

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

**FACULTAD DE INGENIERÍA ELECTRÓNICA**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL  
TÍTULO EN INGENIERÍA ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UN  
SISTEMA DE CONTROL ADAPTATIVO PARA UNA PLANTA DE  
TEMPERATURA”**

**ANA VERÓNICA GUAMÁN NOVILLO**

**JUAN FERNANDO VÁSQUEZ RODRÍGUEZ**

**SANGOLQUI – ECUADOR**

**2006**

## **CERTIFICACIÓN**

Certificamos que el presente proyecto de grado titulado “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA DE CONTROL ADAPTATIVO PARA UNA PLANTA DE TEMPERATURA” ha sido desarrollado en su totalidad por la señorita ANA VERÓNICA GUAMÁN NOVILLO y el señor JUAN FERNANDO VASQUEZ RODRIGUEZ, bajo nuestra dirección.

Ing. Rodolfo Gordillo.

**DIRECTOR**

Ing. Víctor Proaño.

**CODIRECTOR**

## **AGRADECIMIENTOS**

Después de haber recorrido un largo camino de estudio y esfuerzo me permito rendir mis agradecimientos aquellas personas que me acompañaron y apoyaron durante mi proceso de conocimiento.

A mis padres por que sin ellos no podría estar en el lugar donde estoy.

A mis hermanas y hermano que siempre estuvieron a mi lado dando confianza y alegría en los duros pasos recorridos.

A mis abuelitos que formaron parte de mi primer año de educación.

A mis tías y tíos que supieron ser como unos padres en una ciudad extraña para mí.

A Andrés que siempre estuvo a mi lado dando su cariño y amor en momentos tristes y felices de mi vida.

A todos mis amigos y amigas, ya que la educación no lo es todo ellos complementaron mi crecimiento personal.

A todo el cuerpo administrativo y docente de la FIE por que sin ellos mi conocimiento no se hubiera desarrollado con éxito, en especial al Ing. Rodolfo Gordillo e Ing. Víctor Proaño que nos supieron orientar correctamente para el desarrollo del proyecto de grado.

***ANA VERÓNICA***

## **AGRADECIMIENTO**

A mi madre y a mi padre, es demasiado lo que hay que agradecerles que no puedo ponerlo en palabras.

A mi hermana por su cariño.

A mis tías, tíos, primos y todos mis familiares, en especial a mi tía Laura.

A mis amigos y amigas por todo su apoyo.

A los profesores y personal de la ESPE, en especial al Ing. Rodolfo Gordillo y al Ing. Víctor Proaño por su colaboración

***JUAN FERNANDO***

## DEDICATORIA

Mis padres han sido un ejemplo de superación, son mi modelo a seguir. Todos los logros se los debo a ellos por que la confianza que depositaron en mi no se paga con nada. Por ello esta tesis la dedico a ellos con orgullo por que refleja todo el amor que les tengo y mis triunfos serán también de ellos.

*Ana Verónica*

## **DEDICATORIA**

A todos los que con su esfuerzo, luchan por un mundo mejor.

*Juan Fernando*

## PRÓLOGO

El control de procesos es una parte fundamental del desarrollo industrial actual. Uno de los mayores problemas que tienen los controles tradicionales es que necesitan ser ajustados de acuerdo al proceso que controlan, además si existe una variación en dicho proceso debido a diferentes factores, como por ejemplo el envejecimiento, se pueden producir fallos en el control.

Si se emplea un sistema que pueda identificar el proceso y en base a esta identificación ajustar el controlador, entonces los problemas mencionados se pueden solucionar.

Los sistemas adaptativos autoajustan sus parámetros en base a las condiciones del proceso. Un sistema adaptativo puede realizar este ajuste en línea, es decir sobre la marcha del control realizar cambios en base a las señales de la planta, o también puede calcular los parámetros del control cada vez que se requiera.

El sistema consta de tres partes: la identificación de la planta, el ajuste de parámetros, y el control del proceso. En este proyecto la identificación y el ajuste se lo realizan utilizando un computador. El control se lo realiza utilizando un microcontrolador, este control es capaz de funcionar independientemente del computador.

Este proyecto implementa un control adaptativo para una planta de temperatura, identificando su función de transferencia y sintonizando el controlador automáticamente.

## ÍNDICE

<b>CAPÍTULO 1</b> .....	<b>1</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
1.1. IMPORTANCIA .....	1
1.2. GENERALIDADES DEL CONTROL ADAPTATIVO .....	2
1.2.1. Historia del Control Adaptativo .....	2
1.2.2. Control Robusto.....	3
1.2.3. Control Adaptativo .....	5
1.2.3.1. Diferencias y relaciones entre Control Robusto y Control Adaptativo. ....	6
1.2.3.2. Gain Scheduling .....	6
1.3. MÉTODOS DEL CONTROL ADAPTATIVO .....	7
1.3.1. Método Indirecto .....	7
1.3.2. Método Directo.....	8
1.3.3. Control Adaptativo por Modelo de Referencia (MRAC).....	9
1.3.4. Self Tuning Regulator (STR) .....	11
1.4. ANÁLISIS DEL ESQUEMA DE CONTROL ADAPTATIVO UTILIZADO .....	12
 <b>CAPÍTULO 2</b> .....	 <b>14</b>
<b>IDENTIFICACIÓN DE PARÁMETROS</b> .....	<b>14</b>
2.1. INTRODUCCIÓN .....	14
2.2. ANÁLISIS DEL ALGORITMO DE IDENTIFICACIÓN DE PARÁMETROS ....	14
2.2.1. Identificación de parámetros basada en el error de estimación .....	14
2.2.2. Método de la Gradiente .....	15
2.2.3. Método de los mínimos cuadrados .....	16
2.2.4. Método de los mínimos cuadrados con olvido exponencial.....	18
2.2.5. Simulaciones de los métodos de estimación de parámetros .....	19
2.2.5.1. Parámetro $K$ constante ( $K=10$ ) .....	19
2.2.5.2. Parámetro $K$ variable ( $K=10+5\sin(0.3t)$ ) .....	20
2.3. ALGORITMO DE IDENTIFICACIÓN DE PARÁMETROS DE LA PLANTA...	21

<b>CAPÍTULO 3</b> .....	<b>28</b>
<b>SELECCIÓN DEL MÉTODO DE AJUSTE</b> .....	<b>28</b>
3.1. INTRODUCCIÓN.....	28
3.2. LEY DE CONTROL .....	28
3.2.1. Introducción.....	28
3.2.2. Estructura del PID .....	30
3.3. Mejoras de la ley de control .....	32
3.3.1. Atenuación del rizado.....	32
3.3.2. Atenuación del overshoot.....	32
3.3.2.1. Filtrado del setpoint.....	32
3.3.2.2. Setpoint weighting.....	33
3.4. Criterios de Sintonización .....	33
3.5. AJUSTE DE PARÁMETROS DEL CONTROLADOR .....	34
3.5.1. PID autoajustable basado en un margen de fase prefijado .....	34
3.5.2. PID autoajustable basado en asignación de polos .....	38
3.6. Algoritmo de cálculo de los parámetros de la ley de control .....	41
<b>CAPÍTULO 4</b> .....	<b>43</b>
<b>DISEÑO DEL SOFTWARE</b> .....	<b>43</b>
4.1. INTRODUCCIÓN.....	43
4.2. GENERALIDADES .....	44
4.3. DESARROLLO DEL SOFTWARE EN LABVIEW.....	45
4.3.1. SUBVIs empleados .....	46
4.3.1.1. INDATA.....	46
4.3.1.2. SISTEMA DE CONTROL.....	47
4.3.1.3. Identificación.....	50
4.3.1.4. Calculo ITAE.....	51
4.3.1.5. Funciones convertir grados a voltios/ voltios a grados .....	52
4.3.1.6. OPCIONES AVANZADAS.....	52
4.3.1.7. ERRORES .....	53
4.3.2. Explicación del Diagrama de Bloques General.....	53
4.4. DESARROLLO DEL SOFTWARE PARA EL CONTROLADOR.....	56
4.4.1. Interface con la PC .....	57

4.4.1.1. Protocolo USB .....	57
4.4.1.2. Características de la comunicación.....	58
4.4.1.3. Software de Comunicación.....	58
4.4.1.4. Recepción de datos .....	61
4.4.1.5. Almacenamiento de datos en la memoria EEPROM .....	63
4.4.1.6. Lectura de datos de la memoria EEPROM.....	64
4.4.2. Lectura del sensor .....	64
4.4.2.1. Configuración del módulo A/D .....	64
4.4.2.2. Lectura del valor analógico .....	67
4.4.3. Ley de control.....	68
4.4.3.1. Timer0 .....	69
4.4.3.2. Filtro de Set Point.....	70
4.4.3.3. Actualización de los valores.....	70
4.4.4. Envío de la variable manipulada (transformación digital - análoga).....	71
4.4.5. Diagrama de flujo general .....	73
<b>CAPÍTULO 5 .....</b>	<b>74</b>
<b>SIMULACIÓN DEL SISTEMA .....</b>	<b>74</b>
5.1. INTRODUCCIÓN.....	74
5.2. SIMULACIÓN DEL SISTEMA .....	74
5.2.1. Algoritmo de estimación de parámetros.....	74
5.2.2. Prueba del sistema .....	78
5.2.3. Prueba de la señal de entrada.....	83
5.3. ANÁLISIS DE ESTABILIDAD .....	84
<b>CAPÍTULO 6 .....</b>	<b>86</b>
<b>IMPLEMENTACIÓN DEL HARDWARE .....</b>	<b>86</b>
6.1. INTRODUCCIÓN.....	86
6.2. GENERALIDADES.....	86
6.2.1. Controlador.....	87
6.2.2. Tarjeta de adquisición de datos .....	87
6.2.3. Computadora .....	87
6.3. PARÁMETROS DE DISEÑO .....	88
6.3.1. Controlador.....	88

6.3.1.1. Comunicación con el computador .....	88
6.3.1.2. Lectura de las señales analógicas .....	89
6.3.1.3. Salida de señales analógicas .....	90
6.3.1.4. Características del procesador .....	90
6.3.2. Tarjeta de adquisición de datos .....	91
6.4. ESPECIFICACIÓN DE LOS ELEMENTOS A UTILIZAR .....	91
6.4.1. Planta de Temperatura PCT-2 .....	91
6.4.2. Controlador .....	92
6.4.2.1. Características del procesador .....	92
6.4.2.2. Características del módulo USB .....	93
6.4.2.3. Características de la conversión A/D .....	95
6.4.2.4. Características de la salida analógica .....	95
6.4.3. Tarjeta de adquisición USB PMD-1208LS .....	97
6.4.3.1. Introducción .....	97
6.4.3.2. Detalles de funcionamiento de la PMD-1208LS .....	98
6.4.3.3. Descripción de los pines de la tarjeta .....	98
6.5. DISEÑO DE LA TARJETA .....	100
6.5.1. Generalidades de la tarjeta .....	100
6.5.2. Microcontrolador .....	100
6.5.3. Conexión USB .....	103
6.5.4. Entrada analógica .....	104
6.5.5. Salida analógica .....	105
6.5.6. Fuente de poder .....	105
6.5.7. Amplificadores Operacionales .....	106
6.5.8. Circuito impreso (PCB) .....	107
6.5.8.1. Capacitores de bypass .....	107
6.5.8.2. Planos de tierra .....	107
6.5.8.3. Vías de alimentación .....	109
6.5.8.4. Ubicación de los elementos .....	109
6.5.8.5. Ancho de las pistas empleado .....	110
<b>CAPÍTULO 7 .....</b>	<b>111</b>
<b>PRUEBAS Y RESULTADOS .....</b>	<b>111</b>
7.1. INTRODUCCIÓN .....	111

7.2. PRUEBAS CON LA LEY DE CONTROL TIPO C.....	111
7.2.1. Control Adaptativo .....	111
7.2.1.1. Prueba 1: Control Adaptativo en condiciones normales utilizando la ley de control tipo C .....	113
7.2.1.2. Prueba 2: Control adaptativo con cambios de perturbación utilizando la ley de control Tipo C.....	114
7.2.1.3. Prueba 3: Control adaptativo en condiciones normales y con variación de beta-weight ( $\beta$ ).....	115
7.2.1.4. Prueba 4: Control adaptativo en condiciones normales y ante perturbaciones sin usar ninguna mejora a la ley de control tipo C.....	117
7.2.2. Self-Tuning.....	119
7.2.2.1. Prueba 5: Self-tuning en condiciones normales utilizando la ley de control tipo C. ....	120
7.2.2.2. Prueba 6: Self-Tuning con cambios de perturbaciones utilizando la ley de control tipo C. ....	121
7.2.2.3. Prueba 7: Self-Tuning en condiciones normales y ante perturbaciones sin usar ninguna mejora a la ley de control tipo C.....	123
7.3. PRUEBAS CON LA LEY DE CONTROL TIPO A.....	125
7.3.1. Control Adaptativo .....	126
7.3.1.1. Prueba 8: Control Adaptativo en condiciones normales utilizando la ley de control tipo A .....	126
7.3.1.2. Prueba 9: Control Adaptativo ante perturbaciones utilizando la ley de control tipo A .....	127
7.3.1.3. Prueba 10: Control Adaptativo en condiciones normales y ante perturbaciones utilizando la ley de control tipo A sin mejoras de la ley de control.....	129
7.3.2. Self Tuning .....	131
7.3.2.1. Prueba 11: Self-Tuning en condiciones normales utilizando la ley de control tipo A. ....	132
7.3.2.2. Prueba 12: Self-Tuning ante perturbaciones utilizando la ley de control tipo A.....	133
7.3.2.3. Prueba 13: Self-Tuning en condiciones normales y ante perturbaciones utilizando la ley de control tipo A sin parámetros que mejoran la ley de control .....	135

7.4. RESULTADOS PRELIMINARES OBTENIDOS .....	137
7.4.1. Comparación entre Control Adaptativo Vs Self Tuning .....	137
7.4.2. Comparación entre la ley de control Tipo A y Tipo C .....	138
7.5. PRUEBAS FINALES DEL CONTROL ADAPTATIVO UTILIZANDO EL MICROCONTROLADOR.....	139
7.6. RESULTADOS FINALES.....	141
<b>CAPÍTULO 8.....</b>	<b>142</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>142</b>
8.1. CONCLUSIONES.....	142
8.2. RECOMENDACIONES .....	143

## **REFERENCIAS BIBLIOGRÁFICAS**

### **ANEXOS**

ANEXO 1: DIAGRAMAS DE SIMULACIÓN DE LOS ALGORITMOS DE IDENTIFICACIÓN.

ANEXO 2: SIMULACIÓN DEL SISTEMA

ANEXO 3: CIRCUITO IMPRESO

ANEXO 4: INSTALACIÓN DE SOFTWARE DE LA TARJETA PMD 1208-LS

ANEXO 5: INSTALACIÓN DEL HARDWARE DE LA TARJETA PMD 1208-LS

ANEXO 6: ESPECIFICACIONES TÉCNICAS DE LA TARJETA PMD-1208LS

ANEXO 7: INSTRUMENTOS VIRTUALES BÁSICOS DE LABVIEW PARA LA TARJETA PMD-1208LS.

### **INDICE DE FIGURAS**

### **INDICE DE TABLAS**

### **DATASHEETS**

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1. IMPORTANCIA

Muchos sistemas pueden sufrir un cambio en su funcionamiento normal debido a las variaciones en sus parámetros, los cuales pueden ser producidos por una perturbación externa hacia el mismo y/o cambios propios de los elementos del sistema como por ejemplo envejecimiento de sus componentes, etc.

Las variaciones del sistema no pueden ser controladas por la teoría de control normal debido a que es necesario realizar un ajuste de los parámetros en tiempo real para que el sistema tenga un buen desempeño de acuerdo a las modificaciones que se presenten. Este problema puede ser solucionado con la teoría de control adaptativo de allí la importancia de estudiar esta teoría de control.

El comportamiento de una planta puede ser modificado esencialmente por cambios internos o externos a ella; como por ejemplo, la temperatura de un proceso puede sufrir cambios ya sea por el ambiente (agente externo) o perturbaciones internas como abrir o cerrar un ventilador (agente interno). En la teoría del control normal, se debería realizar un diseño off-line donde se tomarían un número de muestras y mediante alguna ley se encontraría los parámetros para el controlador. Este proceso tomaría un determinado tiempo que en casos de riesgo no se tendría. De igual manera dichos parámetros no servirían para todos los puntos de operación de la planta y/o modificaciones que esta podría sufrir.

La teoría de control adaptativo ayuda a solucionar este problema ya que está formada por la combinación de la estimación de parámetros de la planta con una ley de control que usa la medición de las variables de entrada y salida de la planta.

Por esta razón se estudiarán diferentes esquemas de control adaptativo: directo o indirecto, Modelo de Referencia y Self-Tuning Regulator. De igual manera, se requiere una identificación previa de los parámetros de la planta donde se realizará un análisis para seleccionar el método de identificación dando prioridad al método de identificación RLS,

comparándolo con otros métodos de identificación. Con los parámetros de la planta encontrados se analizará el método de ajuste apropiado que será usado para la ley de control. Finalmente se seleccionará la ley de control utilizando el criterio ITAE y que cumpla los objetivos planteados.

Con la selección tanto del método de identificación de parámetros como de la ley de control, se procederá a realizar la implementación de dicha ley de control en un microcontrolador. Finalmente se obtendrá el diseño e implementación del control adaptativo en tiempo real para controlar la temperatura de un proceso. Los detalles del desarrollo de este control se describirán en los siguientes capítulos.

## **1.2. GENERALIDADES DEL CONTROL ADAPTATIVO**

### **1.2.1. Historia del Control Adaptativo**

El inicio del control adaptativo se remonta a los años 50, cuando se realizó el diseño de autopilotos para aeronaves tomando en cuenta la variación en el rango de velocidades y altitudes, y ambas experiencias con grandes variaciones en sus parámetros. Allí se adoptó un esquema llamado gain scheduling con algunas mediciones auxiliares de la velocidad relativa de vuelo. El Control Adaptativo con Modelo de Referencia fue sugerido por Whitaker para dar solución al problema. Finalmente, Kalman dio el concepto de un controlador self tuning con identificación de los parámetros del proceso y la estimación para actualizar los parámetros de un control cuadrático lineal óptimo.

Un importante desarrollo llegó en 1960 cuando se desarrolló la teoría de control y de manera particular la teoría de control adaptativo. La teoría de estabilidad de Lyapunov fue establecida como una herramienta para evaluar la robustez del control adaptativo. Los progresos en programación dinámica, control compartido y control estocástico formaron parte crucial en la reformulación y diseño del control adaptativo. En 1966, Parks encontró una manera de rediseñar las leyes propuestas en 1950 en los esquemas MRAC lo cual produjo una mejoría en sus controladores.

En 1970 muchas pruebas de estabilidad y robustez fueron implementadas en los esquemas del control adaptativo. El éxito de los años 70 fue seguido por la controversia de la implementación de este control. El inicio de los años 80 presentó problemas con la inestabilidad y robustez de los esquemas de este control debido a frecuencias parásitas, ruidos, retardos, etc. A mediados de los años 80 se realizaron nuevas modificaciones y fueron propuestos y realizados nuevos diseños, llegando así al control adaptativo robusto.

Este trabajo continuó en los 80 originando nuevas modificaciones de robustez unificando un esquema general. Al final de los años 80 se realizaron publicaciones sobre el control adaptativo y su desarrollo. Actualmente, existe una basta literatura con diferentes técnicas de diseño, análisis, desarrollo y aplicaciones.

### 1.2.2. Control Robusto

La teoría de control o la teoría de los sistemas marcan una estrecha relación entre la realimentación y la incertidumbre sobre la conducta del sistema a controlar. Esta incertidumbre es inherente a cualquier sistema físico o a la limitación en comprender o controlar la conducta del sistema. En el sistema existen tres tipos de incertidumbre. La primera existe en el modelo de la planta, perturbaciones en el sistema, y ruido a la entrada de los sensores. Como se muestra en la Figura. 1.1 [12].

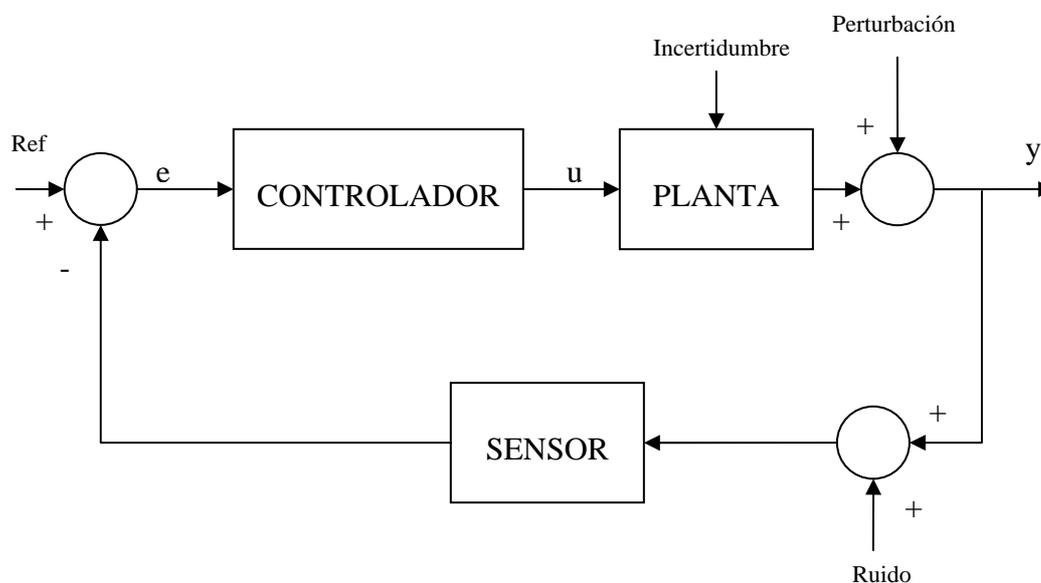


Figura. 1.1. Esquema de Control en Lazo Cerrado con incertidumbres

Chandrasekharan dice que “El Control Robusto se refiere al control de plantas desconocidas con desconocimiento del comportamiento del sistema dinámico así como también de las perturbaciones.[12]” Así el control robusto está diseñado para trabajar incluso con modelo de plantas que no tiene una respuesta 100% confiables. Este control debe garantizar la estabilidad en lazo cerrado y un rendimiento aceptable no solo de la planta nominal sino de plantas similares a esta. En otras palabras quiere decir que para pequeños cambios en el sistema este no produce grandes cambios en la respuesta.

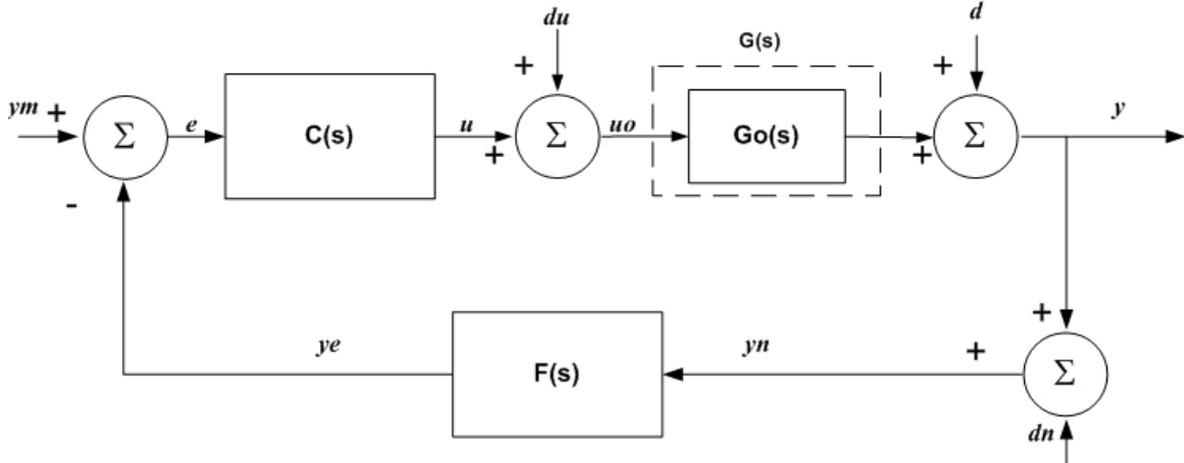


Figura. 1.2. Sistema retroalimentado

Donde:

$y_m$ : es el valor de referencia

$du$ : incertidumbres de la planta.

$d$ : perturbaciones a la salida de la planta

$dn$ : ruido

$e$ : señal de error.

$u$ : señal de salida del controlador.

$u_o$ : variable manipulada.

$y$ : variable de salida de la planta

$C(s)$ : Función de transferencia del controlador.

$G_o(s)$ : Función de transferencia nominal de la planta

$G(s)$ : Familia de plantas a la que pertenece la función de transferencia nominal de  $G_o(s)$ .

$F(s)$ : Función de transferencia del sensor.

El sistema de la Figura. 1.2 [5], está diseñado para estabilizar la parte nominal del modelo de la planta cuya función de transferencia es  $G_o(s)$ . La diferencia entre  $G(s)$  y  $G_o(s)$  son las incertidumbres que se pueden presentar.  $G(s)$  representa una familia de plantas a la que pertenece la función de transferencia nominal de  $G_o(s)$  y la incertidumbre de la planta esta caracterizada por algunos límites en el dominio de la frecuencia. Como indicamos anteriormente decimos que el control  $C(s)$  es robusto con respecto a las incertidumbres de la planta  $G_o(s)$  si éste también se estabiliza para  $G(s)$ . La propiedad de que el control se estabilice en  $G(s)$  se llama como estabilidad robusta.

Las ventajas principales de tener un control robusto es que para cualquier planta de una misma familia, este mantendrá un buen control a pesar de trabajar en distintos puntos de operación, logrando así un proceso robusto.

### 1.2.3. Control Adaptativo

De acuerdo al diccionario Webster, adaptar significa “cambiar el comportamiento dadas ciertas circunstancias”. De esta manera podemos deducir que Control Adaptativo es un control que cambia sus parámetros constantemente debido a cambios ya sea por un sistema dinámico o por perturbaciones. Es una técnica que aplica algunas leyes de identificación para obtener un modelo del proceso y de su ambiente, finalmente usar este modelo para diseñar un controlador adecuado para el sistema.

El control adaptativo está formado por: una estimación en línea de los parámetros de la planta obtenida al procesar las señales de entrada  $u$  y salida  $y$  (Figura. 1.3), un cálculo de los parámetros del controlador ( $\theta(t)$ ) en base a los parámetros estimados de la planta, y una ley de control cuyos parámetros son  $\theta(t)$ .

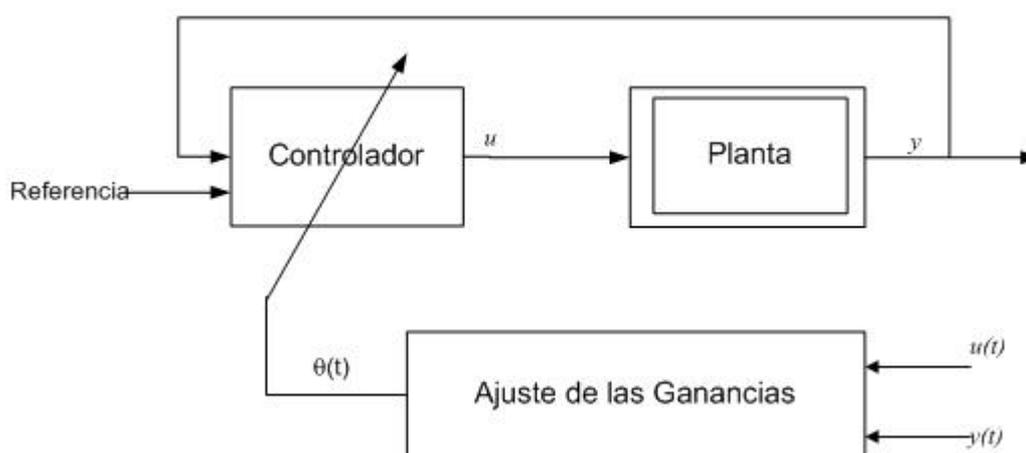


Figura. 1.3. Diagrama de un controlador con ganancias ajustables controladas.[5]

El control adaptativo varía del control normal en que ajusta parámetros que son variables, usualmente es usado para manejar parámetros desconocidos que varían en el tiempo. La variación en el tiempo de la planta tiene que ser considerablemente lenta con respecto a la adaptación de los parámetros, se debe tomar en cuenta que el cambio muy rápido de los parámetros indica que el modelo es inapropiado.

### 1.2.3.1. Diferencias y relaciones entre Control Robusto y Control Adaptativo.

Control Adaptativo	Control Robusto
Mejora el rendimiento con una adaptación en línea.	Intenta mantener un constante rendimiento.
Necesita poca información a priori acerca de los parámetros desconocidos.	Usualmente necesita una información a priori razonable de los límites de los parámetros
Para que el sistema se adapte nuevamente ante cambios como perturbaciones, se demora un tiempo considerable.	Mejor rendimiento ante perturbaciones y variación rápida de los parámetros.

Tabla 1.1. Comparación entre Control Adaptativo y Control Robusto [11]

Actualmente se han combinado ambas características llegando a formar lo que se llama Control Adaptativo Robusto en el cual las incertidumbres o las variaciones de los parámetros es reducido por la adaptación de los parámetros y otras fuentes de incertidumbres serán reducidas por las técnicas de robustez. También existen técnicas para sistemas no lineales que generalmente requiere una parametrización lineal de la planta.

### 1.2.3.2. Gain Scheduling

Gain Scheduling es una técnica que trata con procesos no lineales, procesos que sufren variaciones en el tiempo, o cuando los requerimientos en el control cambian con las condiciones de operación. La idea es encontrar variables auxiliares que correlacionen bien los cambios del sistema dinámico. Esto es posible compensando las variaciones de los parámetros de la planta cambiando los parámetros del controlador en función de las variables auxiliares. Dicho comportamiento se observa en la Figura. 1.4 [5].

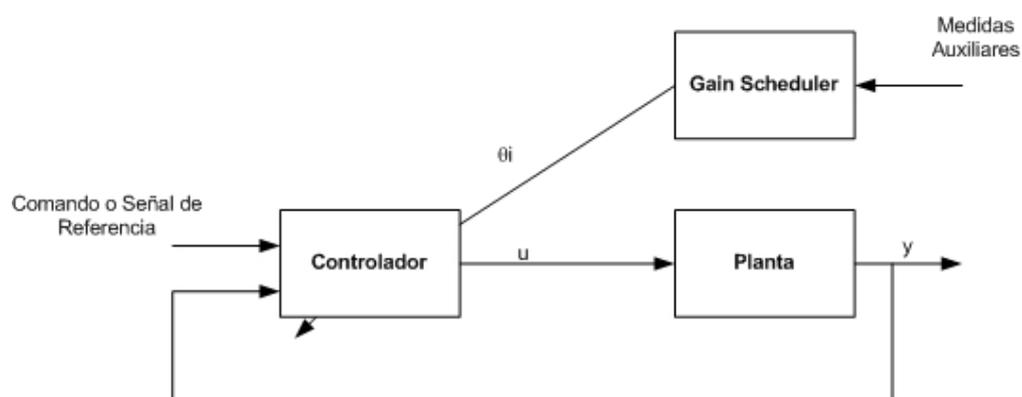


Figura. 1.4. Gain Scheduling

Gain Scheduling consiste en buscar en una tabla y con la lógica apropiada la ganancia que detecte el punto de operación óptimo. La variable programada (scheduling variable) puede ser una señal de medida, una señal de control, o una señal externa.

Las ventajas de Gain Scheduling es que las ganancias del controlador pueden ser cambiadas tan pronto como las medidas auxiliares respondan a los cambios de los parámetros. Esta velocidad también puede originar inestabilidad, por ello se debe tomar en cuenta cuan seguido y a que velocidad el control de las ganancias pueden ser cambiadas.

La desventaja es que se necesita realizar un pre-computo off-line de un mecanismo de adaptación de las ganancias de control, también se requiere un esfuerzo enorme para la implementación aunque este esfuerzo puede ser reducido con <automatic tuning> ya que las ganancias programadas pueden ser determinadas experimentalmente.

A pesar de todo gain scheduling es un método popular para manejar la variación de los parámetros en control de vuelo y otros sistemas.

### 1.3. MÉTODOS DEL CONTROL ADAPTATIVO

#### 1.3.1. Método Indirecto

Un diagrama general de este método se muestra en la Figura. 1.5. En el método indirecto el modelo de la planta  $P(\theta^*)$  está parametrizado en función de un vector de parámetros desconocidos  $\theta^*$ . Procesando la entrada  $u$  y la salida  $y$  de la planta se puede estimar los parámetros  $\theta^*$ , con lo que se genera un vector de parámetros estimados  $\hat{\theta}(t)$  para cada tiempo  $t$ . En base a  $\hat{\theta}(t)$  se tiene un modelo estimado de la planta representado por  $\hat{P}(\hat{\theta}(t))$ . Utilizando los parámetros estimados  $\hat{\theta}(t)$  se calculan los parámetros del controlador mediante una función  $F$ , con lo que:

$$\theta_c(t) = F(\hat{\theta}(t))$$

Donde  $\theta_c(t)$  son los parámetros del controlador en cada tiempo  $t$ .

Para el método indirecto se necesita escoger una la ley de control  $C(\theta_c)$ , un estimador de parámetros para encontrar  $\hat{\theta}(t)$ , y una función para generar los parámetros del controlador  $\theta_c(t)$  en base a los parámetros estimados  $\hat{\theta}(t)$ .

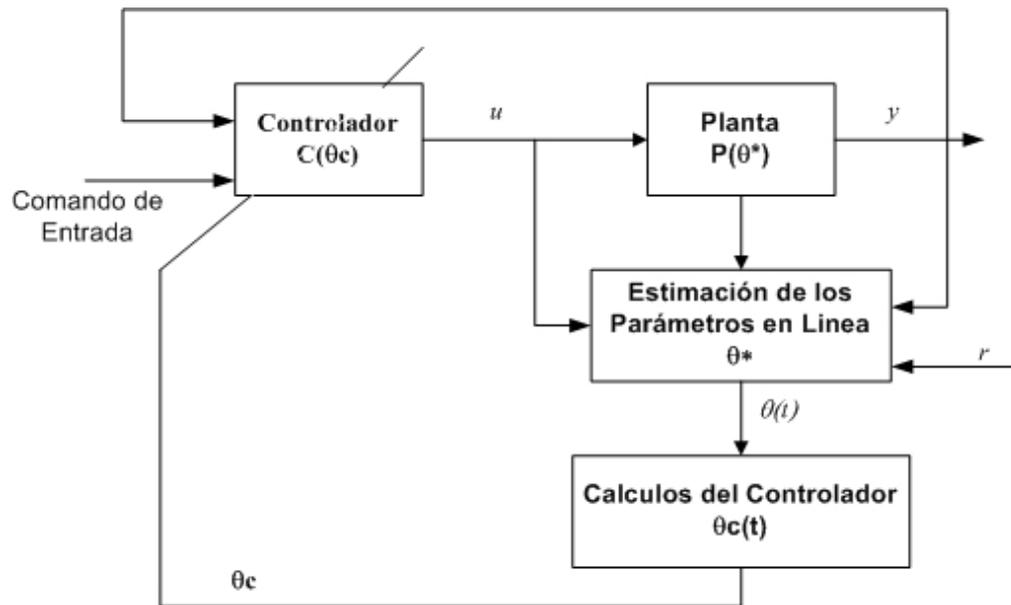


Figura. 1.5. Control Adaptativo Indirecto [5]

Donde:

$\theta^*$ : Parámetros desconocidos de la planta

$P(\theta^*)$ : Modelo de la planta parametrizada en función de  $\theta^*$ .

$\theta(t)$ : Vector de parámetros estimados en el tiempo  $t$ .

$\theta_c(t)$ : Parámetros del controlador en el tiempo  $t$ .

$C(\theta_c)$ : Ley de control con parámetros  $\theta_c$ .

### 1.3.2. Método Directo

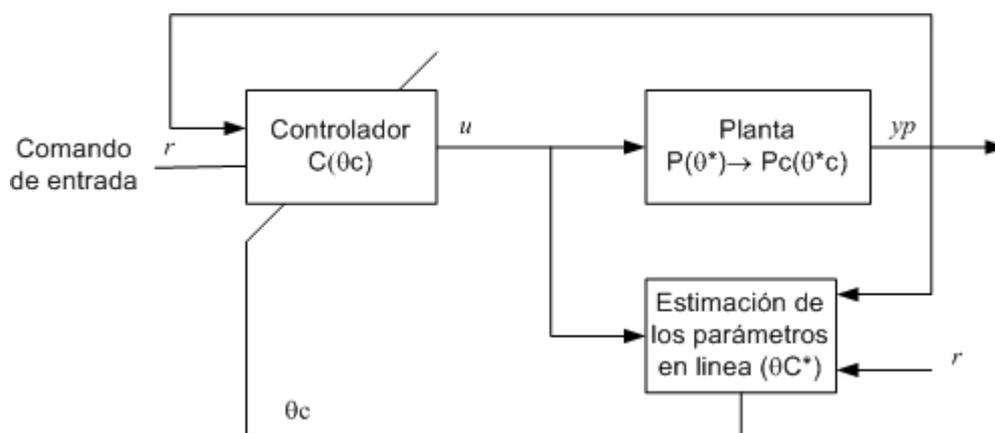


Figura. 1.6. Control adaptativo directo [5]

Donde:

$\theta^*$ : Parámetros de la planta desconocidos.

$\theta^*c$ : Parámetros desconocidos del controlador.

$P(\theta^*)$ : Modelo de la planta parametrizada en función de  $\theta^*$ .

$Pc(\theta^*c)$ : Modelo de la planta parametrizada en función de  $\theta^*c$ .

$\theta c(t)$ : Parámetros del controlador

$C(\theta c(t))$ : Ley de control con parámetros  $\theta c$ .

Un esquema de este método se muestra en la Figura. 1.6. El modelo de la planta  $P(\theta^*)$  se parametriza en función de  $\theta^*c$ , siendo  $\theta^*c$  los parámetros desconocidos del controlador, con lo que se tiene el modelo  $Pc(\theta^*c)$  que es el modelo de la planta en función de  $\theta^*c$ . El estimador se diseña en base al modelo  $Pc(\theta^*c)$ , con lo que procesando la entrada  $u$  y la salida  $y$  de la planta se puede generar un vector  $\theta c(t)$  que es la estimación de  $\theta^*c$  para cada tiempo  $t$ . Entonces  $\theta c(t)$  se usa para actualizar directamente los parámetros del controlador.

Para el método directo se necesita escoger una ley de control  $C(\theta c)$  y un método de estimación para generar  $\theta c(t)$ .

Una limitación de éste método es que no todos los modelos de plantas pueden ser parametrizados en función de los parámetros del controlador.

### 1.3.3. Control Adaptativo por Modelo de Referencia (MRAC)

En un control por modelo de referencia se busca que el lazo de control tenga el mismo comportamiento que el modelo de referencia. En la Figura. 1.7 se puede ver el esquema de este tipo de control sin aplicar un mecanismo adaptativo.

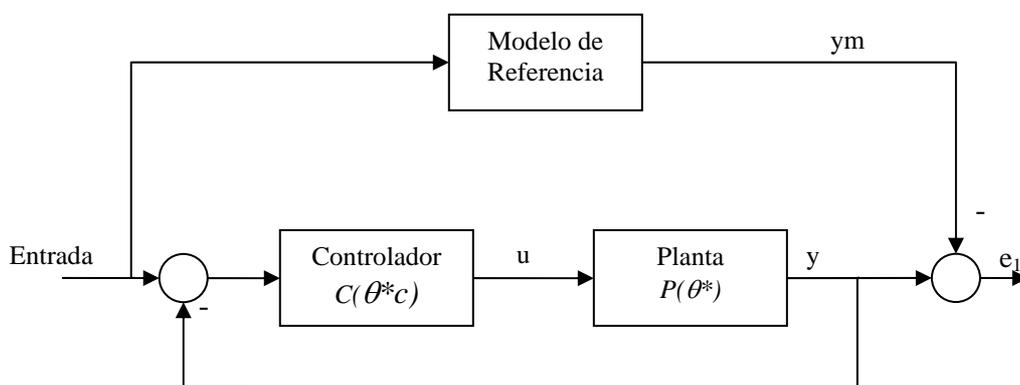


Figura. 1.7. Control por modelo de referencia[5]

Donde:

$ym$ : Respuesta del modelo de referencia.

$y$ : Respuesta del lazo de control.

$e_1$ : Diferencia entre  $y$  e  $ym$ . ( $e_1 = y - ym$ ).

$P(\theta^*)$ : Modelo de la planta en función de  $(\theta^*)$ .

$C(\theta^*_c)$ : Ley de control con parámetros  $\theta^*_c$ .

La respuesta del modelo de referencia a una entrada representa la respuesta ideal que la planta debería tener. Por lo tanto el controlador se debe diseñar de tal manera que la respuesta del lazo de control sea igual que la del modelo de referencia. Si se conocen los parámetros de la planta, entonces se puede hallar una función  $F$ , tal que:

$$\theta^*_c = F(\theta^*)$$

Donde  $\theta_c$  son los parámetros del controlador y  $\theta^*$  son los parámetros de la planta. La función  $F$  se diseña de tal manera que el comportamiento del lazo de control sea igual que el del modelo de referencia.

Cuando no se conocen los parámetros de la planta, se puede aplicar un mecanismo de estimación de dichos parámetros, con lo que se tiene un control adaptativo por modelo de referencia.

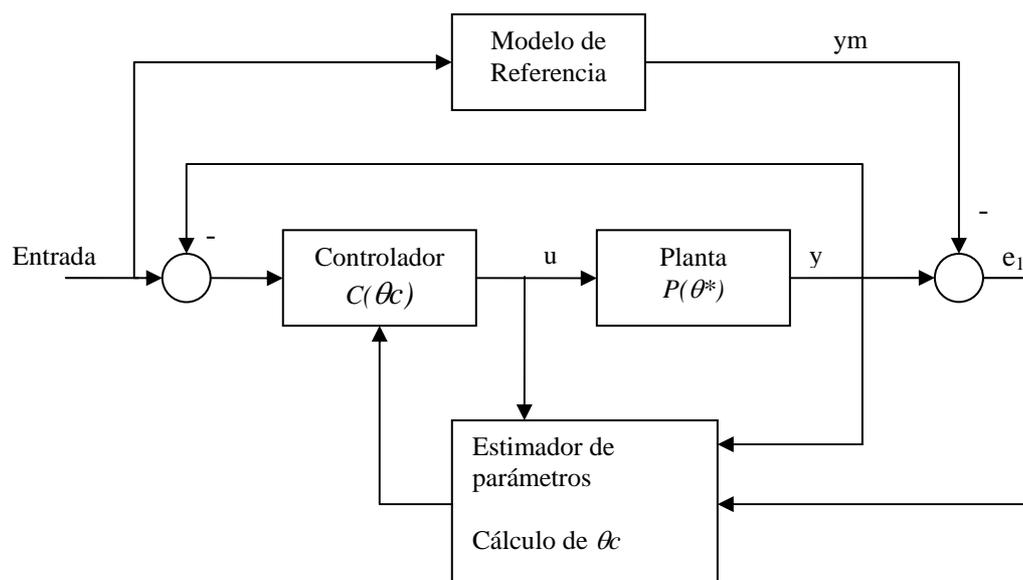


Figura. 1.8. Control adaptativo por modelo de referencia[5]

Donde:

$y_m$ : Respuesta del modelo de referencia.

$y$ : Respuesta del lazo de control.

$e_1$ : Diferencia entre  $y$  e  $y_m$ . ( $e_1 = y - y_m$ ).

$P(\theta^*)$ : Modelo de la planta en función de  $(\theta^*)$ .

$\theta_c$ : Parámetros del controlador

$C(\theta_c)$ : Ley de control con parámetros  $\theta_c$ .

La estimación de los parámetros desconocidos puede hacerse utilizando tanto el método directo como el indirecto. Además los parámetros del controlador se calculan de tal manera que el error  $e_1$ , que es la diferencia entre la salida de la planta y la salida del modelo de referencia, tienda a cero. (Figura. 1.8)

### 1.3.4. Self Tuning Regulator (STR)

Esta técnica de control adaptativo fue creada como un método de diseñar un control para plantas desconocidas. Este diseño es la combinación de una estructura de controlador, y la relación entre los parámetros de la planta y los parámetros del control. Debido a que los parámetros de la planta son desconocidos estos deben ser estimados usando un algoritmo recursivo de identificación. Luego estos parámetros son usados para obtener los parámetros del controlador, como *si estos fueran los parámetros verdaderos*. Esta idea se conoce como el principio de *equivalencia con certeza*.

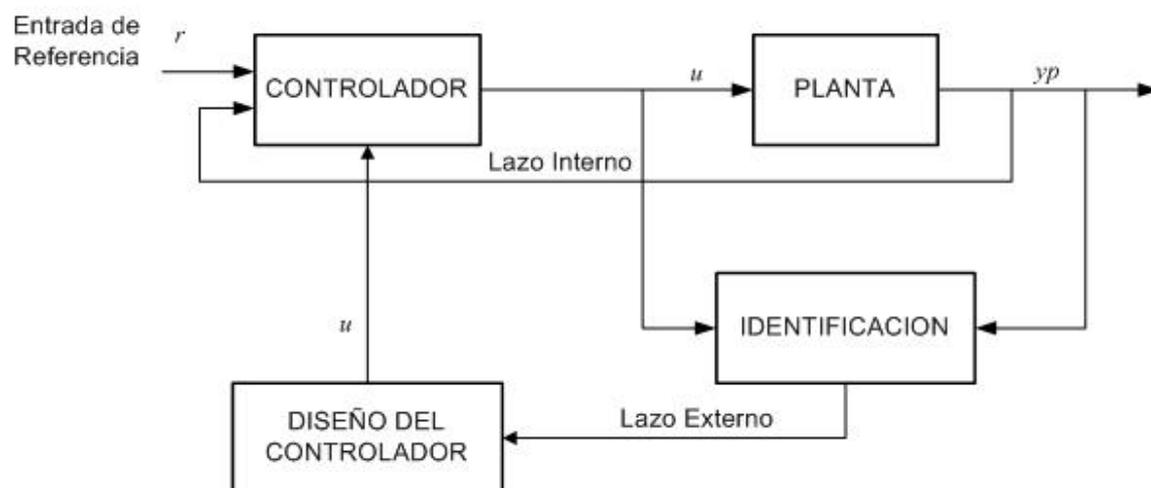


Figura. 1.9. Esquema Self Tuning Regulator [10]

La estimación de los parámetros puede ser entendida como el proceso de encontrar un conjunto de parámetros que se ajusten a las entradas y salidas disponibles de la planta. El método más popular es el *método de mínimos cuadrados* y sus extensiones.

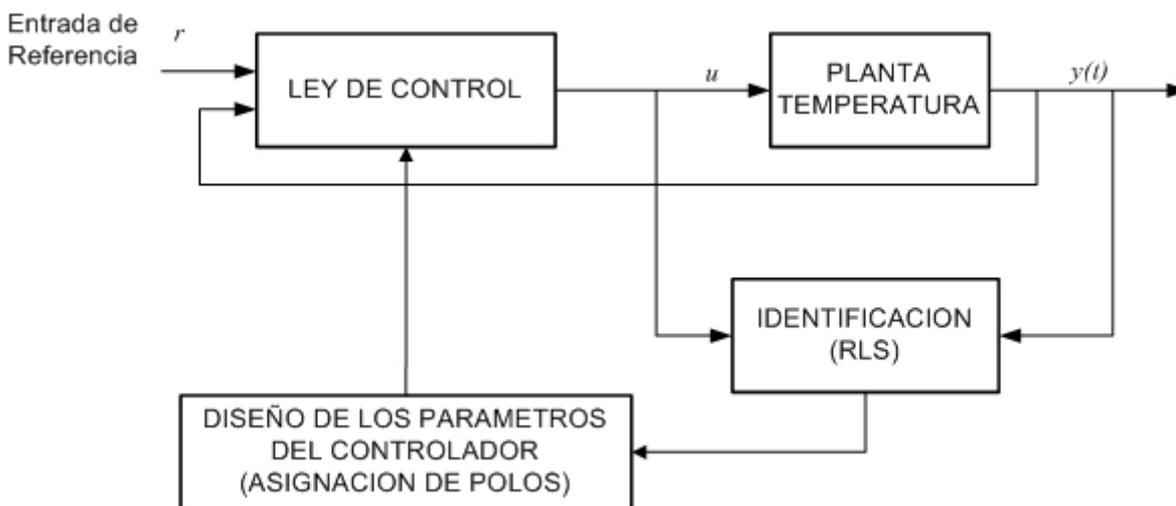
Se asume una separación entre la identificación y el control, este método es llamado self tuning debido a la capacidad de sintonizar sus propios parámetros.

Este esquema tiene dos lazos como se puede observar en la Figura 1.9. El lazo interno consiste en un control convencional, pero con variación de sus parámetros, y el lazo externo consiste en la identificación y un bloque de diseño (representa una solución en línea para un problema de diseño pero con parámetros conocidos) el cual ajustará los parámetros del controlador.

Este método es flexible pues permite escoger el diseño del controlador y el esquema de identificación. Sin embargo, el análisis del sistema del self tuning regulator es complejo debido a la transformación de los parámetros de identificación de la planta a los parámetros del controlador.

#### 1.4. ANÁLISIS DEL ESQUEMA DE CONTROL ADAPTATIVO UTILIZADO

En la Figura. 1.10 se muestra el esquema ha utilizar en este proyecto de grado.



**Figura. 1.10. Esquema de control adaptativo usado**

Como se observa en la figura anterior el esquema a utilizar es el de self tuning regulator debido a que no tiene un modelo de planta de referencia y con este esquema se puede calcular en todo momento los parámetros de planta desconocida. Para ello es necesario usar los valores de entrada y salida de la planta.

Con este esquema se tiene la certeza que a pesar de que un agente externo influya en el rendimiento del proceso, el control se ajustará nuevamente buscando los nuevos parámetros que logre que el controlador estabilice el proceso. Y así se logrará que la planta pueda trabajar en distintos puntos de operación.

## CAPÍTULO 2

### IDENTIFICACIÓN DE PARÁMETROS

#### 2.1. INTRODUCCIÓN

Para implementar un controlador STR (Self-Tuning Regulator) se necesita encontrar los parámetros de la planta que está siendo controlada, para lo cual se emplea un estimador de parámetros. Una alternativa puede ser utilizar un estimador basado en el método de la gradiente. Otra alternativa, quizá una de las de las formas más conocidas de estimación de parámetros, es la utilización del estimador de mínimos cuadrados. El estudio que existe sobre este tipo de estimador es extenso, y las variaciones y formas de aplicarlo también son amplias.

Es así que en el presente capítulo se revisan diferentes métodos de estimación de parámetros, se presentan unas simulaciones que permiten observar el comportamiento de los mismos y además se expone el algoritmo de identificación usado en el proyecto.

#### 2.2. ANÁLISIS DEL ALGORITMO DE IDENTIFICACIÓN DE PARÁMETROS

Para este análisis primero se definirá un modelo del sistema a identificar, una de las formas más comunes utilizada en la bibliografía especializada es:

$$y(t) = z(t)a(t) \tag{2.1}$$

Donde  $y(t)$  es la salida de la planta,  $z(t)$  es el vector de señales y  $a(t)$  son los parámetros desconocidos de la planta que se quieren estimar [11]. A este modelo se le conoce como el *modelo de parametrización lineal* o como el *modelo de regresión lineal*.

##### 2.2.1. Identificación de parámetros basada en el error de estimación

Una forma para la identificación de los parámetros desconocidos en el esquema de control adaptativo STR es utilizando el error de estimación ( $e_1$ ), que es la diferencia entre la salida real del sistema y la salida estimada.

$$e_1(t) = y(t) - \hat{y}(t) \tag{2.2}$$

Donde  $y(t)$  es la salida real e  $\hat{y}(t)$  es la salida estimada que en base a (2.1) viene dada por:

$$\hat{y}(t) = z(t)\hat{a}(t) \quad (2.3)$$

Donde  $z(t)$  es un vector que contiene las entradas del sistema y  $\hat{a}(t)$  es un vector que contiene los parámetros estimados. Entonces, para la identificación de los parámetros del sistema se debe buscar un valor  $\hat{a}(t)$  tal que el error de estimación sea lo más pequeño posible.

En base a (2.1), (2.2) y a (2.3), se tiene:

$$e_1(t) = z(t)[\hat{a}(t) - a(t)] \quad (2.4)$$

En (2.4) se evidencia que si  $e_1(t) \rightarrow 0$ , quiere decir que  $\hat{a}(t) \rightarrow a(t)$ , por lo tanto se debe emplear un método para minimizar  $e_1(t)$  de tal modo que este converja hacia cero. Para lograr esto existen varios métodos, se revisa a continuación el de la gradiente y el de los mínimos cuadrados.

### 2.2.2. Método de la Gradiente

Es uno de los métodos más simples. Primero, si se define la función que se debe minimizar como:

$$\begin{aligned} J(\hat{a}(t)) &= \frac{1}{2} e_1^2 \\ J(\hat{a}(t)) &= \frac{1}{2} [y(t) - z(t)\hat{a}(t)]^2 \end{aligned} \quad (2.5)$$

Entonces se tiene que la gradiente de dicha función es:

$$\nabla J(\hat{a}(t)) = -z^T(t)[y(t) - z(t)\hat{a}(t)]$$

El método de la gradiente se basa en la siguiente ecuación [5]:

$$\dot{\hat{a}} = -P\nabla J(\hat{a}(t))$$

Donde P es una matriz simétrica que es conocida como *matriz de ganancia*, finalmente se tiene:

$$\dot{\hat{a}} = -P \left[ -z^T(t) [y(t) - z(t)\hat{a}(t)] \right]$$

Con lo que se llega al estimador:

$$\dot{\hat{a}} = Pz^T(t)e_1(t)$$

Según [11] y [5] con el empleo de este estimador, dadas las condiciones adecuadas de las señales  $z(t)$ ,  $\hat{a}$  converge hacia  $a$ ; pero por otra parte si existe demasiado ruido en el sistema el estimador puede volverse inestable.

### 2.2.3. Método de los mínimos cuadrados

Este método es uno de los más conocidos y difundidos en la identificación de parámetros, sobre todo cuando se trata de estimaciones en tiempo discreto.

A diferencia del método de la gradiente que emplea una función de costo  $J$  que toma en cuenta solo el error instantáneo, este procedimiento toma en cuenta todos los errores del pasado para tratar de minimizar la función de costo  $J$ . Este método tiene un gran desenvolvimiento en presencia de ruido, debido a que el mismo afecta muy poco a la estimación. Esta es una de las razones del uso extendido de los mínimos cuadrados.

