasta alcanzar la temperatura de referencia de 32°C durante 250 segundos aproximadamente, posteriormente un incremento de 3°C a la temperatura de referencia para alcanzar 35°C durante 200 segundos aproximadamente.
- Señal aleatoria distribuida normalmente (gaussiana) con 0.05 de varianza de los números aleatorios para la generación del ruido que permite la simulación de los controladores con ruido en simulink.
- Tolerancia del 5% del valor final de temperatura para determinar el tiempo de establecimiento más rápido.
- Porcentaje sobreimpulso que permite determinar el controlador con mayor valor de temperatura pico.
- Tiempo de subida que permite determinar el controlador que más rápido alcanza la temperatura de referencia.
- Consumo promedio de corriente y voltaje RMS de la planta.

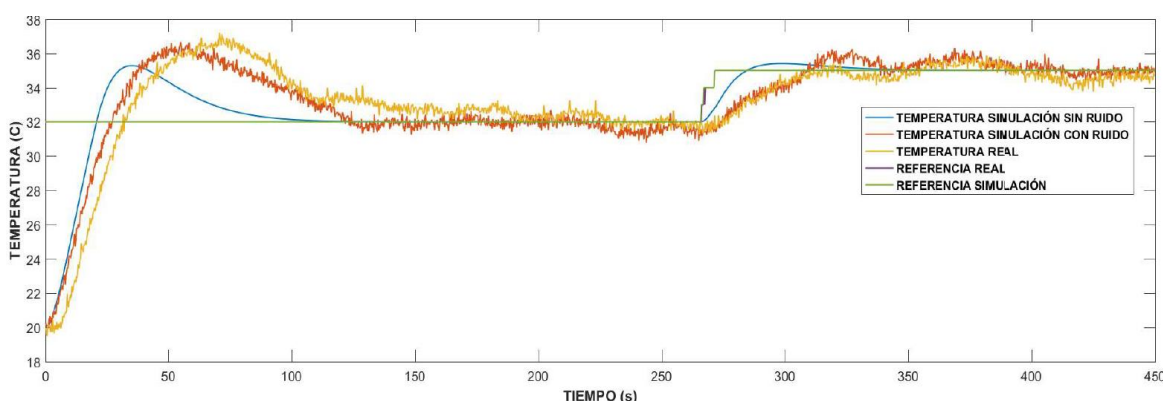
#### 4.1 Evaluación del Desempeño del control embebido con PID Discreto.

Para realizar una adecuada evaluación del desempeño del control embebido con los PID discretos de forma posicional y velocidad se establece los siguientes criterios de comparación:

- Comparación del PID posicional simulado y real.
- Comparación del PID de velocidad simulado y real.

#### 4.1.1 Comparación del PID posicional simulado y real.

La figura 4.1 muestra la comparación de las señales de temperatura de la planta del PID de forma posicional simulado con o sin ruido y real, obtenidas mediante el programa desarrollado en Matlab que se muestra en el **anexo 21**, así como las temperaturas de referencia de 32°C y 35°C respectivamente, donde la respuesta de la planta simulada sin ruido tiene menor tiempo de establecimiento con respecto a los simulados con o sin ruido, mientras la respuesta de la planta real es semejante a la respuesta simulado con ruido con una diferencia de 1°C en algunos tramos.



**Figura 4.1: Respuestas del PID posicional simulado y real.**

La tabla 8 muestra los resultados obtenidos para alcanzar la temperatura de referencia de 32°C partiendo de una temperatura inicial de agua de la ducha eléctrica de 20°C, donde la temperatura pico real de 37.22°C con sobreimpulso 43.467% es el mayor con respecto a los simulados, mientras el tiempo de subida entre las respuestas simuladas con o sin ruido y real tiene una diferencia promedio de 5.25 segundos, además el tiempo de establecimiento de 188.5 segundos del sistema real es el más lento con respecto a los simulados.

La tabla 9 muestra los resultados al realizar un incremento de 3°C a la temperatura de referencia de 32°C a los 266 segundos aproximadamente

para alcanzar 35°C, donde la temperatura de establecimiento en la simulación con ruido tiene picos de temperatura que varían entre 34.5 a 35.5°C, mientras que el sistema real tiene picos de temperatura que varía entre 34 a 35.15°C, además el tiempo de establecimiento del sistema real de 123 segundos con una diferencia 7.2 y 60 segundos con respecto al sistema simulado con o sin ruido respectivamente.

| PID FORMA POSICIONAL             |                |                |        |
|----------------------------------|----------------|----------------|--------|
| Temperatura Inicial =20°C        |                |                |        |
| Temperatura de referencia =32°C  |                |                |        |
| Características de respuesta     | SIM. SIN RUIDO | SIM. CON RUIDO | REAL   |
| TEMPERATURA PICO (°C)            | 35,29          | 36,7           | 37,22  |
| SOBREIMPULSO MP (%)              | 27,399         | 39,158         | 43,467 |
| TIEMPO DE SUBIDA tr (s)          | 21             | 26,5           | 31,5   |
| TIEMPO PICO tp (s)               | 34,75          | 55             | 70,75  |
| TIEMPO DE ESTABLECIMIENTO ts (s) | 77             | 129,3          | 188,5  |

**Tabla 8: Características de respuestas del PID posicional simulados y real.**

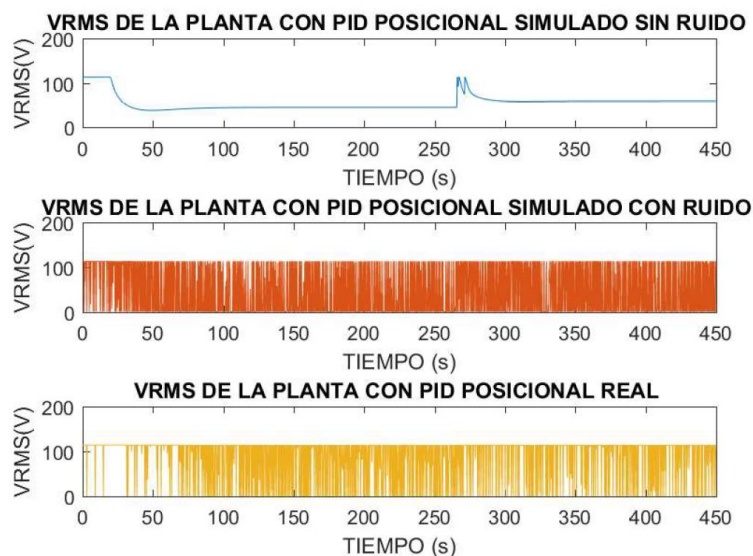
| PID FORMA POSICIONAL  |                |                |          |
|---|----------------|----------------|----------|
| Incremento de 3°C a la Temperatura de 32°C para alcanzar los 35°C de referencia |                |                |          |
| El cambio inicia a los 266 segundos   |                |                |          |
| Características de respuesta  | SIM. SIN RUIDO | SIM. CON RUIDO | REAL     |
| TEMPERATURA DE ESTABLECIMIENTO(°C)  | 35             | 35±0,5         | 34-35,15 |
| TIEMPO DE SUBIDA tr (s)   | 18,3           | 45             | 45,8     |
| TIEMPO DE ESTABLECIMIENTO ts (s)  | 63             | 115,8          | 123      |

**Tabla 9: Características de respuestas al incremento 3°C al PID posicional simulados y real.**

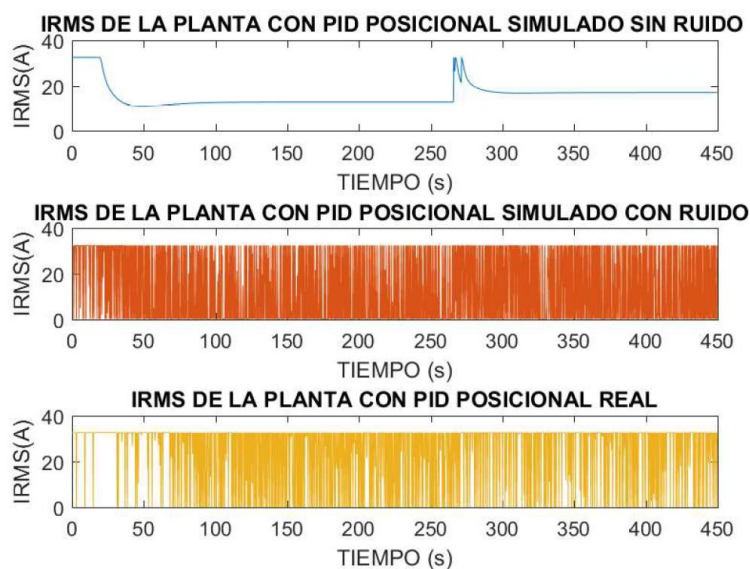
La figura 4.2 muestra el Voltaje RMS aplicado a la planta, donde la simulación sin ruido inicia con 113.6V durante 19.5 segundos, después del tiempo de establecimiento el voltaje baja a 44.8V y permanece constante, cuando se realiza el incremento de 3°C, el voltaje sube nuevamente a 113.6V por cortos periodos de tiempo, para posteriormente alcanzar un voltaje constante de 59.48V, mientras las simulaciones con ruido y real tienen valores variables entre 0 a 113.6V de acuerdo al ángulo de disparo enviado por el controlador.

La figura 4.3 muestra la corriente RMS consumida por la planta, donde la simulación sin ruido inicia con una corriente de 32.46A durante 19.5 segundos, después del tiempo de establecimiento la corriente permanece

constante en 12.8A y cuando se realiza el incremento de 3°C el amperaje sube a 32.46A por un corto periodo de tiempo, después del tiempo de establecimiento permanece constante en 16.97A, mientras la simulación con ruido y real tienen valores variables entre 0 a 32.46A de acuerdo al ángulo de disparo enviado por el controlador.



**Figura 4.2: Voltaje RMS de la planta con PID posicional simulados y real.**



**Figura 4.3: Corriente RMS de la planta con PID posicional simulados y real.**

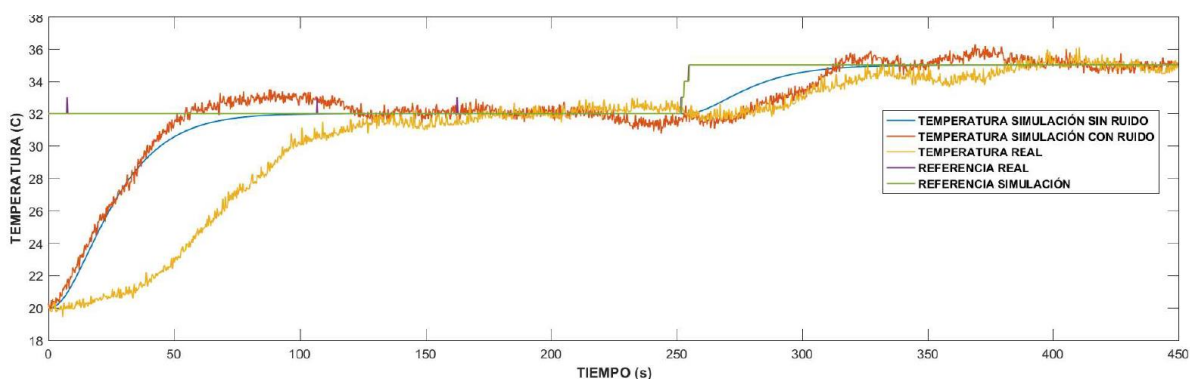
| PID FORMA POSICIONAL |                |                |           |
|----------------------|----------------|----------------|-----------|
| PARAMETROS           | SIM. SIN RUIDO | SIM. CON RUIDO | REAL      |
| VRMS PROMEDIO (V)    | 55,0187        | 48,1449        | 86,2148   |
| IRMS PROMEDIO (A)    | 15,7196        | 13,7557        | 24,6328   |
| POTENCIA (w)         | 864.8720       | 662.2668       | 2123.7120 |

**Tabla 10: Voltaje y Corriente RMS de la planta con PID posicional simulados y real.**

La tabla 10 muestra los resultados promedios de voltaje, corriente y potencia RMS, desde una temperatura inicial de 20°C hasta alcanzar 35°C durante el intervalo de tiempo de 450 segundos, donde el PID posicional real tiene la potencia mayor de 2123,71 vatios.

#### 4.1.2 Comparación del PID de velocidad simulado y real.

La figura 4.4 muestra la comparación de las señales de temperatura de la planta con los PID de velocidad simulado con o sin ruido y real, obtenidas mediante el programa desarrollado en Matlab que se muestra en el **anexo 22**, donde el PID simulado con ruido posee el pico más alto de temperatura, además el sistema simulado con ruido y real tocan la temperatura de referencia de 32°C casi al mismo tiempo, aunque el tiempo de establecimiento del sistema real sea mayor, al realizar el incremento de 3°C para alcanzar 35°C el tiempo de establecimiento tiene pequeñas diferencias entre los sistemas simulados con ruido y real.



**Figura 4.4: Respuestas del PID de velocidad simulados y real.**

La tabla 11 muestra los resultados obtenidos para alcanzar la temperatura de referencia de 32°C, donde la simulación sin ruido no posee sobreimpulso y la temperatura pico simulada con ruido de 33.37°C es el mayor y tienen una diferencia de apenas 0.48°C respecto al real de 32.89°C, mientras que el tiempo de establecimiento real de 168 segundos es mayor en 39 segundos con respecto al simulado con ruido.

| PID FORMA VELOCIDAD              |                |                |       |
|----------------------------------|----------------|----------------|-------|
| Temperatura Inicial =20°C        |                |                |       |
| Temperatura de referencia =32°C  |                |                |       |
| Características de respuesta     | SIM. SIN RUIDO | SIM. CON RUIDO | REAL  |
| TEMPERATURA PICO (°C)            | 32             | 33,37          | 32,89 |
| SOBREIMPULSO MP (%)              | 0              | 11,378         | 7,45  |
| TIEMPO DE SUBIDA tr (s)          | 63.5           | 66,25          | 154,8 |
| TIEMPO PICO tp (s)               | 63.5           | 110,3          | 167,5 |
| TIEMPO DE ESTABLECIMIENTO ts (s) | 63.5           | 129            | 168   |

**Tabla 11: Características de respuestas del PID de velocidad simulados y real.**

La tabla 12 muestra los resultados al realizar el incremento de 3°C a los 251.5 segundos a la temperatura referencia de 32°C para alcanzar 35°C, donde la temperatura de establecimiento simulado con ruido posee picos de temperatura que varía entre 34.5 a 35.5°C, mientras que la temperatura de asentamiento real tiene picos de voltaje entre 38.85 a 36°C, además el tiempo de establecimiento real de 134 segundos es el mayor con una diferencia de 4.2 segundos con respecto al tiempo de establecimiento real.

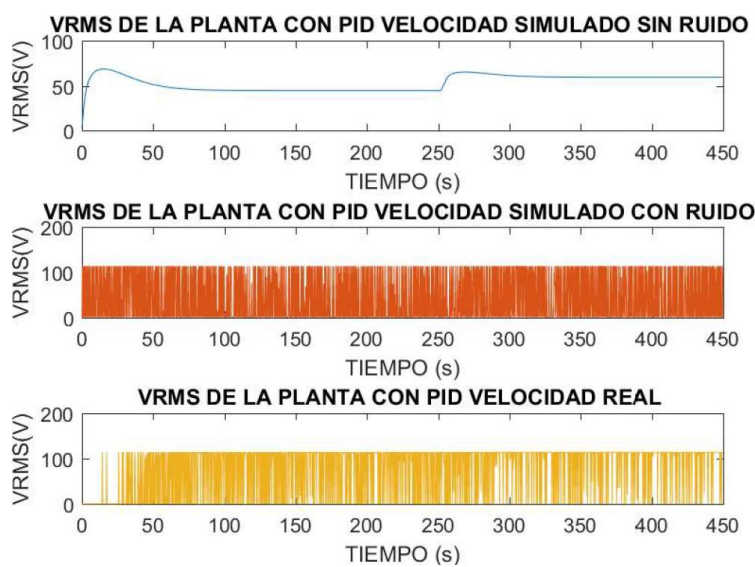
| PID FORMA VELOCIDAD   |                |                |          |
|---|----------------|----------------|----------|
| Incremento de 3°C a la Temperatura de 32°C para alcanzar los 35°C de referencia |                |                |          |
| El cambio inicia a los 251,5 segundos   |                |                |          |
| Características de respuesta  | SIM. SIN RUIDO | SIM. CON RUIDO | REAL     |
| TEMPERATURA DE ESTABLECIMIENTO(°C)  | 35             | 35±0,5         | 34,85-36 |
| TIEMPO DE SUBIDA tr (s)   | 65             | 83,8           | 74,5     |
| TIEMPO DE ESTABLECIMIENTO ts (s)  | 65             | 129,8          | 134      |

**Tabla 12: Características de respuestas al incremento 3°C al PID de velocidad simulados y real.**

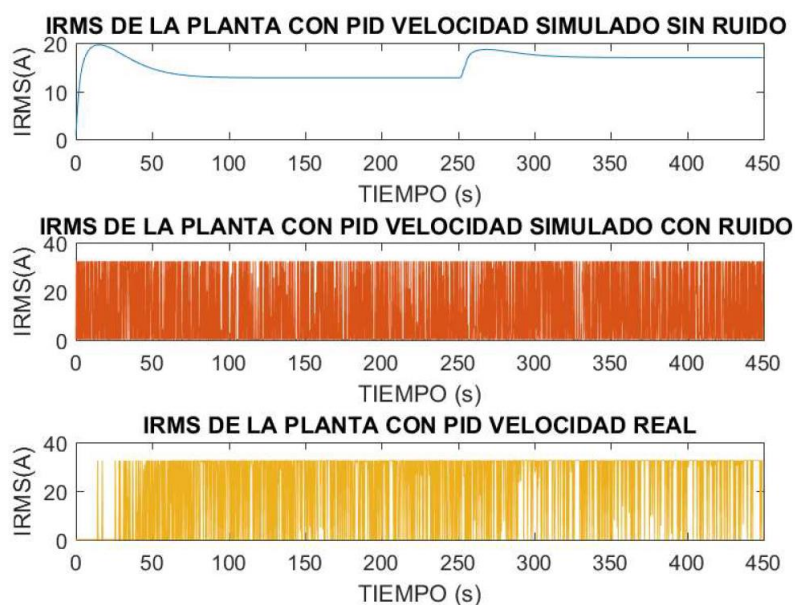
La figura 4.5 muestra el Voltaje RMS aplicado a la planta, donde la simulación sin ruido alcanza un voltaje pico 68.72V a los 15 segundos, después del tiempo de asentamiento el voltaje baja a 44.8V y permanece



constante, cuando se realiza el incremento de  $3^{\circ}\text{C}$  el voltaje alcanza un pico de  $65.36\text{V}$ , para posteriormente bajar a  $59.87\text{V}$  y permanecer constante, mientras las simulaciones con ruido y real tienen variaciones de voltaje entre  $0$  a  $113.6\text{V}$  de acuerdo al ángulo de disparo enviado por el controlador.



**Figura 4.5: Voltaje RMS de la planta con PID de velocidad simulados y real.**



**Figura 4.6: Corriente RMS de la planta con PID de velocidad simulados y real.**

La figura 4.6 muestra la corriente RMS aplicada a la planta, donde la simulación sin ruido inicia con una corriente pico de 19.63<sup>a</sup>, a los 15 segundos después del tiempo de establecimiento la corriente permanece constante en 12.87A y cuando se realiza el incremento de 3°C la corriente alcanza un pico de 18.67A, después del tiempo de establecimiento alcanza una corriente de 17.3A, mientras en la simulación con ruido y real tienen variaciones de corriente entre 0 a 32.46A de acuerdo al ángulo de disparo enviado por el controlador.

| PID FORMA VELOCIDAD |                |                |           |
|---------------------|----------------|----------------|-----------|
| PARAMETROS          | SIM. SIN RUIDO | SIM. CON RUIDO | REAL      |
| VRMS PROMEDIO (V)   | 53,5059        | 45,5332        | 74,939    |
| IRMS PROMEDIO (A)   | 15,2874        | 13,0095        | 21,411    |
| POTENCIA (W)        | 817.9660       | 592.3642       | 1604.5190 |

**Tabla 13: Voltaje y Corriente RMS de la planta con PID de velocidad simulados y real.**

La tabla 13 muestra los resultados promedios de voltaje, corriente y potencia RMS obtenidos desde una temperatura inicial de 20°C hasta alcanzar 35°C durante el intervalo de tiempo de 450 segundos, donde el PID de velocidad real tiene la mayor potencia de 1604.52 vatios.

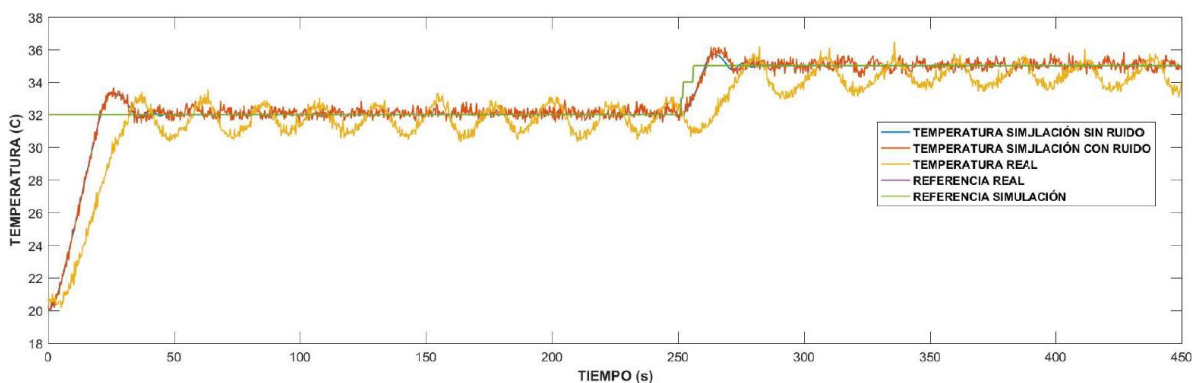
## 4.2 Evaluación del Desempeño del control embebido con PID Difuso.

Para la evaluación del desempeño del control embebido se establece la comparación del PID difuso simulados con o sin ruido y real.

### 4.2.1 Comparación del PID difuso simulados y real.

La figura 4.7 muestra la comparación entre las señales de temperatura de la planta con PID difuso simuladas con o sin ruido y real, obtenidas mediante el programa desarrollado en Matlab que se muestra en el **anexo 23**, así como las temperaturas de referencia de 32°C y 35°C respectivamente, donde las tres señales tienen la temperatura pico que no sobrepasa los 34°C, además las respuestas de temperatura simulada con y sin ruido tienen respuestas similares, mientras que la temperatura real muestra un comportamiento oscilatorio entre valores de 31 a 33°C aproximadamente cuando la temperatura de referencia es 32°C, al

realizar el incremento de 3°C a los 251 segundos aproximadamente para alcanzar la temperatura de referencia de 35°C, la temperatura simulada con y sin ruido tienen comportamiento similar, mientras que la respuesta de la temperatura real permanece oscilatoria.



**Figura 4.7: Respuestas del PID difuso simulados y real.**

La tabla 14 muestra los resultados de las señales de temperatura simuladas con o sin ruido y real, donde la temperatura pico de 33.98°C simulada con ruido es el más alto con un porcentaje de sobreimpulso de 14.824%, además el tiempo de establecimiento simulado con o sin ruido poseen una diferencia aproximada de 5.5 segundos, mientras que la señal de temperatura real a partir de 52 segundos mantiene una oscilación entre 31 a 33°C aproximadamente.

| PID DIFUSO                       |                |                |         |
|----------------------------------|----------------|----------------|---------|
| Temperatura Inicial =20°C        |                |                |         |
| Temperatura de referencia =32°C  |                |                |         |
| Características de respuesta     | SIM. SIN RUIDO | SIM. CON RUIDO | REAL    |
| TEMPERATURA PICO (°C)            | 33,33          | 33,98          | 33,55   |
| SOBREIMPULSO MP (%)              | 11,084         | 14,824         | 12,942  |
| TIEMPO DE SUBIDA tr (s)          | 21             | 20,5           | 32      |
| TIEMPO PICO tp (s)               | 25,75          | 26,25          | 63,5    |
| TIEMPO DE ESTABLECIMIENTO ts (s) | 32,56          | 39,75          | 52(osc) |

**Tabla 14: Características de respuestas del PID difuso simulados y real.**

La tabla 15 muestran los resultados cuando se realiza el incremento de 3°C a la temperatura de referencia de 32°C de manera escalonada para alcanzar 35°C, donde el tiempo de establecimiento promedio simulado con o sin ruido es aproximadamente es de 26.125 segundos, mientras la

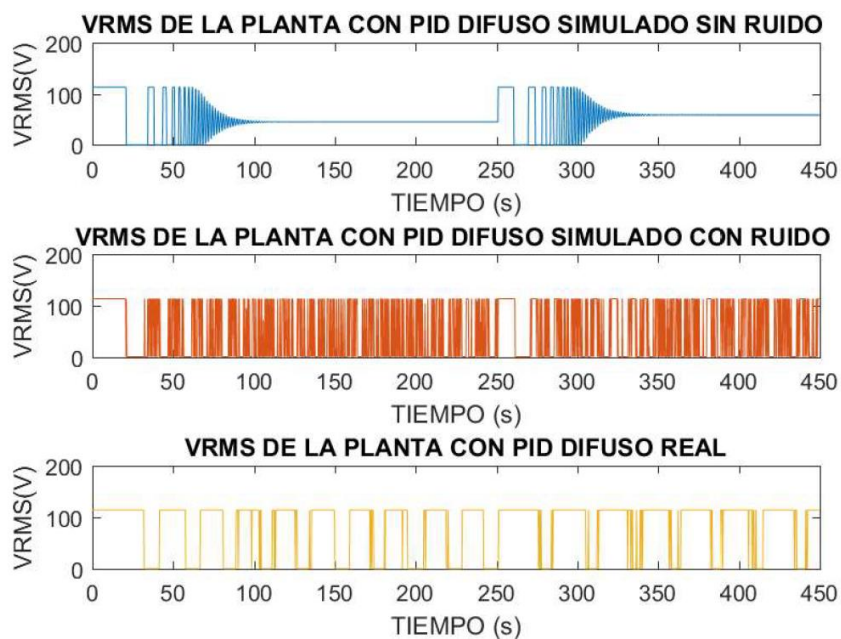
temperatura real oscila entre 33.5 a 35.8°C a partir de los 40 segundos de haber recibido el incremento de temperatura.

| PID DIFUSO   |                |                |                |
|--|----------------|----------------|----------------|
| Incremento de 3°C a la Temperatura de 32°C para alcanzar los 35°C de referencia<br>El cambio inicia a los 251 segundos |                |                |                |
| Características de respuesta   | SIM. SIN RUIDO | SIM. CON RUIDO | REAL           |
| TEMPERATURA ESTABLECIMIENTO(°C)  | 35             | 35±0,5         | 33,5-35,8(osc) |
| TIEMPO DE SUBIDA tr (s)  | 9,8            | 8,8            | 25             |
| TIEMPO DE ESTABLECIMIENTO ts (s)   | 25,75          | 26,5           | 40(osc)        |

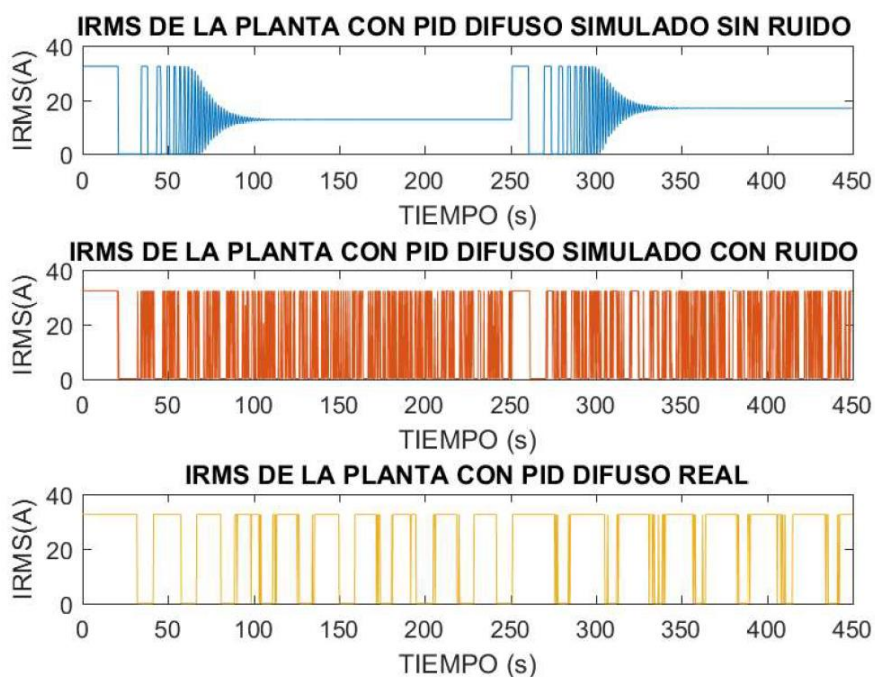
**Tabla 15: Características de respuestas al incremento 3°C al PID difuso simulados y real.**

La figura 4.8 muestra el Voltaje RMS aplicado a la planta, donde el voltaje de simulación sin ruido tiene un voltaje subamortiguado con un valor máximo de 113.6V, después de 111.8 segundos permanece aproximadamente constante en 44.8V, cuando se realiza el incremento de 3°C a los 251 segundos el voltaje es subamortiguado y alcanza un pico de 113.6V, después de 92.8 segundos permanece aproximadamente constante en 59.1V, mientras las simulaciones con ruido y real tienen variaciones de voltaje entre 0 a 113.6V, pero el simulado con ruido posee mayor número de variaciones que el real de acuerdo al ángulo de disparo enviado por el controlador.

La figura 4.9 muestra la corriente RMS aplicada a la planta, donde la simulación sin ruido tiene una señal de corriente subamortiguado que inicia con 32.46A, después de 111.8 segundos la corriente permanece aproximadamente constante en 12.92A y cuando se realiza el incremento de 3°C la señal de corriente subamortiguado es parecida a la mencionada anteriormente, después de 92.8 segundos la corriente permanece aproximadamente constante en 17.15A, mientras en la simulación con ruido y real tienen variaciones de corriente entre 0 a 32.46A, pero el simulado con ruido posee mayor número de variaciones de acuerdo al ángulo de disparo enviado por el controlador.



**Figura 4.8: Voltaje RMS de la planta con PID difuso simulados y real.**



**Figura 4.9: Corriente RMS de la planta con PID difuso simulados y real.**

La tabla 16 muestra los resultados promedios de voltaje, corriente y potencia RMS obtenidos desde una temperatura inicial de 20°C hasta

alcanzar 35°C durante el intervalo de tiempo de 450 segundos, donde el PID difuso real tiene la mayor potencia de 1806.92 vatios.

| PID DIFUSO        |                |                |           |
|-------------------|----------------|----------------|-----------|
| PARAMETROS        | SIM. SIN RUIDO | SIM. CON RUIDO | REAL      |
| VRMS PROMEDIO (V) | 52,8312        | 46,8598        | 79,525    |
| IRMS PROMEDIO (A) | 15,0946        | 13,3885        | 22,7214   |
| POTENCIA (W)      | 797.4658       | 613.9939       | 1806.9193 |

**Tabla 16: Voltaje y Corriente RMS de la planta con PID difuso simulados y real.**

### 4.3 Comparación de los controladores PID discreto y difuso.

La comparación entre los resultados de los controladores PID discretos y difuso simulados con o sin ruido y reales para el control de la planta, obtenidas mediante el programa desarrollado en Matlab que se muestra en el **anexo 24**, donde se establecen dos eventos, el primer evento inicia con 20°C hasta alcanzar una temperatura de referencia de 32°C y el segunda evento al realizar un incremento de 3°C para obtener una temperatura de referencia de 35°C, estableciendo los siguientes parámetros de comparación.

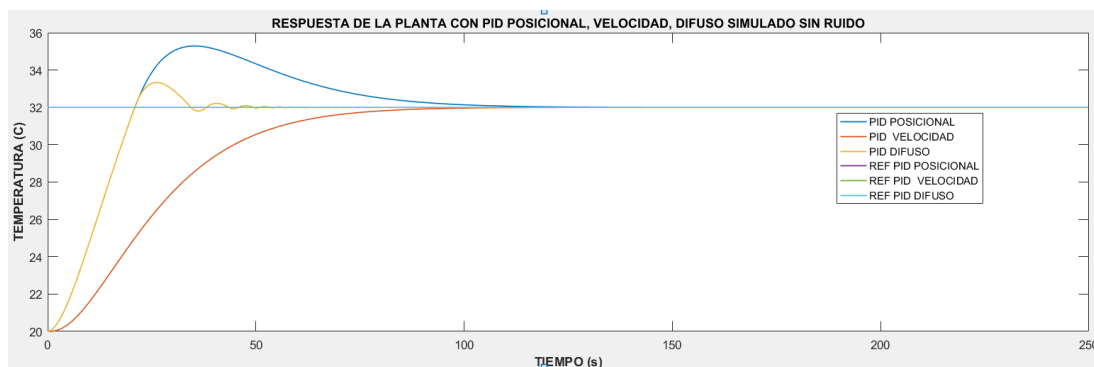
- Comparación de PID discretos y difuso sin ruido.
- Comparación de PID discretos y difuso con ruido.
- Comparación de PID discretos y difuso reales.

#### 4.3.1 Comparación de PID discretos y difuso sin ruido.

La figura 4.10 muestra la comparación entre las señales de temperatura de la planta con los controladores PID discretos y difuso simulados sin ruido, cuando inicia con una temperatura de 20°C hasta alcanzar la temperatura de referencia de 32°C, donde la temperatura pico del PID posicional es el mayor con respecto a las demás señales, mientras que el PID de velocidad no posee sobreimpulso y el tiempo de establecimiento del PID difuso es el más rápido.

La tabla 17 muestra los resultados obtenidos en la simulación de los controladores discreto y difuso sin ruido, cuando inicia con una temperatura de 20°C hasta alcanzar la temperatura de referencia de 32°C, donde el porcentaje de sobreimpulso de 27.399% del PID Posicional con

temperatura pico de 35.29°C es el más alto, mientras que el PID de velocidad no posee sobreimpulso, además el tiempo de subida de los PID posicional y difuso de 21 segundos son iguales, mientras el tiempo de establecimiento del PID difuso de 32.56 segundos es el más rápido de todos.

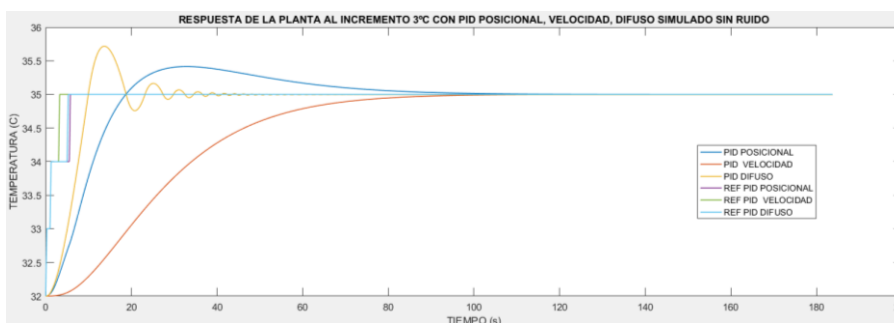


**Figura 4.10: Respuestas de los PID simulados sin ruido.**

| PID SIMULADOS SIN RUIDO             |            |           |        |
|-------------------------------------|------------|-----------|--------|
| Temperatura Inicial =20°C           |            |           |        |
| Temperatura de referencia =32°C     |            |           |        |
| Características de respuesta        | POSICIONAL | VELOCIDAD | DIFUSO |
| TEMPERATURA PICO (°C)               | 35,29      | 32        | 33,33  |
| SOBREIMPULSO MP (%)                 | 27,399     | 0         | 11,084 |
| TIEMPO DE SUBIDA $t_r$ (s)          | 21         | 63.5      | 21     |
| TIEMPO PICO $t_p$ (s)               | 34,75      | 63.5      | 25,75  |
| TIEMPO DE ESTABLECIMIENTO $t_s$ (s) | 77         | 63.5      | 32,56  |

**Tabla 17: Características de Respuestas de los PID simulados sin ruido.**

La figura 4.11 muestra la comparación entre las señales de temperatura de la planta con los controladores PID discretos y difuso sin ruido al realizar el incremento de 3°C de manera escalonada a la temperatura de referencia de 32°C hasta alcanzar los 35°C, donde el PID difuso y posicional no sobrepasan la temperatura de 36°C, mientras que el PID de velocidad no tiene sobreimpulso y el PID difuso tiene el tiempo de establecimiento más rápido.



