

Diseño e implementación de un emulador de plantas SISO analógicas

Santana Cobo José David*

Facultada de Ingeniería Electrónica, Escuela Politécnica del Ejército
Av. El Progreso S/N, Sangolquí, Ecuador

Resumen: El presente artículo describe la implementación de un hardware emulador de plantas industriales analógicas de una entrada y una salida (SISO) basándose en un controlador digital de señales (DSC) que integra la adquisición de datos, el procesamiento digital de los mismos y la interfase de comunicaciones. El equipo es autónomo en la simulación del proceso y la configuración se realiza por medio de una interfase de usuario radicada en un PC. Este equipo está orientado al ámbito académico con el fin de complementar ó reemplazar el uso de plantas a escala a un precio económico, permitiendo simular sistemas industriales como hornos, tanques, motores, sistemas de flujo, entre otros de tal manera que los estudiantes técnicos sean capaces de practicar y entrenarse en los métodos de control de estos sistemas industriales.

Introducción

A lo largo de los años los educadores han reconocido la necesidad e importancia de realizar prácticas de control en laboratorio con el fin de reforzar los conocimientos teóricos adquiridos en clase respecto al diseño de controladores, identificación de plantas y aplicación de conceptos de control industrial. La realización de prácticas en laboratorio ayuda a los estudiantes a relacionar los conceptos teóricos con aplicaciones reales de tal forma que tengan una mayor motivación en el aprendizaje de técnicas de control y permita tomar en cuenta otros aspectos técnicos involucrados en la implementación de lazos de control (conexiones eléctricas, perturbaciones, manejo de software comercial, etc.).

La motivación original detrás de la construcción de modelos a escala de procesos industriales para laboratorios es la de ofrecer a los estudiantes un sistema completo para el diseño de ingeniería, empezando por el modelamiento y simulación, hasta terminar con la validación experimental de un sinnúmero de estrategias de control.

La gran utilidad que se puede dar a estos equipos se ve muchas veces limitada por el costo de los mismos y el espacio que pueden llegar a ocupar, forzando su utilización a prácticas con pocos estudiantes. Adicionalmente cada módulo representa un proceso en particular, lo que los vuelve inflexibles si se quiere variar parámetros del proceso o involucra la necesidad de contar con un modelo a escala distinto dependiendo del proceso que se desea controlar [1].

Ante los problemas que tienen los modelos a escala, en lo últimos años se ha visto una mayor tendencia a desarrollar modelos virtuales que se ejecutan en tiempo real ya sea en un computador personal (PC) o en módulos especializados basados en procesadores digitales de señales (DSP) ó incluso microcontroladores. A este hardware se le equipa adicionalmente con la electrónica necesaria que contiene conversores A/D, D/A, entradas y salidas digitales, sistemas de comunicación, etc. La idea de reemplazar los modelos a escala (o incluso la planta real) con un equipo basado en software se denomina Hardware en el Lazo ó HIL, por sus siglas en inglés.

Filosofía HIL

Un montaje típico para una práctica de control (Fig. 1a) consiste en un controlador (PLC, prototipo de controlador, etc.) conectado a un computador personal para monitorear y configurar el controlador y una planta (modelo a escala de un proceso específico). Las entradas a la planta son por lo general actuadores eléctricos: motores, válvulas, amplificadores, etc., mientras que las salidas son sensores que trasladan los parámetros de la planta (temperatura, presión, velocidad) a variables eléctricas analógicas o digitales.

Un simulador HIL (Fig. 1b) consiste en reemplazar la planta o proceso por un hardware que simule en tiempo real el modelo de la planta, los sensores y actuadores. Una simulación en tiempo real involucra que la simulación sea efectuada de tal manera que las señales de entrada y de salida muestren los

* jdsantana@yahoo.com

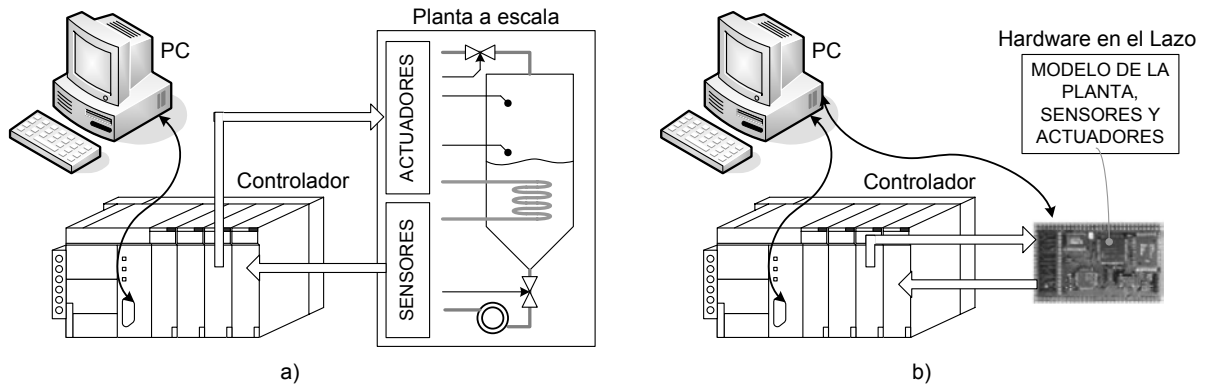


Figura 1. Montaje típico para práctica de control. (a) método clásico y (b) método HIL.

mismos valores dependientes en el tiempo que su contraparte real [2].

Esto significa que la simulación debe tener un comportamiento tal que el sistema de control no note la diferencia entre la planta real y el simulador HIL. Es por esta razón que un ciclo del simulador HIL debe ser algunas veces más rápido que el ciclo de control del sistema del controlador [3]; esto determina la potencia computacional que debe tener un simulador HIL dada la complejidad del modelo asignado a la planta y el método de simulación que se utilice.

Periféricos de interfase con el usuario - permiten configurar y modificar al sistema que se desea simular. Una forma de interfase son controles directos como botones, perillas o teclados mientras que otra manera es usando un canal de comunicación (RS232, LAN, etc.) que, enlazado por un programa de interfaz de usuario, modifica los parámetros de la planta a través de una PC.

