



## MEMORIAS SOMI XV

# DISEÑO Y APLICACIÓN DEL CONTROLADOR DIGITAL *SPV*isor

R. Garibay\*, A. Rosas y V. Andrade  
Departamento de Ingeniería de Control, Facultad de Ingeniería, UNAM  
Ciudad Universitaria, D. F., C. P. 04510  
\*Fax: 52-5-56161855; e-mail: rgaribay@servidor.unam.mx

### Resumen

El programa *SPV*isor es el núcleo de un sistema de control digital en tiempo real, económico y accesible, en el cual, se realiza el esquema de **control por realimentación de estado** con un **observador de orden completo**, para plantas de una entrada y una salida, de hasta quinto orden. Ha sido desarrollado para satisfacer las necesidades de enseñanza y experimentación en los laboratorios de control digital de nivel licenciatura. Su enfoque didáctico permite consolidar el aprendizaje y motivar la creatividad en la aplicación de controladores lineales. Las características de su interfaz hombre-máquina brindan al usuario las facilidades para interactuar en línea con el sistema, permitiéndole introducir el valor del punto de ajuste y actualizar las matrices de ganancia que definen al controlador y al observador.

### Abstract

*SPV*isor is the core of a low-cost and easy-to-use computer controlled system, which applies the well-known strategy of state feedback control with full state estimation, to single input/output plants, up to 5<sup>th</sup> order. It has been developed as a simple solution for teaching and practicing, some of the most important topics in linear control theory. Its didactic focus allows a more efficient learning and increases creativity to solve control application problems. Its man-machine interface provides graphical display of the main variables and the facilities to interact with the system, on line, entering the set point value for the loop and updating, according to the design performed by the customer, the gain matrixes that constitutes the controller and the observer.

### 1. Introducción

En la actualidad, para lograr un adecuado dominio de la teoría y el diseño de sistemas de control, se requiere cubrir diversas etapas de aprendizaje y desarrollo, que van desde el planteamiento del problema por resolver y el modelado de la planta, pasando por el análisis, la simulación y la realización del sistema, hasta la aplicación real del esquema propuesto, las pruebas operativas y la valoración de los resultados experimentales.

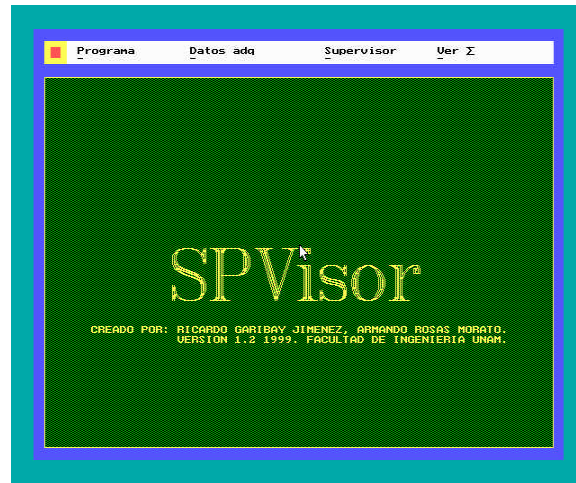
La parte teórica ha sido profusamente presentada en una gran cantidad de textos y artículos, y día con día se dan a conocer nuevos esquemas y algoritmos que resuelven una amplia variedad de problemas. A su vez, la simulación ha cobrado un gran auge debido al uso cada vez más extendido de la computadora personal y de paquetes matemáticos. Sin embargo, para consolidar el aprendizaje, motivar la creatividad y la generación de nuevos proyectos, es indispensable la aplicación de los algoritmos de control en plantas y procesos de laboratorio, probando bajo condiciones reales las soluciones propuestas, mismas que deben asegurar un desempeño confiable y válido de acuerdo con el planteamiento teórico correspondiente. También, es importante considerar que por medio de las prácticas se realimenta el proceso de aprendizaje, ya que éstas aportan la experiencia necesaria para mejorar, en lo sucesivo, el planteamiento, el análisis y el desarrollo de este tipo de proyectos.

El paquete de programación que se presenta, *SPV*isor, resuelve la necesidad de contar con dispositivos de laboratorio que hagan posible la realización de prácticas de control digital, a través



