

# 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

Byron Acuña A, Alexander Ibarra J, Víctor Proaño R.

**Resumen--** El presente proyecto “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”, abarca el estudio de la herramienta RTW en donde se analizó la arquitectura, el algoritmo, el proceso y las etapas en el desarrollo de un modelo en tiempo real por medio del sistema SIMULINK® de MATLAB®, además se realizó una HMI la cuál mediante subsistemas .mdl desarrollados en SIMULINK® permitieron la identificación de la planta, el control PID experimental y la simulación del control PID, utilizando para esto la tarjeta de adquisición y generación de datos National Instruments PCI 6221, la misma que, luego de haberla configurado permitió interconectar la planta real con los subsistemas de la HMI desarrollada.

**Palabras-Clave--** RTW, RTWT, tiempo real, PID, PCI, HMI, Solver, Ziegler Nichols, ESPE.

## I. INTRODUCCIÓN

En el aprendizaje 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 lo realizaba externamente y siempre existían variaciones en los resultados obtenidos.

Con 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 producido en la planta debido a cambios en los parámetros

---

Byron Acuña A, Alexander Ibarra J, Víctor Proaño R.  
Departamento de Eléctrica y Electrónica, ESPE, Sangolquí, Ecuador.  
E-mails: oaibarra@espe.edu.ec, byronacuania@hotmail.com, voproano@espe.edu.ec

del controlador ampliando notablemente la interacción del estudiante y el aprendizaje de sistemas de control.

Se consideró oportuno desarrollar un sistema computarizado en el cual se pueda realizar pruebas en tiempo real con los controladores diseñados.

El sistema propuesto permite diseñar el controlador en el entorno de SIMULINK® pero se ejecuta en una planta real específicamente en la unidad AIR FLOW TEMPERATURE CONTROL SYSTEM.

Un sistema como el indicado es de mucha utilidad por cuanto facilita el diseño de controladores de todo tipo como control difuso, control óptimo, control digital, etc. aprovechando las capacidades de MATLAB® aplicándolo en forma rápida a la planta.

## II. HERRAMIENTA RTW

El Real Time Workshop (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 se haya 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 se aprecia en la Figura 1. [1]

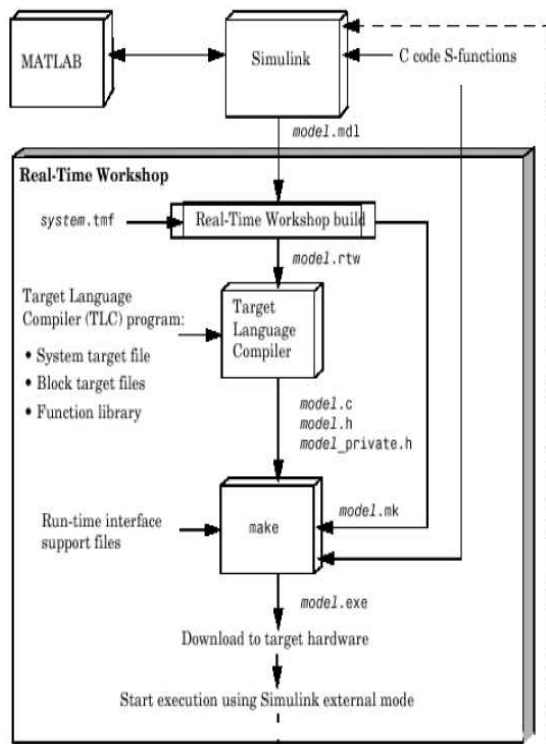


Fig. 1. Arquitectura RTW

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, tal como se lo puede apreciar en la Figura 2. Las etapas en la generación RTW son de la siguiente manera:

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 construcción 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 dota 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.