Existen múltiples maneras de realizar un simulador HIL pero en general se dividen en dos grupos: sistemas basados en PC y sistemas embebidos. En el primer grupo se utiliza el hardware común y barato de una computadora personal la cual es equipada con tarjetas de adquisición de datos. Este método es uno de los más usados por su economía y poder de cómputo. El modelo corre generalmente sobre un sistema operativo en tiempo real (RTOS) proporcionado por el fabricante del producto. En sistemas más sofisticados se utiliza co-simulación que involucra el uso de varios procesadores para la ejecución de los modelos.

El segundo grupo pertenece a sistemas embebidos y se relaciona con todos los equipos que son capaces de funcionar independientemente. Su uso como simuladores HIL surge por la necesidad de contar con características especiales no incluidas en una PC, pueden ser más baratos y usados para realizar pruebas de la tecnología; muchos de los fabricantes de DSPs (Procesadores Digitales de Señal), procesadores, microcontroladores y FPGA proporcionan tarjetas de evaluación de sus productos, lo que los hace ampliamente disponibles.

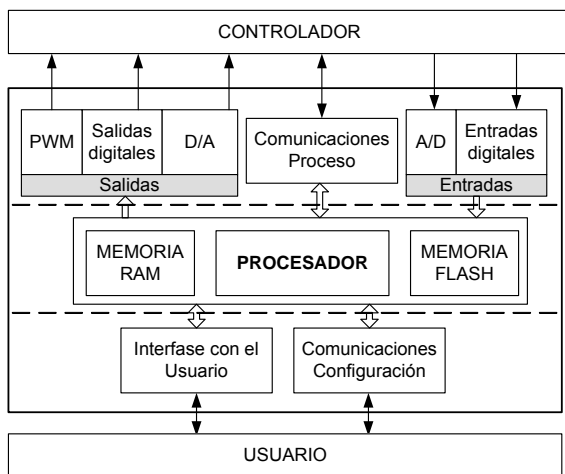


Figura 2. Componentes de un sistema HIL

En general un simulador HIL tendrá (Fig. 2):

Procesador y memoria - en la memoria (temporal o permanente) se guarda el código que representa el modelo de la planta, mientras que el procesador ejecuta los cálculos del código contenido.

Periféricos de interfase con el controlador - constituye la interfase eléctrica necesaria de las señales de entrada y salida para interactuar con el controlador. Estas interfasas pueden ser convertidores A/D, D/A, entradas y salidas digitales discretas, encoders de pulsos e interfasas de comunicación como RS232, LAN, CAN, LIN, etc.

Hardware del emulador

El objetivo principal de este proyecto es emular un sistema analógico de una entrada y una salida (sistemas SISO). Esto se debe a que en la mayoría de casos las prácticas de control se limitan a estos sistemas.

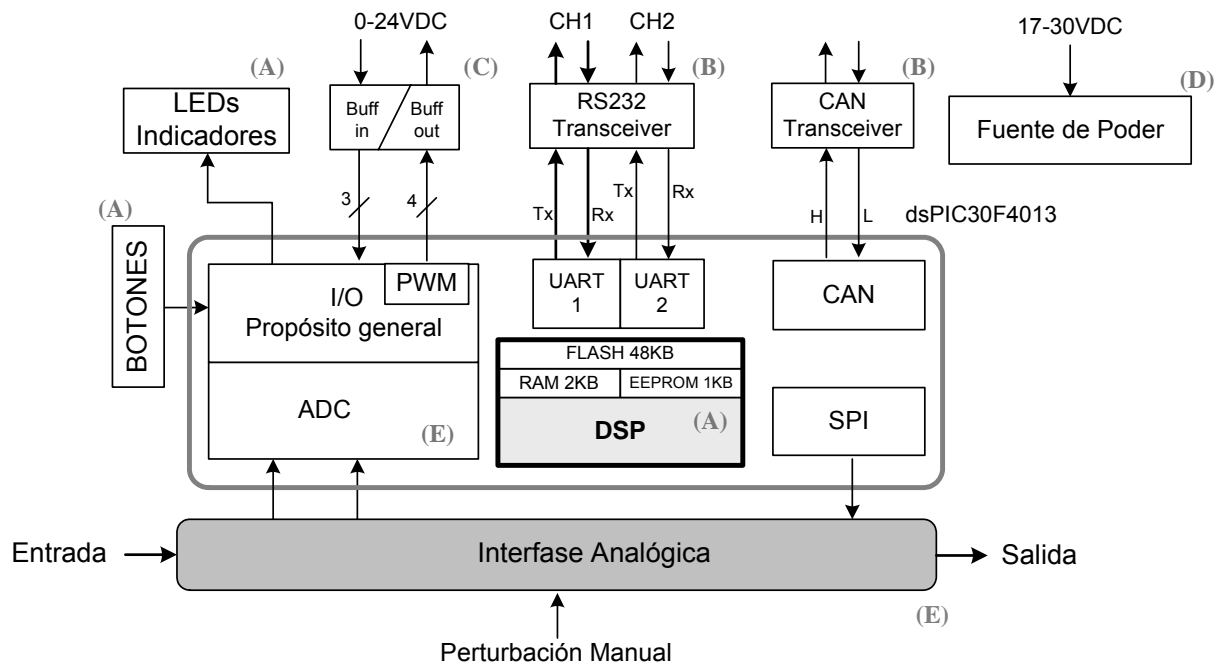


Figura 3. Diagrama general del Hardware del *emulador SISO*

De manera general el circuito es un sistema de adquisición de datos de 12 bits con el fin de tener un sistema embebido capaz de procesar digitalmente señales de baja frecuencia (DC – 1kHz) típicas de control. Los rangos de funcionamiento en señales analógicas son unipolares de 0 a 12 V. Cuenta adicionalmente con Entradas/ Salidas digitales con el fin de complementar la funcionalidad y tener la posibilidad de generación PWM. Los rangos de voltaje digitales son flexibles, de 0 a 24 V.

