

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

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

**“DISEÑO E IMPLEMENTACION DE UN SISTEMA
CONTROLADOR DE TEMPERATURA PID PARA LA
UNIDAD *AIR FLOW TEMPERATURE CONTROL
SYSTEM* MEDIANTE LA UTILIZACIÓN DE LA
HERRAMIENTA RTW (REAL TIME WORKSHOP) DE
MATLAB.”**

BYRON ALEJANDRO ACUÑA ACURIO

OSWALDO ALEXANDER IBARRA JÁCOME

SANGOLQUI – ECUADOR

2010

CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado **“DISEÑO E IMPLEMENTACION DE UN SISTEMA CONTROLADOR DE TEMPERATURA PID PARA LA UNIDAD AIR FLOW TEMPERATURE CONTROL SYSTEM MEDIANTE LA UTILIZACIÓN DE LA HERRAMIENTA RTW (REAL TIME WORKSHOP) DE MATLAB.”** Ha sido desarrollado en su totalidad por el señor BYRON ALEJANDRO ACUÑA ACURIO con CI: 1718688813 y el señor OSWALDO ALEXANDER IBARRA JÁCOME con CI: 1719535427 bajo nuestra dirección.

Atentamente;

Ing. Víctor Proaño

DIRECTOR

Ing. Paul Bernal

CODIRECTOR

AGRADECIMIENTO

A mis padres, los cuales han caminado siempre a mi lado con infinito amor y dedicación.

A mi hermano cuyo ejemplo de sacrificio y dedicación me ha mostrado que los milagros si existen y que es uno quién los construye solo con la ayuda de Dios.

A mis ti@s quienes siempre han trabajado de forma honrada por un mejor futuro.

A mis prim@s con quienes siempre hemos sido amigos y hermanos.

A mis Abuelit@s quienes son un ejemplo de amor y abnegación.

A Dianita mi pequeña princesita por su gran amor y comprensión.

A tod@s mis amig@s por todo su apoyo.

A los profesores y personal de la ESPE, en especial al Ing. Víctor Proaño y al Ing. Paúl Bernal por toda su colaboración.

BYRON ALEJANDRO

AGRADECIMIENTO

Luego de un largo caminar en el sendero de la formación personal y profesional que la vida universitaria me ha brindado, me permito lleno de fervor rendir mis más profundos agradecimientos a todos aquellos que con su apoyo me han brindado diferentes virtudes para alcanzar una meta que hoy por hoy se ha cumplido con arduo pero satisfactorio trabajo.

Primeramente a Dios y a la Santísima Virgen Del Cisne por la energía y fuerzas que me han dado para poder llegar al final de mis estudios y completar esta etapa de mi vida.

Con mucho cariño a mis padres Msc. Dr. Oswaldo Ibarra y a Esp. Lcda. Teresa Jácome que me dieron la vida y han estado conmigo en todo momento, gracias por todo mamá y papá por darme una carrera para mi futuro, por creer en mí, por enseñarme con el ejemplo la honestidad, la perseverancia, la paciencia, la fortaleza, el amor y el pensamiento que rige mi vida el cuál es “El que tiene a Dios, lo tiene todo”.

A mi hermana Msc. Ing. Gabriela Ibarra que con peleas y disgustos ha caminado conmigo brindándome su amor de hermana y la enseñanza de un carácter fuerte y firme apegado siempre a la verdad.

A mi mamá Antuquita que junto al Padre Gerardo Barriga Naranjo+ me enseñaron desde pequeño a ser un hombre de bien.

A mi abuelita María, a mis Tíos Juan y Miguel inspiraciones para decidir mi futuro, a mis Tías Cristina y Verónica, a mis primos en especial a mi prima Silvana, y en general a toda mi familia por el apoyo y bien intencionado interés por mí.

A Daniela, que con su amor, comprensión y cariño ha caminado a mi lado brindándome lo mejor de sí.

A los profesores y personal de la ESPE, en especial al Ing. Víctor Proaño y al Ing. Paúl Bernal por que con su paciencia y gran espíritu de colaboración ayudaron a que se cumpla esta meta.

Y por último a mi fiel mascota Tobías que con su forma de ser me ha acompañado de día y de noche regalándome su alegría y compañía.

OSWALDO ALEXANDER

DEDICATORIA

El presente logro se lo dedico a mis padres y a mi hermano quienes desde muy joven me enseñaron que no hay que tener miedo al mundo, a confiar en Dios, a tener la tranquilidad de que todo va a estar bien, mostrándome que las habilidades no son nada sin las destrezas y que ellas solo se las obtienen ganando y perdiendo pequeñas batallas, con miras siempre a conseguir un sueño ser un gran ser humano como mis padres, mi hermano y mi familia, quienes siempre serán mis únicos héroes.

BYRON ALEJANDRO

“La gran victoria que hoy parece fácil fue el resultado de pequeñas victorias que pasaron desapercibidas”

Paulo Coelho

DEDICATORIA

Esta Tesis se la dedico a mis padres por toda la gratitud y amor que les tengo, porque sé que cada logro alcanzado no solamente es mío sino también es de ellos.

OSWALDO ALEXANDER

Mucho tiempo después pude entender que al igual que Santiago en el Alquimista, *“Cada persona posee su Leyenda Personal, que cuando una persona desea realmente algo, el Universo entero conspira para que pueda realizar su sueño, basta con aprender a escuchar los dictados del corazón y a descifrar el lenguaje que está más allá de las palabras”*

Paulo Coelho - El Alquimista

ÍNDICE DE CONTENIDO

CAPÍTULO 1.....	1
INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	1
1.1.1 Objetivo General.....	1
1.1.2Objetivos Específicos.....	1
1.2 DEFINICIÓN DEL PROYECTO.....	1
1.2.1 Antecedentes.....	1
1.2.2 Justificación e Importancia.....	2
1.2.3 Alcance del Proyecto.....	3
CAPÍTULO 2.....	4
CARACTERÍSTICAS DEL SISTEMA A IMPLEMENTAR.....	4
2.1 ESTUDIO Y ANÁLISIS DE LA HERRAMIENTA RTW (REAL TIME WORKSHOP) DE MATLAB®.....	4
2.1.1 Introducción.....	4
2.1.2 Arquitectura del Real Time Workshop.....	4
2.1.3 Algoritmo del Real Time Workshop.....	6
2.1.4 Proceso del Real Time Workshop.....	6
2.1.5 Pasos En El Desarrollo De Un Algoritmo Para Control En Tiempo Real.....	9
2.1.6 Etapas En La Generación RTW.....	10
2.1.7 Archivos Generados Por RTW.....	10
2.1.8 Estructuras Declaradas por el Modelo RTW.....	11
2.1.9 Código Generado en “C” por el RTW.....	12
2.1.10 Ejecución del Modelo en Tiempo Real.....	13
2.1.10.1 Integración de Estados Continuos.....	14
2.2 CARACTERÍSTICAS DEL REAL TIME WINDOWS TARGET Y DE LA TARJETA DE ADQUISICIÓN DE DATOS.....	15

2.2.1 Real Time Windows Target (RTWT).....	15
2.2.1.1 Características Generales Del RTWT.....	15
2.2.1.2 Características Principales del RTWT.....	15
2.2.2 Tarjeta de Adquisición de Datos National Instruments PCI 6221.....	17
2.2.2.1 Información General.....	17
2.2.2.2 Resumen de Especificaciones.....	18
2.2.2.3 Conector SCB-68.....	20
2.2.2.4 Cable NI SHC68-68-EPM.....	21
2.2.2.5 Distribución de Pines de la Tarjeta de Adquisición de Datos NI PCI 6221.....	22
2.3 CARACTERÍSTICAS DE LA UNIDAD AIR FLOW TEMPERATURE CONTROL SYSTEM.....	23
CAPÍTULO 3.....	25
DISEÑO DEL SISTEMA DE CONTROL RTW.....	25
3.1 DISEÑO DEL HARDWARE.....	25
3.1.1 Características de la PC.....	26
3.1.2 Descripción del diagrama de conexiones.....	26
3.1.3 Diagrama de bloques del sistema de control.....	28
3.2 DISEÑO DEL SOFTWARE.....	29
3.2.1 Instalación y Configuración del Real Time Windows Target.....	30
3.2.1.1 Real Time Kernel.....	30
3.2.1.2 Instalación Del Kernel.....	31
3.2.2 Ingreso a SIMULINK®.....	35
3.2.3 Utilización De La Librería Real Time Windows Target (RTWT).....	36
3.2.4 Instalación y reconocimiento de la tarjeta de adquisición de datos.....	39
3.2.5 Configuración de parámetros del RTW.....	52
3.2.6 Configuración de las opciones de los elementos de visualización.....	57
3.2.7 Ejecución de la aplicación en tiempo real.....	67
3.2.8 Obtención de resultados.....	70
3.2.9 Desinstalación Del Real Time Kernel.....	70

CAPÍTULO 4.....	71
APLICACIÓN DEL SISTEMA DE CONTROL RTW EN UN CONTROL PID.....	71
4.1 ESPECIFICACIONES DEL PROBLEMA.....	71
4.2 MODELO DE LA PLANTA.....	74
4.2.1 Interpretación Física De La Planta	74
4.2.2 Análisis De La Planta Mediante Las Ecuaciones De Transferencia De Calor.....	77
4.2.3 Análisis De La Planta Mediante Las Ecuaciones De Transferencia De Calor Considerando Las Variaciones De Caudal Generadas Por Las Perturbaciones.....	80
4.2.4 Identificación De La Planta.....	83
4.2.4.1 Adquisición de Datos.....	83
4.2.4.2 Identificación Del Modelo Matemático De La Planta.....	92
4.2.5 Identificación De Los Modelos Matemáticos Para Cada Perturbación.....	96
4.2.6 Diseño Del Controlador PID.....	98
4.2.6.1 Sintonización Por Medio Del Segundo Método De Zeigler – Nichols.....	98
4.3 DESARROLLO PROGRAMAS EN SIMULINK.....	99
4.3.1 Identificación De La Planta.....	99
4.3.2 Control PID Experimental.....	99
4.3.3 Simulación Del Control PID.....	100
4.3.4 Descripción De La HMI.....	100
CAPITULO 5.....	101
PRUEBAS Y RESULTADOS.....	101
5.1 PRUEBAS.....	101
5.1.1 Pruebas Del Objeto De Experimentación Y Del Modelo De Simulación.....	101
5.2 RESULTADOS.....	112

CAPITULO 6.....	117
CONCLUSIONES Y RECOMENDACIONES.....	117
6.1 CONCLUSIONES.....	117
6.2 RECOMENDACIONES.....	118
REFERENCIAS BIBIOGRÁFICAS.....	20
ANEXOS.....	121
ANEXO 1 GLOSARIO.....	122
ANEXO 2 MANUAL DE USUARIO.....	133
ANEXO 3 PROGRAMA PARA REALIZAR LA IDENTIFICACIÓN DE LA PLANTA (Modelamiento.m).....	145
ANEXO 4 PROGRAMA PARA REALIZAR LA COMPARACIÓN DE LAS GRAFICAS DE LOS RESULTADOS OBTENIDOS (Comparacion.m).....	148
ÍNDICE DE FIGURAS.....	150
ÍNDICE DE TABLAS.....	154

CAPÍTULO 1

INTRODUCCIÓN

1.1. OBJETIVOS

1.1.1 Objetivo General

Realizar el diseño e implementación de un sistema controlador de temperatura PID para la unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM* mediante la utilización de la herramienta RTW (*Real Time Workshop*) de MATLAB®.

1.1.2 Objetivos Específicos

1. Estudiar y analizar la herramienta RTW de MATLAB®.
2. Verificar el adecuado funcionamiento de la tarjeta de adquisición de datos con MATLAB®.
3. Modelar la planta *AIR FLOW TEMPERATURE CONTROL SYSTEM* y el controlador PID como bloques en SIMULINK®.
4. Verificar el funcionamiento de la interfaz de adquisición de datos usando el software controlador creado en MATLAB®.
5. Describir y verificar mediante la implementación el adecuado funcionamiento del sistema controlador de temperatura PID para la unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM* mediante el software creado en MATLAB® específicamente en la herramienta SIMULINK®.

1.2 DEFINICIÓN DEL PROYECTO

1.2.1 Antecedentes

En el modelamiento y diseño de los sistemas de control se utiliza ampliamente el software MATLAB® como simulador por su gran capacidad de cálculos y versatilidad de librerías. Actualmente este software se encuentra utilizado como material didáctico para la enseñanza de los sistemas de control en las Universidades y como herramienta científica de diseño para prototipos.

En el aprendizaje de modelamiento de sistemas de control, siempre ha existido el inconveniente que un controlador diseñado, queda teóricamente explicado y susceptible a errores porque su implementación resulta cara y demorosa, por lo que en muchas ocasiones, muchos diseños terminan siendo teóricos. El estudiante no logra ver los reales problemas que ocasionaría un controlador mal diseñado. Se suelen realizar proyectos para implementar el controlador diseñado pero el proyecto tiene un alto componente de las asignaturas de electrónica que en muchos casos no son del total dominio del alumno.

La existencia del laboratorio de servo - mecanismos con algunos prototipos, en el Departamento de Ciencias de la Energía y Mecánica de la ESPE refleja la necesidad, funcionalidad y aplicabilidad de poseer una interacción en tiempo real entre un controlador diseñado en MATLAB® y una planta física.

1.2.2 Justificación e Importancia

En la cátedra de sistemas de control se ha visto la necesidad que el estudiante tenga una interacción más cercana con los comportamientos y respuestas que pueden sufrir una planta y su estrecha relación con las curvas de respuesta. Por mucho tiempo MATLAB® se utilizó simplemente como una herramienta de modelamiento y para la implementación del controlador se la realizaba externamente y siempre existían variaciones en los resultados obtenidos. Con la llegada de la herramienta RTW de MATLAB® se ha logrado comprobar al instante si un controlador está correctamente diseñado y se puede verificar en tiempo real el efecto en la planta de cambios en los parámetros del controlador ampliando notablemente la interacción del estudiante y el aprendizaje de esta cátedra.

De acuerdo a los antecedentes anteriormente expuestos se considera oportuno desarrollar un sistema computarizado en el cual se pueda realizar pruebas en tiempo real con los controladores diseñados en la cátedra.

El sistema diseñado permitirá diseñar el controlador en el entorno de SIMULINK® pero se ejecutará en una planta real disponible en el laboratorio específicamente en la unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM*.

Un sistema como el indicado será de mucha utilidad por cuanto permitirá el diseño de controladores de todo tipo como control difuso, control óptimo, control digital, etc. aprovechando las capacidades de MATLAB[®] y aplicarlo en forma rápida a la planta.

1.2.3 Alcance del Proyecto.

El presente proyecto de tesis abarca el estudio, diseño e implementación del código en MATLAB[®] para la unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM* (control de temperatura) en tiempo real mediante la herramienta RTW.

Se utilizará una tarjeta de adquisición y generación para interconectar la planta real con el sistema SIMULINK[®] de MATLAB[®].

Se realizará el acondicionamiento de señales para la conexión de la tarjeta con el proceso. Se configurará la tarjeta de adquisición y se realizará los archivos .m y .mdl de MATLAB[®] para facilitar el uso de la herramienta.

CAPÍTULO 2

CARACTERÍSTICAS DEL SISTEMA A IMPLEMENTAR

2.1 ESTUDIO Y ANÁLISIS DE LA HERRAMIENTA RTW (REAL TIME WORKSHOP) DE MATLAB®

2.1.1 Introducción

Real Time Workshop (RTW) es una parte integral del entorno de Simulink® que genera y ejecuta archivos en código C para desarrollar y probar los algoritmos modelados en Simulink® y códigos en MATLAB®. El código en C resultante puede ser utilizado en aplicaciones en tiempo real porque se basa en una programación por interrupciones.

Cabe señalar que el Real Time Workshop es un componente utilizado por el Real Time Windows Target (RTWT), el que se describirá más adelante en detalle, el cual está diseñado para operar con una gran flexibilidad y rapidez durante la creación de modelos, esta ventaja permite modificar parámetros durante la ejecución en tiempo real, incrementando el tamaño de código generado. El RTWT se considera como la tarjeta de adquisición de datos en el software Simulink® ya que posee bloques de entradas analógicas, salidas analógicas, contadores, temporizadores, entre otras que mediante una tarjeta de adquisición de datos trabaja en tiempo real.

2.1.2 Arquitectura del Real Time Workshop

El RTW trabaja sobre la plataforma de MATLAB® y Simulink® interpretando los códigos .m o los diagramas de bloques de Simulink® .mdl para generar un modelo en tiempo real .rtw el cual mediante un compilador (Visual C++) y un programa identificador del lenguaje del compilador (TLC) genera los archivos .c con sus respectivas librerías .h

para de esta manera generar un ejecutable que pueda ser interpretado fácilmente en lenguaje de bajo nivel y que a su vez sea compatible con la interfaz gráfica de Simulink® en modo externo, el archivo ejecutable .exe generado por la herramienta RTW se forma a partir del make file que hayamos seleccionado y su función principal es poder ejecutarse en una plataforma de Windows o Unix independientemente y que pueda ser llamada y controlada por MATLAB®, este archivo es llamado durante la ejecución del modelo en tiempo real para trabajar conjuntamente con la interfaz de usuario, como lo podemos apreciar en la Figura 2.1 que se presenta a continuación [1].

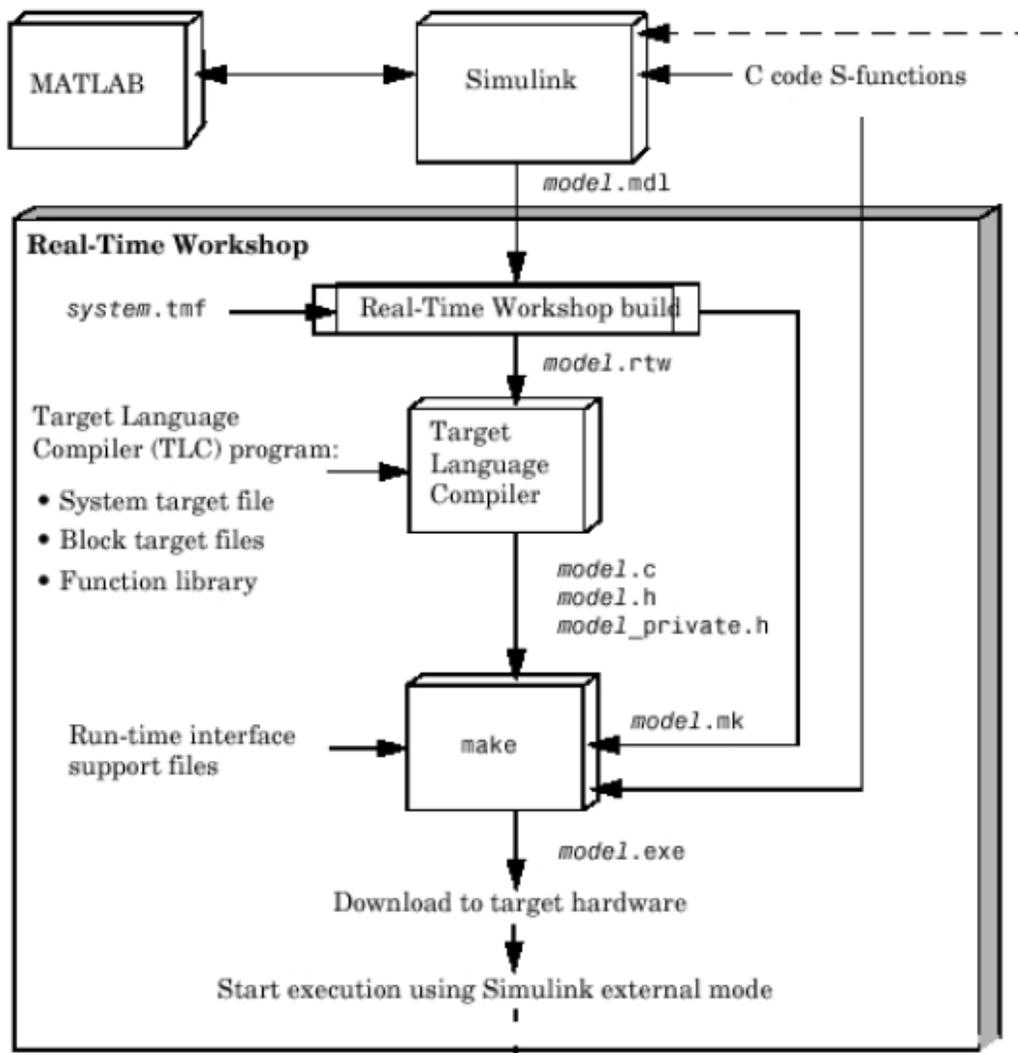


Figura 2.1 Arquitectura RTW

2.1.3 Algoritmo del Real Time Workshop

El Real Time Workshop a parte de generar múltiples archivos genera relaciones de dependencia entre estos archivos y las herramientas y funciones de MATLAB[®], trabajando simultáneamente los componentes de MATLAB[®] y los archivos generados en C con las interrupciones del computador provocando de esta manera el trabajo en tiempo real como se muestra a continuación en la Figura 2.2.

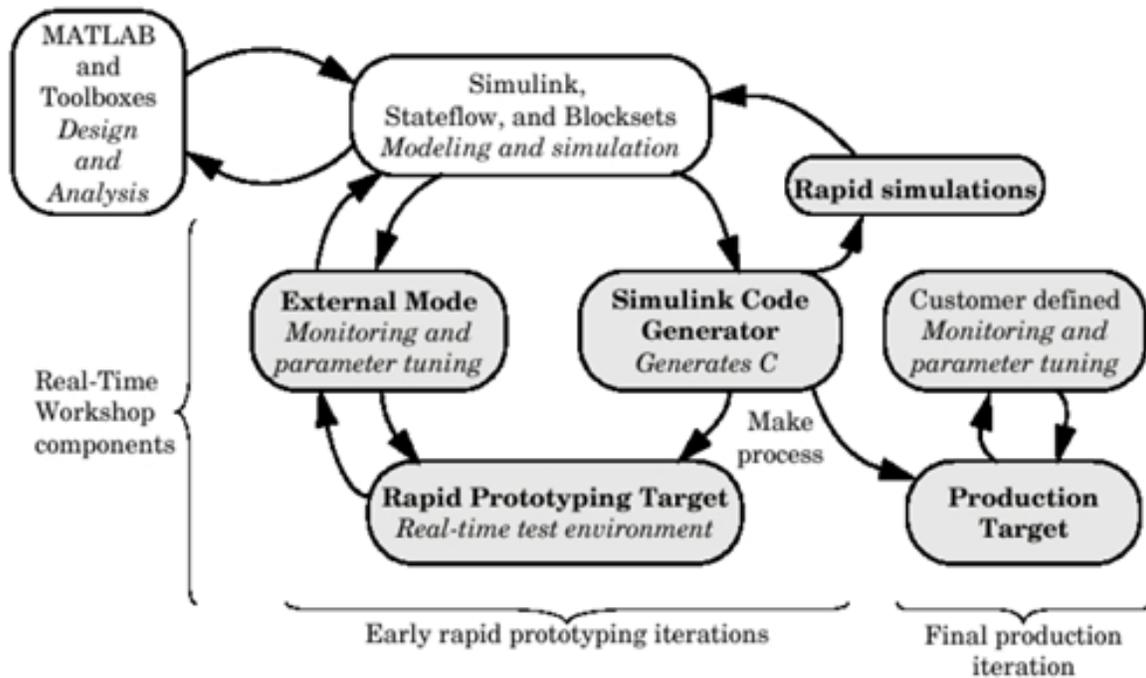


Figura 2.2 Algoritmo RTW

2.1.4 Proceso del Real Time Workshop

Como se mencionó anteriormente el Real Time Workshop sirve de intérprete o traductor entre MATLAB[®] y un compilador en lenguaje C por lo cual el parámetro más importante de configuración es el compilador el cual debe ser compatible con el TLC de

MATLAB® para poder generar los archivos necesarios para una aplicación en tiempo real como se puede apreciar en la Figura 2.3 que se presenta a continuación.

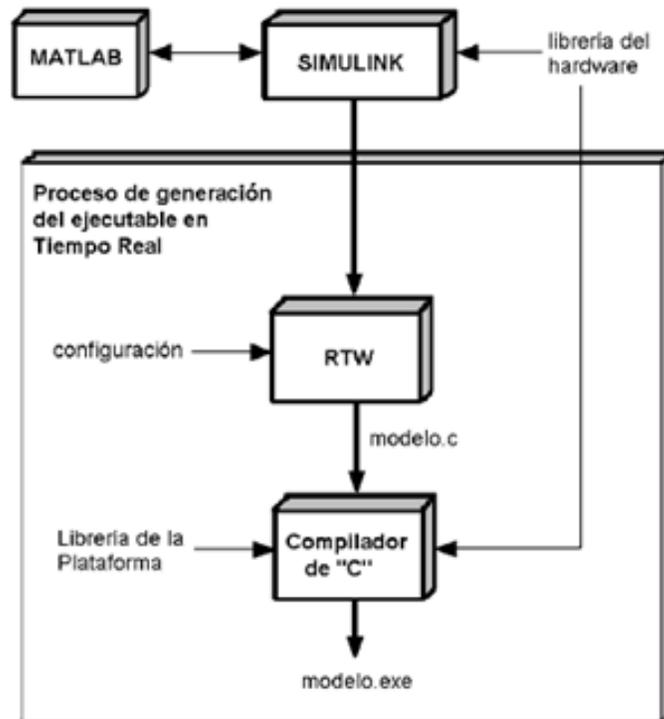


Figura 2.3 Proceso RTW

El Real Time Workshop presenta algunos sistemas identificadores de archivos o identificadores de lenguajes de compiladores TLC para diferentes aplicaciones en tiempo real como por ejemplo para plataformas UNIX, para aplicaciones en Visual C++, funciones S, etc, en general para las aplicaciones en tiempo real con MATLAB® y Simulink® se utilizará siempre el archivo identificador de sistema `rtwin.tlc` el cual nos permitirá realizar aplicaciones en tiempo real genéricas para Pc o UNIX, del mismo modo el RTW posee los archivos `Template Makefile` que generarán el archivo en tiempo real utilizando los archivos creados en C a partir de una plantilla predefinida por lo cual estos archivos `.tmf` también permitirán obtener aplicaciones en tiempo real para diferentes plataformas, por ejemplo para nuestra aplicación se utilizará el `rtwin_tmf` el cual construirá un programa en tiempo real genérico, a continuación se muestra en la Figura 2.4 un esquema del funcionamiento de

la herramienta RTW y de la misma manera en la Figura 2.5 se presenta los valores por defecto que aparecen y que serán utilizados para realizar una aplicación en tiempo real genérica.

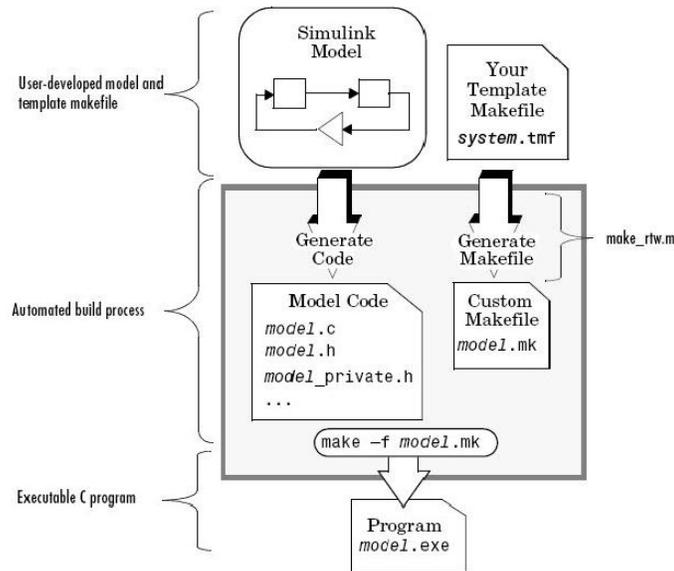


Figura 2.4 Funcionamiento Del RTW

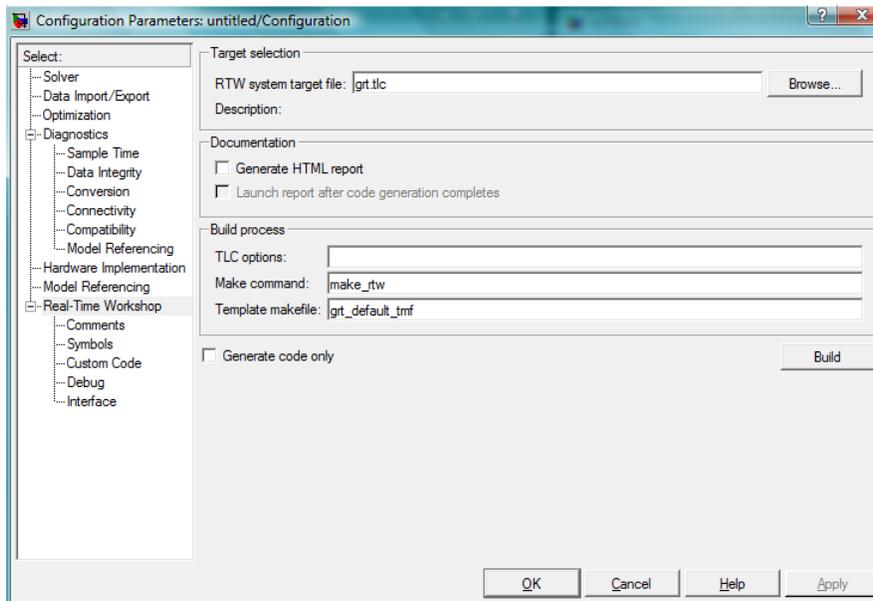


Figura 2.5 Configuración Genérica del RTW

2.1.5 Pasos En El Desarrollo De Un Algoritmo Para Control En Tiempo Real

1. Diseño del modelo en Simulink®.
2. Simulación del modelo en Simulink® (Simulink® en modo normal).
3. Generación del código fuente en lenguaje “C” mediante el RTW.

Real-Time Workshop permite generar archivos con código C de las siguientes dos maneras:

a. Utilizando la función `emlc` directamente en la ventana de comandos de MATLAB® y llenando los parámetros requeridos por la función anteriormente mencionada, como lo explica la ayuda de MATLAB® al digitar el comando `help emlc` tal como se puede apreciar en la Figura 2.6.

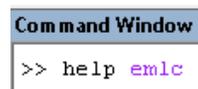


Figura 2.6 Generación De Código Para Modelos En MATLAB®

b. Mediante la barra de tareas de Simulink®, Tools, Real Time Workshop, Build Model como se puede apreciar en la Figura 2.7.

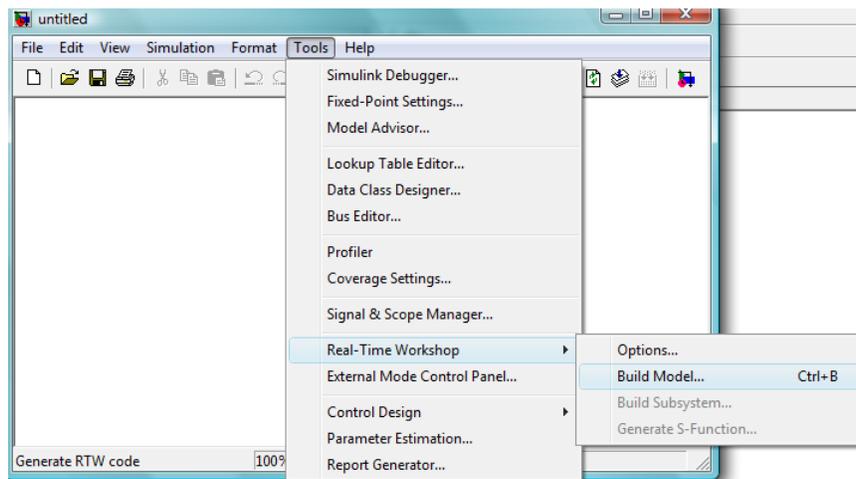


Figura 2.7 Generación De Código Para Modelos En Simulink®.

El RTW realiza la compilación, enlace y descarga del ejecutable al Target.

4. Conexión del Host con el Target para sintonía del modelo y ajustes de parámetros (Simulink[®] en modo externo).

El RTW realiza las iteraciones hasta encontrar el modelo y los parámetros adecuados.

2.1.6 Etapas En La Generación RTW

1. A partir del modelo en Simulink[®] (modelo.mdl) el RTW genera un archivo intermedio llamado modelo.rtw.
2. El TLC toma el archivo intermedio y lo transforma a lenguaje “C” u otro lenguaje con la ayuda de los archivos system.tlc, blocks.tlc y librerías tlc.
3. El proceso de BUILD puede terminar aquí o seguir construyendo el ejecutable dependiendo de la opción “Generate code only”.
4. El RTW construye el archivo modelo.mk llamado makefile con instrucciones para el compilador de “C” y el enlazador del target seleccionado, para la construcción de modelo.mk se utiliza el archivo de configuración system.tmf.
5. RTW llama al comando MAKE del compilador instalado y le pasa las instrucciones para la construcción del ejecutable (modelo.mk).
6. El programa MAKE construye el ejecutable con el nombre “modelo.---” y lo deja en el directorio de trabajo.

2.1.7 Archivos Generados Por RTW

1. Archivo objeto “modelo.rtw”.
2. Template makefile “system,tmf”.

3. El TLC está constituido por: System Target File, Block Target File, Funciones del TLC.

4. Archivos generados por el TLC:

- Modelo.c.- algoritmo del modelo.
- Modelo.h.- inclusión de cabeceras, declaración de estructuras a: parámetros, bloques, estados, entradas y salidas.
- Modelo_data.c.- inicializa las variables de parámetros con los valores iniciales.
- Modelo_types.h.- declara estructuras como tipos de datos.
- Modelo_private.c.- definiciones comunes necesarios por el modelo y los subsistemas.

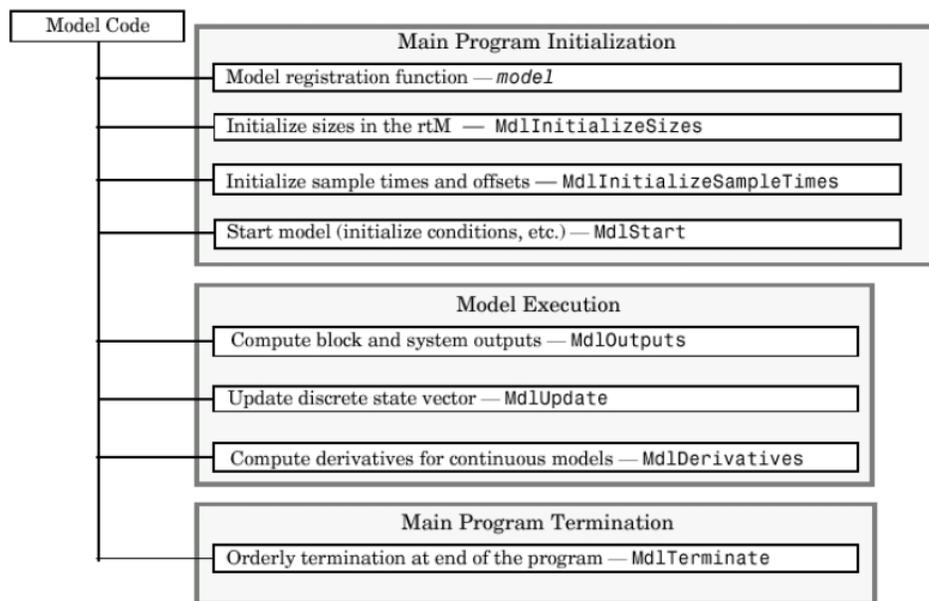


Figura 2.8 Estructura Del Código Del Modelo

2.1.8 Estructuras Declaradas por el Modelo RTW

- rtU.entrada_n.- referencia a cada bloque de tipo InPort que se encuentran en la raíz de Simulink®.

- rtY.salida_n.- referencia a cada bloque de tipo OutPort que se encuentran en la raíz de Simulink®.
- rtP.bloque_parámetro.- referencia a cada parámetro de cada bloque en Simulink®.
- rtB.bloque.- referencia a las salidas de cada bloque en Simulink®.

2.1.9 Código Generado en “C” por el RTW

```
#include <math.h>
#include <string.h>
#include "ejemp1.h"
#include "ejemp1_private.h"

ExternalOutputs rtY;

static rtModel_ejemp1 model_S;
rtModel_ejemp1 *const rtM_ejemp1 = &model_S;

void MdlStart(void)
{
... /* código de arranque del modelo */
}

void MdlOutputs(int_T tid)
{
... /* código de actualización de salidas */
/* parte principal del código */
}

void MdlUpdate(int_T tid)
{
... /* código para actualización de estados discretos */
}

void MdlTerminate(void)
{
... /* código de finalización */
}
```

```

void Md1InitializeSizes(void)
{
... /* código p/ inic. de Nro de bloques, estados, entradas, salidas, etc. */
}

void Md1InitializeSampleTimes(void)
{
... /* código p/ inic. tiempos de muestreo */
}

rtModel_ejemp1 *ejemp1(void)
{
... /* código de registro del modelo */
}

```

2.1.10 Ejecución del Modelo en Tiempo Real

La ejecución del modelo en tiempo real se basa en una programación por interrupciones donde el tiempo de simulación va a estar ligado con el tiempo real del reloj de la computadora y con las interrupciones de esta, como se puede apreciar en la Figura 2.9 el tiempo se incrementa solamente si se ha ejecutado todo el algoritmo de lectura, cálculo, escritura y actualización de datos del sistema, una vez concluido todo un ciclo el tiempo se incrementa lo cual nos da una percepción de trabajar sin retardos[2].

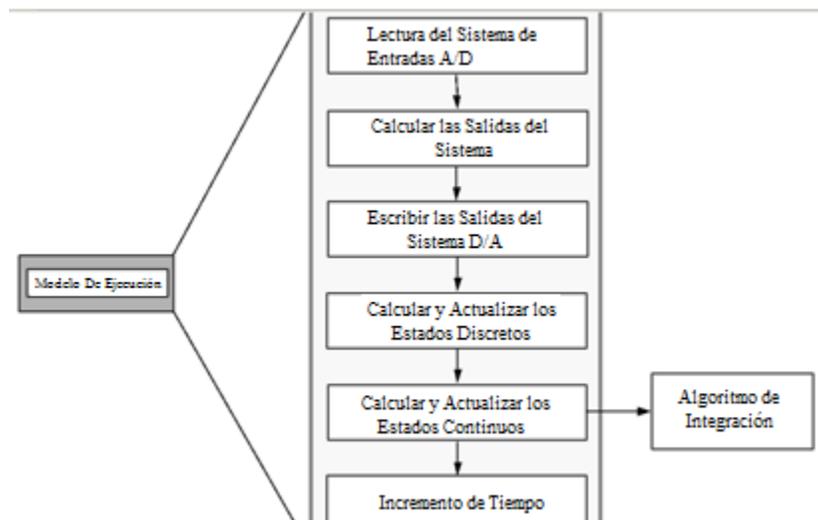


Figura 2.9 Ejecución del Modelo en Tiempo Real

2.1.10.1 Integración de Estados Continuos

El programa en tiempo real calcula los siguientes valores para los estados continuos basado en la derivada del vector dx / dt , de los valores actuales de los insumos y del vector de estado. Estas derivadas se utilizan para calcular los siguientes valores de los estados con un estado de actualización de la ecuación. Este es el estado de actualización de la ecuación para el primer método de Euler.

$$x = x + \frac{dx}{dt}h$$

Donde h es el tamaño del paso de la simulación, x representa el vector estado y dx/dt es el vector de derivadas.

Otros algoritmos pueden hacer varias llamadas a la producción y a las rutinas de derivadas para producir estimaciones más precisas. Hay que tomar en cuenta, sin embargo, que los programas en tiempo real utilizan un tamaño fijo de pasos, porque es necesario para garantizar el cumplimiento de todas las tareas dentro de un período de tiempo determinado.

Esto significa que, si bien debe utilizar métodos de mayor orden de integración de los modelos con la dinámica de una amplia variedad, los métodos de orden superior requieren tiempo de cálculo adicional. A su vez, el tiempo de cálculo adicional podría obligar a utilizar un tamaño de paso más grande, que puede disminuir la mejora de la precisión que inicialmente se trató para la integración.

2.2 CARACTERÍSTICAS DEL REAL TIME WINDOWS TARGET Y DE LA TARJETA DE ADQUISICIÓN DE DATOS

2.2.1 Real Time Windows Target (RTWT)

2.2.1.1 Características Generales Del RTWT

Matlab[®] presenta una aplicación para hacer simulaciones en tiempo real, esta toolbox es Real Time Windows Target (RTWT). Esta herramienta permite diferentes aplicaciones como control en tiempo real, simulación en tiempo real para plantas físicas como por ejemplo turbinas de avión, modelamiento de sistemas físicos, etc.

Real Time Windows Target es una solución aplicada en computadores para prototipos y pruebas de sistemas en tiempo real, donde permite realizar visualizaciones de señales con el bloque de osciloscopios de Simulink[®], modificación de parámetros y control en tiempo real.

La toolbox Real Time Windows Target es una herramienta de Matlab[®] que permite adquirir y generar señales en tiempo real. La comunicación entre el equipo y el hardware exterior se realiza por medio de una tarjeta de adquisición de datos, en este caso National Instruments PCI 6221 [3] la cual permite operar con señales de entrada y/o salidas análogas y digitales que se analizará posteriormente.

2.2.1.2 Características Principales del RTWT

- Ejecuta modelos en tiempo real en computadores de escritorio y personales.
- Permite la adquisición de señales.
- Trabaja con tarjetas de adquisición de datos incluyendo a tarjetas PCMCIA para computadoras portátiles.
- Permite el control de la ejecución en tiempo real directamente desde el modelo con la barra de herramientas de Simulink[®].
- Alcanza velocidades de muestreo superior a 5 kHz para la mayoría de los modelos.

- Requiere un compilador de C para la creación de código en tiempo real por medio de la herramienta RTW.
- Real-Time Windows Target incluye un conjunto de bloques que proporcionan conexiones entre las entradas y salidas físicas de la tarjeta de adquisición de datos y el modelo en tiempo real, dichos bloques son: Digital Input (entradas digitales), Digital Output (salidas digitales), Analog Input (entradas analógicas), Analog Output (salidas analógicas), Counter Input (contador de pulsos), Frequency Output (modulación PWM), Encoder Input (entrada de codificador), Packet Input (recibe datos binarios sin formato), Packet Output (envía datos binarios sin formato), Stream Input (recibe datos en formato ASCII), Stream Output (envía datos en formato ASCII), tal como lo podemos apreciar en la Figura 2.10 que se presenta a continuación[4].