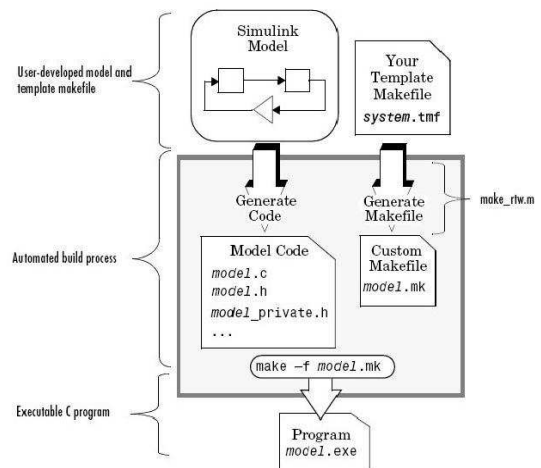


Fig. 2. Funcionamiento del RTW

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 3 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 da una percepción de trabajar sin retardos.

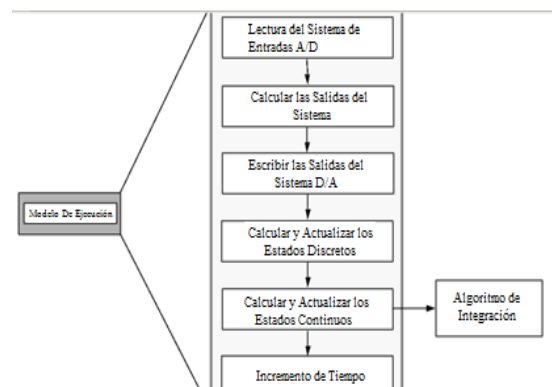


Fig. 3. Ejecución del modelo en tiempo real

### III. HERRAMIENTAS APLICADAS

Las herramientas utilizadas y analizadas de forma específica en el estudio, análisis e implementación del 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® fueron:

- Simulink®.- Software para modelado, simulación y análisis de sistemas dinámicos. Es compatible con sistemas lineales y no lineales, modelados en tiempo

continuo, tiempo discreto, tiempo real o híbrido.

- Real Time Workshop.- 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.
- Real Time Windows Target.- 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 la cual permite operar con señales de entrada y/o salidas análogas y digitales.

#### IV. DESARROLLO

- Se instaló la tarjeta de adquisición de datos PCI 6221. [2]
- Se instaló Visual Estudio 2005.
- Se instaló y configuró el Real Time Windows Target.
- Se reconoció la tarjeta de adquisición de datos PCI 6221 en Simulink®.
- Se configuró los parámetros del Real Time Workshop.
- Se configuró las opciones de los parámetros de simulación en SIMULINK® (Solver).
- Se configuró las opciones de los elementos de visualización.
- Se desarrollo el modelo en tiempo real.
- Se realizaron las conexiones de hardware necesarias.
- Se realizó la identificación de la planta.
- Se realizó el diseño del controlador PID mediante el segundo método de Ziegler Nichols en tiempo real.
- Se realizaron pruebas.
- Se obtuvieron y analizaron los resultados.

En tal virtud se realizó la siguiente HMI en Simulink® (Figura 4) basada en el diagrama de conexiones que se presenta en la Figura 5.



Fig. 4. Pantalla De Ingreso Al Sistema.

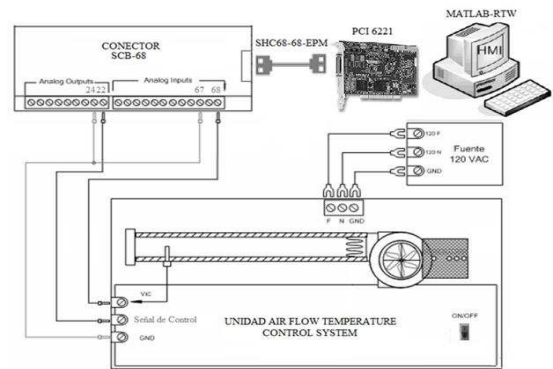


Fig. 5. Diagrama De Conexiones[5]

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.



