

"simPROCes" Simulador de Procesos

C. Domínguez, J. Albaladejo, J. M. Martínez, H. Hassan, Á. Perles

Departamento de Informática de Sistemas y Computadores
Universidad Politécnica de Valencia
Camino de Vera, s/n. 46022 Valencia
e-mail: carlostd@disca.upv.es

Resumen

Presentamos simPROCes, un simulador para la validación de sistemas informáticos de control, que empleamos en la docencia de la informática industrial.

simPROCes simula el proceso bajo control junto con el sistema de entrada/salida que le sirve de interfaz.

Tanto el modelo del proceso como el del sistema de entrada/salida se cargan en forma de librerías de enlace dinámico, lo que da gran flexibilidad para la elección de distintos sistemas de prueba.

La comunicación entre el simulador y la aplicación de control se realiza mediante TCP/IP, y la facilidad que esto proporciona para elegir las ubicaciones de ambos procesos, tiene ventajas aprovechables en la docencia, aunque es cierto que también puede introducir limitaciones para simular en tiempo real.

Actualmente tenemos desarrollada la simulación de la tarjeta de entrada/salida ADLink PCI9112 y de un depósito de agua, elementos reales que usamos en el laboratorio.

1. Justificación

La validación es clave en el desarrollo de sistemas informáticos industriales, e insistimos mucho a nuestros alumnos sobre este punto. Los simuladores de procesos físicos son una buena herramienta para realizar pruebas.

Hace ya años que enfocamos la docencia desde una realización práctica [1][3][7], haciendo que los alumnos desarrollen un miniproyecto de control por computador [4][8][9]. Actualmente estos miniproyectos tienen ya un alcance

multidisciplinar que abarca las materias de todo un curso [10].

En [2] realizábamos una descripción de las herramientas seleccionadas para nuestra asignatura troncal. El criterio fundamental es facilitar el trabajo que los alumnos deben realizar fuera del horario del laboratorio.

El número de módulos de entrada/salida disponibles en las aulas, así como de procesos reales o modelos a escala reducida donde desarrollar ensayos, es bastante limitado.

Una de las herramientas empleadas, el simulador de señales simseny [11], ha facilitado la actividad de grupos numerosos en aulas equipadas con pocos sistemas de entrada/salida. Salvando la escasez de procesos, simPROCes ayudará en la validación, y también en la especificación de los sistemas de control, ya que disponer del modelo nos aligera el trabajo de describirlo para después establecer los requerimientos de su controlador.

En este tema de la especificación, nos han estado sirviendo los procesos modelados con applets Java que los alumnos descargan desde uno de los servidores de la signatura [5][6], pero al estar limitados a interacciones con un operador, no es posible el desarrollo de un control automático.

La pieza que cubra estos dos temas podría ser simPROCes.

2. Arquitectura del simulador

La figura 1 muestra la arquitectura simPROCes. Se trata de una extensión del simulador de señales simseny, al que se añade el modelo de un proceso físico a controlar.

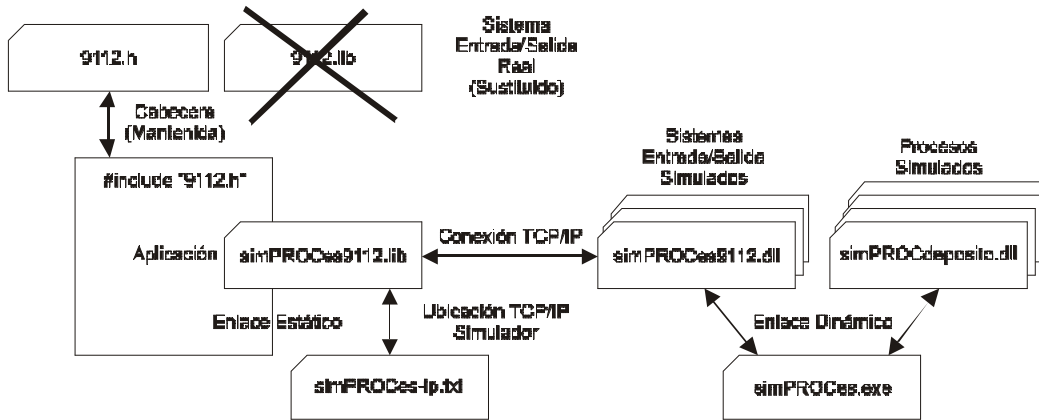


Figura 1. La arquitectura simPROces

2.1. Los módulos

El módulo que define la arquitectura es el programa simPROces.exe (en la parte inferior derecha del gráfico), que permite al usuario la carga en tiempo de ejecución de una librería con el simulador del proceso y de otra con el simulador del sistema de entrada/salida, ambos a elección de entre los modelos que puedan estar desarrollados.

simPROces.exe es el nexo entre el proceso y el sistema de entrada/salida, definiendo la asignación de puertos a señales físicas de accionamiento y de sensorización.