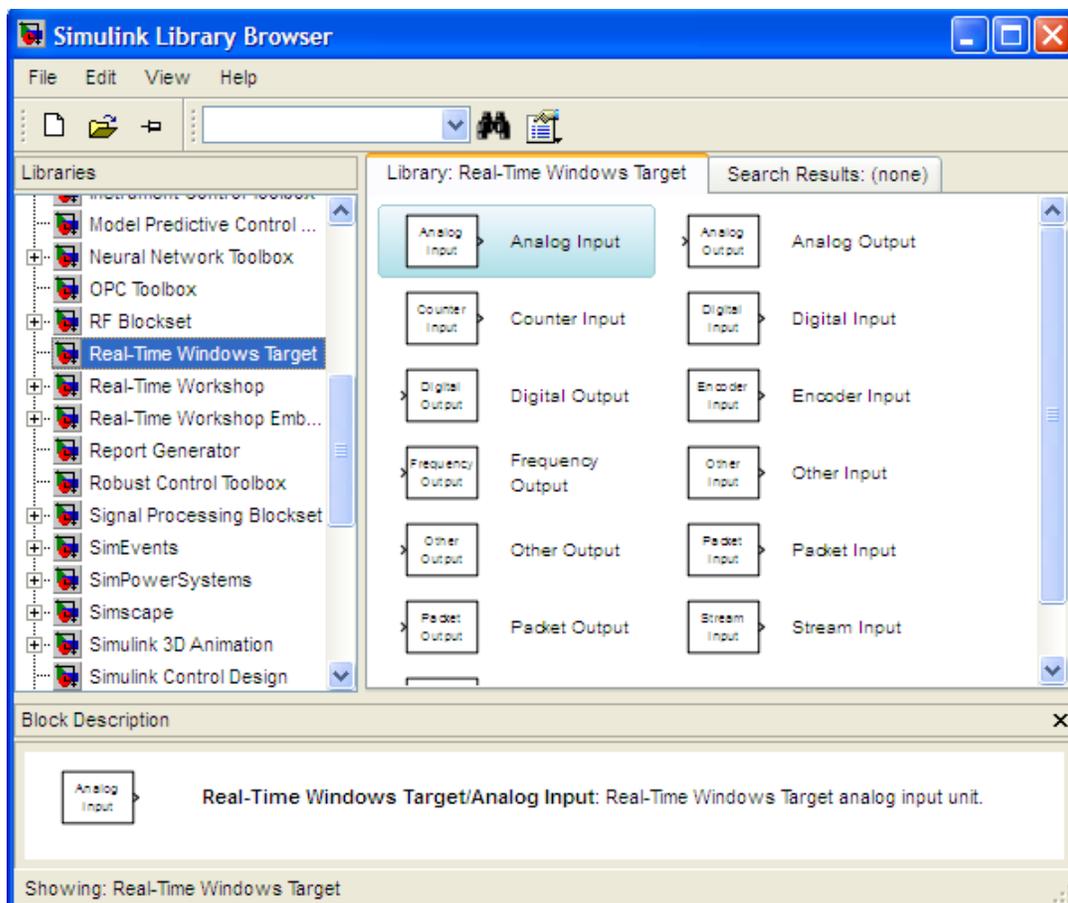


Figura 2.10 Real Time Windows Target Library

2.2.2 Tarjeta de Adquisición de Datos National Instruments PCI 6221



Figura 2.11 Tarjeta de Adquisición de Datos National Instruments PCI 6221

2.2.2.1 Información General

La PCI-6221 de National Instruments es una tarjeta de adquisición de datos (DAQ) multifunción de la Serie M optimizada para aplicaciones condicionadas por costos. Se selecciona los dispositivos de la Serie M de alta velocidad para alcanzar velocidades de muestreo 5 veces más rápidas o dispositivos de alta precisión para obtener 4 veces más resolución y superior precisión en las medidas.

Las tarjetas de la Serie M ofrecen características avanzadas como el controlador de sistema NI-STC 2, el amplificador programable NI-PGIA 2 y la tecnología de calibración NI-MCal para mejorar el rendimiento y la precisión. Los dispositivos de la Serie M funcionan con diversos sistemas operativos usando tres opciones de software incluyendo NI-DAQmx, NI-DAQmx Base y el Hardware de Medida DDK.

2.2.2.2 Resumen de Especificaciones

Información General	
Formato Físico	PCI
Sistema Operativo / Objetivo	Windows, Real-Time, Linux, Mac OS
Tipos de Medida	Digital, Frecuencia, Codificadores de cuadratura, Voltaje
Familia de Productos DAQ	Serie M
Soporte para LabVIEW RT	Sí
Compatibilidad con RoHS	Sí
Compatibilidad con Matlab®	Sí

Tabla 2.1 Información General

Entrada Analógica	
Número de Canales	16 SE/8 DI
Velocidad de Muestreo	250 kS/s
Resolución	16 bits
Muestreo Simultáneo	No
Rango de Voltaje Máximo	-10..10 V
Precisión del Rango	3100 μ V
Sensibilidad del Rango	97.6 μ V
Rango de Voltaje Mínimo	-200..200 mV
Precisión del Rango	112 μ V
Sensibilidad del Rango	5.2 μ V
Número de Rangos	4
Memoria Interna	4095 muestras

Tabla 2.2 Entrada Analógica

Salida Analógica	
Número de Canales	2
Razón de Actualización	833 kS/s
Resolución	16 bits
Rango de Voltaje Máximo	-10..10 V
Precisión del Rango	3230 μ V
Rango de Voltaje Mínimo	-10..10 V
Precisión del Rango	3230 μ V
Capacidad de Corriente (Canal/Total)	5 mA

Tabla 2.3 Salida Analógica

E/S Digital	
Número de Canales	24 DIO
Temporización	Hardware, Software
Máximo Rango de Tiempo	1 MHz
Niveles Lógicos	TTL
Máximo Rango de Entrada	0..5 V
Máximo Rango de Salida	0..5 V
Entrada de Flujo de Corriente	Sinking, Sourcing
Filtros de Entrada Programables	Sí
Salida de Flujo de Corriente	Sinking, Sourcing
Capacidad de Corriente (Canal/Total)	24 mA/448 mA
Temporizador Watchdog	No
¿Soporta Estados de Encendido Programables?	Sí
¿Soporta Protocolo de Sincronización para E/S?	No
¿Soporta E/S de Patrones?	Sí

Tabla 2.4 Entrada / Salida Digital

Contadores/Temporizadores	
Número de Contadores/Temporizadores	2
Resolución	32 bits
Frecuencia Máxima de la Fuente	80 MHz
Entrada Mínima de Ancho de Pulso	12.5 ns
Niveles Lógicos	TTL
Rango Máximo	0..5 V
Estabilidad de Tiempo	50 ppm
Sincronización GPS	No
Generación de Pulso	Sí
Operaciones a Búfer	Sí
Eliminación de Rebotes	Sí
Número de Canales DMA	2

Tabla 2.5 Contadores / Temporizadores

Temporización/Disparo/Sincronización	
Bus de Sincronización (RTSI)	Sí
Disparo	Digital

Tabla 2.6 Temporización/Disparo/Sincronización

2.2.2.3 Conector SCB-68

El SCB-68 es un bloque conector de E/S para realizar interfaz de señales de E/S a dispositivos DAQ insertables con conectores de 68 pines. En combinación con los cables blindados, el SCB-68 brinda terminación de señal robusta con muy poco ruido. El SCB-68 es compatible con dispositivos de la Serie M de un solo conector o doble conector con conectores de 68 pines. El bloque conector es también compatible con la mayoría de los dispositivos DAQ de la Serie E, B, S y R, el cual posee las siguientes características:

- 68 pines de montaje vertical, terminales de tornillo para conectar fácilmente a dispositivos DAQ de 68 pines.
- Dimensiones: 14.35 x 10.74 cm (5.65 x 10.74 cm.).
- Monturas metálicas para uso en PC de escritorio o para montaje en un panel personalizado.

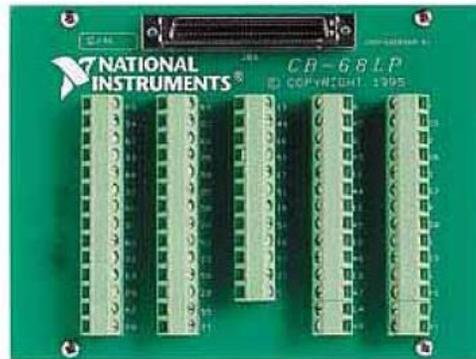


Figura 2.12 Conector SCB-68

2.2.2.4 Cable NI SHC68-68-EPM

El SHC68-68-EPM de National Instruments está especialmente diseñado para funcionar con dispositivos de la Serie M. Este cable ofrece secciones digitales y analógicas separadas, pares cruzados blindados individualmente para entradas analógicas, salidas analógicas blindadas individualmente y pares cruzados para E/S digitales críticas, el cuál presenta las siguientes características:

- Presenta pares analógicos cruzados blindados individualmente para reducir interferencia con tarjetas de alta velocidad.
- 2 m de longitud.
- Compatible con RoHS.
- Conecta dispositivos de 68 pines de la Serie M directamente a accesorios de 68 pines.



Figura 2.13 Cable NI SHC68-68-EPM

2.2.2.5 Distribución de Pines de la Tarjeta de Adquisición de Datos NI PCI 6221

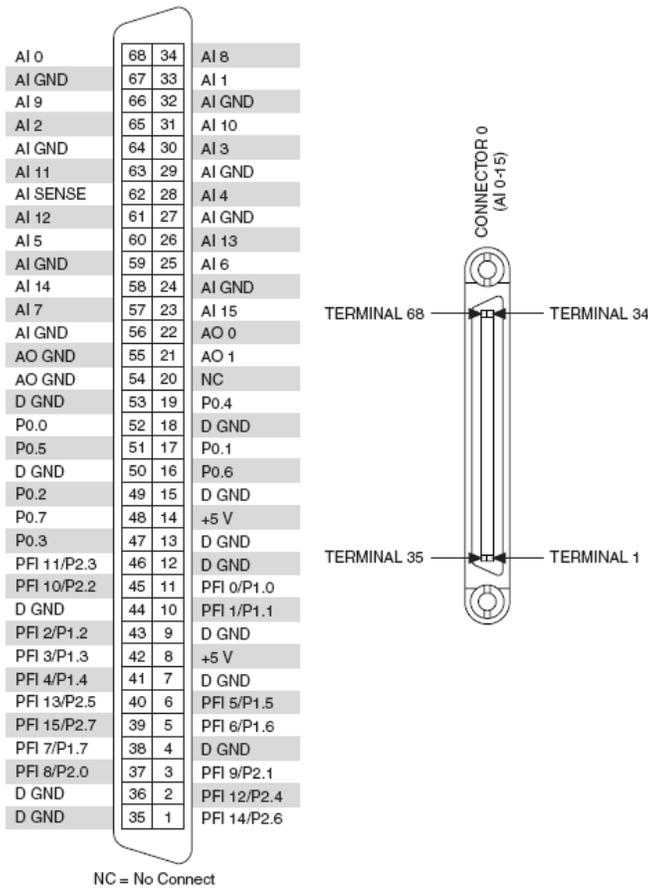


Figura 2.14 Distribución de Pines de la Tarjeta de Adquisición de Datos NI PCI 6221

2.3 CARACTERÍSTICAS DE LA UNIDAD AIR FLOW TEMPERATURE CONTROL SYSTEM

En la Figura 2.15 se muestra el modelo físico del módulo de temperatura, que consta de un sensor tipo IC de estado sólido, el cual por su tipo, es lineal. Además de esto, se tiene un conducto de flujo de aire, el cual mediante un ventilador, que renueva el aire que se encuentra dentro de él. Una resistencia o niquelina, que se encuentra en la entrada del conducto, es la encargada de calentar el aire que ingresa. De esta forma y gracias al ventilador, se tienen diferentes temperaturas a lo largo del conducto de aire.

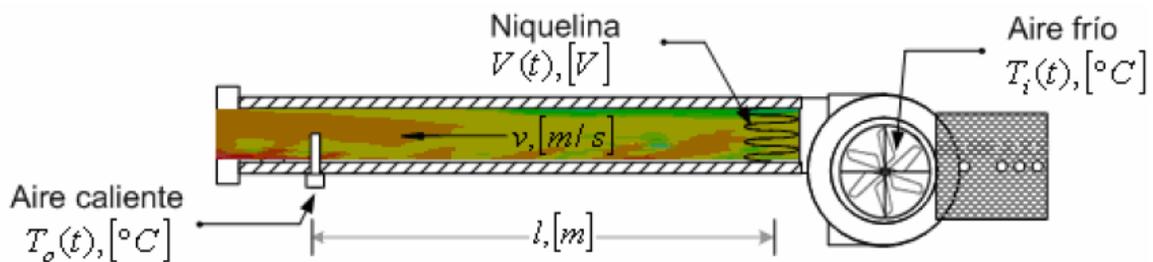


Figura 2.15 Proceso Térmico de la Unidad Air Flow Temperature Control System

Nótese que para asegurar que se mida una solución homogénea, el punto de supervisión se localiza a cierta distancia del punto de mezcla. Por tanto, se tiene interés en conocer la forma en que responde la temperatura de salida, a los cambios en el voltaje de la niquelina en la entrada [5].

La unidad Air Flow Temperature Control System posee:

- **Elemento primario y de medición:** sensor-transmisor de temperatura IC (variación lineal). Toma la temperatura a la salida del calentador ($20 - 70$ [°C]), y envía una señal analógica de voltaje ($0 - 5$ [V]) hacia el controlador.
- **Actuador:** módulo Air Flow Temperature Control System que simula un calentador eléctrico de temperatura, para simulación de procesos de control de temperatura, gobernado mediante una señal de voltaje en el rango de 0 a 10 [V].

- **Placa Metálica:** La cual posee cuatro orificios, que determinan la cantidad de aire que ingresará al proceso.

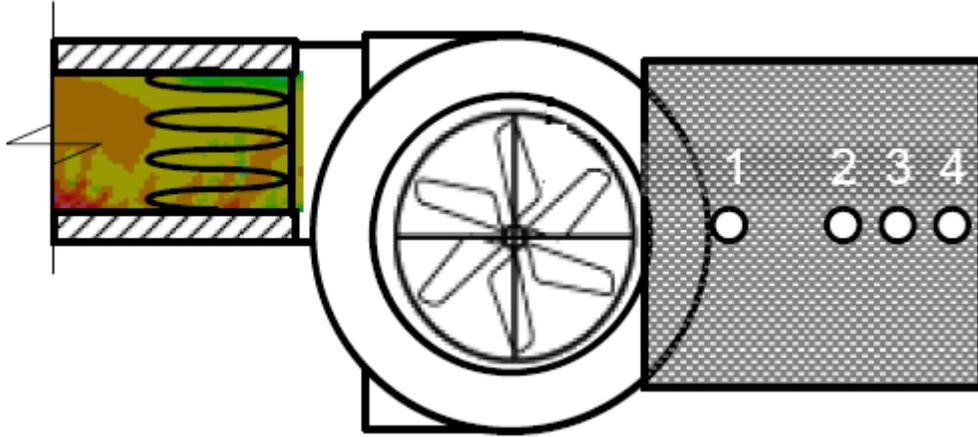


Figura 2.16 Placa Metálica de la Unidad Air Flow Temperature Control System

CAPÍTULO 3

DISEÑO DEL SISTEMA DE CONTROL RTW

3.1 DISEÑO DEL HARDWARE

El hardware implementado en el presente proyecto de tesis está constituido por una planta intercambiadora de temperatura (Unidad Air Flow Temperature Control System), un computador de escritorio, una tarjeta de adquisición de datos PCI 6221, un conector SCB-68, un cable tipo SHC68-68-EPM y cableado en general como se muestra en la Figura 3.1.

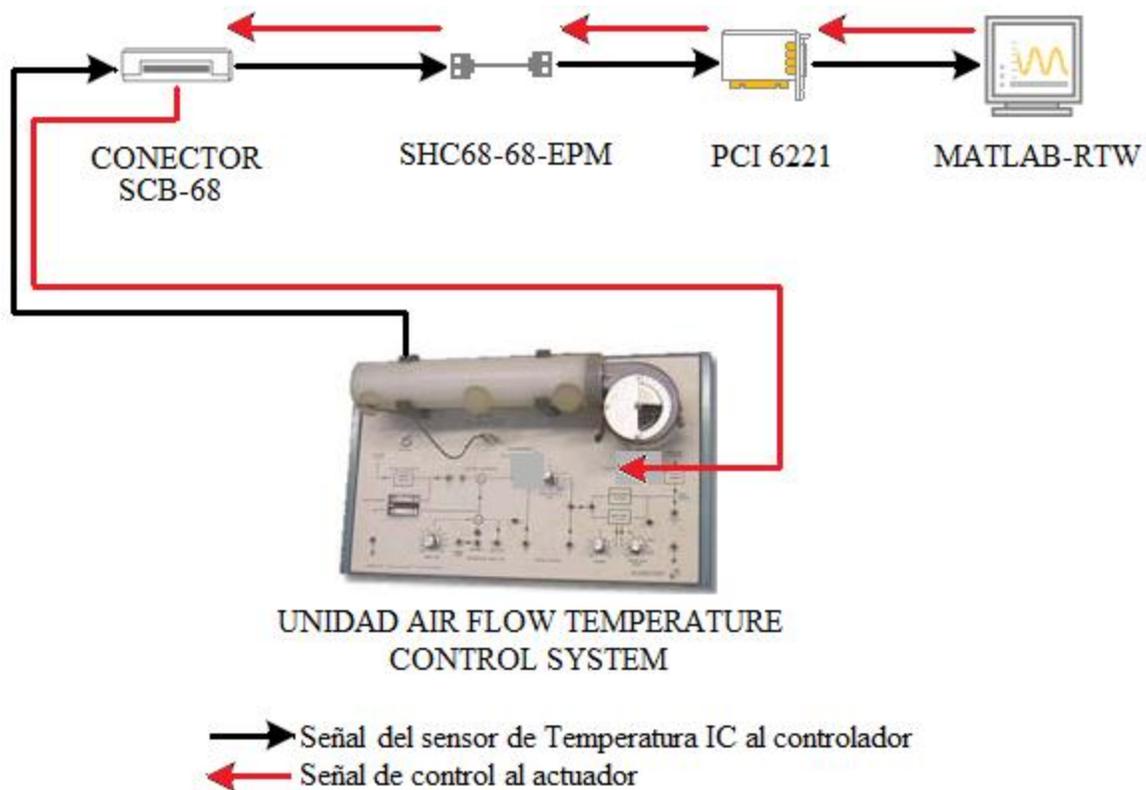


Figura 3.1 Diagrama General de Componentes Del Sistema

Cabe recalcar que las características de la Unidad *Air Flow Temperature Control System*, de la tarjeta de adquisición de datos PCI 6221, del conector SCB-68 y del cable tipo SHC68-68-EPM ya fueron descritas en el Capítulo 2.

3.1.1 Características de la PC

Al trabajar con la herramienta RTW de MATLAB[®] el programa interacciona directamente con las interrupciones del procesador de la computadora por lo que para lograr un rendimiento óptimo del control en tiempo real es necesario una tarjeta de adquisición de datos con gran capacidad de muestreo por segundo y un reloj en el computador que sea capaz de trabajar con un tiempo acorde con el muestreo de la tarjeta, porque todos los tiempos van a estar regidos por el reloj del computador y como se utilizó la tarjeta de adquisición de datos PCI 6221 cuyo driver en MATLAB[®] se encuentra disponible desde la versión 7.5 se vio la necesidad de utilizar un computador que cumpla con las características detalladas a continuación en la Tabla 3.1.

<i>HARDWARE</i>	<i>REQUERIMIENTOS MINIMOS</i>	<i>UTILIZADO</i>
CPU	Pentium	Pentium IV 3,2 GHz
Disco Duro	16 MB libres	120 GB
Tarjeta de adquisición de datos	Compatible con RTWT y RTW	PCI 6221 Nationals Instruments
RAM	256 MB	1 GB

Tabla 3.1 Características De La PC

3.1.2 Descripción del diagrama de conexiones

Se utilizó una entrada analógica de 0-10 VDc (AI0 pin 68) conectada a la salida del sensor de temperatura tipo IC y una salida analógica (AO0 pin 22) conectada al actuador de la planta, además se utilizó las tierras analógicas pines 67 y 24 de la tarjeta de adquisición de datos PCI 6221 para conectarlas a la tierra común de la planta, teniendo el diagrama de conexión representado en la Figura 3.2 que se presenta a continuación.

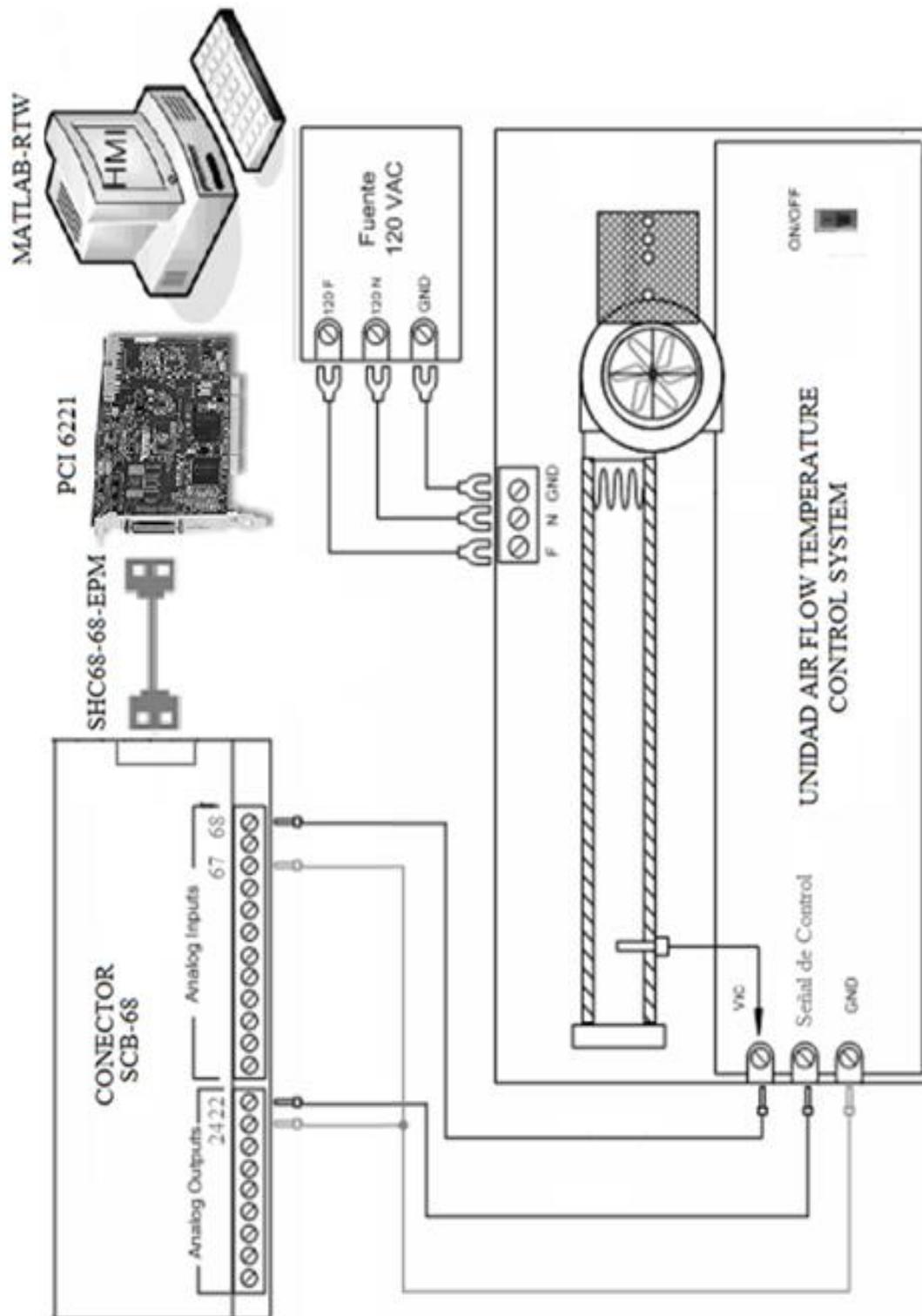


Figura 3.2 Diagrama De Conexiones

3.1.3 Diagrama de bloques del sistema de control

El sistema de control de temperatura de la Unidad Air Flow Temperature Control System posee un actuador que está gobernado por voltajes de 0-10 VDC, de la misma manera posee un sensor de temperatura del tipo IC que emitirá señales de voltaje de 0-5 VDC, estos dos se encuentran linealizados por lo que no hace falta una etapa de acondicionamiento de la señal, el proceso de conversión análogo digital y digital análogo se produce internamente mediante la tarjeta de adquisición de datos PCI 6221, y estos datos son llevados a Simulink[®] mediante la herramienta RTW de MATLAB[®] donde se encuentra el controlador y el set point, que gracias a la herramienta RTWT puede variar sus parámetros en tiempo real desde esta plataforma, a continuación se presenta en la Figura 3.3 el diagrama de bloques del sistema de control.

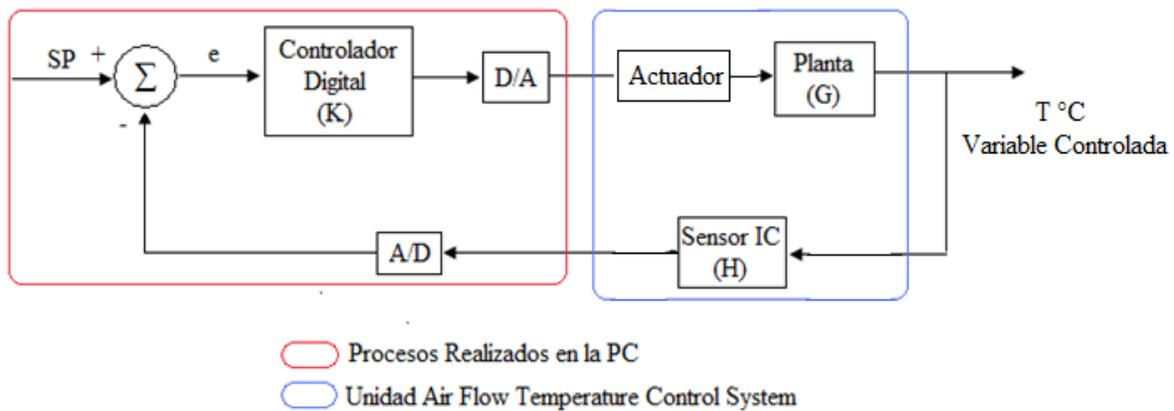


Figura 3.3 Diagrama De Bloques Del Sistema De Control

3.2 DISEÑO DEL SOFTWARE

Para poder comprender de una mejor manera el diseño de una aplicación en tiempo real mediante MATLAB[®] se utilizará a continuación el desarrollo de un ejemplo didáctico en el cuál se verán inmersos los siguientes pasos para la creación de un modelo de Simulink[®] en tiempo real.

- Instalación y Configuración del Real Time Windows Target.
- Ingreso a Simulink[®].
- Utilización de la librería RTWT.
- Instalación y reconocimiento de la tarjeta de adquisición de datos.
- Configuración de parámetros del RTW.
- Configuración de las opciones de los elementos de visualización.
- Ejecución de la aplicación en tiempo real.
- Obtención de resultados.
- Desinstalación del Real Time Kernel.

El ejemplo que se va a desarrollar consiste en retroalimentar una señal analógica generada por las herramientas de Simulink[®] mediante la utilización de una salida y una entrada analógica de la tarjeta de adquisición de datos PCI 6221, dando seguimiento mediante un Osciloscopio de Simulink[®] como muestra la Figura 3.4.

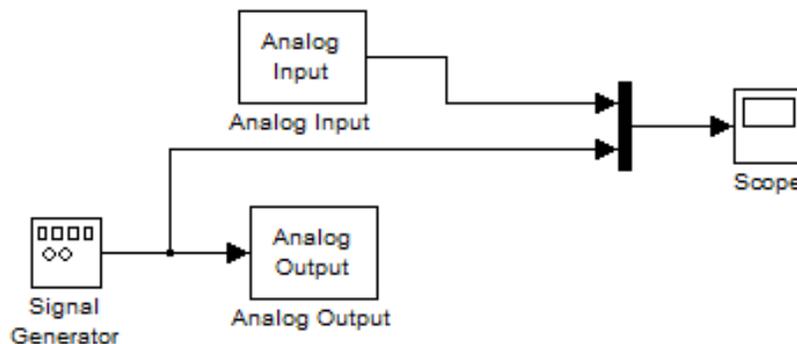


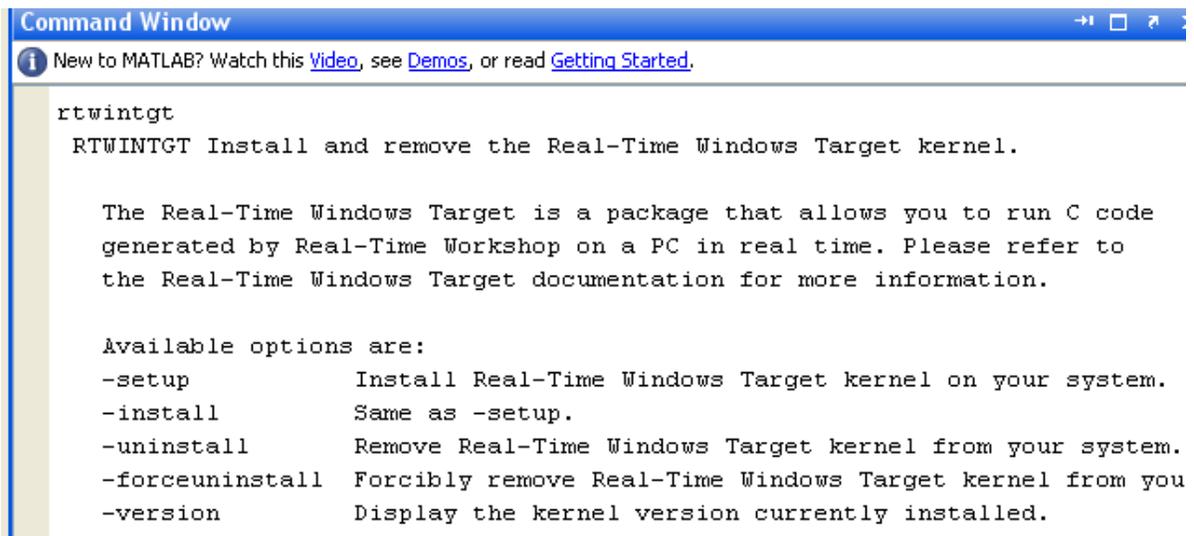
Figura 3.4 Diagrama De Ejemplo

3.2.1 Instalación y Configuración del Real Time Windows Target

3.2.1.1 Real Time Kernel

El software Real – Time Windows Target incluye un Real Time Kernel que sirve de interface con el sistema operativo de Windows. El Real – Time Windows Target kernel asigna altas prioridades de ejecución al código ejecutable, el cual permite correr sin interferencias los tiempos de muestreo seleccionados. Durante la ejecución en tiempo real del modelo, el kernel interviene cuando necesita asegurarse que el modelo tenga prioridad al usar la CPU para ejecutar cada modelo actualizado al prescribir tiempos de muestreo. Cada vez que un modelo se ha ejecutado completamente, el kernel libera la CPU para correr alguna otra aplicación basada en Windows, que podría necesitar este servicio.

Para poder observar todas las opciones del RTWT Kernel se digita en la ventana de comandos de MATLAB® el comando `rtwintgt` como se puede apreciar en la Figura 3.5.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

rtwintgt
RTWINTGT Install and remove the Real-Time Windows Target kernel.

The Real-Time Windows Target is a package that allows you to run C code
generated by Real-Time Workshop on a PC in real time. Please refer to
the Real-Time Windows Target documentation for more information.

Available options are:
-setup          Install Real-Time Windows Target kernel on your system.
-install       Same as -setup.
-uninstall     Remove Real-Time Windows Target kernel from your system.
-forceuninstall Forcibly remove Real-Time Windows Target kernel from you
-version       Display the kernel version currently installed.
```

Figura 3.5 Propiedades del Real Time Windows Target Kernel

3.2.1.2 Instalación Del Kernel

Durante la instalación del software Matlab[®], todo el software del Real – Time Windows Target es copiado en el disco duro, pero el Real – Time Windows Target kernel no es instalado automáticamente en la instalación de Matlab[®]. Se debe instalar el kernel antes de correr una aplicación Real – Time Windows Target. Cada vez que se instale o se desinstale se debe reiniciar la computadora para que se actualicen los cambios, en tal virtud antes de empezar a trabajar en el modelo en tiempo real en Simulink[®] se debe realizar los siguientes pasos para la instalación del Kernel.

La instalación del kernel se la puede realizar de dos maneras las cuales son:

1. Se utiliza el comando `rtwintgt – install` o en su defecto el comando `rtwintgt – setup` teniendo en cuenta que se deben realizar los siguientes pasos para la instalación.

a) En la ventana de comandos de Matlab[®] se escribe:

```
Rtwintgt – install
```

La ventana de comandos de Matlab[®] puede mostrar los siguientes mensajes:

i. You are going to install the Real – Time Windows Target kernel.

Do you want to proceed? [y]:

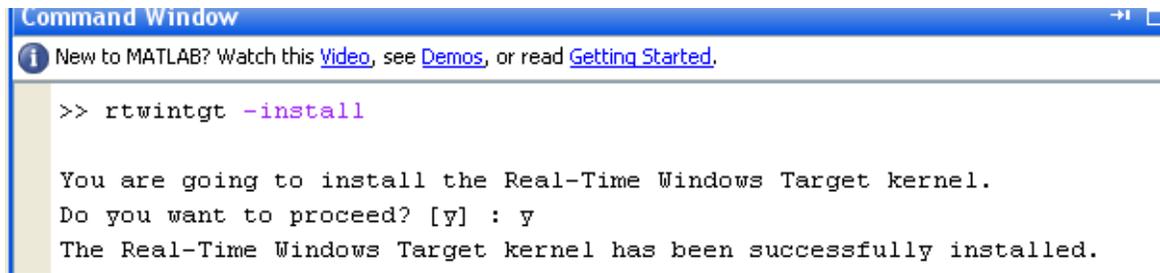
ii. There is a differnt versión of the Real-Time Windows Target kernel installed.

Do you want to update to the current versión? [y]:

b) Se procede a escribir Y y se continúa instalando el kernel, o n para cancelar la instalación sin hacer ningún cambio.

Si se escribe Y, el entorno de Matlab® instala el kernel y muestra el siguiente mensaje:

The Real – Time Windows Target Kernel has been successfully installed, tal como se muestra en la Figura 3.6



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> rtwintgt -install

You are going to install the Real-Time Windows Target kernel.
Do you want to proceed? [y] : y
The Real-Time Windows Target kernel has been successfully installed.
```

Figura 3.6 Confirmación de la Instalación del Real Time Kernel

c) Si un mensaje aparece preguntando si desea reiniciar la computadora, haciéndolo antes de utilizar el kernel, quiere decir que el modulo del Real – Time Windows Target no corrió correctamente.

d) Después de instalar el kernel, hay que verificar que este se ha instalado correctamente escribiendo el comando:

```
rtwho
```

En la ventana de comandos de Matlab® debe aparecer un mensaje que muestre la versión del número del kernel, seguido por el desempeño, *timeslice* y otra información, tal como se muestra en la Figura 3.7.

```
>> rtwho  
  
Real-Time Windows Target version 3.3.0 (C) The MathWorks, Inc. 1994-2008  
Running on Multiprocessor APIC computer.  
MATLAB performance = 100.0%  
Kernel timeslice period = 1 ms
```

Figura 3.7 Estado de la Instalación del Real Time Kernel

Los mensajes que aparecen como resultado del comando se interpretan como se indica a continuación: **MATLAB performance = 100.0%**, este mensaje indica que el entorno de MATLAB® y otras aplicaciones en tiempo no real (por ejemplo, un procesador de palabras) están corriendo al 100% del desempeño porque las aplicaciones en tiempo no real están continuamente ejecutándose. Cuando una aplicación en tiempo real se está ejecutando, el desempeño de MATLAB® está por debajo del 100%. Por ejemplo, si el desempeño de MATLAB = 90%, entonces la aplicación en tiempo real está usando el 10% del tiempo del CPU. Se recomienda que se seleccione un tiempo de muestreo de tal forma que la `rtwho` retorne un desempeño de MATLAB® de por lo menos 80% y **Kernel timeslice period =1ms**, en donde el periodo del kernel time slice es la frecuencia actual de los tiempos de interrupción del hardware. Un milisegundo es el máximo valor para los modelos con largos tiempos de muestreo (bajas frecuencias de muestreo) o cuando una aplicación no ha sido construida. Este valor cambia cuando se selecciona tiempos menores que 1 milisegundo.

Una vez instalado el Kernel, se lo puede dejar instalado. El kernel permanece inactivo después de haberlo instalado, permitiendo que el sistema operativo de Windows ejecute cualquier aplicación que necesite de este servicio.

2. La segunda manera de instalar el Real Time Kernel es Mediante el explorador de MATLAB® como se muestra a continuación:

- a) Click en el botón Start de Matlab®.
- b) Se selecciona Links and Targets > Real – Time Windows Target > Install real - time kernel, tal como se muestra en la Figura 3.8

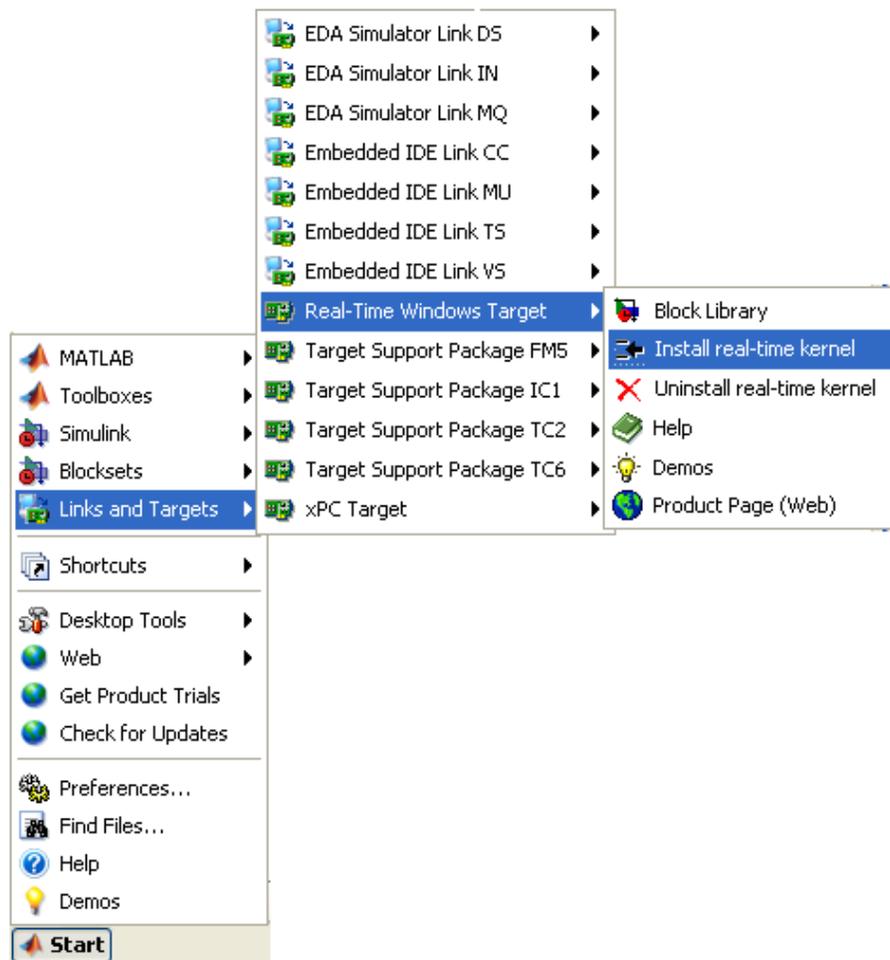


Figura 3.8 Instalación del Real Time Kernel

Posteriormente se debe seguir el mismo procedimiento de verificación de la instalación del Kernel anteriormente descrita.

3.2.2 Ingreso a SIMULINK®

Una vez instalado el Kernel procedemos a abrir Simulink® en la ventana de comandos de Matlab® se escribe *Simulink* como se puede apreciar en la Figura 3.9.

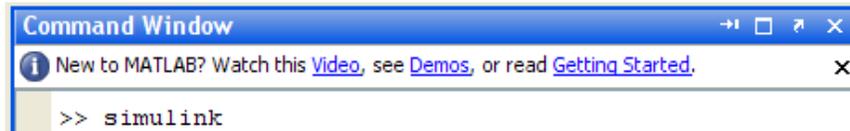


Figura 3.9 Primera Manera de Ingreso a SIMULINK®

También se puede acceder a Simulink® a través de File>New>Model, como se puede apreciar en la Figura 3.10

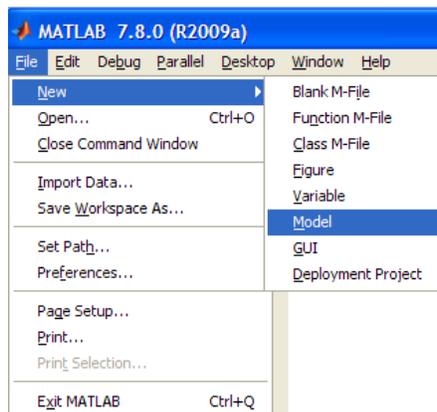


Figura 3.10 Segunda Manera de Ingreso a SIMULINK®

Y por último se puede dar un click sobre el botón New Model de la barra de herramientas, como se puede apreciar en la Figura 3.11.

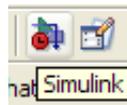


Figura 3.11 Tercera Manera de Ingreso a SIMULINK®

3.2.3 Utilización De La Librería Real Time Windows Target (RTWT)

Para poder observar la librería del Real Time Window Target desde la ventana de comandos de Matlab® podemos utilizar el siguiente comando.

Rtwinlib

Para ver la librería del Real – Time Window Target desde la ventana del modelo de Simulink®.