La función de costo  $J(\hat{a}(t))$  se define como:

$$J(\hat{a}(t)) = \int_0^t e_1^2(t, \nu) d\nu$$

Donde el error de estimación  $e_1(t, \nu) = y(\nu) - z(\nu)\hat{a}(t)$ , con lo que se tiene:

$$J(\hat{a}(t)) = \int_0^t [y(\nu) - z(\nu)\hat{a}(t)]^2 d\nu$$

La minimización de  $J(\hat{a}(t))$  de tal manera que  $\hat{a}(t)$  converja hacia  $a(t)$  se logra igualando a cero la gradiente de la función  $J(\hat{a}(t))$ , así:

$$\begin{aligned} \nabla J(\hat{a}(t)) &= 0 \\ -2 \int_0^t z^T(\nu) [y(\nu) - z(\nu)\hat{a}(t)] d\nu &= 0 \\ \int_0^t z^T(\nu) y(\nu) d\nu - \int_0^t z^T(\nu) z(\nu) \hat{a}(t) d\nu &= 0 \end{aligned}$$

Con lo que se obtiene:

$$\left[ \int_0^t z^T(\nu)z(\nu)d\nu \right] \hat{a}(t) = \int_0^t z^T(\nu)y(\nu)d\nu \quad (2.6)$$

Se define la matriz  $C(t)$  como:

$$C(t) = \left[ \int_0^t z^T(\nu)z(\nu)d\nu \right]^{-1} \quad (2.7)$$

Derivando (2.6) se tiene:

$$\begin{aligned} \frac{d}{dt} \left\{ \left[ \int_0^t z^T(\nu)z(\nu)d\nu \right] \hat{a}(t) \right\} &= \frac{d}{dt} \left\{ \int_0^t z^T(\nu)y(\nu)d\nu \right\} \\ \frac{d}{dt} \left[ \int_0^t z^T(\nu)z(\nu)d\nu \right] \hat{a}(t) + \left[ \int_0^t z^T(\nu)z(\nu)d\nu \right] \frac{d}{dt} \hat{a}(t) &= \frac{d}{dt} \left\{ \int_0^t z^T(\nu)y(\nu)d\nu \right\} \\ z^T(t)z(t)\hat{a}(t) + \left[ \int_0^t z^T(\nu)z(\nu)d\nu \right] \dot{\hat{a}}(t) &= z^T(t)y(t) \end{aligned}$$

Si se utiliza (2.7) se obtiene:

$$\begin{aligned} C^{-1}(t)\dot{\hat{a}}(t) &= z^T(t)y(t) - z^T(t)z(t)\hat{a}(t) \\ C^{-1}(t)\dot{\hat{a}}(t) &= z^T(t)[y(t) - z(t)\hat{a}(t)] \end{aligned}$$

Con lo que finalmente se llega a:

$$\dot{\hat{a}}(t) = C(t)z^T(t)e_1(t) \quad (2.8)$$

Partiendo de la ecuación (2.7), se tiene:

$$\begin{aligned} C^{-1}(t) &= \int_0^t z^T(\nu)z(\nu)d\nu \\ \dot{C}^{-1}(t) &= z^T(t)z(t) \end{aligned}$$

Para la actualización de  $C(t)$ , se puede emplear el procedimiento descrito en [11], el cual mediante el uso de ciertas identidades matriciales a partir de la expresión anterior llega a:

$$\dot{C}(t) = -C(t)z^T(t)z(t)C(t) \quad (2.9)$$

Como se dijo anteriormente, la gran ventaja de este método es su buena inmunidad al ruido. Además su discretización no es complicada y no representa grandes costos computacionales. El mayor problema de este método es que cuando los parámetros reales del sistema varían, pierde efectividad debido a que el método toma en cuenta todos los

errores de estimación pasados. Este inconveniente puede ser resuelto introduciendo un *coeficiente de olvido*, como se explica en el siguiente punto.

#### 2.2.4. Método de los mínimos cuadrados con olvido exponencial

Como se dijo en el punto anterior, uno de los mayores problemas con el método de los mínimos cuadrados, como el presentado en el punto anterior, es que el mismo no es bueno cuando existe una variación de los parámetros reales del sistema. Esto se debe a que el estimador basado en mínimos cuadrados procura estimar los parámetros tomando en cuenta todos los errores de estimación anteriores. Para lograr un mejor desenvolvimiento de este método en presencia de variaciones de parámetros, se puede introducir un coeficiente que anule los errores de estimación más *viejos*. Es decir, que el estimador *olvida* exponencialmente los errores de estimación anteriores.

Para esto se define nuevamente la función de costo  $J(\hat{a}(t))$  como sigue:

$$J(\hat{a}(t)) = \int_0^t e^{-\beta(t-\nu)} [y(\nu) - z(\nu)\hat{a}(t)]^2 d\nu$$

Donde  $\beta$  es el coeficiente de olvido, el mismo que  $\beta \geq 0$ . Si  $\beta = 0$ , se tiene una estimación sin coeficiente de olvido, (todos los errores de estimación tienen el mismo peso).

Siguiendo un procedimiento similar al que se utilizó para llegar a (2.8), se obtiene:

$$\dot{\hat{a}}(t) = C(t)z^T(t)e_1(t) \quad (2.10)$$

Esta vez la ganancia  $C(t)$  se define como:

$$C(t) = \left[ \int_0^t e^{-\beta(t-\nu)} z^T(\nu)z(\nu)d\nu \right]^{-1}$$

Derivando  $C^{-1}(t)$  en función del tiempo se llega a:

$$\begin{aligned} \dot{C}^{-1}(t) &= -\beta \left[ \int_0^t e^{-\beta(t-\nu)} z^T(\nu)z(\nu)d\nu \right] + z^T(t)z(t) \\ \dot{C}^{-1}(t) &= -\beta C^{-1}(t) + z^T(t)z(t) \end{aligned} \quad (2.11)$$

Empleando las mismas identidades que se usaron para llegar a (2.9), se tiene:

$$\dot{C}(t) = \beta C(t) - C(t)z(t)^T z(t)C(t)$$

Con este método se soluciona el problema que se da cuando se estiman parámetros que varían en el tiempo. De todos modos se debe tomar en cuenta que dicha variación no puede ser más rápida que la velocidad de estimación. Un problema que se tiene con este método como se describe en [11] y [5] es que debido al factor  $\beta$  la matriz  $C(t)$  puede crecer desmedidamente, lo que lleva a que los parámetros estimados no converjan hacia los parámetros reales.

### 2.2.5. Simulaciones de los métodos de estimación de parámetros

A continuación se presentan simulaciones de la estimación de parámetros utilizando las técnicas revisadas en los puntos anteriores. Por simplicidad, se va a utilizar el siguiente sistema escalar:

$$y = uK$$

Donde la ganancia  $K$  se supone desconocida y es la que se quiere estimar, La entrada del sistema es  $u$  y su salida es  $y$ . Se presentan simulaciones con  $K$  fijo en la que  $K$  es igual a 10, y con  $K$  variable en la que  $K$  es igual a  $10 + 5 \sin(0.3t)$ , tanto en ausencia de ruido (a la izquierda) y en presencia de ruido (a la derecha). La simulación fue realizada utilizando Simulink, los esquemas se pueden encontrar en el Anexo 1, el diagrama de bloques de la simulación se presenta en la Figura. 2.1

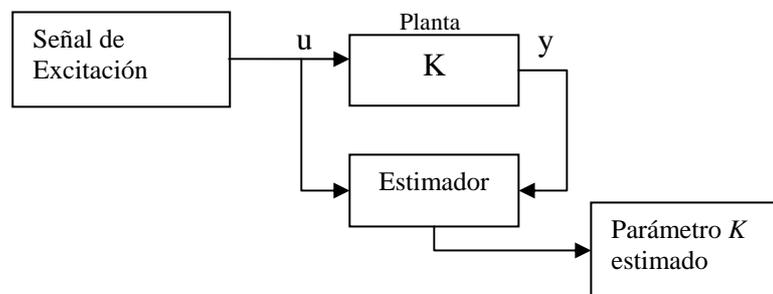


Figura. 2.1. Diagrama de simulación

#### 2.2.5.1. Parámetro $K$ constante ( $K=10$ )

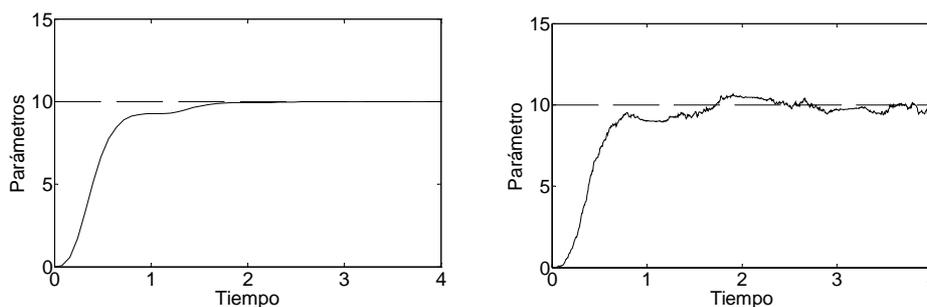
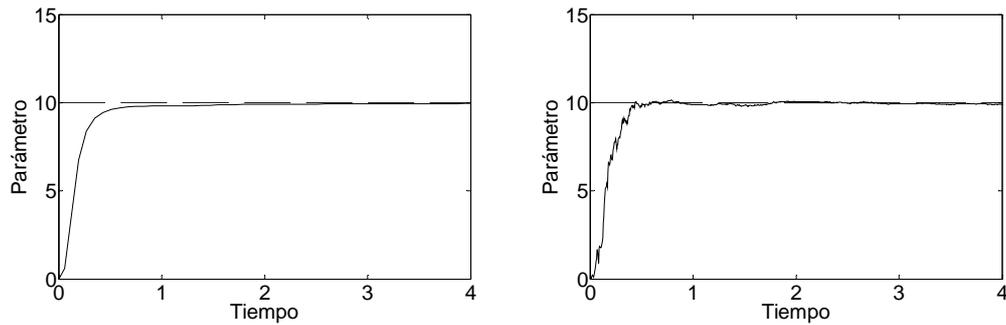
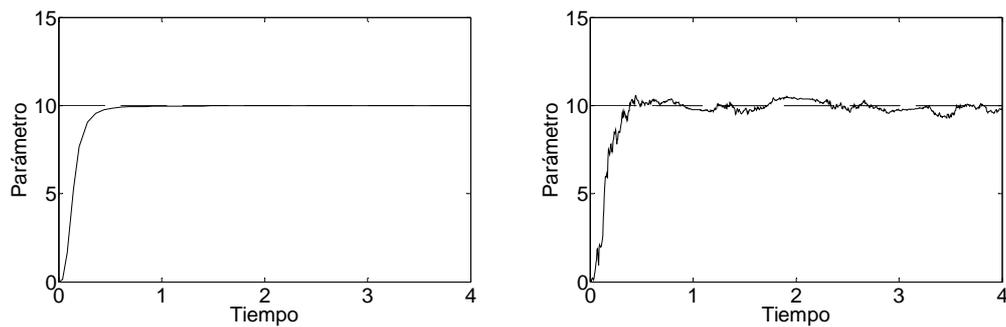


Figura. 2.2. Método de la gradiente



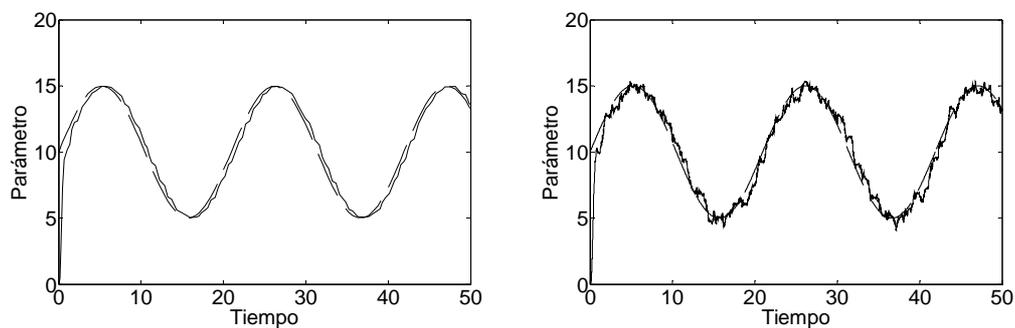
**Figura. 2.3. Método de los Mínimos Cuadrados**



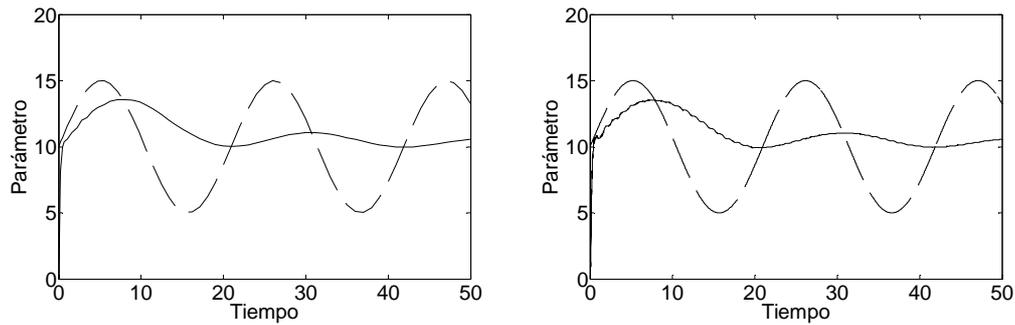
**Figura. 2.4. Método de los Mínimos Cuadrados con olvido exponencial**

El parámetro real se representa con línea entrecortada (es el valor constante en 10), mientras que el parámetro estimado se representa con línea continua. Como se puede ver en las gráficas, en ausencia de ruido los tres métodos logran hacer que el parámetro  $K$  estimado converja satisfactoriamente hacia el valor real de  $K$ . Con la presencia de ruido, se puede ver que la mejor estimación es la de los mínimos cuadrados normal, que presenta una gran inmunidad a dicha perturbación.

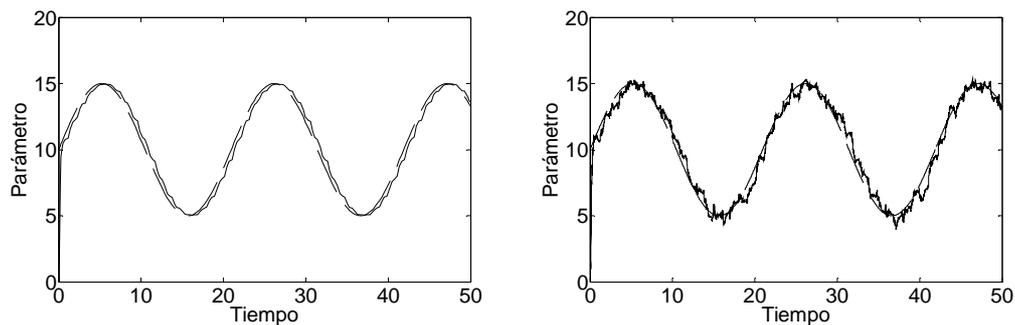
### 2.2.5.2. Parámetro $K$ variable ( $K=10+5\sin(0.3t)$ )



**Figura. 2.5. Método de la gradiente**



**Figura. 2.6. Método de los Mínimos Cuadrados**



**Figura. 2.7. Método de los Mínimos Cuadrados con olvido exponencial**

El parámetro real se representa con línea entrecortada, mientras que el parámetro estimado se representa con línea continua. Se puede ver claramente la incapacidad que tiene el método de los mínimos cuadrados normal para seguir variaciones en el valor real de los parámetros estimados, mientras que el método de la gradiente y el de los mínimos cuadrados con olvido exponencial pueden resolver con éxito el problema de los parámetros variables.

### 2.3. ALGORITMO DE IDENTIFICACIÓN DE PARÁMETROS DE LA PLANTA

Como se concluyó anteriormente, el método de mínimos cuadrados con olvido exponencial representa una buena solución para la estimación de parámetros en línea. A continuación se desarrollará un algoritmo recursivo en tiempo discreto para la estimación de parámetros usando el mencionado método. El desarrollo que aquí se presenta fue tomado de [1]. El algoritmo tomado de dicha fuente es llamado REFIL, y es el algoritmo implementado en este proyecto para la identificación de parámetros

La forma de la función de transferencia discreta de orden  $n$  que se va a estimar es la siguiente:

$$G(z^{-1}) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}}{1 - a_1 z^{-1} - a_2 z^{-2} - \dots - a_n z^{-n}}$$

Se define la función de costo  $J(\hat{a}(t))$  en el tiempo discreto como:

$$J(\hat{a}(t)) = \frac{1}{t} \sum_{k=1}^t \varphi^{2(t-k)} e_1(k)^2$$

Siendo  $e_1(k)$  el error de estimación,  $\hat{a}(t)$  es el vector de parámetros estimados en el tiempo  $t$ , y que tiene la forma:

$$a(t)^T = [b_0 \ a_1 \ b_1 \ a_2 \ b_2 \ \dots \ a_n \ b_n]$$

Mientras que  $\varphi$  es el coeficiente de olvido el cual debe cumplir  $0 < \varphi \leq 1$ . Un coeficiente de  $\varphi$  igual a 1 significa una estimación sin olvido exponencial.  $J(\hat{a}(t))$  puede ser representado en la siguiente forma:

$$J(\hat{a}(t)) = \frac{1}{t} E_1(t)^T W(t) E_1(t)$$

$E(t)$  es un vector que contiene los errores de estimación desde  $k=1$  hasta  $k=t$ , mientras  $W(t)$  es una matriz diagonal con la forma:

$$W(t) = \begin{bmatrix} \varphi^{2(t-1)} & 0 & \dots & 0 & 0 \\ 0 & \varphi^{2(t-1)} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \varphi & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Minimizando la función de costo  $J(\hat{a}(t))$ , se obtiene:

$$\hat{a}(t) = [Z(t)^T W(t) Z(t)]^{-1} Z(t)^T W(t) Y(t)$$

Donde  $Z(t)$  es una matriz que contiene las entradas y las salidas del sistema hasta el tiempo  $t$  y tiene la siguiente forma:

$$Z(t) = \begin{bmatrix} u(1) & u(2) & \dots & u(t) \\ y(0) & y(1) & \dots & y(t-1) \\ u(0) & u(1) & \dots & u(t-1) \\ \vdots & \vdots & \ddots & \vdots \\ y(1-n) & y(2-n) & \dots & y(t-n) \\ u(1-n) & u(2-n) & \dots & u(t-n) \end{bmatrix}$$

La ganancia  $C(t)$  se define como:

$$C(t) = [Z^T(t)W(t)Z(t)]^{-1}$$

Realizando algunas operaciones se llega al algoritmo recursivo para  $C(t)$ , que se presenta a continuación:

$$C(t+1) = \frac{1}{\varphi^2} \left[ C(t) - [\varphi^2 + z^T(t+1)C(t)z(t+1)]^{-1} C(t)z(t+1)z^T(t+1)C(t) \right]$$

Donde el vector  $z(t+1)$  contiene las entradas y salidas del sistema, es decir:

$$z^T(t+1) = [u(t+1) \ y(t) \ u(t) \ \dots \ y(t-n+1) \ u(t-n+1)]$$

Como  $[\varphi^2 + z^T(t+1)C(t)z(t+1)]$  es un escalar, se puede definir:

$$\alpha^2(t+1) = \varphi^2 + z^T(t+1)C(t)z(t+1) \quad (2.12)$$

Con lo que se obtiene

$$C(t+1) = \frac{1}{\varphi^2} \left[ C(t) - \frac{1}{\alpha^2(t+1)} C(t)z(t+1)z^T(t+1)C(t) \right] \quad (2.13)$$

El algoritmo recursivo de  $\hat{a}(t)$  es:

$$\hat{a}(t+1) = \hat{a}(t) + \frac{1}{\alpha^2(t+1)} C(t)z(t+1)[y(t+1) - z^T(t+1)\hat{a}(t)]$$

O lo que es lo mismo

$$\hat{a}(t+1) = \hat{a}(t) + \frac{1}{\alpha^2(t+1)} C(t)z(t+1)e_1(t+1) \quad (2.14)$$

Haciendo unas modificaciones al algoritmo se puede mejorar su precisión, esto es de gran ayuda sobre todo para disminuir los errores de truncamiento en las estimaciones en línea. La idea consiste que en vez de actualizar la matriz  $C(t) = [Z^T(t)W(t)Z(t)]^{-1}$ , se actualiza su raíz cuadrada, es decir una matriz  $G(t)$  tal que:

$$G(t)G^T(t) = C(t) \quad (2.15)$$

Reemplazando (2.15) en (2.13) se tiene:

$$G(t+1)G^T(t+1) = \frac{1}{\varphi^2} \left[ G(t)G^T(t) - \frac{1}{\alpha^2(t+1)} G(t)G^T(t)z(t+1)z^T(t+1)G(t)G^T(t) \right]$$

Si  $f(t+1) = G^T(t)z(t+1)$ , entonces:

$$G(t+1)G^T(t+1) = \frac{1}{\varphi^2} \left[ G(t)G^T(t) - \frac{1}{\alpha^2(t+1)} G(t)f(t+1)f^T(t+1)G^T(t) \right]$$

Con lo que:

$$G(t+1)G^T(t+1) = \frac{1}{\varphi^2} \left\{ G(t) \left[ I - \frac{1}{\alpha^2(t+1)} f(t+1)f^T(t+1) \right] G^T(t) \right\}$$

La expresión entre corchetes da como resultado una matriz simétrica, por lo que se puede hacer:

$$\left[ I - \frac{1}{\alpha^2(t+1)} f(t+1)f^T(t+1) \right] = QQ^T$$

Siendo  $Q$  una matriz triangular superior. De esta manera se tiene:

$$G(t+1)G^T(t+1) = \frac{1}{\varphi^2} [G(t)QQ^TG^T(t)]$$

$$G(t+1)G^T(t+1) = \frac{G(t)Q}{\varphi} \cdot \frac{Q^TG^T(t)}{\varphi}$$

Por lo tanto:

$$G(t+1) = \frac{G(t)Q}{\varphi} \tag{2.16}$$

Para calcular la matriz  $Q$  se emplea el método de Cholevsky, el cual permite factorizar una matriz simétrica en dos matrices triangulares, es decir que si se tiene una matriz simétrica  $A$ , mediante este método se puede encontrar una matriz triangular superior  $B$  tal que:

$$B \cdot B^T = A$$

Es así que empleando el método de Cholevsky teniendo que:

$$A = \left[ I - \frac{1}{\alpha^2(t+1)} f(t+1)f^T(t+1) \right]$$

$$B = Q$$

Para el cálculo de la matriz  $Q$  se tiene:

$$q_{jj} = \sqrt{1 - \frac{f_j^2}{\alpha_j^2}} \quad j = n, n-1, \dots, 1$$

$$q_{ij} = -\frac{f_i f_j}{\alpha_j^2 q_{jj}} \quad i = j-1, \dots, 1$$
(2.17)

De (2.12) y (2.15) se deduce que:

$$\alpha^2 = \varphi^2 + f^T f$$

$$\alpha^2 = \varphi^2 + \sum_{k=1}^n f_k^2$$

Por lo que:

$$\alpha_j^2 = \varphi^2 + \sum_{k=1}^j f_k^2$$

Con lo que se tiene:

$$\alpha_0 = \varphi$$

$$\alpha_j^2 = \alpha_{j-1}^2 + f_j^2$$

$$f_j^2 = \alpha_j^2 - \alpha_{j-1}^2$$
(2.18)

Reemplazando (2.18) en (2.17) se obtiene:

$$q_{jj} = \frac{\alpha_{j-1}}{\alpha_j} \quad j = n, n-1, \dots, 1$$

$$q_{ij} = -\frac{f_i f_j}{\alpha_j \alpha_{j-1}} \quad i = j-1, \dots, 1$$
(2.19)

La expresión (2.16) puede reescribirse como:

$$G(t+1)_{ij} = \frac{1}{\varphi} \sum_{k=1}^j G(t)_{ik} q_{kj} \quad (2.20)$$

Con lo que si se reemplaza (2.19) en (2.20) se tiene:

$$G(t+1)_{ij} = \frac{\alpha_{j-1}}{\varphi \cdot \alpha_j} \left[ G(t)_{ij} - \frac{f_j}{\alpha_{j-1}^2} \sum_{k=1}^j G(t)_{ik} f_k \right]$$

$$j = 1, 2, \dots, n$$

$$i = 1, 2, \dots, j$$

Se define el vector  $g(t+1)$  como:

$$g(t+1) = C(t)z(t+1) \quad (2.21)$$

Utilizando (2.15) y (2.21) se llega a:

$$g(t+1) = G(t)f(t+1) \quad (2.22)$$

La expresión (2.22) puede reescribirse como:

$$g_i^{(r)} = \sum_{k=1}^r G(t)_{ik} f_k \quad i = 1, 2, \dots, r$$

$$g_i^{(r)} = 0 \quad i > r$$

Con lo que finalmente se tiene:

$$G(t+1)_{ij} = \frac{\alpha_{j-1}}{\varphi \cdot \alpha_j} \left[ G(t)_{ij} - \frac{f_j g_i^{(j-1)}}{\alpha_{j-1}^2} \right]$$

Resumiendo, las expresiones necesarias para actualizar  $G(t)$  son:

$$\alpha_0 = \varphi$$

$$f(t+1) = G^T(t)z(t+1)$$

$$\alpha_j = \sqrt{\alpha_{j-1}^2 + f_j^2}$$

$$g_i^{(r)} = \sum_{k=1}^r G(t)_{ik} f_k \quad i = 1, 2, \dots, r$$

$$G(t+1)_{ij} = \frac{\alpha_{j-1}}{\varphi \cdot \alpha_j} \left[ G(t)_{ij} - \frac{f_j g_i^{(j-1)}}{\alpha_{j-1}^2} \right]$$

Ordenando las ecuaciones anteriores, se tiene el algoritmo REFIL, como es presentado en [1]

$$\begin{aligned}
 \alpha &= \varphi \\
 \alpha^2 &= \varphi^2 \\
 j &= 1, 2, \dots, p \\
 f &= \sum_{i=1}^j G(t)_{ij} z(t+1)_i \\
 a &= \frac{\alpha}{\varphi} \\
 b &= \frac{f}{\alpha^2} \\
 \alpha^2 &= \alpha^2 + f^2 \\
 \alpha &= \sqrt{\alpha^2} \\
 a &= \frac{a}{\alpha} \\
 g_j &= G(t)_{jj} f \\
 G(t+1)_{jj} &= aG(t)_{jj} \\
 i &= 1, 2, \dots, j-1 \\
 d &= G(t)_{ij} \\
 G(t+1)_{ij} &= a(d - bg_i) \\
 g_i &= df + g_i \\
 &fin \\
 &fin
 \end{aligned}$$

Para calcular el vector de parámetros primero se define al error de estimación como:

$$e_1(t+1) = y(t+1) - \hat{y}(t+1)$$

Si  $\hat{a}(t)$  es el vector de parámetros estimados, entonces:

$$e_1(t+1) = y(t+1) - \hat{a}^T(t)z(t+1)$$

Empleando (2.14) y (2.21) se llega a:

$$\hat{a}(t+1) = \hat{a}(t) + \frac{g(t+1)e_1(t+1)}{\alpha^2(t+1)}$$

De esta manera en el vector  $\hat{a}(t+1)$  se encontrarán los valores  $[b_0 \ a_1 \ b_1 \ a_2 \ b_2 \ \dots \ a_n \ b_n]$  que son los parámetros de la función de transferencia discretizada de la planta.

## **CAPÍTULO 3**

### **SELECCIÓN DEL MÉTODO DE AJUSTE**

#### **3.1. INTRODUCCIÓN**

Después de haber estudiado los esquemas de control adaptativo existentes y los algoritmos de identificación, es necesario hacer un estudio de los métodos de ajuste de los controladores así como también de la ley de control ha emplear con el objeto de tener un control adaptativo eficiente.

Los cambios en el funcionamiento normal del controlador requieren una buena ley de control que mejore el rendimiento, pero si los cambios no son constantes entonces será necesario seleccionar un método de ajuste para la ley de control que permita estabilizar el proceso de la planta.

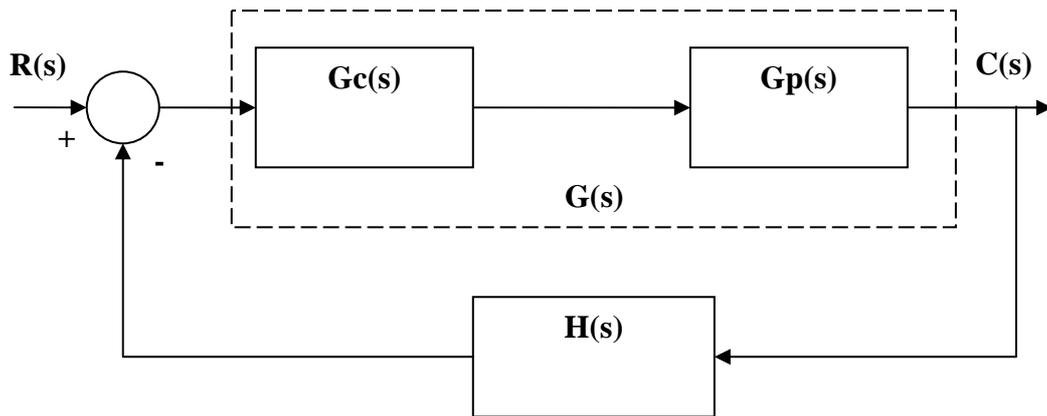
Por lo tanto este capítulo está dividido en dos partes importantes, en el ítem 3.2 se hace un análisis de las leyes de control existentes para finalmente hacer la selección de la mejor ley de control. En el ítem 3.3 se estudian mejoras para la ley de control. En el ítem 3.4 se tratara de los diferentes criterios de sintonización. En el punto 3.5 se estudian los métodos de ajuste de parámetros para la ley de control. Finalmente, en el ítem 3.6 se realiza el desarrollo del software en Matlab del método de ajuste de parámetros.

#### **3.2. LEY DE CONTROL**

##### **3.2.1. Introducción**

La ley de control es el siguiente paso más importante en el diseño de un sistema autoajutable, esta ley de control deberá cumplir con los requerimientos deseados por el operador.

El esquema más general de un sistema de lazo cerrado retroalimentado se muestra en la Figura.3.1, donde  $G(s)$  es nuestro sistema controlado y está compuesto por la función de transferencia del controlador  $G_c(s)$  y la planta  $G_p(s)$ ; y la función de transferencia de la señal del sensor o del acondicionamiento de señal  $H(s)$ .



**Figura.3.1. Esquema de lazo cerrado retroalimentado**

Donde  $C(s)$  es la variable controlada a ser retroalimentada, y  $R(s)$  es la señal deseada o set point. La función de transferencia en lazo cerrado es la siguiente:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (3.1)$$

Existen diferentes métodos de ajuste de ganancia para el controlador pero muchas veces no es suficiente ajustar esta ganancia ya que no se cumple con las especificaciones dadas. Hacer un rediseño o agregar un dispositivo nuevo se llama *compensación* [7], la finalidad del compensador es mejorar las deficiencias que existen en el sistema original.

Antes de diseñar el controlador hay que tomar en cuenta varios parámetros que se deberán analizar como son los polos, ceros, margen de ganancia y fase del sistema.

Los polos tienen una estrecha relación con las ganancias del sistema, esto implica que un simple ajuste de la ganancia logrará desplazarlos a otras posiciones deseadas. El agregar un polo al sistema puede provocar inestabilidad y aumentar el tiempo de establecimiento de la respuesta.

El margen de fase es la cantidad de retraso de fase que se requiere añadir a la frecuencia de transición para llevar al sistema al borde de inestabilidad [7]. El sistema es inestable cuando el margen de fase y ganancia son negativos.

Los márgenes de ganancia y fase son importantes por que adicionan seguridad en los componentes del sistema y se especifican para ciertos valores de frecuencia.

Existen muchas clases de compensadores como los de adelanto, atraso, PI, PD y PID, en este capítulo nos centraremos en el estudio del regulador PID.

### 3.2.2. Estructura del PID

La mayoría de sistemas de control se utilizan en ambientes industriales, las características principales de usar este tipo de controlador son:

- Elimina el error de estado estacionario ya que usa un control integral.
- La respuesta es más rápida por la acción derivativa.
- Disminuye el overshoot.
- La parte derivativa puede amplificar el ruido de alta frecuencia.

La función de salida del controlador es la siguiente:

$$m(t) = Kc e(t) + \frac{1}{Ti} \int_0^t e(t) dt + Td \frac{de(t)}{dt} \quad (3.2)$$

Donde:

$Kc$ : Constante proporcional.

$Ti$ : Tiempo integral.

$Td$ : Tiempo derivativo.

Para cumplir con los objetivos necesitamos discretisar la ley de control para ello tomaremos en cuenta las siguientes igualdades:

$$\left. \frac{dy(t)}{dt} \right|_{t=kT} = \frac{y(k) - y(k-1)}{T} \quad (3.3)$$

y,

$$\left. \frac{d^2 y(t)}{dt^2} \right|_{t=kT} = \frac{y(k) - 2y(k-1) + y(k-2)}{T^2} \quad (3.4)$$

entonces, si se deriva 3.2 se tiene que:

$$\frac{dm(t)}{dt} = Kc \frac{de(t)}{dt} + Kie(t) + K \frac{de^2(t)}{dt} \quad (3.5)$$

reemplazando 3.3 y 3.4 en 3.5 se tiene:

$$m(k) = m(k-1) + [Kc + \frac{T}{Ti} + \frac{Td}{T}]e(k) - [Kc + \frac{2Td}{T}]e(k-1) + \frac{Td}{T}e(k-2) \quad (3.6)$$

donde:

Kc: Constante proporcional

Ti: Tiempo Integral

Td: Tiempo derivativo

La ecuación 3.10 se la conoce como PID digital *Tipo A o versión de velocidad*, donde T es el periodo de muestreo. Esta ecuación ya se puede implementar en un componente digital y será una primera ley de control. Si se toma en cuenta que:

$$e(t) = SP - PV(t) \quad (3.7)$$

siendo,

SP: Set Point.

PV: Present value.

y haciendo la primera y segunda derivada de 3.6 se tiene:

$$\frac{de(t)}{dt} = -\frac{dPV(t)}{dt} \quad (3.8a)$$

$$\frac{d^2e(t)}{dt^2} = -\frac{d^2PV(t)}{dt^2} \quad (3.8b)$$

entonces, reemplazando 3.8a en 3.5 se tiene 3.9 la ecuación digital PID Tipo B

$$m(k) = m(k-1) + Kc[e(k) - e(k-1)] + \frac{T}{Ti}e(k) - \frac{Td}{T}[PV(k) - 2PV(k-1) + PV(k-2)] \quad (3.9)$$

finalmente se obtiene la ley de control empleada, reemplazando 3.8a y 3.8b en 3.5 se tendrá la ecuación digital PID Tipo C:

$$m(k) = m(k-1) - Kc[PV(k) - PV(k-1)] + \frac{T}{Ti}e(k) - \frac{Td}{T}[PV(k) - 2PV(k-1) + PV(k-2)] \quad (3.10)$$

La ecuación 3.10 es una variación de la ecuación 3.7 que mejora el cálculo de la variable de salida del controlador ya que se toma en cuenta el valor presente o present value de la planta y no solo el error de la planta.

### 3.3. Mejoras de la ley de control

#### 3.3.1. Atenuación del rizado

Para el rizado que se presentan por la utilización de la ley de control PID presentada en la fórmula 3.10, es necesario agregar una ganancia  $K_m$  a la ley como se muestra en la siguiente fórmula.

$$m(k) = \left[ \begin{array}{l} m(k-1) - K_c[PV(k) - PV(k-1)] + \frac{T}{T_i} e(k) - \frac{T_d}{T} [PV(k) - 2PV(k-1)] \\ + PV(k-2) \end{array} \right] K_m \quad (3.11)$$

#### 3.3.2. Atenuación del overshoot

Los errores en el overshoot están íntimamente ligados entre el setpoint y la salida del proceso. Ya que el control opera con la señal de error del sistema llamado “system with error feedback” [2]. Para obtener una estructura mejor entre la señal de setpoint y la señal de salida van a ser tratadas por separado.

##### 3.3.2.1. Filtrado del setpoint

Para el filtrado del setpoint se usó un filtro de la forma:

$$Filtro = \frac{1}{Fs + 1} \quad (3.12)$$

Para la eliminación del los componentes de alta frecuencia se obtiene las siguientes ecuaciones:

$$Filtro = \frac{b}{z - a} \quad (3.13)$$

$$SP(k) = a * SP(k-1) + b * SPo \quad (3.14)$$

donde:

a: es el polo de la función

b: es el cero de la función

$SP_0$ : es el setpoint ingresado por el usuario y es un valor constante.

### 3.3.2.2. Setpoint weighting

Sabiendo que el error es la diferencia entre la señal del setpoint y la señal de salida del proceso.

$$e_{sp} = SP - y \quad (3.15)$$

Al ponderar el error se obtendrá una mejora en el control, ya que se disminuye la señal de ruido y perturbaciones al mismo tiempo. La ecuación final será la siguiente:

$$e_{sp} = \beta(SP - y) \quad (3.16)$$

donde,  $\beta$  es una constante que varía entre 0 y 1

## 3.4. Criterios de Sintonización

Existen dos criterios de sintonización según el dominio Temporal [13]:

- Criterios basados en algunos puntos de la respuesta: Este criterio trata de alcanzar el  $\pm 2\%$  del overshoot y minimizar el error en estado estacionario, entre ellos tenemos el método de Ziegler Nichols y Cohen-Coon.
- Criterio basado en la respuesta completa, o criterio integral: Es el criterio de la integral del error ISE, IAE o ITAE. Este criterio se profundizará debido a que se usa para la selección de la mejor ley de control.

Estos métodos del error se analizan para tener una mejor idea del impacto que este tiene en la respuesta independientemente del signo que este tome.

*ISE* (Integral de Error Cuadrático) Se emplea siempre que la magnitud del error no sea mucho menor a 1.

$$ISE = \int_0^{\infty} [e(t)]^2 dt \quad (3.17)$$

*IAE*(Integral del valor absoluto del error) Se emplea con buenos resultados a pesar de que el error sea mucho menor a 1

$$IAE = \int_0^{\infty} |e(t)| dt \quad (3.18)$$

*ITAE*(Integral del valor absoluto ponderado en el tiempo) Se emplea cuando el error es mucho menor a 1 y además el tiempo es una variable importante en el sistema.

$$ITAE = \int_0^{\infty} t|e(t)| dt \quad (3.19)$$

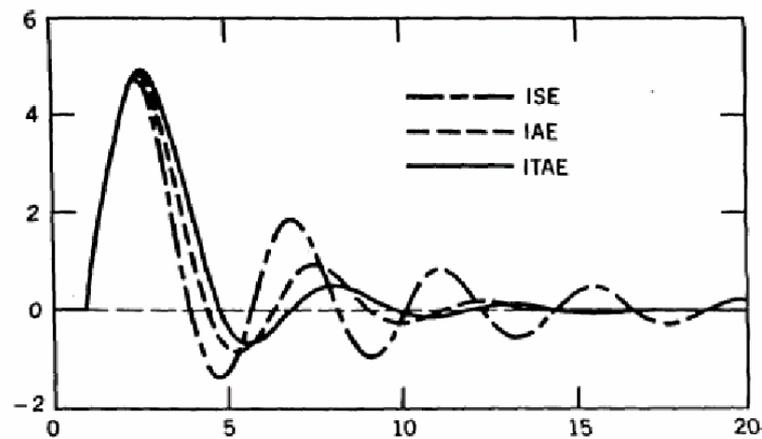


Figura.3.2. Características de rendimiento para cada índice.[14]

### 3.5. AJUSTE DE PARÁMETROS DEL CONTROLADOR

Luego de realizar la identificación de la dinámica del proceso o planta es necesario encontrar el método para ajustar los parámetros del controlador y así lograr un comportamiento estable de acuerdo a algunos parámetros de rendimiento establecido.

#### 3.5.1. PID autoajustable basado en un margen de fase prefijado

Este método de diseño se basa en el criterio clásico de prefijar el margen de fase [1]. Si se considera una función de transferencia discreta de un proceso de segundo orden incompleto con  $k$  pasos de retardo:

$$Gp(z^{-1}) = \frac{b_0 z^{-k}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (3.20)$$

Al aplicar la transformada inversa de  $z$  a 3.6 se tendrá que el controlador es:

$$Gc(z^{-1}) = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2}}{1 - z^{-1}} = \frac{P(z^{-1})}{1 - z^{-1}} \quad (3.21)$$

donde,

$$\begin{aligned} p_0 &= Kc \left( 1 + \frac{T}{Ti} + \frac{Td}{T} \right) \\ p_1 &= -Kc \left( 1 + 2 \frac{Td}{T} \right) \\ p_2 &= Kc \frac{Td}{T} \end{aligned} \quad (3.22)$$

La función de transferencia del sistema en lazo abierto será:

$$G = GpGc = \frac{b_0 p_0 z^{-k}}{1 - z^{-1}} \quad (3.23)$$

Se observa que la ecuación 3.23 contiene un elemento integrador en serie con un elemento de retardo se tiene que:

$$K_I = b_0 p_0 \quad (3.24)$$

La ecuación 3.23 se puede analizar con mayor detenimiento [1] se llega a la conclusión que:

$$K_I = \frac{\pi - 2\varphi_a}{2(2k + 1)} \quad (3.25)$$

donde,  $\varphi$  es el margen de fase del sistema, y si se considera que  $\pi/3$  es un valor adecuado se tiene:

$$K_I = \frac{1}{T^2(2k + 1)} \quad (3.26)$$

Si, se reemplaza 3.26 en 3.24 se tiene que:

$$p_0 = \frac{1}{\hat{b}_0 T^2(2k + 1)} \quad (3.27)$$

donde,  $\hat{b}_0$  es uno de los parámetros realizada en la identificación.

La idea general de este método es encontrar un polinomio de la forma:

$$P(z^{-1}) = p_0(1 + p_1'z^{-1} + p_2'z^{-2}) = p_0(1 - a_1z^{-1} + a_2z^{-2}) = p_0A(z^{-1}) \quad (3.28)$$

donde;

$$\begin{aligned} p_1' &= \frac{p_1}{p_0} = -a_1 \\ p_2' &= \frac{p_2}{p_0} = -a_2 \end{aligned} \quad (3.29)$$

Finalmente se tiene:

$$\begin{aligned} p_1 &= -\hat{a}_1 \hat{p}_0 \\ p_2 &= -\hat{a}_2 \hat{p}_0 \end{aligned} \quad (3.30)$$

donde,  $\hat{a}_1, \hat{a}_2$  y  $\hat{b}_0$  son las estimaciones de los parámetros de la planta realizada en la identificación.

Para calcular las constantes del controlador se usan las siguientes expresiones:

$$\begin{aligned} K_p &= -(p_1 + 2p_2) \\ K_I &= p_0 + p_1 + p_2 \\ K_D &= p_2 \end{aligned}$$

Este método tiene un buen comportamiento con sistemas de primer y segundo orden con varios periodos de retardo [1]. A continuación se muestran unas gráficas comparativas de las variaciones del parámetro  $k$  que son los pasos de retardo y  $K_m$  que es una ganancia autoajutable que atenúa el rizado que son propias de este algoritmo. Este proceso se probó para el sistema:

$$G(s) = \frac{0,04627}{s + 0,0461}$$

La Figura.3.3 presenta el esquema en cual se basan las siguientes comparaciones:

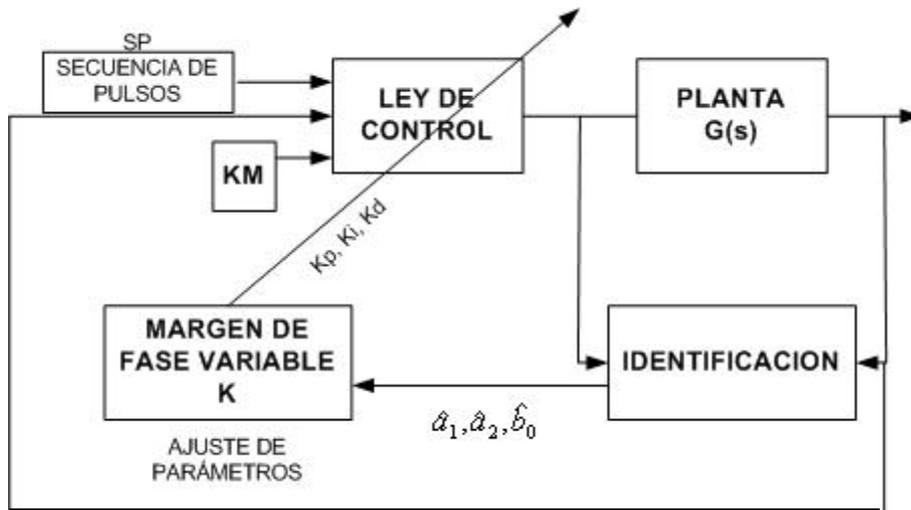


Figura.3.3. Esquema margen de fase

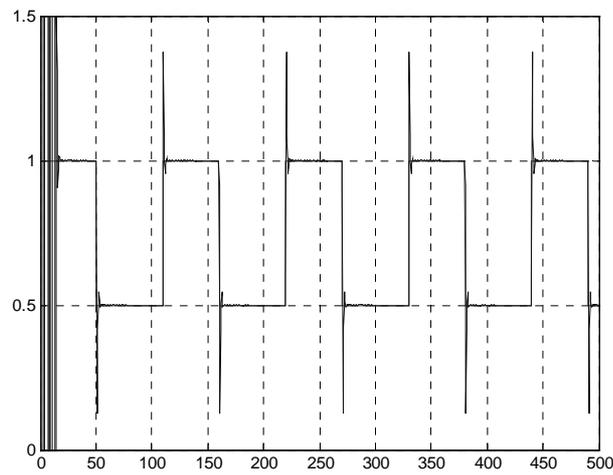


Figura.3.4. Comportamiento de un PID autoajustable con asignación de margen de fase  $k=50$ ,  $K_m=1$

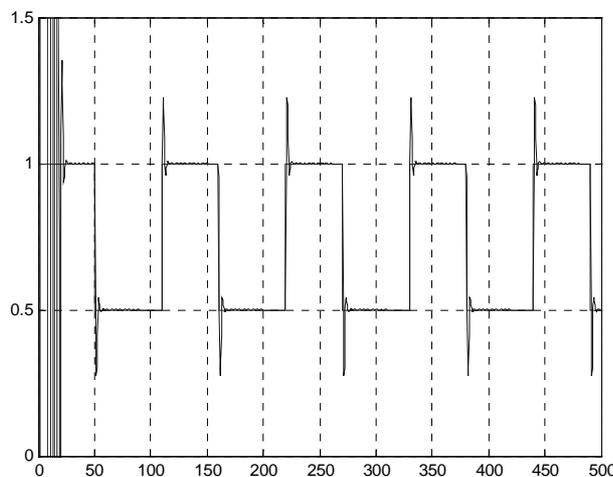
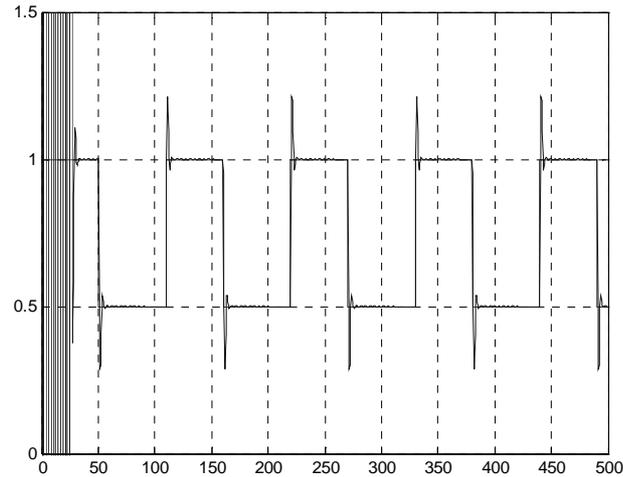


Figura.3.5. Comportamiento de un PID autoajustable con asignación de margen de fase  $k=100$ ,  $K_m=1$



**Figura.3.6. Comportamiento de un PID autoajustable con asignación de margen de fase  $k=10$ ,  $K_m=0.1$**

### 3.5.2. PID autoajustable basado en asignación de polos

Este método se basa en la asignación de polos de un sistema en lazo cerrado. Parte de un modelo de planta [1]:

$$Gp(z^{-1}) = \frac{b_1 z^{-1}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (3.31)$$

La función del controlador es:

$$Gc(z^{-1}) = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2}}{1 - z^{-1}} \quad (3.32)$$

El objetivo de este método resulta cuando se hace  $1 + GpGc = 0$ , entonces se obtiene la siguiente ecuación:

$$z^{-3} + \frac{a_1 - a_2 + b_1 p_1}{b_1 p_2 + a_2} z^{-2} + \frac{b_1 p_0 - a_1 - 1}{b_1 p_2 + a_2} z^{-1} + \frac{1}{b_1 p_2 + a_2} = 0 \quad (3.33)$$

Imponemos ahora la condición de que el sistema en bucle cerrado tenga un polo triple en el plano  $z^{-1}$ , de valor  $a$ , que debe estar en la zona de estabilidad de  $z^{-1}$ . Luego  $a$  debe ser real de módulo mayor que 1. Esto hace suponer que el polinomio característico del sistema tenga la forma:

$$(z^{-1} - a)^3 = z^{-3} - 3az^{-2} + 3a^2 z^{-1} - a^3 \quad (3.34)$$

Al igualar las ecuaciones 3.33 y 3.34 se obtendrá

$$\begin{aligned}
 p_0 &= \frac{1}{\hat{b}_1} \left(1 + \hat{a}_1 - \frac{3}{a}\right) \\
 p_1 &= \frac{1}{\hat{b}_1} \left(\hat{a}_2 - \hat{a}_1 + \frac{3}{a^2}\right) \\
 p_2 &= -\frac{1}{\hat{b}_1} \left(\hat{a}_2 + \frac{1}{a^3}\right)
 \end{aligned} \tag{3.35}$$

De la ecuación 3.6 y sabiendo que:

$$\begin{aligned}
 K_I &= \frac{K_p T}{T_I} \\
 K_D &= \frac{K_p T_D}{T}
 \end{aligned} \tag{3.36}$$

Se calculan las constantes del controlador entonces se tiene:

$$\begin{aligned}
 K_p &= -(p_1 + 2p_2) \\
 K_I &= p_0 + p_1 + p_2 \\
 K_D &= p_2
 \end{aligned} \tag{3.37}$$

La ventaja de este método es que se tiene un solo parámetro de diseño que es el polo triple, pero para evitar inestabilidades es necesario agregar una ganancia autoajustable  $K_m$  que se compensa los polos y ceros inestables y el retardo del tiempo de estabilización.

Este método es empleado para sistemas que calculan constantemente los parámetros del controlador. A continuación unas gráficas comparativas variando el polo triple y la ganancia autoajustable.

La Figura.3.7 se presenta un esquema que será utilizado para observar el comportamiento de un sistema de ajuste de polos en un proceso. Este proceso se probó para el sistema:

$$G(s) = \frac{0,04627}{s + 0,0461}$$

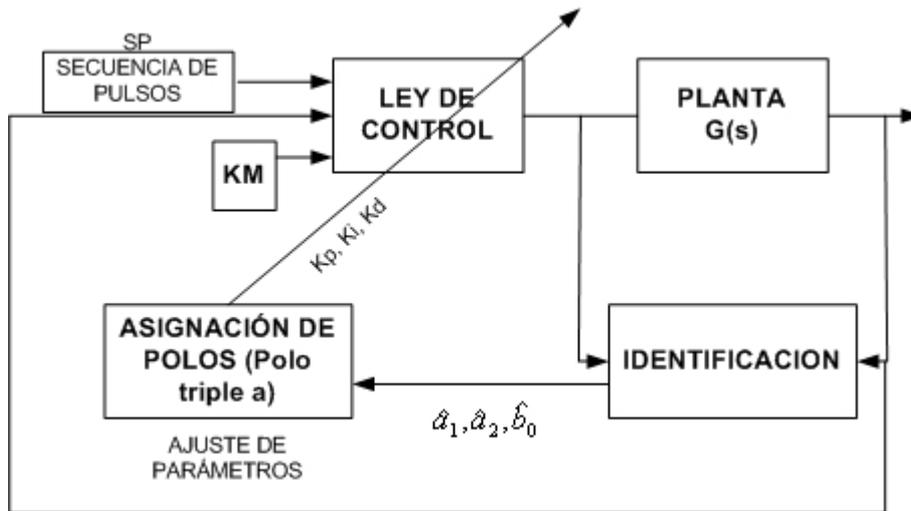


Figura.3.7. Esquema Asignación de polos

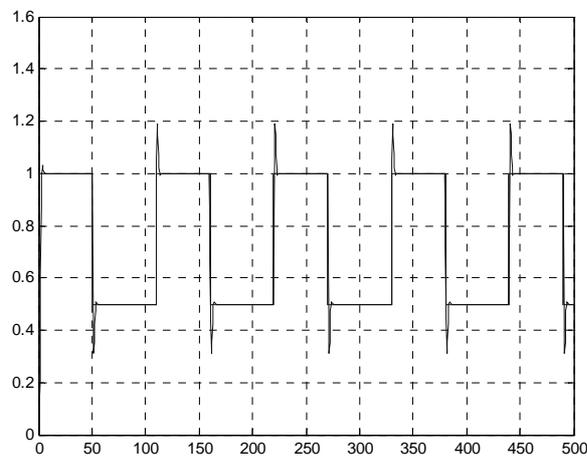


Figura.3.8. Comportamiento de un PID autoajustable con asignación de polos  $a=5$  y  $K_m=0.8$

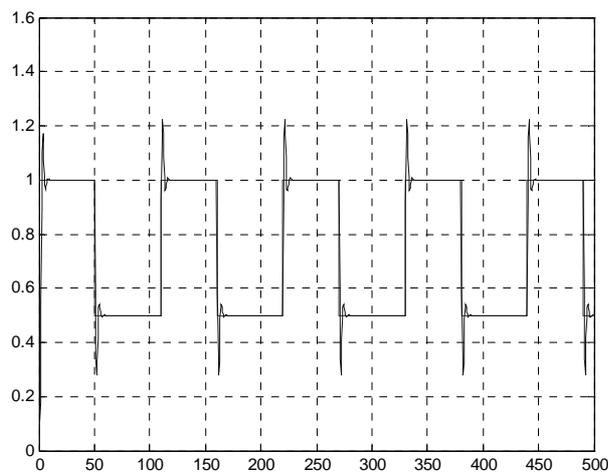
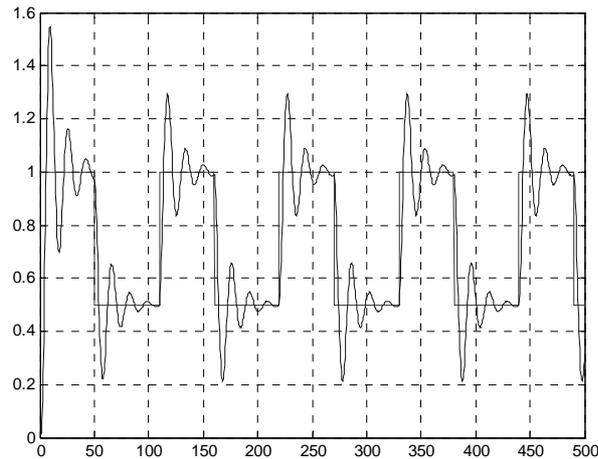
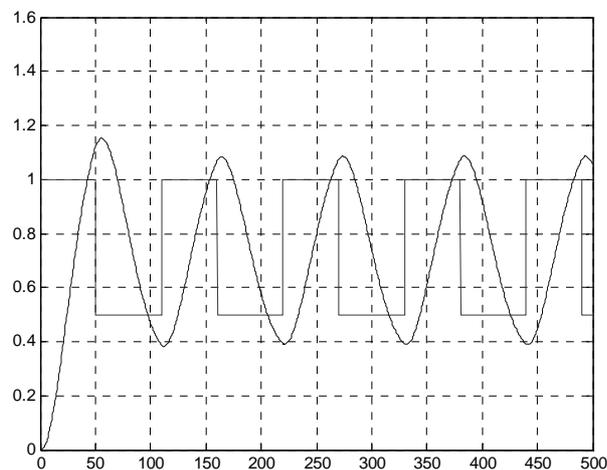


Figura.3.9. Comportamiento de un PID autoajustable con asignación de polos  $a=10$  y  $K_m=0.5$



**Figura.3.10. Comportamiento de un PID autoajustable con asignación de polos  $a=10$  y  $K_m=0.05$**



**Figura.3.11. Comportamiento de un PID autoajustable con asignación de polos  $a=10$  y  $K_m=0.001$**

Como se puede observar en los resultados presentados podemos concluir que entre más grande es el polo triple ( $a$ ) el sistema lo vuelve prácticamente inestable aunque variando el parámetro  $K_m$  se mejora la respuesta del proceso.