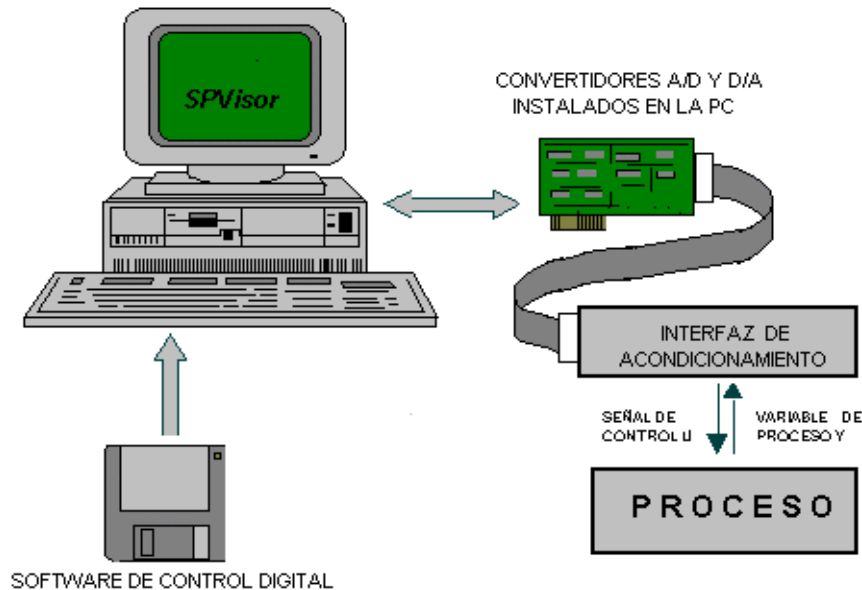
## MEMORIAS SOMI XV

de soluciones muy económicas, accesibles para el usuario y de acuerdo con los temas más importantes del control automático.



### 2. Arquitectura del sistema

En la figura 1 se muestran los componentes que constituyen el controlador digital.



**Figura 1. Componentes del sistema de control digital.**

#### 2.1. SPVisor

A partir de su estructura de programación basada en la metodología de programación orientada a objetos (**POO**), el paquete **SPVisor** permite el control de procesos continuos, por medio de una computadora personal, de acuerdo con los esquemas de **realimentación de estado y observador de orden completo**. Codificado en C++ se conecta con el proceso a través de una tarjeta de adquisición de datos **PCL-812**, con un rango de  $(-10,10)$  volts para los canales **A/D** y **D/A**. El **SPVisor** puede mantener bajo control una planta de orden  $n < 6$ , de una entrada-una salida,



## MEMORIAS SOMI XV

estimando ininterrumpidamente el estado, a pesar de que se le introduzca un cambio de parámetros del controlador, del observador o una breve historia del control de la planta. El despliegue de variables incluye las señales de control  $u(k)$ , la salida de la planta  $y(k)$ , la estimación de la salida  $\hat{y}(k)$  y el error de estimación  $\tilde{y}(k)$ , desplegando gráficas comparativas de estas señales en forma de barras y contra el tiempo.

En el esquema de retroalimentación de estado [3], la *regulación* de la planta se lleva a cabo por medio de la señal de control que se calcula de acuerdo con la ecuación (1), en la cual, se observa que es necesario disponer del estado  $x(k)$  en cada instante de muestreo. Sin embargo, la medición del estado no siempre es posible o recomendable, por lo que se recurre al empleo de un observador que proporcione un estimado  $\hat{x}(k)$  de dicho estado; el estimado se calcula por la ecuación (2). Para que este esquema se lleve a cabo, es indispensable contar con el modelo discreto de la planta, expresado genéricamente por las matrices  $A_d$ ,  $B_d$ , y  $C_d$ , la medición de la salida  $y(k)$  y las matrices de ganancia de control  $K_d$  y del observador  $L_d$ .

$$u(k) = K_d \hat{x}(k) \quad (1)$$

$$\begin{aligned} \hat{x}(k+1) &= A_d \hat{x}(k) + B_d u(k) + L_d [y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C_d \hat{x}(k) \end{aligned} \quad (2)$$

Sobre la base de los enunciados anteriores, se desarrolló otra de las características relevantes del paquete, el cual, no solo efectúa la función de regulación, sino también la de servo, es decir, opera bajo cambios de carga y cambios de referencia. También considera la muy importante variante de que la planta tenga o no polos en el origen, por lo cual, la matriz de control  $K_d$  debe incluir las siguientes posibilidades: si la planta es tipo 0, el término correspondiente a la ganancia de integración  $k_i > 0$ , lo que introduce un integrador en la trayectoria directa, según se muestra en la figura 2; si la planta es de tipo 1 o mayor,  $k_i = 0$ , con lo cual, esta parte del control no se realiza. Para aplicar el controlador se deberá tener muy clara esta diferencia al momento de calcular la matriz de control y declarar el valor de dicho término.

Lo anterior hace que la aplicación del controlador sea más versátil y completa, simplemente modificando la ley de control de acuerdo con el planteamiento de Franklin y Powell [2], mismo que se resume en las ecuaciones siguientes, donde la matriz  $N_x$  compensa el estado para que la salida se ajuste a los cambios de referencia:

$$u(k) = u_1(k) + u_2(k) \quad (3)$$

$$u_1(k) = K_d N_x r(k) - K_d \hat{x}(k) \quad (4)$$

$$u_2(k+1) = u_2(k) + k_i [r(k) - y(k)] \quad (5)$$

Para emplear el paquete, las matrices  $K_d$ , y  $L_d$  deben ser previamente calculadas tomando como base las especificaciones del desempeño deseado de malla cerrada; también deben ser calculadas la matriz  $N_x$  y el término  $k_i$ , de acuerdo con el método propuesto en [2]. Por su amplitud, el método no se detalla en este documento. Otros parámetros que también deben introducirse al iniciar la aplicación son: el período de muestreo  $T$  y el orden  $n$  del modelo del proceso. La figura 4 muestra la pantalla de configuración que da acceso a los elementos del modelo, del controlador y del observador que se han indicado arriba.