1. En la ventana del modelo de Simulink® seleccionamos View, Library Browser, Real Time Window Target tal como se muestra en la Figura 3.12, en donde se tiene:

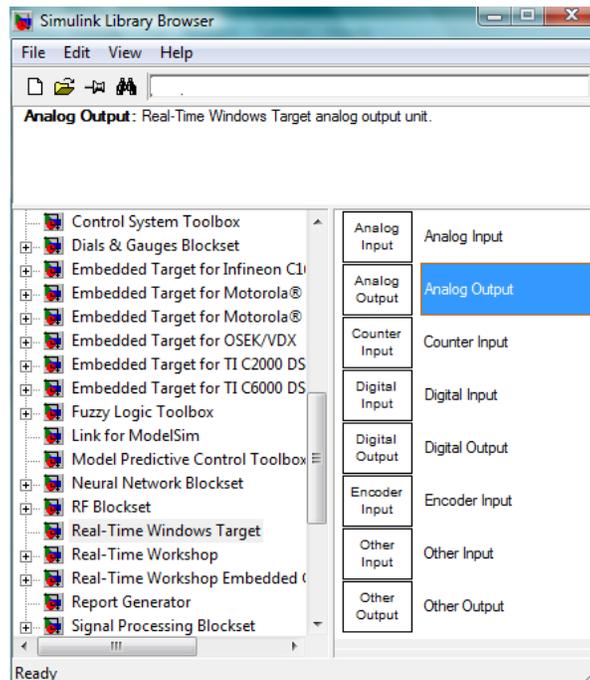


Figura 3.12 Librería Real Time Windows Target

Continuando con el ejemplo mencionado vamos a configurar una entrada y una salida analógica de la siguiente manera:

- **Entradas y Salidas Analógicas.-** Los drivers utilizados por las aplicaciones Real – Time Windows Target en contraste con los autónomos de las tarjetas, capturan un único punto de datos en cada intervalo de muestreo. Un esfuerzo considerable es realizado para minimizar los retardos entre la recolección de puntos de información y la utilización de estos datos en los algoritmos de los sistemas de control. El software Real Time Window Target permite normalizar internamente a los bloques de los Diagramas. Generalmente, las entradas representan los valores del mundo real como es una velocidad angular, posición, temperatura, presión, etc. Esta capacidad para seleccionar señales normalizadas permite: Utilizar nuestros propios factores de escala o trabajar con unidades significativas sin tener que convertirlas en voltajes

Cuando utilizamos una entrada analógica, se debe seleccionar los rangos de los voltajes externos que se reciben en la tarjeta de adquisición de datos, y se escoge la señal que queremos sacar. Por ejemplo el voltaje de ingreso del sensor de temperatura de tipo IC que ingresa por la primera entrada analógica posee un rango de 0 [V] a +5 [V] pero la salida del bloque en Simulink® puede ser normalizada unipolar o bipolarmente añadiendo ganancias y *offset's* para trabajar con unidades de ingeniería, si solo se desea trabajar con voltios se escoge la opción voltios y si se desea convertir esta señal analógica a digital se escoge la opción *Raw* como se muestra en la Figura 3.13.

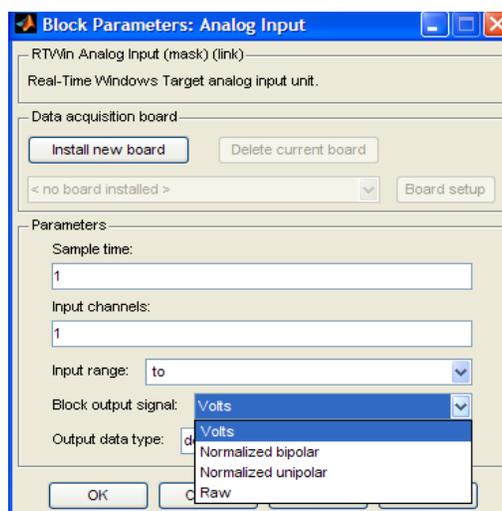


Figura 3.13 Bloque De La Señal De Salida

- **Entrada Analógica.**- Se utiliza esta opción para señales que están conectadas a los canales de las entradas analógicas en la tarjeta. En el bloque de parámetros en la sección *Input Channels* ingresamos 1 ó [1] (el uso de los corchetes es opcional siempre y cuando no se desee leer dos entradas al mismo tiempo donde se utilizaría la siguiente nomenclatura [1,2] para así poder leer la primera y la segunda entrada analógica) tal como se muestra en la Figura 3.14.

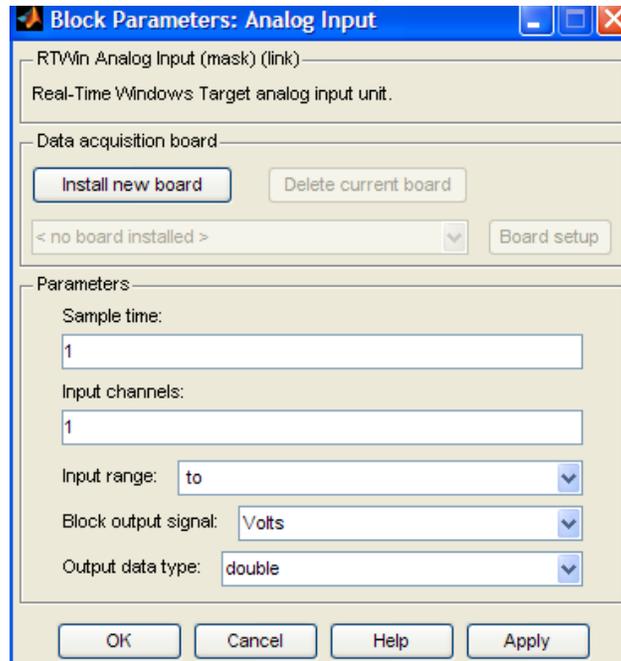


Figura 3.14 Bloque De Parámetros De La Entrada Análoga

Con esto indicamos que se utilizará la primera entrada analógica.

- **Salida Analógica.**- Se utiliza esta opción para señales que están conectadas a los canales de las salidas analógicas en la tarjeta. En el bloque de parámetros en la sección *Output Channels* ingresamos 1 ó [1] (el uso de los corchetes es opcional siempre y cuando no se desee leer dos salidas al mismo tiempo donde se utilizaría la siguiente nomenclatura [1,2] para así poder leer la primera y la segunda salida analógica) tal como se muestra en la Figura 3.15.

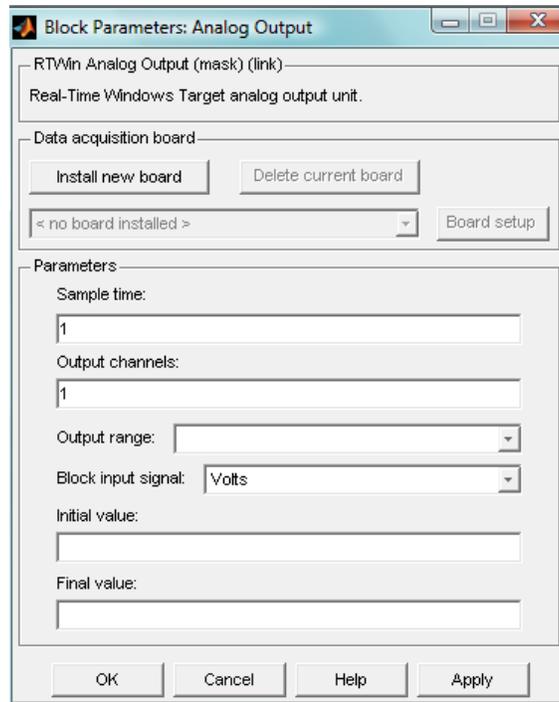


Figura 3.15 Bloque De Parámetros De La Salida Análoga

Con esto indicamos que se utilizará la primera salida analógica, hay que considerar que el valor de *Initial value* va a ser el valor por defecto con el cual va a empezar la simulación en tiempo real y de igual manera el valor de *Final Value* va a ser el valor por defecto con el cual va a terminar la simulación.

3.2.4 Instalación y reconocimiento de la tarjeta de adquisición de datos

Para poder configurar la Tarjeta de Adquisición de Datos es necesario primeramente saber si la tarjeta de adquisición soporta aplicaciones en tiempo real y si es compatible con Matlab[®] para lo cual se debe constatar de que la tarjeta que se utilice o vaya a ser adquirida se encuentre en la siguiente página web.

<http://www.mathworks.com/products/supportedio.html?prodCode=WT>

Una vez que se constate que la tarjeta de adquisición cumpla con los parámetros anteriormente descritos se procede a la instalación de la misma en donde básicamente un modelo RTWT se encuentra conectado a una tarjeta mediante un bloque de manejo de entradas y salidas, este bloque provee de una interface a los dispositivos de la tarjeta y de todas las especificaciones de configuración de la tarjeta.

Los drivers de la tarjeta para utilizarlos en la plataforma Matlab[®] se encuentran incluidos en el software Real Time Windows Target, normalmente esta librería provee la misma flexibilidad de configuración ofertada por las tarjetas manufacturadas.

Luego de haber analizando la información anteriormente presentada se procede a la instalación de la tarjeta de adquisición de datos siguiendo los siguientes pasos:

1) Se debe ubicar la tarjeta en este caso la tarjeta de adquisición de datos PCI 6221 correctamente sobre la ranura PCI de la computadora, posteriormente se procede a la instalación de los Drivers de la tarjeta los cuales deben venir incluidos en un Cd con la compra de la tarjeta, caso contrario se puede descargar fácilmente los drivers de la página oficial de la NATIONAL INSTRUMENT la cual es:

www.ni.com

En la sección *support* en la opción *Drivers and Updates* aparece una barra de búsqueda donde podemos buscar los *drivers* correspondientes de nuestra tarjeta al igual que los *datasheets*, como se muestra en la Figura 3.16 que se presenta a continuación:

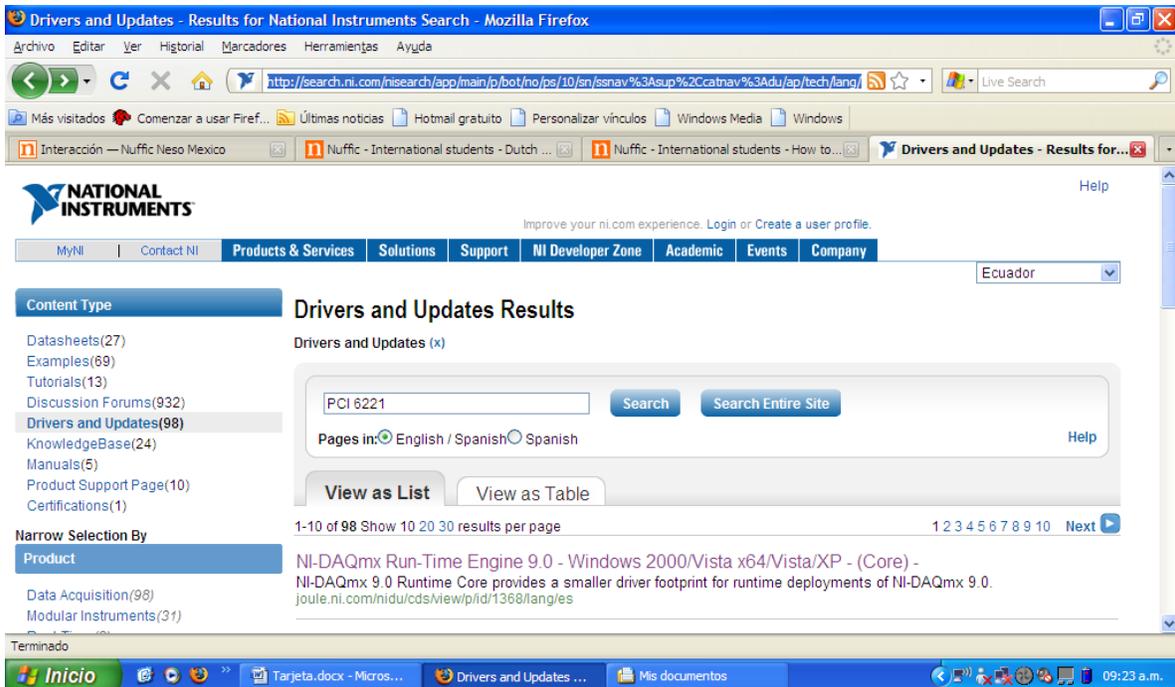


Figura 3.16 National Instruments Support Web

2. A continuación se debe ejecutar el software de instalación de los drivers de la tarjeta en donde debe aparecer una ventana similar a la presentada en la Figura 3.17

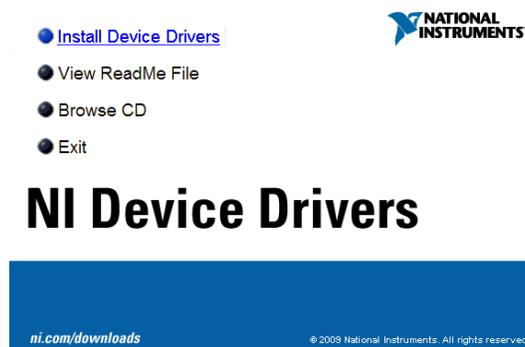


Figura 3.17 National Instruments Auto Run

Para continuar con la instalación se selecciona la opción Install Device Drivers con el cual se ejecuta el software de instalación en el cual en una primera instancia realiza una comprobación del sistema y del PC para asegurarse de la compatibilidad de la tarjeta y de

los drivers con la computadora, esto se demorará unos cuantos segundos si existe algún problema aparecerá algún mensaje caso contrario aparecerá una pantalla similar a la mostrada en la Figura 3.18 que se presenta a continuación.

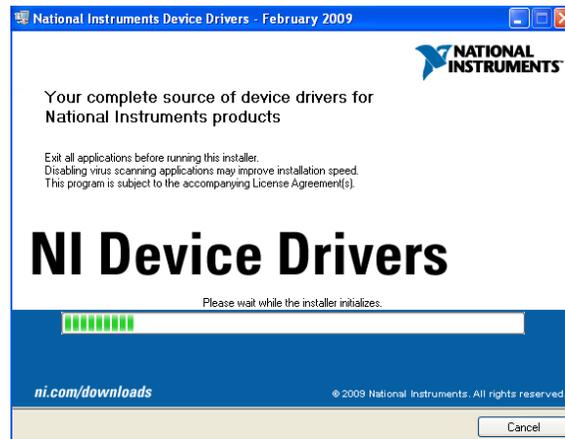


Figura 3.18 National Instruments Device Drivers

Una vez finalizada las pruebas de compatibilidad aparecerá una barra de selección de directorio en donde se instalará el software necesario para el funcionamiento de la tarjeta de la National Instrument, por defecto se llenará esta barra con el directorio sugerido por el fabricante pero si se requiere puede cambiarse, hacemos click en *Next* y continuamos.

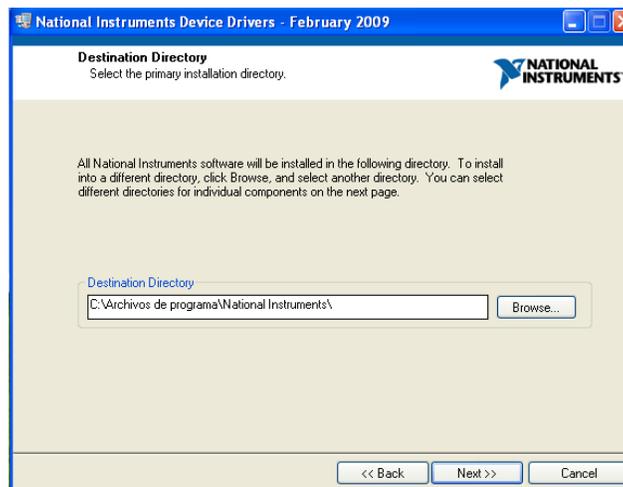


Figura 3.19 National Instruments Destination Directory

En la siguiente ventana podemos observar que se seleccionan los drivers y el software automáticamente de acuerdo con la tarjeta colocada en el computador, hacemos click en *Next* y continuamos.

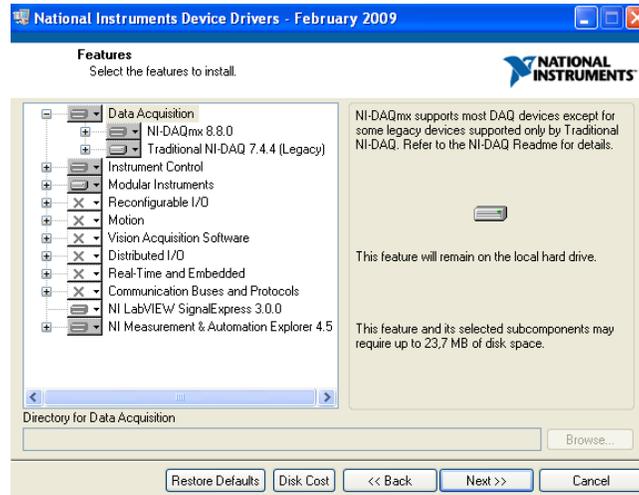


Figura 3.20 National Instruments Instalación Del Driver

En la siguiente ventana nos aparecerá una ventana de contrato en donde se especifica que National Instrument se reserva todos los derechos de software credo, una vez aceptado se continúa con la instalación haciendo click en *Next*.

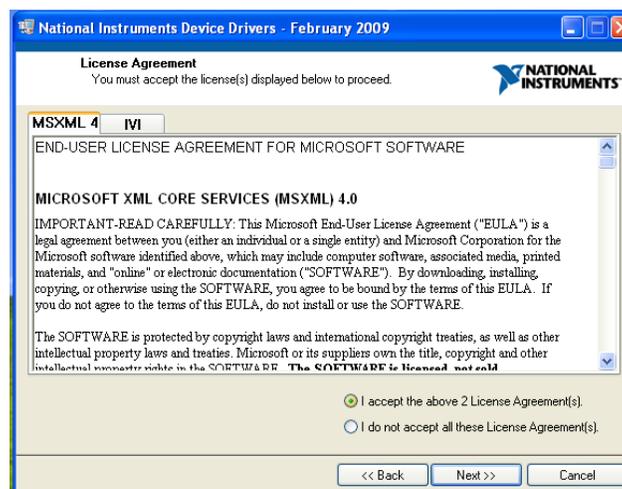


Figura 3.21 National Instruments License Agreement

Luego de aceptar el contrato se los presentará la siguiente ventana en donde se muestra el software que va a ser instalado, hacemos click en *Next* y continuamos con la instalación.

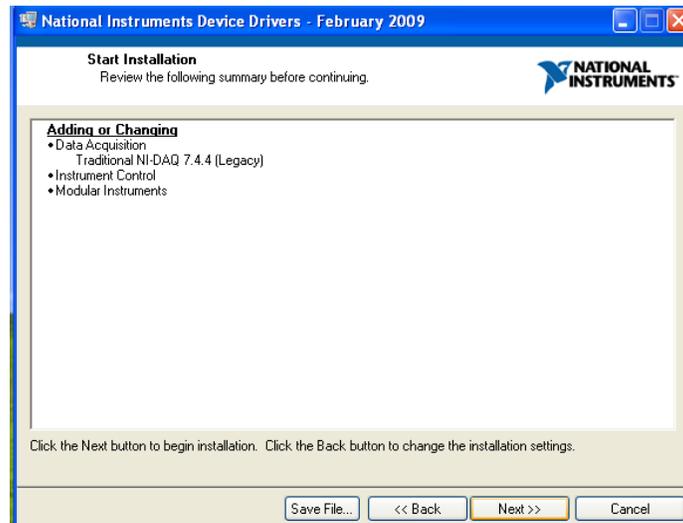


Figura 3.22 National Instruments Start Installation

En la siguiente ventana aparecerán las barras de progreso de la instalación, una vez terminado el proceso hacemos click en *Next* y continuamos con la instalación.

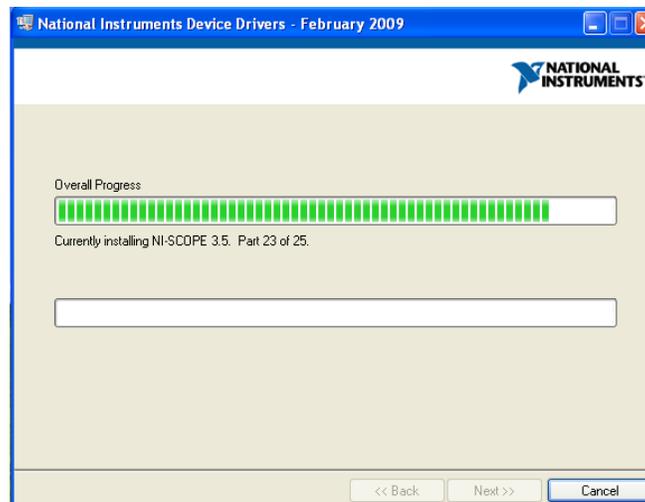


Figura 3.23 Progreso De La Instalación

Al concluir aparecerá una ventana mostrando que la instalación ha finalizado correctamente tal como se muestra en la Figura 3.24.

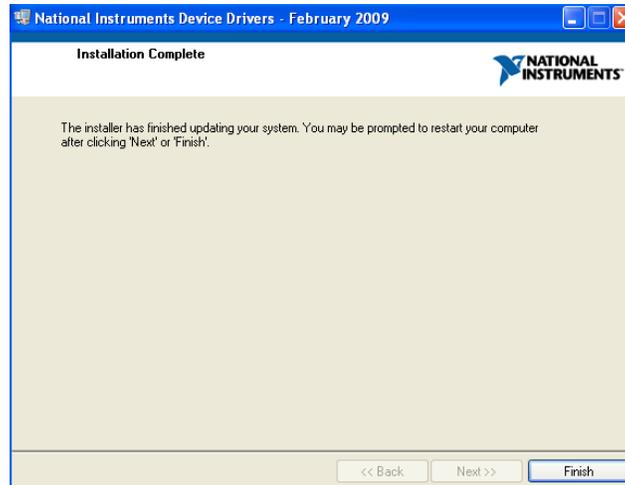


Figura 3.24 National Instruments Installation Complete

Damos click en la opción *Finish*, posteriormente nos aparecerá una ventana que nos sugiere apagar, reiniciar o reiniciar después, es recomendable una vez finalizada la instalación apagar el computador para que arranque nuevamente el sistema con las actualizaciones realizadas por el software instalado.

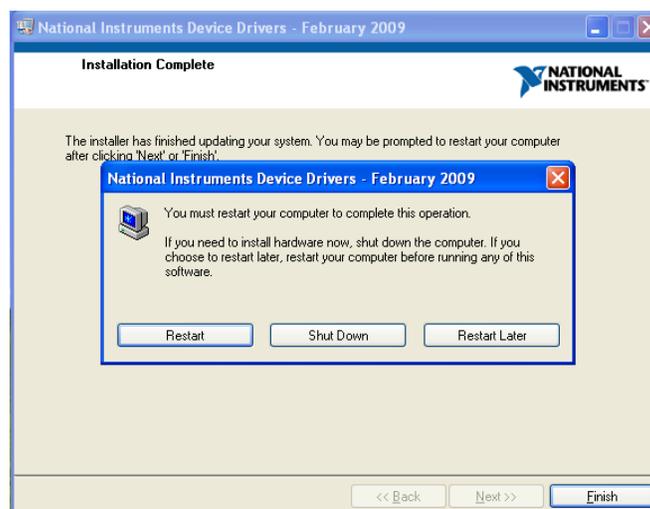


Figura 3.25 Ventana De Reinicialización Del Sistema

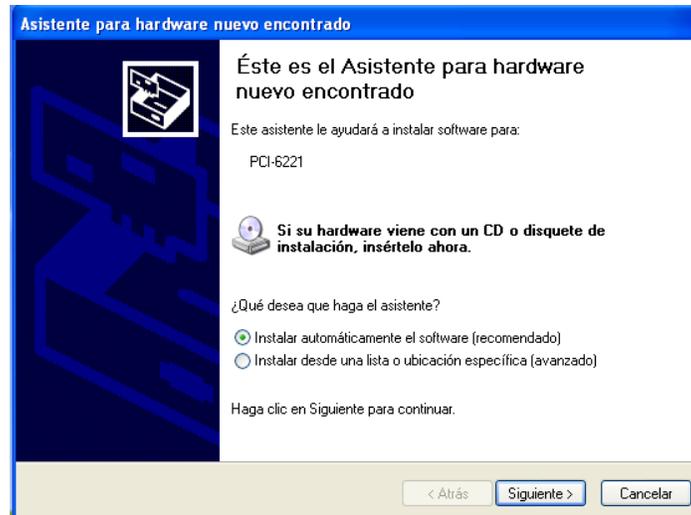


Figura 3.27 Asistente Para Hardware Nuevo Encontrado

A continuación se inicia la busca automática de los drivers necesarios para reconocer la tarjeta de adquisición de datos PCI 6221 como se puede apreciar en la Figura 3.28.

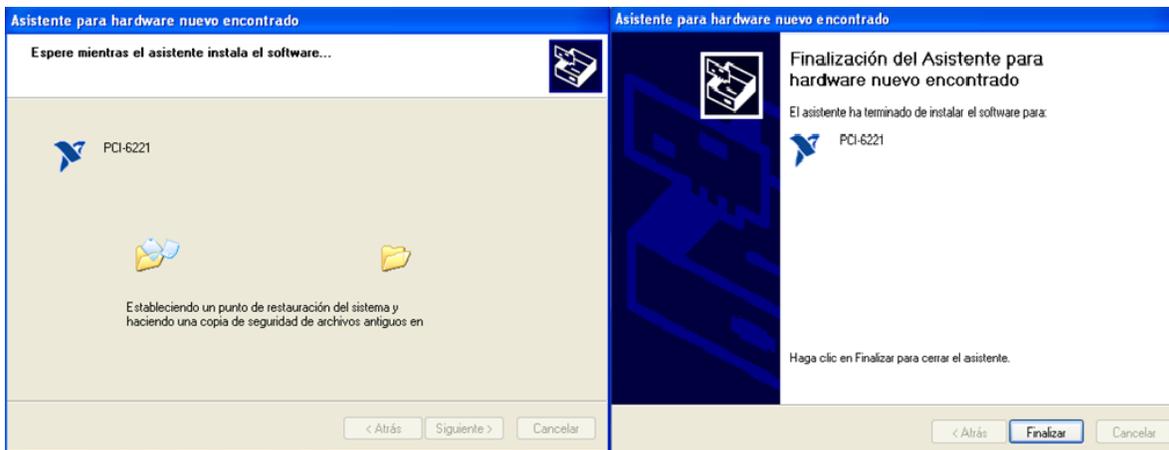


Figura 3.28 Finalización Del Asistente Para Hardware Nuevo Encontrado

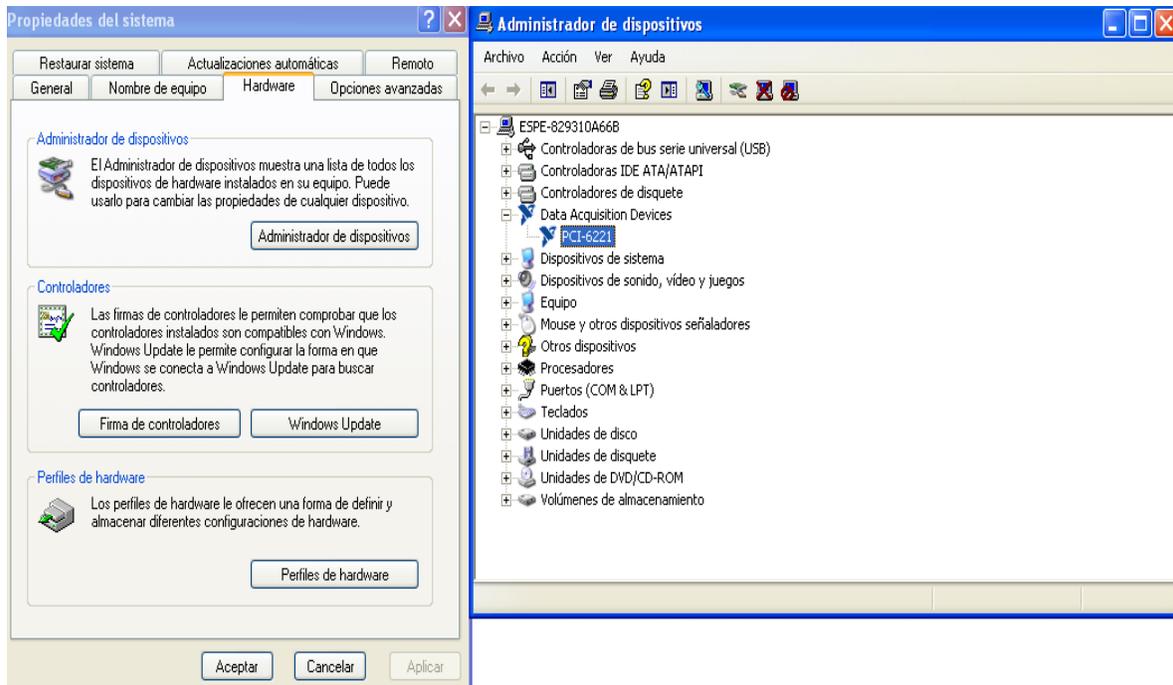


Figura 3.30 Administrador De Dispositivos

3. En la ventana del modelo, se escoge View, Library Browser para mostrar el buscador de la librería de Simulink[®].

4. Inserte una entrada y una salida analógica apropiada al modelo de la librería Real – Time Window Target tal como se muestra en la Figura 3.31.

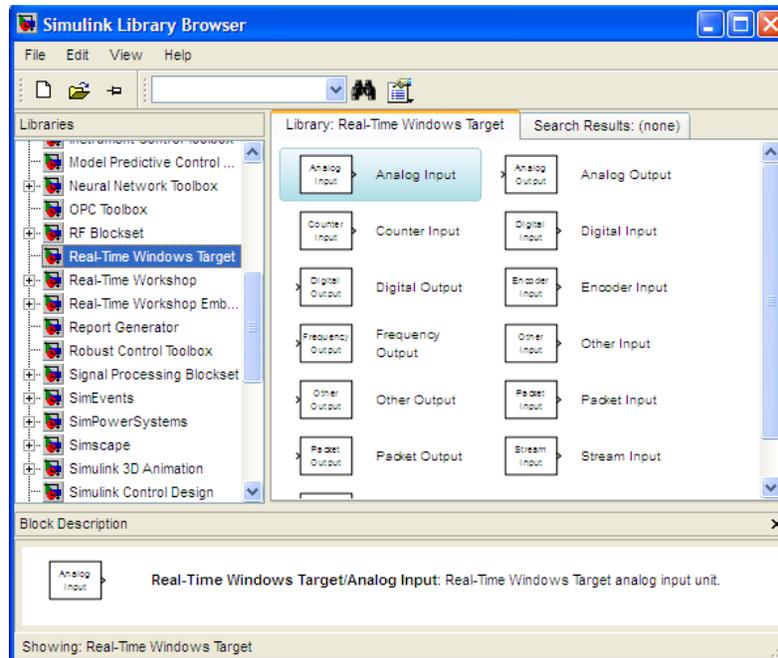


Figura 3.31 Simulink® Library Browser

5. Al dar doble click sobre el bloque de Entrada o de salida se abre el cuadro de dialogo en donde podemos instalar la tarjeta de adquisición de datos como se puede apreciar en la Figura 3.32.

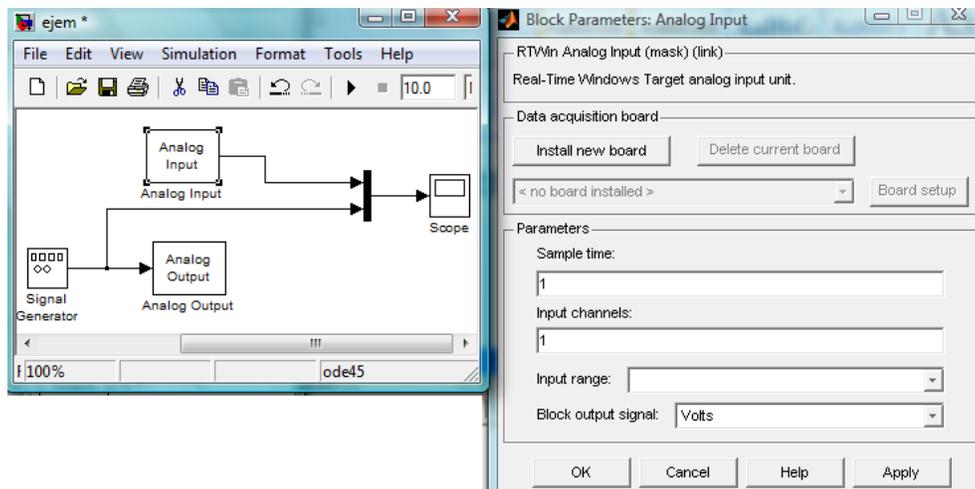


Figura 3.32 Block Parameters Analog Input

6. Se da click en el botón *install new board*. De la lista que aparece ponemos el cursor sobre el nombre del Fabricante, en este caso *National Instrument* y posteriormente seleccionamos el modelo de la tarjeta que para nuestro proyecto es la número PCI-6221, como podemos apreciar en la Figura 3.33.

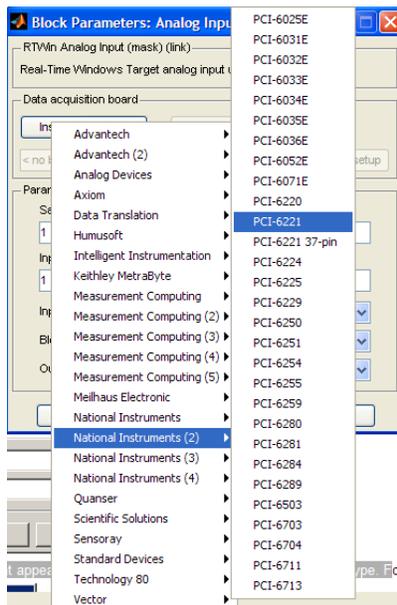


Figura 3.33 Selección De La Tarjeta De Adquisición De Datos

A continuación el Panel de Configuración de la Tarjeta de Adquisición de Datos aparece como podemos apreciar en la Figura 3.34.

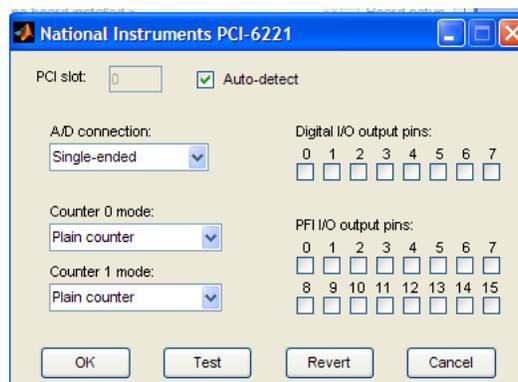


Figura 3.34 Ventana De Configuración De La Tarjeta De Adquisición De Datos PCI-6221

7. Seleccionamos el Slot PCI o ponemos *Auto – Detect* para que se reconozca.

8. Damos Click en el *Botón Test*.

El *real time window target kernel* intenta conectarse a la tarjeta seleccionada y si lo ha conseguido satisfactoriamente muestra el siguiente mensaje ilustrado en la Figura 3.35.

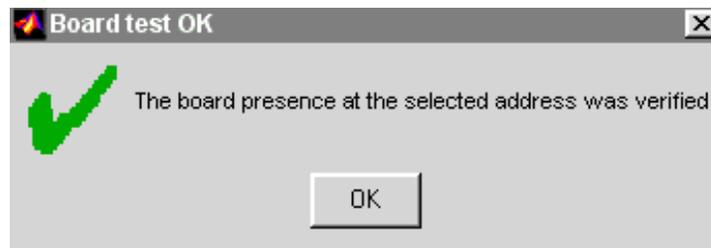


Figura 3.35 Mensaje de Prueba de Comunicación Satisfactoria con la Tarjeta de Adquisición de Datos

9. Por último damos click en Ok

3.2.5 Configuración de parámetros del RTW

Los parámetros de configuración dan al software Simulink® la información necesaria para ejecutar la simulación, en tal virtud tenemos:

1. En la ventana de Simulink®, se ingresa en el menú Simulación, a continuación *Configuration Parameters*. En el cuadro de diálogo de la Configuración de Parámetros se da click sobre la opción *Solver*.
2. En el campo *Start Time* se ingresa 0.0, en el campo *Stop Time* se ingresa el tiempo que se desee que se ejecute el modelo creado, por ejemplo ingresaremos el valor de 10.0.
3. En la lista *Type* se escoge la opción *Fixed-Step*. El software Real Time Workshop Code Generation no soporta variable step.

4. De la lista *Solver* seleccionamos un algoritmo. Por ejemplo escogemos el algoritmo de solución de propósito general ode5.
5. En el campo *Fixed Step Size*, se ingresa el tiempo de muestreo. Por ejemplo se ingresa 0.001 segundos para un tiempo de muestreo de 1000 muestras por segundo.
6. De la lista *Tasking Mode for Periodic samples times*: escogemos la opción *SingleTasking* (si en el modelo que se está elaborando existen bloques con diferentes tiempos de muestreo, se escoge la opción *MultiTasking*).
7. Finalmente seleccionamos *Apply* y *Ok* para aceptar y guardar los cambios tal como se muestra en la Figura 3.36.

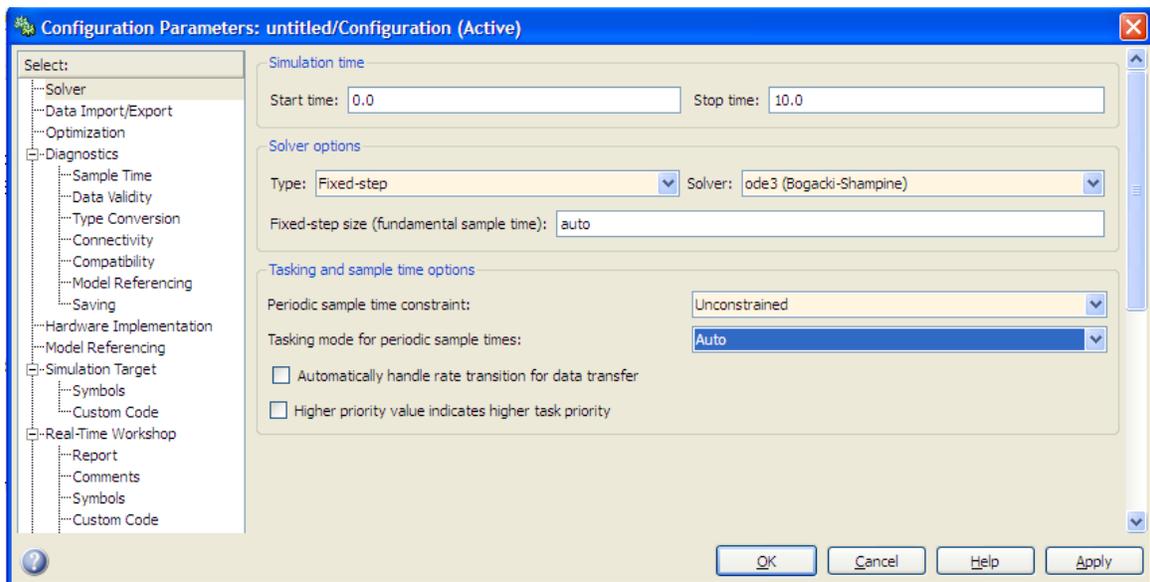


Figura 3.36 Ventana De Configuración del Solver

➤ **Configuración de los Parámetros del RTW**

1. En la ventana de Simulink®, damos click en el menú *Simulation*, y posteriormente en *Configuration Parameters*.
2. Seleccionamos la opción Real-Time Workshop.

Se abre la ventana de configuración del Real time Workshop

3. En la sección *Target Selection*, damos click en el botón *Browse* de la lista RTW *system target file*.

Se abre el sistema de exploración de archivos de destino.

4. Seleccionamos el sistema de identificación de archivos para construir una aplicación RTWT, y damos OK como podemos apreciar en la Figura 3.37.

<code>rsim.tlc</code>	Rapid Simulation Target
<code>rtwin.tlc</code>	Real-Time Windows Target
<code>rtwinert.tlc</code>	Real-Time Windows Target (ERT)
<code>rtwsfcn.tlc</code>	S-function Target

Figura 3.37 Archivos De Identificación Del Sistema

5. El cuadro de dialogo automáticamente ingresa el archivo de identificación de sistema `rtwin.tlc`, la plantilla de realización de archivo `rtwin.tmf` y el comando de realización `make_rtw` dentro del panel del Real - Time Workshop.

Aunque no se encuentre visible en panel Real – Time Workshop, cuando se da click en OK, también se puede configurar la interface de identificación externa MEX file rtwinext. Este modo externo permite pasar nuevos parámetros a la aplicación en tiempo real y retornar señales de datos de la aplicación en tiempo real. Los datos son mostrados en los bloques de osciloscopio o guardados con señales de identificación.

Por último se debe verificar que el lenguaje de compilación esté acorde con el compilador que vamos a utilizar que para nuestro caso será lenguaje C.

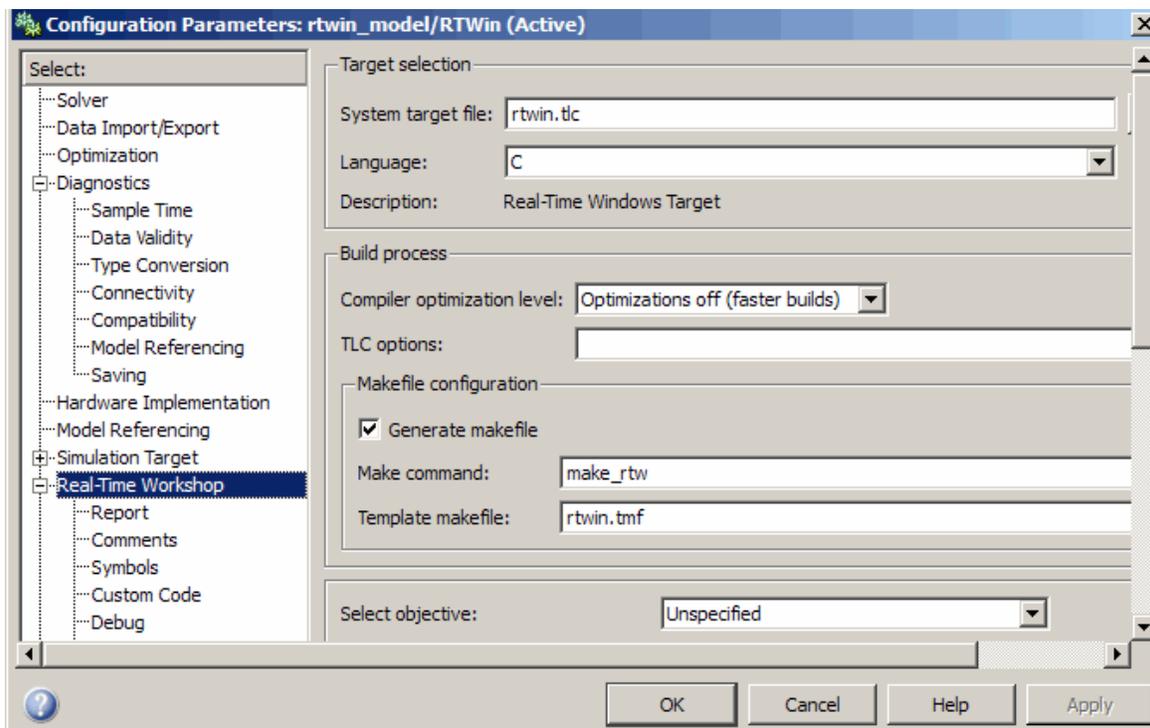


Figura 3.38 Panel Real Time Workshop

No se debe seleccionar Inline parameters en el nodo de Optimización. Los parámetros en línea son utilizados habitualmente como identificadores cuando se desea reducir la cantidad de RAM o ROM con sistemas integrados. También si se selecciona los parámetros en línea, se desactiva los parámetros característicos de autoajuste. Por estas razones es recomendable no ingresar a los parámetros en línea.