Desde un principio se estableció que el emulador debería tener la mayor cantidad de funciones relacionadas al *emulador SISO* incorporadas en un solo chip, esto es, que al menos se encuentre incorporado el procesador, la memoria RAM y Flash, comunicaciones y salidas/entradas digitales.

La línea de productos denominados controladores digitales de señales (DSC), incorporan la robustez e integración de un microcontrolador con la capacidad computacional de un DSP. Para la selección se tomaron en cuenta no solo aspectos de potencia computacional sino de facilidad en conseguir los elementos, forma de programarlos, disponibilidad de herramientas de desarrollo y empaques DIP.

Para el *emulador SISO* se seleccionó el dsPIC30F4013 de Microchip [4], [5] el cual, entre otras funciones, integra:

- un procesador RISC de 16 bits de alto desempeño con arquitectura Harvard modificada con capacidad de DSP;

- rendimiento de hasta 30 millones de operaciones por segundo (MIPS);
- 48KB de memoria FLASH programable en campo, 2KB RAM y 1KB de EEPROM;
- convertor A/D a 100ksps, hasta 13 canales multiplexados.
- comunicaciones UART, CAN, SPI, I²C;
- comparadores de entrada y salida (PWM);
- 5 temporizadores de 16 bits;
- oscilador integrado.

Adicional al dsPIC30F4013 se incluye la circuitería necesaria para el acondicionamiento y manejo de las señales analógicas, los drivers de las entradas/salidas digitales, la interfase de comunicaciones y la fuente de poder. Todos estos elementos van integrados en un solo circuito impreso.

El sistema se divide en cinco bloques componentes básicos (Ver Fig. 3 y Fig. 4):

(A) *Controlador e Interfase* – El controlador es el dsPIC30F4013, encargado de la adquisición de datos analógicos a través de su ADC interno y las entradas/salidas digitales. Realiza todos los cálculos del modelo de la planta y el manejo respectivo de los distintos canales de comunicación. Se tiene una interfase básica por medio de tres botones de uso general más un pulsador de reset. Tres indicadores luminosos indican el estado de simulación y condiciones de saturación en la entrada y salida analógicas.

(B) *Comunicaciones* – Se tiene tres tipos de comunicaciones: USART, CAN y SPI. El DSC

tiene dos canales independientes USART los cuales son compatibilizados con RS232 por medio del transceiver MAX203. Uno de los canales sirve para la carga/descarga de la configuración al emulador. La comunicación SPI se utiliza para enlazarse con los 3 potenciómetros digitales y el DAC. Se implementó comunicación CAN usando el transceiver MCP2551 para futuras aplicaciones.

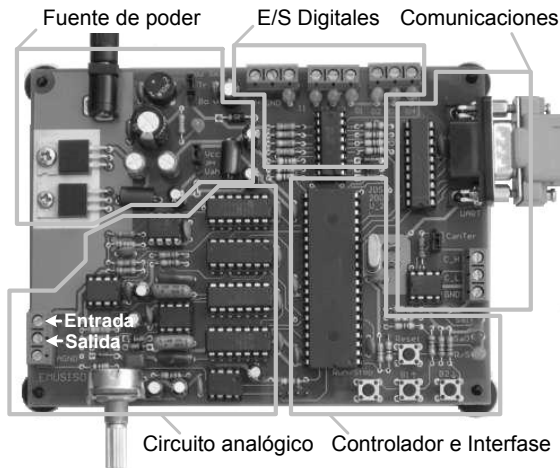


Figura 4. Foto del hardware del emulador SISO

(C) *Entradas/Salidas Digitales* - Se tiene 3 entradas y 4 salidas digitales que se centran en el driver DS2003 más los componentes auxiliares e indicadores visuales que permiten ver el estado de las E/S. Soportan voltajes de 0 a 24V y se fija el rango por medio de polarización externa. Las cuatro salidas pueden configurarse como señales de PWM.

(D) *Fuente de Poder* - El diseño es puramente lineal debido a que no genera ruido, consideración importante para tener un buen sistema de adquisición de datos. Se generan tres voltajes de placa, uno de +15V para los operacionales de la etapa analógica, +5V para los elementos mixtos análogo-digitales y +5V digitales. Se tomó especial cuidado en la ubicación correcta de los elementos y planos separados de tierra analógica y digital.

(E) *Circuito analógico* - Representa la parte más compleja e importante del diseño y su

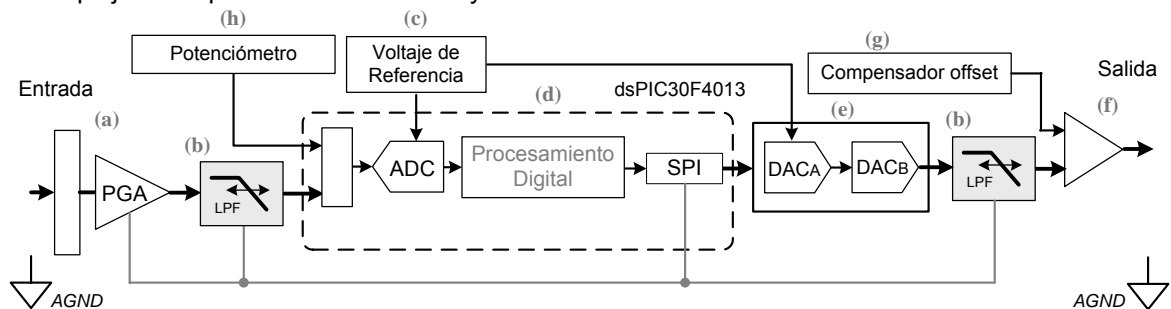


Figura 5. Diagrama funcional del circuito analógico

diagrama funcional se puede ver en la Fig. 5. Las funciones de cada uno se describen a continuación:

- Buffer de entrada analógico* - Tiene las protecciones contra sobrevoltajes y voltajes negativos. Adicionalmente posee un Amplificador de Ganancia Programable (PGA) para adaptar los rangos de voltaje de la entrada (0-12) a los 4.096 que maneja el ADC.
- Filtros antialiasing* (Fig. 6) - Estos filtros suprimen las frecuencias altas que pueden producir aliasing [6] tanto a la entrada como a la salida. Cada uno es de segundo orden implementados con amplificadores operacionales y potenciómetros digitales [7] que vía SPI varían su frecuencia de corte entre 15 y 400 Hz al variar su resistencia.