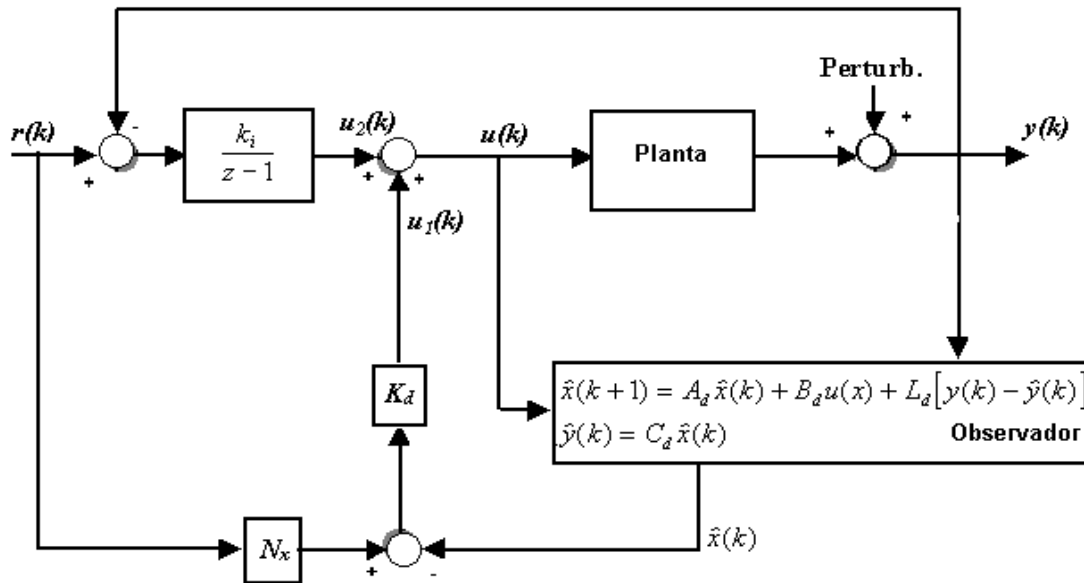


Figura 2. Esquema de control de SPVisor.

Con respecto a la especificación del comportamiento de malla cerrada, cabe destacar que si se consideran dos polos complejos como los dominantes, se pueden definir los parámetros  $M_p$  y  $tr$  como los representativos de la respuesta deseada; el resto de los polos no dominantes,  $n-2$ , se definen reales negativos. A partir de estas definiciones se calcula la matriz  $K_d$ , y por lo tanto, el desempeño en malla cerrada debe satisfacer la especificación de  $M_p$  y  $tr$ .

Para definir los polos del observador se sigue un criterio directo basado en la idea de la rapidez de la estimación, por lo cual, basta con definir valores reales negativos para los  $n$  polos del observador y con ellos calcular la matriz  $L_d$ . De acuerdo con este criterio la respuesta del observador debe satisfacer el requisito de la rapidez especificado.

El cálculo de las matrices  $K_d$  y  $L_d$  se efectúa de acuerdo con los métodos teóricos conocidos de asignación de polos en variables de estado, el más conocido de éstos es la Fórmula de Ackermann [4] para sistemas de una entrada, una salida.

### 3. Estructura y características del paquete

El programa se desarrolló en lenguaje **C++** mediante la metodología de programación orientada a objetos (**POO**) [5]. Esta metodología facilita el diseño estructurado a través de:

- El concepto de clase o de encapsulación.
- La herencia de clase
- Polimorfismo

Debido a las características **POO** de **SPVisor**, su grado de estructuración es alto, por lo que se logró disminuir el tiempo de desarrollo y se incrementó la capacidad de mantenimiento y de evolución, realizando eficientemente el ciclo *reducir-reusar-reciclar* código. La reducción se logró en la programación, la reutilización con el mantenimiento sencillo y el reciclado al momento de desarrollar las variantes de control de **SPVisor**. Este programa tiene una serie de características que proporcionan ventajas en su funcionamiento didáctico, como son: aplicación ininterrumpida de la acción de control, *interfaz sistema usuario* muy amigable, asignación dinámica de la memoria

