TESIS

ANALISIS E IMPLEMENTACION DE UN SISTEMA DE CONTROL ADAPTATIVO EN TIEMPO REAL BASADO EN MICROCOMPUTADOR

por

Hilario López García

presentada en el

DEPARTAMENTO DE INGENIERIA ELECTRICA, ELECTRONICA, DE COMPUTADORES Y SISTEMAS

de la

UNIVERSIDAD DE OVIEDO

para la obtención del

Grado de Doctor Ingeniero Industrial

GIJON, SEPTIEMBRE DE 1989

CAPITULO 3

IMPLEMENTACION

3. IMPLEMENTACION.

3.1 ESTRUCTURA Y PRINCIPALES MODULOS DE UN SISTEMA DE CONTROL ADAPTATIVO EN TIEMPO REAL.

La implementación física del sistema de control adaptativo más supervisor, cuyo esquema se ha visto en la figura 2.9, conduce al diagrama de bloques genérico de la figura 3.1. En ella se ve que la señal de salida del proceso es analógica, obteniéndose la digital correspondiente con el conversor A/D después de ser medida la señal por un sensor.

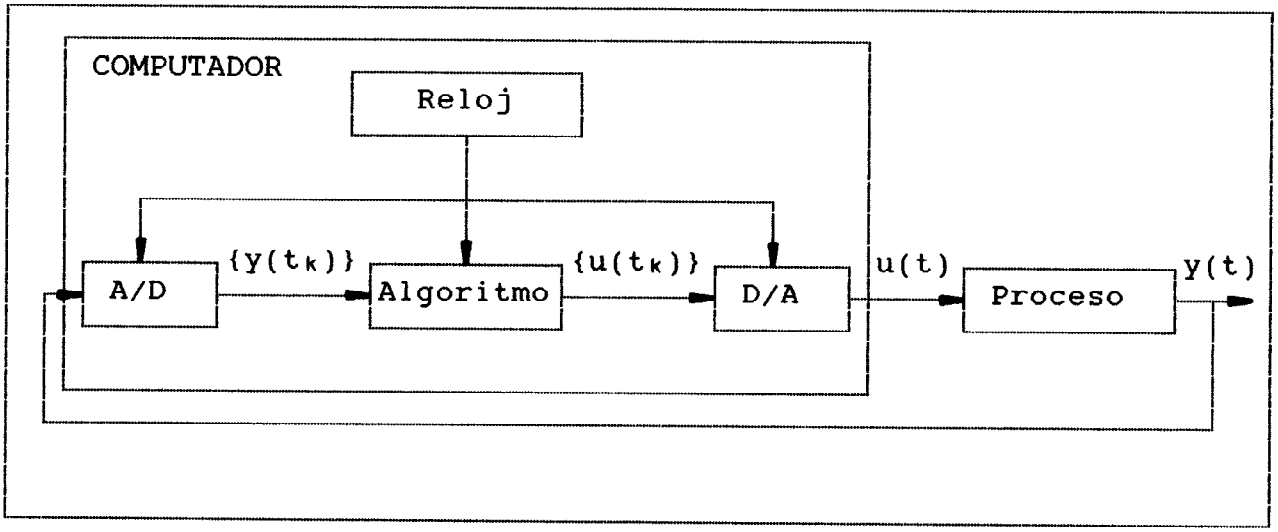


Fig 3.1. Diagrama esquemático de control por computador.

La conversión se realiza en los instantes de muestreo t_k. El computador interpreta la señal convertida, $\{y(t_k)\}$, como una secuencia de números, procesa la misma con los algoritmos necesarios para cada aplicación y produce una nueva secuencia $\{u(t_k)\}$.

Esta se convierte a analógica en el conversor D/A y se aplicará al proceso a través del accionador correspondiente. Como el conversor D/A debe producir una señal continua a su salida, normalmente trabaja dejando la señal de control, u(t), constante entre conversiones, con un funcionamiento de bloqueador de orden cero.

Cada uno de los eventos está sincronizado con el reloj de tiempo real del computador, aunque es necesario tener en cuenta que las operaciones a realizar en el computador consumirán un tiempo finito.

La implementación en el computador de los algoritmos necesarios (figura 3.2), cuando el sistema no es muy complicado, se suele realizar adoptando la típica estructura bitarea (background y foreground). Cada una de estas tareas consta de una serie de módulos que no son más que procedimientos que se ejecutan secuencialmente. El reloj se programa para que se obtenga una interrupción en cada instante de muestreo y el control del programa pasa al foreground. Cuando este finaliza, devuelve el control al background en el punto en que fue interrumpido.

Los módulos incluidos en un sitema de control adaptativo en tiempo real, pueden clasificarse en los siguientes grupos:

- Módulos de control adaptativo.
- Módulos de interface con el usuario.
- Módulos de interface con el proceso.

- Módulos de gestión de dispositivos.

Los datos necesarios para el interface entre los distintos módulos se agruparán en una base de datos inicializada por la fase de gestión de menús del interface con el usuario. Será necesario realizar una utilización correcta de esta base de datos para evitar situaciones que puedan ocasionar errores, por ejemplo, que el background tome datos de la misma para su representación mientras el foreground esté actualizando alguna de dichas variables.

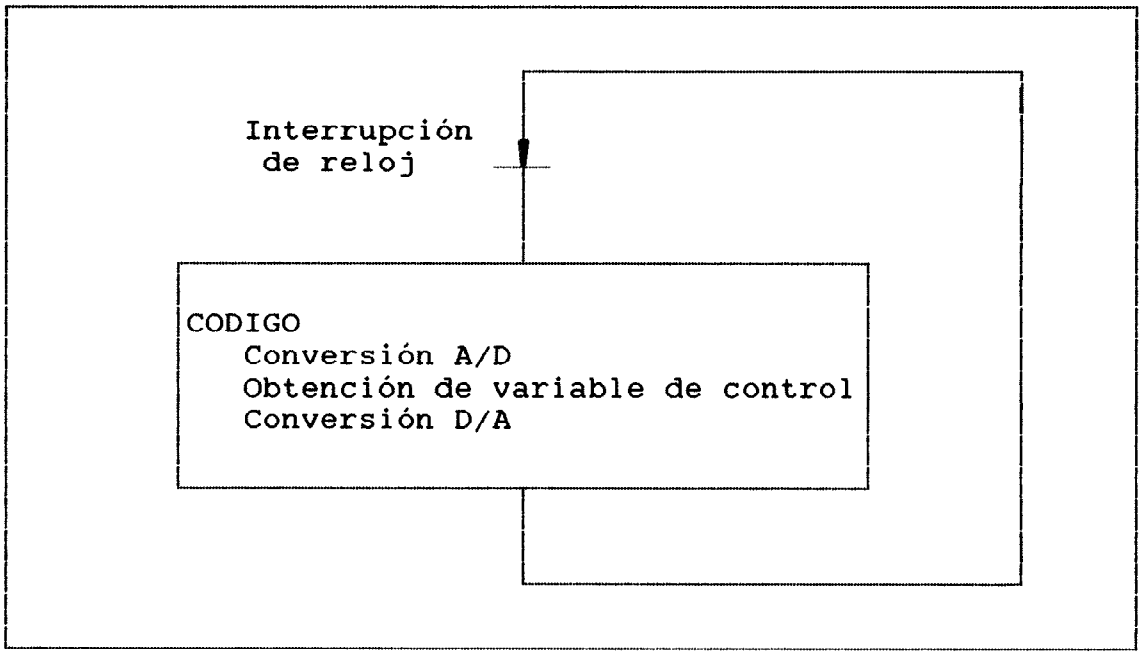


Fig 3.2. Implementación de un sistema discreto.

3.1.1 Módulos de control adaptativo.

Dentro de estos módulos se pueden incluir:

- Estimación de parámetros.

- Cálculo de los coeficientes del controlador.
- Obtención de la señal de control.
- Supervisión.
- Interface de parámetros.
- Fase de preidentificación.

El número y características de los módulos de control o estimación dependen de cada sistema en particular y la elección de la combinación de dichos algoritmos para una aplicación práctica depende del propio proceso a controlar, las perturbaciones existentes, restricciones de los actuadores, etc.

Los algoritmos que pueden incluirse en estos módulos se han visto en el capítulo anterior y la necesidad de inclusión de una fase de preidentificación previa al arranque del sistema, así como de un interface de parámetros entre los módulos de estimación y diseño del controlador, se justificará más adelante.

3.1.2 Módulos de interface con el usuario.

Dentro de estos módulos se pueden incluir:

- Gestión de menús en la inicialización.
- Intérprete de comandos interactivo.
- Intérprete de comandos batch.
- Gestión de alarmas.
- Funcionamiento manual/automático.

- Salida de datos en forma gráfica o numérica.

Este grupo de módulos tiene por objeto la gestión de todo lo que signifique relación entre el sistema y el usuario, desde la introducción de los parámetros iniciales o comandos durante el funcionamiento, hasta la gestión, en el nivel más cercano al usuario, de las representaciones de los resultados.

Así mismo pueden estar aquí comprendidas la definición por parte del usuario, detección y aviso por parte del sistema, de todo tipo de situaciones anómalas (alarmas), además de los algoritmos necesarios para el cambio de modo de funcionamiento de manual a automático y viceversa.

3.1.3 Módulos de interface con el proceso.

Dentro de estos módulos se pueden incluir:

- Conversión analógico/digital.
- Conversión digital/analógico.
- Gestión de E/S digital.
- Programación del reloj de tiempo real.
- Filtrado digital de perturbaciones y outliers.
- Módulos de definición de procesos para simulación.

El sistema de control adaptativo podrá disponer de opciones que permitan realizar simulaciones del comportamiento de diversos tipos

de procesos, además del control en tiempo real. Para ello se deben incorporar módulos de definición de dichos procesos, además de los de gestión de todos los dispositivos de bajo nivel necesarios para el interface entre un proceso físico y el computador.

Así se incluyen los algoritmos que permitirán la entrada y salida a este último tanto de señales digitales como analógicas, siendo preciso en este caso la incorporación, dependiendo de la frecuencia de muestreo, de algoritmos de filtrado digital. Así mismo será necesaria la incorporación de algoritmos de programación del reloj en tiempo real para que sincronice el funcionamiento de las tareas que se ejecutan en el sistema.

3.1.4 Módulos de gestión de dispositivos.

Dentro de estos módulos se pueden incluir todos los algoritmos necesarios para el control de los dispositivos empleados en el interface con el usuario (teclado, pantalla, impresora, etc.).

El software estará estructurado generalmente en varios niveles:

- Nivel cercano al usuario (alto nivel). En él estarán incluidos los algoritmos de representaciones gráficas o numéricas comprendidas en el grupo de interface con el usuario.

- Nivel cercano al dispositivo (bajo nivel). En él estarán incluidos los drivers de los distintos dispositivos que se

emplean para las representaciones.

De esta forma los algoritmos del nivel superior utilizan los del inferior, con lo que un cambio de dispositivo de representación supone sólo un cambio en el nivel bajo.

Esto mismo que se suele realizar en el interface con el usuario puede realizarse en el interface con el proceso, aunque en este caso no es tan usual. Esto es debido a que los algoritmos del nivel alto de éste último, simplemente suponen tomar o entregar un dato a los de nivel bajo y no una compleja gestión de menús o de representaciones gráficas o numéricas como en el primero.

3.2 INTERFACE CON EL USUARIO.

Cuando se piensa en control adaptativo, muchas veces aparece la idea de una caja negra, de un sistema sin parámetros que ajustar, que dé lugar a un buen funcionamiento en cadena cerrada nada más efectuar su conexión.

Hoy en día, esta solución está aún lejana y se cuestiona que sea la apropiada, ya que siempre debe ser posible, al menos, indicar al controlador lo que se espera que haga. De esta manera, sólo para aplicaciones especiales puede ser posible y deseable disponer de un controlador sin parámetros de ajuste y usualmente es necesario proporcionarle bastante información a priori.

De todas formas, es importante la determinación del conjunto de parámetros que es necesario ajustar. Existen dos categorías de parámetros en controladores adaptativos, enteros y reales. En cuanto a los primeros es necesario incluir siempre la definición del orden y tiempo muerto del modelo del proceso a controlar, siendo su valor indicado por el usuario o estimado automáticamente, como se indica en el apartado 3.6.3.

Los parámetros reales servirán para indicar el funcionamiento deseado del sistema en bucle cerrado así como valores iniciales en los procedimientos de identificación (factor de olvido, matriz de covarianza, etc.) pudiendo igualar los valores iniciales de los parámetros estimados a cero, cuando no se tiene ningún conocimiento previo acerca de su valor. Una discusión de la elección de parámetros en controladores adaptativos se realiza en Isermann y Lachmann (1982), Wellstead y Zanker (1982), Wittenmark y Astrom (1984)...

Debido a lo anteriormente indicado, el interface con el usuario se convierte en una parte muy importante de un sistema de control en tiempo real y, más aún, cuando pensamos en la aplicación de éste en labores de formación (en universidades o empresas), como son las de enseñanza y entrenamiento en control de procesos por computador.

El interface con el usuario ha de ser pues fácil de usar, sobre todo en la fase de inicialización del sistema, para lo cual el usuario podrá ser guiado por una estructura de menús. Así mismo, deberá permitirle, durante el ciclo normal de control, el cambio de

consignas u otras especificaciones de funcionamiento.

Igualmente, el interface debe dar opción al usuario a decidir la realización o no de una fase de preidentificación antes del comienzo del ciclo de control. En caso de no efectuarla debe ser posible visualizar y modificar los valores por defecto obtenidos de la última realizada.

Para cumplir estos objetivos en las tres etapas, es conveniente dividir funcionalmente el interface en otros tantos modos: modo menú, modo preidentificación y modo comando.

Por último, es conveniente disponer de mecanismos de detección de anomalías en la aplicación del sistema de control a un determinado proceso. La reacción ante las más importantes debe incluir, entre otras acciones que son competencia del supervisor, un aviso al usuario del mal funcionamiento ocurrido (competencia del interface a través de la gestión de alarmas). Así mismo es frecuentemente deseable tener la posiblilidad de control manual en el sistema.

3.2.1 Modo menú.

En este modo de funcionamiento, el interface debe ayudar al usuario (mediante menús, indicación de valores por defecto, etc.) a que introduzca las condiciones operativas, previamente al control o identificación del proceso.

Con los datos introducidos por el usuario se elabora una base de datos de inicialización, en la que deben estar contenidos, entre otros:

- Modo de operación del sistema (simulación, ejecución en tiempo real, control fijo o adaptativo, ...).
- Algoritmo de control a emplear.
- Parámetros de control.
- Parámetros de identificación.
- Función de transferencia del proceso, en caso de simulación.
- Parámetros para las representaciones gráficas o numéricas.
- Parámetros de la preidentificación (orden del proceso, período de muestreo...), que son valores por defecto necesarios únicamente en caso de no llevarse a cabo posteriormente la ejecución de dicha fase.

Es prioritario, con el interface trabajando en este modo, su facilidad de uso. Para conseguir este objetivo, la entrada de datos por parte del usuario se puede simplificar con diversas técnicas:

- La más importante es la del empleo de una estructura arborescente de menú principal y varios submenús para la elección de parámetros a visualizar o modificar. Las opciones se seleccionan fácilmente, generalmente mediante un ratón o con las teclas de movimiento del cursor.

- Antes de la inicialización de cada submenú o del menú princi-

pal, el valor de cada una de las variables que se visualizarán puede ser leído de un fichero descriptor. Al presentar al usuario estos valores por defecto, los cuales memorizan el último cambio efectuado, la configuración necesita, en la mayoría de los casos, sólo unas pocas modificaciones fáciles de realizar.

- En la pantalla se exhibe la información actual de las variables modificables, de forma que con el movimiento del cursor se selecciona cualquiera de ellas y, con sólo la pulsación de una tecla, el usuario deja como contenido de la misma su valor por defecto y se pasa a la siguiente.

- Es posible efectuar así mismo en esta etapa un chequeo de los datos numéricos introducidos, tanto de su formato como de si su valor está comprendido entre los límites definidos para cada opción. Si existe algún error, el valor por defecto no se altera y el usuario podrá introducir de nuevo el dato deseado.

- En cada opción puede disponerse de información de ayuda, que facilite la introducción de datos por parte del usuario no experimentado.

- No es necesario que el usuario visualice el contenido de todos los submenús, si no quiere efectuar cambios, ya que en aquellos casos en que el fichero descriptor no haya sido leído para su presentación en pantalla, al finalizar el funcionamiento en

este modo, es posible leer los valores por defecto incluidos en los mismos.

En cuanto a la representación de los resultados producidos, estos pueden ser observados durante el funcionamiento normal del ciclo de control y además, dependiendo de las opciones elegidas, ser almacenados para su posterior volcado por impresora o plotter. Así mismo la visualización puede ser efectuada de forma numérica o gráfica, pudiéndose seleccionar las variables a representar en cada período de muestreo.

La representación gráfica podrá constar de varias ventanas para la representación de las variables. En el eje de abscisas se representa el número de período de muestreo en cada instante y en el de ordenadas distintas variables (consigna,variable de control, variable controlada, etc.).

Además de la visualización del valor actual de las variables, se puede reservar en la pantalla un número reducido de líneas para la interacción con el intérprete de comandos en tiempo real.

3.2.2 Modo preidentificación.

Cuando el sistema entra en la fase de preidentificación, si el usuario lo decide así en el modo menú, ésta se realiza automáticamente. Evidentemente, la ejecución de esta fase tiene una gran aplicabilidad en el caso de control de un proceso físico en tiempo

real, cuya función de transferencia sea desconocida.

El interface con el usuario se reduce a la visualización de una pantalla de presentación de resultados finales en esta fase (período de muestreo inicialmente aconsejado, orden y tiempo muerto del proceso, etc.) pudiendo incluir así mismo los resultados parciales que conducen a los mismos (como, por ejemplo, los errores cometidos en la identificación suponiendo distintos órdenes para la función de transferencia del proceso).

Los resultados pueden ser visualizados en el momento de su obtención, existiendo siempre la posibilidad de la modificación posterior de todos o alguno de ellos, si el usuario lo cree conveniente, antes de efectuar el control del sistema en tiempo real.

En caso de no requerir modificación, el sistema pasa automáticamente al modo de funcionamiento de control en tiempo real y el interface con el usuario al modo comando.

3.2.3 Modo comando.

Cuando el sistema está controlando la planta en tiempo real, el diálogo con el usuario se suele efectuar a través de un intérprete de comandos. Esta solución es la más indicada debido a que el número de operaciones a efectuar y parámetros a ajustar es relativamente reducido, con lo que no es necesaria la utilización de menús.

Además, simultáneamente al diálogo con el operador, se exhiben los resultados producidos en forma gráfica o numérica, con lo que es necesario utilizar la mayor parte posible de la superficie útil para tal fin. Para la zona de diálogo se utiliza así un número reducido de líneas.

El intérprete de comandos deberá permitir al usuario cambiar en tiempo real (durante la ejecución y sin interrumpir el ciclo normal de control) un subconjunto de los párametros especificados en el modo menú. Así mismo, durante la operación en tiempo real, es necesario refrescar cada período de muestreo las gráficas y datos numéricos visualizados en la pantalla, mientras el intérprete traslada un conjunto de nemónicos a acciones del sistema de control. De esta forma, se podrán observar en tiempo real el efecto de los cambios por medio de las salidas gráficas o numéricas.

El intérprete puede aceptar una línea como entrada y realizar la partición en los distintos comandos, subcomandos o parámetros, según el formato:

comando [subcomando] [parámetro 1, parámero 2 ...]

Esto permitirá, junto con las técnicas que se especifican a continuación, simplificar la entrada de comandos por parte del usuario:

- Cualquier forma corta del comando no-ambigua es aceptada por el intérprete.
- Existencia del comando "ayuda" que indica al usuario sus posibilidades de actuación.
- Comprobación de que el comando introducido está incluido en la tabla de comandos, así como que su sintaxis es correcta.
- Aceptación de los nuevos valores de los parámetros introducidos después de la comprobación de los límites para ellos establecidos.
- Interrupción del proceso del intérprete cuando se detecte algún error y exhibición de un mensaje indicando su causa.

Así mismo, la tabla de comandos debe ser fácilmente modificable y ampliable mediate editor de textos. También es posible la fácil incorporación de otras utilidades:

- Posibilidad de realizar registros de todos los comandos procesados, de forma que, cuando el sistema falla se pueda investigar la causa.
- Trabajo en modo batch, con la ejecución de los comandos contenidos en un fichero. El instante de ejecución de cada comando puede especificarse en forma absoluta (tiempo medido desde el inicio) o relativa (desde la ejecución del último comando).
- Redireccionamiento de las salidas de errores y/o alarmas hacia ficheros, de forma simultánea o no a la visualización por la terminal.

3.2.4 Gestión de alarmas.

La gestión de anomalías en el funcionamiento del sistema de control en tiempo real es un tema complejo que incluye varios aspectos: definición, detección, aviso al usuario y reacción (bien de éste, o del propio sistema la mayoría de las veces).

Los diferentes casos de anomalías que se pueden producir son dependientes de cada aplicación particular, pero siempre se puede establecer de forma general una tabla con los límites permitidos para cada una de las variables que intervienen. Esta tabla debe ser fácilmente modificable para adaptarla a la aplicación que se esté llevando a cabo.

La detección se efectuará en tiempo real comparando el valor de las variables con el contenido en dicha tabla, o bien más directamente, vía interrupción, en alguno de los casos.

Se deberán definir así mismo las situaciones que serán corregidas por el propio sistema y aquéllas que necesiten la intervención del usuario, así como los mecanismos de aviso al mismo (software o hardware) en los casos en que sea indicado.

3.2.5 Control manual.

Una sencilla forma de implementar control manual en el sistema es

la que se indica en la figura 3.3. En ella se ve que es posible ajustar manualmente el valor de la variable de control, incrementándolo o decrementándolo con pulsadores. Es útil así mismo disponer de un método que permita variar la velocidad de dicha modificación.

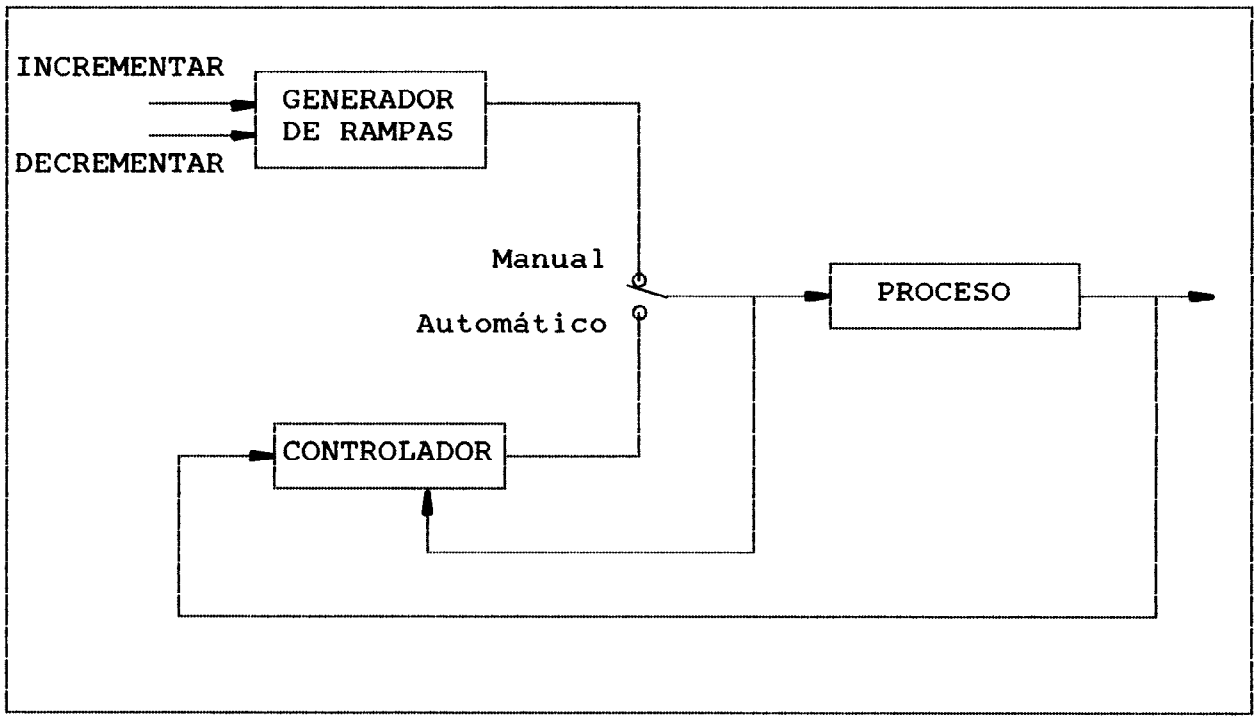


Fig 3.3. Control del sistema en los modos manual y automático.

Puede existir un problema en la aplicación práctica del sistema anterior, cuando se cambia el modo de funcionamiento de manual a automático. Esto sucede ya que, debido a la dinámica del controlador, su estado debe tener el valor correcto cuando se efectúa el cambio. Si no es así ocurrirá una transición indeseable que puede evitarse con diversas técnicas para dar lugar a una transición más suave (bumpless transition).

Esto se realiza frecuentemente introduciendo un modo de seguimiento que ajuste el estado del controlador de tal forma que sea compatible con sus entradas y salidas. Este modo de seguimiento se realiza automáticamente en sistemas que tengan prevista la posible saturación de los actuadores haciendo simplemente:

$$U_{min} = U_{max} = U_{manual}$$

lo que implica que la señal de control es siempre igual a la señal de entrada manual.

Así mismo, en un sistema de control adaptativo, es necesario incluir aún en el modo manual, la estimación de parámetros del modelo del proceso y el ajuste de los coeficientes del controlador, de forma que se obtenga una transición correcta posteriormente al modo automático.

3.3 FILTRADO DE PERTURBACIONES.

Existen varias fuentes de perturbaciones de alta frecuencia en un sistema de control, como son las producidas en la planta a controlar y en el proceso de medida. En este último caso se trata de ruido de los sensores, vibraciones o interferencias eléctricas en la transmisión de la señal medida.

En los sistemas de control o de adquisición de datos se requerirá

así la medición de señales contaminadas por ruido, siendo preciso la inclusión en ellos de los métodos de filtrado apropiados para separar la señal del ruido.

3.3.1 Efecto aliasing.

Los ruidos de alta frecuencia, no tienen en general influencia significativa en el funcionamiento de los dispositivos de control analógicos, debido a su comportamieto paso-bajo general. Sin embargo, en el caso de los sistemas digitales, el ruido es muestreado y transmitido. Por esta razón debe eliminarse, en la medida de lo posible, la influencia de las fuentes de ruido existentes, así como realizar un filtrado de las señales, a la entrada del computador digital.

Considerando la señal analógica procedente del captador como:

$$y(t) = s(t) + v(t) \tag{3.1}$$

siendo v(t) el ruido y s(t) la señal no perturbada, si se efectúa el muestreo de y(t) con una freuencia:

$$w_m = 2\pi / T_m \tag{3.2}$$

el teorema de Shannon indica que si $w_{max} > w_m/2$, siendo w_{max} la máxima frecuencia de la señal, el espectro de frecuencias de las bandas laterales se superpone al básico y la señal continua no

podrá ser reconstruida sin error.

Así, si un ruido v(t) de alta frecuencia está contenido en la señal medida, el espectro lateral se superpondrá al básico. Si el ruido es de frecuencia w_r con:

$$w_m/2 < w_r < w_m \qquad (3.3)$$

se genera después del muestreo con frecuencia w_m una componente de baja frecuencia:

$$w_2 = w_m - w_r \qquad (3.4)$$

con la misma amplitud, según se muestra en la figura 3.4. Esto es conocido como efecto aliasing y, para evitar su aparición, es necesario que las señales contaminadas con ruido con densidades espectrales significativas para $w > \pi/T_m$, sean filtradas antes de ser muestreadas (filtrado previo).

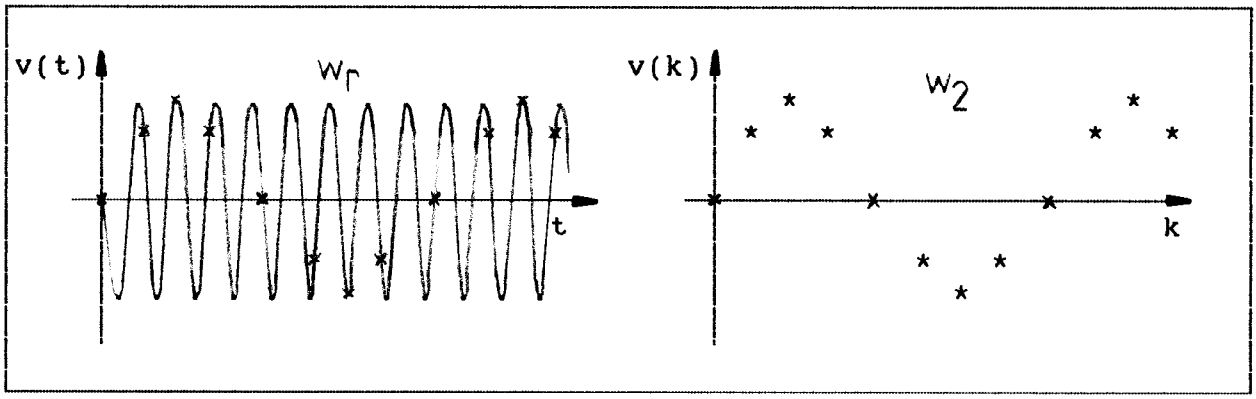


Fig 3.4. Efecto aliasing.

3.3.2 Filtrado previo.

Prácticamente todos los sensores analógicos disponen de alguna clase de filtro, pero éste es rara vez escogido en función de un problema particular de control. Sin embargo, dado un periodo de muestreo determinado T_m, siempre es posible realizar la modificación del filtro para que las señales obtenidas no tengan frecuencias superiores a la de Nyquist.

Una solución más simple es la del empleo de un filtro analógico antes del muestreo, cuyo ancho de banda sea inversamente proporcional a T_m y dependiente del orden del filtro, así como del carácter de la señal medida. Un inconveniente para la aplicación de este método es que el filtro analógico debe ser cambiado cada vez que se emplee un T_m distinto en el sistema.

Una variante para lograr que el cambio en T_m implique únicamente cambios de software es el empleo del prefiltrado analógico utilizando un T_m de 1 seg. o menor, aplicando posteriormente a la señal muestreada un filtrado digital.

Es necesario tener en cuenta la inclusión de la dinámica del filtro, además de la del captador, en el diseño del sistema de control. Si el filtro o la frecuencia de muestreo cambian, se deben calcular de nuevo los coeficientes de la ley de control.

3.3.3 Filtrado analógico.

Los filtros analógicos usados antes del muestreo deben tener un comportamiento paso-bajo con suficiente atenuación a la frecuencia $w=w_m/2$ (entre -20 y -40 dB dependiendo de la amplitud del ruido). En general, un filtro paso bajo de orden n puede ser representado por:

$$G_F(jw) = \frac{1}{(1+jwT)^n} = \frac{1}{(1+j\Omega_e)^n} \quad \text{con } n = 1,2,3,... \qquad (3.5)$$

siendo $\Omega_e=w/w_e=wT$ la frecuencia normalizada y w_g la frecuencia a la cual la amplitud cae -3 dB. Un filtro que responda a la ecuación anterior puede ser obtenido conectando redes de primer orden en serie.

Filtros especiales paso-bajo son, por ejemplo, los de Butterworth, Bessel, ITAE (Integral Time Absolute Error) y Tschebyscheff, que responden a la función de transferencia:

$$G_F(s) = \frac{w^2}{s^2 + 2\zeta ws + w^2} \qquad (3.6)$$

para orden 2, donde los parámetros varían según el filtro. Los filtros de orden superior se obtienen conectando en cascada varios de los anteriores.

En la tabla 3.1 se observa la constante de amortiguamiento f, la frecuencia w para algunos de los filtros citados (Astrom y Wittenmark, 1984) con w_0 como frecuencia natural deseada del filtro. En la figura 3.5 se observa la respuesta en frecuencia de varios filtros de orden 4.

Los filtros simples diseñados únicamente en base a elementos pasivos (redes RC) están especialmente indicados en el rango de altas frecuencias (f_0 > 5 Hz). Para frecuencias inferiores se requieren filtros activos diseñados en base a amplificadores operacionales y redes RC (0.1 Hz < f_0 < 5 Hz).

Esto lleva a utilizar filtros pasivos, considerando un amortiguamiento de 20 dB y orden 2, para periodos de muestreo T_m < 0.03 seg. y activos para 1.5 seg > T_m > 0.03 seg.

3.3.4 Filtrado digital.

El ruido de baja frecuencia debe ser tratado con filtros digitales, ya que los filtros analógicos para frecuencias inferiores a 0.1 Hz resultan excesivamente caros.

La función de transferencia en S de un filtro paso bajo de primer orden será:

$$G_F(s) = \frac{1}{1 + Ts} \qquad (3.7)$$

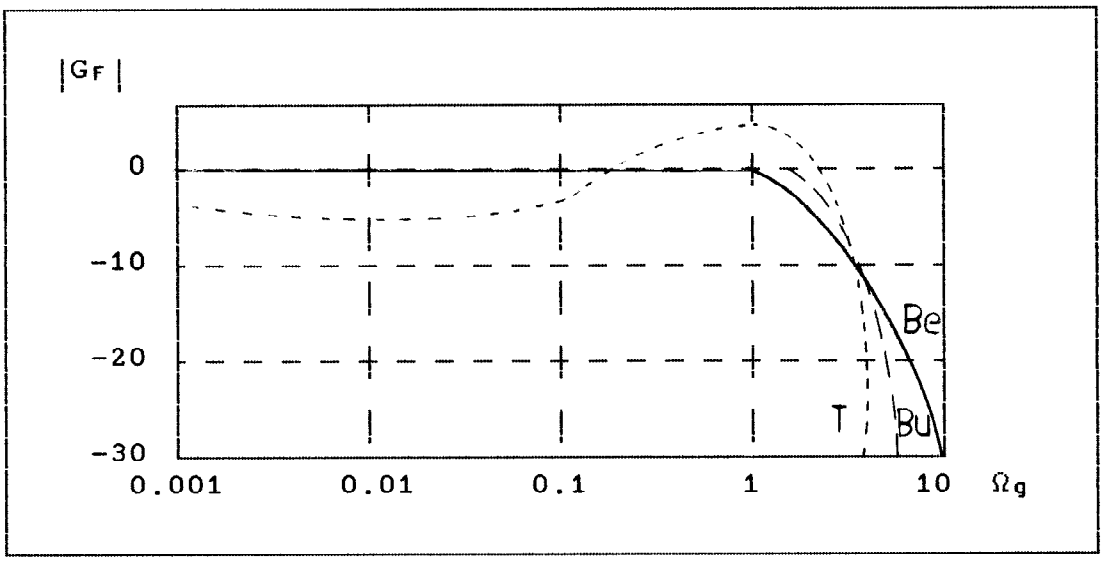


Fig 3.5. Respuesta en frecuencia de varios filtros paso-bajo.

	BUTTERWORTH		ITAE		BESSEL	
Orden	w/w0	f	w/w0	f	w/w0	f
2	1	0.71	1.00	0.71	1.00	0.87
4	1	0.38	1.33	0.32	3.39	0.62
	1	0.92	0.75	0.83	3.02	0.96
6	1	0.26	1.30	0.32	5.14	0.49
	1	0.71	0.98	0.60	5.57	0.82
	1	0.97	0.79	0.93	4.34	0.98

Tabla 3.1. Características de algunos filtros analógicos.

la cual, usando la transformada en z, se convertirá en:

$$G_F(z) = \frac{b}{1 + a_1 z^{-1}} \qquad (3.8)$$

con $a_1=-e^{-T_m/T}$ y $b=1+a_1$ para tener ganancia unitaria.

En la figura 3.6 se representan las amplitudes de los filtros continuo y discreto de primer orden, donde se observa que este último presenta el primer mínimo para $wT_m=\pi$. Esto implica que los filtros discretos no están indicados para filtrar señales con mayor frecuencia que la de Nyquist (para éstas deben usarse filtros analógicos).

3.3.5 Filtrado de outliers.

Además del ruido de medida, pueden existir otras causas de error en el mecanismo de lectura de datos por parte del computador, como son perturbaciones en el sensor o en la línea de transmisión, mal funcionamiento de instrumentación o errores de conversión. Estas causas de error se caracterizan por producir grandes desviaciones en la señal medida con una probabilidad de ocurrencia muy baja. A estas desviaciones se les denomina outliers.

Debido a las características citadas, es muy importante proveer al sistema de control de mecanismos para la detección y eliminación de estos outliers, antes de que influyan en el control. En sistemas con muy altos requerimientos de fiabilidad la detección de los outliers puede generalmente efectuarse en la fuente del error,

incorporando dos o incluso tres sensores para la medición de una misma señal. En el segundo caso se detecta fácilmente cual falla si los otros dos dan la misma medida. En el primer caso se incorpora una cierta lógica que produce una alarma cuando la diferencia entre las medidas dadas por los sensores supera un cierto valor.

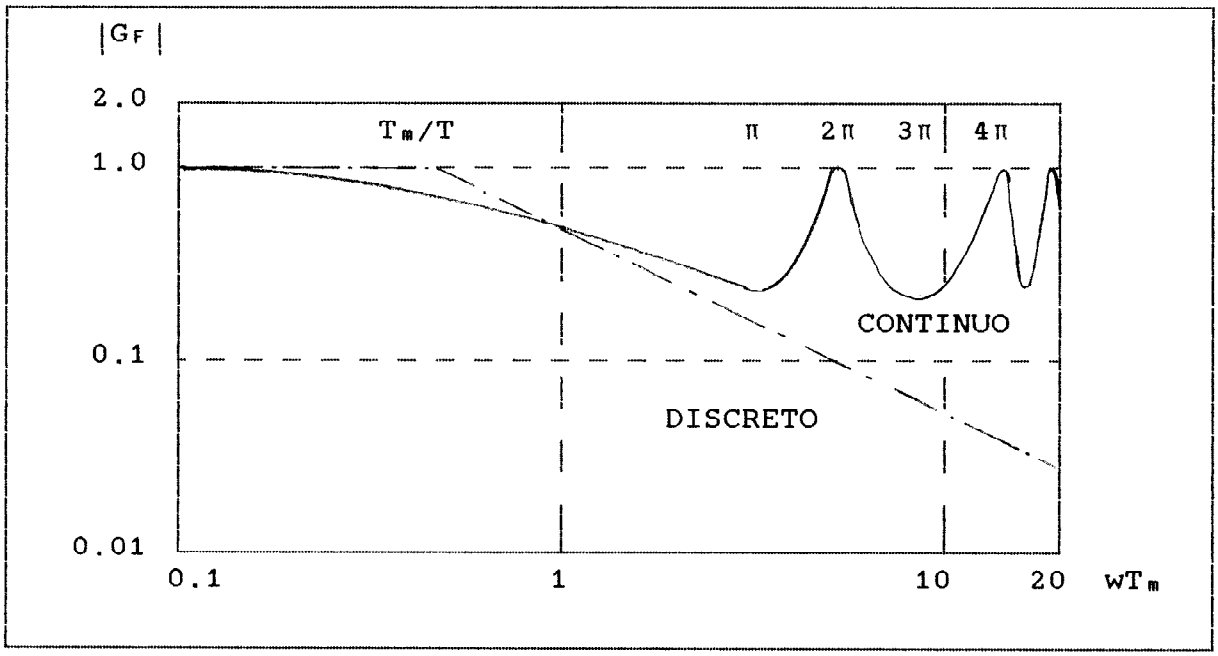


Fig 3.6. Amplitud de un filtro paso-bajo de primer orden.

Existen otras posibilidades para la detección de otros errores hardware. Por ejemplo, algún canal extra en el convertidor A/D puede ser usado, conectado a una tensión fija, para calibración o test de dicho convertidor.

Así mismo, los métodos software de filtrado de outliers están especialmente indicados en el funcionamiento en tiempo real de sistemas pequeños o medios. Estos métodos tienen por objetivo la

eliminación de las medidas erróneas sin que esto produzca una disminución en la capacidad de obtener valores precisos de la variable medida, aún en el caso en que esta sea variante.

Varios de estos métodos se basan en la estimación del valor medio de la señal medida ($\hat{y}$) o de una predicción de la misma $\hat{y}(k/k-1)$, así como de la estimación de la varianza $\hat{\sigma}_y^2$. Estas estimaciones pueden ser realizadas por promediado recursivo:

$$\hat{y}(k+1) = \hat{y}(k) + \frac{1}{k+1} [y(k+1) - \hat{y}(k)] \tag{3.9}$$

$$\hat{\sigma}_y^2(k+1) = \hat{\sigma}_y^2(k) + \frac{1}{k+1} [(y(k+1) - \hat{y}(k+1))^2 - \hat{\sigma}_y^2(k)] \tag{3.10}$$

La detección de outliers se lleva a cabo considerando que las señales que cumplan la siguiente ecuación lo son:

$$|y(k+1) - \hat{y}(k+1)| > K\ \hat{\sigma}_y(k+1) \tag{3.11}$$

donde K depende de la densidad de probabilidad P(y).

Para una distribución normal, se observa que una diferencia entre la medida y la media superior al triple de la desviación típica se presenta con un 0.27% de probabilidad. Así una buena elección en este caso será K=3. Evidentemente, tomando K mayor o menor, se modifica el rango de medidas válidas y la probabilidad de que una medida correcta sea tomada por outlier.

Una variante de la solución anterior consiste precisamente en variar K en tiempo real, acomodándolo a la magnitud de las variaciones en la señal medida. De esta forma el máximo valor permitido para la diferencia entre la medida y la media no será el mismo en distintos períodos de muestreo.

Una vez detectado el outlier la eliminación es secilla, sustituyendo el valor medido $y(k+1)$ por la media estimada $\hat{y}(k+1)$, o bien por ésta incorporando el máximo cambio permitido con el signo correspondiente.

3.4 INTERFACE DE PARAMETROS.

En un controlador adaptativo los parámetros del mismo se calculan generalmente en cada período de muestreo en base al conjunto de parámetros estimados del modelo del proceso de que se disponga en dicho período de muestreo.

Debido al funcionamiento, generalmente sometido a diversas condiciones de ruido, del sistema, así como la aparición ocasional de perturbaciones como picos o outliers en las medidas, pueden ocasionarse cambios en los coeficientes calculados del controlador en base a estimaciones no apropiadas de los parámetros del modelo.

Isermann y Lachmann (1982) proponen la inclusión en el sistema de control adaptativo de un interface de parámetros de forma que las

estimaciones sean evaluadas antes de su uso para el diseño del controlador (como se muestra en la figura 3.7). Para ello se proponen dos mecanismos:

- filtrado de las estimaciones de parámetros.
- uso de diferente período de muestreo en la estimación de parámetros y en control.

3.4.1 Filtrado de las estimaciones de parámetros.

Se pueden usar varios tipos de algoritmos, indicándose a continuación los más apropiados.

3.4.1.1 Filtro recursivo paso-bajo.

La ecuación de actualización de parámetros vendrá dada por:

$$\Theta_{fi}(k) = K\ \Theta_{fi}(k-1) + (1-K)\ \Theta_i(k) \qquad (3.12)$$

siendo Θ_i las estimaciones de parámetros y Θ_{fi} las filtradas. En este filtro se toma el factor de corrección K constante, dando lugar a una función de transferencia con ganancia estática igual a la unidad:

$$G_F(z^{-1}) = \frac{1 - K}{1 - K\ z^{-1}} \qquad (3.13)$$

3.4.1.2 Filtro recursivo paso-bajo con detección de picos.

Las ecuaciones serán:

$$\hat{\theta}_{mi}(k) = K\,\hat{\theta}_{mi}(k-1) + (1-K)\,\hat{\theta}_i(k)$$

Si $|\hat{\theta}_i(k) - \hat{\theta}_{mi}(k)| \leq \varepsilon_\theta$ entonces $\hat{\theta}_{fi}(k) = \hat{\theta}_{mi}(k)$

Si $|\hat{\theta}_i(k) - \hat{\theta}_{mi}(k)| > \varepsilon_\theta$ entonces

$$\hat{\theta}_{fi}(k) = \hat{\theta}_i(k) - \varepsilon_\theta \,\mathrm{sign}[\hat{\theta}_i(k) - \hat{\theta}_{mi}(k)]$$

$$\hat{\theta}_{mi}(k) = \hat{\theta}_i(k) \qquad (3.14)$$

Como se ve, el resultado será el mismo que en el caso anterior siempre que la medida no se separe de la media más de lo permitido. En caso contrario se supondrá la existencia de una variación en la medida, la cual será seguida por las estimaciones filtradas de los parámetros usando un factor de corrección.

3.4.1.3 Filtro con memoria finita.

La ecuación será:

$$\hat{\theta}_{fi}(k) = \frac{1}{n} \sum_{j=k-n+1}^{k} [\hat{\theta}_i(j)] \qquad (3.15)$$

donde se ve que se realiza la media de los últimos n valores de los parámetros, ponderados con igual peso. La función de transferencia resultante será:

$$G_F(z^{-1}) = \frac{1}{n} \; \frac{1 - z^{-n}}{1 - z^{-1}} \qquad (3.16)$$

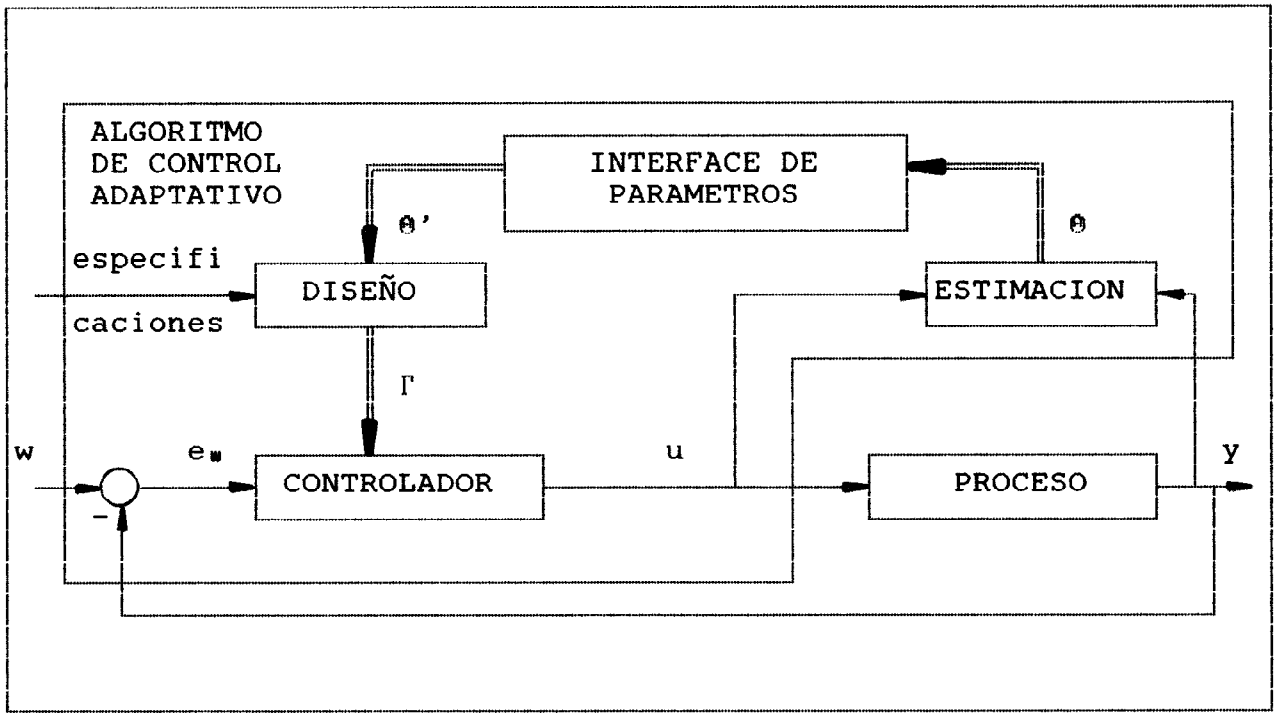


Fig 3.7. Diagrama de bloques de un controlador adaptativo con interface de parámetros.

3.4.1.4 Filtro con factor de olvido exponencial.

La ecuación será:

$$\hat{\theta}_{fi}(k) = \frac{1}{n} \sum_{j=k-n+1}^{k} \tau^{k-j} \, [\hat{\theta}_i(j)] \quad \text{con} \quad 0 < \tau < 1 \qquad (3.17)$$

de forma que se da más valor a los datos más recientes, realizando una ponderación exponencial de los mismos. La función de transferencia resultante será:

$$G_F(z^{-1}) = \frac{1}{n} \frac{1 - \tau}{1 - z^{-1}} \tag{3.18}$$

3.4.2 Diferente período para estimación y control.

Para muchos de los algoritmos de control apropiados para aplicaciones de control adaptativo (por ejemplo tiempo finito) la elección de un período de muestreo adecuado es crítica respecto a la acción de control, de forma que ésta será más fuerte cuanto menor sea aquél. De esta manera existe un límite inferior a T_m, impuesto por las restricciones del actuador o del propio proceso en cuanto a la acción de control que conlleva. En cambio, es posible realizar la elección de un T_m apropiado para la estimación de parámetros dentro de un amplio rango de valores, a veces inferiores a los impuestos por la restricción de control.

Una posible solución a este problema es la propuesta por Schumann et al (1981) de usar diferentes períodos de muestreo para estimación y control. Este posibilita ventajas adicionales como es la posibilidad de cambiar on-line T_{mc} de control, de forma que se obtenga en cada momento una adecuada acción de control, dejando fijo T_{me} para la estimación de parámetros (con un valor menor que el anterior).

La adaptación on-line de T_{mc} se puede realizar comprobando cual es el menor $T_{mc}(i)=i\ T_{me}$ $(i=1,2...)$ que ocasiona una acción de control

apropiada (en el diseño con tiempo finito esto se realiza fácilmente comprobando el parámetro $q_0=u(0)$). Evidentemente, es necesario que el interface transforme los parámetros estimados del modelo con período T_{me} en los correspondientes a un período T_{mc}.

Para realizar dicha trasformación, siendo h(k) y h*(t) las respuestas discreta y continua respectivamente ante una entrada escalón, se tendrá:

$$h(k) = h^*(k\ T_m) \tag{3.19}$$

Para un período $T_m(i)=i\ T_m$ la respuesta discreta será:

$$h^{(i)}(k) = h^*(k\ T_m(i)) = h^*(k\ i\ T_m) \qquad i = 1,2,... \tag{3.20}$$

De esta manera se puede obtener la respuesta discreta con $T_m(i)$ a partir de la misma con período T_m, de la forma:

$$h^{(i)}(k) = h(k\ i) \qquad i = 1,2... \tag{3.21}$$

Teniendo $h^{(i)}(k)$ es fácil obtener el modelo de la función de transferencia discreta con período $i\ T_m$, que es lo necesario para realizar el diseño del controlador.

3.5 INFLUENCIA DEL ACTUADOR.

La conexión entre el algoritmo de control y el proceso a controlar

se realiza a través de un actuador (además del conversor D/A para actuadores analógicos), que transforma la señal de salida del controlador, de acuerdo con los requerimientos del proceso. Por ello, será necesario tener en cuenta en cada aplicación en particular, la forma de control del mismo así como su dinámica, ya que el actuador no es un elemento ideal e incorpora comportamientos no lineales al sistema.

Una no linealidad que se presenta frecuentemente es la saturación. En el caso de las válvulas, usadas a menudo en control de procesos, los límites corresponden a totalmente abierta o cerrada. El sistema que la incorpora puede ser descrito linealmente cuando la válvula no está saturada, pero la consideración de la no-linealidad es importante cuando se efectúan grandes cambios en la señal de control.

3.5.1 Anti-reset windup.

Uno de los problemas más frecuentes que sucede cuando no se tiene en cuenta la saturación del actuador es lo que se denomina "reset windup" o saturación del integrador, en un controlador con acción integral.

Debido a que un integrador es un sistema inestable, puede ocurrir que la integral asuma valores muy elevados si la salida se satura y el controlador continúa integrando el error. Entonces será necesario que transcurra un mayor número de periodos de muestreo para

retornar al valor normal y el bucle continúa saturado, pudiendo llegar a la inestabilidad.

La solución a este problema pasa por la detección de la saturación en el actuador, para parar la integración cuando la salida está limitada. Otra posible solución es la obtención del valor real de la posición del actuador, en vez del calculado en función a la ley de control, que será erróneo cuando sobrepase los límites.

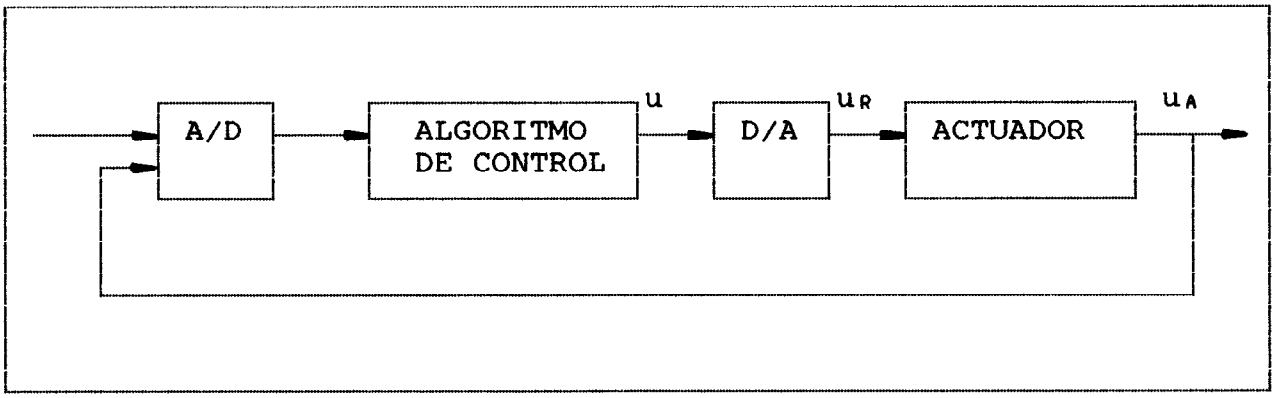


Fig 3.8. Realimentación de la posición del actuador.

Esto último se puede realizar realimentando por hardware la posición real del actuador, como se muestra en la figura 3.8 (Isermann, 1981), o bien por software comparando cada nueva acción de control con límites previamente fijados en función del actuador. Así, si se tiene un controlador lineal en la forma:

$$R(z^{-1})\ u(k) = -\ S(z^{-1})\ y(k) + T(z^{-1})\ w(k) \qquad (3.22)$$

la incorporación de la compensación anti-reset windup se realiza

mediante:

$$A_0\, n = T\, w - S\, y + (A_0 - R)\, u \tag{3.23}$$

$$u = \mathrm{sat}[n] \tag{3.24}$$

siendo la función sat definida como:

$$\mathrm{sat}[u] = \begin{cases} U_{min} & \text{si } u \leq U_{min} \\ u & \text{si } U_{min} < u < U_{max} \\ U_{max} & \text{si } u \geq U_{max} \end{cases} \tag{3.25}$$

Este controlador es equivalente al dado por (3.22) cuando la señal de control no está saturada.

El polinomio A_0, que debe ser estable, puede ser interpretado como la dinámica del observador asociada al controlador. Un diagrama de bloques puede verse en la figura 3.9. Un caso particularmente simple sucede cuando $A_0=1$, quedando el controlador:

$$u = \mathrm{sat}[T\, w - S\, y + (1-R)\, u] \tag{3.26}$$

Otro problema debido a la saturación del actuador puede producirse en la estimación. Para evitarlo es importante que al estimador le llegue la señal que se envía al proceso, pues en otro caso podrán producirse estimaciones incorrectas, por ejemplo, en la ganancia del proceso.

3.6 ARRANQUE DEL SISTEMA.

Dependiendo de la información de que se disponga acerca del proceso a controlar, existen varios caminos para inicializar el sistema de control adaptativo.

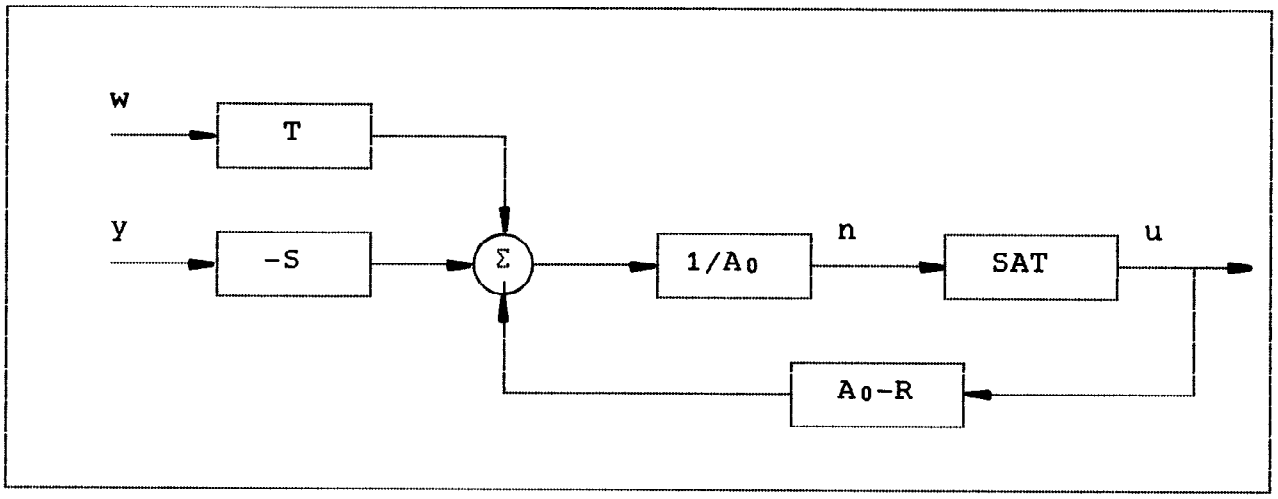


Fig 3.9. Configuración con compensación anti-reset windup.

3.6.1 Estimación previa de parámetros.

Aún en el caso del conocimiento previo del orden y tiempo muerto del proceso, no es deseable la aplicación del controlador adaptativo desde el inicio, ya que, en los primeros períodos de muestreo, las estimaciones de los parámetros del proceso aún no son buenas y por tanto, la acción de control será indefinida.

Una precaución que se puede incorporar fácilmente es la inclusión de límites inferiores y superiores, por software, a la salida del controlador. Para evitar que las señales de entrada y salida del

proceso tomen valores inaceptables durante los primeros períodos de muestreo, es aconsejable la utilización de una fase de estimación previa de los parámetros del proceso.

Esta fase, que puede ser realizada en bucle abierto en el caso de sistemas estables, puede llevarse a cabo introduciendo una señal a la entrada del proceso durante unos pocos períodos de muestreo (de 10 a 20, dependiendo del orden, tiempo muerto del modelo del proceso, perturbaciones, etc.), de forma que se active el estimador y se obtenga un razonable modelo para realizar, en base a él, el diseño del controlador.

3.6.2 Arranque con procesos inestables.

En el caso de procesos inestables, es necesario efectuar el arranque en bucle cerrado, incluyendo un controlador fijo (PID) que estabilice la planta (Isermann, 1982), pero ejecutando, al mismo tiempo, el algoritmo de identificación. Una vez transcurrido el número de períodos de muestreo necesarios para una estimación suficientemente precisa de los parámetros de la planta, se conmutará del controlador fijo al adaptativo.

En el caso de que el proceso haya sido gobernado anteriormente con un controlador convencional o adaptativo, los valores iniciales pueden ser los correspondientes a éste. Así mismo están especialmente indicados para el diseño del controlador fijo de arranque de procesos inestables, los métodos de control robusto.

3.6.3 Fase de preidentificación.

Para inicializar el algoritmo de control adaptativo, es necesario especificar a priori varios factores (entre ellos el período de muestreo, el orden del modelo del proceso y el tiempo muerto de dicho modelo) pudiendo suceder que ninguno de ellos sea conocido.

En este caso es necesario la realización de una fase de preidentificación, cuyo diagrama de bloques se muestra en la figura 3.10, que dé un valor inicial para los mismos. Posteriormente, será posible la estimación on-line de dichos factores, durante el funcionamiento normal del sistema o incluso seguir las posibles variaciones de alguno de ellos.