6. Se da click en el modo de *Hardware Implementation*. Los siguientes valores están configurados por defecto:

- Device vendor - Generic
- Device Type – 32 – bit X86 Compatible
- Emulation hardware – None

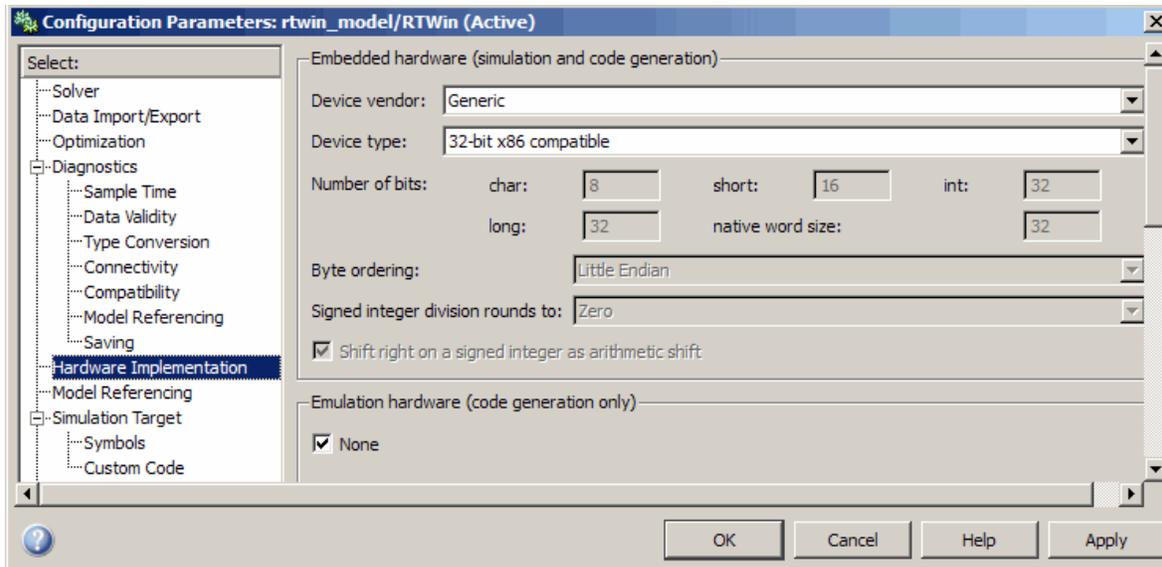


Figura 3.39 Panel De Implementación Del Hardware

7. Por último aplicamos los cambios dando click en Apply, guardamos y salimos dando click en Ok.

Como se mencionó anteriormente el RTW necesita de un compilador en C para crear los ficheros necesarios para realizar la aplicación en tiempo real por esto es necesario previamente tener instalado un compilador en lenguaje C específicamente Visual C++, para seleccionar el compilador se procede a dar un click en la opción Tools de la barra de tareas de Simulink®, luego seleccionamos la opción Real Time Workshop y escogemos la opción xPC Target Explorer como muestra la Figura 3.40.

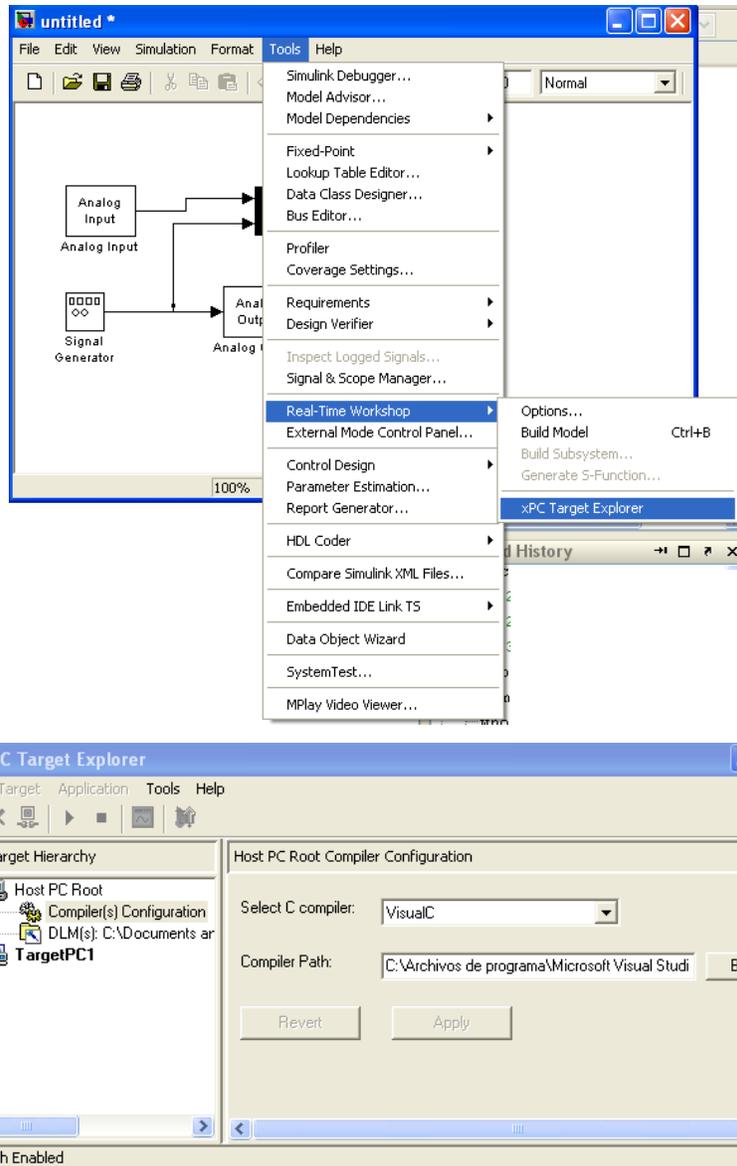


Figura 3.40 Selección Del Compilador

3.2.6 Configuración de las opciones de los elementos de visualización

- **Ingreso De Los Parámetros Del Osciloscopio.-** Primeramente se guardan los datos a través del bloque del osciloscopio que posee Matlab® (SCOPE) a su entorno.

1. En la ventana de Simulink® se da doble click sobre el bloque del osciloscopio (SCOPE) y una ventana de osciloscopio aparece, tal como se muestra en la Figura 3.41.

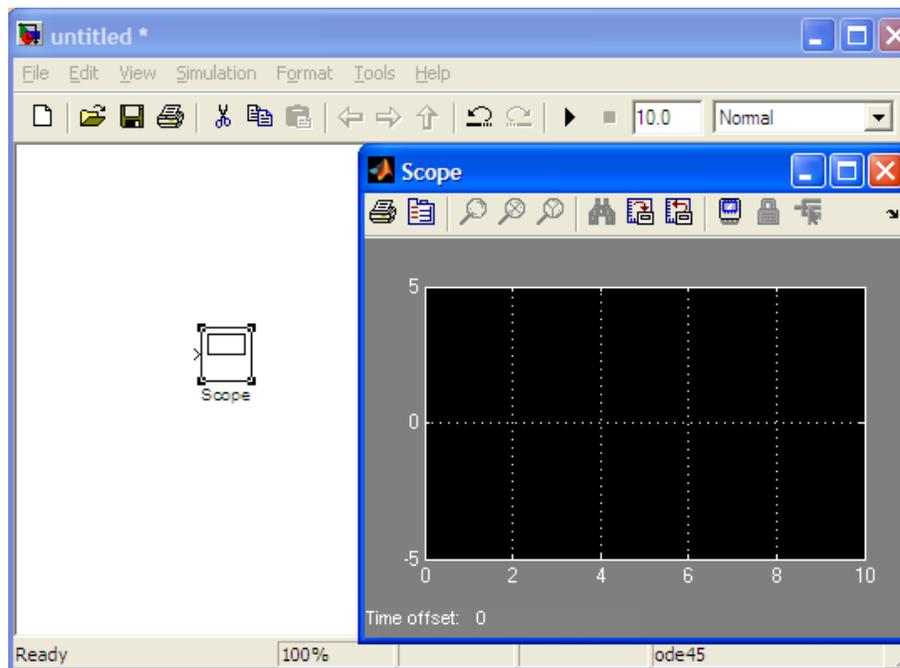


Figura 3.41 Scope

2. En la barra de herramientas se da click sobre el ícono de parámetros, tal como se muestra en la Figura 3.42.



Figura 3.42 Parámetros del Scope

3. A continuación una caja de parámetros aparece, tal como se muestra en la Figura 3.43.

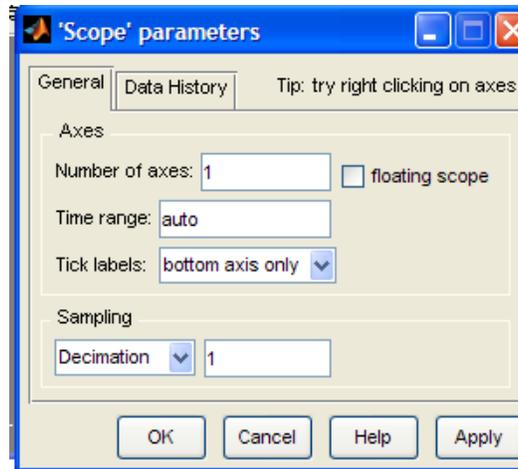


Figura 3.43 Caja de Parámetros del Scope

4. Se da click en *Data History*
5. Se realiza una de las siguientes opciones.
 - a. Si está corriendo una simulación se puede seleccionar la opción *Limit data points to last* (Límite de duración de los datos) y se ingresa el número de puntos para grabar.
 - b. Si está corriendo una aplicación en tiempo real, no se selecciona *Limit data points to last*, porque los límites de puntos de datos están dados por la configuración del *External Signal & Triggering*.
6. Se selecciona *Save data to workspace* (Guardar archivos al entorno), en *Variable name* (Nombre de la Variable) se ingresa el nombre con el cual se grabará en Matlab®. El nombre por defecto es *ScopeData*.
7. El formato que se selecciona para grabar señales o muestras en el tiempo es *Structure with time*, también se puede seleccionar un array, si es un flujo de información o un structure, si se requiere.
8. Posteriormente se selecciona *Apply* para aprobar los cambios realizados y *Ok* para aprobar y cerrar todo el cuadro de parámetros, tal como se muestra en la Figura 3.44.

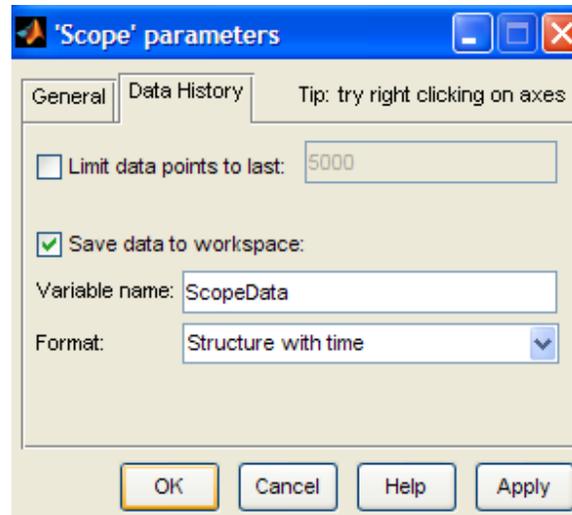


Figura 3.44 Definición de Parámetros del Scope

9. Por último se debe reconstruir la aplicación en tiempo real antes de conectarse con la aplicación, si esto no se realiza aparecerá un error.

➤ **Propiedades del External Signal & Triggering (Disparo y Señales Externas).**-

Las propiedades de las señales externas y de disparo deben ser configurados antes de ejecutar una aplicación en tiempo real, en esta misma configuración se determinará el límite de muestras que se almacenará en el entorno de SIMULINK[®], teniendo que configurarse de la siguiente manera:

1. En la ventana de Simulink[®] se da click en la pestaña Tools (Herramientas), posteriormente se da click en External Mode Control Panel, como se muestra en la Figura 3.45.

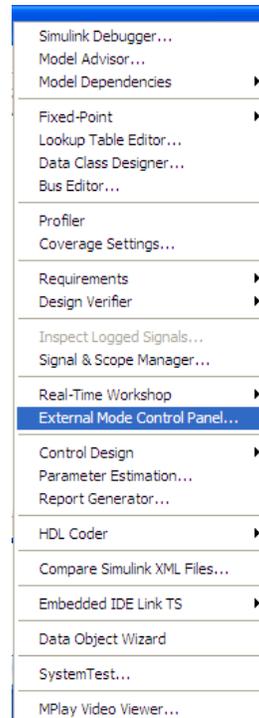


Figura 3.45 Ingreso al External Mode Control Panel

2. Se da click en el botón Signal & Triggering en donde se desplegará una caja de opciones tal como se muestra en la Figura 3.46.

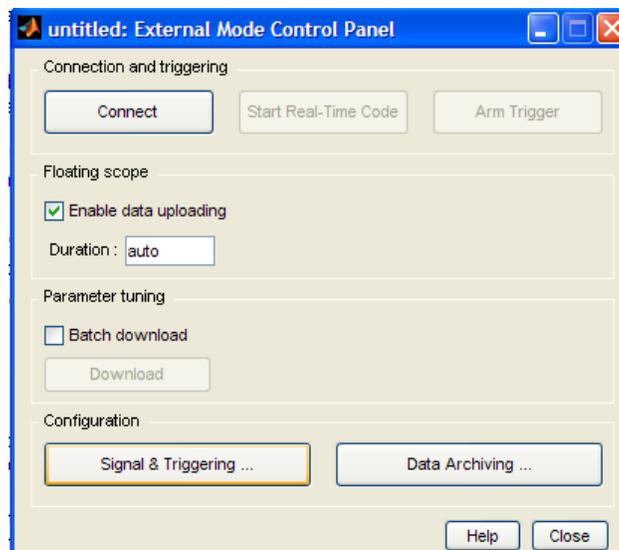


Figura 3.46 Signal & Triggering

3. Aparece una ventana de configuración a continuación seleccionamos el check list que dice Select all, en trigger seleccionamos Source Manual y Modo Normal, en Duration se determina el tiempo que va a almacenar información el osciloscopio, y el retardo igualmente determinará si deseamos un retardo en el osciloscopio.

4. La X que aparece en la parte inferior de la etiqueta Signal Selección, indica que la señal ha sido marcada para recolección de datos. Si aparece T simboliza que la señal está etiquetada como señal de disparo.

En el campo de duración se escribe el número de puntos de muestra en un buffer de datos. Por ejemplo, si se tiene una frecuencia de muestreo de 1000 muestras por segundo y un tiempo de parada de 10 segundos, a continuación, se deberá ingresar 10000.

El valor de Duration (duración) está relacionado con el valor del Limit Data Points to Last en los parámetros del osciloscopio, este valor especifica el número de puntos contiguos de los Datos recogidos por cada buffer de Datos. Se recomienda que se introduzca un valor de duración igual al número total de puntos de muestreo para que solo se guarden los datos que han corrido en la aplicación.

Si se selecciona un número menor que el número total de muestras se perderá información y si se selecciona un valor mayor se tendrá valores vacíos.

Es recomendable también siempre configurar el valor de los ejes de los tiempos de los osciloscopios de Simulink[®] con los valores de muestreo para observar todos los puntos en cada buffer de datos, tal como se puede apreciar en la Figura 3.47.

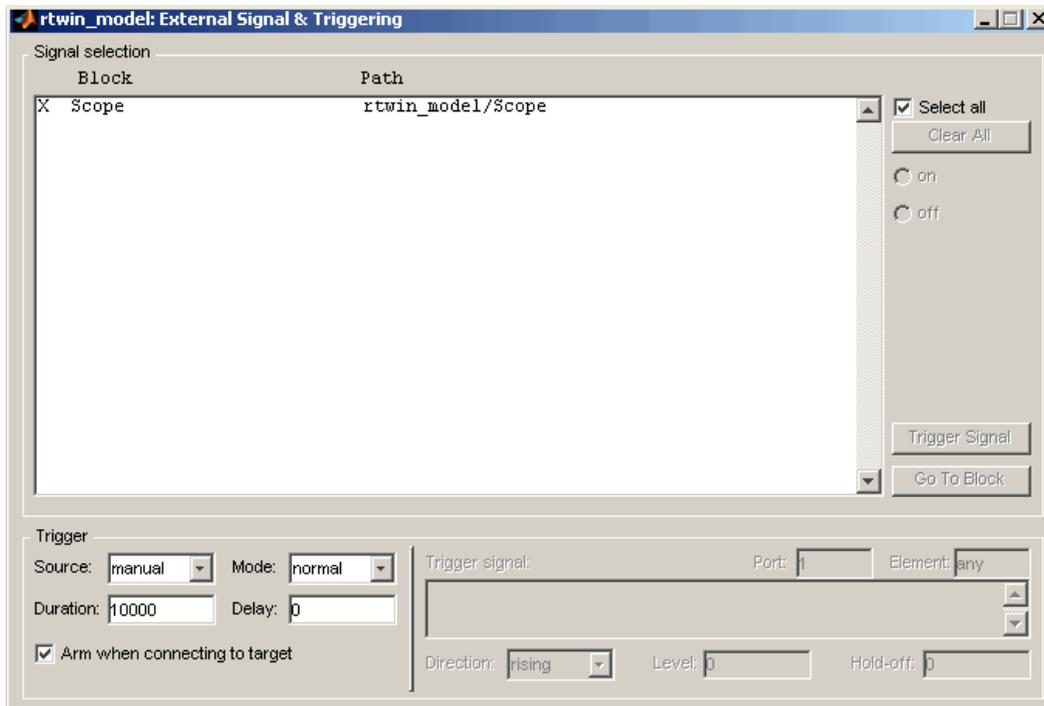


Figura 3.47 Configuración del External Signal & Triggering

Finalmente se deben aceptar todos los cambios y cerrar la ventana

- **Configuración de Parámetros de Data Archiving (Archivo de Datos).**- Los parámetros de configuración del Archivo de datos están relacionados con los parámetros del osciloscopio. En la configuración de los parámetros del osciloscopio se debe seleccionar la opción Save data to workspace (Guardar archivos a área de trabajo) para que:
 - Los datos se transfieren primero desde el buffer de datos del osciloscopio al área de trabajo de MATLAB®, antes de escribir un archivo MAT.
 - El Variable name (nombre de la variable) que se ingresó en la configuración de parámetros del osciloscopio es el mismo que se tiene en el área de trabajo de Matlab® y

es el mismo que se tiene como un archivo MAT. Este nombre también permite llamar a esta variable MAT desde la ventana de comandos o en cualquier entorno de Matlab®.

- Si no se seleccionó Save data to workspace en la configuración del osciloscopio se generará el archivo pero este va a estar vacío.

Después de estas consideraciones se puede ingresar a la configuración de Data Archiving para grabar la información en una unidad de disco.

1. En la ventana de Simulink® se da click en la pestaña Tools (Herramientas), posteriormente se da click en External Mode Control Panel, tal como se mostró en la Figura 3.45.

2. Se da Click en el botón Data Archiving, como se muestra en la Figura 3.48.

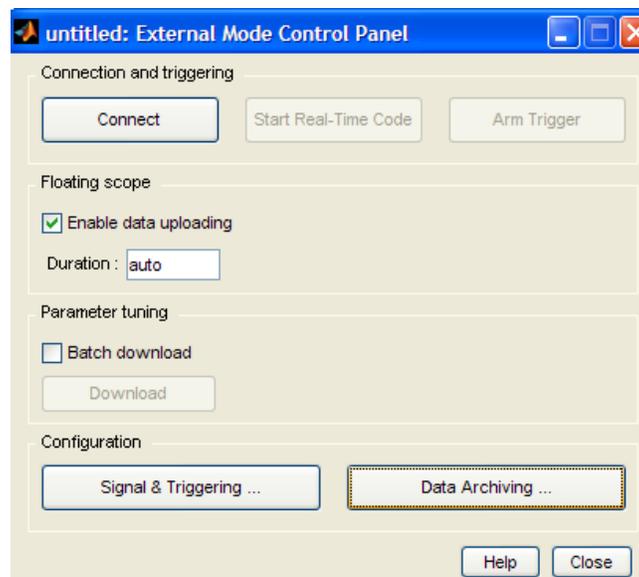


Figura 3.48 Ingreso al Data Archiving

3. Seleccionamos la casilla de verificación Enable Archiving (Habilitar Archivo).
4. En la caja de texto del Directory (Directorio), ingresamos la ruta del directorio de la unidad de disco, en donde se grabará el archivo por ejemplo: c:\ESPE

5. En la caja de texto File (Archivo), ingresamos el nombre o prefijo con el cuál guardaremos los archivos. Por ejemplo, ingresamos: PID

El entorno de MATLAB® nombrará PID_0.mat, PID_1.mat, etc. El número de archivos será igual al número de periodos de muestreo que se ejecuto. Por ejemplo si se configuró el Duration para todos los puntos de muestreo, entonces se generará solo un archivo.

6. Seleccionamos la casilla de verificación Append File suffix to variable names tal como se puede apreciar en la Figura 3.49.

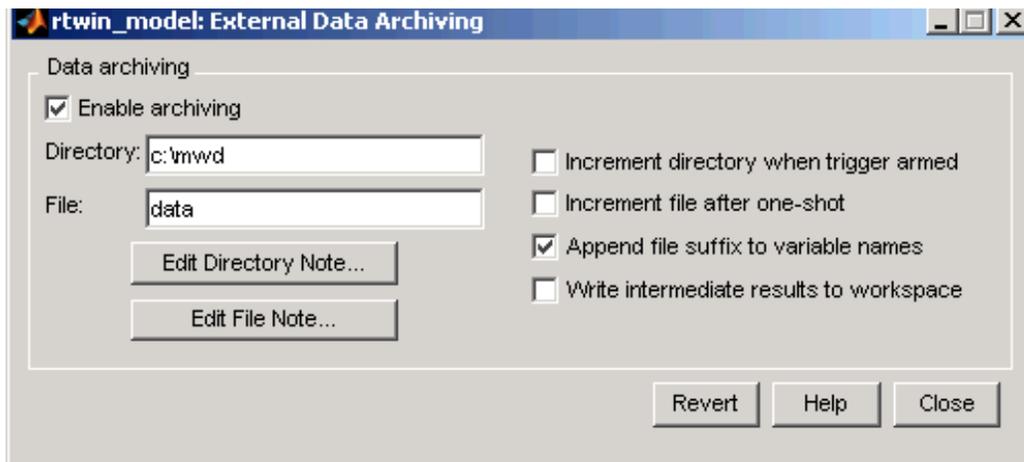


Figura 3.49 Configuración Del Data Archiving

En esta ventana no existe el botón APPLY por lo que al poner el botón Close se guardan los cambios.

Cabe indicar que dentro de cada archivo MAT, una variable es guardada con el mismo sufijo por ejemplo, dentro de PID_0.mat estará la variable ScopeData_0.

- **Gráfica de los Archivos Almacenados.**- Se puede utilizar las funciones de MATLAB® para visualizar los archivos guardados de una aplicación en tiempo real, para lograrlo se debe de seguir el siguiente procedimiento:

1. En la ventana de comandos de MATLAB[®] se escribe: ScopeData (O el nombre de la variable que hayamos guardado).

MATLAB[®] devolverá la descripción de esta estructura con tiempo donde se tienen los campos time, el cuál es un vector, en signals, contendrá los valores de la señal y por último se tiene un string con el nombre del bloque.

```
ScopeData =  
time: [10000x1 double]  
signals: [1x1 struct]  
blockName: 'rtwin_model/Scope'
```

2. Para visualizar todos los archivos MAT guardados en la unidad de disco, se debe escribir: dir *.mat

3. La ventana de comandos de MATLAB[®] mostrará los archivos MAT en el directorio en el cual actualmente se encuentra trabajando.

4. Para limpiar el espacio de trabajo se utiliza el comando clear

5. Para cargar un archivo MAT se utiliza el comando load seguido del archivo MAT que utilizaremos, por ejemplo: load ScopeData

6. Y whos muestra todas las variables que tiene el entorno de trabajo de MATLAB[®]

7. Para graficar utilizamos la función plot con cada una de las partes de la estructura, por ejemplo:

```
plot(ScopeData.time(1:1000), ScopeData_0.signals.values(1:1000))
```

Donde graficaremos las 1000 primeras muestras. El entorno de MATLAB[®] graficará las 1000 primeras muestras desde 0,0000 a 0,9990 segundos, obteniendo una grafica como la de la Figura 3.50

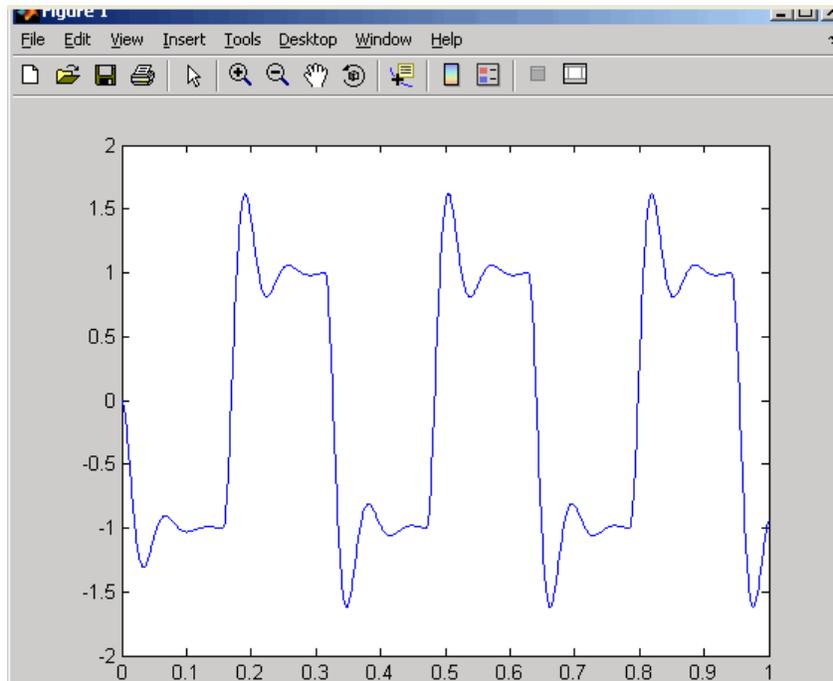


Figura 3.50 Gráfica de Archivo Almacenado

3.2.7 Ejecución de la aplicación en tiempo real

Por ultimo para la ejecución en tiempo real se debe cambiar el modo de simulación en la barra de Menús en la opción de simulación a External tal como se muestra en la Figura 3.51.

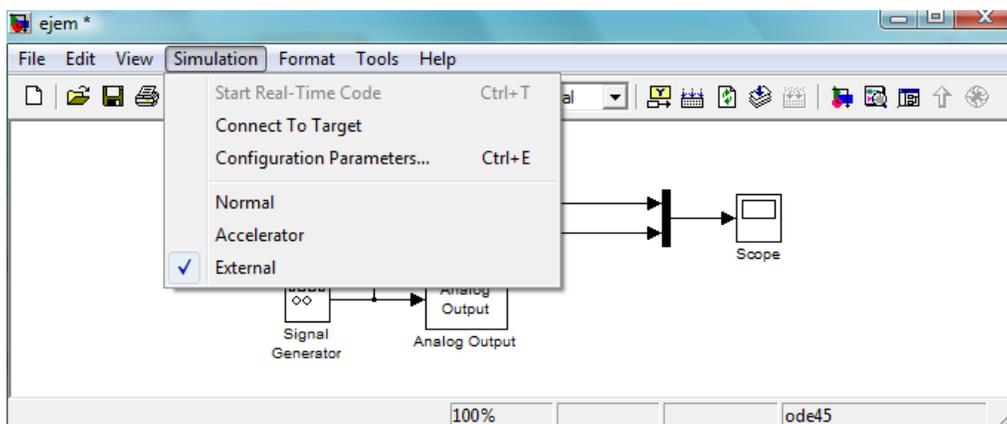


Figura 3.51 Cambio De Modo De Simulación

Posteriormente en la barra de tareas debe aparecer la opción Connect To Target como se aprecia en la Figura 3.52.

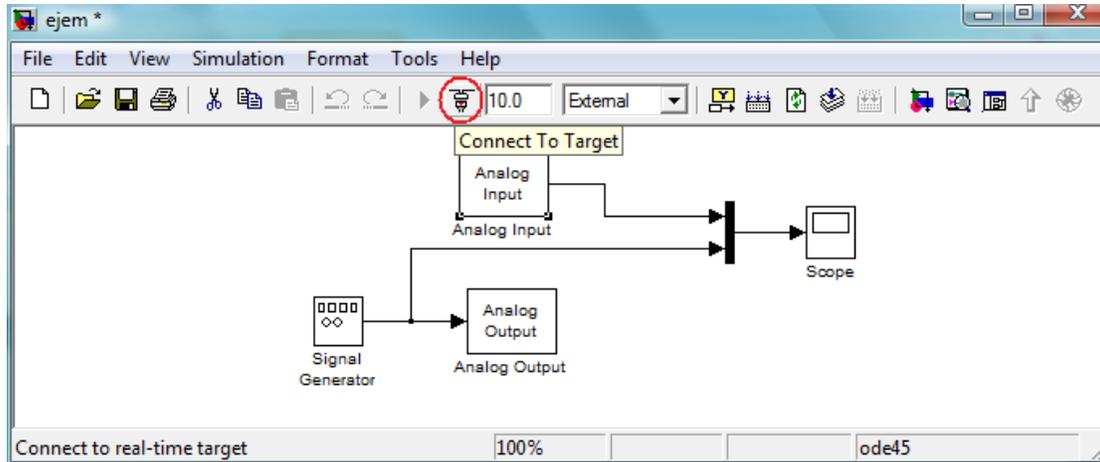


Figura 3.52 Icono Connect To Target

Posteriormente se debe construir el modelo en Tools, Real Time Workshop, Build Model o en su defecto presionando las teclas Ctrl+B como se puede apreciar en la Figura 3.53, cabe recalcar que se tiene que construir el modelo cada vez que se realicen cambios.

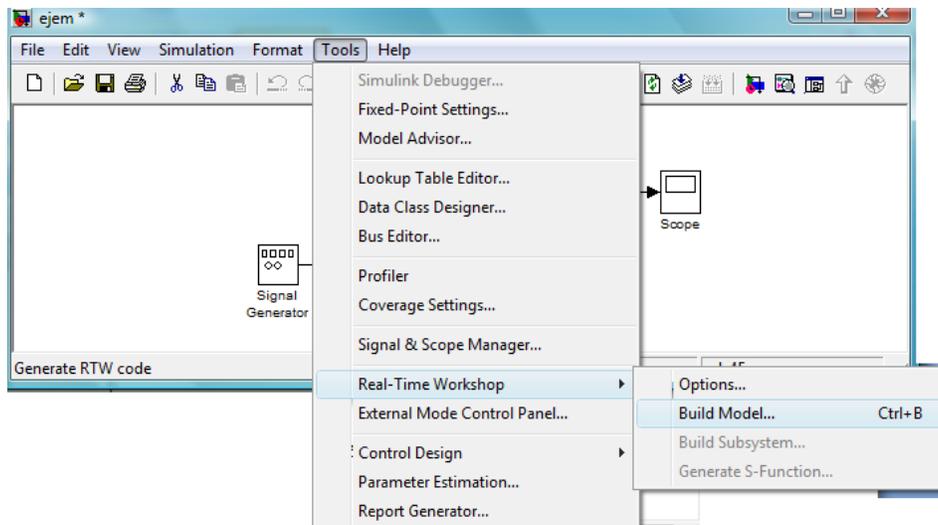


Figura 3.53 Construcción Del Modelo En Tiempo Real

Una vez realizada la construcción del modelo damos click sobre el ícono Connect To Target y posteriormente en el ícono de Run como se muestra en la Figura 3.54.

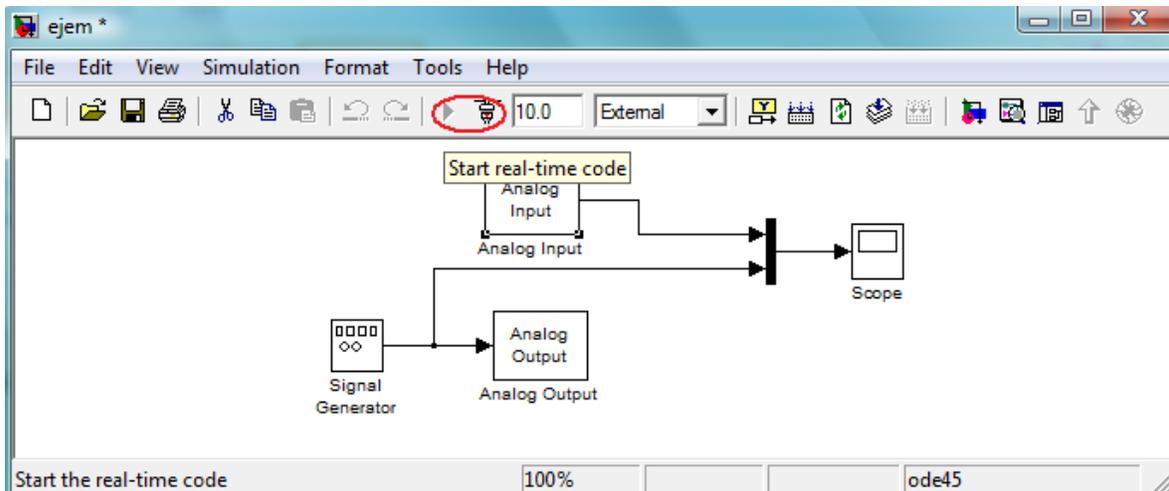


Figura 3.54 Ejecución Del Modelo En Tiempo Real

Al ejecutarse la aplicación en tiempo real la simulación deberá tener el siguiente comportamiento tal como muestra la Figura 3.55, en donde una barra de estado se incrementará constantemente en la parte inferior de la ventana, pudiéndose además observar el tiempo transcurrido de ejecución de la simulación.

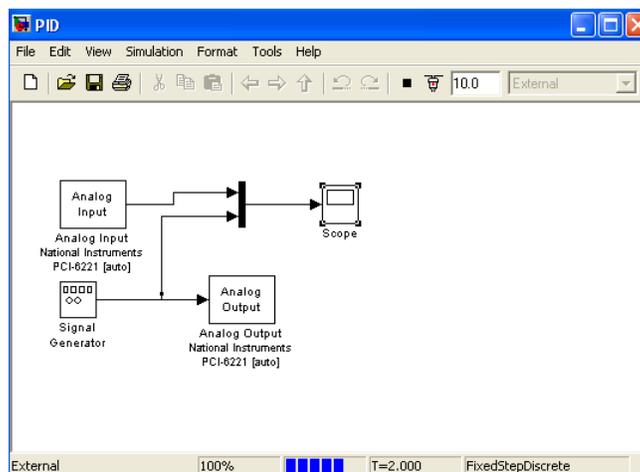


Figura 3.55 Comportamiento De La Ejecución Del Modelo En Tiempo Real

3.2.8 Obtención de resultados

Una vez que hemos seguido todos los pasos anteriormente detallados se obtiene como resultado de la simulación de ejemplo la Figura 3.56

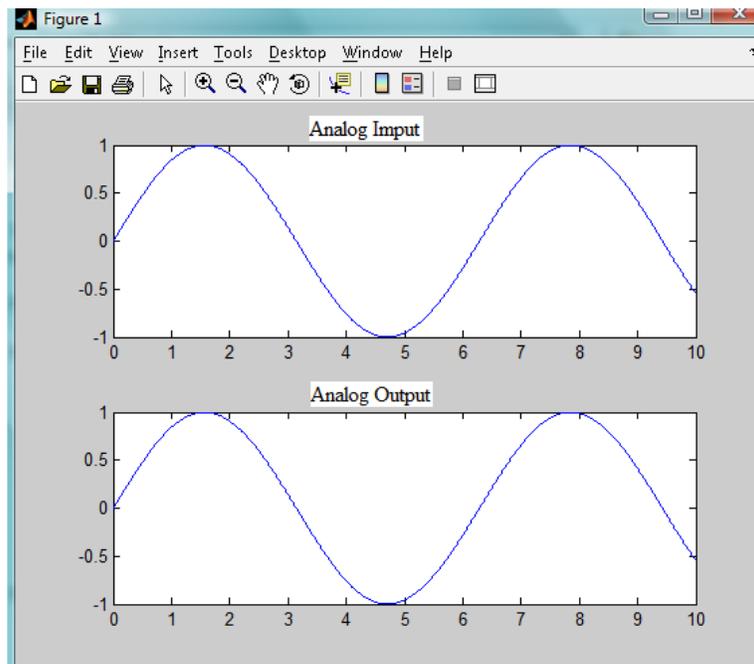


Figura 3.56 Gráfica Resultante Del Ejemplo De Simulación En Tiempo Real

3.2.9 Desinstalación Del Real Time Kernel

Si se encuentran algunos problemas con el software Real – Time Windows Target, se lo puede desinstalar. Una vez desinstalado, el kernel no estará más activo y no tendrá impacto en la operación del computador. Como los instaladores están grabados en el disco duro se lo podrá reinstalar fácilmente después, entonces el procedimiento de desinstalación es el siguiente:

1. En la ventana de comandos de Matlab® o en una ventana de DOS se escribe:

```
Rtwintgt –uninstall
```

O en su defecto también se puede hacer:

- a. Click en el botón de Start.
- b. Se selecciona Links and Targets> Real – Time Windows Target > Uninstall real – time kernel, tal como se muestra en la Figura 3.57.

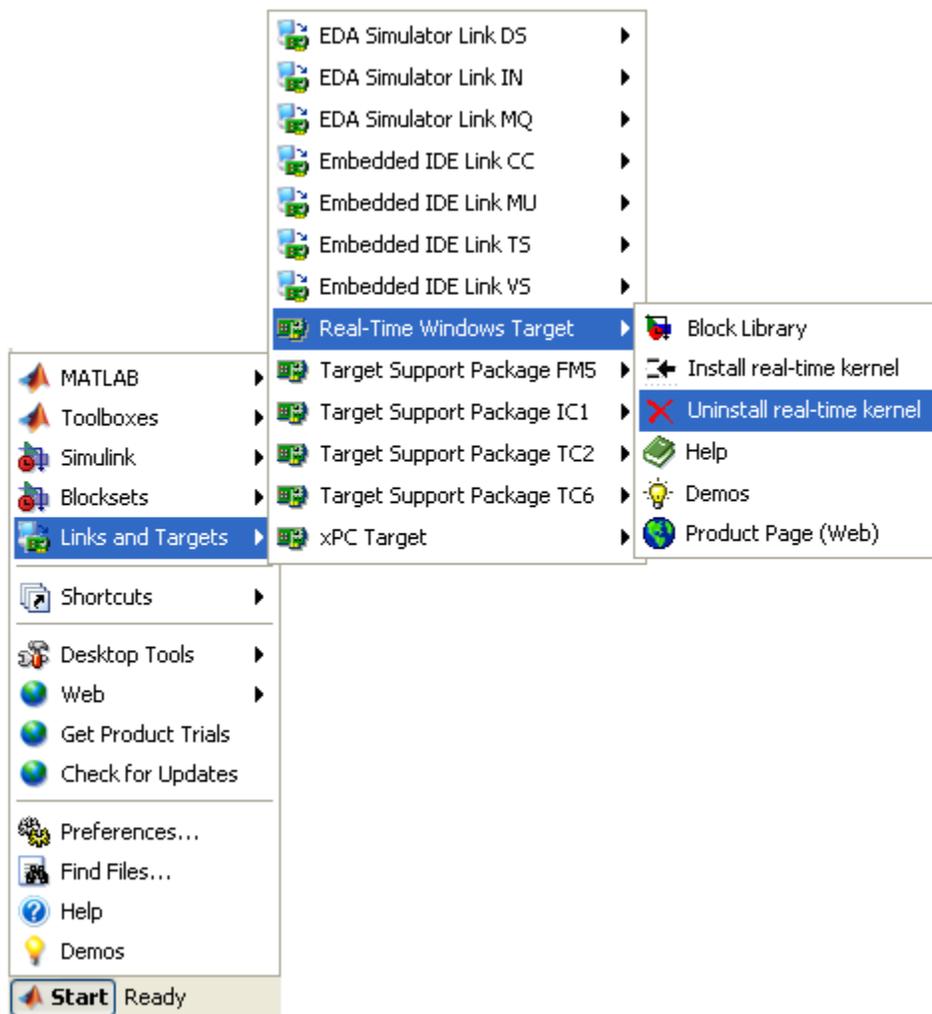


Figura 3.57 Desinstalación del Real Time Kernel

La ventana de comandos de Matlab® muestra el siguiente mensaje:

You are going to uninstall the Real – Time Windows Target kernel.
Do you want to proceed? [Y]:

2. Escriba Y para continuar desinstalando el Kernel, o N para cancelar la instalación sin hacer ningún cambio.

Si se digito Y, el ambiente de Matlab® desinstala el kernel removiéndolo de la memoria, entonces muestra este mensaje:

The Real – Time Windows Target kernel has been succesfully uninstalled.

3. Después de desinstalar el kernel, hay que verificar que este fue correctamente desinstalado escribiendo rtwho

El entorno de Matlab® debe mostrar el siguiente mensaje.

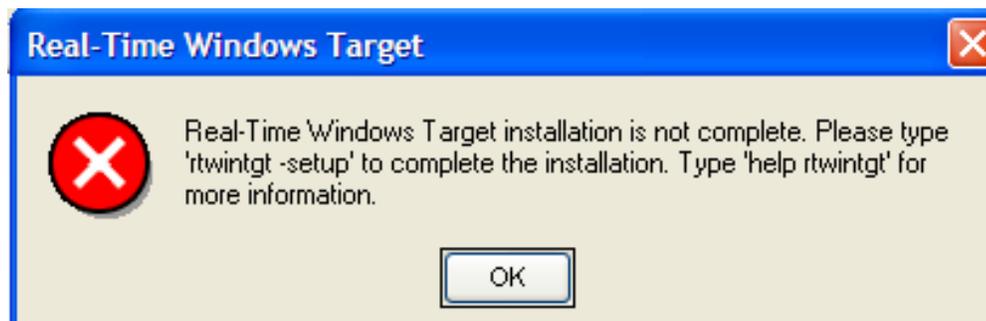


Figura 3.58 Verificación de la Desinstalación del Real Time Kernel

CAPÍTULO 4

APLICACIÓN DEL SISTEMA DE CONTROL RTW EN UN CONTROL PID

4.1 ESPECIFICACIONES DEL PROBLEMA

Se busca diseñar e implementar un sistema controlador de temperatura PID para la unidad *Air Flow Temperature Control System* mediante la utilización de la herramienta RTW de Matlab® con un error menor que $\pm 0,2^{\circ}\text{C}$ sobre la lectura actual, y que además posea un tiempo de establecimiento menor que 2 min. El control debe tener la capacidad de soportar las perturbaciones que en el módulo se provocan mediante la apertura del orificio de entrada de aire.