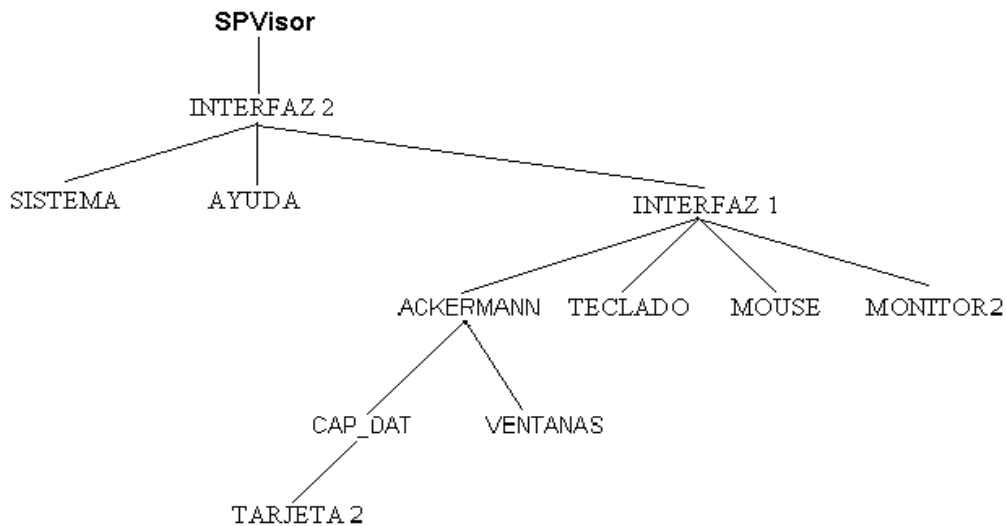


## MEMORIAS SOMI XV

para objetos y en general, para cualquier estructura de datos, opciones del sistema y ayuda durante la operación fuera de línea, así como la estructura del programa que facilita la localización de errores y la reducción del tiempo de compilación.

### 3.1. Estructura modular

La figura 3 presenta la estructura de programación de **SPVisor** con sus correspondientes reglas de herencia. A partir de los conceptos de programación orientada a objetos, se plantea la estructura modular formada por clases [1]. Cada clase representa un elemento que se diferencia de los demás y cuyas características son determinantes. El programas se organiza en cinco módulos principales descritos mas adelante.



**Figura 3. Estructura de programación de SPVisor**

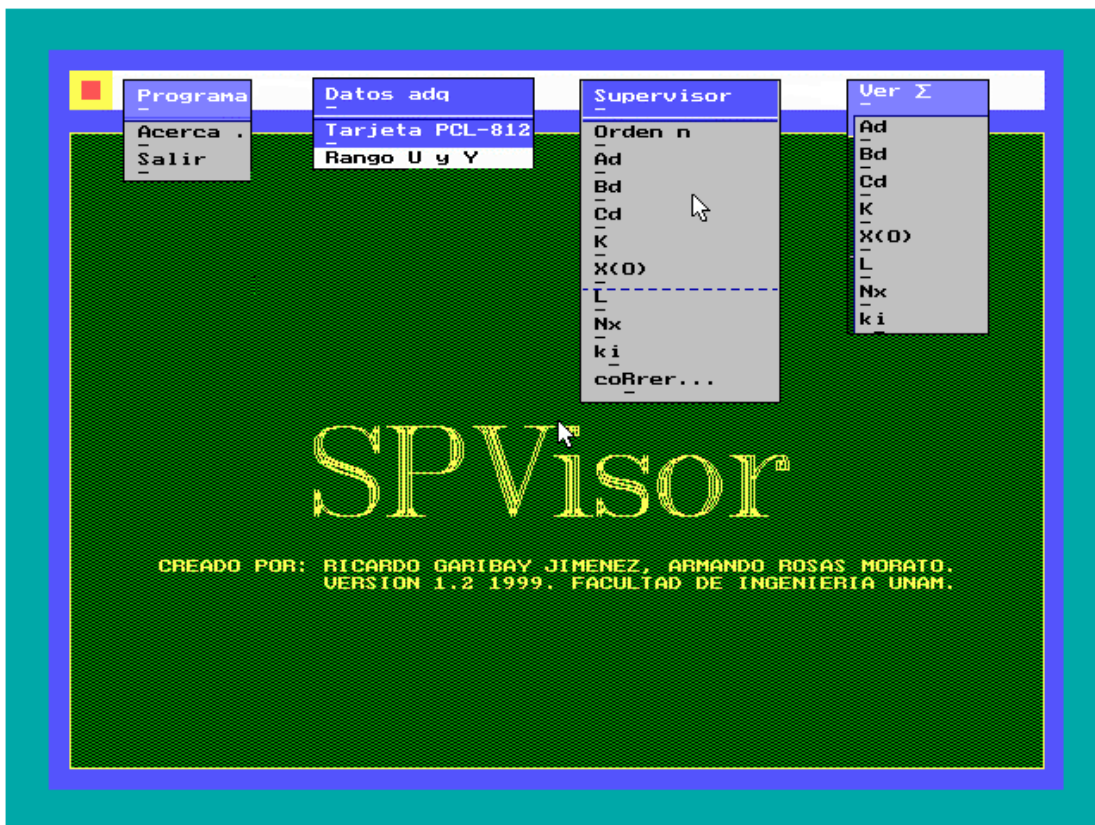
- (i) **De comunicaciones (Cap\_dat).** Encargado de la entrada/salida de datos del controlador; esto es, inicializar la operación de la tarjeta **PCL-812**, seleccionar canales **A/D** y **D/A**, deshabilitar su operación. La tarjeta fue configurada para lecturas y salidas analógicas en un rango de  $(-10, 10)$  volts.
- (ii) **De algoritmos de control (Ackermann).** Conjunta la iniciación y validación de los parámetros de los controladores, las ecuaciones en diferencias que definen a éstos y la traslación al tiempo de los valores de la señal de control.
- (iii) **De interfaces sistema usuario.** Hace uso de dispositivos de entrada/salida del usuario como son el monitor, teclado, y mouse. En primera instancia establece subrutinas de iniciación, aplicación y terminación para cada dispositivo; las de aplicación comprenden el uso de primitivos de gráficos para el monitor; de funciones dedicadas a la lectura y validación de datos numéricos y alfanuméricos del teclado; de funciones encargadas de informar la posición del mouse así como el censado de sus botones. Con las anteriores subrutinas, empleadas en forma coordinada, se ofrece al usuario una interfaz integral.
- (iv) **De apoyo al sistema.** Comprende subrutinas que verifican el estado de las unidades de disco, leen directorios, hacen cambios en ellos y copian archivos.
- (v) **De apoyo al usuario.** Consta de funciones que leen un archivo de disco y permiten desplegarlo en pantalla durante la operación fuera de línea. Este el archivo contiene información para el manejo del paquete.