El simulador del sistema de entrada/salida, al igual que en simseny, se realiza con dos módulos enlazados por una conexión TCP/IP, lo cual permite tanto la ubicación de los componentes en una sola máquina, como en dos máquinas conectadas en la red.

Desde el punto de vista del programador de la aplicación de control (a la izquierda del gráfico), el uso del simulador afecta sólo en el proceso de montaje, cuando se sustituye un módulo de librería facilitado por el fabricante de la tarjeta de adquisición de datos por el módulo simulado. El proceso de traducción no se altera, ya que se sigue empleando el mismo archivo de cabecera, ofreciendo el simulador de entrada/salida una interfaz equivalente al sistema real.

Así, el enfoque de construcción de una versión de pruebas para la aplicación de control, difiere del empleado en simseny, en el que la aplicación de control incorpora funciones que tienen que ver con la activación del simulador. En simPROces se intenta validar la aplicación para distribuirla tal cual como se tradujo, por tanto, información relativa al funcionamiento del simulador, como por ejemplo la ubicación del simulador del proceso, se consigue abriendo el archivo de texto simPROces-ip.txt de forma totalmente transparente.

2.2. Las interfaces

Construir el módulo de nexo simPROces.exe implica la definición de las interfaces que deben respetar las librerías de enlace dinámico que pretendan conectarse a él.

Así, por ejemplo, la figura 2 muestra la interfaz que los módulos de simulación de proceso deben exponer a simPROces.exe.

```
int interrogar(char *facilidades);
int parametrizar(double *pars);
int iniciar(double *estado);
int parar(void);
int continuar(void);
int accionar(double *accs);
int perturbar(double *perts);
```

```
int observar(double *obs);
int informar(int *opciones);
```

Figura 2. Interfaz del simulador de proceso

Comentemos brevemente dichas funciones.

La función de interfaz `interrogar()` permite a `simPROCs` determinar las características básicas del simulador del proceso, relativas a las señales de accionamiento y de sensorización, así como a conjuntos seleccionados de variables de estado y de parámetros de configuración.

`parametrizar()` permite configurar el sistema modelado. Por ejemplo, la figura 3, que muestra la vista de la realización `simPROCdeposito.dll` del depósito de agua del laboratorio, marca en negro ciertos parámetros ajustables, como la superficie de la base o las constantes características de las válvulas. La parametrización permite ensayar un

conjunto de casos a partir del mismo esquema de proceso.

`iniciar()`, `parar()` y `continuar()`, controlan el proceso de simulación. La primera de las funciones acepta un estado inicial de partida.

`accionar()` y `perturbar()` provocan las excitaciones en el proceso. Las acciones vienen, a través del simulador de entrada/salida, desde la aplicación de control; mientras que las perturbaciones son generadas desde el propio `simPROCs`.

`observar()` traslada las variables observables de los sensores, de nuevo a través del simulador de entrada/salida, hacia el controlador.

Finalmente, `informar()` permite desplegar una representación gráfica del estado del proceso que el propio simulador ofrece. Aunque los alumnos construyen una interfaz gráfica con el operador en su aplicación de control, les ayuda disponer de estas vistas ya en la fase de especificación.

