

SISTEMA INTEGRADO PARA LA FORMACIÓN Y REALIZACIÓN DE PROYECTOS EN ELECTRÓNICA, AUTOMÁTICA Y SISTEMAS INFORMÁTICOS

M.J.. LÓPEZ, L. GARCÍA, J. LORENZO, J. TERRÓN
*Departamento de Ingeniería de Sistemas y Automática.
Universidad de Cádiz. España.
manueljesus.lopez@uca.es,*

En este artículo se presenta el sistema EDECOS, que supone un entorno para simulación, diseño y evaluación de sistemas de control, utilizable por los estudiantes de ingeniería electrónica y automática, así como de informática de sistemas. Supone una integración de múltiples disciplinas, y su objetivo es servir desde el punto de vista docente, como un entorno didáctico para la realización de trabajos de curso y proyectos fin de carrera, así como para tender una pasarela hacia el ámbito de la investigación y desarrollo de sistemas de control en electrónica, automática y sistemas de tiempo real.

1. Introducción

Es un hecho habitual que los estudiantes universitarios perciban una perspectiva local de cada asignatura, de forma que su principal objetivo consiste en superar las pruebas de evaluación de cada materia; ocurriendo que la deseable integración de conocimientos no se produce de forma “on-line”, sino a posteriori cuando se enfrenta a casos prácticos reales; como por ejemplo frente a la realización de proyectos fin de carrera, a prácticas en empresas, a trabajos de laboratorio, o una vez que están inmersos en la actividad profesional. Dicha integración de conocimientos, habilidades, capacidades y aptitudes debe ser un objetivo prioritario en la formación, de forma que cuanto antes el estudiante conciba la realidad como un sistema interactivo multidisciplinar que requiere de muy diversas fuentes de información y de un conocimiento integrado, mucho mejor será para su aprendizaje y posterior incorporación al ámbito laboral.

En el contexto de la automatización y control de procesos industriales, el profesional de ingeniería electrónica se tiene que enfrentar con retos cada vez mayores. Por ello, si se consigue que el estudiante universitario esté formado en las disciplinas de electrónica y automática de una forma integrada, dicho estudiante podrá abordar de forma más cómoda y con mayores posibilidades los nuevos retos.

En este contexto, se plantea en este trabajo un entorno para la simulación, diseño, análisis de sistemas y control en tiempo real estricto; cuya finalidad principal consiste en que sirva a los alumnos como banco de pruebas para sus diseños, así como para la integración de conocimientos de disciplinas relacionadas con la electrónica, informática y automática. La electrónica, en cuanto a que se tratan circuitos electrónicos que implementan reguladores, o que se usan como osciladores, así como la interfaz necesaria entre diferentes sistemas (adaptación de señales, etapa de potencia, etc.). La automática, en lo que se refiere al diseño e implementación de algoritmos de control convencionales y avanzados para la regulación de las variables del proceso. La informática, en el contexto que se requiere el empleo de entornos para programación en tiempo real estricto, la adquisición y el procesamiento de datos, así como a la conectividad mediante tecnologías estándares a sistemas de comunicación industriales. En este artículo presentamos un entorno que estamos desarrollado en la universidad de Cádiz y que en lo que sigue vamos a denominar EDECOS (*Environment for Design and Evaluation of Control Systems*), que tiene una aplicabilidad académico-didáctica, pero que también se puede usar para aplicaciones de otra índole, más orientada a la industria o a la investigación.

El resto del artículo está estructurado como sigue: en el apartado segundo se plantean las características de un entorno de desarrollo y se indican algunos productos comerciales del tipo CACSD (*Computer Aided Control System Design*); mientras que en el apartado 3 se pasa a describir el sistema EDECOS propuesto y sus posibilidades docentes. En el apartado 4 se trata sobre algunas líneas de continuación a seguir en el contexto del empleo de laboratorios virtuales y remotos, así como la utilización del e-learning. Finalmente, se resumen las conclusiones.

2. Funcionalidades básicas del sistema

La simulación en tiempo real con prestaciones de “*hardware-in-the-loop*” (HIL) es una de las principales herramientas para el testeo y análisis de sistemas de control; de forma que pueden realizarse pruebas realistas de hardware y software antes de pasar a construir prototipos. La idea básica consiste en que los diferentes componentes del sistema de control van a recibir señales que pueden proceder de forma indiferente de procesos reales o de procesos simulados. Los componentes software van a estar desarrollados en C/C++; por lo que van a utilizar propiedades de la programación orientada a objetos tales como la encapsulación, la herencia y el polimorfismo, muy útiles para la implementación de componentes [1], [2], [4], [13], [15]. Como elementos hardware que interconectan los diferentes elementos del sistema, se usan tarjetas de adquisición de datos y dispositivos para adaptación de señales.