3.6.3.1 Período de muestreo.

Uno de los parámetros que puede influir grandemente en el comportamiento del algoritmo de control adaptativo es el período de muestreo. En general, es preciso que el valor de T_m esté de acuerdo con el funcionamiento deseado del sistema en cadena cerrada.

Así mismo, para varios algoritmos apropiados para control adaptativo (por ejemplo tiempo finito o mínima varianza), la elección de un período de muestreo apropiado es crítico para la acción de control ante cambios de consigna o perturbaciones. En efecto, en los dos casos citados, cuanto menor es T_m mayor es la acción de

control, que, sin embargo, debe ser compatible con las restricciones impuestas por el actuador y el proceso a controlar.

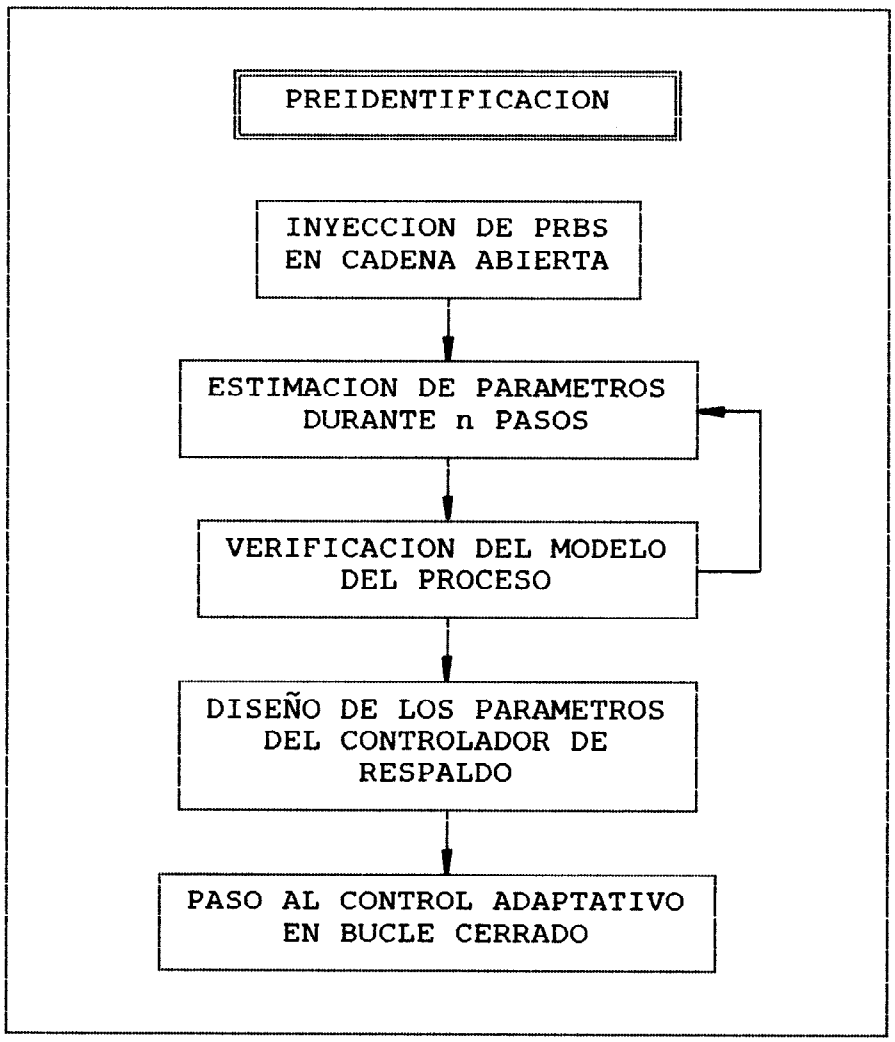


Fig 3.10. Diagrama de bloques del procedimiento de preidentificación.

Una posible solución a este problema es la adaptación del período de muestreo on-line para obtener una apropiada acción de control (Schumann, Lachmann e Isermann, 1981), partiendo de un T_m inicial. Para la obtención de este T_m inicial en la fase de preidentificación se pueden dar varias reglas. Una de ellas es relacionarlo con

el tiempo de subida según:

$$N_r = T_r/T_m \tag{3.27}$$

donde N_r es el número de muestras apropiadas durante T_r, que debe estar comprendido en el rango de 2 a 4 (Wittenmark y Astrom, 1984). Según Isermann y Lachmann (1982) una buena acción de control puede ser obtenida si se cumple:

$$1/15\ T_{95} \leq T_m \leq 1/4\ T_{95} \tag{3.28}$$

donde T_{95} es el tiempo de establecimiento de la respuesta transitoria.

3.6.3.2 <u>Orden y tiempo muerto del modelo del proceso.</u>

Estos dos factores están directamente relacionados con la estructura exacta de la ecuación en diferencias lineal del modelo del proceso, por lo que su obtención está ligada a la determinación de la estructura del sistema.

Diversas simulaciones demuestran que un orden incorrecto del modelo del proceso puede causar serios problemas, sobretodo si es subestimado, en el mecanismo de control adaptativo. Esto no sucede así generalmente si el orden que se toma, $\hat{m}$, está incluido dentro del siguiente rango (en el que m_0 es el verdadero):

$$m_0 - 1 \leq \hat{m} \leq m_0 + 2 \qquad (3.29)$$

Sin embargo, los algoritmos de control adaptativo son mucho más sensibles en cuanto a la elección del tiempo muerto, $\hat{d}$. Si d no es conocido exactamente o varía con el tiempo, el control obtenido será pobre o incluso inestable.

Un estudio del comportamiento de los algoritmos de control frente a una parametrización errónea del estimador en cuanto a orden y tiempo muerto del modelo se incluye en Isermann y Lachmann (1982) y García (1988). En éste se explora separadamente el comportamiento del control, al subestimar y sobreestimar el orden y el tiempo muerto del modelo del proceso dado por la función de transferencia:

$$G_P(s) = \frac{1 + 2s}{(1 + 3s)\,(1 + 5s + 25s^2)}\, e^{-6s} \qquad (3.30)$$

empleando un período de muestreo de 3 segundos, un estimador RLS y tres algoritmos de control (tiempo finito de orden normal, asignación de polos de estructura completa y controlador predictivo integral extendido).

Los resultados se muestran en la figura 3.11, en la que en abscisas se representa el orden del modelo y en ordenadas el tiempo muerto. El círculo representa la parametrización correcta y la letra E los casos en que se obtiene un lazo estable.

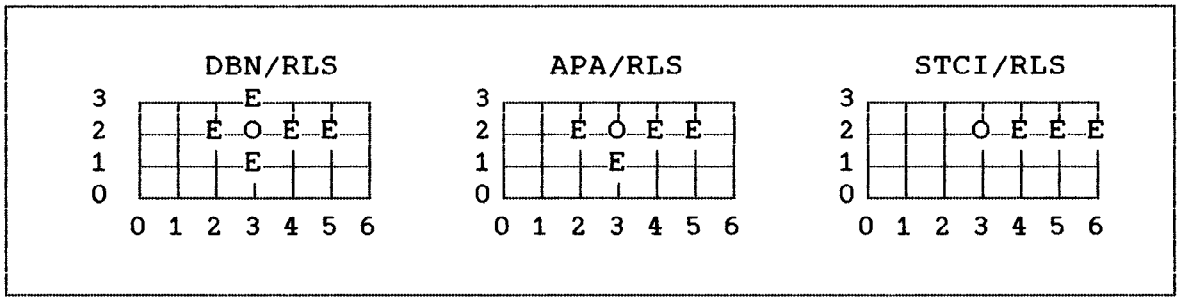


Fig 3.11. Comportamiento de los controladores adaptativos ante especificaciones erróneas del orden o retardo del modelo del proceso.

Al igual que en el caso de T_m, en la fase de preidentificación se determina un valor inicial de estos factores, pudiendo incorporarse una estimación on-line de los mismos durante el funcionamiento normal del sistema, lo que permite ajustar aún más su valor al correcto así como seguir sus posibles variaciones (Schumann, Lachmann e Isermann, 1981, Kurz, 1979 y 1981, De Keyser, 1986).

Un método simple pero efectivo para la comprobación del orden del modelo es el descrito en Hsia (1977). Este método está basado en la comprobación del ajuste de los datos de entrada-salida en el modelo, suponiendo éste de un orden incluido en un cierto rango. La comprobación anterior puede efectuarse mediante la función V de suma de errores al cuadrado:

$$V = (y - \boldsymbol{\varphi}^T \hat{\boldsymbol{\theta}})^T (y - \boldsymbol{\varphi}^T \hat{\boldsymbol{\theta}}) \qquad (3.31)$$

siendo $\boldsymbol{\varphi}$ y $\hat{\boldsymbol{\theta}}$ los vectores de datos y de parámetros estimados mediante mínimos cuadrados, para un orden del modelo dado.

En general, V decrece cuando m se incrementa. Sin embargo, la reducción de V deja de ser significativa cuando m es más grande que el verdadero m_0.

Considerando dos modelos M_1 y M_2 con i_1 y i_2 parámetros independientes respectivamente, de forma que $i_1 < i_2$, el método de prueba debe comparar los dos modelos y juzgar, dado que M_1 es un subconjunto de M_2, cuando el mayor grado de libertad (dado por el mayor número de parámetros independientes) en M_2 proporciona un significativo mejor ajuste.

Dado que en general se requiere un mayor esfuerzo computacional para obtener y usar el modelo M_2, debido a su mayor número de parámetros, en los métodos de prueba suele existir una penalización progresiva según se incrementa la complejidad de la estructura del modelo.

Uno de los criterios propuestos por Akaike para la elección de la estructura óptima para el modelo, consiste en obtener lo que denomina FPE (Error Final de Predicción) para cada una de las estructuras candidatas (Soderstrom, 1977), de forma que aquélla que dé el menor valor para FPE será la elegida. Este índice se obtiene como sigue:

$$FPE(M) = F(\hat{\theta}^n) \frac{1 + i/n}{1 - i/n} \qquad (3.32)$$

donde n es el número de medidas realizadas para obtener las estimaciones $\hat{\theta}^n$ y $F(\hat{\theta}^n)$ es una función de coste obtenida en base a los errores producidos en la estimación.

Este principio puede ser convenientemente usado para determinar el orden apropiado del modelo de una forma simple, obteniendo las estimaciones de parámetros $\hat{\theta}$ y las correspondientes funciones de error V para una secuencia de órdenes del modelo m=1,2,3,...

Si tenemos por ejemplo el sistema de segundo orden dado por la ecuación:

$$y(k) - 1.5\ y(k-1) + 0.7\ y(k-2) = u(k-1) + 0.5\ u(k-2) + v(k) \quad (3.33)$$

al cual le aplicamos el método anterior, con una estimación de parámetros mediante RLS y distintos órdenes del modelo, se obtendrá la tabla 3.2, (Astrom y Eykhoff, 1971) para cinco niveles de ruido. Se puede comprobar que mediante el experimento se obtiene el orden correcto del modelo del proceso $\hat{m}=2$, ya que, aunque los valores de V son menores para m=3, debido a las penalizaciones progresivas de las estructuras más complejas, se obtiene el FPE mínimo para m=2.

En cuanto al tiempo muerto del proceso, d, vendrá dado por el retardo del mismo medido en número de períodos de muestreo. Así, siendo t_r el retardo:

$$d = INT(t_r/T_m) + 1 \qquad (3.34)$$

Este factor d, puede ser visto como el horizonte de previsión del controlador (Wittenmark y Astrom, 1984) y éste puede tenerlo en cuenta trabajando con un largo horizonte, lo que disminuiría la varianza de la señal de control a expensas de aumentar la de la variable de salida. En todo caso, los controladores de tipo predictivo son más sensibles que los clásicos ante una elección errónea del tiempo muerto del proceso, tanto por exceso como por defecto. Como contrapartida son prácticamente insensibles a la sobreestimación del orden.

nivel de ruido (σ)	m=1	m=2	m=3
0.0	265.8	0.000	—
0.1	248.4	0.987	0.983
0.5	337.8	24.55	24.45
1.0	308.1	99.86	98.70
5.0	5131.	2462.	2440.

Tabla 3.2. Variaciones de V en función del orden del modelo para diferentes niveles de ruido.

3.7 PROGRAMACION.

La programación es un aspecto importante en la implementación de un sistema de control, respecto tanto a la eficiencia del mismo como

al tiempo empleado en dicha implementación. En el caso en que se use un microcomputador y la velocidad de cálculo no sea crucial, la programación no causará generalmente problemas, pudiendo emplearse lenguajes de alto nivel, y aritmética de coma flotante, aunque siempre será necesario el cumplimiento de algunos requisitos específicos para la operación en tiempo real.

3.7.1 Requerimientos.

Debido a la problemática especial existente en el campo de control de procesos por computador, es necesario definir, aparte de los criterios básicos de programación que debe cumplir cualquier aplicación genérica (legibilidad, mantenibilidad, sencillez, portabilidad), algunas exigencias específicas en el control en tiempo real (De la Puente y Crespo, 1984):

- Es esencial una organización software en multiproceso, en la que se puedan identificar distintas tareas (por ejemplo interface con el usuario, interface con el proceso, control, tratamiento de alarmas, etc.). Así es necesario disponer de construcciones en el lenguaje que permitan describir, crear e intercomunicar procesos concurrentes.

- Programación directa de dispositivos de bajo nivel para poder gobernar dispositivos de E/S no estándares (convertidores A/D y D/A, entrada/salida digital, etc.), ya que en un sistema en tiempo real es esencial la relación entre el computador y su

entorno y, por tanto, este acceso al hardware de la máquina. Para realizarlo desde un lenguaje de alto nivel, se necesitan mecanismos para especificar, de forma explícita, la disposición y el mapeado de los tipos abstractos de datos del lenguaje, sobre la memoria física y los registros de E/S.

- Para conseguir una gran fiabilidad, el software en tiempo real debe ser capaz de una recuperación de las condiciones de error ya que, a diferencia del científico y comercial, debe funcionar, muy a menudo, ininterrumpidamente aún en presencia de condiciones de error. Esto es importante, ya que un fallo en un sistema de estas características puede ser muy caro.

  Existen dos aspectos en el tratamiento de errores. El primero es la detección (en la fase de compilación o en la de ejecución) y el segundo la respuesta al error detectado (que debe ser capaz de restaurar el sistema a un estado consistente, reparar cualquier daño que se haya producido y continuar funcionando normalmente).

- A diferencia de otras aplicaciones en las que, debido a la caída de los costes hardware y subida de los software, se sacrifica en parte la eficiencia por la flexibilidad y la sencillez, en los sistemas en tiempo real aquélla es importante, ya que se debe procurar un alto grado de flujo interno de cálculo para satisfacer las restricciones impuestas por el proceso a controlar. A veces, el no obtener el resultado de un

cálculo dentro de un tiempo especificado, puede ser tan malo como recibir un resultado erróneo.

El objetivo es garantizar los tiempos de respuesta del computador ante los cambios del sistema controlado. En este sentido es importante evitar los mecanismos que conduzcan a sobrecargas de tiempo de cálculo, maximizando la velocidad de ejecución de los mismos.

- El uso de un lenguaje apropiado de alto nivel puede servir de apoyo ya en la fase de determinación de la estructura más idónea del sistema y subdivisión en módulos. Así, puede realizarse el diseño de forma descendente, mediante depuraciones sucesivas. El modelo definido en la etapa de especificación, en base al funcionamiento deseado del sistema, se va detallando progresivamente en forma de módulos, la interface entre los mismos, las funciones que cumplirá cada uno y los algoritmos que permitirán dicho cumplimiento. De esta forma se puede realizar un diseño modular de modo natural, pudiéndose diseñar los distintos subsistemas independientemente del resto.

3.7.2 Programación concurrente.

Es natural razonar sobre los bucles de control como actividades concurrentes que se están ejecutando en paralelo mientras que, sin embargo, el computador digital trabaja de forma secuencial.

Se presenta entonces el problema de trasladar un número dado de actividades concurrentes a un programa secuencial. Para ello se presentan las nociones de proceso o tarea, que representan actividades que deben ser ejecutadas en paralelo. Cada una de estas actividades en tiempo real puede estar estructurada de idéntica manera que si fuera secuencial, usando subrutinas o procedimientos.

Para la ejecución de actividades concurrentes, es necesario asociar una prioridad a cada proceso. Este puede ser programado para su ejecución periódica o en respuesta a eventos tales como interrupciones o finalización de otras tareas.

Varios son los problemas que pueden presentarse, como son:

- Si procesos diferentes comparten los mismos datos, es necesario asegurarse que un proceso no trata de usarlos mientras son modificados por otro.

- Si procesos diferentes comparten recursos, es necesario prever las situaciones que se pueden producir para evitar, por ejemplo, que dos procesos estén esperando uno por el otro para el uso de un mismo recurso.

- El computador debe tener potencia suficiente para que todas las actividades se completen en el tiempo requerido.

Uno de los ejemplos más sencillos de necesidad de programación

concurrente es el de control con interface con el operador. Uno de los procesos (control) debe ejecutarse cada período de muestreo y el otro (comunicación con el operador) cuando el primero deje libre el procesador. En este caso el control tiene mayor prioridad (tarea foreground) que el interface con el operador (tarea background).

3.7.3 Lenguajes.

En los primeros años de control por computador no existían lenguajes de alto nivel disponibles, apropiados para dicha aplicación, por lo que era necesaria la codificación en ensamblador, aunque el tiempo de desarrollo del sistema en este caso es mucho mayor, el código es mucho menos legible, la modificación del mismo, por tanto, mucho más complicada, etc.

Hoy en día, sin embargo, existen varios lenguajes de alto nivel que pueden ser usados, como son el Forth, Basic, Fortran, C, Pascal, Concurrent Pascal, Modula 2, Ada. Varios de ellos soportan así mismo las operaciones básicas para trabajo en tiempo real, cumpliendo con los criterios anteriormente especificados (concurrencia, modularidad, etc.), como el Modula 2, Ada, algunas versiones del Pascal y Forth.

En otros casos las operaciones en tiempo real se realizan a través de un ejecutivo diseñado para servir de interface para programas en C o Pascal (Maier, 1982, Elmquist y Mattsson, 1982). Sin embargo, un kernel de sistema operativo usualmente requiere un considerable

tiempo del procesador para la gestión de tareas. El cambio de una tarea a otra puede necesitar alrededor de 100 μs con procesadores de 16 bits, (Hanselmann, 1987) lo cual puede ser excesivo en ciertas aplicaciones.

Para problemas con sólo dos procesos, no es difícil escribir un programa monitor que se encarge de su gestión, en un relativamente reducido número de líneas de código ensamblador. Es, sin embargo, más complicado extender este monitor para que gestione situaciones más complejas, incluyendo varios procesos, peticiones de E/S, interrupciones, etc. (Cavalcanti, 1988).

3.8 IMPLICACIONES ARITMETICAS.

Cuando se implementa un sistema de control por computador es necesario conocer los efectos producidos por la limitación en el número de bits de los convertidores A/D y D/A, así como en la longitud de palabra del procesador, para realizar una apropiada elección de los mismos, así como de si los cálculos deben ser realizados en aritmética de coma fija o flotante.

Estos efectos estarán en función de la compleja interacción de la realimentación, el algoritmo, el período de muestreo, la amplitud de los cambios producidos en las señales, etc. En cuanto a este último factor, en el caso de que los cambios producidos en las señales sean significativos y la longitud de palabra del procesador apropiado, el efecto de cuantización de amplitud es tan pequeño que

ésta puede ser considerada cuasi-continua. Sin embargo, para pequeños cambios en las señales y procesadores con pequeña longitud de palabra, los efectos son apreciables.

Se pueden considerar como fuentes de error numérico las siguientes:

- cuantización en los convertidores A/D.
- cuantización de parámetros.
- redondeo o rebasamiento en adiciones, multiplicaciones y otras operaciones.
- cuantización en los convertidores D/A.

Hay que tener en cuenta así mismo que las cuantizaciones en los cálculos tienen mucha mayor importancia en el caso de emplear aritmética de coma fija.

Estas cuantizaciones de amplitud introducen no-linealidades en el bucle de control por computador, mostrando la figura 3.12 un diagrama de bloques simplificado. En el caso de la entrada analógica, las tensiones o corrientes normalizadas (0...10V, 0...20 mA, 4...20 mA) son muestreadas por conversor A/D y digitalizadas. La resolución vendrá dada por la unidad de cuantización, función así mismo del número de bits del conversor (n).

El rango numérico será así:

$$RN = 2^n - 1 \qquad (3.35)$$

y la unidad de cuantización:

$$\delta = \frac{1}{RN} = \frac{1}{2^n - 1} \qquad (3.36)$$

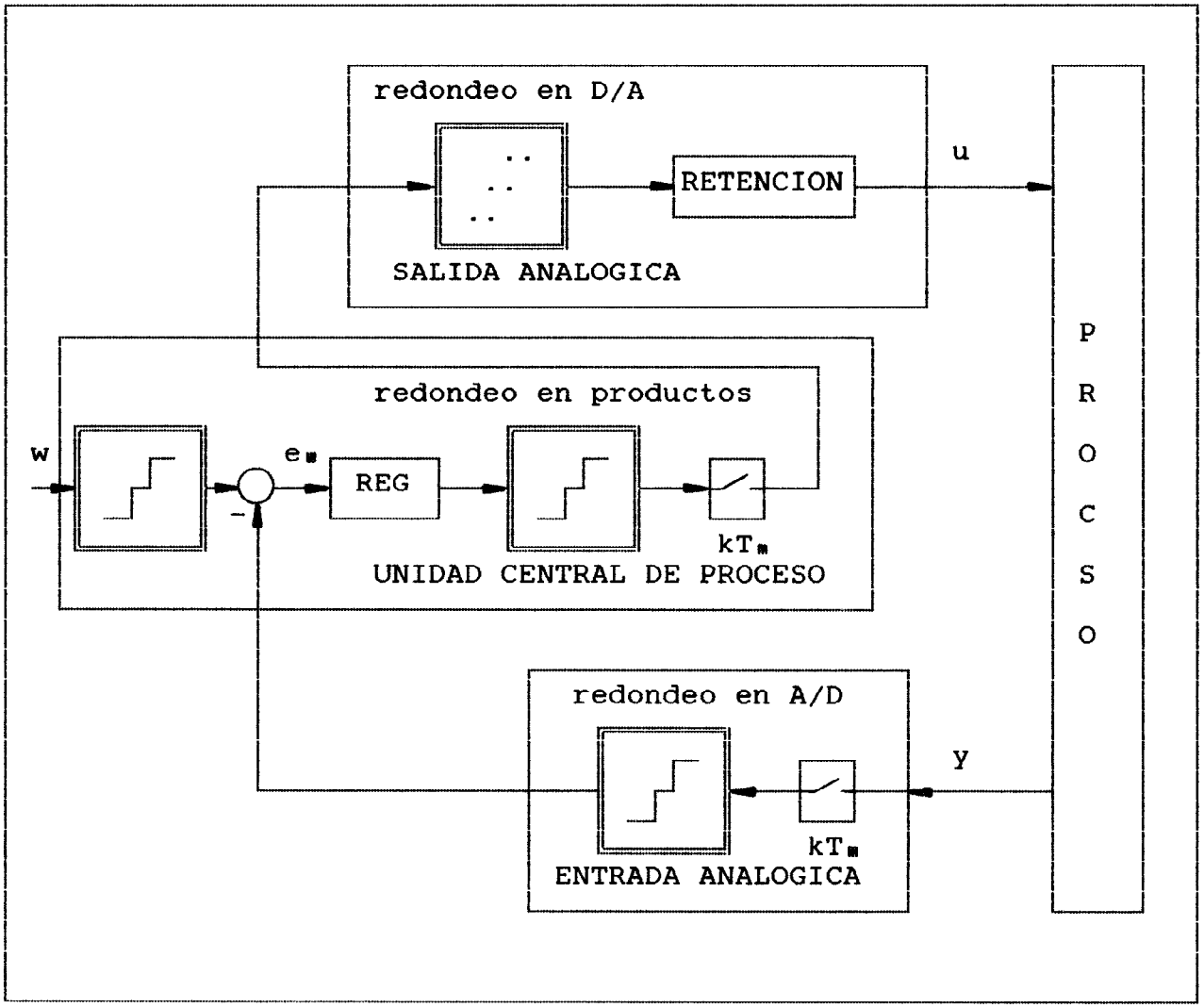


Fig 3.12. Diagrama de bloqes simplificado de los puntos en que se produce cuantización de amplitud en control digital.

En la tabla 3.3 se muestran estos valores para un conversor A/D con

n=8, 10, 12 y 15 bits. Estos conversores cuentan el múltiplo de δ que más se aproxima a la tensión analógica de entrada Y:

$$Y_Q = K\,\delta \quad \text{con} \quad Y = Y_Q + \delta_Y \tag{3.37}$$

El resto ($\delta_Y < \delta$) será redondeado o truncado, cometiéndose un error de cuantización dado por (3.38) o (3.39), respectivamente:

$$-0.5 \leq \delta_Y/\delta < 0.5 \tag{3.38}$$

$$0 \leq \delta_Y/\delta < 1 \tag{3.39}$$

n (bits)	8	10	12	15
Rango numérico	255	1023	4095	32767
Unidad cuantización	0.00392	0.00098	0.00024	0.00003
% Unidad cuantiz.	0.392	0.098	0.024	0.003

Tabla 3.3. Unidad de cuantización en función del número de bits.

3.8.1 Aritmética de coma fija/coma flotante.

Además de estos dos tipos de aritmética, existen otras no estándar que pueden ser usadas en la implementación de un sistema de control, como con la logarítmica o la de residuos.

El uso de la representación logarítmica para control digital fue

propuesta por Hall et al y Kingsbury y Rayner y, más recientemente, en el caso de procesadores para control de propósito especial, por Lang. La principal ventaja de la de residuos es la gran velocidad de computación al trabajar con dígitos en vez de con la totalidad del número (Waser y Flynn, 1982).

Las aritméticas más conocidas son, sin embargo la de coma fija y la de coma flotante, siendo además las que generalmente soportan los procesadores estándar. En muchas aplicaciones es preferida la de coma fija debido a la mayor velocidad de computación que se obtiene y al menor espacio de memoria necesario al trabajar con unidades aritméticas relativamente simples, sin embargo es necesario tener mucho cuidado con respecto a las cuantizaciones de amplitud.

Los formatos de datos en coma fija o flotante hacen uso de la representación de complemento a dos, siendo así el valor decimal de un número:

$$v_d = 2^{-i} \left[-g_{n-1}\, 2^{n-1} + \sum_{j=2}^{n-2} g_j\, 2^j\right] \quad \text{con} \quad g_i \in [0,1] \qquad (3.40)$$

donde g_j (j=0 ... l-2) representa los digitos binarios, g_{n-1} contiene la información del signo, i la localización del punto decimal y n es la longitud total de palabra. El valor de i puede ser diferente para cada número en la representación en coma flotante, mientras que, en coma fija, permanece constante.

La razón de que la representación anterior sea llamada de complemento a 2 es obvia en el importante caso de los números fraccionales (i=n-1), donde se tendrá:

$$v_d = -g_{n-1} + \sum_{j=0}^{n-2} g_j \, 2^{j-(n-1)} \qquad (3.41)$$

Debido a que la substracción de 2 tiene el mismo efecto que un cambio de peso en g_{i-1} de +1 a -1, se tendrá que para un número $v_d<0$, $2-|v_d|$ presenta los mismos bits que $|v_d|$.

La principal ventaja de esta representación resulta así la simplicidad del hardware necesario para realizar adiciones o substracciones, sin necesidad de hacer distinciones en cuanto al signo o magnitud de los operandos (realizando un escalado apropiado). Unicamente es necesario con esta representación un circuito de complementación y otro sumador para realizar sumas o substracciones.

Otra ventaja es que, en la computación del producto escalar, siempre se produce el resultado correcto mientras no se sobrepase el rango numérico, con lo que los rebosamientos intermedios o sumas parciales no influyen en el resultado final. En este caso es muy útil que el procesador disponga de bits de rebosamiento para comprobar si el resultado final está o no dentro del rango numérico.

En cuanto a la multiplicación, puede ser realizada con una apropiada extensión de signo en la representación de números negativos o bien usando el algoritmo de Booth (Booth, 1951) o modificaciones del mismo. Estos algoritmos están incorporados en los multiplicadores hardware de los procesadores de señal, ya que la velocidad de cálculo que se alcanza con ellos es superior que con la simple técnica de desplazamiento y suma (la multiplicación se realiza con sólo una adición y dos substracciones, conjuntamente a los necesarios desplazamientos).

El producto de dos números de n bits será uno de 2n-1, ya que el resultado sólo necesita un bit de signo. Es posible multiplicar tanto enteros como fracciones ya que la interpretación de los bits sólo tiene lugar cuando se almacena el resultado, como se muestra en la figura 3.13.

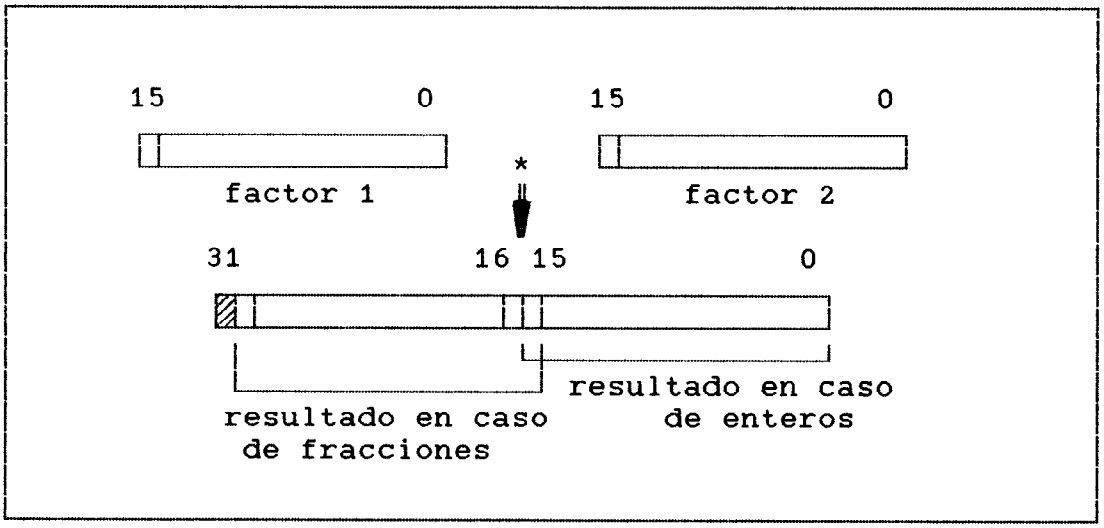


Fig 3.13. Multiplicación en coma fija.

Para control digital aplicando aritmética de coma fija es preferible el uso de números fraccionales, ya que, aunque no es tan exacto como el uso de enteros, el rebosamiento puede ocurrir en sumas o restas, pero no en multiplicaciones. Con números enteros la precisión se paga con un excesivo riesgo de rebosamiento y es necesaria la inclusión de un apropiado escalado de los datos.

En cambio, si es posible el uso de aritmética de coma flotante, los valores numéricos de los datos estarán en un rango apropiado y las pequeñas diferencias de números elevados podrán ser realizados apropiadamente sin problemas.

Son comunes las siguientes representaciones numéricas:

- coma fija, precisión simple (16 bits).
- coma fija, precisión doble (32 bits).
- coma flotante, precisión simple (32 bits).
- coma flotante, precisión doble (64 bits).

El formato de coma flotante, precisión simple (estándar IEEE 754) consiste en 24 bit para la mantisa f (uno de ellos para el signo) y 8 para el exponente e. El valor decimal vendrá dado por:

$$(-1)^s \, [2^{e-127}] \, (1+f) \qquad (3.42)$$

El rango va así desde $2^{-126} \approx 10^{-38}$ hasta $2^{+128} \approx 3 \; 10^{+38}$, estando la precisión en función del valor del bit menos significativo en f.

Este valor es $2^{-23} \approx 10^{-7}$. Si se utiliza doble precisión (56 bits de mantisa + 8 de exponente), el valor del bit menos significativo será de $2^{-55} \approx 3\ 10^{-17}$.

Se debe hacer notar que la mayor diferencia entre los formatos en coma fija o flotante es que en este último caso el error producido por la cuantización es relativo, dependiendo de la amplitud de la señal, mientras que en coma fija es absoluto.

3.8.2 Rebosamiento.

Debido a la finita longitud de palabra del procesador, es necesario que las variables que intervienen en los algoritmos del sistema de control estén dentro de un rango limitado. Sin embargo, no es deseable que sean demasiado pequeñas para que no afecte excesivamente la cuantización de amplitud.

Con la representación de complemento a dos, se produce un efecto indeseable con la saturación, es el "wrap-around". Debido a que el rango numérico no es simétrico:

$$-1.0 \leq v_d \leq 1.0 - 2^{-(n-1)} \qquad (3.43)$$

(habiendo aproximado el límite superior a 1.0 por simplicidad), está claro que el producto -1.0 (-1.0)=+1.0 nunca podrá ser representado y será interpretado erróneamente como -1.0, como se indica en la figura 3.14 (a). Para combatir este efecto, los proce-

sadores de señal frecuentemente incorporan hardware para saturación opcional en estos casos, aunque también es posible realizar la saturación por software. El efecto se indica en la figura 3.14 (b).

3.8.3 Efectos de la cuantización de amplitud.

Las consecuencias de la cuantización de amplitud debidas a redondeos o truncamientos dependen en gran medida del sistema a implementar y de aspectos de los propios algoritmos.

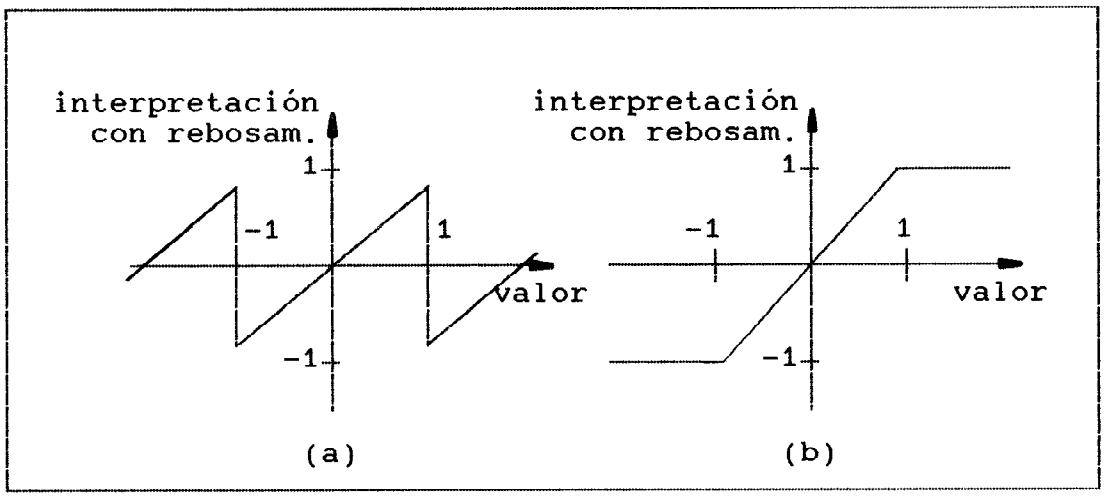


Fig 3.14. Rebosamiento (a) sin saturación (b) con saturación.

Si el acumulador del procesador es de doble longitud que los datos, lo cual es frecuente, los productos son acumulados en la total longitud y el truncado o redondeo se produce únicamente con el resultado final. En el caso de adiciones y sustracciones, estas serán exactas mientras no existan problemas de rebosamiento.

La cuantización ocasiona fundamentalmente tres efectos: sesgo de

los datos, ruido y ciclos límites. El sesgo es introducido por la truncación debido a que en complemento a dos se tiene TRUNC(x) < x tanto para x positivos como para negativos. Así es más recomendable el uso de redondeos en vez de truncar los datos.

El modelo de ruido de la cuantización es una aproximación ampliamente usada, reemplazando las operaciones por sus modelos ideales conjuntamente a una perturbación aditiva e_a. Como se ve en la figura 3.15, los conversores A/D y D/A son representados por una ganancia lineal más una perturbación que modele el truncado o redondeo. Aunque las adicciones son exactas efectuando los cálculos con coma fija, habrá sin embargo errores en la multiplicación. Así, ésta será representada por una multiplicación exacta más un error aditivo.

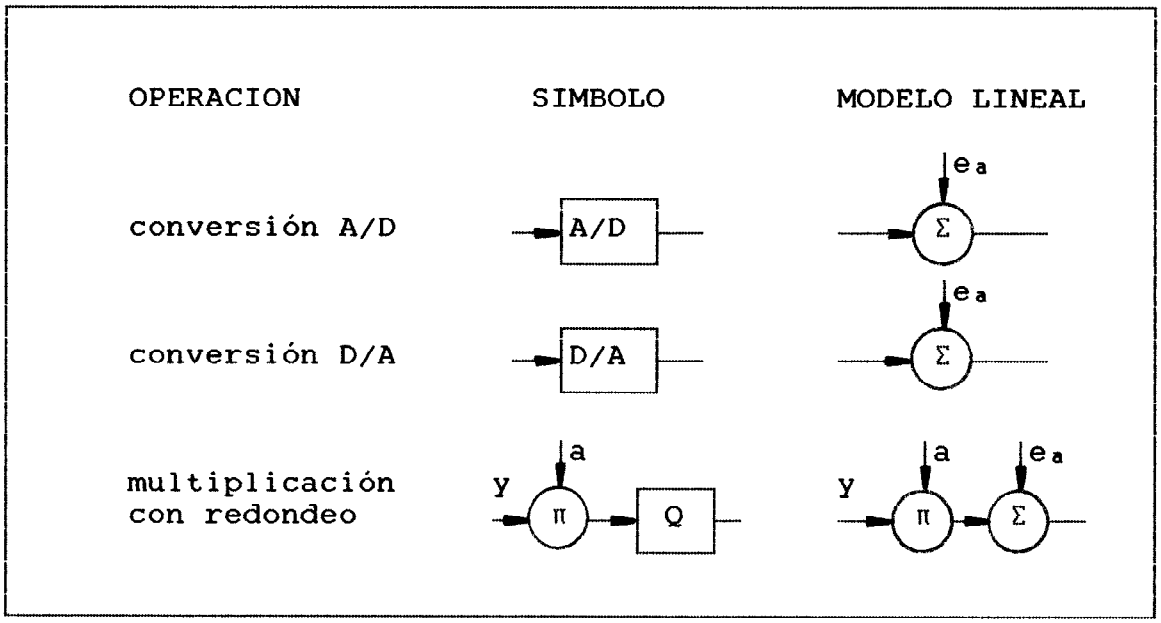


Fig 3.15. Modelos lineales para cuantización y redondeo.

Los errores pueden ser modelados como señales deterministas o estocásticas. En el primer caso serán constantes, dependiendo de la resolución de los operaciones aritméticas y del tamaño de la cuantización. En el segundo, el error es descrito como ruido blanco aditivo con una distribución rectangular. Siendo $\delta=2^{-i}$ la unidad de cuantización, el error estará igualmente distribuido en el intervalo $(-\delta/2, \delta/2)$ en el caso de redondeo y en $(0, \delta)$ en el de truncación, como se muestra en la figura 3.16. Como en los 2 casos el intervalo es de longitud δ se tendrá:

varianza $\sigma^2 = \delta^2/12$ (3.44)

media $e_c = \delta/2$ (truncación) (3.45)

$e_c = 0$ (redondeo) (3.46)

Esta expresión sólo es válida suponiendo la distribución uniforme del error en el intervalo de longitud δ, lo cual es cierto únicamente bajo ciertas condiciones, como se indica en Boite (1983), particularmente si la amplitud de la señal que se cuantiza no es demasiado pequeña.

En el caso en que no se cumplan las condiciones anteriores, por ejemplo cerca del régimen permanente, en que las variaciones de las señales son pequeñas comparadas con δ, se muestra la naturaleza no lineal de la cuantización dando como resultado ciclos límite.

Si sólo existe una no-linealidad en el bucle, es posible usar el método de la función descriptiva para determinar aproximadamente

los ciclos límite. La función descriptiva caracteriza la transmisión de una señal sinusoidal de amplitud A a través de la no-linealidad.

El método predice la existencia de un ciclo límite cuando:

$$G(e^{j\omega T}) = -1/Y(A) \tag{3.47}$$

siendo Y(A) la función descriptiva de la no-linealidad (véase la figura 3.17(a)). La frecuencia w_1 y amplitud A_1 en el punto de intersección serán las estimadas para el ciclo límite.

La función descriptiva de una cuantización por redondeo tiene por expresión:

$$Y(A) = \begin{cases} 0 & 0 < A < \delta/2 \\ \dfrac{4\delta}{A} \displaystyle\sum_{i=1}^{n} \sqrt{\left(1-\left[\frac{2j-1}{2A}\delta\right]^2\right)} & \dfrac{2n-1}{2}\delta < A < \dfrac{2n+1}{2}\delta \end{cases} \tag{3.48}$$

cuya representación se muestra en la figura 3.17(b).

Estos ciclos se deben frecuentemente a la cuantización en la conversión A/D y D/A, pero pueden ser causados también fácilmente por cuantizaciones en los cálculos. Cuando existen diversas fuentes al mismo tiempo, el análisis de los ciclos límites en el sistema de control en bucle cerrado es complicado. En estos casos puede resultar más atractivo comprobar los efectos directamente vía

simulación, como se muestra en la figura 3.18, para el sistema con función de transferencia:

$$G(z) = \frac{0.25\ K}{(z - 1)\ (z - 0.5)} \qquad (3.49)$$

con un nivel de cuantización δ=0.2 y K=1.2 y 1.6. Se muestra claramente (Astron y Wittenmark, 1984) el ciclo límite que aparece en el segundo caso.

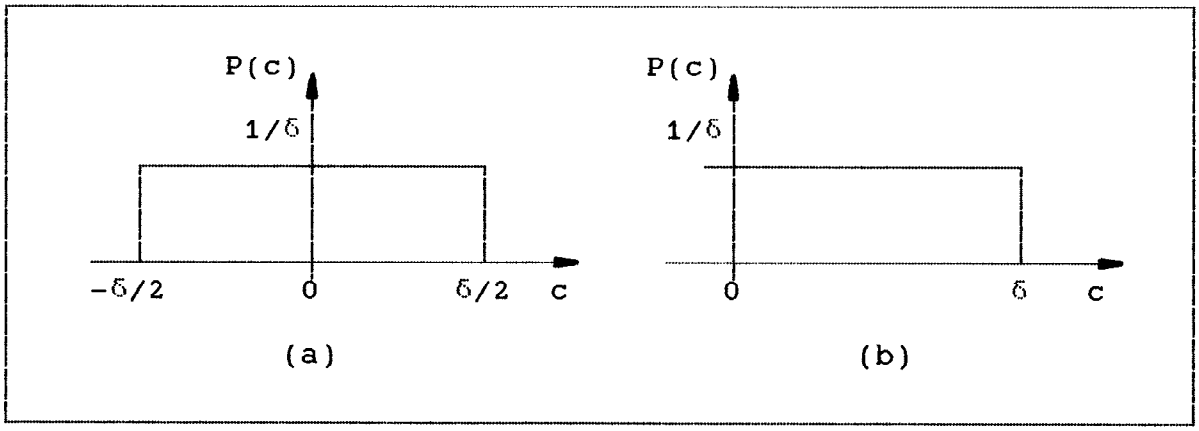


Fig 3.16. Densidad de probabilidad del error de cuantización para (a) redondeo (b) trucación

3.8.4 Influencia del período de muestreo.

Considerando un sistema de primer orden con constante de tiempo T:

$$x((k+1)\ T_m) = a\ x(k\ T_m) + b\ u(k\ T_m) \qquad a = e^{-T_m/T} \qquad (3.50)$$

se tendrá:

$$\frac{dT}{T} = - \frac{T}{T_m} \frac{da}{a} \tag{3.51}$$

con lo que, para una precisión relativa dada, la constante de tiempo equivalente será inversamente proporcional al período de muestreo.

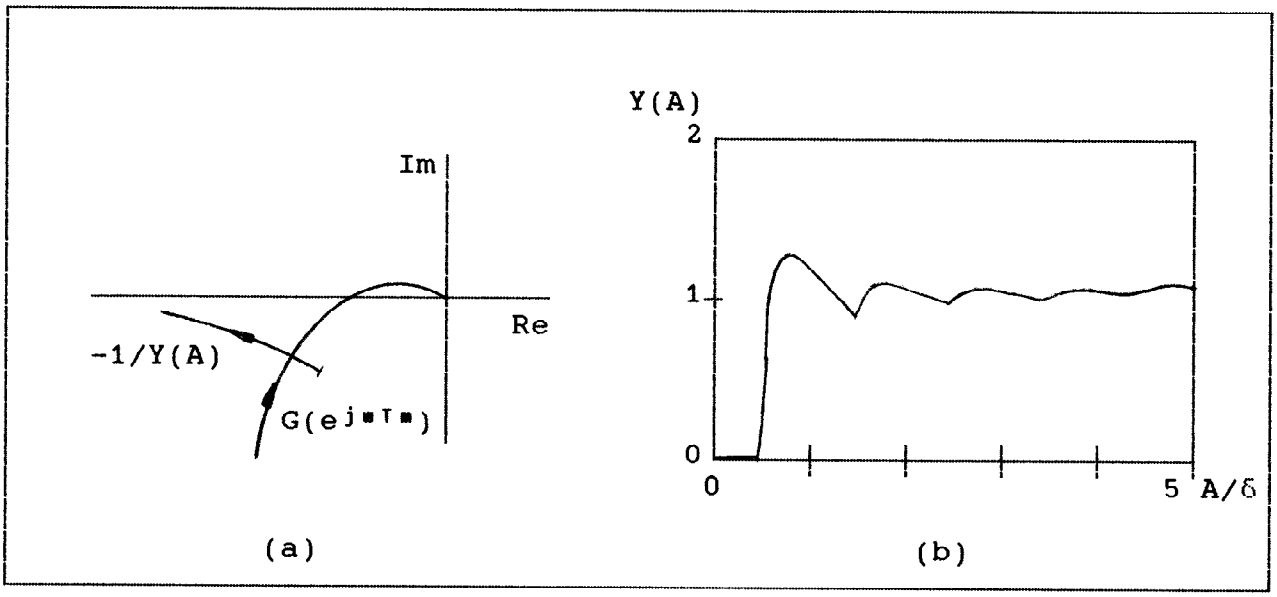


Fig 3.17. Función descriptiva en caso de cuantización por redondeo.

Así mismo, en un controlador PI se tendrá, como corrección de la integral en cada período de muestreo:

$$i((k+1)\ T_m) = i(k\ T_m) + e_m(k\ T_m)\ T_m/T_i \tag{3.52}$$

siendo T_i la constante de integración. Asumiendo que T_i sea mucho mayor que T_m, el cambio en la salida puede ser menor que la unidad

de cuantización. Por ello generalmente la integral se implementa con un mayor número de bits en los controladores PI dedicados.

Debido a lo anteriormente citado, ante un aumento de la frecuencia de muestreo se requiere generalmente mayor precisión en los coeficientes.

3.9 HARDWARE.

Una descripción del espectro del hardware disponible en la actualidad, para la implementación de sistemas de control digital, puede encontrarse en Hanselmann (1987), aunque, como el mismo autor reconoce, es necesario recalcar el término "en la actualidad", pues en pocos años tal descripción podrá parecer obsoleta.

En el escalón más alto del espectro, en cuando a potencia computacional y coste, se encuentran las máquinas especiales para experimentación rápida, computadores de alta velocidad específicamente diseñados para computación y adquisición de datos en tiempo real. Un ejemplo de los mismos es el AD10 de Applied Dynamics International.

Estos sistemas son particularmente atractivos en primeros pasos de proyectos de diseño e implementación de sistemas de control, en orden a obtener la realimentación necesaria, a partir de los resultados experimentales, tan pronto como sea posible, con las facilidades de programación flexible, aritmética de coma flotante,

gran velocidad ...

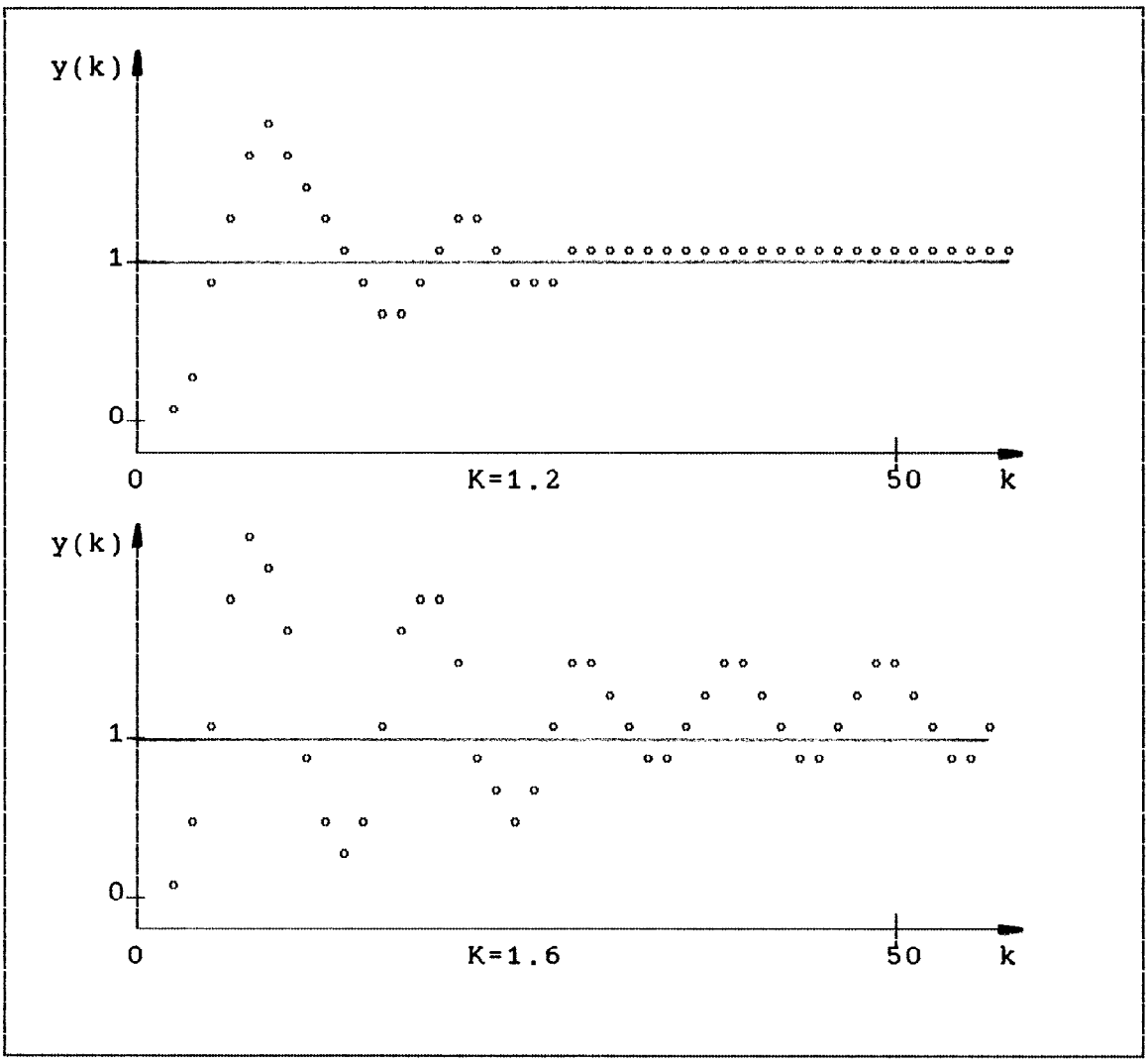


Fig 3.18. Análisis de los ciclos límites vía simulación.

Pueden ser usados, con los mismos objetivos, minicomputadores dotados con varios procesadores en paralelo, obteniéndose una potencia de cálculo similar pero aún a mayor coste. Sin procesadores en paralelo, la velocidad de los minicomputadores es considerablemente menor. Un sistema de este tipo de bajo coste es el PC 1000 de Systolic Systems Inc., especialmente recomendado para

evaluación experimental de sistemas de control lineal.

También puede conseguirse aproximadamente la misma velocidad de computación, con sistemas basados en los denominados conjuntos de chips "word-slice" de Advanced Micro Devices y Analog Devices, los cuales comprenden todos los bloques necesarios para desarrollar procesadores de señal microprogramados de alta velocidad, con solo unos pocos chips, entre los cuales se encuentran chips aritméticos de propósito especial para cálculos en coma flotante.

Es posible trabajar con aritmética de coma flotante en el mismo rango de velocidad con chips aritméticos separados para adición y multiplicación de 32 bits, por ejemplo de Weitek Corporation.

La implementación y prueba de controladores a mucho menor coste es posible usando computadores personales o tarjetas microcomputador, equipados con interfaces con el proceso y coprocesadores matemáticos, como los conjuntos NS 32016/32081 de National Semiconductor o 80286/80287 de Intel.

Estos sistemas son fáciles de programar en lenguajes de alto nivel y proporcionan velocidad media suficiente para implementar complejos sistemas de control en muchos casos. Para el tratamiento de sistemas especialmente complejos y rápidos puede incorporarse a la solución anterior multiplicadores hardware rápidos.

Otra solución para la implementación puede ser el uso de microcon-

troladores, término usado para designar a microprocesadores con funciones especiales en un único chip, diseñados para su uso como procesadores dedicados. Dispositivos de este tipo son el MK68200 de Motorola, el Nippon Electric NEC µPD 78312 y el Intel 8096. La velocidad de cálculo no es muy superior, típicamente, a la de los µP de propósito general, pero incluyen en el chip funciones como conversión A/D, líneas de E/S digital, contadores, etc., que los hacen especialmente atractivos.

Con la implementación en base a procesadores de señal es posible conseguir una alta velocidad de cálculo con un coste bajo. En la figura 3.19 se muestra el núcleo de la arquitectura de un procesador de señal típico.

Aparte del bus múltiple y de la estructura del camino de datos, la diferencia más significativa con un µP o µC general es el multiplicador hardware paralelo integrado, que puede realizar el producto de 16x16 bit en cada ciclo de máquina, con lo se realiza tan rápidamente como las adiciones o las operaciones lógicas.

Existe una gran actividad en el desarrollo de este tipo de dispositivos, existiendo varios disponibles como el Nippon Electric NEC 7720, Fujitsu MB 8784, STC DSP 128, Texas Instruments TMS 32010 y TMS 32020.

Para aplicaciones de medio a alto volumen, el diseño "a la medida" proporciona bloques como multiplicadores, unidades aritméticas,

etc. Existen así mismo compiladores de silicio que parten de una descripción de alto nivel de lo que se desea que el procesador de señal realice, eligiendo el software los paramétros óptimos de la arquitectura del chip.

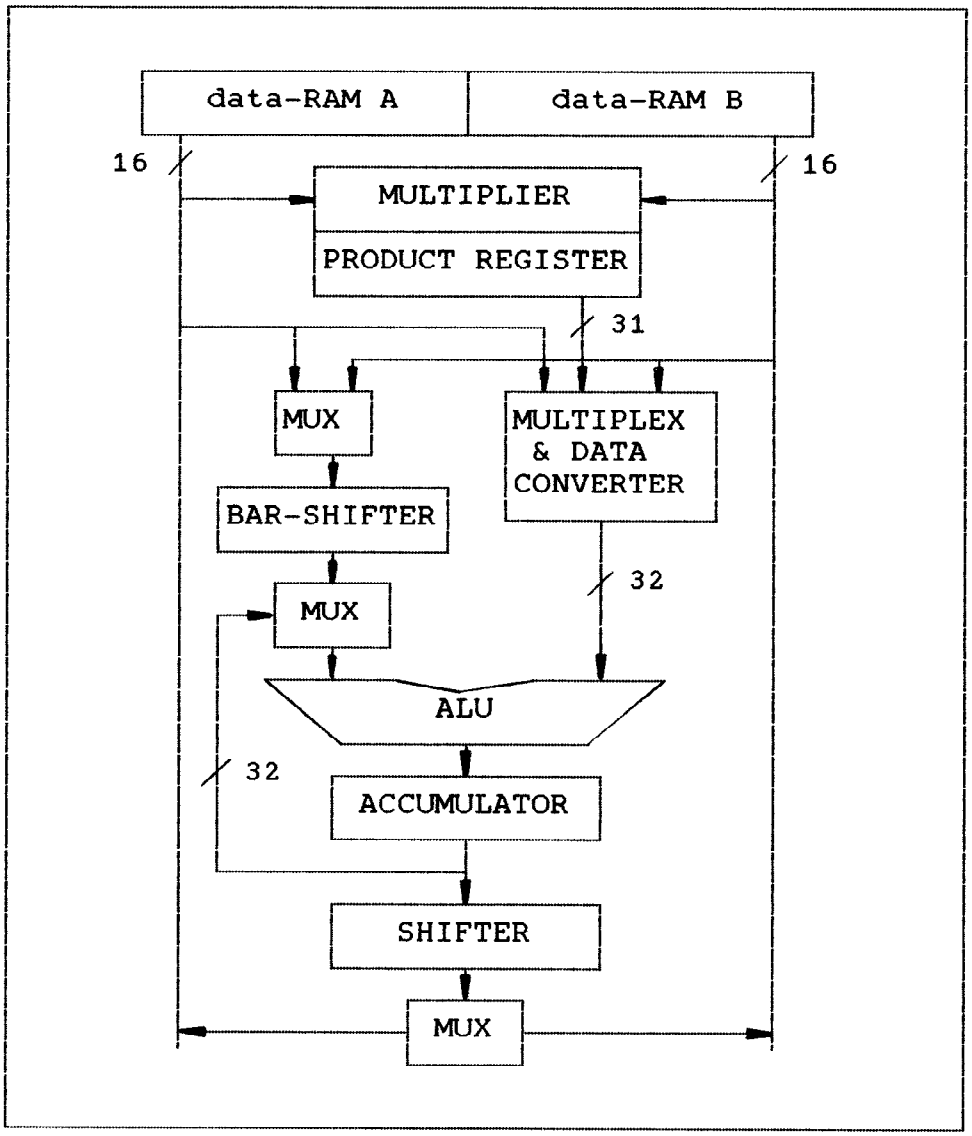


Fig 3.19. Núcleo de procesador de señal típico.

Una comparación entre algunos microprocesadores y procesadores de señal puede verse en Hanselmann y Loges (1984) en cuanto a la

máxima frecuencia de muestreo que se puede obtener. En la tabla 3.4 se hace dicha comparación con el procesador de señal TMS 32010, basada en la implementación de un controlador de orden 9, con sólo una entrada y una salida, usando aritmética de coma fija.

μP	Clock (MHz)	f_m (kHz)
8086	8	<2
Z8000	5	<2
68000	10	<4
32016	10	<5
TMS 32010 (proc. señal)		31

Tabla 3.4. Frecuencia de muestreo con diferentes μP y procesadores de señal.

3.10 INFLUENCIA DEL RETARDO COMPUTACIONAL.

Aunque generalmente se considera que existe sincronismo entre la generación de la secuencia por el muestreador a partir de la salida del proceso $\{y_k\}$, y la aplicación de la calculada por el computador en base a la ley de control $\{u_k\}$, en la realidad entre ambos sucesos existe un tiempo necesario para realizar las conversiones A/D y D/A, así como los cálculos por parte del computador.

Este tiempo hace que la acción de control se aplique a la entrada del actuador con un cierto retraso respecto a los instantes de muestreo. Es lo que se denomina retardo computacional o tiempo de cálculo.