Cada módulo tiene una estructura de clase formada por un nodo y varias raíces. La clase raíz reúne información específica del módulo, mientras que las clases nodo ó enlace facilitan la comunicación inter-módulos. Para su operación cada módulo requiere una serie de datos, mismos que son proporcionados por otro(s) módulo (s).

### 3.2. Características didácticas de aplicación

Durante la operación en línea la acción de control es ininterrumpida, las variables se despliegan en forma de barras y como gráficas contra el tiempo. La señal de control  $u$  se muestra en porcentaje y la salida  $y$  en volts. Empleando gráficos en VGA y el mouse, las interfaces son muy amigables.

La figura 4 muestra la pantalla de operación fuera de línea de **SPVisor**, con las diferentes opciones del menú principal; cada una de estas opciones habilita la configuración de los módulos del programa en donde el usuario debe definir parámetros o datos para el algoritmo de control. Un paso muy importante de la aplicación de **SPVisor** se tiene en esta pantalla, ya que a través de ella se introducen el modelo discreto de la planta y las matrices que conforman el esquema descrito en el punto 2.1 anterior. Lo que persigue esta forma de acceso es que el usuario efectúe, fuera de línea, todo el diseño de control y verifique su aplicación a través de **SPVisor**.



**Figura 4. SPVisor fuera de línea.**

La pantalla de operación en línea se muestra en las figuras 6 y 7, en donde se observa que se despliega un par de variables de interés de acuerdo con las combinaciones:  $u/y$ ,  $\hat{y}/y$ ,  $\tilde{y}/y$ ,  $\tilde{y}/\hat{y}$ . La combinación se selecciona en la ventana de la parte inferior izquierda de la pantalla, misma que permite introducir un cambio en el valor de la referencia  $r$ .

4. Pruebas de la aplicación de SPVísor

4.1 Procesos de prueba

Para efectuar la prueba de los controladores se ha empleado un “Simulador de Procesos”, de tipo electrónico, de la marca “Feedback”, el cual, proporciona un proceso continuo lineal de hasta tres etapas de retraso, de acuerdo con el diagrama que se muestra en la figura 5. Cada etapa puede simular un retraso de primer orden, con constante de tiempo de 1 seg., en el modo de operación “lag”, o un integrador en el modo de operación “integrator”, en ambos casos la ganancia del bloque es unitaria. Dispone también de un sumador que permite introducir una perturbación de carga “load disturbance”. El uso de las etapas, la aplicación de la carga y los modos de operación son a criterio del experimentador de acuerdo con sus propósitos experimentales.

En las pruebas se emplearon dos procesos: uno de segundo orden simple y uno de tercer orden con un polo en el origen. Abajo se muestra, para cada uno de ellos, la función de transferencia continua, la función de transferencia discreta y el modelo discreto de variables de estado. Los equivalentes discretos fueron obtenidos con el método de aproximación **ROC** y período de muestreo  $T = 0.1$  seg.

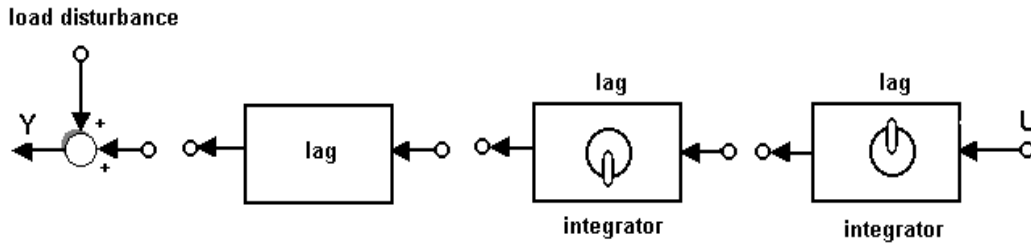


Figura 5. Simulador de Procesos.