Los procesos simulados se implementan según dos opciones: 1) utilizando componentes de electrónica analógica, 2) digitalmente en un computador tipo PC, junto con el hardware de entrada/salida.

Además de procesos simulados, se dispone de procesos de laboratorio como son: sistema de control de temperatura, sistema de control de nivel, sistema de control de velocidad angular y sistema de control de posición angular. También se dispone de circuitos electrónicos que dependiendo de los parámetros que tomen los componentes pueden funcionar como osciladores electrónicos convencionales, o como osciladores caóticos; siendo susceptibles de control en tiempo real.

En el simulador analógico se implementan sistemas escalares (SISO) y multivariantes (MIMO 2x2) lineales con variables entrada/salida acotadas de hasta orden cinco en cada función de transferencia, con la opción de incorporarle elementos de retardo puros (dead-time) en cada uno de los canales, a la entrada o a la salida del proceso.

El simulador de procesos basado en PC es mucho más versátil y flexible, dado que la programación se hace mediante software. En este caso, se implementan procesos SISO y MIMO (de dimensión hasta 4x4) lineales y no lineales. Las simulaciones se hacen en tiempo real estricto; si bien, para acelerar las pruebas en el caso que se requiera o convenga, se realizan escalados temporales. Por ello, se pueden simular en tiempo real estricto procesos con dinámicas relativamente lentas, como ocurre con determinados procesos industriales; pero también sistemas con dinámicas relativamente más rápidas, como ocurre con manipuladores robóticos o aeronaves. En estos casos, la restricción en cuanto a qué tipo de proceso es implementable en el simulador la pone las prestaciones de la unidad de procesamiento digital utilizado (basado en PC), así como las características de las tarjetas de adquisición de datos. Las tarjetas de adquisición de datos utilizadas son de la casa comercial National Instruments [10] para bus PCI y frecuencias de muestreo de señal superiores a 10 kHz, con dos o cuatro salidas analógicas y 16 o 32 entradas analógicas, además de múltiples entradas y salidas digitales.

La complejidad de los algoritmos de control que se quieran probar va a determinar la elección de la forma de implementar dichos algoritmos. Esto es evidente, dado que si se dispone de un circuito que realiza un controlador PID o un compensador de atraso/adelanto en electrónica analógica, se puede utilizar dicho circuito y realizar el ajuste de los parámetros del controlador; pero si se quiere implementar un controlador más complejo, éste ya se realizaría de forma digital. El controlador del sistema se va a realizar con alguna de las siguientes opciones:

1. En hardware mediante electrónica analógica (en caso de controladores sencillos de implementar).

2. Digitalmente mediante software y procesador digital, en cuyo caso se tienen las siguientes alternativas a realizar: 2.a) basado en PC y sistema de tiempo real estricto, 2.b) basado en sistema DSP, 2.c) basado en sistema microcontrolador tipo PIC).

3. Mediante un controlador industrial tipo PID implementado de forma autónoma (en el laboratorio se tienen PID industriales de la casa Omron), o como módulo de un autómata programable (PLC).

Como ya se ha indicado anteriormente, el sistema ha de realizarse e interconectarse de forma que cada componente no tenga que distinguir si las señales que recibe son de un elemento real o uno simulado, e igualmente que tampoco deba distinguir si las señales que da a la salida van a un elemento real o a uno simulado. De esta forma, por problemas de interconexión que aparecen en los sistemas reales, también van a aparecer en el entorno EDECOS que proponemos.

3. Sistema EDECOS

Basado en el planteamiento descrito en el apartado anterior, estamos desarrollando un entorno hardware/software para el diseño y evaluación de sistemas de control (*Environment for Design and Evaluation of Control Systems*, EDECOS). En la figura 1 se pueden ver dos pantallas típicas que corresponden a dos de los elementos fundamentales del entorno, como son EPESC y ControlAvH Tune; mientras que la figura 2 se muestra un esquema con los diferentes elementos que forman EDECOS.

El ordenador PC0 hace de host computer, por lo que tiene instaladas las diferentes aplicaciones para diseño y análisis, así como los sistemas de desarrollo para Windows XP y Linux respectivamente. TuHiCo Toolbox y GARCO Toolbox son librerías o toolboxes que hemos desarrollado en nuestro grupo para Matlab (Mathworks) [9], y que se utilizan como primer escalón o paso en el proceso de diseño.