### 3.6. Algoritmo de cálculo de los parámetros de la ley de control

Como se vio en el capítulo anterior el método de identificación da como resultado un vector de parámetros como puede ser  $[b_0 \ a_1 \ b_1 \ a_2 \ b_2 \ \dots \ a_n \ b_n]$ , estos parámetros son luego utilizados para el ajuste de parámetros del controlador.

Si se considera un sistema de segundo orden vamos a obtener un vector de parámetros de identificación de la siguiente manera  $[0 \ a_1 \ b_1 \ a_2 \ 0]$ . Este vector sirve para calcular los parámetros de ajuste del controlador utilizando las siguientes fórmulas.

$$\begin{aligned}
 p_0 &= \frac{1}{b_1} \left( 1 + a_1 - \frac{3}{a} \right) \\
 p_1 &= \frac{1}{b_1} \left( a_2 - a_1 + \frac{3}{a^2} \right) \\
 p_2 &= -\frac{1}{b_1} \left( a_2 + \frac{1}{a^3} \right)
 \end{aligned} \quad (3.38)$$

donde,  $a$  será el polo triple del método de asignación de polos,  $a_1$ ,  $a_2$ ,  $b_1$  son los parámetros del proceso de identificación, y  $p_0$ ,  $p_1$ ,  $p_2$  son los parámetros del método de ajuste.

Como se observa en la fórmula 3.10 se necesitan obtener las constantes de la ley de control para ello se hace el uso de las siguientes fórmulas

$$\begin{aligned}
 K_p &= -(p_1 + 2p_2) \\
 K_I &= p_0 + p_1 + p_2 \\
 K_D &= p_2 \\
 T_i &= 1/K_I \\
 T_d &= 1/K_D
 \end{aligned} \quad (3.39)$$

Donde,  $K_D$  es la constante derivativa y  $K_I$  es la constante de integración.

## **CAPÍTULO 4**

### **DISEÑO DEL SOFTWARE**

#### **4.1. INTRODUCCIÓN**

En este capítulo se va a desarrollar el software basándose en lo estudiado en los capítulos anteriores. Como se estudió en el capítulo 1 se escogió el método de self-tuning regulator para lo cual se necesita un proceso de identificación de la planta y el ajuste de la ley de control como se muestra en la Figura.1.10., para ello hemos escogido el método de identificación de mínimos cuadrados tratado en el capítulo 2 y el método de ajuste de asignación de polos estudiado en el capítulo 3.

Se diseñaron tres programas. Uno en Matlab donde se realiza la identificación de los parámetros de la planta y el cálculo de los parámetros del controlador. El segundo programa es la interface HMI que se realizó utilizando LabView. El tercero es el programa del microcontrolador donde se ejecuta la ley de control.

Matlab tiene la capacidad de resolver procesos matemáticos complejos por esta razón fue escogido para ejecutar la identificación de los parámetros de la planta que exige gran capacidad de procesamiento.

Para realizar el control adaptativo se necesita la adquisición de datos de la planta de temperatura en tiempo real por esta razón se utiliza LABVIEW ya que este programa permite gestionar fácilmente el manejo de ingreso y salida de datos hacia un dispositivo exterior, además permite la adición de código de MATLAB

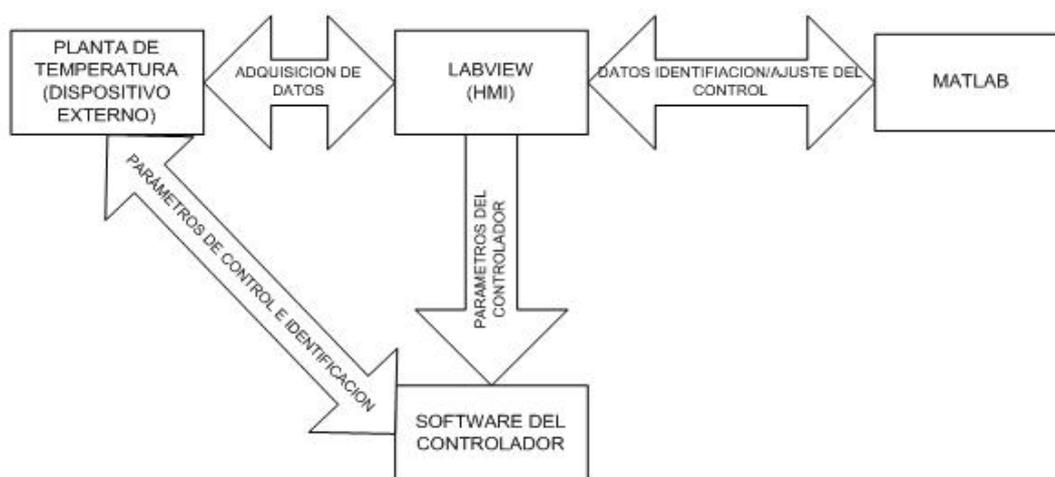
La tarea de control se implementó en un microcontrolador debido a que se requiere que esta tarea se ejecute independientemente del computador. Por lo tanto se desarrolló un software para el microcontrolador en el que se incorpora la ley de control con parámetros ajustables que son enviados desde la computadora.

Con lo anteriormente mencionado éste capítulo está distribuido de la siguiente manera; en el ítem 4.2 Generalidades, se realizará una breve explicación sobre el funcionamiento de cada uno de los programas utilizados así como también del funcionamiento en conjunto, en

el ítem 4.3 se desarrolla el funcionamiento del software en LABVIEW y para finalmente en el ítem 4.4 se expone el desarrollo del software para el dispositivo de control utilizado.

## 4.2. GENERALIDADES

La interfaz HMI donde se encuentra el panel visual de cada uno de los parámetros de la planta para la identificación y para el controlador es desarrollada en LABVIEW y es por medio de esta que se envían los datos para ser procesados hacia MATLAB y hacia el controlador. Como se muestra en la siguiente figura.



**Figura.4.1. Diagrama de bloques del software**

El programa de LABVIEW desarrollado, adquiere los datos de la planta de temperatura por medio de una tarjeta de adquisición de datos compatible con National Instrument y cuyas funciones son instaladas y pueden ser usadas por este software. Para que estos datos sean procesados en MATLAB se necesita de una función proporcionada por LABVIEW llamada MATLAB Script y las funciones de Serial VIs que sirve para enviar los datos hacia el dispositivo de control.

El programa en LABVIEW realiza un bucle infinito pero con un tiempo de muestreo determinado por el operador. Luego de adquirir el valor de salida de la planta, éste en primera instancia entra en la etapa de desarrollada dentro de un MATLAB Script. Este programa realizado en MATLAB determina los parámetros de identificación basado en la lectura de la planta, tiempo de muestreo y otros valores. Los parámetros de la planta determinados son enviados hacia una segunda función realizada en MATLAB Script que determina los parámetros de ajuste del controlador basándose además en otros valores de acuerdo al método de ajuste escogido.

Los parámetros del controlador y otros valores de sintonización finalmente son enviados a través del puerto USB hacia el controlador, el cual contiene el programa de la ley de control.

Cuando la variable de control llega a un valor cercano al SETPOINT el lazo de identificación se detiene, guardándose así los últimos parámetros de identificación y funcionando únicamente el lazo de control. También, al inicio se espera un tiempo hasta que se realiza una identificación preliminar antes que empiece a controlar debido a que siempre los primeros valores no son siempre los correctos.

El programa del controlador por su parte guarda los últimos parámetros de la ley de control, para que el control funcione sin necesidad de usar una computadora.

### 4.3. DESARROLLO DEL SOFTWARE EN LABVIEW

Este software nos permite tener controles visuales tanto de entrada como de salida de los datos de la planta. En la Figura.4.2., muestra el panel del control adaptativo realizado.

En dicha interfaz se muestran controles indicadores de salida y entrada de texto, que permiten visualizar y modificar las constantes de la identificación y control de la planta, también se muestra un control que grafica la variable controlada, el valor del setpoint y los parámetros de identificación. Consta de dos botones los cuales nos sirve para ver los errores que se pueden presentar en la ejecución del programa, y opciones avanzadas que permiten cambiar los parámetros de la identificación y controlador para los usuarios que quieran otra forma de ejecución.



Figura.4.2. Interfaz HMI

### 4.3.1. SUBVIs empleados

El programa contiene diferentes SubVI, que son pequeños programas, debidos a que la programación visual es muy extensa, los SubVIs utilizados se muestran en la siguiente figura.

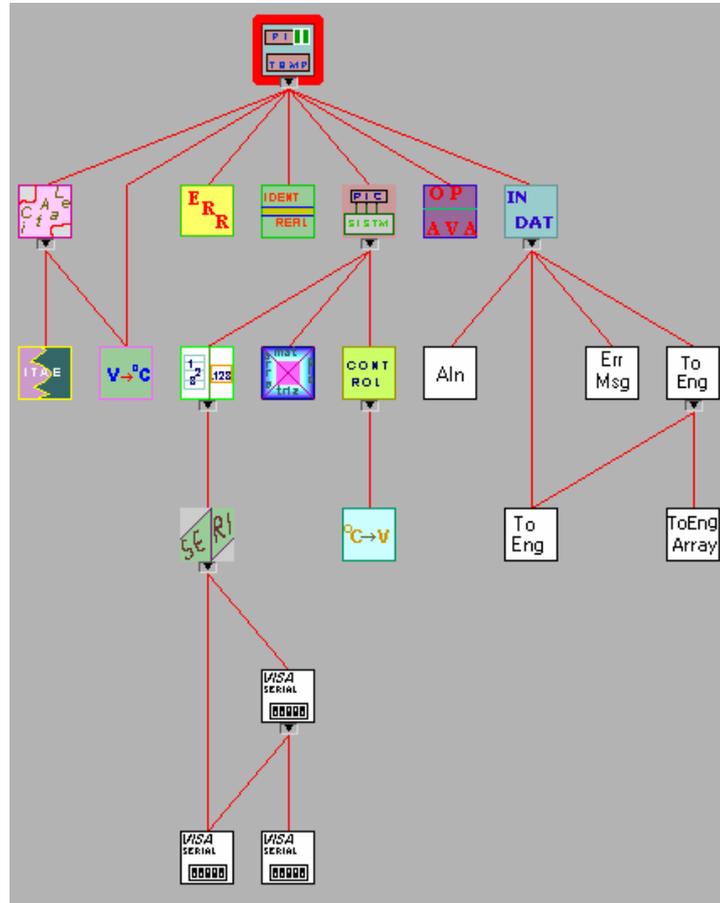


Figura.4.3. Diagrama Jerárquico de los SubVIs usados.

#### 4.3.1.1. INDATA



Figura.4.4. SubVI lectura de la tarjeta de adquisición de datos

Este SubVI sirve para leer los datos de la tarjeta de adquisición de datos PMD-1208LS, para ello es necesario haber instalado la tarjeta en la PC como se muestra en el ANEXO 4. Luego de haber instalado la tarjeta los siguientes instrumentos virtuales básicos como son Aln, ErrMsg, To Eng, To Eng Array que son explicados en el ANEXO 7.

El diagrama de bloques de este SubVI es el siguiente:

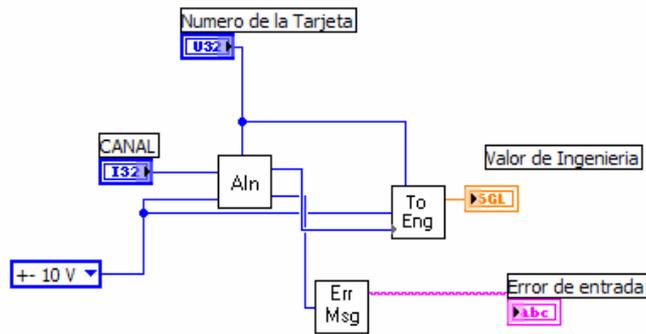


Figura.4.5. Diagrama de bloques del SubVI de INDATA

### 4.3.1.2. SISTEMA DE CONTROL

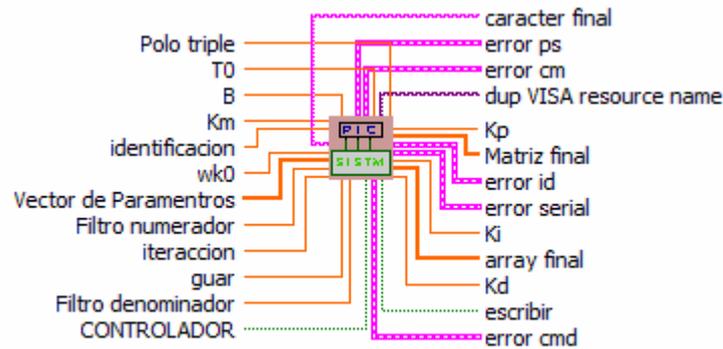


Figura.4.6. Diagrama del SubVI del Sistema de Control

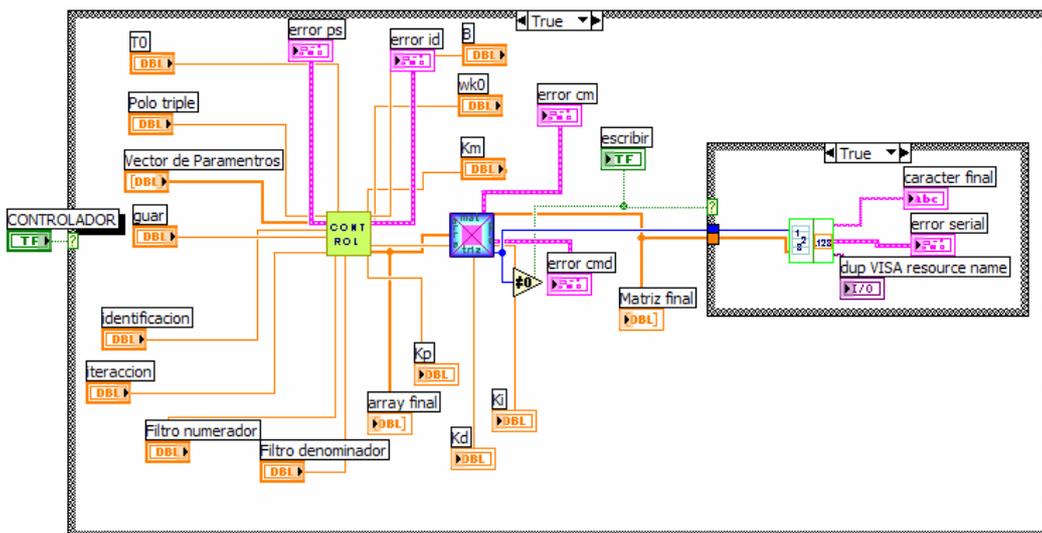
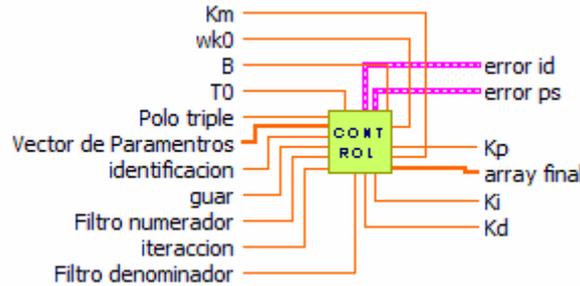


Figura.4.7. Diagrama de Bloques del SubVI del Sistema de Control

Este SubVI se encarga de realizar todas las funciones para el controlador y el envío de datos hacia el dispositivo de control, esta función a su vez tiene tres diferentes SubVI. El diagrama de bloques de esta función es el que se muestra en la Figura.4.7

**4.3.1.2.1. CONTROL**

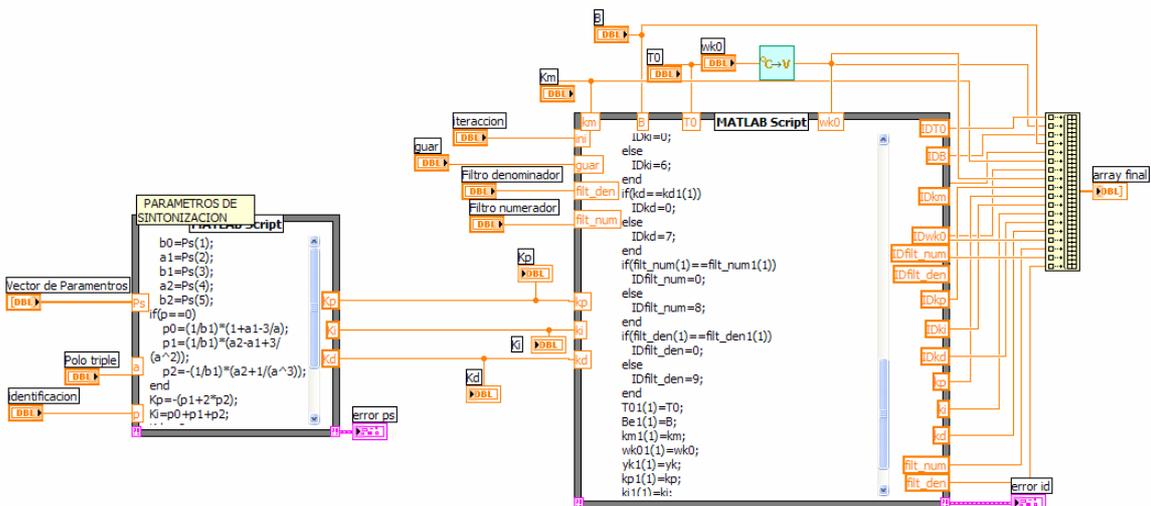


**Figura.4.8. Diagrama del SubVI de control**

Este SubVI tiene el programa de ajuste de parámetros para el controlador que se explicó en el ítem 4.3, así como la conformación de una matriz que envía todos los parámetros hacia el dispositivo de control como son: tiempo de muestreo, valor del setpoint, constantes de sintonización para la ley de control, y constantes del filtro de setpoint.

Si alguna variable ha sido cambiada, entonces los identificadores de cada variable cambian de 0 a su valor respectivo (ver Tabla. 4.1) para luego ser enviadas hacia el controlador.

El diagrama de bloques de esta función es la siguiente.



**Figura.4.9. Diagrama de Bloques del SubVI de CONTROL**

### 4.3.1.2.2. Arreglo Serial

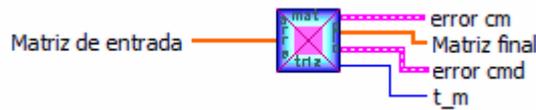


Figura.4.10. Diagrama de la función arreglo serial

Esta función ordena la matriz formada por las constantes explicadas en el ítem anterior de tal manera que solo los datos cambiados sean enviados por el puerto USB hacia el dispositivo de control, de esta manera se optimiza el envío de datos, si no se realizó ningún cambio envía una señal que evita que se transfieran datos innecesarios.

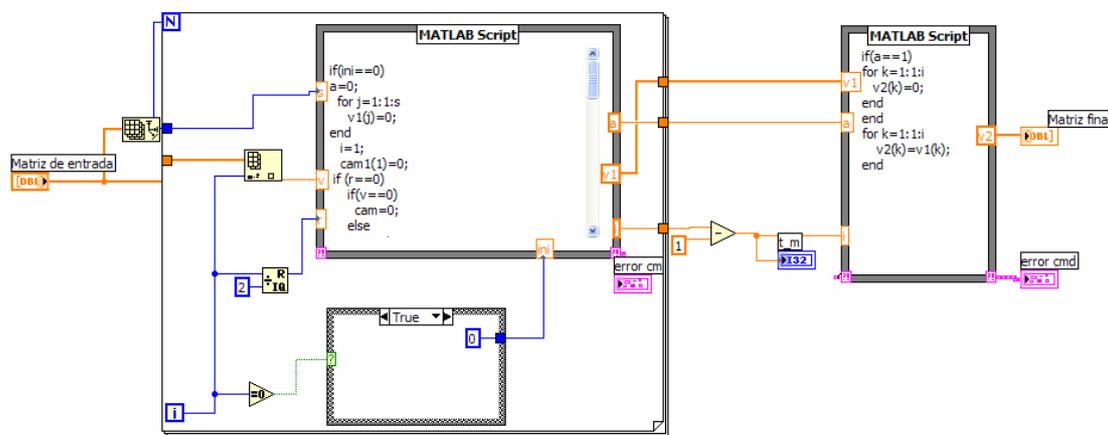


Figura.4.11. Diagrama de bloques del SubVI de arreglo serial

### 4.3.1.2.3. Arreglo\_envio

Esta función convierte en cadena de caracteres los datos de la matriz enviada, el identificador es cambiado por letras desde la a hasta la i dependiendo del valor del identificador, como se puede ver en la Tabla. 4.1, y el valor del dato es transformado por medio de la función number to engineering string, esta función convierte el número en formato de ingeniería pero en cadena de caracteres. Finalmente se agrega un espacio que es un carácter que sirve para la sincronización.



Figura.4.12. Diagrama del SubVI arreglo\_envio

El diagrama de bloques de este SubVI es la que se muestra en la siguiente figura.

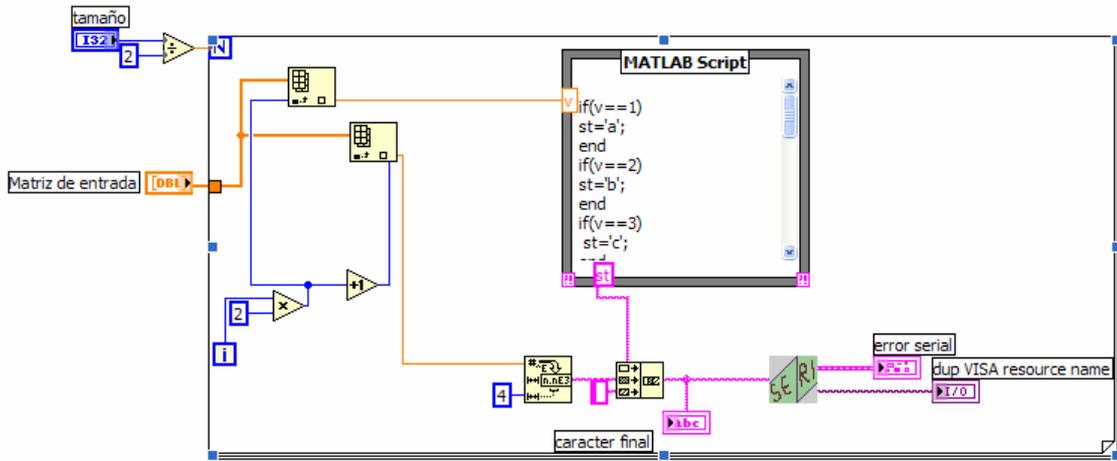


Figura.4.13. Diagrama de bloques del SubVI arreglo\_envio

4.3.1.2.4. Comunicación Serial

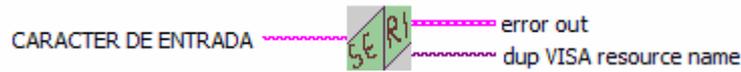


Figura.4.14. Diagrama del SubVI comunicación serial

Esta función configura el puerto serial para el envío de datos. Es necesario cerrar el puerto serial después de enviar los datos, pero al enviar constantemente valores es necesario cerrar el puerto cuando todos los valores se han enviados. El puerto USB es detectado como un puerto serial virtual en este caso COM 4, la velocidad de envío de datos es de 30000 bps debido a la gran cantidad de datos enviados hacia el dispositivo de control. El diagrama de bloques es el siguiente.

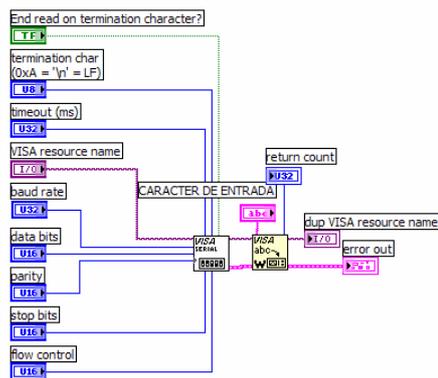


Figura.4.15. Diagrama de Bloques del SubVI comunicación serial

4.3.1.3. Identificación

Esta función realiza la identificación de la planta por medio de MATLAB la cual se estudio en el ítem 4.3. El diagrama de bloques es el de la Figura.4.17.

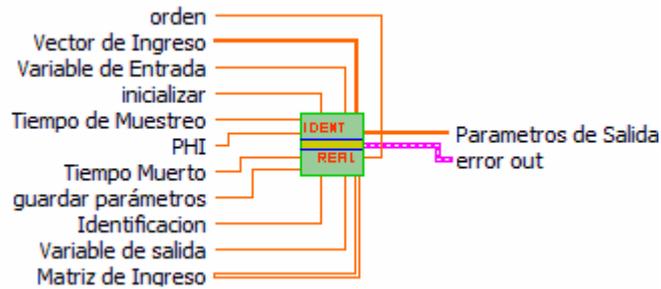


Figura.4.16. Diagrama de la Función de Identificación

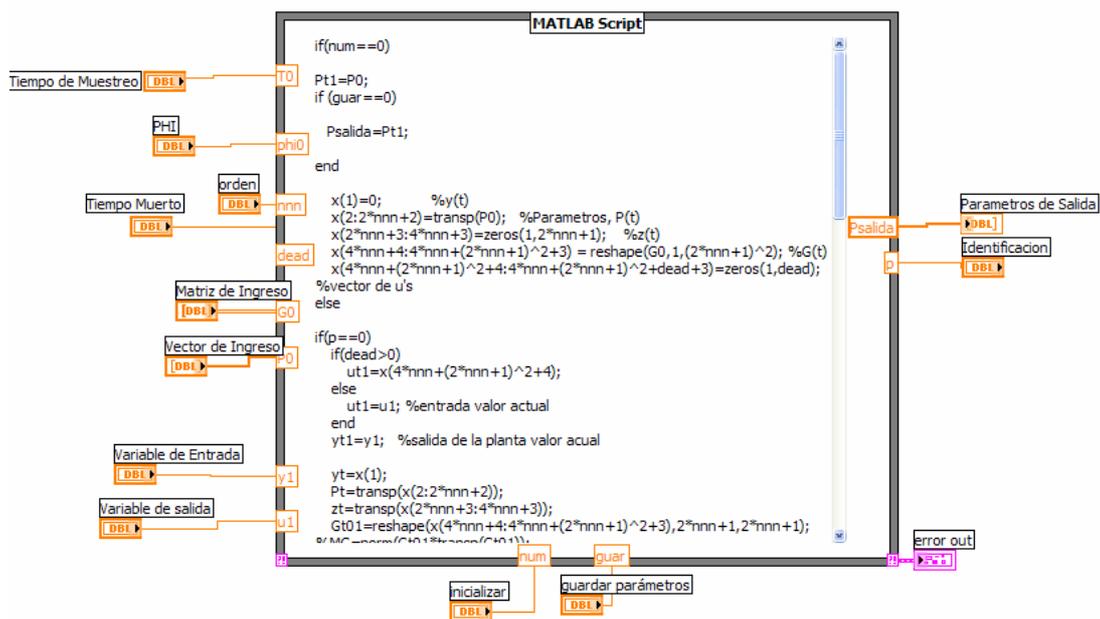


Figura.4.17. Diagrama de bloques del SubVI identificación

#### 4.3.1.4. Calculo ITAE

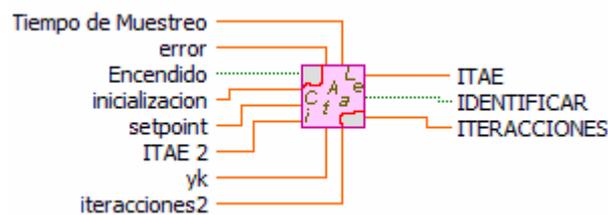


Figura.4.18. Diagrama de la función Calculo ITAE

Esta función realiza el cálculo del ITAE del sistema de control, este cálculo es importante ya que nos determinó según este criterio la ley de control a utilizar según los valores obtenidos en las pruebas realizadas, los cuales se estudiarán en el capítulo 7. El diagrama de bloques es el siguiente.

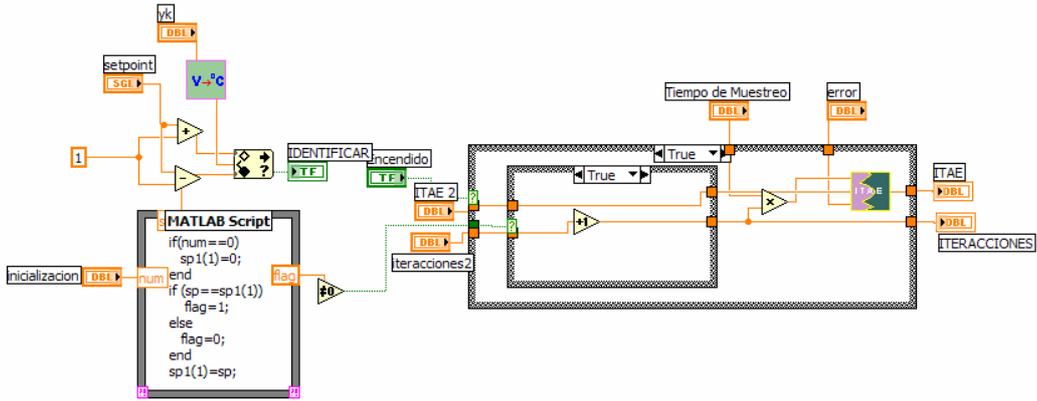


Figura.4.19. Diagrama de bloques del SubVI Cálculo ITAE

Esta función también nos envía una señal de detención de identificación. Cuando este valor indica 1 el valor del setpoint está en el rango de  $\pm 1$  grado del setpoint y la identificación se detiene, caso contrario la identificación continúa y el valor indicador será 0.

**4.3.1.5. Funciones convertir grados a voltios/ voltios a grados**

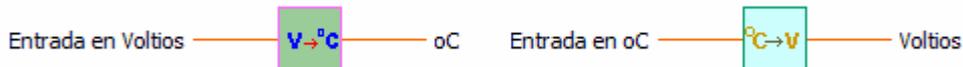


Figura.4.20. Diagrama SubVI conversión

Estas funciones nos facilitan la conversión de grados a voltios y viceversa, con estas funciones nos ahorramos mucho espacio en el programa final.

**4.3.1.6. OPCIONES AVANZADAS**

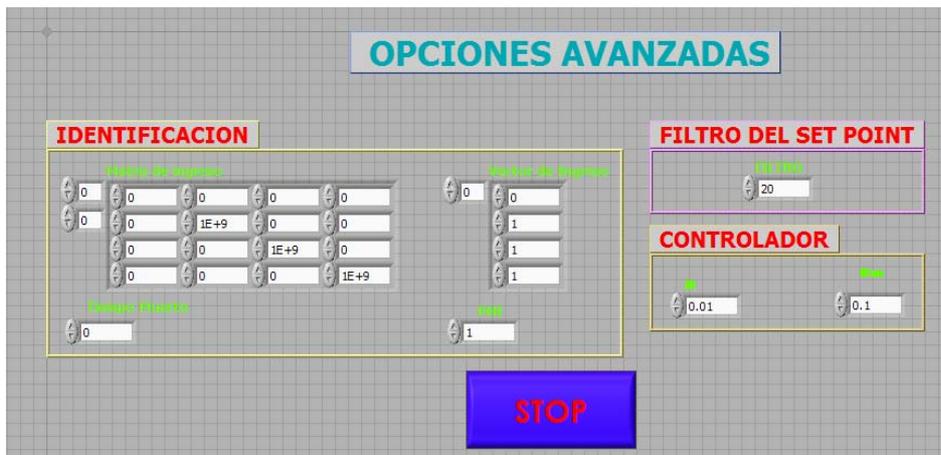


Figura.4.21. Panel de Opciones Avanzadas

Esta función abre un panel que contiene valores que modificarán tanto la identificación de la planta como la ley de control. Los valores que se encuentran en este panel fueron obtenidos luego de realizar varias pruebas y están colectados como default.

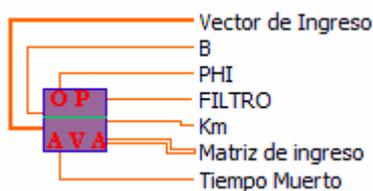


Figura.4.22. Diagrama del SubVI de Opciones Avanzadas

#### 4.3.1.7. ERRORES

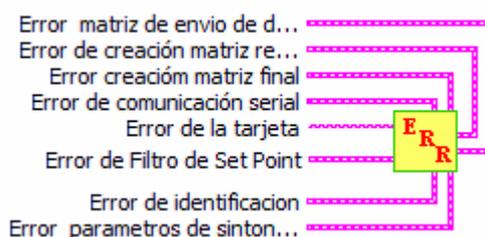


Figura.4.23. Diagrama del SubVI de Errores.

Esta función presenta un panel con los errores que se podrían presentar durante la ejecución del programa. El panel se muestra en la siguiente figura.

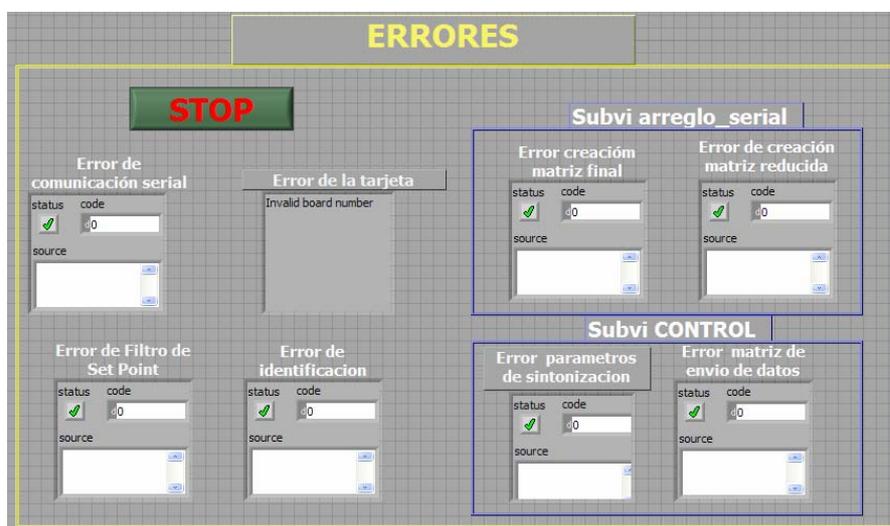


Figura.4.24. Diagrama del Panel Errores

#### 4.3.2. Explicación del Diagrama de Bloques General

Luego de haber explicado la funcionalidad de cada SubVI se hace más fácil la explicación del programa general. El diagrama de bloques se presenta en la Figura.4.25.





Como se ve en la Figura.4.25, el diagrama está compuesta por diferentes estructuras de control, la estructura más externa es una Flant sequence structure que divide en dos estructuras principales, esto se realizó de esta manera debido a que era necesario cerrar el puerto serial virtual una vez éste haya terminado con la ejecución del programa, por lo tanto la primera secuencia contiene el programa principal y la segunda secuencia cierra el puerto serial.

El segundo bucle es un bucle infinito o while loop en el interior de este se encuentra otra secuencia infinita. La secuencia interna es ejecutada en cada tiempo de muestreo, dicho tiempo es asignado por el operario, en este bucle se encuentran los SubVI de identificación y control que se explicaron anteriormente así como también el cálculo del filtro del setpoint dependiendo del valor ingresado en opciones avanzadas (Figura.4.26).

En el bucle externo se leen los datos de la tarjeta de adquisición, se calcula el valor del ITAE, se presentan el sistema de control gráficamente, y se presentan los SubVI de opciones avanzadas y errores.

#### **4.4. DESARROLLO DEL SOFTWARE PARA EL CONTROLADOR**

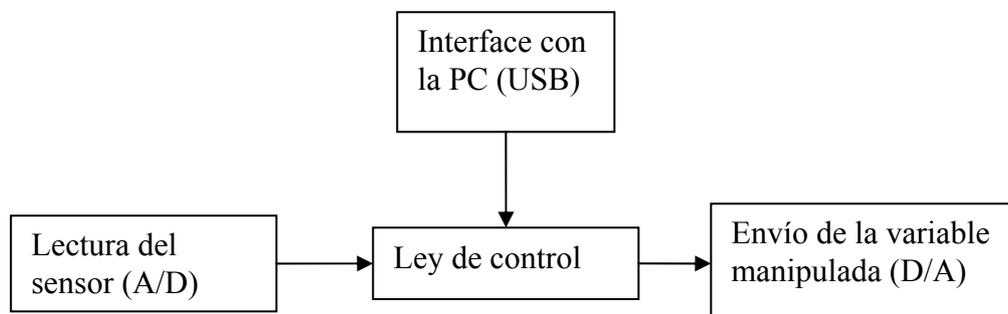
El microcontrolador escogido es el PIC 18F2550, los detalles acerca de este microcontrolador se describen en el capítulo 6, además las especificaciones se pueden encontrar en el datasheet de dicho producto [16], de dicho documento se han tomado los datos y ciertas fórmulas para poder desarrollar el programa estudiado en este ítem.

En el microcontrolador se encuentra la ley de control que calcula el valor de la variable manipulada que se debe enviar a la planta según los parámetros enviados desde el computador. Además, como ya se dijo anteriormente, el controlador debe ser capaz de ejecutar su tarea de control aunque se desconecte del PC. Esto se logra guardando los últimos parámetros recibidos en la memoria EEPROM del microcontrolador; de esta manera, aunque se desconecte el controlador de la PC e inclusive se lo apague y se lo encienda sin conectarlo al computador el controlador podrá realizar su tarea sin inconvenientes.

El microcontrolador se programó utilizando el compilador MPLAB C18, de Microchip. Es un compilador ANSI C diseñado para la familia de microcontroladores PIC18. Sigue el estándar ANSI X3.159-1989 en todo lo que es posible para el microcontrolador, por lo que

su programación y tipos de datos son muy similares al los del lenguaje C. Esto facilita en gran manera el manejo de datos de punto flotante.

El software en el micro controlador se puede dividir en cuatro grandes bloques, que son:



**Figura.4.27. Diagrama de bloques del software del microcontrolador**

#### **4.4.1. Interface con la PC**

Mediante la interface con la PC llegan al microcontrolador diferentes parámetros del control, como por ejemplo el período de muestreo y el set point; así como también los valores de sintonización de la ley de control.

La transferencia de datos entre el microcontrolador y el computador se realiza utilizando un bus USB (Universal Serial Bus). Las funciones para implementar la comunicación USB fueron tomadas del firmware que la Microchip, el fabricante del microcontrolador, provee para implementar esta comunicación. Dicho software es gratuito y la licencia del mismo permite su modificación y su uso libre en productos de la Microchip [9].

##### **4.4.1.1. Protocolo USB**

El estudio del protocolo USB es amplio y está fuera del objetivo de esta tesis, por lo tanto a continuación se describen solamente ciertas generalidades del mismo. Otras características técnicas importantes, se detallan en el capítulo 6.

Primero, se pueden definir dos tipos de componentes USB: el PC host, y el periférico (dispositivo USB). Considerando la comunicación USB como un sistema maestro-esclavo, el PC host vendría a ser el *maestro* del sistema. Cuando se conecta un dispositivo al bus USB, el host inicia un proceso llamado *numeración*, en el cual el se identifica y se asigna una dirección al periférico. Una vez identificado, el host carga los drivers necesarios para

manejar la comunicación con dicho dispositivo. El host puede proporcionar además energía para el funcionamiento del periférico<sup>1</sup>.

El periférico es el dispositivo conectado al bus USB. Los dispositivos se pueden agrupar en *clases (USB class)*. Una *clase* “describe un grupo de dispositivos o interfaces con atributos o servicios similares” [15]. Algunas clases de dispositivos son: clase de audio, clase de dispositivos de comunicación, clase de dispositivos de interfaz humana (HID), clase de almacenamiento masivo y clase de impresoras.

#### 4.4.1.2. Características de la comunicación

A continuación se detallan algunos aspectos de la comunicación. Debido a que se basa en un firmware de la Microchip, los detalles que a continuación se presentan son tomados de [9].

- Las especificaciones USB en las que se basa la comunicación se refieren a las *Especificaciones USB revisión 2.0 (Universal Serial Bus Specification)*.
- La comunicación USB implementada emula una comunicación RS-232. Es decir crea en la computadora un puerto COM virtual. Para esto la computadora emplea dos drivers incluidos en las versiones de Microsoft Windows 98 y posteriores; estos drivers son *usbser.sys* y *ccport.sys*.<sup>2</sup>
- Un dispositivo de emulación serial se define en el protocolo USB dentro de la clase de dispositivos de comunicación (*Communication Device Class*) o CDC. Los drivers de Microsoft Windows cumplen con las especificaciones de esta clase. El dispositivo desarrollado también cumple con la especificación CDC.

#### 4.4.1.3. Software de Comunicación

Las principales funciones del firmware de la Microchip que se emplean son las siguientes:

***mInitializeUSBDriver()***: Inicializa la comunicación USB, especificando los siguientes parámetros:

---

<sup>1</sup> Máximo hasta 500mA.

<sup>2</sup> Microsoft Windows no tiene un archivo *.inf* estándar para los drivers CDC, por esto se emplea el archivo *mchpcdc.inf* provisto por la Microchip

*Velocidad:* La velocidad empleada es Full-Speed, que permite una comunicación de hasta 12Mb/s.

*Usar transeiver interno:* Usa el transeiver interno del PIC, es decir que las salidas desde el PIC son directamente las señales D+ y D-.

*PING-PONG buffering:* La función *ping-pong buffering* se deshabilita. Esta función separa los buffers de entrada pares de los de buffers impares, de esta manera mientras el CPU procesa un buffer, la interface procesa el otro. Así se consigue una mayor velocidad.

*Resistencias de Pull-up internas.* Las resistencias de pull-up internas están habilitadas. Si se quiere una comunicación *full-speed*, se debe poner una resistencia de pull-up a la línea D+, mientras que para una velocidad *low-speed*, se debe colocar una resistencia de pull-up en la línea D-.

*USBTasks():* En esta función se llaman otras funciones que cumplen la tarea de comunicación con el host:

*USBCheckBusStatus():* Verifica si el dispositivo esta conectado al bus USB.

*USBDriverServices():* Inicializa la comunicación con el host cuando se conecta el dispositivo al bus USB.

*CDCTxService():* Se encarga de la comunicación dispositivo-host.

*getsUSBUSART(char \*buffer, byte len):* Copia en *buffer* los datos recibidos a través del bus USB. El número de bytes que se espera recibir se indica con *len*. Si los bytes recibidos son mayores a *len*, solo se copian el número de bytes indicados por *len*. Si los bytes recibidos son menores a *len*, se copian solamente el número de bytes recibidos. Si no se han recibido datos, la función devuelve 0.

Cabe resaltar que ninguna de estas funciones “bloquea” el programa, ni se pueden usar en programas con funciones “bloqueadoras”. Por ejemplo, una función bloqueadora es la siguiente:

```
while(PORTBbits.RB0==0);
```

El programa no saldrá de este lazo hasta que en el puerto RB0 exista un 1. Del mismo modo, la función `getsUSBUSART(char *buffer, byte len)` estaría utilizada incorrectamente si se hace lo siguiente:

```
while(getsUSBUSART(buffer_entrada, 10));
```

El modo correcto de utilizar estas funciones es llamarlas en cada repetición del bucle del programa, de esta manera:

```
void main(void)
{
    ...
    ...
    while(1)
    {
        ...
        if((getsUSBUSART(buffer_entrada, 10))
        {
            recibido=1;
        }
        ...
    }
}
```

En cada iteración del bucle principal se revisa si han llegado datos, y si es así, se guardan en `buffer_entrada` y se activa la bandera `recibido`.

En el software implementado en el microcontrolador, la función `getsUSBUSART` se llama con los con los parámetros `USBent`, que es el arreglo donde se guardan los datos recibidos, y el largo que se espera recibir es `15`, que es el máximo número de bytes que se envían desde la PC.

En la Figura.4.28 se presenta un diagrama de flujo que indica el funcionamiento del software de la parte de comunicación del microcontrolador. Se puede observar que siempre se revisa si hay datos en el buffer de entrada, aunque el dispositivo no esté conectado. Esto es para que se lean todos los datos del buffer que pudieron quedar ahí cuando se desconectó el dispositivo. Es evidente que no llegarán datos nuevos al buffer cuando el dispositivo esté desconectado.

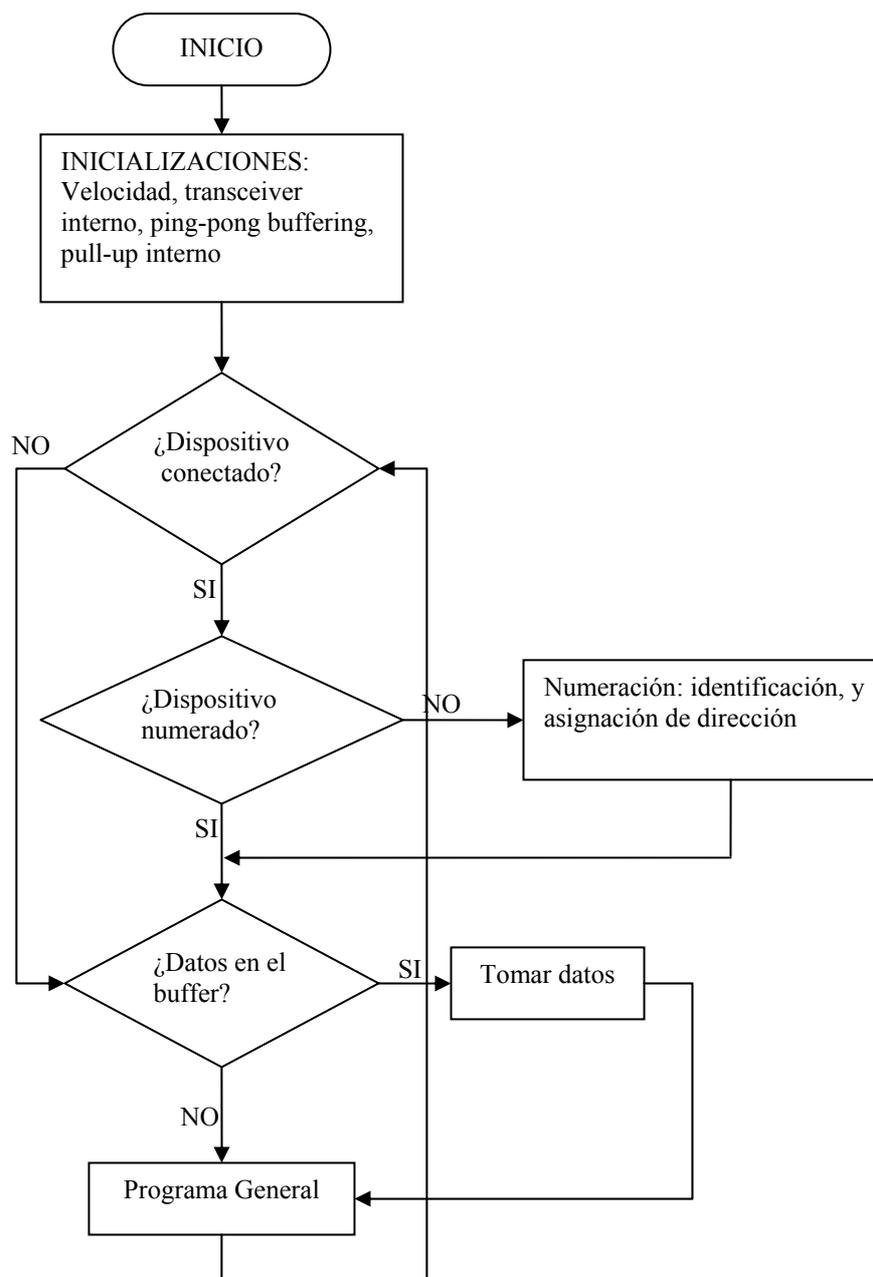


Figura.4.28. Diagrama de flujo, comunicación USB

#### 4.4.1.4. Recepción de datos

Una vez que los datos han sido recibidos, estos deben interpretarse y se deben actualizar los parámetros del controlador con la nueva información. Primero, se debe identificar a que variable corresponde el nuevo dato recibido. El primer caracter de la cadena recibida sirve de identificador, como se ve en la Tabla. 4.1.

IDENTIFICADOR	VARIABLE	SIGNIFICADO
a	T0	Período de muestreo
b	B	Constante $\beta$
c	km	Constante $km$
d	wk0	Set Point
e	Kp	Constante proporcional
f	Ki	Constante integral
g	Kd	Constante derivativa
h	vB	Constante B del filtro de SP
i	vA	Constante A del filtro de SP

**Tabla. 4.1. Identificadores de parámetros**

El valor recibido llega en forma de cadena de caracteres. Por ejemplo si desde la PC se requiere enviar el número  $-0.00035124$  en realidad se envía la cadena “-351.24E-6”. Esta cadena necesita ser transformada a su valor real, para eso se diseñó la función *transformacion()*.

Para explicar el funcionamiento de *transformacion()*, tomemos la cadena “-351.24E-6” como ejemplo. Primero se hace la conversión de la parte entera es decir de 351, para esto se tiene que:

$$351 = 1 \times 1 + 5 \times 10 + 3 \times 100$$

Si se tiene el número entero en un arreglo, solamente se tiene que multiplicar el primer valor del arreglo por 1, el segundo por 10, el tercero por 100 y así sucesivamente; y al final sumar todos los resultados.

Para la parte decimal se tiene que:

$$0.24 = \frac{2}{10} + \frac{4}{100}$$

Es decir que si se tiene el número decimal en un arreglo, se tiene que dividir el primer valor posterior al punto para 10, el siguiente valor para 100, el siguiente para 1000 y así sucesivamente; al final se tiene que sumar los resultados de todas las divisiones.

Para tener el número completo se suman los resultados del cálculo de la parte entera y de la parte decimal. “E-6” indica que el resultado obtenido debe ser dividido para 1000000 o para  $10^6$ , es decir que:

$$0.00035124 = \frac{351.24}{1000000}$$

Si se tuviera “E+6” en vez de “E-6” el resultado se debería multiplicar por  $10^6$  en vez de dividirlo para esa cantidad. Si se trata de un número negativo, “-” al principio de la cadena, el resultado total se multiplica por  $-1$ , con lo que al final obtenemos

$$-0.00035124 = 0.00035124 \times (-1)$$

Este es el resultado final, con lo que la transformación ha sido concluida. Este resultado se almacena en una variable de tipo flotante. (El compilador MPLAB C18 soporta variables flotantes de 32 bits).

#### 4.4.1.5. Almacenamiento de datos en la memoria EEPROM

Cada vez que se recibe un dato nuevo, este se almacena en la memoria EEPROM. Para lograr esto, primero los datos a guardar que son flotantes, se deben pasar a cuatro registros de 8 bits puesto que las variables de tipo flotante son de 32 bits. Debido a la dificultad de manejar procesos de bajo nivel en un lenguaje de alto nivel como el C, se introdujo unas líneas en assembler para poder llevar a cabo esta tarea.

Primero se debe guardar el dato original en un espacio de memoria conocido, para esto se mapeó en la dirección 0x200 la variable flotante *mem*. Así, cada dato que deba ser guardado primero debe ser almacenado en *mem*. Una vez hecho esto, se ejecuta el siguiente código en assembler

```
MOVFF    0x200,    flo03
MOVFF    0x201,    flo02
MOVFF    0x202,    flo01
MOVFF    0x203,    flo00
```

Con esto los 4 bytes que componen la variable flotante se almacenan en *flo00*, *flo01*, *flo02* y *flo03*.

Luego se debe almacenar los valores de estos cuatro registros en la memoria EEPROM, para ello se llama a la función *grabar()*. Esta función emplea el algoritmo provisto en [16] para guardar datos en la memoria EEPROM.

#### 4.4.1.6. Lectura de datos de la memoria EEPROM

Cada vez que se enciende el controlador, este recupera los datos guardados en la memoria EEPROM y los asigna a sus respectivas variables. Como los datos son de tipo flotante de 32 bits, se guardan en cuatro espacios de memoria. Por ello, se debe primero recuperar los datos en cuatro registros de 8 bits para luego unirlos en un solo registro de 32 bits. Los datos son leídos y almacenados en las variables *flo00*, *flo01*, *flo02* y *flo03* empleando la función *leer()* que utiliza un algoritmo mostrado en [16] para la lectura de la memoria EEPROM. Una vez que se tiene los datos listos en los registros de 8 bits, se los debe unir en un solo registro flotante de 32 bits. Como en el ítem anterior, esto se lo hace en assembler, así se tiene:

```
MOVFF    flo03,    0x200
MOVFF    flo02,    0x201
MOVFF    flo01,    0x202
MOVFF    flo00,    0x203
```

La variable *mem* mapeada en la dirección 0x200 contiene de esta manera el valor flotante recuperado de la memoria EEPROM. Después de esto simplemente se asigna a la variable correspondiente el valor de *mem*.

#### 4.4.2. Lectura del sensor

La planta entrega a su salda una señal de 0-5V que refleja la temperatura que posee. Para leer este dato, simplemente se utiliza el módulo de conversión análogo-digital del microcontrolador. Los detalles de hardware y de las características del módulo se discuten en el capítulo 6, por lo que en este capítulo se asume que se tiene disponible la señal de 0-5V en la entrada del conversor A/D del microcontrolador.