**Figura 4.11: Respuestas de los PID simulados sin ruido al incremento de 3°C.**

La tabla 18 muestran los resultados obtenidos en la simulación de los controladores al realizar un incremento de 3°C de manera escalonada a la temperatura de referencia de 32°C para alcanzar los 35°C, donde la temperatura de subida del PID difuso de 9.8 segundos es la más rápida con respecto al PID posicional y velocidad, mientras el PID difuso alcanza la temperatura de establecimiento de 35°C más rápido con respecto a los PID posicional y velocidad en 25.75 segundos.

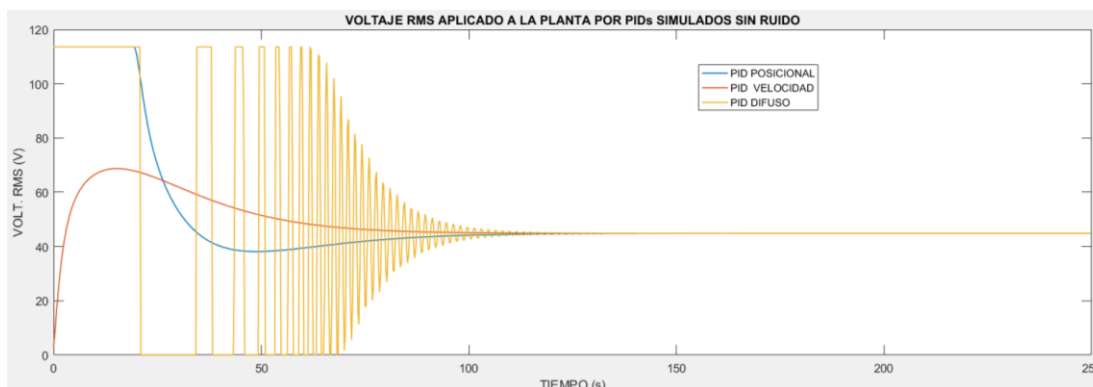
| PID SIMULADOS SIN RUIDO   |            |           |        |
|---|------------|-----------|--------|
| Incremento de 3°C a la Temperatura de 32°C para alcanzar los 35°C de referencia |            |           |        |
| Características de respuesta  | POSICIONAL | VELOCIDAD | DIFUSO |
| TEMPERATURA DE ESTABLECIMIENTO  | 35         | 35        | 35     |
| TIEMPO DE SUBIDA $t_r$ (s)  | 18,3       | 65        | 9,8    |
| TIEMPO DE ESTABLECIMIENTO $t_s$ (s)   | 63         | 65        | 25,75  |

**Tabla 18: Características de respuestas de los controladores PID simulados sin ruido al incremento de 3°C.**

La figura 4.12 muestra el voltaje RMS aplicado a la planta con simulación sin ruido de los controladores PID discretos y difuso, donde el PID posicional inicia con un voltaje RMS de 113.6V y permanece constante durante 19.5 segundos después baja a 38.15V para luego permanecer constante 44.8V aproximadamente, mientras el PID de velocidad inicia en 0V hasta alcanzar un voltaje máximo de 68.72V, posteriormente baja a



44.8V y permanece constante, mientras el PID difuso tiene un voltaje subamortiguado conmutada desde 0 a 113.6V, hasta llegar a 44.8V y permanecer constante.



**Figura 4.12: Voltaje RMS Aplicado a la planta por PID simulados sin ruido.**

La tabla 19 muestra los resultados de voltaje, corriente y potencia RMS de consumo de la planta desde 20°C hasta alcanzar los 35°C donde el controlador difuso posee menor consumo de energía con una potencia de 797.47 vatios.

| PID SIMULADOS SIN RUIDO |            |           |          |
|-------------------------|------------|-----------|----------|
| PARAMETROS              | POSICIONAL | VELOCIDAD | DIFUSO   |
| VRMS PROMEDIO (V)       | 55,0187    | 53,5059   | 52,8312  |
| IRMS PROMEDIO (A)       | 15,7196    | 15,2874   | 15,0946  |
| POTENCIA (W)            | 864.8720   | 817.9660  | 797.4658 |

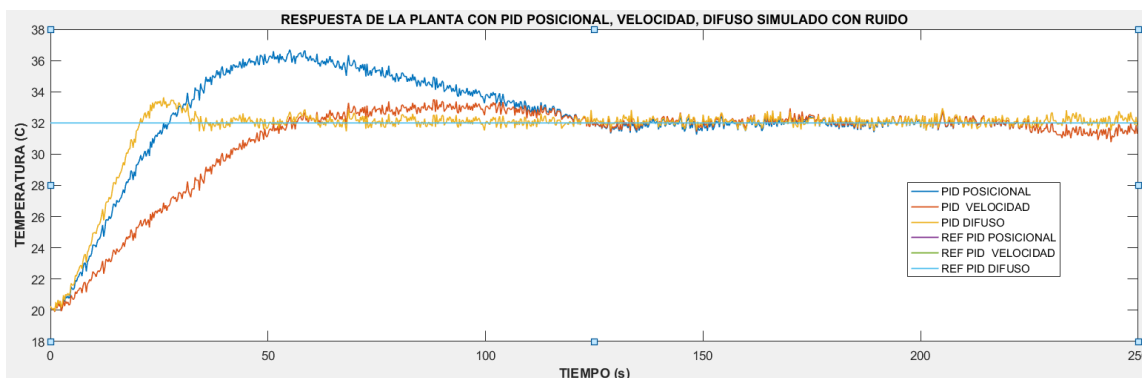
**Tabla 19: Voltaje y Corriente RMS de la planta con PID simulados sin ruido.**

Considerando los resultados de las tablas 17, 18, 19 se determina que el controlador difuso tiene el tiempo de establecimiento más rápido de 32.56 segundos para alcanzar la temperatura de 32°C, al realizar el incremento de 3°C para alcanzar 35°C el tiempo de establecimiento es de 25.75 segundos, y la potencia de consumo promedio de 797.4658 vatios en la simulación en condiciones ideales sin ruido durante 7.5 minutos.

#### 4.3.2 Comparación de PID discretos y difuso con ruido.

La figura 4.13 muestra la comparación entre las señales de temperatura de la planta con los controladores discretos y difuso simulados con ruido

cuando la temperatura de referencia es de 32°C, donde se puede observar que la temperatura de pico del PID posicional es el mayor, mientras que el tiempo de establecimiento del controlador PID difuso es más rápido, además el PID posicional y velocidad poseen tiempo de establecimiento similares.



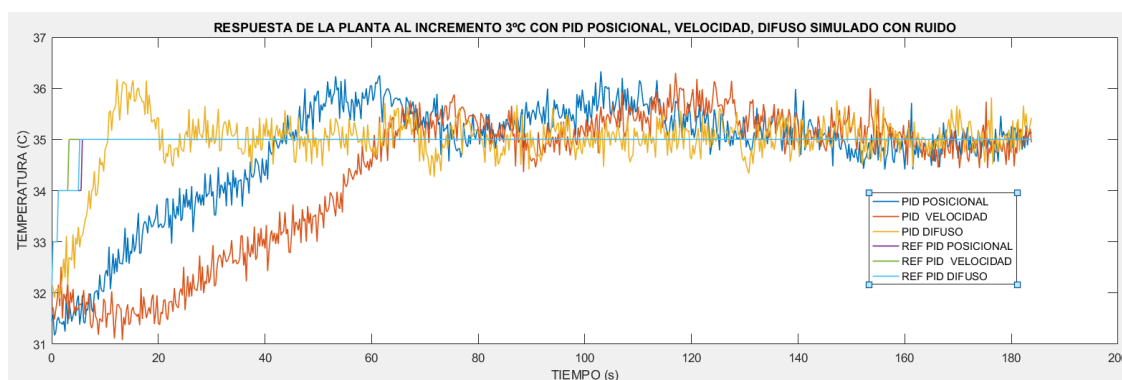
**Figura 4.13: Respuestas de los PID simulados con ruido.**

La tabla 20 muestra los resultados obtenidos en la simulación de los controladores discreto y difuso con ruido, cuando inicia con 20°C hasta alcanzar la temperatura de referencia de 32°C, donde la temperatura pico del PID posicional de 36.7°C con sobreimpulso de 39.158% es la más alta, mientras la temperatura pico de los PID velocidad y difuso alcanzan el mismo niveles parecidos con una diferencia de 0.61°C y el tiempo de establecimiento de 39.75 segundos del PID difuso es el más rápido con respecto a los otros controladores.

| PID SIMULADOS CON RUIDO          |            |           |        |
|----------------------------------|------------|-----------|--------|
| Temperatura Inicial =20°C        |            |           |        |
| Temperatura de referencia =32°C  |            |           |        |
| Características de respuesta     | POSICIONAL | VELOCIDAD | DIFUSO |
| TEMPERATURA PICO (°C)            | 36,7       | 33,37     | 33,98  |
| SOBREIMPULSO MP (%)              | 39,158     | 11,378    | 14,824 |
| TIEMPO DE SUBIDA tr (s)          | 26,5       | 66,25     | 20,5   |
| TIEMPO PICO tp (s)               | 55         | 110,3     | 26,25  |
| TIEMPO DE ESTABLECIMIENTO ts (s) | 129.3      | 129       | 39,75  |

**Tabla 20: Características de respuestas de los PID simulados con ruido.**

La figura 4.14 muestran las señales de temperatura de la planta con los controladores simulados con ruido al realizar el incremento de 3°C de manera escalonada hasta alcanzar los 35°C, donde el PID posicional, velocidad y difuso no sobrepasan la temperatura de 36°C, además el PID difuso tiene el tiempo de establecimiento más rápido, mientras que los PID posicional y velocidad tienen el tiempo de establecimiento similares.



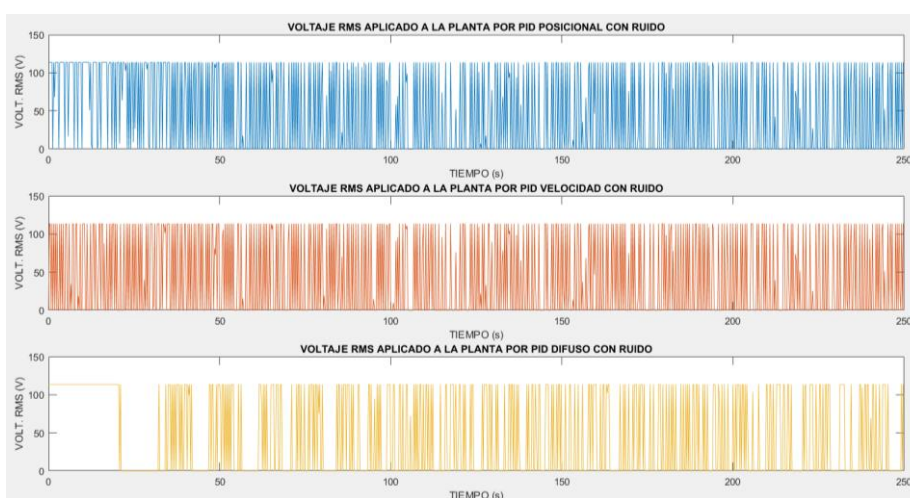
**Figura 4.14: Respuestas de los PID simulados con ruido al incremento de 3°C.**

La tabla 21 muestra los resultados obtenidos al realizar el incremento de manera escalonada de 3°C a la temperatura de referencia de 32°C para alcanzar la temperatura de 35°C, donde el PID difuso tiene el tiempo de establecimiento más rápido de 26.5 segundos respecto a los PID posicional y velocidad, mientras que la temperatura de establecimiento de los tres PID varía entre valores de 34.5°C a 35.5°C, debido al ruido incorporado en la simulación que permite obtener resultados semejantes a los reales.

| PID SIMULADOS CON RUIDO   |            |           |        |
|---|------------|-----------|--------|
| Incremento de 3°C a la Temperatura de 32°C para alcanzar los 35°C de referencia |            |           |        |
| Características de respuesta  | POSICIONAL | VELOCIDAD | DIFUSO |
| TEMPERATURA DE ESTABLECIMIENTO(°C)  | 35±0,5     | 35±0,5    | 35±0,5 |
| TIEMPO DE SUBIDA $t_r$ (s)  | 45         | 83,8      | 8,8    |
| TIEMPO DE ESTABLECIMIENTO $t_s$ (s)   | 115,8      | 129,8     | 26,5   |

**Tabla 21: Características de respuestas de los controladores PID simulados con ruido al incremento de 3°C.**

La figura 4.15 muestra el voltaje RMS aplicado a la planta por los PID discretos y difuso simulados con ruido, donde el PID posicional tiene conmutaciones de voltaje entre 0 a 113.6V larga y corta al inicio provocando un sobreimpulso de temperatura alto, mientras el PID velocidad tiene conmutaciones entre 0 a 113.6V cortas provocando un sobreimpulso menor, además el PID difuso tiene una conmutación de voltaje entre 0V a 113.6V muy larga al inicio provocando llegar rápido a la temperatura de referencia.



**Figura 4.15: Voltaje RMS Aplicado a la planta por PID simulados con ruido.**

La tabla 22 muestra los resultados de voltaje, corriente y potencia RMS de consumo de la planta, desde una temperatura inicial de 20°C hasta alcanzar los 35°C con los controladores PID discreto y difuso con ruido, donde el controlador de velocidad posee menor consumo de potencia de 592.36 vatios y muy cercano el controlador difuso posee una potencia de 627.38 vatios y se establece una diferencia de 35.0182 vatios entre ambos.

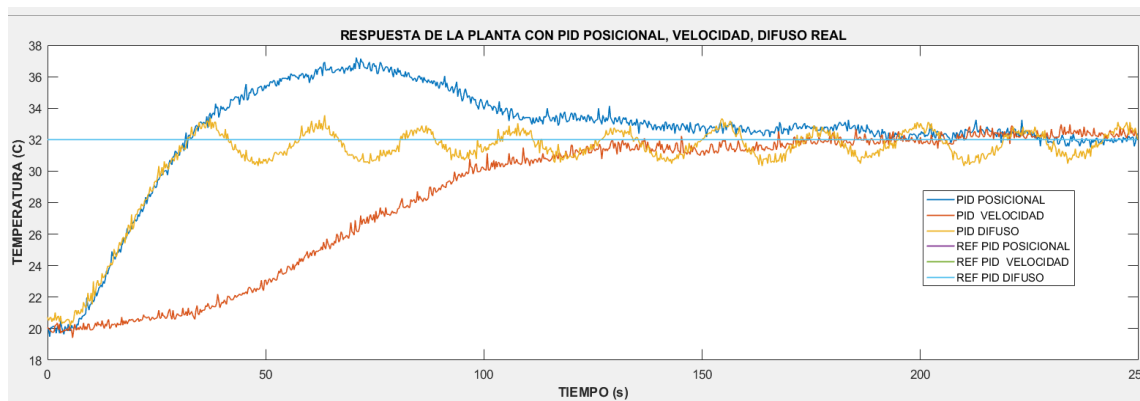
| PID SIMULADOS CON RUIDO |            |           |          |
|-------------------------|------------|-----------|----------|
| PARAMETROS              | POSICIONAL | VELOCIDAD | DIFUSO   |
| VRMS PROMEDIO (V)       | 48,1449    | 45,5332   | 46,8598  |
| IRMS PROMEDIO (A)       | 13,7557    | 13,0095   | 13,3885  |
| POTENCIA (W)            | 662.2668   | 592.3642  | 627.3824 |

**Tabla 22: Voltaje y Corriente RMS de la planta con PID simulados con ruido.**

Considerando los resultados de las tablas 20, 21, 22 se determina que el controlador PID difuso simulado con ruido es la mejor opción, debido a que tiene el tiempo de establecimiento más rápido de 39.75 segundos para alcanzar la temperatura de 32°C, al realizar el incremento de 3°C para alcanzar 35°C el tiempo de establecimiento es de 26.5 segundos y la potencia de consumo promedio es de 627.38 vatios durante 7.5 minutos de simulación.

#### 4.3.3 Comparación de PID discretos y difuso reales.

La figura 4.16 muestra la comparación entre las señales de temperatura de la planta con los controladores PID discretos y difuso reales cuando la temperatura de referencia es de 32°C, donde la temperatura de pico del PID posicional es mayor, mientras que el PID difuso permanece oscilatorio entre 31 a 33°C aproximadamente y el PID velocidad posee el menor tiempo de establecimiento.



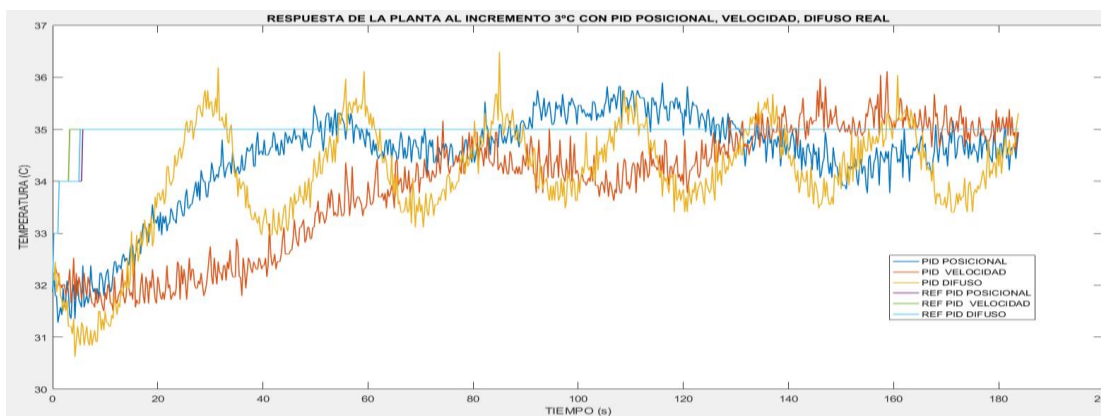
**Figura 4.16: Respuestas de los PID reales.**

La tabla 23 muestra la comparación entre las señales de temperatura de la planta con los controladores PID discretos y difuso reales, cuando inicia con 20°C hasta alcanzar la temperatura de referencia de 32°C, donde la temperatura pico del PID posicional de 37.22°C con 43.467% de sobreimpulso es el más alto, mientras que el tiempo de subida del PID posicional y difuso son similares, además el PID difuso permanece oscilatorio entre 31 y 33°C aproximadamente desde los 52 segundos.

| PID REALES                       |            |           |         |
|----------------------------------|------------|-----------|---------|
| Temperatura Inicial =20°C        |            |           |         |
| Temperatura de referencia =32°C  |            |           |         |
| Características de respuesta     | POSICIONAL | VELOCIDAD | DIFUSO  |
| TEMPERATURA PICO (°C)            | 37,22      | 32,89     | 33,55   |
| SOBREIMPULSO MP (%)              | 43,467     | 7,45      | 12,942  |
| TIEMPO DE SUBIDA tr (s)          | 31,5       | 154,8     | 32      |
| TIEMPO PICO tp (s)               | 70,75      | 167,5     | 63,5    |
| TIEMPO DE ESTABLECIMIENTO ts (s) | 188,5      | 168       | 52(osc) |

**Tabla 23: Características de respuestas de los PID reales.**

La figura 4.17 muestran los resultados al realizar el incremento de 3°C de manera escalonada hasta alcanzar los 35°C, donde el PID posicional real tiene el tiempo de establecimiento más rápido respecto al PID velocidad real, mientras el PID difuso real mantiene una oscilación.



**Figura 4.17: Respuestas de los PID reales al incremento de 3°C.**

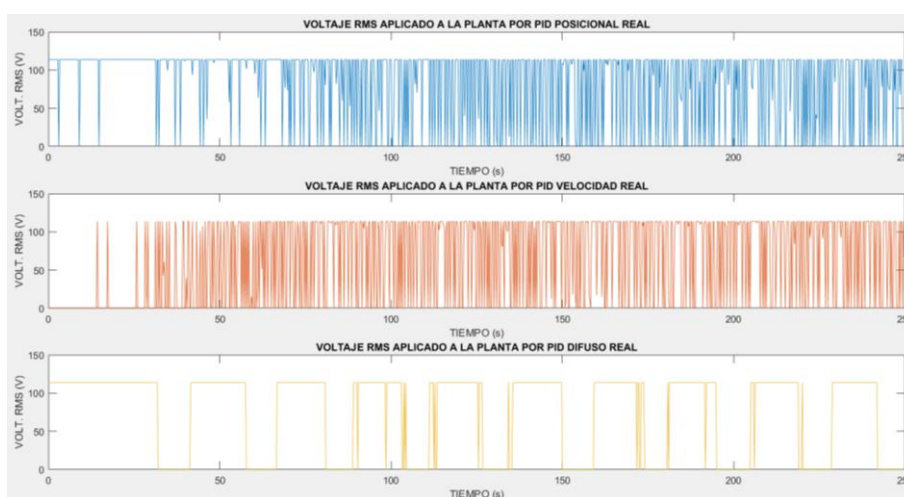
La tabla 24 muestra los resultados al realizar el incremento de 3°C de manera escalonada a los 32°C hasta alcanzar la temperatura de 35°C, donde la temperatura de establecimiento aproximado del PID posicional real varía entre 34 a 35.15°C desde los 126 segundos, el PID velocidad real varía entre 34.85 a 36°C desde los 134 segundos y el PID difuso real oscila entre 33.5 y 35.8°C desde los 40 segundos.

La figura 4.18 muestra el voltaje RMS aplicado a la planta con PID reales, donde el PID posicional tiene conmutaciones desde 0 a 113.6V en periodos largos y muy cortas que determina que tenga sobreimpulso alto,

mientras el PID velocidad al inicio permanece en 0V por un periodo largo y posteriormente conmutaciones desde 0 a 113.6V con periodos cortos provocando un tiempo de establecimiento lento respecto a los otros controladores, además el PID difuso tiene conmutaciones entre 0 a 113.6V, con periodos más largos provocando que la temperatura oscile alrededor de la referencia.

| PID REALES  |            |           |                |
|---|------------|-----------|----------------|
| Incremento de 3°C a la Temperatura de 32°C para alcanzar los 35°C de referencia |            |           |                |
| Características de respuesta  | POSICIONAL | VELOCIDAD | DIFUSO         |
| TEMPERATURA ASENTAMIENTO  | 34-35,15   | 34,85-36  | 33,5-35,8(osc) |
| TIEMPO DE SUBIDA $t_r$ (s)  | 45,8       | 74,5      | 25             |
| TIEMPO DE ESTABLECIMIENTO $t_s$ (s)   | 123        | 134       | 40(osc)        |

**Tabla 24: Características de respuestas de los controladores PID reales al incremento de 3°C.**



**Figura 4.18: Voltaje RMS Aplicado a la planta por PID real.**

La tabla 25 muestra los resultados de voltaje, corriente y potencia RMS de consumo de la planta desde 20°C hasta alcanzar los 35°C donde el controlador velocidad posee el menor consumo de potencia de 1604.5189 vatios y muy cercano el controlador difuso posee una potencia de 1806.9193 vatios y se establece una diferencia de 202.4004 vatios entre ambos.

| <b>.PID REALES</b> |                   |                  |               |
|--------------------|-------------------|------------------|---------------|
| <b>PARAMETROS</b>  | <b>POSICIONAL</b> | <b>VELOCIDAD</b> | <b>DIFUSO</b> |
| VRMS PROMEDIO (V)  | 86,2148           | 74,939           | 79,525        |
| IRMS PROMEDIO (A)  | 24,6328           | 21,411           | 22,7214       |
| POTENCIA (W)       | 2123.7192         | 1604.5189        | 1806.9193     |

**Tabla 25: Voltaje y Corriente RMS de la planta con PID reales.**

Considerando los resultados de las tablas 23, 24, 25 se determina que el controlador PID difuso real es la mejor opción, debido a que alcanza más rápido la temperatura de referencia de 32 y 35°C en aproximadamente 52 y 40 segundos respectivamente, además tiene una potencia media de consumo de 1806.92 vatios, pero manteniendo una oscilación de temperatura de  $\pm 1.5^{\circ}\text{C}$ , que es imperceptible para el usuario.



## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES.

- El circuito embebido desarrollado con la tarjeta STM32 Discovery y la interfaz gráfica desarrollada en LabView, permitió la adquisición de datos de temperatura del agua de la planta (ducha eléctrica) en un archivo de Excel a una frecuencia de muestreo de 250 milisegundos y mediante la aplicación de identificación de sistemas Ident de Matlab se determinó las funciones de transferencia, donde se eligió la función de transferencia discreta con dos polos y un cero por tener el mayor porcentaje de estimación de 96.47%.
- El circuito de potencia desarrollado con dos triacs BTA41 debido a la corriente máxima de consumo de 43.3A dado por el fabricante de la ducha eléctrica, permitió el control por ángulo de fase para variar la potencia entregada a la planta y por lo tanto el control de temperatura del sistema de transferencia de calor.
- El sistema embebido desarrollado permitió la adquisición de datos mediante bluetooth y el control de temperatura del sistema de transferencia de calor, incorporando en la programación los algoritmos de adquisición datos, PID discretos o difuso respectivamente de manera individual para no causar sobrecarga de procesamiento en la tarjeta STM32 Discovery.
- La programación de los algoritmos de control PID discreto y difuso mediante Matlab-Simulink en el sistema embebido permitió controlar la temperatura del sistema de transferencia de calor (ducha eléctrica), donde se establece el PID difuso como la mejor opción, debido a que alcanza más rápido la temperatura de referencia de 32 y 35 grados centígrados en aproximadamente 52 y 40 segundos respectivamente, con una potencia media de consumo de 1806.92 vatios, pero manteniendo una oscilación de temperatura de  $\pm 1.5^{\circ}\text{C}$  sobre la referencia que resulta imperceptible para el usuario, además se considera como antecedente que el PID difuso simulado con o sin ruido tiene el menor tiempo de establecimiento, determinado que el sistema embebido con PID

difuso provoca un ahorro de consumo de energía variable debido también a la temperatura inicial del agua.

- El voltaje RMS aplicado a la planta por los PID discretos y difuso reales tienen conmutaciones entre 0 a 113.6V, donde el PID difuso desataca por tener periodos de conmutación más largos respecto a los PID de posición y velocidad que provocan una señal de temperatura con oscilaciones de  $\pm 1.5^{\circ}\text{C}$  sobre la referencia, este comportamiento es provocado por el caudal de agua variable de la ducha eléctrica.

## **RECOMENDACIONES.**

- Para realizar la adquisición de datos y pruebas de funcionamiento en la planta es necesario procurar tener el flujo de agua constante, esto permitirá que el sistema embebido tenga mejor desempeño y las señales más parecidas a las obtenidas en la simulación con ruido.
- Para evaluar el desempeño del sistema embebido antes de conectar a la ducha eléctrica, se debe conectar una lámpara incandescente al circuito de potencia para observar las señales de control por ángulo de fase mediante un osciloscopio.
- El circuito de potencia del sistema embebido tiene dos triacs de 40 amperios, puede ser remplazado por un triac de 60A.
- El circuito de potencia que utiliza el control por ángulo de fase para disparar los tiristores, puede ser remplazado para futuros trabajos por el control por modulación de ancho de pulso (PWM) que mejora el factor de potencia, armónicos, calidad de corriente y utiliza dispositivos semiconductores de conmutación forzada como BJTs, MOSFETs, IGBTs.
- Para mejorar el desempeño del sistema embebido con PID difuso para futuros trabajos se puede variar el número de reglas que permitirá mejorar el control de temperatura sobre la planta.

## BIBLIOGRAFÍA

[1] G. Espín y B. Remache, "Diseño y construcción de un prototipo de ducha eléctrica de alta eficiencia", Tesis de Pregrado, Escuela Politécnica Nacional, Quito, 2011.

[2] L. Ayala y R. Donoso, "Diseño, Modelación, Simulación e Implementación de un Sistema de Control de Temperatura para la Ducha Eléctrica", Tesis de Pregrado, Escuela Superior Politécnica de Chimborazo, Riobamba, 2015.

[3] A. Acosta, "Sistema de Control Automático de una ducha eléctrica mediante Reconocimiento de la Voz para personas con discapacidad en las Extremidades Superiores", Tesis de Pregrado, Universidad Técnica de Ambato, Ambato, 2015.

[4] J. Ortiz, "Diseño y Análisis comparativo de un sistema de control de temperatura con lógica difusa vs control proporcional integral derivativo (PID) en un prototipo de una torre de enfriamiento en una planta de procesos", Tesis de Posgrado. Escuela Superior Politécnica del Litoral, Guayaquil. 2016.

[5] C. Carrasco y G. Alvares. (2014). Diseño e implementación de un PID Profesional y un PID Difuso utilizando un microcontrolador PIC18F4550. ICIMAF. Recuperado de: [https://www.researchgate.net/publication/273439309\\_Diseño\\_e\\_implementación\\_de\\_un\\_PID\\_Profesional\\_y\\_un\\_PID\\_Difuso\\_utilizando\\_un\\_microcontrolador\\_PIC18F4550-2014](https://www.researchgate.net/publication/273439309_Diseño_e_implementación_de_un_PID_Profesional_y_un_PID_Difuso_utilizando_un_microcontrolador_PIC18F4550-2014)

[6] J. Hernández, J. Monsalve, y E. Rico. (2005). Comparación del control de velocidad de un motor de DC utilizando lógica difusa vs PID tradicional. CINTEX, 1(11), 72-18. Recuperado de: <http://www.pascualbravo.edu.co:5056/cintexpb/index.php/cintex/article/view/144/146>

[7] S. Martínez, E. Tarifa, y A. Núñez, (2012). Comparación de desempeño entre Controladores Fuzzy y Controladores Proporcionales. Avances en Ciencias e Ingeniería, 3(3), 91-101. Recuperado de: [http://www.exeedu.com/publishing.cl/av\\_cienc\\_ing/2012/Vol3/Nro3/10-ACI1094-11-full.pdf](http://www.exeedu.com/publishing.cl/av_cienc_ing/2012/Vol3/Nro3/10-ACI1094-11-full.pdf)

- [8] Y. Cengel, *Transferencia de calor y masa*, 3rd ed. México: Mcgraw-Hill Interamericana Editores, 2007, pp. 17-30.
- [9] D. Hart, *Electrónica de potencia*, 1st ed. Madrid: Pearson educación, 2001, pp. 177-198.
- [10] K. OGATA, *Sistemas de Control en tiempo discreto*. Editorial Prentice Hall, México, 1996.
- [11] J. Espinosa, J. Vandewalle and V. Wertz, *Fuzzy logic, identification, and predictive control*. London: Springer, 2004, pp. 124-128.
- [12] P. Escamilla, "A Novel Design and Tuning Procedure for PID Type Fuzzy Logic Controllers", First International IEEE Symposium "Intelligent Systems", pp. 36-41, 2002.

## **ANEXOS**

ANEXO 1: CARACTERISTICAS DE STM32DISCOVERY

ANEXO 2: LIBRERÍA WAIJUNG PARA SIMULINK

ANEXO 3: CARACTERISTICAS TÉCNICAS DE DUCHA ELECTRICA LORENZETTI.

ANEXO 4: CIRCUITO DEL SISTEMA EMBEBIDO.

ANEXO 5: CIRCUITOS IMPRESOS

ANEXO 6: DIAGRAMA DE FLUJO DEL DISPARO DE TIRISTOR CON ATMEGA328P

ANEXO 7: DIAGRAMA DE FLUJO DE ADQUISICIÓN DE DATOS

ANEXO 8: PROGRAMA CAPTURA DE DATOS LABVIEW

ANEXO 9: PASOS PARA CAPTURA DE DATOS

ANEXO 10: PASOS PARA IDENTIFICAR PLANTA

ANEXO 11: PASOS PARA IDENTIFICAR CONSTANTES DE PID

ANEXO 12: PROGRAMA PARA CALCULAR CONSTANTES DE PID DISCRETO Y DIFUSO

ANEXO 13: DIAGRAMA DE FLUJO DEL PID DE FORMA POSICIONAL

ANEXO 14: DIAGRAMA DE FLUJO DEL PID FORMA DE VELOCIDAD

ANEXO 15: DISEÑO DEL PID DIFUSO

ANEXO 16: DIAGRAMA DE FLUJO DEL PID DIFUSO

ANEXO 17: SIMULACIÓN DEL PID POSICIONAL CON Y SIN RUIDO

ANEXO 18: SIMULACIÓN DEL PID VELOCIDAD CON Y SIN RUIDO

ANEXO 19: SIMULACIÓN DEL PID DIFUSO CON Y SIN RUIDO

ANEXO 20: GRÁFICAS DEL PID POSICIONAL, VELOCIDAD, DIFUSO REALES

ANEXO 21: COMPARACIÓN DEL PID POSICIONAL SIMULADO Y REAL

ANEXO 22: COMPARACIÓN DEL PID VELOCIDAD SIMULADO Y REAL

ANEXO 23: COMPARACIÓN DEL PID DIFUSO SIMULADO Y REAL

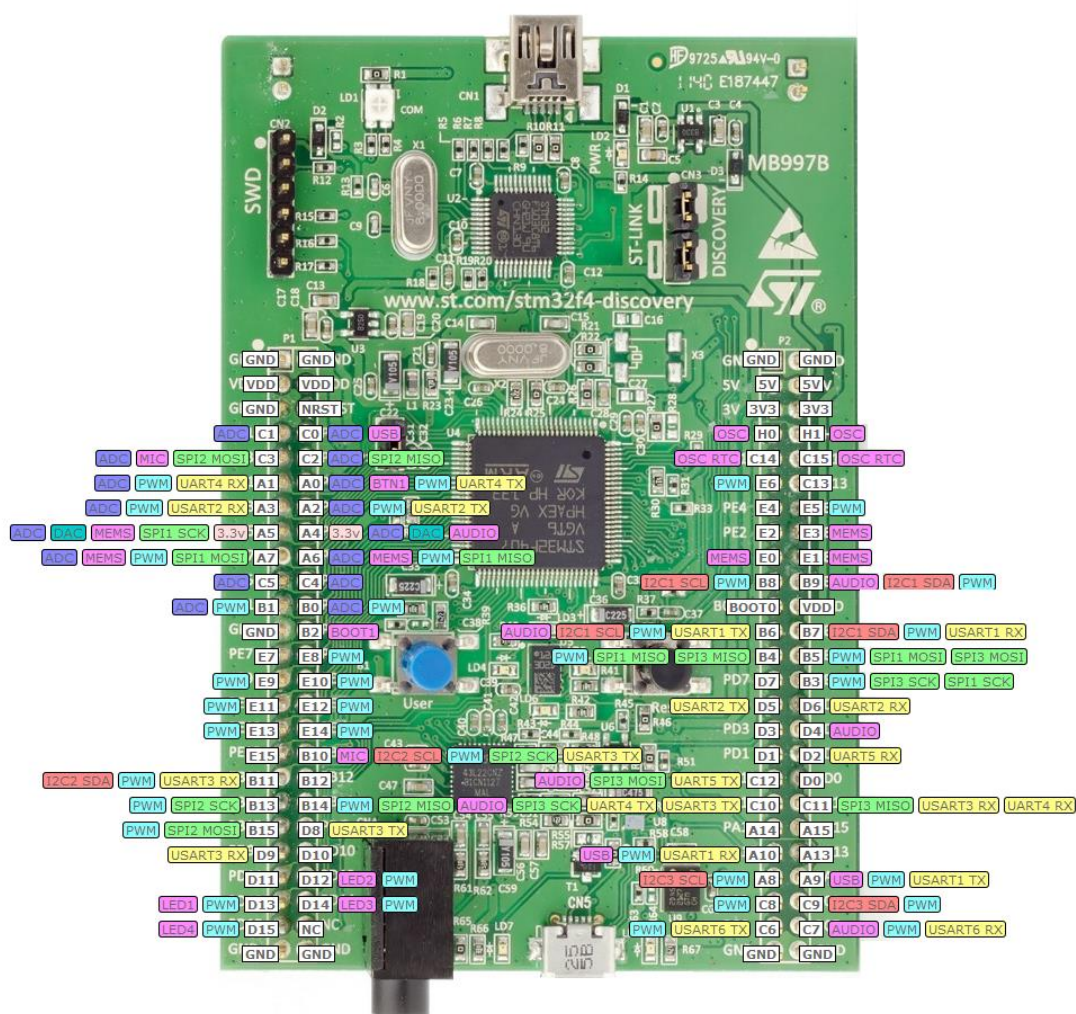
ANEXO 24: COMPARACIÓN DEL PID POSICIONAL, VELOCIDAD, DIFUSO SIMULADO Y REAL

## ANEXO 1

### CARACTERÍSTICAS DE STM32DISCOVERY

#### Función de Pines

La función de cada uno de los pines de la tarjeta Discovery se muestra en la figura



#### Características

- STM32F407VGT6 Microcontrolador de 32 Bits ARM Cortex- M4, 1Mbyte memoria Flash, 192 Kbyte RAM
- Alimentación de la placa: USB, externa 5 y 3 Vdc.
- Acelerómetro 3 ejes LIS302DL o LIS3DSH

- Micrófono digital omnidireccional MP45DT02
- DAC integrado CS43L22
- 8 Leds: LD1(rojo/verde) para comunicación USB; LD2 (red) para alimentación 3.3Vdc; leds de usuario LD3 (tomate), LD4 (verde), LD5 (rojo), LD6 (azul); 2 USB OTG Leds LD7 (verde) VBUS; LD8 (rojo) sobre corriente.
- Dos botones para usuario y reset respectivamente.