Además, se utilizan algunas toolbox comerciales de Matlab y también Simulink (Mathworks), aunque se dispone también de VisSim (Visual Solutions) [17] como alternativa a Simulink [9]. Como herramienta de programación y desarrollo para Windows XP utilizamos Builder C++ (Borland) [3], y para las tarjetas de adquisición de datos los drivers correspondientes a la librería NI-DAQ de National Instruments (NI) [10]. Para desarrollar las aplicaciones de tiempo real duro, utilizamos Linux y Real Time Linux (Linux-RTAI) (freeware), así como los drivers para tarjetas de adquisición de datos que proporciona la librería Comedi (freeware) [5], [7], [8], [12], [14]. Además, para la simulación electrónica se emplea el software Multisim o Electronic-Workbench.

El ordenador PC1 dispone de una aplicación informática para sistemas operativos Windows (Windows 98, Windows 2000, Windows XP, Windows Live), ControlAvH Tune [16], que se ha desarrollado en Builder C++ , y que permite el diseño de controladores del tipo PID, así como también de controladores basados en la teoría de control robusto (H2, Hinf). ControlAvH Tune lo puede utilizar un operador de planta, dado que dispone de una interface apropiada fácil de manejar, para realizar el ajuste de controladores convencionales PID y de controladores robustos.

El controlador en tiempo real estricto (HRTC, hard real time controller) está implementado en el ordenador PC2. Esta aplicación se ha desarrollado en Linux utilizando Real Time Linux (RTAI) [19]. El algoritmo de control se calcula en el PC1 mediante ControlAvH Tune, se recibe por PC2 mediante comunicación TCP/IP [6] y conexión en red local Ethernet a la que tienen acceso PC0, PC1, PC2 y PC3. El controlador se conecta con el proceso a controlar mediante una interface entrada/salida basada en tarjetas de adquisición de datos NI (ver figura 4). Como alternativa, también se puede utilizar un controlador PID industrial autónomo, o un PID instalado en un autómata programable (PLC) (ver figura 9); con lo que se han considerado diferentes opciones que se dan en la práctica.

La simulación del proceso a controlar también se realiza en tiempo real estricto (HRTS, hard real time simulator), y se implementa en el PC3. La conexión con el controlador se realiza a través de tarjetas de adquisición de datos NI. Como se ha indicado antes, también se puede utilizar como proceso a controlar un sistema simulado con electrónica analógica.

Como sistema real a controlar se podría utilizar un proceso dentro de una planta industrial si se tuviera accesibilidad; o procesos de laboratorio dentro de los disponibles, que son actualmente: proceso térmico para control de temperatura, proceso para control de nivel y caudal de un líquido (ver figura 6), servosistema para control de posición o velocidad angular (ver figura 7). Está previsto que se disponga en breve, además de un robot de tres grados de libertad y una plataforma con helicóptero con tres grados de libertad, en el que se puedan incorporar los algoritmos y probarlos de forma real.

Otros casos particulares de sistemas reales de laboratorio a controlar se consideran los osciladores electrónicos caóticos. En estos sistemas se consideran dos problemas de control básicos: 1) la sincronización de dos osciladores caóticos, 2) la obtención de órbitas periódicas inestables. Estos dos problemas tienen una aplicabilidad importante en el ámbito de las ingenierías electrónica y de control (ver figura 8).

Los modelos matemáticos de sistemas implementados actualmente, o que está prevista su realización en el simulador de tiempo real estricto HRTS cubren una amplia gama de procesos, siendo ampliable según se vaya requiriendo; por lo que se dispone de un entorno flexible, ampliable y de propósito general. Los sistemas que se están incorporando son los siguientes (en algunos casos se deja la denominación original en inglés, puesto que es la forma habitual de denominar dichos sistemas, y no se tiene una traducción consensuada): manipuladores robóticos multi-articulares, péndulo inverso o “cart-pole system”, “TORA system”, “convey-crane system”, “pendubot system”, vehículos marinos (buques de superficie y submarinos), vehículos terrestres, aeronaves, misiles, helicópteros, columnas de destilación, intercambiadores de calor, turbinas para generación de energía eléctrica, reactores químicos, antenas y plataformas estables, proceso para control de pH, entre otros.