Fig. 6. Identificación De La Planta

Mediante la utilización de subsistemas en la herramienta Simulink® [3] se procedió a la implementación de un subsistema para obtener la respuesta al escalón de la planta (Figura 7), además se implementó también la ecuación de respuesta del sensor para lograr convertir las unidades crudas en unidades de ingeniería (Figura 8).

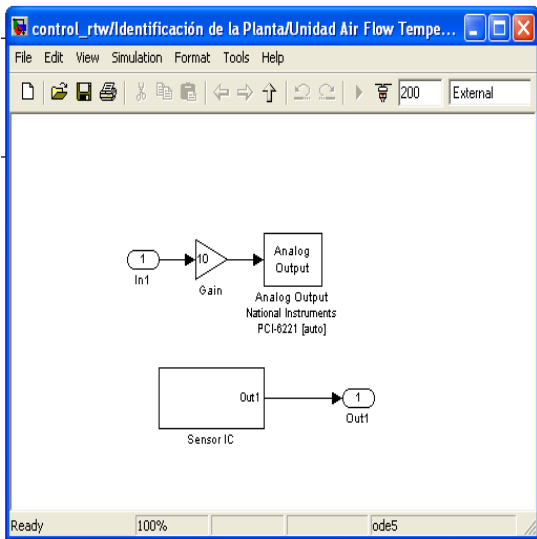


Fig. 7. Subsistema de respuesta al escalón

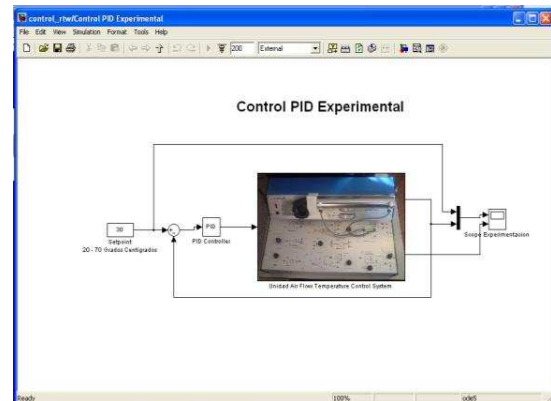


Fig. 9. Control PID Experimental

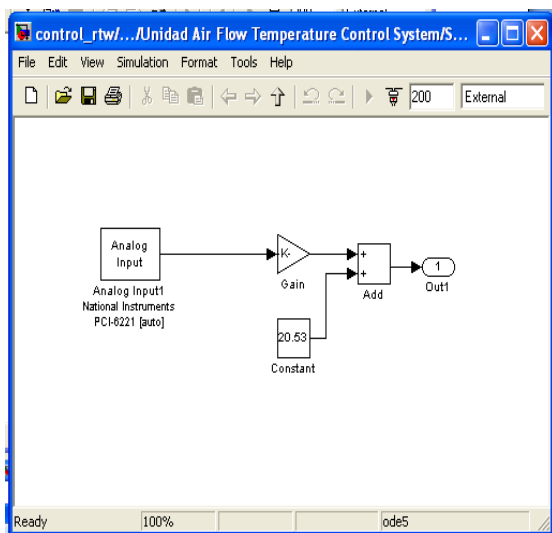


Fig. 8. Subsistema del Sensor

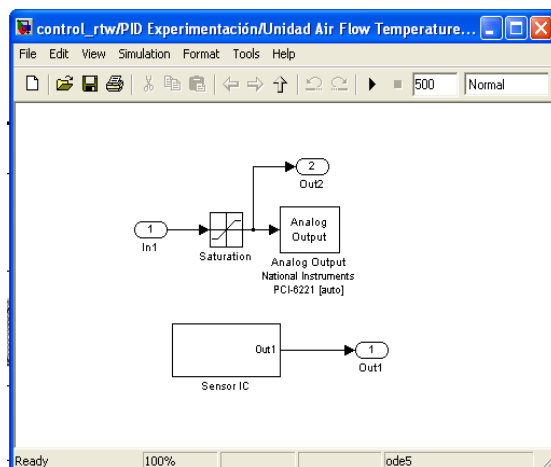


Fig. 10. Subsistema para adquisición de datos

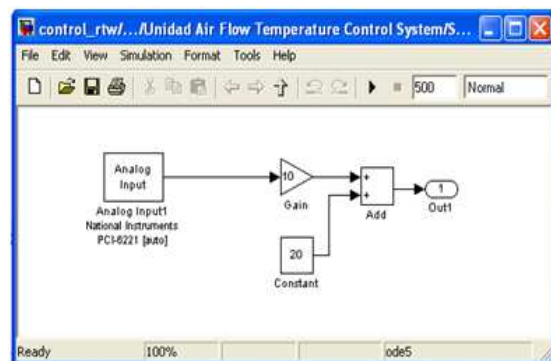


Fig. 11. Subsistema del Sensor Tipo IC

El Control PID experimental (Figura 9) se lo realizó mediante la utilización de subsistemas en la herramienta Simulink®, en los cuales se procedió a la