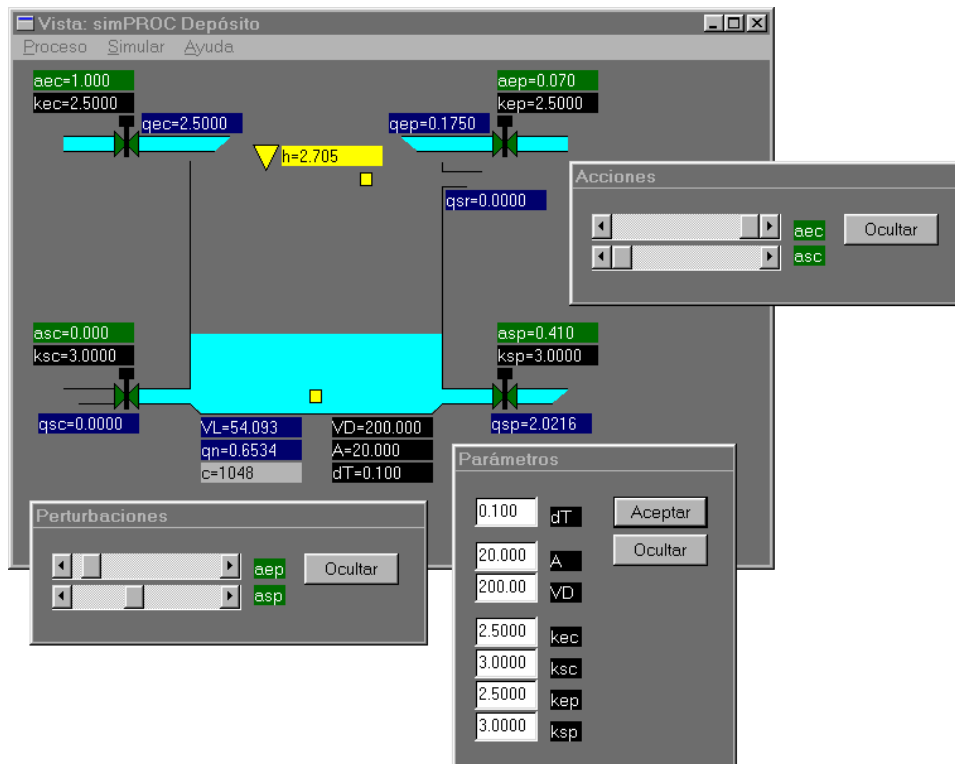


Figura 3. Vista de `simPROCdeposito` en acción

3. La versión actual, y después

Actualmente disponemos de un prototipo con simulación para la tarjeta de adquisición de datos ADLink PCI9112 y para el depósito de agua.

Para sacar partido a simPROCes deberemos desarrollar otros modelos de procesos, seguramente sobre los temas que tradicionalmente hemos estado empleando en la asignatura, como las expendedoras, los ascensores, etc.

Tener la posibilidad de trabajar con enlaces de red nos puede facilitar exponer procesos a los alumnos, sin informar de si se trata de maquetas físicas o de programas. Un registro de simulaciones de control remoto, podría aligerar el proceso de evaluación.

Los problemas de carga de la red sin embargo, dificultan la simulación de procesos de dinámica más rápida que la de nuestro depósito.

Referencias

- [1] Albaladejo; Perles; Domínguez; Hassan; Martínez; "Informática Industrial: de la Teoría a la Práctica". Póster. Barcelona: CIDUI'2000
ISBN 84-7653-752-2 Página 172
- [2] Domínguez; Albaladejo; Martínez; Hassan; Perles; "Informática Industrial para Ingenieros Técnicos en Electrónica Industrial". Donostia, San Sebastián: CUIE'2000
ISBN 84-7585-402-8 Páginas 619 a 632
- [3] Domínguez; Hassan; "Innovación Educativa en Informática Industrial". Madrid: JNIEI'1996
ISBN 84-88760-09-4 Páginas 659 a 668
- [4] Domínguez; Perles; Martínez; Hassan; "Especificación de Problemas Reales para la Enseñanza de la Informática Industrial". La Almunia de Doña Godina: JENUIT'1999
ISBN 84-605-9617-6 Páginas 45 a 53
- [5] Hassan; Martínez; Domínguez; Perles; "Herramientas Multimedia de Apoyo a la Docencia de la Asignatura Informática Industrial de la EUITIV". Huelva: CUIE'1999
ISBN 84-931043-0-2 Páginas 274 a 283
- [6] Hassan; Perles; Albaladejo; Martínez; Domínguez; "Diseño de las Páginas Web de las Asignaturas de Informática Industrial con Capacidad para Realizar Adquisición y Envío de Datos a Bajo Nivel Utilizando Lenguaje Java". Alcalá de Henares: JENUIT'2000
ISBN 84-8138-409-7
- [7] Martínez; Hassan; Domínguez; "Informática Industrial: Una Visión de su Enseñanza a través de la Innovación Educativa". Las Palmas de Gran Canaria: CUIE'1998
ISBN 84-89728-69-0 Páginas 413 a 419
- [8] Martínez; Hassan; Domínguez; Perles; "Un Modelo de Evaluación para la Asignatura Informática Industrial de la EUITIV". Huelva: CUIE'1999
ISBN 84-931043-0-2 Páginas 1611 a 1617
- [9] Perles; Domínguez; Martínez; Hassan; "Enseñanza de la Informática Industrial mediante Proyectos Reales Simplificados". La Almunia de Doña Godina: JENUIT'1999
ISBN 84-605-9617-6 Páginas 55 a 60
- [10] Perles; Hassan; Martínez; Domínguez; "Los Miniproyectos como Herramienta de Evaluación en la Asignatura Informática Industrial en la Escuela Universitaria de Ingeniería Técnica Industrial de Valencia". Valencia: JJD-UPV'1998
- [11] Perles; Martínez; Albaladejo; Domínguez; "SimSeny: Un Simulador Didáctico para Tarjetas de Adquisición de Datos". Barcelona: TAAE'2000
ISBN 84-600-9596-7 Páginas 493 a 496