- Planta de segundo orden:

$$G_p(s) = \frac{1}{(s+1)^2} \tag{6}$$

$$G_p(z) = \frac{0.0047(z+0.9355)}{(z-0.9048)^2} \tag{7}$$

$$A_d = \begin{bmatrix} 1.8097 & -0.8187 \\ 1 & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{8}$$

$$C_d = [0.0047 \quad 0.0044]$$

- Planta de tercer orden con un polo en el origen:

$$G_p(s) = \frac{1}{s(s+1)^2} \tag{9}$$

$$G_p(z) = \frac{(0.15858)10^{-3}(z+0.2548)(z+3.55)}{(z-1)(z-0.9048)^2} \tag{10}$$



## MEMORIAS SOMI XV

$$A_d = \begin{bmatrix} 2.8097 & -2.6284 & 0.8187 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (11)$$
$$C_d = 10^{-3} [0.1586 \quad 0.6035 \quad 0.1435]$$

### 4.2. Prueba del control por retroalimentación de estado con observador de orden completo en una planta tipo 0

En esta prueba se empleó el proceso de segundo orden definido por las ecuaciones (7) y (8), el cual, no tiene polos en el origen y por lo mismo debe calcularse e introducirse la ganancia de integración  $k_i$ , de tal forma que **SPVisor** genere la función de integración para lograr que el error de estado estable sea nulo ante cambios de referencia y cambios de carga. La inclusión de este polo determina que el diseño deba ser desarrollado sobre una planta aumentada de tercer orden. A continuación se enuncian la especificación y las matrices de diseño de esta prueba.

- Especificación. (12))

Polos dominantes:  $Mp = 0.2$  y  $tr = 2.2$  seg.

Otros polos:  $s_3 = -9$ .

Polos del observador:  $q_{1,2} = -15$ .

- Matrices del diseño: (13)

$$K_d = [0.5138 \quad -0.4434], \quad k_i = 1.3029, \quad L_d^T = [170.7 \quad 129.01], \quad N_x^T = [110.42 \quad 110.42]$$

El resultado del experimento se muestra en la figura 6, en la cual, la gráfica inferior, de color rojo, corresponde a la respuesta del sistema ante un cambio del valor de referencia  $\Delta r = 4$  y a la aplicación de una carga  $\Delta p = 2.5$ , en distintos momentos. En la misma gráfica se pueden medir los términos del comportamiento dinámico especificado; adicionalmente, se observa que el sistema logra un error de estado estable nulo gracias a la presencia del integrador incorporado por el controlador.

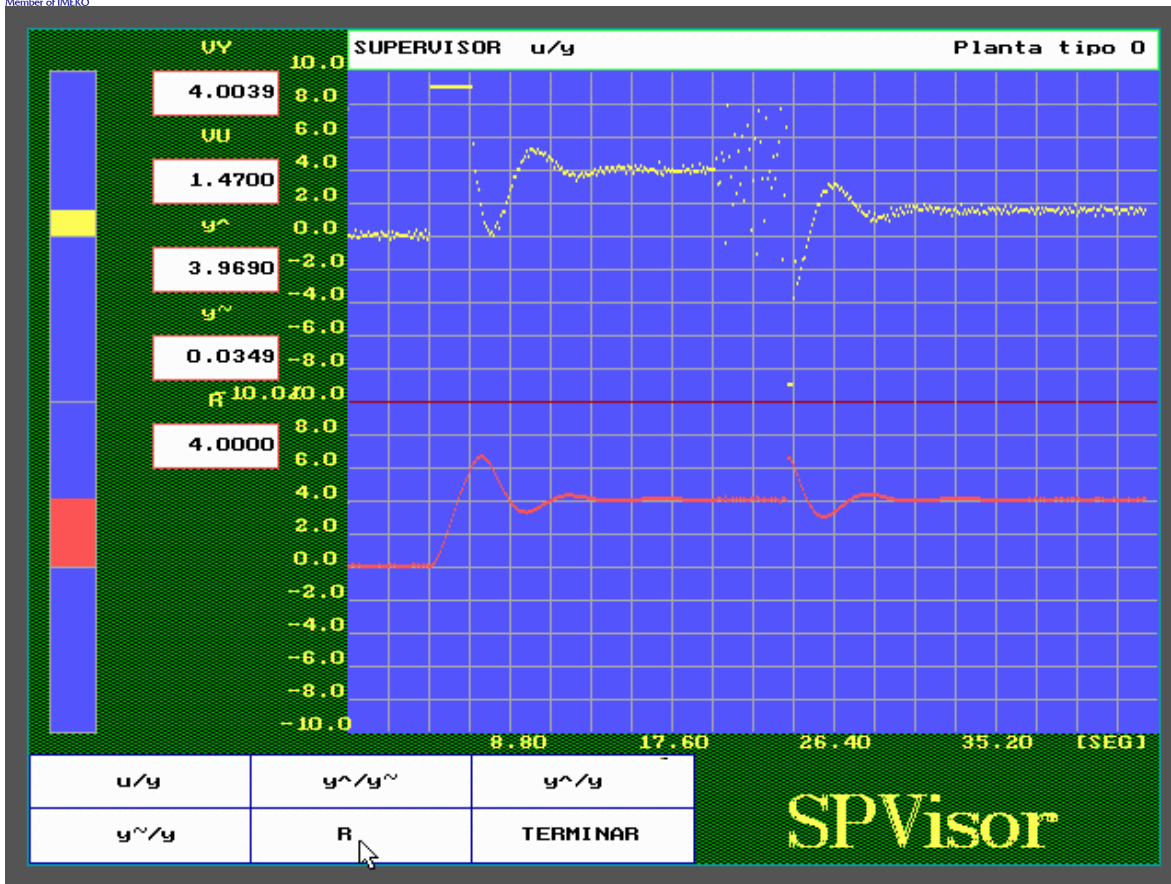


Figura 6. Resultado de la prueba 4.2.