4. Líneas de continuación

A lo largo de la última década se han ido incorporando los nuevos recursos que proporcionan las tecnologías de la información (TIC) en el proceso de aprendizaje de la electrónica, la automática y la informática. Los resultados más innovadores son la proliferación de numerosos laboratorios virtuales y remotos, además de los recursos ya más asentados de utilizar aplicaciones gráficas interactivas que explotan la interactividad como herramienta pedagógica [18]. Una utilidad de interés consiste en desarrollar unos laboratorios virtuales que permitan a los estudiantes reproducir, ampliar, modificar las prácticas presenciales en puestos de trabajo fuera del laboratorio de la universidad. Esto se debe a que en la práctica, sucede que en muchas ocasiones los estudiantes no consiguen finalizar las prácticas de laboratorio en el tiempo estipulado, debido a que cada individuo o cada grupo tiene su propia dinámica de trabajo, y por tanto lo que para la mayoría es tiempo suficiente, resulta que para algunos no lo es. Esto provoca que el proceso de aprendizaje de estos estudiantes no se complete como sería deseable. Con el objetivo de proporcionar una solución a este problema se plantea el proporcionar un laboratorio virtual; así como un laboratorio de acceso remoto. Sin embargo, esto no lo tenemos aún desarrollado, estando en una fase preliminar. Adicionalmente se pretende disponer de sistemas para aprendizaje basado e-learning; para lo cual se tiene formado un grupo de innovación docente centrado en la enseñanza de la electrónica y automática industrial.

5. Conclusiones

El sistema EDECOS propuesto en este artículo se está actualmente desarrollando, habiendo realizado en la actualidad algunas de los componentes. Para los estudiantes de ingeniería electrónica y automática, así como de informática de sistemas supone un entorno en el que pueden aplicar los contenidos de las diferentes disciplinas estudiadas, integrarlas y evaluar los resultados. El sistema es flexible y versátil, aplicable a nivel docente así como en el ámbito de la realización de proyectos fin de carrera y de la investigación dentro de las TIC.

Como línea de continuación se va a abordar el problema de forma que se pueda utilizar además en el contexto de laboratorios remotos y e-learning. Además se tiene la idea de que la comunidad universitaria tenga acceso al sistema, a fin de compartir experiencias, detectar problemas, fallos, etc., y de las sugerencias que surjan mejorar y ampliar las prestaciones del sistema.

Referencias

- [1] A. Abran and J.W. Moore, executive editors, *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2004.
- [2] B.S. Heck, L.M. Wills and G.J. Vachtsevas, *Software Technology for Implementing Reusable, Distributed Control System*. IEEE Control System Magazine, pp. 21-35, February 2003.
- [3] Builder C++. <http://www.borland.com>
- [4] Burns A., A. Wellings, *Sistemas de Tiempo Real y Lenguajes de Programación*, Addison Wesley (2003).
- [5] COMEDI –The Linux Control and Measurement Device Interface. <http://www.comedi.org/>
- [6] D.E. Comer, D.L. Stevens, *Internetworking with TCP/IP (0Vol. III): Client-Server Programming and applications*. Prentice-Hall, 1996.
- [7] K.A. Robbins, S. Robbins, *UNIX Programación práctica. Guía para la Concurrencia, la Comunicación y los Multihilos*. Prentice-Hall, 1997
- [8] Mantegazza, y S. Papacharalambous, “Real time application interface”, Linux Journal (2000).
- [9] MathWorks Inc. *Matlab*, www.mathworks.com.
- [10] National Instruments Corporation. *Measurement and automation catalog*. <http://www.ni.com>.
- [11] RTAI. <http://www.aero.polimi.it/rtai>.
- [12] RTnet – Hard Real-Time Networking for Linux RTAI. <http://www.rts.uni-hannover.de/rtnet/>
- [13] S. Bennet, *Real Time Computer Control*. Prentice-Hall, 1998.
- [14] SuSE Linux 9.0, *Administration and user guide*, SuSE Linux AG (2003).
- [15] V. Gazi, M.L. Moore, K.M. Passino, W.P. Shackleford, F.M. Proctor, J.S. Albus, The RCS Handbook. *Tools for Real-Time Control Systems Software Development*. Wiley, 2001.
- [16] J.Lorenzo. Software integrado para el diseño y control de sistemas en tiempo real. Tesis Doctoral. Universidad de Cádiz, 2007.
- [17] Visual Solutions Inc. *VisSim (Visual Simulator)*. <http://www.vissim.com>.
- [18] L. Gomes, J. García-Zubía (eds.), *Advances on remote laboratorios and e-learning experiences*. University of Deusto, 2007.
- [19] L. García, M. J. López, J. Lorenzo, *Hard Real Time Based on Linux/RTAI for Plant Simulation and Control Systems Evaluation*. WSEAS Transactions on Systems and Control. Issue 2, Vol. 1, pp 161-168, 2006