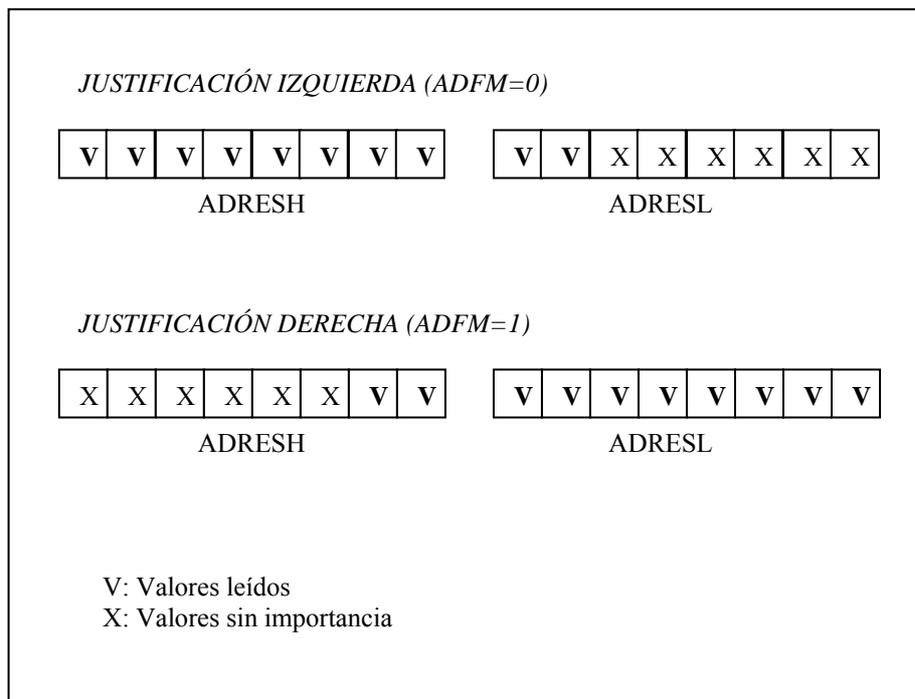
##### 4.4.2.1. Configuración del módulo A/D

El módulo de conversión A/D es manejado por 3 registros: ADCON0, ADCON1 y ADCON2. Con los bits <5:2> del registro ADCON0 se configura el canal a utilizar, en este caso el canal 0, por lo que dichos bits deben ser cargados con 0000. El bit <1> se llama

GO/DONE, poniendo este bit en 1 se inicia la conversión y cuando esta acaba vuelve automáticamente a 0. Escribiendo un 1 en el bit <0> se habilita el módulo A/D.

Con el registro ADCON1 se selecciona el voltaje de referencia mediante los bits <5:4>. Como en este caso la referencia es  $V_{DD}$  y  $V_{SS}$ , estos bits tienen el valor de 0. Con los bits <3:0> se configura las entradas del microcontrolador para hacerlas analógicas o digitales. Al utilizar el canal 0, se requiere que el puerto RA0 sea analógico, por lo que en estos bits se cargan los valores 1110.

En el bit <7> del registro ADCON2, llamado ADFM, se selecciona si se quiere justificación izquierda o derecha. El módulo A/D es de 10 bits y los valores leídos se guardan en los registros ADRESH y ADRESL; si ADFM es 1, entonces los dos bits más significativos del valor leído se guardan en los 2 bits de menos significativos de ADRESH y los 8 bits restantes en ADRESL (justificación derecha). Si por el contrario ADFM es 0, entonces los 8 bits más significativos del valor leído se guardan en ADRESH y los 2 bits restantes se guardan en los 2 bits más significativos de ADRESL (justificación izquierda). Esto se ilustra en la figura Figura.4.29



**Figura.4.29. Justificación izquierda y justificación derecha**

En el programa se escogió la justificación izquierda, por lo que ADFM toma el valor de 0.

Los bits <5:3> del registro ADCON2 controlan el tiempo de espera entre el seteo del bit GO/DONE y el inicio de la conversión. El módulo A/D del PIC18F2550 posee un capacitor llamado  $C_{HOLD}$  que debe cargarse completamente para que el valor leído sea correcto, por lo que se debe esperar que dicho capacitor se cargue completamente. Este tiempo de espera es llamado tiempo de adquisición. En [16] podemos encontrar que este tiempo se calcula de la siguiente manera:

$$T_{ADQ} = T_A + T_C + T_{COFF} \quad (4.1)$$

Donde  $T_A$  es el tiempo de establecimiento del amplificador y es igual a  $0.2\mu s$  para el PIC18F2550.  $T_C$  es el tiempo de carga de  $C_{HOLD}$ .  $T_{COFF}$  es el coeficiente de temperatura.

El cálculo realizado en [16] empleando la fórmula anterior da como resultado un tiempo de adquisición de  $6.4\mu s$ , asumiendo que el valor de  $C_{HOLD}$  es  $25pF$ , la resistencia de salida del sensor es  $2.5k\Omega$ , la resistencia de entrada del módulo A/D es  $2k\Omega$  y la temperatura es  $50^\circ$ . Como los valores anteriores son altos, para un funcionamiento en condiciones normales del controlador de este proyecto el tiempo de adquisición será menor.

El fabricante del microcontrolador, Microchip, sugiere que se espere el tiempo de adquisición después de cambiar de canal, después de una conversión y después de que el módulo A/D es encendido (poniendo un 1 en el bit <0> del registro ADCON0). Dado que en el software diseñado para este proyecto el módulo se habilita al encender el controlador, que no se realizan cambios de canal de entrada analógica, que se realiza una conversión en cada período de muestreo, y que dicho período de muestreo es mucho mayor que el tiempo de adquisición, los valores seleccionados en los bits <5:3> del registro ADCON2 son 0, con lo que solamente se espera una instrucción para iniciar la conversión después de haber seteado el bit GO/DONE.

Mediante los bits <2:0> del registro ADCON2 se selecciona el reloj de la conversión. En la tabla 21-1 de [16] se indica que valor se debe seleccionar de reloj según la frecuencia a la que funciona el microcontrolador. En este proyecto funciona a  $24MHz$ , y los valores de los bits <2:0> del registro ADCON2 escogidos son 010.

Resumiendo, los valores en ADCON0, ADCON1 y ADCON2 para el microcontrolador son:

$$ADCON0 = 0x01$$

```
ADCON1 = 0x0E
```

```
ADCON2 = 0x02
```

El puerto por donde se realiza la lectura debe ser configurado como entrada, por lo tanto el bit <0> del registro TRISA (el puerto RA0 es el canal 0 de la entrada analógica) debe tener el valor de 1.

Para habilitar la interrupción de A/D el valor del ADIE (bit <6> del registro PIE1) se pone en 1. De esta manera cuando concluye una conversión ocurre una interrupción y además el bit ADIF (bit <6> del registro PIR1) se pone en 1. El bit ADIF debe ser reseteado por software.

#### 4.4.2.2. Lectura del valor analógico

La lectura del valor analógico en la entrada del módulo A/D se debe dar en cada período de muestreo, por lo tanto en cada uno de ellos el bit GO/DONE debe ser seteado para iniciar una conversión.

Cuando la conversión termina, el bit GO/DONE vuelve a 0, además se produce una interrupción y el bit ADIF es seteado por el microcontrolador. En este momento se puede leer el valor analógico en los registros ADRESH y ADRESL; como la justificación izquierda fue seleccionada (como se puede revisar en 4.4.2.1), ADRESH tendrá los 8 bits más significativos del valor de entrada y en ADRESL estarán los dos bits restantes.

Para procesar el valor analógico leído, se debe poner en este en una variable del tipo *flotante*. El compilador MPLAB C18 admite variables de tipo flotante que son de 32bits, además de variables tipo *int* que son de 16 bits. Lo primero que se hace es mover los valores de ADRESH y ADRESL a un solo registro de 16 bits, en este caso llamado *vang*:

```
vang=ADRESH;
vang=(vang<<2)&0x03FC;
vang=vang/(ADRESL>>6)&0x03;
```

Una vez que se tiene los 10 bits de la conversión en un solo registro, dicho registro contendrá un valor en una escala de 0 y 1023 (0x000 y 0x3FF), para pasar a un valor en una escala de 0 a 5 que se utiliza en la ley de control, se realiza la siguiente operación:

```
yk=(vang*5.0/0x3FF);
```

De esta manera en  $y_k$ , que es un dato de tipo flotante, se tiene el valor de 0 a 5 leído en la entrada analógica.

En la siguiente figura se tiene un diagrama de flujo que explica el funcionamiento de la lectura del valor analógico.

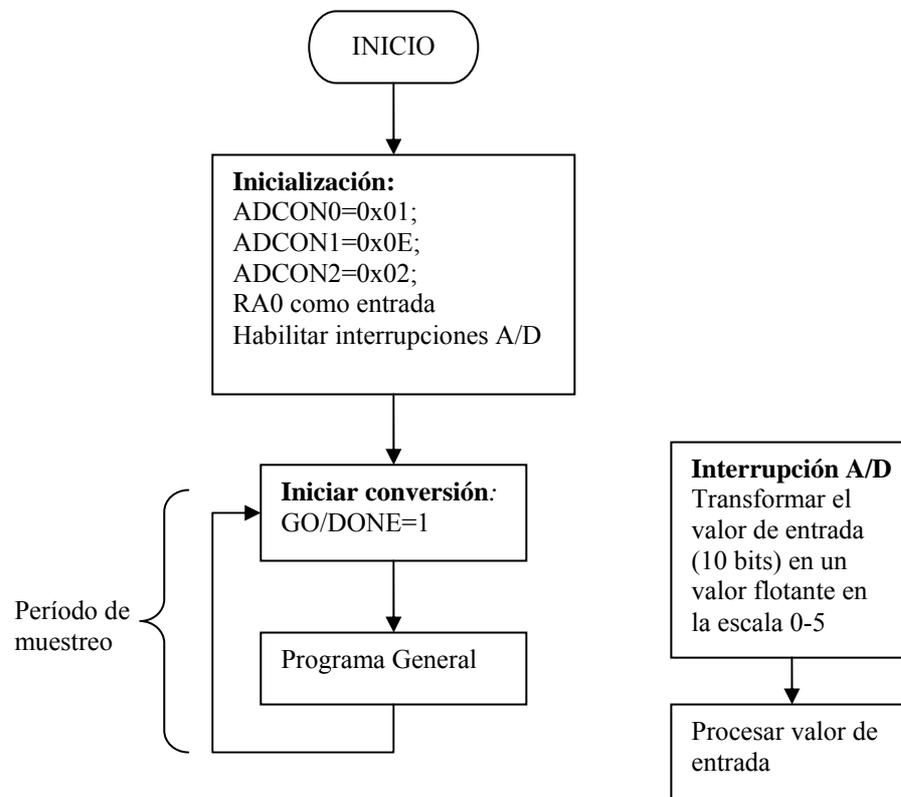


Figura.4.30. Diagrama de flujo de la lectura del valor del analógico

#### 4.4.3. Ley de control

La ley de control empleada es la siguiente:

$$u_k = u_{k1} + \left[ -Kp(y_k - y_{k1} + B(w_k - w_{k1})) + Ki \cdot T0 \cdot e_k - \left( \frac{Kd}{T0} \right) (y_k - 2y_{k1} + y_{k2}) \right] \cdot km$$

El subíndice  $kj$  representa los valores anteriores, por ejemplo  $u_{k1}$  es el valor que tenía  $u_k$  en el período de muestreo anterior. Las constantes  $Kp$ ,  $Ki$ ,  $Kd$ ,  $B$ ,  $T0$  y  $km$  son enviadas desde el computador, o son recuperadas de la memoria EEPROM del microcontrolador si es que no se conecta el PC.  $y_k$  es el valor de salida de la planta.

La ley de control debe ser llamada en cada período de muestreo. Para establecer el periodo de muestreo se utiliza el Timer0 del microcontrolador, de tal manera que cuando

haya transcurrido el tiempo requerido, se genera una interrupción. Cuando ocurre la interrupción se lee el valor analógico de la entrada (4.4.2) y se ejecuta la ley de control.

#### 4.4.3.1. Timer0

El Timer0 es un timer del PIC18F2550 de 8 o de 16 bits, que genera una interrupción cuando se desborda, es decir cuando pasa de 0xFF a 0x00 en el modo de 8 bits o de 0xFFFF a 0x0000 en el modo de 16 bits. Para configurarlo se emplea el registro T0CON, cargando este registro con un valor de 0x06 se configura el timer en modo de 16 bits, se elige el reloj interno ( $f_{osc}/4$ ) como fuente de reloj, y se elige un valor de pre-escalado de 1:128.

Como las interrupciones se generan cada vez que se desborda el registro del timer, y este registro se incrementa en cada ciclo del reloj ( $f_{osc}/4$ ), el período con que cada interrupción es generada es:

$$T = \frac{4 \cdot preescaler}{f_{osc}} (2^{16} - VAL)$$

Donde VAL es el valor de 16 bits formado por los registros TMR0H y TMR0L, por lo tanto para un período de 0.5s por ejemplo

$$VAL = 2^{16} - \frac{T \cdot f_{osc}}{4 \cdot preescaler}$$

$$VAL = 2^{16} - \frac{T \cdot 24e6}{4 \cdot 128}$$

$$VAL = 42098,5 = 0xA472$$

Por lo tanto para un período de 0.5s se debe cargar un valor de 0xA4 en TMR0H y 0x72 en TMR0L. Los valores de TMR0H y TMR0L deben actualizarse cuando se envía un nuevo período de muestreo desde el computador, para esto se emplea el siguiente algoritmo:

```
TM=0xFFFF-(T0*46875)+1;
TMH=TM>>8;
TML=TM;
```

En TMH (8 bits) se guarda el valor con el que debe cargarse TMR0H y en TML (8 bits) el valor para TMR0L. TM es un registro de 16 bits que contiene el valor completo.

#### 4.4.3.2. Filtro de Set Point

El filtro de Set Point, que se estudio en el capítulo 3, debe ser actualizado también en cada iteración del período de muestreo. El filtro emplea las constantes  $vA$  y  $vB$ , además del valor anterior del set point o  $w_{k1}$ , y del set point real o  $w_{k0}$ . El set point filtrado o  $w_k$  es igual a:

$$w_k = vA \cdot w_{k1} + vB \cdot w_{k0}$$

#### 4.4.3.3. Actualización de los valores

En cada iteración de la ley de control, se deben actualizar los valores, es decir que los valores actuales deben transformarse en los valores anteriores, para que en el siguiente período estén listos para su uso. Por lo tanto se tiene:

$$\begin{aligned} u_{k1} &= u_k \\ y_{k2} &= y_{k1} \\ y_{k1} &= y_k \\ w_{k1} &= w_k \end{aligned}$$

En la Figura.4.31 se presenta el diagrama de flujo de la ley de control.

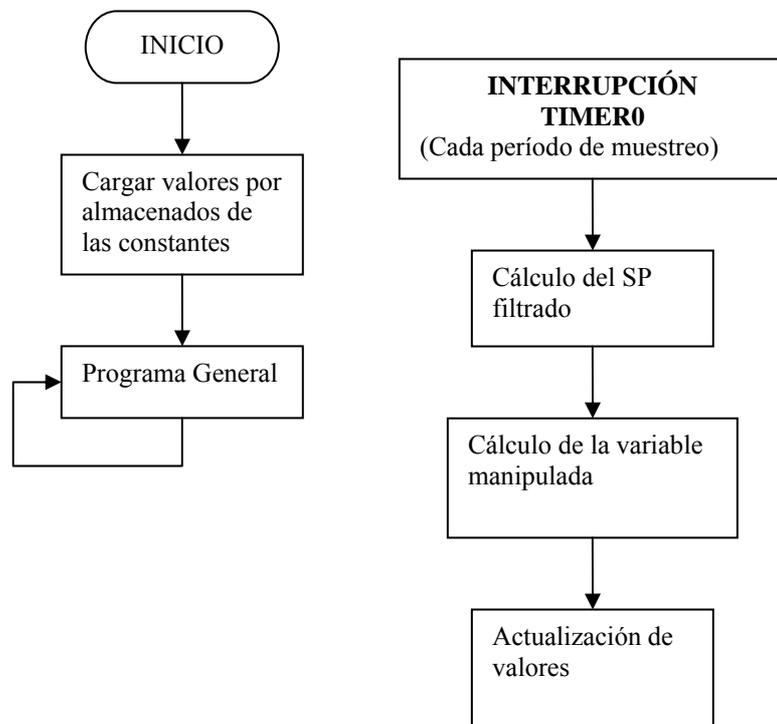


Figura.4.31. Diagrama de flujo de la ley de control

#### 4.4.4. Envío de la variable manipulada (transformación digital - análoga)

Una vez que la ley de control calcula la variable manipulada, se procede a su envío a la planta. Para esto se necesita una transformación digital – análoga (D/A). Aunque el PIC18F2550 no tiene un módulo A/D, posee un módulo PWM que puede ser usado con ese fin, mas detalles sobre este punto se revisarán en el capítulo 6, aquí solamente se estudiará la generación de la señal PWM necesaria para tener un nivel de voltaje de 0 a 5V en la salida PWM del microcontrolador.

El PIC18F2550 posee dos salidas PWM, en este caso se empleara el módulo 1. Este módulo es controlado por el registro CCP1CON. Los bits <5:4> llamados DC1B1 y DC1B0 contienen los bits menos significativos del valor del ciclo de trabajo del PWM, este punto se explicará con más detalle más adelante. Los bits <3:0> se utilizan para seleccionar el modo de operación, como lo que se requiere es el modo PWM los valores de estos bits deben ser 1100. Por lo tanto, el registro CCP1CON debe inicializarse con el valor 0x0C.

El período de PWM se determina por el Timer2 del microcontrolador, y es igual a [16]:

$$T_{PWM} = [PR2 + 1] \cdot 4 \cdot T_{OSC}(\text{preescalerTMR2})$$

PR2 es el registro donde se guarda el valor que se compara con el de que tiene TMR2, un período de PWM se completa cuando el valor de TMR2 es igual a PR2. Para lograr la resolución máxima, es decir 10 bits, el valor de PR2 debe ser igual a 255 o 0xFF, y para lograr las frecuencias más altas, el preescaler del Timer2 debe ser 1. Por lo tanto el registro T2CON que controla el Timer2 se carga con 0x00. Con esto el período de PWM es:

$$T_{PWM} = [255 + 1] \cdot 4 \cdot \frac{1}{24e6} (1)$$

$$T_{PWM} = 42,66e - 6$$

$$f_{PWM} = 23.44kHz$$

El ciclo de trabajo se calcula utilizando un valor de 10bits. Los 8 bits más significativos se encuentran en el registro CCPR1L, mientras que los 2 bits restantes lo forman los bits DC1B1 y DC1B0 del registro CCP1CON. El ciclo de trabajo del PWM viene dado por [16]:

$$PWM_{\%} = (CCPR1L : CCP1CON < 5 : 4 >) \cdot T_{OSC} \cdot (\text{preescalerTMR2})$$

Es decir que si el valor de 10 bits dado por  $CCPR1L:CCP1CON <5:4>$  se carga con 512 por ejemplo, el ciclo de trabajo será de  $21.33e-6$ , es decir del 50% con lo que en la salida se tendría un valor de 2.5V.

En vez de trabajar con 10 bits para el ciclo de trabajo, se pueden trabajar solamente con los 8 del registro  $CCPR1L$ , conservando los bits  $DC1B1$  y  $DC1B0$  en 0. De esta manera se simplifica el programa pues para cambiar el ciclo de trabajo solamente se necesita actualizar un registro en vez de los dos que se necesitan actualizar al trabajar con 10 bits. De este modo, si se quiere un ciclo del trabajo del 50% por ejemplo, simplemente hay que cargar 128 (0x80) el registro  $CCPR1L$ . Aunque se baja la resolución a 8 bits, ésta continúa dentro de los márgenes necesarios (6.3.2.3)

El puerto a través del cual se obtiene el PWM debe ser configurado como salida. En este caso corresponde al puerto  $RC2$ , por lo que el bit  $<2>$  del registro  $TRISC$  debe ser puesto en 0.

Resumiendo, para inicializar el PWM, se cargan los siguientes valores en los registros mostrados:

$$CCP1CON = 0x0C$$

$$PR2 = 0xFF$$

$$T2CON = 0x00$$

$$CCP1CON<5:4> = 0$$

$$TRISC<2> = 0$$

Como la ley de control entrega el valor de la variable manipulada en una escala de 0 a 5, este valor es transformado a una escala de 0 a 255 para que pueda ser cargado en  $CCPR1L$ . Primero se lo almacena en la variable tipo *char* (8 bits) llamada *valan*, eliminándose la parte decimal, mediante la siguiente instrucción:

$$valan = \frac{u_k}{5} \cdot 0xFF$$

Donde  $u_k$  es la variable manipulada. Luego se carga el valor de *valan* en  $CCPR1L$ .

#### 4.4.5. Diagrama de flujo general

A continuación se presenta un diagrama de flujo general de todo el software del microcontrolador. Las interrupciones del módulo A/D y del Timer0 no se presentan pues estas ya se mostraron en la Figura.4.30 y en la Figura.4.31., se debe recordar que es en esas interrupciones donde se realiza la lectura del valor de salida de la planta, se ejecuta la ley de control y se envía la variable manipulada.

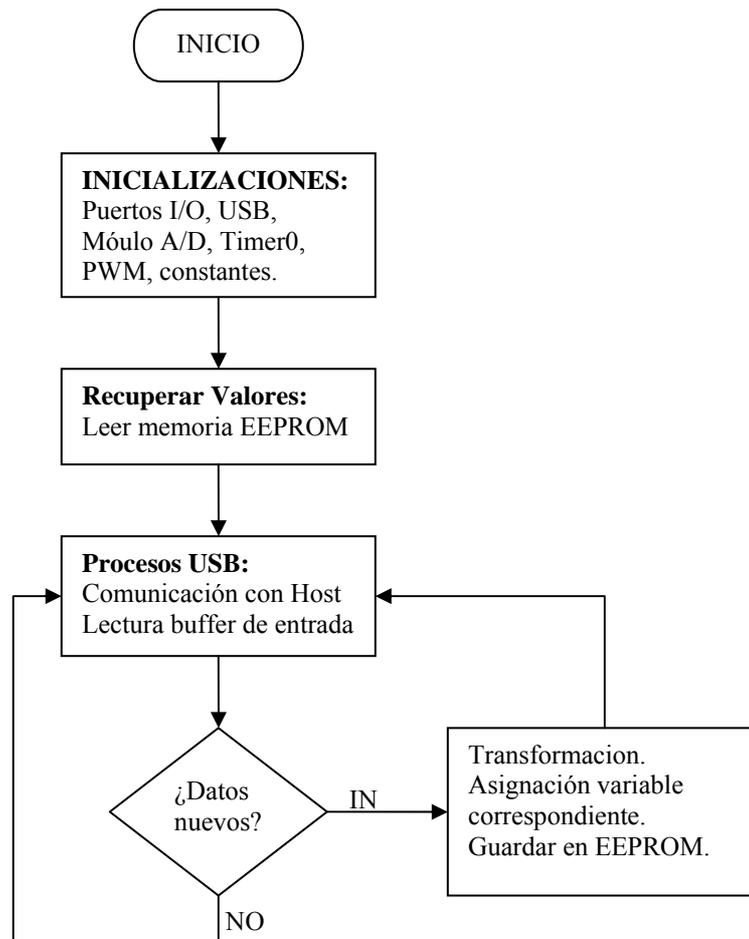


Figura.4.32. Diagrama General

## **CAPÍTULO 5**

### **SIMULACIÓN DEL SISTEMA**

#### **5.1. INTRODUCCIÓN**

Antes de realizar la implementación del sistema, se realizaron simulaciones del sistema completo utilizando Matlab, que permitieron determinar posibles fallos y poder conocer de antemano la manera como dicho sistema responderá. Las simulaciones fueron realizadas en Simulink. Los esquemas se encuentran en el Anexo 2.

Principalmente se trató de determinar si el sistema responde convenientemente, si existe una adecuada estimación de parámetros y además como se comporta el sistema frente al ruido. Para esto se simuló sobre un modelo que corresponde al comportamiento de la planta real sobre la que se trabaja en el presente proyecto.

Una condición importante para que el método de mínimos cuadrados funcione correctamente es que la señal de entrada de la planta sea de excitación persistente (PE). En este capítulo se muestra que en el sistema la señal que llega a la planta y que es usada en la identificación de la misma si es PE.

Es así que en el punto 5.2 de este capítulo se verán mediante las pruebas realizadas, el algoritmo de identificación y el sistema general, además de mostrar que la señal de entrada a la planta es PE. En el punto 5.3 se analiza la estabilidad del sistema.

#### **5.2. SIMULACIÓN DEL SISTEMA**

##### **5.2.1. Algoritmo de estimación de parámetros**

Primero se probará el algoritmo REFIL, que es el utilizado en la estimación de los parámetros del sistema. Las pruebas que se efectuaron fueron tanto en presencia como en ausencia de ruido, además se probará la efectividad del algoritmo cuando los parámetros son variables; es decir que las simulaciones son similares a las efectuadas en el capítulo 2, pero esta vez se trabajará sobre un modelo un poco más complejo.

Primero se define un sistema sobre el que se realizó las pruebas, que es el siguiente:

$$G(s) = \frac{0,04627}{s + 0,0461} \quad (5.1)$$

Se escoge este modelo debido a que es el sistema que representa el comportamiento de la planta real en la que se implementó el control desarrollado en la presente tesis. Discretizando el sistema con un período de muestreo de 0,5 segundos, (que es suficiente para un sistema lento como el estudiado) se obtiene:

$$G(z) = \frac{0,02287}{z - 0,9772}$$

O lo que es lo mismo:

$$G(z) = \frac{0,02287z^{-1}}{1 - 0,9772z^{-1}}$$

Es decir que aplicando el algoritmo REFIL al sistema dado, sus respuestas deberían ser:

$$\begin{aligned} b_0 &= 0 \\ a_1 &= 0,9772 \\ b_1 &= 0,02287 \end{aligned}$$

Para probar el comportamiento del algoritmo cuando los parámetros son variables, se hace variar el parámetro  $b_1$  de la siguiente manera:

$$b_1 = 0,02287 + 0,025 * \sin(0,01t)$$

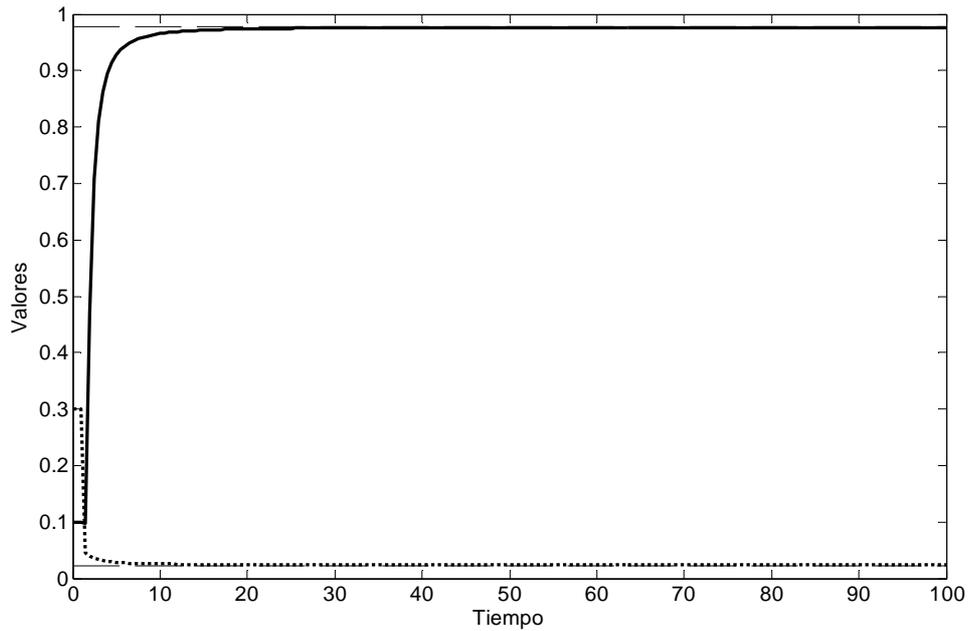
Para las pruebas con ruido se introduce ruido blanco utilizando el bloque llamado *band limited white noise* de Simulink con los siguientes parámetros:

$$\text{noise power}=[0.00001]$$

$$\text{sample time}=0.5$$

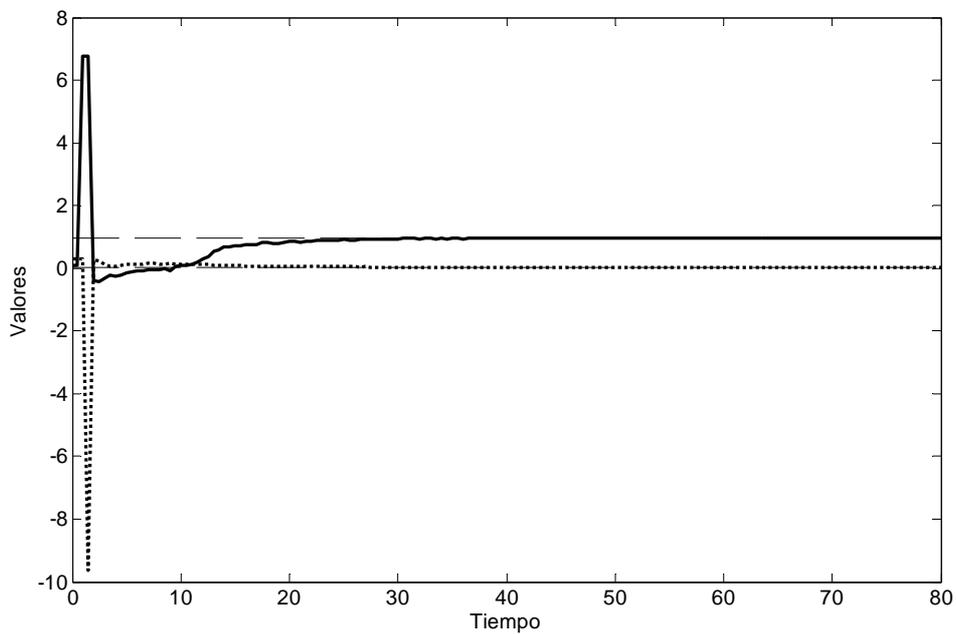
Para las simulaciones se ha graficado los valores reales con líneas cortadas, el valor de  $a_1$  en línea continua y el valor de  $b_1$  en línea punteada.

Al realizar la simulación con parámetros fijos y sin la presencia de ruido, se obtiene el siguiente resultado:



**Figura. 5.1. Estimación de parámetros REFIL (sin ruido)**

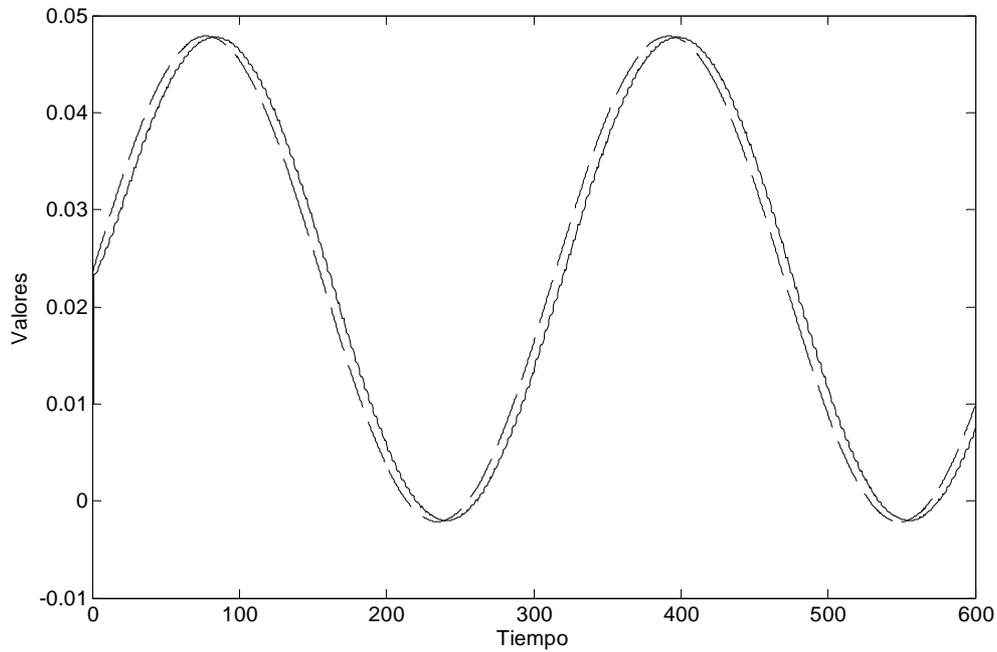
Cuando se introduce ruido, se tiene:



**Figura. 5.2. Estimación de parámetros REFIL (con ruido)**

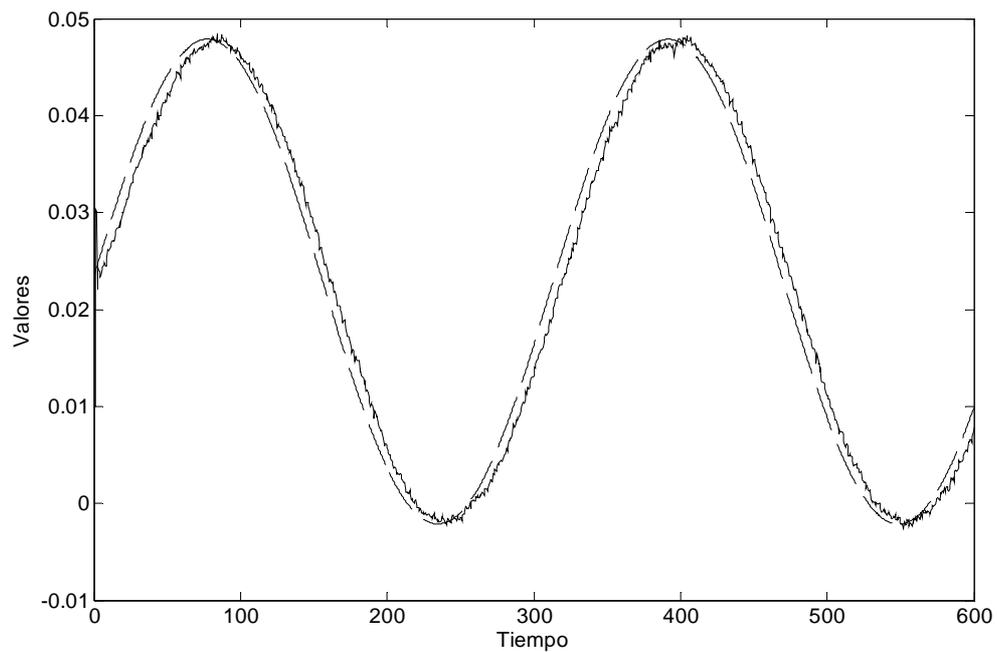
Se puede ver que el algoritmo es capaz de estimar adecuadamente los parámetros del sistema; Cuando existe ruido, a pesar de que la convergencia es más lenta, se logra alcanzar una estimación satisfactoria, aunque las estimaciones iniciales producen un error considerable.

En la gráfica de la simulación con parámetros variables se muestra el valor real (línea cortada) y el valor estimado de  $b_I$  (línea continua), que es el parámetro variable. El resultado obtenido es el siguiente cuando no existe ruido:



**Figura. 5.3.** Estimación de  $b_I=0,02287+0,025\text{sen}(0,01t)$  en ausencia de ruido

Con ruido el resultado es:



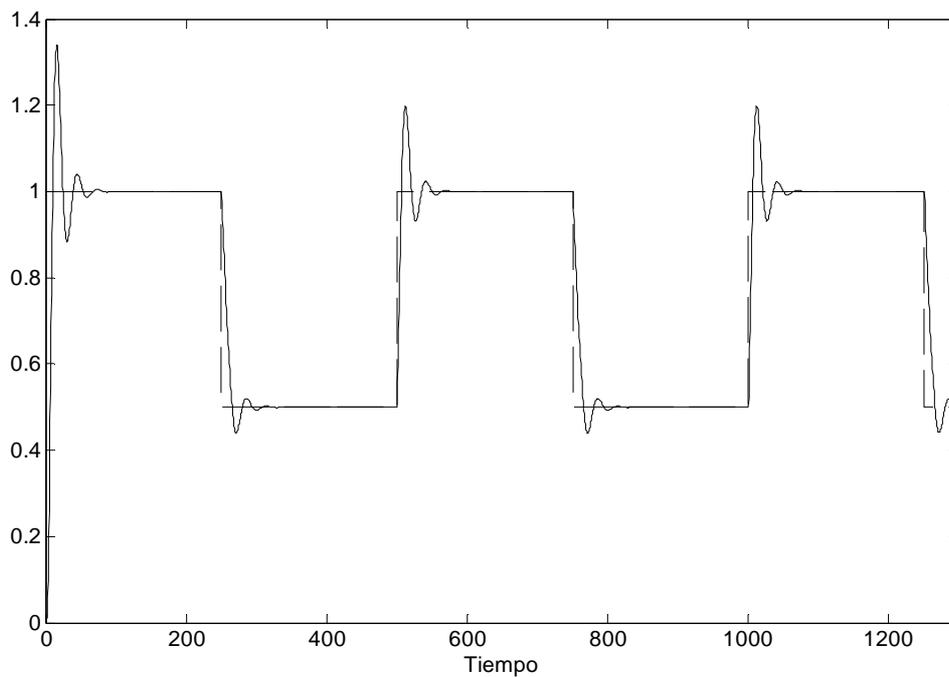
**Figura. 5.4.** Estimación de  $b_I=0,02287+0,025\text{sen}(0,01t)$  en presencia de ruido

Se puede ver que el algoritmo logra estimar el parámetro a pesar de que este varía, gracias a que el algoritmo utiliza la técnica del olvido exponencial.

### 5.2.2. Prueba del sistema

A continuación se procede con las simulaciones del sistema completo, es decir se simulará el esquema presentado en el capítulo 2, con una estimación de parámetros utilizando el algoritmo REFIL, y con asignación de polos como se revisó en el capítulo 3 para determinar los parámetros del controlador<sup>1</sup>. Para las pruebas se genera una señal de pulsos como entrada para el Set Point (SP) del sistema.

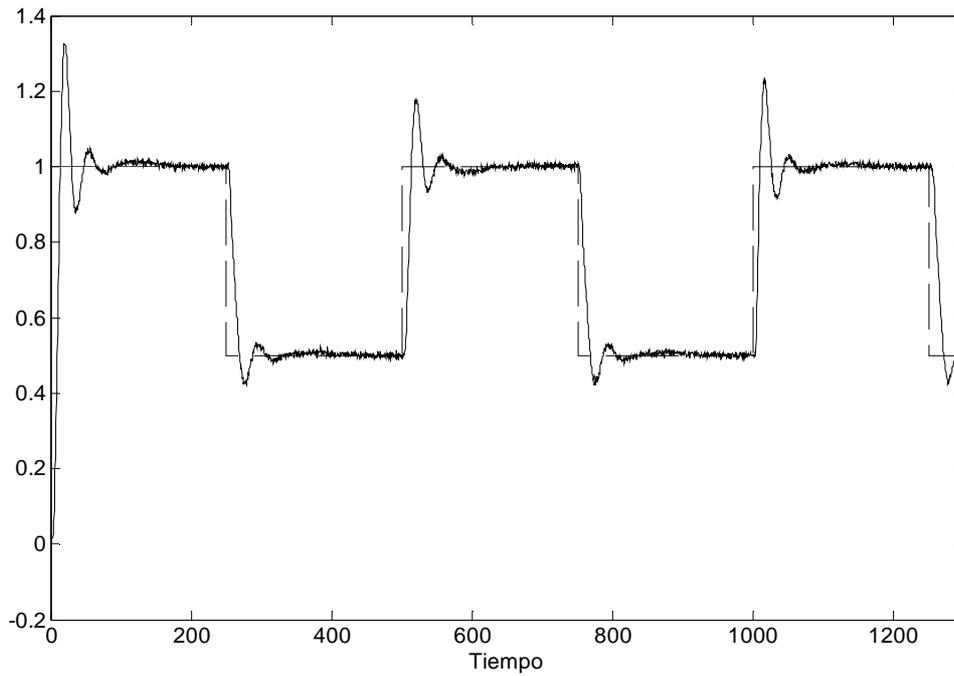
En un sistema sin filtro en el SP, con una ley de control tipo A se tienen las siguientes respuestas. El ITAE es 58,85



**Figura. 5.5. Control clase A, sin ruido, sin filtro de SP**

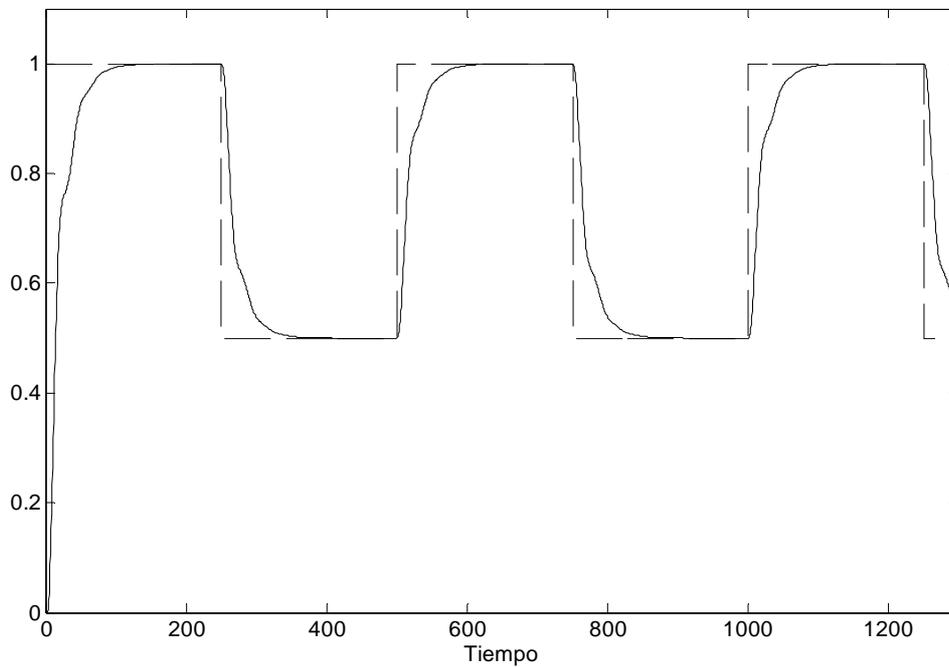
Con ruido se obtiene la siguiente respuesta:

<sup>1</sup> Todos los esquemas de simulación se encuentran en el Anexo I



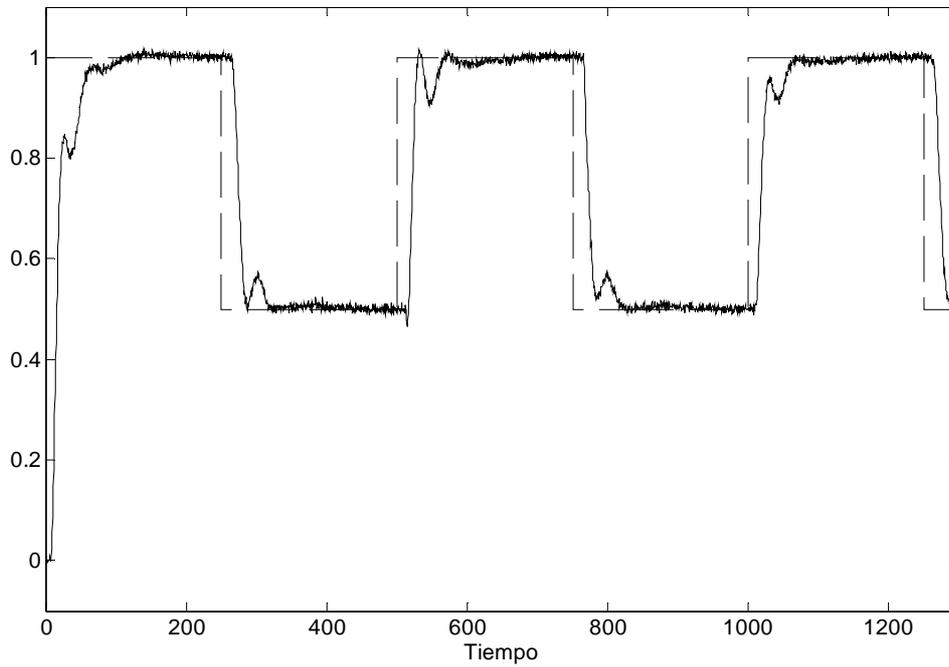
**Figura. 5.6. Control clase A, con ruido, sin filtro de SP**

Si se aplica un filtro en el SP, se obtiene un ITAE de: 199,8 y la salida es



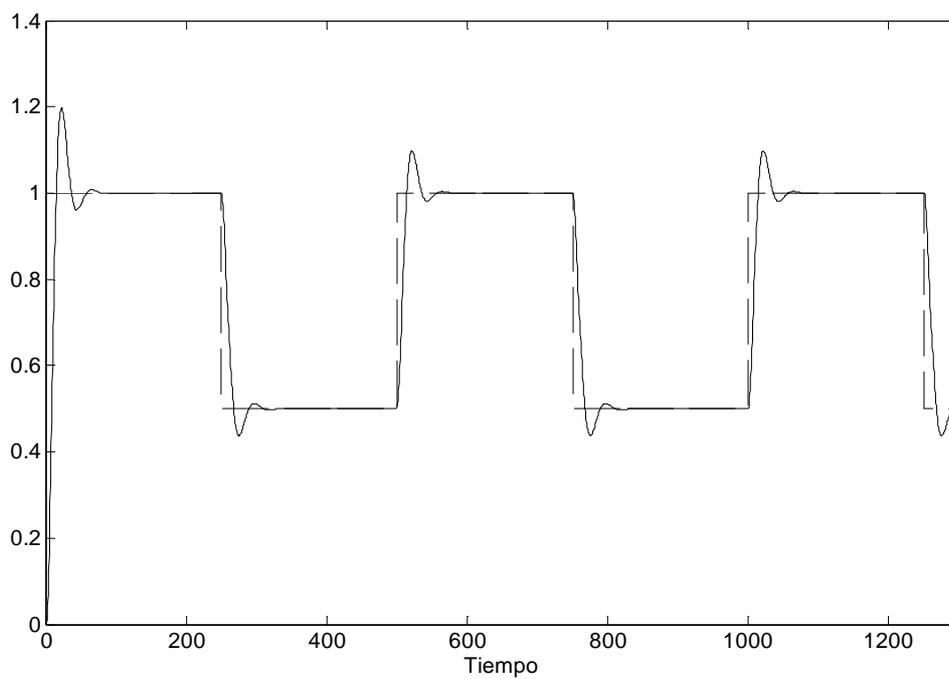
**Figura. 5.7. Clase A con filtro SP sin ruido**

Con ruido se obtiene



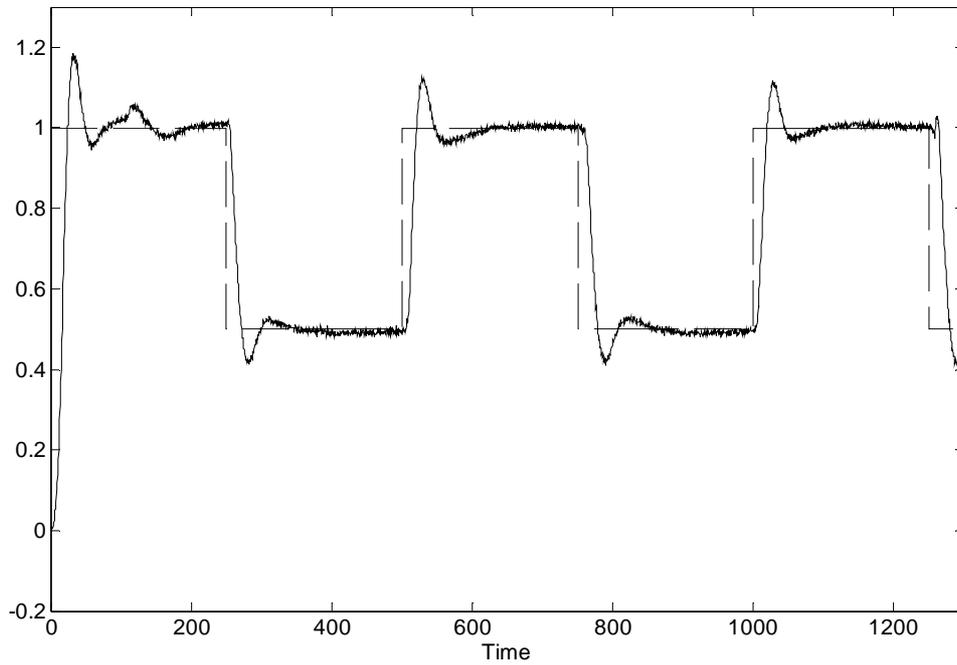
**Figura. 5.8. Control clase A, con filtro SP, con ruido**

Un sistema clase C sin filtro SP y sin SetPoint Weighting (B) tiene la siguiente salida, con un ITAE de 62,05



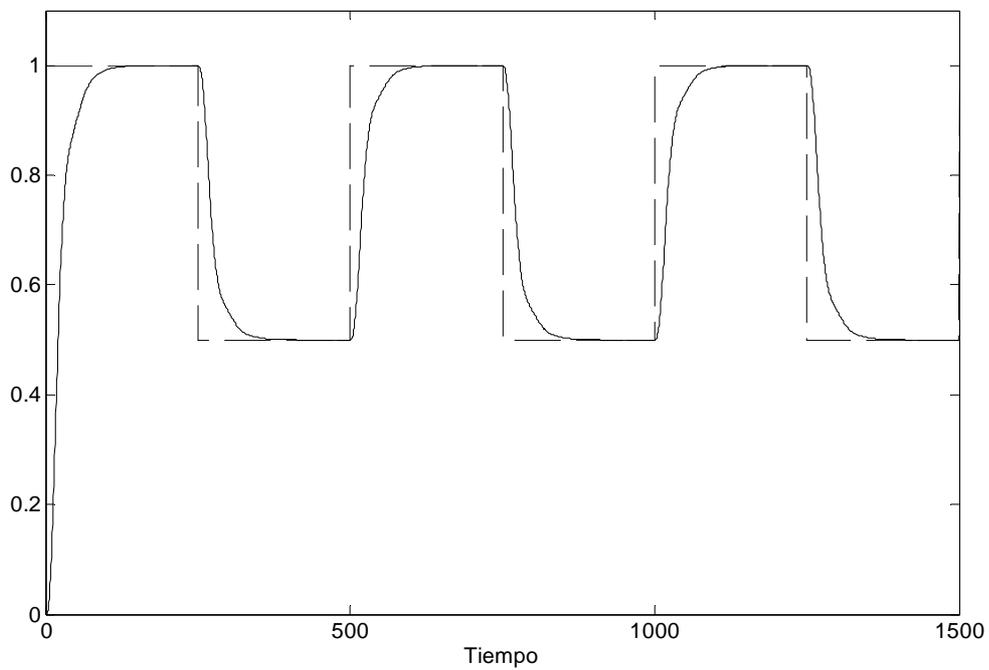
**Figura. 5.9. Control clase C, con B=0 y sin filtro de SP**

Con ruido se obtiene



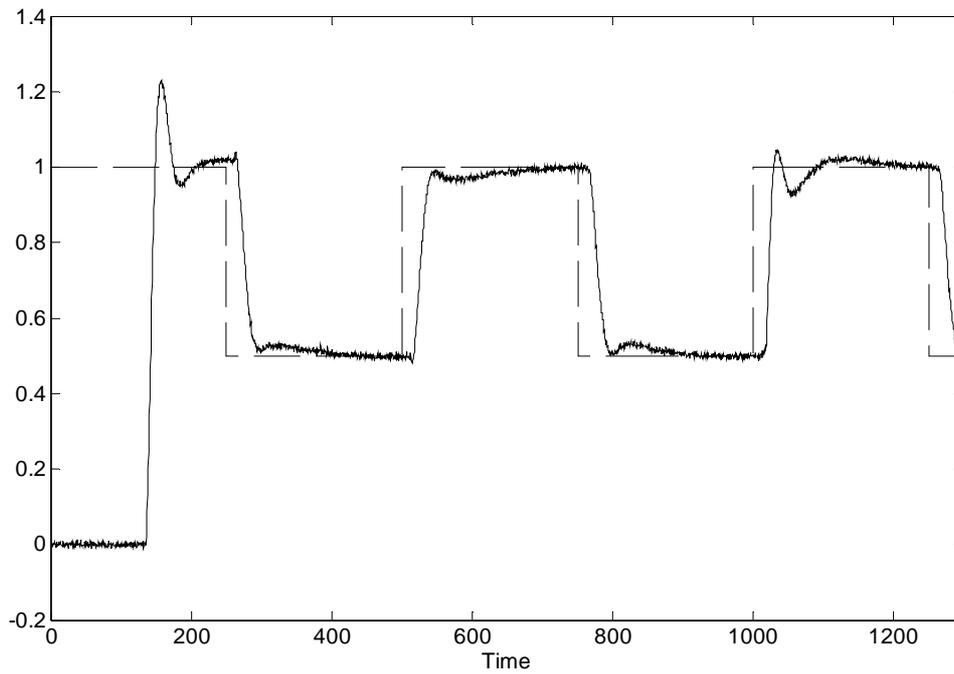
**Figura. 5.10. Control clase C, con  $B=0$ , sin filtro de SP y con ruido**

Ahora si se aplica el filtro a la entrada se tiene un ITAE de 252,7



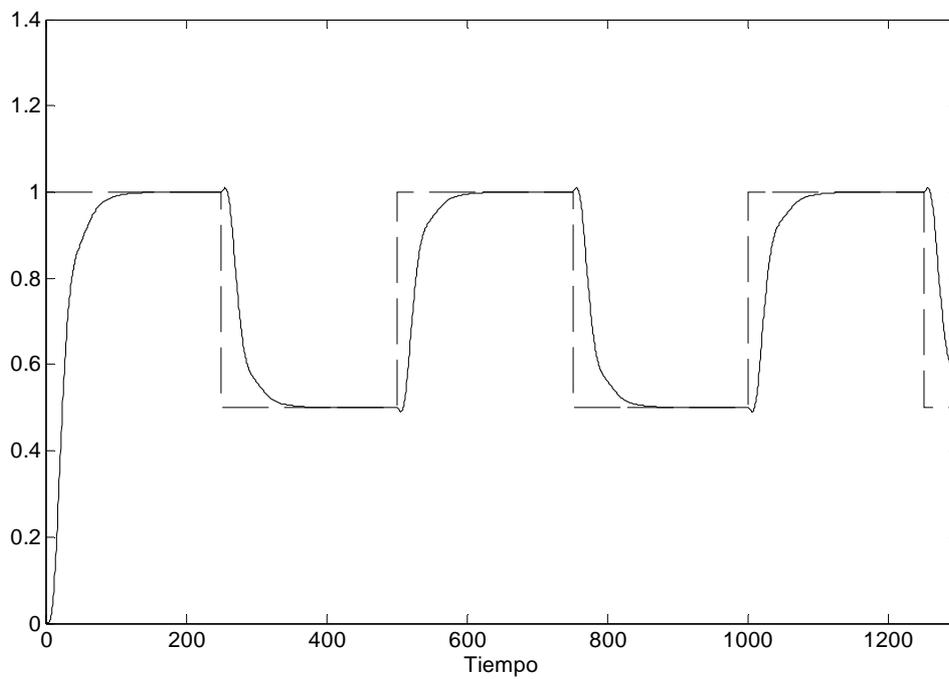
**Figura. 5.11. Sistema clase C con filtro de SP**

Y con ruido se tiene



**Figura. 5.12. Control clase C con filtro de SP y con ruido**

Si se aplica el SetPoint weighting (B), se tiene un ITAE de 301.9



**Figura. 5.13. Control clase C, con filtro de SP y con B=1**

En la Tabla. 5.1 se muestra el resumen de los resultados de las simulaciones realizadas

PRUEBA			
Clase	Filtro SP	B	ITAE
A	No	0	58,85
A	Si	0	199,8
C	No	0	62,05
C	Si	0	252,7
C	Si	1	301,9

Tabla. 5.1. Resumen de resultados

### 5.2.3. Prueba de la señal de entrada

En este punto se muestra que la señal de entrada de la planta es de excitación persistente (PE), que es una condición necesaria para que el método de mínimos cuadrados funcione correctamente.