#### 4.3. Prueba del control por retroalimentación de estado con observador de orden completo en una planta tipo 1

En esta última prueba se empleó el proceso de tercer orden con un polo en el origen, definido por las ecuaciones (10) y (11), por lo cual, **SPVIsor** genera la ley de control sin la función de integración, quedando un diseño de tercer orden. A continuación se enuncia la especificación y los elementos calculados para esta prueba.

- Especificación. (14)

Polos dominantes:  $M_p = 0.2$  y  $t_r = 2.2$  seg.

Tercer polo:  $s_3 = -9$ .

Polos del observador:  $q_{1,2,3} = -15$ .

- Matrices del diseño: (15)

$$K_d = [0.5043 \quad -0.9472 \quad 0.4491], L_d^T = [3841.7 \quad 2295.7 \quad 1014.4],$$

$$N_x^T = [1104.3 \quad 1104.3 \quad 1104.3]$$

El resultado de esta prueba se muestra en la figura 7, en la cual, la gráfica inferior, en color rojo, es la respuesta del sistema ante un cambio del valor de referencia  $\Delta r = 4$  y a la aplicación de una

carga  $\Delta p = 2.5$ . En esta respuesta se observa que el sistema logra un error de estado estable nulo gracias a la presencia del integrador de la planta; asimismo, se satisfacen las especificaciones de comportamiento dinámico respecto a los polos dominantes, dando un sobrepaso y un tiempo de levantamiento iguales a los especificados.

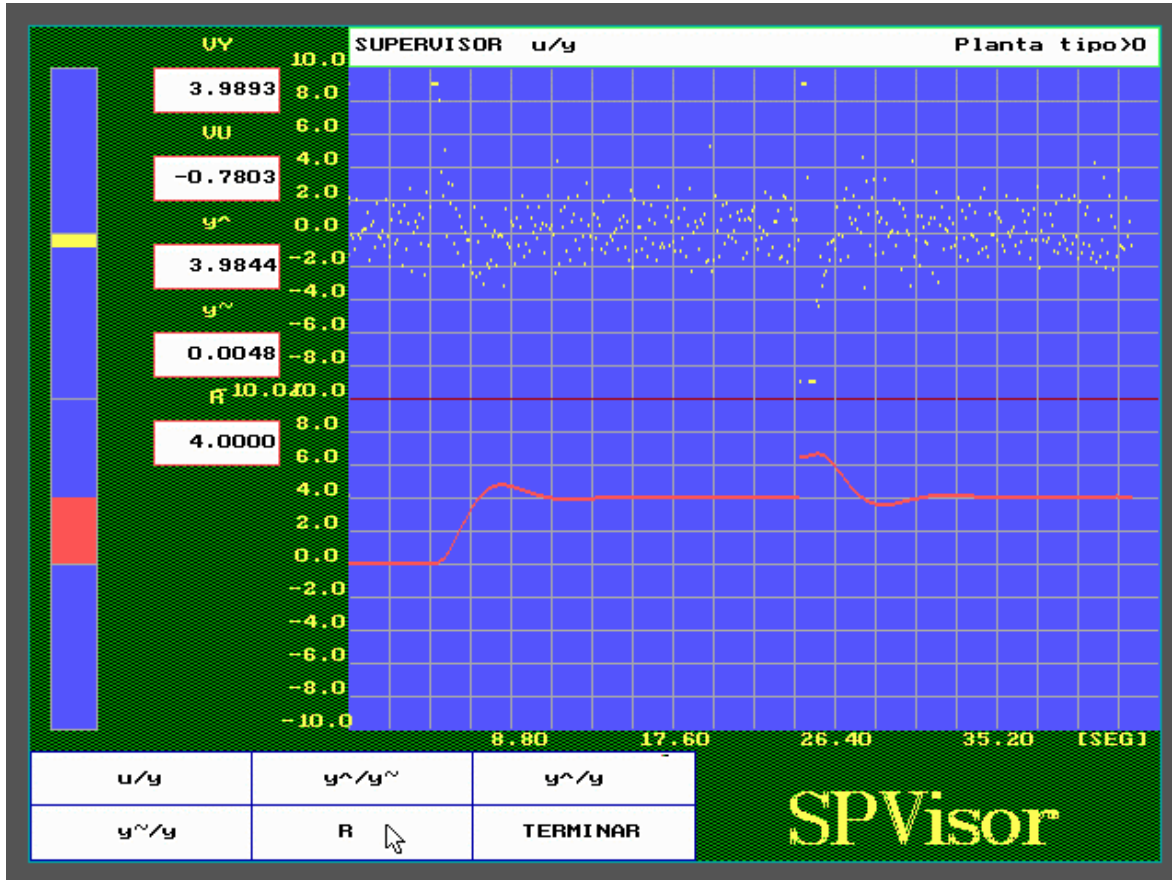


Figura 7. Resultado de la prueba 4.3.