Generalmente este tiempo de cálculo (αT_m) es suficientemente pequeño en comparación con el período de muestreo y, si no es necesaria una elevada precisión en los cálculos, puede despreciarse su efecto. En el caso de que αT_m sea cercano a T_m, puede tenerse en cuenta en los cálculos incluyendo un nuevo bloque con un retardo de valor αT_m a la entrada del muestreador, como se indica en la figura 3.20. La función de transferencia entre $\{y_k\}$ y $\{w_k\}$ será en este caso (Aracil y Jiménez, 1982):

$$\frac{y(z)}{w(z)} = \frac{G_R(z)\ BG(z)}{1 + G_R(z)\ BGF(z,1-\alpha)} \tag{3.53}$$

siendo:

$$BG(z) = (1-z^{-1}) \sum_{\text{polos } G_P(p)/p} \text{Residuos} \left[\frac{G_P(p)}{p} = \frac{1}{1-e^{pT_m}\ z^{-1}}\right] \tag{3.54}$$

$$BGF(z,1-\alpha)=z^{-1}\ (1-z^{-1}) \sum_{\text{polos } G_P(p)F(p)/p} \text{Residuos} \left[\frac{G_P(p)\ F(p)}{p} = \frac{e^{pT_m(1-\alpha)}}{1-e^{pT_m}\ z^{-1}}\right] \tag{3.55}$$

Otra forma de afrontar el problema es usar las variables leídas en el período de muestreo k para calcular la señal de control a aplicar en el k+1 (figura 3.21 (a)) en vez de realizar la conversión D/A tan pronto como sea posible (figura 3.21 (b)). De esta manera se sustituye un retardo variable por uno fijo de un período de muestreo, siendo necesario tenerlo en cuenta en la ley de

control. Esto puede realizarse incluyendo un retardo de T_m en el modelo del proceso.

En el caso de la figura 3.21 (b), es conveniente que α sea lo menor posible. Para ello se deben realizar las mínimas operaciones posibles entre las conversiones A/D y D/A.

Para juzgar las consecuencias del retardo computacional, es útil ver la influencia del comportamiento del sistema en cadena cerrada con respecto a α, por ejemplo realizando el lugar de las raíces. Esto es más sencillo de realizar en el caso de suponer un retardo fijo de un período de muestreo.

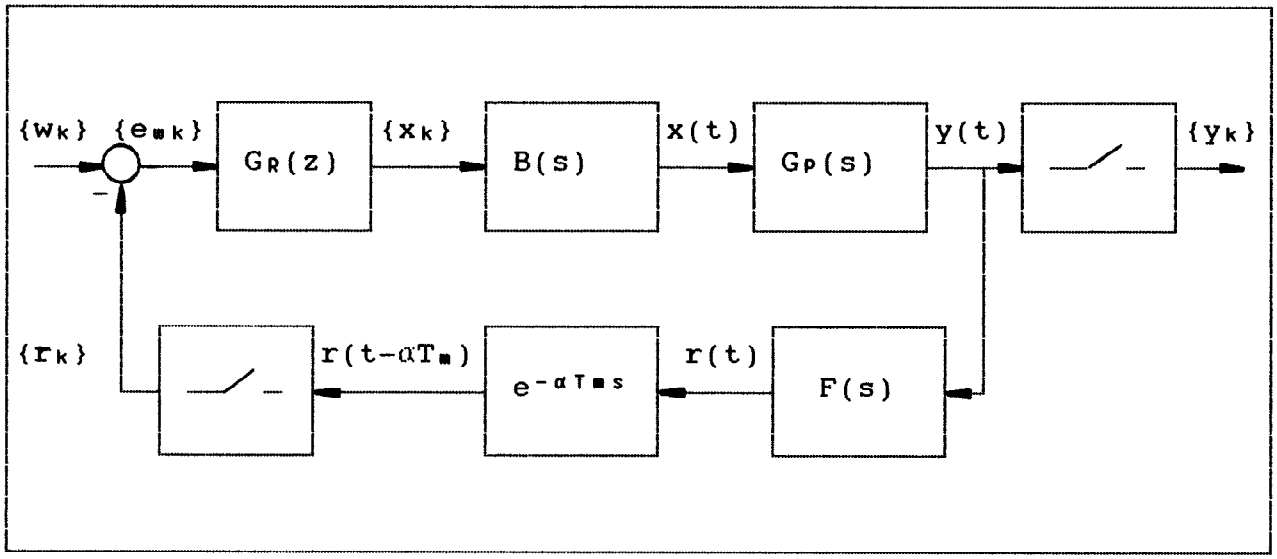


Fig 3.20. Inclusión del efecto del tiempo de cálculo.

Así mismo es útil tener buenas estimaciones de los tiempos empleados en la computación de los diferentes algoritmos de control, estimación, etc. que componen el sistema. En microcomputadores

simples, que no disponen de hardware que implemente la aritmética en coma flotante, existe una diferencia sustancial en el tiempo de cálculo cuando se usa coma fija o flotante. En el caso de la existencia de dicho hardware, la diferencia es mucho menor.

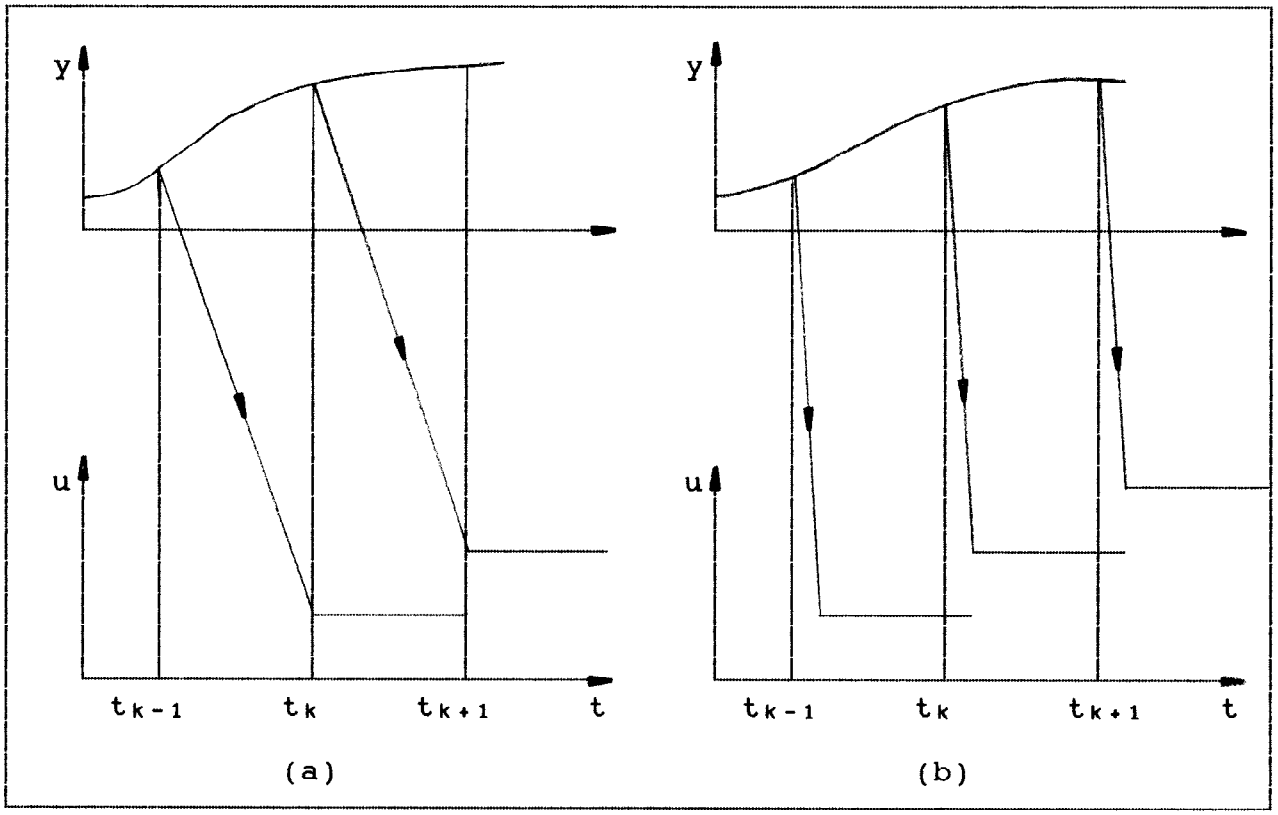


Fig 3.21. Sincronización de entrada/salida.

CAPITULO 4

SISTEMA DESARROLLADO Y RESULTADOS OBTENIDOS

4. SISTEMA DESARROLLADO Y RESULTADOS OBTENIDOS.

4.1 DESCRIPCION DEL SITEMA MASTER.

4.1.1 Estructura.

El sistema MASTER (Sistema Modular para Control Adaptativo en Tiempo Real) se ha implementado en base a un computador compatible PC-AT, dotado de una tarjeta de interface DASH-16F de Metrabyte Corporation.

Se ha utilizado Modula-2 como lenguaje de programación, por ser un lenguaje de alto nivel pequeño y compacto, adecuado para la implementación de sistemas en tiempo real. De esta forma, cumple con los criterios previamente especificados en el apartado 3.7, a la vez que es especialmente adecuado en el caso de sistemas de complejidad media-baja, como el MASTER.

La estructura del sistema MASTER es bitarea (background y foreground), constando cada tarea de una serie de procedimientos (incluidos en módulos), que se ejecutan secuencialmente. El paso del control por parte del background al foreground, se realiza cuando se produce una interrupción, que marca un nuevo período de muestreo o de adquisición de datos en tiempo real.

4.1.2 Módulos.

Los módulos incluidos en el sistema MASTER pueden ser agrupados funcionalmente en la forma siguiente:

- Módulos de control adaptativo.
- Módulos de interface con el usuario.
- Módulos de interface con el proceso.
- Módulos de gestión de dispositivos.
- Módulos auxiliares.

4.1.3 Evaluación de los algoritmos implementados.

Para la evaluación de los algoritmos implementados en tiempo real se dispone de una placa de pruebas cuyas características se especifican en el apartado 4.4.2, a la que es posible conectar directamente procesos externos. Esto se realiza así con la planta piloto descrita en el apartado 4.7.

Para incrementar el número y la variedad de características de los procesos disponibles para la evaluación de los algoritmos, se ha seleccionado un subconjunto de los existentes en el subdirectorio de procesos, en orden a realizar diversas simulaciones. Las funciones de transferencia de estos procesos son las siguientes:

$$P1.1 \quad G_P(z^{-1}) = \frac{0.4}{1.0 - 0.6\, z^{-1}}\, z^{-1} \qquad (4.1)$$

$$P2.1\ G_P(z^{-1}) = \frac{0.087 + 0.063\ z^{-1}}{1.0 - 1.227\ z^{-1} + 0.376\ z^{-2}}\ z^{-1} \qquad (4.2)$$

$$P2.2\ G_P(z^{-1}) = \frac{0.087 + 0.063\ z^{-1}}{1.0 - 1.227\ z^{-1} + 0.376\ z^{-2}}\ z^{-3} \qquad (4.3)$$

$$P2.3\ G_P(z^{-1}) = \frac{0.035 + 0.029\ z^{-1}}{1.0 - 1.492\ z^{-1} + 0.556\ z^{-2}}\ z^{-1} \qquad (4.4)$$

$$P2.4\ G_P(z^{-1}) = \frac{-0.1 + 0.173\ z^{-1}}{1.0 - 1.425\ z^{-1} + 0.496\ z^{-2}}\ z^{-1} \qquad (4.5)$$

$$P2.5\ G_P(z^{-1}) = \frac{0.105 + 0.089\ z^{-1}}{1.0 - 1.609\ z^{-1} + 0.609\ z^{-2}}\ z^{-1} \qquad (4.6)$$

$$P2.6\ G_P(z^{-1}) = \frac{1.0 - 2.0\ z^{-1}}{1.0 - 4.0\ z^{-1} + 3.0\ z^{-2}}\ z^{-1} \qquad (4.7)$$

$$P2.7\ G_P(z^{-1}) = \frac{0.38 - 0.3\ z^{-1}}{1.0 - 2.78\ z^{-1} + 1.87\ z^{-2}}\ z^{-1} \qquad (4.8)$$

$$P3.1\ G_P(z^{-1}) = \frac{0.11 + 0.08\ z^{-1} - 0.023\ z^{-2}}{1.0 - 1.654\ z^{-1} + 1.022\ z^{-2} - 0.202\ z^{-3}}\ z^{-4} \qquad (4.9)$$

$$P3.2\ G_P(z^{-1}) = \frac{0.11 + 0.08\ z^{-1} - 0.023\ z^{-2}}{1.0 - 1.654\ z^{-1} + 1.022\ z^{-2} - 0.202\ z^{-3}}\ z^{-1} \qquad (4.10)$$

$$P3.3\ G_P(z^{-1}) = \frac{0.105 + 0.089\ z^{-1}}{1.0 - 1.9\ z^{-1} + 1.27\ z^{-2} - 0.18\ z^{-3}}\ z^{-2} \qquad (4.11)$$

$$P4.1\ G_P(z^{-1}) = \frac{0.018 + 0.17\ z^{-1} + 0.036\ z^{-2} - 0.0024\ z^{-3}}{1.0 - 1.11\ z^{-1} + 0.36\ z^{-2} - 0.03\ z^{-3} + 0.05\ z^{-4}}\ z^{-1} \qquad (4.12)$$

El proceso P1.1 tiene por coeficientes de la función de transferencia una aproximación de los de la planta de primer orden implementada en la placa de pruebas, cuando se emplea un período de muestreo de T_m=500 ms. La función de transferencia de los procesos P2.1 y P2.2 se aproxima a la de la planta de segundo orden implementada físicamente, cuando T_m=500 ms y d=1 o d=3, respectivamente. P2.3 corresponde a la misma planta con T_m=300 ms.

4.2 CONTROL ADAPTATIVO.

4.2.1 Sistema de control adaptativo.

En éste módulo se realizan las llamadas necesarias, en cada período de muestreo, a los procedimientos incluidos en el resto de los que componen el sistema, de forma que se ejecute la opción seleccionada por el usuario, mediante el menú correspondiente, de entre las indicadas a continuación (las cuatro primeras pueden ser llevadas a cabo trabajando en tiempo real o mediante simulación):

- Estimación de los parámetros del proceso.
- Control fijo de un proceso con el método seleccionado.
- Control adaptativo de un proceso con el método seleccionado.
- Control adaptativo de un proceso con el método de control seleccionado, incorporando un nivel de supervisión.
- Cálculo de la función de transferencia del controlador para un proceso determinado, con el método seleccionado.

- Adquisición de datos.

De esta forma el sistema MASTER puede encontrarse, según las opciones elegidas, funcionando en modo Menú o modo Ejecución. En este segundo caso, en los modos Simulación o Tiempo Real y, en cada uno de éstos, en Identificación o Control (Fijo, Adaptativo o con Supervisión). En Tiempo Real se tienen así mismo el modo Data Logging, los modos Interactivo o Batch para el intérprete de comandos, los modos Manual o Automático en Control y el modo Preidentificación en Control Adaptativo. Un esquema con todos los modos de funcionamiento puede observarse en la figura 4.1.

En el trabajo en tiempo real, se ejecuta en cada período de muestreo el procedimiento "Interrupción", llamado cuando se produce la interrupción indicando que existe dato disponible de la conversión analógico/digital o bien el procedimiento "IntData", en el caso de adquisición de datos, cuando se ha completado la transferencia de un bloque vía DMA.

También en este módulo se realizan las llamadas necesarias a los procedimientos de los módulos de interface con el usuario, interface con el proceso y gestión de dispositivos para representación gráfica o numérica.

4.2.2 Diseño de parámetros del controlador.

En este módulo se realiza la obtención, con el método de control

seleccionado mediante menú, de los coeficientes de la función de transferecia del controlador, en base a las estimaciones de los parámetros del proceso (para control adaptativo) obtenidas.

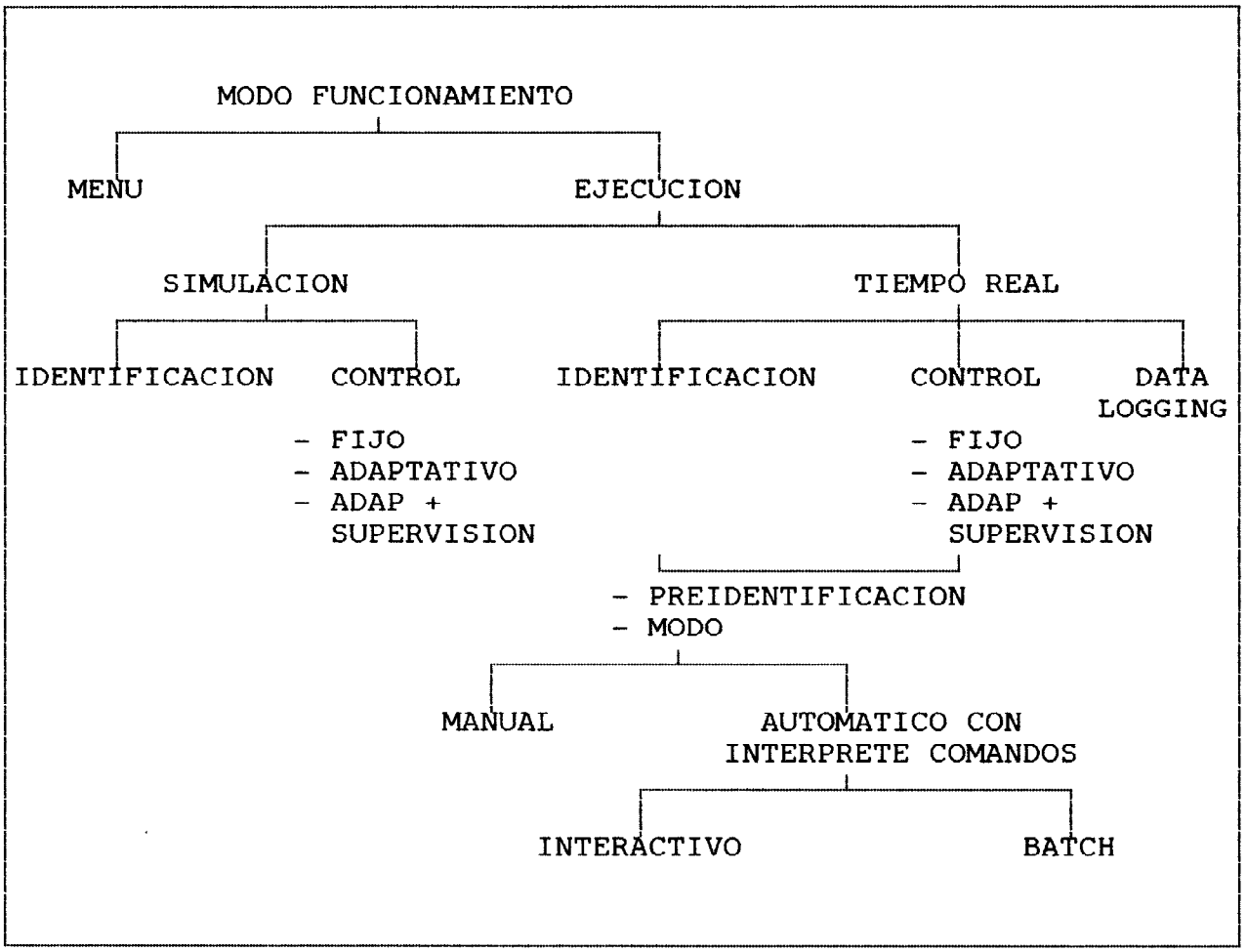


Fig 4.1. Modos de funcionamiento del sistema MASTER.

Hay que hacer salvedad de que esto no sucede así en el caso del algoritmo LRP (Long Range Predictive Control, efectuado con el método EPSAC de De Keyser y Van Cauwenberghe), en que la salida del mismo es directamente la señal de control que actúa sobre el proceso, necesitando como dato adicional el valor de la variable de

salida en el anterior período de muestreo.

El resto de los métodos de control programados son:

- DBN: Tiempo finito de orden normal.
- DBA: Tiempo finito de orden aumentado.
- APN: Asignación de polos de estructura simple.
- APA: Asignación de polos de estructura completa.
- APR: Asignación de polos robusta.

4.2.2.1 Controladores de Tiempo Finito (DBN y DBA).

Estos controladores, de orden normal o incrementado, no presentan parámetros de ajuste de la ley de control, una vez establecido el período de muestreo T_m, por lo que su aplicación es muy sencilla. El problema de los valores elevados resultantes para la señal de control se puede reducir en parte (para T_m fijo) utilizando un controlador de orden incrementado en lugar del de orden normal.

En las figuras 4.2 y 4.3 se muestran las señales de consigna, salida y control cuando se aplica DBN y DBA, respectivamente, al proceso P2.1. Como se ve, en el primer caso las señales u e y se estabilizan dos períodos de muestreo después del cambio de consigna. Cuando se aplica DBA las señales u e y se estabilizan un período de muestreo después que en el caso anterior, tomando la variable de control el mismo valor en los dos períodos de muestreo posteriores a un cambio y siendo mucho más reducido este valor que

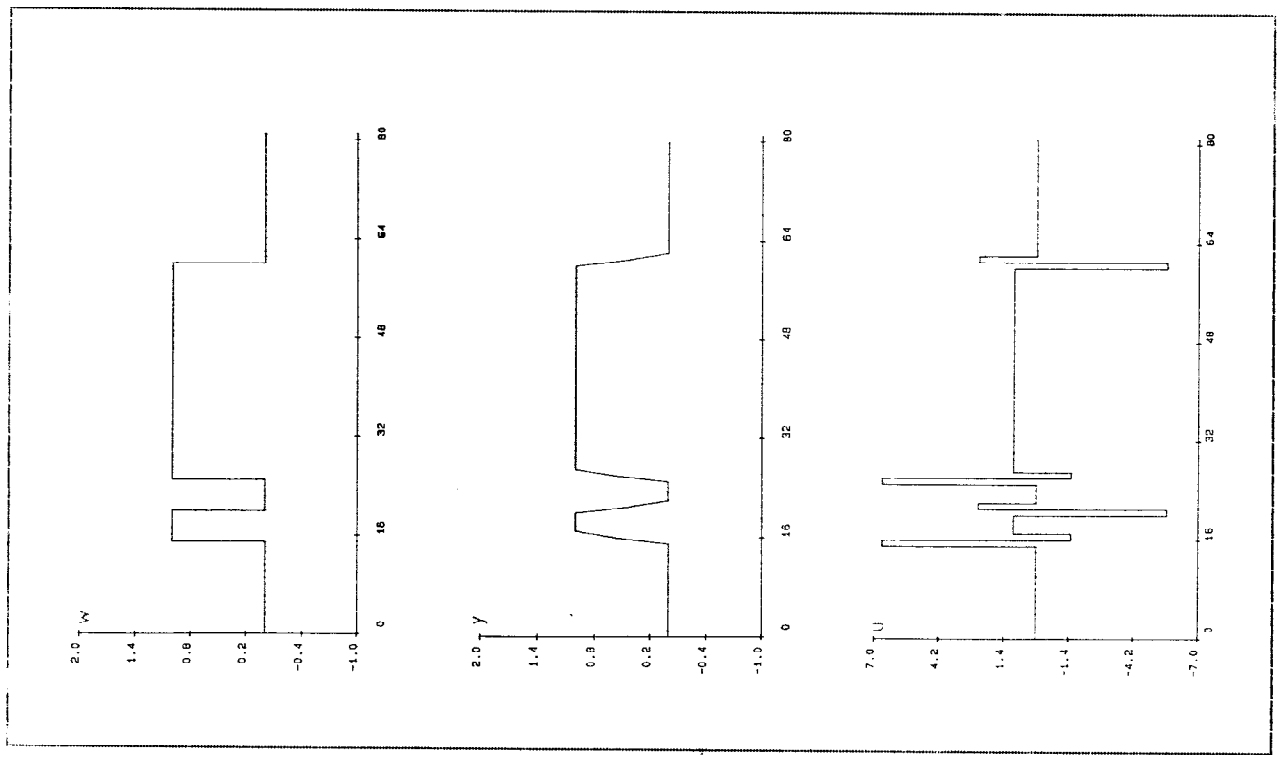


Fig 4.2. Control fijo con DBN del proceso P2.1.

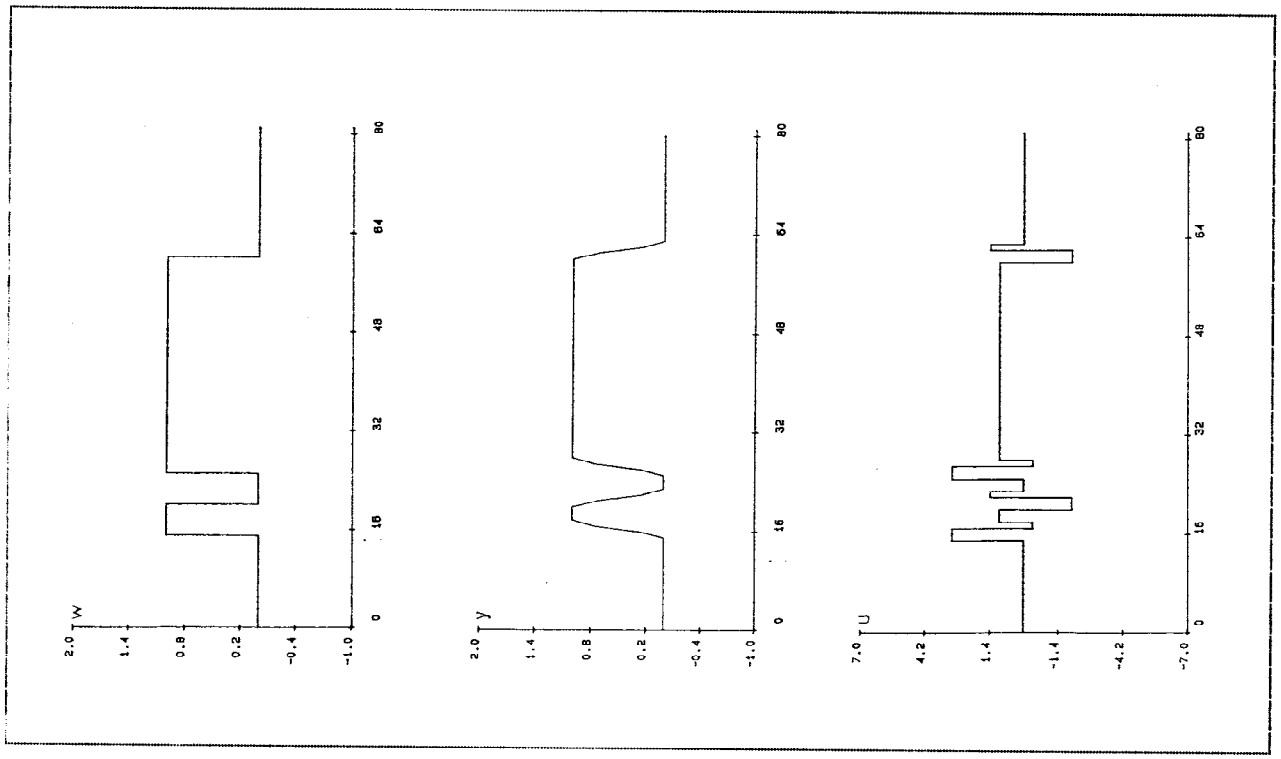


Fig 4.3. Control fijo con DBA del proceso P2.1.

el máximo que se produce con DBN.

Como se puede deducir de las figuras anteriores, no existen desviaciones en régimen permanente entre la variable de salida y la referencia. Esto es debido a la acción integral que presenta este tipo de controladores.

- Efecto de las perturbaciones deterministas.

En las figuras 4.4 y 4.5 se muestran las señales de consigna, salida y control cuando se aplica DBN y DBA, respectivamente, al proceso anterior, cuando actúa una perturbación determinista a la entrada del mismo, en escalón, desde k=60.

Como se puede comprobar, la señal de control se adecúa a la nueva situación en el mismo número de períodos de muestreo que ante cambio de consigna y no existe error en régimen permanente.

- Control adaptativo.

El comportamiento del sistema con control adaptativo es semejante al observado con control fijo, una vez que ha transcurrido el número de períodos de muestreo necesario para que se determine, con el método de identificación aplicado, un valor suficientemente aproximado de los coeficientes de la función de transferencia del proceso. Esto se podrá comprobar con las gráficas obtenidas con el sistema funcionando en tiempo real con control adaptativo.

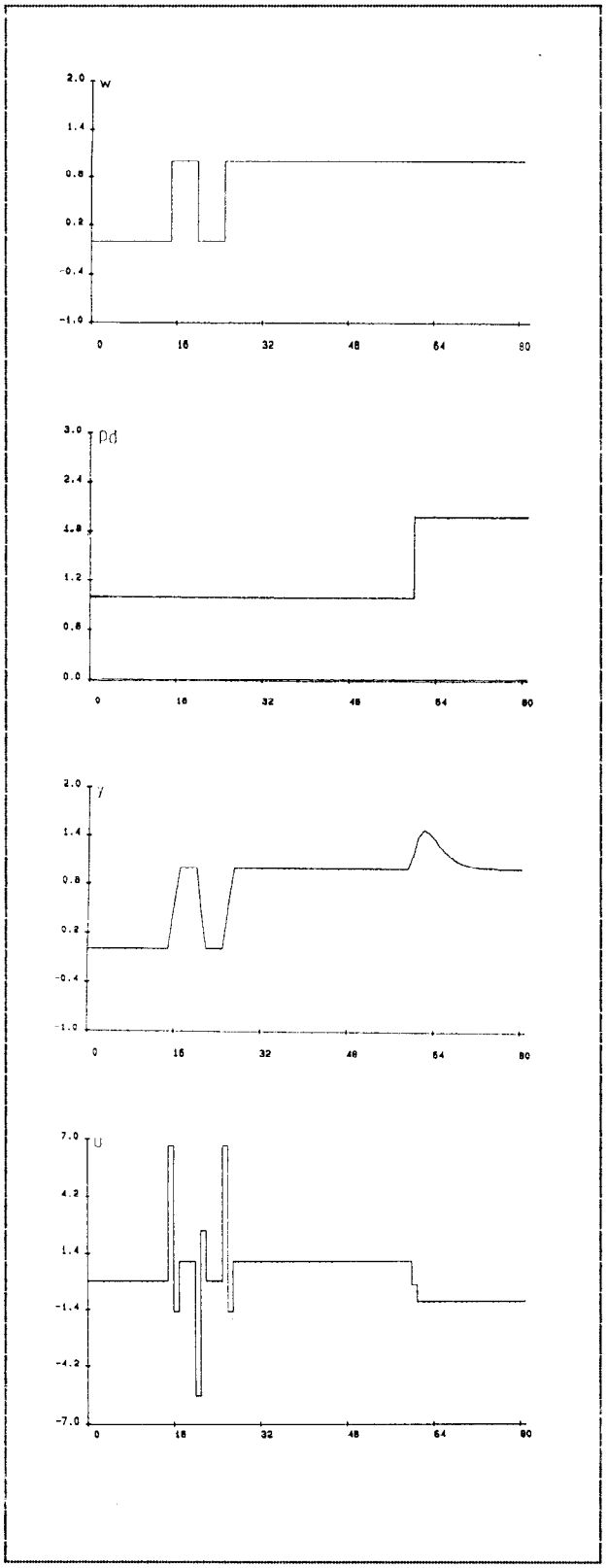


Fig 4.4. Efecto de perturbaciones deterministas con DBN.

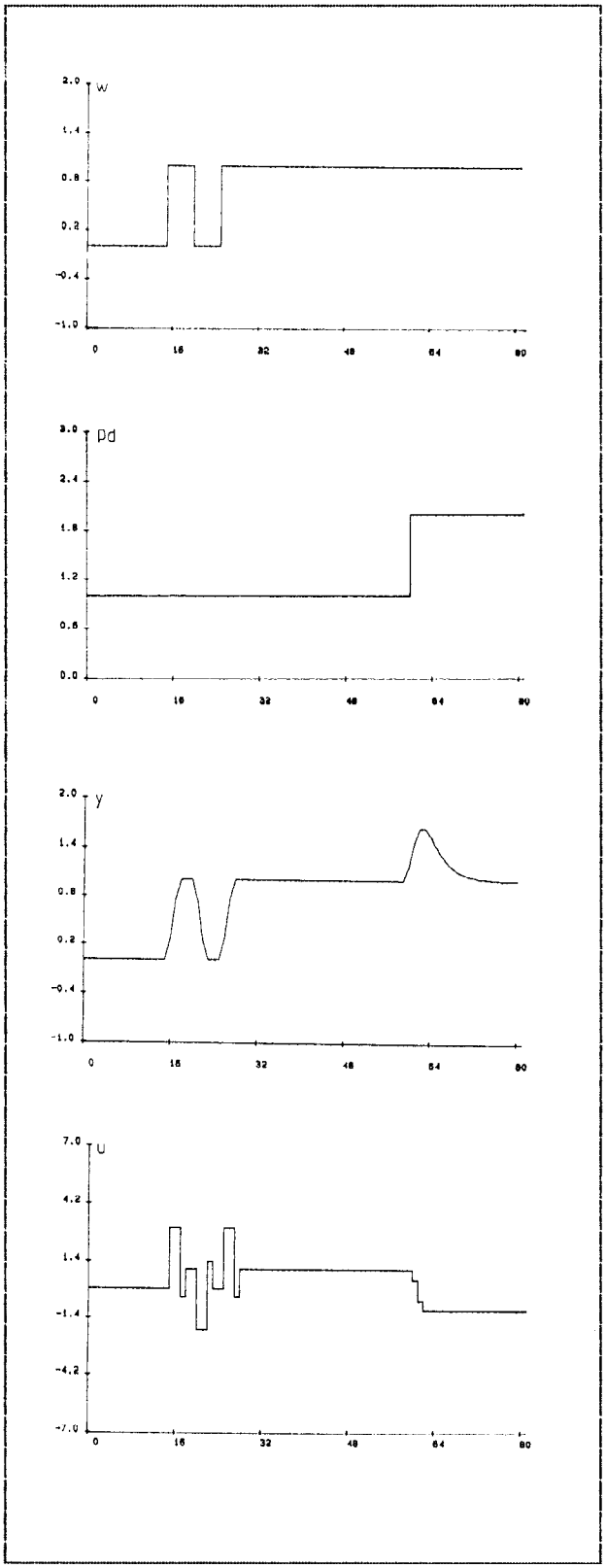


Fig 4.5. Efecto de perturbaciones deterministas con DBA.

- Control adaptativo en tiempo real.

En la figura 4.6 se pueden comprobar los efectos de una disminución del período de muestreo, cuando se aplica DBN al proceso de segundo orden implementado físicamente en la placa de pruebas, pasando de T_m=500 ms (4.6(a)) a T_m=300 ms (4.6(b)). En el segundo caso la acción de control en los cambios de consigna (se aplica la misma señal de consigna que en la figura 4.2) debe ser más fuerte que en el primero. En la figura 4.7 se muestran los resultados cuando, en los mismos casos anteriores, se aplica DBA.

En las figuras 4.8 y 4.9 se muestra el efecto de una perturbación determinista a la entrada del proceso, con DBN y DBA respectivamente, trabajando el sistema con control adaptativo en tiempo real y T_m=500 ms. Se aplica la misma consigna que en la figura 4.4.

4.2.2.2 Controladores de Asignación de Polos (APN y APA).

- Efecto de los parámetros de ajuste y las perturbaciones deterministas.

Los parámetros de ajuste tomados para el método de asignación de polos de estructura simple son:

- Coeficientes del polinomio formado con la posición deseada de los polos del sistema en cadena cerrada (A_m).

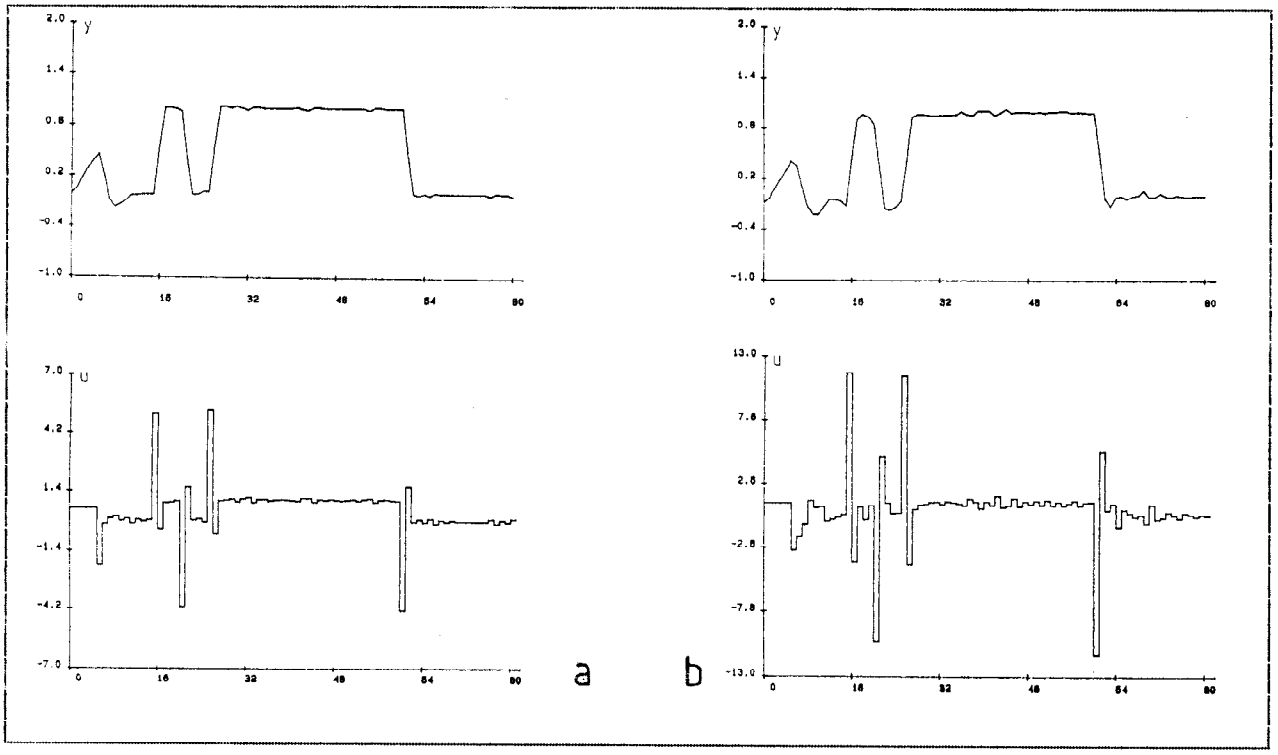


Fig 4.6. Efecto del cambio de período de muestreo con DBN.

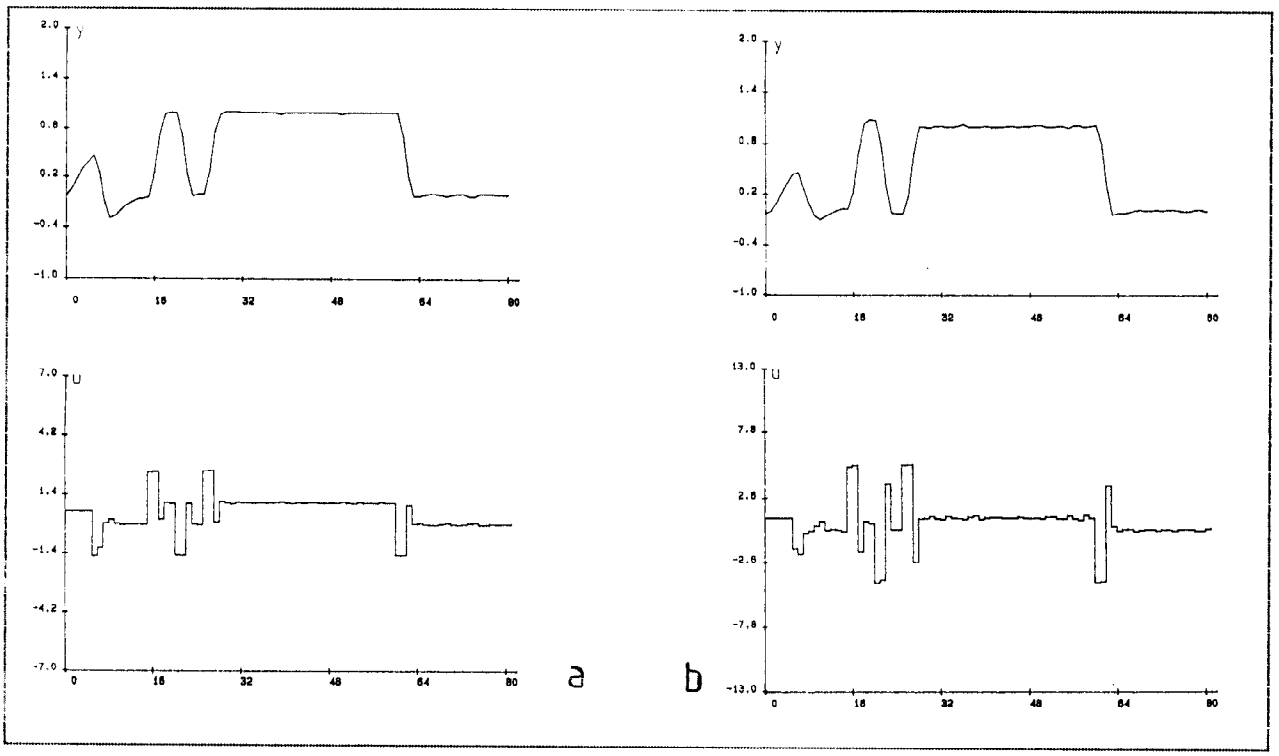


Fig 4.7. Efecto del cambio de período de muestreo con DBA.

- Inclusión adicional o no de un polo en z=1 en la función de transferencia resultante del controlador.

En el caso de asignación de polos de estructura completa los parámetros son los mismos, incluyendo:

- Coeficientes de polinomio observador de la perturbación (A_0).

El efecto de un cambio en los coeficientes del polinomio A_m es conocido. Si el comportamiento deseado para el sistema puede ser obtenido muestreando un sistema de segundo orden, la ecuación característica será:

$$z^2 + t_1 z + t_2 = 0 \tag{4.13}$$

$$t_1 = -2 e^{-f w T_m} \cos (w T_m \sqrt{(1-f^2)}) \tag{4.14}$$

$$t_2 = e^{-2 f w T_m} \tag{4.15}$$

influyendo f en el amortiguamiento y w en la velocidad de respuesta. En el caso en que los polos deseados para el problema de asignación se sitúen en el origen, la ecuación característica para el sistema en bucle cerrado será $z^n=0$, con lo que sólo queda como parámetro de diseño el período de muestreo T_m y se obtiene un comportamiento de tipo deadbeat (tiempo finito).

En las figuras 4.10 y 4.11 se muestran las señales de consigna, salida y control cuando se aplican APN y APA respectivamente, con control fijo, al proceso P2.5. Adicionalmente, se aplica una

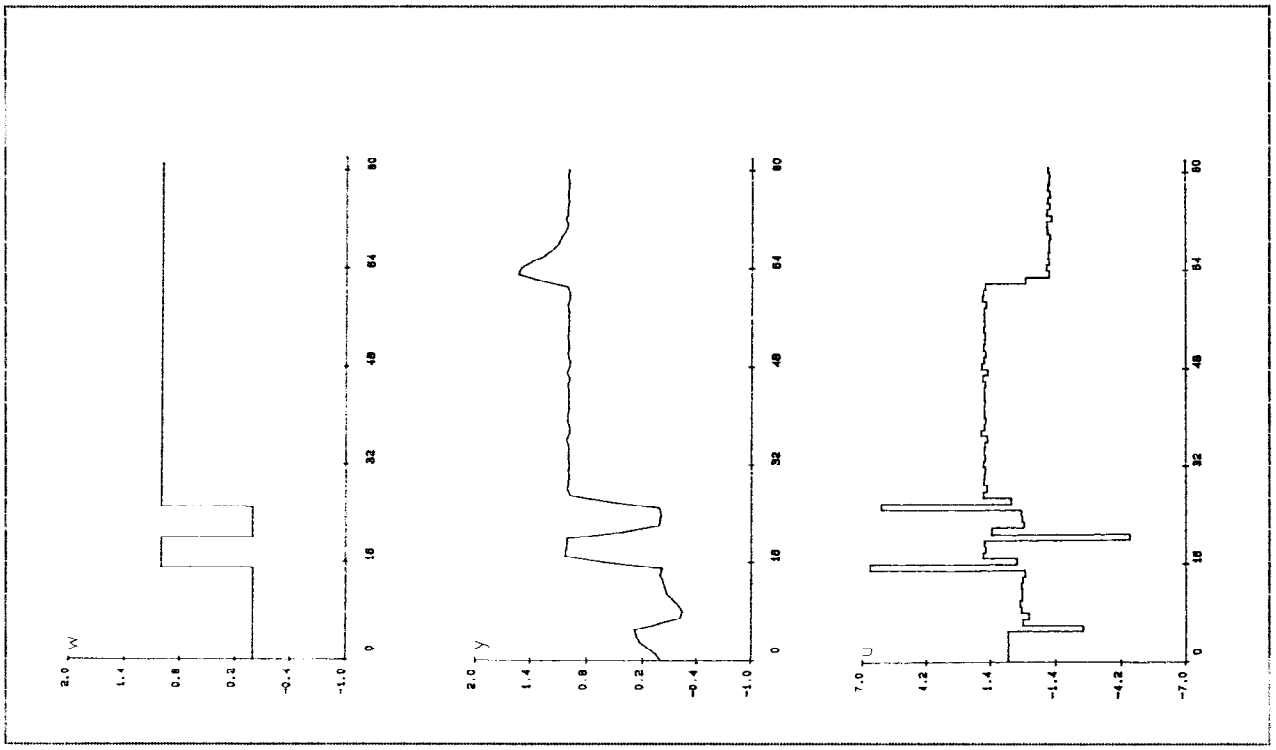


Fig 4.8. Efecto de perturbación determinista en tiempo real (DBN).

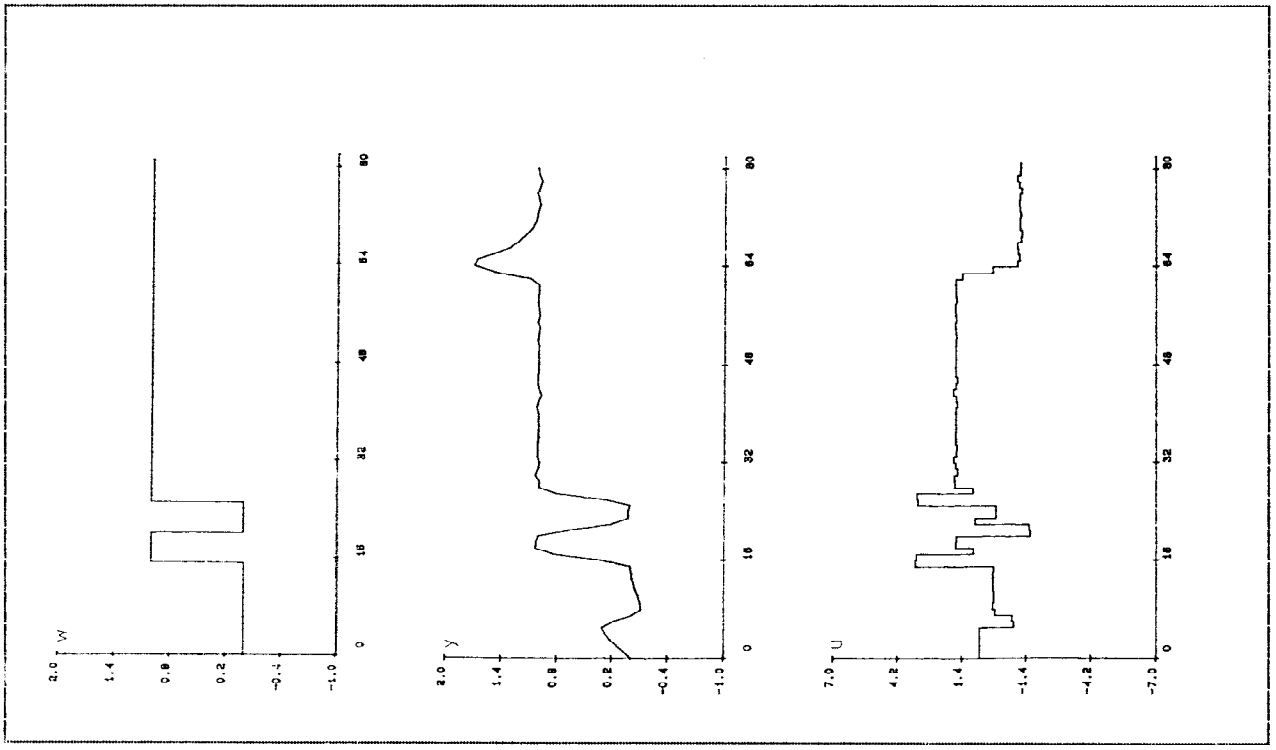


Fig 4.9. Efecto de perturbación determinista en tiempo real (DBA).

perturbación determinista a la entrada del proceso, en escalón, en k=60. Se ha elegido la posición de las raíces de A_m en el origen, con lo que se obtendrá un comportamiento del sistema en cadena cerrada de tipo tiempo finito.

Como se puede observar, en el primer caso la sobreoscilación es mucho mayor. Esto es debido a que con APN los ceros de la función de transferencia en cadena cerrada vienen en parte determinados por la ecuación (2.7), influyendo en el comportamiento. Como con APA esto no sucede así, se obtiene un mejor comportamiento con un menor esfuerzo de control. Se hace notar que, al poseer la función de transferencia del proceso un polo en z=1 la salida sigue a la consigna sin error en régimen permanente.

En las figuras 4.12 y 4.13 se muestra el efecto de la inclusión de un integrador en la función de transferencia del controlador, en los dos casos anteriores. Se puede observar en ambas figuras la rápida adecuación de la señal de control a su nuevo valor, haciendo que desaparezcan los efectos de la perturbación sobre la señal de salida.

En la figura 4.14 se puede comprobar el efecto del cambio en la velocidad de respuesta deseada para el sistema en cadena cerrada. Se ha aplicado control fijo con APN, incluyendo integrador en el controlador, sobre el proceso P2.5 con la misma consigna que en el caso anterior y con f=0.7 y w=2 (a), w=1 (b) y w=0.5 (c) respectivamente. Como se ve, la velocidad de respuesta (tanto ante cambio

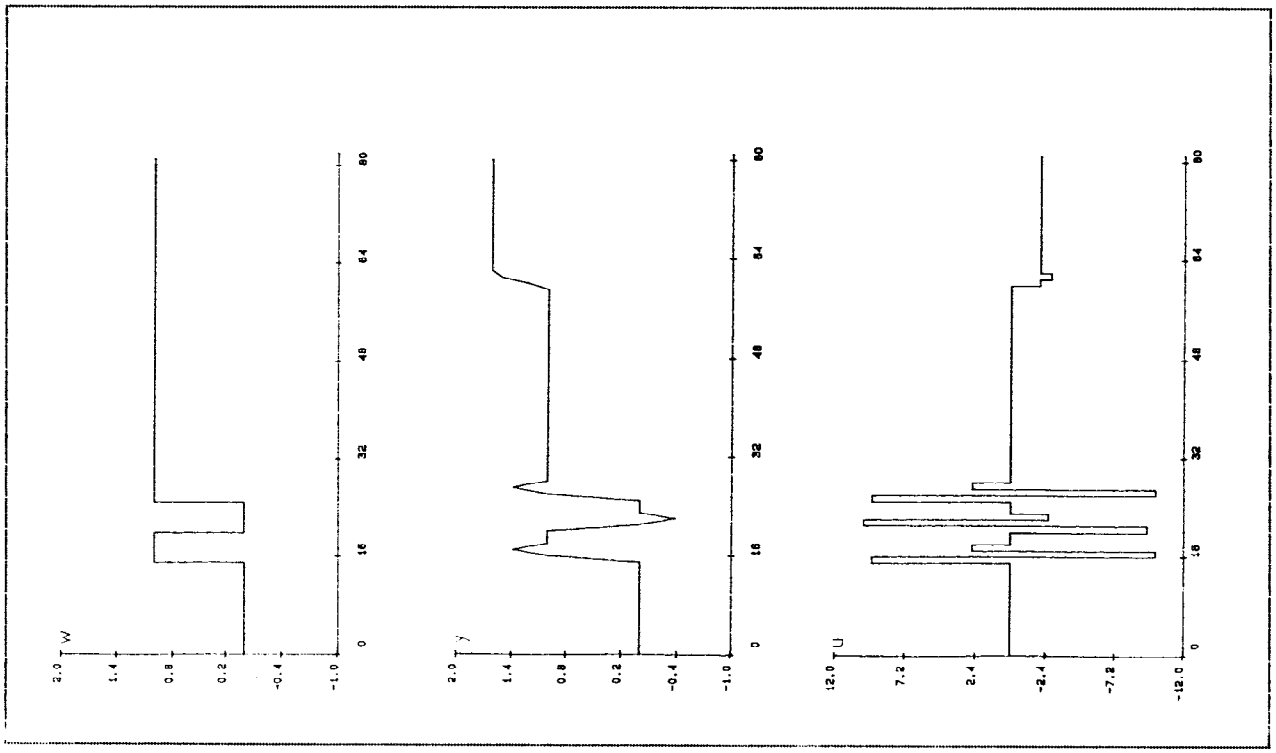


Fig 4.10. Control fijo con APN del proceso P2.5.

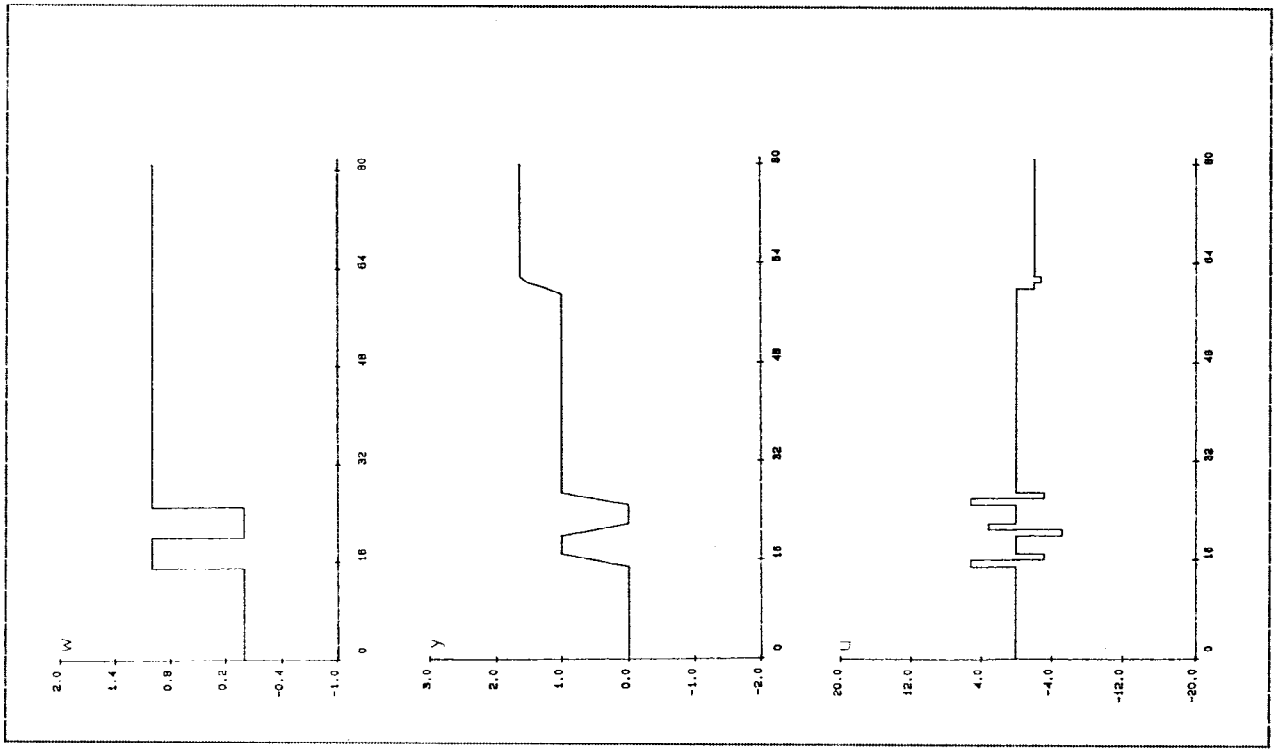


Fig 4.11. Control fijo con APA del proceso P2.5.

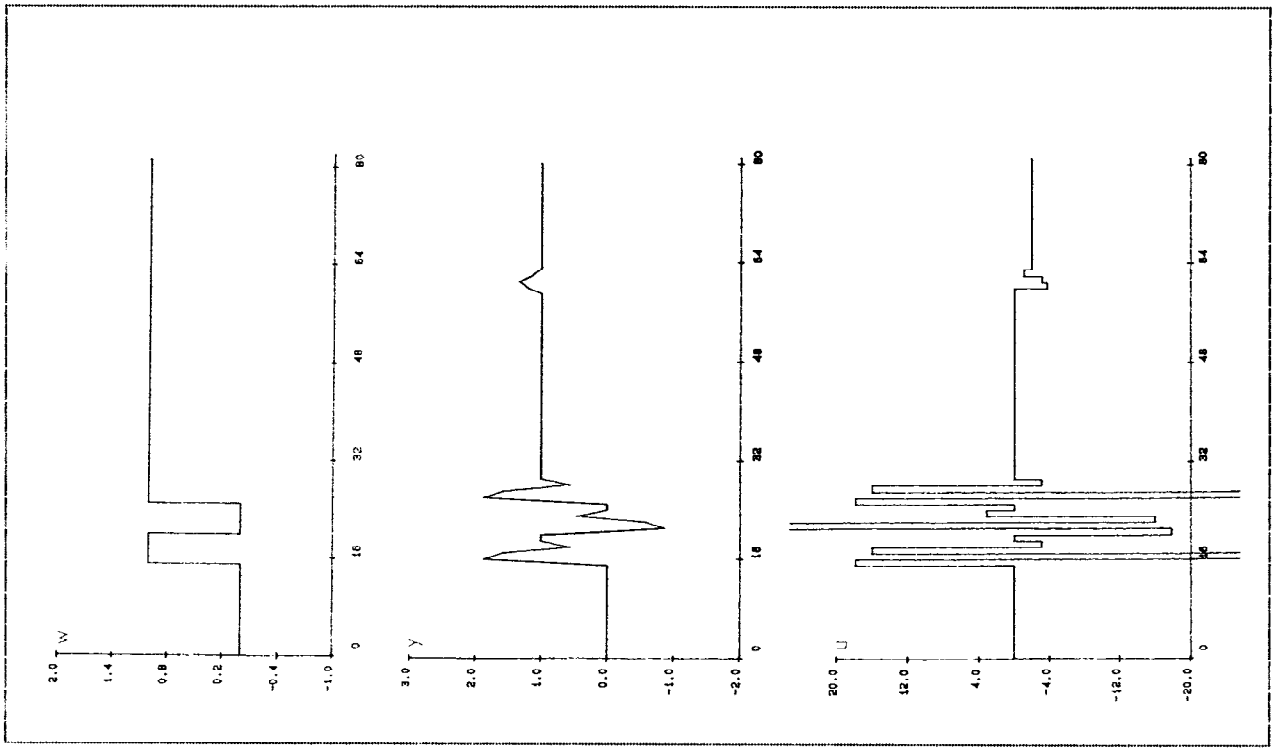


Fig 4.12. Efecto de la inclusión de integrador en APN.