Para esto se emplea la condición de señal PE, tomada de [1], que es la siguiente:

$$\bar{u} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^t u(k) \quad (5.2)$$

$$R_{ui} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^t [u(k) - \bar{u}] [u(k+i) - \bar{u}]$$

Si estos límites existen y además se cumple la condición de que la matriz D dada por

$$D = \{d_{ij} = R_{u(i-j)}\} \quad i, j=1, 2, 3, \dots$$

es definida positiva, entonces  $u(k)$  es una señal PE.

Empleando Simulink para calcular los valores de (5.2) en el sistema dado por (5.1), se obtiene los siguientes resultados<sup>1</sup>:

$$R_{u0} = 0,134$$

$$R_{u(-1)} = 0,132$$

<sup>1</sup> El esquema de simulación se encuentra en el Anexo J2

$$R_{u1} = 0,132$$

$$R_{u(-2)} = 0,130$$

$$R_{u2} = 0,130$$

$$R_{u(-3)} = 0,127$$

$$R_{u3} = 0,127$$

$$R_{u(-4)} = 0,124$$

$$R_{u4} = 0,124$$

Por lo tanto, la matriz  $D$  es:

$$D = \begin{bmatrix} 0,134 & 0,132 & 0,130 & 0,127 & 0,124 \\ 0,132 & 0,134 & 0,132 & 0,130 & 0,127 \\ 0,130 & 0,132 & 0,134 & 0,132 & 0,130 \\ 0,127 & 0,130 & 0,132 & 0,134 & 0,132 \\ 0,124 & 0,127 & 0,130 & 0,132 & 0,134 \end{bmatrix}$$

Una matriz simétrica es definida positiva si todos sus valores propios son mayores a cero ( $\lambda_i > 0$ ) [5]. Los valores propios de la matriz  $D$  son:

$$\lambda_1 = 0,652432$$

$$\lambda_2 = 0,012831$$

$$\lambda_3 = 0,002258$$

$$\lambda_4 = 0,00131$$

$$\lambda_5 = 0,001169$$

Y como  $D$  es simétrica, entonces  $D$  es definida positiva y por lo tanto  $u(k)$  es una señal PE.

### 5.3. ANÁLISIS DE ESTABILIDAD

Para el análisis de estabilidad del control adaptativo que se ha diseñado se empleará el diagrama Bode. Mediante ese análisis se determina el margen de fase y el margen de ganancia obtenidos. Si ambos son positivos, se concluye que la función es estable.

Para realizar el diagrama de Bode y determinar los márgenes de fase y de ganancia se empleó el toolbox SISOTOOL de MATLAB. La función de transferencia de la planta es la que se da en (5.1). Como el controlador es un PID, su función de transferencia es:

$$G_C = \frac{K_d s^2 + K_p s + K_i}{s}$$

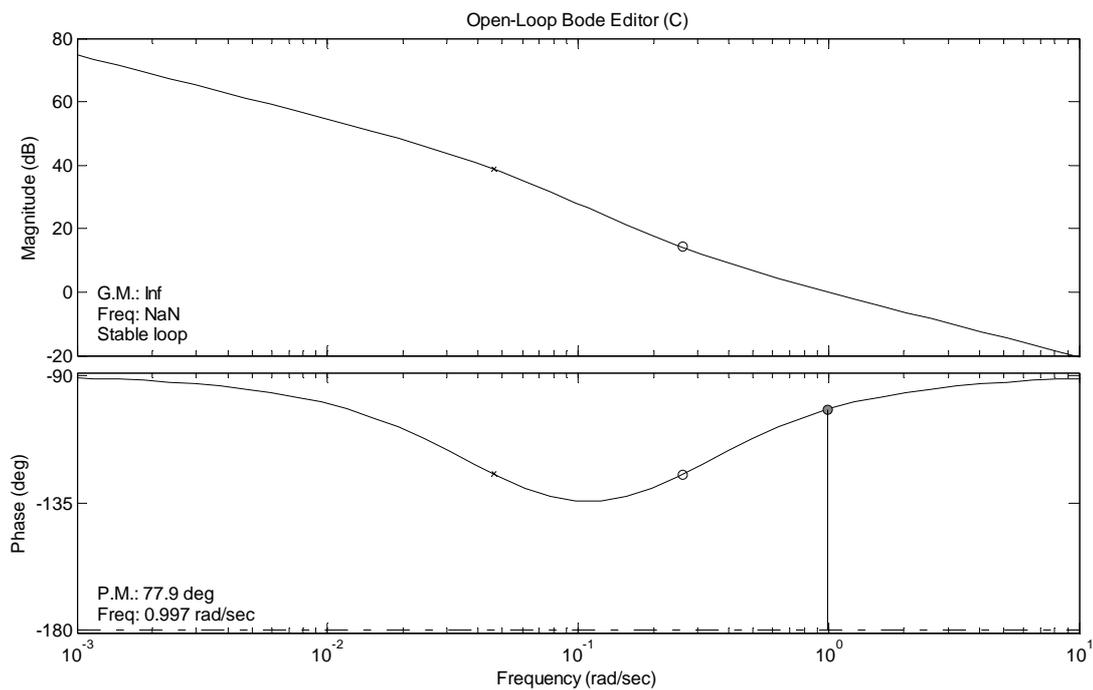
Los parámetros  $K_d$ ,  $K_p$  y  $K_i$  se determinaron realizando una simulación completa del sistema, es decir realizando la identificación de los parámetros del sistema dado por (5.1) y en base a estos resultados calculando las constantes del controlador. Los resultados fueron:

$$K_p = 20.87$$

$$K_i = 5.466$$

$$K_d = 0$$

El diagrama de Bode obtenido fue



**Figura. 5.14. Diagrama de Bode del sistema**

Se puede ver que tanto el margen de fase como el de ganancia son positivos, por lo tanto el sistema es estable.

## CAPÍTULO 6

### IMPLEMENTACIÓN DEL HARDWARE

#### 6.1. INTRODUCCIÓN

Una vez que el software desarrollado para el proyecto ha sido probado mediante simulación, el siguiente paso lógico es el desarrollo del hardware sobre el cual funcionará dicho software.

El hardware es en realidad un conjunto de elementos que conforman el sistema de control adaptativo. El objetivo del presente capítulo es explicar la función de cada una de estos elementos dentro del sistema. Es así que en el punto 6.2 se detalla el funcionamiento del sistema en su conjunto; en el punto 6.3 se explican los parámetros de diseño, donde se analizan las características del sistema para poder determinar los diferentes elementos que deben utilizarse; en el punto 6.4 se detallan las características de los elementos que son utilizados y en el punto 6.5 se tratan aspectos sobre la tarjeta diseñada para el controlador.

#### 6.2. GENERALIDADES

El hardware puede ser dividido en varios bloques, cada uno de los cuales cumple una función dentro del sistema, es así que se tiene:

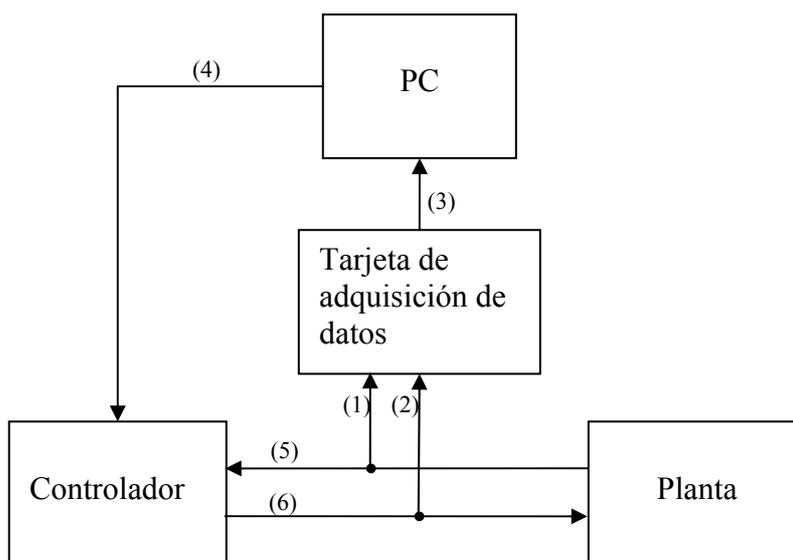


Figura. 6.1. Diagrama de bloques del Hardware

La PC es la encargada de realizar la estimación de los parámetros de la planta y en base a ello calcular las constantes de control; para esto necesita información de la entrada recibida por la planta y de la salida de la misma. Dicha información es adquirida a través de la tarjeta de adquisición de datos. En la Figura. 6.1 esto es representado por (1), (2) y (3). Una vez calculadas las constantes de control, las mismas deben ser enviadas al controlador, como se aprecia en (4) de la Figura. 6.1.

El controlador se encarga de determinar la señal que debe ser enviada a la planta (6), para esto necesita conocer la salida de la misma (5). Los parámetros del controlador son recibidos desde el computador (4).

### **6.2.1. Controlador**

Es aquí donde se ejecuta la ley de control. El controlador recibe la señal del sensor que indica la temperatura de la planta, procesa esa información y genera una señal que permite controlar el comportamiento de la misma. Además el computador envía al controlador los parámetros de la ley de control. Por lo tanto, el controlador debe ser capaz de leer y generar señales analógicas, además debe comunicarse con la PC.

### **6.2.2. Tarjeta de adquisición de datos**

Una de las funciones de la tarjeta de adquisición es leer y escribir datos análogos enviados desde un elemento externo hacia la computadora.

En este caso la tarjeta de adquisición cumple con la función de adquirir los datos enviados desde la salida de la planta y de la salida del controlador hacia la computadora. La información recolectada de la tarjeta de adquisición luego será empleada para realizar la identificación de la planta y las funciones de control como se explicó en el capítulo 4.

### **6.2.3. Computadora**

La computadora procesa la información enviada desde la tarjeta de adquisición y hacia el controlador, en este se encuentra el HMI que permite modificar y observar los parámetros a ser procesados por la planta mediante el software de LabView como se vio en el ítem 6.4. También este dispositivo contiene la interfaz de comunicación hacia el controlador a utilizar en este caso se realizará por medio del dispositivo USB que será detectado como un puerto serial virtual.

## 6.3. PARÁMETROS DE DISEÑO

### 6.3.1. Controlador

El controlador debe cumplir ciertos requisitos para poder funcionar correctamente dentro del sistema de control adaptativo diseñado. Las necesidades que el controlador debe ser capaz de cubrir son: comunicación con la PC, lectura de señales analógicas y salida de señales analógicas. Además debe cumplir ciertos requerimientos de capacidad de procesamiento.

#### 6.3.1.1. Comunicación con el computador

El controlador debe ser capaz de comunicarse con el computador de un modo eficaz. Para ello existen muchas opciones de comunicación, protocolos, interfaces y demás. Uno de los más populares y más utilizados en la comunicación PC-hardware es el RS-232. Aunque su uso industrial sigue siendo muy importante, su uso en otras áreas ha ido disminuyendo a tal punto que los últimos modelos de computador han dejado de incorporar esta interface, sobre todo en lo que se refiere a computadores portátiles. Por esta razón se escogió el bus USB como medio de comunicación entre el PC y el controlador. A pesar que el sistema es pensado para un medio industrial, se creyó conveniente utilizar USB debido a que prácticamente todas las computadoras modernas poseen una interface de este tipo, lo que no ocurre con el RS-232. Por lo tanto el controlador será capaz de manejar una interface USB.

##### 6.3.1.1.1. Características de la comunicación

En las pruebas realizadas, se determinó que se requiere una velocidad de 20 Kb/s de transmisión para que no existan problemas en la comunicación. Las distancias que se manejan no son grandes, una distancia de hasta 3 metros entre el computador y el controlador es suficiente. Se necesita solamente una comunicación punto a punto.

Parámetro	Valor	Unidad
Número de dispositivos	1	-
Velocidad	20	Kb/s
Distancia	3	metros

**Tabla. 6.1. Características de la comunicación**

En la Tabla. 6.1 se pueden ver las características mínimas requeridas para la comunicación

### 6.3.1.2. Lectura de las señales analógicas

El controlador debe ser capaz de leer señales analógicas. Como solamente se debe leer el sensor de temperatura de la planta, el controlador debe tener un canal de conversión análogo – digital (A/D).

Los períodos de muestreo son relativamente largos comparados con la velocidad de conversión de los dispositivos A/D, Por ello la velocidad de conversión del A/D no es un punto crítico.

La salida de la planta de temperatura varía entre 20 y 70 grados centígrados, que se traduce en una salida del sensor de 0 a 5V, con eso:

$$20 \leftrightarrow 70 \Rightarrow 0 \leftrightarrow 5$$

$$50 \Rightarrow 5$$

$$\Delta T = \Delta V \cdot 10$$

$$\Delta V = \frac{\Delta T}{10}$$

Es decir que por la variación de 1 grado en la temperatura se produce una variación de 0.1V en el sensor. Si para el control se admite un error máximo del  $\pm 2\%$  de la escala completa, quiere decir que a una determinada temperatura fija y controlada podrá existir una error máximo de  $\pm 1^\circ\text{C}$ , o lo que es lo mismo un error de  $\pm 0.1\text{V}$ .

El parámetro  $LSb$  indica el cambio que debe haber en la entrada analógica para que se produzca un cambio del código de salida del conversor.

$$LSb = \frac{Vref}{2^N}$$

Donde  $N$  indica la resolución del conversor y  $Vref$  es el voltaje de referencia. Si el conversor tiene una resolución de 8 bits, entonces:

$$LSb = \frac{5}{2^8} = 19,53 \times 10^{-3} \text{V}$$

Lo que quiere decir que un conversor de 8 bits producirá 256 valores, en una escala de 0 a 5V significa que cada valor tendrá un ancho de 19,53mV. Como este valor es menor que el error admisible, entonces el conversor debe tener una resolución de por lo menos 8 bits.

Una característica importante que debe cumplir el conversor A/D es que sea monótono, es decir que cuando la entrada aumente, el código de salida siempre aumente y viceversa. Además no debe haber códigos faltantes, es decir que si se recorre todo el rango de entrada, deben aparecer a la salida todos los códigos.

### **6.3.1.3. Salida de señales analógicas**

La variable manipulada calculada por la ley de control debe ser enviada a la entrada de la planta. Esta entrada admite señales de 0 a 5V, por lo que la salida del controlador deberá ser capaz de generar una señal en esa escala.

Para ello se necesita una conversión digital-analógica. Haciendo un análisis similar al hecho para calcular la resolución del conversor A/D y debido a que la salida del sensor de la planta también tiene una salida de 0 a 5V, podemos concluir que en este caso la resolución mínima también es 8 bits.

### **6.3.1.4. Características del procesador**

#### **6.3.1.4.1. Memoria de programa**

El programa desarrollado en el punto 4.5 una vez compilado tiene 5511 instrucciones. En un PIC cada instrucción ocupa 2 bytes por lo que se necesita 11022 bytes de espacio para el programa, es decir que se requiere un procesador que tenga por lo menos 12KB de memoria de programa.

#### **6.3.1.4.2. Memoria RAM**

Se necesitan 111 bytes de memoria RAM para el programa. Estos bytes corresponden a las 19 variables declaradas como *unsigned char* (1 byte), las 4 declaradas como *int* (2 bytes) y las 21 declaradas como *float* (4 bytes).

#### **6.3.1.4.3. Memoria EEPROM**

Los datos guardados en la memoria EEPROM corresponden a nueve variables de tipo flotante, cada una de ellas ocupa 4 bytes por lo tanto se necesita una memoria EEPROM de 36 bytes.

### 6.3.2. Tarjeta de adquisición de datos

La tarjeta de adquisición de datos debe tener entradas y salidas análogas con un rango de 0 a 5 Voltios, debe tener una precisión de lectura de datos de por lo menos 8 bits. También la instalación de la tarjeta debe poseer librerías para ser utilizadas por LabView, además debe tener posibilidad de comunicación hacia la PC ya sea por puerto serial o USB debido al muestreo en tiempo real.

## 6.4. ESPECIFICACIÓN DE LOS ELEMENTOS A UTILIZAR

### 6.4.1. Planta de Temperatura PCT-2

El objetivo principal de este proyecto de grado es diseñar e implementar un sistema de control adaptativo para una planta de temperatura.

La planta de temperatura consta de un sensor lineal tipo IC, además tiene un conducto de aire, el cual mediante un ventilador refresca el aire que se encuentra dentro de él. También en la entrada del conducto tiene una resistencia o niquelina que se encarga de calentar el aire que ingresa. De esta manera y gracias al ventilador se tiene diferentes temperaturas a lo largo del conducto de aire.

En su interior posee un amplificador de corriente debido a que la niquelina demanda mucha corriente para su funcionamiento. Al usar éste amplificador logramos que el módulo caliente únicamente con un voltaje de ingreso.

Para obtener la ecuación del sensor transductor es necesario tomar los datos del sensor tipo IC, en la Tabla. 6.2 se hace una relación del voltaje tomado del sensor tipo IC versus Temperatura.

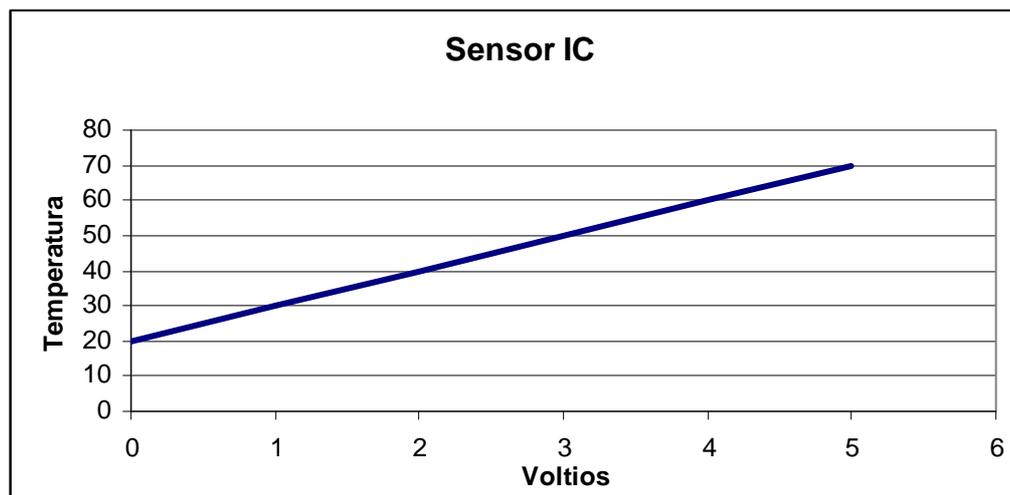
Voltaje de Salida ( Sensor IC)	Temperatura
0	20
1	30
2	40
3	50
4	60
5	70

Tabla. 6.2. Datos de Voltaje de salida vs. Temperatura para el sensor tipo IC del módulo PCT-2[4]

Por lo tanto la ecuación característica es la siguiente:

$$T = 10 * V + 20$$

La Figura. 6.2 muestra el comportamiento del sensor:



**Figura. 6.2. Respuesta del sensor tipo IC del módulo PCT-2**

## 6.4.2. Controlador

El procesador usado en el controlador es el PIC18F2550. Este microcontrolador cumple con todas las características descritas en 6.3.1. Dispone de una interface USB para la comunicación con la PC. Tiene un módulo de conversión A/D que es usado para la lectura de la señal analógica de la salida de la planta. Tiene además un módulo PWM que es usado para generar una señal analógica. Cabe recalcar que todos los datos que se presentan aquí son tomados de [16]

### 6.4.2.1. Características del procesador

El PIC18F2550 pertenece a la familia de microcontroladores PIC18 de la Microchip. Las principales características se describen en la Tabla. 6.3, donde además se compara con las características mínimas establecidas en 6.3.1

Característica	PIC18F2550	Mínimo requerido
Frecuencia de Operación	DC - 48MHz	
Memoria de Programa	32768 bytes	11022 bytes
	16384 instrucciones	5511 instrucciones
Memoria de Datos (RAM)	2048 bytes	111 bytes
Memoria EEPROM	256 bytes	36bytes
Puertos de I/O	21	2

Timers	4	1
Módulo PWM	2	1
Módulo USB	1	1
Convertor A/D de 10 bits	10 canales	1 canal
Set de instrucciones	75	

**Tabla. 6.3. Características del microcontrolador PIC18F2550**

Otras características importantes que describen al microprocesador se presentan en la Tabla. 6.4

<b>Característica</b>	<b>Valor</b>
Voltaje $V_{DD}$ respecto a $V_{SS}$	-0.3V hasta 7.5 V
Potencia total consumida	1W
Corriente máxima de salida de $V_{SS}$	300mA
Corriente máxima de entrada en $V_{DD}$	250mA
Máxima corriente de entrada o salida en un pin I/O	25mA
Máxima corriente de entrada o salida en todos los puertos	200mA
Voltaje de alimentación	3.0V hasta 5.5V

**Tabla. 6.4. Características PIC 18F2550**

#### **6.4.2.2. Características del módulo USB**

En la Tabla. 6.5 se presentan las principales características del protocolo USB, tomadas de [17].

Característica	
Velocidad	Low-speed: 1.5 Mb/s
	Full-speed: 12Mb/s
Tipo	Diferencial
Número de dispositivos	127
Tipo de transmisión	Control: Usada para la configuración
	Bulk: (Gran cantidad de datos con integridad de datos asegurada)
	Interrupt: (Tiempo de entrega e integridad de datos asegurado para pequeñas cantidades de datos)
	Isocrónica (gran cantidad de datos con tiempo de entrega asegurado: p.e. audio)
Topología	Estrella en niveles (tiered star)
Interface física	Eléctrica: Cable de cuatro conductores
	Mecánica: Conectores <i>upstream</i> y <i>downstream</i>
Protocolo de Bus	polled bus controlado por el Host  hosts-scheduled, token-based protocol.

**Tabla. 6.5. Características del protocolo USB**

Como la comunicación implementada simula un protocolo RS-232 sobre un protocolo USB, en la tabla se comparan las características mínimas requeridas (6.3.1.1.1) con las proporcionadas por el RS-232 simulado.

<b>Característica</b>	<b>RS-232 simulado</b>	<b>Mínimo requerido</b>
Número de dispositivos	1	1
Velocidad	Hasta 30Kb/s	20Kb/s
Distancia	5 metros	3 metros

**Tabla. 6.6. Características RS-232 simulado**

#### 6.4.2.3. Características de la conversión A/D

El PIC18F2550 tiene un módulo de conversión análoga digital con las características indicadas en la Tabla. 6.7, que se compara además con las características mínimas requeridas (6.3.1.2)

<b>Característica</b>	<b>Módulo A/D</b>	<b>Mínimo requerido</b>
Resolución	10 bits	8 bits
No-linealidad integral	$\leq \pm 1$ LSb	
Error de offset	$\leq \pm 1.5$ LSb	
Error de ganancia	$\leq \pm 1$ LSb	
Monotonicidad	Garantizada	Garantizada
Códigos faltantes	No	No

**Tabla. 6.7. Características conversor A/D**

Mediante el seguidor de voltaje se logra que la impedancia de salida del filtro sea muy baja, con esto se cumple la condición impuesta por el fabricante del controlador de que la impedancia que se conecta al módulo A/D sea menor o igual a 2.5k $\Omega$ .

#### 6.4.2.4. Características de la salida analógica

Como se vio en 6.3.1.3, la conversión análoga digital debe ser de por lo menos 8 bits. Aunque el microcontrolador PIC18F2550 no cuenta con un módulo de conversión D/A, cuenta con un módulo PWM que puede ser usado para el mismo fin. Dicho módulo tiene una resolución de 10 bits, pero como se vio en 4.5.4, se lo usará solamente con una

resolución de 8 bits que de todos modos cumple con el requisito mínimo estimado en 6.3.1.3.

La señal de PWM tiene frecuencia fija en la cual varía su ciclo de trabajo. Una desventaja de esta señal es que produce armónicos en los múltiplos de la frecuencia del PWM, es decir en  $2f_{PWM}$ ,  $3f_{PWM}$ ...

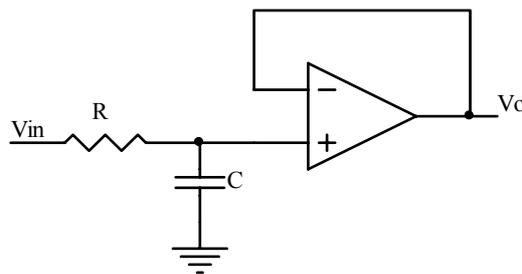
Para tener una salida DC, se debe filtrar la frecuencia de PWM y de esta manera también se eliminan los armónicos que se producen, para esto se pone un filtro pasa-bajo en la salida PWM, de tal manera que la frecuencia de corte del filtro sea menor que la frecuencia de PWM. Para este propósito un filtro R-C es suficiente, con lo que la frecuencia de corte debe ser unas 7 veces menor que la frecuencia de PWM.

$$f_c = \frac{f_{PWM}}{7}$$

La  $f_{PWM}$  es 23.44kHz, como se determinó en 4.5.4

$$f_c = \frac{23.44e3}{7}$$

$$f_c = 3.35kHz$$



**Figura. 6.3. Filtro R-C**

Usando el filtro de la Figura. 6.3, tenemos que:

$$RC = \frac{1}{2\pi f_c}$$

$$RC = \frac{1}{2\pi 3.35 \times 10^3}$$

$$RC = 47.53 \times 10^{-6}$$

Si se escoge un capacitor de 10nF, entonces:

$$R = \frac{47.53 \times 10^{-6}}{10 \times 10^{-9}}$$
$$R = 4.75 k\Omega$$

Por lo tanto se necesita un capacitor C de 10nF y una resistencia R de 4.7KΩ.

### 6.4.3. Tarjeta de adquisición USB PMD-1208LS

#### 6.4.3.1. Introducción

La tarjeta de adquisición de datos USB PMD-1208LS es un dispositivo de baja velocidad que es usado para adquisición de datos y control por medio del puerto USB 1.1, o USB 2.0. [6].

Este dispositivo es alimentado por el suministro de +5 Voltios del puerto USB, no requiere ninguna alimentación externa. La siguiente figura muestra la tarjeta USB PMD-1208LS, donde se observa claramente que contiene borneras donde se realizan las conexiones de entrada y salida del dispositivo.



**Figura. 6.4. Tarjeta USB PMD-1208LS**

Esta tarjeta es un dispositivo HID (Human Interface Device). Tiene las siguientes ventajas [3]:

- Un dispositivo USB HID utiliza drivers de clase Microsoft USB HID como interfaz, no requiere de ningún otro driver.
- La aplicación puede ser conectada por medio de un cable estándar de 4 hilos, este conector reemplaza a los conectores de los puertos seriales y paralelos con una combinación de conector y puerto estandarizado en uno solo.
- Es un dispositivo tipo plug-and-play.

- Se puede conectar la HID antes o después de instalar su software, y sin la necesidad de encender la computadora primero.
- Los datos pueden fluir en ambos sentidos entre una PC y periféricos bajo conexiones USB.

La instalación del hardware se presenta en el ANEXO 5.

#### 6.4.3.2. Detalles de funcionamiento de la PMD-1208LS

La PMD-1208LS puede adquirir los datos de las siguientes maneras:

- **Adquisición de datos a velocidad determinada por software:** En este modo la tarjeta realiza una sola adquisición o en grupos de únicas adquisiciones, la conversión es determinada por el software y el valor es enviado hacia la PC. La velocidad máxima cuando se trabaja en este modo es de 50 muestras por segundo.
- **Adquisición continua:** La adquisición se realiza ya sea por comando por software o por un evento disparado por hardware, por un solo canal o en una secuencia de canales, esta secuencia convierte, transfiere y almacena datos en un buffer FIFO hasta que la adquisición de datos sea detenida. La máxima tasa acumulada de adquisición de datos es de 1200 muestras por segundo por esta razón se puede adquirir en un canal de 1200 muestras, 2 canales de 600 muestras o 4 canales de 300 muestras.
- **Adquisición de datos forzada:** Lo realiza utilizando la capacidad de su buffer FIFO de 4000 muestras. Se realiza una única adquisición de una, dos o cuatro canales y los datos capturados después son leídos desde el buffer y enviados hacia la PC. La capacidad de memoria es limitada por la velocidad de transferencia de datos. Las tasas de transmisión son 8000, 4000 y 2000 muestras por segundo.

#### 6.4.3.3. Descripción de los pines de la tarjeta

La tarjeta tiene dos bancos de borneras una en la parte superior y otra en la parte inferior como se muestra en la siguiente figura.

Los pines del 1 al 20 tienen las siguientes características:

- Ocho entradas analógicas (CH0 IN a CH7 IN)
- Dos salidas analógicas (D/A OUT 0 a D/A OUT 1)

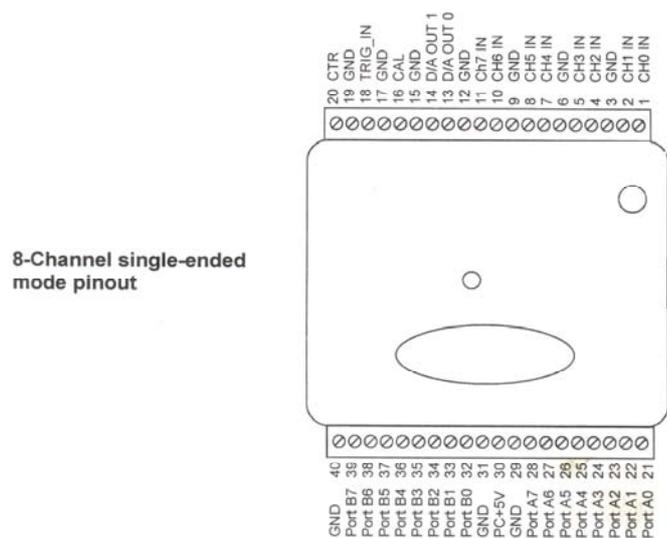
- Un disipador externo (TRIG\_IN)
- Un contador de eventos externos (CTR)
- Siete pines de tierra
- Un pin de calibración



**Figura. 6.5. Numeración de pines en la PMD-1208LS**

Los pines del 21 al 40 tienen las siguientes características:

- 16 entradas/salidas digitales (Port A0 a Port A7, y Port B0 a Port B7)
- Un pin de Vcc (PC +5V)
- Tres pines de tierra



**Figura. 6.6. Diagrama de pines**

La Figura. 6.6 enseña más detalladamente el diagrama de pines.

Debido a que las especificaciones de este proyecto se basan en la adquisición de datos análogos, este capítulo solo se refiere a este punto en particular, para más información puede referirse a [6] o [3].

#### **6.4.3.3.1. Pines de entradas analógicas (CH0 IN – CH7 IN)**

Se pueden usar 8 entradas análogas, hay dos tipos de configuración 8 tipo de entradas individuales o 4 entradas diferenciales.

- **Configuración Individual:** En este tipo de configuración hay 8 entradas disponibles, tiene como referencia el valor de tierra. Para cada entrada análoga se necesita un valor de tierra, el rango de entrada es de  $\pm 10$  V.
- **Configuración diferencial:** Cuatro entradas están disponibles, esta medida es comparada con respecto al valor ingresado en el pin Low input. La señal es entregada a través de tres cables. Un cable a tierra, el otro a la entrada alta y el otro a la entrada baja. Los rangos de voltajes son de  $\pm 20$  V,  $\pm 10$ V,  $\pm 5$  V,  $\pm 2.5$ V,  $\pm 2$ ,  $\pm 1.25$ V y  $\pm 1$ V. La entrada debe mantenerse en el rango de -10V a +20V para obtener los resultados deseados.

Las características técnicas de la tarjeta PMD-1208LS se presentan en el ANEXO 6.

### **6.5. DISEÑO DE LA TARJETA**

#### **6.5.1. Generalidades de la tarjeta**

Tomando en cuenta lo expuesto en 6.3.1 y en 6.4.2, la tarjeta diseñada debe tener las siguientes características:

- Conexión USB
- Entrada analógica de 0 a 5V
- Salida analógica de 0 a 5V

Se habilitaron además para diversos propósitos 4 puertos de salidas digitales, cuatro entradas digitales (conectadas a un pulsante) y se conectó las salidas RA1 hasta RA4 a una bornera.

#### **6.5.2. Microcontrolador**

La Figura. 6.7 presenta el diagrama esquemático general.

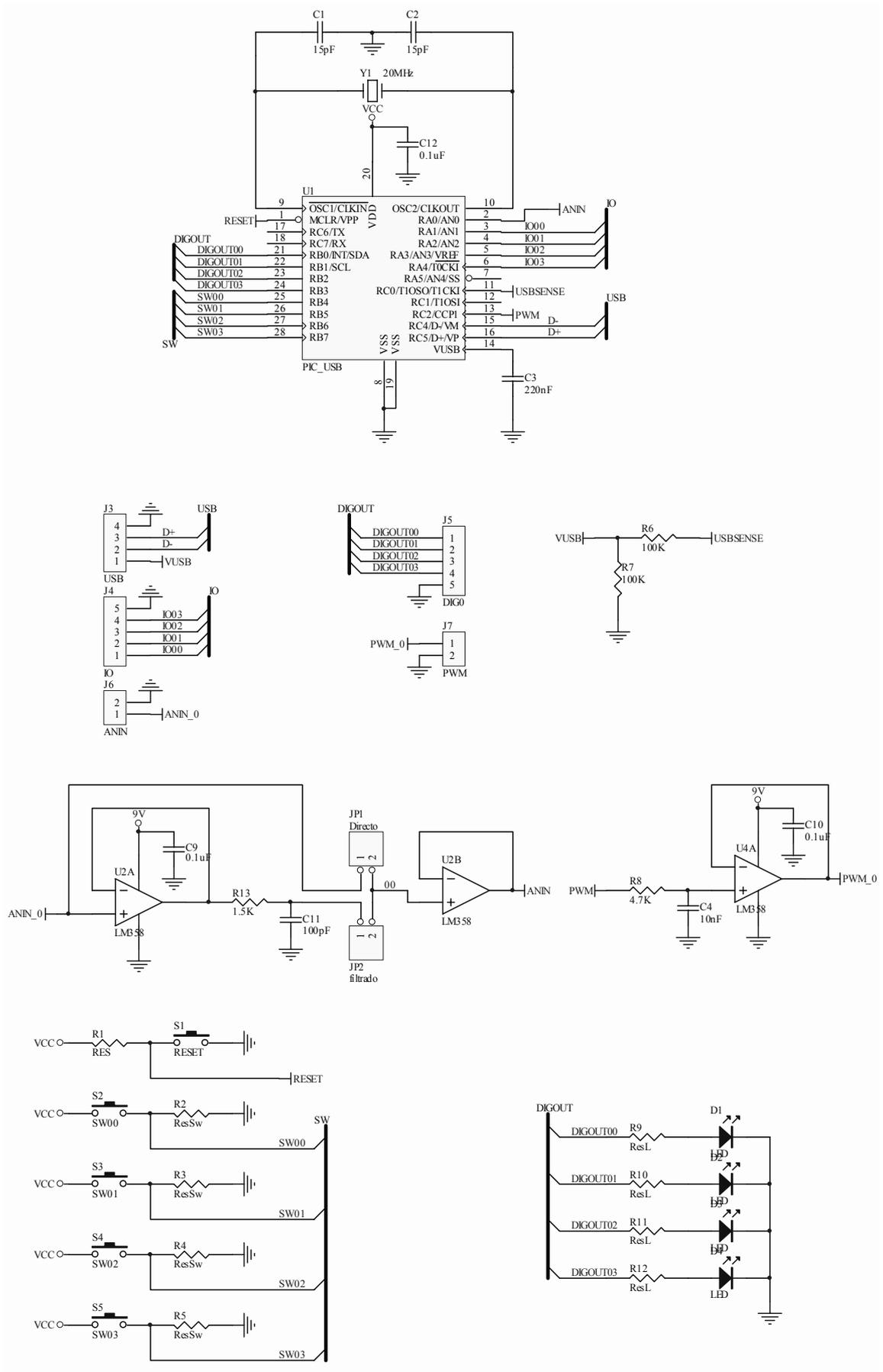
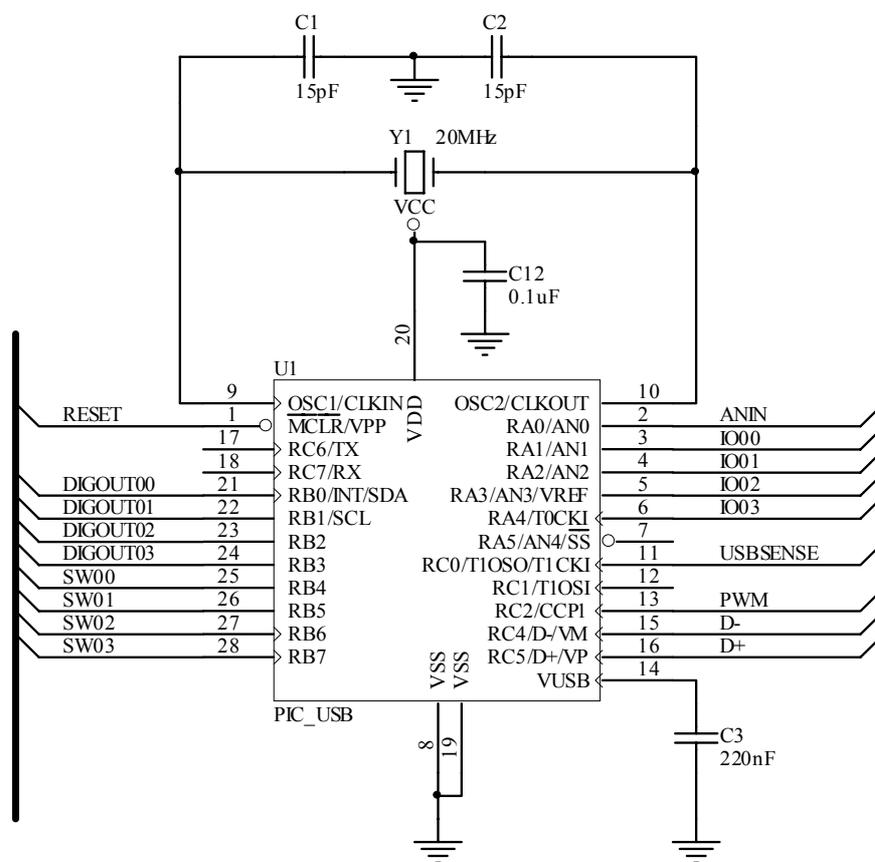


Figura. 6.7. Diagrama esquemático general

El esquemático de la parte del microcontrolador es el de la Figura. 6.8



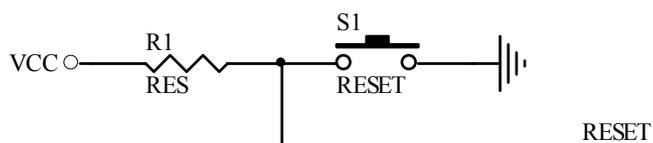
**Figura. 6.8. Esquemático del microcontrolador**

La entrada de la señal analógica es ANIN, conectada en el puerto RA0. La salida de PWM está conectada en el puerto RC2. D+ y D- son las señales diferenciales para la comunicación USB. La entrada USBSENSE en RC0 está conectada a  $V_{USB}$  del bus USB, y sirve para detectar si el dispositivo está conectado al host USB.

Las salidas digitales son DIGOUT00 hasta DIGOUT03 que están conectadas a la parte baja del puerto B. Las señales de los pulsantes son SW00 hasta SW03 y están conectadas a la parte alta del puerto B. IO00 hasta IO03 no están habilitadas en software, pero físicamente están conectadas a una bornera.

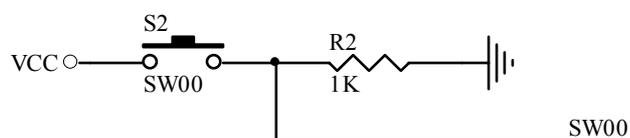
Se observa que se han colocado dos capacitores de 15pF entre las entradas del oscilador y tierra, esto se lo hace por recomendación del fabricante. El capacitor C3 de 220nF colocado en el pin  $V_{USB}$  es requerido para dar estabilidad. El capacitor C12 es un capacitor de bypass, más información sobre esto se encuentra en 6.5.8.1.

Se habilitó además un pulsante de RESET, como se ve en la Figura. 6.9



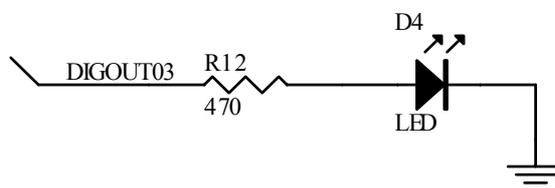
**Figura. 6.9. Pulsante de reset**

El esquema de entradas de los pulsantes es el de la Figura. 6.10 y es el mismo para los cuatro pulsantes implementados. Se puede ver que en estado de reposo, la señal es baja en la entrada del microcontrolador.



**Figura. 6.10. Señal de pulsante**

Para las salidas digitales se implementó leds indicadores, esto se puede ver en la Figura. 6.11, que es la misma para las cuatro salidas.

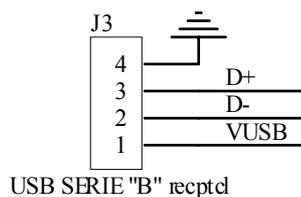


**Figura. 6.11. Led indicador de salida digital**

Las señales DIGOUT00 hasta DIGOUT03 están conectadas a una bornera de salida.

### 6.5.3. Conexión USB

La conexión USB se realiza a través de un receptáculo USB Serie "B", como lo indica la norma USB 2.0. A través del puerto RC0 se verifica si el dispositivo está o no conectado al host. De esta manera, la interface USB es la indicada en las figuras



**Figura. 6.12. Entrada USB**

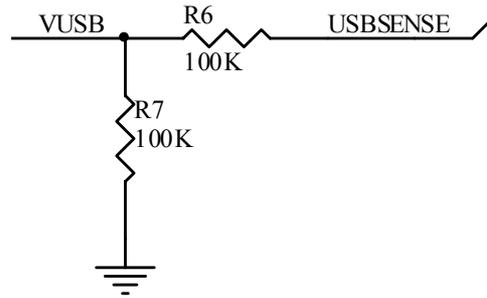


Figura. 6.13. Sensor de conexión con el host

#### 6.5.4. Entrada analógica

La entrada analógica está conectada en el canal 0 del módulo de conversión A/D del microcontrolador, es decir en el puerto RA0. Se implementó un filtro en la entrada A/D, el mismo que puede ser o no usado a consideración del operador si es que esta tarjeta se utiliza para trabajar con velocidades más altas. Esta selección se realiza a través de los jumpers JP1 y JP2 como se observa en la Figura. 6.14. Al seleccionar JP2 la señal es filtrada, mientras que con JP1 pasa directamente al módulo A/D.

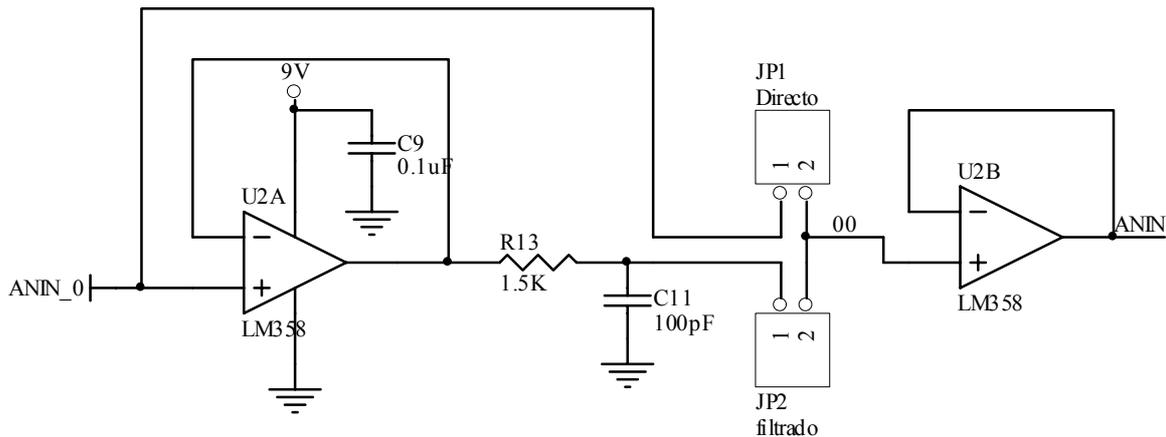


Figura. 6.14. Entrada analógica

Mediante U2A se implementa un seguidor de voltaje para que no exista problemas de acople de impedancias entre el sensor de la planta y el filtro. De igual manera U2B está configurado como un seguidor de voltaje para acoplar las impedancias del filtro (o del sensor si no se usa el filtro) con el módulo A/D. La señal ANIN\_0 está conectada en una bornera de entrada.

El capacitor C9 es un capacitor de bypass, más información sobre esto se encuentre en 6.5.8.1.

### 6.5.5. Salida analógica

La salida analógica se implementa usando el módulo PWM del microcontrolador y filtrando esa señal, como se vio en 6.4.2. El esquema de dicha salida se presenta en la Figura. 6.15

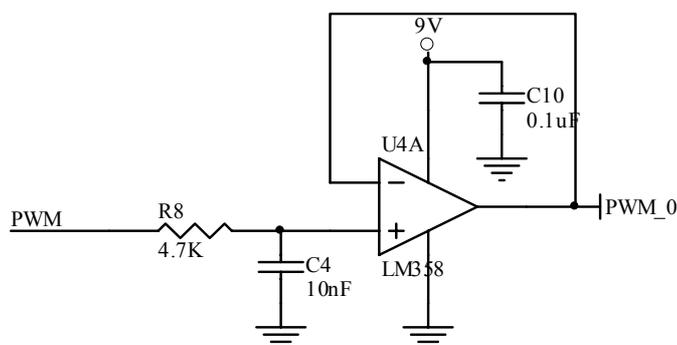


Figura. 6.15. Salida analógica

La señal PWM\_0 está conectada en una bornera de salida. C10 es un capacitor de bypass (revisar 6.5.8.1)

### 6.5.6. Fuente de poder

Característica	LM7805	LM7805
Corriente de Salida	1 A (sin disipador de calor)	1A (sin disipador de calor)
Voltaje de entrada máximo	35V	35V
Rango de temperatura de operación	0-125° C	0-125° C
Voltaje de salida	5V	9V
Impedancia de salida	15mΩ	17mΩ
Corriente pico de salida	2.2A	2.2A

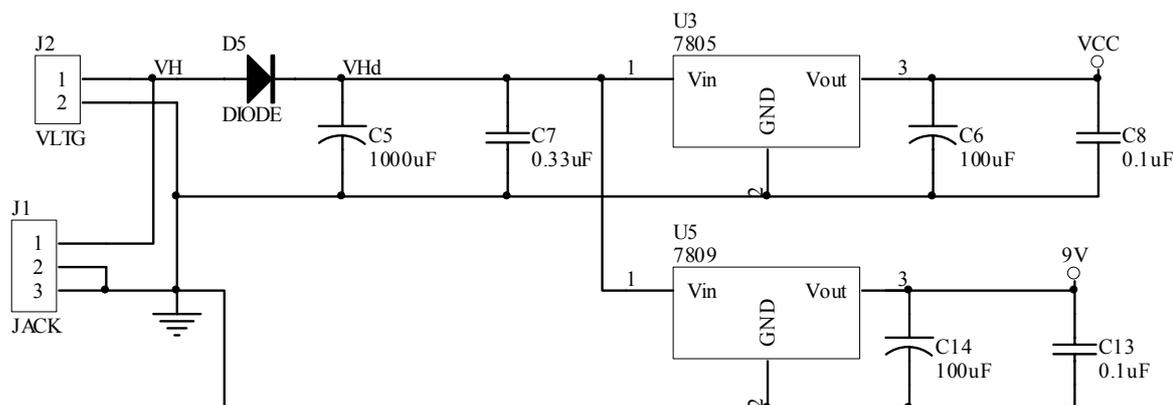
Tabla. 6.8. Características LM7805 y LM7809

Debido a que los amplificadores operacionales requieren un voltaje de por lo menos 7V<sup>1</sup> para funcionar adecuadamente y el resto del circuito necesita 5V, se decidió diseñar una

<sup>1</sup> Los detalles de las características de los amplificadores se encuentran en 6.5.7

fuente para proveer 9V y 5V. Para esto se utilizó los reguladores LM7805 y LM7809, cuyas características principales se detallan en la Tabla. 6.8<sup>1</sup>

Una ventaja adicional que se obtiene al usar los reguladores es que se elimina el ruido de la fuente producido cuando esta es una fuente conmutada. El esquema implementado es el de la Figura. 6.16.



**Figura. 6.16. Fuente de poder**

Se habilitaron dos tipos de entradas, una con JACK y otra con bornera. El diodo D5 evita daños al circuito si es que por error se polariza inversamente el mismo. Los capacitores de entrada eliminan ruidos de la fuente y son recomendados por el fabricante del regulador al igual que el capacitor de 0.1uF en la salida. El capacitor C14 es un capacitor de bypass (6.5.8.1)

Cabe recalcar que el voltaje de entrada se debe mantener por lo menos 2V sobre el valor de la salida.

### 6.5.7. Amplificadores Operacionales

El amplificador operacional escogido para implementar los filtros y los seguidores de voltaje del circuito es el LM358. La principal razón por la que se usa este operacional es que puede funcionar con una sola fuente, lo que elimina la necesidad de una fuente dual. Las principales características del LM358 se describen en la Tabla. 6.9 [19]

Cuando se alimenta con una sola fuente, la salida puede variar entre 0 y  $V^+ - 1.5$ , por esta razón en el circuito diseñado los amplificadores se alimentan con una fuente de 9V, lo que permite un rango de salida de 0 a 7.5V. Para el controlador se necesitan señales en el rango de 0 a 5V, con lo que dicho rango queda cubierto.

<sup>1</sup> Todas las características de los reguladores son tomadas de [18]

<b>Característica</b>	<b>Valor</b>
Voltaje de alimentación (V+)	3 hasta 32V
Voltaje de entrada	-0.3 hasta 32V
Potencia consumida	830mW
Temperatura de operación	0 – 70° C
Corriente de alimentación	1mA
Offset de voltaje en la entrada	2mV
Bias de corriente de entrada	45nA
Ganancia de lazo abierto	100V/mV
Rechazo de fuente de poder	100dB

**Tabla. 6.9. Características LM358**

### **6.5.8. Circuito impreso (PCB)**

Para diseñar un circuito impreso o PCB se deben tomar en cuenta diversos parámetros para evitar problemas de ruido. A continuación se describen los criterios tomados en cuenta en el diseño del PCB.

#### **6.5.8.1. Capacitores de bypass**

Los capacitores de bypass se colocan lo más cerca posible del pin de alimentación de los dispositivos. Estos capacitores sirven para eliminar el ruido producido por variaciones de la fuente. Se recomienda que para compuertas lógicas y microcontroladores este capacitor tenga un valor de 0.1 $\mu$ F. Para los amplificadores operacionales que funcionan en un ancho de banda de hasta 1MHz, como en este caso, también se recomienda capacitores de bypass de 0.1 $\mu$ F. Además se debe poner un capacitor de bypass en la salida de la fuente, cuyo valor debe estar entre 10 $\mu$ F y 100 $\mu$ F

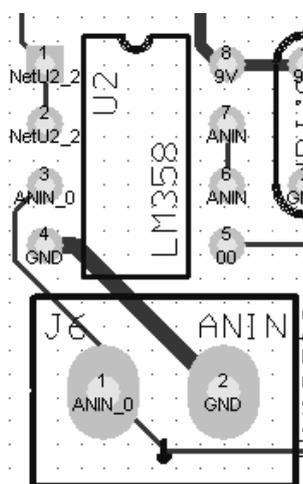
#### **6.5.8.2. Planos de tierra**

Un dispositivo conectado a tierra inyectará cierta cantidad de corriente en el conductor de tierra. La cantidad de corriente inyectada puede variar, por ejemplo cuando se activa o se desactiva una salida del microcontrolador conectada a un led. Debido a que los

conductores siempre tienen una resistencia, habrá una caída de voltaje que varía en el conductor debido a las corrientes que circulan por él. Si por ejemplo se usa la tierra como referencia en una conversión A/D, las variaciones de voltaje que sufre dicha tierra pueden afectara seriamente la conversión.

Dos razones son las principales por las que el manejo de las tierras se vuelve crítico. Primero, todos los dispositivos de la tarjeta y los dispositivos en las entradas y en las salidas se conectan a tierra, cada uno contribuyendo con ruido. Segundo, los dispositivos usan la tierra como referencia de 0V, por lo que variaciones de voltaje en ésta pueden producir serios efectos.

Como la magnitud de las variaciones de voltaje (ruido) en el conductor de tierra es directamente proporcional a la resistencia de dicho conductor, para disminuir este ruido se debe mantener dicha resistencia lo más pequeña posible haciendo que los conductores de tierra sean lo más anchos y lo más cortos posible. Una mejor solución es poner un solo plano de tierra en toda la tarjeta, con lo que la resistencia de la tierra se disminuye al mínimo.