Como el principal requerimiento es que el error de lectura actual no sea mayor a $\pm 0.2^{\circ}\text{C}$; tenemos el requerimiento de un valor de PV que no sea mayor o menor a $\text{SP} \pm 0.2^{\circ}\text{C}$. Para interpretar este error en forma de error relativo porcentual ($\text{er}(\%)$), se plantea la Tabla 4.1 en donde se calculan cinco muestras para obtener un promedio del error en estado estable:

Set Point (SP)	$\text{er} = 0.2/\text{SP}$	$\text{er}(\%)$
50	0,004	0,4
51	0,003921569	0,39215686
52	0,003846154	0,38461538
53	0,003773585	0,37735849
54	0,003703704	0,37037037
55	0,003636364	0,36363636

Tabla 4.1 Cálculo Del Error En Estado Estable

De aquí, que el error relativo porcentual requerido (el error en estado estable) es de 0.36%. El resto de datos de entrada para este diseño se seleccionan mediante el criterio de

Ts sea menor que 2 min y que el mayor sobrepico sea $< 25\%$, así que se establecen los siguientes objetivos de diseño:

- Porcentaje de sobrepico: $\%MP < 25\%$
- Tiempo de estabilidad: $t_s < 2 \text{ min.}$
- Error de estado estable a la entrada escalón: $e_{ss} \text{ de step} \leq 0.36\%$

4.2 MODELO DE LA PLANTA

4.2.1 Interpretación Física De La Planta

La unidad *Air Flow Temperature Control System* es un sistema abierto ya que existe un intercambio de masa y energía con el ambiente puesto que al cambiar las perturbaciones mediante la placa se varía el intercambio de calor entre aire caliente y aire frío.

Por definición de equilibrio térmico toda sustancia por encima de los 0 grados Kelvin, ($-273,15 \text{ }^\circ\text{C}$) emite calor [6], si dos sustancias en contacto se encuentran a diferente temperatura una de ellas emitirá más calor y calentará a la más fría tendiendo a un equilibrio térmico que se alcanzaría cuando ambas sustancias emitan y reciban la misma cantidad de calor por gramo.

Las variables termodinámicas que tienen relación con nuestro sistema son la masa de aire (variable por las perturbaciones), el volumen (constante por la longitud y el área del tubo), la densidad y presión (constantes por ser la misma sustancia dentro del tubo) y la temperatura (variable por la niquelina y las perturbaciones).

La propagación del calor en la planta se realiza mediante conducción (transferencia de energía calorífica de partícula a partícula dentro del mismo cuerpo), por convección (transmisión de calor por las corrientes de aire) y radiación (transmisión de calor por movimiento vibratorio de las partículas).

Dentro de la planta el aire es calentado por conducción mediante una niquelina allí ascenderá por convección y se pondrá en contacto con el ventilador cediendo parte de su calor al aire frío volviendo hacia el ventilador para que posteriormente sea enviado de forma laminar (todas las moléculas pasan ordenadamente dentro del tubo) al tubo en donde se verá afectado también por fenómenos de convección y radiación de la niquelina.

Nuestro sistema se rige a las leyes de conservación, tales como la ley de la conservación de la masa [7], la cual establece que la materia es indestructible, aun cuando la teoría de la relatividad de Einstein postula que la masa puede transformarse en energía esto solo ocurriría si se mueve la materia a la velocidad de la luz, como se enuncia en su famosa fórmula $E=m*c^2$ como la velocidad a la cual se mueve el aire dentro de nuestro sistema es muy por debajo de la velocidad de la luz esta no se verá afectada por las variaciones de velocidad.

La segunda ley fundamental que se consideró fue la ley de la conservación de la cantidad de movimiento lineal en la cual se postula que la cantidad de movimiento lineal de un sistema permanece constante si no hay fuerzas externas que actúen en él, como la velocidad del aire es igual antes y después de pasar por la niquelina se sabría que la energía cinética inicial y final es la misma por lo que no afectará a la transmisión de energía (calor) del sistema.

Por último se consideró la primera ley de la termodinámica, también conocida como principio de conservación de la energía para la termodinámica, la cual establece que si se realiza un trabajo sobre un sistema o bien éste intercambia calor con otro, la energía interna del sistema cambiará.

La ecuación general de la conservación de la energía es la siguiente:

$$\text{Energía que entra} - \text{Energía que sale} = \Delta \text{Energía del Sistema}$$

Como se menciono anteriormente las únicas Energías que interaccionan son el Calor dado por la niquelina y la temperatura ambiente del aire.

Obteniendo la siguiente ecuación.

$$Q \text{ de la niquelina} - Q \text{ del ambiente} = \Delta Q \text{ del Sistema}$$

Para interpretar de una mejor manera lo anteriormente expuesto se utilizo las siguientes ecuaciones.

$$P = V * I \quad \text{Ec(1)}$$

$$V = I * R \quad \text{Ec(2)}$$

Reemplazando la Ec(2) en la Ec(1):

$$P = V^2 / R \quad \text{Ec(3)}$$

La ecuación tres es conocida como el efecto Joule.

Ahora por definición se tiene que:

$$P = E / t \quad \text{Ec(4)}$$

Reemplazando la Ec(4) en la Ec(3) obtenemos la siguiente ecuación.

$$\text{Energía} = \text{Calor} = (\text{Voltaje}^2 * \text{Tiempo}) / \text{Resistencia}$$

$$E = q = (V^2 * t) / R \quad \text{Ec(5)}$$

La interpretación de esta ecuación sería que nosotros podemos generar mayor calor al variar el voltaje, puesto que nosotros no podemos variar la resistencia de la niquelina ni el tiempo que va a permanecer el aire en contacto con la niquelina porque el motor del

ventilador se encuentra en constante movimiento, independientemente del intercambio de calor que se realice, por lo tanto el tiempo que se demora en interactuar la masa de aire con la niquelina basándonos en el movimiento constante del ventilador es el mismo.

Ahora para entender el fenómeno físico de las perturbaciones se analizará la fórmula de transferencia de calor la cual se presenta a continuación.

$$Q = \text{Calor} = \text{masa de aire} * (T_{\text{final}} - T_{\text{inicial}}) * \text{calor específico del aire} \quad \text{Ec(6)}$$

La interpretación física de la Ec(6) sería de la siguiente manera, como el calor específico del aire es constante, la variación de temperatura ($T_{\text{final}} - T_{\text{inicial}}$) se verá afectada por la masa de aire que podemos variarla mediante las perturbaciones o el calor de la niquelina que es variado por el voltaje.

Entonces se tiene una cantidad de aire en contacto con una fuente de energía durante un tiempo determinado en el que gana una cantidad de calor. Por lo que el tiempo de retardo de la planta sería casi constante en cada perturbación pero tomando en cuenta que la mezcla aire caliente aire frío cambia los valores netos de la planta, porque al poner la placa en la primera perturbación existirá un intercambio mayor de masa y energía entre el aire caliente dentro del ventilador y el aire frío del ambiente, ocurriendo lo contrario cuando se encuentra la placa en la cuarta perturbación.

4.2.2 Análisis De La Planta Mediante Las Ecuaciones De Transferencia De Calor

La resistencia térmica R para transferencia de calor entre dos sustancias se define como [8]:

$$R = \frac{\text{cambio en la diferencia de temperatura, } ^\circ\text{C}}{\text{cambio en el flujo de calor, kcal/seg}}$$

La capacitancia térmica C se define mediante:

$$C = \frac{\text{cambio en el calor almacenado, kcal}}{\text{cambio en la temperatura, } ^\circ\text{C}} = \text{masa del aire} * \text{calor específico del aire}$$

Suponemos que el cilindro contenedor está aislado para eliminar las pérdidas de calor hacia el aire circundante, también se despreció la temperatura del aislamiento y que el aire caliente está perfectamente mezclado con el aire frío por lo que a la salida se obtiene una temperatura estable, de esta forma solo se utiliza una temperatura para referirse al intercambiador de temperatura. Tomando en cuenta las consideraciones anteriormente planteadas se obtiene el diagrama de bloques del sistema el cuál se lo puede apreciar en la Figura 4.1 que se presenta a continuación.

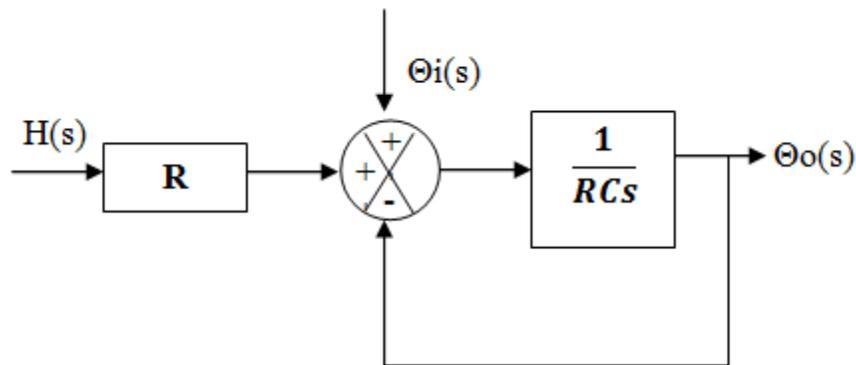


Figura 4.1 Diagrama de Bloques del Sistema Air Flow Control Temperature System

Donde:

Θ_i = Perturbaciones en la temperatura generadas por la posición de la placa en el orificio, $^\circ\text{C}$.

Θ_o = Temperatura en estado estable del aire que sale, $^\circ\text{C}$.

R = resistencia térmica, $^\circ\text{C seg} / \text{Kcal}$.

C = capacitancia térmica, $\text{Kcal} / ^\circ\text{C}$

H = entrada del flujo de calor en estado estable, Kcal/seg .

El diagrama de bloques presentado en la Figura 4.1 se obtuvo considerando que la entrada de flujo de calor se verá afectada por una resistencia térmica propia del aire, de la

misma manera se puede afectar la temperatura por las perturbaciones como se explicó anteriormente, esta a su salida se encontrará con una mezcla de aire que ya posee una capacitancia térmica la cuál gracias al ventilador será realimentada.

Del diagrama de bloques anteriormente expuesto se puede simplificar obteniendo como resultado el siguiente diagrama de bloques que se presenta en la Figura 4.2.

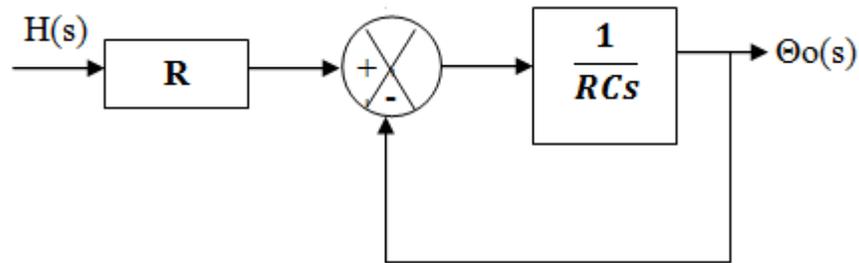
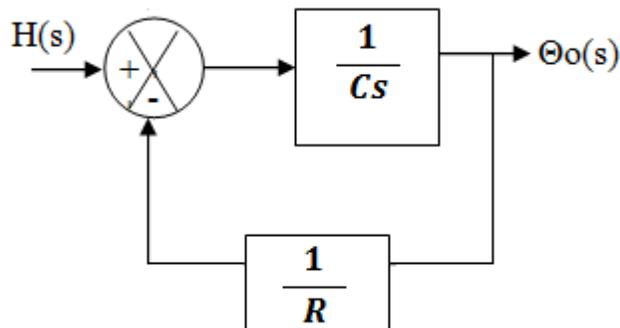


Figura 4.2 Diagrama de bloques del sistema Air Flow Control Temperature System considerando que no existe variaciones en la perturbación

Como se puede apreciar al variar la perturbación varía la planta pero al estabilizarse, esta perturbación desaparece puesto que no se varía de forma continua la placa, y como se mencionó anteriormente el efecto de las perturbaciones se ve reflejado únicamente en los valores de resistencia y capacitancia térmica.

Mediante algebra de bloques operamos el diagrama presentado en la Figura 4.2 obteniendo como resultado:



Por último continuando con la operación se obtiene:

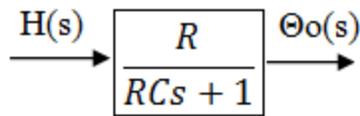


Figura 4.3 Diagrama de bloques del sistema Air Flow Control Temperature System simplificado

Como se puede observar en la Figura 4.3 se obtiene el modelo matemático de primer orden característico de un sistema intercambiador de temperatura, el mismo que se verá afectado por la resistencia térmica que es controlada mediante las perturbaciones en la entrada de flujo del aire ambiente y de la capacitancia térmica que será afectada por la potencia de la niquelina como se explicó en la interpretación física de la planta.

4.2.3 Análisis De La Planta Mediante Las Ecuaciones De Transferencia De Calor Considerando Las Variaciones De Caudal Generadas Por Las Perturbaciones

El calor puede fluir cuando un flujo de masa caliente fluye hacia una masa fría o viceversa: $q = W C_v (T_1 - T_2)$ en donde:

- q = calor [Joules/seg] o [Calorias/seg].
- W = tasa del flujo de masa de aire con una temperatura T_1 que fluye a reservas con temperatura T_2 [gr/seg].
- C_v = calor específico a volumen constante [Joules/seg] o [Cal / gr °C].
 C_v del aire = 0,094 [Cal / gr °C].

Ahora se pretende encontrar las ecuaciones diferenciales que describen la dinámica de la temperatura del aire que sale como función del área A_s de la apertura de las perturbaciones de aire frío, para muchas aplicaciones no es necesario tener una gran exactitud por que la realimentación corrige en gran cantidad los errores del modelo, por ello

tiene sentido suponer que el calor transferido por la niquelina al aire es proporcional a la diferencia de sus temperaturas.

Para lograr lo anteriormente descrito se procede al siguiente desarrollo matemático basándonos en el gráfico de la Figura 4.4 que se presenta a continuación:

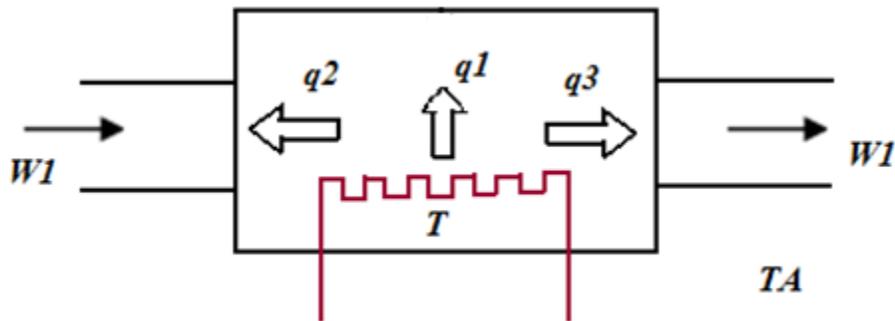


Figura 4.4 Diagrama De Transferencia De Calor Del Sistema

Donde:

- q_1 = calor de la niquelina.
- q_2 = calor que se va hacia el ventilador.
- q_3 = calor que sale del tubo hacia el ambiente.
- W = tasa del flujo de masa de aire con una temperatura T_1 que fluye a reservas con temperatura T_2 [gr/seg].

Para el análisis de transferencia de calor se analiza el paso del aire a través de la niquelina dentro del tubo en donde el caudal que entra es igual al caudal que sale y nos referiremos a una sola temperatura dentro del tubo y a una temperatura ambiente fuera del tubo, no se considerarán pérdidas de calor por lo que se tendría la ecuación:

$$q_1 - q_2 - q_3 = C (dT/dt), \text{ donde:}$$

$$q_2 = W_1 C_v (T - T_A)$$

$$q_3 = W_1 C_v (T - T_A)$$

$$\Delta T = T - T_A$$

$$\delta T = \delta \Delta T = \Delta \dot{T}$$

$$q_1 - 2W_1 C_v \Delta T = C \Delta \dot{T}$$

$$Q_1(s) - 2W_1 C_v \Delta T(s) = s C \Delta T(s)$$

$$\Delta T(s) (s C + 2W_1 C_v) = Q_1(s)$$

$$\frac{\Delta T(s)}{Q_1(s)} = \frac{1}{sC + 2W_1 C_v}$$

Para obtener la ecuación característica de un sistema de primer orden acomodamos la ecuación anterior obteniendo:

$$\frac{\Delta T(s)}{Q_1(s)} = \frac{K}{s\tau + 1} e^{-st}$$

$$K = \frac{1}{2W_1 C_v}$$

$$\tau = \frac{C}{2W_1 C_v}$$

El tiempo de retardo se obtuvo de las siguientes ecuaciones:

- $W = A * V$, donde W = Caudal en m^3/s .
- A = Área del tubo.
- V = Velocidad.

Como el sensor se encuentra ubicado a una cierta distancia de la niquelina se puede decir que:

Distancia = velocidad * tiempo

Velocidad = Caudal / Área.

Por lo tanto, Tiempo = (Distancia * Área) / Caudal

4.2.4 Identificación De La Planta

Para el modelamiento de la planta se recurrió a un procedimiento experimental, siguiendo los pasos que se detallan a continuación:

- Adquisición de datos.
- Identificación del modelo matemático de la planta.

4.2.4.1 Adquisición de Datos

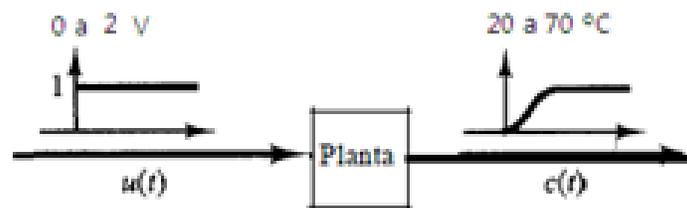


Figura 4.5 Respuesta De La Planta Al Impulso

Con la planta a lazo abierto, se la llevó a un punto de operación estable, al dejarla sin voltaje de entrada en el actuador y observando su estabilización en una temperatura inicial de aproximadamente $T_0 = 24.5^\circ \text{C}$ sin ninguna perturbación como se puede apreciar en la Figura 4.6 que se presenta a continuación.

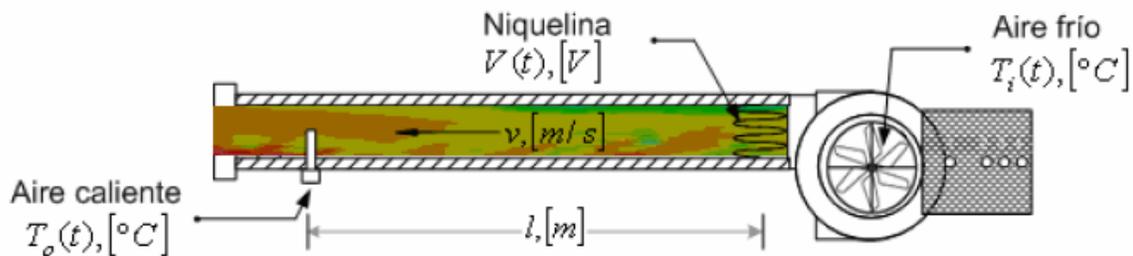


Figura 4.6 Unidad Air Flow Temperature Control System Sin Perturbación

A partir de esto, se aplicó en el tiempo t_0 , un cambio constante de escalón en la entrada $u(t) = 10 \text{ V}$. Los datos de la respuesta de temperatura de la planta se tomaron

mediante la tarjeta de adquisición de datos PCI 6221, para tener los datos en unidades de ingeniería y no en unidades crudas se obtuvo la ecuación del sensor IC de la siguiente manera: el rango de voltaje del sensor de temperatura es de 0 a 5 voltios teniendo en cuenta que el rango de temperatura es de 20 a 70 °C correspondientemente, en tal virtud con estos datos se procedió a obtener la gráfica de respuesta del sensor IC mediante la utilización de una hoja electrónica basado en los datos de la Tabla 4.2 que se presenta a continuación.

<i>Voltios</i>	<i>Temperatura en °C</i>
0	20
1	30
2	40
3	50
4	60
5	70

Tabla 4.2 Datos De Respuesta Del Sensor Tipo IC

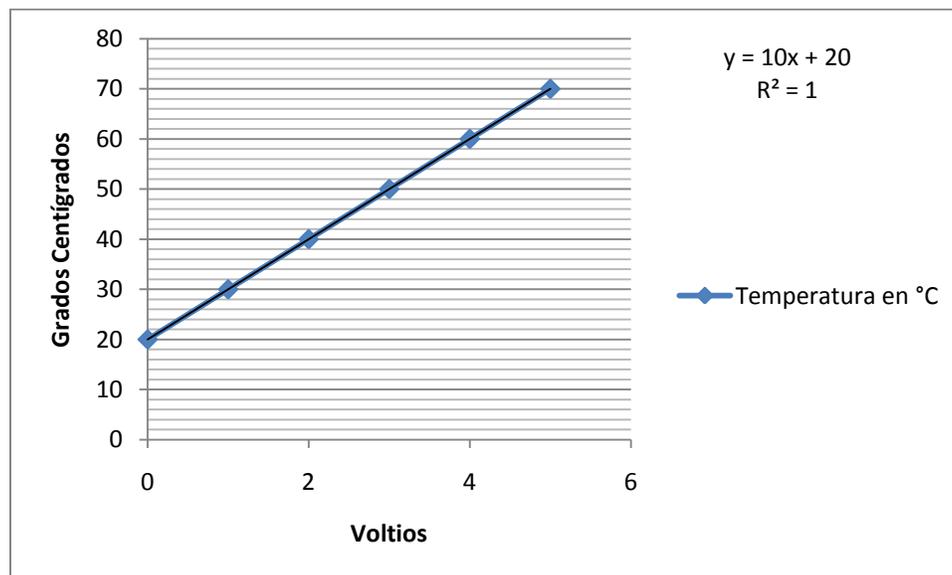


Figura 4.7 Gráfica De La Respuesta Del Sensor Tipo IC

Como se puede observar en la Figura 4.7 existe una relación lineal entre el voltaje de entrada y la temperatura obteniéndose la ecuación $T [^{\circ}\text{C}] = 10 * V + 20$, una vez obtenida la ecuación de respuesta del sensor se procedió a realizar un programa en Simulink® con el cual se obtendrá los datos necesarios para el modelamiento de la planta.

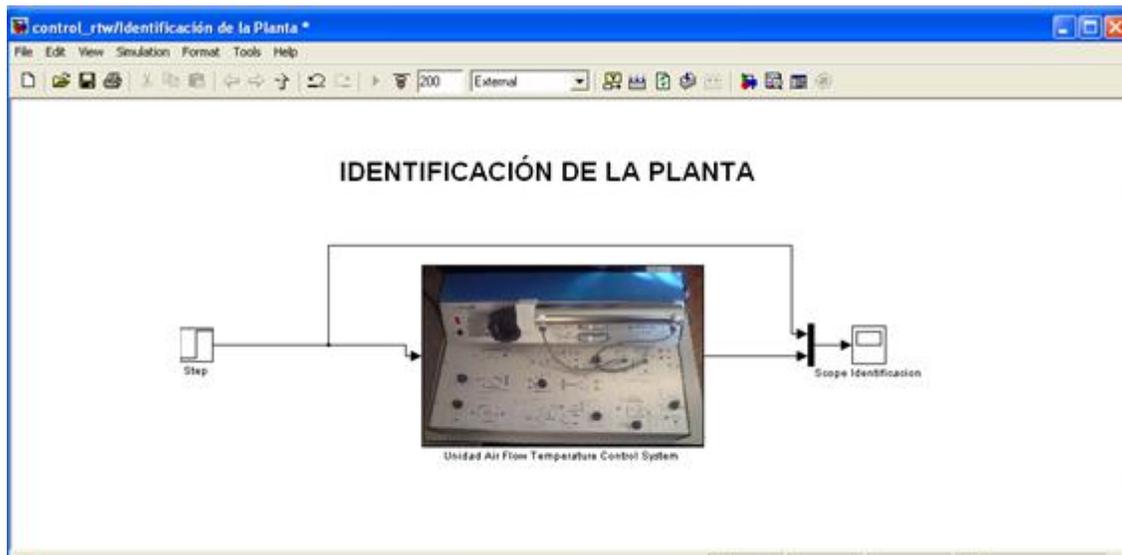


Figura 4.8 Identificación De La Planta

Terminadas las conexiones necesarias para realizar el esquema expuesto en la Figura 4.8 se debe configurar todas las opciones necesarias para realizar una aplicación en tiempo real como se explico anteriormente en el Capítulo 3.

Se debe configurar el *data archiving* para generar un archivo de respaldo con nombre “modelado” que se grabará en la raíz C:\, tal como se muestra en la Figura 4.9 que se presenta a continuación

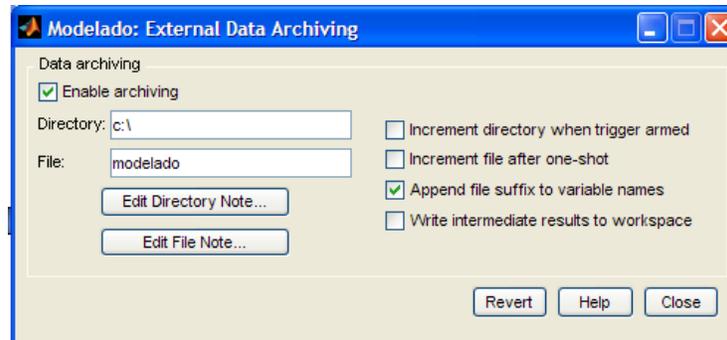


Figura 4.9 Archivo De Datos Adquiridos

Posteriormente se debe configurar el bloque step para un valor de 10 voltios como se puede apreciar en la Figura 4.10

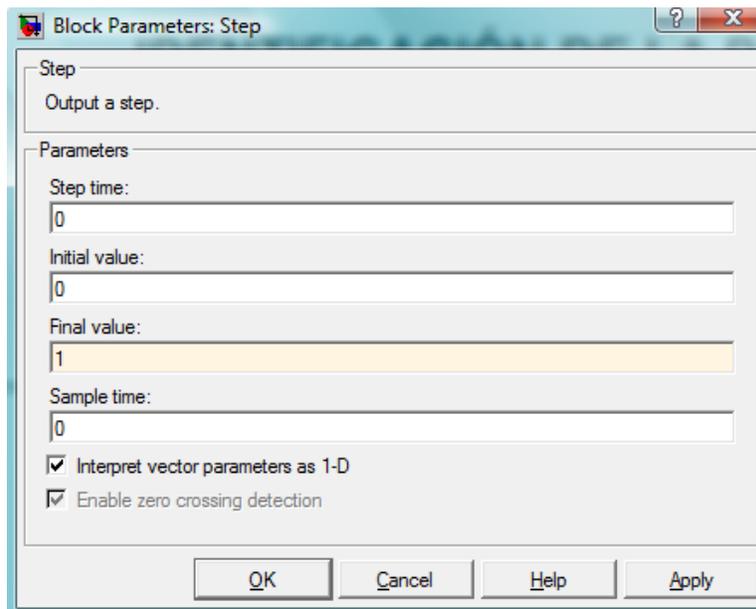


Figura 4.10 Configuración Del Bloque Step

El tiempo de simulación utilizado fue 200 segundos, y la duración de muestras recogidas por el osciloscopio es de 20000 porque la frecuencia de muestreo es 0.01, entonces dando doble click en los bloques de entrada y salida analógicas se debe configurar el mismo tiempo de muestreo como se muestra en la Figura 4.11 que se presenta a continuación.

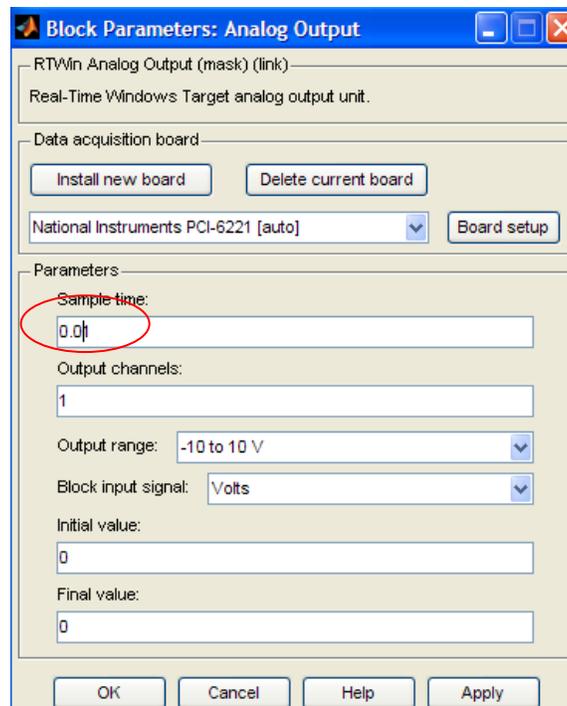


Figura 4.11 Configuración Del Tiempo de Muestreo Del Analog Input

Este valor representa el tiempo en segundos con la cual se va a adquirir y a sacar datos de la tarjeta de adquisición de datos, el número de muestras que se debe seleccionar para los osciloscopios en tiempo real se relaciona con el tiempo de muestreo de la siguiente forma:

$$\text{Numero de Muestras} = \frac{\text{Tiempo de Duración de la Simulación}}{\text{Tiempo de Muestreo}}$$

El tiempo de duración de la simulación se puede cambiar fácilmente desde la barra de herramientas como se muestra en la Figura 4.11, este valor se encuentra expresado en segundos.

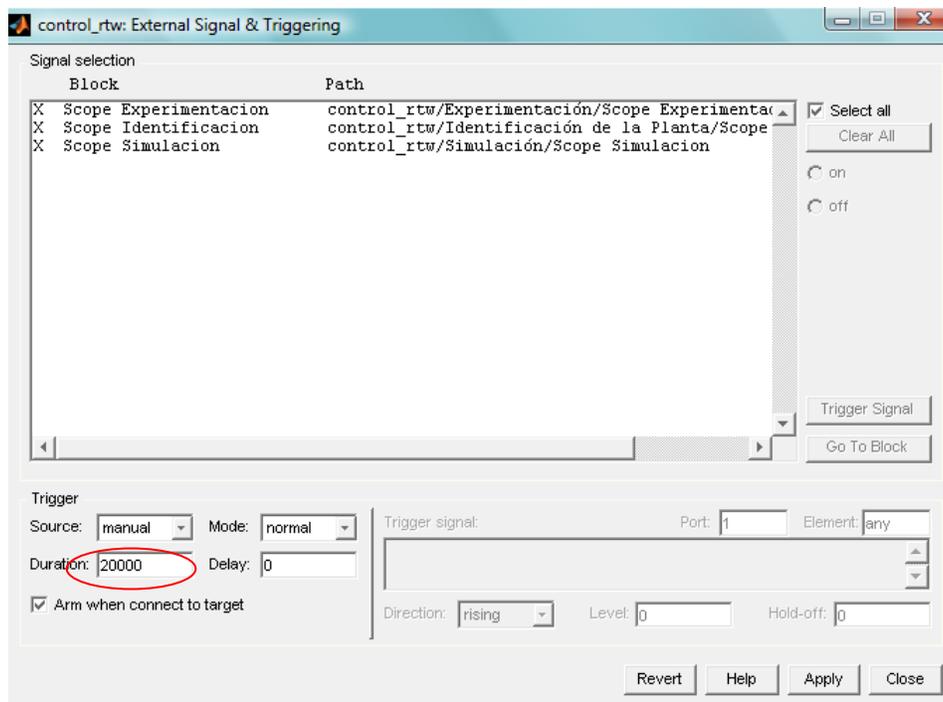


Figura 4.13 Configuración Del Número de Muestras

Si se selecciona un número menor que el número total de muestras se perderá información y se generarán varios archivos guardados, caso contrario si se selecciona un valor mayor se tendrá valores vacíos por lo que es aconsejable seleccionar un valor adecuado de número de muestras para que se pueda observar todas las señales en un solo archivo, al ejecutarse el programa y finalizar la aplicación en tiempo real, debe crearse en la siguiente dirección un archivo “C:\modelado_0.mat”, con el cuál procederemos a trabajar para poder identificar la planta.

En la ventana de comandos de Matlab[®] procederemos a realizar el siguiente programa:

%Modelamiento de la Planta Air Flow Temperature Control System

%Byron Acuña

%Alexander Ibarra

Close all;%Cierra todas las ventanas de figuras.

Clear all;%Borra todas las variables, funciones y links.

Clc;%Limpia la ventana de comandos de Matlab

load ('c:\modelado_0')%Cargamos la respuesta de la Planta a un Impulso de 2 Voltios

Cuando se ejecuta este comando se carga automáticamente en el *workspace* la variable *ScopeData_0* el cual corresponde al osciloscopio que tenemos en el esquema, como se muestra en la Figura 4.14 que se presenta a continuación

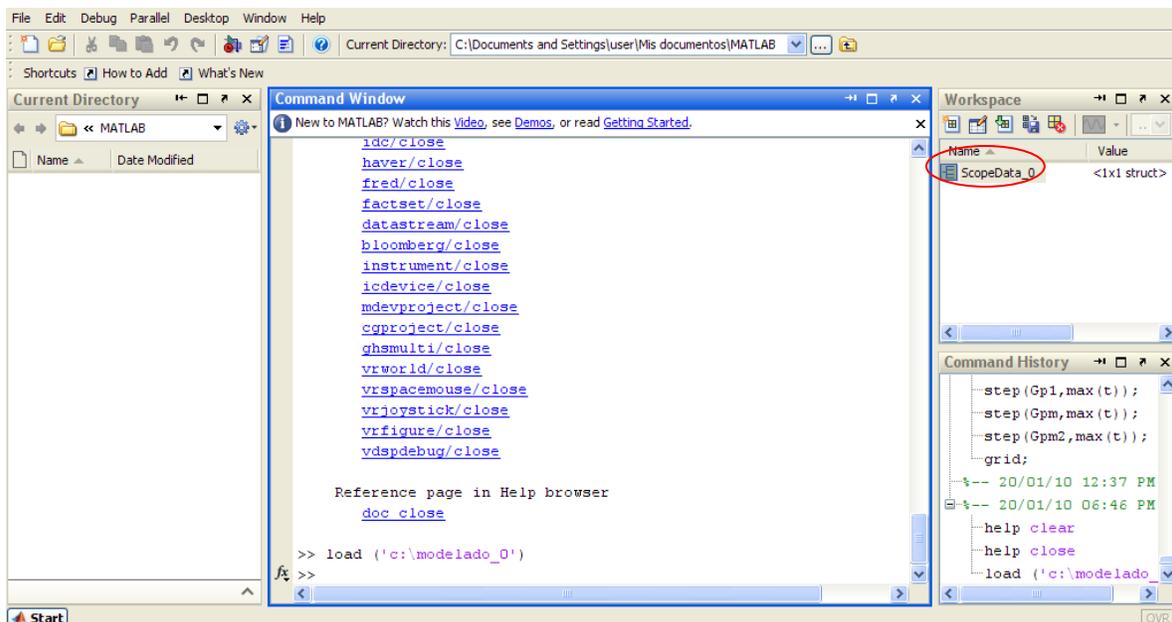


Figura 4.14 Variable ScopeData_0

Esta estructura posee el tiempo y los valores de las señales, los cuales se pueden observar fácilmente si damos doble click sobre el mismo, para graficar la respuesta al impulso escribimos el siguiente comando:

plot(ScopeData_0.time,ScopeData_0.signals.values)

%Graficamos la respuesta al impulso de la planta

Obteniendo la Figura 4.14 que se presenta a continuación.

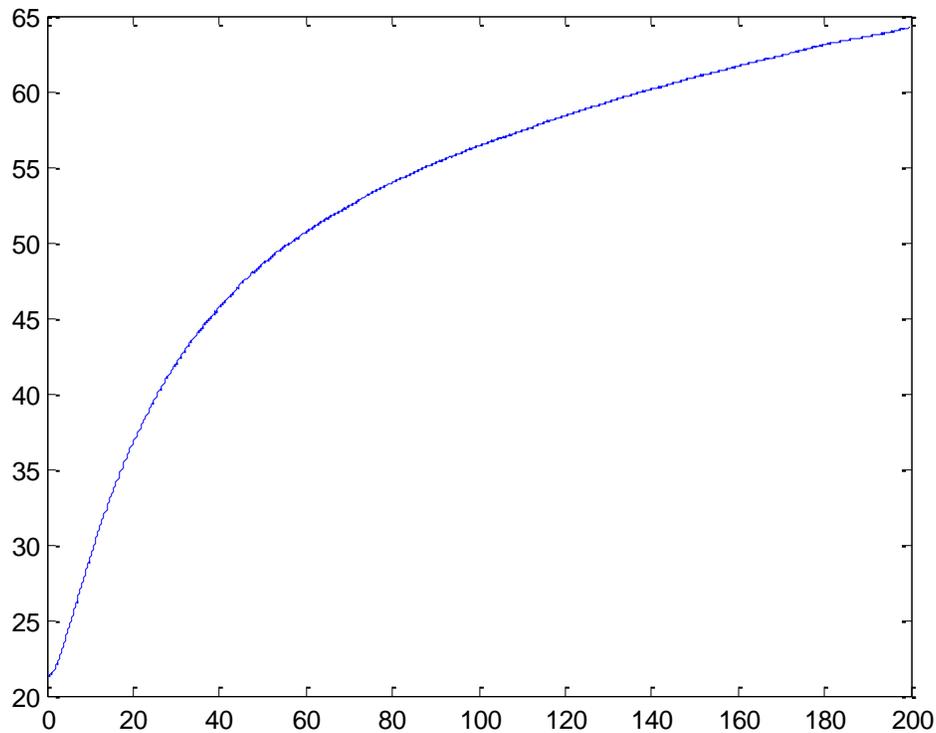


Figura 4.15 Respuesta De La Planta A Un Step De 10 Voltios

```
t=ScopeData_0.time; %Guardamos el vector tiempo en t
u=ScopeData_0.signals.values(:,2);%Definimos el Escalón Unitario.
```

Como se explico anteriormente el tiempo de muestreo fue 0.01 por lo que se lo definirá igualmente para el análisis con Matlab[®].

```
Ts=0,01;
y=ScopeData_0.signals.values(:,1);%Definimos la respuesta de la Planta.
```

A continuación utilizaremos el siguiente comando para analizar esta planta mediante la toolbox de Matlab[®] System Identification.

```
data=iddata(y,u,Ts);
%Planta estimada por MATLAB (System Identification Toolbox)
```

Una vez que se teclea el comando $data=iddata(y,u,Ts)$; en el workspace debe aparecer una variable que podamos utilizar para la herramienta *System Identification Toolbox* (*ident*) como se puede apreciar en la Figura 4.16 que se presenta a continuación.

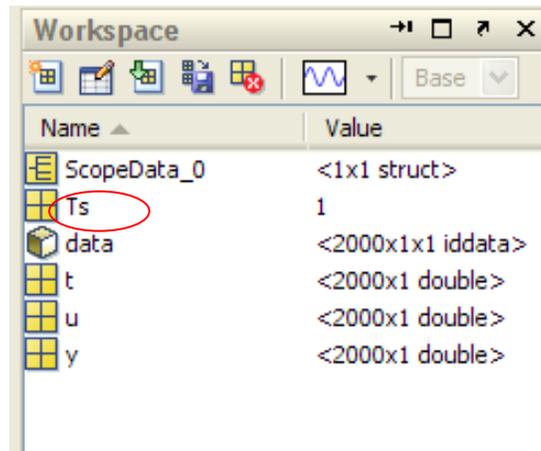


Figura 4.16 Variable Para La Herramienta Ident

4.2.4.2 Identificación Del Modelo Matemático De La Planta

Con los datos de la respuesta al paso $u(t)= 10V$, se pueden realizar diversos métodos de sintonización. Algunos implican la disposición de un modelo matemático de la planta, el cual se realizó en este proyecto mediante la herramienta de Matlab[®] llamada *System Identification Toolbox* (*ident*), manejada como se indica a continuación:

Tecleando en la ventana de comandos de Matlab[®] *ident* se tiene acceso a la herramienta, posteriormente se abre la siguiente ventana:

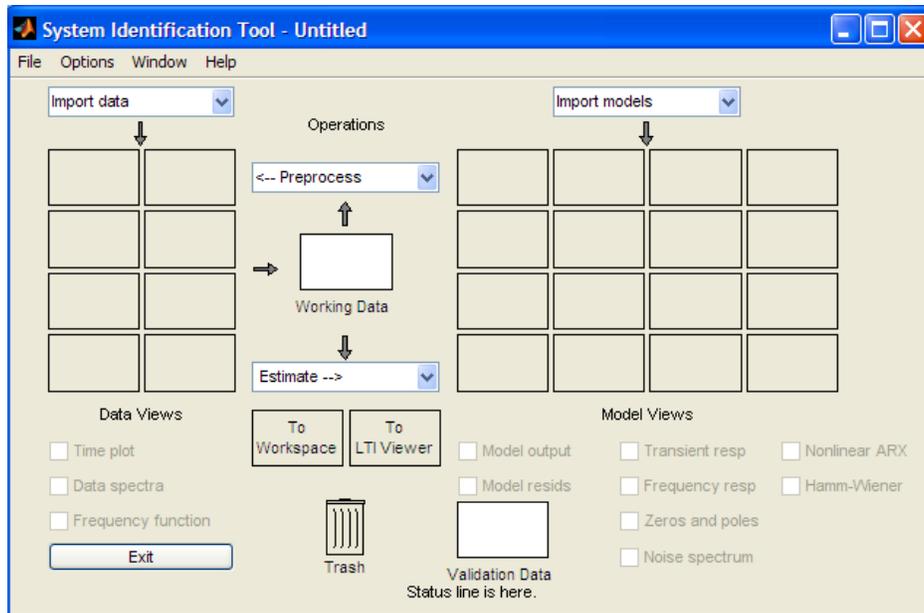


Figura 4.17 System Identification Tool (Ident)

Para la importación de los datos desde el Workspace a ident se debe seguir los siguientes pasos:

- En la Pestaña de Import data seleccionamos Time domain data...
- Se abre una ventana Import Data con una pestaña en la cual seleccionaremos Data Object.
- Posteriormente cambiara el formato de esta ventana y en la opción “Object:” escribiremos el nombre de la variable que en nuestro caso es “data”.
- Con el cual debe llenarse automáticamente las casillas inferiores.
- Damos click en el botón Import
- Finalmente damos click en close

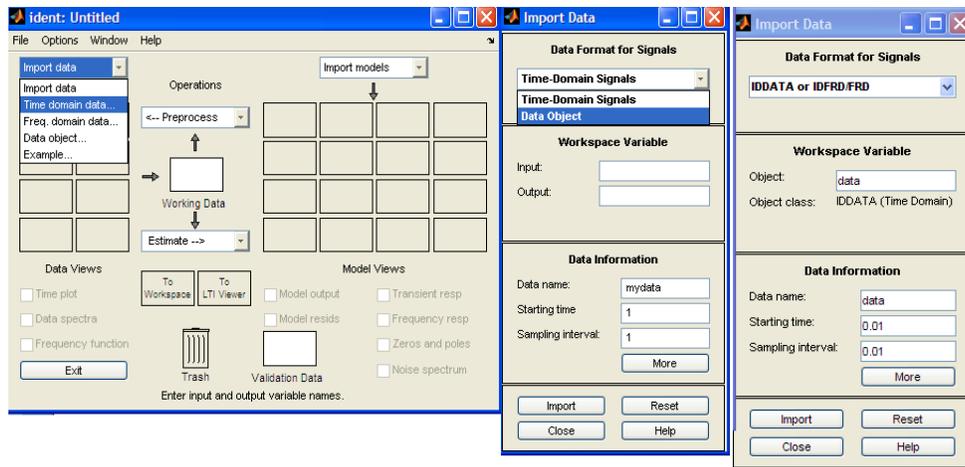


Figura 4.18 Carga Del Dato En La Herramienta Ident

A continuación esta herramienta nos permite obtener la descripción de la planta en Data Views y la estimación matemática del modelo seleccionando el método mediante la opción Estimate - - >. En este caso, se ha utilizado la opción Estimate → Process models... y Estimate → Quick starts, para una elección de la mejor estimación. Con esto se obtuvieron las siguientes estimaciones:

Al seleccionar Process Model nos aparece una ventana como la que se muestra en la Figura 4.19 que se presenta a continuación, en la cual seleccionamos la opción Estimate.