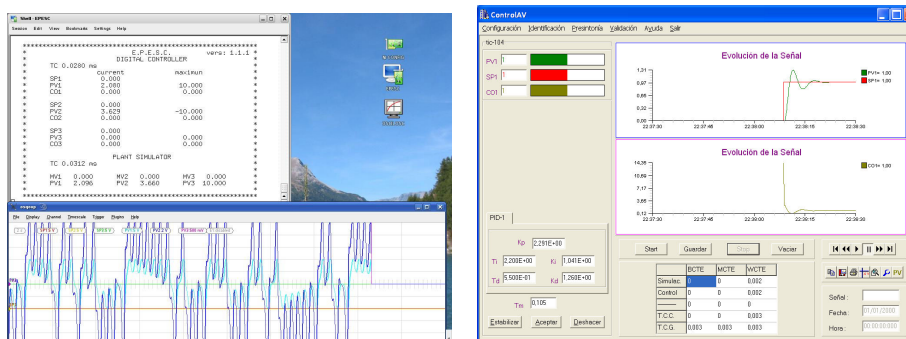


Figura 1. Pantallas respectivas de EPESC y ControlAvH Tune

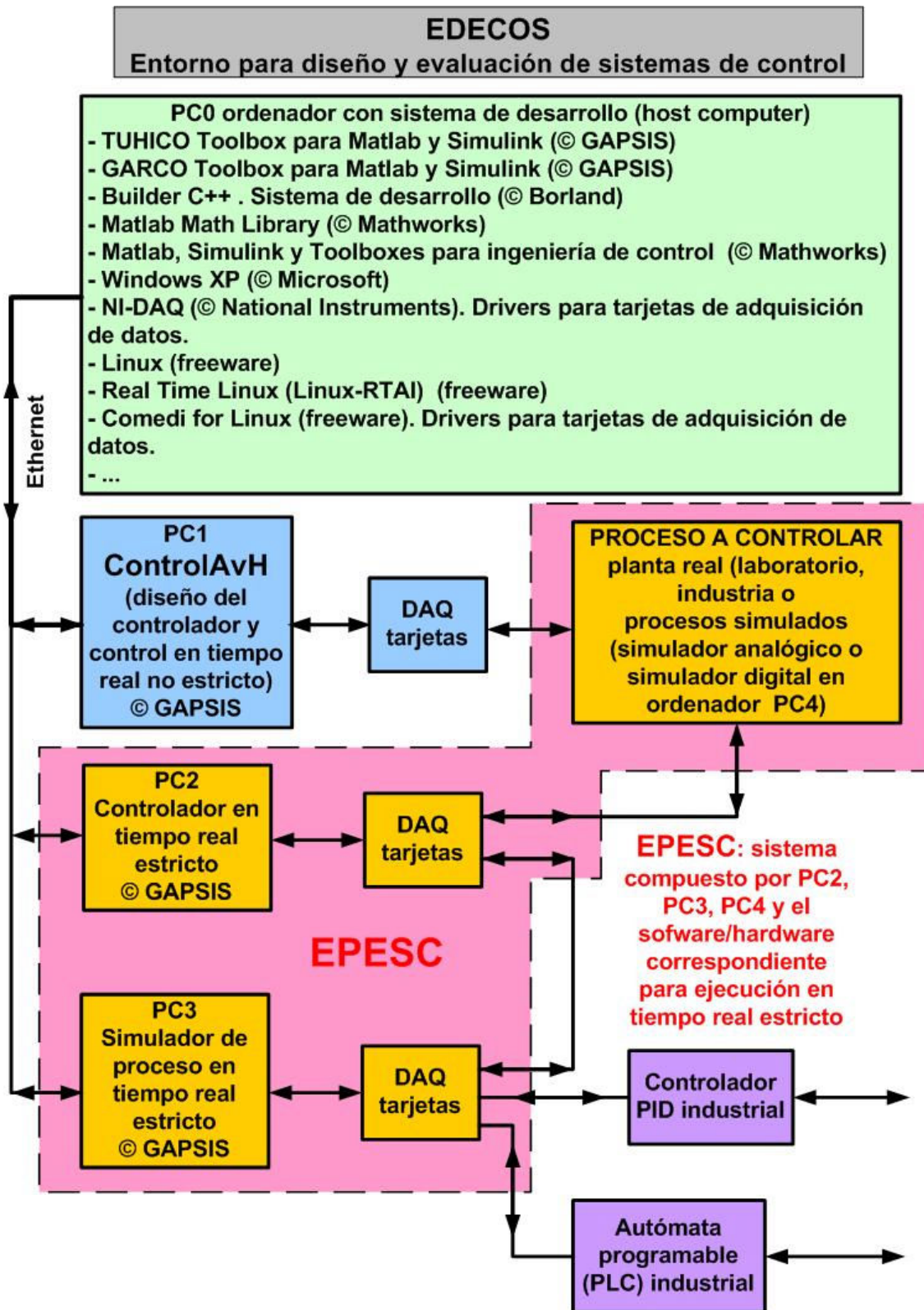