**Figura. 6.17. Conexión del sensor con el plano de tierra**

Existen dispositivos muy sensibles a los ruidos en la tierra, como por ejemplo los sensores. Se recomienda que los mismos se conecten al plano de tierra en un solo punto, para mantener “separada” la tierra del sensor de la tierra *ruidosa* del sistema. La conexión se debe mantener lo más ancha y corta posible, y se debe hacer en la tierra del primer dispositivo de la tarjeta a la que llega la señal del sensor. Esto se puede ver en la Figura. 6.17 en la cual se muestra la conexión que se realizó desde la entrada del sensor de temperatura hasta el amplificador operacional usado para su filtrado.

La Figura. 6.17 muestra la cara superior de la tarjeta, el plano de tierra se encuentra en la cara inferior.

### 6.5.8.3. Vías de alimentación

El mismo criterio aplicado al diseño de tierra se podría aplicar al del diseño de las vías de alimentación, es decir que se podría crear un plano de alimentación. Esto no es necesario debido a que los ruidos en las vías de alimentación no son tan críticos por dos razones: primero, la alimentación no es usada como referencia de 0V. Segundo, muchos dispositivos analógicos, como los amplificadores operacionales, son diseñados con un buen nivel de inmunidad al ruido producido por la variación de la fuente. Por estas razones para minimizar el ruido en las vías de alimentación basta con hacerlas dos o tres veces más anchas que las pistas normales, y se debe colocar los capacitores de bypass como se describió en 6.5.8.1.

### 6.5.8.4. Ubicación de los elementos

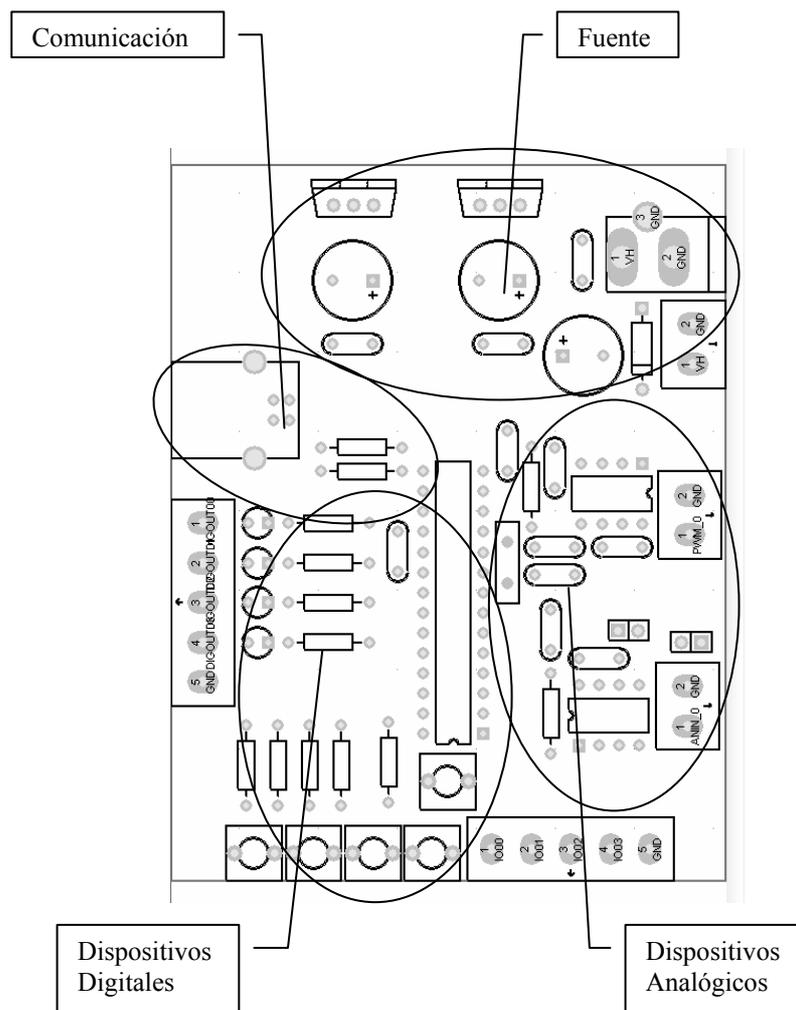


Figura. 6.18. Ubicación de los elementos en la placa

Algunos criterios deben seguirse para la correcta ubicación de los elementos. Se debe procurar que la parte analógica quede separada de la parte digital, pues las señales digitales generan gran cantidad de ruido que puede interferir con las señales analógicas. La parte de las comunicaciones debe tener también su espacio propio y debe estar la más cerca posible de su conexión con el microcontrolador, para evitar su contaminación con ruido. La fuente debe estar separada del resto del circuito. La ubicación empleada en el PCB diseñado puede verse en la Figura. 6.18

#### 6.5.8.5. Ancho de las pistas empleado

Tomando en cuenta los criterios expuestos, y bajo el parámetro de que se debe usar un ancho de pista de 0.25mm para las señales digitales, los anchos de pista empleados son los que se encuentran en la Tabla. 6.10

Descripción		Valor
Líneas de alimentación	Vcc	0.5mm
	9V	0.5mm
	GND	0.75mm
Líneas de entrada a la fuente		0.5mm
Señales digitales		0.25mm
Plano de tierra		SI

**Tabla. 6.10. Descripción de las pistas realizadas**

## **CAPÍTULO 7**

### **PRUEBAS Y RESULTADOS**

#### **7.1. INTRODUCCIÓN**

Luego de haber estudiado las bases de control adaptativo, y haber diseñado e implementado el correcto software y hardware para la planta de temperatura PCT-2, se procede a realizar varias pruebas. Las dos primeras pruebas fueron realizadas mediante software sin necesidad de utilizar un dispositivo de control externo, en éstas pruebas se determinan la ley de control de mejor desempeño según el criterio de ITAE, así como de algunas variables de ajuste. La última prueba fue realizada con el microcontrolador ya implementado, y esta comprobará el buen funcionamiento del hardware implementado.

Este capítulo está dividido de la siguiente manera en el ítem 7.2 se realizan las pruebas para la ley de control tipo C, la ley de control tipo A se estudia en el ítem 7.3. Luego de realizar estas pruebas se analizan los resultados en el ítem 7.4. Las pruebas finales y el análisis de los resultados se los observan en los ítems 7.5 y 7.6.

#### **7.2. PRUEBAS CON LA LEY DE CONTROL TIPO C**

##### **7.2.1. Control Adaptativo**

Estas pruebas fueron realizadas totalmente en línea esto quiere decir que continuamente se estiman los parámetros de la planta lo que implica que los coeficientes de la ley de control no son constantes.

Los elementos utilizados para esta prueba son: la PC que realiza los procesos de identificación de la planta, calculan los coeficientes de la ley de control, y ejecuta la ley de control tipo C. La tarjeta de adquisición de datos PMD-1208LS la cual enviará la variable manipulada y leerá la variable controlada de la planta. Finalmente la planta de temperatura la cual se encuentra disponible en los laboratorios de la FIE.

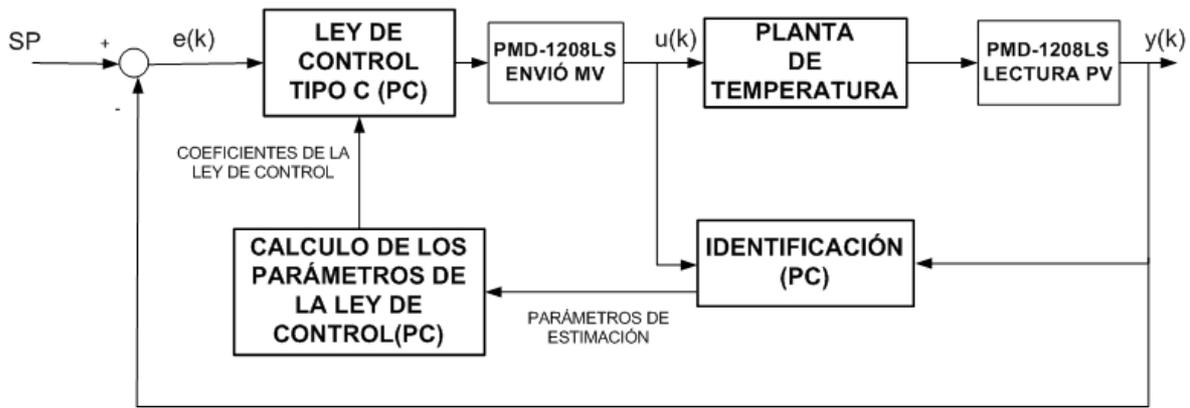


Figura.7.1. Esquema Control Adaptativo en línea mediante software ley de control tipo C

Donde:

$u(k)$  = Variable manipulada

$y(k)$  = Variable controlada

$e(k) = SP(k) - y(k)$

Las siguientes 4 pruebas usan el esquema presentado en la Figura.7.1. Las constantes para el cálculo de los parámetros del controlador y para la ley de control tipo C son las siguientes:

Constantes	Valor	Aplicación
F (filtro del setpoint)	20	Ley de control tipo C
Km	0.1	Ley de control tipo C
$\beta$ (beta-weight)	0.001	Ley de control tipo C
Tiempo de muestreo	0.5(seg)	Ley de control tipo C
Polo triple	1.2	Cálculo del controlador

Tabla.7.1. Constantes utilizadas para la ley de control tipo C

Utilizando la información de la Tabla.7.1 se puede calcular la ecuación del set point utilizada:

$$SP(k) = 0.9753 * SP(k - 1) + 0.02469SP0 \quad (7.1)$$

donde, SP es el setpoint filtrado y SP0 es el valor del setpoint ingresado.

La ecuación final de la ley de control tipo C será la siguiente.

$$u(k) = u(k - 1) + (-Kp * (y(k) - y(k - 1) + 0.001 * (SP - SP(k - 1))) + Ki * T0 * e(k) - \left(\frac{Kd}{T0}\right) * (y(k) - 2 * y(k - 1) + y(k - 2))) * 0.1 \quad (7.2)$$

Donde:

$T_0$  = Tiempo de muestreo

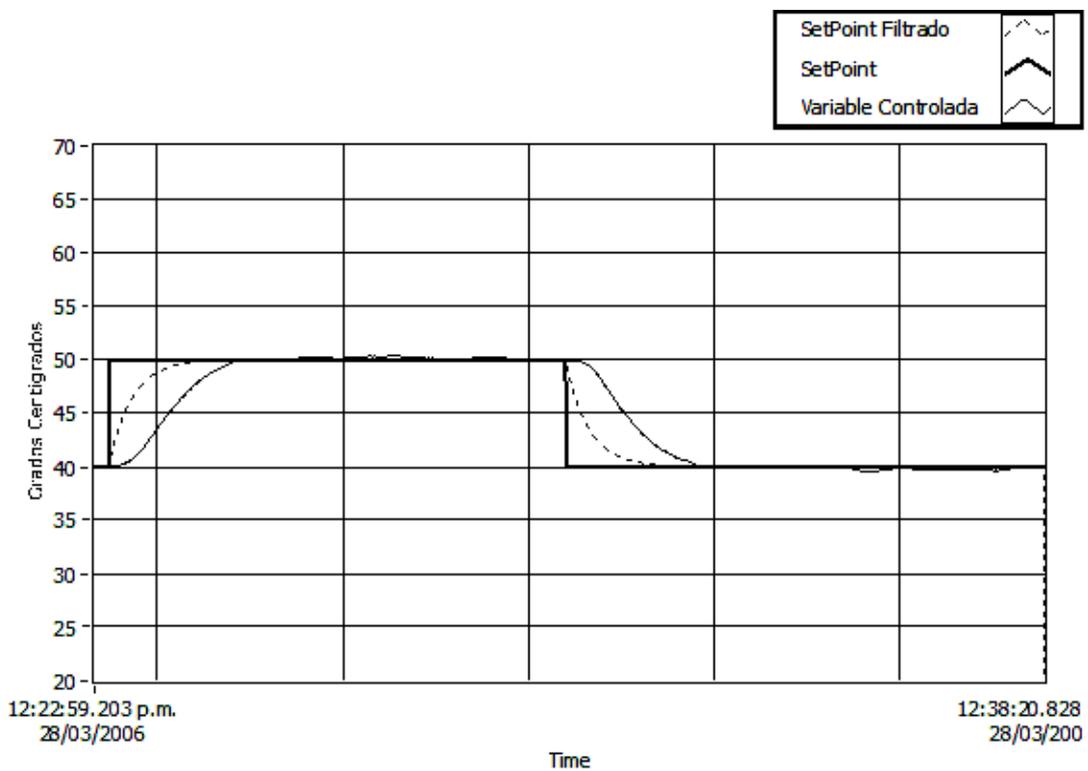
$K_i$  = Constante de integración

$K_d$  = Constante de derivación

$K_p$  = Constante proporcional

**7.2.1.1. Prueba 1: Control Adaptativo en condiciones normales utilizando la ley de control tipo C**

Esta primera prueba se realizó en condiciones normales de la planta, esto implica que el ventilador estaba 50% abierto y se utilizó las ecuaciones 7.1 y 7.2 para la ley de control. Los resultados se muestran en la Figura.7.2.



**Figura.7.2.** Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C.

La Tabla.7.2 indica el comportamiento de la respuesta de la primera prueba.

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	4350	3.5	0
50 a 40	4574	2.3	0

**Tabla.7.2.** Tabla de resultados de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C

Al realizar un breve análisis general de esta prueba se puede notar primeramente en la Figura.7.2 que la respuesta de la planta en condiciones normales no obtuvo ningún overshoot, y eso se clarifica en la Tabla.7.2. Los valores de tiempo de establecimiento e ITAE se analizarán más adelante.

### 7.2.1.2. Prueba 2: Control adaptativo con cambios de perturbación utilizando la ley de control Tipo C

Para realizar la prueba 2 se agregó perturbación a la planta y se mantuvo las mismas condiciones de la prueba 1. La Figura.7.3., enseña la respuesta ante una perturbación primero cerrando el ventilador y luego con el ventilador totalmente abierto.

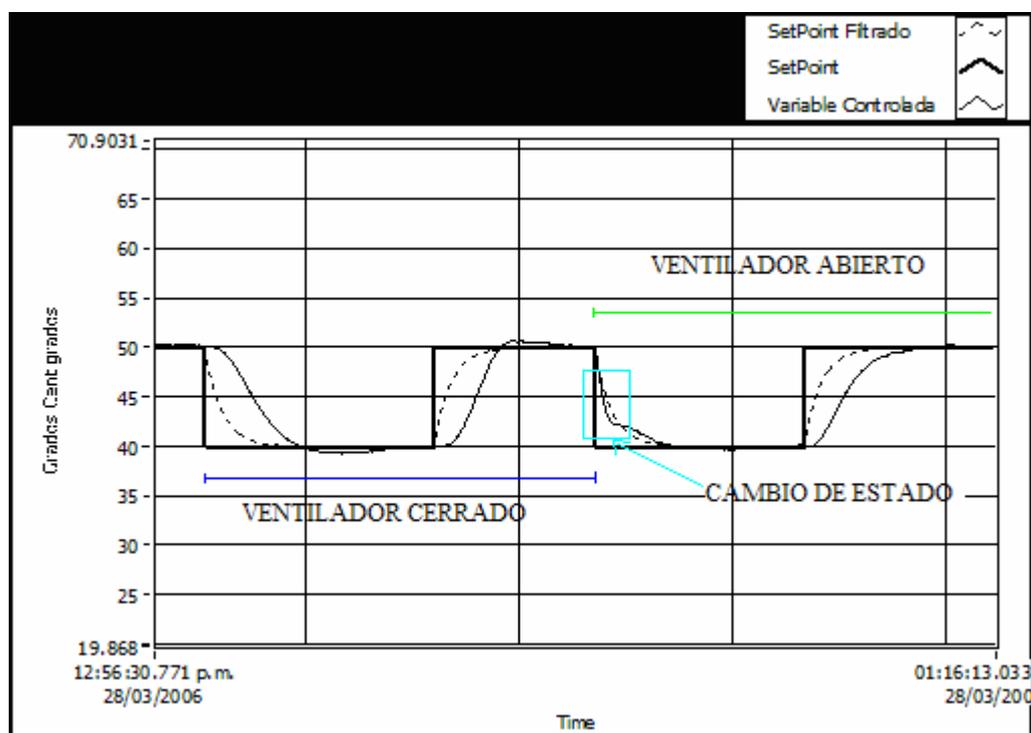


Figura.7.3. Respuesta de la planta con control adaptativo con cambios de perturbaciones utilizando la ley de control tipo C

Se puede observar que a pesar de los cambios de perturbación de la planta el sistema funciona correctamente, logrando así overshoot mínimos especialmente cuando el ventilador se encuentra cerrado.

En la Tabla.7.3 se encuentran los valores obtenidos de la respuesta de la planta.

En la Figura.7.4 se observa la respuesta de la planta ante cambio de perturbaciones, ventilador abierto-cerrado-abierto.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
50 a 40	cerrado	5020	4.9	0.5
40 a 50	cerrado	3060	3.7	1
50 a 40	abierto	3400	1.2	0
40 a 50	abierto	2500	2.5	0

Tabla.7.3. Tabla de resultados de la planta con control adaptativo con cambios de perturbaciones utilizando ley de control tipo C.

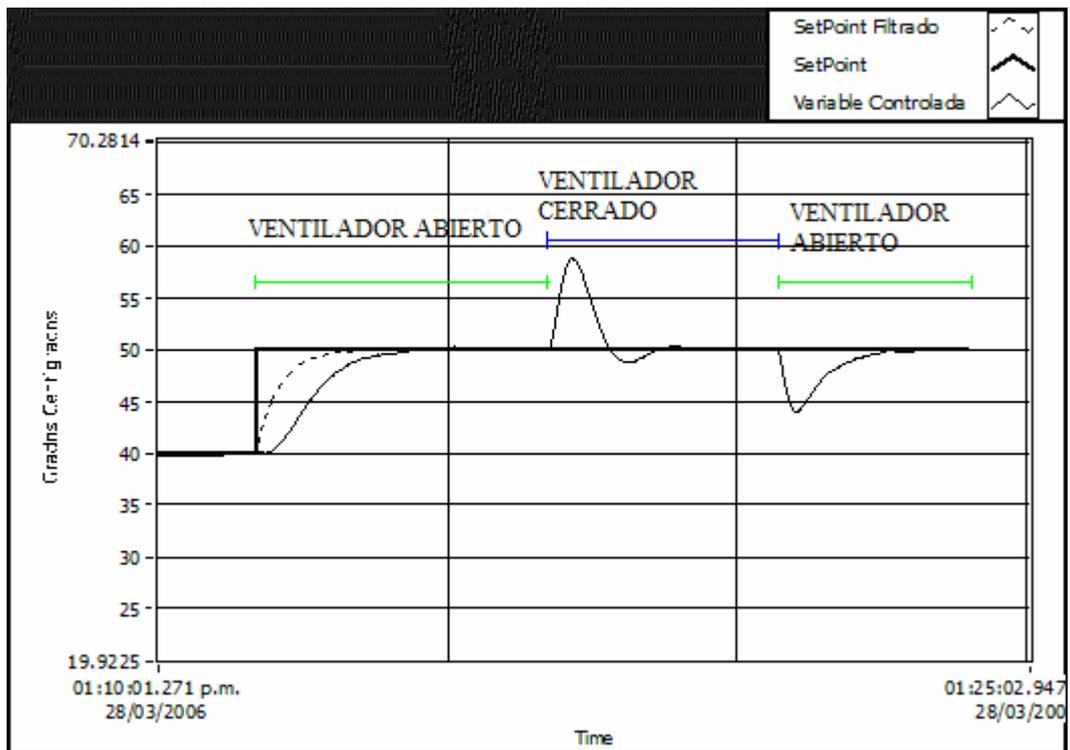


Figura.7.4. Respuesta de control adaptativo cambio de perturbaciones en un mismo punto de operación utilizando ley de control tipo C

### 7.2.1.3. Prueba 3: Control adaptativo en condiciones normales y con variación de beta-weight ( $\beta$ ).

La prueba 3 fue realizada para ver el comportamiento de la planta ante el cambio del factor B que atenúa el cambio de setpoint en condiciones normales. Se realizarán tres cambios con  $\beta$  igual a 1, luego será igual a 0.5 y finalmente tomará el valor de 0, como se ve en las Figura.7.5 y Figura.7.6.

- Para  $B=1$  y  $B=0.5$

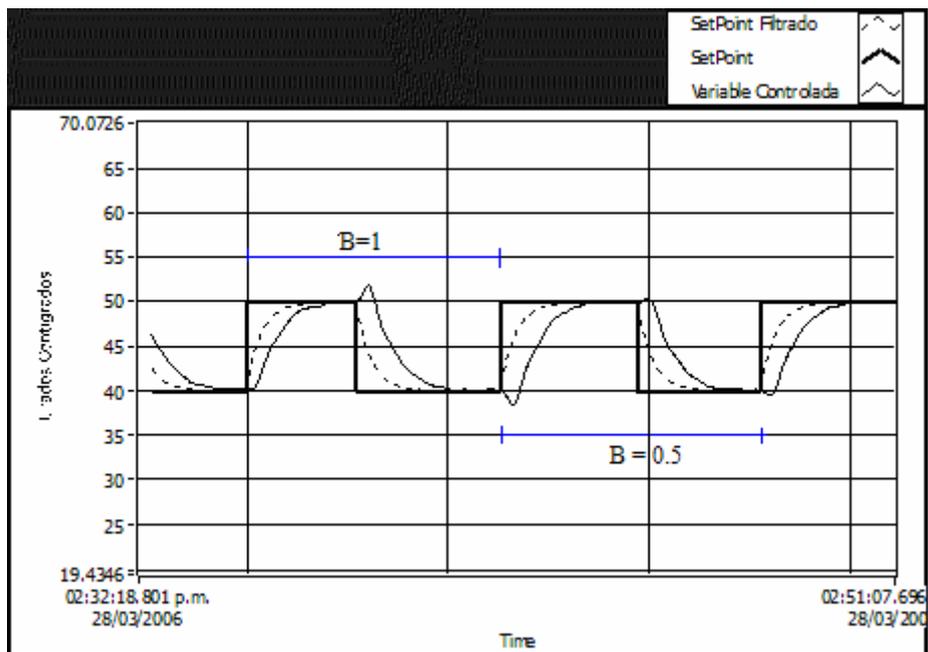


Figura.7.5. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C para  $B=1$  y  $B=0.5$

- Para  $B=0.5$  y  $B=0$ .

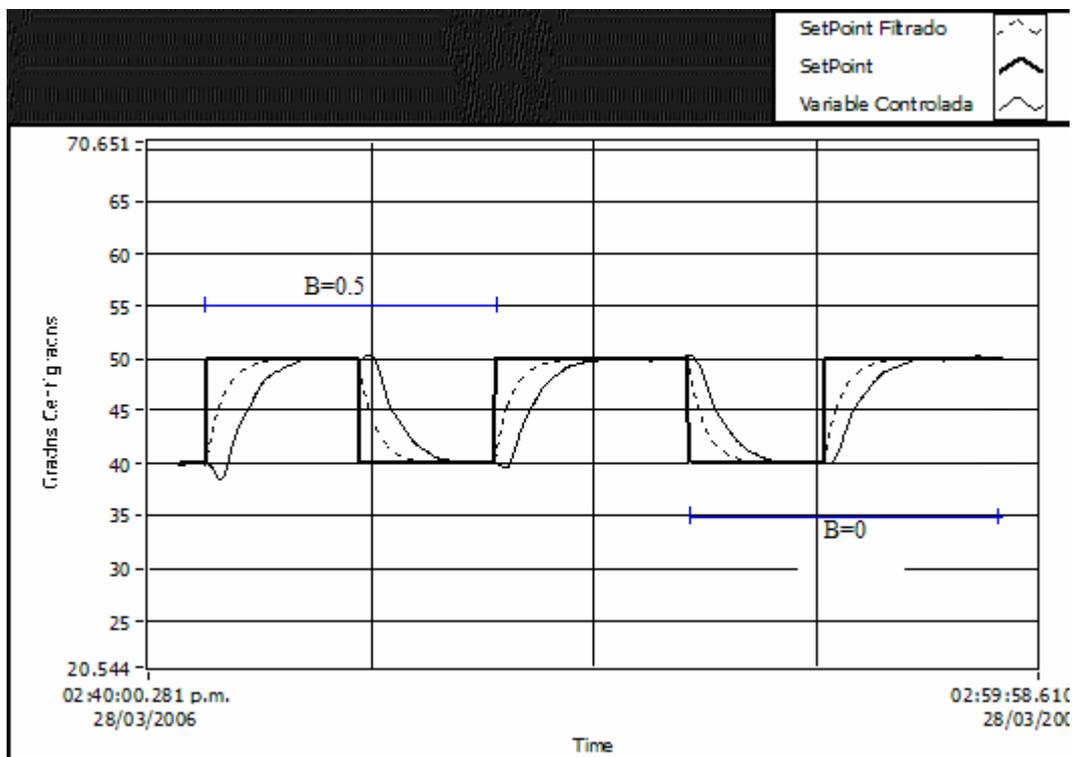


Figura.7.6. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C para  $B=0.5$  y  $B=0$

Los resultados de la respuesta de la planta ante la variación de B se presentan en la Tabla.7.4.

Variación de Setpoint (°C)	B	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	1	4800	3.8	0
50 a 40	1	5000	3.2	0
40 a 50	0.5	4000	3.7	0
50 a 40	0.5	4200	2.7	0
40 a 50	0	3050	2.9	0
50 a 40	0	3075	2.5	0

Tabla.7.4. Tabla de resultados con variación de B prueba 3

En la Tabla.7.4 se observa claramente que entre más pequeño sea el valor de B mas pequeño es el valor ITAE, que es la integral del error cuadrático, y el tiempo de establecimiento de la respuesta, lo mismo se puede observar en la Figura.7.6 que se observa un mejor rendimiento entre más pequeño es el valor de B. Pero si se coloca el parámetro en 0 no mejora nada la ley de control, por lo tanto lo mejor será un valor que se aproxime a 0.

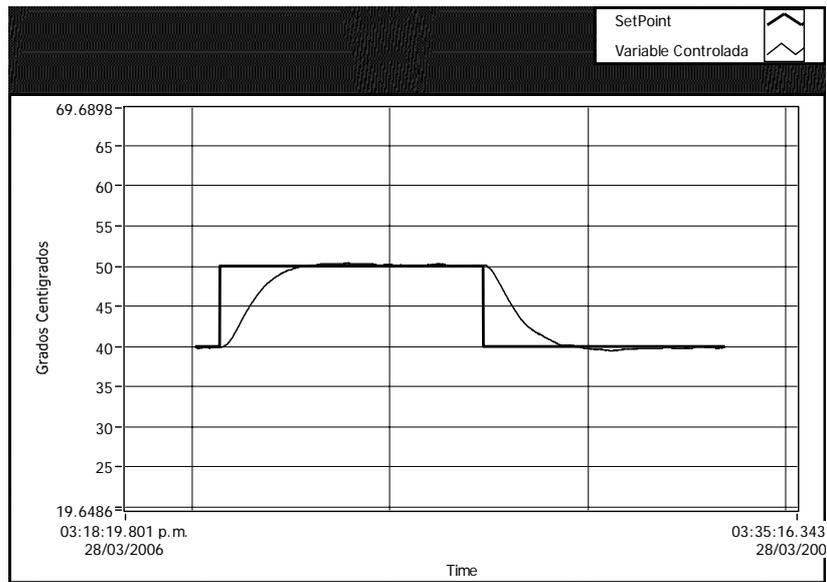
#### 7.2.1.4. Prueba 4: Control adaptativo en condiciones normales y ante perturbaciones sin usar ninguna mejora a la ley de control tipo C

La prueba número 4 fue realizada utilizando solamente la ley de control tipo C sin usar parámetros que mejoren el desempeño de esta ley, sin perturbación y con perturbación. La ecuación de la ley de control a utilizar será la siguiente:

$$u(k) = u(k-1) - Kp * (y(k) - y(k-1)) + Ki * T0 * e(k) - \left( \frac{Kd}{T0} \right) * (y(k) - 2 * y(k-1) + y(k-2)) \quad (7.3)$$

Los resultados se pueden observar en las Figura.7.7 que fue realizada en condiciones normales y la Figura.7.8 donde se aplicó perturbaciones.

- Sin perturbación



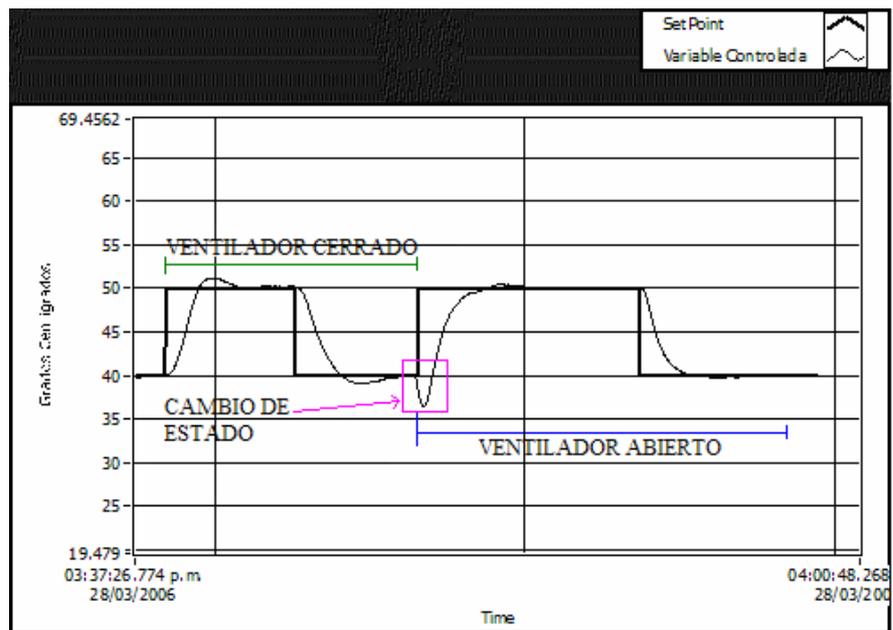
**Figura.7.7.** Gráfica de respuesta con control adaptativo en condiciones normales utilizando la ley de control tipo C sin mejoras en la ley

Los resultados de ésta prueba se pueden observar en la Tabla.7.5

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	4500	2.7	0
50 a 40	5000	3.2	0

**Tabla.7.5.** Tabla de resultados prueba 4

- Con perturbación



**Figura.7.8.** Gráfica de respuesta para la ley de control tipo C con perturbación ventilador cerrado y abierto sin ninguna mejora en la ley.

La Tabla.7.6 muestra los valores tomados en ésta prueba.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	5800	2.9	5
50 a 40	cerrado	3500	2.5	5
40 a 50	abierto	4200	2.9	1
50 a 40	abierto	2900	2.5	0

Tabla.7.6. Tabla de resultados para la prueba 4 con perturbación

Esta prueba fue realizada esencialmente para comprobar la necesidad de la utilización de parámetros que mejoren la ley de control.

Como se observa en la Tabla.7.5 los valores ITAE obtenidos tanto en condiciones normales como ante perturbaciones siguen siendo bajos esto indica que se tiene un buen desempeño de la ley de control tipo C a pesar de no usar ninguna constante que mejore su rendimiento. Pero al comparar los valores ITAE de esta prueba con los obtenidos en la Tabla.7.2 y Figura.7.3., se observa que se obtiene un mejor desempeño cuando se utilizan las mejoras en la ley de control ya que los valores ITAE son menores a los obtenidos en estas pruebas.

### 7.2.2. Self-Tuning

Para las siguientes tres pruebas solo se identificó la planta una vez y con los valores obtenidos se realizó las pruebas con la ley de control. En el esquema de la Figura.7.9 se realiza el proceso de estimación de los parámetros una sola vez, luego de calcular los coeficientes de la ley de control se desconecta el proceso de identificación y solamente se ejecuta la ley de control como se ve en el esquema de la Figura.7.10

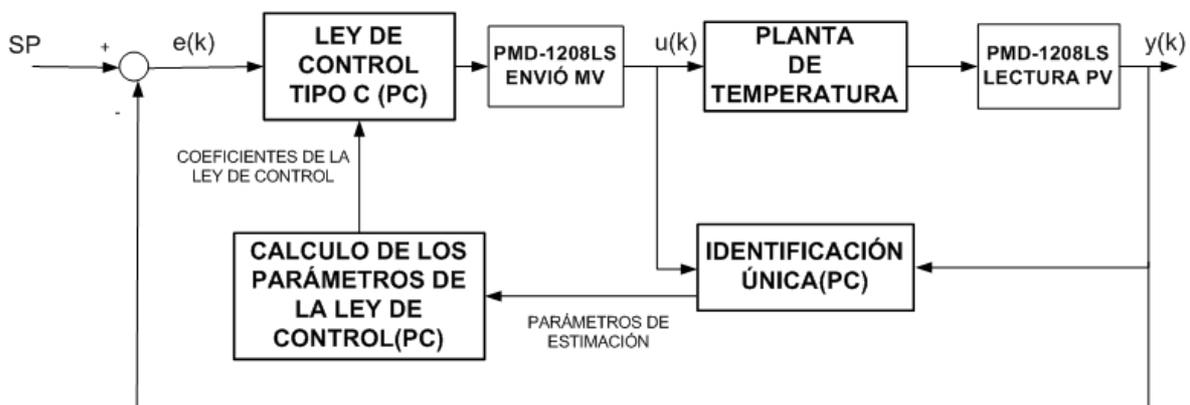


Figura.7.9. Esquema de Self-tuning con identificación única de la planta ley de control tipo C.

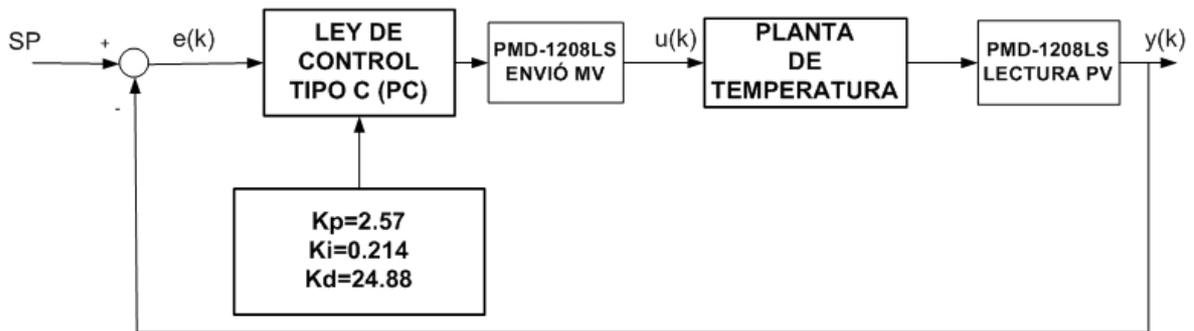


Figura.7.10. Esquema de self-tuning luego de realizar la identificación

Al realizar la identificación se obtuvo los siguientes valores para el controlador:

$$K_p = 2.57$$

$$K_i = 0.214$$

$$K_d = -24.88$$

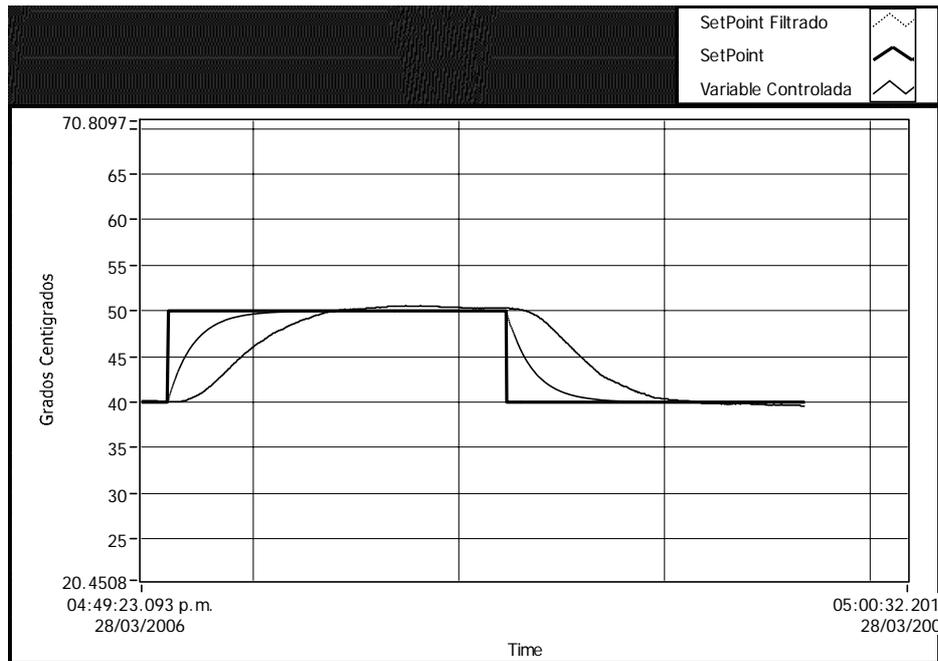
La ecuación final es la siguiente manteniendo los valores anteriores de ajuste de la ley de control y utilizando las mismas constantes para la ley de control de la Tabla.7.1

$$u(k) = u(k-1) + (-2.5714 * (y(k) - y(k-1) + 0.001 * (SP - SP(k-1))) + 0.214926 * 0.5 * e(k) - \left(\frac{24.8881}{0.5}\right) * (y(k) - 2 * y(k-1) + y(k-2))) * 0.1 \quad (7.4)$$

### 7.2.2.1. Prueba 5: Self-tuning en condiciones normales utilizando la ley de control tipo C.

Esta prueba fue realizada usando la ley de control 7.4. En condiciones normales esto quiere decir que el ventilador estaba el 50% abierto.

En la Figura.7.11 se muestra el resultado de la planta y en la Tabla.7.7 se presentan los resultados obtenidos.



**Figura.7.11. Gráfica de resultado con self-tuning en condiciones normales utilizando la ley de control tipo C.**

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	6500	4.03	1
50 a 40	5100	3.1	0

**Tabla.7.7. Tabla de resultados con self-tuning en condiciones normales utilizando la ley de control tipo C.**

Al tener parámetros constantes en la ley de control, se observa que el rendimiento de la respuesta de la planta ha decaído debido a que se han incrementado los valores ITAE. Esto es debido a que la planta de temperatura no tiene el mismo funcionamiento en distintos puntos de operación, y al identificar una sola vez la planta no se garantiza un buen desempeño en todos los puntos de operación.

**7.2.2.2. Prueba 6: Self-Tuning con cambios de perturbaciones utilizando la ley de control tipo C.**

La prueba 6 fue realizada tomando en consideración lo expuesto en la prueba 5 pero se agregó perturbación a la planta primero con el ventilador cerrado y luego abierto. En la Figura.7.12 se muestra la respuesta de la planta ante estas perturbaciones.

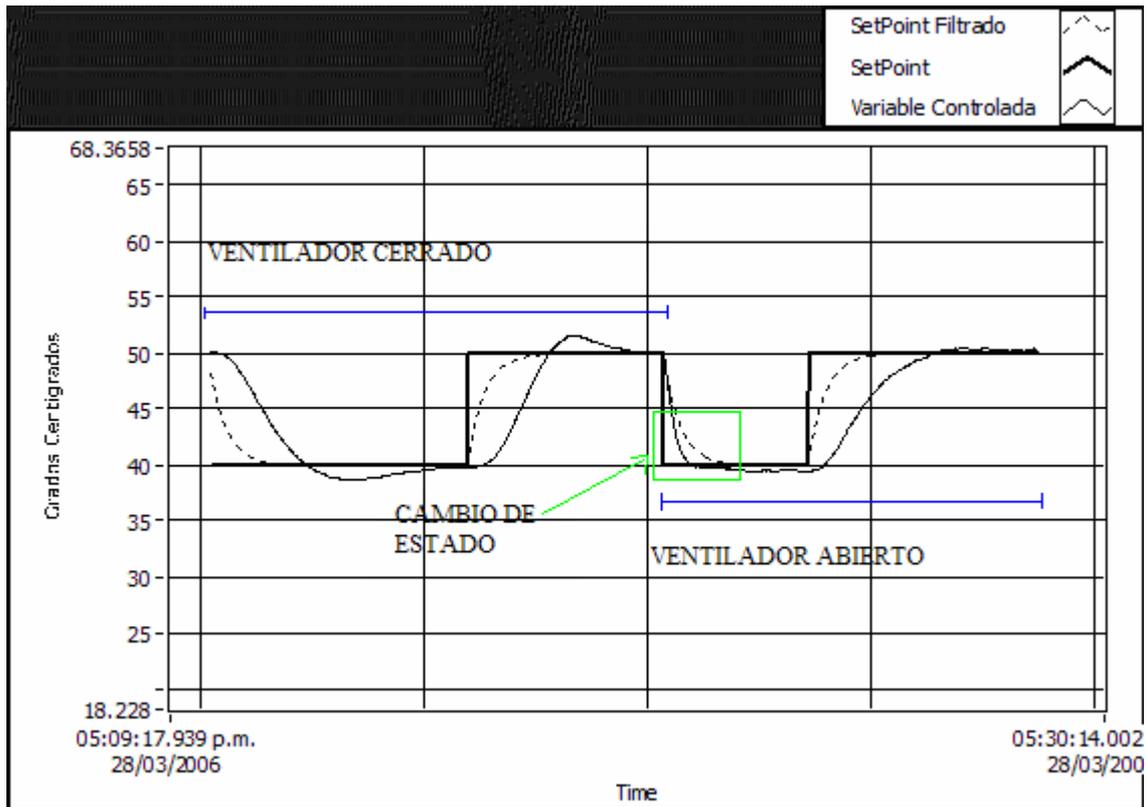


Figura.7.12. Respuesta de la planta con self-tuning con cambios de perturbaciones utilizando la ley de control tipo C

La Tabla.7.8 muestra los resultados obtenidos.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
50 a 40	cerrado	10000	6.5	30
40 a 50	cerrado	9600	4.9	30
50 a 40	abierto	1200	3.9	20
40 a 50	abierto	7500	5.5	10

Tabla.7.8. Tabla de resultados con self-tuning con cambios de perturbaciones utilizando la ley de control tipo C.

Como se habló en el ítem 7.2.2.1 de la necesidad de utilizar una identificación constante, en esta prueba se evidencia esta necesidad debido a que el valor de los ITAE, tiempo de establecimiento y overshoot han aumentado. Lo cual implica que el rendimiento del proceso es menor.

En la Figura.7.13 se muestra la respuesta de la planta ante cambios bruscos de perturbaciones con el ventilador abierto-cerrado-abierto.

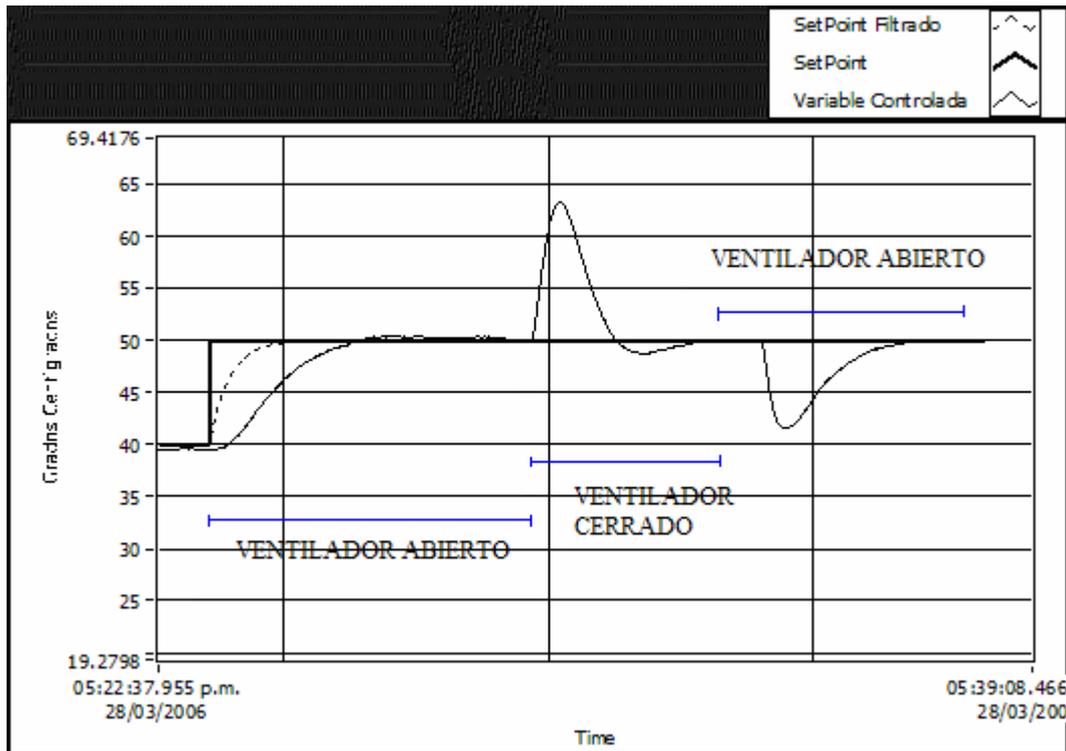


Figura.7.13. Gráfica de Self-Tuning ante cambios de perturbación en un mismo Setpoint

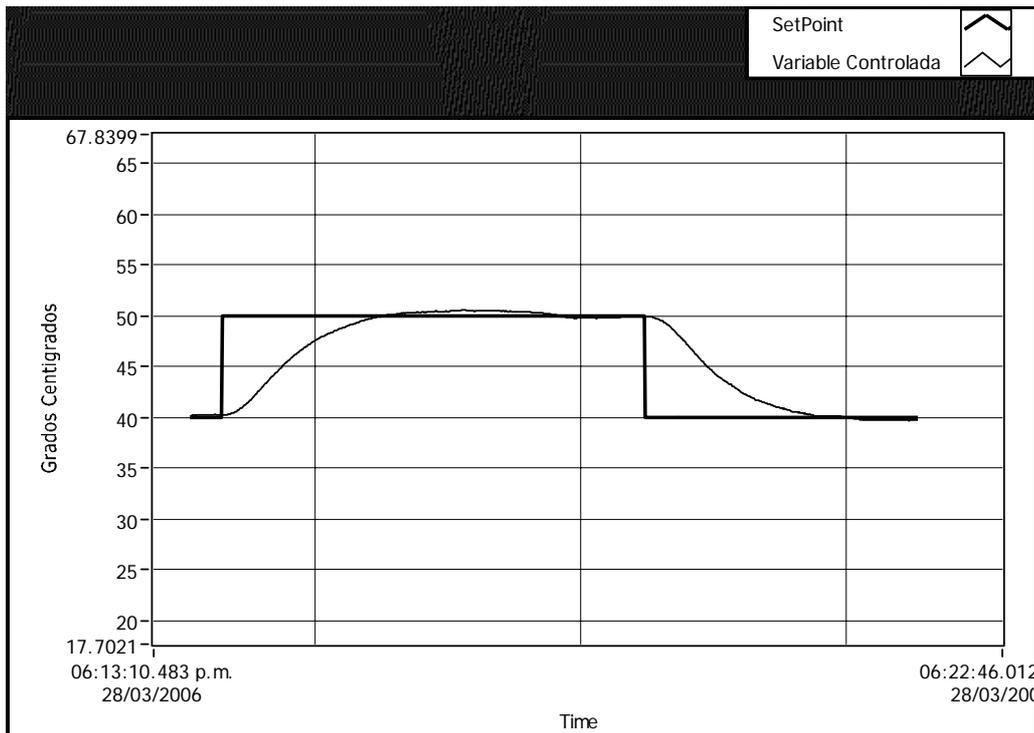
### 7.2.2.3. Prueba 7: Self-Tuning en condiciones normales y ante perturbaciones sin usar ninguna mejora a la ley de control tipo C

La prueba 7 se la realizó usando la ley de control tipo C sin utilizar parámetros adicionales que puedan mejorar el rendimiento de la ley de control. La Figura.7.14 muestra el resultado de la planta en condiciones normales y la Figura.7.15 muestra el resultado de la planta ante distintas perturbaciones.

La ley de control utilizada será la siguiente:

$$u(k) = u(k-1) - 2.5714 * (y(k) - y(k-1)) + 0.001 * +0.214926 * 0.5 * e(k) - \left( \frac{24.8881}{0.5} \right) * (y(k) - 2 * y(k-1) + y(k-2)) \quad (7.5)$$

- Sin perturbación



**Figura.7.14. Gráfica de respuesta ley de control tipo C sin perturbación prueba 7**

En la Tabla.7.9 se muestra los valores obtenidos de la respuesta de la planta en condiciones normales

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	3700	5.4	5
50 a 40	2700	2.8	0

**Tabla.7.9. Tabla de resultados prueba 7**

- Con perturbación

Las perturbaciones aplicadas primero el ventilador completamente cerrado y luego el ventilador completamente abierto.

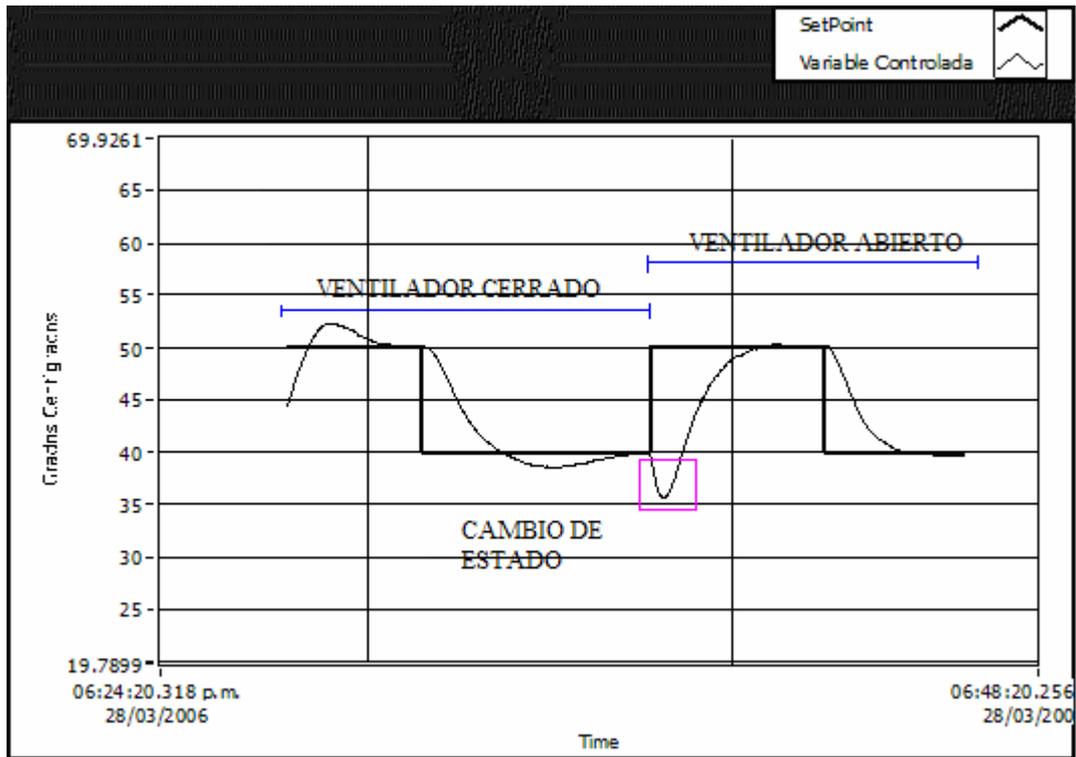


Figura.7.15. Gráfica de respuesta para la ley de control tipo C con perturbación ventilador cerrado y abierto

Los valores de la Tabla.7.10 muestran los resultados de la respuesta de la planta ante perturbaciones.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	6100	6.8	30
50 a 40	cerrado	14200	6.0	20
40 a 50	abierto	8800	4.3	1
50 a 40	abierto	3500	4.5	0

Tabla.7.10. Tabla de resultados para la prueba 7 con perturbación

Con esta prueba se confirma la necesidad de utilizar parámetros que mejoren la ley de control. Debido a que los valores de ITAE son elevados especialmente cuando el proceso actúa ante una perturbación ocasionada por el cierre total del ventilador.

### 7.3. PRUEBAS CON LA LEY DE CONTROL TIPO A

Para este tipo de pruebas se cambió la ley de control empleada la cual fue la siguiente:

$$u(k) = u(k - 1) + [Kc + \frac{T0}{Ti} + \frac{Td}{T}]e(k) - [Kc + \frac{2Td}{T}]e(k - 1) + \frac{Td}{T}e(k - 2) \quad (7.6)$$

### 7.3.1. Control Adaptativo

En la Figura.7.16 se muestra el esquema utilizado para las pruebas 8 hasta la 10.

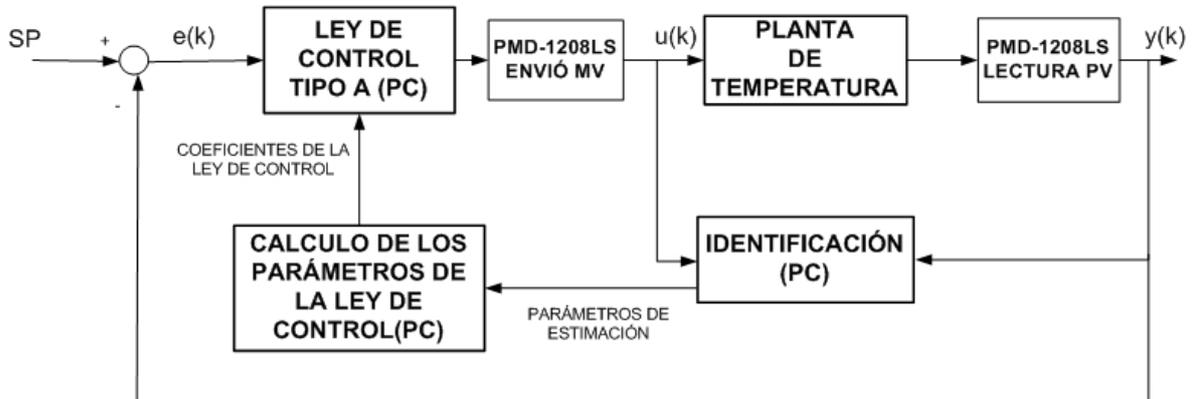


Figura.7.16. Esquema de control adaptativo mediante software utilizando la ley de control tipo A.

Los elementos a utilizar en estas pruebas son: PC que identifica los parámetros de la planta, calcula los coeficientes de la ley de control tipo A y ejecuta la ley de control tipo A, la tarjeta de adquisición de datos que envía la variable manipulada y lee la variable controlada del proceso y una planta de temperatura.

La Tabla.7.11 presenta los parámetros que mejoran el desempeño de la ley de control tipo A.