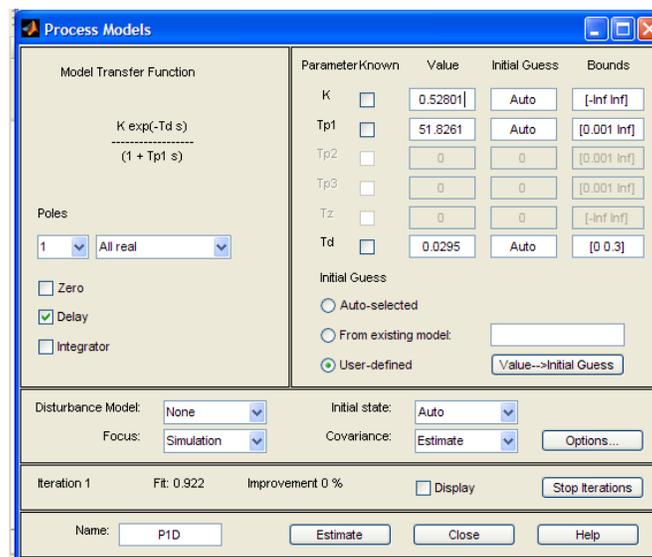


Figura 4.19 Process Model

Con la selección de Quick start nos aparecen otras estimaciones.

La Figura 4.20 nos muestra diferentes estimaciones, y nos dice cual es la mejor opción.

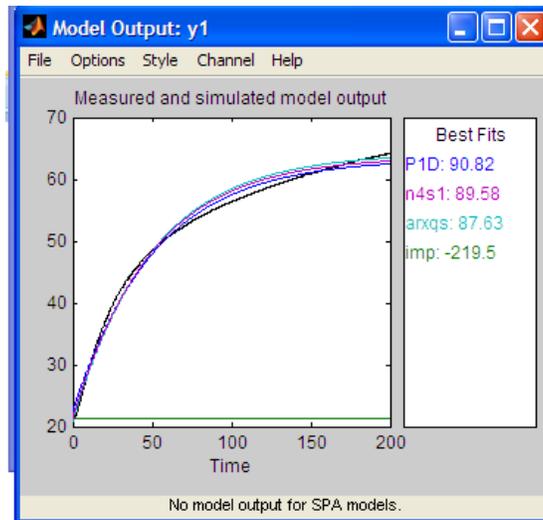


Figura 4.20 Model Output

Finalmente para continuar con el modelamiento exportamos estas estimaciones arrastrándolas hasta el recuadro To Workspace, como se muestra en la Figura 4.21 que se presenta a continuación.

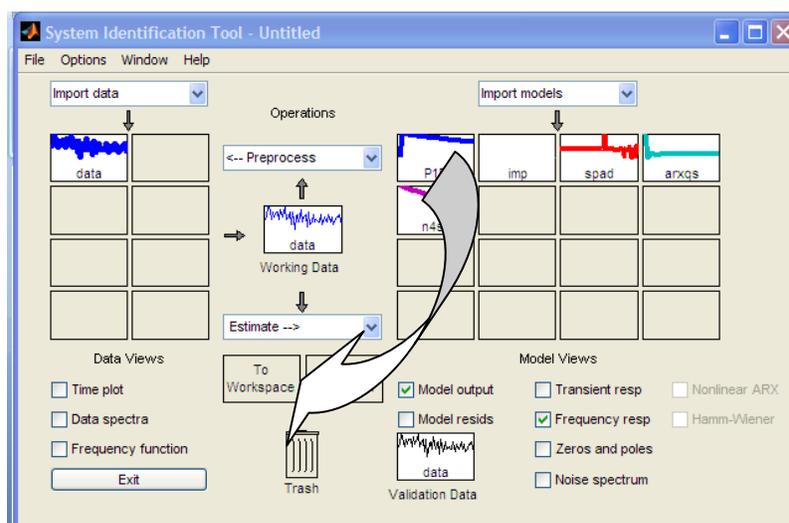


Figura 4.21 Carga De La Aproximación En El Workspace

La estimación que se cargó fue la PID porque apareció como la mejor estimación calculada por la herramienta ident.

En la ventana de comandos de Matlab® se continua con la obtención de las funciones de transferencias.

```
%Estimación de primer orden simple
```

```
sys=tf(PID);
```

```
Gp1=sys(1);
```

```
Gp1
```

A continuación nos debe aparecer la función de transferencia estimada de la planta la cual es:

```
>> Gp1 = tf(PID)
```

```
Transfer function from input "u1" to output "y1":
```

```
          0.010188  
exp(-0.295*s) * -----  
                s + 0.01893
```

4.2.5 Identificación De Los Modelos Matemáticos Para Cada Perturbación

Siguiendo el mismo procedimiento anteriormente detallado se obtuvieron las funciones de transferencia de la planta sometida a diferentes perturbaciones como se puede apreciar en la Tabla 4.3 que se presenta a continuación.

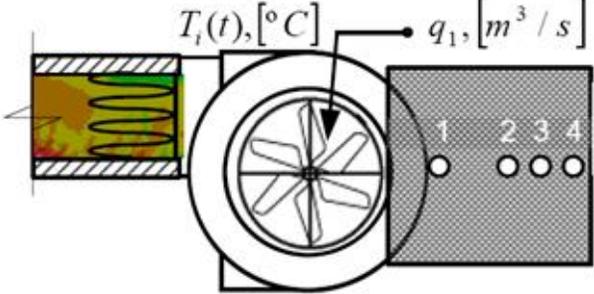
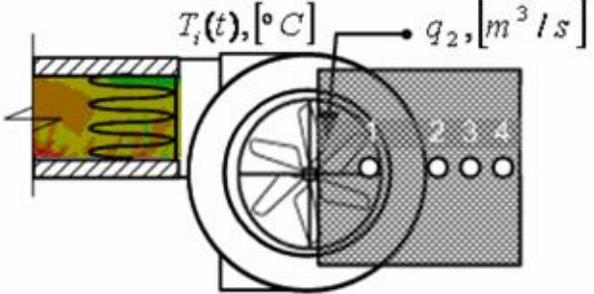
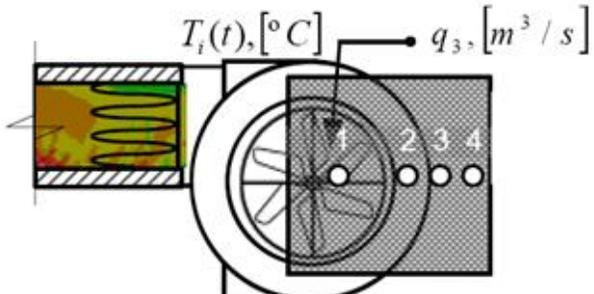
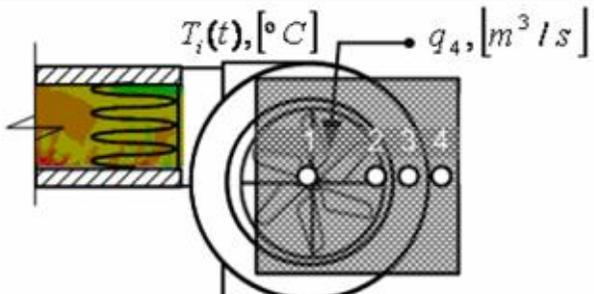
Ingreso de aire	Representación	Modelo
1		$G_1(s) = \frac{0,52801}{51,8261s + 1} e^{-0,295s}$
2		$G_2(s) = \frac{0,50239}{39,5624s + 1} e^{-0,3s}$
3		$G_3(s) = \frac{0,56354}{47,143s + 1} e^{-0,3s}$
4		$G_4(s) = \frac{0,71508}{64,4921s + 1} e^{-0,3s}$

Tabla 4.3 Respuesta de la Unidad Air Flow Temperature Control System Ante Perturbaciones

4.2.6 Diseño Del Controlador PID

4.2.6.1 Sintonización Por Medio Del Segundo Método De Zeigler – Nichols

Utilizando el potencial de Matlab[®] y con la flexibilidad del trabajo en tiempo real se procedió a realizar el método de oscilaciones sostenidas con la planta en tiempo real, para lo cual se procedió a buscar un K crítico (K_{cr}) obteniendo la Figura 4.22 que se presenta a continuación.

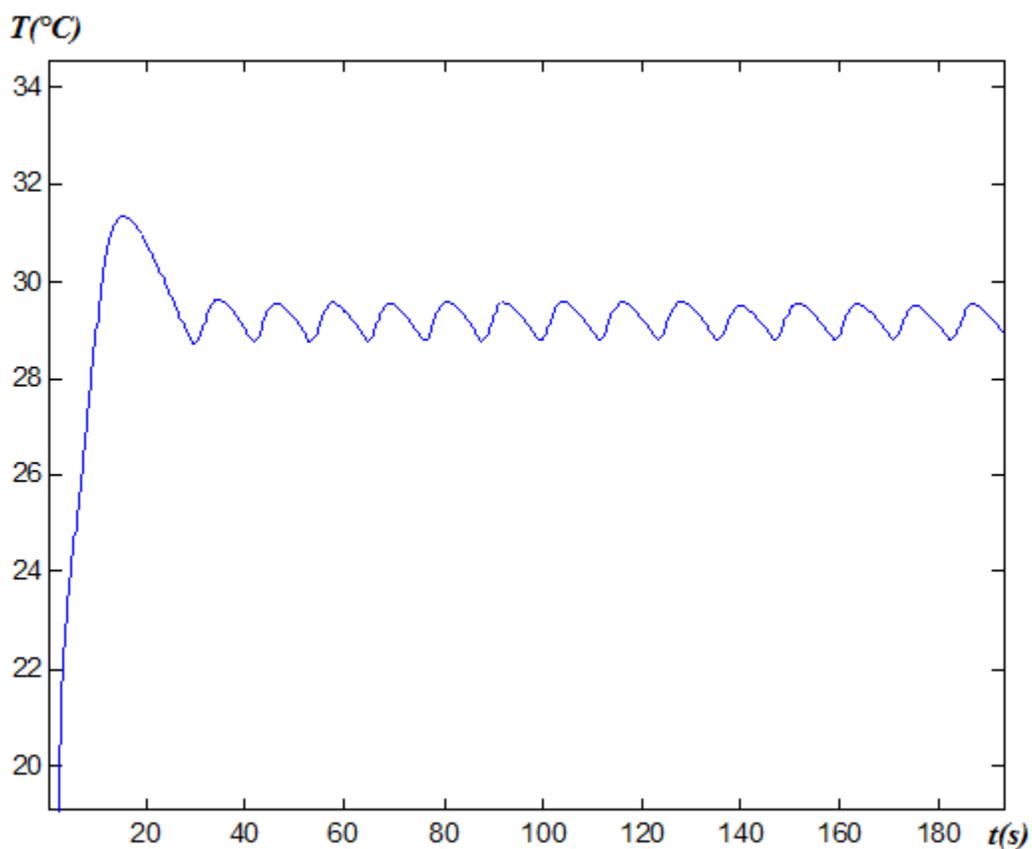


Figura 4.22 Oscilaciones Sostenidas De La Planta Kcr En Tiempo Real

De la Figura 4.22 el K_{cr} encontrado es 34, analizando los datos entre crestas se encontró el P crítico (P_{cr}) que es 12 seg, el segundo método de Zeigler Nichols establece que:

Tipo de controlador	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Tabla 4.4 Regla Se Sintonización De Ziegler Nichols Segundo Método

A partir de la Tabla 4.4 se obtuvieron los parámetros:

- $K_p = 20,4$
- $T_i = 6$
- $T_d = 1,5$
- $K_i = K_p/T_i = 3,4$
- $K_d = K_p * T_d = 30,6$

4.3 DESARROLLO PROGRAMAS EN SIMULINK

4.3.1 Identificación De La Planta

Mediante la utilización de subsistemas en la herramienta Simulink[®] se procedió a la implementación de un subsistema para obtener la respuesta al escalón de la planta, además se implementó también la ecuación de respuesta del sensor para lograr convertir las unidades crudas en unidades de ingeniería.

4.3.2 Control PID Experimental

Mediante la utilización de subsistemas en la herramienta Simulink[®] se procedió a la implementación del controlador PID para el Control PID Experimental en donde se utilizaron los parámetros $K_p = 20,581$, $K_i = 3,43$ y $K_d = 30,871$, además se implementó también la ecuación de respuesta del sensor para lograr convertir las unidades crudas en unidades de ingeniería.

4.3.3 Simulación Del Control PID

Para la realización de los modelos de simulación se crearon cuatro subsistemas en donde se implementó las funciones de transferencia obtenidas con las perturbaciones que se detallaron en la Tabla 4.3 y un controlador con los valores obtenidos por el segundo método de Ziegler – Nichols los cuales son: $K_p= 20,581$, $K_i= 3,43$ y $K_d= 30,871$.

4.3.4 Descripción De La HMI

Al abrir el archivo control_rtw.mdl se despliega la siguiente ventana la cual podemos apreciar en la Figura 4.23.

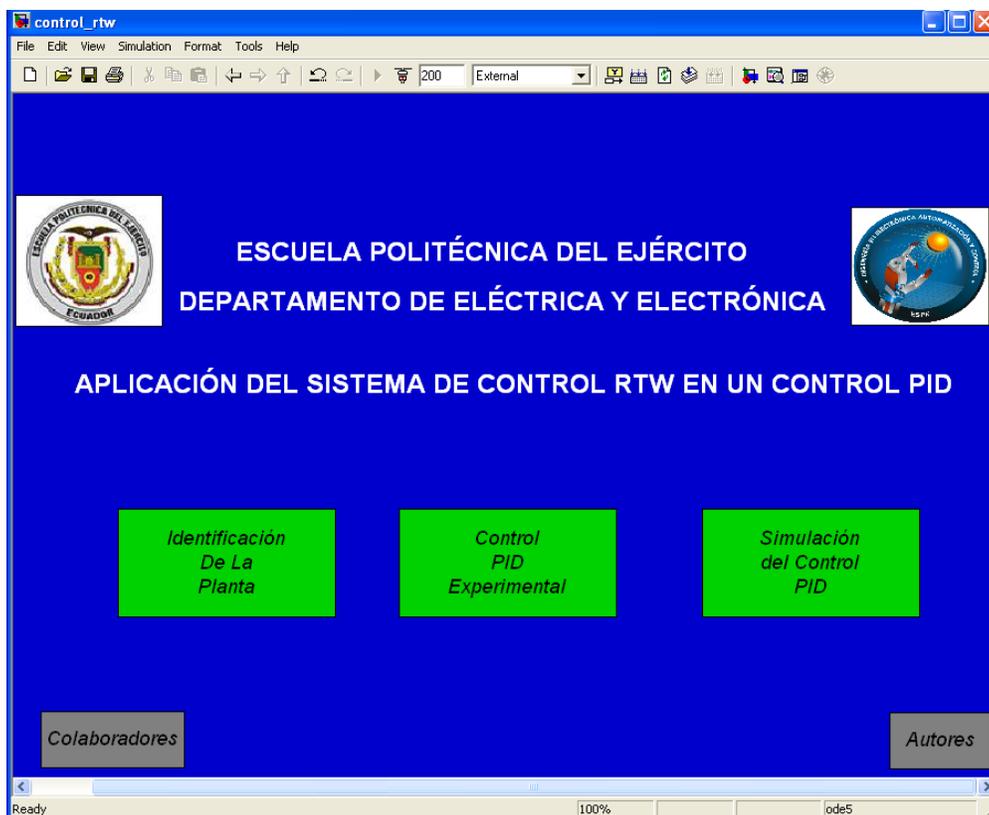


Figura 4.23 Pantalla De Ingreso Al Sistema

En esta ventana consta información general del sistema y los accesos a los subsistemas: Identificación De La Planta, Control PID Experimental y Simulación del control PID.

Al dar doble click sobre el botón de Identificación De La Planta se abre el mismo, el cuál contiene subsistemas destinados a obtener la respuesta de la planta ante un escalón.

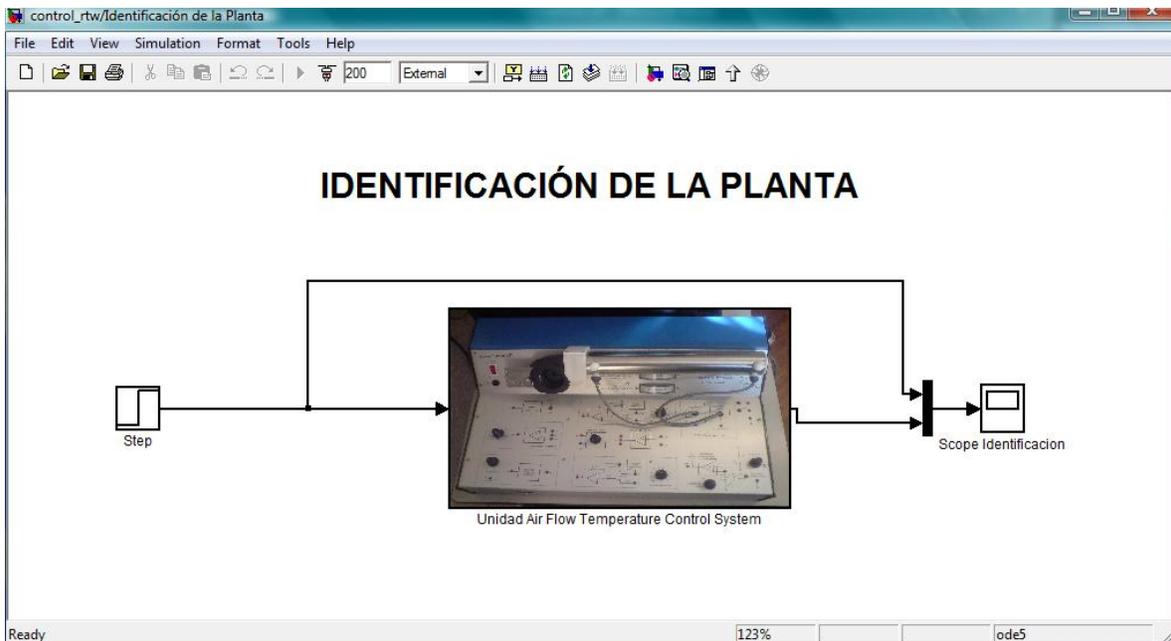


Figura 4.24 Identificación De La Planta

Al dar doble click sobre la imagen de la planta Air Flow Temperature Control System se abrirá el subsistema que contiene una entrada que enlaza la salida del bloque Step con el bloque de ganancia y posteriormente a la salida analógica asociada a la tarjeta de adquisición de datos PCI6221 la cual enviará la señal del paso que esta comprendida entre 0 a 10 V.

La salida de este subsistema es la señal del sensor en unidades de ingeniería tal como se puede apreciar en la Figura 4.25 que se presenta a continuación.

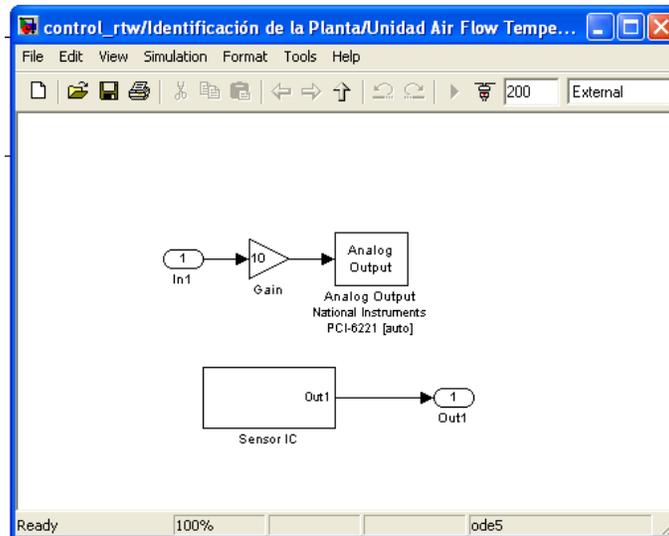


Figura 4.25 Subsistema De La Planta Air Flow Temperature Control System

De la misma manera se realizó otro subsistema con la etiqueta sensor IC la misma que es la encargada de la realización de la adquisición y transformación de la señal del sensor de unidades crudas a unidades de ingeniería, tal como se puede apreciar en la Figura 4.26 que se presenta a continuación.

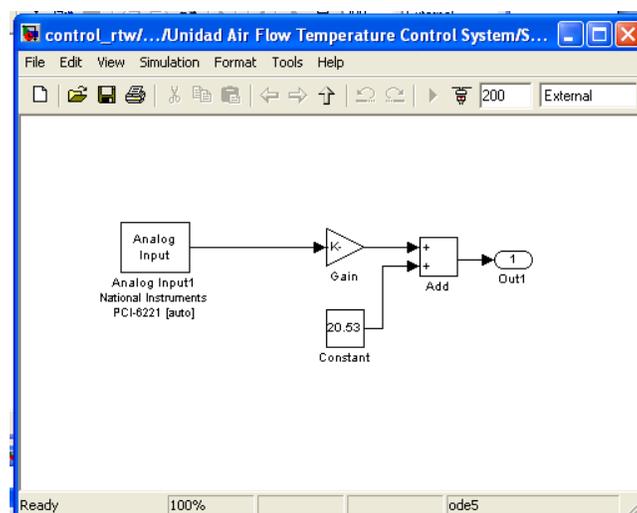


Figura 4.26 Subsistema Del Sensor Tipo IC

Al dar doble click sobre el botón de Control PID Experimental se abre el mismo, el cuál contiene subsistemas destinados a implementar de manera física la conexión entre la planta y el software creado para de esta manera dar ejecución al control en tiempo real.

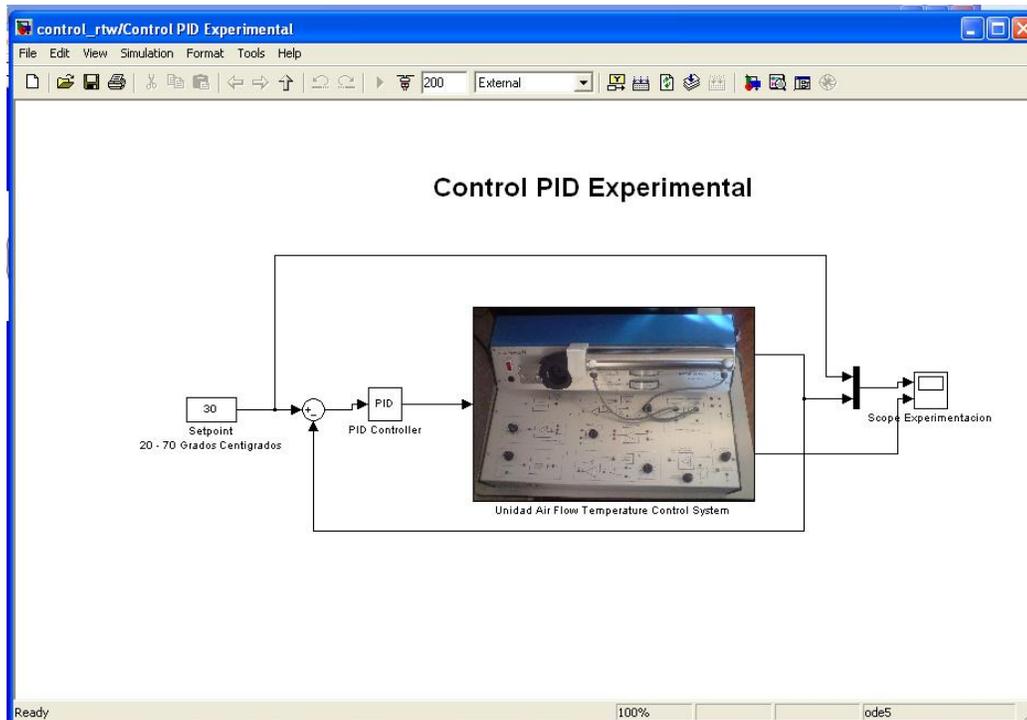


Figura 4.27 Control PID Experimental

Al dar doble click sobre la imagen de la planta Air Flow Temperature Control System se abrirá el subsistema que contiene una entrada que enlaza la salida del bloque de control PID con el bloque de saturación y posteriormente a la salida analógica asociada a la tarjeta de adquisición de datos PCI6221 la cual enviará la señal de control de 0 a 10 V.

Las dos salidas de este subsistema son la señal de control y la señal del sensor en unidades de ingeniería tal como se puede apreciar en la Figura 4.28 que se presenta a continuación.

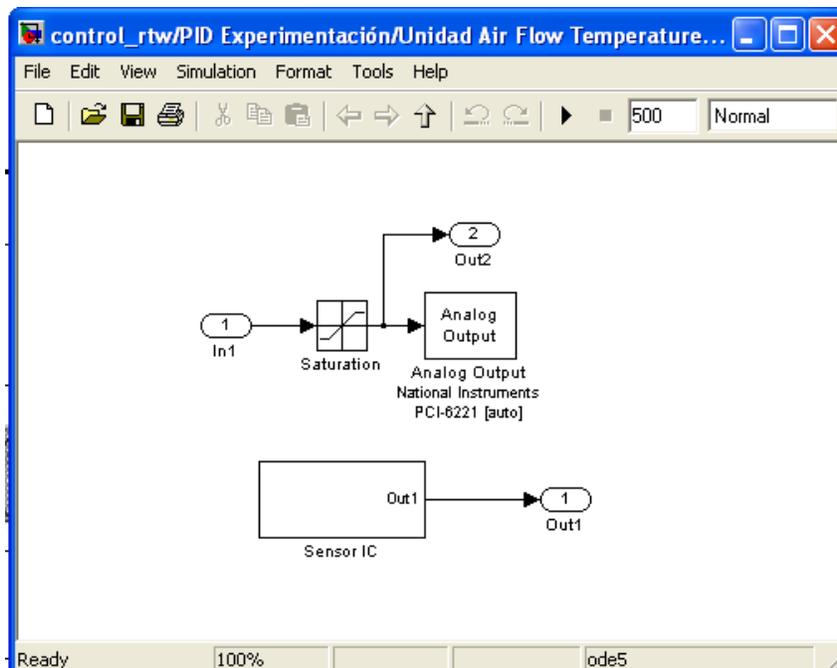


Figura 4.28 Subsistema De La Planta Air Flow Temperature Control System

De la misma manera se realizó otro subsistema con la etiqueta sensor IC la misma que es la encargada de la realización de la adquisición y transformación de la señal del sensor de unidades crudas a unidades de ingeniería, tal como se puede apreciar en la Figura 4.29 que se presenta a continuación.

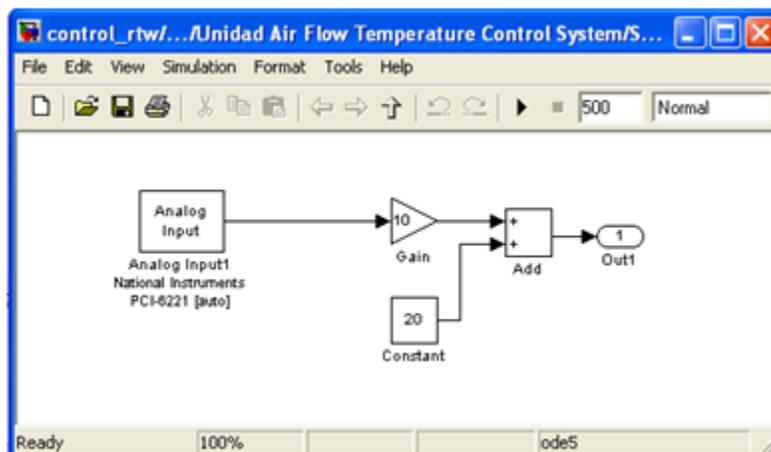


Figura 4.29 Subsistema Del Sensor Tipo IC

Al dar doble click sobre el botón de Simulación del Control PID de la pantalla principal se abre el mismo, el cuál contiene subsistemas destinados a implementar las funciones de transferencia obtenidas en la Tabla 4.3 y el controlador diseñado como se puede apreciar en la Figura 4.30 que se presenta a continuación.

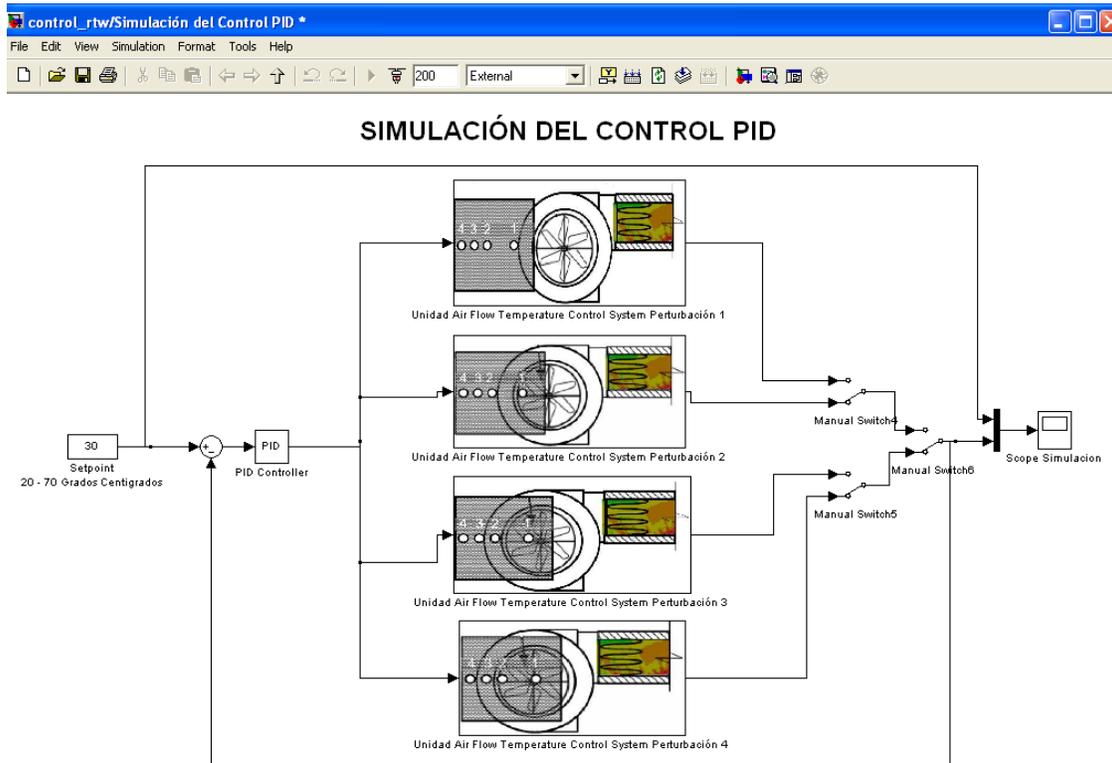


Figura 4.30 Simulación Del Control PID

Al dar doble click sobre cada una de las imágenes de la planta con su respectiva perturbación se abrirá el subsistema que contiene la implementación de los modelos matemáticos obtenidos en la Tabla 4.3 como se muestra en la Figura 4.31 que se presenta a continuación.

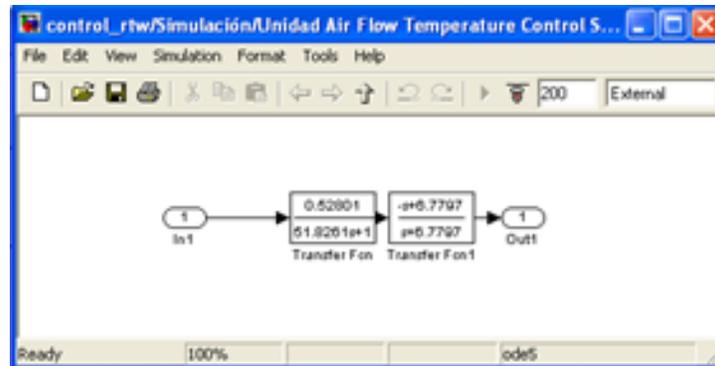


Figura 4.31 Función De Transferencia De La Planta Con Retardo

La transformación del retardo de una forma exponencial a una función de transferencia con coeficientes se lo realizó de la siguiente manera detallada en el ejemplo de cálculo que se presenta a continuación.

$$G_1(s) = \frac{0,52801}{51,8261s + 1} e^{-0,295s}$$

```
>> [num, dem] =pade(0.3, 1)
```

```
num =
```

```
   -1.0000    6.6667
```

```
dem =
```

```
    1.0000    6.6667
```

```
>> delay=tf(num, dem)
```

```
Transfer function:
```

```
-s + 6.667
```

```
-----
```

```
s + 6.667
```

CAPITULO V

PRUEBAS Y RESULTADOS

5.1 PRUEBAS

5.1.1 Pruebas Del Objeto De Experimentación Y Del Modelo De Simulación

Con el software desarrollado en el Capitulo 4 se procedió a realizar las pruebas de la respuesta de la planta en tiempo real con el controlador PID diseñado para cada una de las perturbaciones de la planta a un set point de 30°C obteniéndose las siguientes gráficas descritas en la Tabla 5.1, Tabla 5.2, Tabla 5.3 y Tabla 5.4 que se presentan a continuación.

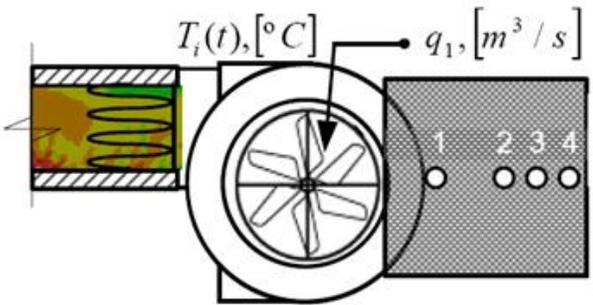
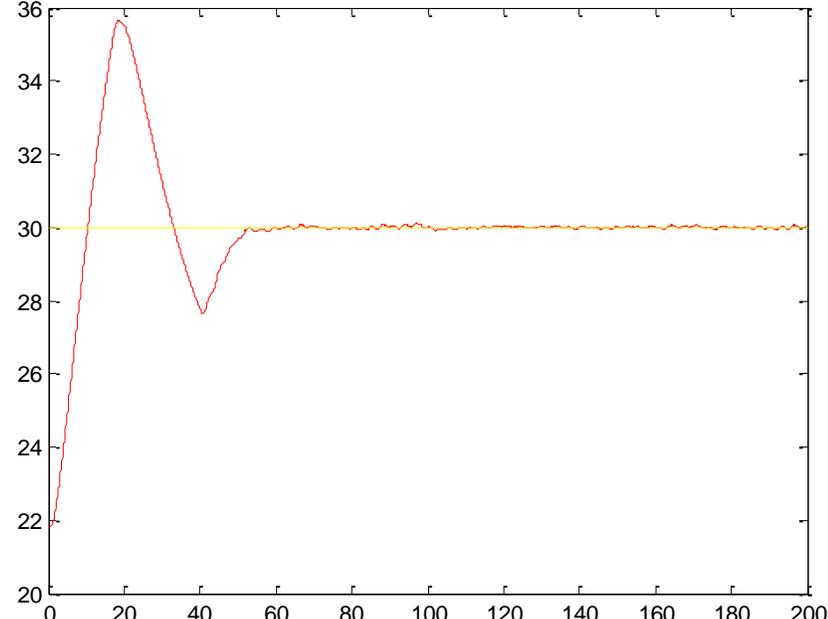
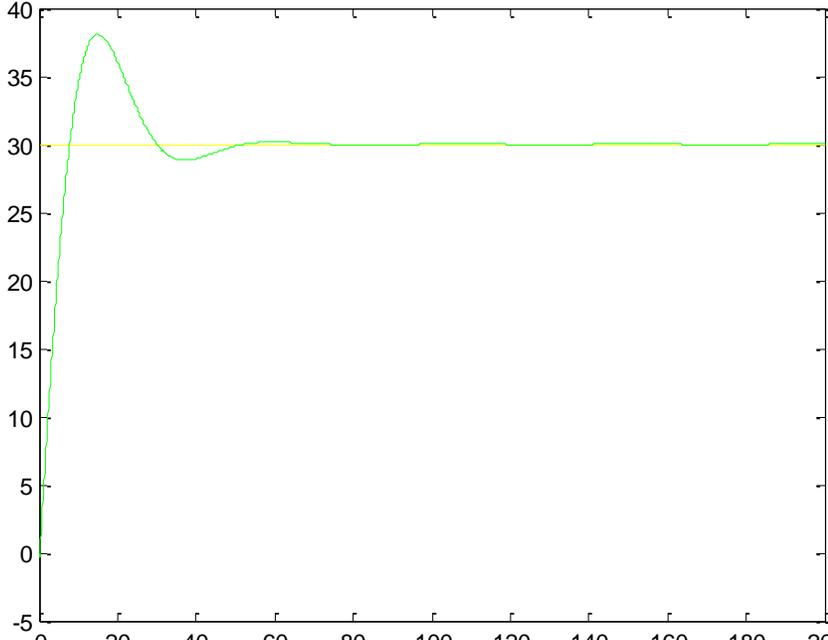
<p>PERTURBACIÓN 1</p>	
<p>CONTROL PID EXPERIMENTAL</p>	
<p>SIMULACIÓN DEL CONTROL PID</p>	