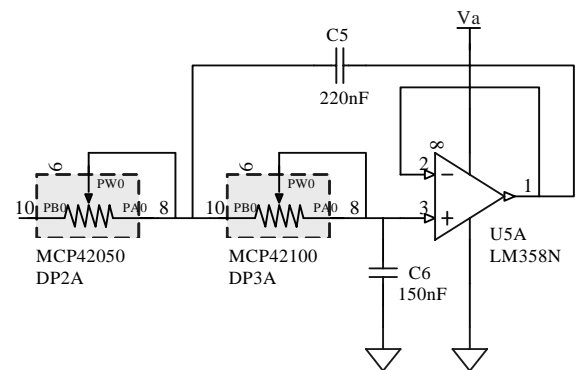


Figura 6. Filtro antialiasing de entrada y salida

- Referencia de voltaje* - Genera un voltaje de 4.096V estable en temperatura y tiempo tanto para el ADC como para el DAC.
- Convertor Analógico - Digital* - Incorporado en el DSC, muestrea alternadamente el potenciómetro y la señal de entrada ya acondicionada.
- Convertor Digital - Analógico* - Compuesto por dos DACs que se programan mediante el canal SPI; el primero (DAC_A) sirve para calibrar el rango de voltajes de salida a generar el voltaje de referencia del DAC_B encargado de convertir los valores calculados en el procesamiento digital.

- f) *Buffer de salida analógico* - Amplifica el voltaje de la etapa anterior y proporciona una salida de baja impedancia.
- g) *Compensación de offset* - Permite compensar el valor de offset a la salida. Se regula por medio de un potenciómetro digital.
- h) *Potenciómetro mecánico*- Es usado como un elemento de interfase con el usuario. Permite variar un parámetro de software dependiendo del valor analógico que genere.

La utilización de potenciómetros digitales posibilita compensar los errores que se tengan en la etapa analógica por la variabilidad de los componentes pasivos. Tras un proceso de calibración se almacenan los datos de corrección en la memoria EEPROM del DSC a través de una aplicación especial integrada a la GUI.

Modelo de la planta

Los sistemas comerciales generan el código que representa el modelo de la planta a partir de un diagrama de flujo, p.e. realizado en Simulink de MATLAB, y lo ejecutan en el procesador de destino [8]. Si bien esta aproximación es la más óptima y adecuada, el desarrollo de la infraestructura de software para soportar un nuevo procesador resulta complejo y costoso.

Para este proyecto se reduce la flexibilidad de modelado por la sencillez en el desarrollo al utilizar una plantilla que, para el *emulador SISO*, corresponde al flujo que se muestra en la Fig. 7. Por medio de una Interfaz Gráfica de Usuario (GUI) se configuran los parámetros de la plantilla y se descargan los datos de la misma a la memoria EEPROM del *emulador SISO* usando el puerto serial de la PC. Complementariamente se cuenta con el software embebido en el DSC que corre el modelo de la planta conforme a los valores almacenados en la memoria.

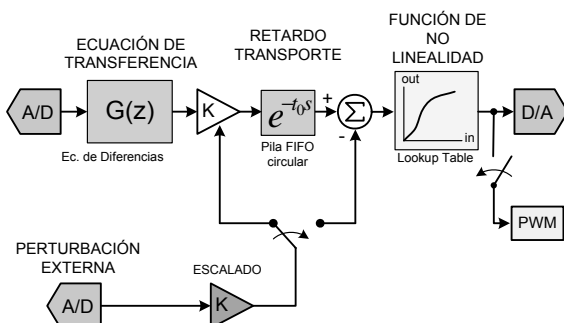


Figura 7. Elementos del modelo de la planta.

A. Función de transferencia

La mayoría de plantas y sistemas pueden ser descritos por medio de ecuaciones diferenciales lineales e invariantes en el tiempo (LTI). Por medio de la transformada de Laplace se expresan estas ecuaciones en términos algebraicos de la variable s , reduciendo el comportamiento del sistema a una función de transferencia ($G(s)$) [9]. Es posible encontrar la ecuación de diferencias correspondiente a la función de transferencia continua para un tiempo de muestreo dado [10]. La ecuación de diferencias corresponde a un filtro IIR de fácil implementación a nivel digital. En el caso del *emulador SISO* los coeficientes del filtro IIR se obtienen en la GUI y son enviados al DSC donde se realizan los cálculos en punto flotante de 64 bits para evitar acumulación de errores. Se trabaja con funciones de hasta 4^{to} orden, aunque en un futuro es posible incrementarlo con ciertas modificaciones en el algoritmo.

B. Función de retardo de transporte

El retardo de transporte ó tiempo muerto es un fenómeno muy común en procesos industriales ya que está asociado al traslado de materiales desde un punto a otro, demoras en el cómputo de partes del proceso, velocidad de difusión de sustancias, entre otras.

El método más práctico es el usar un sistema de grabación [11], en el cual se registra el valor y se lo reproduce después de un tiempo (según el retardo). En el emulador se utiliza una pila circular radicada en la memoria RAM. Dependiendo de la cantidad de memoria y la frecuencia a la que se trabaja se tendrá el máximo retardo disponible.

C. Función de no linealidad

Este trabajo solo incorpora no linealidades estáticas invariantes en el tiempo. Estas son generadas en software con la ayuda de una tabla de búsqueda en la cual se genera la salida no lineal al relacionar la entrada con una curva (tabla) que representa el fenómeno como saturación, zona muerta, fricción de Coulumb o cualquier otra curva arbitraria.

D. Función de perturbación externa

Por medio de la variación del potenciómetro incorporado al emulador se simulan dos tipos de perturbaciones:

- aditiva al flujo principal de la señal y
- de ganancia a la función de transferencia

De esta manera se cambian las condiciones del sistema con el fin de comprobar el desempeño del controlador.

Elementos de Software

Junto con el modelo de la planta se tiene el software de integración embebido en el DSC y el programa de la GUI.