Constantes	Valor	Aplicación
F(Filtro del setpoint)	20%	Ley de control tipo A
Km	0.1	Ley de control tipo A
β(beta-weight)	0.001	Ley de control tipo A
Tiempo de muestreo	0.5(seg)	Ley de control tipo A
Polo triple	1.2	Cálculo de la ley de control

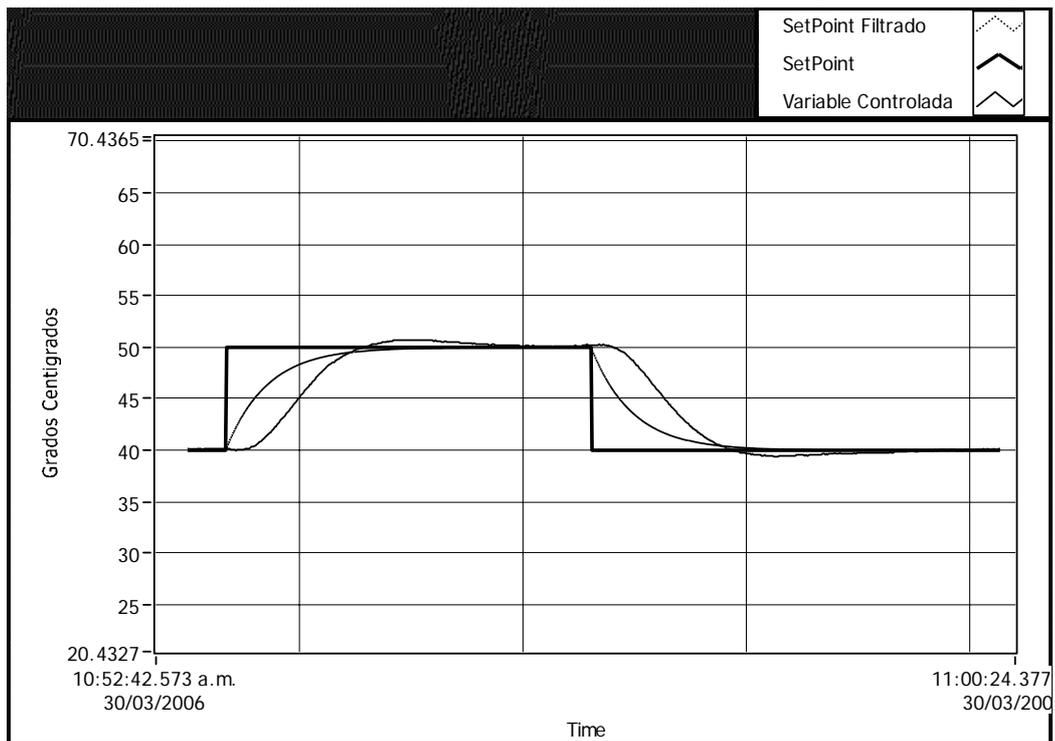
Tabla.7.11. Constantes utilizadas para la ley de control tipo A

Finalmente la ecuación es:

$$u(k) = u(k - 1) + \left[ \left[ Kc + \frac{0.5}{Ti} + \frac{Td}{T} \right] e(k) - \left[ Kc + \frac{2Td}{T} \right] e(k - 1) + \frac{Td}{T} e(k - 2) \right] * 0.1 \quad (7.7)$$

#### 7.3.1.1. Prueba 8: Control Adaptativo en condiciones normales utilizando la ley de control tipo A

La prueba numero 8 fue realizada en condiciones normales esto quiere decir que el ventilador de la planta estaba el 50% abierto y utilizando la ecuación 7.7. En la Figura.7.17 se muestra el resultado de la prueba en condiciones normales.



**Figura.7.17. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo A**

La Tabla.7.12 muestra los valores de la respuesta de la planta.

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	8800	3.2	10
50 a 40	1200	3.7	20

**Tabla.7.12. Tabla de resultados para la prueba 7 con perturbación**

Los valores de ITAE obtenidos con la ley de control tipo A son grandes en comparación a los valores obtenidos en la Tabla.7.2., a pesar de que la planta finalmente logra ser controlada. Por lo tanto bajo el criterio ITAE el rendimiento de este proceso no es bueno en comparación al obtenido con la ley de control tipo C.

### 7.3.1.2. Prueba 9: Control Adaptativo ante perturbaciones utilizando la ley de control tipo A

La prueba número 9 fue realizada usando los mismos parámetros y ley de control de la prueba 8 pero agregando perturbación. Las perturbaciones fueron aplicadas con el ventilador cerrado y luego abierto. En la Figura.7.18 se muestra la respuesta de la planta ante éstas perturbaciones.

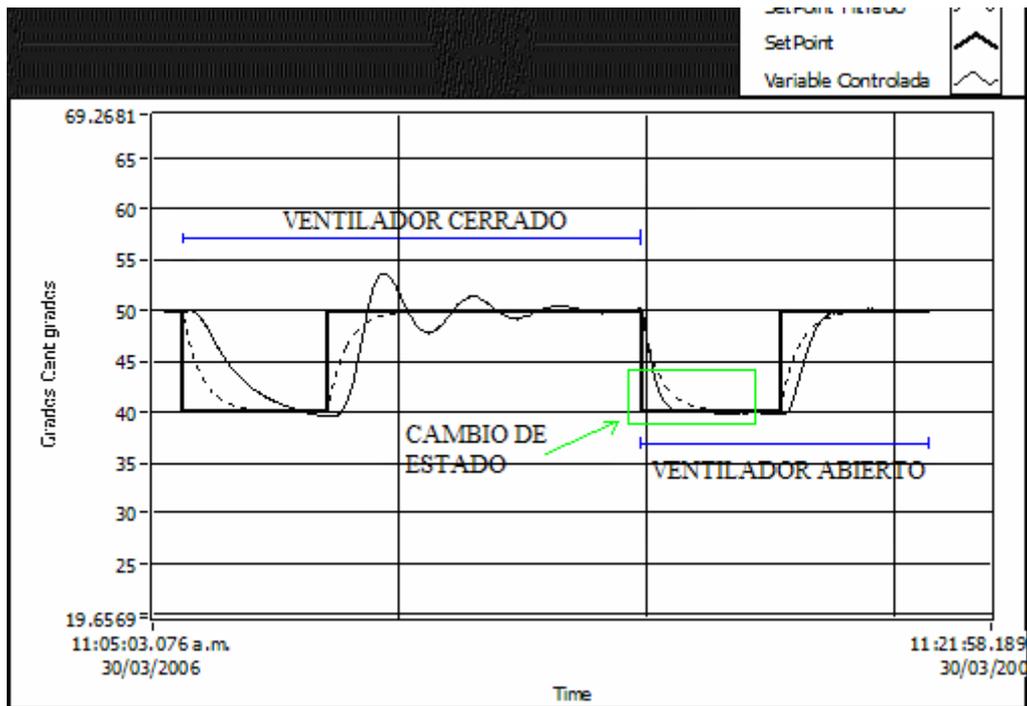


Figura.7.18. Respuesta de la planta con control adaptativo ante perturbaciones utilizando la ley de control tipo A

La Tabla.7.13 indica los valores obtenidos de la planta.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	4000	4.5	10
50 a 40	cerrado	7000	6.23	40
40 a 50	abierto	490	2.9	0
50 a 40	abierto	1045	1.9	0

Tabla.7.13. Tabla de resultados con control adaptativo ante perturbaciones utilizando la ley de control tipo A

Al perturbar la planta observamos en la Figura.7.18 que existe un gran número de oscilaciones antes de estabilizarse cuando el ventilador está cerrado en un cambio de setpoint de 50 a 40, esto se debe a que la ley de control tipo A no toma en cuenta el valor anterior del setpoint y solamente la señal del error. Esto provoca menos exactitud al calcular la variable manipulada que recibirá la planta.

La Figura.7.19 muestra el rendimiento de la planta ante la variación de perturbaciones con ventilador abierto-cerrado-abierto.

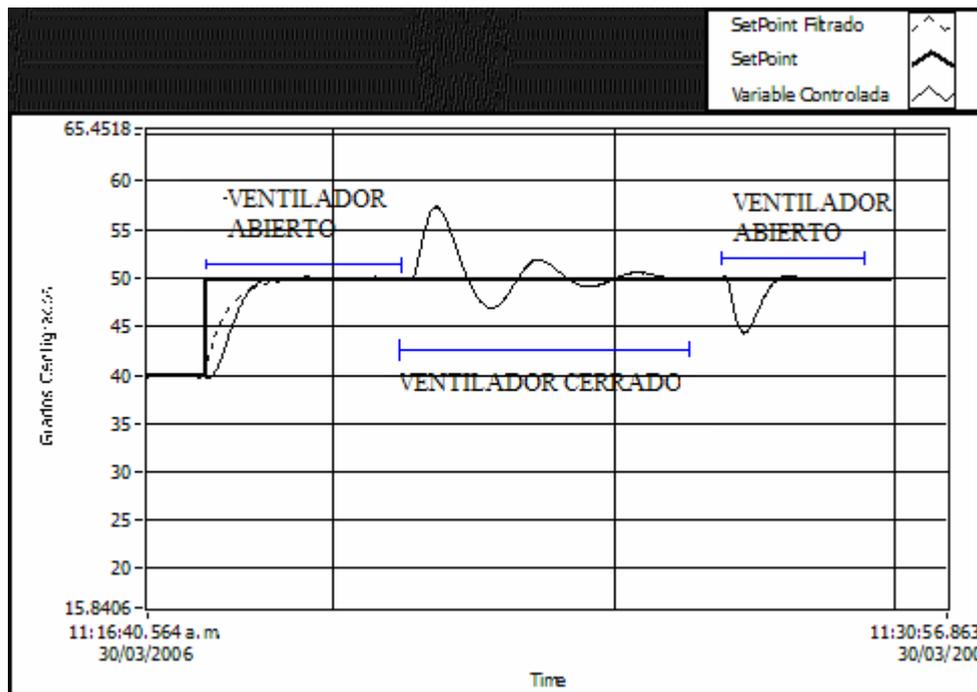
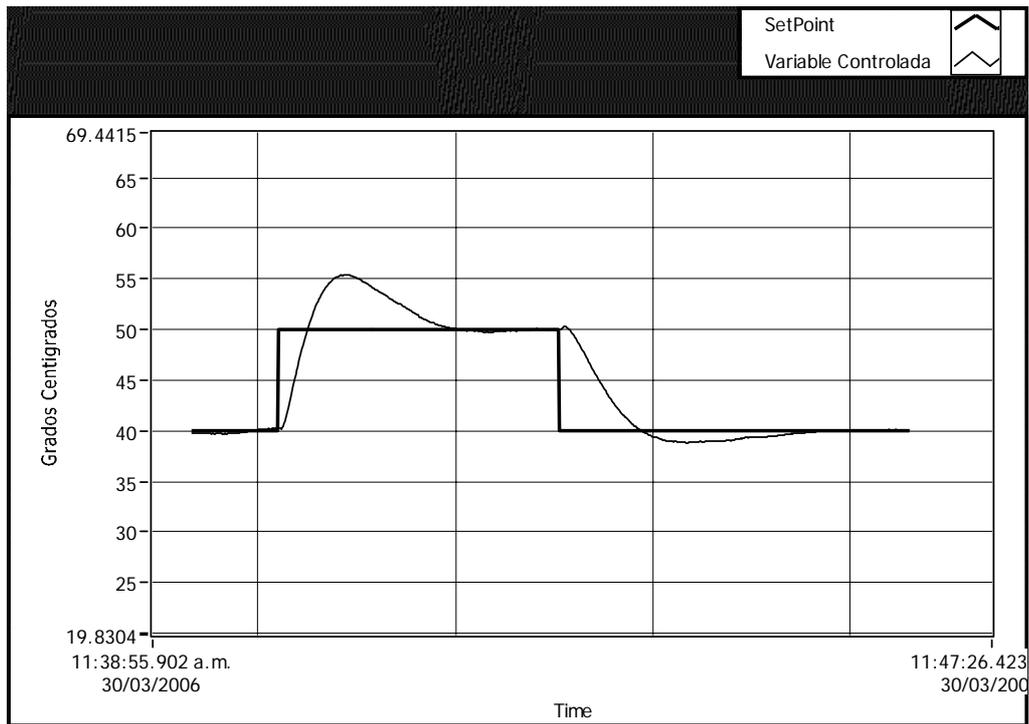


Figura.7.19. Respuesta ante perturbación en un mismo punto de operación con control adaptativo utilizando la ley de control tipo A

### 7.3.1.3. Prueba 10: Control Adaptativo en condiciones normales y ante perturbaciones utilizando la ley de control tipo A sin mejoras de la ley de control

En esta prueba se utilizó únicamente la ley de control tipo A sin usar ningún parámetro que mejore dicha ley, en la Figura.7.20 se muestran los resultados en condiciones normales y en la Figura.7.21 se muestra los resultados de la planta aplicando perturbaciones la primera con el ventilador cerrado y luego con el ventilador abierto.

- Sin perturbación



**Figura.7.20. Respuesta de la planta con control adaptativo utilizando la ley de control tipo A sin mejoras en la ley de control.**

En la Tabla.7.14 se observan los resultados obtenidos de la planta en condiciones normales

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	2400	2.6	51
50 a 40	1800	2.9	30

**Tabla.7.14. Tabla de resultados prueba 10**

- Con perturbación

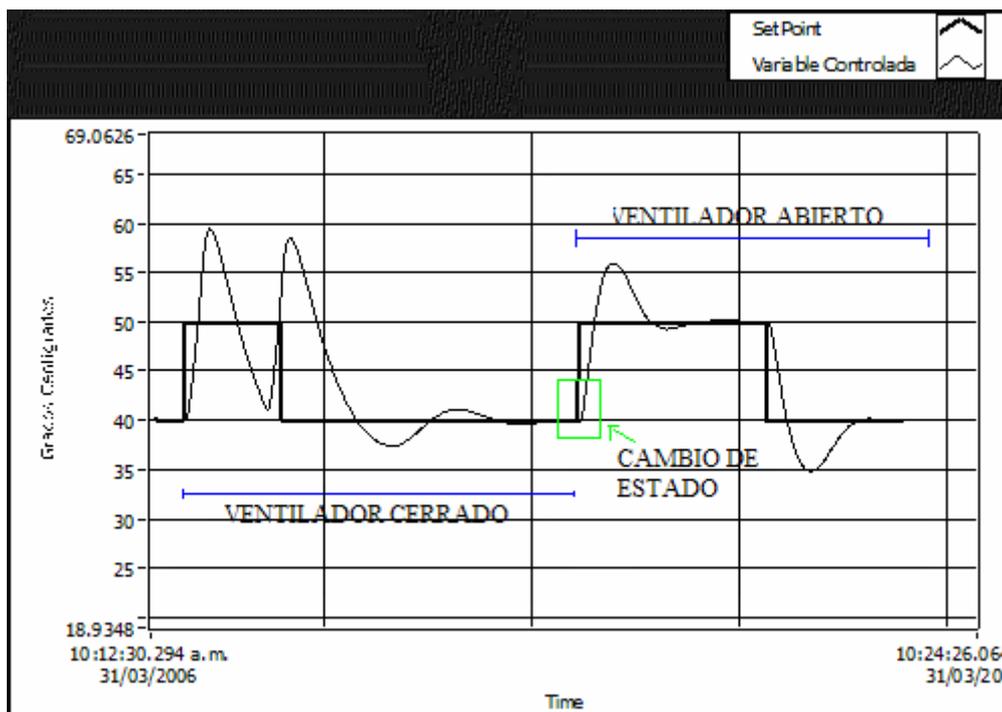


Figura.7.21. Resultado de la ley de control tipo A con perturbación prueba 10

Los resultados obtenidos se muestran en la Tabla.7.15

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	inestable	inestable	100
50 a 40	cerrado	5000	3.8	30
40 a 50	abierto	120	1.89	65
50 a 40	abierto	1200	1.78	53

Tabla.7.15. Tabla de resultados para la prueba 10 con perturbación

Al quitar los parámetros de mejora de la ley de control se observa claramente en la Figura.7.20 que a pesar de trabajar en condiciones normales se obtiene un overshoot del 51% y cuando se agrega perturbaciones al proceso lo hace inestable como se observa en la Tabla.7.15. A pesar que los valores ITAE son bajos la inestabilidad descarta esta ley de control.

### 7.3.2. Self Tuning

Para las siguientes pruebas se realizó una identificación única. Como se observa en la Figura.7.22 se realiza una estimación única de los parámetros de la planta y de los coeficientes de la ley de control. En el esquema de la Figura.7.23 con los coeficientes determinados se realiza el proceso de control.

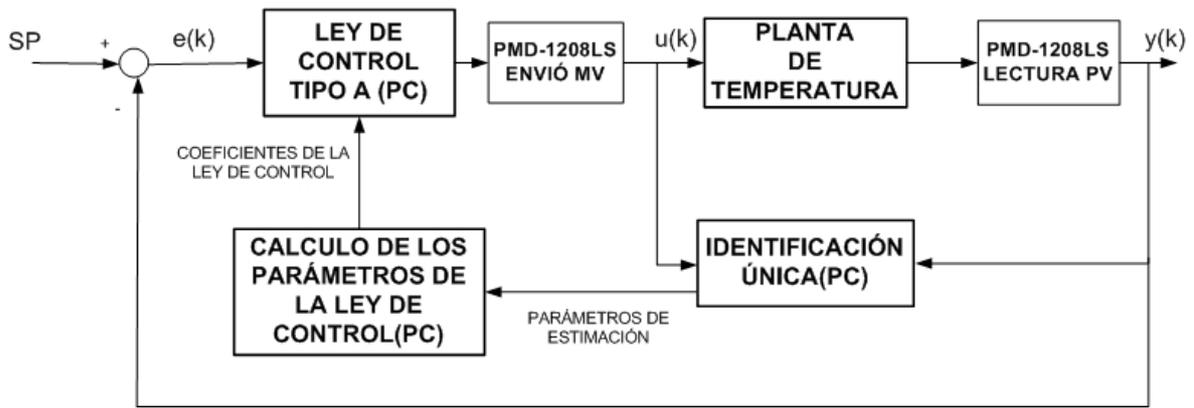


Figura.7.22. Esquema de identificación única de la planta y ley de control tipo A mediante software

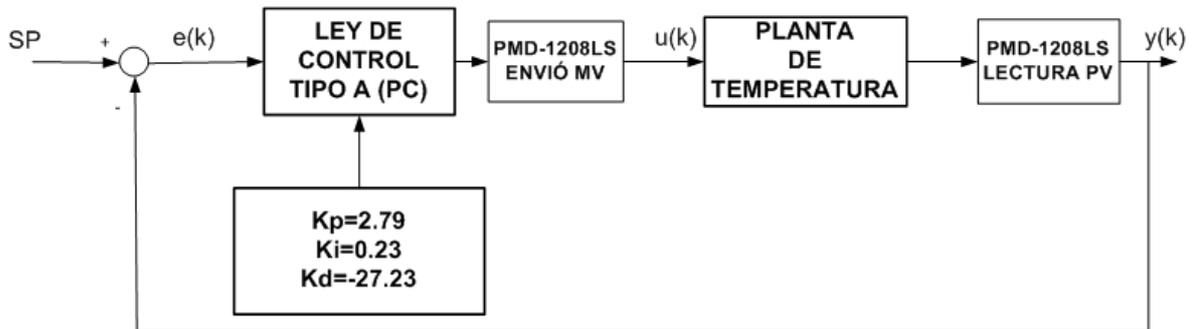


Figura.7.23. Esquema de Self-tuning luego de realizar la identificación

Las constantes para la ley de control tipo A fueron los siguientes:

$$K_p = 2.79$$

$$K_i = 0.23$$

$$K_d = -27.23$$

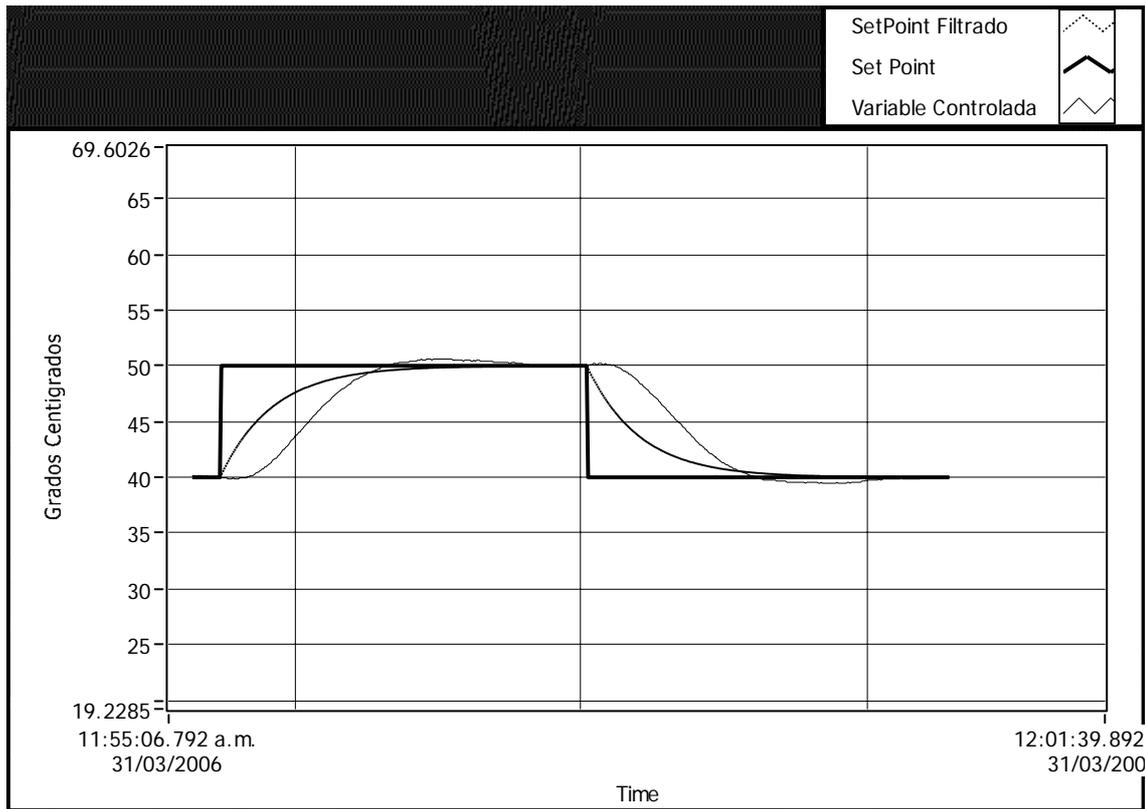
Por lo tanto la ley de control usada fue la siguiente:

$$u(k) = u(k-1) + \left[ 2.7989 + 0.2329 * T_0 + \frac{-27.2316}{T} \right] e(k) - \left[ K_c + \frac{2Td}{T} \right] e(k-1) + \frac{Td}{T} e(k-2) \quad (7.7)$$

Las siguientes pruebas van a corroborar que la ley de control tipo A no tiene un buen desempeño menos aun cuando no se realiza una identificación constante.

### 7.3.2.1. Prueba 11: Self-Tuning en condiciones normales utilizando la ley de control tipo A.

La prueba 11 fue realizada bajo las mismas condiciones de la prueba 9. La Figura.7.24 indica los resultados de la planta obtenidos.



**Figura.7.24. Respuesta para la ley de control A para prueba 11**

En la Tabla.7.16 muestra los resultados obtenidos por la planta.

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	1700	2.18	10
50 a 40	1780	1.9	10

**Tabla.7.16. Tabla de resultados prueba 11**

**7.3.2.2. Prueba 12: Self-Tuning ante perturbaciones utilizando la ley de control tipo A**

La prueba fue realizada usando los mismos parámetros de la prueba 11, pero las perturbaciones aplicadas fueron: primero con ventilador cerrado y luego con ventilador abierto. Como se ve en la Figura.7.25.



**Figura.7.25. Respuesta de la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A**

La Tabla.7.17 exhibe los resultados de la planta obtenidos.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	5400	4.3	20
50 a 40	cerrado	9200	4.7	40
40 a 50	abierto	1000	2.4	10
50 a 40	abierto	2000	3.3	10

**Tabla.7.17. Tabla de resultados para la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A**

La Figura.7.26 muestra la respuesta ante perturbaciones con ventilador abierto-cerrado-abierto.

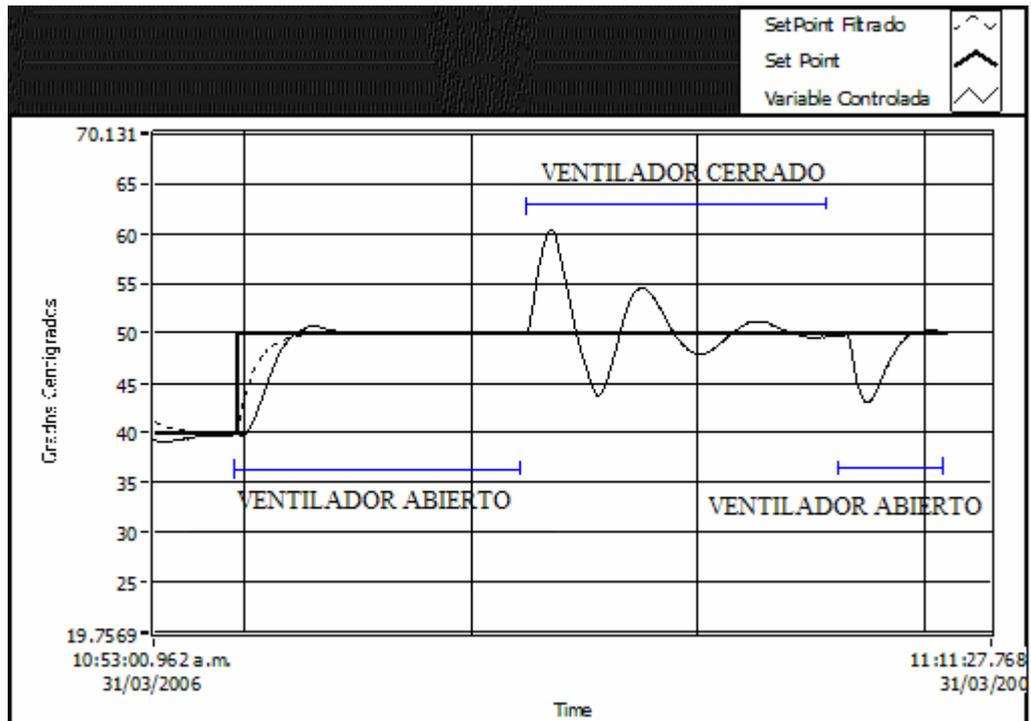


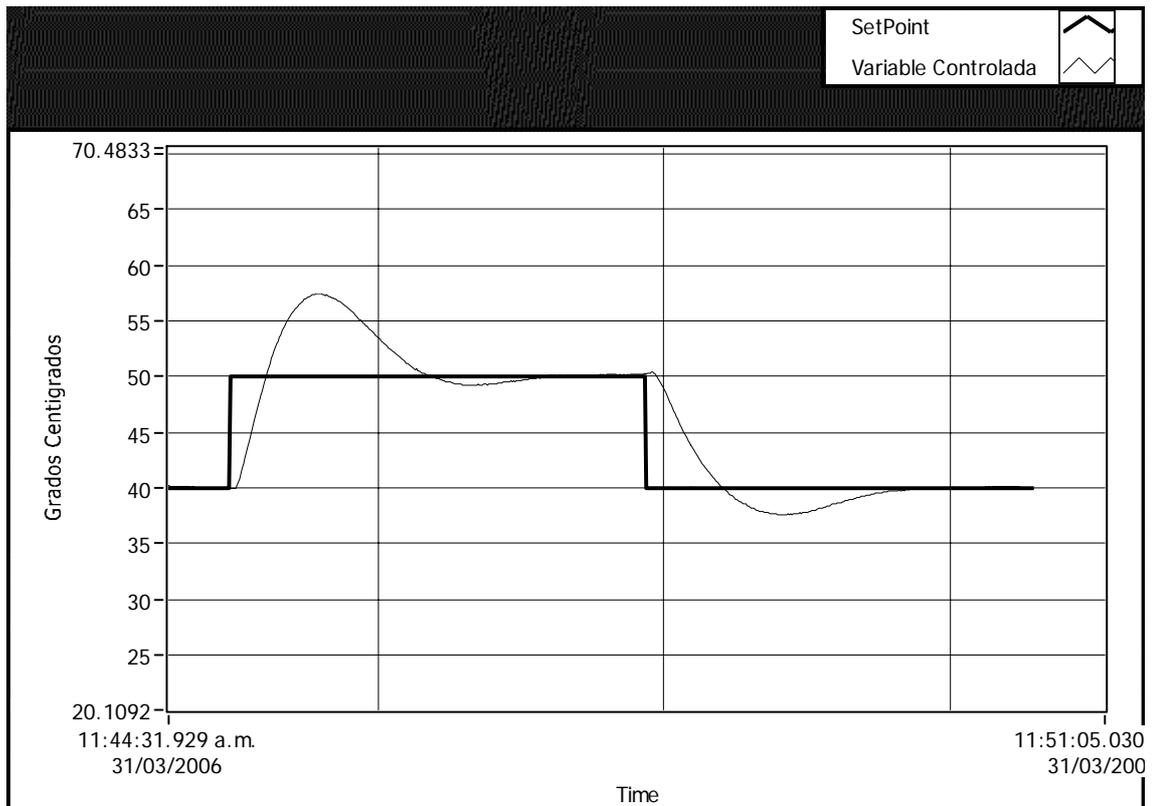
Figura.7.26. Respuesta ante perturbaciones utilizando la ley de control tipo A con Self-Tuning

### 7.3.2.3. Prueba 13: Self-Tuning en condiciones normales y ante perturbaciones utilizando la ley de control tipo A sin parámetros que mejoran la ley de control

En esta prueba se utilizó únicamente la ley de control tipo A, en este ítem se mostrarán los resultados con perturbación y sin perturbación.

En la Figura.7.27 se muestra el resultado de la planta en condiciones normales y en la Figura.7.28 se muestra el resultado de la planta aplicando distintas perturbaciones, con el ventilador cerrado y luego abierto.

- Sin perturbación



**Figura.7.27.** Respuesta de la planta con self-tuning en condiciones normales utilizando la ley de control tipo A sin parámetros que mejoran la ley de control

En la Tabla.7.18 se observa los valores obtenidos en la prueba 13.

Variación de Setpoint (°C)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	2616	3.25	80
50 a 40	1400	2.3	30

**Tabla.7.18.** Tabla de resultados con self-tuning en condiciones normales utilizando la ley de control tipo A sin parámetros que mejoran la ley de control

- Con perturbación

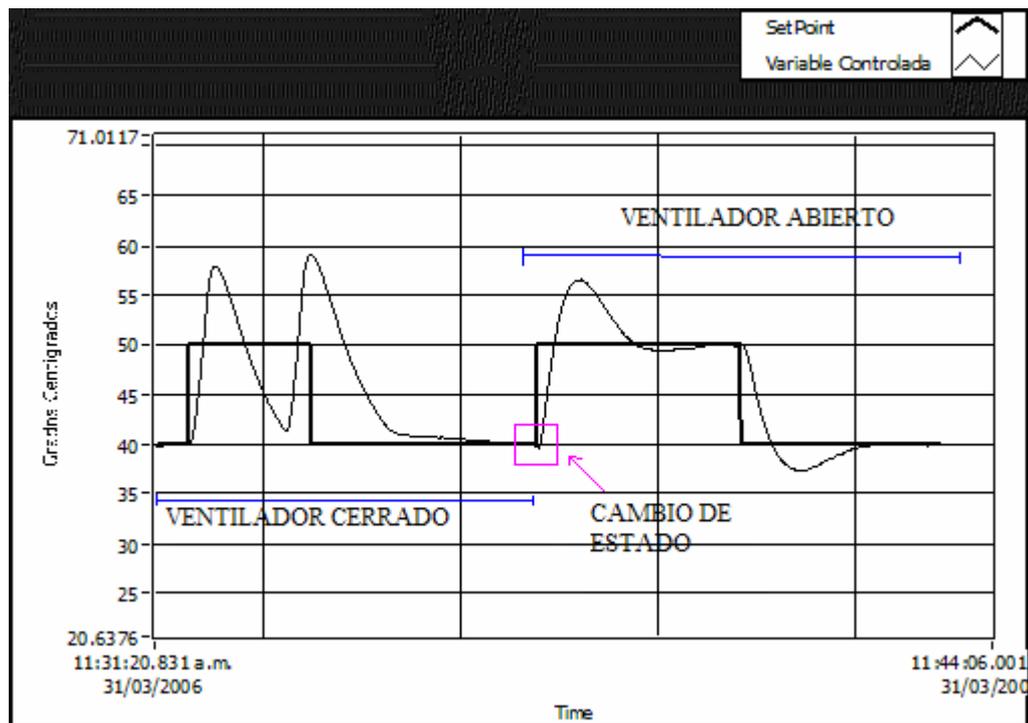


Figura.7.28. Respuesta de la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A sin parámetros que mejoran la ley de control

La Tabla.7.19 muestra los resultados de la planta ante cambio de perturbaciones.

Variación del SP (°C)	Perturbación (Ventilador de la planta)	ITAE	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	inestable	inestable	80
50 a 40	cerrado	3810	3.36	0
40 a 50	abierto	2700	3.64	70
50 a 40	abierto	1643	2.52	30

Tabla.7.19. Tabla de resultados de la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A sin parámetros que mejoran la ley de control

#### 7.4. RESULTADOS PRELIMINARES OBTENIDOS

Finalmente luego de realizar las pruebas vamos a concluir con unas tablas comparativas que corroboraran con lo anteriormente mencionado.

##### 7.4.1. Comparación entre Control Adaptativo Vs Self Tuning

La Tabla.7.20 tiene los valores comparativos de las pruebas 1 y 5, las cuales funcionan en condiciones normales, la diferencia es que en la prueba 1 se realiza una identificación constante, y en la prueba 5 se identifica el proceso de estimación de parámetros se lo realiza una sola vez.

	<b>PRUEBA</b>	<b>ITAE</b>	<b>t.e (min)</b>	<b>Overshoot (%)</b>
<b>Control Adaptativo</b>	1	4350	3, 5	0
<b>Self Tuning</b>	5	6500	4,03	5

**Tabla.7.20. Comparación entre Control Adaptativo vs Self Tuning prueba 1 vs prueba 5**

Como ya se explicó la planta de temperatura no tiene el mismo desempeño en distintos puntos de operación, esto quiere decir que la planta de temperatura tiene un funcionamiento distinto cuando eleva la temperatura y cuando la disminuye.

Por lo tanto es más recomendable utilizar una identificación constante de la planta cuando se realizan cambios de setpoint, esto logra que se tenga un mejor rendimiento en el control. Esto se puede observar en los diferentes valores de ITAE obtenidos, cuando se realiza una identificación constante el valor del ITAE es menor que cuando se realiza una única identificación.

#### **7.4.2. Comparación entre la ley de control Tipo A y Tipo C**

Para analizar las dos leyes se toma como referencia la prueba 1 que utiliza en condiciones normales la ley de control tipo C y la prueba 9 que emplea la ley de control tipo A. La Tabla.7.21 nos muestra un cuadro comparativo.

	<b>PRUEBA</b>	<b>ITAE</b>	<b>t.e (min)</b>	<b>Overshoot (%)</b>
<b>Ley de control C</b>	1	4350	3, 5	0
<b>Ley de control A</b>	9	8800	3,2	10

**Tabla.7.21. Comparación entre la ley de control C y A prueba 1 vs prueba 9**

Se observa claramente que el ITAE en la ley de control tipo A es aproximadamente el doble que el de la ley tipo C. Esto es debido a que la ley de control tipo C tiene mayor rendimiento por que no solo utiliza la señal de error si no también el estado anterior del setpoint ingresado. La ley de control tipo A es más rápida y debido a esto se hace más inestable el usar esta ley.

A pesar que la ley de control tipo C es un poco más lenta que la tipo A, el desempeño en cualquier condición es mucho mejor que la observada en la tipo A. Por lo tanto se empleará la ley de control tipo C para la prueba final con el microcontrolador.

## 7.5. PRUEBAS FINALES DEL CONTROL ADAPTATIVO UTILIZANDO EL MICROCONTROLADOR

Luego de haber implementado el dispositivo de control como se muestra en la Figura.7.29 se realizaron las siguientes pruebas para mostrar el buen desempeño del mismo.

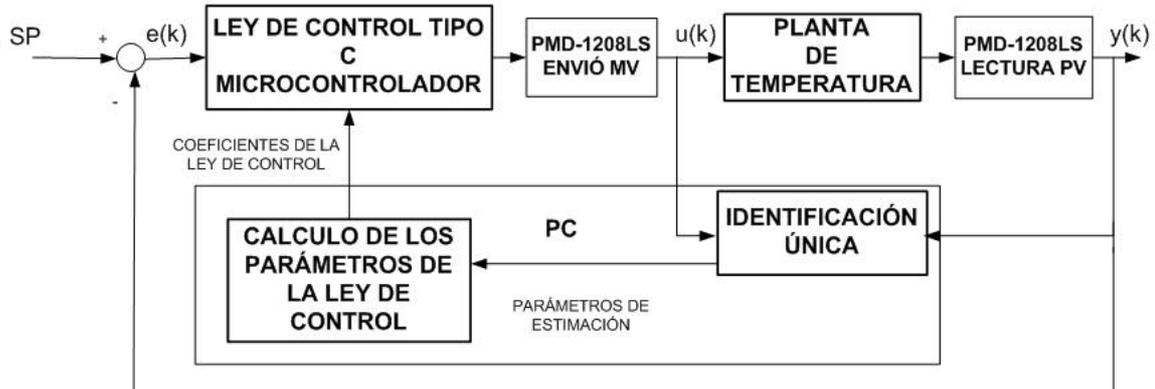


Figura.7.29. Esquema final de funcionamiento con el microcontrolador.

- Control Adaptativo aplicado con la ley de control tipo C sin perturbación.

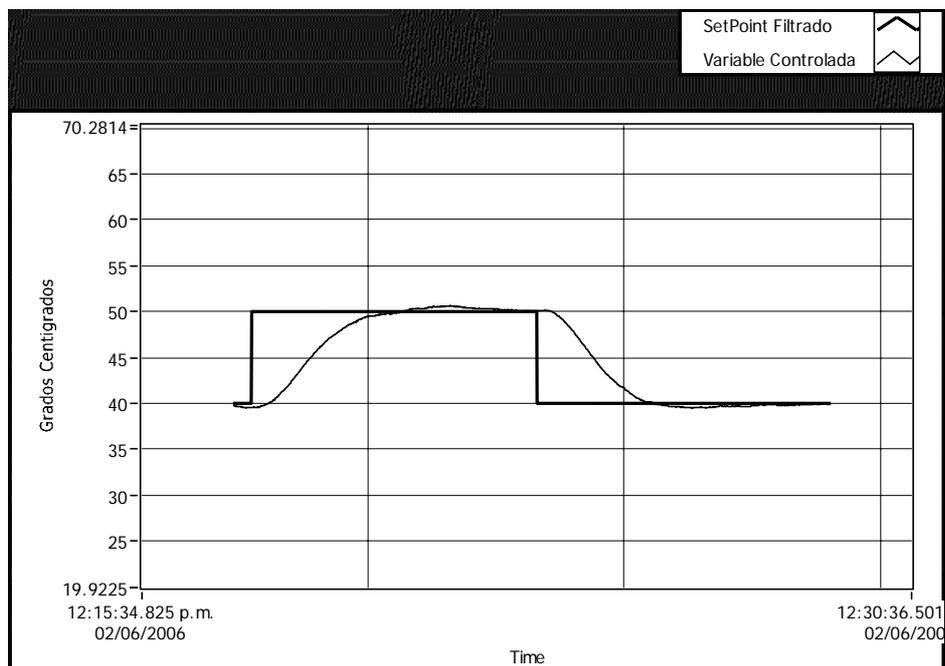


Figura.7.30. Control Adaptativo utilizando la ley de control tipo C implementado en un microcontrolador.

En la Tabla.7.22 se observa los valores obtenidos en la prueba final en condiciones normales.

Variación de Setpoint (°C)	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	4.5	0.5
50 a 40	5.6	1

Tabla.7.22. Tabla de resultados de la planta con control adaptativo utilizando la ley de control tipo C implementado en un microcontrolador

- Control Adaptativo con perturbación, con ventilador abierto y cerrado.

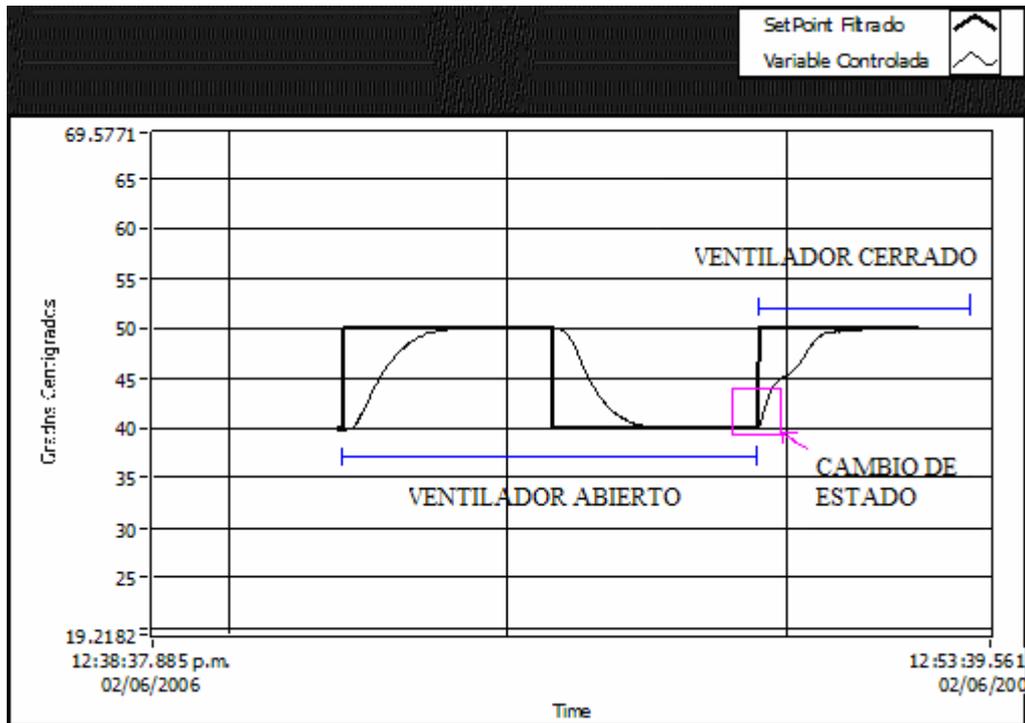


Figura.7.31. Control Adaptativo ante perturbaciones utilizando la ley de control tipo C implementado en un microcontrolador

Variación del SP (°C)	Perturbación (Ventilador de la planta)	Tiempo de establecimiento (min)	Overshoot (%)
40 a 50	cerrado	2.9	0
40 a 50	abierto	3.4	0
50 a 40	abierto	3.4	0

Tabla.7.23. Tabla de resultados con control adaptativo utilizando la ley de control tipo C implementado en un microcontrolador

## Cambio de perturbaciones ventilador cerrado-abierto-cerrado

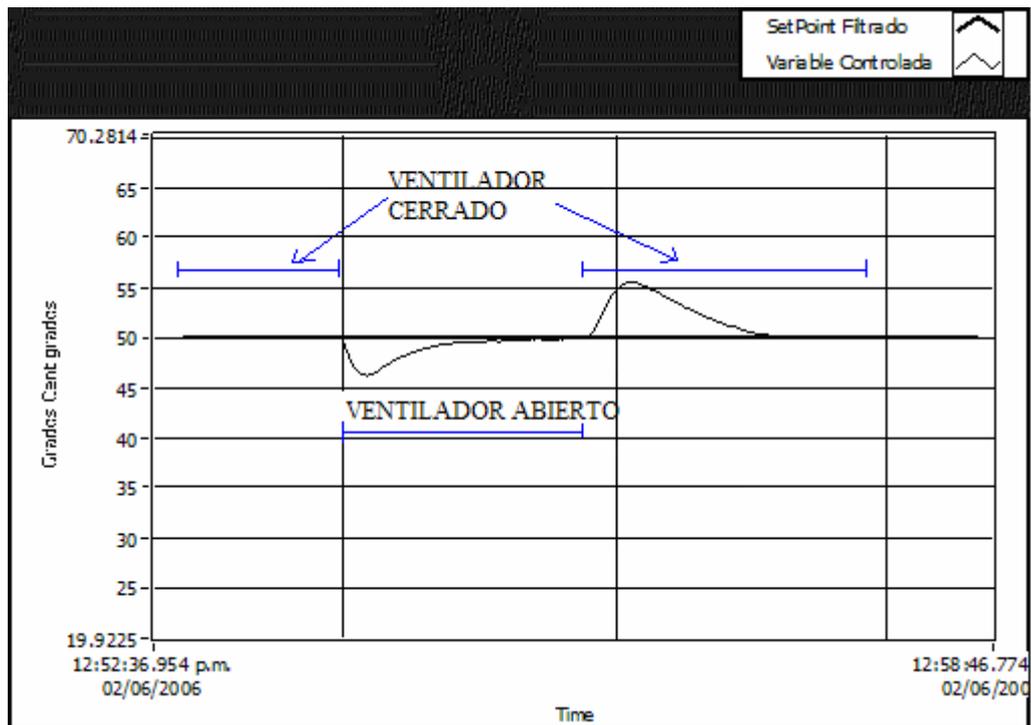


Figura.7.32. Cambio de perturbaciones cerrado-abierto-cerrado

## 7.6. RESULTADOS FINALES

Finalmente al realizar las pruebas con el microcontrolador se puede observar que el comportamiento de la planta es similar al que se obtuvo con las pruebas realizadas mediante software. Esto quiere decir que el objetivo general de este proyecto de grado fue cumplido ya que se diseñó e implementó un prototipo de control adaptativo para una planta de temperatura.

La siguiente tabla enseña la comparación entre el control adaptativo con software y hardware, la cual enseña el buen desempeño de la ley de control tipo C implementada en el microcontrolador.

	PRUEBA	t.e (min)	Overshoot (%)
<b>Control Adaptativo mediante software</b>	1	3,5	0
<b>Control Adaptativo mediante hardware</b>	FINAL	4,5	0,5

Tabla.7.24. Tabla comparativa entre Control Adaptativo realizado por software y por hardware.

## CAPÍTULO 8

### CONCLUSIONES Y RECOMENDACIONES

#### 8.1. CONCLUSIONES

- Mediante la realización de este proyecto se ha diseñado e implementado un sistema de control adaptativo para una planta de temperatura.
- En un sistema de control adaptativo al producirse cambios en la dinámica de la planta los parámetros del controlador del sistema son calculados nuevamente para ajustarse a las nuevas condiciones de funcionamiento.
- La principal ventaja del método de identificación de parámetros por mínimos cuadrados es su inmunidad al ruido. Implementándolo con la técnica de olvido exponencial se consigue también que este método sea capaz de seguir las variaciones de los parámetros siempre y cuando la velocidad de dicha variación sea menor que la velocidad de convergencia de la estimación.
- Empleando la ley de control clase C se logra mejores respuestas que empleando la ley de control clase A. En general la ley de control clase C es más estable. Además implementando un filtro en el set point se mejora la respuesta evitando sobreimpulsos demasiado grandes.
- Empleando la asignación de polos para encontrar los parámetros del controlador se consigue un buen funcionamiento del sistema, además permite modificar su comportamiento cambiando un solo parámetro: la ubicación de un polo triple.
- El microcontrolador PIC18F2550 es una buena solución para implementar la ley de control. Sus características de velocidad y capacidad cumplen con los requerimientos necesarios para su uso en el sistema. Cuenta con periféricos que permiten emplearlo con un mínimo de hardware adicional. Además dispone de un módulo para comunicación USB.
- Tomando el ITAE como medida, se observa que el sistema responde bien ante cambios en la planta. Esto demuestra que la estimación de parámetros funciona

correctamente y permite un buen control de la planta a pesar de que existan cambios en ésta.

- Algunas características de la planta de temperatura, como por ejemplo que su dinámica al calentarse es diferente que su dinámica al enfriarse, la vuelven difícil de controlar con un sistema normal. El control adaptativo implementado logra resolver este inconveniente puesto que calcula los parámetros de la planta constantemente.
- El uso de MatLab para realizar los cálculos facilita en gran medida la programación debido a que este software es una poderosa herramienta para realizar cálculos matemáticos. Además LabView, el programa utilizado para hacer la interface HMI, permite la inclusión de programas de MatLab dentro de su código.
- Una de las dificultades que se presentan en un sistema de control adaptativo es que debido al proceso de estimación el lazo de control puede volverse inestable. El problema puede ser solucionado si se detiene la estimación cuando la salida de la planta alcanza el estado estacionario.

## 8.2. RECOMENDACIONES

- Se debe tener cuidado con la velocidad de transmisión entre el microcontrolador y la PC al usar el emulador RS-232 sobre USB. Una velocidad demasiado alta o demasiado baja puede producir fallos en la comunicación.
- Se debe tener en cuenta que el puerto serial debe ser cerrado en la PC (cerrado vía software) luego de su utilización, pues caso contrario pueden producirse problemas. Como el cierre es relativamente lento, lo mejor es hacerlo al parar la ejecución del programa en el computador.
- Al utilizar MatLab en LabView a través de la función MatLabScript, es recomendable haber hecho previamente pruebas del programa en el entorno del propio MatLab, puesto que así es más fácil la detección de errores.
- El diseño del circuito impreso es crítico, se debe tomar en cuenta las recomendaciones sugeridas en el capítulo 6, de lo contrario se pueden producir problemas de ruido en la placa.

- Se debe filtrar adecuadamente las señales analógicas cuando es necesario hacerlo. En este caso el filtrado de la señal PWM evita que se produzcan armónicos (ruido) asociados con dicho tipo de señal.

## BIBLIOGRAFÍA

- [1] AGUDO, A; MARTINEZ, M, **Identificación y Control Adaptativo**, PEARSON EDUCACION, Madrid 2003, 304 p.
- [2] ASTROM, K; HAGGLUND, T, **PID Controllers: Theory, Design and Tuning**, 1988.
- [3] CUADRADO, Andrés, **Automatización del sistema de entrenamiento de control de procesos CPTS-1 mediante la tarjeta de adquisición de dtos USB PMD-1208LS**.
- [4] FERNÁNDEZ, Adrián; GONZALES ,Humberto, **Diseño e implementación de controladores PID difusos para los modulos experimentales PCT-1 y PCT-2**.
- [5] IOANNOU, Petros, SUN, Jin, **Stable and Robust Adaptive Control**, Prentice Hall PTR , Mayo 1995.
- [6] MEASUREMENT COMPUTING;PMD-1208LS **Personal Measurement Device for Analog and Digital I/O User`s Guide**, 2004.
- [7] OGATA, Katsuhiko, **Ingeniería de control Moderna**, 1993.
- [8] RUIZ, Victor, **Métodos de Sintonización PID de controladores que operan como reguladores**.
- [9] ROJVANIT, Rawin, **Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software**, Microchip Technology Inc, AN956.
- [10] SASTRY, Shankar, BODSON, Marc., **Adaptative Control Stability, Convergence and Robustness**, 1989.
- [11] SLOTINE, Jean-Jaques, **Applied Nonlinear Control**, Prentice Hall, 1990, 457 p.
- [12] [www.ece.cmu.edu/~koopman/des\\_s99/control\\_theory/](http://www.ece.cmu.edu/~koopman/des_s99/control_theory/)
- [13] <http://prof.usb.ve/williamc/PID/Pid.html>

[14] <http://www.eie.fceia.unr.edu.ar/ftp/Control%20II/TP%20control%20Electronicos/tp1/Teor%EDa/Ajuste%20de%20controladores.pdf>

[15] **Universal Serial Bus Common Class Specification**, Systemsoft Corporation - Intel Corporation, 1997.

[16] *PIC18F2455/2550/4455/4550 Data Sheet*.

[17] **Universal Serial Bus Specification**, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, Nec, Philips, Revisión 2.0, Abril 2000.

[18] Datasheet LM78XX.

[19] Datasheet LM358.