Tabla 5.1 Pruebas En La Perturbación 1

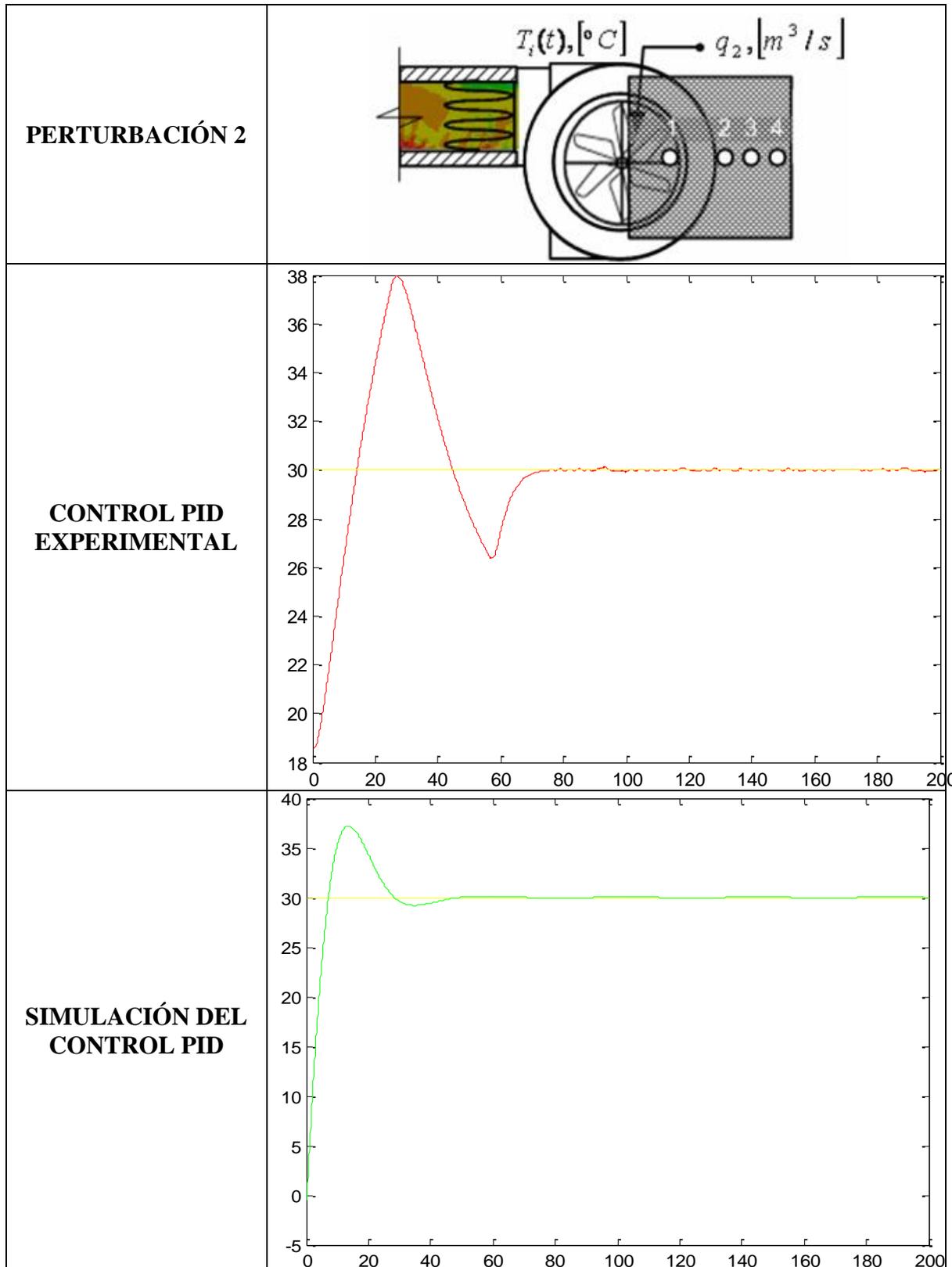


Tabla 5.2 Pruebas En La Perturbación 2

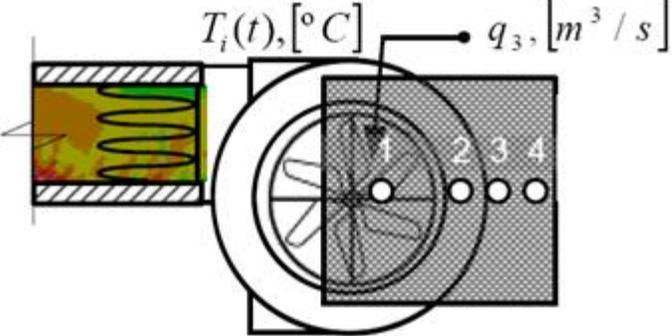
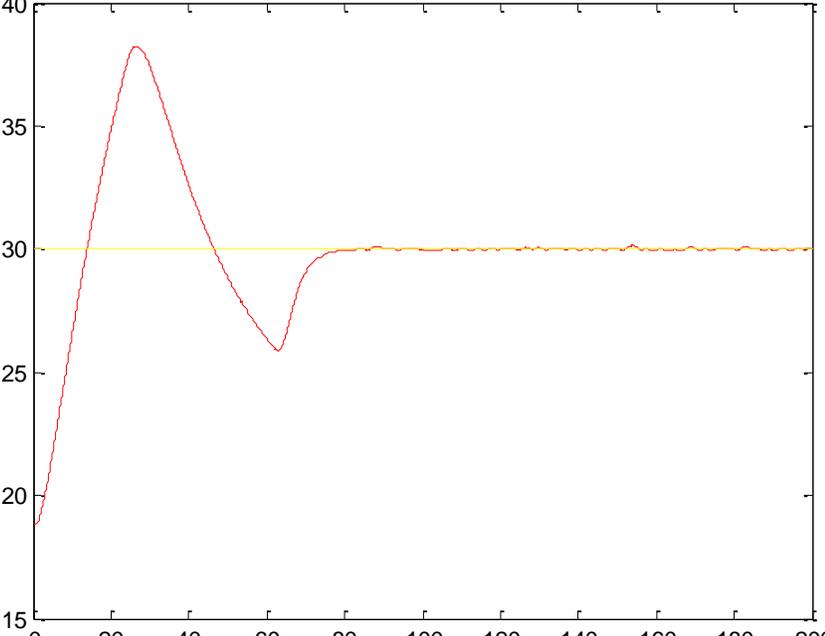
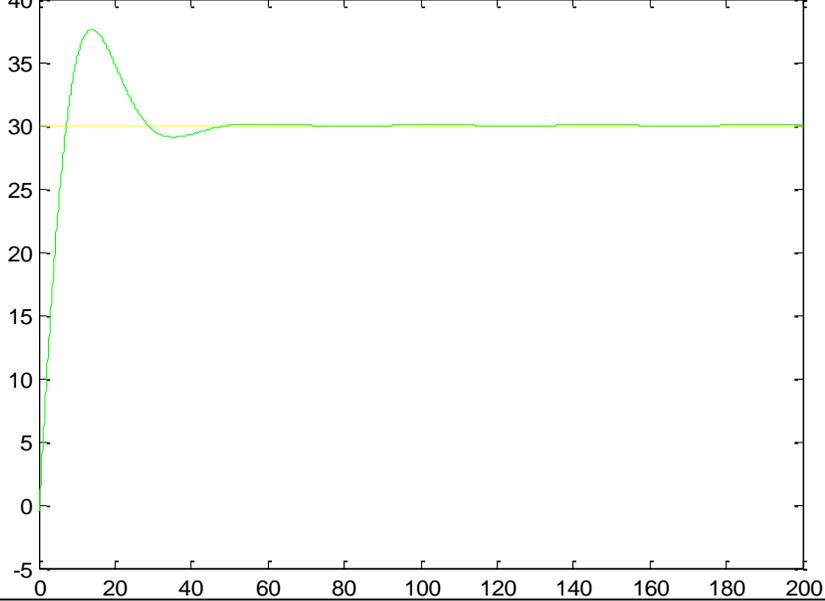
<p>PERTURBACIÓN 3</p>	 <p>The diagram shows a cross-section of a thermal system. On the left, a heater coil is shown with a temperature profile. A fan is in the center, and a sensor array with four points (1, 2, 3, 4) is on the right. Labels include $T_i(t), [^{\circ}C]$ and $q_3, [m^3 / s]$.</p>
<p>CONTROL PID EXPERIMENTAL</p>	 <p>The graph plots temperature $T_i(t)$ in $^{\circ}C$ against time in seconds. The y-axis ranges from 15 to 40, and the x-axis from 0 to 200. A horizontal yellow line is at 30 $^{\circ}C$. The red curve starts at 19 $^{\circ}C$, rises to a peak of 38 $^{\circ}C$ at 25s, dips to 26 $^{\circ}C$ at 65s, and stabilizes at 30 $^{\circ}C$ after 80s.</p>
<p>SIMULACIÓN DEL CONTROL PID</p>	 <p>The graph plots temperature $T_i(t)$ in $^{\circ}C$ against time in seconds. The y-axis ranges from -5 to 40, and the x-axis from 0 to 200. A horizontal yellow line is at 30 $^{\circ}C$. The green curve starts at 0 $^{\circ}C$, rises to a peak of 37 $^{\circ}C$ at 15s, dips to 29 $^{\circ}C$ at 35s, and stabilizes at 30 $^{\circ}C$ after 50s.</p>

Tabla 5.3 Pruebas En La Perturbación 3

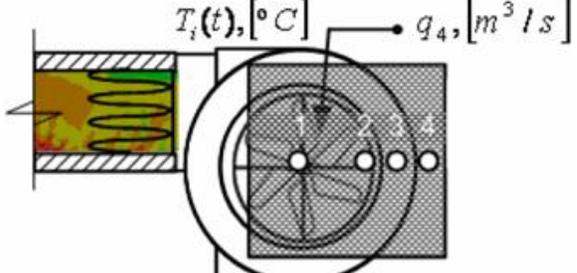
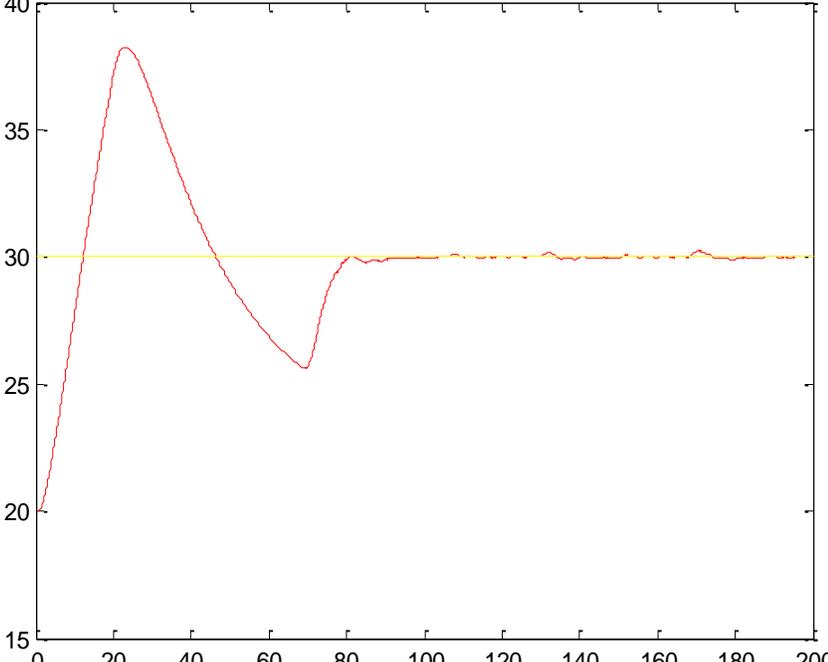
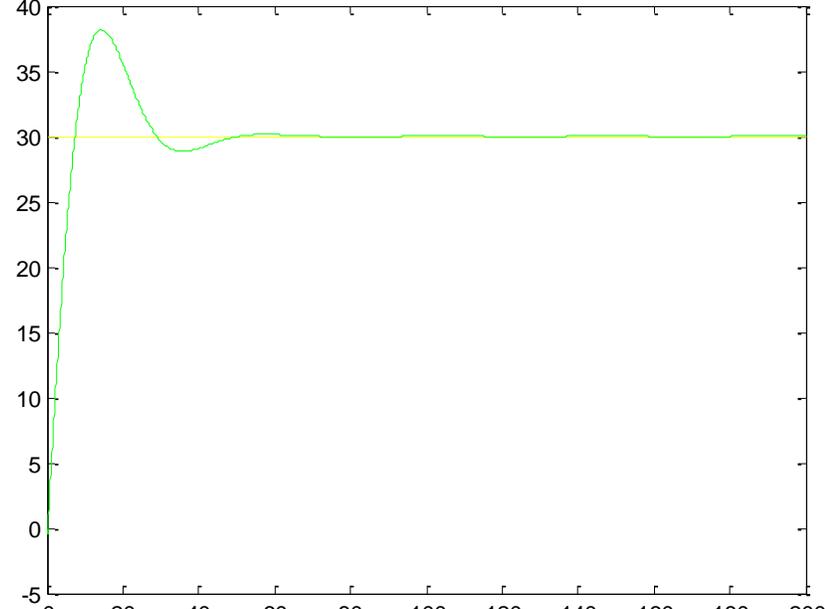
<p>PERTURBACIÓN 4</p>	
<p>CONTROL PID EXPERIMENTAL</p>	
<p>SIMULACIÓN DEL CONTROL PID</p>	

Tabla 5.4 Pruebas En La Perturbación 4

5.2 RESULTADOS

Para un mejor análisis de los resultados obtenidos se ha superpuesto las gráficas resultantes de la Simulación del control PID (curva en verde), el Control PID experimental (curva en rojo) y el *SetPoint* (amarillo).

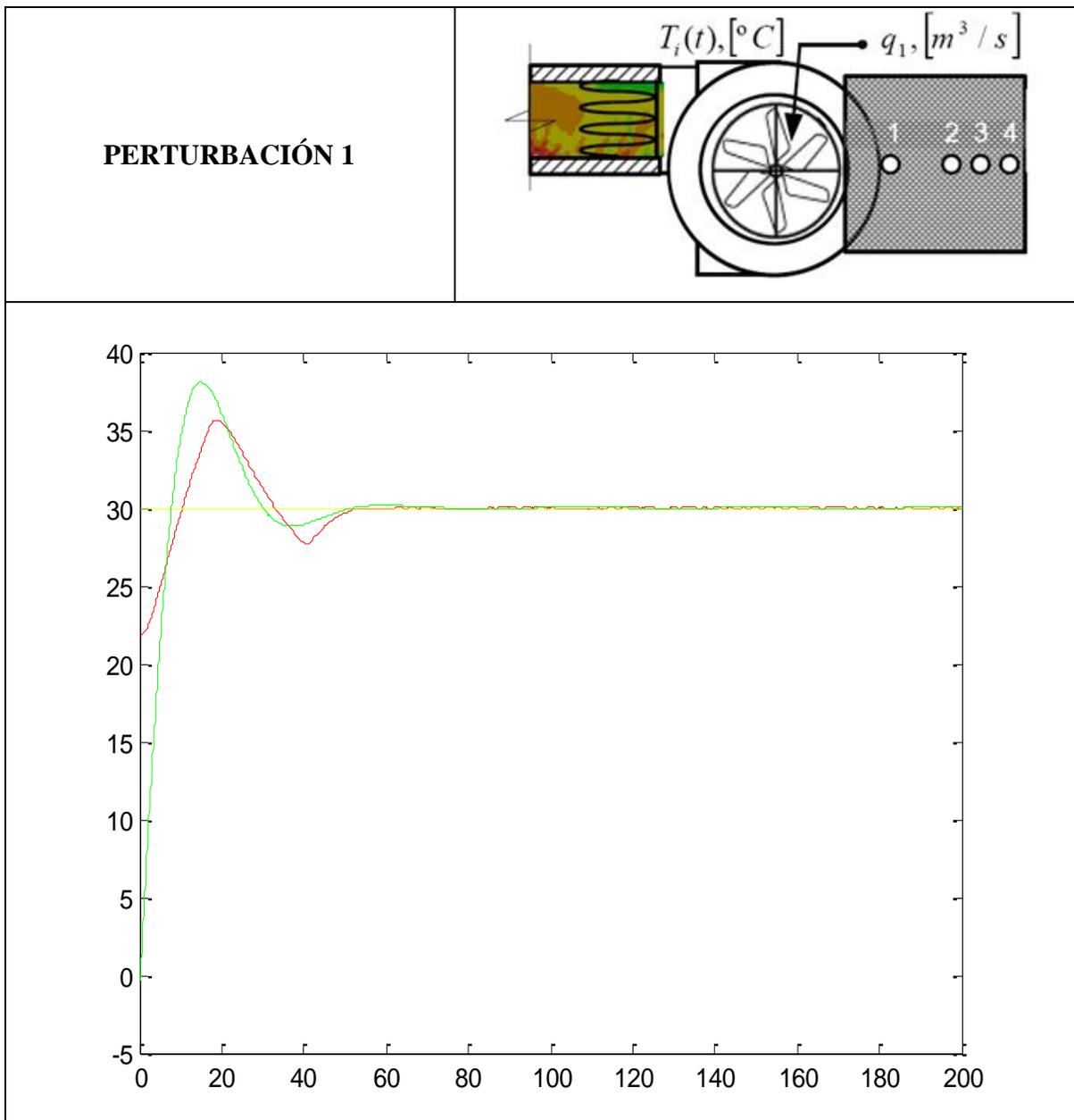


Tabla 5.5 Resultados En La Perturbación 1

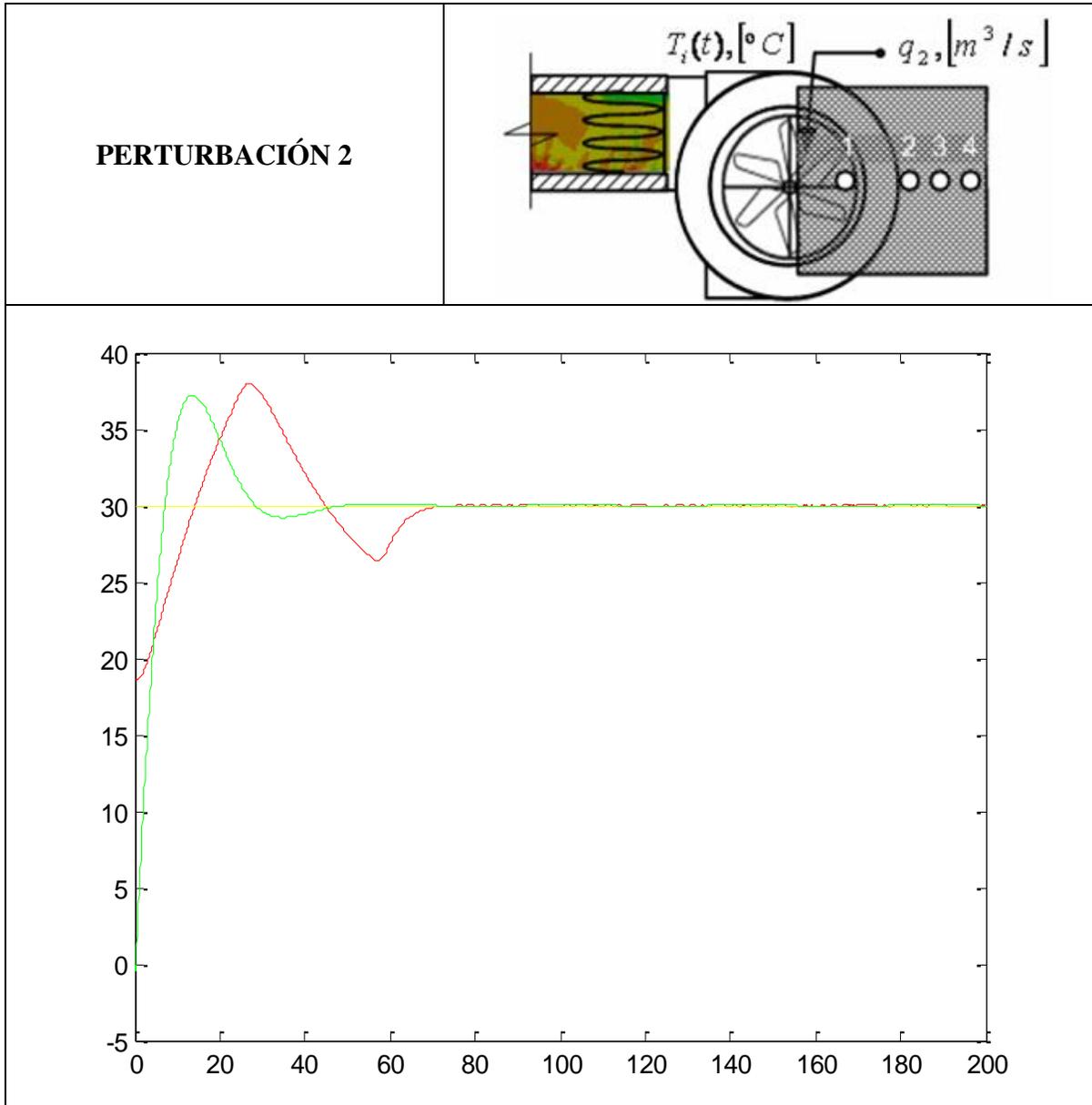


Tabla 5.6 Resultados En La Perturbación 2

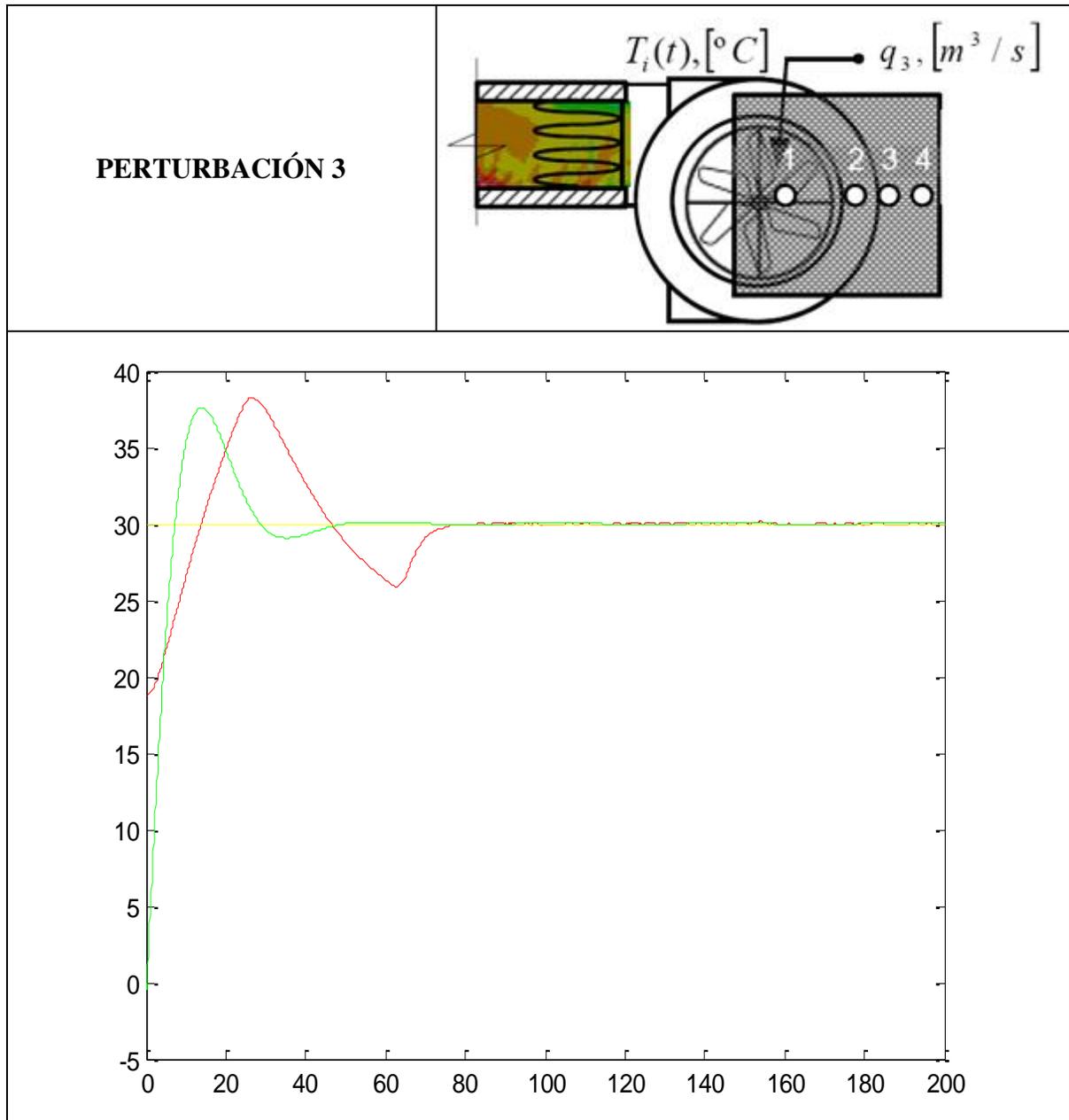


Tabla 5.7 Resultados En La Perturbación 3

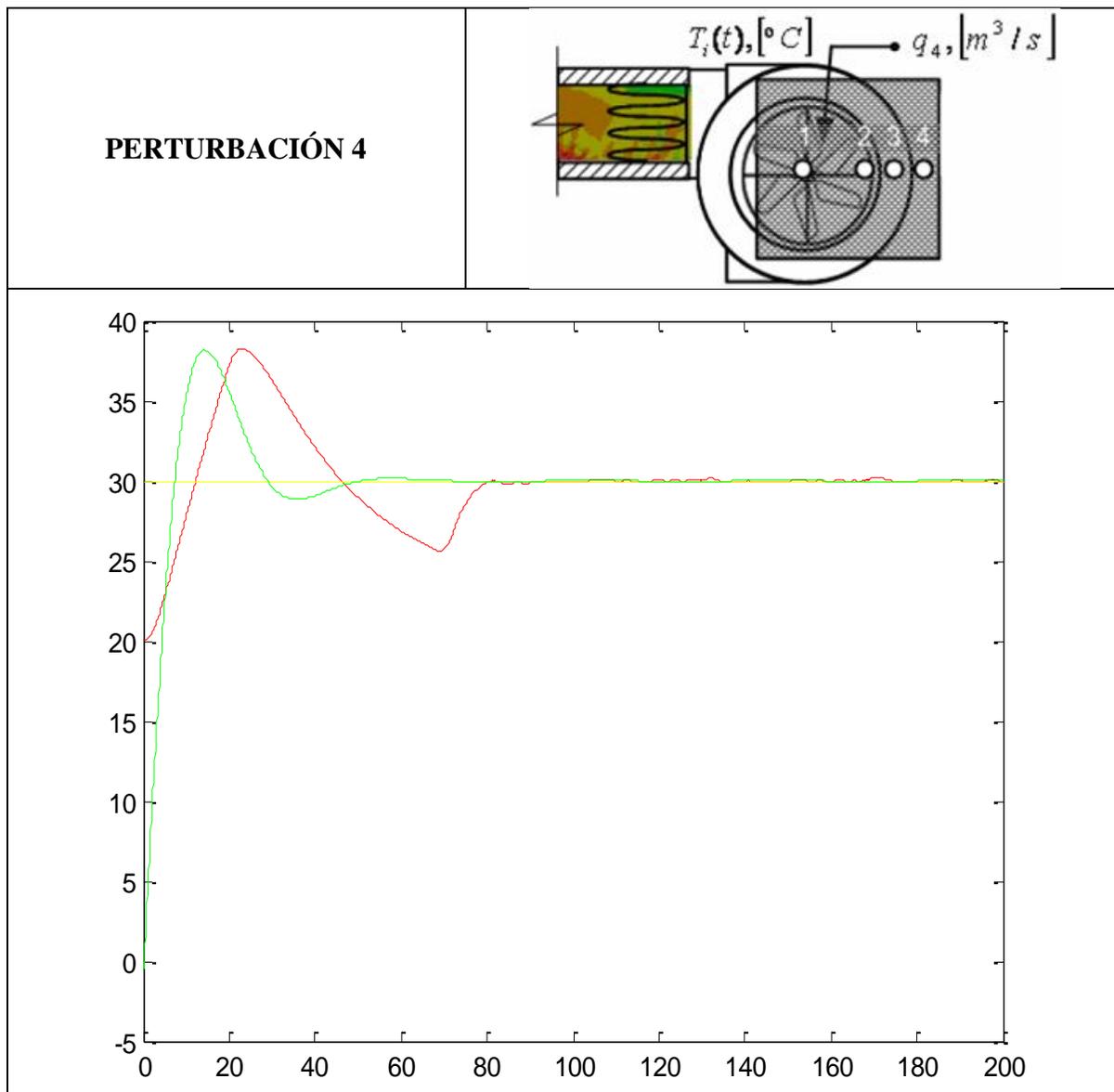


Tabla 5.8 Resultados En La Perturbación 4

Como se puede apreciar en las gráficas de la Tabla 5.2 el Control PID Experimental y la Simulación del control PID guardan cierta relación en los parámetros de importancia como son el Máximo Pico (M_p), el error en estado estacionario (e_{ss}) y el tiempo de estabilización (t_s), las variaciones que se pueden notar son el tiempo de la respuesta transitoria, puesto que como se explicó anteriormente en el Capitulo 4 el tiempo de la

respuesta transitoria dependía de los valores de la resistencia y capacitancia térmica al igual que la ganancia dependía netamente de la resistencia térmica, como estos parámetros varían cuando varía la temperatura ambiente ya que nuestro sistema es abierto existe ese margen de error entre el modelo teórico y el objeto experimental, pero los modelos matemáticos obtenidos tienen gran aproximación con la planta física real puesto que las gráficas fueron obtenidos con los mismos valores del controlador PID y con el mismo tiempo de muestreo dando como resultado una aproximación entre las gráficas altamente aceptable.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Mediante la realización de este proyecto se logró realizar el control en tiempo real por medio de la herramienta RTW de Matlab[®], consiguiendo de esta manera implementar un control PID para la unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM*.
- Para el desarrollo de un control en tiempo real con Matlab[®] es necesario minimizar al máximo los tiempos de retardo producidos en la adquisición de datos, esto se logra por que el programa de Simulink[®] y el procesamiento de datos se encuentran gobernados por el reloj del computador dándole la máxima prioridad a la ejecución del modelo en tiempo real frente a otras aplicaciones del computador.
- Una tarjeta de adquisición de datos con interfaz USB o serial se encuentra limitada por el buffer de comunicación siendo imposible conseguir la ejecución de un modelo en tiempo real puesto que existe un retardo en la adquisición de datos con respecto al computador.
- La herramienta RTW de Matlab[®] posee la ventaja de variar el tiempo de muestreo lo cual nos permite analizar plantas de respuesta rápida como es la unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM* o incluso plantas más complicadas como lo es la velocidad de un motor de Dc o de un servo motor.
- Al diseñar sistemas de control en tiempo real con Matlab[®] se obtienen los resultados reales del controlador diseñado, puesto que no existiría elementos discretos que introduzcan errores o cambios en el controlador.

- Por mucho tiempo MATLAB[®] se utilizó simplemente como una herramienta de modelamiento y para la implementación del controlador se la realizaba externamente y siempre existían variaciones en los resultados obtenidos. Con la aplicación de la herramienta RTW de MATLAB[®] se ha logrado comprobar al instante si un controlador está correctamente diseñado y se puede verificar en tiempo real el efecto en la planta de cambios en los parámetros del controlador ampliando notablemente la interacción del estudiante y el aprendizaje de la cátedra de sistemas de control.

- La unidad *AIR FLOW TEMPERATURE CONTROL SYSTEM* al ser un sistema abierto se ve afectado por las variaciones de temperatura del ambiente pero los modelos matemáticos obtenidos guardan concordancia entre las respuestas de la ejecución del modelo en tiempo real sobre la planta física y las respuestas de la ejecución de la simulación.

6.2 RECOMENDACIONES

- Para el diseño de un control en tiempo real se debe realizar el modelamiento con el mismo tiempo de muestreo que se utilizará en el control en tiempo real, puesto que al variar este parámetro se puede afectar a la identificación de la planta.

- Es recomendable siempre generar archivos de respaldo de extensión .mat como se explico anteriormente en el Capítulo 3 para documentar los resultados obtenidos.

- Cuando se realiza un modelo en tiempo real es necesario que todas las entradas y salidas se encuentren a un mismo tiempo de muestreo, porque como se ha explicado en el Capítulo 3, esta herramienta sincroniza la toma de muestras de datos con el reloj del computador y sus interrupciones.

- Por seguridad es recomendable utilizar bloques de saturación antes de los bloques de las salidas, para evitar comportamientos errados en el funcionamiento de un modelo en tiempo real.
- Cuando se cambian los bloques o las propiedades de los bloques es necesario reconstruir el modelo en tiempo real, por lo general si existe algún problema Matlab® marca el bloque de conflicto de color rojo y da un mensaje breve del error que sucede.
- Se recomienda leer el Capítulo 3 en especial el desarrollo del software donde se explica ampliamente la configuración de todos los parámetros para la realización de un modelo en tiempo real.
- Cuando se está ejecutando un modelo en tiempo real es recomendable no abrir otras aplicaciones puesto que puede producirse conflictos en el funcionamiento del computador, y para evitar dichos conflictos al finalizar el trabajo en tiempo real se deberá realizar la desinstalación del Real Time Windows Kernel.

REFERENCIAS BIBLIOGRÁFICAS

- [1] <http://iaci.unq.edu.ar/materias/laboratorio2/transparencias%5CReal%20Time%20Workshop.pdf>
- [2] MATLAB & Simulink Based Books
- [3] <http://digital.ni.com/manuals.nsf/websearch/8117DF4C5A29C95C862573020061023B>
- [4] <http://www.mathworks.es>
- [5] MINIGUANO, Henry; **Diseño E Implementación De Un Controlador Robusto Para El Módulo Experimental De Control De Temperatura De Un Flujo De Aire PCT-2**, 2008.
- [6] INCROPERA Frank, **Fundamentals of Heat and Mass Transfer**, 2002.
- [7] MERLE C. POTER David, **Mecánica De Fluidos**, 2001.
- [8] OGATA, Katsuhiko, **Ingeniería de control Moderna**, 1993.
- [9] <http://www.iberchip.org/VIII/docs/posters/p9.pdf>
- [10] <http://www.definicion-es.com/red.php>

ANEXOS

ANEXO 1
GLOSARIO

GLOSARIO

.

- **.c.-** extensión de código generado en lenguaje C.
- **.exe.-** extensión de una aplicación ejecutable.
- **.h.-** extensión de los archivos de cabecera o librerías de un lenguaje de programación.
- **.m.-** extensión de un archivo de código generado por Matlab®.
- **.mdl.-** extensión de un archivo de código generado por Simulink®.
- **.mk.-** extensión de un archivo de código generado por el RTW.
- **.rtw.-** extensión de un archivo de código intermedio generado por el RTW.
- **.tmf.-** extensión de un archivo de configuración del TLC.

A

- **ACONDICIONAMIENTO.-** tratamiento de una señal para que se comporte de una manera deseada.
- **ACTUADOR.-** parte del lazo de control encargada de realizar los cambios sobre la variable manipulada.
- **AJUSTE FINO.-** obtención de parámetros satisfactorios mediante prueba y error.

B

- **BARRIDO.-** toma de un número definido de muestras en un determinado tiempo.
- **BLINDADO.-** protección física contra el ruido eléctrico en un cable.
- **BUFFER.-** conjunto de datos que se encuentran en un puerto de entrada o salida.
- **BUILD.-** construcción del modelo en software.



- **C++.-** lenguaje de programación.
- **CONNECT TO TARGET.-** establecer conexión entre el modelo de Simulink® y el ejecutable en tiempo real .exe.
- **CONTROL.-** Medir el valor de la variable controlada del sistema y aplicar al sistema la variable manipulada para corregir o limitar la desviación del valor medido, respecto al valor deseado.



- **DRIVER.-** o gobernador encargado de controlar que se ejecute correctamente un determinado hardware o software.



- **emlc.-** función de Matlab® que permite la generación de código fuente en lenguaje C (archivos .c, .h, .rtw, .exe, etc).
- **ess.-** error en estado estacionario el cual es la diferencia algebraica entre el valor de entrada al lazo de control y el valor realimentado u obtenido.
- **EXPERIMENTACIÓN.-** conexión física entre el modulo real y el software creado en Simulink®.



- **FUNCIÓN DE TRANSFERENCIA.-** otra forma de presentación de las propiedades dinámicas de los elementos y sistemas (lineales) de control automático, la función de transferencia $G(s)$ es la relación de la variable de salida $y(s)$ a la variable de entrada $x(s)$



- **HOST.-** Es el sistema donde se ejecuta MATLAB, Simulink y el Real Time WorkShop. Es allí donde se crea el código que luego será cargado en el target[9].



- **IC.-** tipo de sensor integrado de temperatura con respuesta lineal.
- **IDENT.-** System Identification Toolbox el cuál es un componente de Matlab® utilizado para la identificación de modelos de plantas físicas.
- **IDENTIFICACIÓN.-** obtención de todas las características inmersas en una planta física.
- **InPort.-** puerto de entrada de datos.
- **INTERFERENCIA.-** es cualquier proceso que altera, modifica o destruye una señal durante su trayecto en el canal existente entre el emisor y el receptor.
- **INTERRUPCIÓN.-** es una señal recibida por el procesador de un computador, indicando que debe "interrumpir" el curso de ejecución actual y pasar a ejecutar código específico para tratar una situación en particular.

K

- **Kd.-** constante derivativa la cual determina la reacción del tiempo en el que el error se produce.
- **KERNEL.-** funciona interceptando las interrupciones del reloj de la PC, antes que Windows las llegue a recibir, permitiendo de este modo una ejecución a tiempo real del modelo. Para alcanzar un muestreo preciso, el kernel reprograma el reloj de la CPU a una mayor frecuencia. A causa que el reloj es también la fuente primaria de tiempo para Windows, el kernel también envía una interrupción de tiempo al sistema operativo a la frecuencia original.
- **Ki.-** constante de integración la cual genera una corrección proporcional a la integral del error.
- **Kp.-** constante proporcional la cual determina la reacción del error actual.

L

- **LAZO ABIERTO.-** es un tipo de sistema de control en el cual la salida no se encuentra realimentada.
- **LAZO CERRADO.-** es un tipo de sistema de control en el cual la salida se encuentra realimentada.
- **LIBRERÍA.-** es un conjunto de subprogramas utilizados para desarrollar software.

M

- **MAKE FILE.-** archivo de configuración para generación de archivos para trabajo en tiempo real.

- **MATLAB®**.- abreviatura de MATrix LABoratory, "laboratorio de matrices" es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). es un programa de cálculo numérico orientado a matrices.
- **MODELO**.- Representa gráficamente las relaciones matemáticas dependientes del tiempo a través de las entradas, estados y salidas del sistema.
- **MODO EXTERNO**.- Permite que dos sistemas separados, el host y el target, se comuniquen. El host (donde reside Simulink) transmite mensajes al target requiriendo que acepte cambio de parámetros y solicitando una actualización en datos de señal. El target responde ejecutando el requerimiento. La comunicación de modo externo utiliza la arquitectura cliente / servidor, en la cual el Simulink es el cliente y el target es el servidor.
- **MP**.- máximo pico, el cuál es el valor máximo de la variable controlada.
- **MUESTRA**.- es un dato obtenido de un proceso en el transcurso del tiempo.
- **MUESTREO**.- corresponde a la velocidad de la toma de datos.



- **OFFSET**.- corresponde a la desviación existente entre el valor de ajuste y el valor de estabilización.
- **OSCILACIÓN**.- corresponde a una variación, perturbación o fluctuación en el tiempo de un medio o sistema.
- **OutPort**.- puerto de salida.



- **PID**.- sistema de primer orden.

- **PCI.-** Peripheral Component Interconnect (Interconexión de Componentes Periféricos) consiste en un bus de computador estándar para conectar dispositivos periféricos directamente a su placa base.
- **PERTURBACIÓN.-** Señal que tiende a afectar adversamente el valor de la salida del sistema.
- **PID.-** mecanismo de control por realimentación el cual corrige el error entre un valor medido y el valor que se quiere obtener mediante el cálculo de tres parámetros distintos los cuales son el proporcional, el integral y el derivativo.
- **PLANTA.-** Es un equipo o un juego de piezas de una máquina funcionando conjuntamente, cuyo objetivo es realizar una operación determinada. Cualquier objeto físico que deba controlarse.
- **PLATAFORMA.-** Sistema donde se encuentran las librerías nativas para que se ejecuten otras aplicaciones y programas.
- **PROCESO.-** Es una operación o desarrollo natural, caracterizado por una serie de cambios graduales progresivamente continuos, que se suceden uno a otro de un modo relativamente fijo y que tienden a un determinado resultado final. Un proceso es cualquier operación que deba controlarse.
- **PROCESADOR.-** Dispositivo encargado de procesar y administrar la información recibida y emitir señales de control a los dispositivos periféricos conectados a él dentro del computador.
- **PV.-** variable del proceso.



- **RAW.-** opción de configuración del bloque de parámetros de entrada o salida analógica de Simulink® que permite convertir una señal analógica a digital.
- **RELOJ.-** La frecuencia de reloj indica la velocidad a la que un computador realiza sus operaciones, se mide en ciclos por segundo (hercios). Los diferentes circuitos integrados de un computador pueden funcionar a diferentes frecuencias de reloj, por lo que cuando se usa el término frecuencia de reloj aplicado a un computador, suele

sobreentenderse que se refiere la velocidad de funcionamiento del procesador principal. En el contexto de la electrónica digital, más concretamente en la secuencial, la frecuencia de la señal de reloj es enviada a todos los dispositivos para sincronizar la operación de los mismos entre sí[10].

- **RETARDO.-** es el tiempo que necesita una señal para atravesar un conductor o dispositivo.
- **ROBUSTO.-** sistema que continua con su correcto funcionamiento pese a que las condiciones de funcionamiento se tornen extremas.
- **RTW.-** Real Time Workshop es el encargado de producir código directamente desde los modelos desarrollados en Simulink y automáticamente construye programas que pueden ser corridos en una variedad de entornos. Además permite utilizar Simulink para analizar los datos de salida del ejecutable. Con Real-Time Workshop, se puede generar código C para sistemas de tiempo discreto, de tiempo continuo, tiempo real o híbridos.
- **rtwho.-** comando de Matlab® el cual nos permite verificar la instalación del Real Time Kernel.
- **rtwin.tlc.-** archivo de identificación del sistema.
- **rtwinlib.-** comando de Matlab® el cual nos permite ingresar a la librería del Real Time Windows Target.
- **rtwintgt.-** comando de Matlab® el cual nos permite realizar la desinstalación del real Time Windows Target.,
- **RTWT.-** Real Time Windows Target es un entorno que genera código de tiempo real, utilizando una PC como target y host al mismo tiempo. Es en esencia un complemento del Real Time Workshop.
- **RUIDO.-** se considera ruido a todas las perturbaciones eléctricas que interfieren sobre las señales transmitidas o procesadas.
- **RUN.-** ícono de Simulink® que nos permite iniciar el proceso de simulación.



- **SAMPLE TIME.**- tiempo de muestreo al cual se toma o escribe datos.
- **SATURACIÓN.**- límites de valores a los que estará regido el funcionamiento de un proceso.
- **SCOPE.**- osciloscopio en Simulink®.
- **SENSOR.**- denominado también elemento primario, el cual es el encargado de la medición de la variable controlada.
- **SET POINT.**- es el valor de referencia al cual se desea que llegue el sistema controlado.
- **SIMULACIÓN.**- proceso en el cual se observa el funcionamiento de un proceso de manera virtual.
- **SIMULINK®.**- Es un paquete de Software, complemento de MATLAB® que permite modelar, simular y analizar sistemas dinámicos, esto es sistemas cuyas salidas y estados internos cambien con el tiempo. Es un entorno grafico, donde se crea un modelo en bloques del sistema, utilizando librerías de bloques estándar y un editor que permite interconectar los mismos.
- **SISTEMA.**- Es una combinación de componentes que actúan conjuntamente y cumplen un determinado objetivo.
- **SINTONIZACIÓN.**- obtención de los parámetros de funcionamiento de un controlador por medio de algún método.
- **SOBREPICO.**- valor máximo de la variable controlada antes del tiempo de estabilización.
- **SOLVER.**- cuadro de configuración de los parámetros de simulación en Simulink®.
- **START TIME.**- corresponde al tiempo en el cual deseamos que comience la simulación.
- **STEP.**- función paso o escalón $u(t)$.
- **STOP TIME.**- corresponde al tiempo en el cual deseamos que termine la simulación.

- **SUBSISTEMA.-** corresponde en Simulink® a un subprograma que realiza un proceso para un programa principal.



- **TARGET.-** Es el entorno (de software o hardware) en el cual un código se ejecutará.
- **Td.-** Tiempo diferencial.
- **TEMPLATE MAKEFILE.-** archivos del RTW que generarán el archivo en tiempo real utilizando los archivos creados en C a partir de una plantilla predefinida por lo cual estos archivos .tmf también permitirán obtener aplicaciones en tiempo real para diferentes plataformas.
- **Ti.-** Tiempo integral.
- **TIEMPO REAL.-** se habla de tiempo real cuando el tiempo de actualización o muestreo es menor o igual al tiempo en que se producen cambios considerables en el proceso controlado.
- **TIMESLICE.-** es la frecuencia actual de los tiempos de interrupción del hardware.
- **TLC.-** Target Language Compiler, es un compilador utilizado por el RTW para trasladar el modelo del sistema en Simulink a un código intermedio.
- **TOOLBOX.-** caja de herramientas de Matlab®
- **ts.-** tiempo de estabilidad el cual es el tiempo que se demora en tornarse cero el error en estado estacionario.



- **$u(t)$.-** step o función paso conocida también como escalón.
- **UNIDADES CRUDAS.-** son unidades que no poseen representación física en el mundo real como por ejemplo valores de registro.

- **UNIDADES DE INGENIERÍA.**- son unidades que poseen representación física en el mundo real como por ejemplo metros, °C, kilogramos, etc.



- **VARIABLE CONTROLADA.**- Cantidad o condición que se mide y controla. Normalmente es la salida del sistema.
- **VARIABLE MANIPULADA.**- Cantidad o condición modificada por el controlador, a fin de afectar la variable controlada.



- **WORKSPACE.**- entorno de trabajo de Matlab®.