#### **Herramientas de desarrollo**

- Programación y depuración por ST-LINK/V2
- STM32CubeMX herramienta de configuración
- ARM mbed compilador
- Keil uVision compilador
- Matlab-Simulink con librerías Waijung

## ANEXO 2

# LIBRERÍA WAIJUNG PARA SIMULINK

### Instalación

**Paso 1.** Descargar la librería del link: <https://www.aimagin.com/download/>



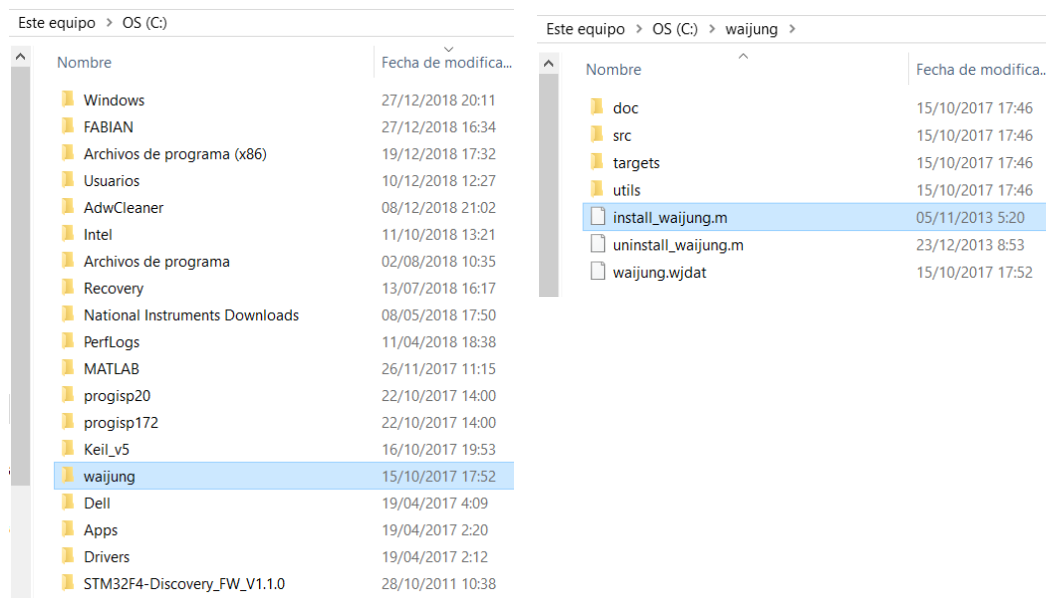
**Waijung Blockset**

STM32F4 Target for Hi-Performance & DSP Applications  
 STM32F0 Target for Generic & Economic Applications  
 nRF51422/822 Target for Ultra Low-Power Wireless (ANT) Network Applications

Release 17.03a (7 Mar 2017)

[Download Now](#) [Change Log](#) [Documentations](#)

**Paso 2.** Extraer el archivo bajado en el archivo raíz C:



| Nombre                         | Fecha de modifica...    |
|--------------------------------|-------------------------|
| Windows                        | 27/12/2018 20:11        |
| FABIAN                         | 27/12/2018 16:34        |
| Archivos de programa (x86)     | 19/12/2018 17:32        |
| Usuarios                       | 10/12/2018 12:27        |
| AdwCleaner                     | 08/12/2018 21:02        |
| Intel                          | 11/10/2018 13:21        |
| Archivos de programa           | 02/08/2018 10:35        |
| Recovery                       | 13/07/2018 16:17        |
| National Instruments Downloads | 08/05/2018 17:50        |
| PerfLogs                       | 11/04/2018 18:38        |
| MATLAB                         | 26/11/2017 11:15        |
| progisp20                      | 22/10/2017 14:00        |
| progisp172                     | 22/10/2017 14:00        |
| Keil_v5                        | 16/10/2017 19:53        |
| <b>waijung</b>                 | <b>15/10/2017 17:52</b> |
| Dell                           | 19/04/2017 4:09         |
| Apps                           | 19/04/2017 2:20         |
| Drivers                        | 19/04/2017 2:12         |
| STM32F4-Discovery_FW_V1.1.0    | 28/10/2011 10:38        |

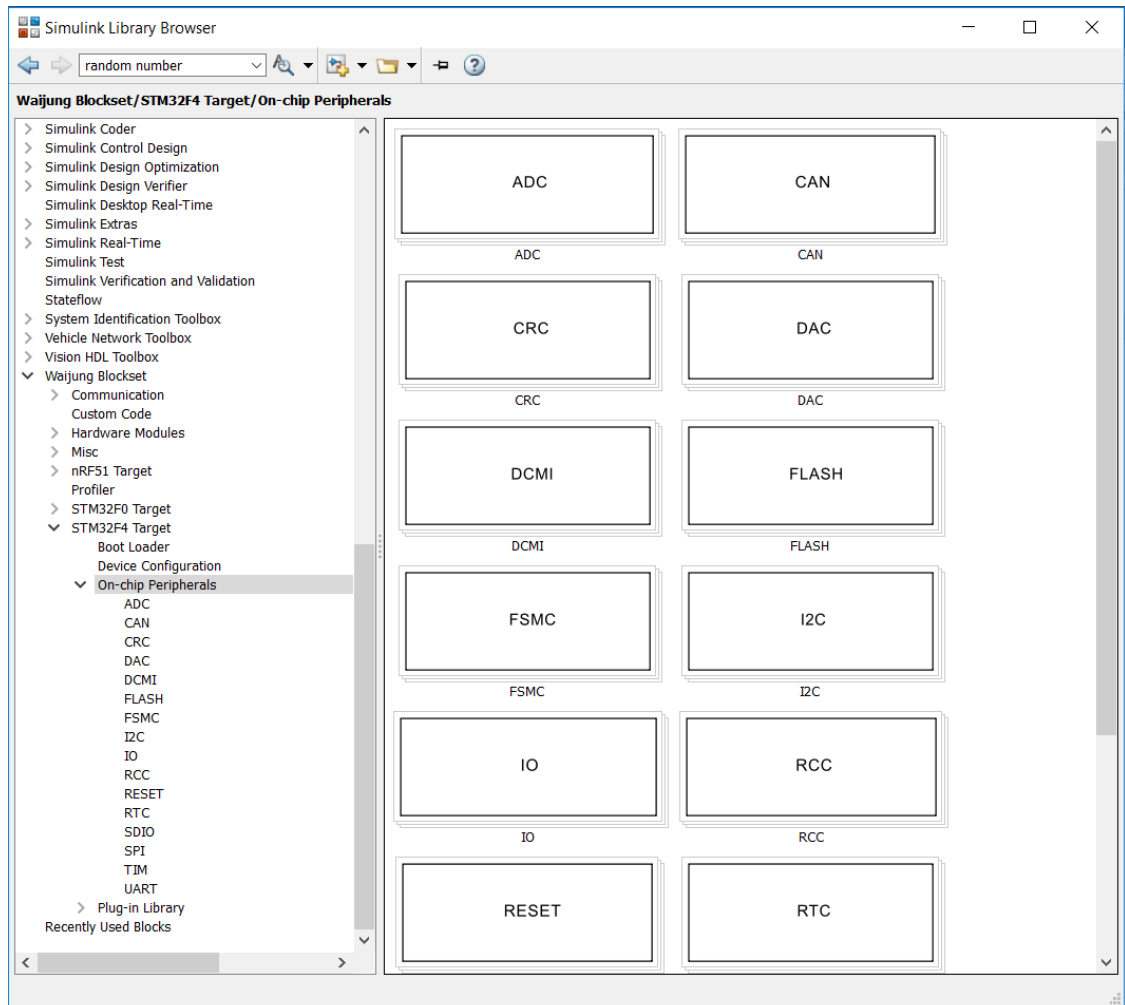
| Nombre                   | Fecha de modifica...   |
|--------------------------|------------------------|
| doc                      | 15/10/2017 17:46       |
| src                      | 15/10/2017 17:46       |
| targets                  | 15/10/2017 17:46       |
| utils                    | 15/10/2017 17:46       |
| <b>install_waijung.m</b> | <b>05/11/2013 5:20</b> |
| uninstall_waijung.m      | 23/12/2013 8:53        |
| waijung.wjdat            | 15/10/2017 17:52       |

**Paso 3.** Instalar el archivo “install\_waijung.m” abriendo en Matlab como muestra la figura y ejecutando el programa dando clic en Run.





## Bloques de trabajo



## ANEXO 3

# CARACTERÍSTICAS TÉCNICAS DE DUCHA ELECTRICA LORENZETTI

**Ducha Eléctrica**



**LORENZETTI**

### Funcionamiento

**Paso 1**  
Abrir la llave de la ducha para que el agua fría ingrese a la ducha

**Paso 2**  
La presión del agua conecta el diafragma que conecta mecánicamente los contactos eléctricos y la resistencia empuja el calentamiento del agua

**Paso 3**  
El agua caliente circula en la ducha hasta salir por el rociador para brindar el baño confortable al usuario

**Paso 4**  
Al cerrar la llave de agua de la ducha se desconectan los contactos eléctricos apagando la resistencia eléctrica



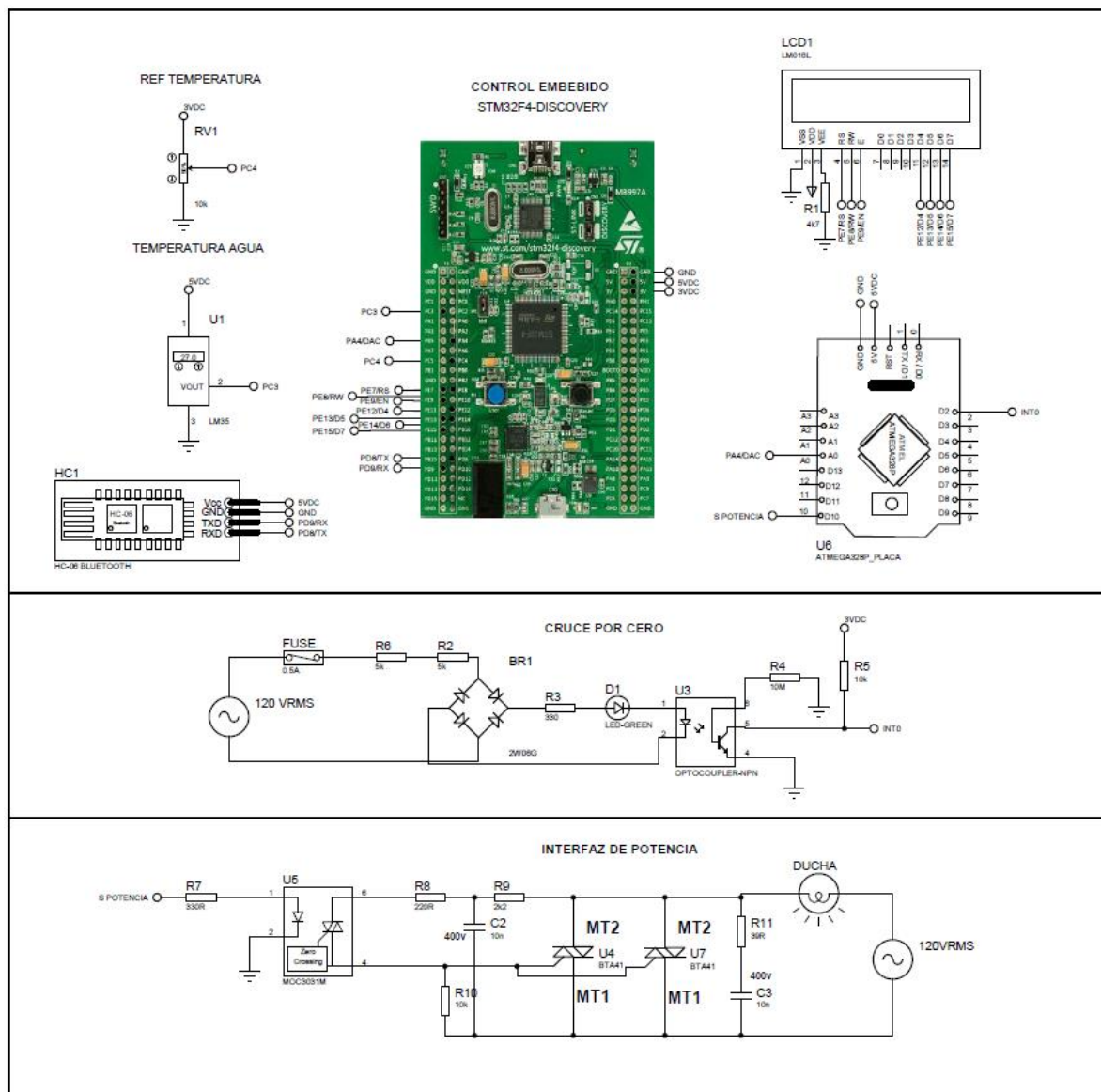
### Características técnicas

| Especificación/ Especification   |  |                             | Grado de protección/Protecting Degree: IP24  |         |         |        |
|--|--|-----------------------------|--|---------|---------|--------|
| Modelo/ Model  |  |                             | Lorenducha *   |         |         |        |
| Tensión / Tension (V~)   |  |                             | 127  |         | 220     |        |
| Potencia<br>Power<br>(Watts)   | Seletor de<br>Temperatura<br>Temperature<br>Selector | Desconec./ Off              | 0  | 0       | 0       | 0      |
|  |  | Tibio/Warm                  | 1800   | **2400  | **2400  | **2600 |
|  |  | Caliente/Hot                | **3200   | 4400    | 4400    | 4400   |
|  |  | Super caliente<br>Super Hot | 4400   | 5400    | 5400    | 6400   |
| Disyuntor o Fusible/ (Amperios)<br>Circuit Breaker or fuse ( Ampers)   |  |                             | 40   | 50      | 30      | 35     |
| Grosor de los cables (mm <sup>2</sup> )***<br>Size of the cables (AWG) |  |                             | 6<br>8   | 10<br>6 | 4<br>10 | 6<br>8 |
| Presión de funcionamiento.<br>Working Pressure.                        |  |                             | Mnima/Minimum<br>Máxima/ Max   |         |         |        |
| Conexión Hidráulica/Hydraulic conection                                |  |                             | Entrada de agua - Rosca/Water inlet - Thread 1/2" BSP  |         |         |        |
| Conexión Eléctrica/ Electric Conection                                 |  |                             | Cables Blancos - Fase/Fase White Wire - Phase/Phase<br>Fase/Neutro Phase Neutral<br>Cable verde/amarillo- puesta a tierra<br>Green/ Yellow wire - Grounding wire |         |         |        |

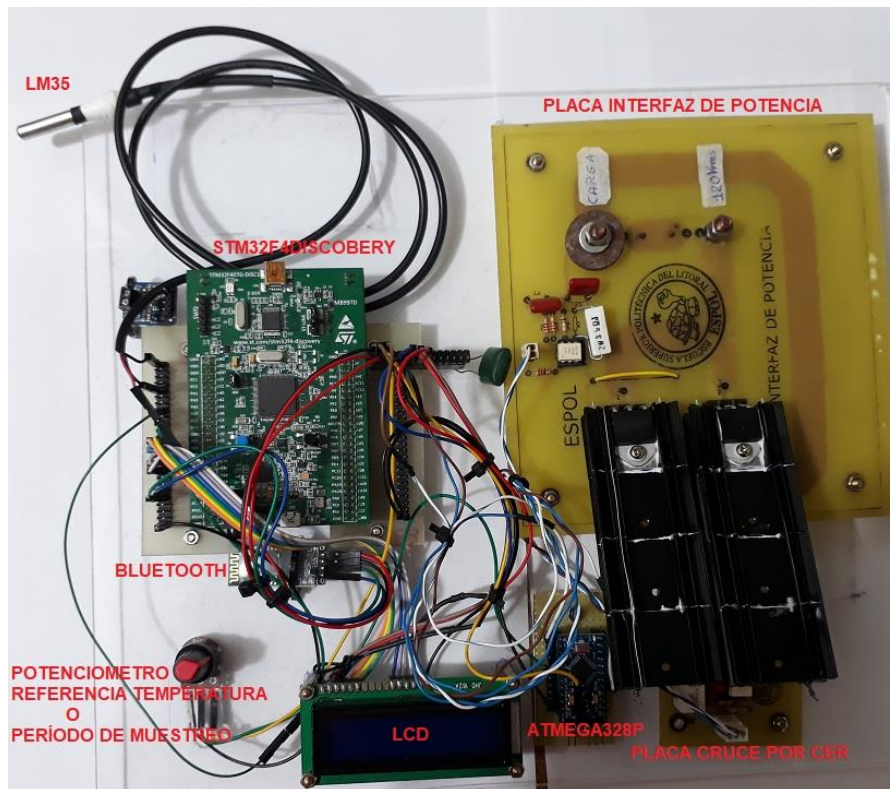
\* La Resistividad del agua suministrada para este producto a 22°C, no debe ser inferior a 1300 ohms x cm.  
 \*\* Potencia económica  
 \*\*\* Para distancias superiores a 30 m, use cables de mayor sección.  
 \*The water resistivity for this product at 22° C, Cab not be lower than 1300 ohms xcm.  
 \*\*Economic power. \*\*\*For distances over 30 meters, thicker cables must be used.

## ANEXO 4

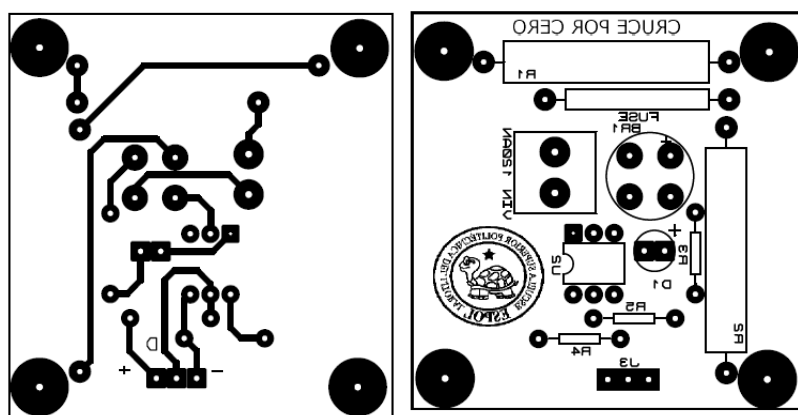
## CIRCUITO DEL SISTEMA EMBEBIDO



## SISTEMA EMBEBIDO REAL



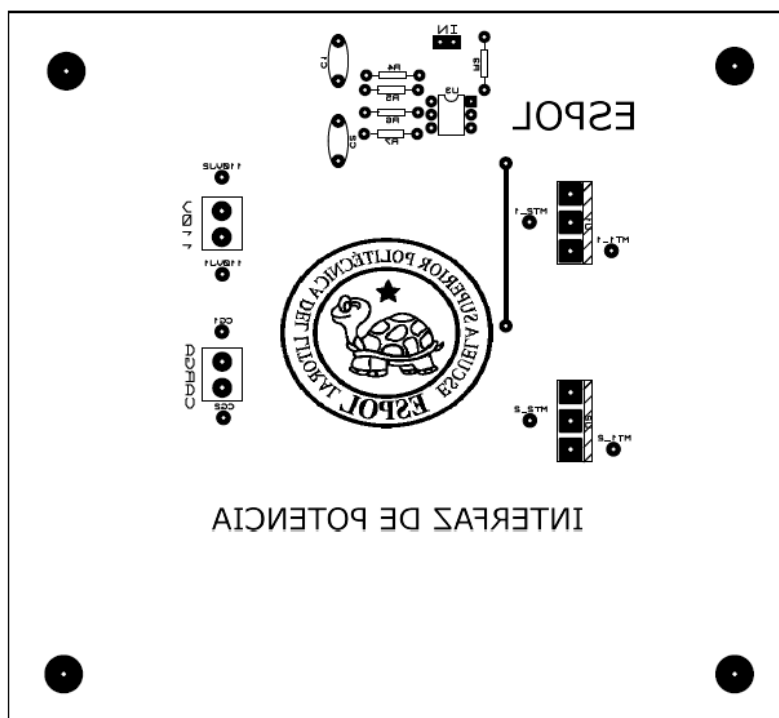
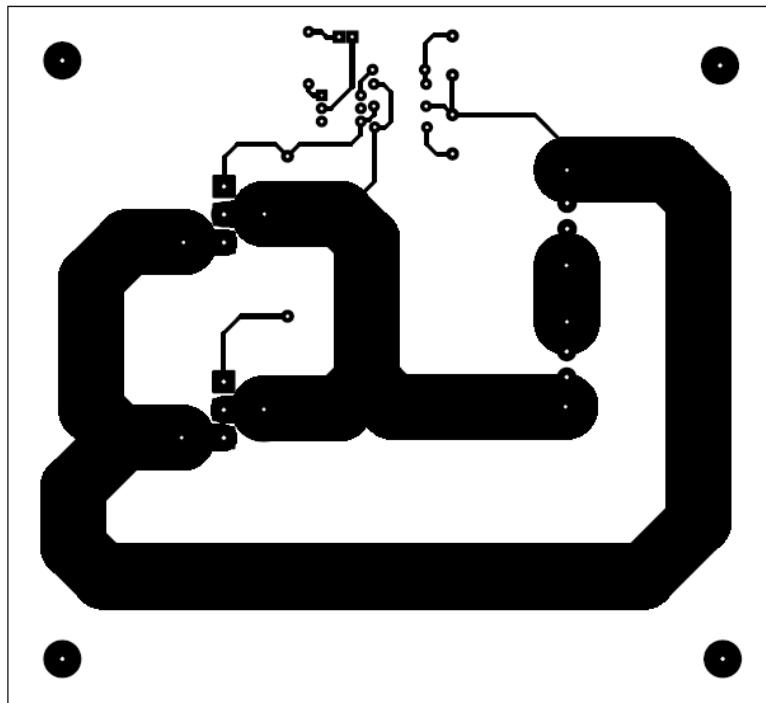
**ANEXO 5**  
**CIRCUITOS IMPRESOS**  
**CIRCUITO CRUCE POR CERO**



**PLACA TERMINADA**

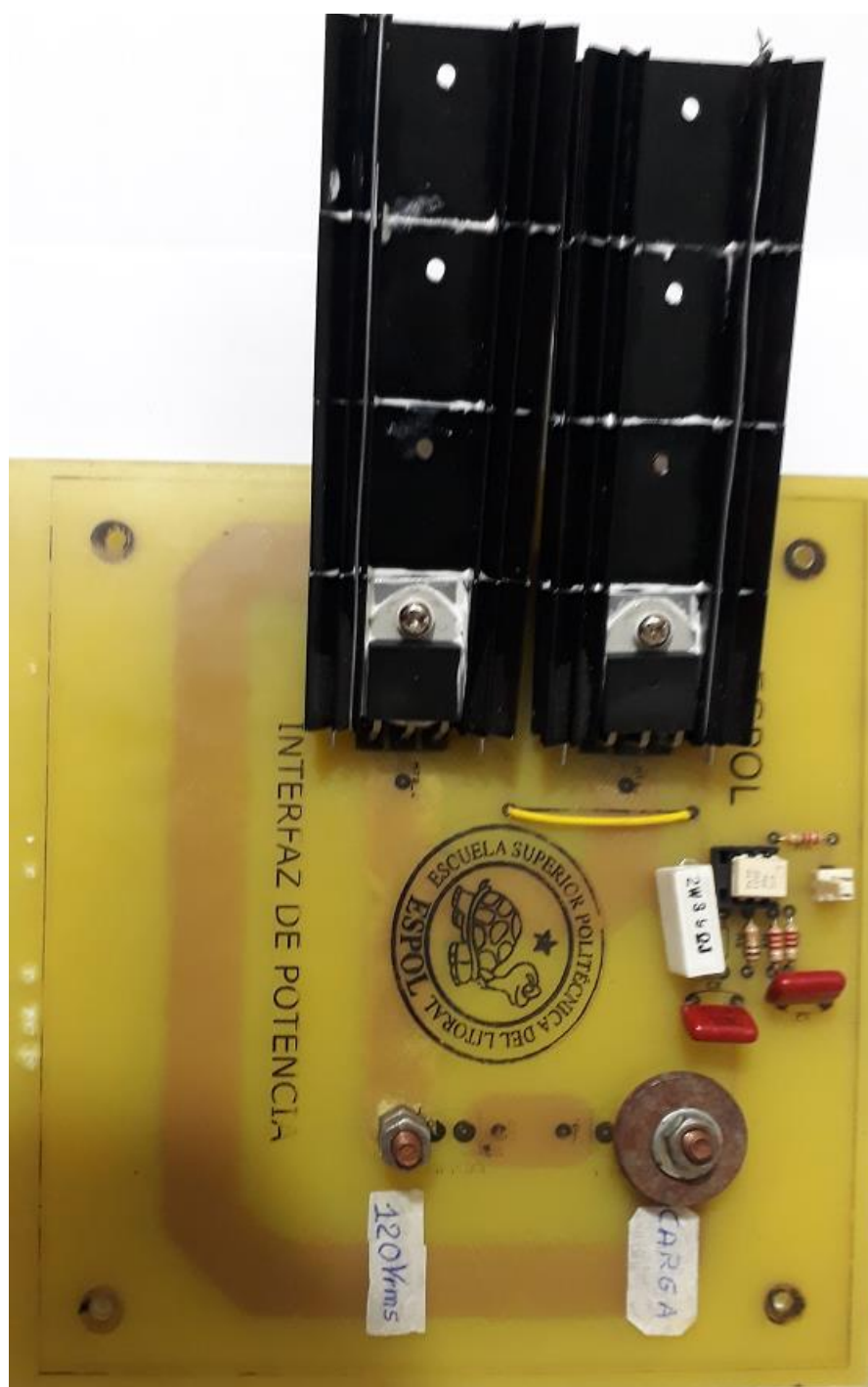


## CIRCUITO INTERFAZ DE POTENCIA



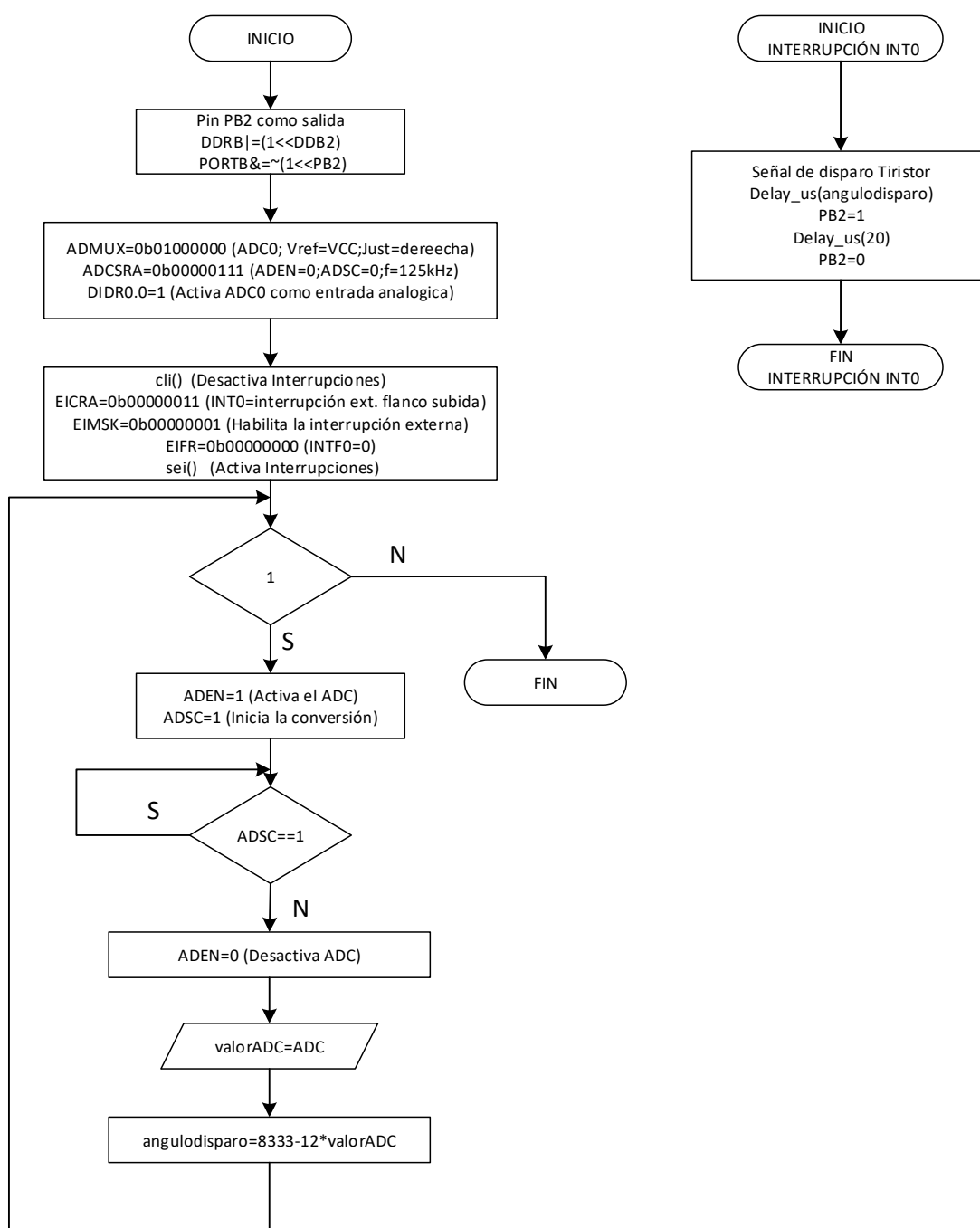


## PLACA TERMINADA





## ANEXO 6

DIAGRAMA DE FLUJO DEL DISPARO DE TIRISTOR  
CON ATMEGA328P

## PROGRAMA DE DISPARO DE TIRISTOR CON ATMEGA328P

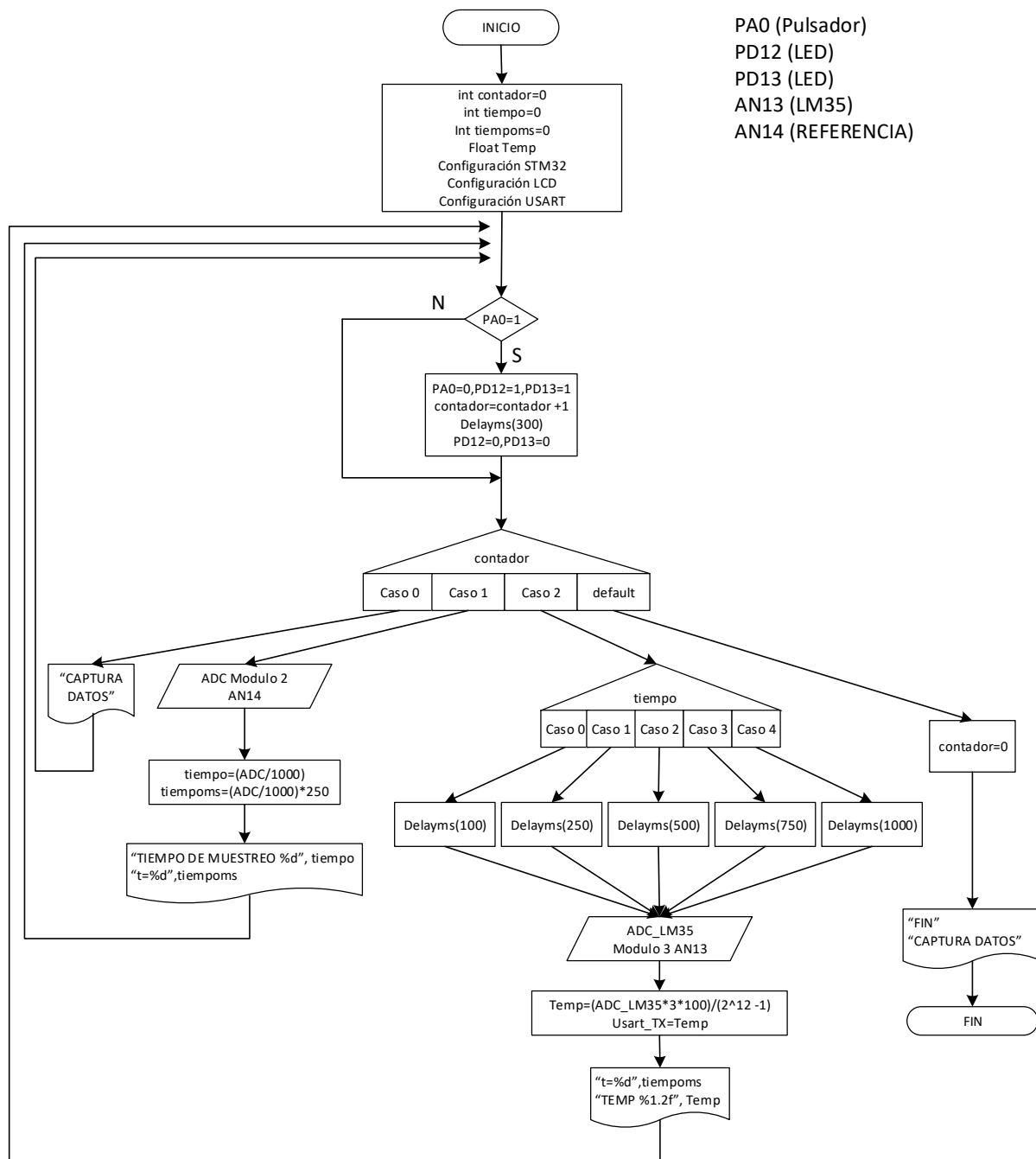
```

#define F_CPU 8000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
uint16_t valorADC=0, angulodisparo=7633;
int main(void)
{
    DDRB|=(1<<DDB2); //Pin PB0 como salida digital
    PORTB&=~(1<<PB2); //Pin PB0 se inicia apagado
    //Configuracion ADC
    ADMUX=0b01000000; // configura ADC0, justificación a la derecha, Vref=Vcc
    ADCSRA=0b00000111; // frecuencia de trabajo 125kHz
    ADCSRB=0;
    DIDR0|=(1<<0);
    //Configuracion interrupcion INT0
    cli(); //desabilita interrupciones
    EICRA=0b00000011; //interrupcion flanco de subida
    EIMSK=0b00000001; //habilita INT0
    EIFR=0b00000000; //INTF0=0 se coloca en 1 cuando se produce interrupcion
    //INT0, se coloca automaticamente en cero al ejecutar ISR
    sei(); //habilita interrupciones
    while (1) {
        ADCSRA|=((1<<ADEN)|(1<<ADSC));|
        while (ADCSRA&(1<<ADSC));
        ADCSRA&=~(1<<ADEN);
        valorADC=ADC;
        angulodisparo=8333-12*valorADC;
    }
}
ISR(INT0_vect) {
    delayMicroseconds(angulodisparo); //tiempo de espera antes que el ciclo se repita
    PORTB|=(1<<PB2); //PB0 cambia de estado, el led parpadea
    _delay_us(20);
    PORTB&=~(1<<PB2);
}