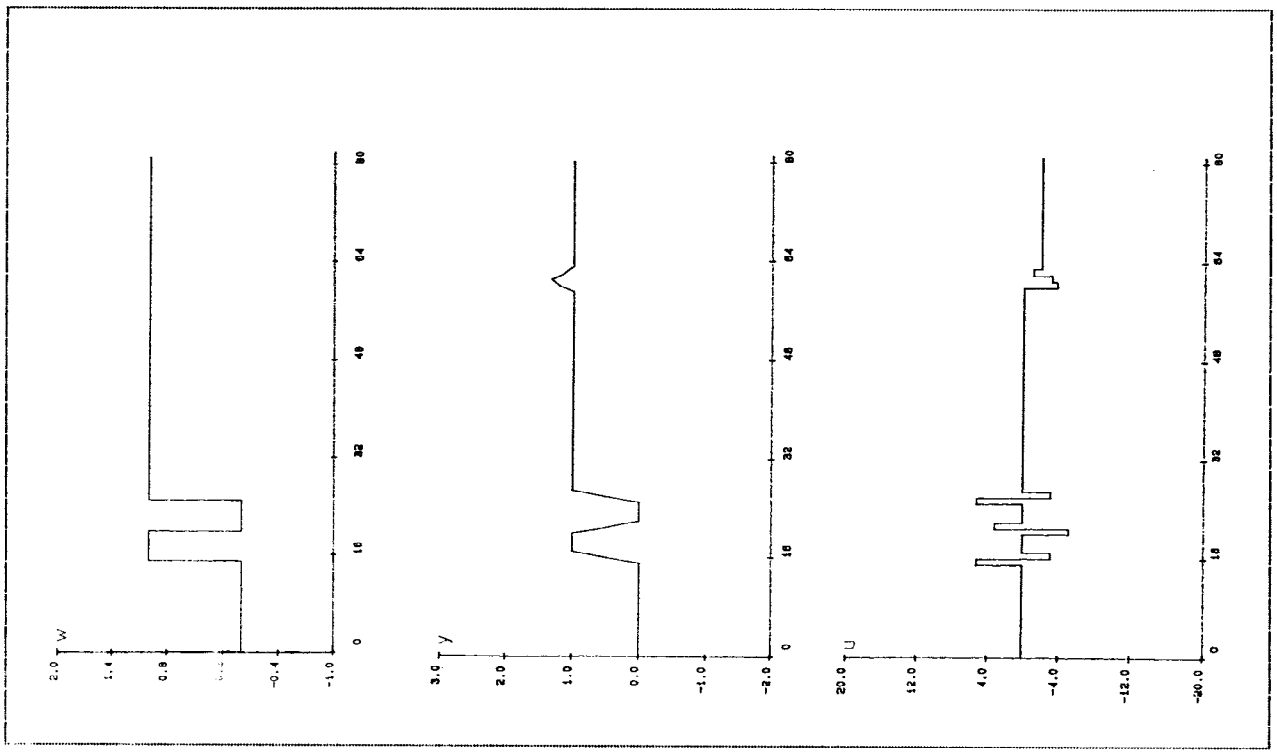


Fig 4.13. Efecto de la inclusión de integrador en APA.

de consigna, como ante la aplicación de perturbación determinista), así como el esfuerzo de control necesario, disminuyen considerablemente cuando w decrece.

Consideremos ahora el caso de la influencia del polinomio observador A_0 en asignación de polos de estructura completa. Mediante una elección adecuada del mismo se podrá introducir una red de compensación que evite la excitación de los modos oscilatorios por parte de la acción de control. Sin embargo, con éste método (Notch Filter Design) no se introduce ningún amortiguamiento de dichos modos oscilatorios con lo que, si fuesen excitados, el sistema respondería del mismo modo que en cadena abierta.

En la figura 4.15 (a) se observan las señales de salida y control del proceso P3.3 con control fijo APA, integrador en el controlador y con la situación de los polos dominantes deseados en cadena cerrada dada por ζ=0.7, w=1. La señal de consigna es la misma que en los casos anteriores y se aplica una perturbación determinista, en escalón, a la entrada del proceso, en k=60.

Se muestra el efecto del diseño de rechazo de banda introduciendo dos raíces en el polinomio observador que responden a la ecuación:

$$z^2 - 1.71\ z + 0.95 = 0 \qquad (4.16)$$

Estas raíces corresponden al par de polos complejos conjugados del proceso, situados en z_1=0.856+0.47j y z_2=0.856-0.47j, o sea con

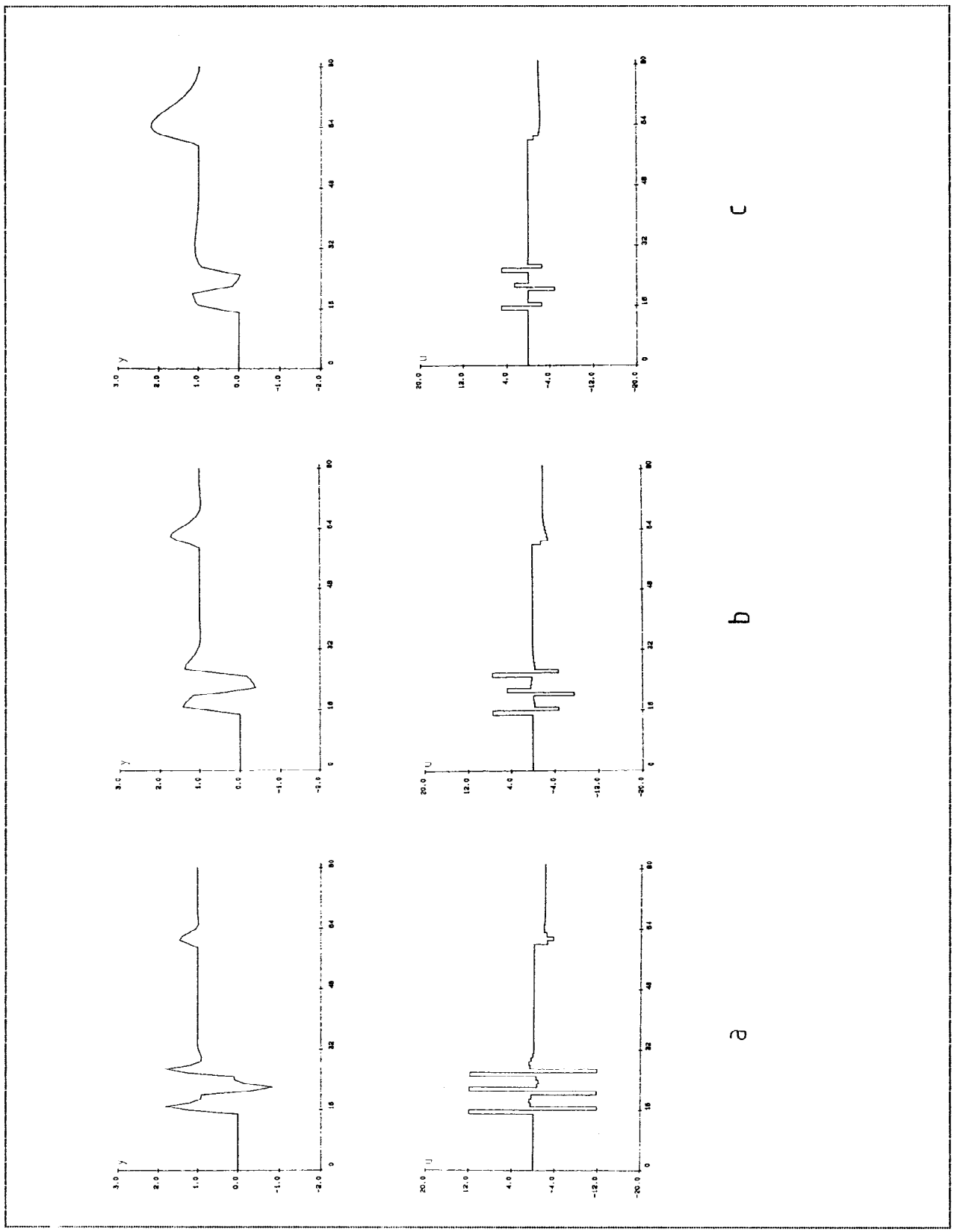


Fig 4.14. Efecto del cambio de w en los polos dominantes.

f=0.05 y w=1 (T_m=0.5).

Como se puede comprobar,la acción de control por cambios en la consigna no excita los modos dados por dichos polos subamortiguados, aunque estos sí son excitados por la perturbación determinista. Nótese que la oscilación no modifica la acción de control; el cambio en ésta se debe únicamente a evitar errores en régimen permanente.

En las figuras 4.15 (b), (c) y (d) se muestra como se puede realizar el diseño de rechazo de banda con amortiguamiento activo de los modos oscilatorios. Se ha realizado control fijo con APA, integrador en el controlador y los polos deseados en cadena cerrada situados en el origen. La señal de consigna y la perturbación determinista son las mismas que en el caso anterior. Se tendrán dos de las raíces del polinomio observador, respectivamente, en:

$z^2 - 1.71\ z + 0.95 = 0$	f=0.05	w=1	T_m=0.5	(b)	(4.17)
$z^2 - 1.53\ z + 0.74 = 0$	f=0.3	w=1	T_m=0.5	(c)	(4.18)
$z^2 - 1.32\ z + 0.5 = 0$	f=0.7	w=1	T_m=0.5	(d)	(4.19)

Como se puede comprobar, el seguimiento de la consigna no se ve afectado, aunque los modos oscilatorios son ahora amortiguados por el controlador.

- <u>Procesos inestables.</u>

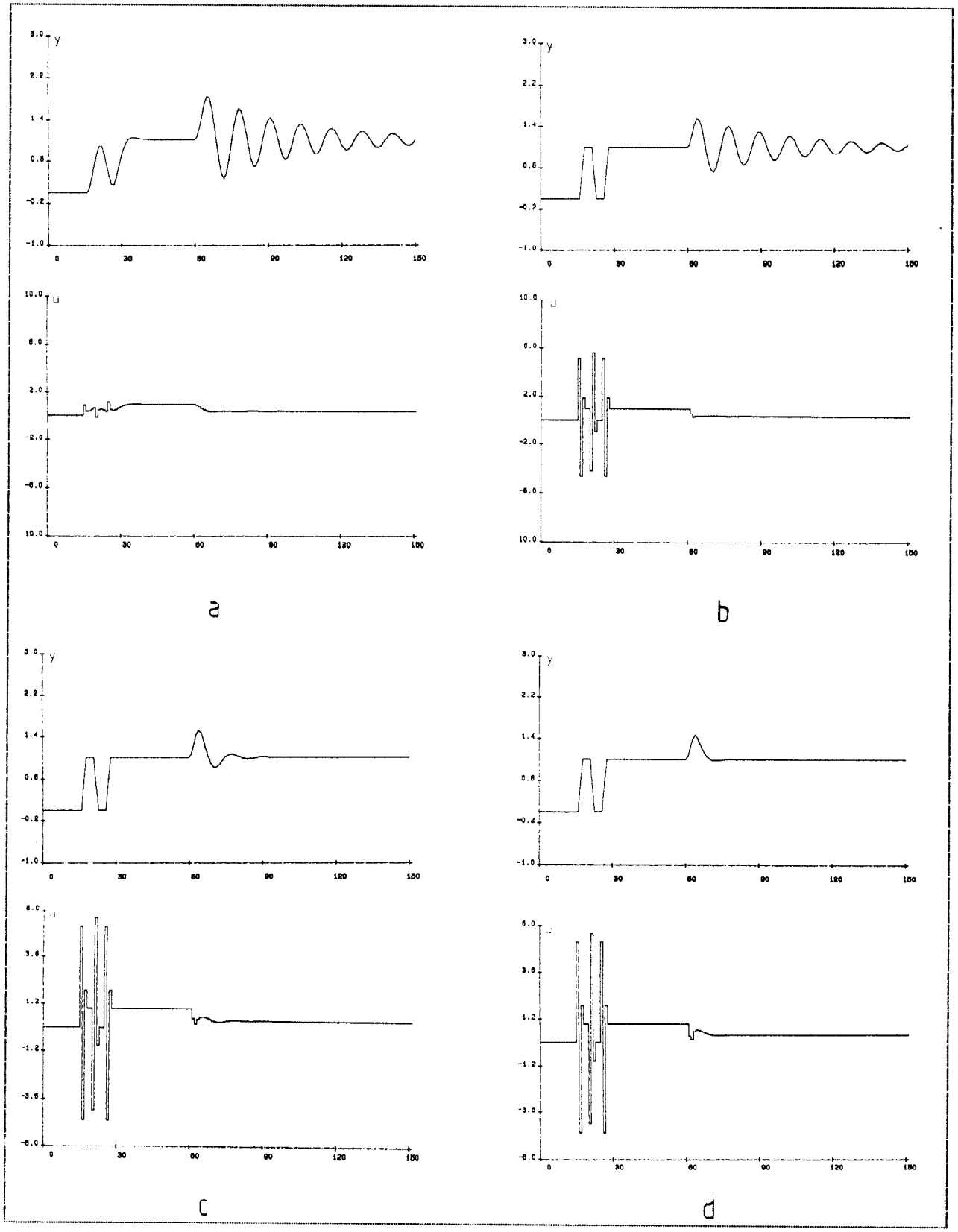


Fig 4.15. Diseño de rechazo de banda con APA.

Estos controladores de asignación de polos son de aplicabilidad muy general, siendo unos de los más adecuados para el control de procesos inestables.

En las figuras 4.16 (a), (b) y (c) se pueden ver las señales de control y de salida del proceso P2.7 controlado con APN (fijo). La señal de consigna cambia a w=0.0 en k=60 respecto a los casos anteriores. Las raíces del polinomio observador se han situado en el origen y no se ha incluido integrador en el controlador.

Se han situado todas las raíces de A_m en el origen, excepto dos de ellas. Estas últimas responden a las ecuaciones:

$z^2 = 0$	comportamiento tiempo finito				(a)	(4.20)
$z^2 - 1.32\ z + 0.5 = 0$	f=0.7	w=1	T_m=0.5		(b)	(4.21)
$z^2 - 1.65\ z + 0.7 = 0$	f=0.7	w=0.5	T_m=0.5		(c)	(4.22)

Como se puede comprobar, al disminuir la rapidez de respuesta exigida, disminuye la sobreoscilación y el esfuerzo de control necesario, aunque siguen siendo elevados.

En la figura 4.17 (a) y (b) se muestra el efecto de la aplicación del controlador APA en los dos últimos casos anteriores (ecuaciones (4.21) (a) y (4.22) (b)). Como se puede comprobar, la sobreoscilación, así como el esfuerzo de control necesario, disminuyen apreciablemente con respecto a la aplicación de APN.

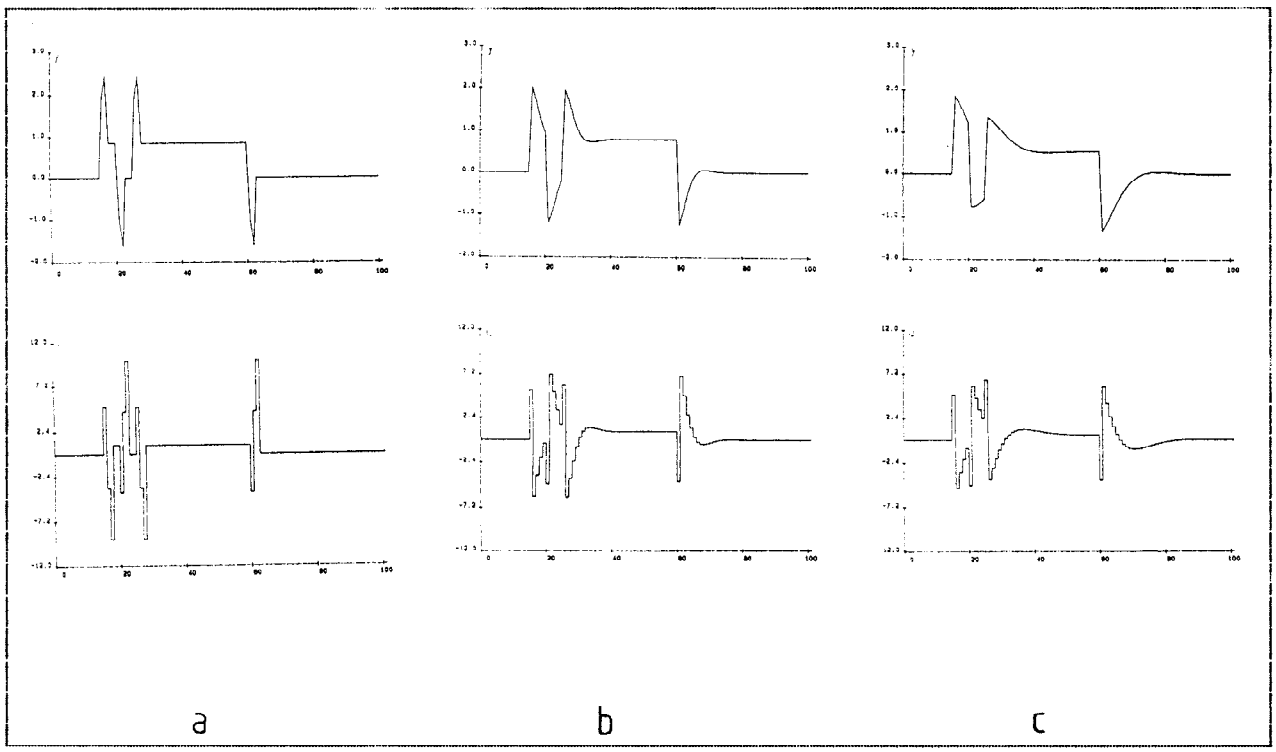


Fig 4.16. Control del proceso inestable P2.7 con APN.

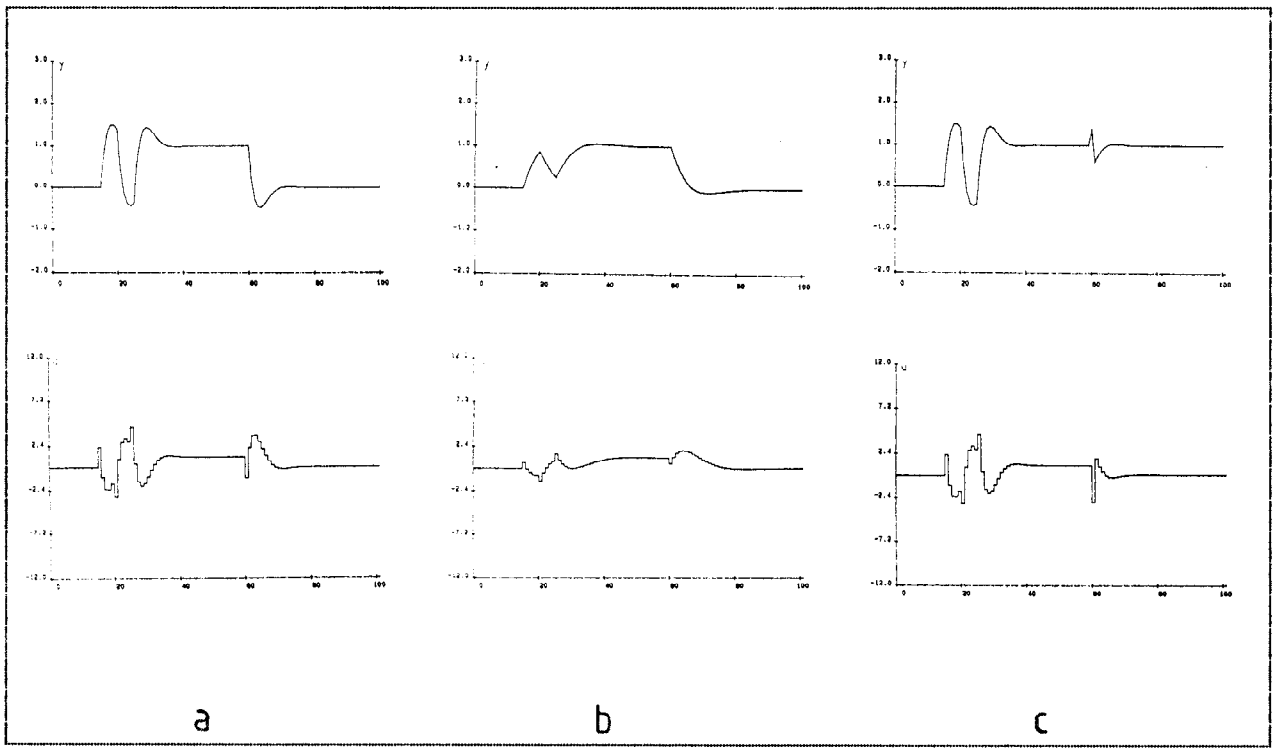


Fig 4.17. Control del proceso inestable P2.7 con APA.

En la figura 4.17 (c) se muestra el efecto de la inclusión de una perturbación determinista a la entrada del proceso, en k=60, en lugar del cambio de consigna, con el mismo polinomio A_m que en el caso (a), aunque incluyendo integrador en el controlador.

- Control adaptativo.

El comportamiento del controlador adaptativo es similar al del fijo correspondiente, una vez que ha transcurrido el número de períodos de muestreo suficiente para que exista una buena estimación de los coeficientes de la función de transferencia del proceso. Esto se puede comprobar aplicando los controladores de asignación de polos a varios de los casos anteriores, con control adaptativo. Los resultados se muestran (variables de control y controlada) en la figura 4.18.

Caso (a):

Proceso: P2.5 Controlador:APA Integrador: SI

Perturbación determinista en k=60.

Raíces de Am en el origen.

Raíces de A_0 en el origen.

Caso (b):

Proceso: P2.5 Controlador:APN Integrador: SI

Perturbación determinista en k=60.

A_m incluye $z^2-1.32z+0.5=0$ (f=0.7 w=1 T_m=0.5)

Raíces de A_0 en el origen.

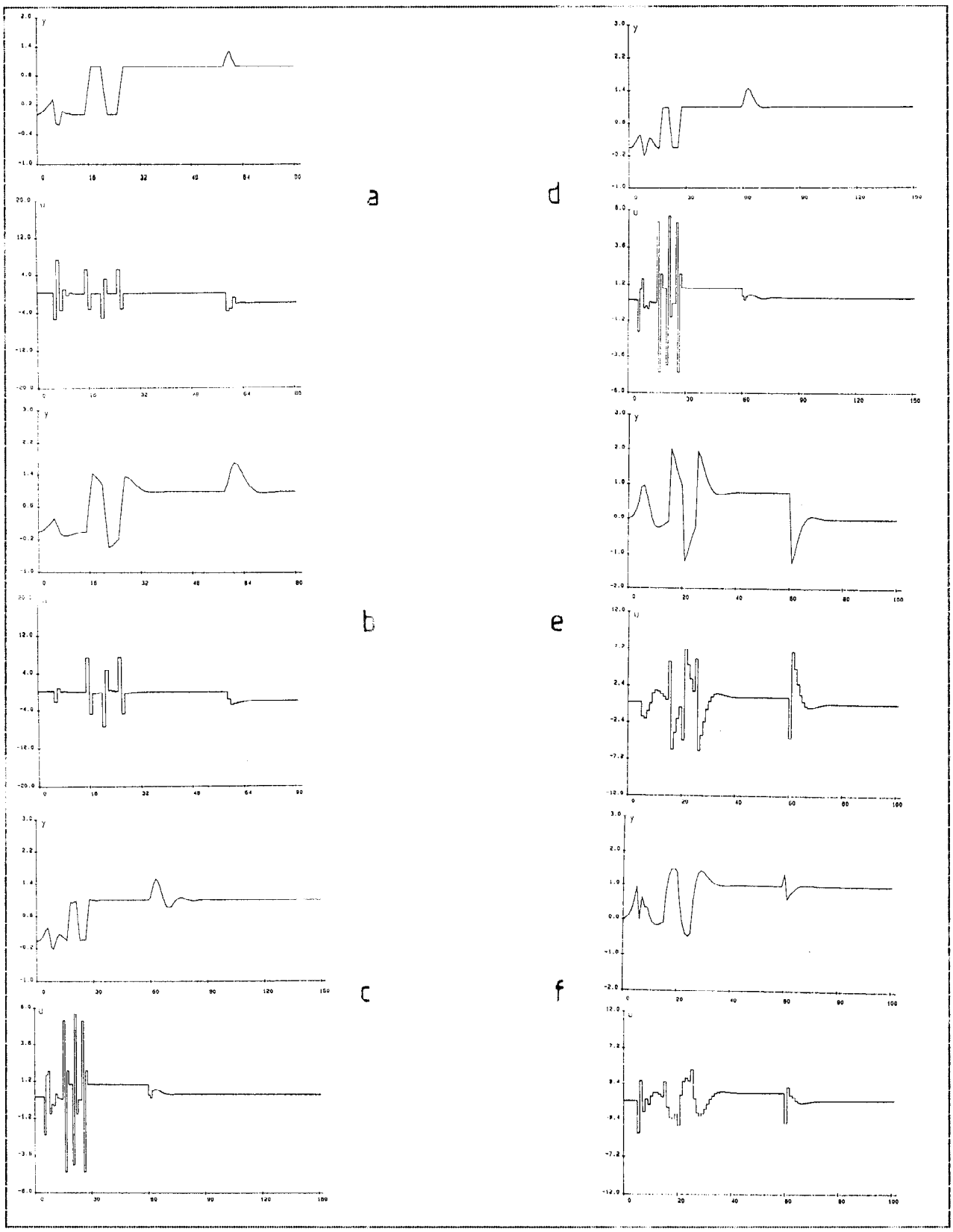


Fig 4.18. Control adaptativo con APN y APA.

Caso (c):

Proceso: P3.3 Controlador:APA Integrador: SI

Perturbación determinista en k=60.

Raíces de Am en el origen.

A_0 incluye $z^2-1.53z+0.74=0$ (f=0.3 w=1 T_m=0.5)

Caso (d):

Proceso: P3.3 Controlador:APA Integrador: SI

Perturbación determinista en k=60.

Raíces de Am en el origen.

A_0 incluye $z^2-1.32z+0.74=0$ (f=0.7 w=1 T_m=0.5)

Caso (e):

Proceso: P2.7 Controlador:APN Integrador: NO

Cambio de consigna en k=60 (w=0.0).

A_m incluye $z^2-1.32z+0.5=0$ (f=0.7 w=1 T_m=0.5)

Raíces de A_0 en el origen.

Caso (f):

Proceso: P2.7 Controlador:APA Integrador: SI

Perturbación determinista en k=60.

A_m incluye $z^2-1.32z+0.5=0$ (f=0.7 w=1 T_m=0.5)

Raíces de A_0 en el origen.

- <u>Control adaptativo en tiempo real.</u>

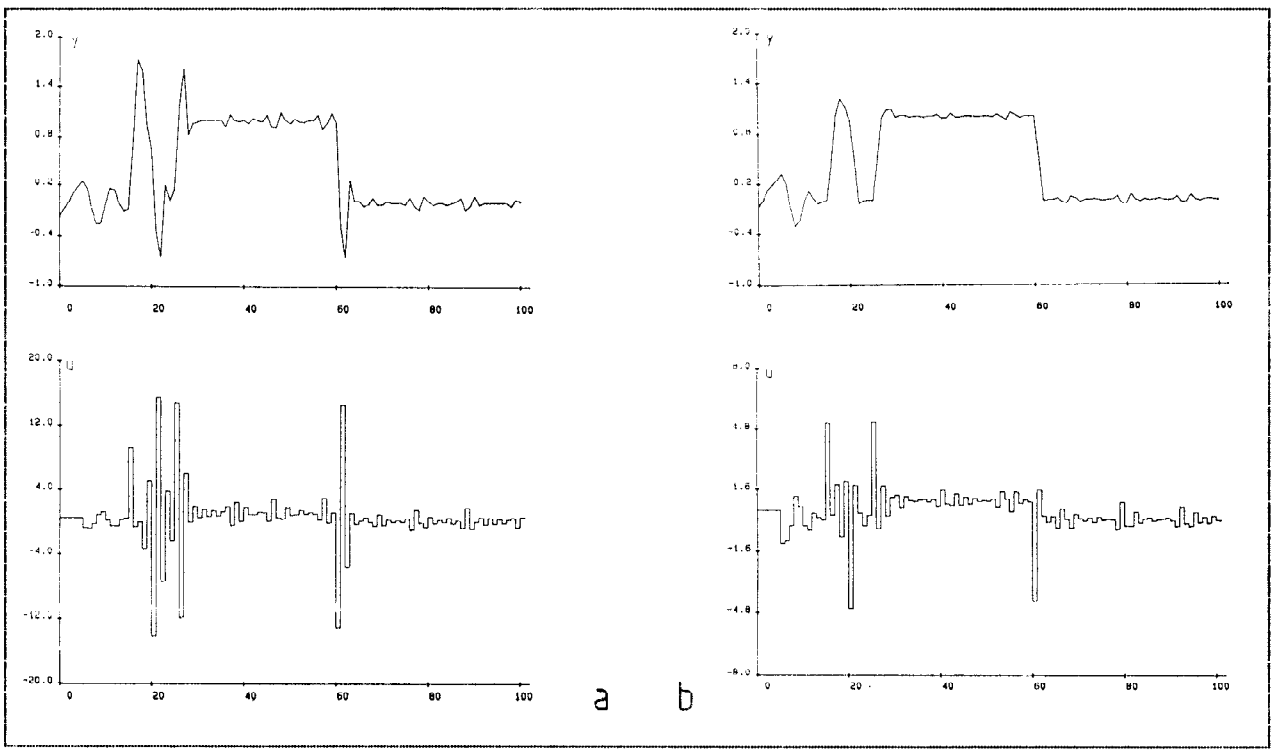


Fig 4.19. Control adaptativo en tiempo real con APN y APA.

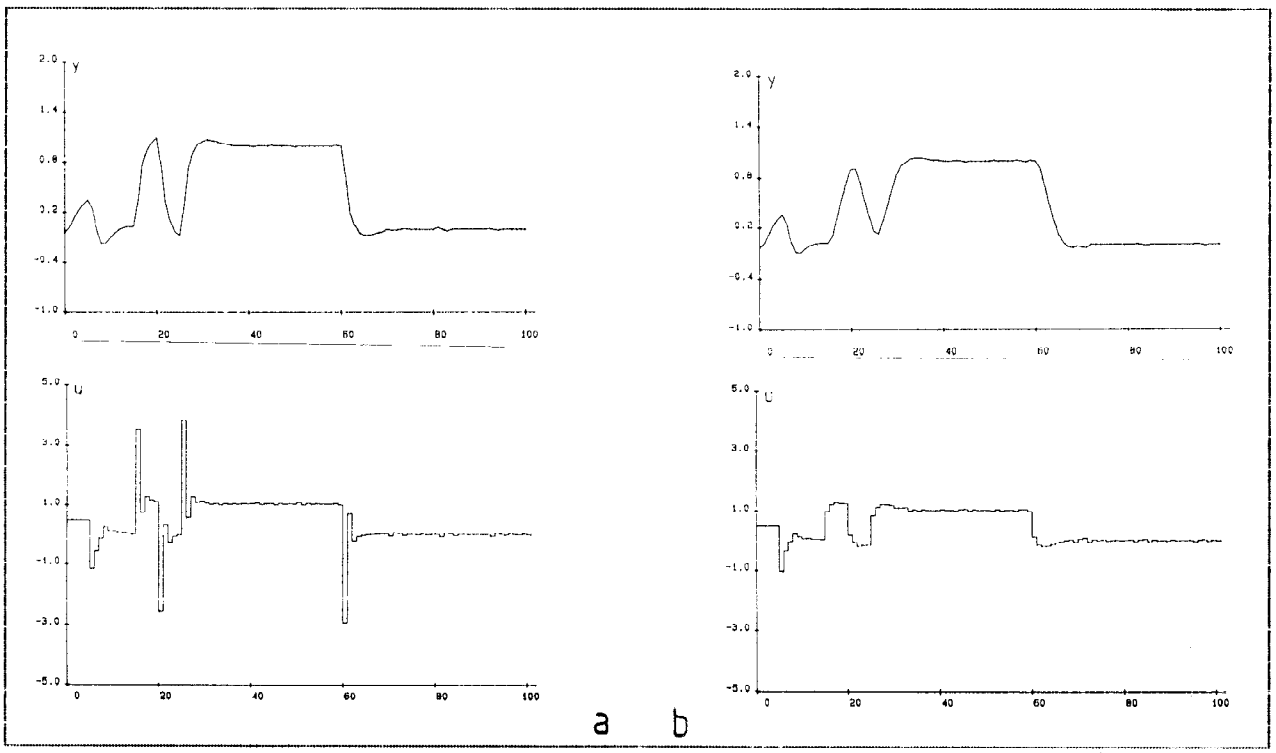


Fig 4.20. Efecto de variaciones de los polos en cadena cerrada.

El comportamiento es similar al observado en control adaptativo mediante simulación. Así se muestra en las figuras 4.19 (a) y (b) las señales de control y de salida del proceso de segundo orden implementado físicamente en la placa de pruebas, controlado con APN y APA respectivamente, incluyendo integrador en el controlador. Las raíces de A_m y A_0 están en el origen y la consigna es la de los casos anteriores.

En las figuras 4.20 (a) y (b) se muestra el efecto de una variación de los polos en cadena cerrada para que incluyan el polinomio:

$$z^2 - 1.32\ z + 0.5 = 0 \qquad (\zeta=0.7 \quad w=1 \quad T_m=0.5) \qquad (4.23)$$

Se aprecia el mismo comportamiento visto en simulación, con una menor sobreoscilación y menor esfuerzo de control que en los casos anteriores.

Así mismo, se ha incorporado al sistema un comando específico para visualización y modificación de la posición de los polos del sistema en cadena cerrada. Así, en las figuras 4.21 (a) para APN y (b) para APA se muestra el efecto de un cambio en A_m en k=60, pasando de tener todos sus polos en el origen, a incluir el polinomio $z^2-1.32z+0.5=0$.

Se puede observar, en la figura 4.22 (a) para APN y (b) para APA, el comportamiento del control adaptativo en tiempo real ante una perturbación determinista, a la entrada del proceso, en escalón,

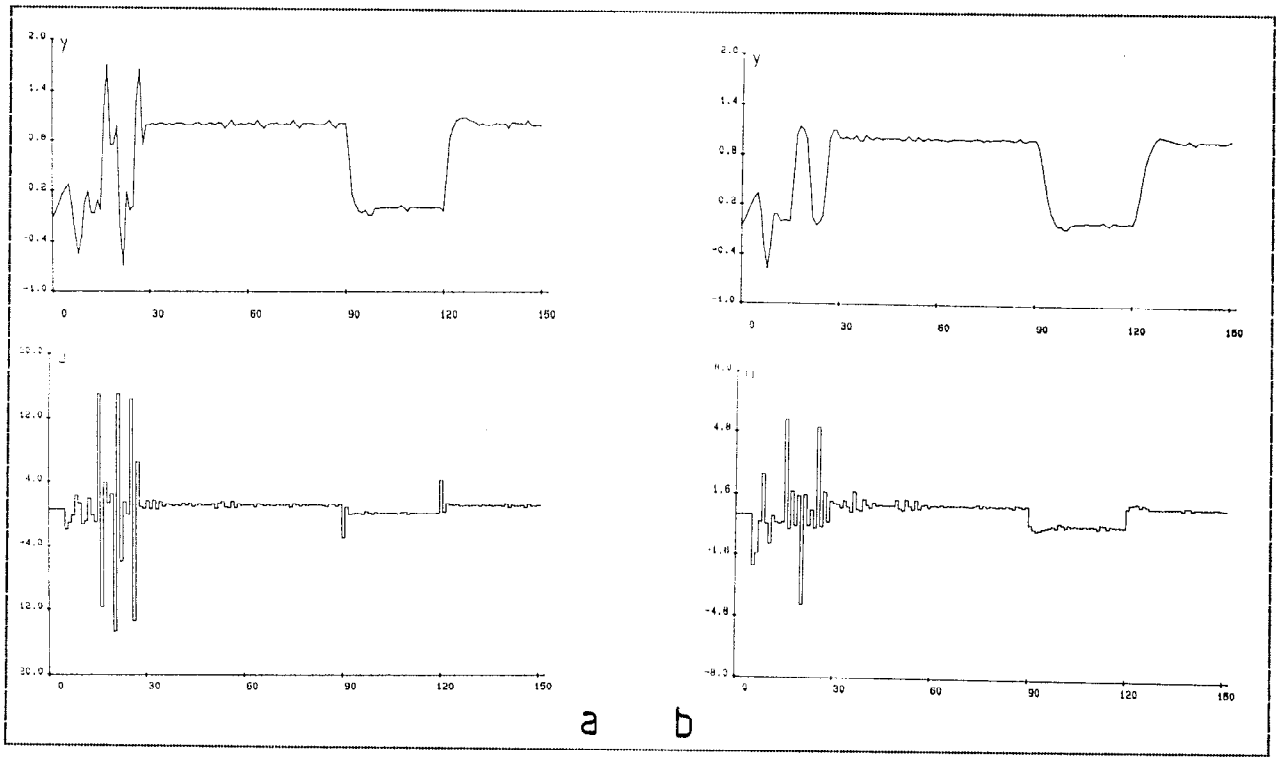


Fig 4.21. Cambio de A_m en tiempo real.

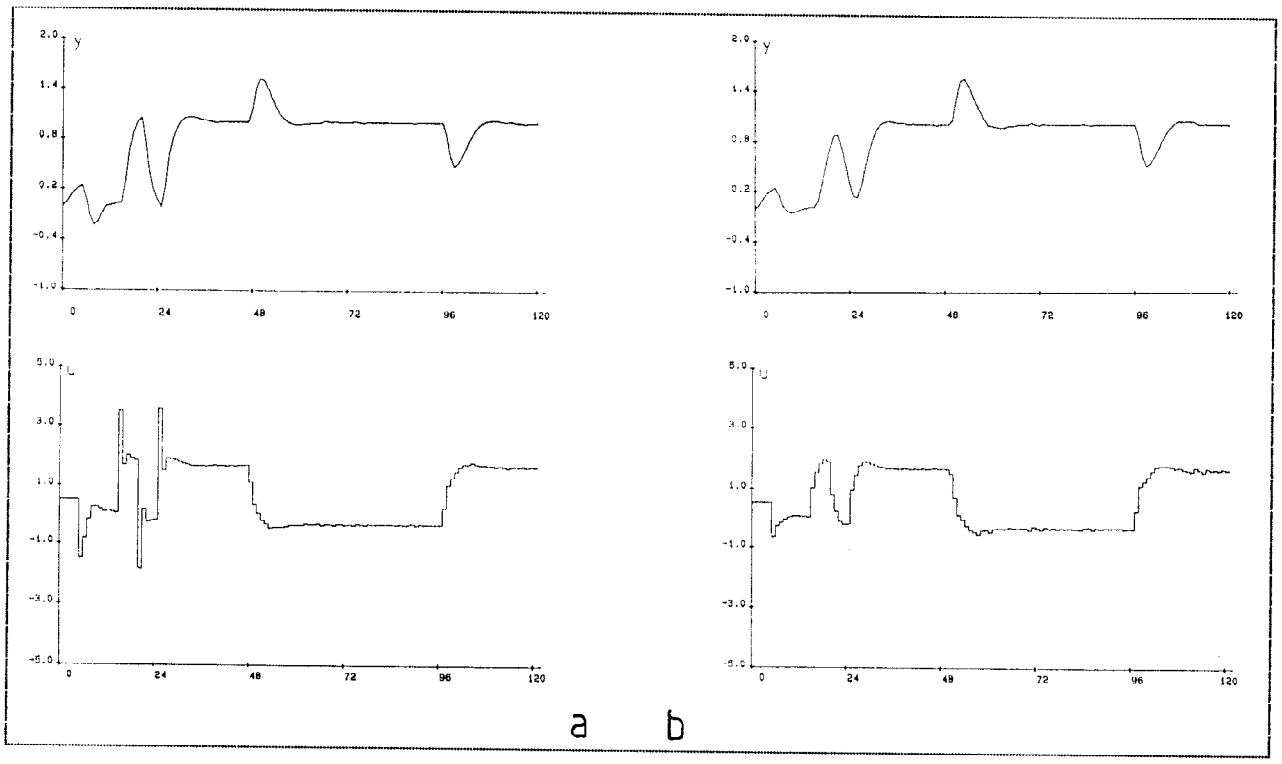


Fig 4.22. Efecto de perturbación determinista en tiempo real.

que afecta al sistema entre k=48 y k=96. Se ha incluido un integrador en el controlador y en A_m el polinomio $z^2-1.32z+0.5=0$. Se observa como, entre k=48 y k=96, la señal de control se adecúa al nuevo valor, necesario para eliminar la variación de la señal de salida debida a la perturbación.

4.2.2.3 Controlador de Asignación de Polos robusto (APR).

- Efecto de los parámetros de ajuste.

Los parámetros del ajuste del método de asignación de polos propuesto por Soh, Evans y otros (1987) son la situación de los polos en cadena cerrada, indicada originalmente por el vector de posiciones nominal d^0 más el vector de variaciones posibles ε, así como el vector de variaciones μ de los coeficientes de la planta. Utilizando el conjunto de regiones propuesto en el apartado 2.1.5, los únicos parámetros necesarios para especificar la situación de los polos en cadena cerrada serán z y r (comprendidos entre 0 y 1).

Si tomamos por ejemplo la planta P2.6, se puede probar si existe una solución para los parámetros del controlador que garantice que los polos en cadena cerrada estén dentro de la región definida por z=0.4 y r=1, cuando el vector de variaciones de los coeficientes de la planta es el siguiente:

$$\mu = [\ 0 \quad 0 \quad 0.035 \quad 0\ ] \tag{4.24}$$

o sea el único coeficiente para el que se supondrá variación será b_1, cumpliéndose $0.965 \leq b_1 \leq 1.035$.

En estas condiciones el controlador robusto estará dado por:

$$G_R(z^{-1}) = \frac{12.8 - 13.73\ z^{-1}}{1 - 9.15\ z^{-1}} \tag{4.25}$$

Si se restringe aún más la posición de los polos en cadena cerrada, especificando la región con igual r y z=0.3:

$$G_R(z^{-1}) = \frac{13.6 - 14.59\ z^{-1}}{1 - 9.72\ z^{-1}} \tag{4.26}$$

Si se baja z a 0.2, el problema de minimización lineal será no factible y no existirá ningún controlador que garantice que los polos en cadena cerrada estén dentro de la región prescrita, para cualquier variación de b_1 dentro de los límites indicados.

Evidentemente, para la misma región, si las variaciones máximas en los coeficientes de la planta son menores, puede llegarse a una solución. Así, para z=0.2 si ahora:

$$\mu = [\ 0 \quad 0 \quad 0.025 \quad 0\ ] \quad \text{o sea} \quad 0.975 \leq b_1 \leq 1.025 \tag{4.27}$$

se tendrá un controlador robusto dado por:

$$G_R(z^{-1}) = \frac{13.13 - 13.01\ z^{-1}}{1 - 9.2\ z^{-1}} \qquad (4.28)$$

En el caso de mantener z=0.4 y tomar r^2=0.9 con μ dado por la ecuación (4.24), se tendrá:

$$G_R(z^{-1}) = \frac{13.08 - 14.13\ z^{-1}}{1 - 9.42\ z^{-1}} \qquad (4.29)$$

Se puede comprobar que los polos en cadena cerrada están incluidos dentro de la región prescrita en cada caso. Así con $G_R(z)$ dado por la ecuación (4.25) (z=0.4, r=1), se tendrán los polos dominantes en 0.19+j0.49 y 0.19-j0.49.

La comparación entre el funcionamiento de un controlador fijo diseñado mediante asignación de polos de estructura simple (sin integrador, todos los polos en el origen) y el controlador robusto por asignación de polos en la región especifiada por z=0.4 y r=1, para una variación en k=60 del coeficiente b_1 del proceso P2.6, pasando de un valor 1.0 a 1.035, se puede realizar observando las figuras 4.23 para APN y 4.24 para APR, en las que se representan las variables de control, consigna y salida. Los controladores vendrán dados por:

$$\text{APN:}\quad G_R(z^{-1}) = \frac{14 - 15\ z^{-1}}{1 - 10\ z^{-1}} \qquad \text{(figura 4.23)} \qquad (4.30)$$

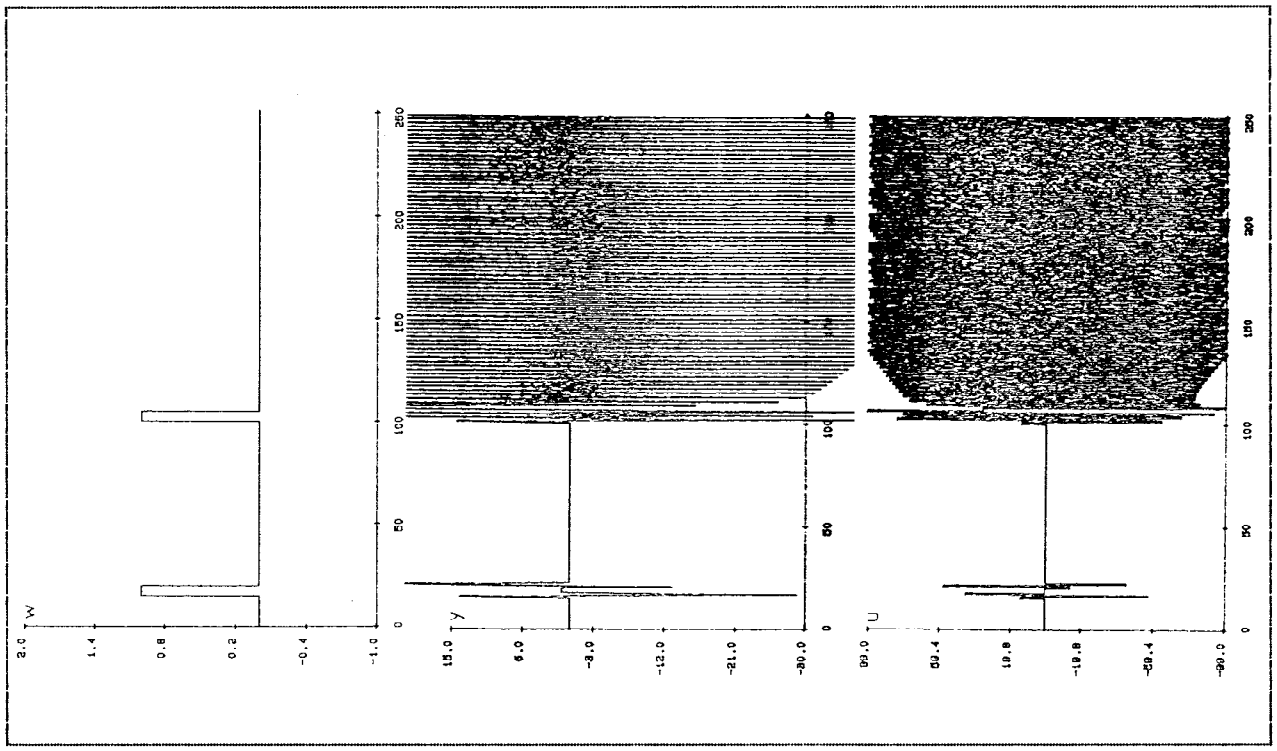


Fig 4.23. Efecto de variaciones en los parámetros de P2.6 con APN.

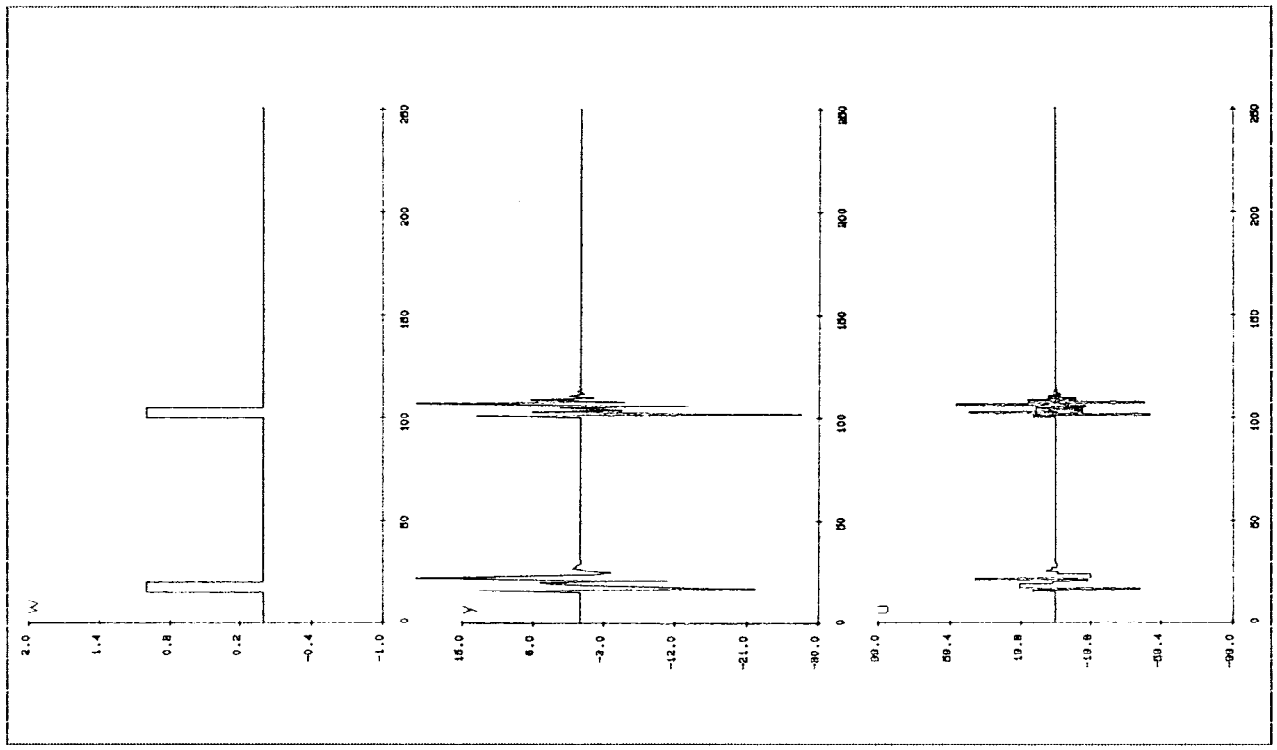


Fig 4.24. Efecto de variaciones en los parámetros de P2.6 con APR.

$$\text{APR:} \quad G_R(z^{-1}) = \frac{12.8 - 13.73\ z^{-1}}{1 - 9.15\ z^{-1}} \qquad \text{(figura 4.24)} \qquad (4.31)$$

En ambas gráficas se observa una elevada sobreoscilación en el seguimiento del primer cambio de consigna, debido a que se le está pidiendo al sistema un comportamiento de tiempo finito cuando se está efectuando el control de una planta "difícil" (polos en z=1 y z=3, cero en z=2).

La diferencia fundamental se observa en el segundo cambio de consigna, después del cambio de valor de b_1, en que el controlador no robusto no logra estabilizar la planta.

En la figura 4.25 se observa el comportamiento del sistema cuando, en el caso anterior, se aplica un controlador de asignación de polos de estructura completa, dado por:

$$R(z^{-1}) = 1-10z^{-1} \qquad S(z^{-1}) = 14-15z^{-1} \qquad T(z^{-1}) = -1 \qquad (4.32)$$

Como se puede observar, no logra estabilizar la planta después del cambio en el proceso, aunque la sobreoscilación en el seguimiento de la consigna, previamente al mismo, es menor que en los casos anteriores. Esta observación conduce a la propuesta de transformación del controlador de asignación de polos robusto simple en uno de estructura completa (apartado 2.1.5) cuya aplicación al caso anterior conduce a:

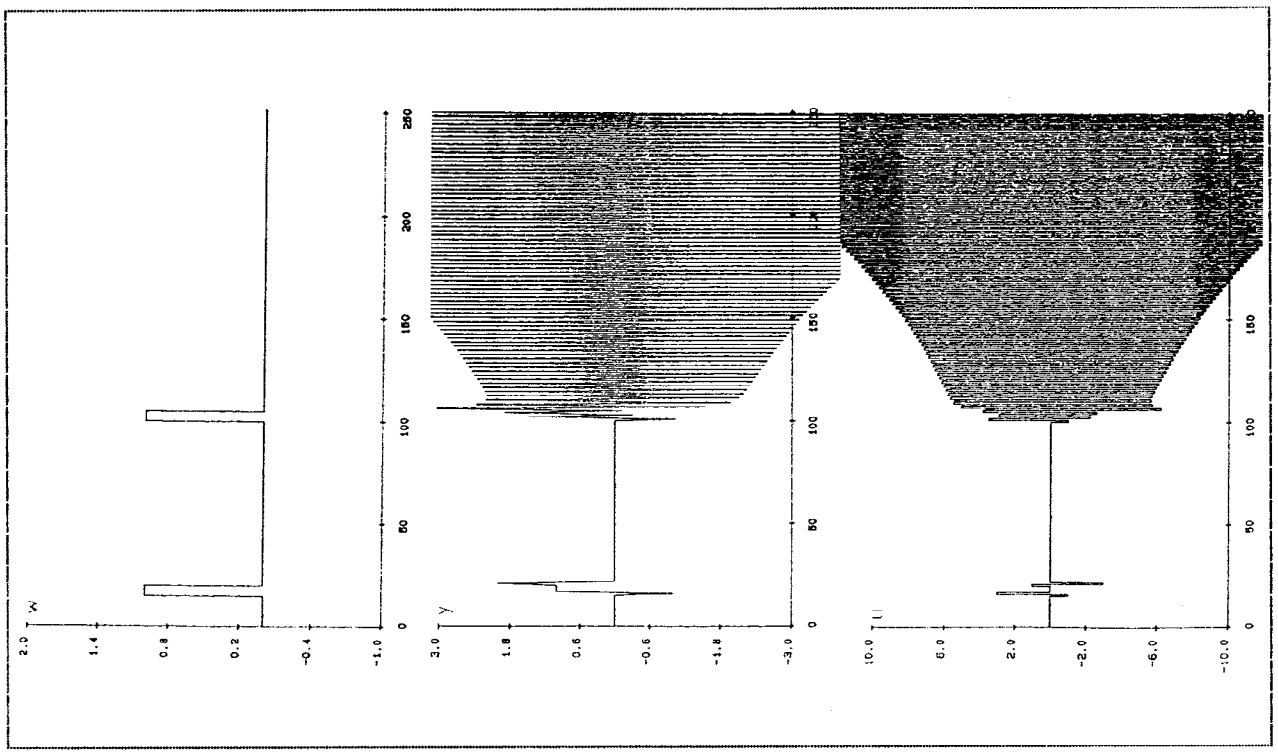


Fig 4.25. Empleo de controlador no robusto de estructura completa.

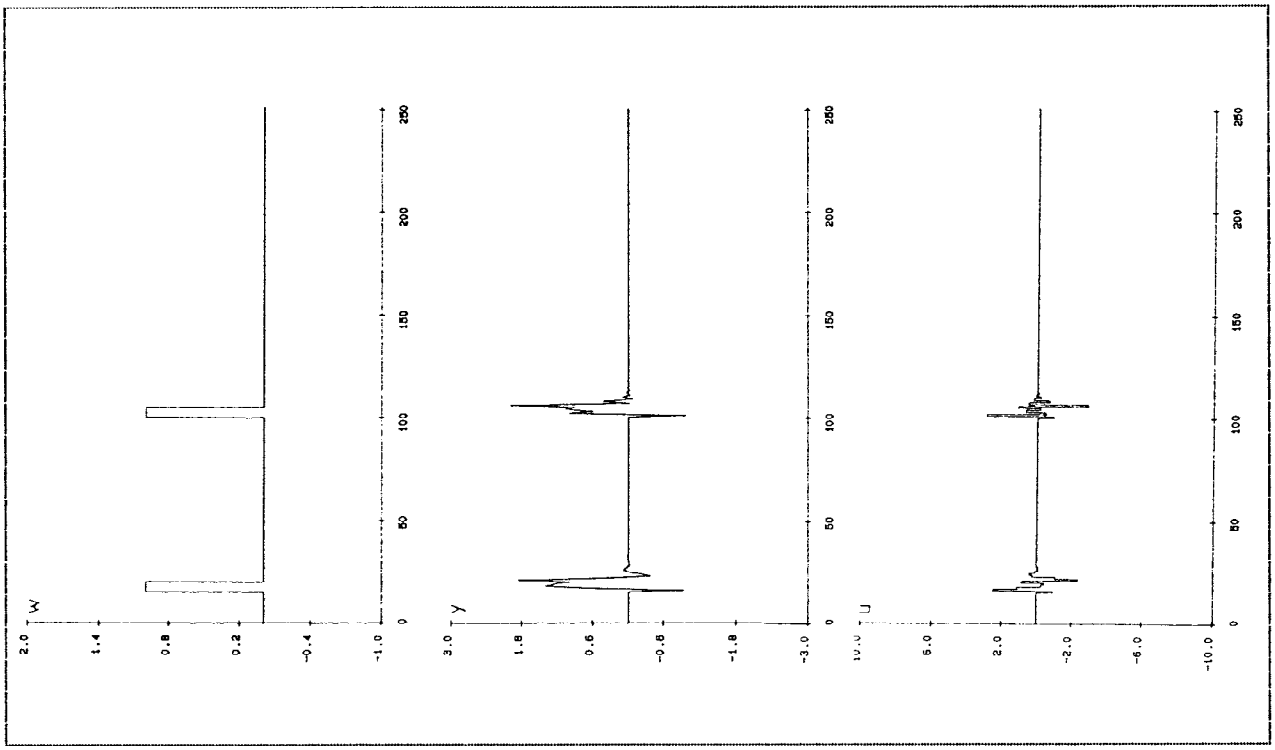


Fig 4.26. Empleo de controlador robusto de estructura completa.

$$R(z^{-1}) = 1-9.15z^{-1} \quad S(z^{-1}) = 12.8-13.73z^{-1} \quad T(z^{-1}) = -0.93 \qquad (4.33)$$

mostrando la figura 4.26 las variables controlada y de control cuando se aplica el mismo.

- Efecto de las perturbaciones deterministas.

Debido a que el controlador robusto original de Soh y Evans no posee efecto integral, no va a poder eliminar el efecto de las perturbaciones deterministas que se presenten, dando éstas lugar a la aparición de un offset en la variable de salida, como se muestra en la figura 4.27 (se aplica el controlador dado por (4.31)). En la figura 4.28 se ve el efecto de la inclusión de una acción integral en la función de transferencia del mismo.

- Empleo del controlador robusto

Aparte de su uso como controlador fijo, el controlador robusto puede ser empleado conjuntamente con un mecanismo de estimación de parámetros robusto, dando lugar a un sistema de control adaptativo robusto. Este uso está bastante limitado, debido al tiempo consumido en resolver el problema de minimización con restricciones presentado. Así mismo hay que tener en cuenta la posibilidad de que, en función de los parámetros de ajuste y los valores actuales de las estimaciones de parámetros del proceso, el problema no posea solución factible en determinados instantes.

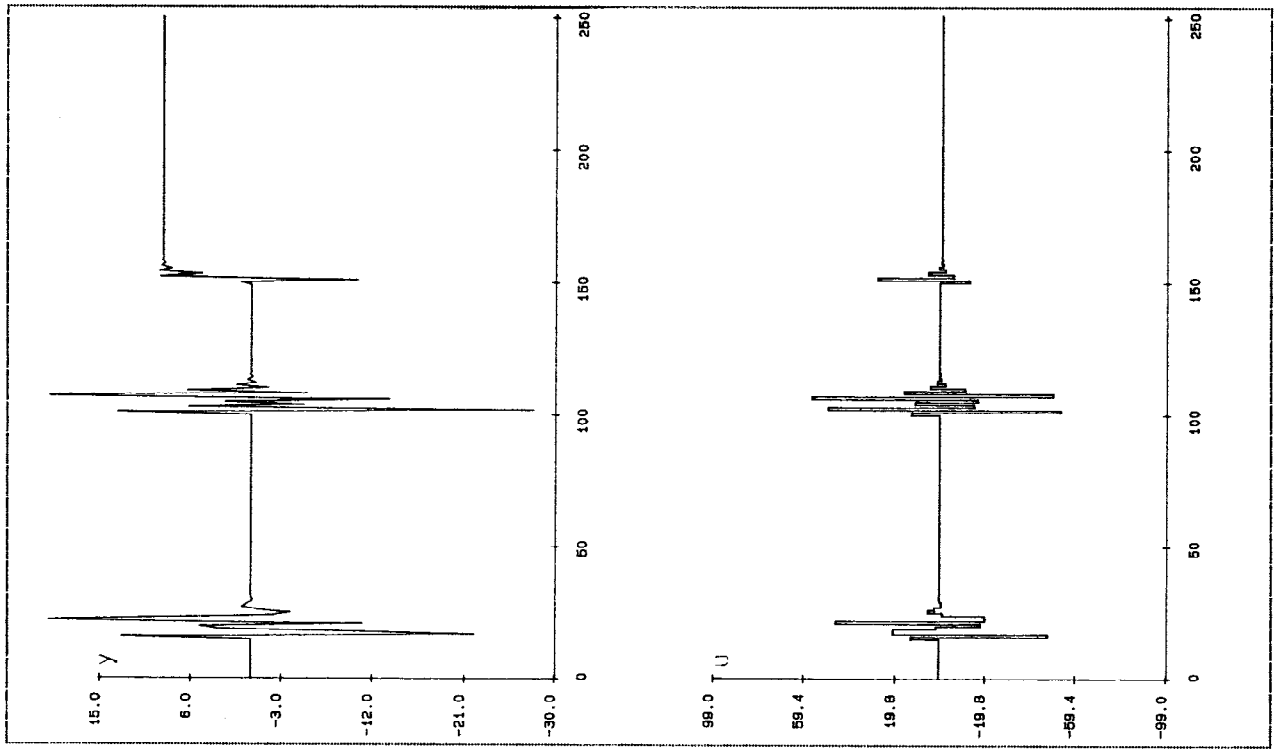


Fig 4.27. Efecto de perturbaciones deterministas con APR.