Figura 2. Elementos del sistema EDECOS

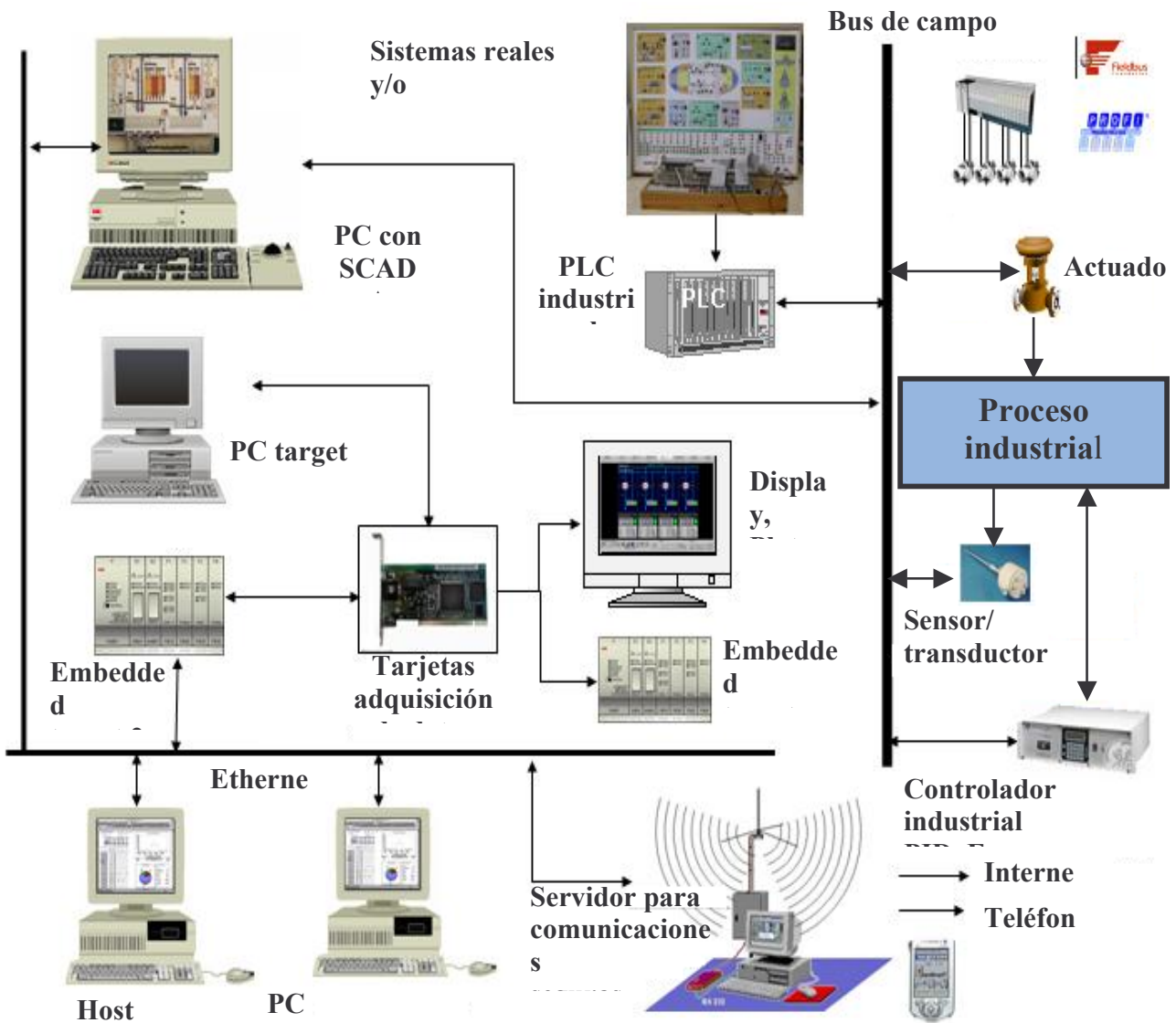


Figura 3. Interconexión de los sistemas mediante red Ethernet y Bus de Campo industrial.