```

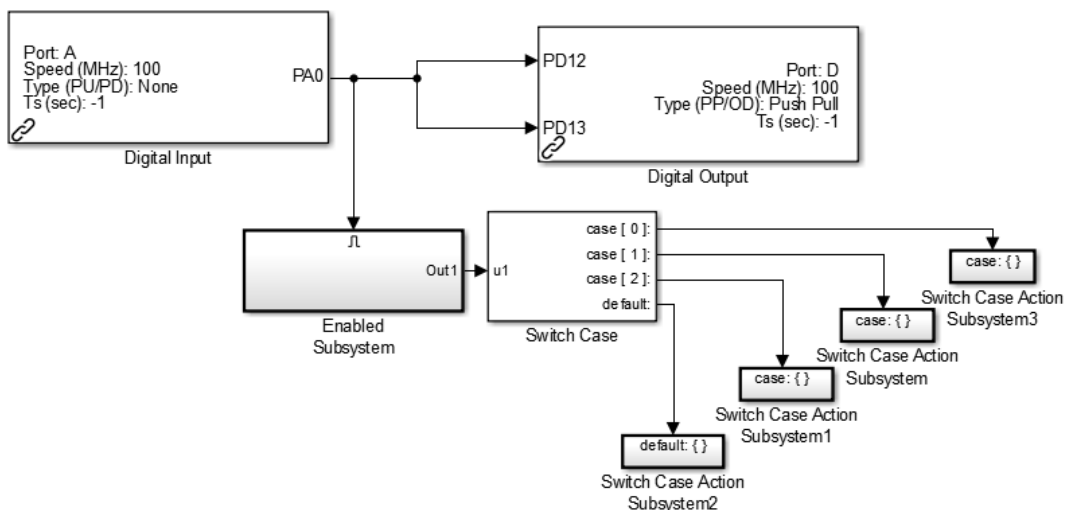
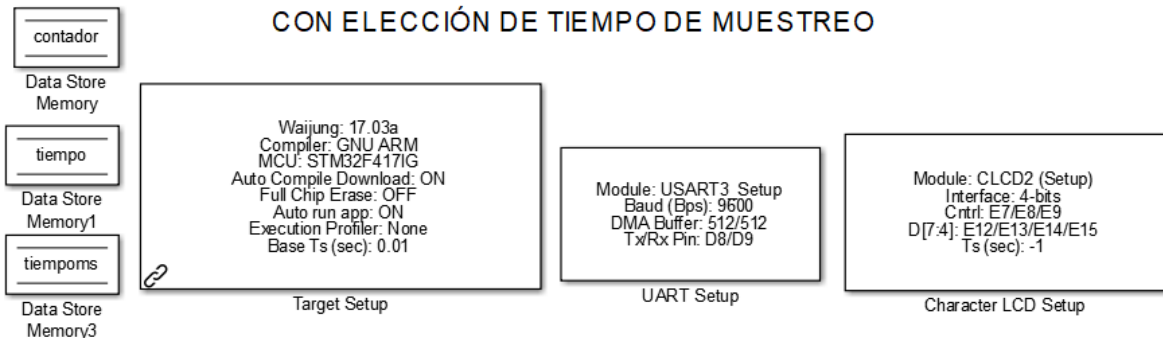
## ANEXO 7

## DIAGRAMA DE FLUJO DE ADQUISICIÓN DE DATOS

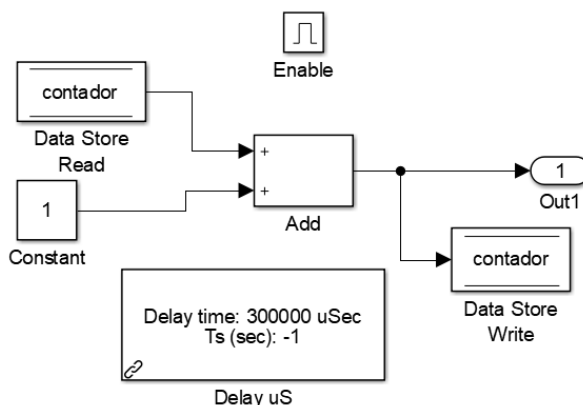


# PROGRAMA DE ADQUISICIÓN DE DATOS

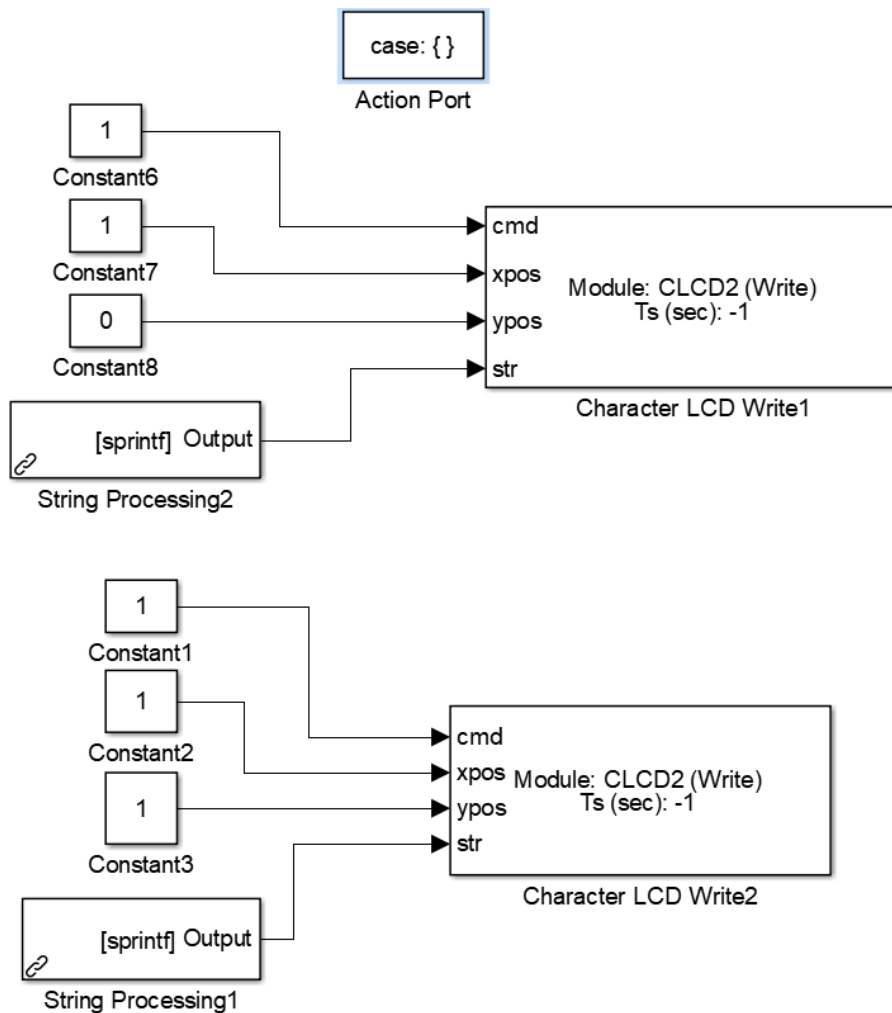
## CAPTURA DE DATOS PARA LM35 CON ELECCIÓN DE TIEMPO DE MUESTREO



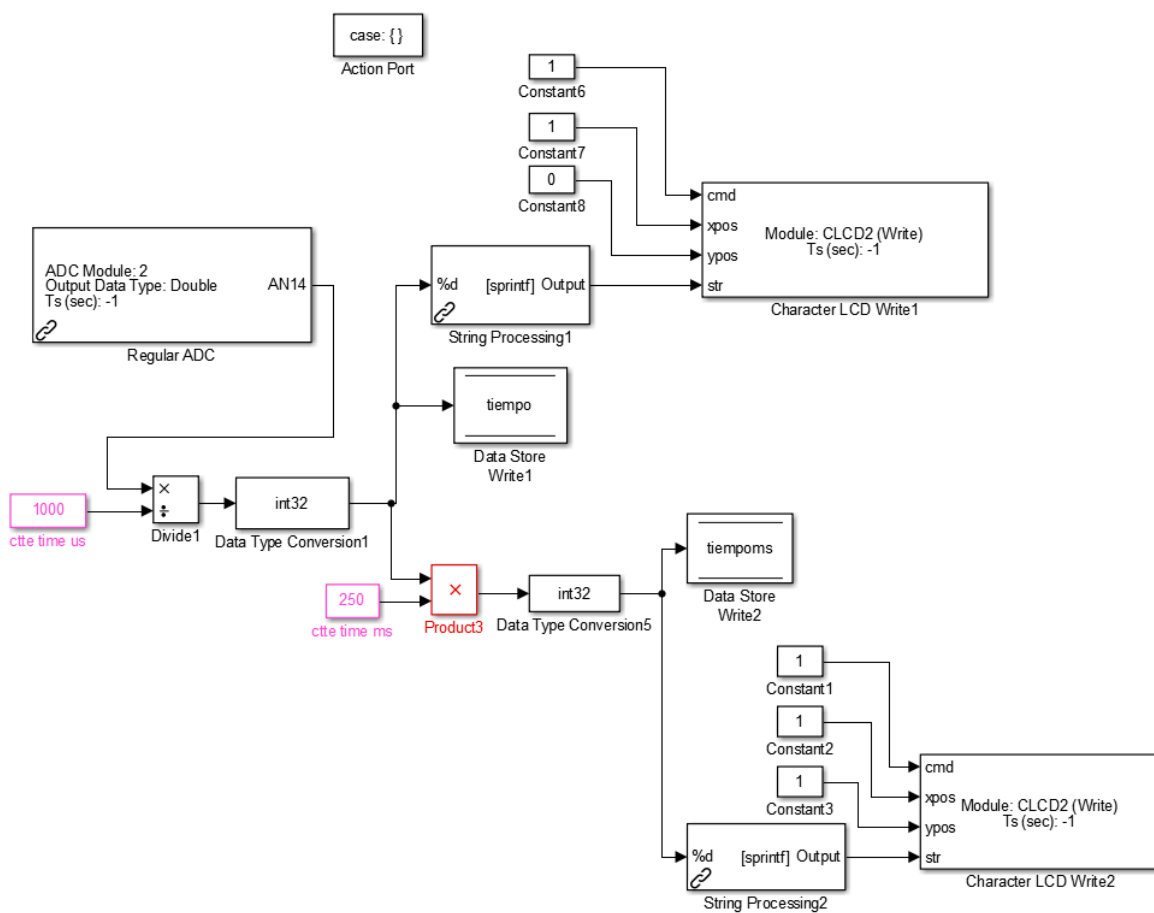
## FUNCIÓN CONTADOR



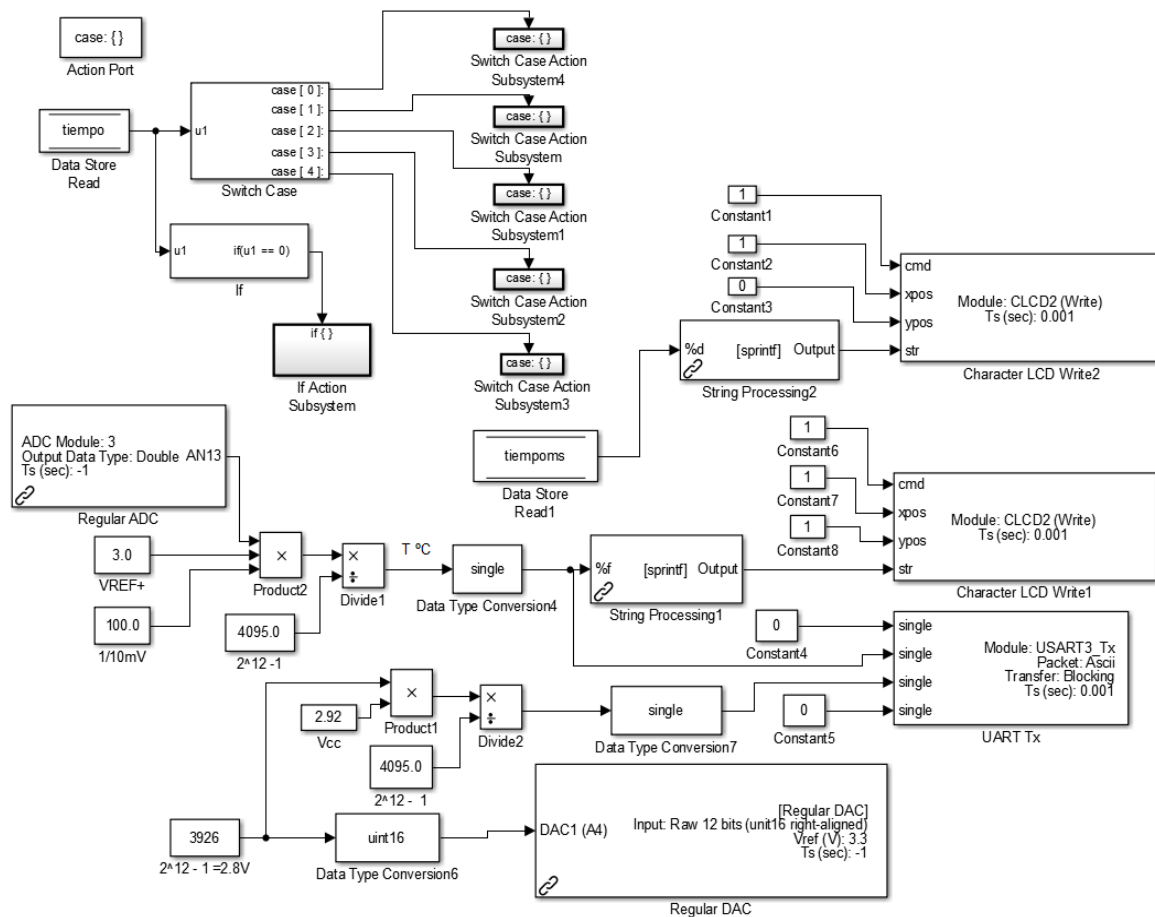
## CASE 0



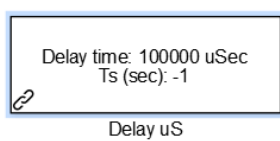
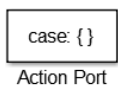
## CASE 1



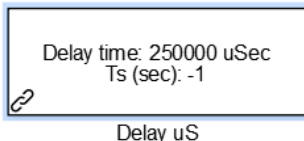
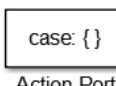
## CASE 2



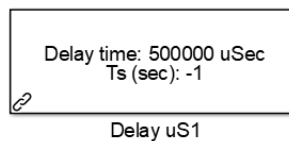
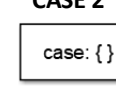
### CASE 0



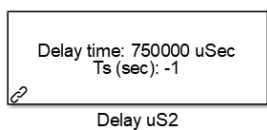
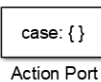
### CASE 1



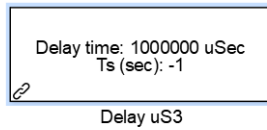
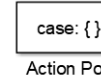
### CASE 2



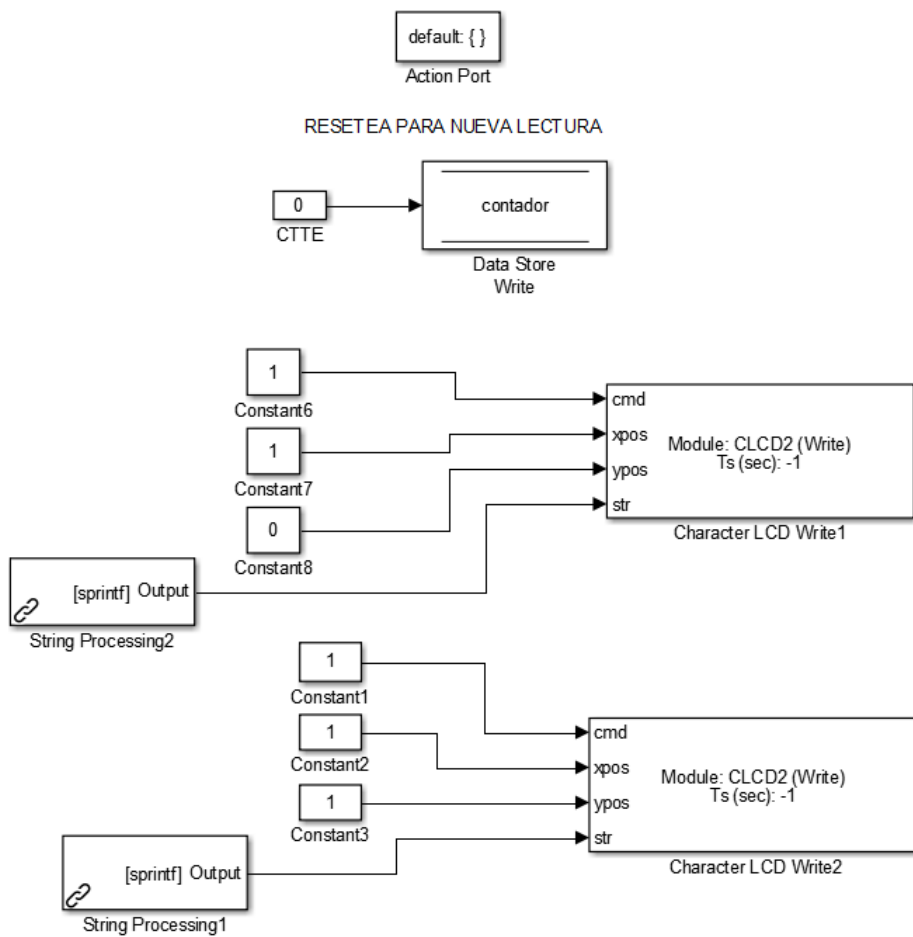
### CASE 3



### CASE 4



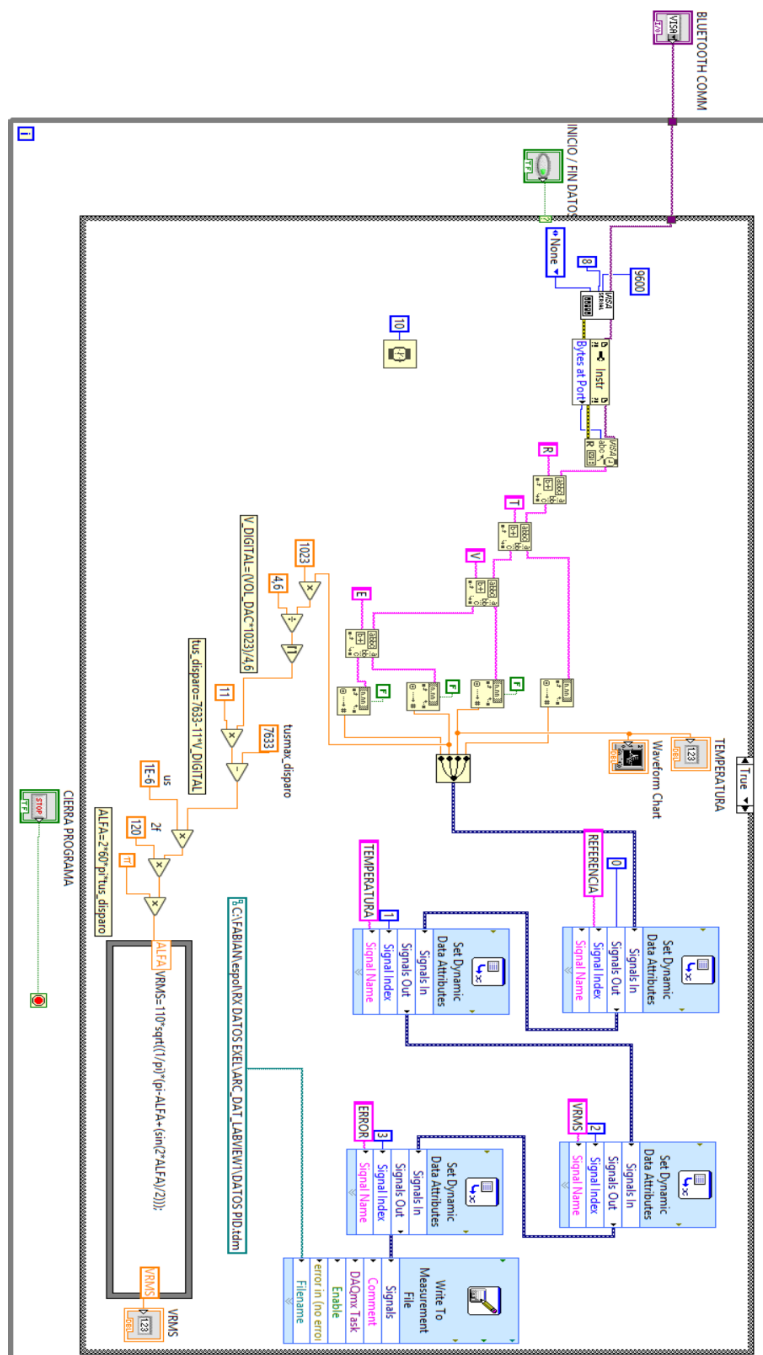
## DEFAULT

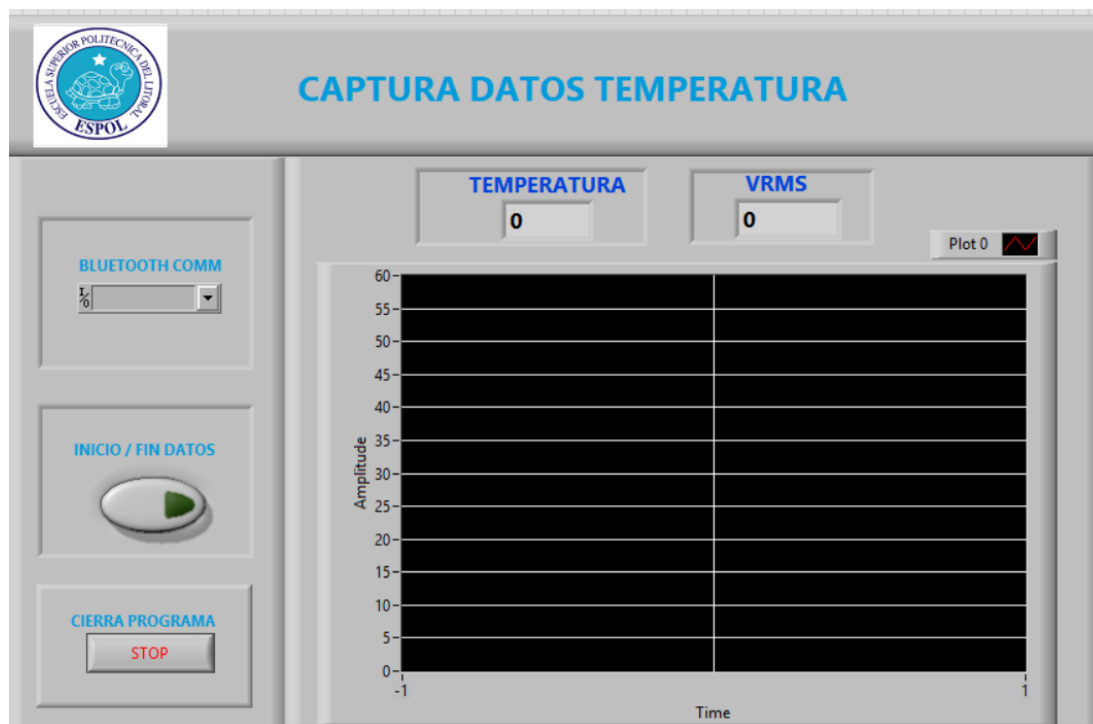




## ANEXO 8

# PROGRAMA CAPTURA DE DATOS LABVIEW

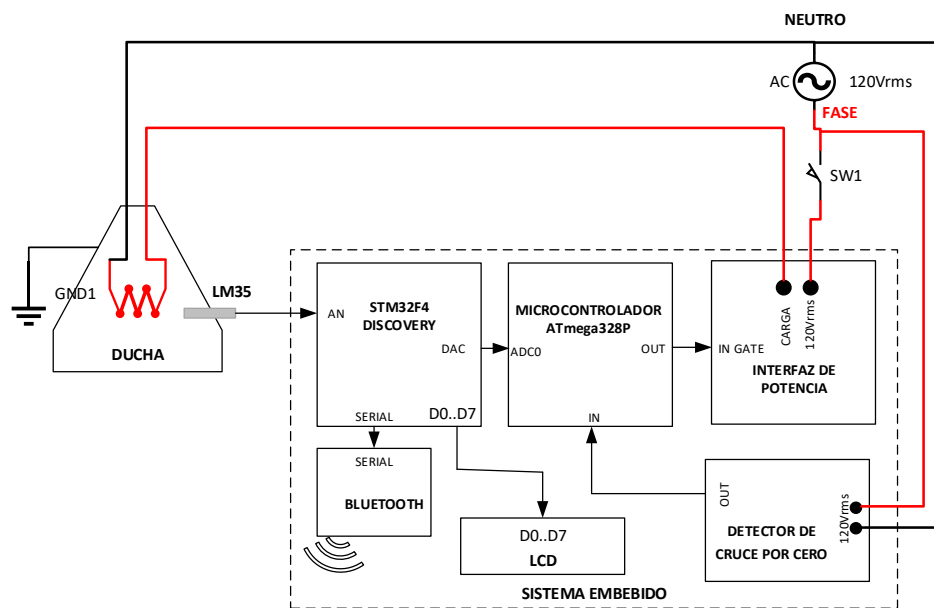




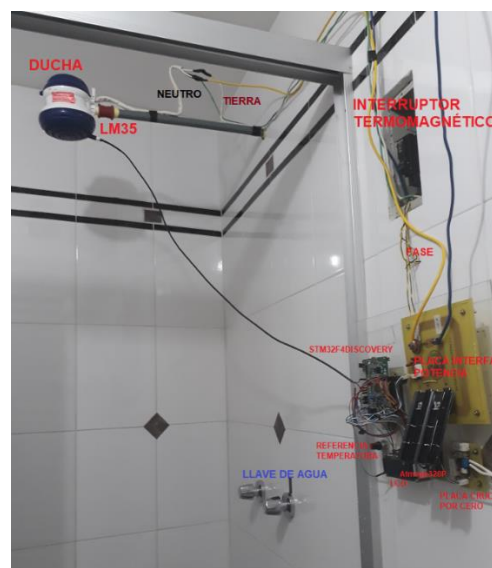
## ANEXO 9

### PASOS PARA CAPTURA DE DATOS

Antes de realizar la captura de datos se debe instalar el sistema embebido como muestran las figuras.

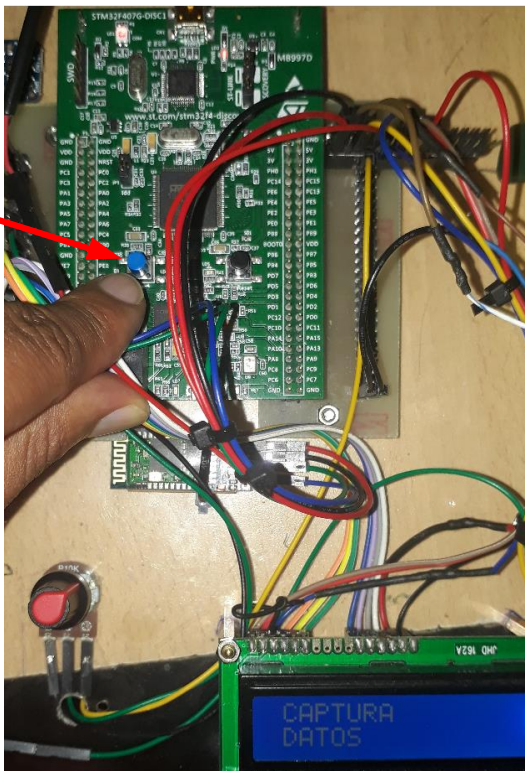


### CONEXIÓN REAL



**Paso 1.** Pulsamos el botón azul del STM32discovery que permitir elegir el tiempo de muestreo.

PULSA BOTON PARA ELEGIR  
TIEMPO DE MUESTREO

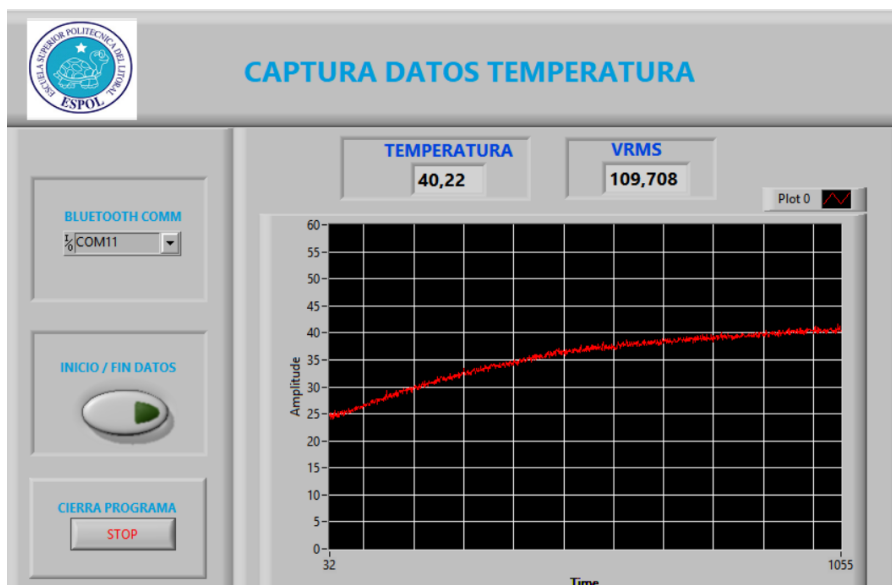


**Paso 2.** Establecemos el tiempo de muestro en 250ms mediante el potenciómetro

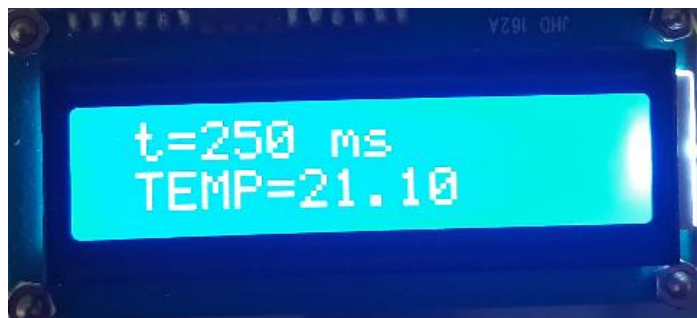
ELEGIR TIEMPO DE  
MUESTREO CON  
POTENCIÓMETRO



**Paso 3.** Ejecutamos la aplicación de LabView “Captura Datos Temperatura” y elegimos el puerto COM perteneciente a la conexión Bluetooth, y pulsamos “INICIO” para iniciar la captura de Datos y “FIN” para terminar la captura, para cerrar el programa se utiliza “STOP”.



**Paso 4.** Pulsamos nuevamente el botón azul del STM32Discovery para iniciar la captura y envió de datos mediante bluetooth hacia la computadora, donde la aplicación de LabView recolecta los datos y lo almacena en el archivo Excel “DATOS PID”, mientras el LCD mostrará el mensaje de tiempo de muestreo y la temperatura actual indicando de esta mera que el programa está en funcionamiento.



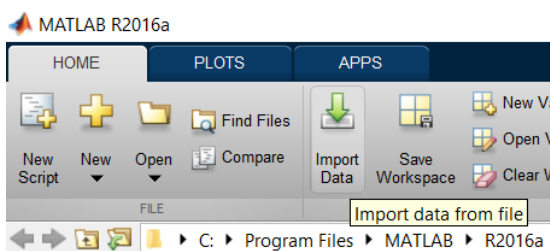
**Paso 5.** Finalmente pulsamos el botón azul del STM32Discovery nuevamente para finalizar la captura de datos y se mostrara en el LCD Final captura de datos.