ANEXO 2

MANUAL DE USUARIO

MANUAL DE USUARIO

El siguiente manual de usuario está enfocado en la explicación y orientación para llevar a cabo el diseño y la ejecución de un modelo en tiempo real basado en la utilización del software creado denominado “APLICACIÓN DEL SISTEMA DE CONTROL RTW EN UN CONTROL PID”, por lo que el presente manual de usuario se lo ha dividido en los siguientes ítems:

- Diagrama de Conexiones.
- Identificación de la Planta.
- Simulación Del Control PID.
- Control PID Experimental.

✓ **Diagrama de Conexiones.**

Para dar ejecución al modelo en tiempo real enfocado al control PID de la Unidad Air Flow Temperature Control System mediante la herramienta RTW de Matlab[®] se deberá realizar la implementación del siguiente diagrama de conexiones tal como se puede apreciar en la Figura 1.

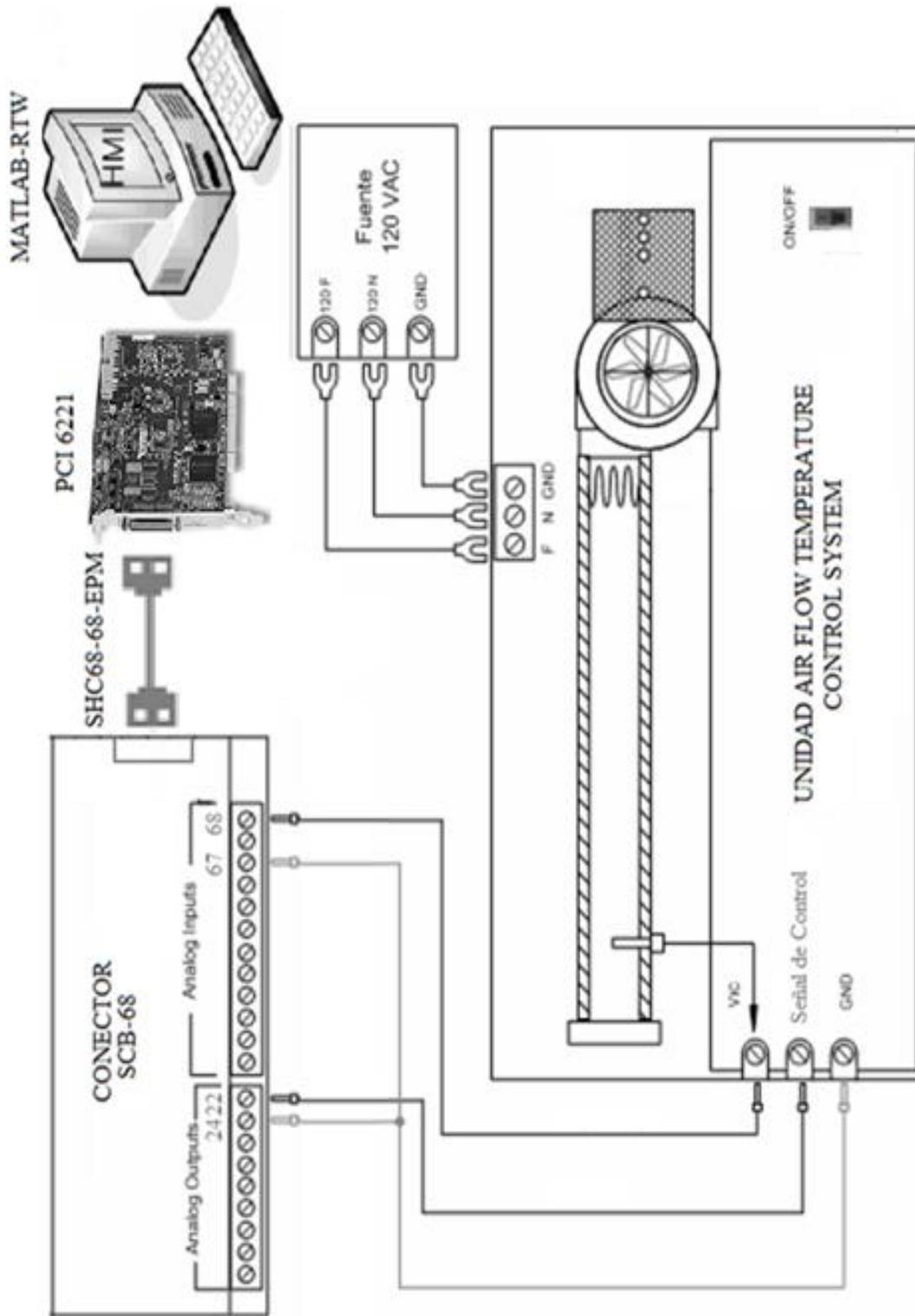


Figura 1 Diagrama De Conexiones

Posteriormente, al abrir el archivo control_rtw.mdl se despliega la siguiente ventana la cual podemos apreciar en la Figura 2.

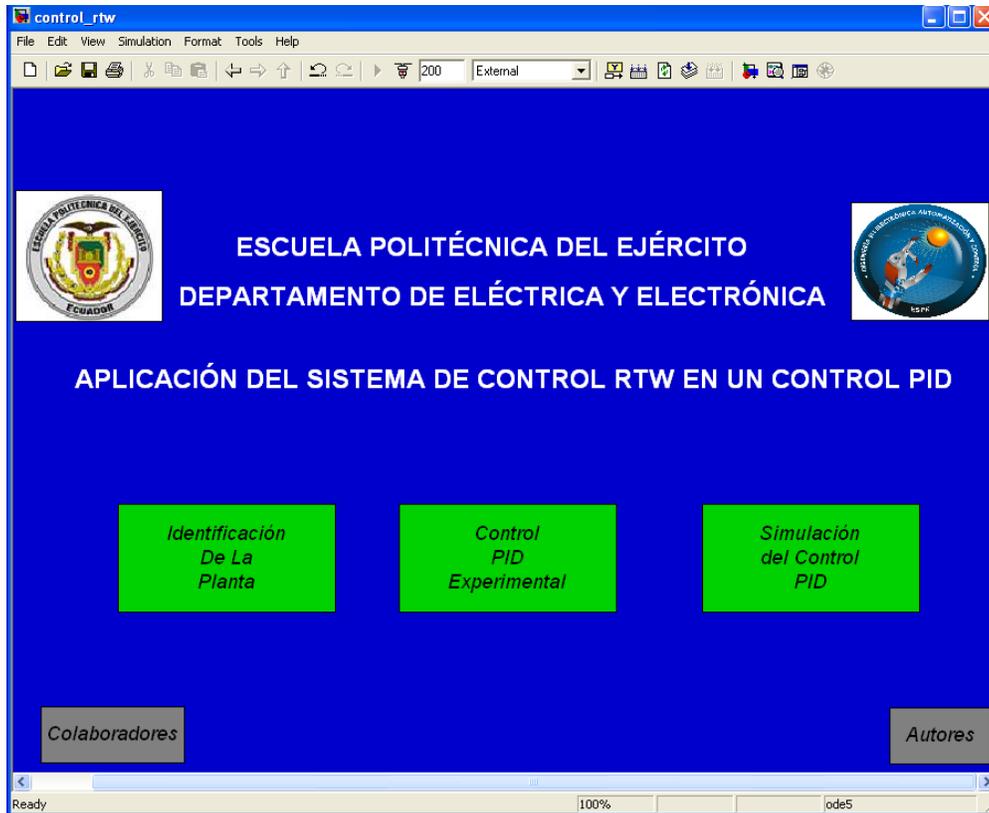


Figura 2 Pantalla De Ingreso Al Sistema

✓ Identificación De La Planta

Al dar doble click sobre el botón de Identificación De La Planta se abre el mismo, el cual contiene subsistemas destinados a obtener la respuesta de la planta ante un escalón.

El valor del step debe estar entre cero y uno puesto que el subsistema posee una ganancia de 10, esto significa que al poner el valor de 1 en el step estaremos dando el valor máximo que soporta el actuador de la planta (10V).



Figura 2 Identificación De La Planta

Una vez concluido el tiempo de simulación en tiempo real utilizaremos el programa para realizar la identificación de la planta “Modelamiento.m”, el mismo que se encuentra detallado en el Anexo 3, para que utilicemos la herramienta de identificación de matlab, al ejecutar el programa se debe revisar si el tiempo de muestreo que se utilizó en el modelo en tiempo real y el que se encuentra en el programa son los mismos.

Posteriormente al ejecutar el programa se abrirá la herramienta de identificación de Matlab®.

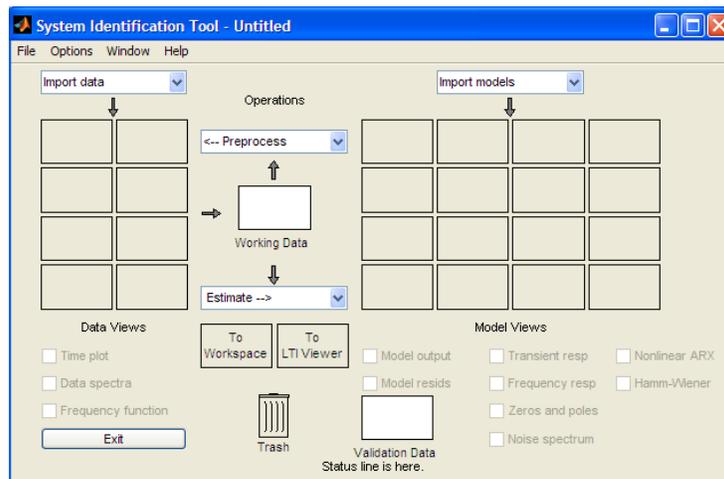


Figura 3 System Identification Tool (Ident)

Para la importación de los datos desde el Workspace a ident se debe seguir los siguientes pasos:

- En la Pestaña de Import data seleccionamos Time domain data...
- Se abre una ventana Import Data con una pestaña en la cual seleccionaremos Data Object.
- Posteriormente cambiara el formato de esta ventana y en la opción “Object:” escribiremos el nombre de la variable que en nuestro caso es “data”.
- Con el cual debe llenarse automáticamente las casillas inferiores.
- Damos click en el botón Import
- Finalmente damos click en close

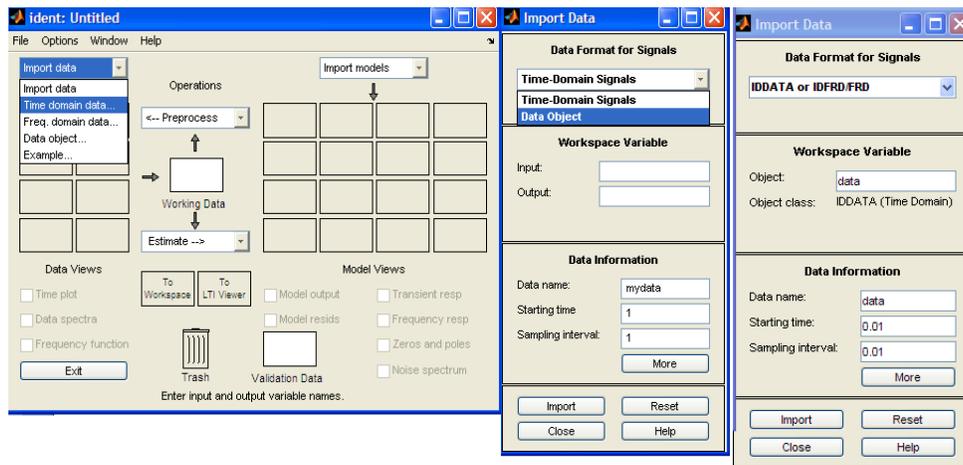


Figura 4 Carga Del Dato En La Herramienta Ident

A continuación esta herramienta nos permite obtener la descripción de la planta en Data Views y la estimación matemática del modelo seleccionando el método mediante la opción Estimate - - >. En este caso, se ha utilizado la opción Estimate → Process models... y Estimate → Quick starts, para una elección de la mejor estimación. Con esto se obtuvieron las siguientes estimaciones:

Al seleccionar Process Model nos aparece una ventana como la que se muestra en la Figura 5 que se presenta a continuación, en la cual seleccionamos la opción Estimate.

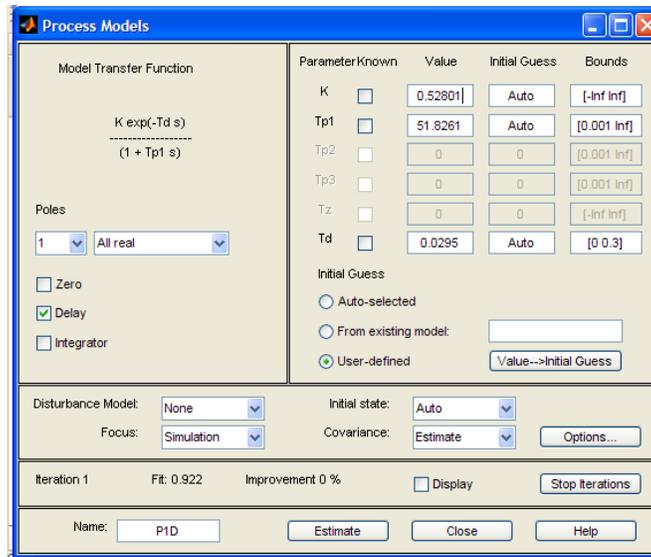


Figura 5 Process Model

Con la selección de Quick start nos aparecen otras estimaciones.

La Figura 6 nos muestra diferentes estimaciones, y nos dice cual es la mejor opción.

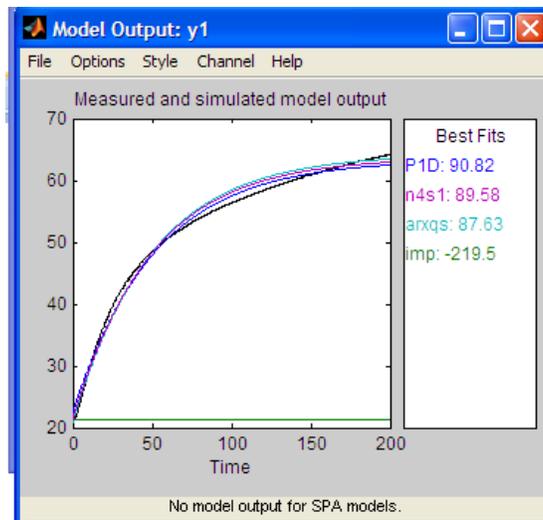


Figura 6 Model Output

Finalmente para continuar con el modelamiento exportamos estas estimaciones arrastrándolas hasta el recuadro To Workspace, como se muestra en la Figura 7 que se presenta a continuación.

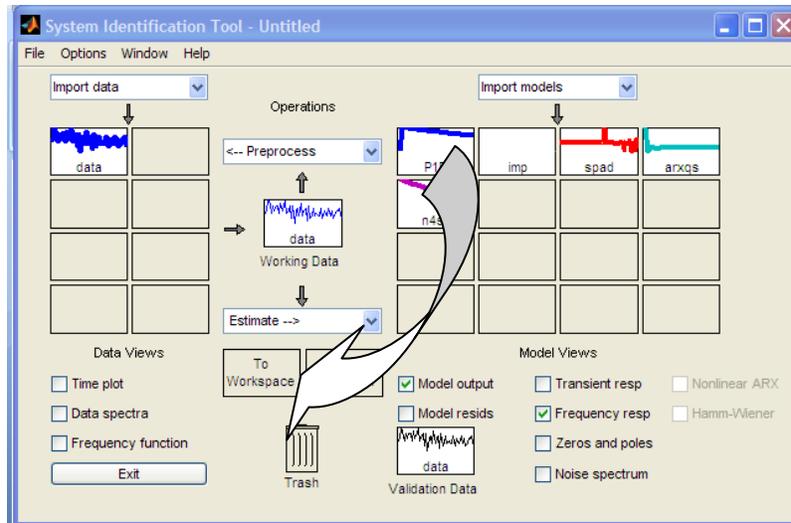


Figura 7 Carga De La Aproximación En El Workspace

Y finalmente se da enter en la ventana de comandos de Matlab para obtener los modelos matemáticos de cada perturbación, dichos modelos matemáticos nos servirán para el modelo de simulación.

Para el diseño del controlador se utilizará el Control PID Experimental.

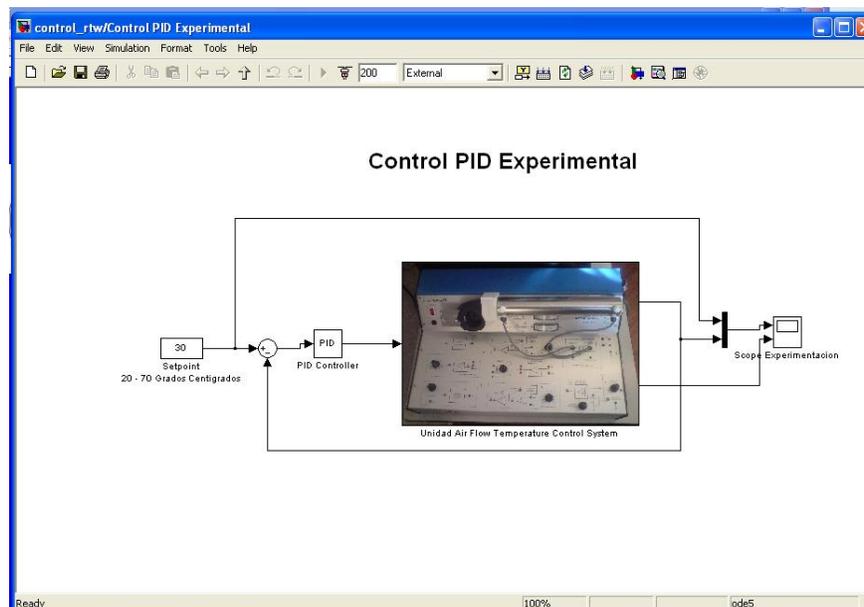


Figura 8 Control PID Experimental

En donde, en el bloque PID utilizaremos los siguientes valores $K_i = 0$, $K_d = 0$ y $K_p = K$ crítico (K_{cr}) obtenido la Figura 9 que se presenta a continuación.

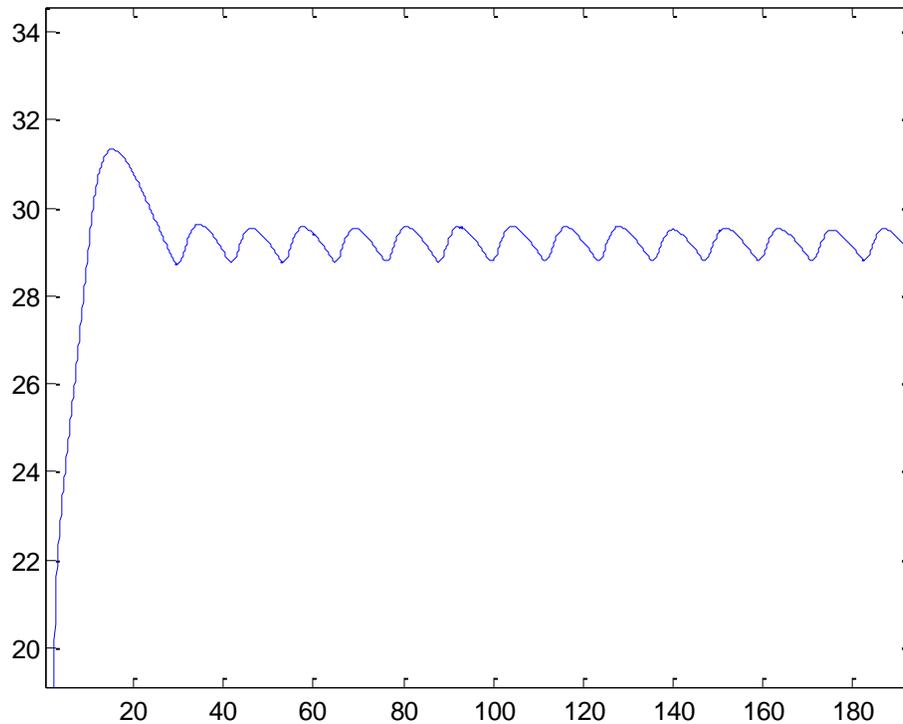


Figura 9 Oscilaciones Sostenidas De La Planta K_{cr} En Tiempo Real

De la Figura 9 el K_{cr} encontrado es 34, analizando los datos entre crestas se encuentra el P crítico (P_{cr}) que es 12 seg, empleando el segundo método de Zeigler Nichols el cuál establece que:

Tipo de controlador	K_p	τ_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Tabla 1 Regla Se Sintonización De Ziegler Nichols Segundo Método

Se obtuvieron los parámetros:

- $K_p = 20,4$
- $T_i = 6$
- $T_d = 1,5$
- $K_i = K_p/T_i = 3.4$
- $K_d = K_p * T_d = 30,6$

Al utilizar los valores de K_p , K_d y K_i anteriormente detallados en el bloque PID del control PID experimental y simulación del control PID, el resultado obtenido es el control PID en tiempo real de la planta física y de los modelos matemáticos calculados correspondientemente.

Se debe tener en cuenta que al utilizar el objeto de experimentación y la identificación de la planta estas no se encuentran asignadas a la misma salida y entrada analógica por lo que es menester realizar el cambio por software de las mismas entrando al bloque de salida o entrada analógica y variando su valor debiendo dejarlo en 1 el que se va a utilizar y en 2 el que no, como se puede apreciar en la Figura 10.

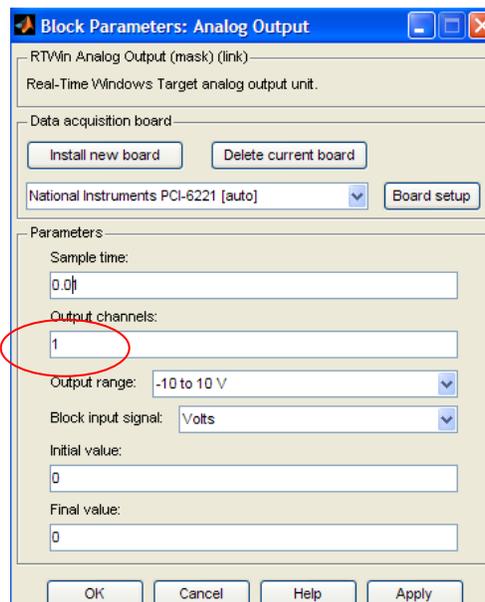


Figura 10 Bloque De Parámetros De La Salida Analógica

Para el la Simulación Del Control PID se anexó un bloque el cual nos permite guardar en un archivo .m los valores resultantes de la simulación dicho bloque es el “*to file*” como se puede apreciar en la Figura 11, pero este se lo utilizó solo en el Modelo de Simulación debiendo borrarlo cuando se utilice la identificación de la planta o el objeto de experimentación ya que el bloque *to file* es incompatible con la ejecución del modelo en tiempo real, además para correr el Modelo de simulación se deberá cambiar la forma de correr el modelo de Simulink® de *External Mode* a *Normal Mode*.

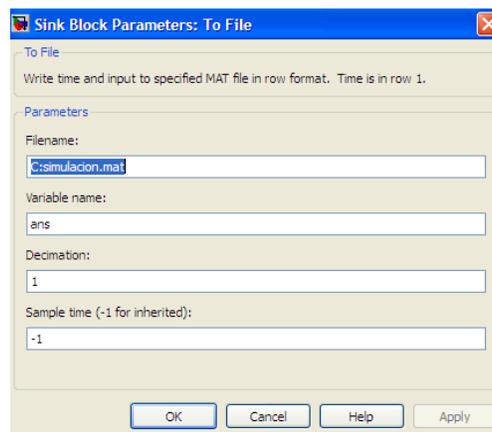
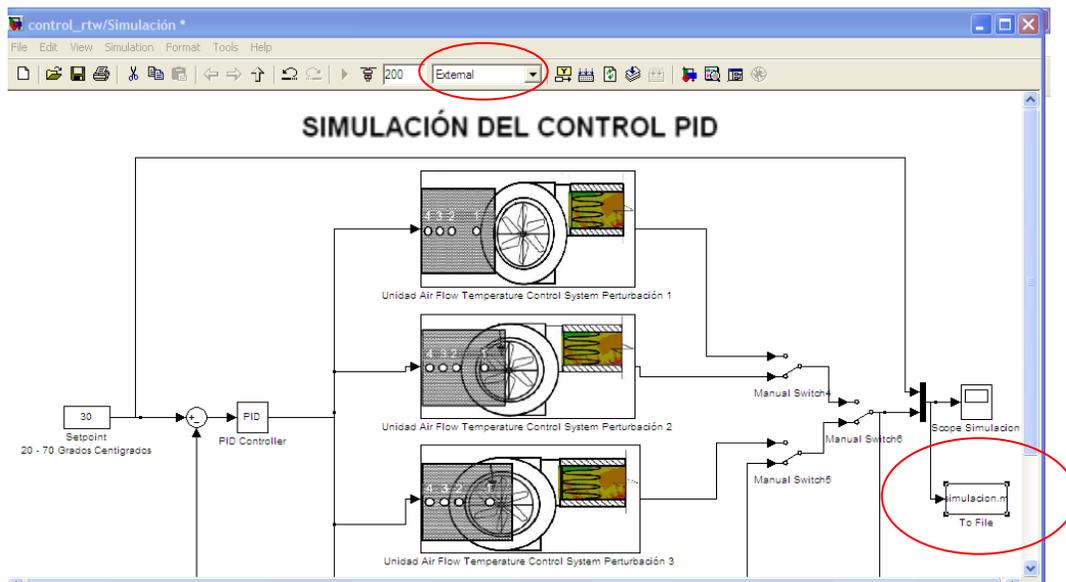


Figura 11 Bloque De Parámetros Del To File

Finalmente se ejecuta el programa para realizar la comparación de las gráficas resultantes obtenidas detallado en el Anexo 4 “Comparación.m”, para de esa manera superponer la grafica obtenida en el modelo de simulación y la gráfica obtenida en el objeto de experimentación como se muestra en la Figura 13.

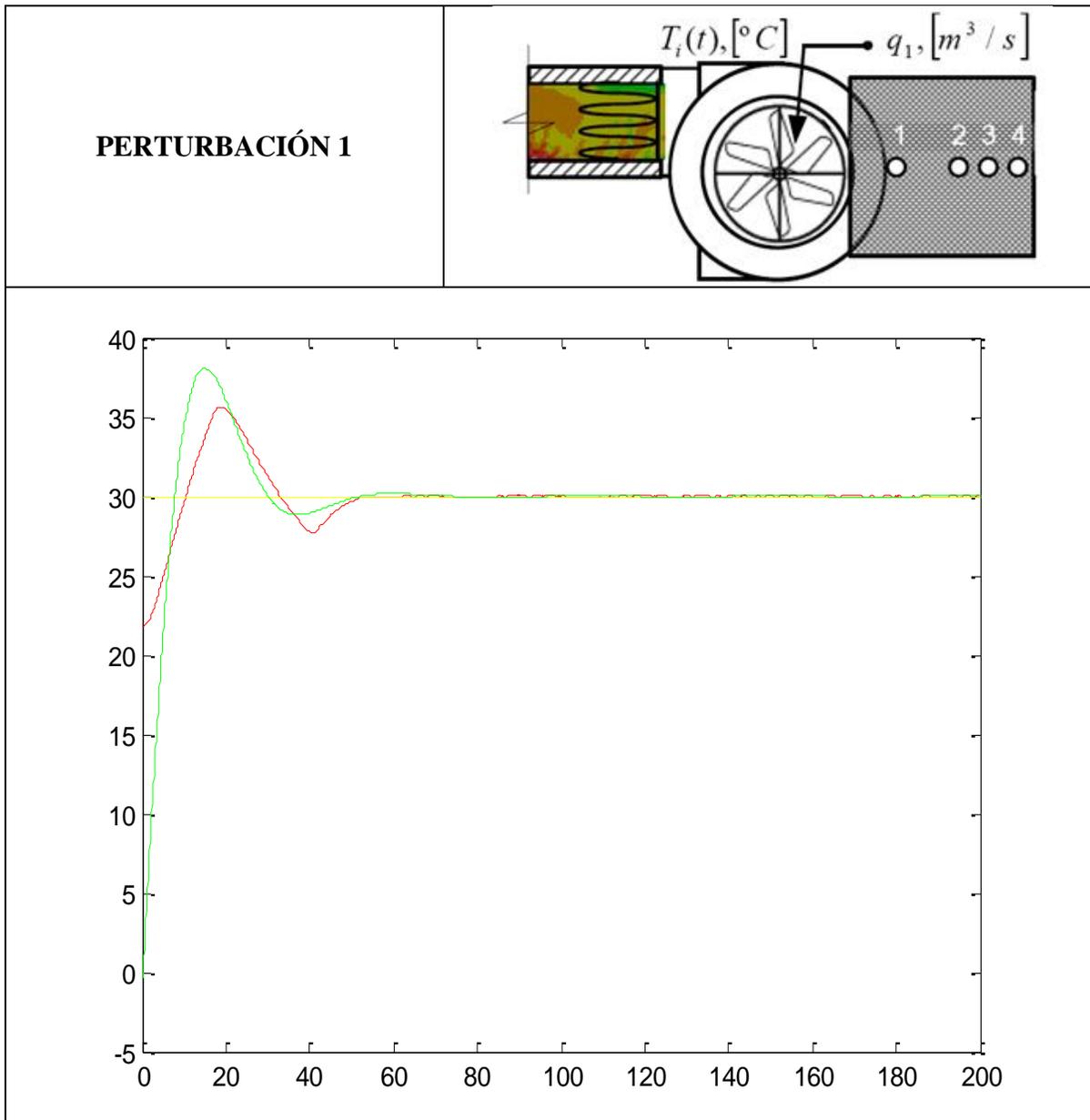


Figura 13 Comparación Del Resultado Obtenido En La Perturbación 1

ANEXO 3

**PROGRAMA PARA REALIZAR LA
IDENTIFICACIÓN DE LA PLANTA**

PROGRAMA PARA REALIZAR LA IDENTIFICACIÓN DE LA PLANTA**(“Modelamiento.m”)**

```
%Modelamiento de la Planta Air Flow Temperature Control System
```

```
%Byron Acuña
```

```
%Alexander Ibarra
```

```
%Cierra toda las ventanas de figuras.
```

```
clear all;%Borra todas las variables, funciones y links.
```

```
clc;%Limpia la ventana de comandos de Matlab
```

```
%Modelamiento discreto de la Planta
```

```
load ('c:\PID_RTW_0')
```

```
%Cargamos la respuesta de la Planta a un Impulso de 2 Voltios
```

```
plot(Scope_Identificacion_0.time,Scope_Identificacion_0.signals.values)
```

```
%Graficamos la respuesta al impulso de la planta
```

```
t=Scope_Identificacion_0.time;
```

```
%Guardamos el vector tiempo en t
```

```
y=Scope_Identificacion_0.signals.values(:,2);%Definimos el escalon Unitario.
```

```
Ts=0.01;%Definimos el Tiempo de Muestreo.
```

```
u=Scope_Identificacion_0.signals.values(:,1);%Definimos la respuesta de la Planta.
```

```
%Planta estimada por MATLAB (System Identification Toolbox)
```

```
data=iddata(y,u,Ts);
```

```
ident
```

```
pause
```

```
fprintf('Exporte las estimaciones desde el System Identification Tool y de Enter...');
```

```
pause;
```

```
% Estimación de primer orden simple
```

```
sys=tf(P1D);
```

```
Gp1=sys(1);
```

```
% n4s1
```

```
sys=tf(n4s2);
```

```
sys=d2c(sys);
```

```
Gpm=sys(1);
```

```
% arxqs
```

```
sys=tf(arxqs);
```

```
sys=d2c(sys);
```

```
Gpm2=sys(1);
```

```
Gp1
```

```
Gpm
```

```
Gpm2
```

```
% Gráficos
```

```
plot(t,y,'g');
```

```
hold on;
```

```
step(Gp1,max(t));
```

```
step(Gpm,max(t));
```

```
step(Gpm2,max(t));
```

```
grid;
```

ANEXO 4

**PROGRAMA PARA REALIZAR LA
COMPARACIÓN DE LAS GRAFICAS DE LOS
RESULTADOS OBTENIDOS**

**PROGRAMA PARA REALIZAR LA COMPARACIÓN DE LAS GRÁFICAS DE
LOS RESULTADOS OBTENIDOS.**

(“Comparacion.m”)

```
%ESCUELA POLITÉCNICA DEL EJÉRCITO
%BYRON ACUÑA
%ALEXANDER IBARRA
%PROGRAMA DE COMPARACIÓN ENTRE LOS ARCHIVOS DE SIMULACIÓN Y
EXPERIMENTACION
clear all%Borra todas las variables del Workspace
close all%Cierra todas las ventanas abiertas dentro de Matlab
clc%Limpia la ventana de comandos de Matlab
load ('C:\PID_RTW_0')%Carga el archivo del objeto de experimentacion
y=Scope_Experimentacion_0.signals(1,1).values(:,2)%Obtenemos los datos de la respuesta
del sensor de temperatura
t=Scope_Experimentacion_0.time(:,1)%Obtenemos el tiempo de simulacion en tiempo real
plot(t,y,'r')%Graficamos la señal de la planta controlada en tiempo real
hold on%Mantenemos la Grafica anteriormente graficada
clear all%Limpiamos las Variables del Workspace
load ('C:\simulacion')%Carga el archivo del modelo de simulacion
t=Scope_Simulacion(1,:)%'Obtenemos el tiempo de Simulacion.
u=Scope_Simulacion(2,:)%'Obtenemos la señal de referencia
plot(t,u,'y')%Graficamos la señal de referencia
y=Scope_Simulacion(3,:)%'Obtenemos la respuesta de la simulacion de la planta
controlada
plot(t,y,'g')%Graficamos la respuesta de la simulacion de la planta controlada.
```

ÍNDICE DE FIGURAS

Figura 2.1 Arquitectura RTW.....	5
Figura 2.2 Algoritmo RTW.....	6
Figura 2.3 Proceso RTW.....	7
Figura 2.4 Funcionamiento Del RTW.....	8
Figura 2.5 Configuración Genérica del RTW.....	8
Figura 2.6 Generación De Código Para Modelos En MATLAB®.....	9
Figura 2.7 Generación De Código Para Modelos En Simulink®.....	9
Figura 2.8 Estructura Del Código Del Modelo.....	11
Figura 2.9 Ejecución del Modelo en Tiempo Real.....	13
Figura 2.10 Real Time Windows Target Library.....	16
Figura 2.11 Tarjeta de Adquisición De Datos National Instruments PCI 6221.....	17
Figura 2.12 Conector SCB-68.....	21
Figura 2.13 Cable NI SHC68-68-EPM.....	22
Figura 2.14 Distribución de Pines De La Tarjeta de Adquisición de Datos NI PCI 6221.....	22
Figura 2.15 Proceso Térmico De La Unidad Air Flow Temperature Control System.....	23
Figura 2.16 Placa Metálica De La Unidad Air Flow Temperature Control System.....	24
Figura 3.1 Diagrama General De Componentes Del Sistema.....	25
Figura 3.2 Diagrama De Conexiones.....	27
Figura 3.3 Diagrama De Bloques Del Sistema De Control.....	28
Figura 3.4 Diagrama De Ejemplo.....	29
Figura 3.5 Propiedades del Real Time Windows Target Kernel.....	30
Figura 3.6 Confirmación de la Instalación del Real Time Kernel.....	32
Figura 3.7 Estado de la Instalación del Real Time Kernel.....	33
Figura 3.8 Instalación del Real Time Kernel.....	34
Figura 3.9 Primera Manera de Ingreso a SIMULINK®.....	35
Figura 3.10 Segunda Manera de Ingreso a SIMULINK®.....	35
Figura 3.11 Tercera Manera de Ingreso a SIMULINK®.....	35
Figura 3.12 Librería Real Time Windows Target.....	36

Figura 3.13 Bloque De La Señal De Salida.....	37
Figura 3.14 Bloque De Parámetros De La Entrada Análoga.....	38
Figura 3.15 Bloque De Parámetros De La Salida Análoga.....	39
Figura 3.16 National Instruments Support Web.....	41
Figura 3.17 National Instruments Auto Run.....	41
Figura 3.18 National Instruments Device Drivers.....	42
Figura 3.19 National Instruments Destination Directory.....	42
Figura 3.20 National Instruments Instalación Del Driver.....	43
Figura 3.21 National Instruments License Agreement.....	43
Figura 3.22 National Instruments Start Installation.....	44
Figura 3.23 Progreso De La Instalación.....	44
Figura 3.24 National Instruments Installation Complete.....	45
Figura 3.25 Ventana De Reinicialización Del Sistema.....	45
Figura 3.26 Icono De La National Instruments.....	46
Figura 3.27 Asistente Para Hardware Nuevo Encontrado.....	47
Figura 3.28 Finalización Del Asistente Para Hardware Nuevo Encontrado.....	47
Figura 3.29 Hardware Instalado Y Listo Para Usarse.....	48
Figura 3.30 Administrador De Dispositivos.....	49
Figura 3.31 Simulink® Library Browser.....	50
Figura 3.32 Block Parameters Analog Input.....	50
Figura 3.33 Selección De La Tarjeta De Adquisición De Datos.....	51
Figura 3.34 Ventana De Configuración De La Tarjeta De Adquisición De Datos PCI-6221.....	51
Figura 3.35 Mensaje de Prueba de Comunicación Satisfactoria con la Tarjeta de Adquisición de Datos.....	52
Figura 3.36 Ventana De Configuración del Solver.....	53
Figura 3.37 Archivos De Identificación Del Sistema.....	54
Figura 3.38 Panel Real Time Workshop.....	55
Figura 3.39 Panel De Implementación Del Hardware.....	56
Figura 3.40 Selección Del Compilador.....	57
Figura 3.41 Scope.....	58

Figura 3.42 Parámetros del Scope.....	58
Figura 3.43 Caja de Parámetros del Scope.....	59
Figura 3.44 Definición de Parámetros del Scope.....	60
Figura 3.45 Ingreso al External Mode Control Panel.....	61
Figura 3.46 Signal & Triggering.....	61
Figura 3.47 Configuración del External Signal & Triggering.....	63
Figura 3.48 Ingreso al Data Archiving.....	64
Figura 3.49 Configuración Del Data Archiving.....	65
Figura 3.50 Gráfica de Archivo Almacenado.....	67
Figura 3.51 Cambio De Modo De Simulación.....	67
Figura 3.52 Icono Connect To Target.....	68
Figura 3.53 Construcción Del Modelo En Tiempo Real.....	68
Figura 3.54 Ejecución Del Modelo En Tiempo Real.....	69
Figura 3.55 Comportamiento De La Ejecución Del Modelo En Tiempo Real.....	69
Figura 3.56 Gráfica Resultante Del Ejemplo De Simulación En Tiempo Real.....	70
Figura 3.57 Desinstalación del Real Time Kernel.....	71
Figura 3.58 Verificación de la Desinstalación del Real Time Kernel.....	72
Figura 4.1 Diagrama de Bloques del Sistema Air Flow Control Temperature System.....	78
Figura 4.2 Diagrama de Bloques del Sistema Air Flow Control Temperature System Considerando Que No Existe Variaciones En La Perturbación.....	79
Figura 4.3 Diagrama de Bloques del Sistema Air Flow Control Temperature System Simplificado.....	80
Figura 4.4 Diagrama De Transferencia De Calor Del Sistema.....	81
Figura 4.5 Respuesta De La Planta Al Impulso.....	83
Figura 4.6 Unidad Air Flow Temperature Control System Sin Perturbación.....	83
Figura 4.7 Gráfica De La Respuesta Del Sensor Tipo IC.....	84
Figura 4.8 Identificación De La Planta.....	85
Figura 4.9 Archivo De Datos Adquiridos.....	86
Figura 4.10 Configuración Del Bloque Step.....	86
Figura 4.11 Configuración Del Tiempo de Muestreo Del Analog Input.....	87
Figura 4.12 Configuración Del Tiempo De Duración De La Simulación.....	88

Figura 4.13 Configuración Del Número de Muestras.....	89
Figura 4.14 Variable ScapeData_0.....	90
Figura 4.15 Respuesta De La Planta A Un Step De 10 Voltios.....	91
Figura 4.16 Variable Para La Herramienta Ident.....	92
Figura 4.17 System Identification Tool (Ident).....	93
Figura 4.18 Carga Del Dato En La Herramienta Ident.....	94
Figura 4.19 Process Model.....	94
Figura 4.20 Model Output.....	95
Figura 4.21 Carga De La Aproximación En El Workspace.....	95
Figura 4.22 Oscilaciones Sostenidas De La Planta Kcr En Tiempo Real.....	98
Figura 4.23 Pantalla De Ingreso Al Sistema.....	100
Figura 4.24 Identificación De La Planta.....	101
Figura 4.25 Subsistema De La Planta Air Flow Temperature Control System.....	102
Figura 4.26 Subsistema Del Sensor Tipo IC.....	102
Figura 4.27 Control PID Experimental.....	103
Figura 4.28 Subsistema De La Planta Air Flow Temperature Control System.....	104
Figura 4.29 Subsistema Del Sensor Tipo IC.....	104
Figura 4.30 Simulación Del Control PID.....	105
Figura 4.31 Función De Transferencia De La Planta Con Retardo.....	106

ÍNDICE DE TABLAS

Tabla 2.1 Información General.....	18
Tabla 2.2 Entrada Analógica.....	18
Tabla 2.3 Salida Analógica.....	19
Tabla 2.4 Entrada / Salida Digital.....	19
Tabla 2.5 Contadores / Temporizadores.....	20
Tabla 2.6 Temporización/Disparo/Sincronización.....	20
Tabla 3.1 Características De La PC.....	26
Tabla 4.1 Cálculo Del Error En Estado Estable.....	73
Tabla 4.2 Datos De Respuesta Del Sensor Tipo IC.....	84
Tabla 4.3 Respuesta de la Unidad Air Flow Temperature Control System Ante Perturbaciones.....	97
Tabla 4.4 Regla Se Sintonización De Ziegler Nichols Segundo Método.....	99
Tabla 5.1 Pruebas En La Perturbación 1.....	108
Tabla 5.2 Pruebas En La Perturbación 2.....	109
Tabla 5.3 Pruebas En La Perturbación 3.....	110
Tabla 5.4 Pruebas En La Perturbación 4.....	111
Tabla 5.5 Resultados En La Perturbación 1.....	112
Tabla 5.6 Resultados En La Perturbación 2.....	113
Tabla 5.7 Resultados En La Perturbación 3.....	114
Tabla 5.8 Resultados En La Perturbación 4.....	115

FECHA DE ENTREGA:

AUTORES

Byron Alejandro Acuña Acurio
171868881-3

Oswaldo Alexander Ibarra Jácome
171953542-7

AUTORIDAD

Ing. Víctor Proaño

**DIRECTOR DE LA CARRERA EN INGENIERÍA ELECTRÓNICA
AUTOMATIZACIÓN Y CONTROL**