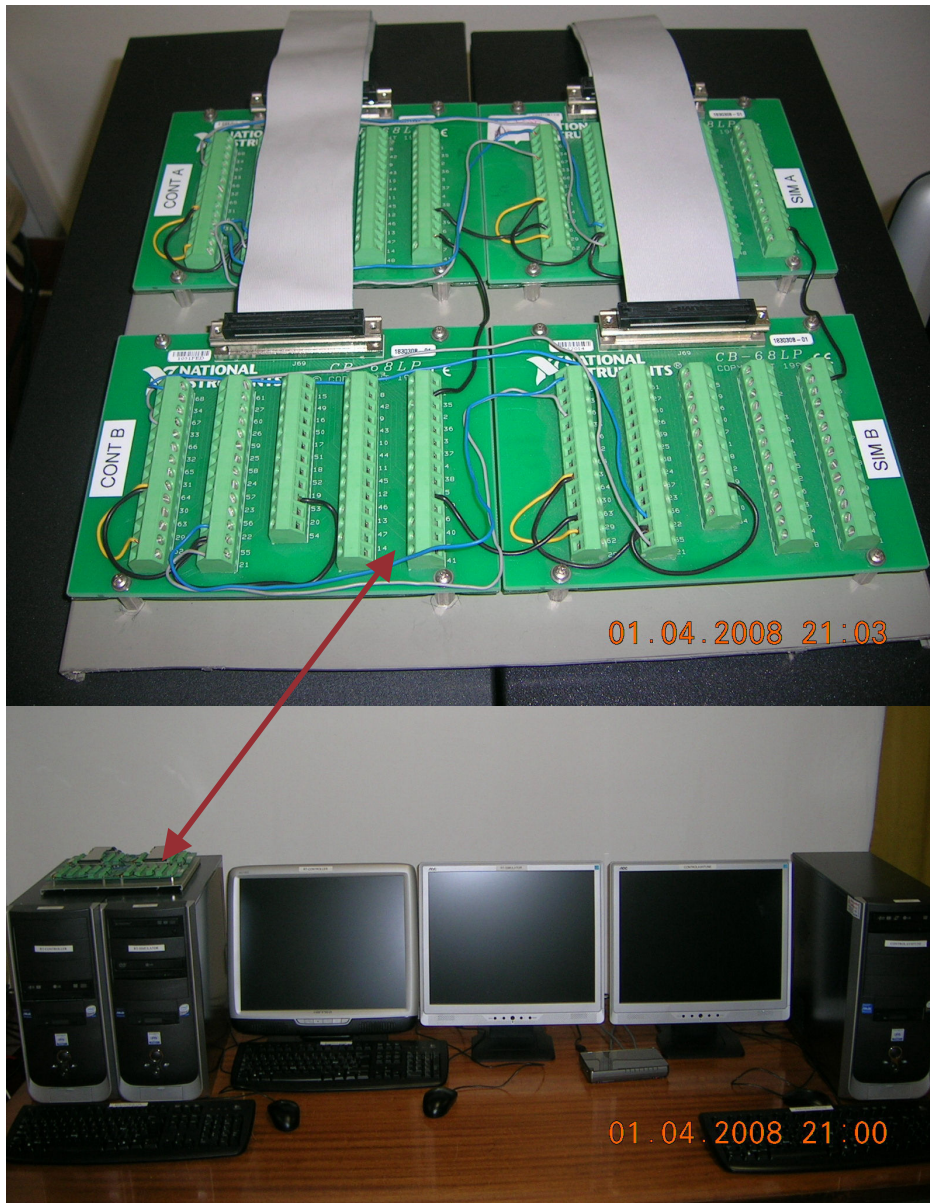


Figura 4. Entorno para realizar “hardware in the loop” compuesto de tres PC: PC1 controlador de tiempo real estricto, PC2 simulador de tiempo real estricto del proceso a controlar, PC3 contiene la aplicación ControlAvH Tune para diseño de controladores PID y controladores robustos H