implementación del controlador PID [4] en donde se utilizaron los parámetros  $K_p$ ,  $K_i$  y  $K_d$  (Figura 10), además se implementó también la ecuación de respuesta del sensor para lograr convertir las unidades crudas en unidades de ingeniería (Figura 11).

Para la realización de la Simulación del Control PID (Figura 12), se crearon cuatro subsistemas en donde se implementó las funciones de transferencia obtenidas con las cuatro diferentes perturbaciones (Figura 13) y un controlador con los valores de  $K_p$ ,  $K_i$  y  $K_d$  obtenidos por el segundo método de Ziegler – Nichols.

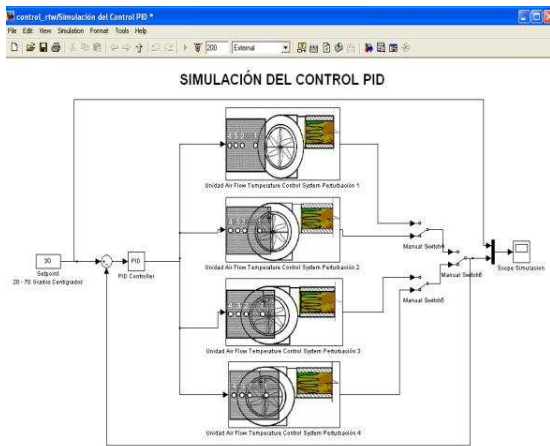


Fig. 12. Modelo De Simulación

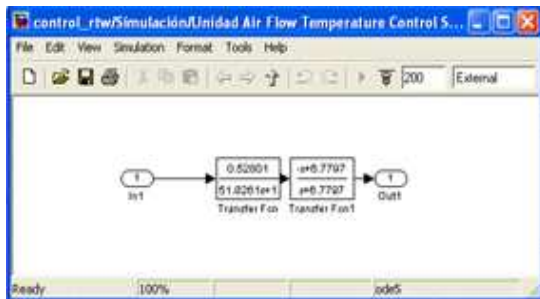


Fig. 13. Función de transferencia Perturbación 1

En donde, en el bloque PID utilizaremos los siguientes valores  $K_i = 0$ ,  $K_d = 0$  y  $K_p = K_{cr}$  obteniendo la Figura 14 que se presenta a continuación [6].