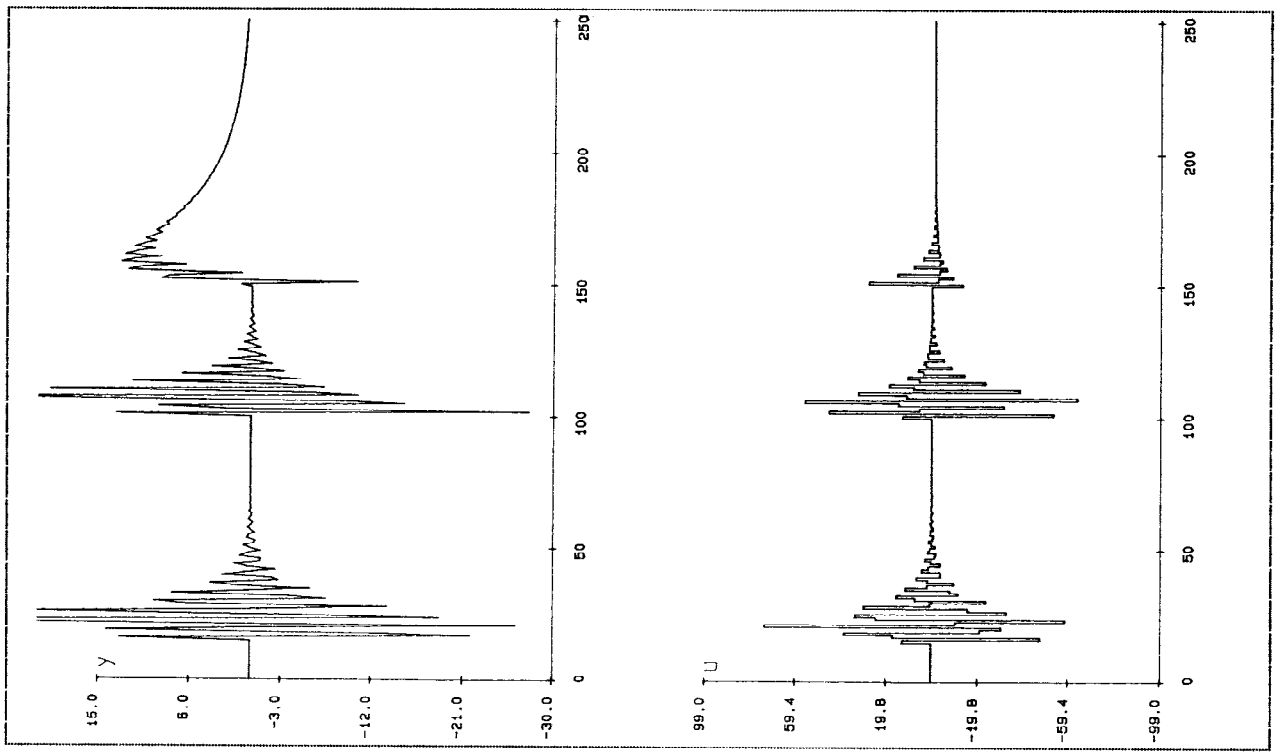


Fig 4.28. Inclusión de integrador en el controlador robusto.

El controlador robusto puede servir así mismo como controlador de respaldo en el sistema de control adaptativo con supervisión diseñado. Este controlador entraría en funcionamiento siempre que el supervisor lo indicara (ante cancelaciones incorretas, saturaciones continuadas, etc.), así como en el arranque del sistema de control adaptativo con procesos inestables.

En la figura 4.29 (a) se muestran las variables de consigna, controlada y de control para el proceso P2.6 con control adaptativo y asignación de polos de estructura completa (con integrador y polos del sistema en cadena cerrada en el origen). El arranque se efectúa con un controlador fijo obtenido por el método de asignación de polos robusto de estructura completa, que es el que actúa en los primeros cambios de consigna (hasta k=20). En este período de muestreo entra el control adaptativo, una vez que se tienen unos valores suficientemente aproximados de los parámetros del proceso. En k=150, b_1 pasa de 1.0 a 1.035.

Evidentemente se pueden conseguir mejores prestaciones situando los polos en cadena cerrada en una posición más adecuada, por ejemplo la dada por un coeficiente de amortiguamiento f=0.9 y w=1. En este caso se obtienen las gráficas de la figura 4.29 (b) donde se observa igual comportamiento durante los primeros 20 períodos de muestreo (controlador robusto), pero una disminución apreciable a partir de ahí de la sobreoscilación, respecto al caso anterior, así como de los valores de la variable de control.

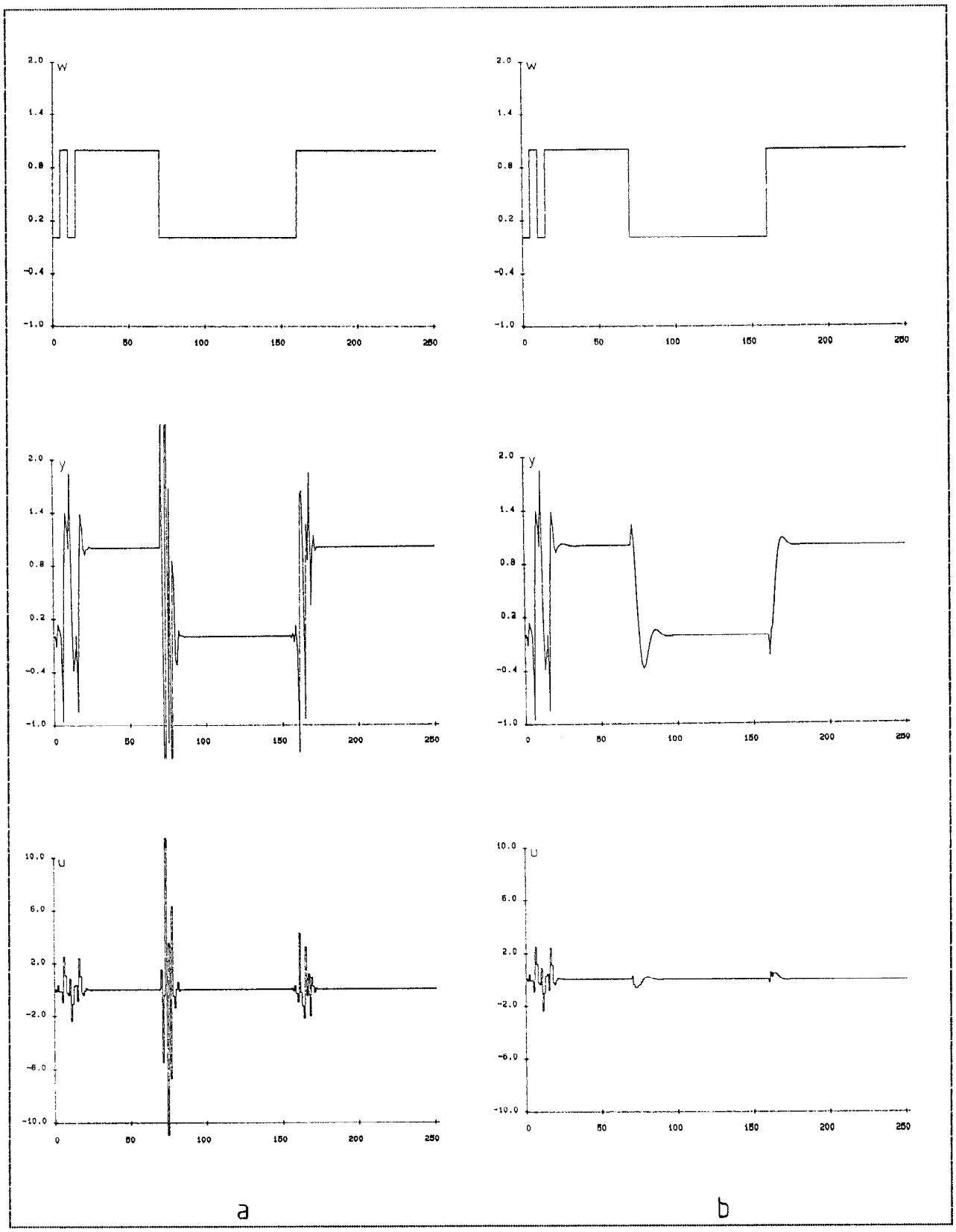


Fig 4.29. Control con APA y arranque con controlador robusto.

4.2.2.4 Controlador Predictivo Extendido (LRP).

El método de control predictivo programado es EPSAC (Extended Prediction Self-Adaptive Control, de De Keyser y Van Cauwenberghe), basado en la realización de una previsión de la salida del proceso a lo largo de un horizonte de l períodos de muestreo.

- Efecto de los párametros de ajuste.

Los parámetros de ajuste tomados para este método de control son dos. El primero es l, el horizonte de previsión, y el segundo β, el cual va a dar lugar al factor de peso de la acción de control según la expresión $\Phi(k)=\beta^{l-k}$.

Ambos parámetros pueden ser modificados off-line mediante menú, aunque funcionando on-line el horizonte de previsión no va a ser considerado como parámetro de ajuste debido a su influencia en la estructura del método de control. Es mucho más simple efectuar el ajuste con β, y, como se verá en los resultados que se presentan a continuación, variando β se puede controlar perfectamente el horizonte de previsión efectivo.

En las figuras 4.30 (a), (b) y (c) se ve un ejemplo de la influencia del parámetro l sobre las variables controlada y de control, tomando los valores l=10 (a), l=6 (b) y l=3 (c). Se aplica el controlador fijo al proceso P2.3 con β=1.0 y la evolución de la consigna es la misma que en la figura 4.31. Como se ve, un mayor

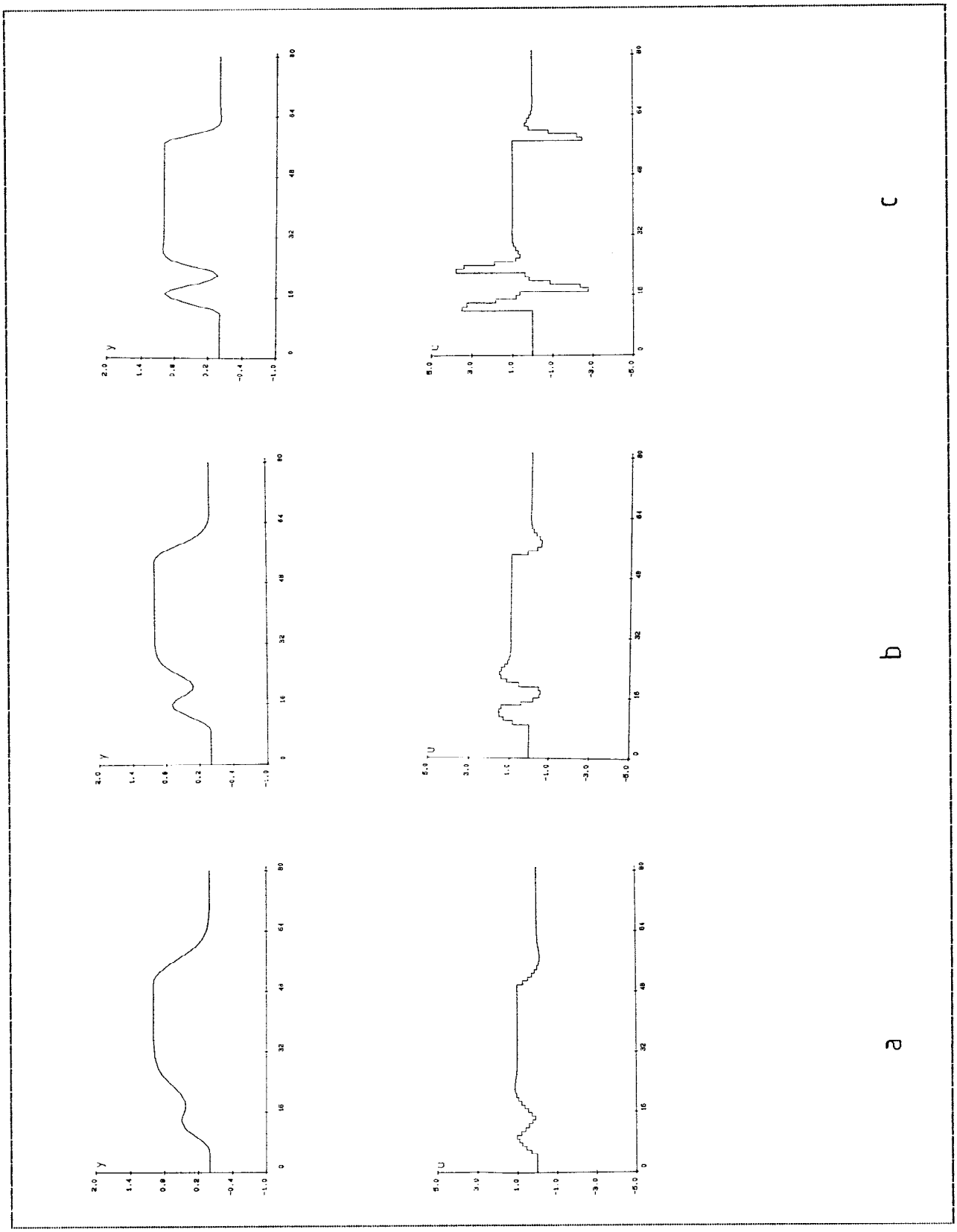


Fig 4.30. Influencia del parámetro l en LRP.

horizonte de previsión ocasiona una acción de control más suave mientras que, si se disminuye l, el esfuerzo de control es mayor con lo que la salida se aproxima más a la consigna.

El mismo efecto que se consigue con la variación de l se puede conseguir con β. En la figura 4.31 (a) se ve el efecto, sobre el mismo proceso y con la misma consigna que en los casos anteriores, al tomar β=0.5 con l=6. En la figura 4.31 (b) se ha tomado β=4.0 con l=6. En el primer caso, el error de control a largo plazo es más importante que a corto, mientras que en el segundo sucede todo lo contrario y la acción de control es más fuerte.

Se hace notar que con éste método de control, como se observa en los casos anteriores, una consigna constante es seguida sin error estático.

- <u>Efecto de las perturbaciones deterministas.</u>

Si el proceso está afectado por una perturbación determinista a la entrada del mismo en forma de escalón, la eliminación de la misma dependerá de si el método de predicción la tiene en cuenta para realizar la previsión de la salida del proceso o no.

En caso de no ser tenida en cuenta se producirá un error en régimen permanente en la salida del sistema. Así se muestra en la figura 4.32 (a) (control fijo, proceso P2.3, l=6, β=4.0) el efecto de una perturbación determinista en escalón, en k=60.

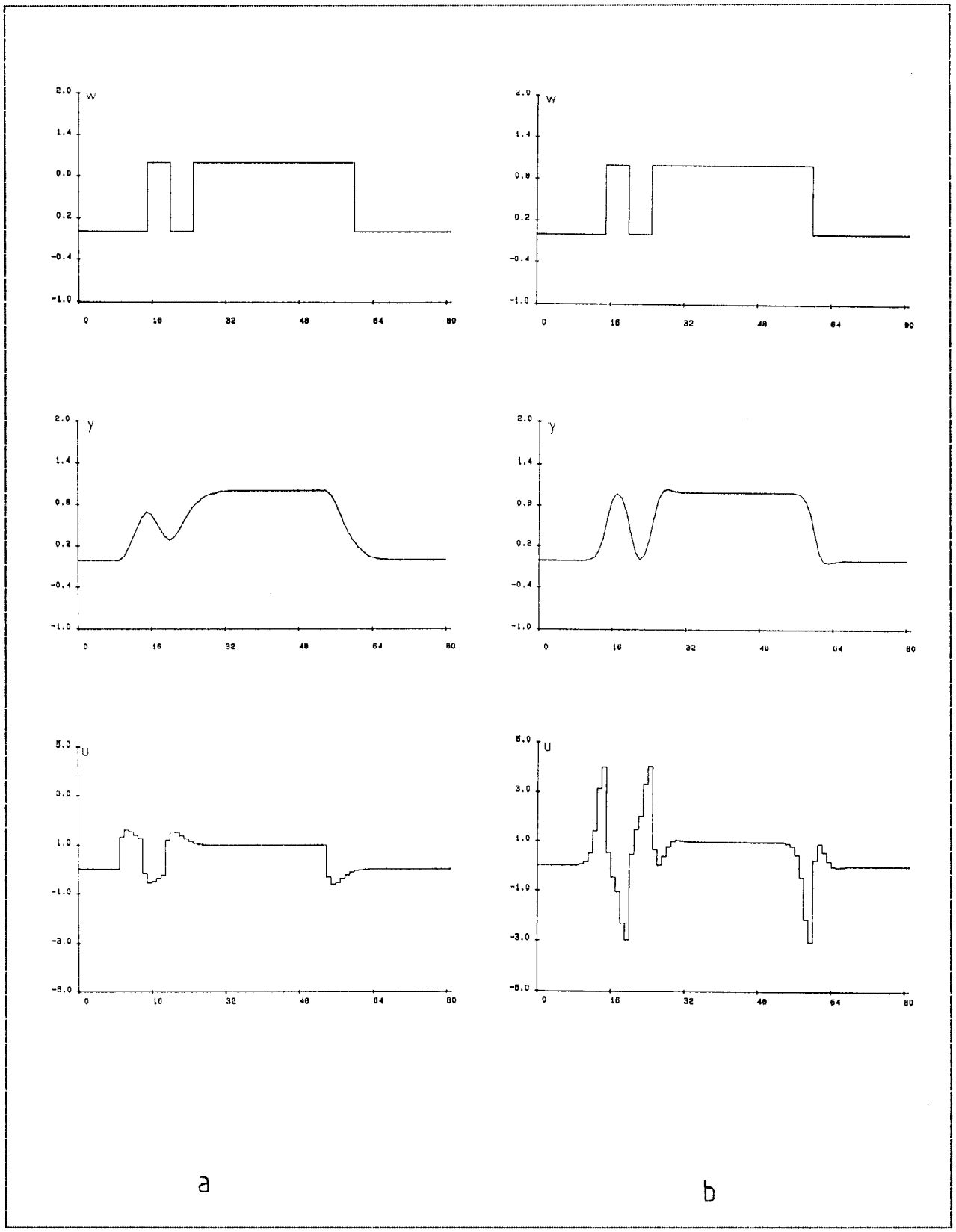


Fig 4.31. Influencia del parámetro β en LRP.

Si en cambio la perturbación es modelada, el predictor puede tomar en cuenta su efecto sobre la salida del proceso y ser compensada por el médodo de control, como se ve en la figura 4.32 (b).

Si l o β varían, se influirá sobre la rapidez de eliminación de la perturbación, pero nunca se producirá error estático. Así en la figura 4.32 (c) se muestra como, variando únicamente β respecto al caso anterior (ahora β=0.5), la variación en la variable de control no es tan brusca cuando se produce la perturbación, por lo que la eliminación de los efectos de la misma no es tan rápida.

- Procesos de fase no-mínima.

Debido a que los ceros en cadena abierta aparecen como ceros en la función de transferencia en cadena cerrada, no van a existir cancelaciones incorrectas al aplicar el sistema de control a procesos de fase no-mínima.

En la figura 4.33 (a) se muestran la variable de control y la de salida del proceso P2.4 con control fijo. Las variaciones de la consigna son las mismas que en la figura 4.31. Se ha tomado l=6 y β=1.0.

Es importante hacer notar, sin embargo, que si los parámetros de ajuste se eligen de forma que se deba ejercer un control muy rápido (cercano a dead-beat), se presentan problemas con los procesos de

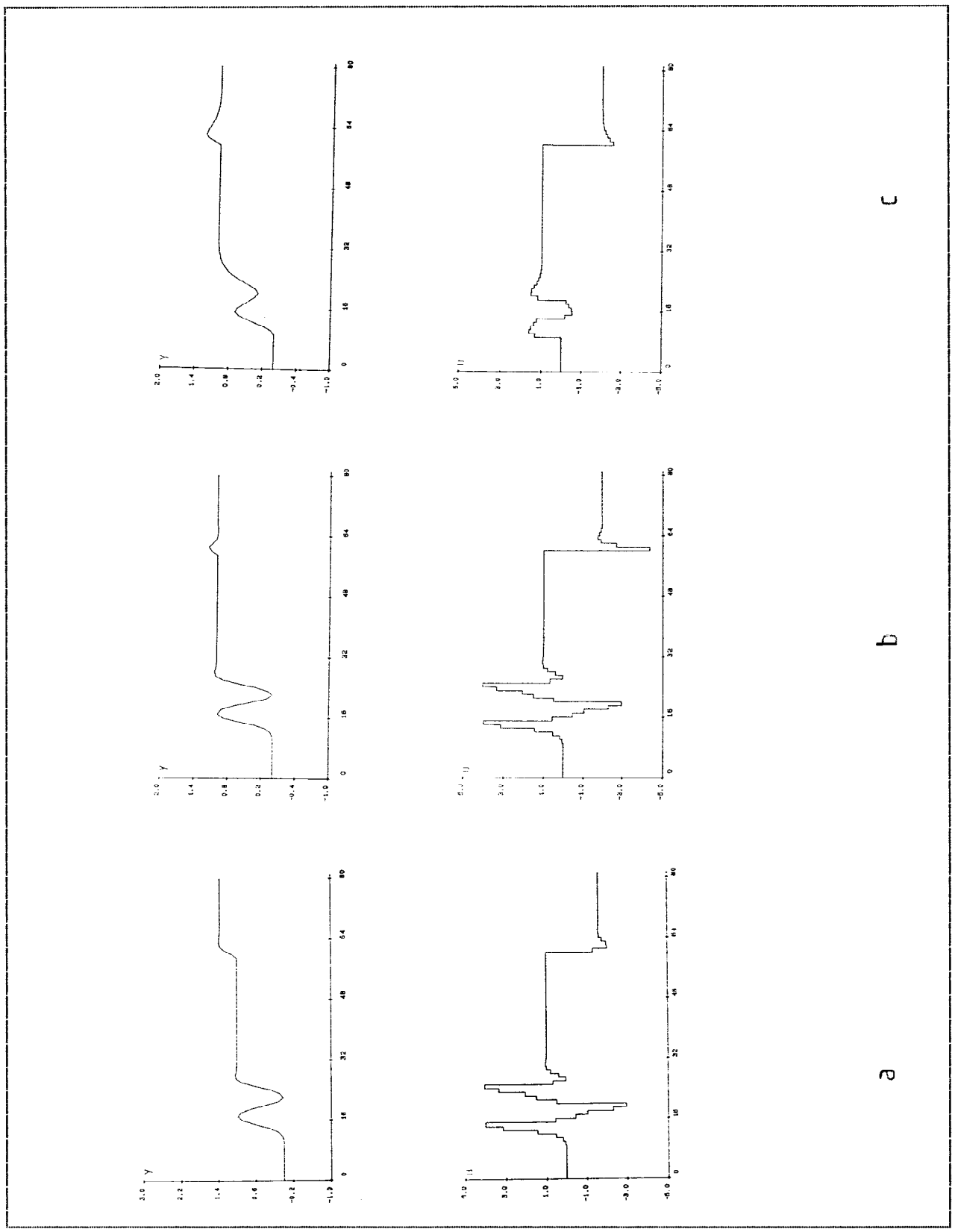


Fig 4.32. Efecto de perturbaciones deterministas en LRP.

fase no-mínima. En las figuras 4.33 (b) y 4.33 (c) se muestran los efectos de disminuir el horizonte de previsión ($l=4$, $\beta=1.0$) o de incrementar el factor de peso de acción de control ($l=6$, $\beta=1.7$) respectivamente, respeto a los aplicados en el caso anterior.

- Procesos con tiempo muerto.

La acción de control con EPSAC es independiente del retardo del proceso, siempre que el rango de predicción efectivo $l-d$ no varíe. Para comprobar esto, así como ver la diferencia que se produce en el seguimiento de la consigna entre este controlador y uno no predictivo (p. ej. el de tiempo finito de orden normal) se van a aplicar ambos sobre el proceso P3.1 con $d=4$.

En la figura 4.34 (a) se ve la señal de salida y la consigna cuando se aplica DBN con control fijo. En la figura 4.34 (b) se ha aplicado al mismo proceso el controlador fijo predictivo con $l=5$ y $\beta=1.0$. Como se ve, al ser el rango de predicción mayor que el tiempo muerto del proceso, y estar la consigna preprogramada, la salida sigue las variaciones de la consigna mucho más rápidamente que en el primer caso.

En la figura 4.34 (c) se observa la aplicación de EPSAC sobre el proceso P3.2, que es idéntico a P3.1 pero con $d=1$. Tomando ahora $l=2$ y $\beta=1.0$ se ve que la salida del proceso es idéntica en ambos casos así como la acción de control, aunque ésta evidentemente está adelantada en el primer caso.

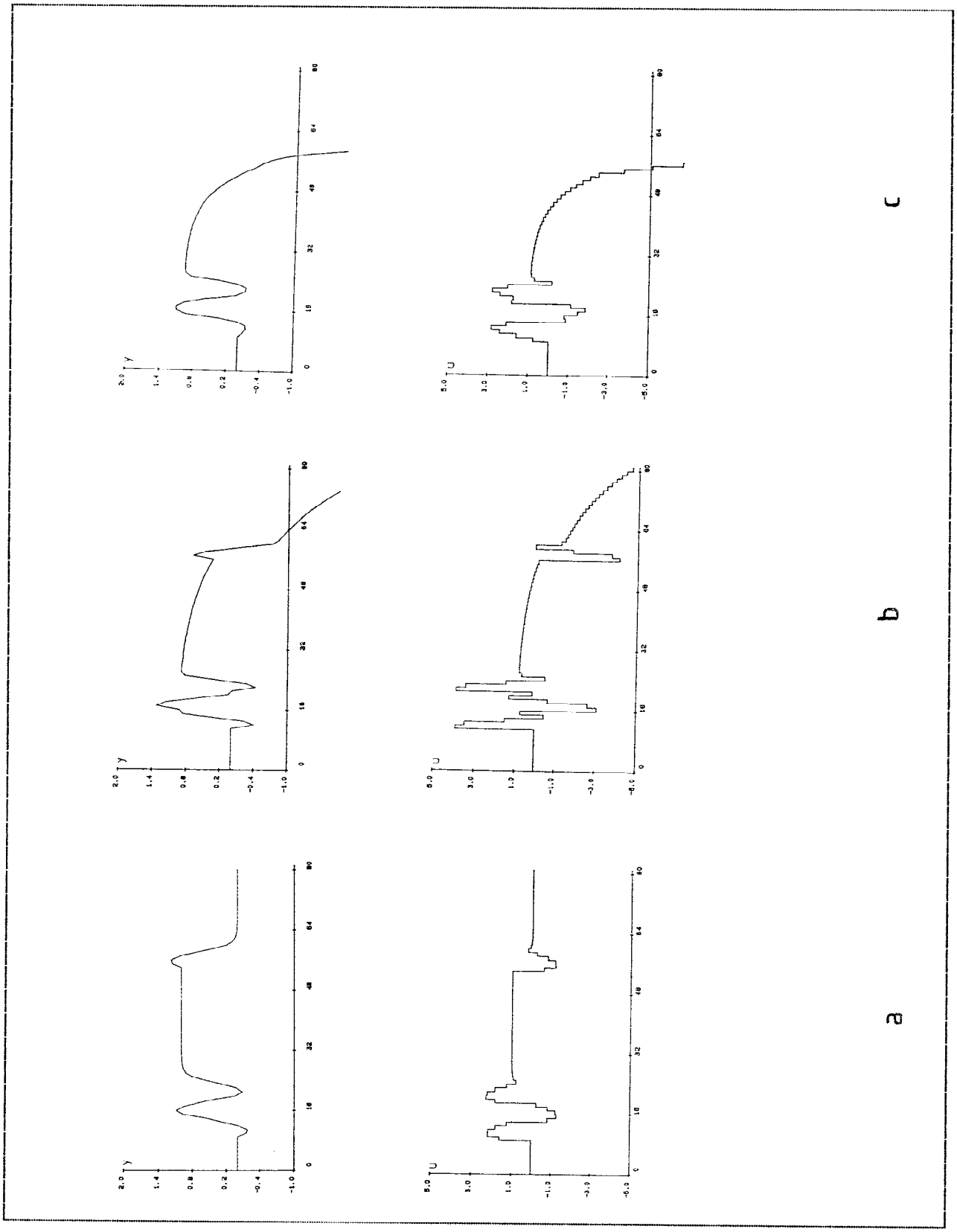


Fig 4.33. Control del proceso de fase no-mínima P2.4 con LRP.

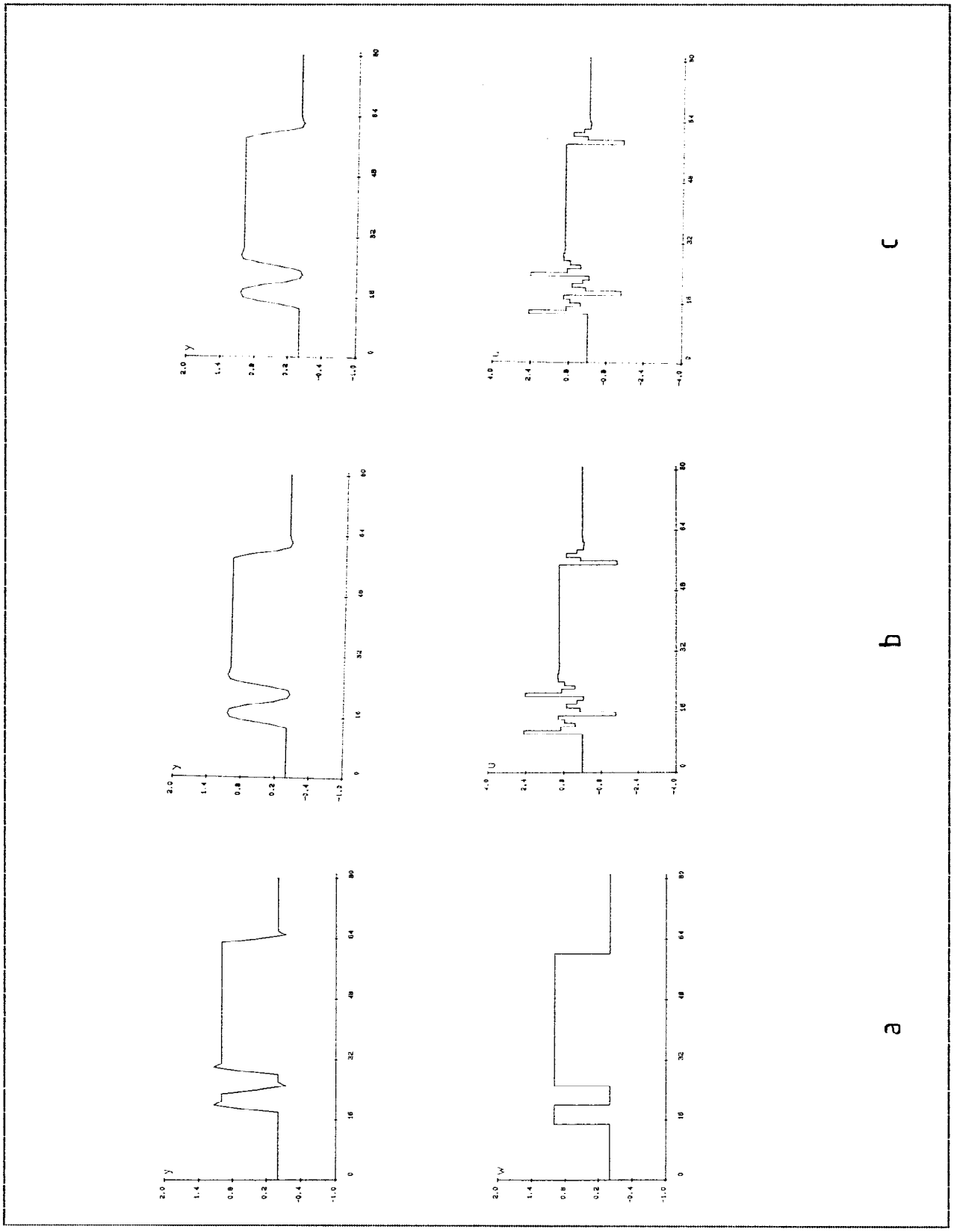


Fig 4.34. Control del proceso P3.1 (d=4) con LRP.

- Control adaptativo.

Una vez transcurridos el número de períodos necesario para realizar la preidentificación, determinando un valor aproximado de los coeficientes de la función de trasferencia del proceso, se observa que el control adaptativo se comporta igual que el fijo. Esto se puede comprobar aplicando los controladores de asignación de polos a varios de los casos anteriores, con control adaptativo. Los resultados se muestran (variables de control y controlada) en la figura 4.35.

Así en el caso (a) se observa el efecto de una perturbación en escalón en k=60 sobre el mismo proceso cuando no es tomada en cuenta por el método de predicción, con l=6 y β=4.0. Como se observa se produce error en régimen permanente. En el caso (b) se muestra la reacción ante la perturbación cuando ésta es modelada, con los mismos parámetros en el método de control.

En los casos (c) y (d) se observan las señales de salida y de control cuando se aplica el método de control sobre un sistema de fase no mínima (P2.4). En el primer caso l=6 y β=1.0 y en el segundo, como se ve, se presentan problemas al bajar el horizonte de previsión a l=4, como ocurría con control fijo.

En los casos (e) y (f) se observa el efecto del control adaptativo sobre los procesos P3.1 (con d=4) y P3.2 (idéntico al anterior con

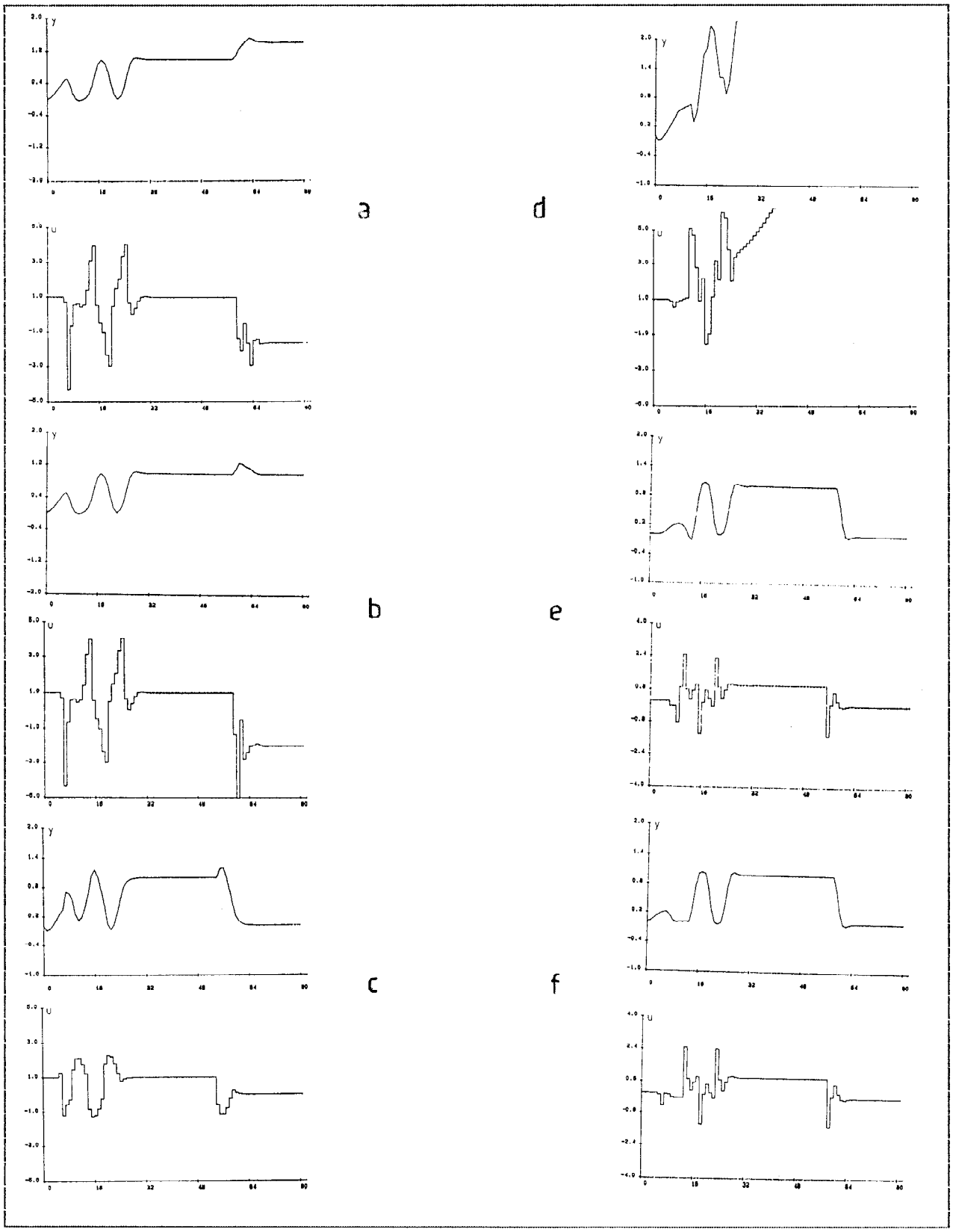


Fig 4.35. Control adaptativo con LRP.

d=1). Como se ve el resultado es el mismo (una vez transcurrida la preidentificación) al tomarse en el primer caso β=1.0 y l=5 y en el segundo β=1.0 y l=2.

- Control adaptativo en tiempo real.

El comportamiento es similar al observado en control adaptativo mediante simulación. Hay una diferencia en cuanto al comportamiento ante cambios de consigna no pre-programadas, introducidas al intérprete de comandos en funcionamiento on-line.

En este caso, para que el método de control predictivo funcione correctamente es necesario que el cambio de consigna no se active inmediatamente, sino un número de períodos después de su introducción, que coincida con el horizonte de previsión. Así, a todo los efectos se entenderá, cuando se trabaje con EPSAC, que un cambio de consigna on-line en $k=k_1$ se producirá en $k=k_1+1$.

En la figura 4.36 (a) se ven las variables de consigna, salida y control cuando se aplica EPSAC al proceso de segundo orden implementado físicamente, con l=4 y β=2.0, siendo los cambios de consigna en k=15, k=20 y k=32 preprogramados y los cambios en k=48 y k=64 efectuados on-line.

Así mismo, se ha incorporado al sistema un comando especifico para controlar el horizonte de previsión efectivo (así como la magnitud de la señal de control), con el parámetro β.

En la figura 4.36 (b) se observa las señales de consigna, salida y control cuando se aplica control adaptativo al sistema anterior con tres cambios de consigna preprogramados (en k=15, k=20 y k=25) y otros dos on-line (en k=60 y k=80). Así mismo l=5 y β=0.3 y se produce un cambio en k=30 del factor de peso, pasando el valor de β a 3.0. Como se observa, la reacción a los cambios de consigna anteriores a k=30 es mucho más lenta que a los posteriores.

4.2.3 Estimación de parámetros del proceso.

Este módulo gestiona la estimación de los parámetros del proceso, en cada período de muestreo, cuando el modo de funcionamiento es control adaptativo o estimación de parámetros. Así mismo, en la fase de preidentificación, es llamado para aportar datos que permitan estimar el orden del proceso.

Para realizar lo anteriormente citado, consta de dos procedimientos. El primero de ellos (PRB) sirve para la generación de PRBS (Pseudo Random Binary Sequence) con características seleccionables mediante menú, necesaria para llevar a cabo la fase de preidentificación, así como la de estimación previa de parámetros en el arranque de un controlador adaptativo.

El segundo procedimiento (RLS), realiza la estimación de parámetros de un proceso de orden y tiempo muerto especificados, por el método de mínimos cuadrados recursivos.

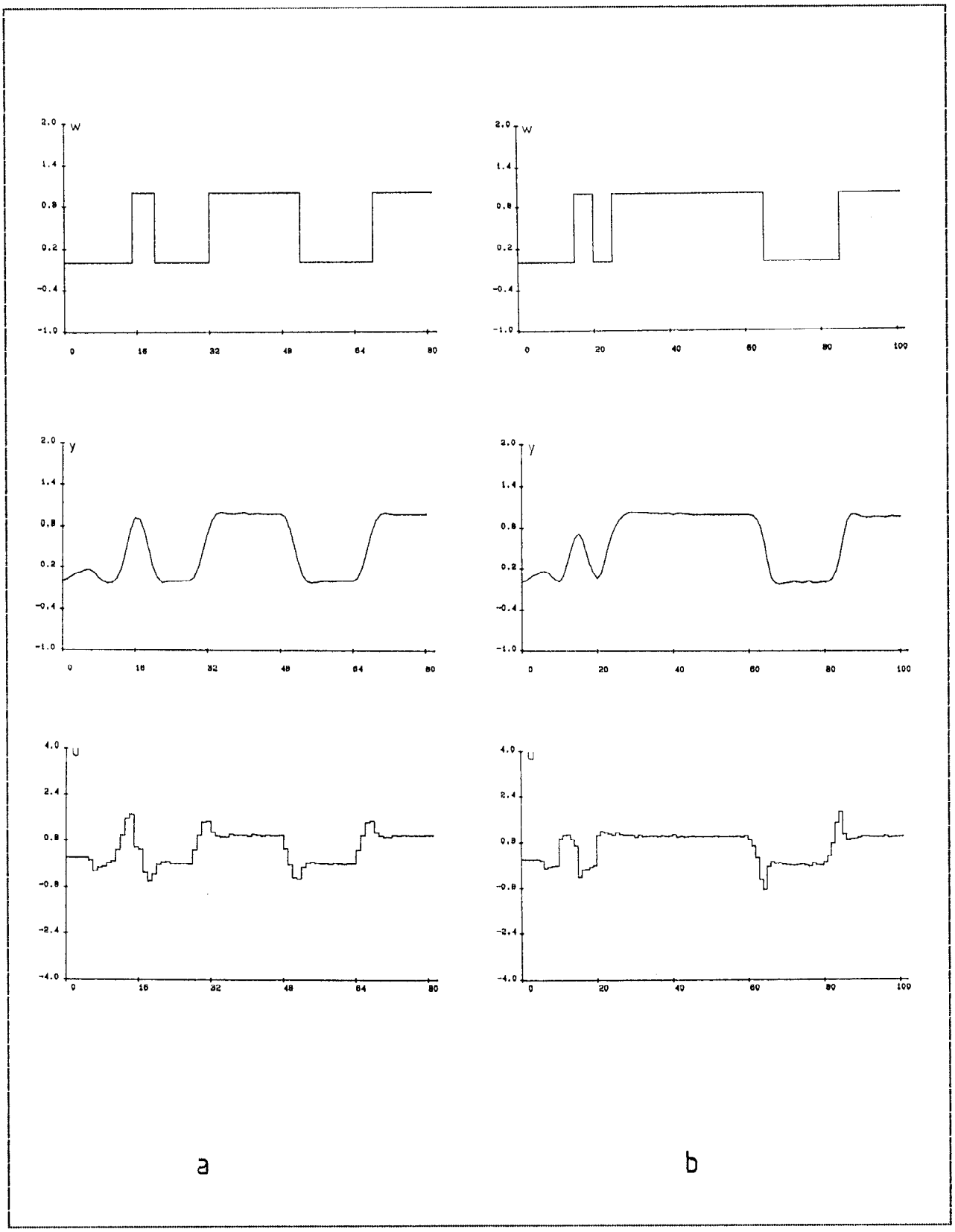


Fig 4.36. Control adaptativo en tiempo real con LRP.

4.2.4 Salida del controlador.

Este módulo se emplea de forma genérica para la obtención de la salida de un sistema discreto, conocida la entrada y la función de transferencia del mismo.

De forma particular se usa, mediante el procedimiento SIM del mismo, cuando se está simulando el comportamiento de un sistema discreto dado por su función de transferencia.

Así mismo, mediante el procedimiento CTR se obtiene el valor de la señal de salida del controlador, con el método seleccionado (excepto LRP), en cada período de muestreo, teniendo en cuenta la posible saturación del actuador. Para esto se ha incorporado un tercer procedimiento (SAT), que tiene en cuenta los valores de U_{max} y U_{min} seleccionados mediante menú, para comprobar dicha saturación.

En las figuras 4.37 (a), (b) y (c) se muestra este efecto, con la representación gráfica de las señales u e y cuando se aplica control adaptativo al proceso de segundo orden implementado en la placa de pruebas (con APN, integrador, polos en cadena cerrada con f=0.5 w=1). En el primer caso U_{min}=-4.0 y U_{max}=4.0 con lo que no se produce la saturación del actuador. En el segundo U_{min}=-1.7 y U_{max}=1.7, así como en el tercero (aunque en éste se selecciona el mecanismo anti-reset windup).

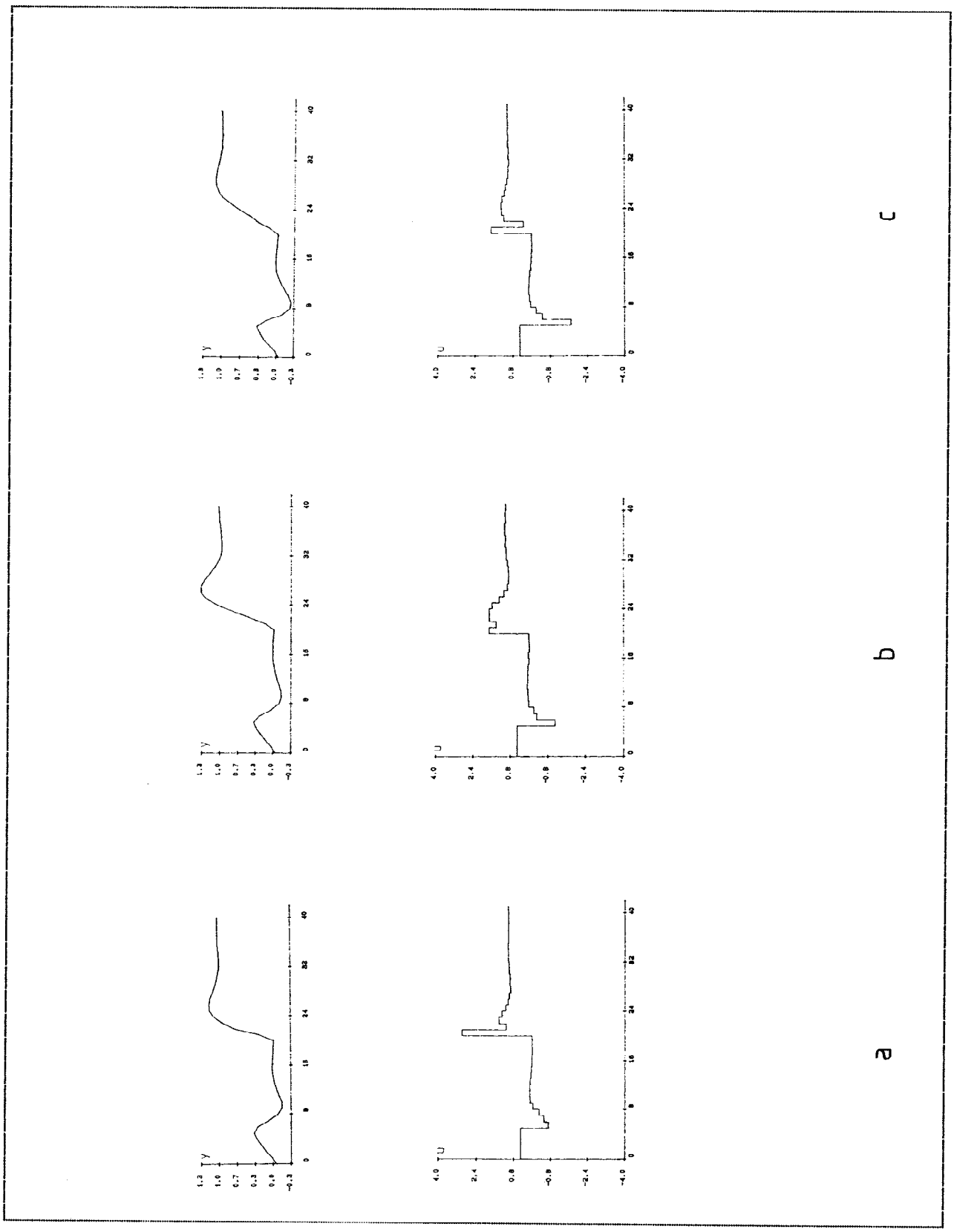


Fig 4.37. Saturación del actuador sin/con anti-reset windup.

Como se ve, en (b) y (c) se produce la saturación del actuador, siendo continuada en (b) y tardando varios períodos de muestreo en recuperar el valor de régimen permanente posteriormente (efecto windup). En (c), con la aplicación del mecanismo correspondiente, se soluciona el problema.

4.2.5 Preidentificación.

Este módulo gestiona la fase de preidentificación anterior a la de control en tiempo real de un sistema físico, cuando la opción para la ejecución de dicha fase es seleccionada por el usuario mediante menú.

El procedimiento "Preidentificación" gobierna la ejecución de esta fase, obteniendo el período de muestreo conveniente para el control del sistema físico, su orden, retardo así como el valor del parámetro Covlím del mecanismo de supervisión de la estimación apropiado.

Cuando la fase finaliza, se presentan estos datos al usuario, pudiendo ser modificados por el mismo, antes de su utilización o su almacenamiento en un fichero en disco llamado datos.pre. El formato de dicho fichero es:

T_m d m Covlím (T_m viene dado en ms * 10).

De esta manera no es necesaria la realización de la fase de preidentificación cada vez que se ejecuta el sistema para trabajo en tiempo real, sino únicamente cuando se modifica el proceso conectado físicamente al mismo.

El resto de procedimientos del módulo son auxiliares del anterior. Así "THex" transforma el período de muestreo obtenido T_m en las variables necesarias, en hexadecimal, para programar los registros del temporizador. El procedimiento "Fin" espera hasta que la salida del proceso llegue al régimen permanente, una vez que se ha producido una variación en la entrada del mismo. De esta forma se puede obtener el valor final ante una determinada entrada.

El procedimiento "OrdenMod" realiza la identificación de los parámetros del proceso, suponiendo éste de un orden dado, obteniendo la suma de los componentes del vector de errores al cuadrado, para dicho orden. Mediante la comparación de los resultados obtenidos suponiendo distintos órdenes (de 1 a 4), en "Preidentificación", se obtendrá el mejor ajuste.

En la figura 4.38 se pueden observar los resultados parciales y las conclusiones finales, en la ejecución de esta fase, así como la modificación por parte del usuario de alguna de las mismas.

Aún cuando se posean los datos correspondientes al orden, tiempo muerto del proceso, etc. existen diversas consideraciones a realizar respecto al arranque de los controladores adaptativos.

Así, en la figura 4.39 (a) se muestran las variables w, u e y durante los primeros períodos de muestreo del control adaptativo del proceso P4.1 (con DBN, τ=0.95, k estimación previa =1, multiplicador PRBS =0.7). En la figura 4.39 (b) se observa el mismo caso anterior con k estimación previa =3.

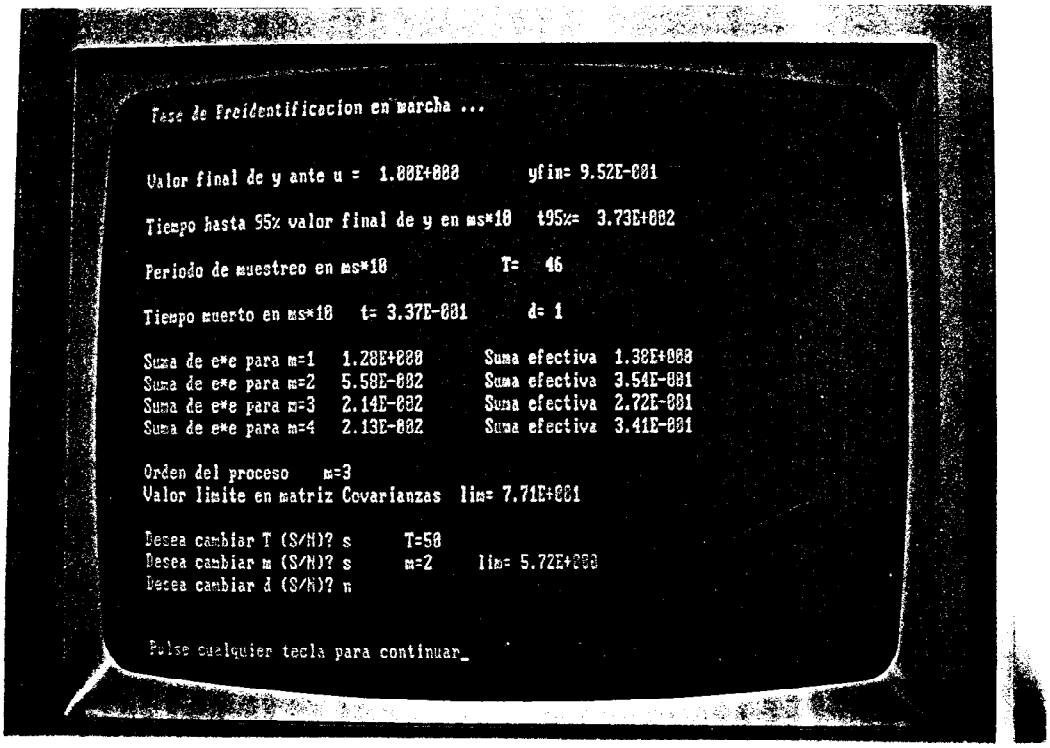


Fig 4.38. Ejecución de la fase de preidentificación.

Como se ve, en el primer caso el arranque se efectúa más bruscamente, al no poseer el módulo de diseño estimaciones suficientemente aproximadas de los parámetros del proceso, cuando sintetiza el

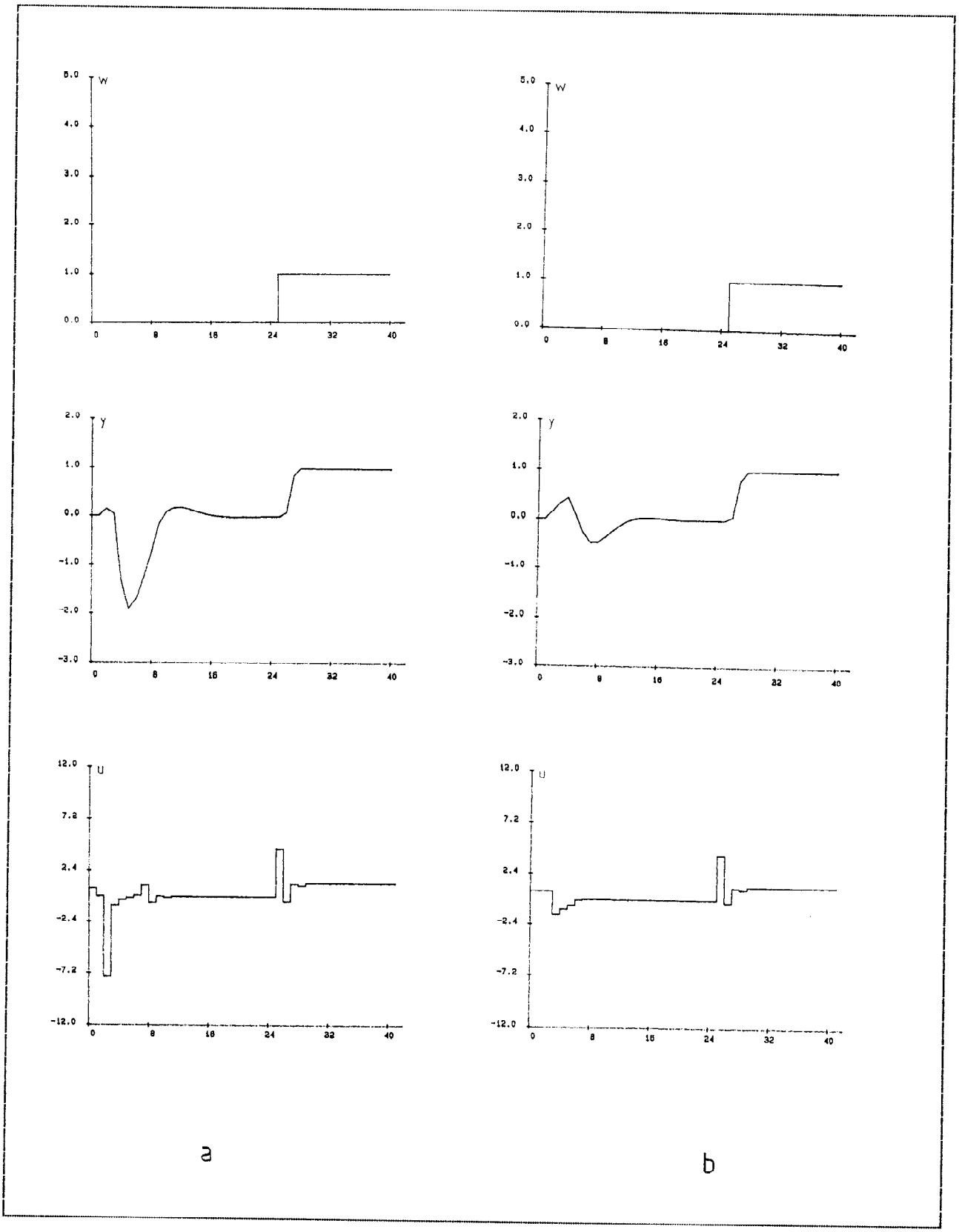


Fig 4.39. Efecto de la estimación previa de parámetros.

controlador en los primeros períodos de muestreo.

Así mismo cuando el proceso a controlar es inestable, es necesario efectuar el arranque con un controlador fijo en cadena cerrada, como se ha visto en el apartado 4.2.2.3.

4.2.6 Supervisión.

El sistema desarrollado incluye los mecanismos de supervisión adecuados para el correcto funcionamiento en tiempo real, tanto en lo que se refiere a la estimación de parámetros y a la síntesis del controlador, como al comportamiento en bucle cerrado.

4.2.6.1 Inclusión de factor de olvido en el estimador.

En la figura 4.40 (a) puede observarse el efecto del empleo de un factor de olvido constante y menor que la unidad, en el algoritmo de estimación de parámetros RLS. En dicha figura se muestran los valores de P(1,1) y P(3,3), con τ=0.9, para el sistema de segundo orden implementado en la placa de pruebas, que corresponden al primer parámetro del denominador de la función de transferencia (asociado con salida del proceso) y también al primero del numerador (asociado con la entrada), respectivamente.