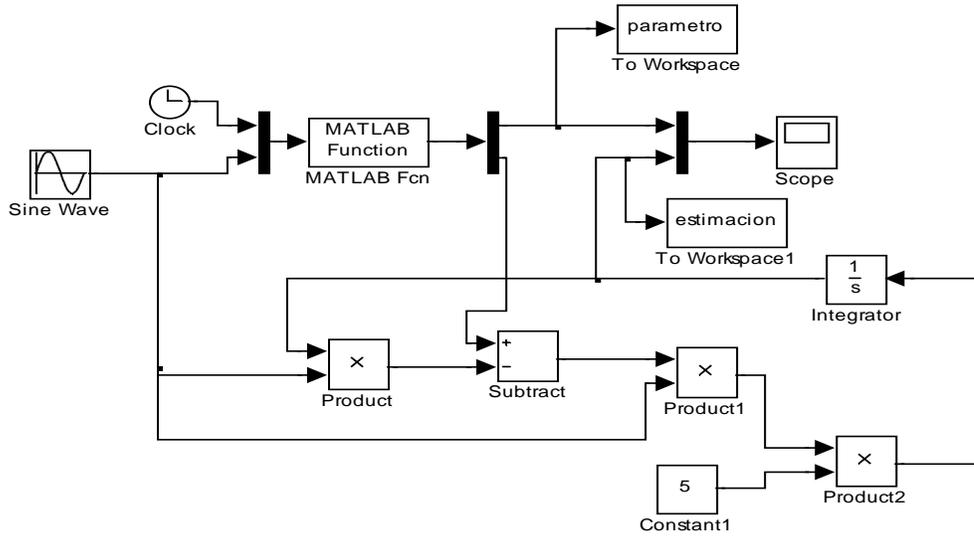
# ***ANEXOS***

## **ANEXO 1**

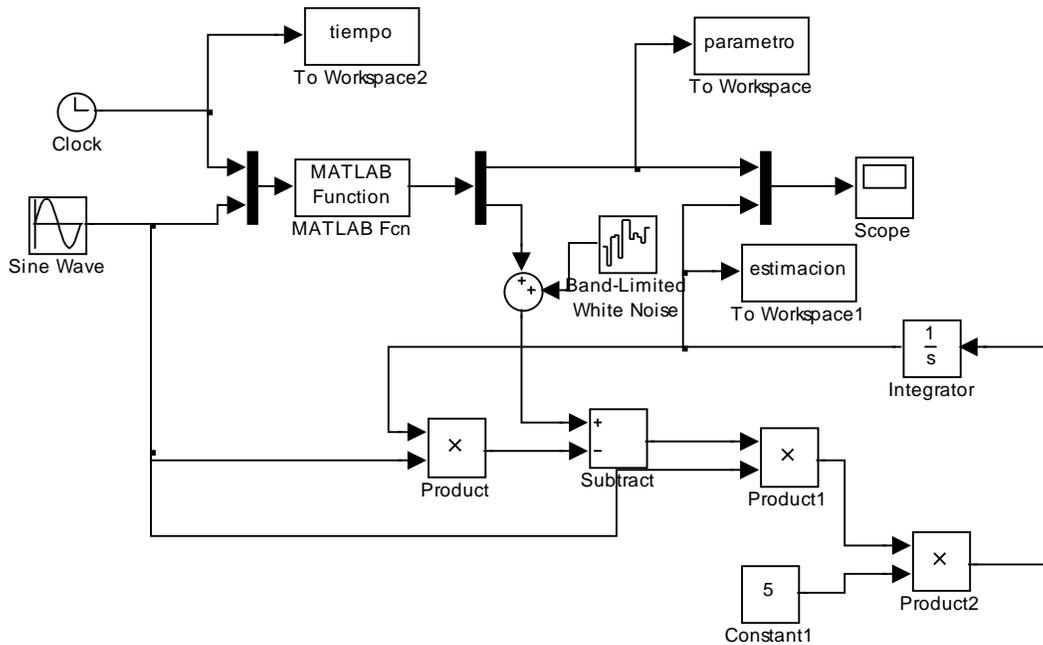
# **DIAGRAMAS DE SIMULACIÓN DE LOS ALGORITMOS DE IDENTIFICACIÓN**

# DIAGRAMAS DE SIMULACIÓN DE LOS ALGORITMOS DE IDENTIFICACIÓN

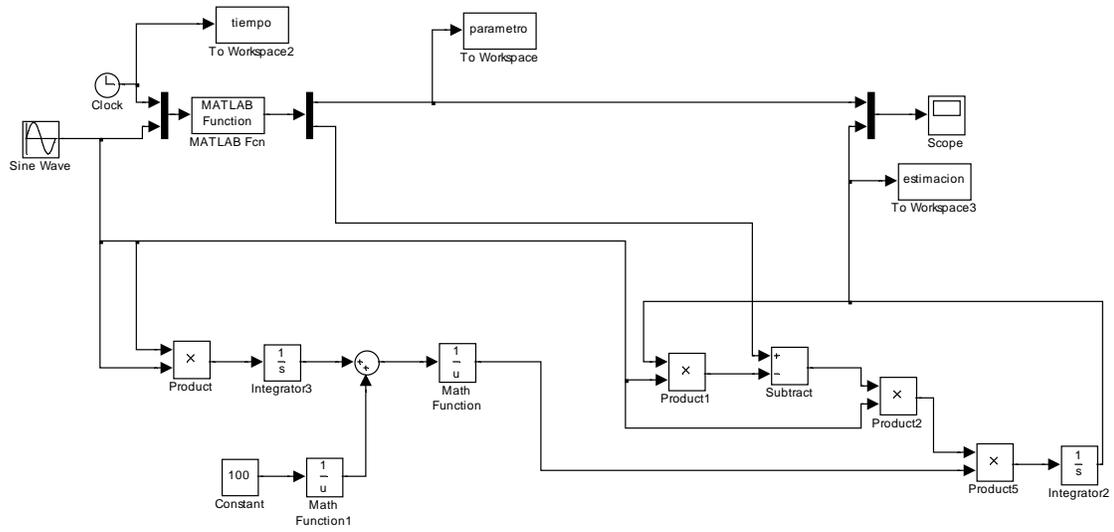
## Método de la Gradiente



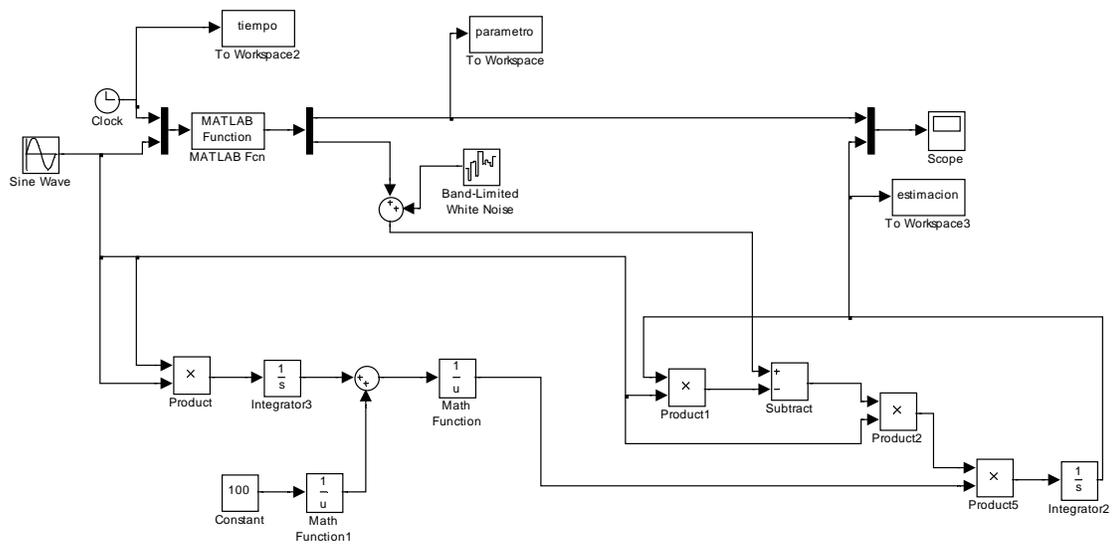
## Método de la Gradiente (con ruido)



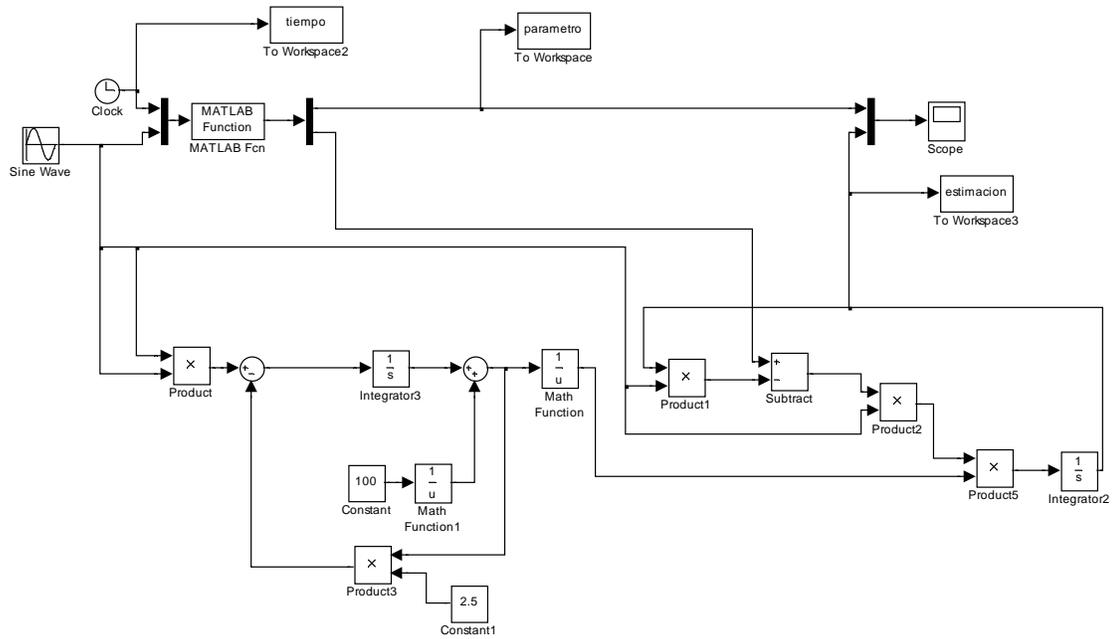
## Método de los mínimos cuadrados



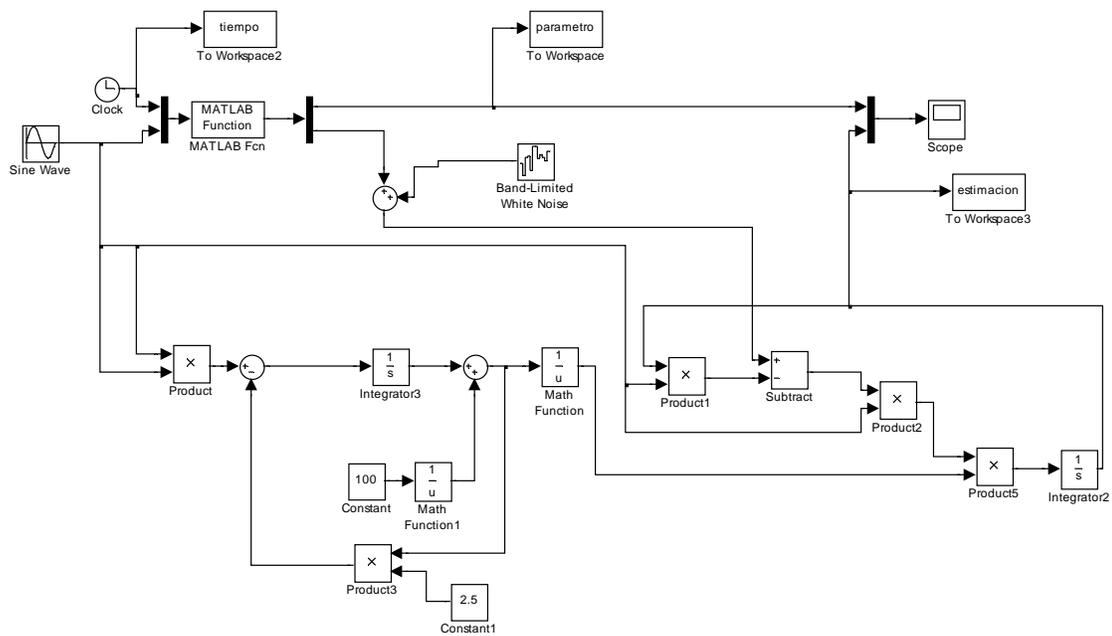
## Método de los mínimos cuadrados (con ruido)



## Método de los mínimos cuadrados con olvido exponencial



## Método de los mínimos cuadrados con olvido exponencial (con ruido)

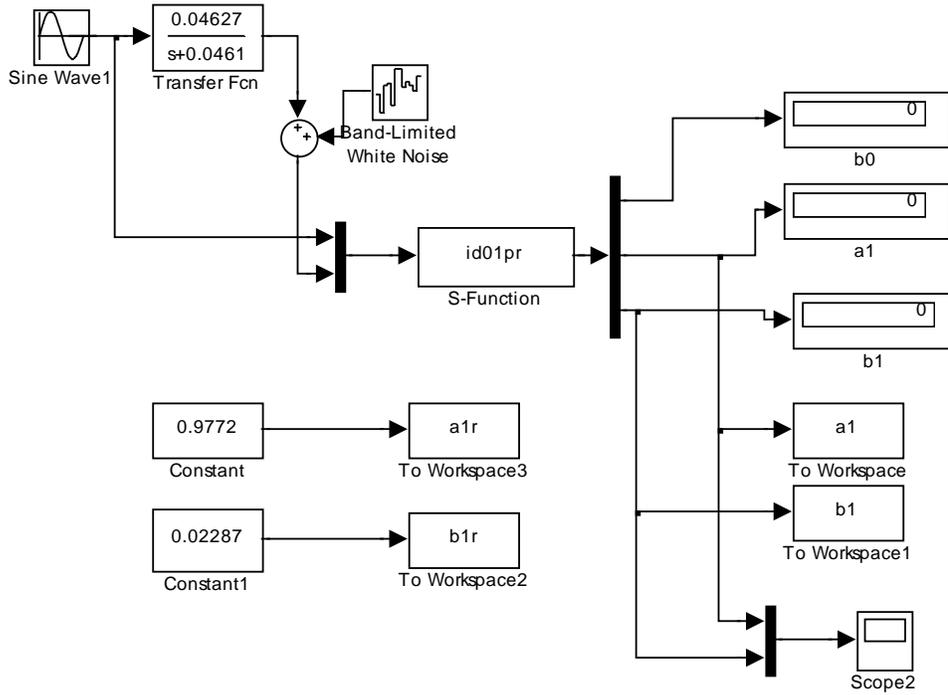


**ANEXO 2**

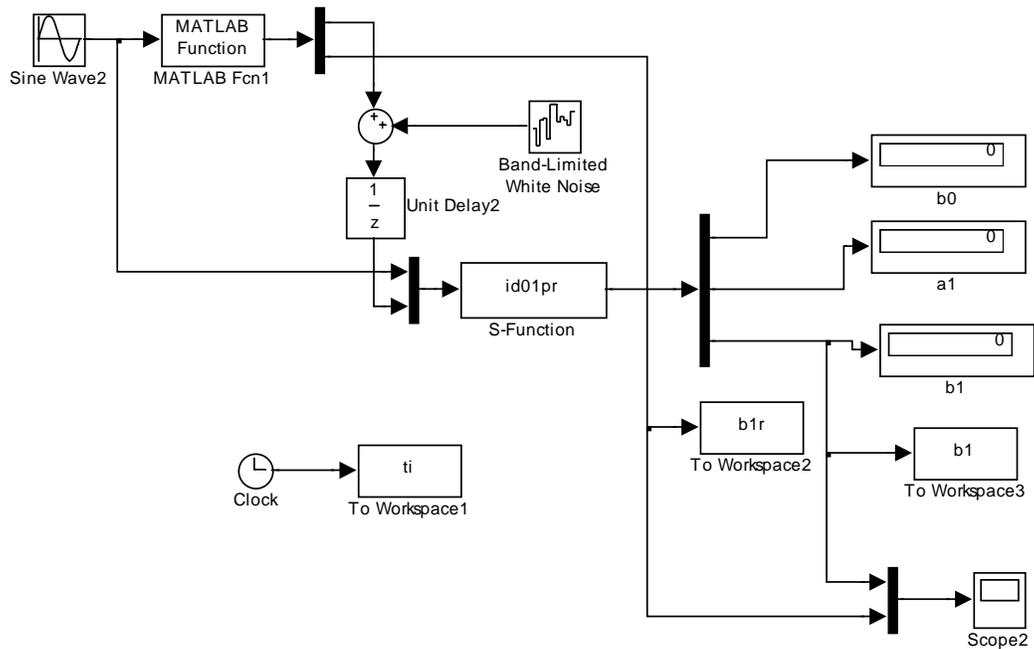
**SIMULACIÓN DEL SISTEMA**

# SIMULACIÓN DEL SISTEMA

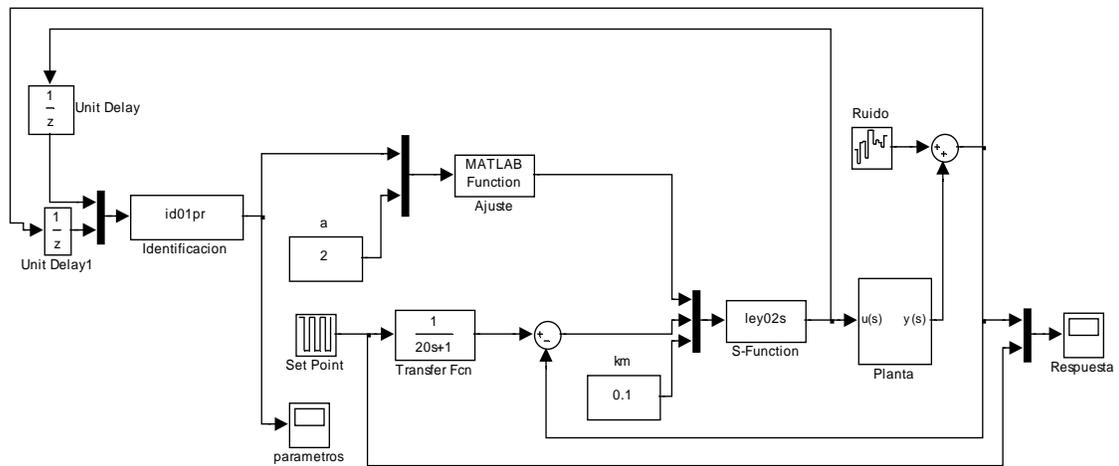
## Estimación de parámetros fijos



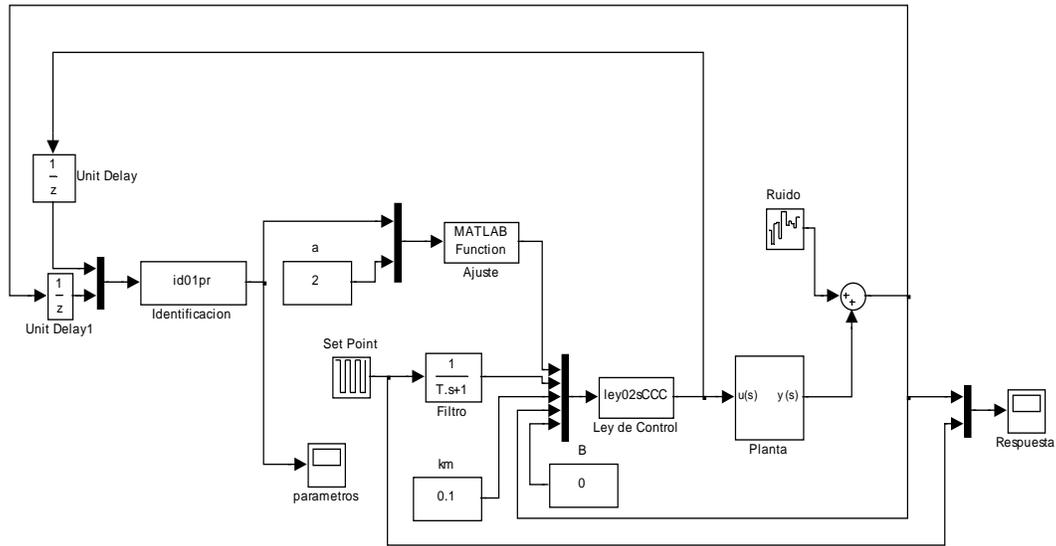
## Estimación de parámetros variables



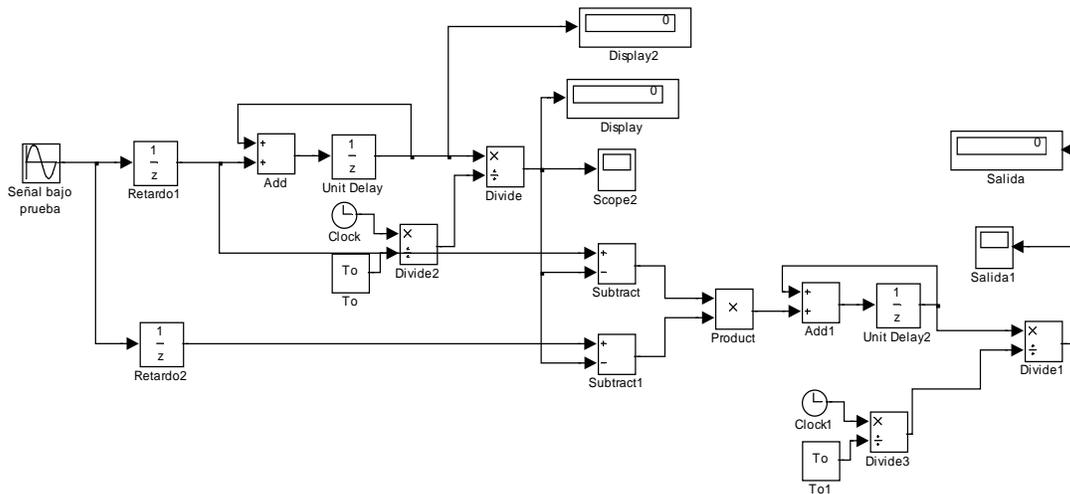
## Sistema completo, ley de control clase A



## Sistema completo, ley de control clase C



## Prueba de la señal de entrada

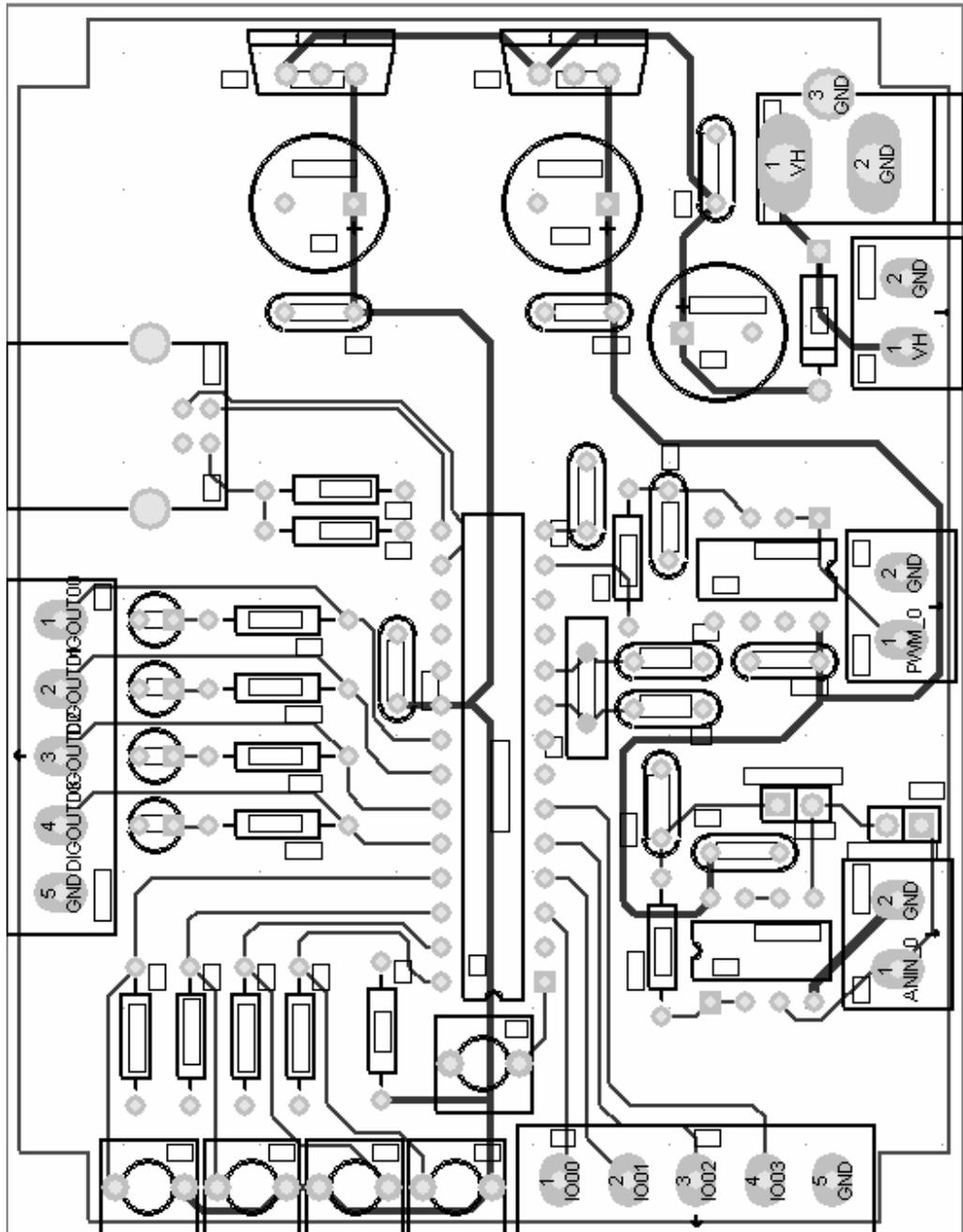


**ANEXO 3**

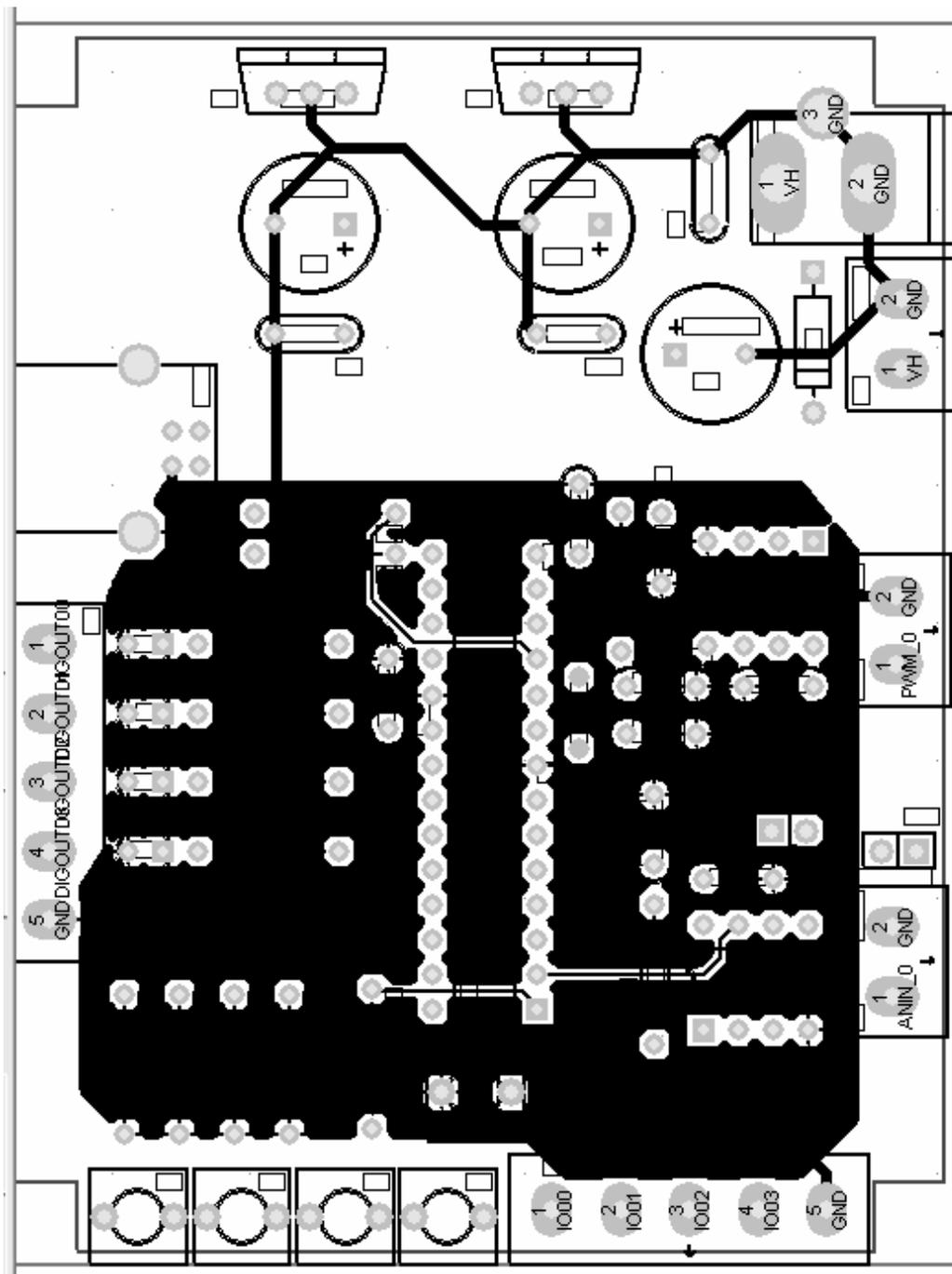
**CIRCUITO IMPRESO**

# CIRCUITO IMPRESO

Cara superior



# Cara Inferior



**ANEXO 4**

**INSTALACIÓN DE SOFTWARE DE LA TARJETA PMD-  
1208LS**

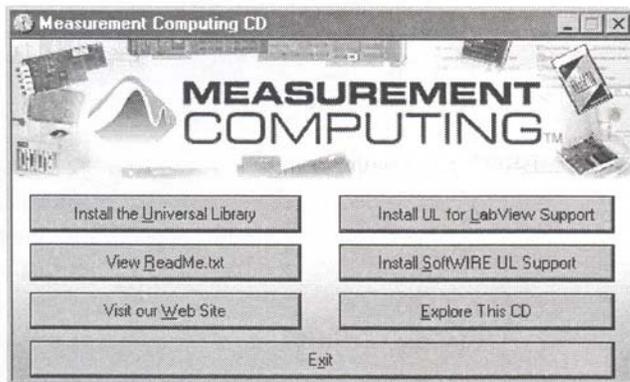
# INSTALACIÓN DE SOFTWARE DE LA TARJETA PMD-1208LS

## Instalación del Software

El CD de instalación de Measurement Computing se debe instalar la Librería Universal, software de configuración de la tarjeta, librería universal para LabVIEW y calibración IstalCal .

Para realizar la instalación se debe realizar los siguientes pasos

Insertar el CD de instalación en el PC. Se muestra la siguiente ventana.



Hacer clic en el botón “Install the Universal Library” y seguir las instrucciones.

Cuando la instalación se ha completado se debe resetear la PC.

Para poder usar la tarjeta con LabVIEW se debe instalar adicionalmente la librería universal para LabVIEW.

Realice los siguientes comandos:

Hacer clic en el botón”Install UL for Labview Support” y seguir las instrucciones.

Cuando haya terminado la instalación resetear la computadora.

**ANEXO 5**

**INSTALACIÓN DE L HARDWARE DE LA TARJETA PMD-  
1208LS**

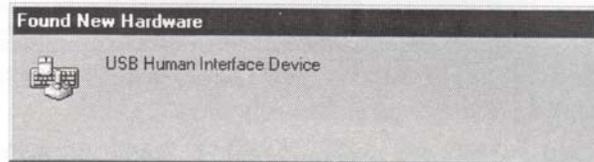
## INSTALACIÓN DE LA TARJETA PMD-1208LS

Con la computadora encendida, conectar el cable USB en el puerto USB del PC o a un hub USB externo que esté conectado a la computadora. Cuando se conecta por primera vez aparece la siguiente ventana:



Primera ventana de instalación

Una segunda pantalla aparece, el sistema operativo es Windows XP como se muestra en la siguiente figura.



Segunda ventana de dialogo de instalación

Cuando esta última ventana se ha cerrado la instalación se ha completado, el led de la tarjeta debe parpadear tres veces y después permanecer encendido ya que esto nos indica que la comunicación entre la tarjeta y el PC está lista.

**ANEXO 6**

**ESPECIFICACIONES TÉCNICAS DE LA TARJETA PMD-  
1208LS**

## ESPECIFICACIONES TÉCNICAS DE LA TARJETA PMD-1208LS

Típicamente para 25°C a menos que se especifique de otra manera

### *Sección Entrada Análoga*

Parámetro	Condiciones	Especificaciones
Tipo Convertidor A/D		Tipo Aproximación Sucesiva
Rango de voltaje de entrada para operación lineal, modo entrada simple	CHx a GND	±10 V máx.
Rango de voltaje de entrada en modo común para operación lineal , Modo Diferencial	CHx a GND	-10 V min, + 20 V máx.
Voltaje de entrada máximo absoluto	CHx a GND	±40 V máx.
Corriente de Entrada(	V <sub>in</sub> = +10 V	70 µA típicamente
	V <sub>in</sub> = 0 V	-10 µA típicamente

Nota 1)	Vin = -10 V	-94 $\mu$ A típicamente
Número de Canales		8 Simples, 4 diferenciales, seleccionables por software
Rangos de Entrada, Modo Simple		$\pm 10$ V, G=2
Rangos de Entrada, Modo Diferencial		$\pm 20$ V, G=1
		$\pm 10$ V, G=2
		$\pm 5$ V, G=4
		$\pm 4$ V, G=5
		$\pm 2.5$ V, G=8
		$\pm 2.0$ V, G=10
		$\pm 1.25$ V, G=16
		$\pm 1.0$ V, G=20
Throughput	Establecido por software	50 S/s
	Escaneo Continuo	1.2 KS/s
	Escaneo Forzado a muestreo de 4K FIFO	8 KS/s
Cola de Canal de Ganancia	Hasta 8 elementos	Canal, rango y ganancia configurables

		por software
Resolución (Nota 2)	Diferencial	12 bits, sin códigos perdidos
	Modo simple	11 bits
Precisión CAL	CAL = 2.5 V	Típico $\pm 0.05\%$ , $\pm 0.25\%$ máx.
Error de Linealidad Integral		Típico $\pm 1$ LSB
Error de Linealidad Diferencial		Típico $\pm 0.5$ LSB
Repetibilidad		Típico $\pm 1$ LSB
Corriente CAL	Fuente	5 mA máx
	Drenaje	20 $\mu$ A min., 200 nA típico
Fuente de Disparo (trigger)	Seleccionable por software	Externo digital: TRIG_IN

Nota 1: La corriente de entrada es una función de un voltaje aplicado en los canales análogos de entrada. Para un voltaje de entrada dado,  $V_{in}$ , el voltaje de fuga es aproximadamente igual a  $(8.181 * V_{in} - 12)\mu A$ .

Nota 2: El convertidor AD7870 solamente retorna 11 bits (0 – 2047 códigos) en modo simple.

#### PRECISIÓN MODO DIFERENCIAL

Rango	Precisión
±20 V	5.1
±10 V	6.1
±5 V	8.1
±4V	9.1
±2.5 V	12.1
±2 V	14.1
±1.25 V	20.1
±1 V	24.1

PRECISIÓN MODO SIMPLE

Rango	Precisión (LSB)
±10 V	4.0

COMPONENTES DE PRECISIÓN MODO DIFERENCIAL – TODOS LOS VALORES SON (±)

Rango	% de Lectura	Ganancia de error(mV)	Offset (mV)	Precisión en FS (mV)
±20 V	0.2	40	9.766	49.766
±10 V	0.2	20	9.766	29.766

±5 V	0.2	10	9.766	19.766
±4V	0.2	8	9.766	17.766
±2.5 V	0.2	5	9.766	14.766
±2 V	0.2	4	9.766	13.766
±1.25 V	0.2	2.5	9.766	12.766
±1 V	0.2	2	9.766	11.766

COMPONENTES DE PRECISIÓN MODO SIMPLE- TODOS LOS VALORES

Rango	% de Lectura	Ganancia de error(mV)	Offset (mV)	Precisión en FS (mV)
±10 V	0.2	20	19.531	39.531

SECCIÓN SALIDA ANÁLOGA

Parámetro	Condiciones	Especificaciones
Tipo de convertidor D/A		PWM
Resolución		10 bits, 1 en 1024
Rango máx de salida		0-5 V
Número de Canales		2 salidas de voltaje
Throughtput	Paso por software	100 S/s en modo canal simple. 50S/s modo canal

		actual
Voltaje de encendido y reset		Inicializa a código 000h
Voltaje máximo ( Nota 3)	Sin Carga	Vs
	Carga de 1 mA	0.99 * Vs
	Carga de 5 mA	0.98 * Vs
Controlador de salida	Cada salida D/A	30 mA
Taza de Slew		0.14 V/mS típico

Nota 3: Vs es ka energía del bus USB + 5 V. El voltaje de salida análogo máximo es igual a Vs sin carga. V es dependiente del sistema y puede ser menos de 5 V.

### MEMORIA NO VOLÁTIL

Tamaño de memoria	8192 bytes		
Configuración de Memoria	Rango de dirección	Acceso	Descripción
	0x0000 - 0x17FF	LECTURA/ESCRITURA	Datos A/D
	0x1800 - 0x1EFF	LECTURA/ESCRITURA	Área de datos del usuario
	0x1F00 - 0x	LECTURA/ESCRITURA	

Datos de calibración

1FEF		
0x1FF0 - 0x1FFF	LECTURA/ESCRITURA	Datos del sistema

## ENERGÍA

Parámetro	Condiciones	Especificación
Suministro de Corriente (Nota 5)		20 mA
Energía disponible de + 5 V USB	Conectado a un hub auto-energizado	4.5 V min, 5.25 V máx
	Conectado a un hub auto-energizado	4.1 V min, 5.25 V máx
Corriente de salida (Nota 7)	Conectado a un hub auto-energizado	450 V min, 500 V máx
	Conectado a un hub auto-energizado	50 V min, 100 V máx

Nota 5: Este es el requisito total de corriente para la tarjeta el cual incluye hasta 5mA para el led de estado

Nota 6: Auto-energizados se refiere a hubs USB y PC con un suministro de energía propio.

Nota 7: Esto se refiere a la cantidad total de corriente que puede ser entregada desde + 5 V del USB, salidas análogas y digitales.

#### GENERAL

Parámetro	Condiciones	Especificación
Error de Temperatura del controlador USB	25°C	±ξ<μ μππ 03
	0 a 70°C	±ξ<μ μππ 05
	-40 a 85°C	±ξ<μ μππ 001
Tipo de dispositivo		USB1.1 baja velocidad
Compatibilidad de dispositivo		USB 1.1, USB 2.0

#### AMBIENTE

Rango de Funcionamiento	Temperatura de	-40 a 85° C
Rango de Almacenamiento	Temperatura de	-40 a 85° C
Humedad		0 a 90% sin condensación

#### MECÁNICO

Dimensiones	79 kk (Largo) x 82 mm ( Ancho) x 25 mm ( Alto)
-------------	--

Longitud del cable USB	3 Metros máx.
Longitud de conexión al usuario	3 Metros máx.

CONECTOR PRINCIPAL Y PINES DE SALIDA

Tipo Conector	Terminales tipo tornillo
Rango del Cable	16 AWG a 30 AWG

MODO DIFERENCIAL, 4 CANALES

PIN	Nombre de la Señal	PIN	Nombre de la señal
1	CH0 IN HI	21	PORT A0
2	CH0 IN LO	22	PORT A1
3	GND	23	PORT A2
4	CH1 IN HI	24	PORT A3
5	CH1 IN LO	25	PORT A4
6	GND	26	PORT A5
7	CH2 IN HI	27	PORT A6
8	CH2 IN LO	28	PORT A7
9	GND	29	GND
10	CH3 IN HI	30	PC + 5 V
11	CH3 IN LO	31	GND
12	GND	32	PORT B0

13	D/A OUT 0	33	PORT B1
14	D/A OUT 1	34	PORT B2
15	GND	35	PORT B3
16	CAL	36	PORT B4
17	GND	37	PORT B5
18	TRIG_IN	38	PORT B6
19	GND	39	PORT B7
20	CAL	40	PORT B8

MODO DIFERENCIAL, 8 CANALES

PIN	Nombre de la Señal	PIN	Nombre de la señal
1	CH0 IN	21	PORT A0
2	CH1 IN	22	PORT A1
3	GND	23	PORT A2
4	CH2 IN	24	PORT A3
5	CH3 IN	25	PORT A4
6	GND	26	PORT A5
7	CH4 IN	27	PORT A6
8	CH5 IN	28	PORT A7

9	GND	29	GND
10	CH6 IN	30	PC + 5 V
11	CH7 IN	31	GND
12	GND	32	PORT B0
13	D/A OUT 0	33	PORT B1
14	D/A OUT 1	34	PORT B2
15	GND	35	PORT B3
16	CAL	36	PORT B4
17	GND	37	PORT B5
18	TRIG_IN	38	PORT B6
19	GND	39	PORT B7
20	CTR	40	PORT B8

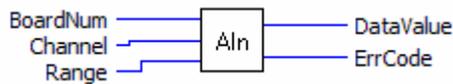
**ANEXO 7**

**INSTRUMENTOS VIRTUALES BÁSICOS DE LABVIEW  
PARA LA PMD-1208LS**

# INSTRUMENTOS VIRTUALES BÁSICOS DE LABVIEW PARA LA PMD-1208LS

Para tener comunicación con la planta de temperatura se requiere la utilización de las entradas análogas de la tarjeta de adquisición de datos y éstas deben ser manejadas por medio de instrumentos virtuales provistos en LabVIEW.

## AIn.VI



## Función AIn y sus argumentos

Lee un valor Análogo desde un canal especificado y este valor es enviado hacia Data Value. Los argumentos que se necesitan son los siguientes.

BoardNum: Es un argumento de entrada, en cual se debe indicar el número de tarjeta, éste número es establecido por el programa *InstalCal*. Puede ser de 0 a 100.

Channel: Especifica el canal A/D a leer.

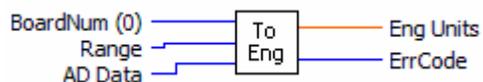
Range: Rango a utilizar en el Terminal de lectura.

Data Value: Es el valor leído en unidades crudas, se recomienda utilizar la función To Eng.

ErrCode: Señal de salida que presenta el estado actual de la tarjeta.

## To Eng.VI

Convierte un valor análogo en unidades crudas a su equivalente en voltaje.



## Función ToEng y sus argumentos

BoardNum: Es un argumento de entrada, en cual se debe indicar el número de tarjeta, éste número es establecido por el programa *InstalCal*. Puede ser de 0 a 100.

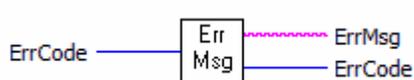
Range: Rango a utilizar en el Terminal de lectura.

AD Data: Valor en unidades cruda de entrada

EngUnits: Valor de salida equivalente a voltaje. El valor está especificado en el rango determinado.

ErrCode: Señal de salida que presenta el estado actual de la tarjeta.

ErrMsg.VI



Función ErrMsg y sus argumentos

Retorna el mensaje de error asociado con un código de error. Si el código de error no es igual a cero, indica que un error ha ocurrido. Este VI convierte un código de error en un mensaje descriptivo.

ErrCode: Señal de entrada retornada por un VI.

ErrMsg: Mensaje de salida retornado.

ErrCode: Código de error o 0 si no lo hubiera.

## ÍNDICE DE FIGURAS

Figura. 1.1. Esquema de Control en Lazo Cerrado con incertidumbres.....	3
Figura. 1.2. Sistema retroalimentado.....	4
Figura. 1.3. Diagrama de un controlador con ganancias ajustables controladas.[5] .....	5
Figura. 1.4. Gain Scheduling .....	6
Figura. 1.5. Control Adaptativo Indirecto [5].....	8
Figura. 1.6. Control adaptativo directo [5] .....	8
Figura. 1.7. Control por modelo de referencia[5].....	9
Figura. 1.8. Control adaptativo por modelo de referencia[5] .....	10
Figura. 1.9. Esquema Self Tuning Regulator [10].....	11
Figura. 1.10. Esquema de control adaptativo usado .....	12
Figura. 2.1. Diagrama de simulación.....	19
Figura. 2.2. Método de la gradiente.....	19
Figura. 2.3. Método de los Mínimos Cuadrados .....	20
Figura. 2.4. Método de los Mínimos Cuadrados con olvido exponencial .....	20
Figura. 2.5. Método de la gradiente.....	20
Figura. 2.6. Método de los Mínimos Cuadrados .....	21
Figura. 2.7. Método de los Mínimos Cuadrados con olvido exponencial .....	21
Figura.3.1. Esquema de lazo cerrado retroalimentado .....	29
Figura.3.2. Características de rendimiento para cada índice.[14] .....	34
Figura.3.3. Esquema margen de fase.....	37
Figura.3.4. Comportamiento de un PID autoajustable con asignación de margen de fase k=50, Km=1.....	37
Figura.3.5. Comportamiento de un PID autoajustable con asignación de margen de fase k=100, Km=1.....	37
Figura.3.6. Comportamiento de un PID autoajustable con asignación de margen de fase k=10, Km=0.1.....	38

Figura.3.7. Esquema Asignación de polos .....	40
Figura.3.8. Comportamiento de un PID autoajustable con asignación de polos $a=5$ y $K_m=0.8$ .....	40
Figura.3.9. Comportamiento de un PID autoajustable con asignación de polos $a=10$ y $K_m=0.5$ .....	40
Figura.3.10. Comportamiento de un PID autoajustable con asignación de polos $a=10$ y $K_m=0.05$ .....	41
Figura.3.11. Comportamiento de un PID autoajustable con asignación de polos $a=10$ y $K_m=0.001$ .....	41
Figura.4.1. Diagrama de bloques del software .....	44
Figura.4.2. Interfaz HMI .....	45
Figura.4.3. Diagrama Jerárquico de los SubVIs usados.....	46
Figura.4.4. SubVI lectura de la tarjeta de adquisición de datos .....	46
Figura.4.5. Diagrama de bloques del SubVI de INDATA .....	47
Figura.4.6. Diagrama del SubVI del Sistema de Control .....	47
Figura.4.7. Diagrama de Bloques del SubVI del Sistema de Control .....	47
Figura.4.8. Diagrama del SubVI de control.....	48
Figura.4.9. Diagrama de Bloques del SubVI de CONTROL .....	48
Figura.4.10. Diagrama de la función arreglo serial .....	49
Figura.4.11. Diagrama de bloques del SubVI de arreglo serial.....	49
Figura.4.12. Diagrama del SubVI arreglo_envio .....	49
Figura.4.13. Diagrama de bloques del SubVI arreglo_envio .....	50
Figura.4.14. Diagrama del SubVI comunicación serial.....	50
Figura.4.15. Diagrama de Bloques del SubVI comunicación serial.....	50
Figura.4.16. Diagrama de la Función de Identificación .....	51
Figura.4.17. Diagrama de bloques del SubVI identificación .....	51
Figura.4.18. Diagrama de la función Calculo ITAE.....	51
Figura.4.19. Diagrama de bloques del SubVI Cálculo ITAE.....	52
Figura.4.20. Diagrama SubVI conversión .....	52
Figura.4.21. Panel de Opciones Avanzadas .....	52
Figura.4.22. Diagrama del SubVI de Opciones Avanzadas .....	53
Figura.4.23. Diagrama del SubVI de Errores.....	53
Figura.4.24. Diagrama del Panel Errores .....	53

Figura.4.25. Diagrama de bloques general .....	54
Figura.4.26. Diagrama del bucle interno .....	55
Figura.4.27. Diagrama de bloques del software del microcontrolador .....	57
Figura.4.28. Diagrama de flujo, comunicación USB .....	61
Figura.4.29. Justificación izquierda y justificación derecha .....	65
Figura.4.30. Diagrama de flujo de la lectura del valor del analógico.....	68
Figura.4.31. Diagrama de flujo de la ley de control .....	70
Figura.4.32. Diagrama General .....	73
Figura. 5.1. Estimación de parámetros REFIL (sin ruido) .....	76
Figura. 5.2. Estimación de parámetros REFIL (con ruido) .....	76
Figura. 5.3. Estimación de $b_I=0,02287+0,025sen(0,01t)$ en ausencia de ruido .....	77
Figura. 5.4. Estimación de $b_I=0,02287+0,025sen(0,01t)$ en presencia de ruido .....	77
Figura. 5.5. Control clase A, sin ruido, sin filtro de SP.....	78
Figura. 5.6. Control clase A, con ruido, sin filtro de SP.....	79
Figura. 5.7. Clase A con filtro SP sin ruido.....	79
Figura. 5.8. Control clase A, con filtro SP, con ruido .....	80
Figura. 5.9. Control clase C, con B=0 y sin filtro de SP .....	80
Figura. 5.10. Control clase C, con B=0, sin filtro de SP y con ruido .....	81
Figura. 5.11. Sistema clase C con filtro de SP .....	81
Figura. 5.12. Control clase C con filtro de SP y con ruido.....	82
Figura. 5.13. Control clase C, con filtro de SP y con B=1 .....	82
Figura. 5.14. Diagrama de Bode del sistema.....	85
Figura. 6.1. Diagrama de bloques del Hardware .....	86
Figura. 6.2. Respuesta del sensor tipo IC del módulo PCT-2 .....	92
Figura. 6.3. Filtro R-C .....	96
Figura. 6.4. Tarjeta USB PMD-1208LS .....	97
Figura. 6.5. Numeración de pines en la PMD-1208LS .....	99
Figura. 6.6. Diagrama de pines.....	99
Figura. 6.7. Diagrama esquemático general .....	101
Figura. 6.8. Esquemático del microcontrolador .....	102
Figura. 6.9. Pulsante de reset.....	103
Figura. 6.10. Señal de pulsante.....	103

Figura. 6.11. Led indicador de salida digital .....	103
Figura. 6.12. Entrada USB .....	103
Figura. 6.13. Sensor de conexión con el host .....	104
Figura. 6.14. Entrada analógica .....	104
Figura. 6.15. Salida analógica .....	105
Figura. 6.16. Fuente de poder .....	106
Figura. 6.17. Conexión del sensor con el plano de tierra .....	108
Figura. 6.18. Ubicación de los elementos en la placa .....	109
Figura.7.1. Esquema Control Adaptativo en línea mediante software ley de control tipo C .....	112
Figura.7.2. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C. ....	113
Figura.7.3. Respuesta de la planta con control adaptativo con cambios de perturbaciones utilizando la ley de control tipo C .....	114
Figura.7.4. Respuesta de control adaptativo cambio de perturbaciones en un mismo punto de operación utilizando ley de control tipo C.....	115
Figura.7.5. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C para $B = 1$ y $B = 0.5$ .....	116
Figura.7.6. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C para $B = 0.5$ y $B = 0$ .....	116
Figura.7.7. Gráfica de respuesta con control adaptativo en condiciones normales utilizando la ley de control tipo C sin mejoras en la ley.....	118
Figura.7.8. Gráfica de respuesta para la ley de control tipo C con perturbación ventilador cerrado y abierto sin ninguna mejora en la ley.....	118
Figura.7.9. Esquema de Self-tuning con identificación única de la planta ley de control tipo C. ....	119
Figura.7.10. Esquema de self-tuning luego de realizar la identificación .....	120
Figura.7.11. Gráfica de resultado con self-tuning en condiciones normales utilizando la ley de control tipo C. ....	121
Figura.7.12. Respuesta de la planta con self-tuning con cambios de perturbaciones utilizando la ley de control tipo C .....	122
Figura.7.13. Gráfica de Self-Tuning ante cambios de perturbación en un mismo Setpoint	123
Figura.7.14. Gráfica de respuesta ley de control tipo C sin perturbación prueba 7 .....	124

Figura.7.15. Gráfica de respuesta para la ley de control tipo C con perturbación ventilador cerrado y abierto.....	125
Figura.7.16. Esquema de control adaptativo mediante software utilizando la ley de control tipo A.....	126
Figura.7.17. Respuesta de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo A .....	127
Figura.7.18. Respuesta de la planta con control adaptativo ante perturbaciones utilizando la ley de control tipo A .....	128
Figura.7.19. Respuesta ante perturbación en un mismo punto de operación con control adaptativo utilizando la ley de control tipo A.....	129
Figura.7.20. Respuesta de la planta con control adaptativo utilizando la ley de control tipo A sin mejoras en la ley de control. ....	130
Figura.7.21. Resultado de la ley de control tipo A con perturbación prueba 10.....	131
Figura.7.22. Esquema de identificación única de la planta y ley de control tipo A mediante software.....	132
Figura.7.23. Esquema de Self-tuning luego de realizar la identificación.....	132
Figura.7.24. Respuesta para la ley de control A para prueba 11 .....	133
Figura.7.25. Respuesta de la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A.....	134
Figura.7.26. Respuesta ante perturbaciones utilizando la ley de control tipo A con Self-Tuning.....	135
Figura.7.27. Respuesta de la planta con self-tuning en condiciones normales utilizando la ley de control tipo A sin parámetros que mejoran la ley de control .....	136
Figura.7.28. Respuesta de la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A sin parámetros que mejoran la ley de control.....	137
Figura.7.29. Esquema final de funcionamiento con el microcontrolador. ....	139
Figura.7.30. Control Adaptativo utilizando la ley de control tipo C implementado en un microcontrolador. ....	139
Figura.7.31. Control Adaptativo ante perturbaciones utilizando la ley de control tipo C implementado en un microcontrolador.....	140
Figura.7.32. Cambio de perturbaciones cerrado-abierto-cerrado .....	141

## ÍNDICE DE TABLAS

Tabla 1.1. Comparación entre Control Adaptativo y Control Robusto [11] .....	6
Tabla. 4.1. Identificadores de parámetros.....	62
Tabla. 5.1. Resumen de resultados .....	83
Tabla. 6.1. Características de la comunicación .....	88
Tabla. 6.2. Datos de Voltaje de salida vs. Temperatura para el sensor tipo IC del módulo PCT-2[4].....	91
Tabla. 6.3. Características del microcontrolador PIC18F2550 .....	93
Tabla. 6.4. Características PIC 18F2550 .....	93
Tabla. 6.5. Características del protocolo USB .....	94
Tabla. 6.6. Características RS-232 simulado.....	95
Tabla. 6.7. Características conversor A/D.....	95
Tabla. 6.8. Características LM7805 y LM7809 .....	105
Tabla. 6.9. Características LM358 .....	107
Tabla. 6.10. Descripción de las pistas realizadas .....	110
Tabla.7.1. Constantes utilizadas para la ley de control tipo C.....	112
Tabla.7.2. Tabla de resultados de la planta con control adaptativo en condiciones normales utilizando la ley de control tipo C.....	113
Tabla.7.3. Tabla de resultados de la planta con control adaptativo con cambios de perturbaciones utilizando ley de control tipo C.....	115
Tabla.7.4. Tabla de resultados con variación de B prueba 3 .....	117
Tabla.7.5. Tabla de resultados prueba 4 .....	118
Tabla.7.6. Tabla de resultados para la prueba 4 con perturbación .....	119
Tabla.7.7. Tabla de resultados con self-tuning en condiciones normales utilizando la ley de control tipo C. ....	121
Tabla.7.8. Tabla de resultados con self-tuning con cambios de perturbaciones utilizando la ley de control tipo C. ....	122
Tabla.7.9. Tabla de resultados prueba 7 .....	124
Tabla.7.10. Tabla de resultados para la prueba 7 con perturbación .....	125
Tabla.7.11. Constantes utilizadas para la ley de control tipo A .....	126

Tabla.7.12. Tabla de resultados para la prueba 7 con perturbación .....	127
Tabla.7.13. Tabla de resultados con control adaptativo ante perturbaciones utilizando la ley de control tipo A .....	128
Tabla.7.14. Tabla de resultados prueba 10 .....	130
Tabla.7.15. Tabla de resultados para la prueba 10 con perturbación .....	131
Tabla.7.16. Tabla de resultados prueba 11 .....	133
Tabla.7.17. Tabla de resultados para la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A .....	134
Tabla.7.18. Tabla de resultados con self-tuning en condiciones normales utilizando la ley de control tipo A sin parámetros que mejoran la ley de control.....	136
Tabla.7.19. Tabla de resultados de la planta con self-tuning ante perturbaciones utilizando la ley de control tipo A sin parámetros que mejoran la ley de control.....	137
Tabla.7.20. Comparación entre Control Adaptativo vs Self Tuning prueba 1 vs prueba 5	138
Tabla.7.21. Comparación entre la ley de control C y A prueba 1 vs prueba 9.....	138
Tabla.7.22. Tabla de resultados de la planta con control adaptativo utilizando la ley de control tipo C implementado en un microcontrolador.....	140
Tabla.7.23. Tabla de resultados con control adaptativo utilizando la ley de control tipo C implementado en un microcontrolador .....	140
Tabla.7.24. Tabla comparativa entre Control Adaptativo realizado por software y por hardware. ....	141

Sangolquí, \_\_\_\_\_

**Elaborado por:**

\_\_\_\_\_  
Ana Verónica Guamán Novillo

\_\_\_\_\_  
Juan Fernando Vásquez Rodríguez

**Autoridades:**

Decano  
Facultad de Ingeniería Electrónica

Secretario Académico  
Facultad de Ingeniería Electrónica

\_\_\_\_\_  
Sr. Ing. Xavier Martínez Carrera  
Tcn. De E.M.

\_\_\_\_\_  
Sr. Dr. Jorge Carvajal