## 5. Conclusiones

A través del presente trabajo se han probado diversos algoritmos de control digital en procesos físicos experimentales y se han comprobado las ventajas que brindan los controladores basados en **PC** para desarrollar con mayor eficacia la enseñanza de la teoría de control, realizar prácticas sencillas y didácticas y evaluar el desempeño de los controladores que se proponen. Es importante señalar que todas las aplicaciones fueron suficientemente probadas, lográndose desempeños de malla cerrada muy apegados a los que se plantearon teóricamente y muy similares a los obtenidos en las simulaciones correspondientes.

Con respecto a las ventajas de la realización de prácticas, se ha podido constatar, que en efecto, el aprendizaje se consolida enormemente, dando lugar a mejores perspectivas para un más amplio desarrollo de la experimentación con algoritmos de control.

El paquete **SPVIsor** opera sobre una plataforma de programación, que ha permitido ahorrar código y evolucionar desde un programa original sencillo hacia diversas aplicaciones. Esto hace que todos los controladores se empleen siguiendo un mismo procedimiento, a través de un solo tipo de pantallas, lo que facilita la realización de prácticas, ya que de esta forma, los usuarios no tienen



## MEMORIAS SOMI XV

que aprender, en cada ocasión, nuevos procedimientos de configuración y operación. Otra ventaja que ofrece este tipo de programas, consiste en que la adquisición de datos permite el despliegue de la dinámica de la planta y el procesamiento de los mismos con propósitos que van desde la verificación de los modelos, hasta la aplicación de algoritmos más sofisticados.

Con respecto a los resultados específicos de las pruebas, se observó que el desempeño del esquema de **control por realimentación de estado de una planta tipo 1**, se apegó muy aceptablemente al planteamiento teórico, lo cual, se comprobó al verificarse los valores de comportamiento dinámico especificados. Además, en este caso, los resultados experimentales tuvieron una gran similitud con los obtenidos en las simulaciones de **Matlab**. En contraste, la prueba del **control por realimentación de estado de una planta tipo 0**, en donde el controlador incorpora un “integrador” en la trayectoria directa, requiere de un análisis más detallado ya que el resultado experimental y el de la simulación mostraron un desempeño de malla cerrada muy similar entre ellos, pero desviado del planteamiento teórico, ya que en ellos no se verificaron exactamente las especificaciones de comportamiento dinámico.

También se comprobó que la aplicación del controlador **SPVisor** puede efectuarse sobre cualquier planta experimental o industrial, de una entrada, una salida, compatible con el rango de señales de  $-10$  a  $10$  volts. Esto hace crecer las perspectivas de uso del paquete, ya que en el ámbito industrial existe una gran cantidad de áreas de aplicación, en donde el uso de dispositivos de control es muy amplio.

Finalmente, es importante mencionar que a partir de la capacidad de evolución de **SPVisor**, se pueden instrumentar esquemas más avanzados como los relacionados con el campo de la identificación de procesos y el control adaptable.

### 6. Referencias

1. A. Rosas *Paquete de controladores digitales mediante programación orientada a objetos*, Tesis profesional, (Facultad de Ingeniería, UNAM, México), (1994).
2. G. F. Franklin, J. D. Powell y M. L. Workman, *Digital Control of Dynamic Systems*, 2<sup>nd</sup> ed., (Addison-Wesley, New York), (1990).
3. K. Ogata, *Ingeniería de Control Moderna*, 3<sup>a</sup> ed., (Prentice-Hall Hispanoamericana, México), (1998).
4. K. J. Astrom y B. Wittenmark, *Computer Controlled Systems: Theory and Design*, (Prentice-Hall, Englewood Cliffs N.J. ), (1984).
5. H. Schildt, *Turbo C/C++: The Complete Reference*, (Osborne Mcgraw-Hill, Berkeley, Ca), (1992).

### Apéndice A

Fórmulas para el cálculo de parámetros y coeficientes.

$$M_p = e^{-\frac{xp}{\sqrt{1-x^2}}}, \quad x = \frac{\sqrt{[\ln(M_p)]^2}}{p^2 + [\ln(M_p)]^2} \quad (A1)$$

$$t_r = \frac{p - \cos^{-1}(x)}{w_n \sqrt{1-x^2}}, \quad w_n = \frac{p - \cos^{-1}(x)}{t_r \sqrt{1-x^2}} \quad (A2)$$

**Apéndice B**

Resultados de la simulación en *Matlab* del controlador por realimentación de estado para las pruebas 4.2 y 4.3 respectivamente.