Como se puede ver en la figura 4.40 (b) los efectos al aplicar un factor de olvido de valor superior (τ=0.95) se siguen produciendo, aunque se puede comprobar que en este caso el crecimiento exponen-

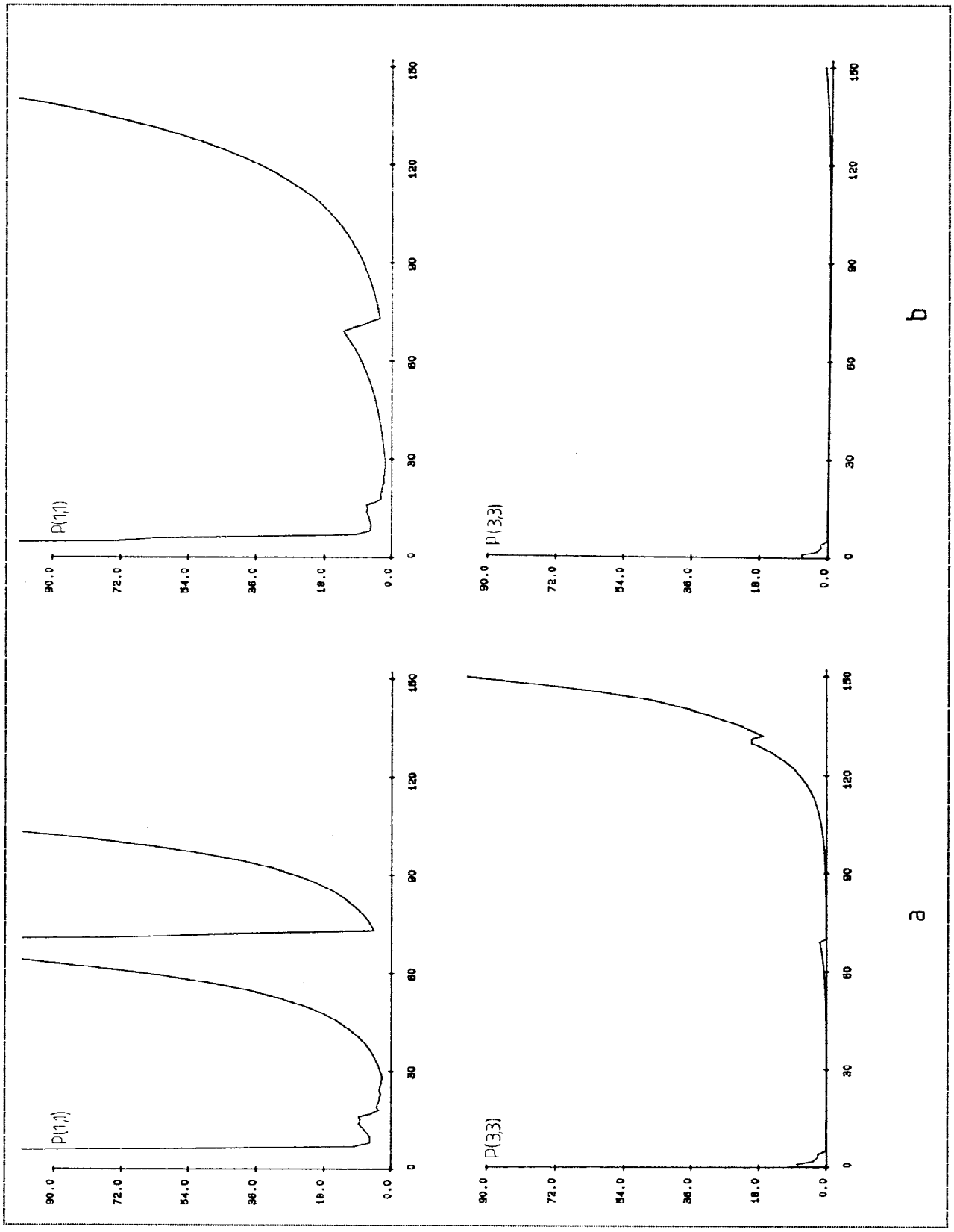


Fig 4.40. Blow-up en la matriz de covarianzas con τ=0.9 y τ=0.95.

cial de los elementos P(1,1) y P(3,3) es más suave. Por tanto únicamente la aplicación de τ=1 o de los mecanismos de supervisión adecuados, aseguran que no se ocasione el "blow-up" en la matriz de covarianzas, con los problemas que lleva asociados.

En la figura 4.41 se muestra el incremento producido en la varianza de las estimaciones de dos de los parámetros, con τ=0.85 fijo, a partir del instante en que se produce el "blow-up" en la matriz de covarianzas. Evidentemente este fenómeno, conocido como "bursting", deteriora el comportamiento en bucle cerrado del sistema, como se observa en la representación de las variables controlada y de control, cuando se sigue la consigna indicada.

Como se puede comprobar en la figura 4.42, la aplicación directa del método de factor de olvido vectorial propuesto por Saelid y Foss, con algoritmo de identificación RLS convencional, da lugar a problemas numéricos. En la figura se muestra el parámetro estimado t_1, así como el elemento de la diagonal de la matriz de covarianzas asociado P(1,1), en el que no se detectan problemas de crecimiento exponencial aunque sí numéricos.

4.2.6.2 <u>Factor de olvido de Saelid y Foss modificado.</u>

Los problemas anteriores quedan resueltos con las modificaciones propuestas en el apartado 2.3.1.3. En la figura 4.43 (a) se muestra el coeficiente de la diagonal de la matriz de covarianzas asociado al parámetro estimado t_1, así como el componente τ_1 del vector de

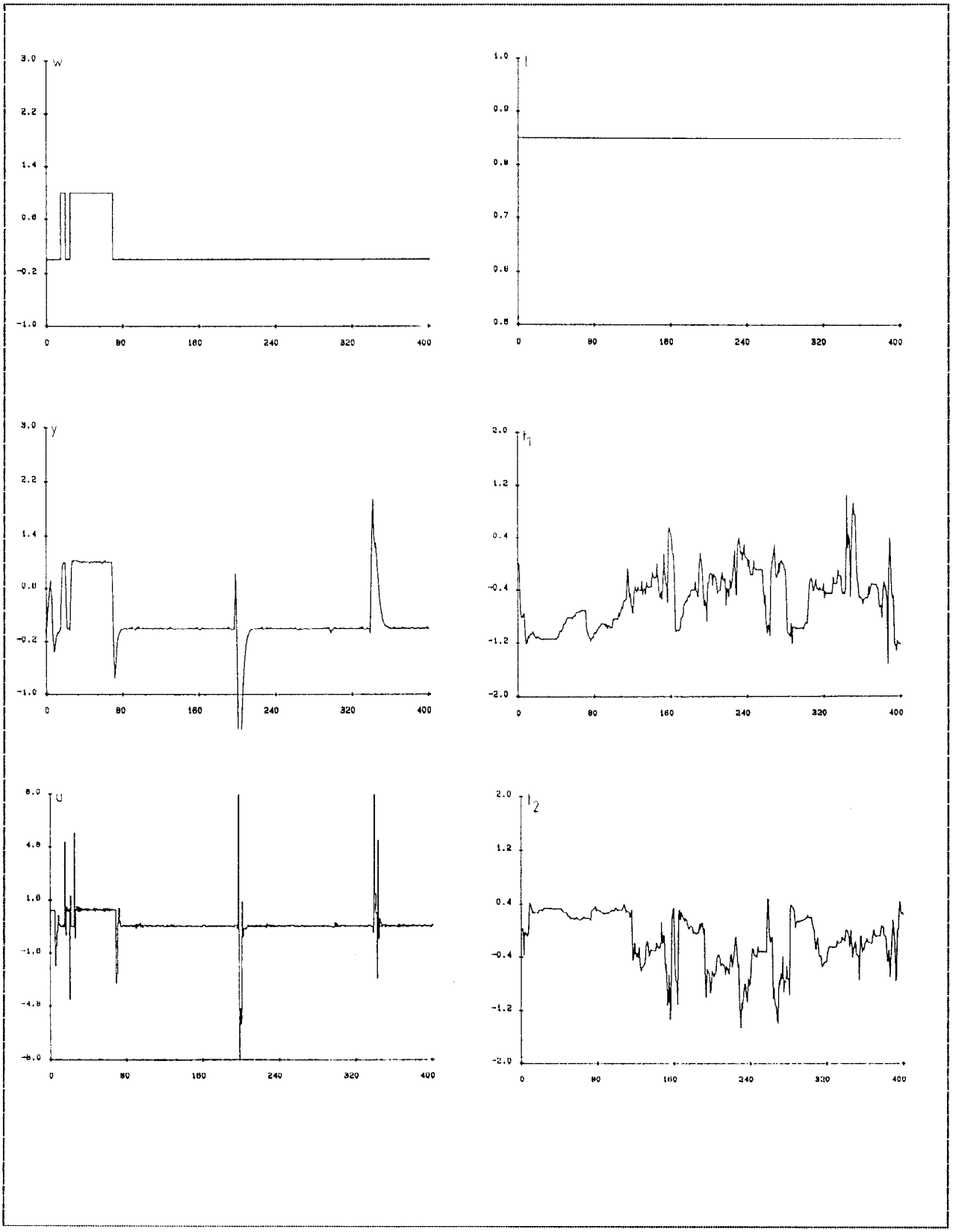


Fig 4.41. Efectos del blow-up en estimación y control.

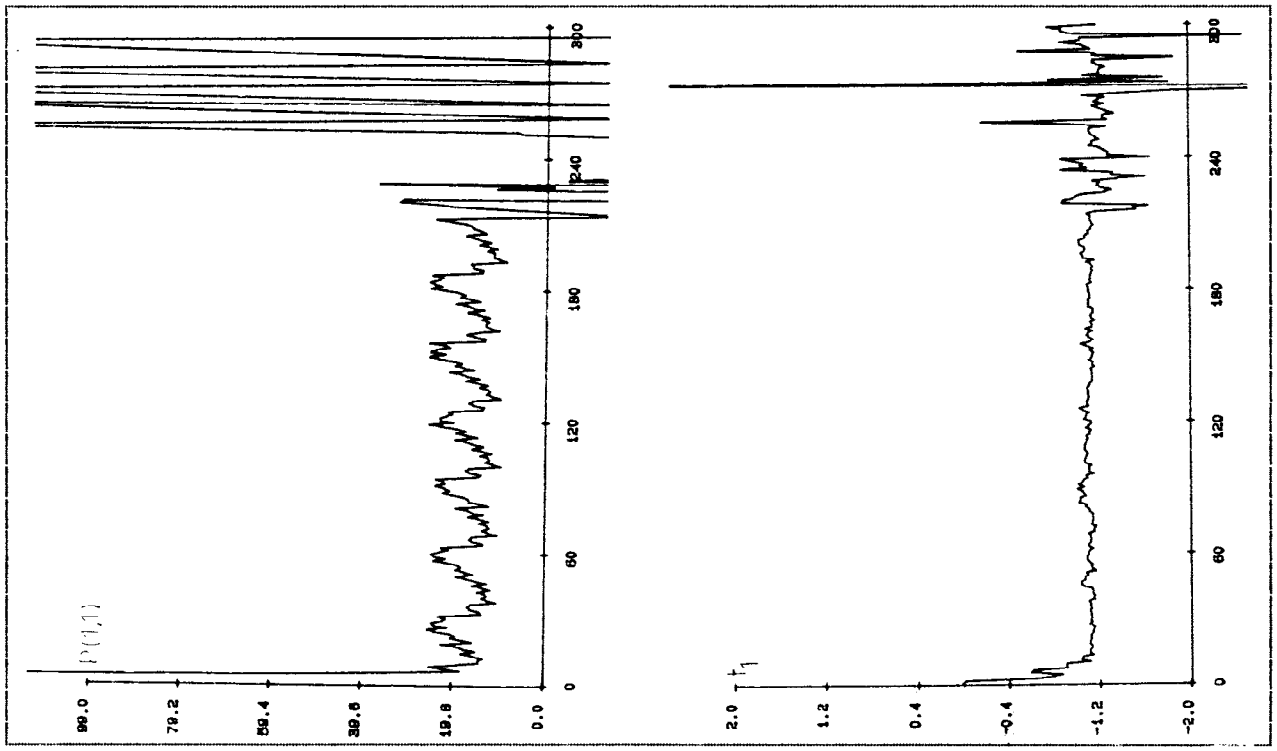


Fig 4.42. Aplicación directa del método de Saelid y Foss con RLS.

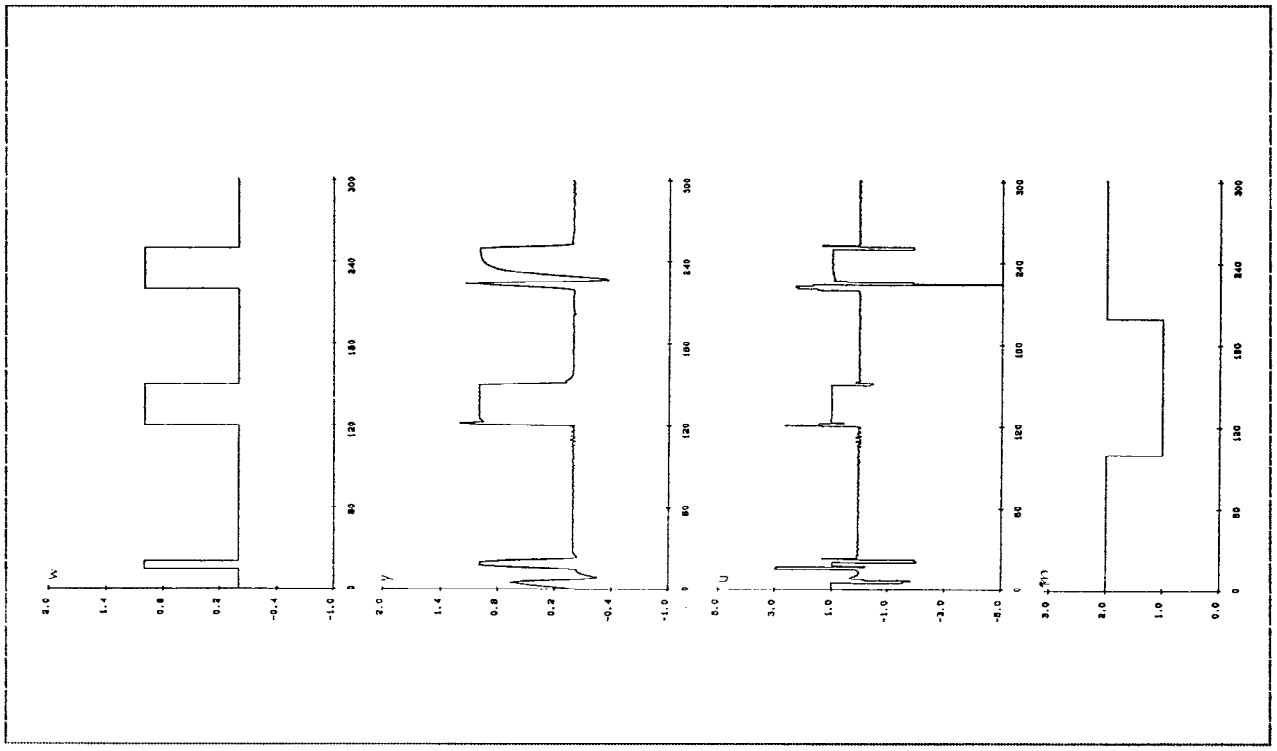


Fig 4.46. Comportamiento ante variaciones en el orden del proceso.

factores de olvido. En (b) se muestran los mismos valores asociados ahora al parámetro t_3.

Se elimina con estas modificaciones el blow-up (adecuando el valor del factor de olvido a la excitación existente), sin entrar en problemas numéricos. El valor en el cual deben permanecer constantes los coeficientes de la diagonal de la matriz P se ha seleccionado a 25.0. Como se puede comprobar tanto P(1,1) como P(3,3) vuelven a dicho valor, después de períodos de elevada excitación, con lo que no se pierde la capacidad de seguimiento de las variaciones de los parámetros.

Así mismo se muestra en (c) la inexistencia del "bursting" en las estimaciones de los parámetros t_1 y t_2, y en (d), el correcto seguimiento de la consigna por parte de la variable de salida, así como la variable de control.

4.2.6.3 <u>Variaciones en los parametros del proceso.</u>

Utilizando el método anterior para el uso de factor de olvido sin problemas adicionales, es posible el seguimiento de variaciones en los parámetros estimados. En la figura 4.44 se muestran la estimación del parámetro t_1, así como el elemento de la diagonal de la matriz P asociado, cuando se somete el proceso P2.1 a una simulación de estimación de parámetros, con τ=0.9, COV INIC =999.0 y se produce un cambio, en k=130, pasando el valor de a_1 -1.227 a -0.6.

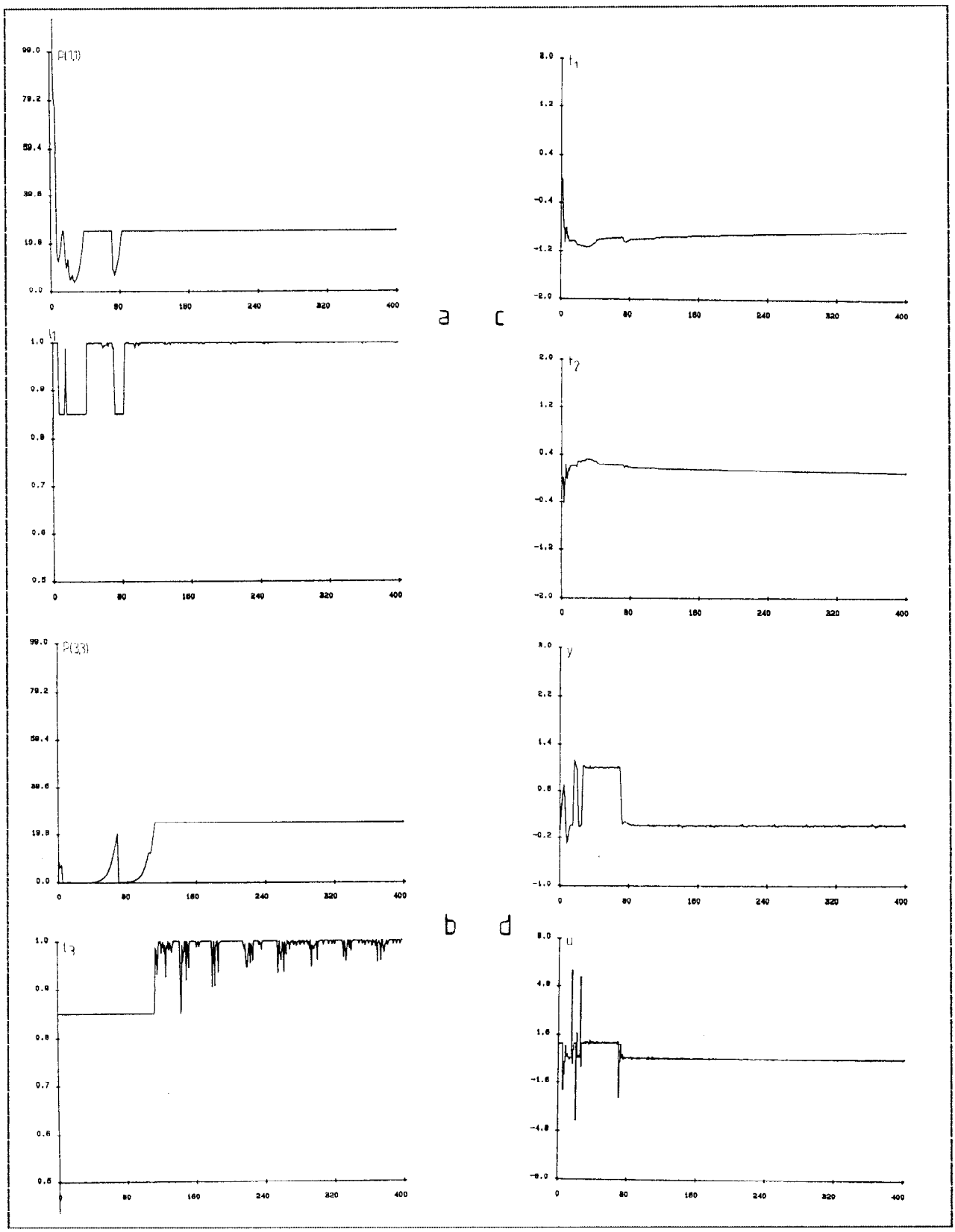


Fig 4.43. Aplicación del método de Saelid y Foss modificado.

Como se puede comprobar, la convergencia al nuevo valor del parámetro estimado se efectúa lentamente, ya que la ganancia del estimador, en función de P, es reducida cuando se produce el cambio. En la misma figura se observa la secuencia de errores de la estimación, que difiere considerablemente de sus valores anteriores en el momento de producirse el cambio, lo cual puede ser aprovechado para realizar la detección del mismo.

En la figura 4.45 se observa la simulación del caso anterior cuando se procede a la detección del cambio y posterior incremento de la ganancia del estimador (reinicialización de la matriz de covarianzas), basándose en el correspondiente error de estimación. Como se puede comprobar la rapidez de convergencia en este caso es muy superior.

4.2.6.4 Variaciones en el orden y tiempo muerto del proceso.

Como se ha indicado en el apartado 3.6.3, el sistema de control adaptativo es extremadamente sensible ante un modelado incorrecto del tiempo muerto del proceso, siendo algo más tolerante en el caso del orden del mismo, sobre todo en la sobreestimación del mismo.

Así se muestran en la figura 4.46 las señales de consigna, salida y control cuando se pasa bruscamente de tomar la salida (en la placa de pruebas) del proceso de segundo orden al de primero (en k=100) y viceversa (k=200). Se ha utilizado control adaptativo (con DBA) más supervisión, para que en los cambios las estimaciones de los

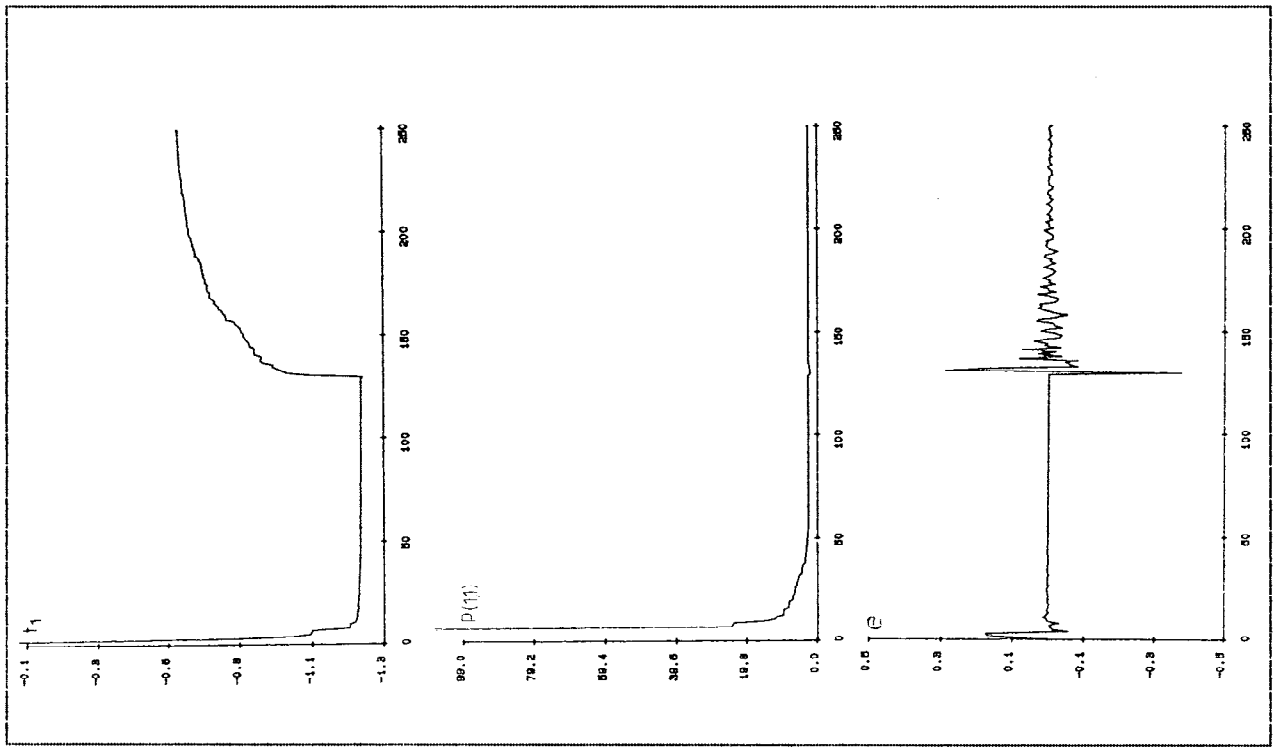


Fig 4.44. Seguimiento de cambios en los parámetros sin supervisión.

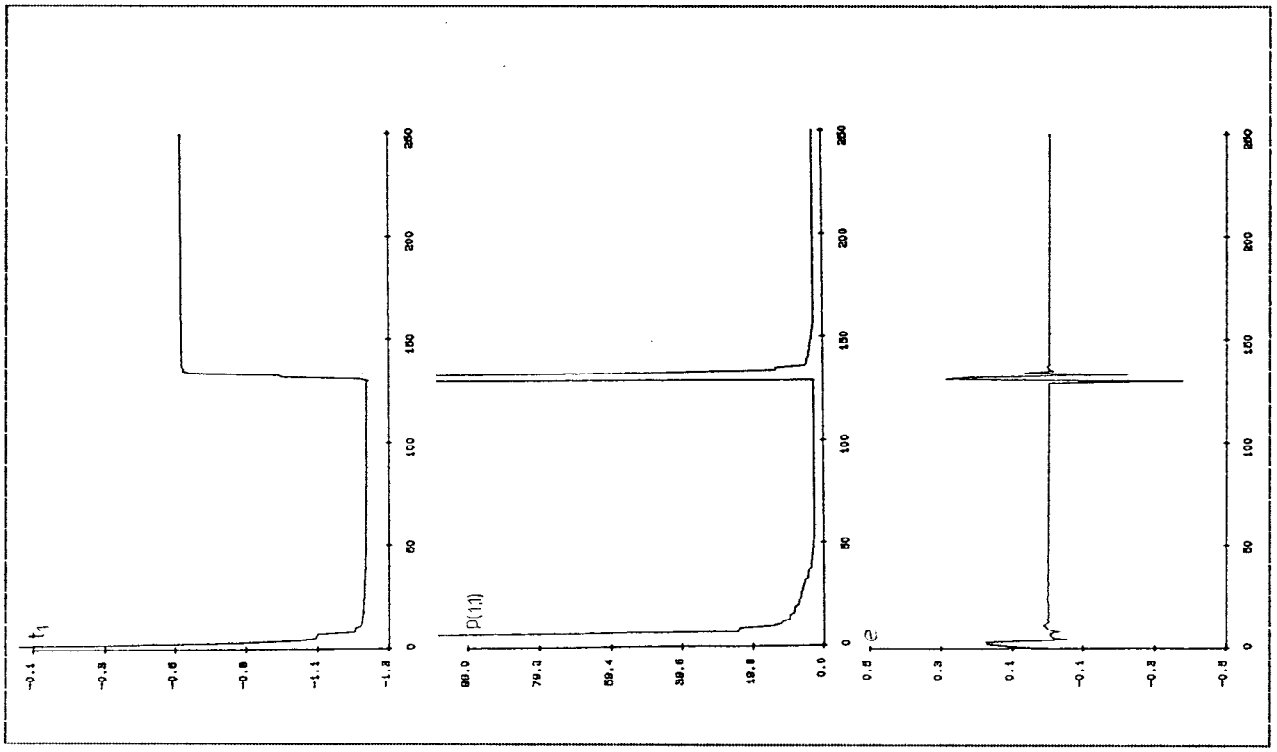


Fig 4.45. Seguimiento de cambios en los parámetros con supervisión.

parámetros converjan rápidamente (dependiendo de la excitación) a los nuevos valores.

En la figura 4.47 (a) se muestran las variables controlada y de control, así como el tiempo muerto real y estimado del proceso y el error de estimación cuando se realiza una simulación de control adaptativo con DBN al proceso P2.1, sin la aplicación de ningún mecanismo del supervisor para seguir las variaciones del tiempo muerto del proceso.

Se comprueba que una diferencia entre el tiempo muerto real y el estimado de dos o más unidades, lo que sucede en este caso a partir de k=180, llevan a la inestabilidad al sistema de control adaptativo en la siguiente variación de las condiciones de funcionamiento (en este caso, cambio de consigna en k=185).

En la figura 4.47 (b) se muestra el resultado de la aplicación del método especificado en el apartado 2.3.1.5 al caso anterior. La aplicación se produce únicamente en los períodos de muestreo siguientes a la detección de un cambio en el proceso (mediante el seguimiento de la señal e(k)), que ocasiona así mismo una reinialización de la matriz de covarianzas (en la misma figura se observa P(1,1)).

Se puede ver que el seguimiento de la consigna por parte de la variable controlada es correcto, aunque el período de tiempo transcurrido entre la detección de una variación en el proceso y la

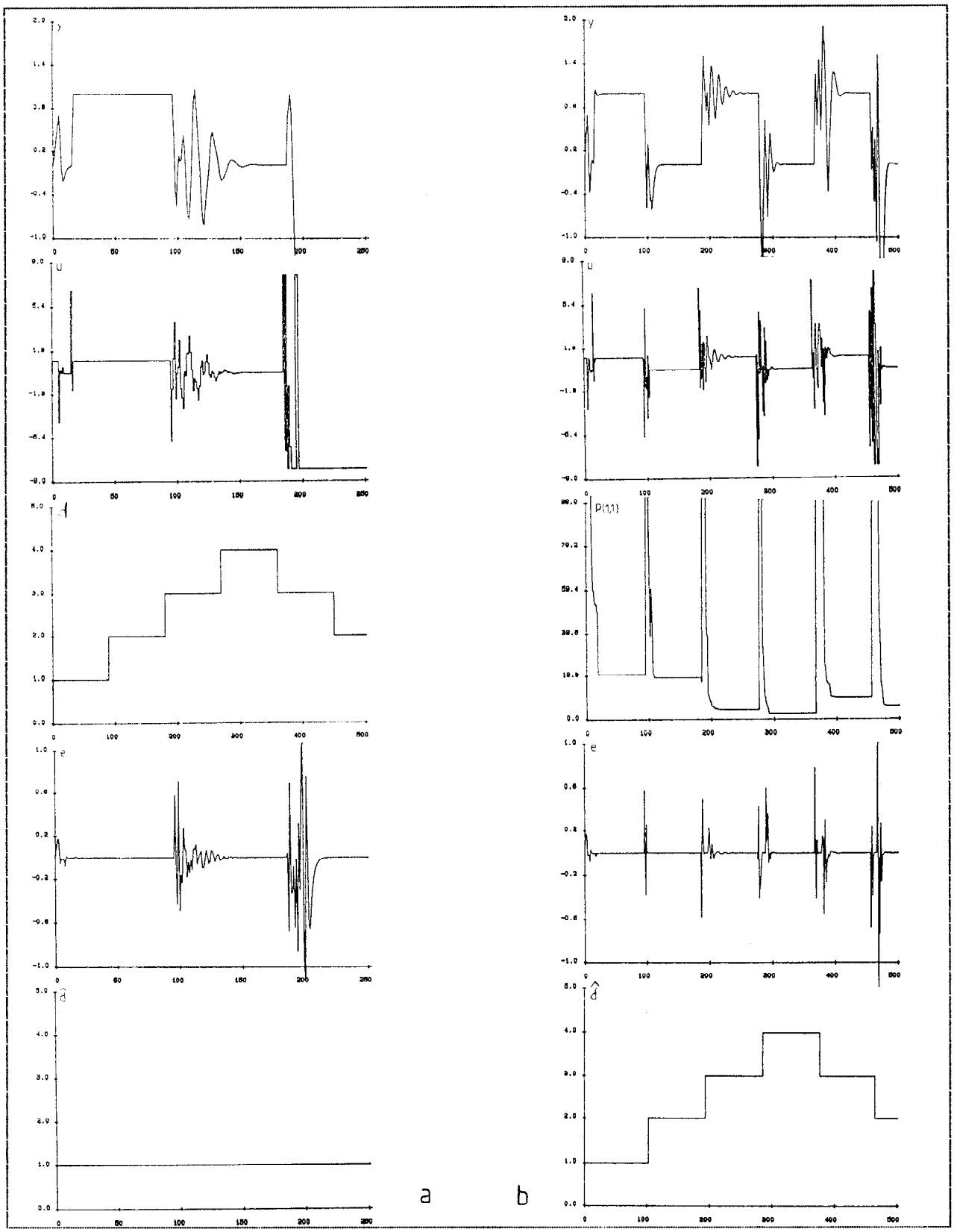


Fig 4.47. Control sin/con seguimiento de las variaciones de "d".

certeza de que la misma conlleva un cambio en el tiempo muerto del mismo, produce un transitorio en cada cambio con sobreoscilaciones elevadas. En la misma figura se muestra el tiempo muerto estimado del modelo del proceso, que coincide así con el real después de un cierto retardo para efectuar la detección del cambio.

El transitorio en cada cambio puede ser mejorado con la aplicación de un controlador de respaldo, previamente definido, una vez se ha detectada una variación en el proceso. Así, en la figura 4.48 se muestran la variable controlada y la de control del proceso anterior con d=3 (P2.2), así como el tiempo muerto real y la consigna. En la figura 4.49 se aplica en el momento del cambio un controlador de respaldo diseñado con DBN y d=3, que como vemos consigue mejorar el transitorio, respecto al caso anterior.

Es posible mejorar aún los resultados con la aplicación de otros métodos de diseño para el controlador de respaldo, aunque evidentemente será necesario disponer $d_{max}-d_{min}$ controladores previamente especificados e ir conmutando al apropiado en función de la estimación realizada del tiempo muerto del proceso.

4.2.6.5 Comportamiento del tratamiento de variaciones en tiempo real.

El comportamiento es similar al visto mediante simulación y se van a resumir los resultados en los tres casos siguientes, aplicados al proceso de segundo orden implementado en la placa de pruebas.

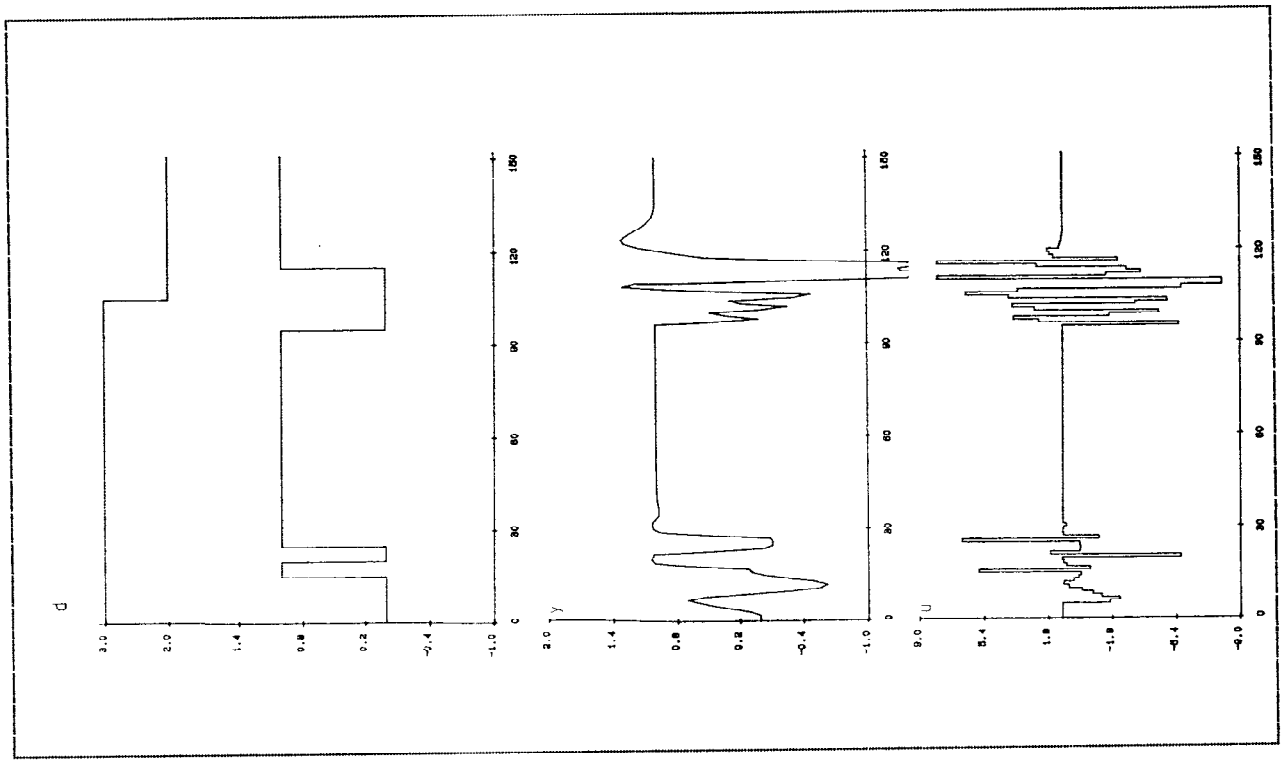


Fig 4.48. Cambio en "d" sin controlador de respaldo.

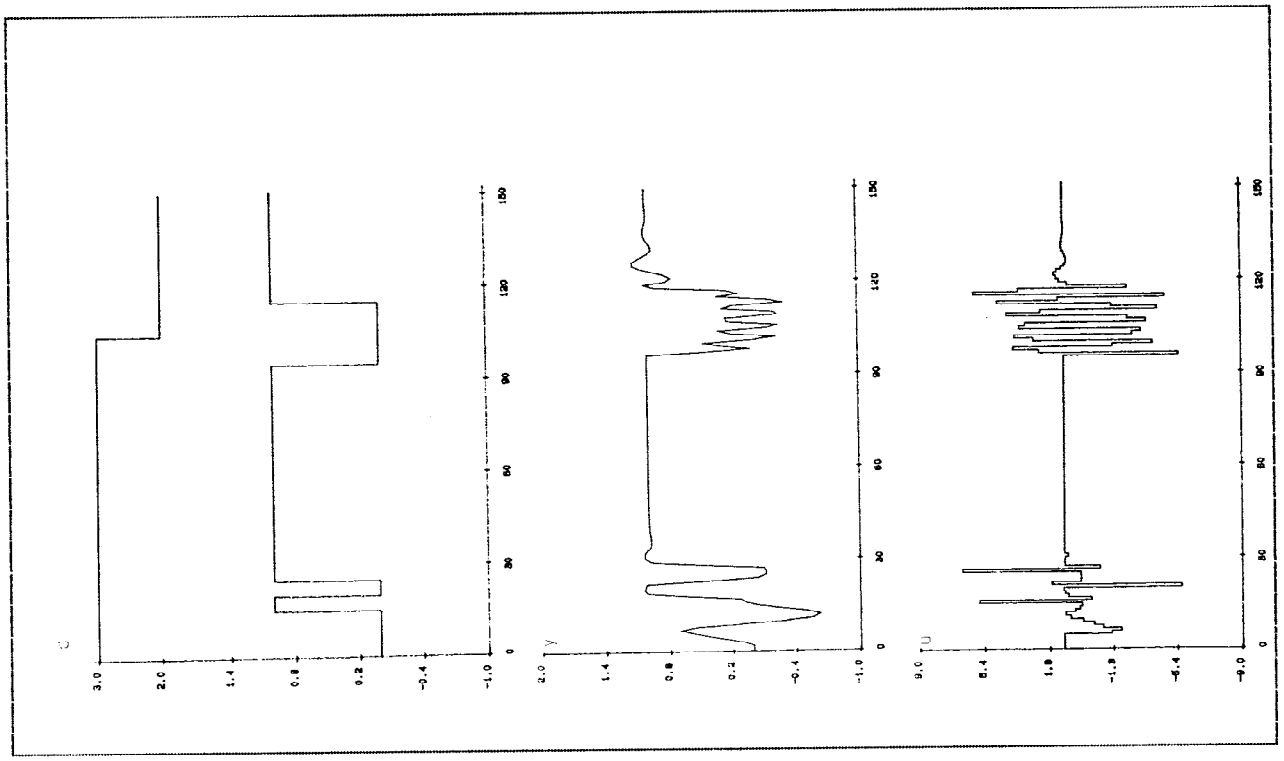


Fig 4.49. Cambio en "d" con controlador de respaldo.

En las figuras 4.50 y 4.51 se muestran los parámetros estimados t_1 y t_2 en los casos en que se emplea o no, respectivamente, detección de cambios de los mismos mediante el supervisor. Se muestra un incremento de la rapidez de convergencia apreciable en el primer caso, cuando se produce un cambio en los mismos en k=100.

En la figura 4.52 se muestra el correcto seguimiento, mediante estimación en tiempo real, de los parámetros así como del tiempo muerto del proceso, cuando se activa el mecanismo de detección de cambios, en el caso de las siguientes variaciones en el proceso:

(a) k=90 d=2 (b) k=180 d=3 (c) k=270 d=4
(d) k=360 cambio de parámetros del proceso con el potenciómetro P1
(e) k=450 d=3

En la figura 4.53 se muestra el comportamiento de la variable controlada y el tiempo muerto estimado, ante las mismas variaciones, en el caso de control adaptativo + supervisión, con DBN, para el correcto seguimiento de la consigna w.

4.2.6.6 <u>Detección de cancelaciones incorrectas.</u>

En la figura 4.54 se muestran los efectos sobre la variable controlada de un cambio en el proceso P2.1 (en k=100, a_2 pasa a valer -0.2), que lo hace inestable, cuando se está utilizando un método de control (DBN) que no es aplicable con polos del proceso

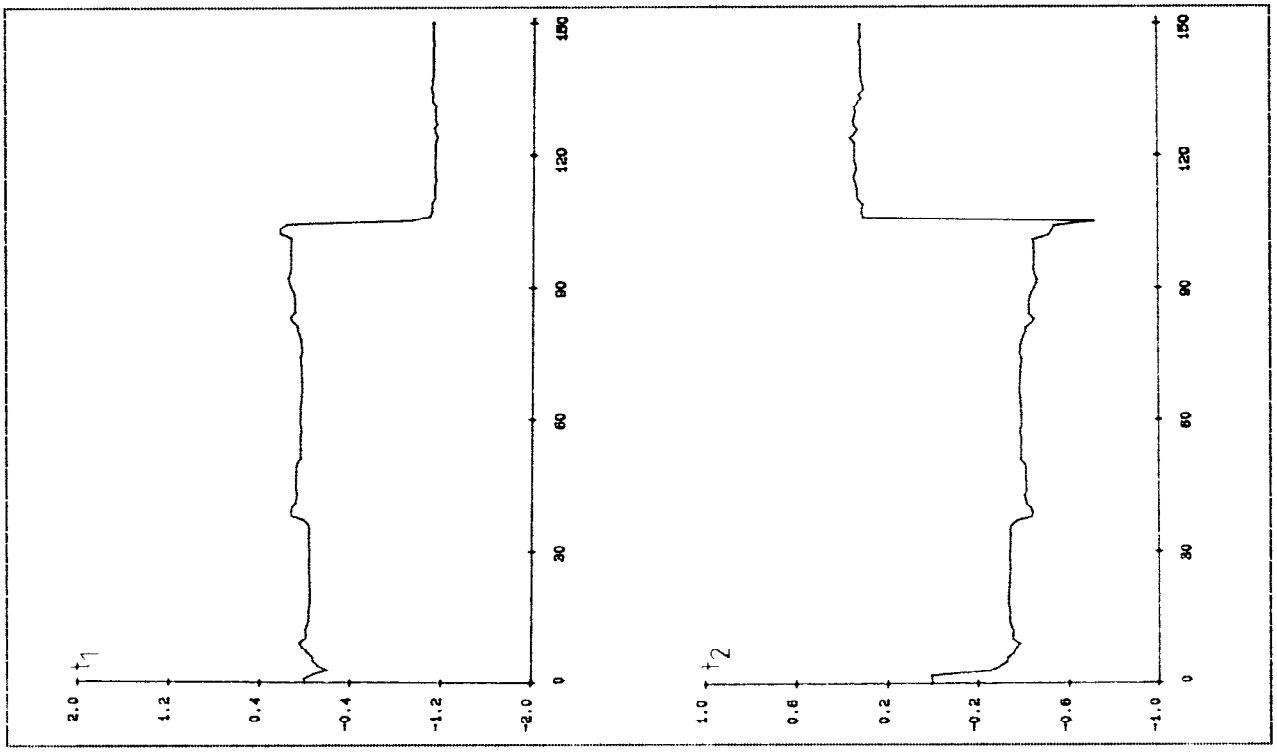


Fig 4.50. Cambio de parámetros en tiempo real con supervisión.

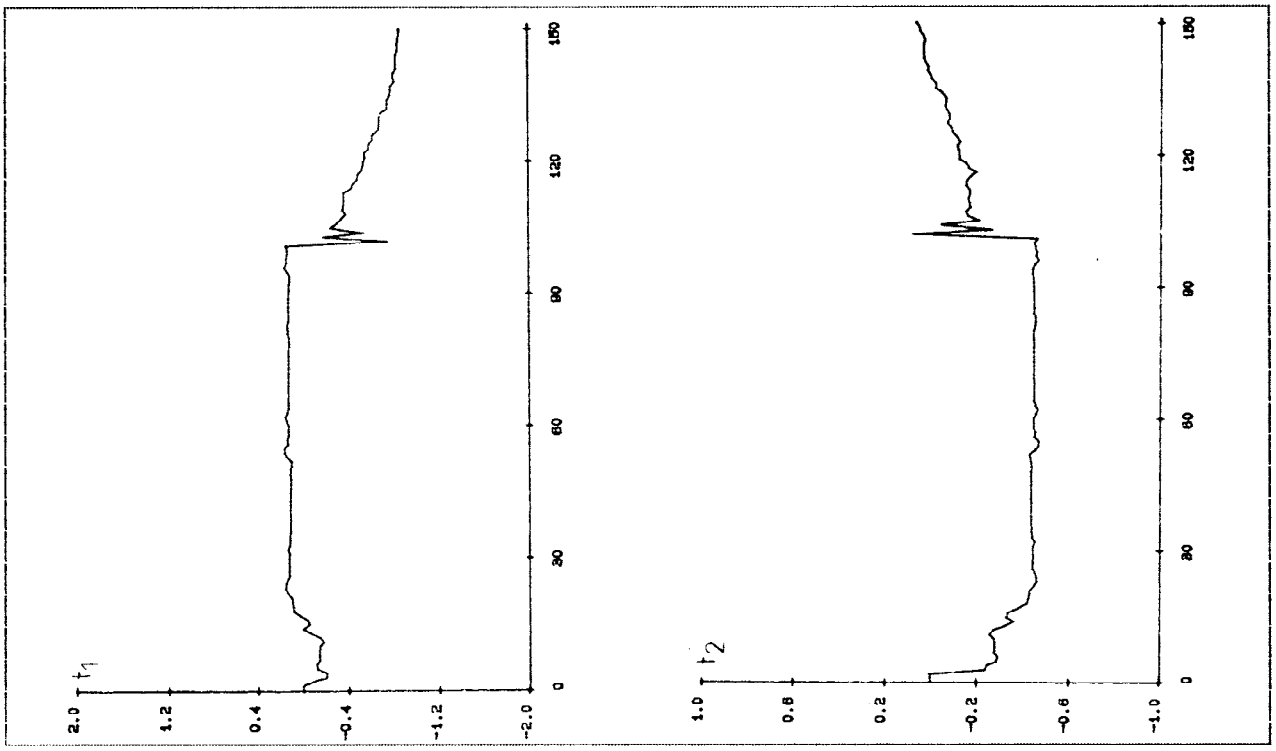


Fig 4.51. Cambio de parámetros en tiempo real sin supervisión.

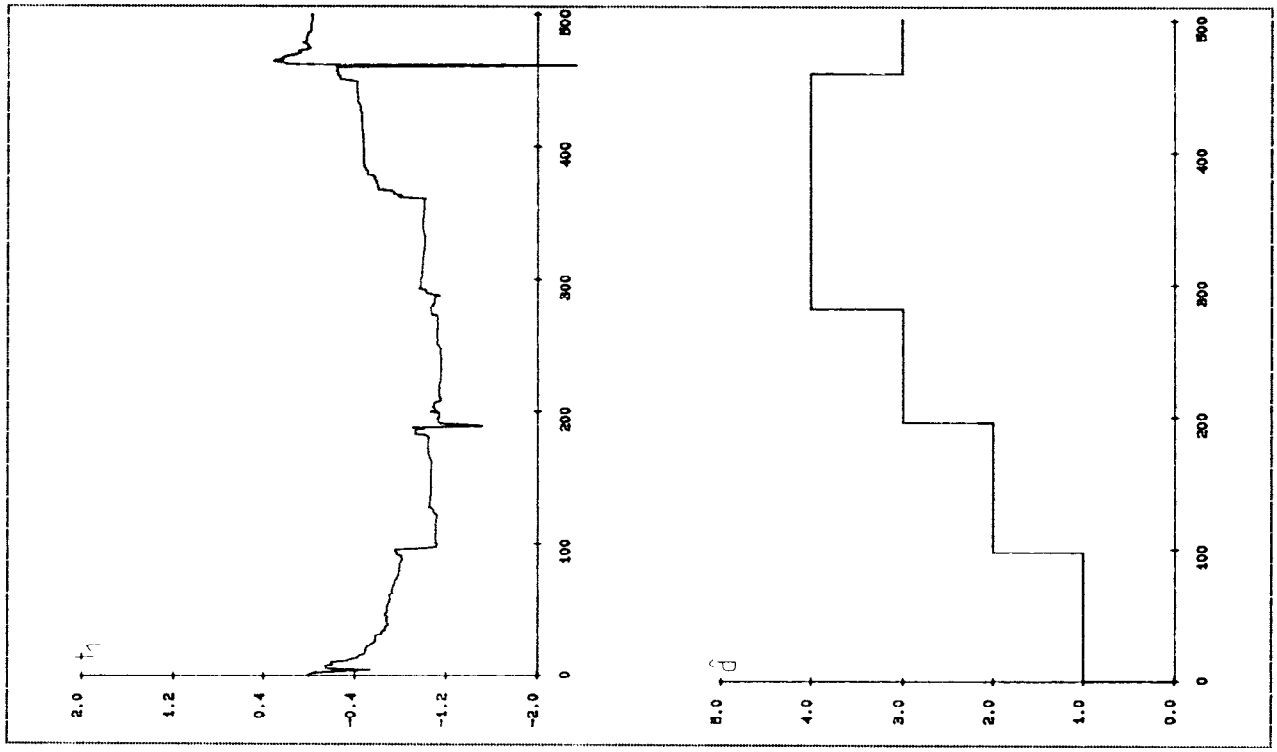


Fig 4.52. Estimación de parámetros variantes en tiempo real.

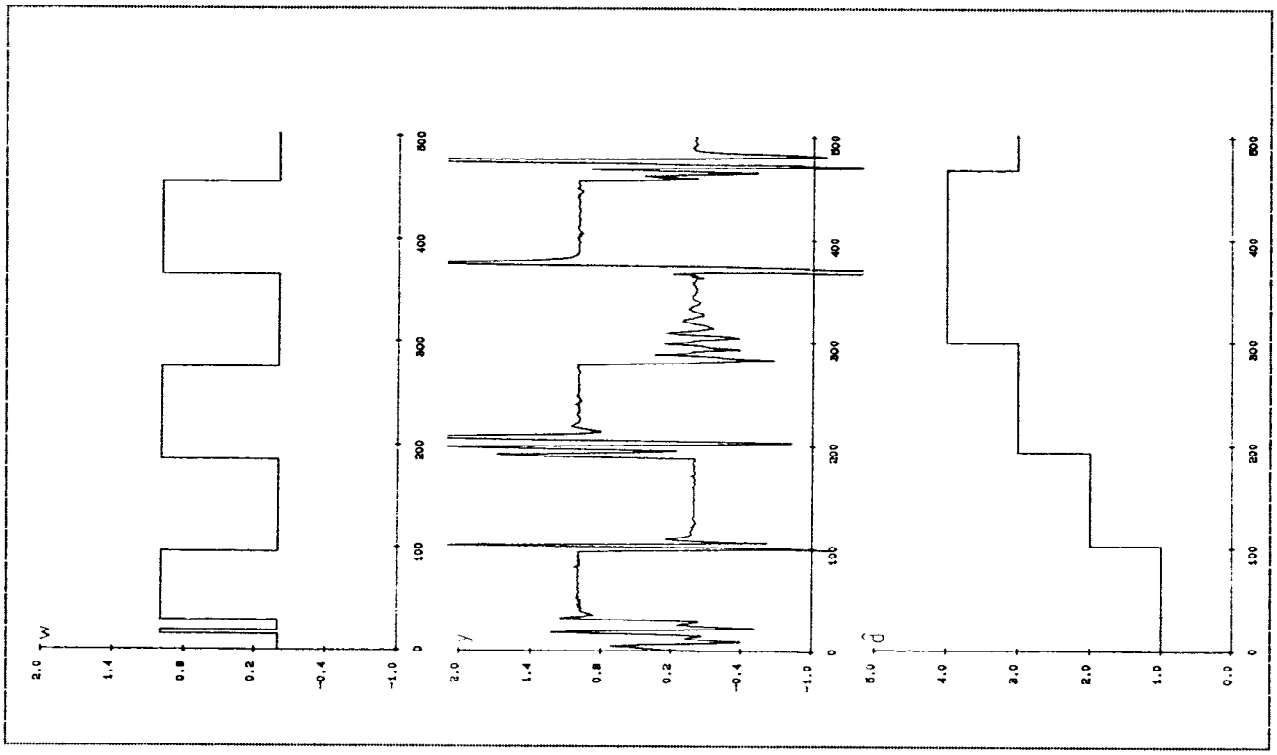


Fig 4.53. Control adaptativo con parámetros variantes en t. real.

fuera del círculo unidad.

Utilizando el supervisor, es posible la detección, previa a la síntesis del nuevo controlador, de la existencia de polos fuera del círculo de radio unidad, invocando entonces a un controlador de respaldo previamente definido, como ocurre en la figura 4.55. Este ha sido calculado por APN con integrador y los polos del sistema en cadena cerrada situados en el origen.

Como se puede comprobar, este controlador actúa unos pocos períodos de muestreo después de producirse el cambio, debido a que el método se basa en las estimaciones de los parámetros para detectar el riesgo de inestabilidad.

4.2.6.7 Supervisión del comportamiento en bucle cerrado.

Para evaluar el funcionamiento del supervisor en relación con el comportamiento en bucle cerrado del sistema, fundamentalmente en lo que respecta a comprobación del error de control y saturaciones del actuador, se va a aplicar al proceso de segundo orden de la placa de pruebas un controlador adaptativo con APN de modo que se incluya integrador y los polos en cadena cerrada estén en el origen.

En la figura 4.56 (a) se muestran la variable de consigna, salida y control en este caso, cuando las restricciones del actuador son $U_{max}=5.0$, $U_{min}=-5.0$. Si en estas condiciones se produce una medida errónea en k=120 se ve en la figura que sus efectos provocan la

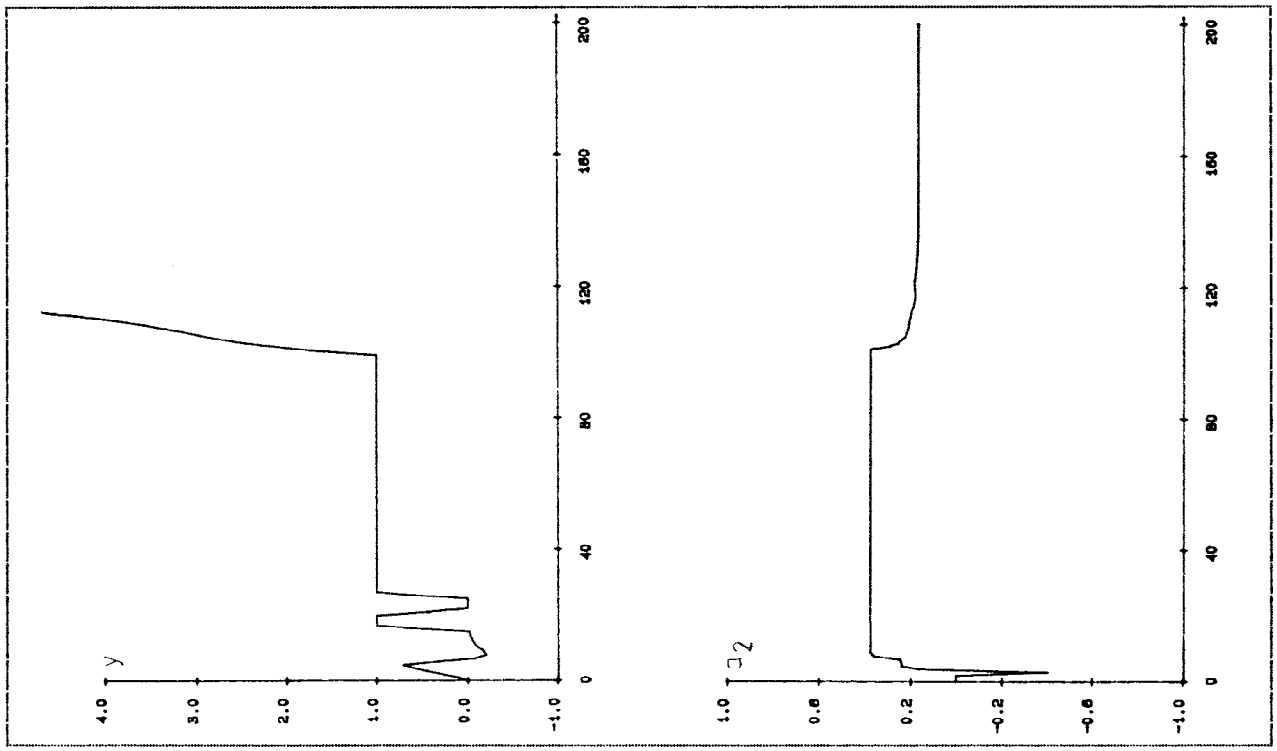


Fig 4.54. Control adaptativo del proceso P2.1 sin supervisión.

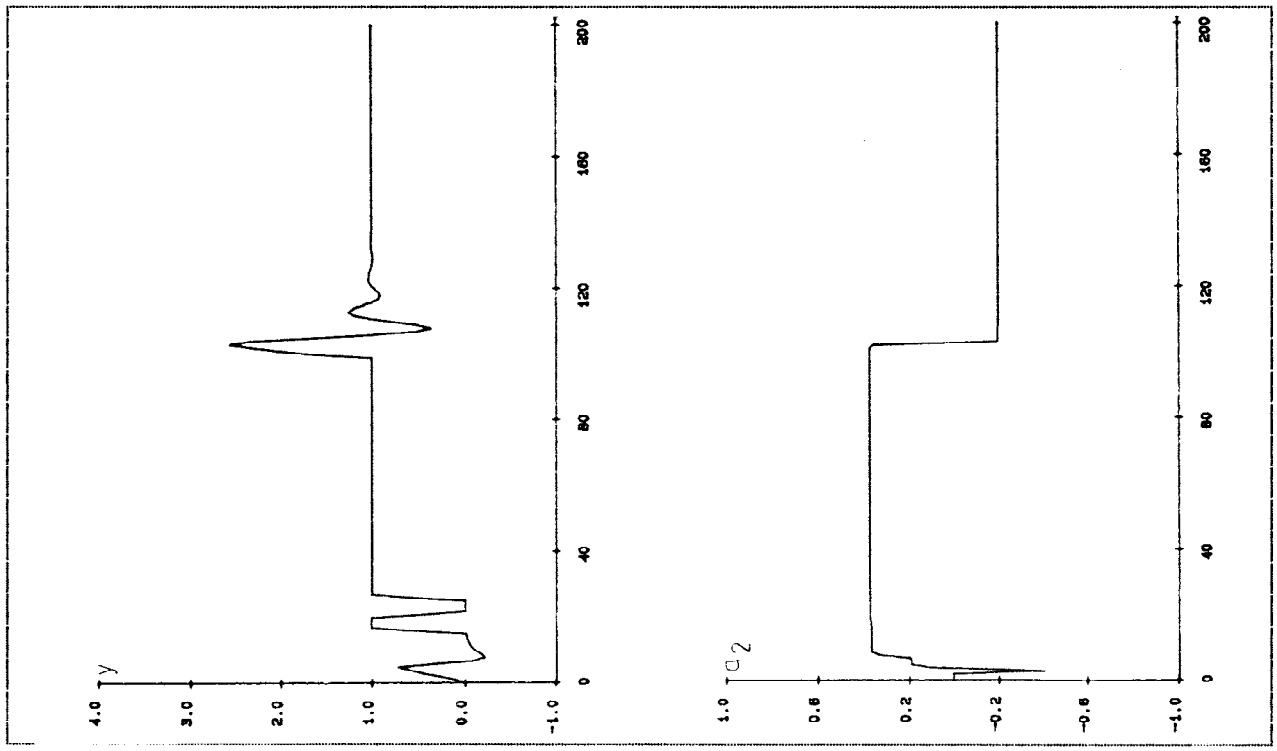


Fig 4.55. Control adaptativo del proceso P2.1 con supervisión.

saturación continuada del actuador, pudiendo llegar a desestabilizar el sistema.

En la figura 4.56 (b) se muestra el efecto del supervisor en este caso, de forma que llame a un controlador de respaldo cuando detecte saturación en el actuador o error de control superior a un cierto límite (seleccionado =3.0) durante un cierto número de períodos de muestreo (seleccionado =2 en este caso). Como se ve el controlador de respaldo fijo (calculado previamente mediante APN con integrador y los polos en cadena cerrada cumpliendo f=0.9 w=1) logra el correcto comportamiento del sistema en bucle cerrado pocos períodos de muestreo después de la detección.