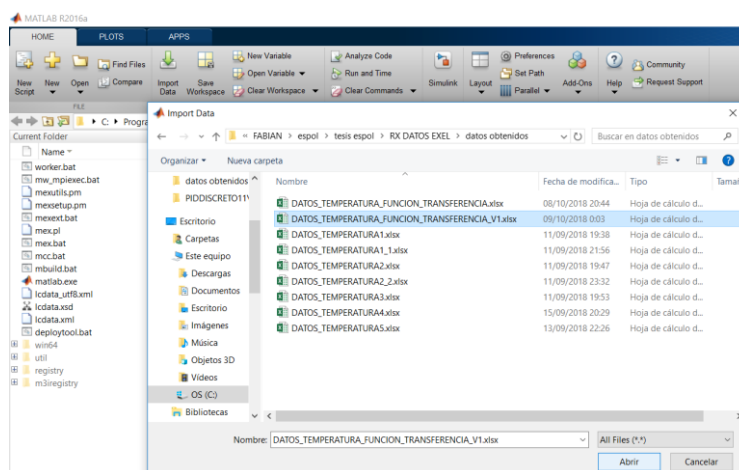
## ANEXO 10

### PASOS PARA IDENTIFICAR PLANTA

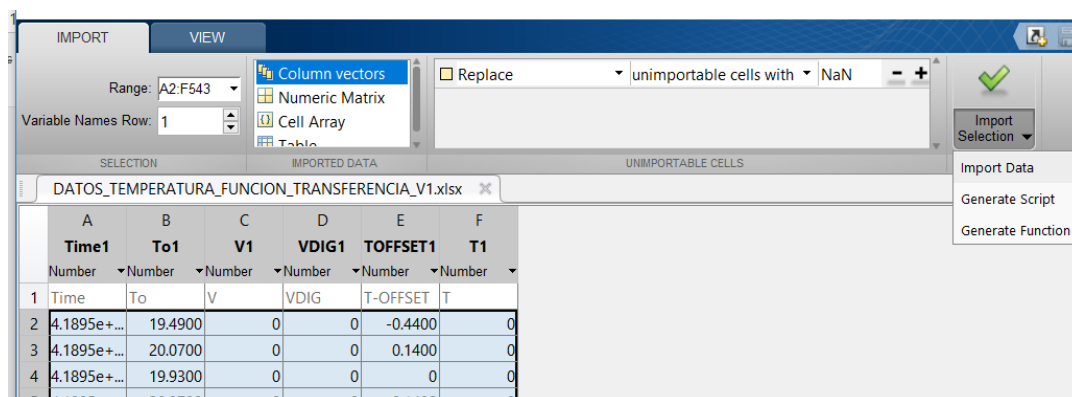
#### Paso 1. Clic "Import Data"



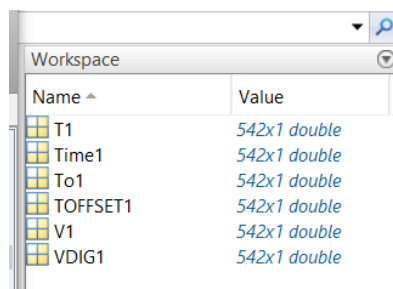
#### Paso 2. Seleccionamos el archivo de Excel y clic en "Abrir"



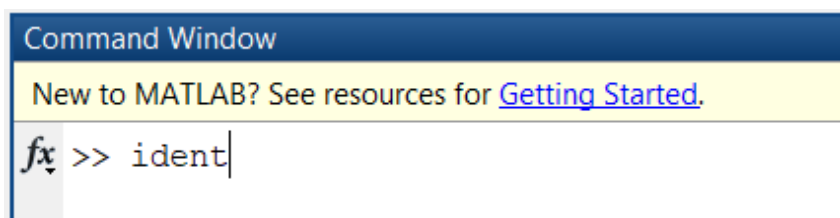
#### Paso 3. Aparece la siguiente imagen y damos clic en "Import Selection", clic en "Import Data"



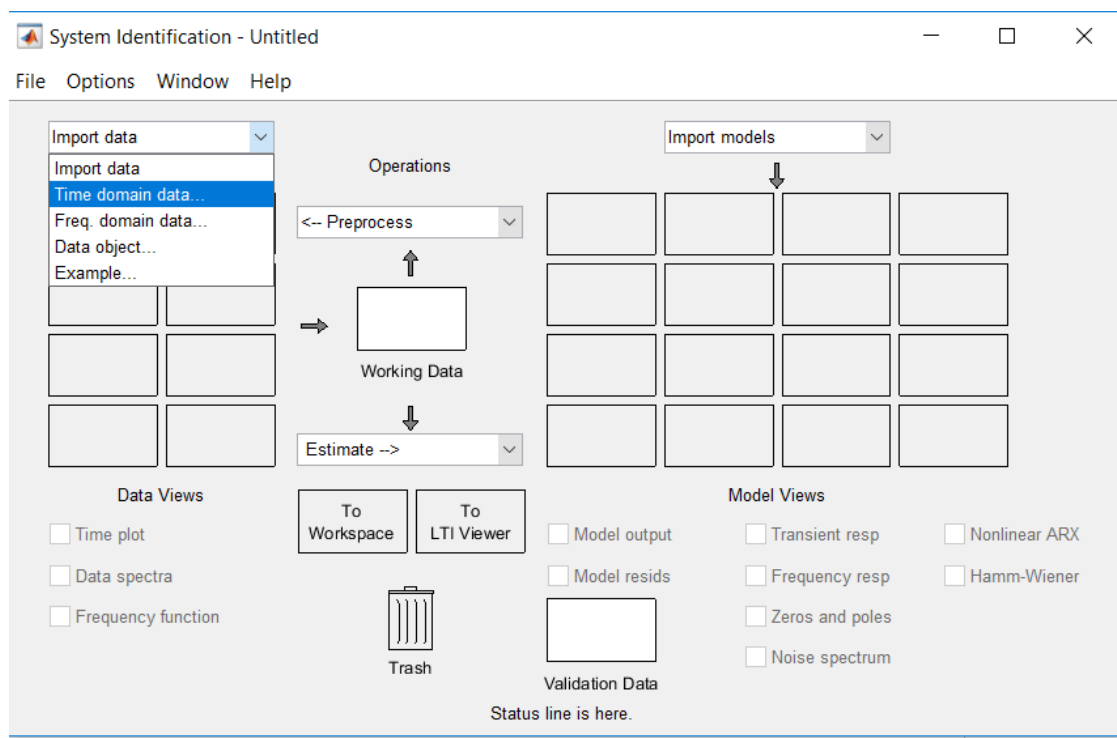
**Paso 4.** Los datos ahora se muestran en **“Workspace”**



**Paso 5.** Digitamos **“Ident”** en el bloque de comandos de Matlab y damos enter.



**Paso 6.** En la ventana **“System Identification”**, seleccionamos **“Time domain data”** y damos clic.



**Paso 7.** En la ventana “**Import Data**” colocamos la variable de entrada **VDIG**, salida **T**, tiempo de inicio 0, frecuencia de muestre 0.25=250ms y clic en “**Impot**”.

**Import Data**

**Data Format for Signals**

Time-Domain Signals

**Workspace Variable**

Input: VDIG

Output: T

**Data Information**

Data name: mydata

Starting time: 0

Sample time: 0.25

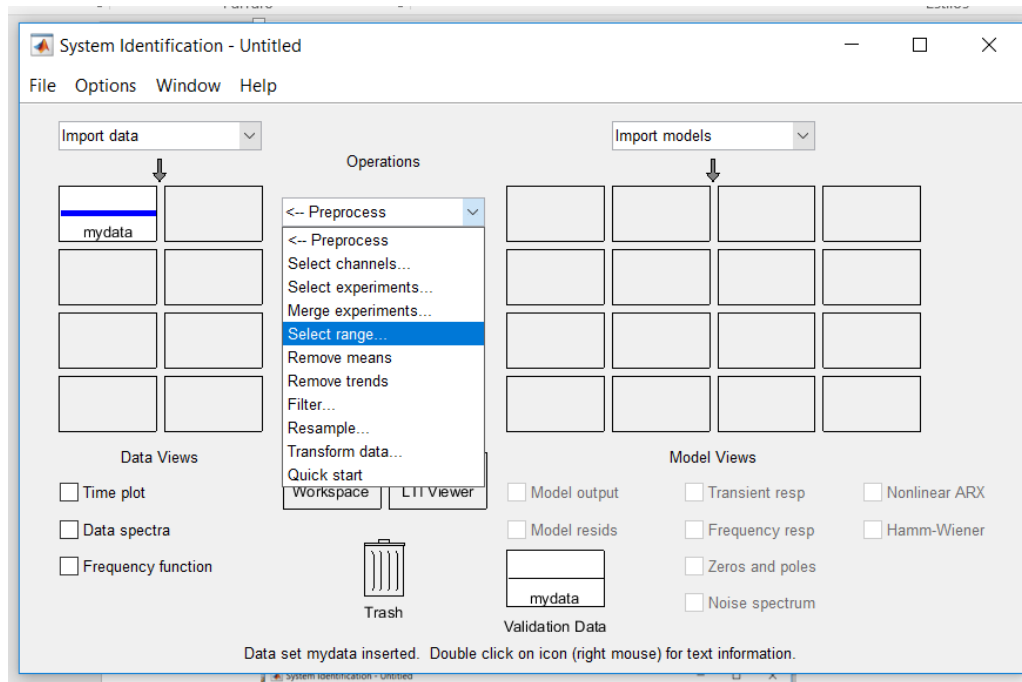
More

Import Reset

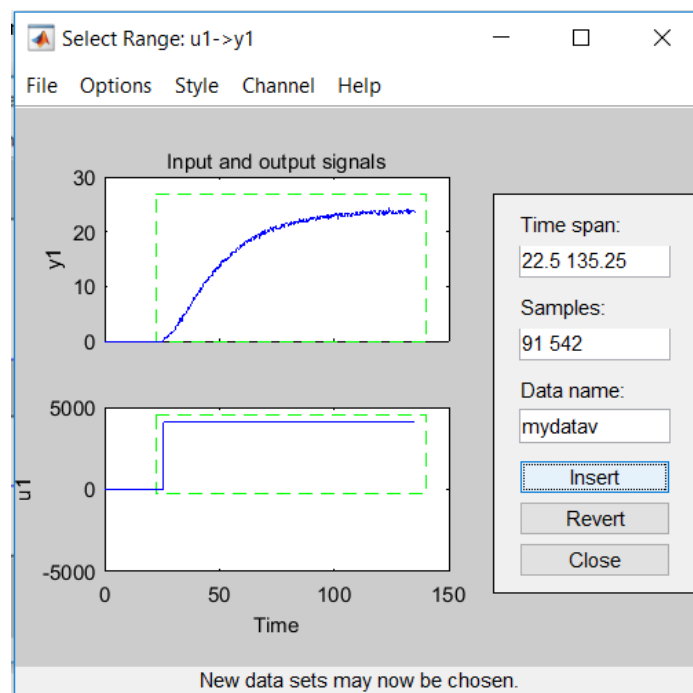
Close Help



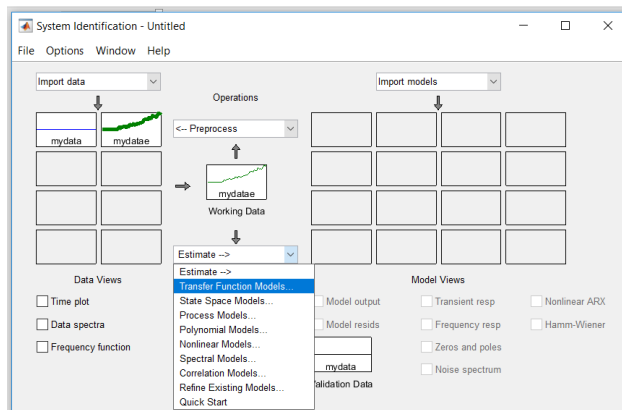
**Paso 8.** Una vez importada los datos damos clic en **“Select range”**



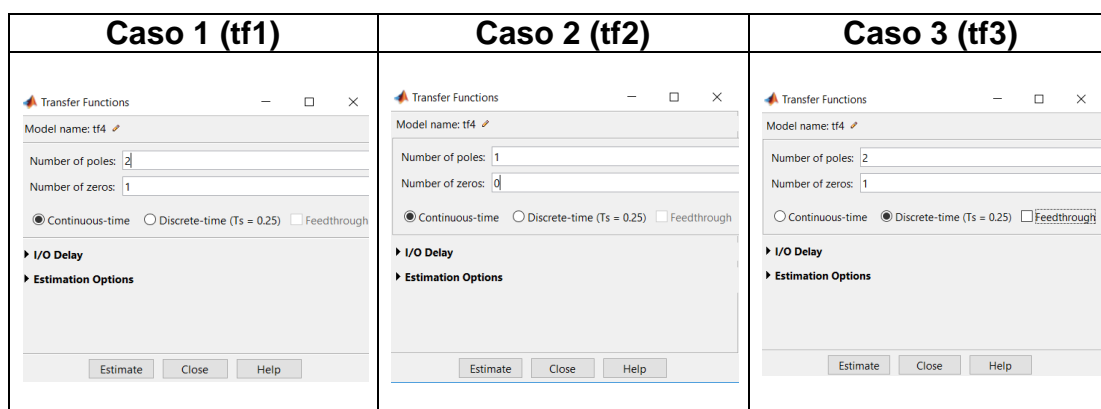
**Paso 9.** En la ventana **“Select Range”** elegimos con el mouse elegimos los datos que analizaremos y después damos clic en **“Insert”**



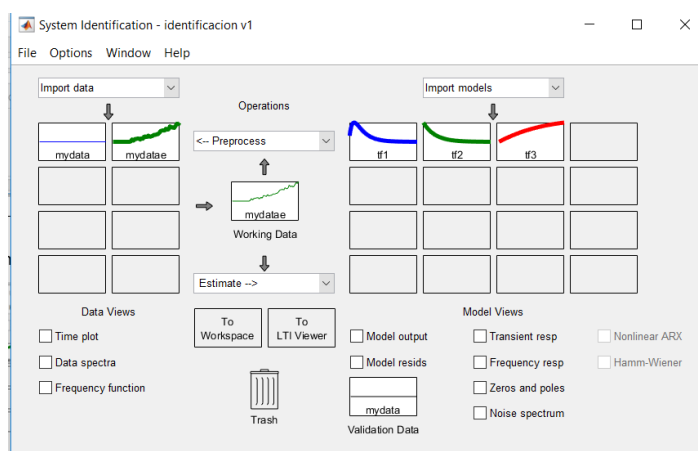
**Paso 10.** Ahora trasladamos la nueva gráfica a la zona “Working Data” y elegimos “Transfer Function Models”



**Paso 11.** Ahora elegimos número de polos, ceros, tiempo-continuo y clic en “Estimate”



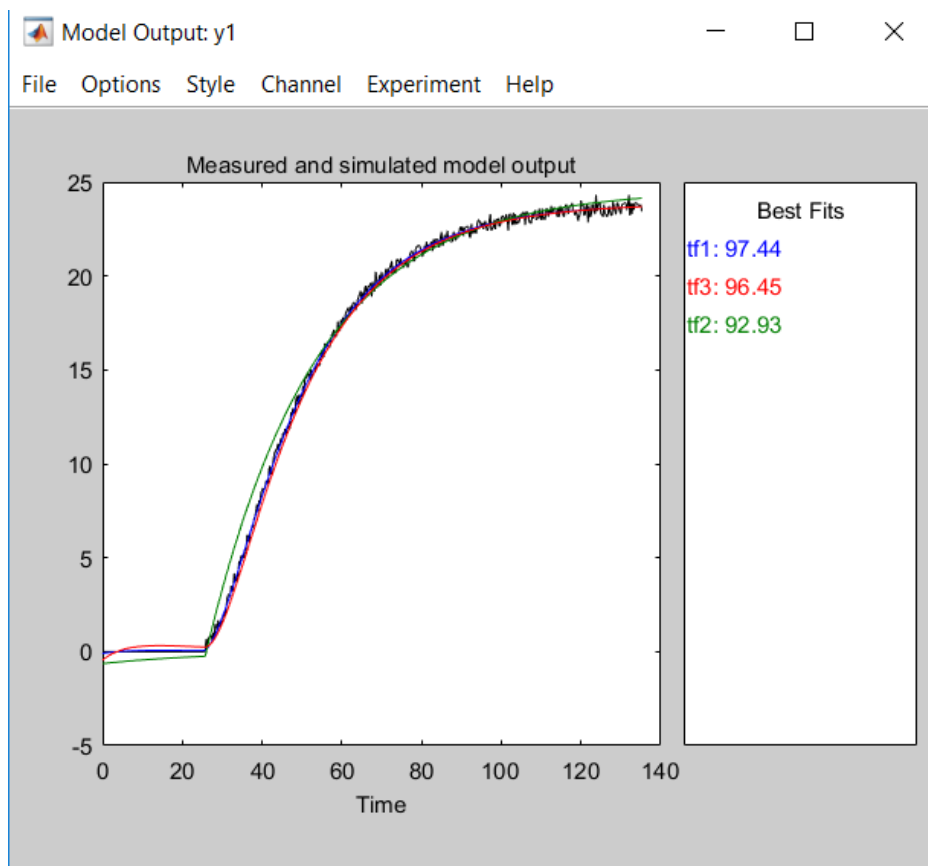
**Paso 12.** Obtenemos la función de transferencia tf1, tf2, tf3



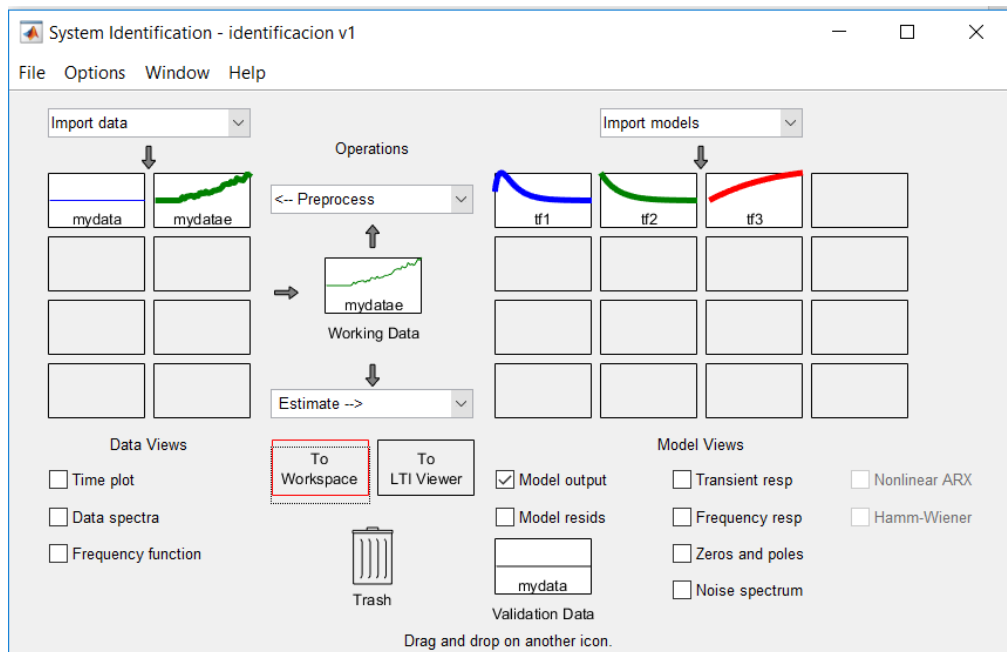
**Paso 13.** Damos clic en la ventana de la función de transferencia obtenida **tf1**, **tf2**, **tf3** y obtenemos los valores de la función de transferencia

| tf1   | tf2  | tf3  |
|---|--|--|
| <p>Data/model Info: tf1</p> <p>Model name: tf1</p> <p>Color: [0,0,1]</p> <p>From input "u1" to output "y1":<br/> <math>5.163e-05 s + 4.172e-05</math><br/> -----<br/> <math>s^2 + 0.198 s + 0.007157</math><br/> Name: tf1<br/> Continuous-time identified transfer function.<br/> Status:<br/> Estimated using TFEST on time domain data "mydatae".<br/> Fit to estimation data: 96.46% (simulation focus)<br/> PFE: 0.06908, MSE: 0.06727</p> | <p>Data/model Info: tf2</p> <p>Model name: tf2</p> <p>Color: [0,0.5,0]</p> <p>From input "u1" to output "y1":<br/> <math>0.0002183</math><br/> -----<br/> <math>s + 0.03633</math><br/> Name: tf2<br/> Continuous-time identified transfer function.<br/> Status:<br/> Estimated using TFEST on time domain data "mydatae".<br/> Fit to estimation data: 92.26% (simulation focus)<br/> PFE: 0.3261, MSE: 0.3218</p> | <p>Data/model Info: tf3</p> <p>Model name: tf3</p> <p>Color: [1,0,0]</p> <p>From input "u1" to output "y1":<br/> <math>2.699e-06 z^{-1}</math><br/> -----<br/> <math>1 - 1.949 z^{-1} + 0.9493 z^{-2}</math><br/> Name: tf3<br/> Sample time: 0.25 seconds<br/> Status:<br/> Estimated using TFEST on time domain data "mydatae".<br/> Fit to estimation data: 96.47% (simulation focus)<br/> PFE: 0.06822, MSE: 0.06673</p> |

**Paso 14.** En la ventana “**System Identification**” elegimos “**Model output**” y tenemos el siguiente resultado



**Paso 15.** Ahora arrastramos la función de transferencia **tf1**, **tf2**, **tf3** a la ventana To Workspace para poder utilizarlo en simulink



**Paso 16.** Podemos verificar la función digitando **tf1**, **tf2**, **tf3** en la ventana de comandos de Matlab.

```

Command Window
New to MATLAB? See resources for Getting Started.
>> tf1

tf1 =

    From input "u1" to output "y1":
    5.163e-05 s + 4.172e-05
    -----
    s^2 + 0.198 s + 0.007157

Name: tf1
Continuous-time identified transfer function.

Parameterization:
    Number of poles: 2    Number of zeros: 1
    Number of free coefficients: 4
    Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
    Estimated using TFEST on time domain data "mydatae".
    Fit to estimation data: 96.46% (simulation focus)
    FPE: 0.06908, MSE: 0.06727
  
```

```

Command Window
New to MATLAB? See resources for Getting Started.

>> ident
>> tf2

tf2 =

    From input "u1" to output "y1":
    0.0002183
    -----
    s + 0.03633

Name: tf2
Continuous-time identified transfer function.

Parameterization:
    Number of poles: 1   Number of zeros: 0
    Number of free coefficients: 2
    Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mydatae".
Fit to estimation data: 92.26% (simulation focus)
FPE: 0.3261, MSE: 0.3218

```

```

Command Window
New to MATLAB? See resources for Getting Started.

Fit to estimation data: 92.26% (simulation focus)
FPE: 0.3261, MSE: 0.3218
>> tf3

tf3 =

    From input "u1" to output "y1":
    2.699e-06 z^-1
    -----
    1 - 1.949 z^-1 + 0.9493 z^-2

Name: tf3
Sample time: 0.25 seconds
Discrete-time identified transfer function.

Parameterization:
    Number of poles: 2   Number of zeros: 1
    Number of free coefficients: 3
    Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

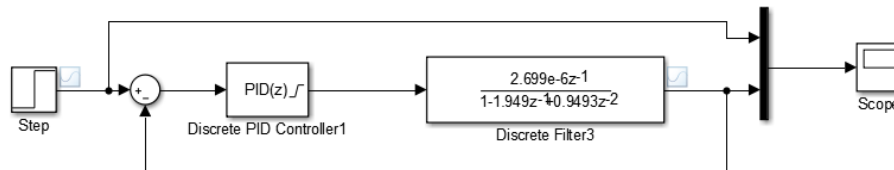
Status:
Estimated using TFEST on time domain data "mydatae".
Fit to estimation data: 96.47% (simulation focus)
FPE: 0.06822, MSE: 0.06673

```

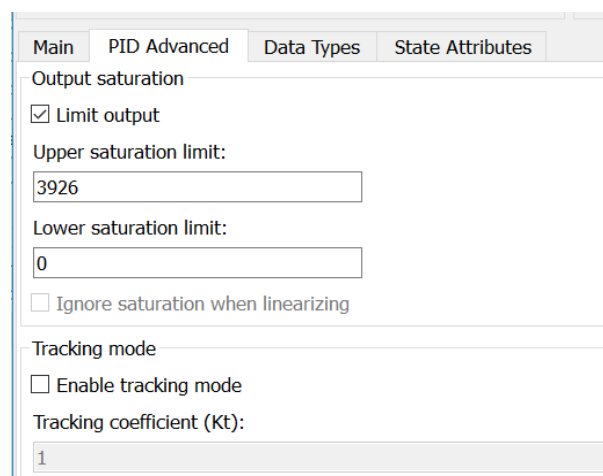
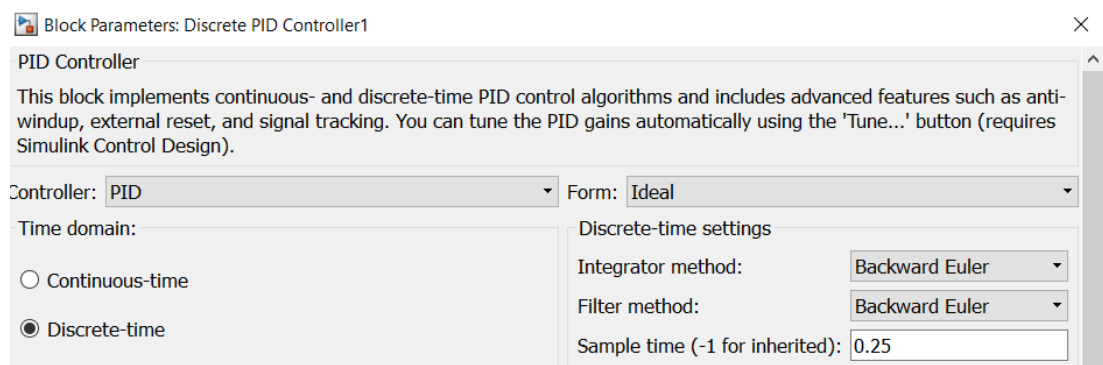
## ANEXO 11

### PASOS PARA IDENTIFICAR CONSTANTES DE PID

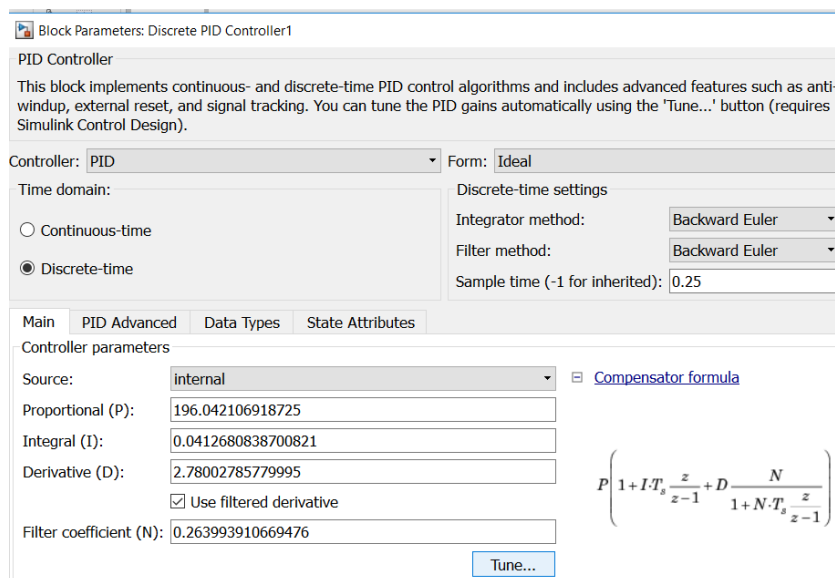
**Paso 1.** Realizamos el diagrama de bloques en simulink con la función de transferencia y PID discreta.



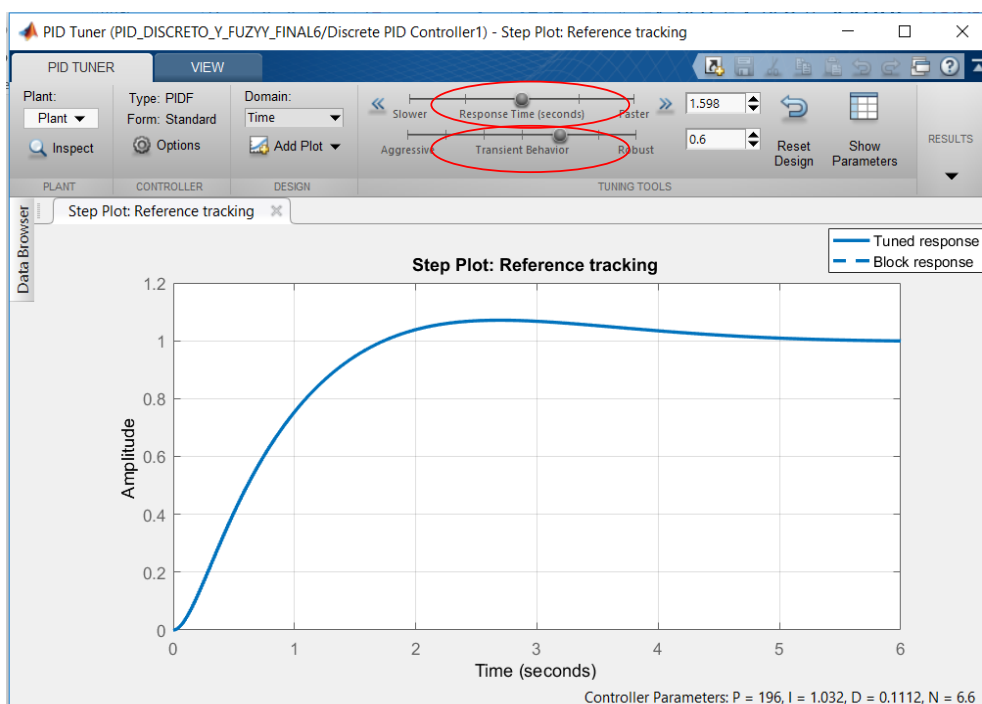
**Paso 2.** Configuramos el bloque de PID discreto con los parámetros que se muestran en la imagen: Controlador=PID, forma=ideal, tiempo discreto, frecuencia de muestreo de 0.25, límite de saturación 3926 que es valor del DAC para alcanzar 2.8Vdc aproximadamente.



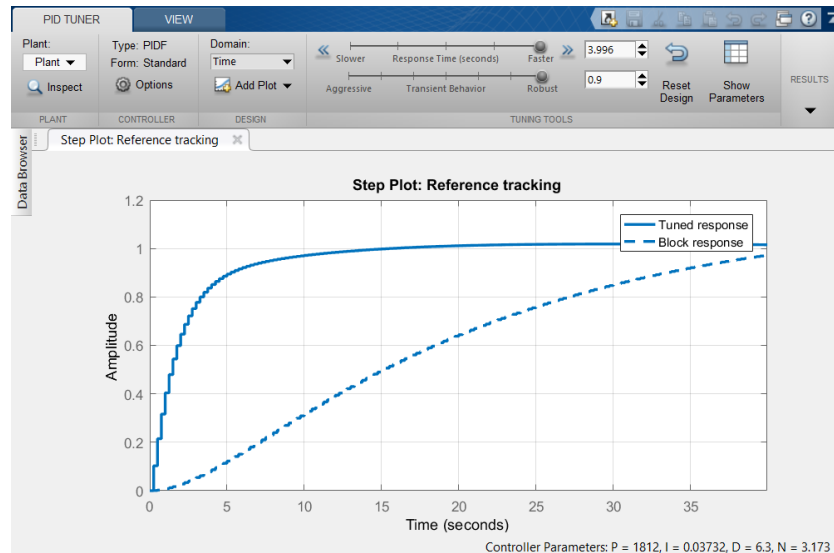
### Paso 3. Clic en “Tune”, para acceder a PID Tuner



### Paso 4. PID Tuner establece una posible respuesta a una señal escalón.



**Paso 5.** Variando el tiempo de respuesta “**Response Time**” y el comportamiento de transiente “**Transient Behavior**” se mejora la respuesta del sistema obteniendo los siguientes resultados.



**Paso 6.** De los valores del PID que se muestran en la figura se establece los valores de  $P=K_p=1812.3529$ ,  $I=K_i=0.037316$ ,  $D=K_d=6.3002$  por tener 1.84% de sobreimpulso, tiempo de establecimiento de 11.3 segundos, adecuados para el diseño de los controladores PID de posición y velocidad.

| Controller Parameters      |                      |                          |
|----------------------------|----------------------|--------------------------|
|                            | Tuned                | Block                    |
| P                          | 1812.3529            | 196.0421                 |
| I                          | 0.037316             | 0.041268                 |
| D                          | 6.3002               | 2.78                     |
| N                          | 3.1725               | 0.26399                  |
|                            |                      |                          |
| Performance and Robustness |                      |                          |
|                            | Tuned                | Block                    |
| Rise time                  | 5 seconds            | 29 seconds               |
| Settling time              | 11.3 seconds         | 112 seconds              |
| Overshoot                  | 1.84 %               | 6.9 %                    |
| Peak                       | 1.02                 | 1.07                     |
| Gain margin                | 27.4 dB @ 12.6 rad/s | 72.7 dB @ 12.6 rad/s     |
| Phase margin               | 90 deg @ 0.501 rad/s | 69.3 deg @ 0.0504 rad... |
| Closed-loop stability      | Stable               | Stable                   |



## ANEXO 12

### PROGRAMA PARA CALCULAR CONSTANTES DE PID DISCRETO Y DIFUSO

```

%CONSTANTES PARA PID Discreto y Difuso
T=0.25;%Periodo de muestreo
%PID(Z) (DISCRETO)
Kp=1812.35289659286; %OKU1
Ki=0.0373164815020297;%OKU1
Kd=6.3001640643544; N=3.17254537006096;%OKU1
%CTTE PID POSICION Y VELOCIDAD
Ti=1/Ki;
Td=Kd;
KI=(Kp*T)/Ti;
KD=(Kp*Td)/T;
KP=Kp-KI/2;
%CTTE PID DIFUSO
GE = 1;
GCE = GE*(Kp-sqrt(Kp^2-4*Ki*Kd))/2/Ki;
GCU = Ki/GE;
GU = Kd/GCE;
GCExGCU=GCE*GCU;

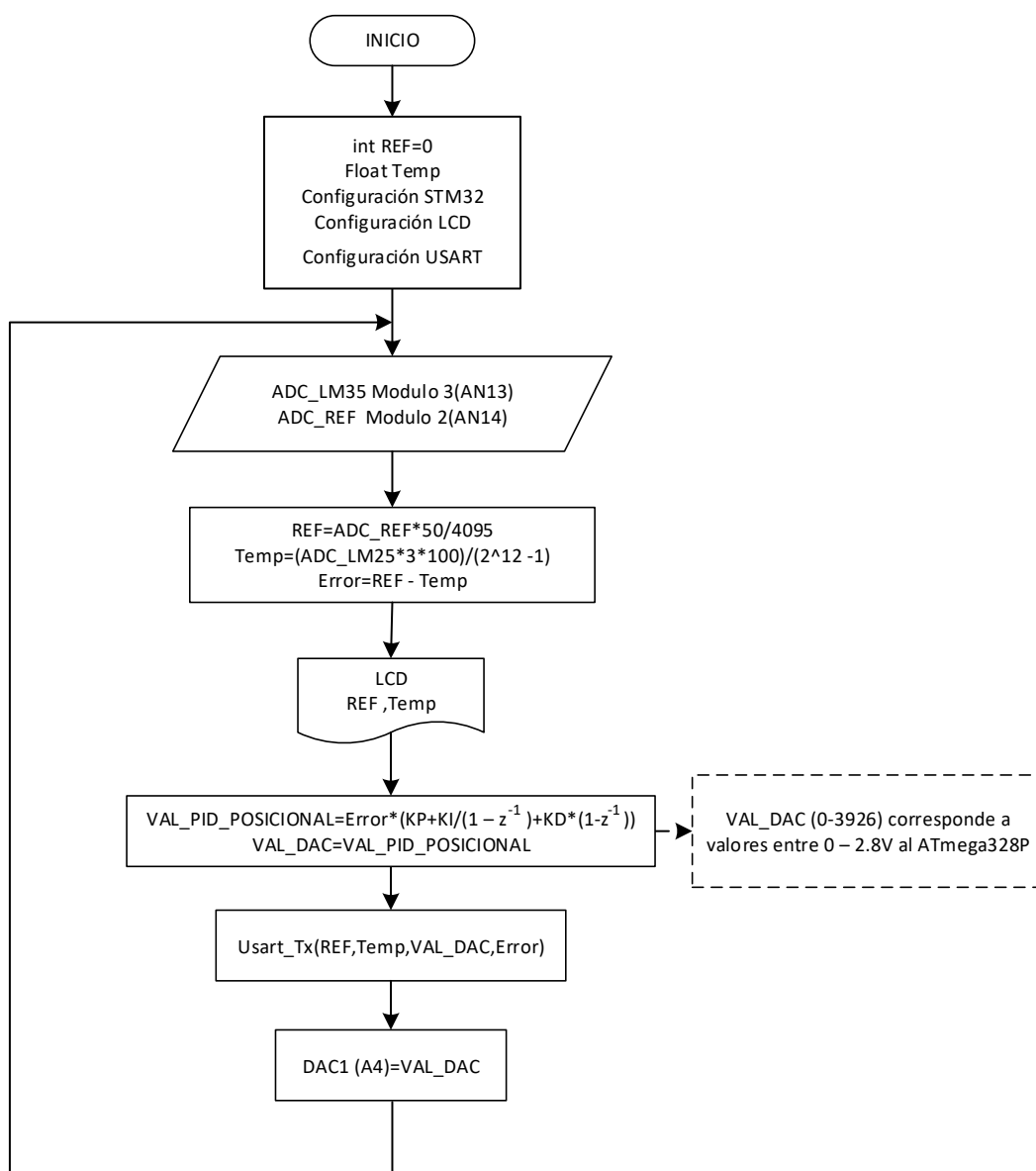
```