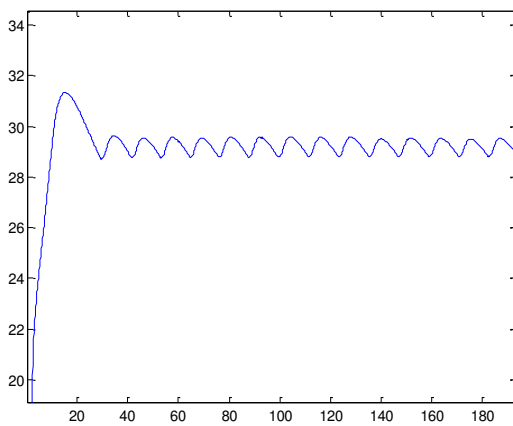


Fig. 14. Oscilaciones Sostenidas De La Planta Kcr En Tiempo Real

De la Figura 14 el  $K_{cr}$  encontrado es 34, analizando los datos entre crestas se encuentra el  $P_{cr}$  que es 12 seg, empleando el segundo método de Zeigler Nichols el cuál establece que:

Tabla 1 Regla Se Sintonización De Ziegler Nichols Segundo Método

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

Se obtuvieron los parámetros:

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

Los valores de  $K_p$ ,  $K_d$  y  $K_i$  anteriormente detallados son utilizados en el bloque PID del control PID experimental y en la simulación del control PID, obteniendo como resultado el control PID en tiempo real de la planta física y su simulación correspondientemente, consiguiendo los resultados que se aprecian en la Figura 15 que se presenta a continuación.

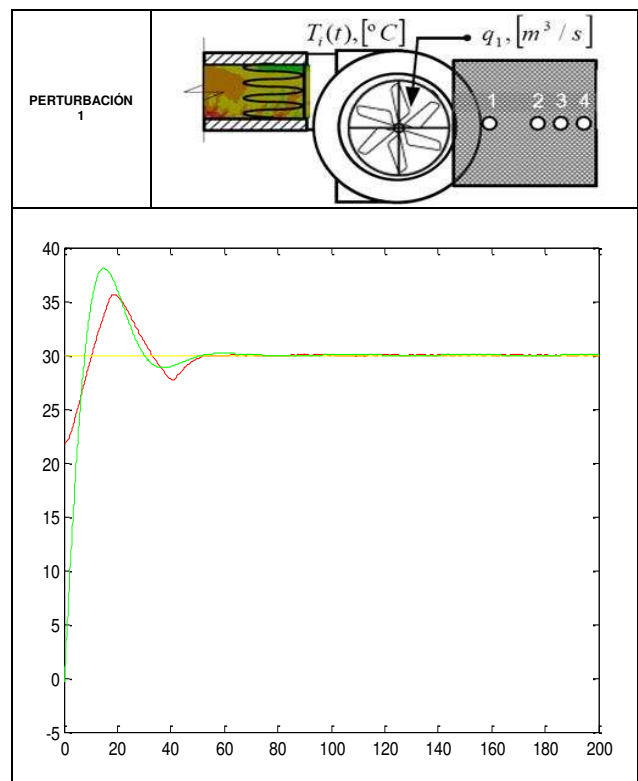


Fig. 15. Comparación Del Resultado Obtenido En La Perturbación 1

## V. ANÁLISIS DE RESULTADOS

Como se puede apreciar en las gráficas de la Figura 15 del control PID experimental y de la simulación del control PID guardan cierta relación en los parámetros de importancia como son el Máximo Pico (Mp), el error en estado estacionario (ess) y el tiempo de estabilización (ts), las variaciones que se pueden notar son el tiempo de la respuesta transitoria, puesto que como se observa depende 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 el 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 obtenidas con los mismos valores del controlador PID y con el mismo tiempo de muestreo dando como resultado una aproximación entre las gráficas altamente aceptable.

## VI. CONCLUSIONES

- 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 gracias a 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 lenta como temperatura e 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.
- 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.

## VII. 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.
- 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 la herramienta RTWT 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.
- 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

- [1] <http://iaci.unq.edu.ar/materias/laboratorio2/transparencias%5CReal%20Time%20Workshop.pdf>
- [2] <http://digital.ni.com/manuals.nsf/websearch/8117DF4C5A29C95C862573020061023B>
- [3] MATLAB & Simulink Based Books
- [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] OGATA, Katsuhiko, Ingeniería de control Moderna, 1993.