La selección de este número de períodos de muestreo de comprobación antes del paso al controlador de respaldo ha de ser cuidadosamente realizada, pues de ella depende la velocidad de respuesta así como la precisión en la necesidad de dicho paso. Así, al tomar en el caso anterior un número de períodos de muestreo de comprobación muy bajo, se incrementa la velocidad de respuesta, pero también se posibilita que entre el controlador de respaldo en k=23 cuando no era necesario (en la variable c_r se muestran los períodos de actividad de dicho controlador).

Si seleccionamos el número de comprobación a 3, se disminuye (como se muestra en la figura 4.56 (c)) ligeramente la velocidad de respuesta pero se evitan los pasos inadecuados al controlador de respaldo.

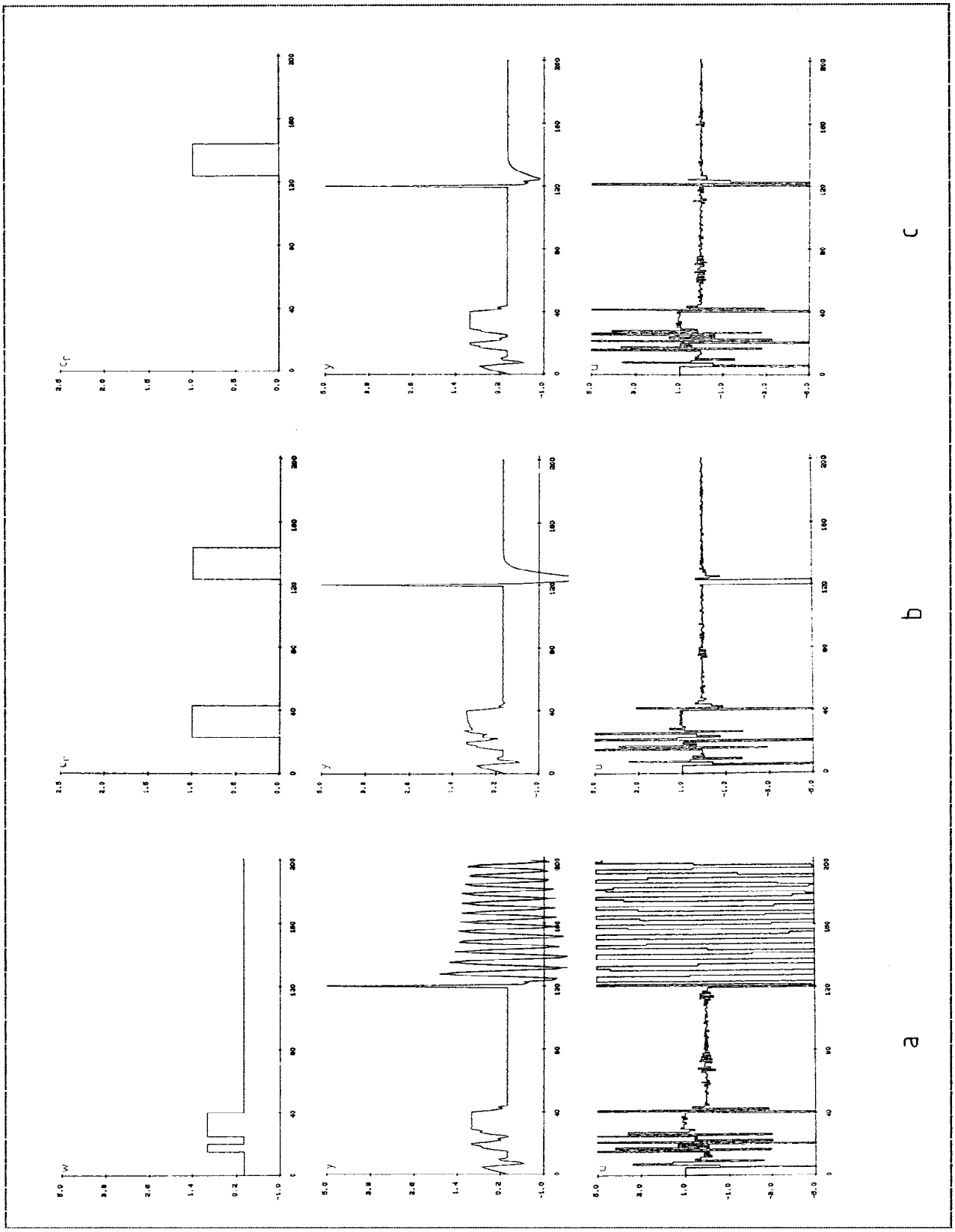


Fig 4.56. Saturación continuada del actuador en tiempo real.

Evidentemente, con un correcto filtrado de outliers se resuelve igualmente el problema en este caso particular, provocado por las medidas erróneas, como se puede comprobar en la figura 4.57.

Por último, en la figura 4.58 se muestran las variables controlada y de control cuando se aplica un mecanismo en el supervisor que resuelva la necesidad de invocación continuada del controlador de respaldo. Las posibles soluciones deberían venir de un nivel superior de toma de decisiones (aplicación permanente del controlador de respaldo, cambio de método de diseño del controlador, paso a modo Manual, etc.).

Aquí se ha tomado esta última posibilidad y se produce la entrada en modo Manual (m_M) a la tercera invocación del controlador de respaldo (c_r) durante 100 períodos de muestreo.

4.3 INTERFACE CON EL USUARIO.

4.3.1 Sistema de menús.

En este módulo se realiza, mediante el procedimiento MENU, la selección de las opciones de funcionamiento del sistema, previamente a su ejecución. El usuario estará asistido por un sistema arborescente de menús que le indicará en cada instante las opciones por defecto, que podrán o no ser modificadas.

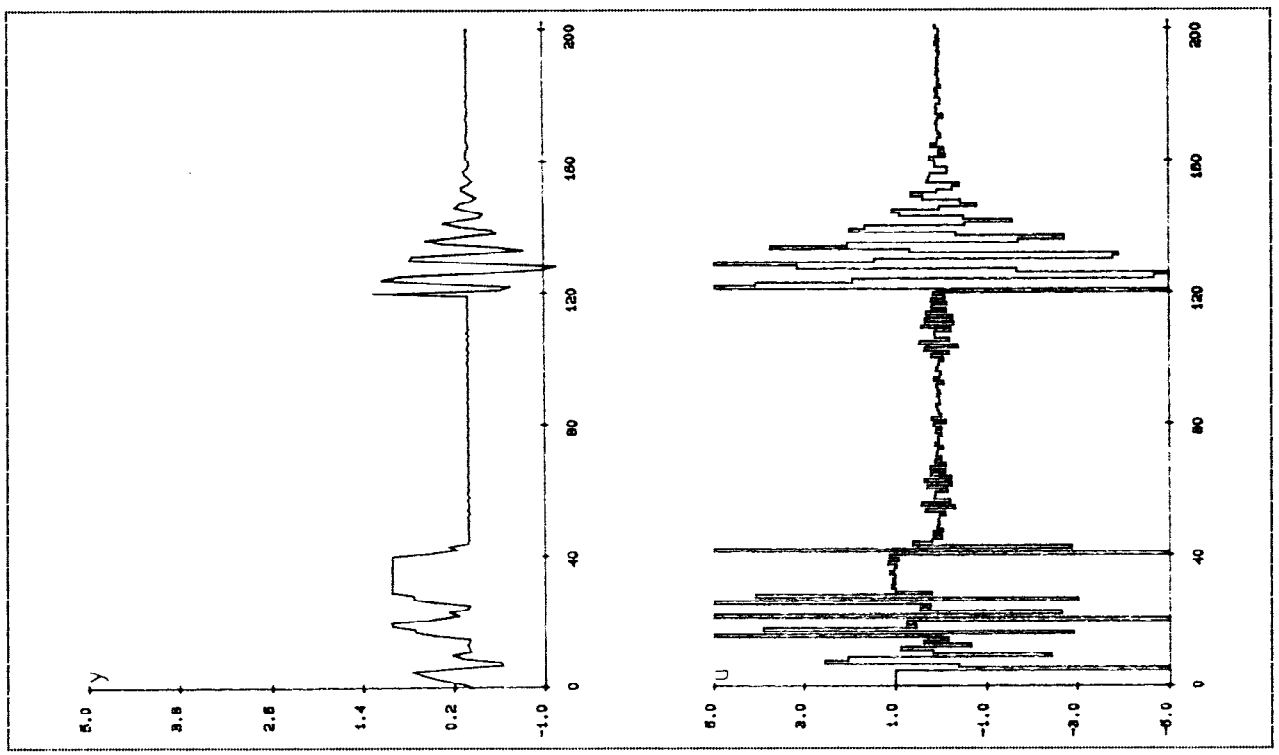


Fig 4.57. Control adaptativo con filtrado de outliers.

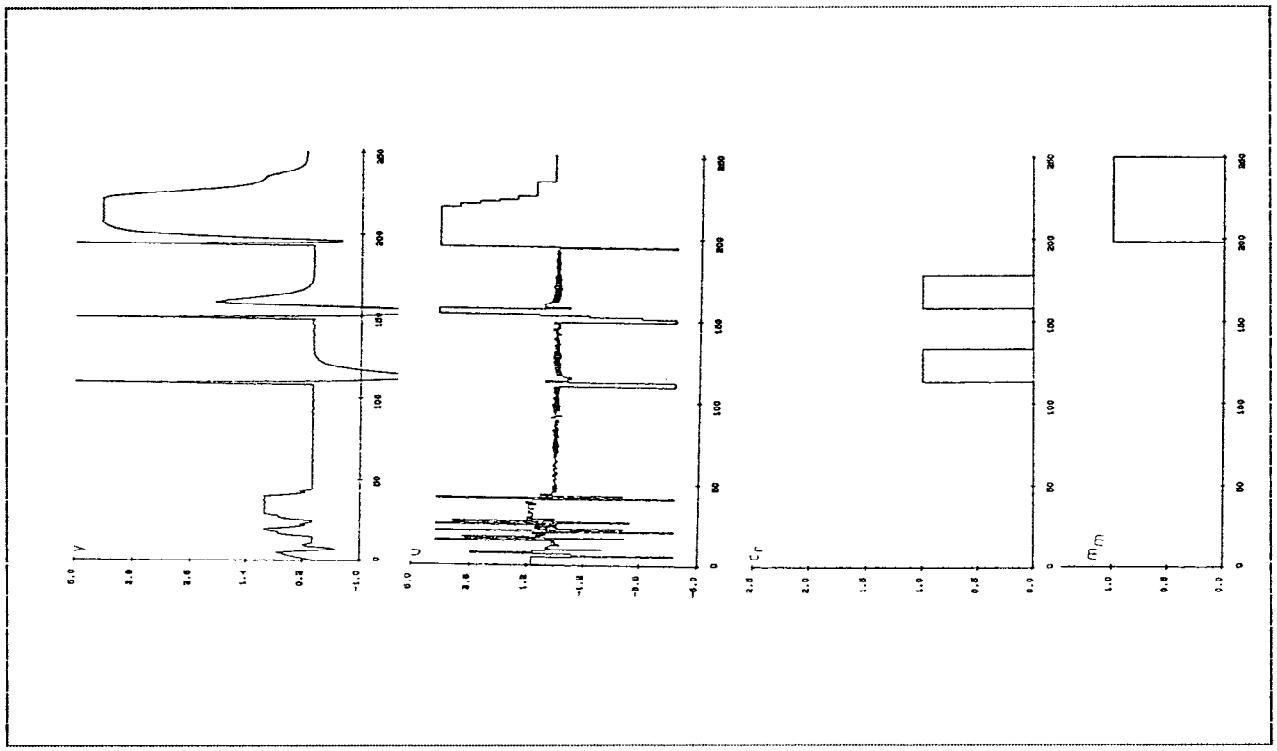


Fig 4.58. Activación continuada del controlador de respaldo.

Mediante el procedimiento Datos del módulo, se almacenan, en el caso de control en tiempo real, los datos procedentes de la preidentificación del proceso, en el caso en que ésta se haya efectuado. Este almacenamiento se realiza en el fichero datos.pre, de donde el mismo procedimiento Datos los lee en cualquier otra ejecución del sistema que no vaya acompañada de la fase de preidentificación.

El mismo procedimiento lee los parámetros del proceso correspondiente al fichero .DAT seleccionado, en todos los casos de simulación además de en los de cálculo de la función de transferencia del controlador y control fijo en tiempo real, para obtener los coeficientes de dicha función de transferencia.

Las opciones que pueden ser selecccionadas con ayuda del procedimiento MENU son las siguientes:

4.3.1.1 Menú principal.

MODO DE FUNCIONAMIENTO: Selección del modo de funcionamiento en la próxima ejecución del sistema MASTER.

SELECCION DE PROCESO: Selección de la función de transferencia del proceso, en el caso de simulación.

CONTROL: Selección del método de control a emplear para hallar la función de transferencia del controlador.

PARAMETROS DE CONTROL: Selección de los parámetros que es necesario ajustar, dependiendo del método de control seleccionado.

COEFICIENTES DEL CONTROLADOR: Selección de los coeficientes de la función de transferencia del controlador, en el caso de que se vaya a emplear uno fijo dado o uno de respaldo en control adaptativo más supervisión, calculado o no con los métodos disponibles.

CONSIGNA: Selección de las opciones relacionada con los cambios de consigna y perturbaciones deterministas en los casos de simulación.

CAMBIO PROCESO: Selección de los cambios a efectuar en la función de transferencia del proceso, en modo simulación.

IDENTIFICACION: Selección de las opciones relacionadas con la estimación de los parámetros del proceso.

ALARMAS: Selección de las opciones relacionadas con la definición de alarmas, límites, aviso al usuario, etc.

FILTRADO: Selección de las opciones relacionadas con el filtrado, analógico o digital, de perturbaciones y outliers.

SUPERVISION: Selección de las opciones correspondientes a la supervisión, cuando ésta se activa en el submenú de modo de funcionamiento.

DATA LOGGING: Selección de las opciones relacionas con el funcionamiento del sistema en modo adquisición de datos.

REPRESENTACION: Selección de las opciones relacionadas con la representación, gráfica o numérica, de las variables apropiadas, dependiendo del modo de funcionamiento seleccionado.

4.3.1.2 Submenú de modo de funcionamiento.

Es posible seleccionar las siguientes opciones (como se puede observar en la figura 4.59), tanto en el funcionamiento en tiempo real (T R) como en simulación (SIM):

ESTIMACION DE PARAMETROS: Realización de la estimación de parámetros del proceso.

CONTROL FIJO: Empleo de un controlador cuya función de transferencia se obtiene, con el método seleccionado, en función de los coeficientes de la del proceso.

CONTROL ADAPTATIVO: Empleo de un controlador cuya función de transferencia se obtiene, con el método seleccionado, en cada período de muestreo, en función de los parámetros estimados del proceso.

CONTROL CON SUPERVISION: Control del sistema incorporando las

funciones de supervisión que se seleccionen en el submenú correspondiente.

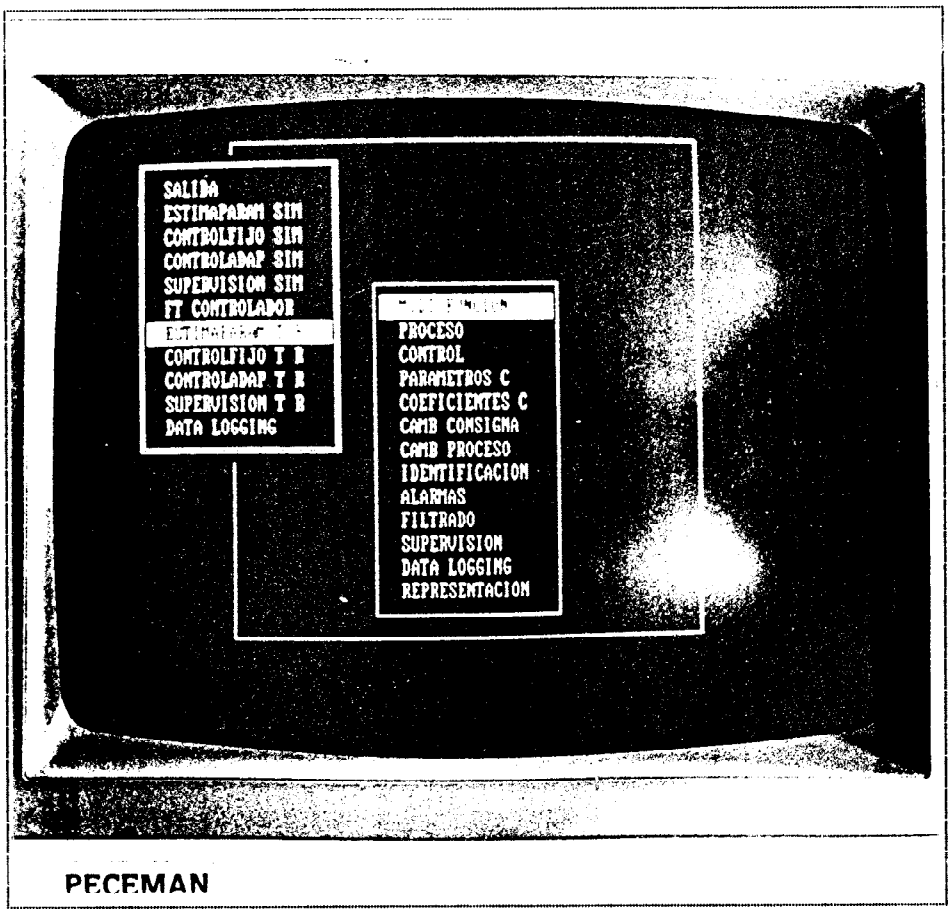


Fig 4.59. Submenú de modo de funcionamiento.

```
# ejecucion
# relativa
escribe k
50
esc relat
1

# consigna
#=0 en k=60
esc k
10
e w
0.0

#lee w k=80
e k
20
lee w

# fin ejec.
# relativa
esc k
20
esc r
0

# haz w=1
e k
120
e w
1.0

#haz w=0
e k
140
escribe w
0.0

# fin batch
batch
```

Fig 4.60. Fichero con órdenes para trabajo en modo batch.

Otras opciones que es posible seleccionar son:

SALIDA: Fin de la ejecución y retorno al sistema operativo.

FT REGULADOR: Obtención de la función de transferencia del controlador, con el método seleccionado.

DATA LOGGING: El sistema funcionará en modo adquisición de datos, con las opciones seleccionadas en el submenú correspondiente.

4.3.1.3 Submenú de selección de proceso.

Será posible la selección de la función de transferencia del proceso con la que se va a efectuar la simulación, de entre las incluidas en los ficheros .DAT almacenados en el subdirectorio de procesos del disco.

El formato de cada fichero, para la representación de la función de transferencia $z^{-d}\ B(z^{-1})\ /\ A(z^{-1})$ será el siguiente:

d m b(1)... b(m) a(1)... a(m) (4.34)

siendo m el orden del proceso, b(0)=0 y a(0)=1. De esta manera es sencillo incorporar, mediante editor de texto, una nueva función de transferencia de un proceso para su simulación.

4.3.1.4 Submenú de control.

FIX: Control con el controlador cuya función de transferencia se especifica en el submenú de Coeficientes del Controlador.

LRP: Control con el método de Control Pedictivo Extendido.

DBN: Control con el método de Tiempo Finito de orden normal.

DBA: Control con el método de Tiempo Finito de orden aumentado.

APN: Control con el método de Asignación de Polos de estructura simple.

APA: Control con el método de Asignación de Polos de estructura completa.

APR: Control con el método de Asignación de Polos robusto.

4.3.1.5 Submenú de parámetros de control.

Es posible la selección de las siguientes opciones, dependiendo del método de control elegido:

FIX: Ninguna.

LRP: PARAM [1] = l (horizonte de previsión).

PARAM [2] = D (modelado o no de perturbación determinista).

PARAM [3] = β (coeficiente de factor de peso de la acción de control).

DBN, DBA: Ninguna.

APN: PARAM [2] = I (inclusión o no de integrador en F.T. del controlador).

PARAM [3] ... PARAM [3+2m+d-2+i] (coeficientes del polinomio A_m, con i=1 si se incluye integrador en el controlador e i=0 en caso contrario).

APA: PARAM [2] = I (inclusión o no de integrador).

PARAM [3] ... PARAM [3+m] (coeficientes del polinomio A_m).

PARAM [4+m] ... PARAM [4+2m+d-2+i] (coeficientes de A_0).

APR: PARAM [1] = z (coeficientes de la región de situación de los

PARAM [2] = r^2 polos en cadena cerrada).

PARAM [3] ... PARAM [3+2m+d-2] (coeficientes de A_m).

PARAM [3+2m+d] ... PARAM [3+3m+d] (coeficientes del vector de variaciones de los parámetros del denominador de la F.T. del proceso).

PARAM [4+3m+d] ... PARAM [4+4m+d] (coeficientes del vector de variaciones de los parámetros del numerador de la F.T. del proceso).

4.3.1.6 <u>Submenú de coeficientes del controlador.</u>

En el caso de que en el submenú de control se seleccione la posición FIX o se emplee control adaptativo más supervisor y controlador de respaldo, es necesario que se incluya en este submenú la estructura de regulación a emplear, así como el orden, retardo y coeficientes de la función de transferencia del controlador. Las opciones son:

ESTRUCTURA DE REGULACION: Normal o Tricaja, respondiendo, según la selección, a una de las ecuaciones:

$$P(z^{-1})\ u(k) = z^{-dr}\ Q(z^{-1})\ (w(k) - y(k)) \qquad (4.35)$$

$$R(z^{-1})\ u(k) = z^{-dr}\ T(z^{-1})\ w(k) - z^{-dr}\ S(z^{-1})\ y(k) \qquad (4.36)$$

ORDEN, RETARDO: mr, dr.

q(0) ... q(mr) / s(0) ... s(mr) según la selección (N o T).

p(0) ... p(mr) / r(0) ... r(mr) según la selección (N o T).

— / t(0) ... t(mr) según la selección (N o T).

4.3.1.7 <u>Submenú de consigna.</u>

KMAX: Cuando LIMITE K esta en ON, esta opción marca el número de período de muestreo en que finaliza la ejecución, siempre que dicho

suceso no se adelante mediante comando on-line.

LIMITE K: Dependiendo de la selección que se efectúe aquí, tendrá o no efecto KMAX, de forma que pueda realizarse la estimación de parámetros o el control de forma indefinida, si se desea. La selección de esta opción sólo tiene sentido en los casos de funcionamiento en tiempo real suponiéndose siempre LIMITE en ON en los casos de simulación.

K PERTURBACION DETERMINISTA: Número de período de muestreo en que va a afectar, cuando se trabaja en modo simulación, una perturbación determinista en escalón, a la entrada del proceso.

VALOR PERTURBACION: Valor de la perturbación determinista introducida.

CAMBIOS DE CONSIGNA: Selección del número de cambios de consigna preprogramados que se van a efectuar. Al modificar este valor, automáticamente se adaptan el número de parejas (K CAMBIO, VALOR CONSIGNA), necesarias para especificar cada cambio.

K CAMBIO: Número de período de muestreo en que se modifica, en forma de escalón, el valor de la consigna.

VALOR CONSIGNA: Nuevo valor de la consigna a partir del período de muestreo en que se efectúa el cambio.

4.3.1.8 Submenú de cambios en el proceso.

En este submenú se va a especificar el número de cambios, así como su tipo y valor, a producir en la función de transferencia del proceso durante una simulación dada. Las opciones serán:

NUMERO DE CAMBIOS: Selección del número de cambios en el proceso que se van a efectuar. Al modificar este valor, automáticamente se adaptan el número de líneas (TIPO, K CAMBIO y VALOR), necesarias para especificar cada cambio.

TIPO: Especificación del tipo de cambio (ganancia del proceso (K), coeficiente del numerador (Bi) o denominador (Ai) de la función de transferencia del proceso o tiempo muerto (D) del mismo).

K CAMBIO: Número de período de muestreo en que se modifica el valor de la variable especificada en TIPO.

VALOR: Nuevo valor de la variable especificada a partir del período de muestreo en que se efectúa el cambio.

4.3.1.9 Submenú de identificación.

En este submenú se seleccionan las opciones relacionadas con la estimación de los coeficientes de la función de transferencia del proceso, dependiendo de las distintas fases de funcionamiento. Igualmente se especifican los parámetros para la generación de

PRBS. Las opciones son:

PREIDENTIFICACION: Ejecución o no de la fase de preidentificación, previamente a la identificación o control en tiempo real.

K ESTIMACION PREVIA: Número de período de muestreo en que finaliza la estimación previa de parámetros del proceso, en el caso de control adaptativo, tanto en tiempo real como en simulación. Esta estimación previa es necesaria, para que existan unos valores razonablemente aproximados de los parámetros del proceso, previamente a la obtención, en base a ellos, de la función de transferencia del controlador.

INCREMENTAL: Con este parámetro se selecciona si se desea que el algoritmo de identificación trabaje con las señales medidas U(k) e Y(k), o con las variaciones de las mismas:

$$\delta u(k) = u(k) - u(k-1) = U(k) - U(k-1) \tag{4.37}$$

$$\delta y(k) = y(k) - y(k-1) = Y(k) - Y(k-1) \tag{4.38}$$

BIT DEL REGISTRO: Selección del número total de bits del registro de desplazamiento.

BIT NUMERO: Selección de cada uno de los bits del registro que servirán como punto de realimentación con los que se efectuará la función OR EXCLUSIVA para obtener el nuevo valor del primer bit del registro.

MULTIPLICADOR: La salida del generador de PRBS vendrá en función de este parámetro, así como del último bit del registro de desplazamiento usado. Si este es 1, la salida será igual a MULTIPLICADOR, sino será -MULTIPLICADOR.

4.3.1.10 Submenú de alarmas.

MANUAL: Funcionamiento del sistema en modo Manual.

TRANSICION: Selección del modo de cambio, si éste se produce en funcionamiento on-line, de Manual a Automático.

UMIN, UMAX: Límites máximos (inferior y superior) para efectuar la detección de la saturación del actuador.

ANTI RESET: Opción para que el método de control tenga o no en cuenta la posible saturación del actuador.

CANAL ANALOGICO: Selección del número de canal analógico de entrada que se comprobará en cada período de muestreo.

ALARMA INFERIOR, ALARMA SUPERIOR: Límites inferior y superior para la señal análogica procedente del canal seleccionado.

CANAL DIGITAL: Selección del número de canal digital de entrada que se comprobará en cada período de muestreo.

REGISTRO: Almacenamiento o no en disco de la notificación de las situaciones anómalas, o cambio de modo de funcionamiento on-line, producidas.

ALARMA i: Definición de alarmas, indicando qué señal (saturación del actuador S, detección de outlier O, superación límite analógico A, canal digital en ON D, controlador de respaldo activo C, u otras que se definan previamente por software) se monitorizará en cada alarma, teniendo en cuenta que la señalización por hardware sólo afecta a las tres primeras.

4.3.1.11 Submenú de filtrado.

Se seleccionan en este submenú las opciones relacionadas con el filtrado, analógico o digital, de perturbaciones, así como la detección y eliminación de outliers o errores de medidas cuando el sistema está funcionando en tiempo real. Las opciones son:

FILTRADO ANALOGICO: Se selecciona aquí si se desea que la señal de salida del sistema físico implementado se lleve al filtro de Butterworth de segundo orden diseñado según se indica en el apartado 4.4.7.1.

T FIJO: Se indica aquí si el sistema va a funcionar con un período de muestreo fijo (en combinación con el filtro analógico implementado, para evitar el efecto aliasing), o con el que figura en el

fichero datos.pre. En el primer caso se aplica posteriormente a la señal muestreada un filtro digital.

VALOR T: Se selecciona el valor del período de muestreo fijo, en el caso de la primera elección de la opción anterior. No es normal variar este valor de T=10 (en ms*10) a no ser que se varíe el filtro analógico implementado.

FILTRADO DIGITAL: Se selecciona aquí si se desea la inclusión en el sistema de un filtro digital, para la eliminación del ruido de baja frecuencia.

ORDEN DEL FILTRO: Selección del orden del filtro digital a emplear.

FRECUENCIA DE CORTE: Selección del valor de la frecuencia de corte para el filtro digital implementado.

FILTRADO DE OUTLIERS: Selección de la inclusión o no del filtro de outliers.

K FILTRO DE OUTLIERS: Selección del número de período de muestreo en que se activa el filtro, una vez que tenga suficiente información para la estimación adecuada de $\bar{y}$ y σ_y antes de su uso en el filtro.

VALOR MINIMO: La máxima variación respecto de la media de una señal, para ser o no considerada outlier, se acomodará en tiempo

real a las variaciones en la señal medida. El tope inferior para la misma será VALOR MINIMO $\star$ $\hat{\sigma}_y$.

4.3.1.12 Submenú de supervisión.

Se seleccionará mediante este submenú las opciones correspondientes a la supervisión de la estimación de parámetros, la síntesis del controlador y el comportamiento en bucle cerrado, cuando se seleccionan los modos de funcionamiento de control adaptativo más supervisión o estimación de parámetros, mediante:

TIPO OLVIDO: Se selecciona aquí si se desea que el método de identificación trabaje con un factor de olvido constante para todo período de muestreo, variable en función de la excitación existente según el método de Saelid y Foss, variable según este mismo método modificado según el apartado 4.2.6.2 o variable según la expresión:

$$\tau(k+1) = \tau_0\ \tau(k) + \tau(1-\tau_0) \qquad (4.39)$$

FACTOR OLVIDO INICIAL: $\tau(0) \leq 1$ en la fórmula anterior.

FACTOR OLVIDO FINAL: $\tau \leq 1$ en la fórmula anterior. Es el valor deseado para $\tau(k)$ una vez completa la estimación previa.

MULTIPLICADOR: $\tau_0 \leq 1$ en la fórmula anterior.

COVARIANZA INICIAL: Valor inicial para los coeficientes de la

diagonal de la matriz de covarianzas.

COVARIANZA LIMITE: Valor límite para los valores de la diagonal de la matriz de covarianzas, según el método de Saelid y Foss modificado, después de la inicialización (sólo válido en simulación).

DETECCION T: Se selecciona aquí si se activa o no, en cada período de muestreo, el mecanismo de supervisión que detecta los cambios que se produzcan en los parámetros del proceso.

DETECCION D: Se selecciona aquí si se activa o no, en cada período de muestreo, el mecanismo de supervisión que detecta los cambios que se produzcan en el tiempo muerto del proceso.

K MINIMO DETECCION: Parámetro que da la separación mínima entre cambios en el proceso. Cuanto mayor sea su valor, mayor será la fiablidad de las detecciones.

CONTROLADOR DE RESPALDO: Selección de si se desea o no que el controlador especificado por la función de transferencia mediante el submenú de coeficientes del controlador, actúe o no en el arranque del control adaptativo o cuando ocurra algún cambio en el proceso (si está seleccionado DETECCION T).

DETECCION I: Se selecciona aquí si se activa o no, en cada período de muestreo, el mecanismo de supervisión que detecta la existencia de polos del proceso fuera del círculo unidad que pudieran dar

lugar a cancelaciones incorrectas, así como activa el controlador de respaldo si esto sucede.

DETECCION E: Se selecciona aquí si se activa o no, en cada período de muestreo, el mecanismo de supervisión que detecta saturaciones continuadas en el actuador o errores de control superiores a un cierto límite, activando el controlador de respaldo.

ERROR MAXIMO: Límite superior admisible para el error de control, cuando está activado el mecanismo anterior.

VECES ERROR: Número de veces que ha de superarse el error máximo o encontrarse saturado el actuador para activar el controlador de respaldo.

VECES C RESPALDO: Número de veces que se activa el controlador de respaldo en un número dado de períodos de muestreo, antes de que se modifique el método de control o se pase a modo Manual.

ROBUSTO: Se selecciona aquí si se desea o no que el estimador incorpore una zona muerta en función del error de estimación.

VALOR: Definición de los límites de la zona muerta.

4.3.1.13 Submenú de adquisición de datos.

CANAL INFERIOR, CANAL SUPERIOR: Selección de los límites inferior y

superior, para el número del canal análogico de entrada, entre los cuales se llevará a cabo la adquisición de datos.

NUMERO DE DATOS: Selección del número de datos a introducir, por cada canal y período de adquisición.

VALOR T ADQUISICION: Selección del período de adquisición de datos, en µs * 10.

VALOR T CONVERSION: Selección del tiempo, en µs * 10, entre dos conversiones A/D sucesivas.

VISUALIZACION: Selección de si se desea o no que se muestren los datos procedentes de la adquisición por pantalla.

4.3.1.14 Submenú de representación.

NUMERICA/GRAFICA: Selección del modo de representación.

REGISTRO: Selección de si se desea obtener o no copia en papel de la representación (numérica o gráfica) que se visualiza en pantalla.

GRAFICA 1, LIMITE INFERIOR 1, LIMITE SUPERIOR 1: Selección de la variable a representar, así como de los límites inferior y superior de representación en la primera gráfica.

GRAFICA 2, LIMITE INFERIOR 2, LIMITE SUPERIOR 2: Selección análoga en la segunda gráfica.

COLUMNAS 2 .. 10: Selección de las variables a representar en las columnas de la salida numérica (en la primera se representa siempre el período de muestreo), de entre consigna, parámetros estimados del proceso, tiempo muerto estimado, error de estimación, coeficientes de la matriz de covarianzas, componentes del vector de factores de olvido, variable de control, variable controlada, coeficientes del numerador y denominador de la función de transferencia del controlador u otras que se definan previamente por software.

Cuando el modo de funcionamiento para el sistema es el de cálculo de la función de transferencia del controlador, independientemente de las elecciones hechas en este submenú, la representación será numérica y sólo de las columnas q y p. En caso de control adaptativo no se representarán dichas columnas.

4.3.2 Intérprete de comandos.

Este módulo se encarga de la recepción y evaluación de los comandos emitidos por el usuario, mientras el sistema MASTER trabaja en tiempo real (estimación de parámetros o control). La representación de los comandos y respuestas del intérprete se realiza en un pequeño número de líneas, al representarse en el resto de la pantalla los resultados que se van produciendo simultáneamente.

El intérprete puede funcionar en dos modos: interactivo y batch. En el primero de ellos los comandos son emitidos por el usuario on-line, mientras que en el segundo proceden del fichero datos.int, completado previamente, como el que se muestra en la figura 4.60.

El instante de ejecución de cada comando podrá especificarse en forma absoluta o relativa (a partir de la ejecución del último comando), con el comando Relativo ejecutado en forma interactiva o batch.

Los comandos que acepta el intérprete son los siguientes:

- Quit: Se finaliza el modo de funcionamiento en curso, pudiendo retornar al sistema operativo o iniciar una nueva ejecución.

- Alarmas: Se define el modo de indicación de alarma por pantalla (ON/OFF), no afectando a la indicación hardware ni al registro en fichero de las mismas.

- Reset: Inicialización de las indicaciones hardware de alarma.

- Batch: Definición del modo de ejecución de comandos en batch (ON/OFF).

- Manual: Definición del modo de control (Manual ON/OFF).

- +, -: Incremento o decremento de la acción de control en modo Manual.

- *, /: Modificación (±100%) del valor de incremento y decremento de la acción de control en modo Manual.

- Lee, Escribe: Visualización o modificación de un subconjunto de los parámetros del sistema de control, de los que pueden ser seleccionados en la inicialización mediante menús. Son los siguientes:

  - Analógico (L,E). Activa o desactiva el filtrado analógico.

  - Digital (L,E). Activa o desactiva el filtrado digital.

  - Outlier (L,E). Activa o desactiva el filtrado de outliers.

  - W (L,E). Lee o modifica el valor de la variable de consigna.

  - Y (L,E). Lee o modifica el valor de la variable de salida.

  - Parámetro β (L,E). Lee o modifica el valor del parámetro β del método de control LRP cuando éste está seleccionado.

  - Coeficientes Am(1), Am(2) (L,E). Lee o modifica el valor de los coeficientes (con $A_m(0)=1.0$) de un polinomio característico con dos polos significativos y el resto en el origen, cuando

están seleccionados los métodos de control APN o APA.

- Región z (L,E). Lee o modifica el valor del parámetro z del método de control APR cuando éste está seleccionado.

- Incremento (L,E). Lee o modifica el valor de incremento de la acción de control cuando el sistema funcione posteriormente en modo Manual.

- Relativo (L,E). Lee o modifica el valor de la variable relativo, que afecta a la ejecución de comandos en modo Batch.

- K (E). Admite el número del período de muestreo en que se ejecutará el comando que se especifique a continuación.

- U (L). Lectura del valor de la variable de control, en el período de muestreo actual.

- E (L). Lectura del valor de la variable de error en la estimación de parámetros, en el período de muestreo actual.

4.3.3 Gestión de Alarmas.

En este módulo se realiza la gestión de anomalías, con el sistema en alguno de los modos de funcionamiento en tiempo real.

Las señales que se monitorizan pueden ser analógicas o digitales.

También se incluyen como alarmas variables del sistema de control adaptativo. En el primer caso, las señales se introducen por cualquiera de los canales de entrada analógicos (que van al conversor A/D) o digitales. La selección de dichos canales puede realizarse mediante menú.

En este módulo se han implementado cuatro procedimientos: Borra, Reset, HardAlarm y Alarma. En el primero se realiza la inicialización de las variables usadas como alarma así como de las indicaciones que se realizan al usuario, previamente al funcionamiento del sistema on-line.

El procedimiento Reset se puede activar mediante un comando del usuario durante el funcionamiento del sistema on-line y sirve para inicializar las indicaciones de alarma que se han presentado. El procedimiento Alarma está concebido para dar respuesta a varios aspectos del tratamiento de situaciones anómalas como son:

- Detección: En el caso de variables del sistema de control, se realiza con ayuda de los procedimientos involucrados de otros módulos. Por ejemplo en el caso de saturación del actuador o detección de outliers, con los procedimientos SAT y FOutlier. En el caso de señales procedentes del sistema físico conectado al computador, analógicas o digitales, mediante el procedimiento HardAlarm del propio módulo ALA.

- Aviso: Se realiza mediante dos métodos. El primero se lleva a

cabo conectando un sistema de señalización a cada una de las líneas de salida digital existentes y gobernándolas en función de lo obtenido en la fase de detección. El segundo método consiste en un mensaje visual por pantalla de la anomalía producida.

- Reacción: En el caso de variables del sistema de control, se realiza con los mismos procedimientos involucrados en la detección. En el caso de que señales analógicas sobrepasen los límites preestablecidos o se activen señales digitales, la reacción debe ser propia de cada sistema físico a controlar. En el procedimiento HardAlarm se ha implementado una situación particular para reacción automática. Así mismo el usuario, mediante comandos on-line o bien poniendo el sistema en modo Manual puede realizar las acciones oportunas en función de la anomalía producida.

  En la figura 4.61 se observa la reacción implementada, cuando, a partir del cambio de consigna en k=80, la variable analógica monitorizada intenta superar el valor 3.0 seleccionado mediante el submenú de definición de alarmas. La reacción consiste en este caso en reducir en un 5% el valor de la consigna en cada detección de superación del límite superior establecido.

- Registro: Si se selecciona por parte del usuario mediante menú, se activa la función de registro automático en un fichero de disco virtual, los mensajes de alarma que envía el sistema, aún

en el caso en que la aparición de los mismos por pantalla haya sido inhibida por parte del usuario mediante comando on-line.

La definición de las anomalías que van a ser tratadas, así como de los límites inferiores y superiores para variables numéricas o señales analógicas del sistema físico, podrá ser realizada mediante menú, previamente al funcionamiento del sistema en tiempo real. Así mismo podrán ser seleccionados los canales de entrada analógicos o digitales para las señales a comprobar, así como para qué alarmas se utilizará aviso mediante sistema de señalización (por ejemplo led) conectado a salida digital y en cuáles sólo indicación por pantalla. También se podrá seleccionar si se desea registro o no en fichero de todas las producidas.

En la figura 4.62 se muestra un ejemplo de fichero de salida (datos.ala) obtenido durante el funcionamiento del sistema en tiempo real, cuando se producen varias situaciones anómalas, cuya definición se ha efectuado previamente mediante menú, así como cambios en el modo de funcionamiento del sistema.

4.3.4 Control manual.

Es posible entrar en el modo de funcionamiento Manual del sistema, bien directamente seleccionando la opción correspondiente en modo Menú o bien por comando on-line en cualquier instante.

En modo Manual será posible ajustar mediante comando on-line el

valor de la variable de control, pero únicamente para incrementarla (+) o decrementarla (-), de forma que se simule una actuación con pulsadores, más apropiada que con comando en determinadas aplicaciones industriales.

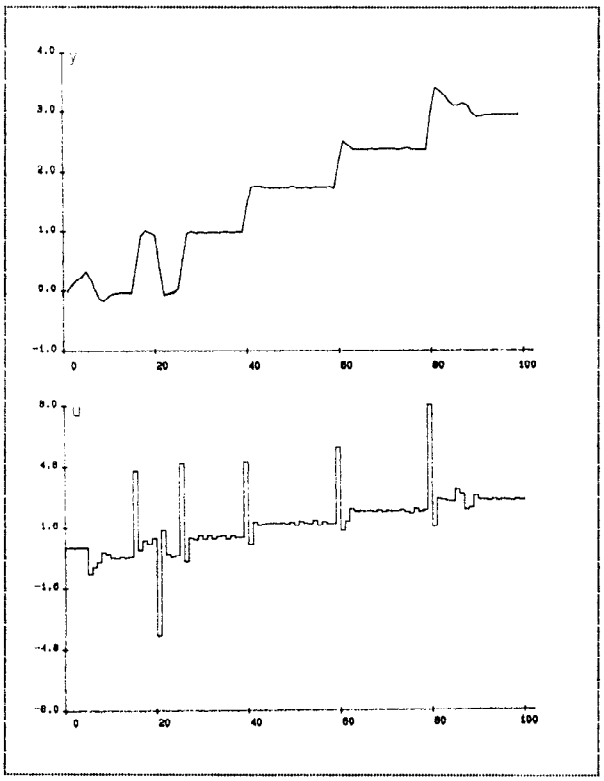


Fig 4.61. Reacción ante alarma.

k	
44	ALARMA NUMERO2
77	Modo Manual ON
147	Modo Manual OFF
164	ALARMA NUMERO4
173	ALARMA NUMERO4
213	Modo Batch ON
265	Modo Batch OFF
393	ALARMA NUMERO2
452	ALARMA NUMERO2
475	ALARMA NUMERO2
476	ALARMA NUMERO1
481	ALARMA NUMERO1
482	ALARMA NUMERO1
482	ALARMA NUMERO2
483	ALARMA NUMERO2
629	Filtro analogico ON
654	Filtro analogico OFF

Fig 4.62. Fichero con registro de situaciones anómalas producidas.

La velocidad de la modificación de la variable de control podrá ser ajustada mediante comando con el sistema en modo Automático (Manual OFF). En modo Manual únicamente podrá ser variada con los comandos * y /, multiplicándola o dividiéndola respectivamente por 2.

El cambio de modo Manual a Automático puede ser problemático en muchos casos, ocurriendo transiciones indeseables si no se realiza

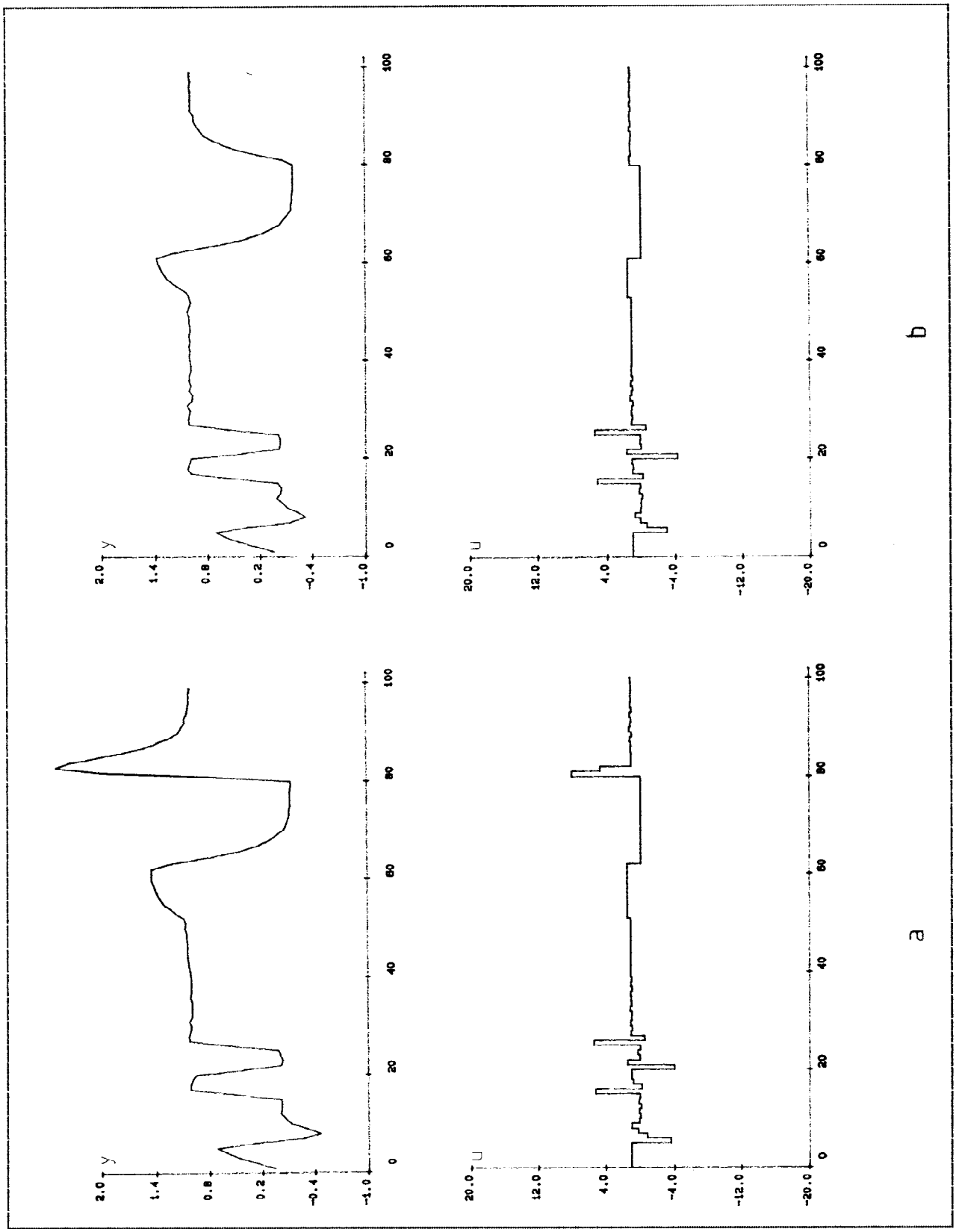


Fig 4.63. Cambio de modo manual a automático.

un correcto seguimiento de la acción de control. Si se selecciona TRANSICION en ON en el submenú correspondiente, se realizará este seguimiento automáticamente, haciendo que la señal de control, calculada en base a las estimaciones de los parámetros del proceso, sea igual a la de entrada Manual, y comprobando la posible saturación del actuador.

En la figura 4.63 se puede comprobar la diferencia en las transiciones, realizando (b) o no (a) el seguimiento indicado, cuando se cambia el modo de Manual a Automático en k=80 (previamente se realizó el cambio opuesto en k=40).

4.4 INTERFACE CON EL PROCESO.

4.4.1 Tarjeta de Interface DASH-16F.

4.4.1.1 Diagrama de bloques.

La tarjeta DASH-16F, mostrada en la figura 4.64, es una tarjeta multifunción de MetraByte para IBM PC o compatible. Las funciones implementadas son las siguientes:

- Conversión A/D. El conversor A/D es HI-774, de aproximaciones sucesivas, tiempo de conversión 8 µs y 12 bits de resolución. La configuración de los canales de entrada se selecciona a 8 diferenciales. Las entradas están multiplexadas y el rango de la tensión en las mismas se selecciona a +/- 10 V. bipolar.

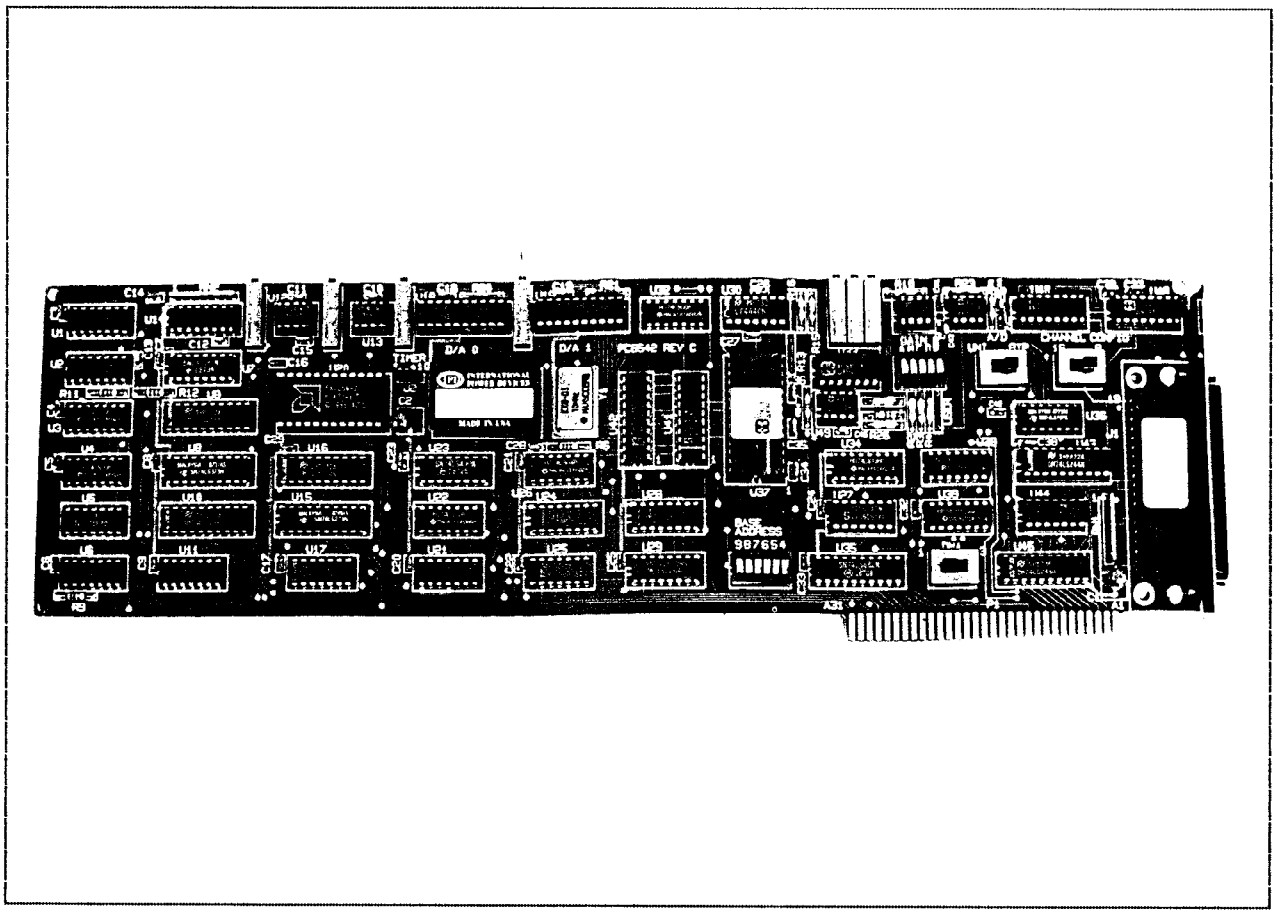


Fig 4.64. Tarjeta DASH-16F.

Las conversiones A/D pueden ser iniciadas por tres métodos: comando software, temporizador programable interno o disparo externo. La transferencia de los datos procedentes de la conversión puede llevarse a cabo por tres caminos: comprobación del final de conversión por programa, interrupción o DMA.

- Conversión D/A. Puede llevarse a cabo a través de cualquiera de los dos conversores AD-7548 existentes. Son conversores multiplicadores de 12 bits con un tiempo de establecimiento de

la salida de 30 µs. La referencia se conecta a una tensión de -10V para tener un rango de salida de 0 - +10V.

- Temporización. Puede llevarse a cabo a través del temporizador programable 8254 de tres canales.

- E/S Digital. Se lleva a cabo a través de un registro de entrada y otro de salida digital, compatible TTL, de 4 bits.

4.4.1.2 Programación.

La tarjeta DASH-16F usa 16 direcciones consecutivas en el mapa de entrada/salida, a partir de la dirección base que se selecciona por hardware. A través de esas direcciones los procedimientos programados acceden a:

- Registros de datos procedentes de la conversión A/D.
- Registro de selección de canal.
- Registro de datos para E/S digital.
- Registros de datos para la conversión D/A.
- Registro de estado.
- Registro de control.
- Registro de habilitación para temporización.
- Registros del temporizador.

4.4.1.3 Disparo periódico del conversor A/D.

El primero de los modos de inicio de conversión A/D, por comando software, así como la obtención del dato procedente de la misma una vez comprobado el bit EOC (End Of Conversion) del registro de estado, se utilizan únicamente en la fase de Preidentificación, así como para la lectura de canales auxiliares de entrada analógica.

Durante el funcionamiento del sistema en tiempo real se programa el registro de control de modo que la fuente de pulsos de inicio de conversión sea la salida del contador 2.

Como se ve en la figura 4.65, la salida del contador 1 está conectada a la entrada del contador 2. Conectando el "clock jumper" en la posición x1 se tendrá 1 MHz a la entrada del contador 1. Dejando el valor del contador 1 fijo igual a 2000H (2^{13}) y variando el de contador 2 en función del período de muestreo T_m (2^1 - 2^{16}), se tendrá un rango de variación para dividir la frecuencia inicial de $2^{14} \div 2^{29}$, o sea, se podrá programar T_m en el rango 16.4 ms $\leq T_m \leq$ 536870 ms.

La conversión A/D se disparará así cada T_m y, al finalizar la misma, se producirá una interrupción, traspasándose el control del programa a su rutina de servicio, en la que se realizará la lectura del dato procedente de la conversión y su utilización en el modo de funcionamiento seleccionado previamente por el usuario vía menú.

4.4.1.4 <u>Disparo periódico con DMA.</u>

El Acceso Directo a Memoria proporciona una transferencia directa de datos entre un dispositivo de E/S y memoria, sin que la CPU tenga que acceder previamente a los mismos. Esto ocasiona un gran incremento en la velocidad de transferencia de datos, lo que hace posible disminuir considerablemente el tiempo necesario entre cada disparo del conversor A/D, hasta aproximarse al tiempo máximo de conversión (8.5 µs para el Harris HI-774).

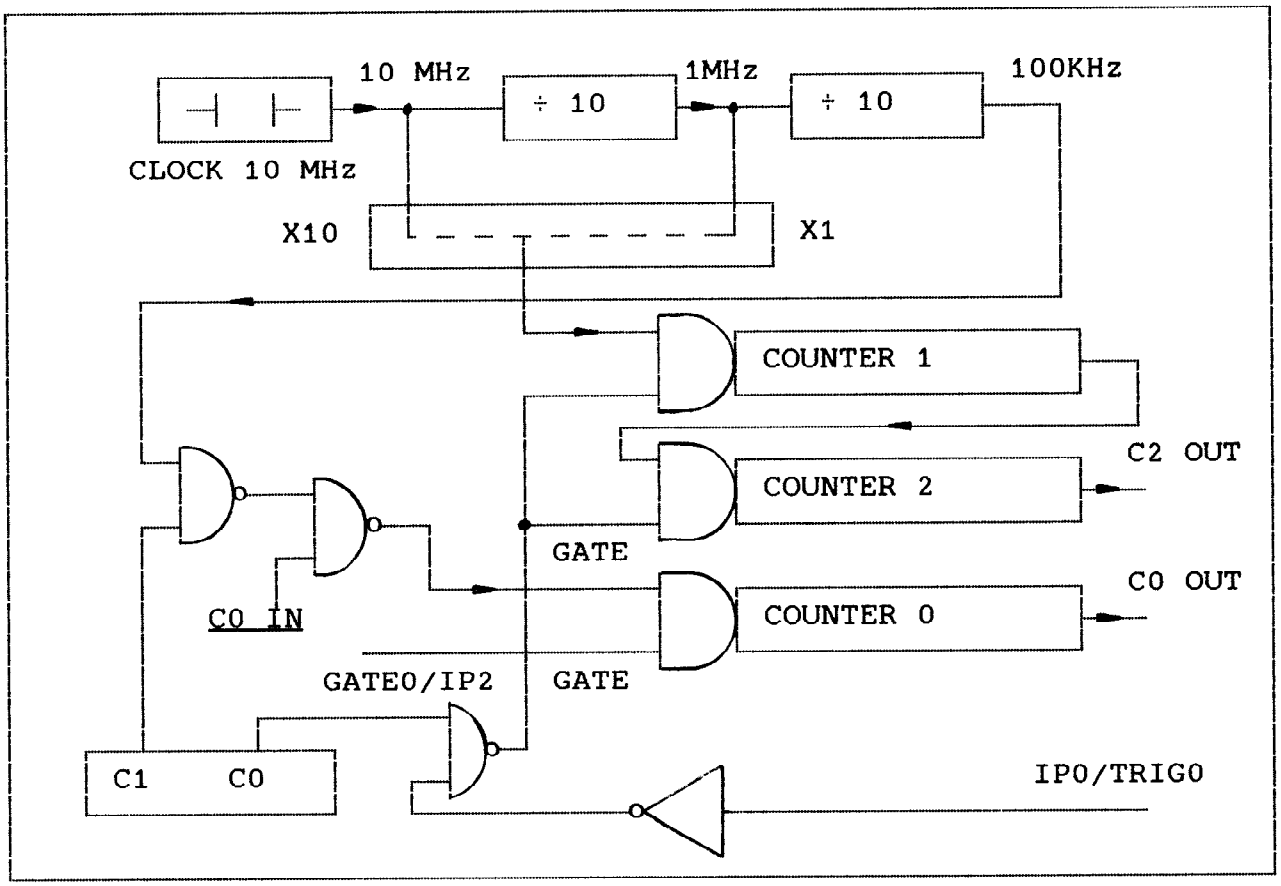


Fig 4.65. Configuración del temporizador programable.

La salida del canal 2 del temporizador seguirá siendo la fuente de pulsos de disparo del conversor A/D, emitiendo éste, al finalizar cada conversión, dos DMA Request (DRQ) para, una vez recibido cada

DMA Acknowledge (DACK), transferir los dos bytes de datos procedentes de cada conversión a la dirección de memoria puesta en el bus de direcciones por el controlador de DMA. Una vez que éste completa la operación de transferencia del número de datos con que se ha programado, emite un Terminal Count (T/C), generándose una interrupción.

El canal 0 del temporizador es usado, programando correctamente C0, como habilitación de conteo de los canales 1 y 2, de forma que podrá producir, después de cada interrupción, la espera seleccionada antes de iniciar el siguiente bloque de conversiones. Como se puede ver en la figura 4.66, la fuente de pulsos de reloj de entrada, programando correctamente C1, será de 100 KHz, con lo que, con un conteo decimal, con el contador 0 se podrán producir períodos de adquisición de datos de hasta 100 ms.

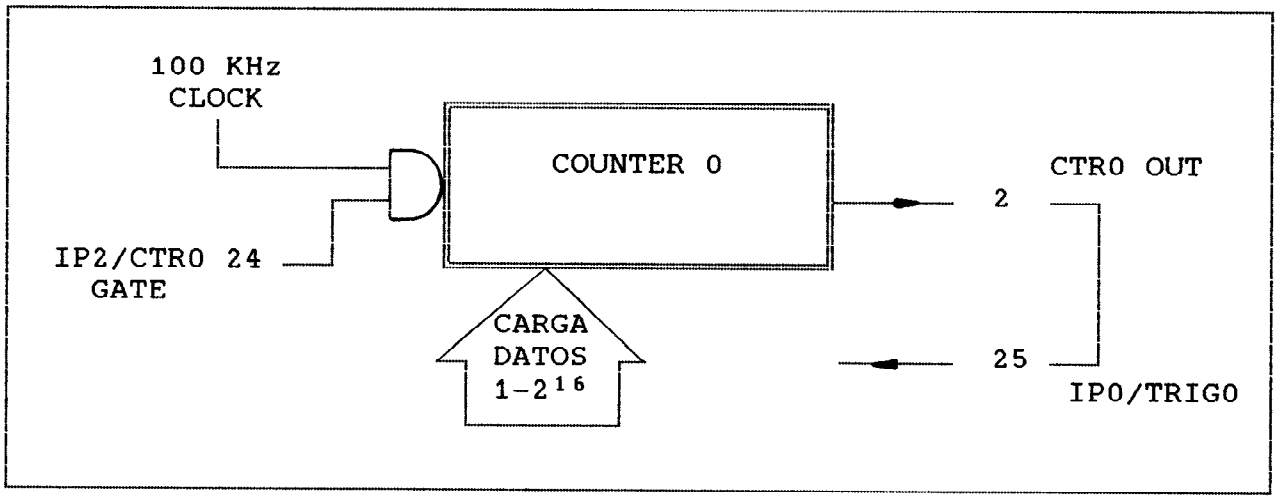


Fig 4.66. Uso del contador 0.