## RESULTADOS

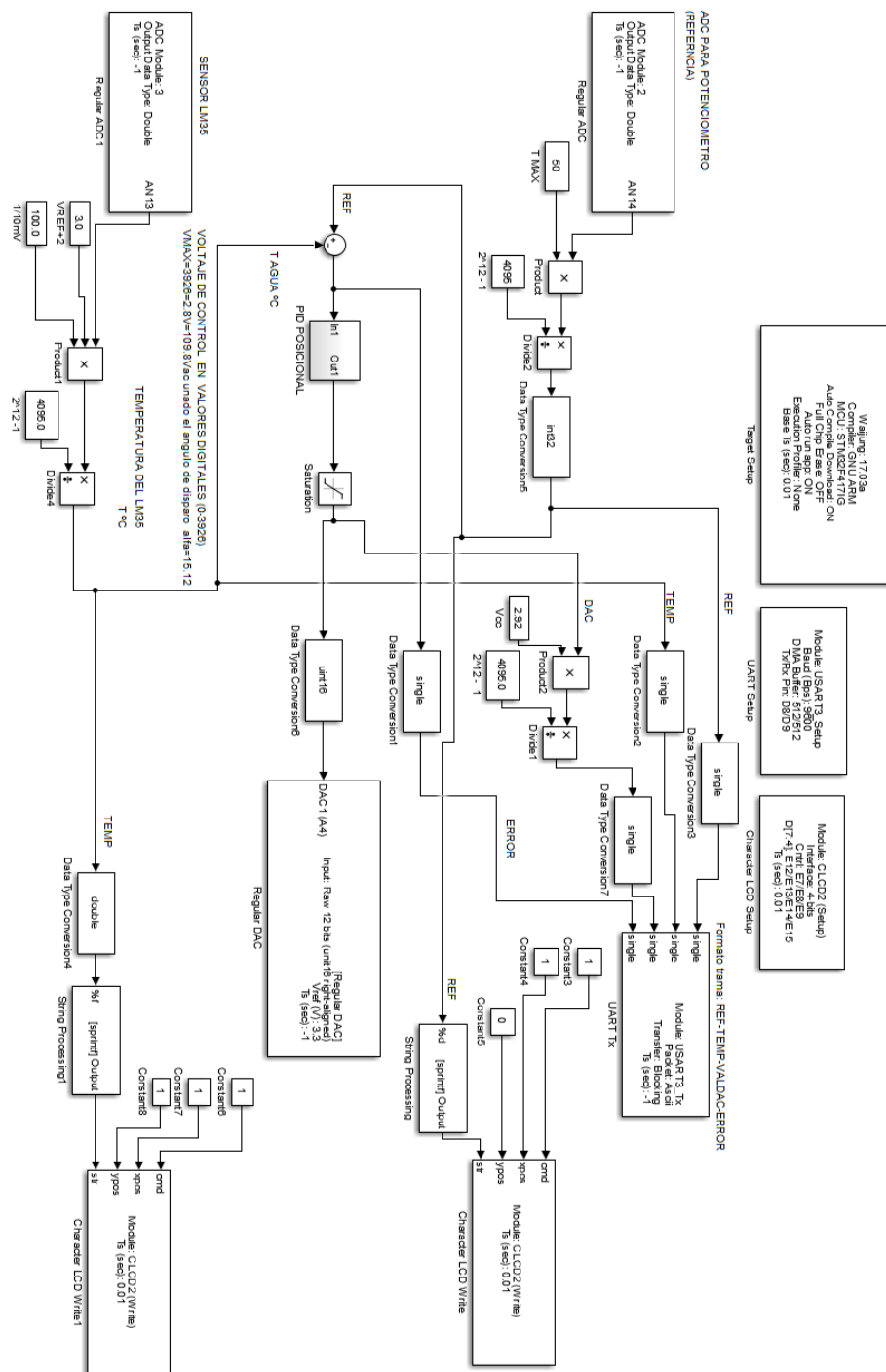
| Name ^  | Value      |
|---------|------------|
| GCE     | 8.8084e-04 |
| GCExGCU | 5.4166e-04 |
| GCU     | 0.6149     |
| GE      | 1          |
| GU      | 235.2510   |
| Kd      | 0.2072     |
| KD      | 4.8749e+03 |
| Ki      | 0.6149     |
| KI      | 1.4467     |
| Kp      | 235.2516   |
| KP      | 234.5282   |
| N       | 83.3863    |
| T       | 0.0100     |
| Td      | 0.2072     |
| Ti      | 1.6262     |

## ANEXO 13

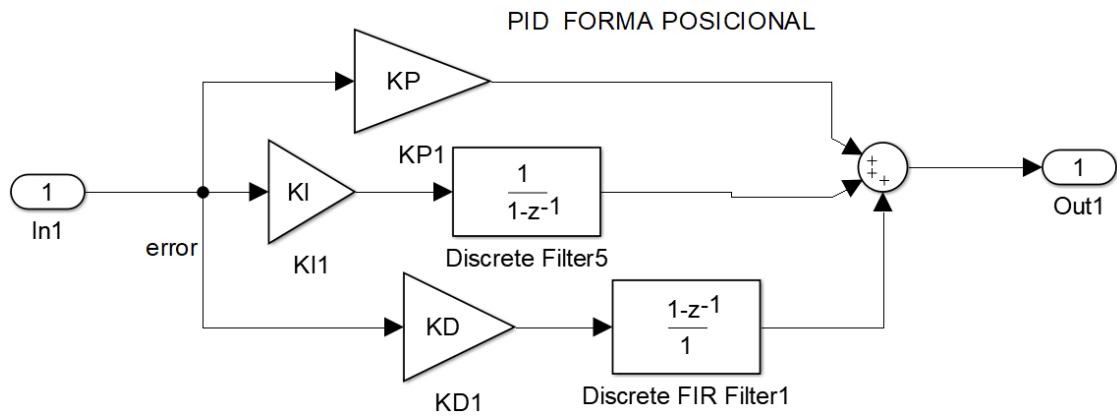
### DIAGRAMA DE FLUJO DEL PID DE FORMA POSICIONAL



# PROGRAMA DEL PID FORMA POSICIONAL EN SIMULINK

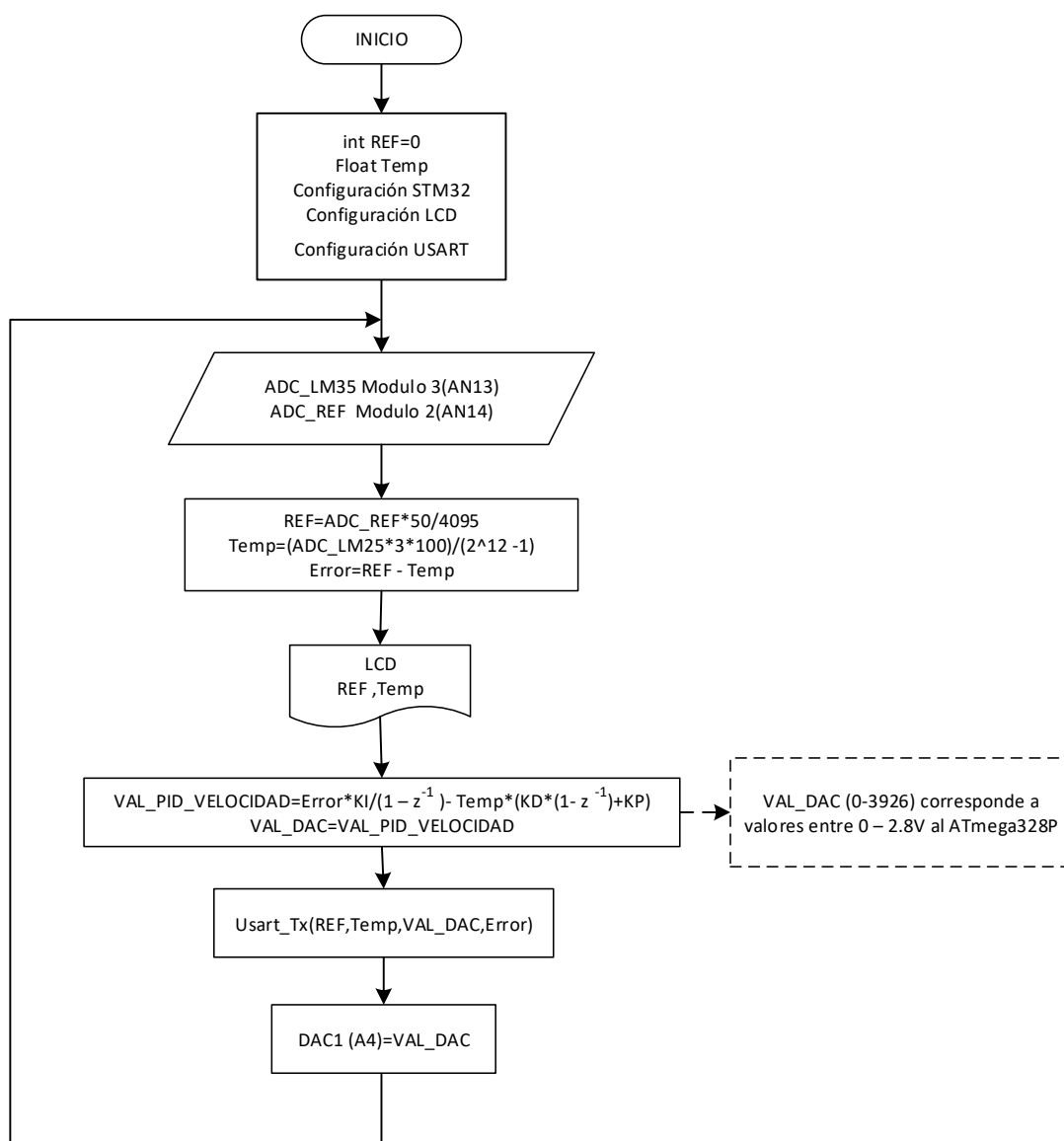


## SUBSISTEMA PID FORMA POSICIONAL

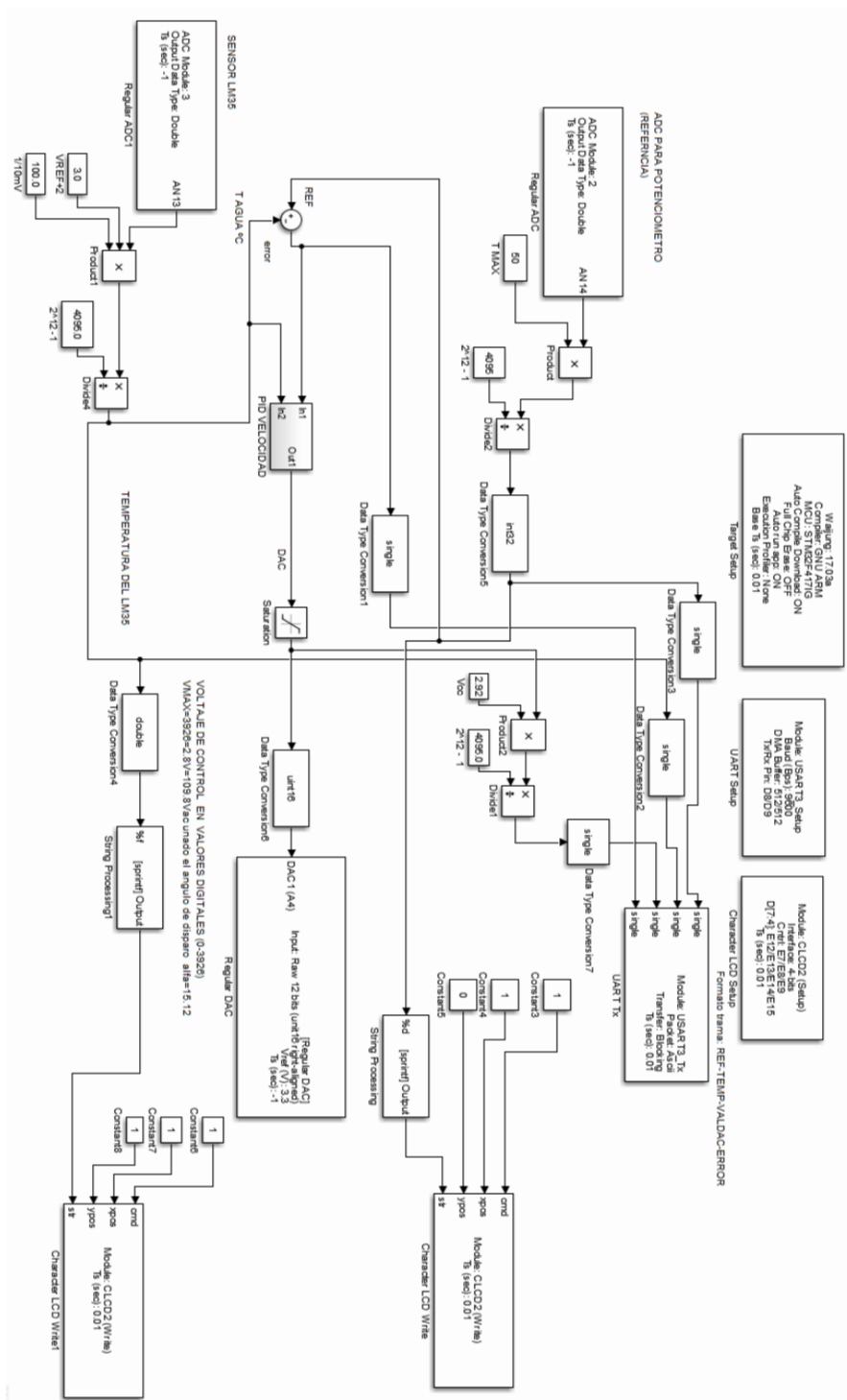


## ANEXO 14

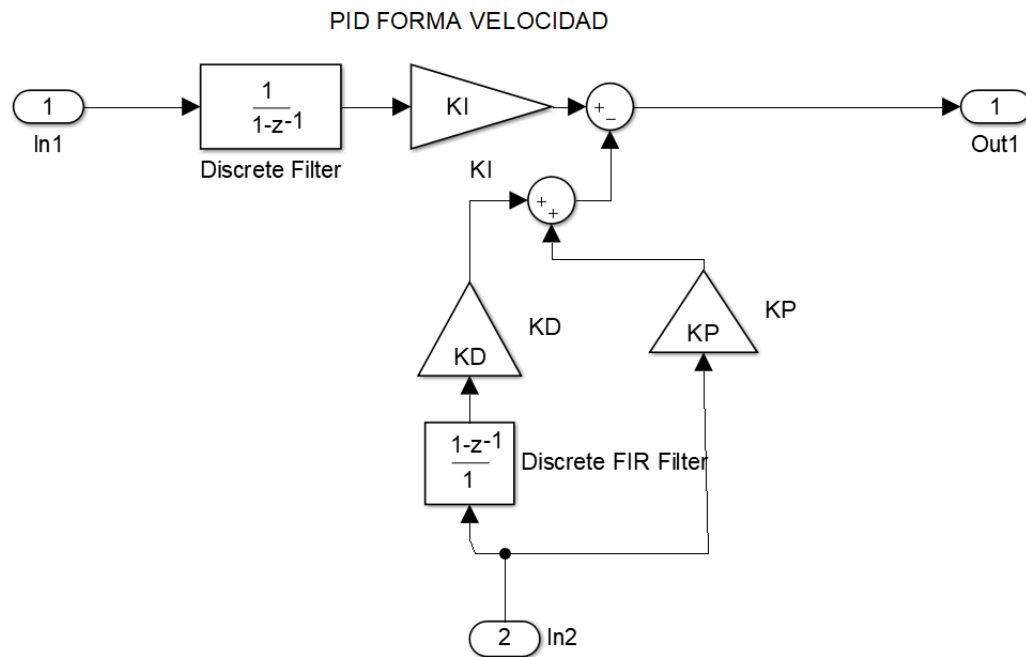
### DIAGRAMA DE FLUJO DEL PID FORMA DE VELOCIDAD



# PROGRAMA DEL PID FORMA DE VELOCIDAD EN SIMULINK



## SUBSISTEMA PID VELOCIDAD



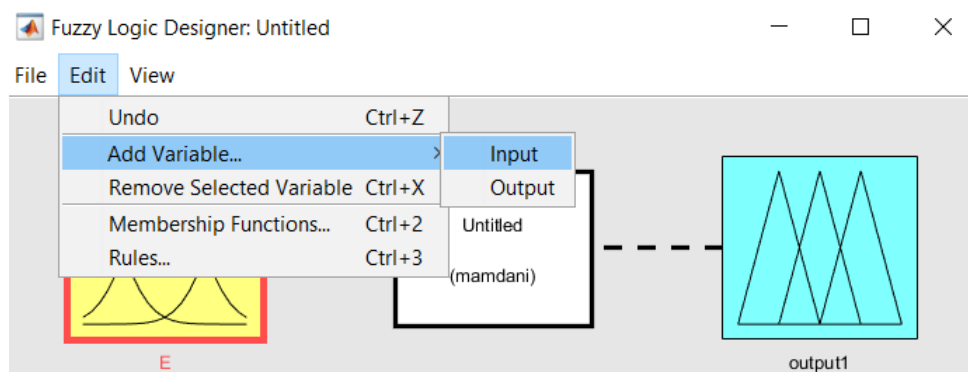
## ANEXO 15

### DISEÑO DEL PID DIFUSO

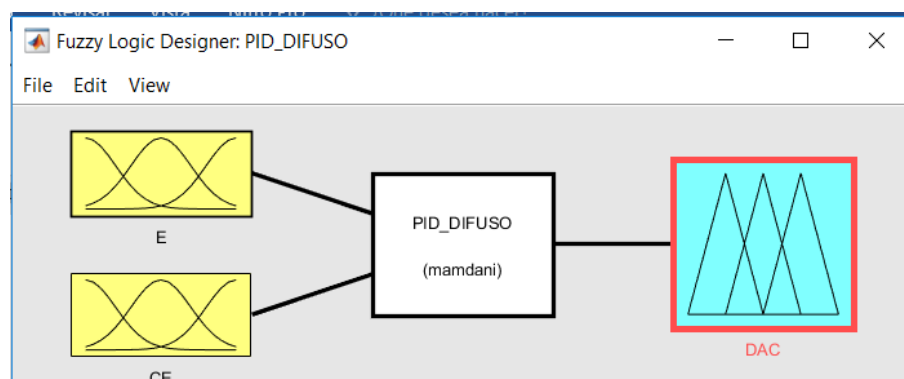
**Paso 1.** Abrimos la aplicación “Fuzzy Logic Designer” ejecutando en el Command Windows de Matlab “**fuzzyLogicDesigner**” y después enter.

```
Command Window
fx >> fuzzyLogicDesigner
```

**Paso 2.** Ingresamos las variables lingüísticas de entradas (E, CE) y salida (DAC) eligiendo Edit/Add Variable/Input u Output



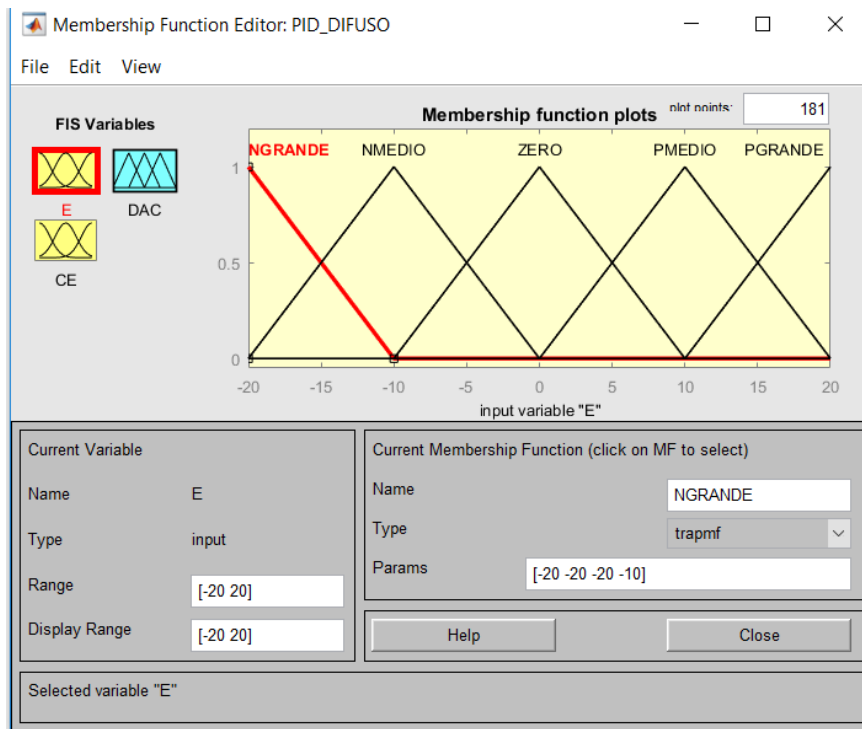
Obtenemos



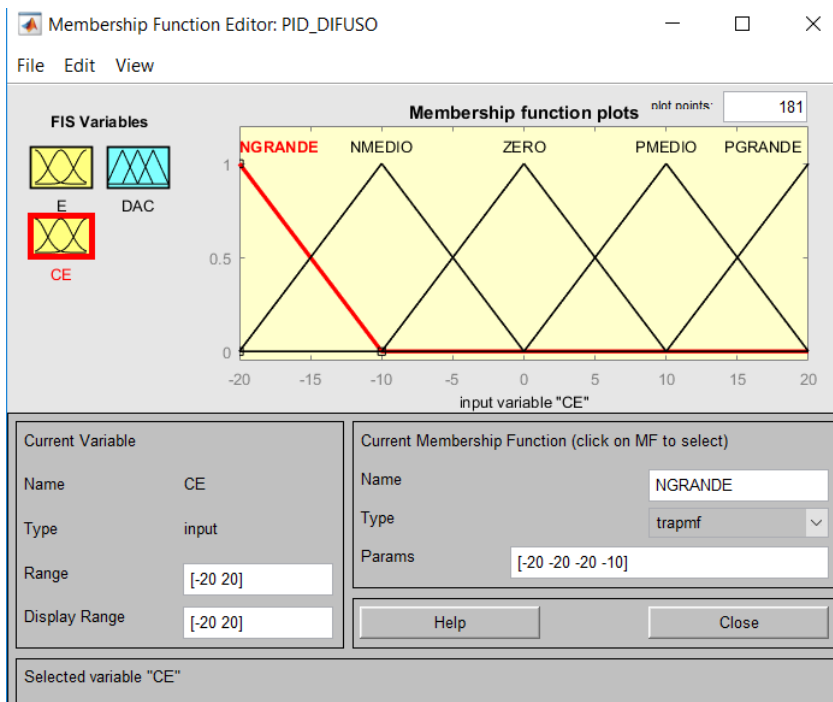
**Paso 3.** Ingresamos el Universo discurso, Términos lingüísticos, Funciones de membresía con doble clic en E, CE, DAC respectivamente, con los valores establecidos.



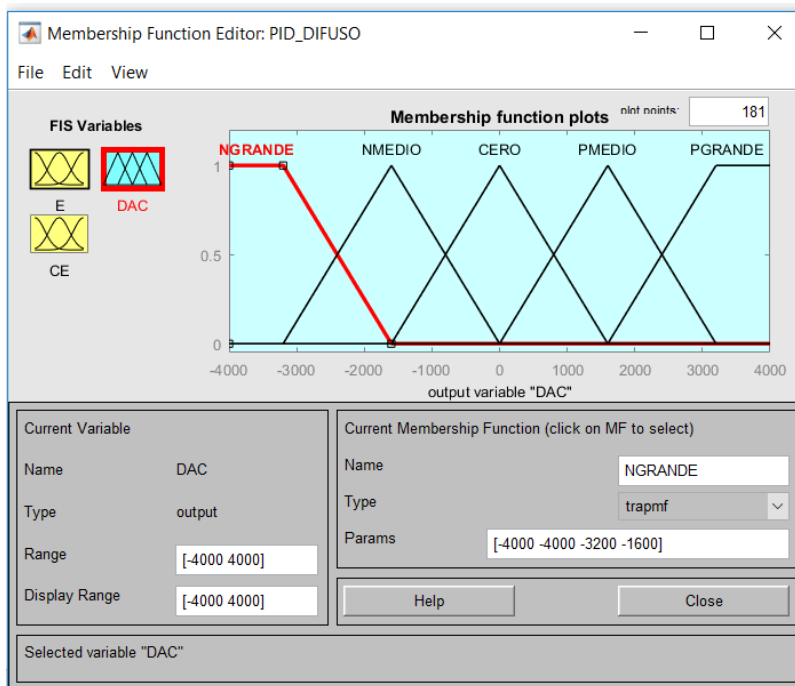
### Datos ingresados para E=Error



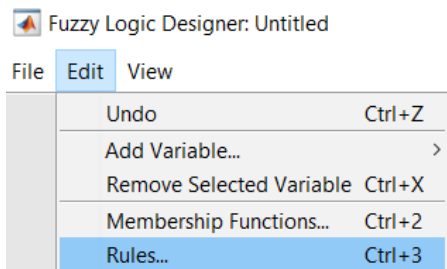
### Datos ingresados para CE=Cambio de Error



Datos ingresados para DAC=Valores del DAC

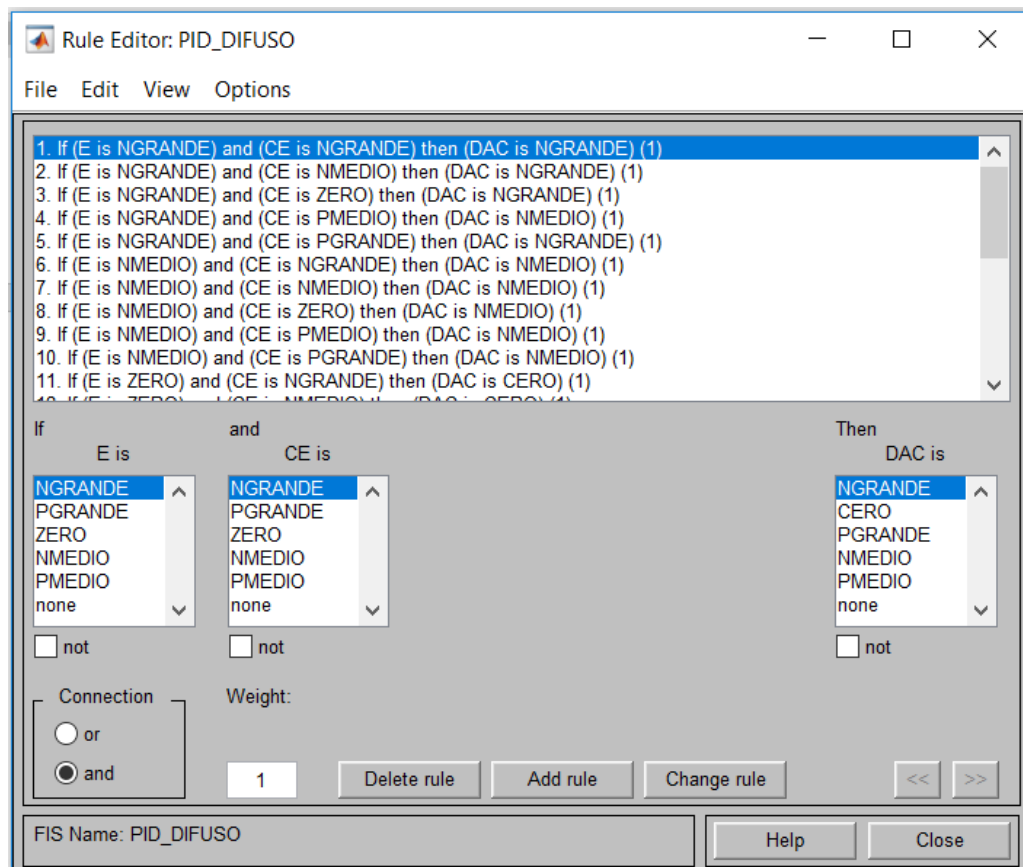


**Paso 4.** Ingresamos las reglas eligiendo Edit/Rules como muestra la figura.

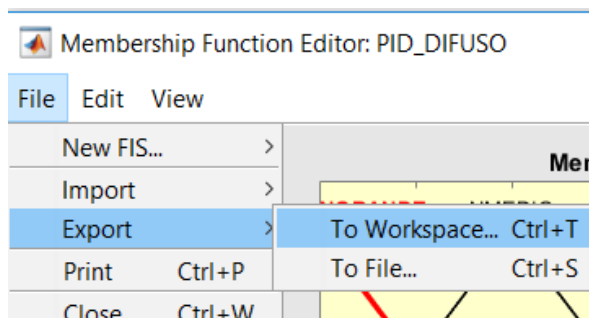


**Paso 5.** Ingresamos todas las reglas de acuerdo al análisis establecido eligiendo E, CE y DAC de la matriz de reglas y finalmente clic en Close.

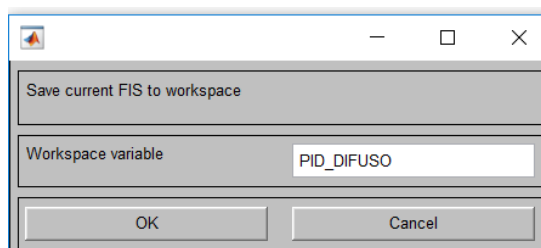
|       |    | Δerror |    |    |    |    |
|-------|----|--------|----|----|----|----|
|       |    | NG     | NM | ZE | PM | PG |
| error | NG | NG     | NG | NG | NM | NG |
|       | NM | NM     | NM | NM | NM | NM |
|       | ZE | ZE     | ZE | ZE | ZE | ZE |
|       | PM | PM     | PM | PM | PM | PM |
|       | PG | PG     | PG | PG | PG | PG |



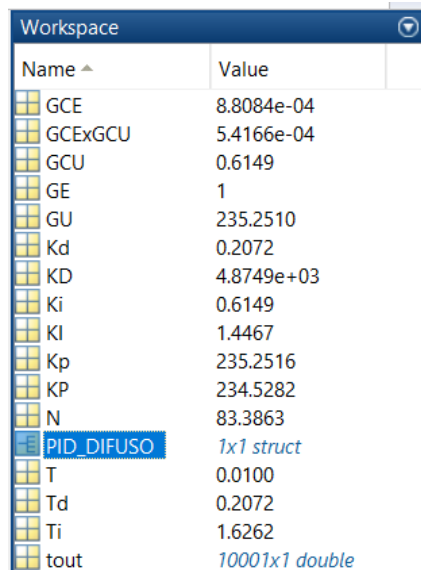
**Paso 6.** Exportamos los datos del PID DIFUSO como muestra la figura



**Paso 7.** Damos clic en OK para enviar al Workspace.



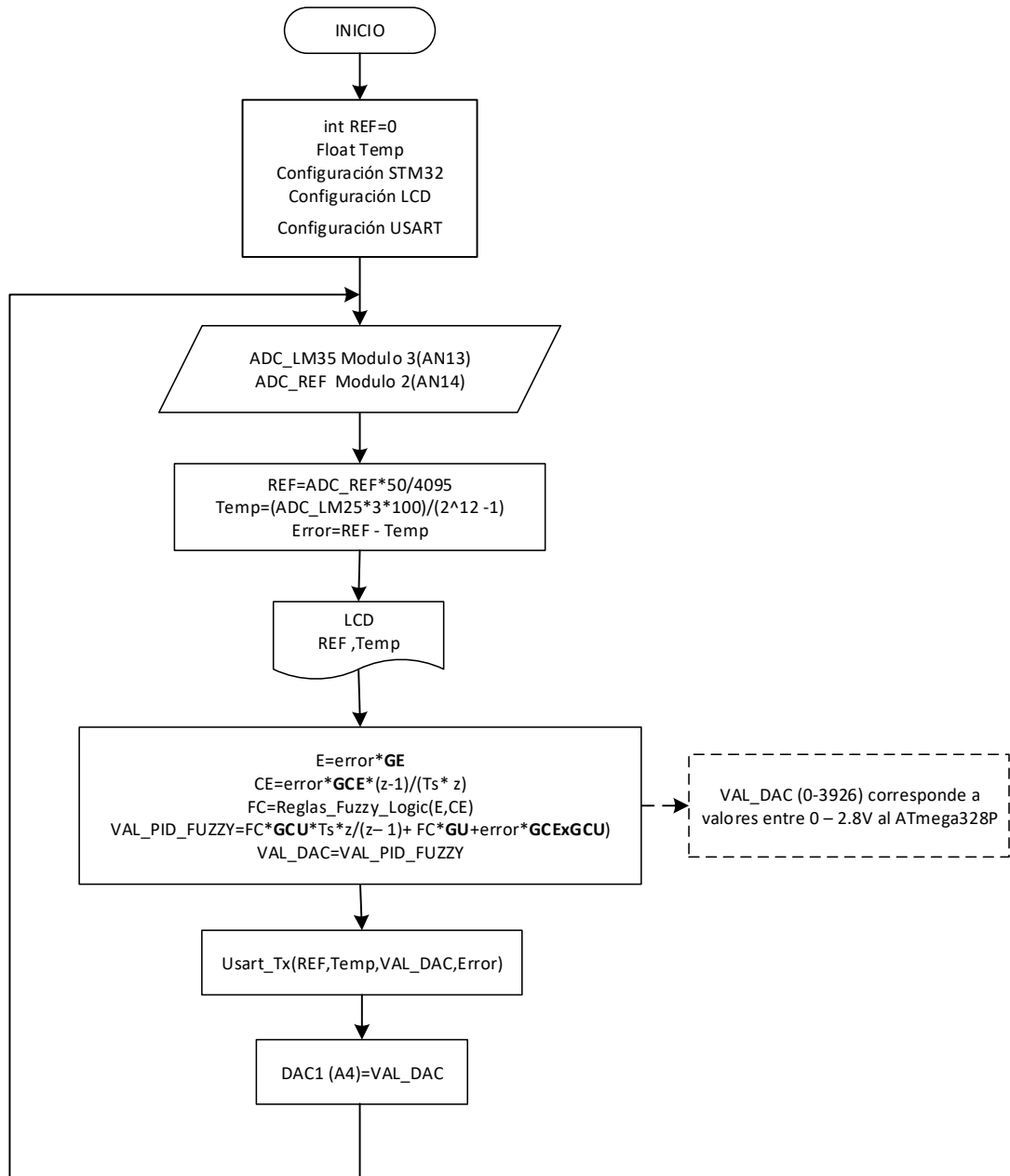
**Paso 8.** Finalmente verificamos que se encuentre en el Workspace los datos del PID\_DIFUSO



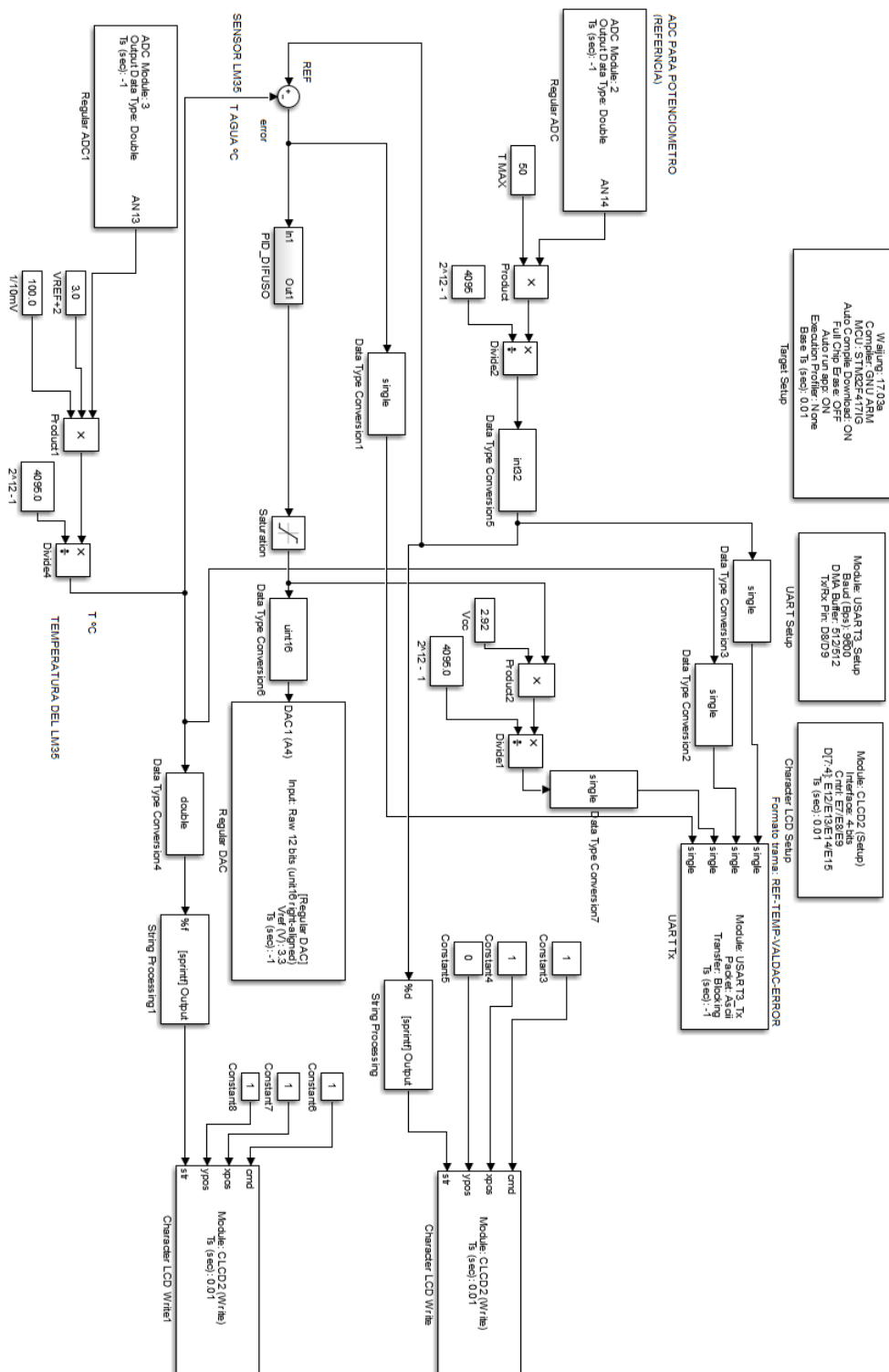
| Name ^            | Value                 |
|-------------------|-----------------------|
| GCE               | 8.8084e-04            |
| GCExGCU           | 5.4166e-04            |
| GCU               | 0.6149                |
| GE                | 1                     |
| GU                | 235.2510              |
| Kd                | 0.2072                |
| KD                | 4.8749e+03            |
| Ki                | 0.6149                |
| KI                | 1.4467                |
| Kp                | 235.2516              |
| KP                | 234.5282              |
| N                 | 83.3863               |
| <b>PID_DIFUSO</b> | <i>1x1 struct</i>     |
| T                 | 0.0100                |
| Td                | 0.2072                |
| Ti                | 1.6262                |
| tout              | <i>10001x1 double</i> |