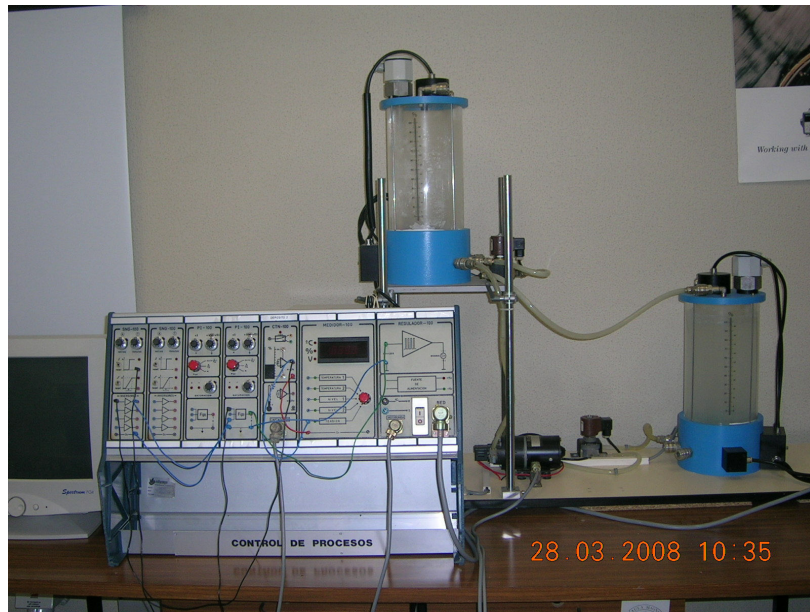


Figura 6. Proceso de laboratorio a controlar: control de nivel y caudal



Figura 7. Proceso de laboratorio: servosistema, control de posición/velocidad

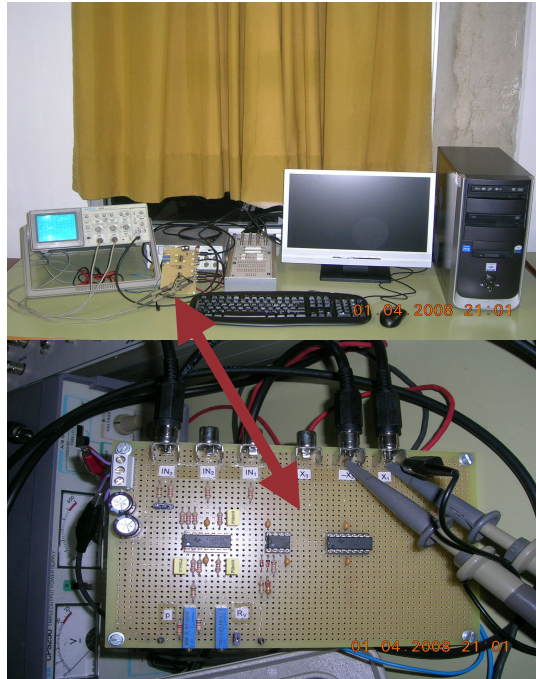


Figura 8. Utilización de circuito electrónico caótico como proceso a controlar

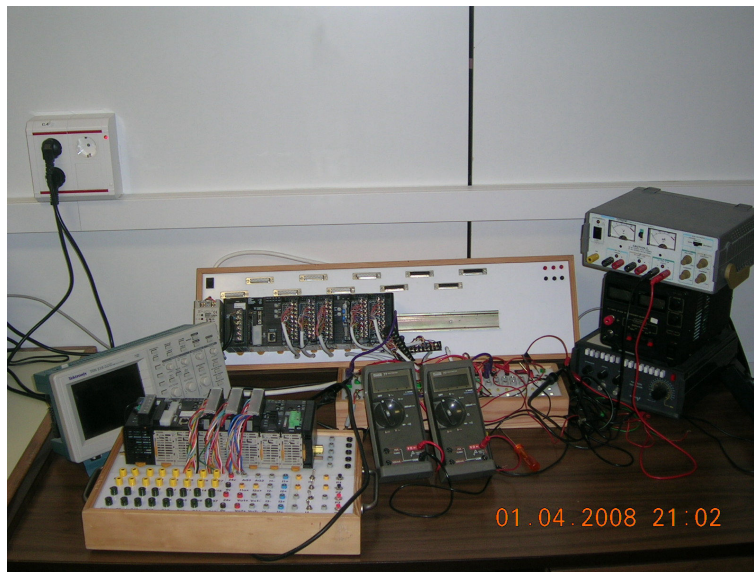


Figura 9. Control mediante módulo PID de autómatas programables