A. Interfaz gráfica de usuario

Este componente es la interfase principal con el usuario y está desarrollada completamente en MATLAB 7 debido a la amplia disponibilidad de herramientas matemáticas que presenta y la gran difusión que tiene en medios educativos y científicos. El programa desarrollado cumple con las siguientes funciones:

- Permite una utilización fácil e intuitiva, con las protecciones, mensajes, advertencias y guías gráficas necesarias para un correcto ingreso de datos del modelo de la planta y los parámetros de calibración del hardware emulador.
- Gestiona las comunicaciones por el puerto serial con el *emulador SISO*.
- La GUI realiza las correcciones de los valores basándose en los parámetros de calibración recuperados de la memoria del DSC.
- Los datos generados por la GUI son formateados de manera adecuada para descargarlos a la memoria del DSC por el puerto serial y recuperarlos del mismo. Estos datos se pueden guardar en cualquier medio magnético.

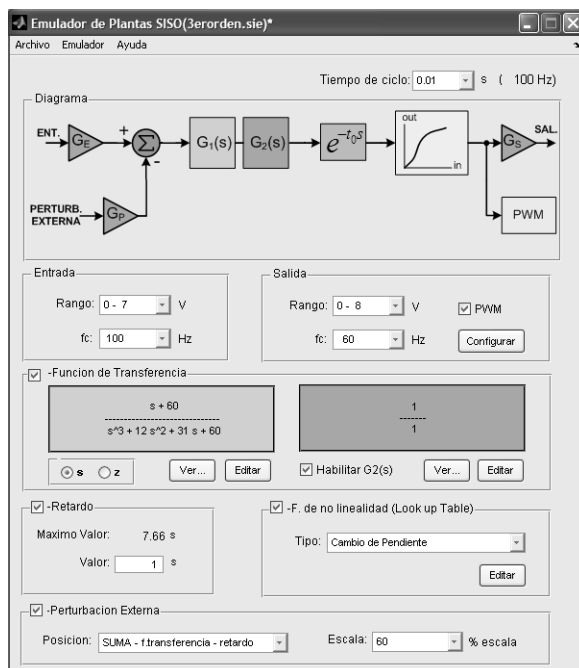


Figura 8. Ventana principal de la GUI

En la Fig. 8 se muestra la ventana principal de la GUI, destacando que cada bloque permite su activación y respectiva configuración. Por

razones didácticas se deja la opción de variar las frecuencias de corte de los filtros antialias y el tiempo de ciclo en la simulación.

B. Programa Embebido

Una de las razones de más peso al seleccionar el dsPIC30F4013 fue la disponibilidad de herramientas que el fabricante Microchip pone al alcance para el desarrollo de programas. Entre estas se encuentra un entorno gratuito que integra el compilador en ensamblador, el simulador y administrador de proyectos. De la misma manera se proporciona una versión estudiantil del compilador en C sin mayores restricciones. El código se desarrolló en su mayoría en ANSI C.

El *primer* componente que se instala en el DSC dsPIC30F4013 es un cargador de arranque (*bootloader*) [12] el cual permite descargar el código máquina a la memoria Flash y EEPROM del DSC por medio del puerto serial. Esto evita el tener que sacar el chip hacia un grabador cada vez que se desea cambiar el programa.

El *segundo* componente de SW son todas aquellas rutinas que constituyen los drivers específicos para el hardware desarrollado. En estas se incluyen las funciones de manejo de los potenciómetros digitales via SPI, manejo del ADC y del DAC, control de los temporizadores y el manejo básico de los periféricos de comunicaciones. Se tiene también la administración de la memoria EEPROM. Ciertas de estas rutinas son en ensamblador para optimizar el rendimiento.

El *tercer* componente es el código que realiza los cálculos del modelo de la planta y el conjunto de otras rutinas de control de la aplicación.

Conforme la complejidad de la aplicación se va incrementando resulta más difícil mantener ordenado el código y retener la lógica de todo el programa. Por esta razón se desarrolló la aplicación usando un Sistema Operativo en Tiempo Real (RTOS): el CMX-Scheduler [13]. Esta estrategia de programación permite dividir la aplicación en tareas y simplifica significativamente el código al ceder el arbitraje de la ejecución de los distintos componentes al núcleo del RTOS. A cada una de las tareas se le asigna una prioridad de ejecución, lo que permite controlar precisamente cuándo y cómo se ejecutan las tareas.

La estructura total del software embebido se muestra en la Fig. 9. No se incluye el componente del bootloader ya que este no influye en la aplicación de *emulador SISO*. Los únicos dos niveles que trabajan simultáneamente son los de procesador y

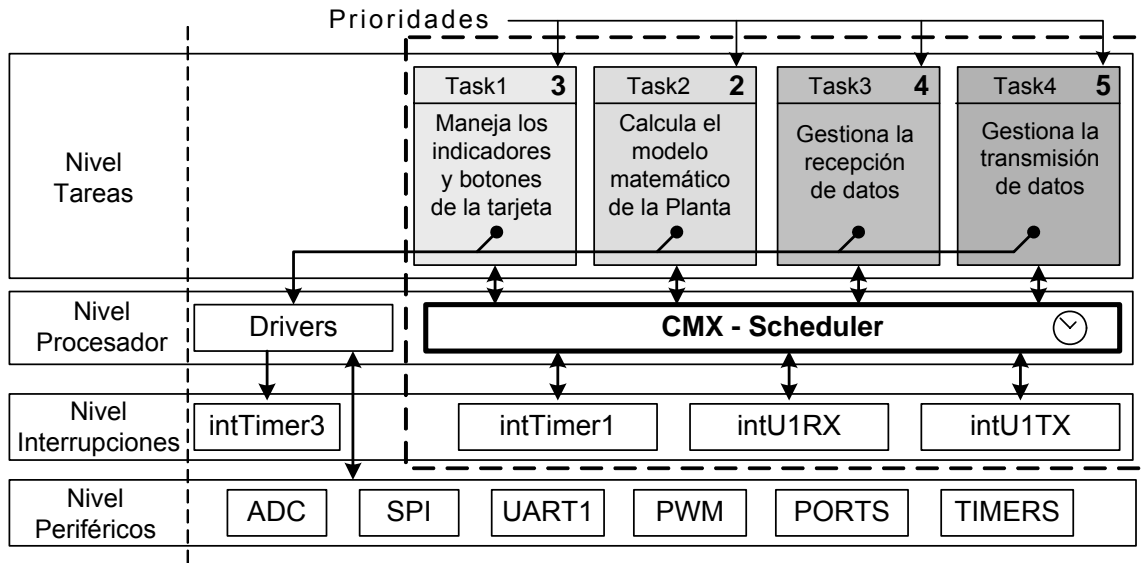


Figura 9. Interacción entre los elementos de Software

periféricos, pero la incorporación del RTOS permite abstraer y suponer que paralelamente a los anteriores están también trabajando los niveles de tareas e interrupciones. La aplicación de emulador se divide en 4 tareas:

- 1) Tarea que maneja los indicadores y botones del emulador según los distintos estados que se tenga en la GUI o en las entradas y salidas analógicas.
- 2) Realiza los cálculos de todo el modelo matemático según la configuración registrada. A esta tarea se la llama repetidamente según el tiempo de simulación.
- 3) Esta tarea se activa ante la recepción de cualquier dato por medio del puerto serial. Gestiona las acciones según el mensaje recibido.
- 4) Administra el envío de datos por el puerto serial desde una pila de salida radicada en RAM.

Resultados

Se hicieron dos tipos de pruebas al sistema: las primeras corresponden a encontrar el comportamiento como un sistema de adquisición de datos y las segundas en su función de emulador SISO.

A. Sistema de adquisición de datos

Ruido - la primera prueba se refiere al contenido de ruido en la salida y entrada analógica. Una manera sencilla de obtener esta característica es conectar la salida analógica a la entrada analógica y realizar un conjunto de lecturas al ADC para un valor fijo en el DAC [14]. Un histograma de estas mediciones nos

proporciona una buena idea del ruido en el sistema (Fig.10). Se obtuvo un desempeño de ± 1 bit de ruido pico-pico.

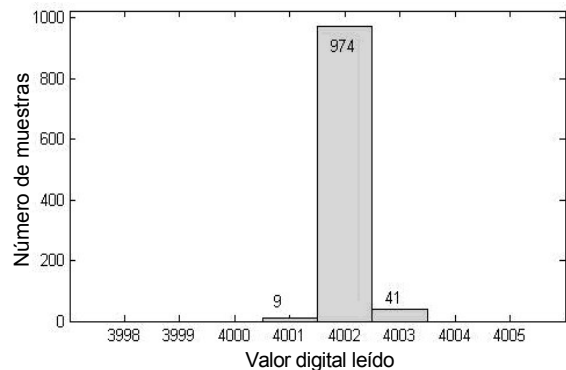


Figura 10. Histograma de distribución de datos

Linealidad - se sigue un procedimiento similar a la prueba de ruido (conectando la salida con la entrada) pero esta vez se varía el valor de la salida analógica desde su mínimo (0 digital) hasta su máximo (4095 digital).

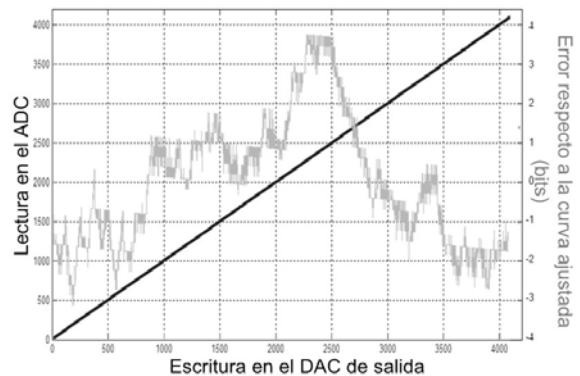


Figura 11. Resultados de prueba de linealidad

Esta prueba demuestra de una manera sencilla la linealidad de todo el sistema que incluye el DAC y el ADC, los filtros antialiasing de entrada y salida, el amplificador de ganancia programable en la entrada y el amplificador a la salida. Usando el conjunto de datos se obtiene la curva que mejor ajusta y respecto a esta se restan todos los puntos de muestra (Fig.11). La linealidad es de +0.121% y -0.106% para la escala completa.

Filtros antialiasing – por medio de puntos de prueba en la placa se tiene acceso a las entradas y salidas de los filtros. Cargando los valores a los potenciómetros digitales se modifica la frecuencia de corte (f_c) e inyectando señales se puede obtener la respuesta del filtro. La Fig.12a muestra el comportamiento frente a una onda de frecuencia variable. La Fig.12b muestra el desempeño ante una señal cuadrada. El comportamiento de filtro como Bessel es muy bueno (la respuesta no muestra sobreimpulso).

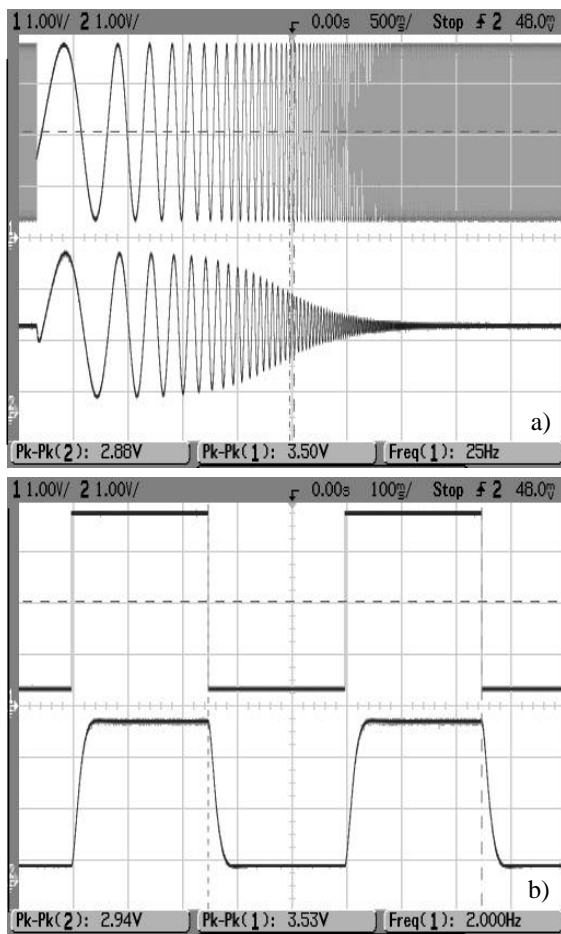


Figura 12. Frecuencia de corte del filtro en 15Hz a) la señal senoidal varía de 1Hz a 1kHz, escala logarítmica b) la frecuencia del pulso rectangular es 3Hz.