## ANEXO 16

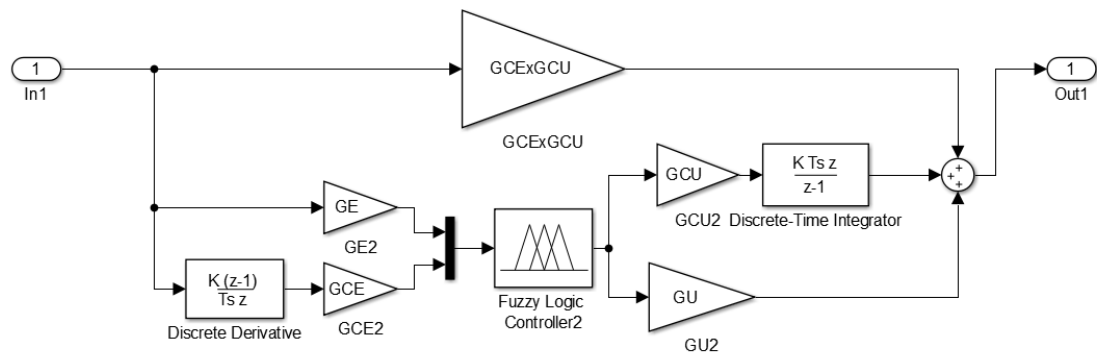
### DIAGRAMA DE FLUJO DEL PID DIFUSO



# PROGRAMA DEL PID DIFUSO EN SIMULINK

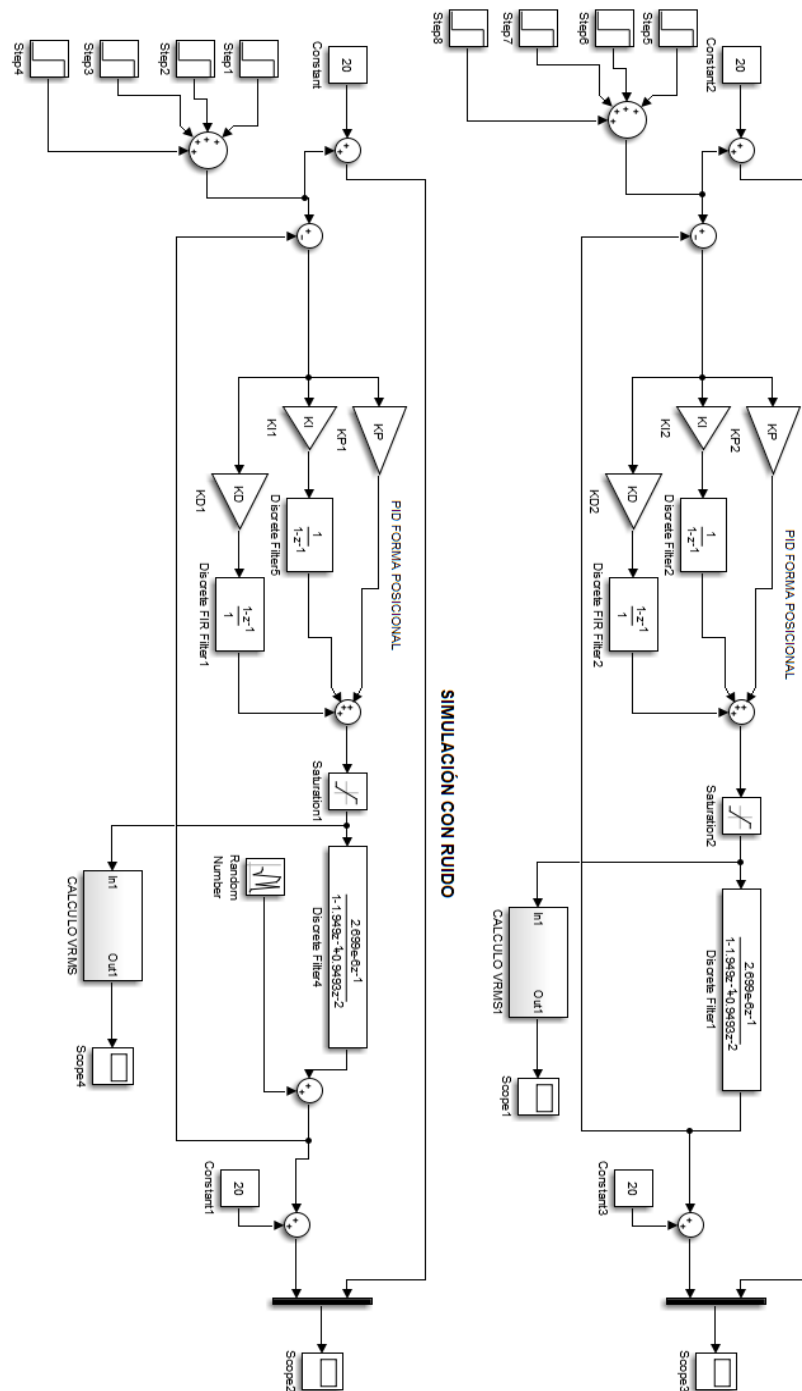


## SUBSISTEMA PID DIFUSO



## ANEXO 17

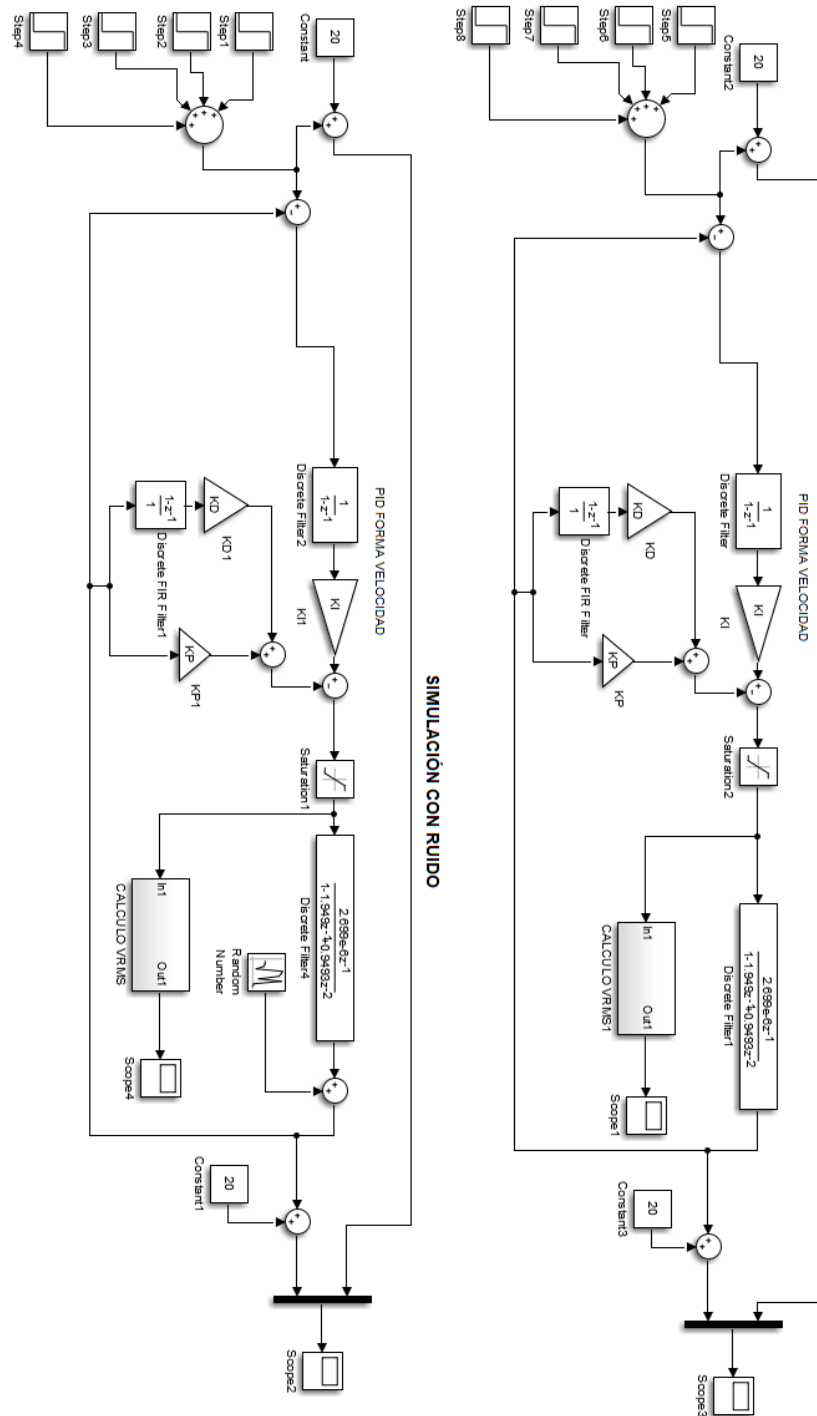
### SIMULACIÓN DEL PID POSICIONAL CON Y SIN RUIDO





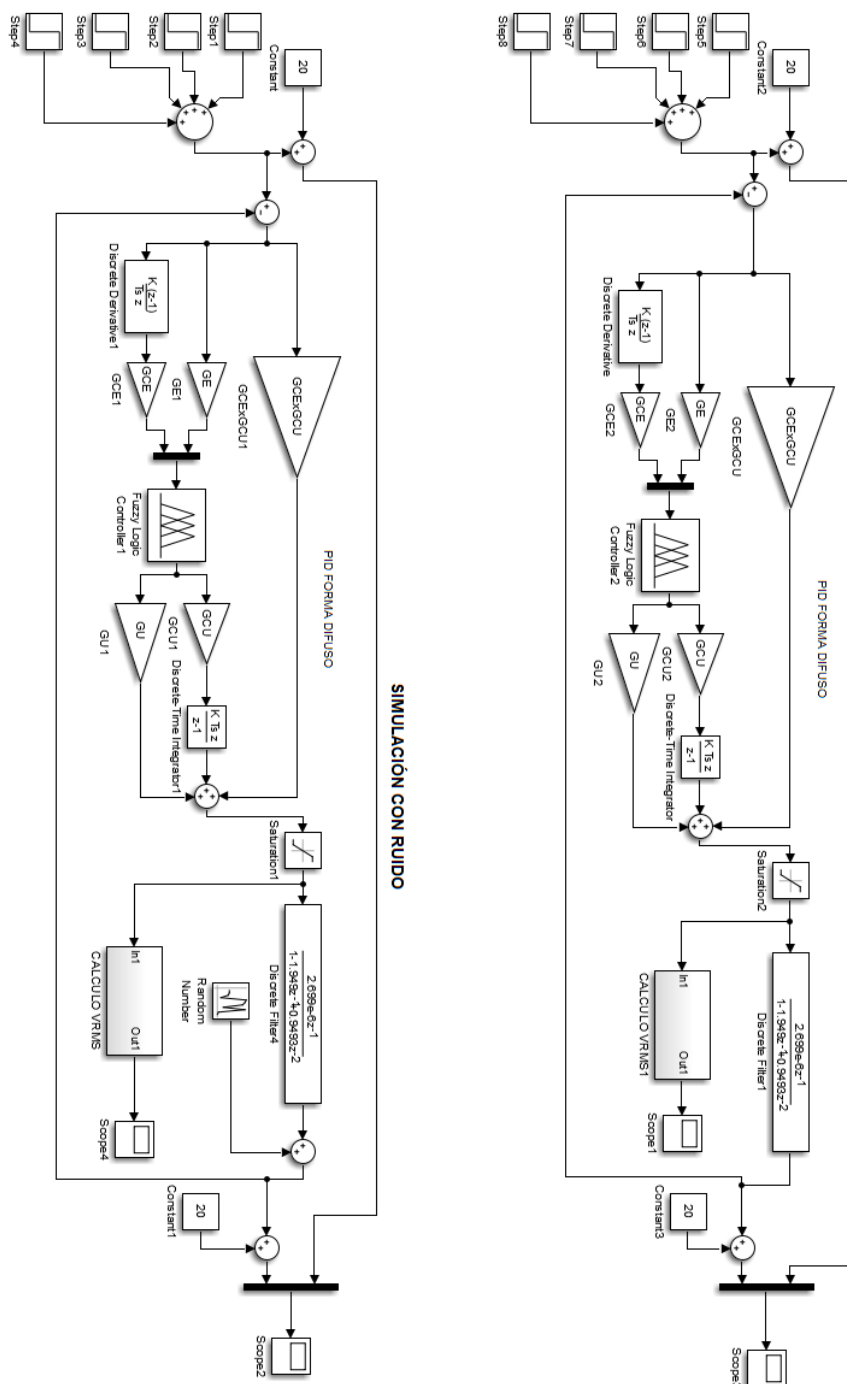
## ANEXO 18

### SIMULACIÓN DEL PID VELOCIDAD CON Y SIN RUIDO



## ANEXO 19

## SIMULACIÓN DEL PID DIFUSO CON Y SIN RUIDO

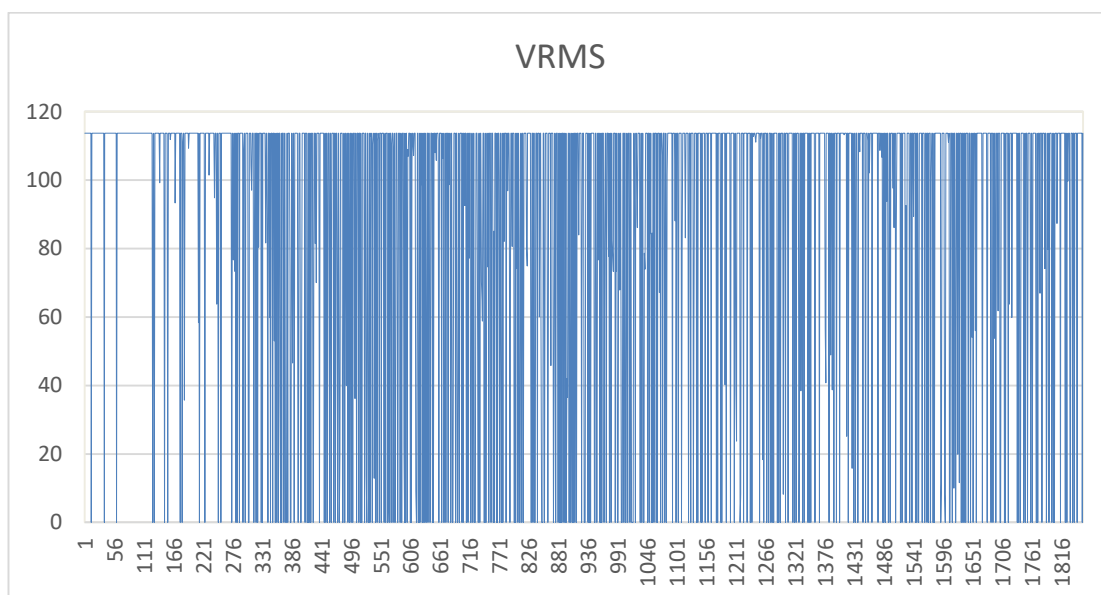
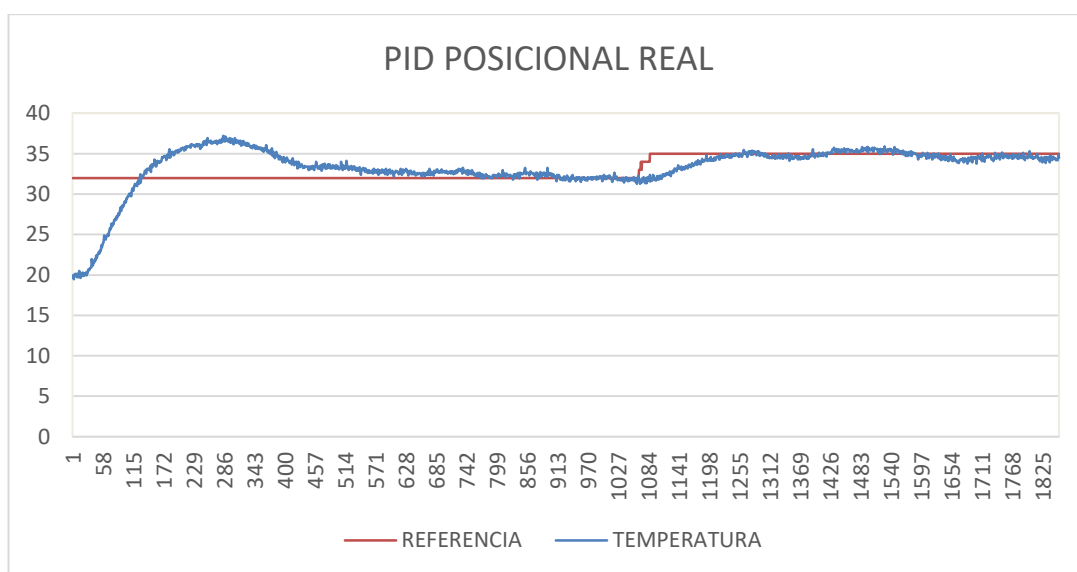


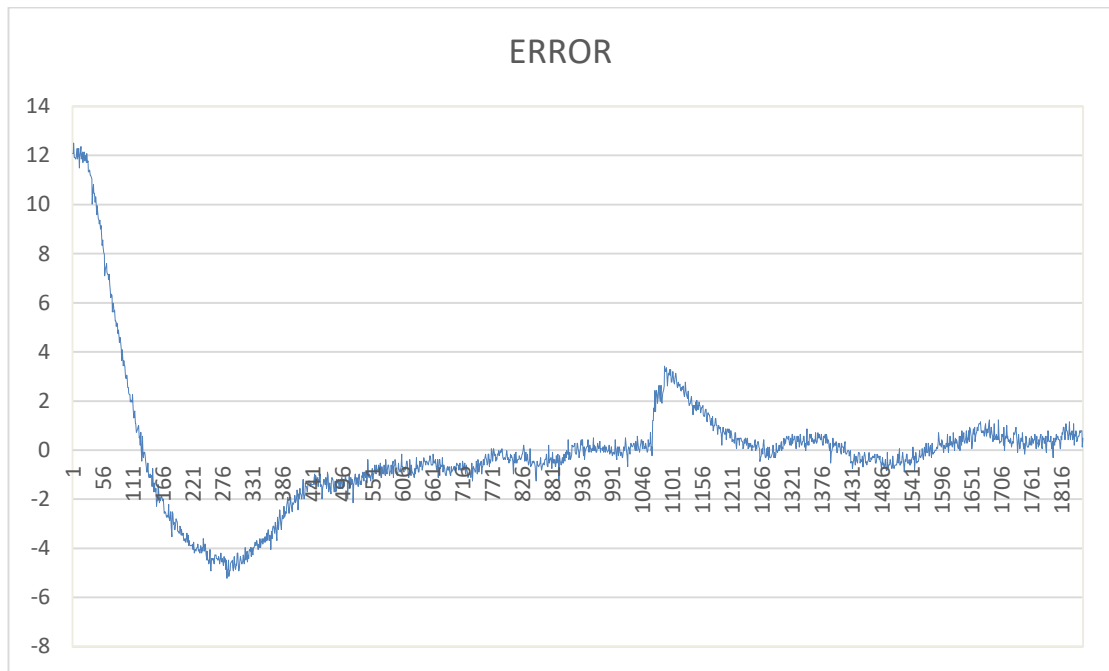
## ANEXO 20

### GRÁFICAS DEL PID POSICIONAL, VELOCIDAD, DIFUSO REALES

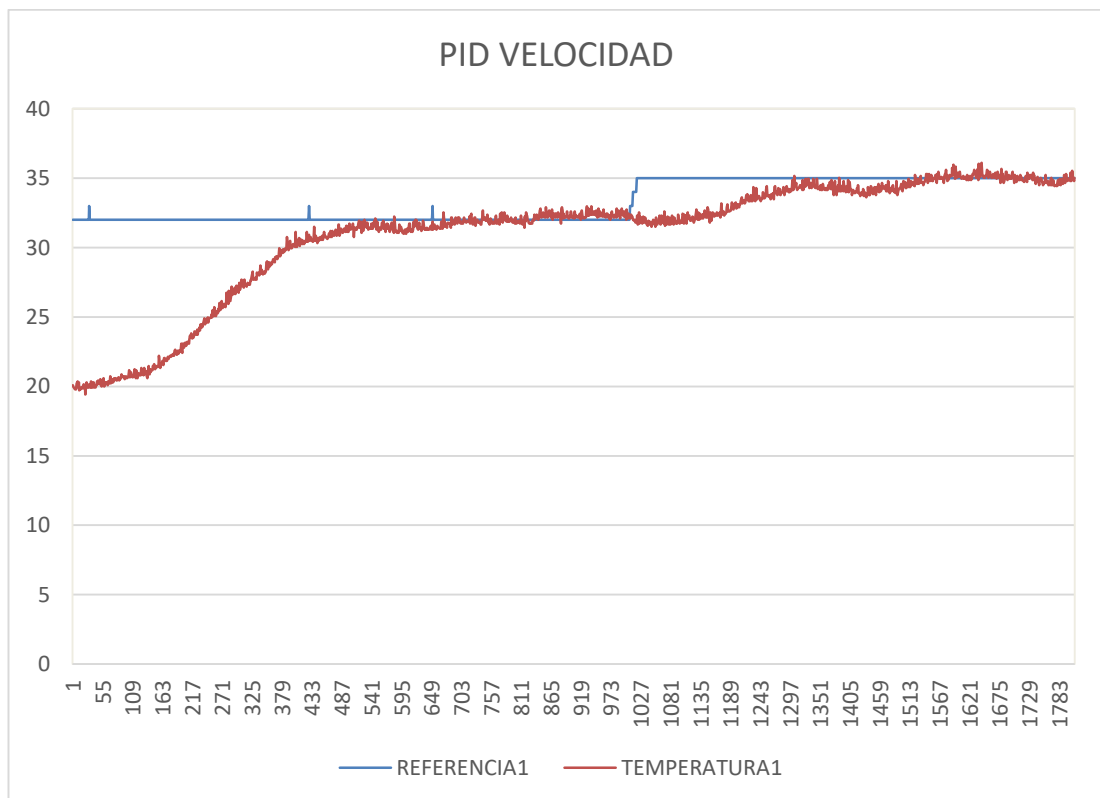
Para realizar la captura de datos se realiza conexión del sistema embebido como muestra el **anexo 9**.

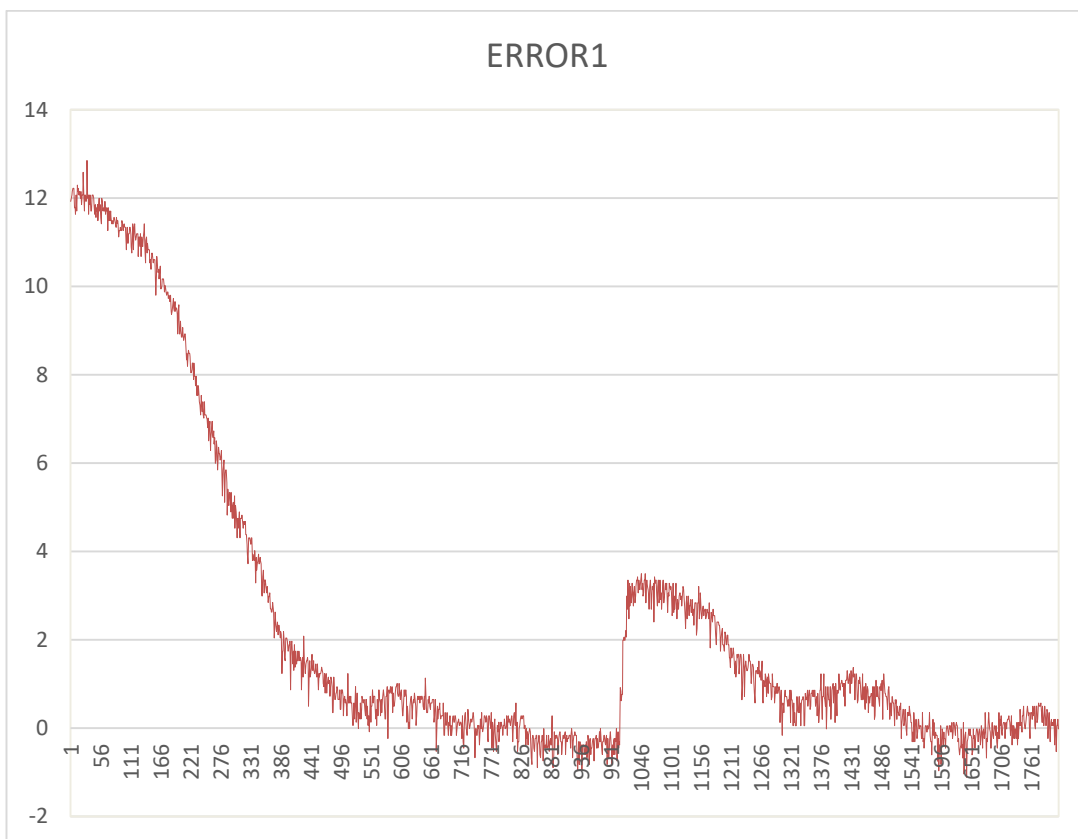
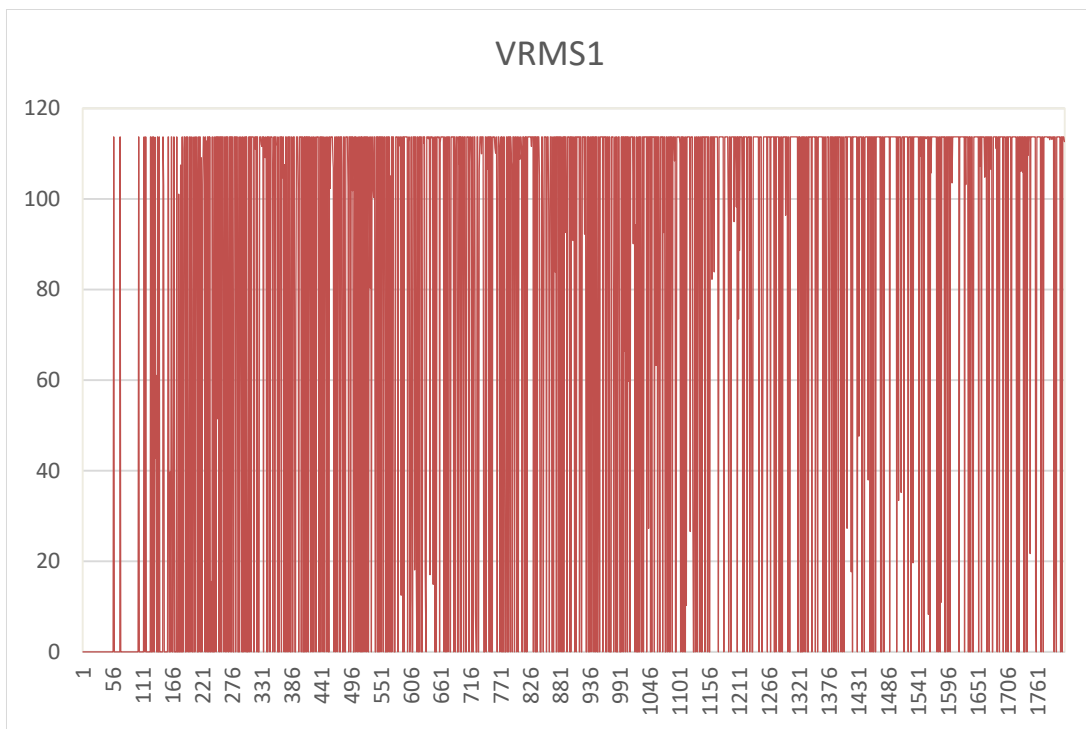
#### PID POSICIONAL

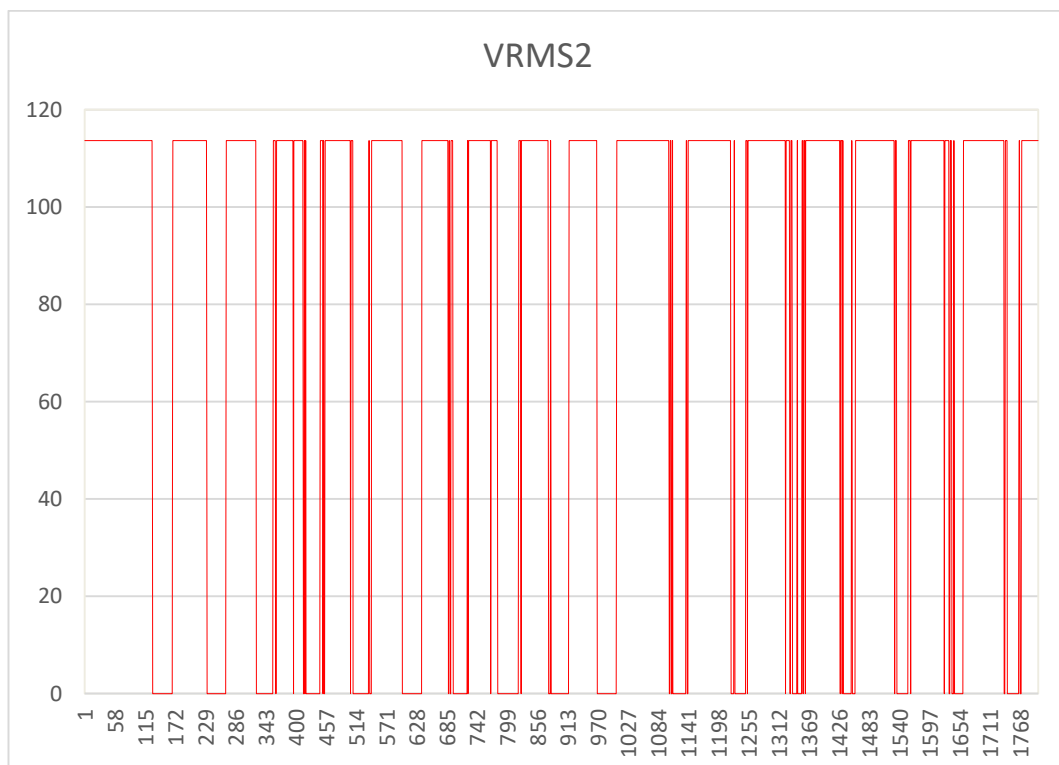
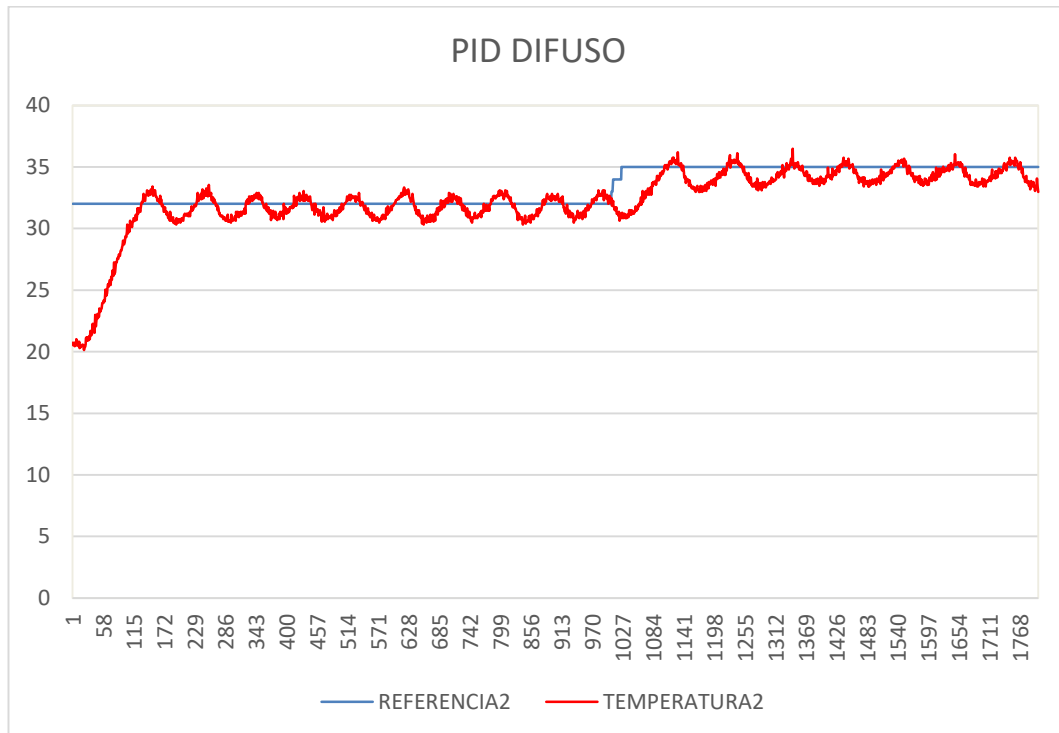