Para valores bajos de la frecuencia de corte se tienen bajos errores (-1% a 15Hz) pero para valores altos de f_c se tiene menor control sobre la frecuencia obteniendo resultados de -2.4% a 100Hz y -19% a 400Hz.

B. Resultados como emulador SISO

Con el fin de comprobar el comportamiento del emulador con respecto a la configuración cargada se realizó una comparación de resultados entre una simulación realizada en MATLAB y los resultados experimentales al excitar con una función escalón la entrada del emulador. La Fig.13a muestra la simulación de la ecuación de transferencia de 2^{do} orden mostrada y en la Fig.13b el resultado experimental ante la excitación escalón. En la Fig.13c se aprecia el resultado cuando se carga una no linealidad estática como la que se muestra en el recuadro superior derecho.

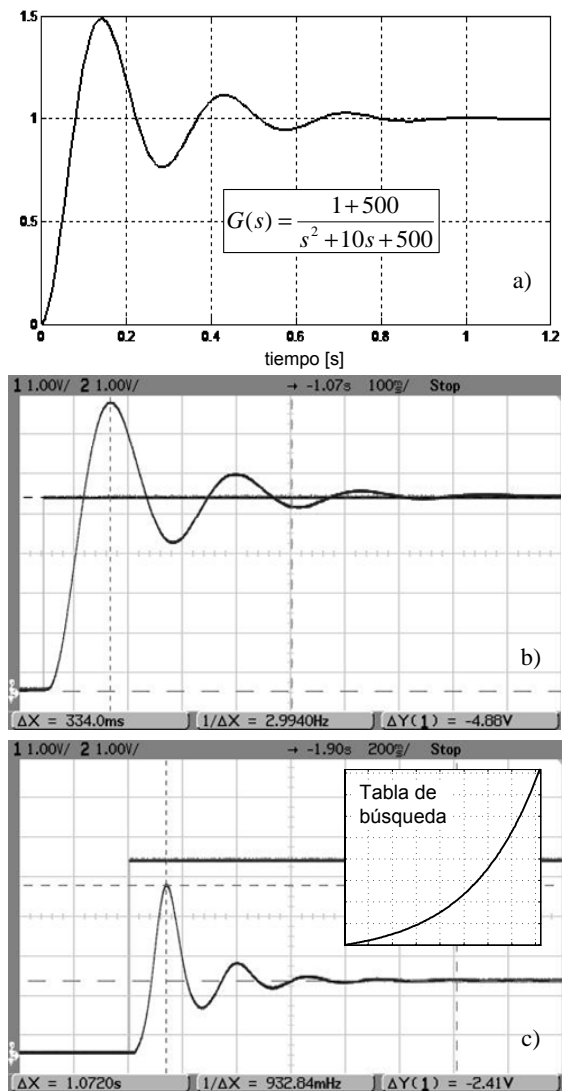


Figura 13. Planta de segundo orden

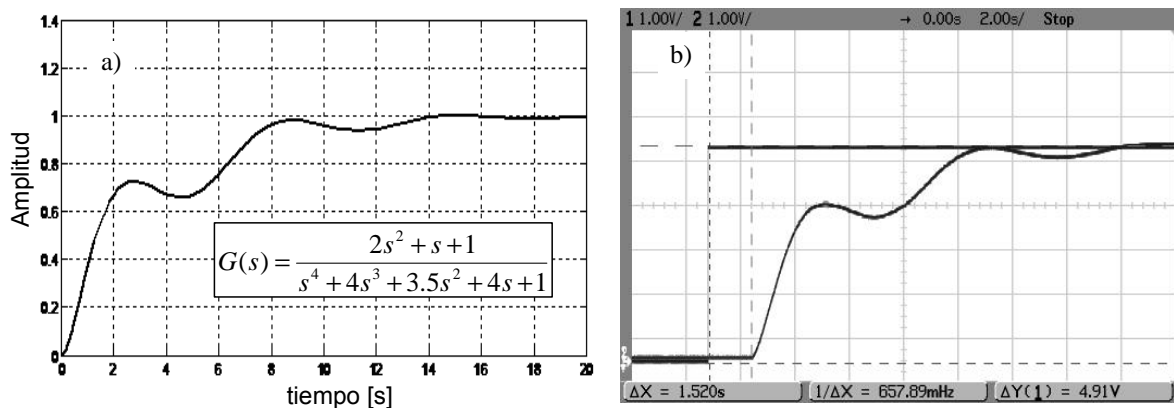


Figura 14. Planta de cuarto orden a) simulado sin retardo y b) experimental con retardo de transporte.

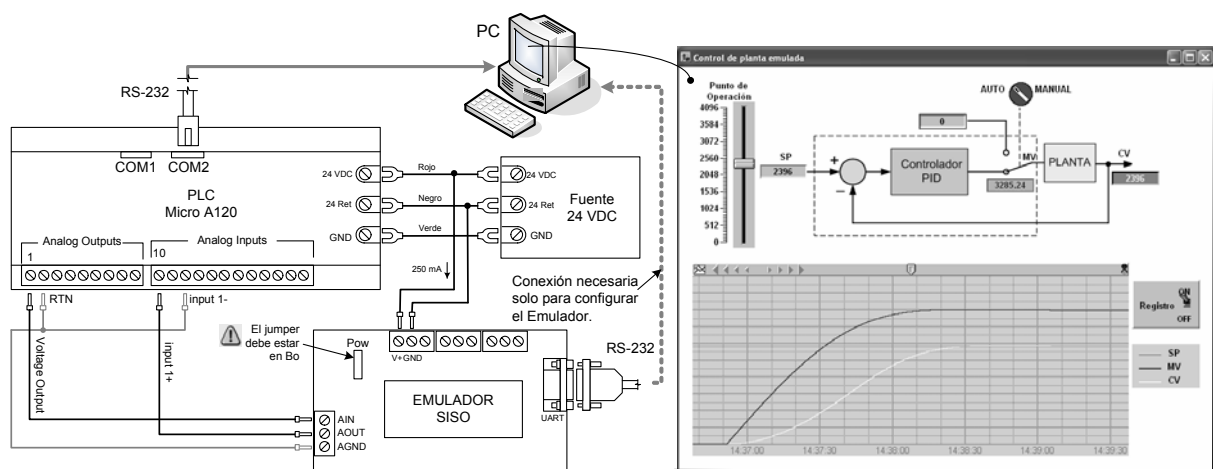


Figura 15. Esquema de una práctica con un PLC Micro A120 y una Interfase Hombre-Máquina (HMI) desarrollada en Lookout.

Al probar el comportamiento del sistema con una ecuación de cuarto orden Fig.14 se puede apreciar que no existe diferencia entre la curva simulada y la obtenida experimentalmente. La ecuación que se utilizó es particularmente sensible a la acumulación de errores en los cálculos de la ecuación de diferencias correspondiente, hecho que destaca la ventaja de usar doble precisión en las operaciones.

El rendimiento del código de todo el sistema en las peores condiciones (ecuación de transferencia de 4to orden, retardo máximo, no linealidad, perturbación multiplicativa, salida PWM activa) logra ejecutar un ciclo de simulación en 8327 ciclos del procesador, lo que a una frecuencia de trabajo de 29.4912 MHz del DSC resulta en un tiempo de 282.35us. Esto permite realizar con tranquilidad simulaciones hasta con una frecuencia de 2kHz.

En la Fig.15 se indica el esquema de conexión del emulador con un PLC industrial típico. Para esta práctica se elaboró una HMI en Lookout 5 (Programa estándar para el

diseño de interfaces de control) con el fin de controlar la planta cargada en el *emulador SISO*. Este esquema puede ser utilizado con varios tipos de estrategias de control dependiendo de la planta emulada que se proponga.

Conclusiones

Se ha descrito una solución HIL de una entrada y una salida analógica al desarrollar un hardware que integra conversores de datos analógicos con precisión de 12 bits y un controlador digital de señales DSC dsPIC30F4013 el cual posee un programa embebido capaz de calcular funciones de transferencia hasta de 4to orden en punto flotante de 64 bits, retardo de transporte y tabla de búsqueda hasta una frecuencia de simulación de 2kHz, junto con inyección de perturbaciones por medio de un potenciómetro mecánico.

Adicionalmente se desarrolló la interfase de usuario (GUI) que permite configurar los parámetros de la planta de manera sencilla e

intuitiva, integrando la generación de parámetros para el programa del DSC y la gestión de las comunicaciones necesarias para enlazarse con el hardware por medio del puerto serial RS232. Posee las herramientas gráficas que facilitan la calibración de los parámetros analógicos del circuito emulador. Queda abierta la posibilidad de mejorar el software de emulación HIL.

El equipo como tal cuesta menos de 90USD y las herramientas para el desarrollo del código en el DSC son gratuitas, lo que posibilita su uso para el desarrollo de varias aplicaciones de control. La incorporación de un DSC da la opción de crear controladores avanzados de tipo adaptativo, difuso, neuronal, entre otros ó permite aumentar la funcionalidad de controladores actuales. También pone a disposición la infraestructura para compatibilizar con señales industriales analógicas y digitales.

Referencias

- [1] Wojciech Grega, "Hardware-in-the-loop simulation and its application in control education" en *29th ASEE/IEEE Frontiers in Education Conf.* San Juan, Puerto Rico, 1999 pp. 12b6-7-12b-9.
- [2] P. M. VISSER, M. A. GROOTHUIS and J. F. BROENINK, "FPGAs as versatile configurable I/O devices in Hardware-in-the-Loop Simulation" en *Work-In-Progress Proceedings*, Lisboa, 2004, pp 41-42.
- [3] Martin Gomez, "Hardware In The Loop Simulation", en <http://www.embedded.com>, 2001.
- [4] "dsPIC30F3014, dsPIC30F4013 Data Sheet", DS70138C, Microchip Technology Inc., USA, 2004.
- [5] "dsPIC30F Family Reference Manual" , DS70046C, Microchip Technology Inc., USA, 2004.
- [6] "Mixed-Signal and DSP Design Techniques" Analog Devices Technical Reference Books, USA, 2001, pp. 2.1-2.8.
- [7] B. Baker, "Using Digital Potentiometers to Design Low-Pass Adjustable Filters", *Microchip Technology Inc.*, AN737, USA, 2004.
- [8] Mathworks, "Real Time Workshop User's Guide", ver. 6, 2006, pp. 1.2-1.19.
- [9] Ogata K., "Ingeniería de Control Moderna", 2da ed., Prentice-Hall, México 1993, pp. 15-54, 98-104.
- [10] Smith, Steven W., "The Scientist and Engineer's Guide to Digital Signal Processing", 2da ed., California Technical Publishing, San Diego 1999. pp. 609-611.
- [11] Smith C., Corripio A., "Control Automático de Procesos", Primera Edición, Ed. Limusa, México 1997, pp. 263,591-593.
- [12] "Ingenia dsPIC bootloader User's Guide" , v1.00, Ingenia-Cat S.L., España, 2005.

[13] "Cmx – Sheduler® User's Manual", CMX Systems Inc., USA, 2002.

[14] B. Baker, "Layout Tips for 12-Bit A/D Converter Application", AN688, Microchip Technology Inc, USA, 2004.



J. David Santana nació en Quito, Ecuador, en 1981. Recibió el título de pregrado en ingeniería electrónica, automatización y control en el 2006 en la Escuela Politécnica del Ejército y es miembro del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) desde el 2005. Entre los campos de interés está la aplicación del procesamiento digital en el campo de control y el diseño de aplicaciones embebidas.