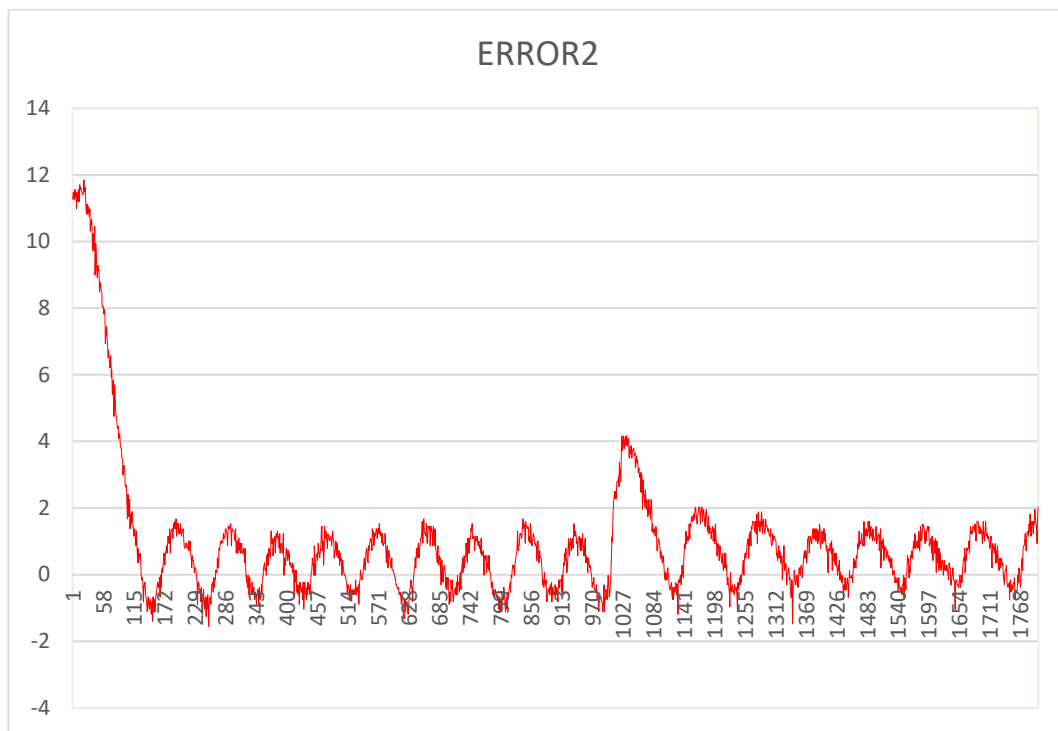


## PID VELOCIDAD





**PID DIFUSO**



## ANEXO 21

# COMPARACIÓN DEL PID POSICIONAL SIMULADO Y REAL

```

%GRAFICAS PID POSICIONAL REAL Y SIMULADO
% DATOS SIMULADOS
% SIN RUIDO
Ndatos=1800;
RESISTENCIA=3.5;
tiempo_SIM_POSICIONAL_SIN_RUIDO=PID_POSICIONAL_SIN_RUIDO(1:Ndatos,1);
REF_SIM_POSICIONAL_SIN_RUIDO=PID_POSICIONAL_SIN_RUIDO(1:Ndatos,2);
TEMP_SIM_POSICIONAL_SIN_RUIDO=PID_POSICIONAL_SIN_RUIDO(1:Ndatos,3);
VRMS_SIM_POSICIONAL_SIN_RUIDO=VRMS_POSICIONAL_SIN_RUIDO(1:Ndatos,2);
VRMS_SIM_PROMEDIO_POSICIONAL_SIN_RUIDO=sum(VRMS_SIM_POSICIONAL_SIN_RUIDO)/Ndatos;
IRMS_SIM_POSICIONAL_SIN_RUIDO=VRMS_SIM_POSICIONAL_SIN_RUIDO/RESISTENCIA;
IRMS_SIM_PROMEDIO_POSICIONAL_SIN_RUIDO=sum(IRMS_SIM_POSICIONAL_SIN_RUIDO)/Ndatos
% CON RUIDO
tiempo_SIM_POSICIONAL_CON_RUIDO=PID_POSICIONAL_CON_RUIDO(1:Ndatos,1);
REF_SIM_POSICIONAL_CON_RUIDO=PID_POSICIONAL_CON_RUIDO(1:Ndatos,2);
TEMP_SIM_POSICIONAL_CON_RUIDO=PID_POSICIONAL_CON_RUIDO(1:Ndatos,3);
VRMS_SIM_POSICIONAL_CON_RUIDO=VRMS_POSICIONAL_CON_RUIDO(1:Ndatos,2);
VRMS_SIM_PROMEDIO_POSICIONAL_CON_RUIDO=sum(VRMS_SIM_POSICIONAL_CON_RUIDO)/Ndatos;
IRMS_SIM_POSICIONAL_CON_RUIDO=VRMS_SIM_POSICIONAL_CON_RUIDO/RESISTENCIA;
IRMS_SIM_PROMEDIO_POSICIONAL_CON_RUIDO=sum(IRMS_SIM_POSICIONAL_CON_RUIDO)/Ndatos
%DATOS REALES
tiempo_REAL_POSICIONAL=Time(1:Ndatos)*0.25;
REF_REAL_POSICIONAL=REFERENCIA(1:Ndatos);
TEMP_REAL_POSICIONAL=TEMPERATURA(1:Ndatos);
VRMS_REAL_POSICIONAL=VRMS(1:Ndatos);
VRMS_REAL_PROMEDIO_POSICIONAL=sum(VRMS_REAL_POSICIONAL)/Ndatos;
IRMS_REAL_POSICIONAL=VRMS_REAL_POSICIONAL/RESISTENCIA;
IRMS_REAL_PROMEDIO_POSICIONAL=sum(IRMS_REAL_POSICIONAL)/Ndatos

%graficas de PID simulado sin ruido con real
figure(1)
plot(tiempo_SIM_POSICIONAL_SIN_RUIDO,TEMP_SIM_POSICIONAL_SIN_RUIDO,'DisplayName','TEMPERATURA
SIMULACIÓN SIN RUIDO');
hold on
plot(tiempo_SIM_POSICIONAL_CON_RUIDO,TEMP_SIM_POSICIONAL_CON_RUIDO,'DisplayName','TEMPERATURA
SIMULACIÓN CON RUIDO');
plot(tiempo_REAL_POSICIONAL,TEMP_REAL_POSICIONAL,'DisplayName','TEMPERATURA REAL');
plot(tiempo_REAL_POSICIONAL,REF_REAL_POSICIONAL,'DisplayName','REFERENCIA REAL');
plot(tiempo_SIM_POSICIONAL_SIN_RUIDO,REF_SIM_POSICIONAL_SIN_RUIDO,'DisplayName','REFERENCIA
SIMULACIÓN');
title('RESPUESTA DE LA PLANTA CON PID POSICIONAL SIMULADO SIN RUIDO Y REAL');
xlabel('TIEMPO (s)');
ylabel('TEMPERATURA (C)');
figure(2)
subplot(3,1,1);
plot(tiempo_SIM_POSICIONAL_SIN_RUIDO,VRMS_SIM_POSICIONAL_SIN_RUIDO,'DisplayName','VRMS SIMULADO
SIN RUIDO');
title('VRMS DE LA PLANTA CON PID POSICIONAL SIMULADO SIN RUIDO ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
%hold on
subplot(3,1,2);
plot(tiempo_SIM_POSICIONAL_CON_RUIDO,VRMS_SIM_POSICIONAL_CON_RUIDO,'DisplayName','VRMS SIMULADO
CON RUIDO','Color',[0.8500, 0.3250, 0.0980]);
title('VRMS DE LA PLANTA CON PID POSICIONAL SIMULADO CON RUIDO ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
subplot(3,1,3);
plot(tiempo_REAL_POSICIONAL,VRMS_REAL_POSICIONAL,'DisplayName','VRMS REAL','Color',[0.9290,
0.6940, 0.1250]);
title('VRMS DE LA PLANTA CON PID POSICIONAL REAL ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
%-----
%-----GRAFICAS DE CORRIENTE-----
figure(3)
subplot(3,1,1);

```



```

plot(tiempo_SIM_POSICIONAL_SIN_RUIDO,IRMS_SIM_POSICIONAL_SIN_RUIDO,'DisplayName','IRMS SIMULADO
SIN RUIDO');
title('IRMS DE LA PLANTA CON PID POSICIONAL SIMULADO SIN RUIDO ');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
subplot(3,1,2);
plot(tiempo_SIM_POSICIONAL_CON_RUIDO,IRMS_SIM_POSICIONAL_CON_RUIDO,'DisplayName','IRMS SIMULADO
CON RUIDO','Color',[0.8500, 0.3250, 0.0980]);
title('IRMS DE LA PLANTA CON PID POSICIONAL SIMULADO CON RUIDO ');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
subplot(3,1,3);
plot(tiempo_REAL_POSICIONAL,IRMS_REAL_POSICIONAL,'DisplayName','IRMS REAL','Color',[0.9290,
0.6940, 0.1250]);
title('IRMS DE LA PLANTA CON PID POSICIONAL REAL');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
%-----
% OVERSHOOT REFERENCIA INICIAL 32 C
Novershoot=900;
figure(4)
tiempo_REAL_POSICIONAL_OVERSHOOT1=tiempo_REAL_POSICIONAL(1:Novershoot);
TEMP_REAL_POSICIONAL_OVERSHOOT1=TEMP_REAL_POSICIONAL(1:Novershoot);
%settlingtime(TEMP_REAL_POSICIONAL,4,0.25)
[os1] =
overshoot(TEMP_REAL_POSICIONAL_OVERSHOOT1,tiempo_REAL_POSICIONAL_OVERSHOOT1,'StateLevels',[20
32],'Tolerance',2);
overshoot_TEMP_REAL_POSICIONAL_32_REF=os1
overshoot(TEMP_REAL_POSICIONAL_OVERSHOOT1,tiempo_REAL_POSICIONAL_OVERSHOOT1,'StateLevels',[20
32],'Tolerance',2);
%-----
figure(5)
tiempo_SIM_POSICIONAL_CON_RUIDO_OVERSHOOT=tiempo_SIM_POSICIONAL_CON_RUIDO(1:Novershoot);
TEMP_SIM_POSICIONAL_CON_RUIDO_OVERSHOOT=TEMP_SIM_POSICIONAL_CON_RUIDO(1:Novershoot);

[os2] =
overshoot(TEMP_SIM_POSICIONAL_CON_RUIDO_OVERSHOOT,tiempo_SIM_POSICIONAL_CON_RUIDO_OVERSHOOT,'State
Levels',[20 32],'Tolerance',2);
overshoot_TEMP_SIM_POSICIONAL_CON_RUIDO_32_REF=os2
overshoot(TEMP_SIM_POSICIONAL_CON_RUIDO_OVERSHOOT,tiempo_SIM_POSICIONAL_CON_RUIDO_OVERSHOOT,'State
Levels',[20 32],'Tolerance',2);

%-----
figure(6)
tiempo_SIM_POSICIONAL_SIN_RUIDO_OVERSHOOT=tiempo_SIM_POSICIONAL_SIN_RUIDO(1:Novershoot);
TEMP_SIM_POSICIONAL_SIN_RUIDO_OVERSHOOT=TEMP_SIM_POSICIONAL_SIN_RUIDO(1:Novershoot);

[os3] =
overshoot(TEMP_SIM_POSICIONAL_SIN_RUIDO_OVERSHOOT,tiempo_SIM_POSICIONAL_SIN_RUIDO_OVERSHOOT,'State
Levels',[20 32],'Tolerance',2);
overshoot_TEMP_SIM_POSICIONAL_SIN_RUIDO_32_REF=os3
overshoot(TEMP_SIM_POSICIONAL_SIN_RUIDO_OVERSHOOT,tiempo_SIM_POSICIONAL_SIN_RUIDO_OVERSHOOT,'State
Levels',[20 32],'Tolerance',2);

```

## ANEXO 22

# COMPARACIÓN DEL PID VELOCIDAD SIMULADO Y REAL

```

%GRAFICAS PID VELOCIDAD REAL Y SIMULADO
% DATOS SIMULADOS
% SIN RUIDO
Ndatos=1800;
RESISTENCIA=3.5;
tiempo_SIM_VELOCIDAD_SIN_RUIDO=PID_VELOCIDAD_SIN_RUIDO(1:Ndatos,1);
REF_SIM_VELOCIDAD_SIN_RUIDO=PID_VELOCIDAD_SIN_RUIDO(1:Ndatos,2);
TEMP_SIM_VELOCIDAD_SIN_RUIDO=PID_VELOCIDAD_SIN_RUIDO(1:Ndatos,3);
VRMS_SIM_VELOCIDAD_SIN_RUIDO=VRMS_VELOCIDAD_SIN_RUIDO(1:Ndatos,2);
VRMS_SIM_PROMEDIO_VELOCIDAD_SIN_RUIDO=sum(VRMS_SIM_VELOCIDAD_SIN_RUIDO)/Ndatos
IRMS_SIM_VELOCIDAD_SIN_RUIDO=VRMS_SIM_VELOCIDAD_SIN_RUIDO/RESISTENCIA;
IRMS_SIM_PROMEDIO_VELOCIDAD_SIN_RUIDO=sum(IRMS_SIM_VELOCIDAD_SIN_RUIDO)/Ndatos
% CON RUIDO
tiempo_SIM_VELOCIDAD_CON_RUIDO=PID_VELOCIDAD_CON_RUIDO(1:Ndatos,1);
REF_SIM_VELOCIDAD_CON_RUIDO=PID_VELOCIDAD_CON_RUIDO(1:Ndatos,2);
TEMP_SIM_VELOCIDAD_CON_RUIDO=PID_VELOCIDAD_CON_RUIDO(1:Ndatos,3);
VRMS_SIM_VELOCIDAD_CON_RUIDO=VRMS_VELOCIDAD_CON_RUIDO(1:Ndatos,2);
VRMS_SIM_PROMEDIO_VELOCIDAD_CON_RUIDO=sum(VRMS_SIM_VELOCIDAD_CON_RUIDO)/Ndatos
IRMS_SIM_VELOCIDAD_CON_RUIDO=VRMS_SIM_VELOCIDAD_CON_RUIDO/RESISTENCIA;
IRMS_SIM_PROMEDIO_VELOCIDAD_CON_RUIDO=sum(IRMS_SIM_VELOCIDAD_CON_RUIDO)/Ndatos
%DATOS REALES
tiempo_REAL_VELOCIDAD=Time1(1:Ndatos)*0.25;
REF_REAL_VELOCIDAD=REFERENCIA1(1:Ndatos);
TEMP_REAL_VELOCIDAD=TEMPERATURA1(1:Ndatos);
VRMS_REAL_VELOCIDAD=VRMS1(1:Ndatos);
VRMS_REAL_PROMEDIO_VELOCIDAD=sum(VRMS_REAL_VELOCIDAD)/Ndatos
IRMS_REAL_VELOCIDAD=VRMS_REAL_VELOCIDAD/RESISTENCIA;
IRMS_REAL_PROMEDIO_VELOCIDAD=sum(IRMS_REAL_VELOCIDAD)/Ndatos

%graficas de PID simulado sin ruido con real
figure(1)
plot(tiempo_SIM_VELOCIDAD_SIN_RUIDO,TEMP_SIM_VELOCIDAD_SIN_RUIDO,'DisplayName','TEMPERATURA
SIMULACIÓN SIN RUIDO');
hold on
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,TEMP_SIM_VELOCIDAD_CON_RUIDO,'DisplayName','TEMPERATURA
SIMULACIÓN CON RUIDO');
plot(tiempo_REAL_VELOCIDAD,TEMP_REAL_VELOCIDAD,'DisplayName','TEMPERATURA REAL');
plot(tiempo_REAL_VELOCIDAD,REF_REAL_VELOCIDAD,'DisplayName','REFERENCIA REAL');
plot(tiempo_SIM_VELOCIDAD_SIN_RUIDO,REF_SIM_VELOCIDAD_SIN_RUIDO,'DisplayName','REFERENCIA
SIMULACIÓN');
title('RESPUESTA DE LA PLANTA CON PID VELOCIDAD SIMULADO SIN RUIDO Y REAL');
xlabel('TIEMPO (s)');
ylabel('TEMPERATURA (C)');
%-----
figure(2)
subplot(3,1,1);
plot(tiempo_SIM_VELOCIDAD_SIN_RUIDO,VRMS_SIM_VELOCIDAD_SIN_RUIDO,'DisplayName','VRMS SIMULADO SIN
RUIDO');
title('VRMS DE LA PLANTA CON PID VELOCIDAD SIMULADO SIN RUIDO ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
subplot(3,1,2);
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,VRMS_SIM_VELOCIDAD_CON_RUIDO,'DisplayName','VRMS SIMULADO CON
RUIDO','Color',[0.8500, 0.3250, 0.0980]);
title('VRMS DE LA PLANTA CON PID VELOCIDAD SIMULADO CON RUIDO ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
subplot(3,1,3);
plot(tiempo_REAL_VELOCIDAD,VRMS_REAL_VELOCIDAD,'DisplayName','VRMS REAL','Color',[0.9290, 0.6940,
0.1250]);
title('VRMS DE LA PLANTA CON PID VELOCIDAD REAL ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');

%-----GRAFICAS DE CORRIENTE-----
figure(3)
subplot(3,1,1);

```

```

plot(tiempo_SIM_VELOCIDAD_SIN_RUIDO,IRMS_SIM_VELOCIDAD_SIN_RUIDO,'DisplayName','IRMS SIMULADO SIN
RUIDO');
title('IRMS DE LA PLANTA CON PID VELOCIDAD SIMULADO SIN RUIDO ');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
subplot(3,1,2);
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,IRMS_SIM_VELOCIDAD_CON_RUIDO,'DisplayName','IRMS SIMULADO CON
RUIDO','Color',[0.8500, 0.3250, 0.0980]);
title('IRMS DE LA PLANTA CON PID VELOCIDAD SIMULADO CON RUIDO ');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
subplot(3,1,3);
plot(tiempo_REAL_VELOCIDAD,IRMS_REAL_VELOCIDAD,'DisplayName','IRMS REAL','Color',[0.9290, 0.6940,
0.1250]);
title('IRMS DE LA PLANTA CON PID VELOCIDAD REAL');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
%-----
% OVERSHOOT REFERENCIA INICIAL 32 C
Novershoot=900;
figure(4)
tiempo_REAL_VELOCIDAD_OVERSHOOT1=tiempo_REAL_VELOCIDAD(1:Novershoot);
TEMP_REAL_VELOCIDAD_OVERSHOOT1=TEMP_REAL_VELOCIDAD(1:Novershoot);
%settlingtime(TEMP_REAL_POSICIONAL,4,0.25)
[os1] =
overshoot(TEMP_REAL_VELOCIDAD_OVERSHOOT1,tiempo_REAL_VELOCIDAD_OVERSHOOT1,'StateLevels',[20
32],'Tolerance',2);
overshoot_TEMP_REAL_VELOCIDAD_32_REF=os1
overshoot(TEMP_REAL_VELOCIDAD_OVERSHOOT1,tiempo_REAL_VELOCIDAD_OVERSHOOT1,'StateLevels',[20
32],'Tolerance',2);
%-----
figure(5)
tiempo_SIM_VELOCIDAD_CON_RUIDO_OVERSHOOT=tiempo_SIM_VELOCIDAD_CON_RUIDO(1:Novershoot);
TEMP_SIM_VELOCIDAD_CON_RUIDO_OVERSHOOT=TEMP_SIM_VELOCIDAD_CON_RUIDO(1:Novershoot);

[os2] =
overshoot(TEMP_SIM_VELOCIDAD_CON_RUIDO_OVERSHOOT,tiempo_SIM_VELOCIDAD_CON_RUIDO_OVERSHOOT,'StateLe
vels',[20 32],'Tolerance',2);
overshoot_TEMP_SIM_VELOCIDAD_CON_RUIDO_32_REF=os2
overshoot(TEMP_SIM_VELOCIDAD_CON_RUIDO_OVERSHOOT,tiempo_SIM_VELOCIDAD_CON_RUIDO_OVERSHOOT,'StateLe
vels',[20 32],'Tolerance',2);
%-----
figure(6)
tiempo_SIM_VELOCIDAD_SIN_RUIDO_OVERSHOOT=tiempo_SIM_VELOCIDAD_SIN_RUIDO(1:Novershoot);
TEMP_SIM_VELOCIDAD_SIN_RUIDO_OVERSHOOT=TEMP_SIM_VELOCIDAD_SIN_RUIDO(1:Novershoot);

[os3] =
overshoot(TEMP_SIM_VELOCIDAD_SIN_RUIDO_OVERSHOOT,tiempo_SIM_VELOCIDAD_SIN_RUIDO_OVERSHOOT,'StateLe
vels',[20 32],'Tolerance',2);
overshoot_TEMP_SIM_VELOCIDAD_SIN_RUIDO_32_REF=os3
overshoot(TEMP_SIM_VELOCIDAD_SIN_RUIDO_OVERSHOOT,tiempo_SIM_VELOCIDAD_SIN_RUIDO_OVERSHOOT,'StateLe
vels',[20 32],'Tolerance',2);

```

## ANEXO 23

# COMPARACIÓN DEL PID DIFUSO SIMULADO Y REAL

```

%GRAFICAS PID DIFUSO REAL Y SIMULADO
% DATOS SIMULADOS
% SIN RUIDO
Ndatos=1800;
RESISTENCIA=3.5;
tiempo_SIM_DIFUSO_SIN_RUIDO=PID_DIFUSO_SIN_RUIDO(1:Ndatos,1);
REF_SIM_DIFUSO_SIN_RUIDO=PID_DIFUSO_SIN_RUIDO(1:Ndatos,2);
TEMP_SIM_DIFUSO_SIN_RUIDO=PID_DIFUSO_SIN_RUIDO(1:Ndatos,3);
VRMS_SIM_DIFUSO_SIN_RUIDO=VRMS_DIFUSO_SIN_RUIDO(1:Ndatos,2);
VRMS_SIM_PROMEDIO_DIFUSO_SIN_RUIDO=sum(VRMS_SIM_DIFUSO_SIN_RUIDO)/Ndatos
IRMS_SIM_DIFUSO_SIN_RUIDO=VRMS_SIM_DIFUSO_SIN_RUIDO/RESISTENCIA;
IRMS_SIM_PROMEDIO_DIFUSO_SIN_RUIDO=sum(IRMS_SIM_DIFUSO_SIN_RUIDO)/Ndatos
% CON RUIDO
tiempo_SIM_DIFUSO_CON_RUIDO=PID_DIFUSO_CON_RUIDO(1:Ndatos,1);
REF_SIM_DIFUSO_CON_RUIDO=PID_DIFUSO_CON_RUIDO(1:Ndatos,2);
TEMP_SIM_DIFUSO_CON_RUIDO=PID_DIFUSO_CON_RUIDO(1:Ndatos,3);
VRMS_SIM_DIFUSO_CON_RUIDO=VRMS_DIFUSO_CON_RUIDO(1:Ndatos,2);
VRMS_SIM_PROMEDIO_DIFUSO_CON_RUIDO=sum(VRMS_SIM_DIFUSO_CON_RUIDO)/Ndatos
IRMS_SIM_DIFUSO_CON_RUIDO=VRMS_SIM_DIFUSO_CON_RUIDO/RESISTENCIA;
IRMS_SIM_PROMEDIO_DIFUSO_CON_RUIDO=sum(IRMS_SIM_DIFUSO_CON_RUIDO)/Ndatos
%DATOS REALES
tiempo_REAL_DIFUSO=Time2(1:Ndatos)*0.25;
REF_REAL_DIFUSO=REFERENCIA2(1:Ndatos);
TEMP_REAL_DIFUSO=TEMPERATURA2(1:Ndatos);
VRMS_REAL_DIFUSO=VRMS2(1:Ndatos);
VRMS_REAL_PROMEDIO_DIFUSO=sum(VRMS_REAL_DIFUSO)/Ndatos
IRMS_REAL_DIFUSO=VRMS_REAL_DIFUSO/RESISTENCIA;
IRMS_REAL_PROMEDIO_DIFUSO=sum(IRMS_REAL_DIFUSO)/Ndatos
%Gráficas de PID Simulado Real
figure(1)
plot(tiempo_SIM_DIFUSO_SIN_RUIDO,TEMP_SIM_DIFUSO_SIN_RUIDO,'DisplayName','TEMPERATURA SIMULACIÓN SIN RUIDO');
hold on
plot(tiempo_SIM_DIFUSO_CON_RUIDO,TEMP_SIM_DIFUSO_CON_RUIDO,'DisplayName','TEMPERATURA SIMULACIÓN CON RUIDO');
plot(tiempo_REAL_DIFUSO,TEMP_REAL_DIFUSO,'DisplayName','TEMPERATURA REAL');
plot(tiempo_REAL_DIFUSO,REF_REAL_DIFUSO,'DisplayName','REFERENCIA REAL');
plot(tiempo_REAL_DIFUSO,REF_SIM_DIFUSO_SIN_RUIDO,'DisplayName','REFERENCIA SIMULACIÓN');
title('RESPUESTA DE LA PLANTA CON PID DIFUSO SIMULADOS Y REAL');
xlabel('TIEMPO (s)');
ylabel('TEMPERATURA (C)');
%-----
figure(2)
subplot(3,1,1);
plot(tiempo_SIM_DIFUSO_SIN_RUIDO,VRMS_SIM_DIFUSO_SIN_RUIDO,'DisplayName','VRMS SIMULADO SIN RUIDO');
title('VRMS DE LA PLANTA CON PID DIFUSO SIMULADO SIN RUIDO ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
subplot(3,1,2);
plot(tiempo_SIM_DIFUSO_CON_RUIDO,VRMS_SIM_DIFUSO_CON_RUIDO,'DisplayName','VRMS SIMULADO CON RUIDO','Color',[0.8500, 0.3250, 0.0980]);
title('VRMS DE LA PLANTA CON PID DIFUSO SIMULADO CON RUIDO ');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
subplot(3,1,3);
plot(tiempo_REAL_DIFUSO,VRMS_REAL_DIFUSO,'DisplayName','VRMS REAL','Color',[0.9290, 0.6940, 0.1250]);
title('VRMS DE LA PLANTA CON PID DIFUSO REAL');
xlabel('TIEMPO (s)');
ylabel('VRMS (V)');
%-----
%-----GRAFICAS DE CORRIENTE-----
figure(3)
subplot(3,1,1);
plot(tiempo_SIM_DIFUSO_SIN_RUIDO,IRMS_SIM_DIFUSO_SIN_RUIDO,'DisplayName','IRMS SIMULADO SIN RUIDO');
title('IRMS DE LA PLANTA CON PID DIFUSO SIMULADO SIN RUIDO ');
xlabel('TIEMPO (s)');

```

```

ylabel('IRMS(A)');
subplot(3,1,2);
plot(tiempo_SIM_DIFUSO_CON_RUIDO,IRMS_SIM_DIFUSO_CON_RUIDO,'DisplayName','IRMS SIMULADO CON
RUIDO','Color',[0.8500, 0.3250, 0.0980]);
title('IRMS DE LA PLANTA CON PID DIFUSO SIMULADO CON RUIDO ');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
subplot(3,1,3);
plot(tiempo_REAL_DIFUSO,IRMS_REAL_DIFUSO,'DisplayName','IRMS REAL','Color',[0.9290, 0.6940,
0.1250]);
title('IRMS DE LA PLANTA CON PID DIFUSO REAL');
xlabel('TIEMPO (s)');
ylabel('IRMS(A)');
%-----
% OVERSHOOT REFERENCIA INICIAL 32 C
Novershoot=900;
figure(4)
tiempo_REAL_DIFUSO_OVERSHOOT1=tiempo_REAL_DIFUSO(1:Novershoot);
TEMP_REAL_DIFUSO_OVERSHOOT1=TEMP_REAL_DIFUSO(1:Novershoot);
%settlingtime(TEMP_REAL_POSICIONAL,4,0.25)
[os1] = overshoot(TEMP_REAL_DIFUSO_OVERSHOOT1,tiempo_REAL_DIFUSO_OVERSHOOT1,'StateLevels',[20
32],'Tolerance',2);
overshoot_TEMP_REAL_DIFUSO_32_REF=os1
overshoot(TEMP_REAL_DIFUSO_OVERSHOOT1,tiempo_REAL_DIFUSO_OVERSHOOT1,'StateLevels',[20
32],'Tolerance',2);
%-----
figure(5)
tiempo_SIM_DIFUSO_CON_RUIDO_OVERSHOOT=tiempo_SIM_DIFUSO_CON_RUIDO(1:Novershoot);
TEMP_SIM_DIFUSO_CON_RUIDO_OVERSHOOT=TEMP_SIM_DIFUSO_CON_RUIDO(1:Novershoot);

[os2] =
overshoot(TEMP_SIM_DIFUSO_CON_RUIDO_OVERSHOOT,tiempo_SIM_DIFUSO_CON_RUIDO_OVERSHOOT,'StateLevels',
[20 32],'Tolerance',2);
overshoot_TEMP_SIM_DIFUSO_CON_RUIDO_32_REF=os2
overshoot(TEMP_SIM_DIFUSO_CON_RUIDO_OVERSHOOT,tiempo_SIM_DIFUSO_CON_RUIDO_OVERSHOOT,'StateLevels',
[20 32],'Tolerance',2);

%-----
figure(6)
tiempo_SIM_DIFUSO_SIN_RUIDO_OVERSHOOT=tiempo_SIM_DIFUSO_SIN_RUIDO(1:Novershoot);
TEMP_SIM_DIFUSO_SIN_RUIDO_OVERSHOOT=TEMP_SIM_DIFUSO_SIN_RUIDO(1:Novershoot);

[os3] =
overshoot(TEMP_SIM_DIFUSO_SIN_RUIDO_OVERSHOOT,tiempo_SIM_DIFUSO_SIN_RUIDO_OVERSHOOT,'StateLevels',
[20 32],'Tolerance',2);
overshoot_TEMP_SIM_DIFUSO_SIN_RUIDO_32_REF=os3
overshoot(TEMP_SIM_DIFUSO_SIN_RUIDO_OVERSHOOT,tiempo_SIM_DIFUSO_SIN_RUIDO_OVERSHOOT,'StateLevels',
[20 32],'Tolerance',2);

```

## ANEXO 24

### COMPARACIÓN DEL PID POSICIONAL, VELOCIDAD, DIFUSO SIMULADO Y REAL

```

%COMPARACION PID POSICIONAL VELOCIDAD DIFUSO REAL Y SIMULADO
% DATOS SIMULADOS
Ndatos=1000;
RESISTENCIA=3.5;
%----- SIN RUIDO
%+++++
tiempo_SIM_POSICIONAL_SIN_RUIDO=PID_POSICIONAL_SIN_RUIDO(1:Ndatos,1);
REF_SIM_POSICIONAL_SIN_RUIDO=PID_POSICIONAL_SIN_RUIDO(1:Ndatos,2);
TEMP_SIM_POSICIONAL_SIN_RUIDO=PID_POSICIONAL_SIN_RUIDO(1:Ndatos,3);
%+++++
tiempo_SIM_VELOCIDAD_SIN_RUIDO=PID_VELOCIDAD_SIN_RUIDO(1:Ndatos,1);
REF_SIM_VELOCIDAD_SIN_RUIDO=PID_VELOCIDAD_SIN_RUIDO(1:Ndatos,2);
TEMP_SIM_VELOCIDAD_SIN_RUIDO=PID_VELOCIDAD_SIN_RUIDO(1:Ndatos,3);
%+++++
tiempo_SIM_DIFUSO_SIN_RUIDO=PID_DIFUSO_SIN_RUIDO(1:Ndatos,1);
REF_SIM_DIFUSO_SIN_RUIDO=PID_DIFUSO_SIN_RUIDO(1:Ndatos,2);
TEMP_SIM_DIFUSO_SIN_RUIDO=PID_DIFUSO_SIN_RUIDO(1:Ndatos,3);

%----- CON RUIDO
%+++++
tiempo_SIM_POSICIONAL_CON_RUIDO=PID_POSICIONAL_CON_RUIDO(1:Ndatos,1);
REF_SIM_POSICIONAL_CON_RUIDO=PID_POSICIONAL_CON_RUIDO(1:Ndatos,2);
TEMP_SIM_POSICIONAL_CON_RUIDO=PID_POSICIONAL_CON_RUIDO(1:Ndatos,3);
%+++++
tiempo_SIM_VELOCIDAD_CON_RUIDO=PID_VELOCIDAD_CON_RUIDO(1:Ndatos,1);
REF_SIM_VELOCIDAD_CON_RUIDO=PID_VELOCIDAD_CON_RUIDO(1:Ndatos,2);
TEMP_SIM_VELOCIDAD_CON_RUIDO=PID_VELOCIDAD_CON_RUIDO(1:Ndatos,3);
%+++++
tiempo_SIM_DIFUSO_CON_RUIDO=PID_DIFUSO_CON_RUIDO(1:Ndatos,1);
REF_SIM_DIFUSO_CON_RUIDO=PID_DIFUSO_CON_RUIDO(1:Ndatos,2);
TEMP_SIM_DIFUSO_CON_RUIDO=PID_DIFUSO_CON_RUIDO(1:Ndatos,3);
%----- DATOS REALES
%+++++
tiempo_REAL_POSICIONAL=Time(1:Ndatos)*0.25;
REF_REAL_POSICIONAL=REFERENCIA(1:Ndatos);
TEMP_REAL_POSICIONAL=TEMPERATURA(1:Ndatos);
%+++++
tiempo_REAL_VELOCIDAD=Time1(1:Ndatos)*0.25;
REF_REAL_VELOCIDAD=REFERENCIA1(1:Ndatos);
TEMP_REAL_VELOCIDAD=TEMPERATURA1(1:Ndatos);
%+++++
tiempo_REAL_DIFUSO=Time2(1:Ndatos)*0.25;
REF_REAL_DIFUSO=REFERENCIA2(1:Ndatos);
TEMP_REAL_DIFUSO=TEMPERATURA2(1:Ndatos);
%graficas de PID simulado sin ruido con real
figure(1)
plot(tiempo_SIM_POSICIONAL_SIN_RUIDO,TEMP_SIM_POSICIONAL_SIN_RUIDO,'DisplayName','PID POSICIONAL');
hold on
plot(tiempo_SIM_VELOCIDAD_SIN_RUIDO,TEMP_SIM_VELOCIDAD_SIN_RUIDO,'DisplayName','PID VELOCIDAD');
plot(tiempo_SIM_DIFUSO_SIN_RUIDO,TEMP_SIM_DIFUSO_SIN_RUIDO,'DisplayName','PID DIFUSO');
plot(tiempo_SIM_POSICIONAL_SIN_RUIDO,REF_SIM_POSICIONAL_SIN_RUIDO,'DisplayName','REF PID POSICIONAL');
plot(tiempo_SIM_VELOCIDAD_SIN_RUIDO,REF_SIM_VELOCIDAD_SIN_RUIDO,'DisplayName','REF PID VELOCIDAD');
plot(tiempo_REAL_DIFUSO,REF_SIM_DIFUSO_SIN_RUIDO,'DisplayName','REF PID DIFUSO');
title('RESPUESTA DE LA PLANTA CON PID POSICIONAL, VELOCIDAD, DIFUSO SIMULADO SIN RUIDO');
xlabel('TIEMPO (s)');
ylabel('TEMPERATURA (C)');

figure(2)
plot(tiempo_SIM_POSICIONAL_CON_RUIDO,TEMP_SIM_POSICIONAL_CON_RUIDO,'DisplayName','PID POSICIONAL');
hold on
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,TEMP_SIM_VELOCIDAD_CON_RUIDO,'DisplayName','PID VELOCIDAD');
plot(tiempo_SIM_DIFUSO_CON_RUIDO,TEMP_SIM_DIFUSO_CON_RUIDO,'DisplayName','PID DIFUSO');

```

```

plot(tiempo_SIM_POSICIONAL_CON_RUIDO,REF_SIM_POSICIONAL_CON_RUIDO,'DisplayName','REF PID
POSICIONAL');
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,REF_SIM_VELOCIDAD_CON_RUIDO,'DisplayName','REF PID
VELOCIDAD');
plot(tiempo_REAL_DIFUSO,REF_SIM_DIFUSO_CON_RUIDO,'DisplayName','REF PID DIFUSO');
title('RESPUESTA DE LA PLANTA CON PID POSICIONAL, VELOCIDAD, DIFUSO SIMULADO CON RUIDO');
xlabel('TIEMPO (s)');
ylabel('TEMPERATURA (C)');

figure(3)
plot(tiempo_SIM_POSICIONAL_CON_RUIDO,TEMP_REAL_POSICIONAL,'DisplayName','PID POSICIONAL');
hold on
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,TEMP_REAL_VELOCIDAD,'DisplayName','PID VELOCIDAD');
plot(tiempo_SIM_DIFUSO_CON_RUIDO,TEMP_REAL_DIFUSO,'DisplayName','PID DIFUSO');
plot(tiempo_SIM_POSICIONAL_CON_RUIDO,REF_SIM_POSICIONAL_CON_RUIDO,'DisplayName','REF PID
POSICIONAL');
plot(tiempo_SIM_VELOCIDAD_CON_RUIDO,REF_SIM_VELOCIDAD_CON_RUIDO,'DisplayName','REF PID
VELOCIDAD');
plot(tiempo_REAL_DIFUSO,REF_SIM_DIFUSO_CON_RUIDO,'DisplayName','REF PID DIFUSO');
title('RESPUESTA DE LA PLANTA CON PID POSICIONAL, VELOCIDAD, DIFUSO REAL');
xlabel('TIEMPO (s)');
ylabel('TEMPERATURA (C)');

```