El controlador de DMA usado en el IBM PC o compatible es el chip de

Intel 8237A-5, que dispone de cuatro canales para transferencia de datos. El canal 1 no es usado en el funcionamiento normal del computador, por lo que será el empleado por el sistema MASTER.

4.4.2 Hardware implementado.

La placa de pruebas desarrollada se muestra en la figura 4.67, mientras que en la 4.68 se puede ver un diagrama de bloques de la misma conectada a la tarjeta DASH-16F, consistiendo básicamente en:

- Dos sistemas de primer orden conectados en serie, de forma que se tiene accesible la salida de ambos y, seleccionando la segunda, se dispone de un sistema de orden 2.

- Fácil posibilidad de cambio, mediante el potenciómetro P1, de los coeficientes de la función de transferencia en ambos casos.

- Filtro de Butterworth de segundo orden.

- Entrada para la señal procedente de un proceso externo, de forma que se pueda aplicar a la misma el filtro implementado, así como el sistema de señalización, etc. de la tarjeta.

- Sistema de leds para señalización de alarmas y otras indicaciones (Fuente de alimentación ON/OFF, Filtro analógico ON/OFF, Alarmas números 1, 2 y 3). Así mismo se dispone de un sistema de generación de señal digital, que se lleva al puerto de E/S.

- Sistema de introducción de señales de frecuencia variable, así como de continua, para estudio del comportamiento del sistema desarrollado ante perturbaciones.

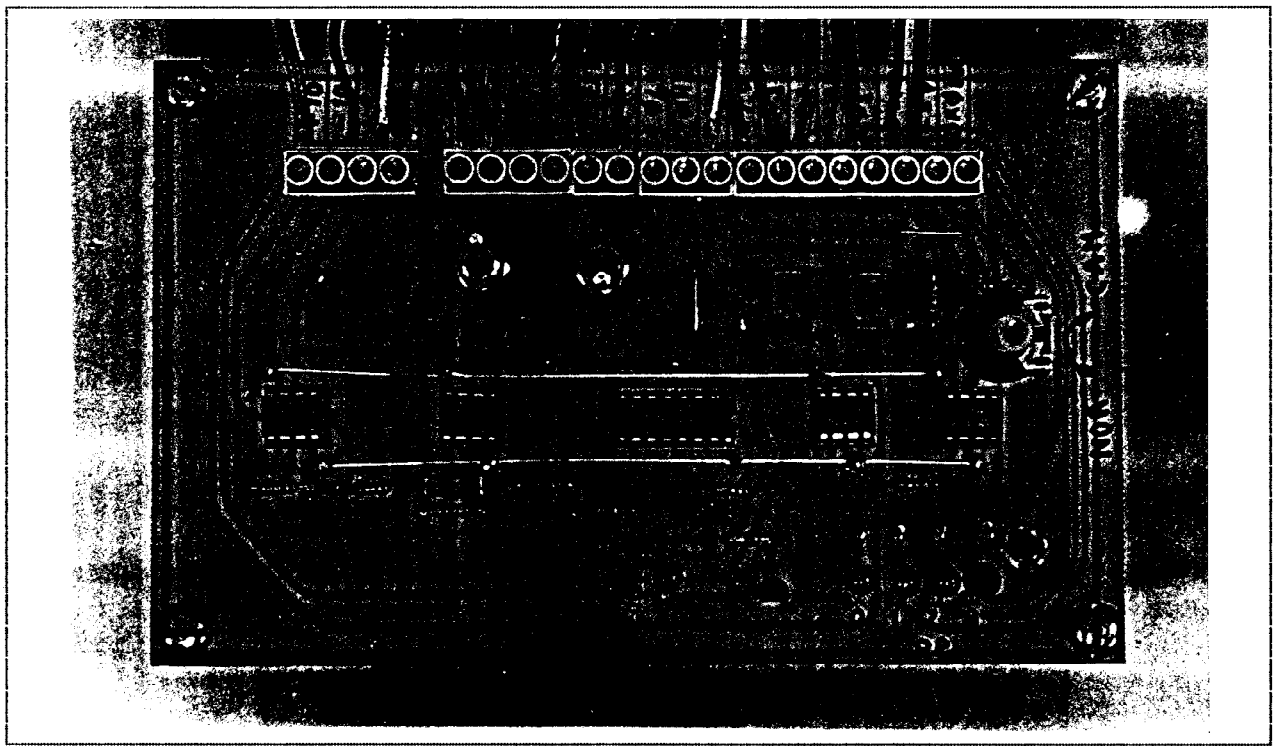


Fig 4.67. Placa de pruebas.

Como se puede comprobar, son seleccionables por software las diversas opciones de funcionamiento (filtrado analógico ON/OFF, funcionamiento o no del canal 0 del temporizador como habilitación del 1 y 2, selección del período de muestreo, etc.), sin que ello requiera ninguna modificación en el hardware.

4.4.3 Conversión Analógico/Digital.

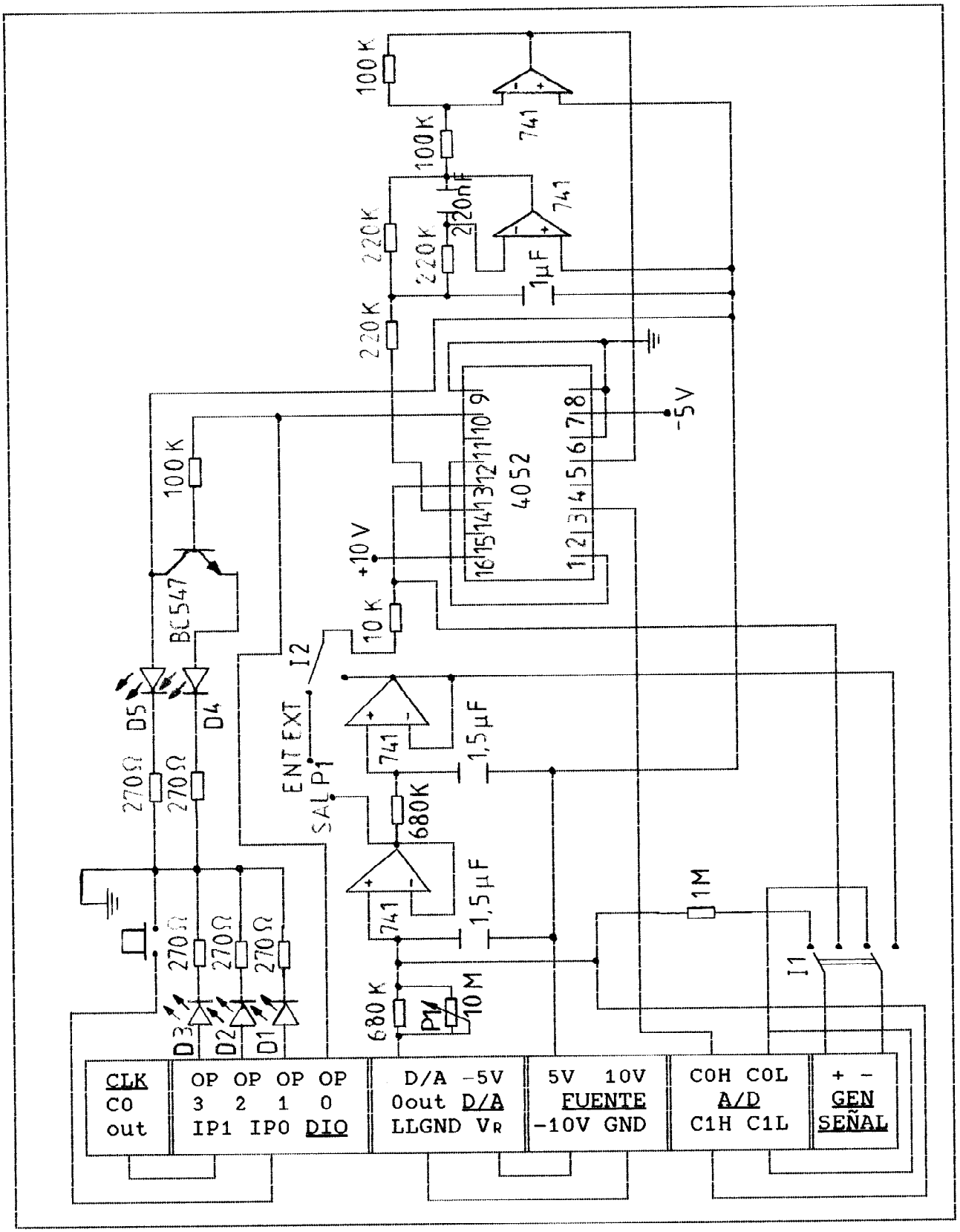


Fig 4.68. Diagrama de bloques del hardware implementado.

Este módulo es un driver del conversor A/D de la tarjeta DASH-16F de Metrabyte, que gestiona la obtención de datos procedentes del mismo al efectuarse cada conversión. Las conversiones podrán dispararse por software o periódicamente por hardware, tomando como fuente de pulsos de disparo una de las salidas del temporizador incluido en la tarjeta. Así mismo, la obtención del dato de la conversión podrá obtenerse comprobando por software el final de la misma, por interrupción o vía DMA.

Durante el funcionamiento en tiempo real del sistema, las conversiones A/D son gestionadas por los procedimientos:

- IADC: Inicializa los registros necesarios de la tarjeta, habilitando interrupciones, que se producirán cuando haya dato válido procedente de la conversión.

- StartADC: Crea el proceso para el driver de interrupciones, lo asocia al vector de interrupción y le transfiere el control.

- DriverADC: Gestiona las interrupciones producidas, obteniendo el dato procedente de la conversión y llamando al procedimiento Interrupción para que se ejecute el ciclo de control o estimación.

Con el sistema funcionando en modo Data Logging, las conversiones A/D son gestionadas por los mismos procedimientos anteriores, teniendo en cuenta que cuando se produzca una interrupción (ahora

al finalizar cada transferencia de bloque de datos), se llamará al procedimiento IntData del módulo DAT. Otros procedimientos necesarios en este modo de funcionamiento son:

- Dirección: Obtiene los bytes con que será necesario efectuar la programación del registro base de dirección del controlador de DMA, así como el registro de página para el canal 1 del mismo.

- ResetADC,IntADC: Inicialización del funcionamiento en modo DMA.

En la fase de preidentificación y en tiempo real para canales auxiliares, las conversiones son gestionadas por el procedimiento DatosADC, que inicializa los registros necesarios en la tarjeta para lanzar la conversión por software, así como introduce los datos procedentes del CA/D al finalizar la misma, sin usar interrupciones. Para la obtención del orden del modelo en preidentificación, se usará el mismo mecanismo que para tiempo real, llamándose ahora a OrdenMod cada vez que se produzca una interrupción.

4.4.4 Conversión Digital/Análogica.

Este módulo es un driver del conversor D/A de la tarjeta DASH-16F de Metrabyte, que realiza la introducción de un dato digital de 12 bits en los registros del conversor D/A para su paso a una magnitud análogica.

4.4.5 Entrada/Salida digital.

Este módulo es un driver de los registros para entrada/salida digital de la tarjeta DASH-16F de Metrabyte, que incluye dos procedimientos: InputDIO y OutputDIO. El primero de ellos lee el contenido del registro de entrada (4 bits) y lo almacena en la variable estent, mientras el segundo manda el contenido de la variable estsal al de salida (4 bits).

4.4.6 Control del temporizador.

Este módulo es un driver del temporizador programable de la tarjeta DASH-16F, que realiza tres funciones:

- Medición de tiempos, mediante los procedimientos Reset (que inicializa los registros de los contadores 1 y 2 del temporizador) y Lectura (que lee el contenido de los mismos y obtiene restando el tiempo transcurrido desde que se ejecutó Reset).

- Generación de pulsos de disparo de la conversión A/D para el funcionamiento del sistema en tiempo real, de forma que se obtenga un muestreo periódico.

- Generación de pulsos de disparo de la conversión A/D, así como del período de adquisición con el canal 0 del temporizador, para el funcionamiento del sistema en modo Data Logging. Para estas dos últimas funciones se emplea el procedimiento Program.

4.4.7 Filtrado de perturbaciones y outliers.

El sistema desarrollado incluye los filtros adecuados para la eliminación de las perturbaciones que puedan actuar sobre el proceso, interferencias en la transmisión de la señal medida, etc. así como la detección y eliminación de medidas erróneas debidas a funcionamiento defectuoso de los sensores, errores de conversión, etc. Este módulo contiene los procedimientos necesarios para que, dependiendo de las opciones elegidas por el usuario, se realice el filtrado de perturbaciones y outliers.

4.4.7.1 Filtrado de perturbaciones.

En la figura 4.69 (a) se puede comprobar el efecto aliasing producido por la influencia de un ruido de frecuencia f_r=1.2 Hz actuando sobre un sistema de segundo orden desde k=60. El período de muestreo es T_m=500 ms. Como se ve, la frecuencia del ruido está comprendida entre la frecuencia de muestreo f_m y la de Nyquist f_N, por lo que la componente generada después del muestreo tendrá por frecuencia f_2:

f_m=2 Hz f_N=1 Hz f_r=1.2 Hz $f_2=f_m-f_r$=0.8 Hz.

En la figura 4.69 (b) se comprueba el efecto, sobre la señal de salida, de un ruido de frecuencia f_r=1.7 Hz. En este caso se tendrá f_2=0.3 Hz.

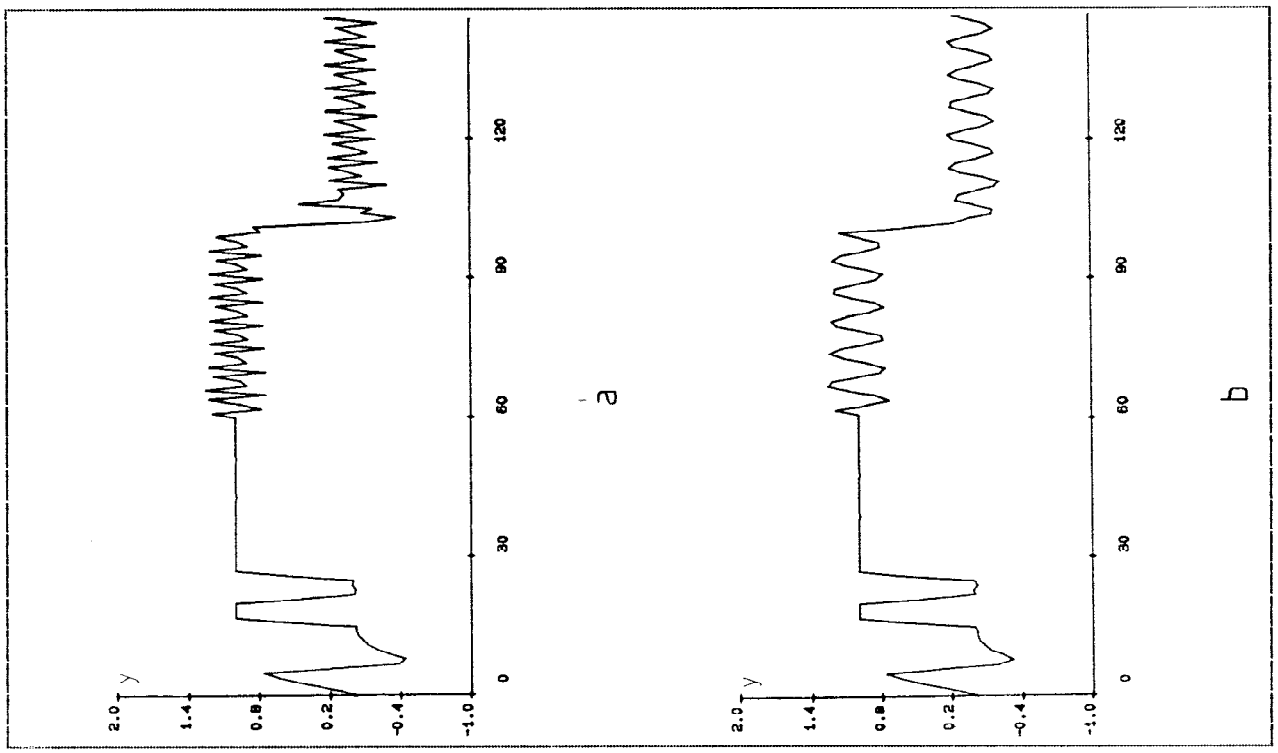


Fig 4.69. Efecto aliasing con fr=1.2 Hz y fr=1.7 Hz.

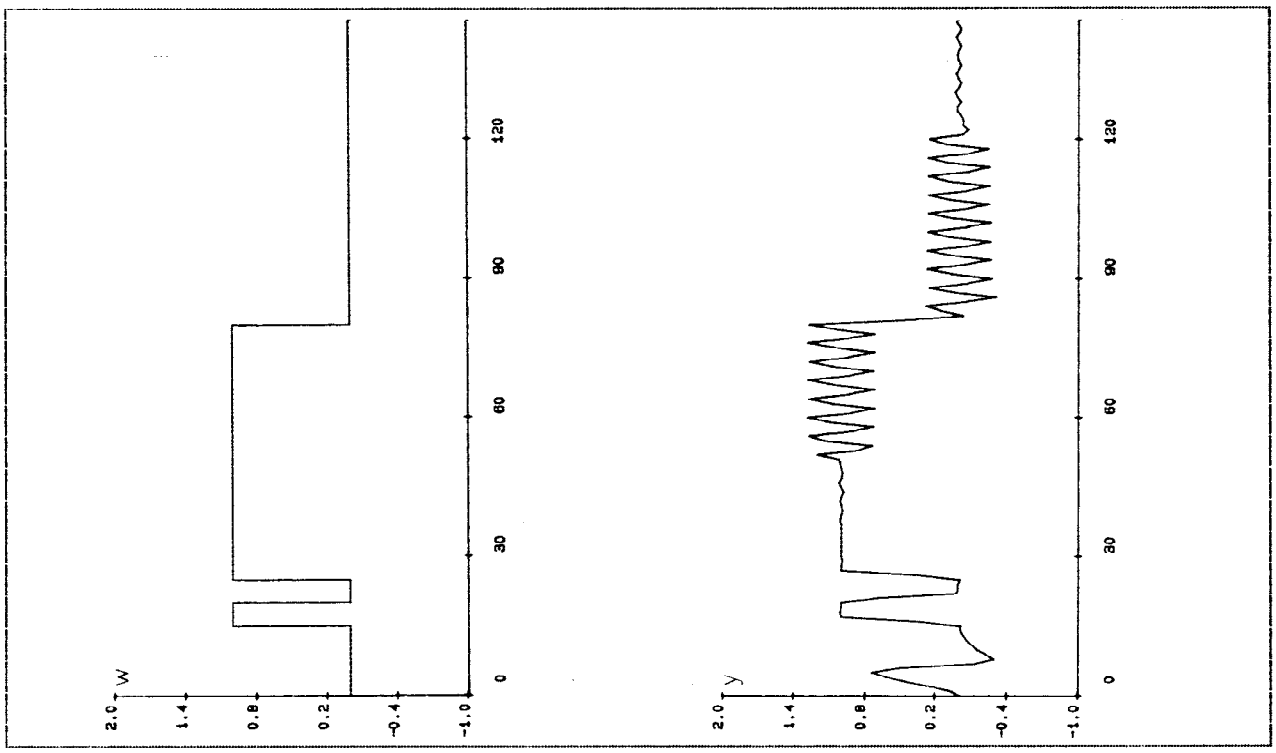


Fig 4.70. Efecto del filtro analógico implementado.

La eliminación del efecto aliasing se realiza con la inclusión en el sistema de un filtrado previo al muestreo. Para evitar que cambios en el período de muestreo conlleven cambios en el hardware, se emplea un filtro analógico con ancho de banda inversamente proporcional a un período fijo T=100 ms. Los procedimientos FAinic y FAnalog se encargan de que se realice la obtención de datos del conversor A/D con un período de muestreo fijo, de acuerdo con la frecuencia de corte del filtro analógico implementado.

Posteriormente se aplicará a la señal muestreada un filtro digital, de modo que el período de muestreo efectivo para el ciclo de estimación o control, T_m*, sea el procedente de la Preidentificación (o del fichero datos.pre). Este período será igual al que se aplica realmente (T_m), siempre que T_m*=iT, siendo i entero positivo. Si no ocurriera así se truncaría al múltiplo de T inmediatamente inferior, salvo en el caso de T_m*<T en que se hace T_m=T.

El filtro analógico utilizado es un filtro de Butterworth paso-bajo de segundo orden y ganancia unitaria, diseñado para que la atenuación a una frecuencia $f_s = f_m/2 = 5$ Hz sea de 20 dB. Para ello se deberá cumplir $f_g \approx 0.3\ f_s = 1.5$ Hz.

Como se ve en Iserman (1981), un filtro analógico activo es adecuado en este caso. El esquema electrónico del mismo se puede ver en la figura 4.68. Su efecto sobre un ruido de frecuencia fr=9.2 Hz, que actúa sobre el proceso desde k=40, se puede comprobar en la figura 4.70. En ella se observa que se ha desconectado el

filtro analógico desde k=50 hasta k=120.

La eliminación del ruido de baja frecuencia se va a efectuar mediante uno de los filtros digitales paso-bajo, de primer o segundo orden, implementados en el procedimiento FDigital y respondiendo, respectivamente, a la funciones de transferencia:

$$G_{F1}(z^{-1}) = b_0 / (1+a_1 z^{-1}) \quad \text{con} \quad a_1=-e^{-T_m/T} \quad b_0=1+a_1. \qquad (4.40)$$

$$G_{F2}(z^{-1}) = b_0 (1+2z^{-1}+z^{-2}) / (1+a_1 z^{-1}+a_2 z^{-2}) \qquad (4.41)$$

$$\text{con} \quad a_1=-2e^{-T_m/T} \quad a_2=e^{-2T_m/T} \quad b_0=(1+a_1+a_2)/4. \qquad (4.42)$$

Como se ve, ambos filtros tienen ganancia unitaria. La frecuencia de corte se tendrá a $f_g=1/2\pi T$.

El filtro digital, como se ve en la figura 3.6, sólo es efectivo hasta la frecuencia $f_m/2$, donde se presenta el primer mínimo en la curva. Los sucesivos mínimos se presentarán a frecuencias múltiplos impares de la anterior y los máximos, en los múltiplos pares.

En la figura 4.71 se ve la señal de salida del sistema de segundo orden implementado físicamente en la placa de pruebas, con T_m=500 ms e incorporando un filtro digital de primer orden con frecuencia de corte f_g=0.1 Hz y perturbado con una señal de ruido como se indica en con las señales r1.0 y r1.8, esto es:

- f_r=1.0 Hz desde k= 50 hasta k=100.

- f_r=1.8 Hz desde k=100 hasta k=150.
- f_r=1.0 Hz desde k=150 hasta k=200.

En la figura 4.72 se aplica al sistema un filtro digital de segundo orden y frecuencia de corte de 0.3 Hz, siendo la señal de ruido de las siguientes frecuencias:

- f_r=3.0 Hz desde k= 50 hasta k=100.
- f_r=3.8 Hz desde k=100 hasta k=150.
- f_r=3.0 Hz desde k=150 hasta k=200.

Como se ve en los dos casos, para k de 100 a 150 prácticamente no existe atenuación y ésta es máxima de 50 a 100 y de 150 a 200, lo que se corresponde con los máximos y mínimos de la figura 3.6.

Como se ha visto, con la inclusión del filtro analógico se atenúan los ruidos de frecuencia superior a 5 Hz y con el digital, los de frecuencia inferior, siempre que ésta sea menor que $f_m/2$. Con la inclusión del muestreo con período fijo, se elimina esta última restricción, como se deduce de la comparación de las figuras 4.73 (a) y (b). En ambas T_m=500 ms, f_r=1.5 Hz (a partir de k=60) y se aplica el filtro digital de primer orden con f_g=0.1 Hz.

En el primer caso se observa el efecto aliasing con $f_2=f_m-f_r$=0.5 Hz. En el segundo se incluye además el muestreo con período fijo T=100 ms con lo que ahora $f_m/2$=5 Hz y el filtro digital es efectivo con f_r=1.5 Hz.

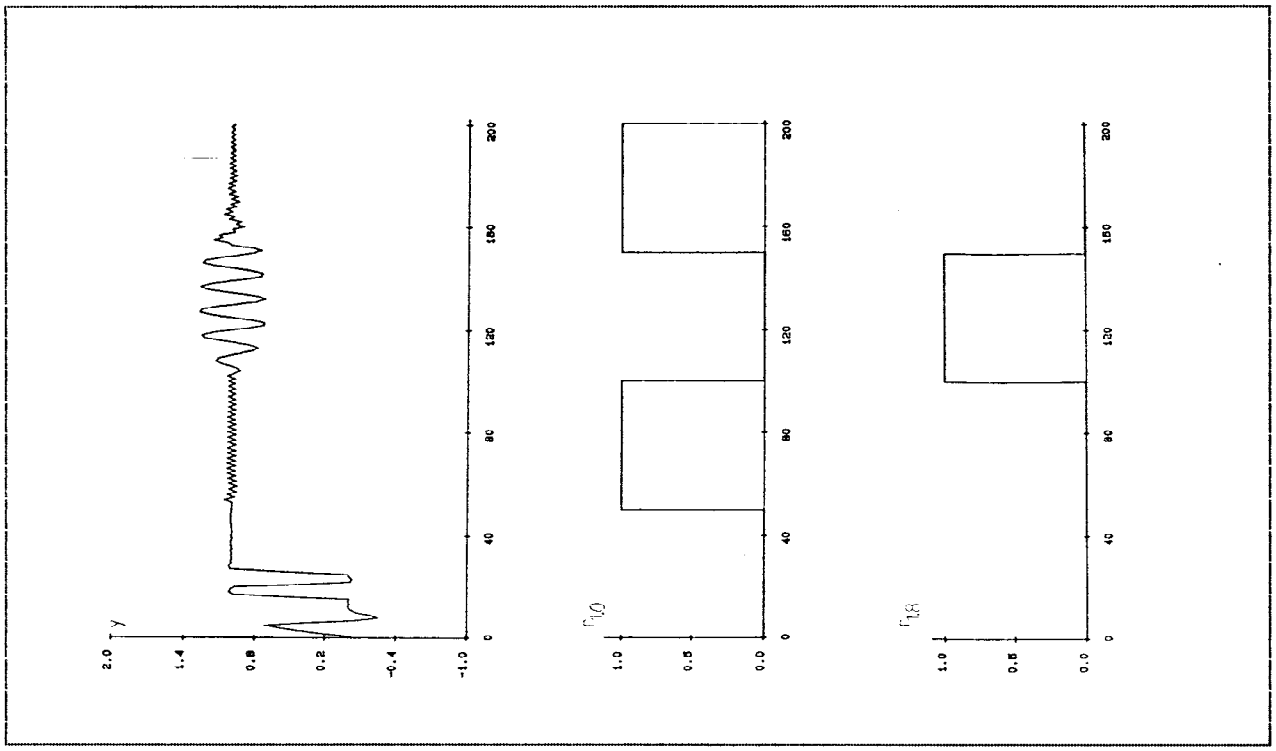


Fig 4.71. Efecto del filtro digital de orden 1.

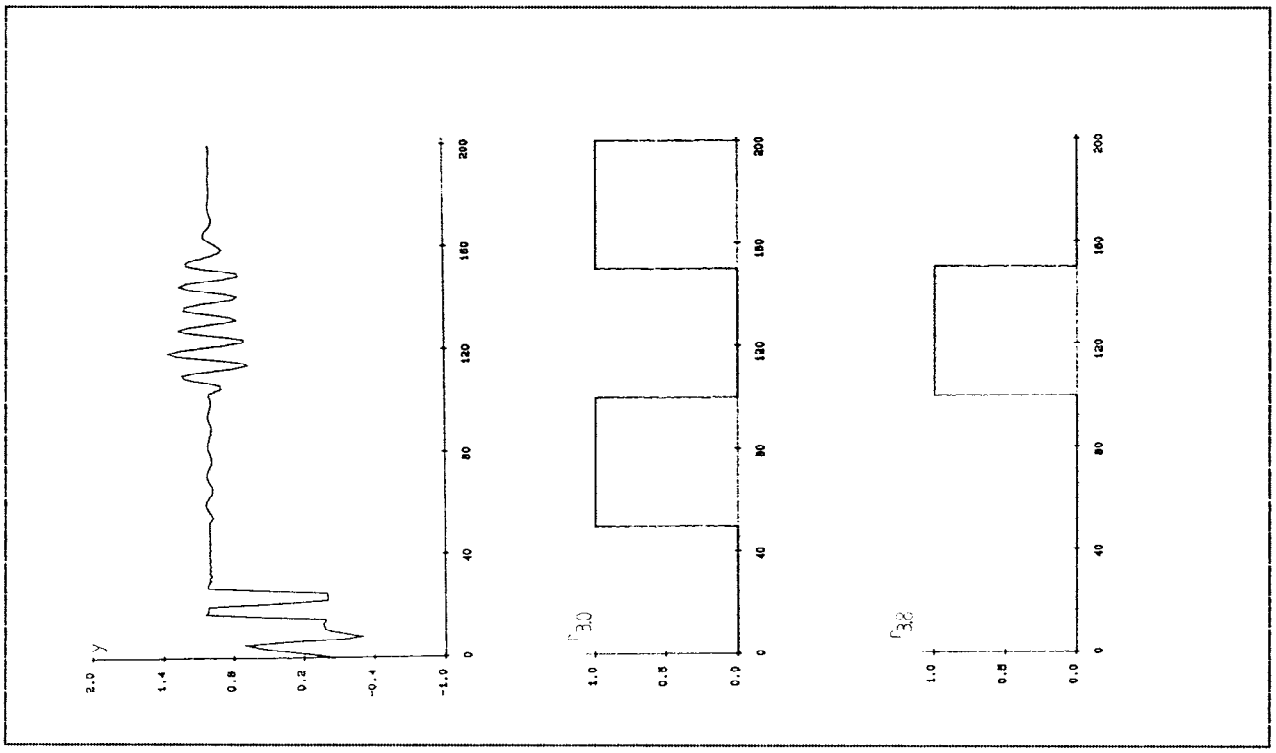


Fig 4.72. Efecto del filtro digital de orden 2.

4.4.7.2 Filtrado de outliers.

Son errores de medida caracterizados por producir grandes desviaciones en la señal medida, con una probabilidad de ocurrencia muy baja. El filtro de outliers incluido en el procedimiento FOutlier, se encarga de detectar y eliminar estas medidas erróneas, sin que se disminuya la capacidad de seguir consignas variantes por la señal de salida. Para ello el filtro se va acomodando en tiempo real a las variaciones de la señal medida.

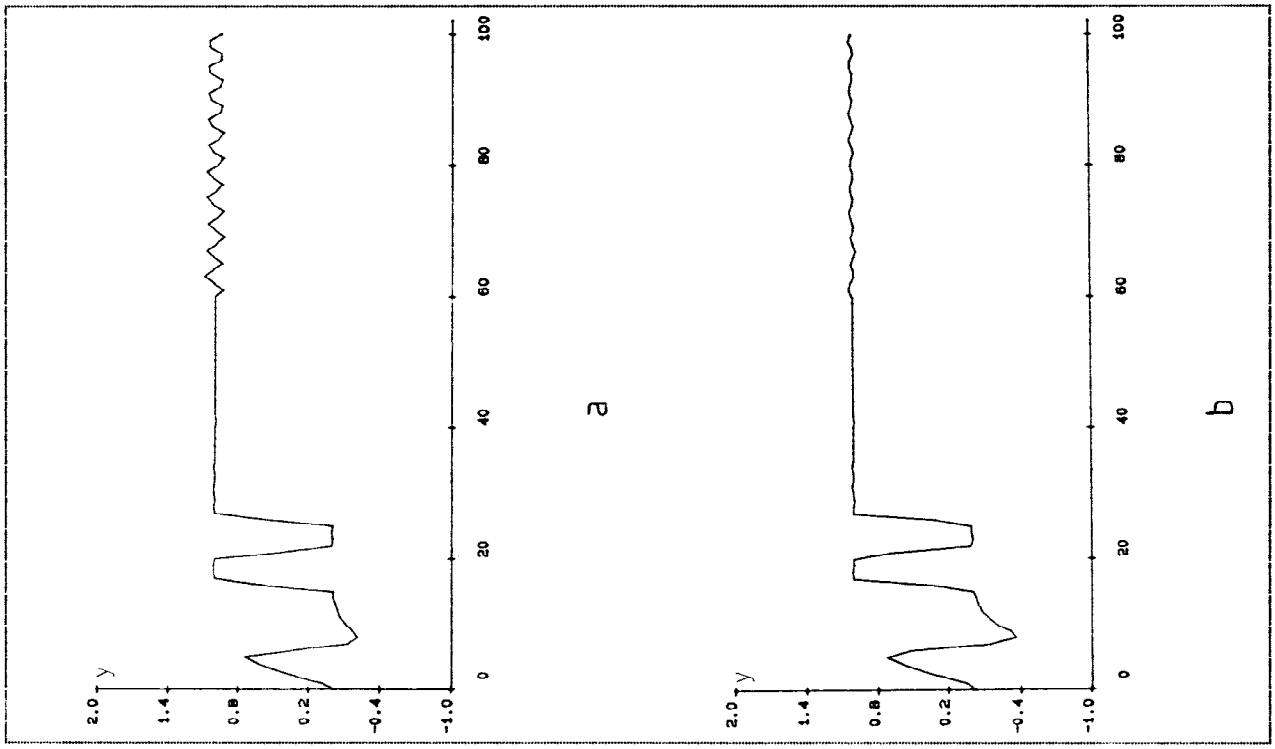


Fig 4.73. Muestreo sin/con período fijo T=100 ms.

La detección se realiza estimando el valor medio y la varianza de la señal de salida, según las ecuaciones (3.9) y (3.10). Se

considerará que son outliers las señales que cumplan:

$$| y(k+1) - \hat{y}(k+1) | > K(k+1)\ \hat{\sigma}_y(k+1) \qquad (4.43)$$

con K(k) acomodándose en tiempo real a la magnitud de las variaciones en la señal medida, mediante un sencillo algoritmo:

```
Si detección_outlier entonces
   K(k+1) := K(k) + 1.0
sino
   si K(k+1) > factor entonces
       K(k+1) := K(k) - 1.0
   fin
fin.
```

siendo factor un parámetro seleccionable funcionando el sistema en modo Menú. Otro parámetro seleccionable por software será el período de muestreo en que comienza la detección y eliminación de outliers, necesitando el algoritmo un tiempo previo para elaborar unas estimaciones de y y σ_y^2 adecuadas, antes de su uso en el filtrado.

Las acciones que toma el sistema ante la detección del outlier son de dos tipos: aviso al usuario y atenuación de la magnitud del mismo. La información al usuario se realiza, si ha sido seleccionado así en la defininción de alarmas, encendiendo uno de los leds incorporados en el sistema.

La eliminación del outlier detectado se realiza fácilmente sustituyendo la medida errónea por:

$$\hat{y}(k+1) = \hat{y}(k+1) \pm K(k+1)\ \hat{\sigma}_y(k+1) \qquad (4.44)$$

o sea, la máxima desviación permitida de la media, en el sentido dado por el signo de $y(k+1) - \hat{y}(k+1)$.

En la figura 4.74 (a) se pueden observar los efectos de un outlier en k=120 sobre la salida del sistema, cuando no está activado el filtro de outliers. Como se ve, el controlador reacciona ante lo que considera una medida válida (se ha introducido y=20.0), llegando a la saturación del actuador, para llevar la variable controlada a su valor correcto.

En la figura 4.74 (b) se observan los efectos de la misma medida errónea cuando se ha incorporado el filtro de outliers a partir de k=30, siendo factor =3.0. La salida en este caso toma un valor $\hat{y}(k+1) \approx 2.0$ en lugar de 20.0 y no se llega a la saturación del actuador. La máxima desviación permitida de la media en este caso es suficiente para que la salida siga a la consigna con un 99.73% de probabilidad (como se ve el filtro de outliers no actúa ante el cambio de consigna en k=60, posterior a su activación).

Para disminuir aún más los efectos del outlier, con la misma media y varianza de la señal medida, se podrá disminuir el valor de

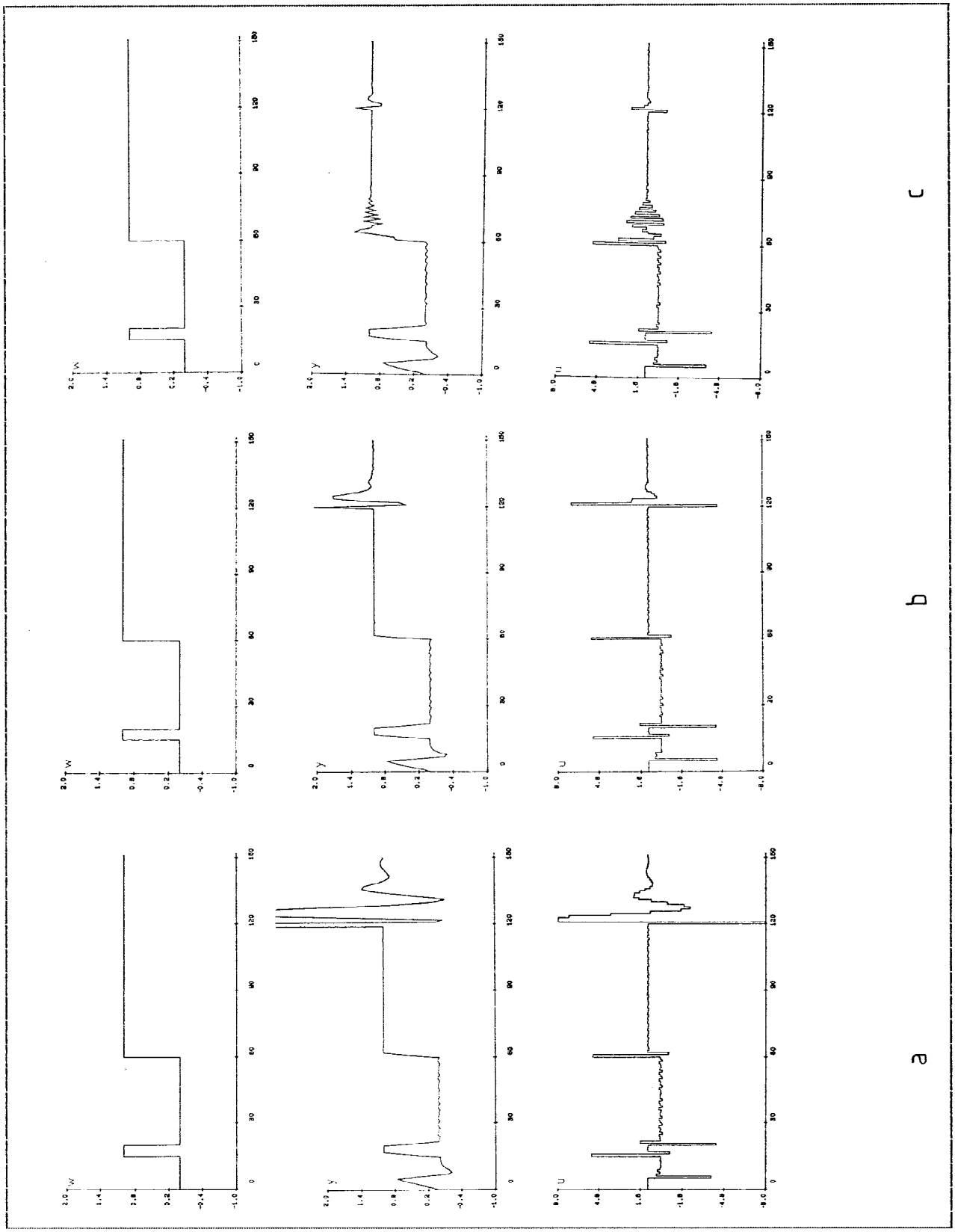


Fig 4.74. Efecto del filtrado de outliers.

factor del algoritmo anterior. En la figura 4.74 (c) se muestran los efectos del mismo outlier cuando factor =1.5. Ahora la salida toma un valor más reducido $\hat{y}(k+1)\approx 1.25$, pero a costa de disminuir la capacidad de seguimiento de la consigna, como se puede apreciar ante el cambio en k=60.

4.5 GESTION DE DISPOSITIVOS

4.5.1 Representación gráfica en pantalla.

Este módulo se utiliza para la representación gráfica en pantalla, utilizando una tarjeta compatible Hércules. Para la representación se utilizan las rutinas de una versión de Turbo Graphix Toolbox de Borland, en Modula-2.

Las funciones que cumplen los procedimientos incluidos en este módulo son básicamente de tres tipos:

- Representación gráfica en pantalla de las variables seleccionadas mediante menú. Esta representación podrá ser realizada, en caso de especificación de no existencia de un valor límite de k, de manera continuada, de forma que siempre se muestren en pantalla los últimos 100 valores, como se ve en la figura 4.75.

- Transferencia de las variables a representar al driver del plotter, en el caso en que se requiera copia en papel de la representación.

- Representación, en las líneas reservadas al interface con el intérprete de comandos, de las órdenes del usuario así como las respuestas a las mismas.

4.5.2 Representación numérica en pantalla.

Este módulo se utiliza para gestionar las representaciones numéricas en pantalla. Se utilizan las rutinas incluidas en el programa Window Machine para Modula-2. Las variables a representar son las seleccionadas mediante menú y se pasarán al driver de la impresora en caso necesario. Así mismo, se representarán las órdenes del usuario y la respuesta a las mismas por parte del interface de comandos, como se muestra en la figura 4.76.

Independientemente del modo de representación seleccionado mediante menú, los datos adquiridos con el funcionamiento del sistema en modo Data Logging serán representados en forma numérica, indicando el período de adquisición actual así como el canal analógico de procedencia de cada uno.

Las funciones que se llevan a cabo por GRA y NUM son realizados por procedimientos de idéntico nombre para representaciones gráficas o numéricas. La selección del conjunto que se aplica, se realiza en MASTER dependiendo de la opción que se selecciona mediante menú. Los procedimientos son:

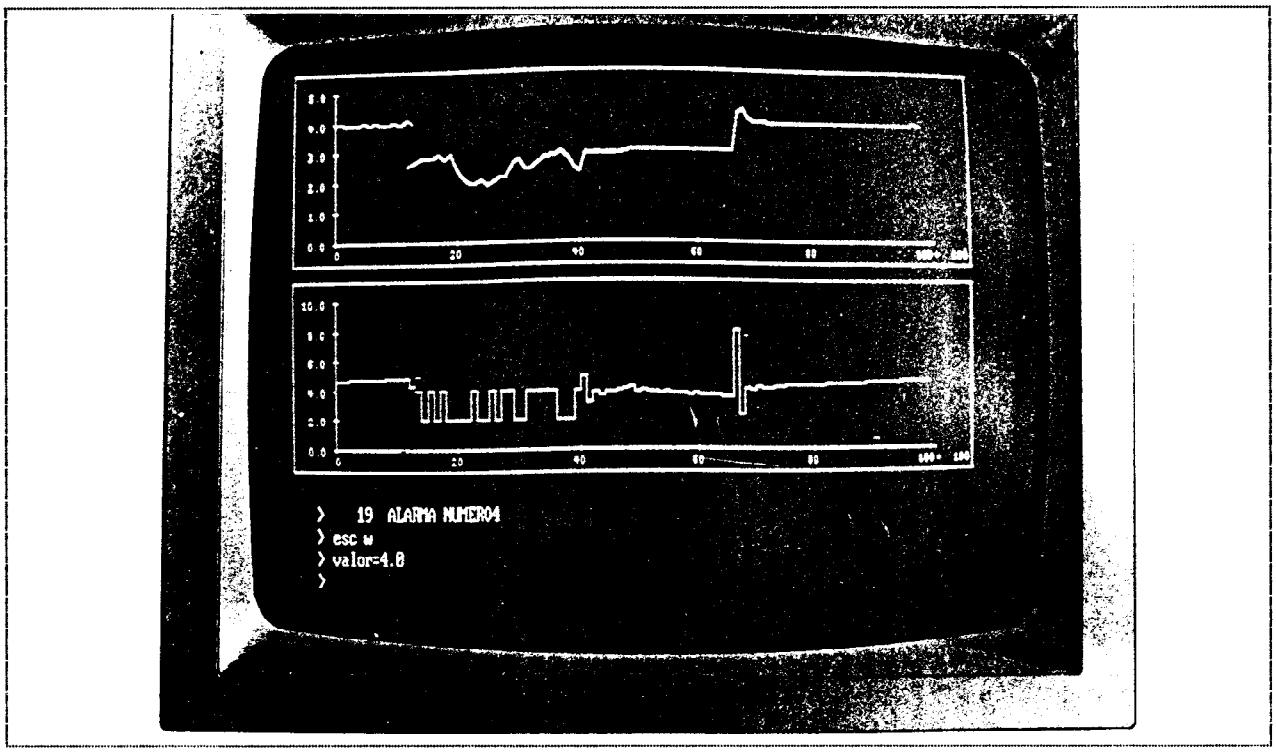
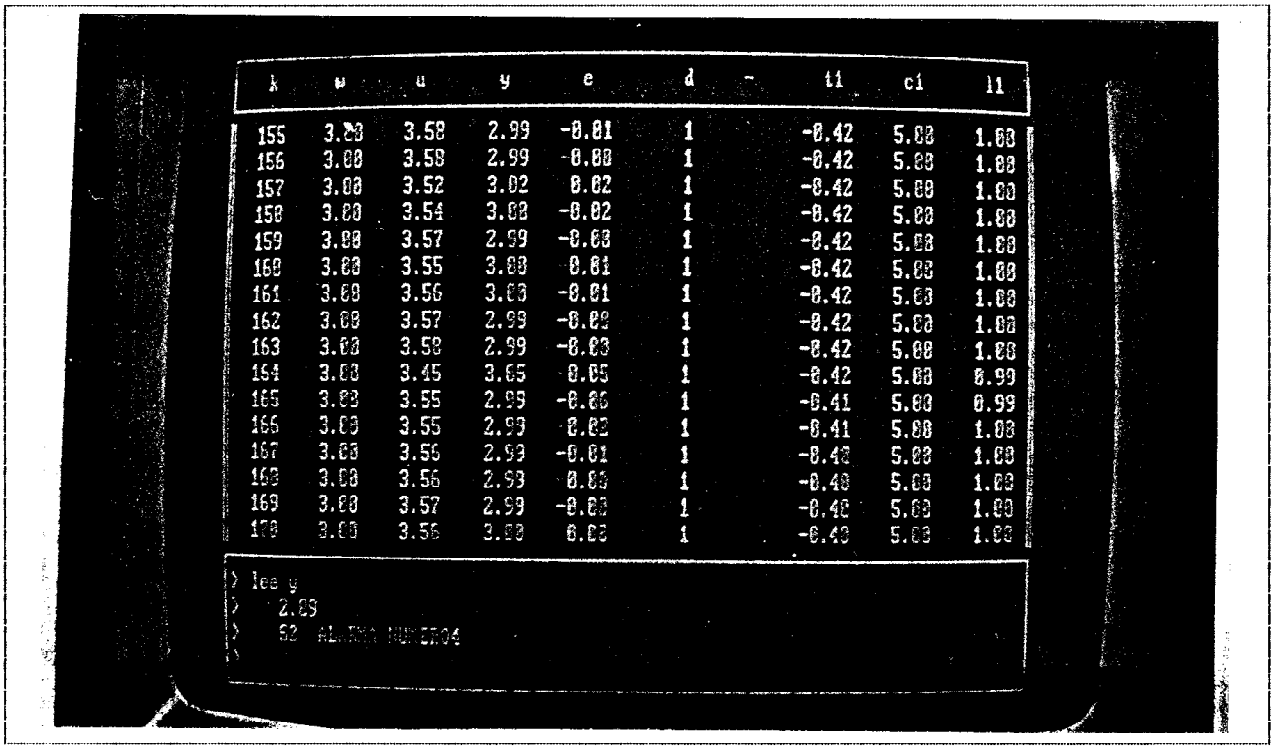


Fig 4.75. Representación gráfica continua en pantalla.



k	w	u	y	e	d	-	ti	ci	li
155	3.00	3.58	2.99	-0.01	1		-0.42	5.00	1.00
156	3.00	3.58	2.99	-0.00	1		-0.42	5.00	1.00
157	3.00	3.52	3.02	0.02	1		-0.42	5.00	1.00
158	3.00	3.54	3.00	-0.02	1		-0.42	5.00	1.00
159	3.00	3.57	2.99	-0.00	1		-0.42	5.00	1.00
160	3.00	3.55	3.00	0.01	1		-0.42	5.00	1.00
161	3.00	3.56	3.00	-0.01	1		-0.42	5.00	1.00
162	3.00	3.57	2.99	-0.00	1		-0.42	5.00	1.00
163	3.00	3.58	2.99	-0.00	1		-0.42	5.00	1.00
164	3.00	3.45	3.05	0.05	1		-0.42	5.00	0.99
165	3.00	3.55	2.99	-0.00	1		-0.41	5.00	0.99
166	3.00	3.55	2.99	0.00	1		-0.41	5.00	1.00
167	3.00	3.56	2.99	-0.01	1		-0.40	5.00	1.00
168	3.00	3.56	2.99	0.00	1		-0.40	5.00	1.00
169	3.00	3.57	2.99	-0.00	1		-0.40	5.00	1.00
170	3.00	3.56	3.00	0.00	1		-0.40	5.00	1.00

> lee y
> 2.09
> 92 ALARMA NUMERO4
>

Fig 4.76. Representación numérica en pantalla.

- Inicio: Inicialización del modo gráfico o numérico, así como los drivers del plotter o la impresora, definiendo las ventanas que van a ser usadas para la representación y la zona reservada para el interface con el usuario.

- REP: Representación, en cada período de muestreo, de las variables seleccionadas, así como su transferencia al driver del plotter o impresora.

- ORDEN Y VARIABLE: Representación de los comandos dados por el usuario y respuesta a los mismos por parte del intérprete.

4.5.3 Representación gráfica en plotter.

Este módulo se utiliza para complementar un fichero con órdenes para el plotter (datos.plo), encaminadas a la representación gráfica de las variables deseadas. El fichero se enviará posteriormente al plotter, mediante línea serie.

Las órdenes son escritas en el fichero utilizando las rutinas (movea, drawa...) incluidas en el driver del mismo, que debe ser cambiado si se cambia de plotter.

4.5.4 Representación mumérica en impresora.

Este módulo se utiliza para complementar un fichero (datos.imp), con la representación numérica de las variables deseadas, que

posteriormente se enviará a una impresora conectada al computador. En el caso de fucionamiento en modo Data Logging los datos adquiridos se almacenarán así mismo en fichero (datos.dat).

Los procedimientos básicos de estos dos últimos módulos tienen idéntico nombre para representaciones gráficas o numéricas, realizándose la selección de los utilizados en función del tipo de representación. Son los siguientes:

- INIC: Apertura del fichero con los datos a imprimir así como de las ventanas con representación de los ejes (gráfica) o escritura de la cabecera (numérica).

- VOLCADO: Representación de las variables seleccionadas en la ventana (gráfica) o columna (numérica) correspondiente.

4.5.5 Driver del plotter.

Este módulo es un driver del plotter 7475A de Hewlett-Packard.

4.6 OTROS MODULOS.

4.6.1 Adquisición de datos.

En este módulo están implementados los procedimientos de alto nivel que se ejecutan cuando se selecciona la opción de Data Logging mediante menú. Dichos procedimientos son los siguientes:

- DataLogging: En él se realiza la inicialización del proceso de recogida de datos, mediante llamada al procedimiento del módulo ADC que programa la transferencia de datos procedentes de la conversión A/D vía DMA.

  Esta transferencia no se programa en modo autoinicialización, de modo que se pueda efectuar un control del período de adquisición, lo que se realiza mediante la señal de salida del contador 0 del temporizador. Esta señal, además de habilitar a los contadores 1 y 2, se llevará a la entrada digital 1 y, mediante su lectura, se sincronizará la adquisición con el período seleccionado por el usuario, como se indica en la figura 4.77.

- IntData: En él se realizan, en cada período de muestreo, las operaciones propias del funcionamiento del sistema en modo Data Loggin, como son:

  a) Adquisición de datos: Recepción de los datos procedentes de la conversión A/D, así como selección del canal analógico de adquisición de los mismos.

  b) Tratamiento: Filtrado digital, linealización, conversión a unidades de ingeniería, cálculos, comprobación de la validez de los datos, promediados, etc. Este grupo de operaciones depende de la aplicación particular del

sistema de adquisición y no está implementado en el sistema MASTER.

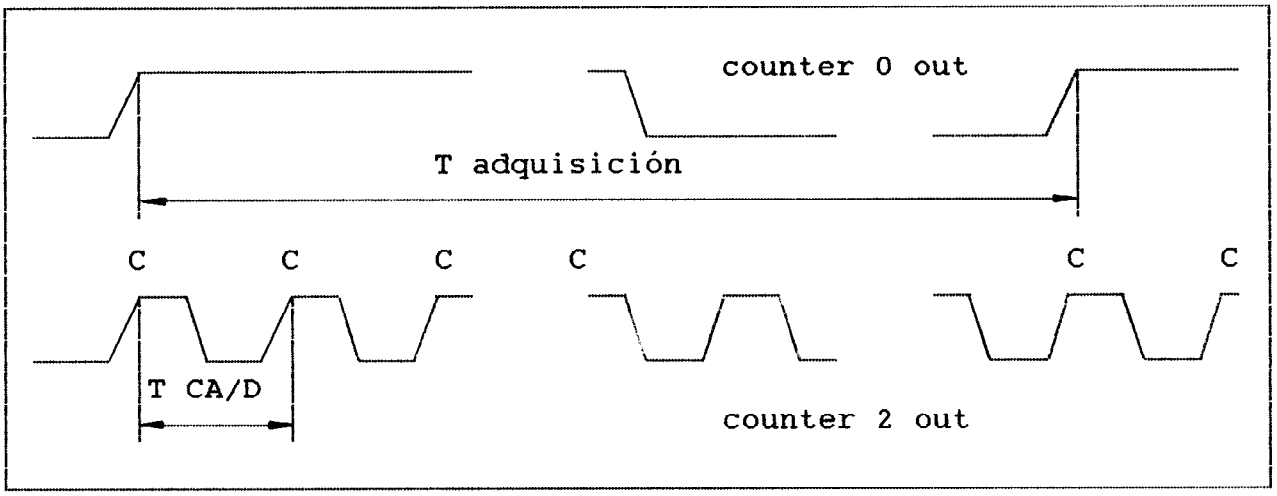


Fig 4.77. Sincronización en adquisición de datos.

c) Presentación: Si se selecciona la opción correspondiente por menú, es posible realizar la visualización de los datos recibidos por pantalla on-line. Sin embargo, no se aconseja la realización de esta presentación por pantalla cuando el período de adquisición es muy bajo, ya que el tiempo necesario para la visualización puede incrementar dicho período, modificando la selección realizada previamente.

d) Almacenamiento: El almacenamiento de los datos se realiza siempre en un fichero "datos.dat" contenido en un disco virtual. Esto implica que las transferencias se realizan memoria RAM a memoria RAM con lo que los tiempos consumidos en las mismas son pequeños. Una vez finalizada la

adquisición es posible almacenar permanentemente los datos en disco, imprimirlos, etc.

En la figura 4.78 se muestra la finalización de la visualización en pantalla de la adquisición de datos procedentes de 3 canales analógicos (canal 0, 1 y 2), con 14 muestras por canal y período. El período de recogida de datos es de 20 ms, realizándose la adquisición durante 100 períodos.

4.6.2 Módulos auxiliares.

De entre otros módulos utilizados, caben resaltar:

- WM (Window Machine): Gestión de ventanas.
- Tubo Graphix: Representaciones gráficas en pantalla.
- MathLib1: Rutinas matemáticas, no incluidas en el MathLib0 para Modula-2, necesarias para el cálculo de la función de transferencia del controlador, con alguna de los métodos implementados. Así MatInv invierte una matriz cuadrada, usando el método de Gauss-Jordan y MatSim resuelve un problema lineal de minimización con restricciones, usando el método Simplex.

4.7 CONTROL ADAPTATIVO DE UNA PLANTA PILOTO.

Además de los sistemas de primer y segundo orden implementados físicamente en la placa de pruebas descrita en el apartado 4.4.2, se ha usado para la experimentación una planta piloto que consta de

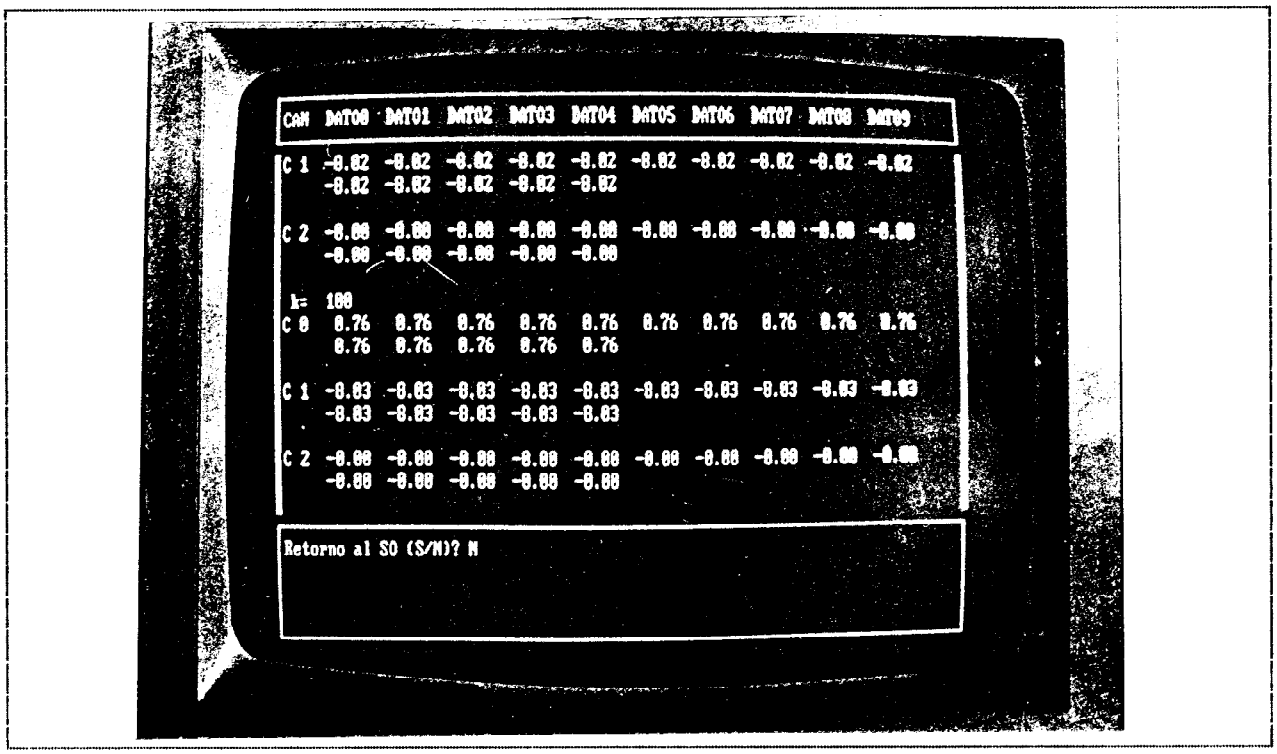


Fig 4.78. Resultado de la adquisición de datos.

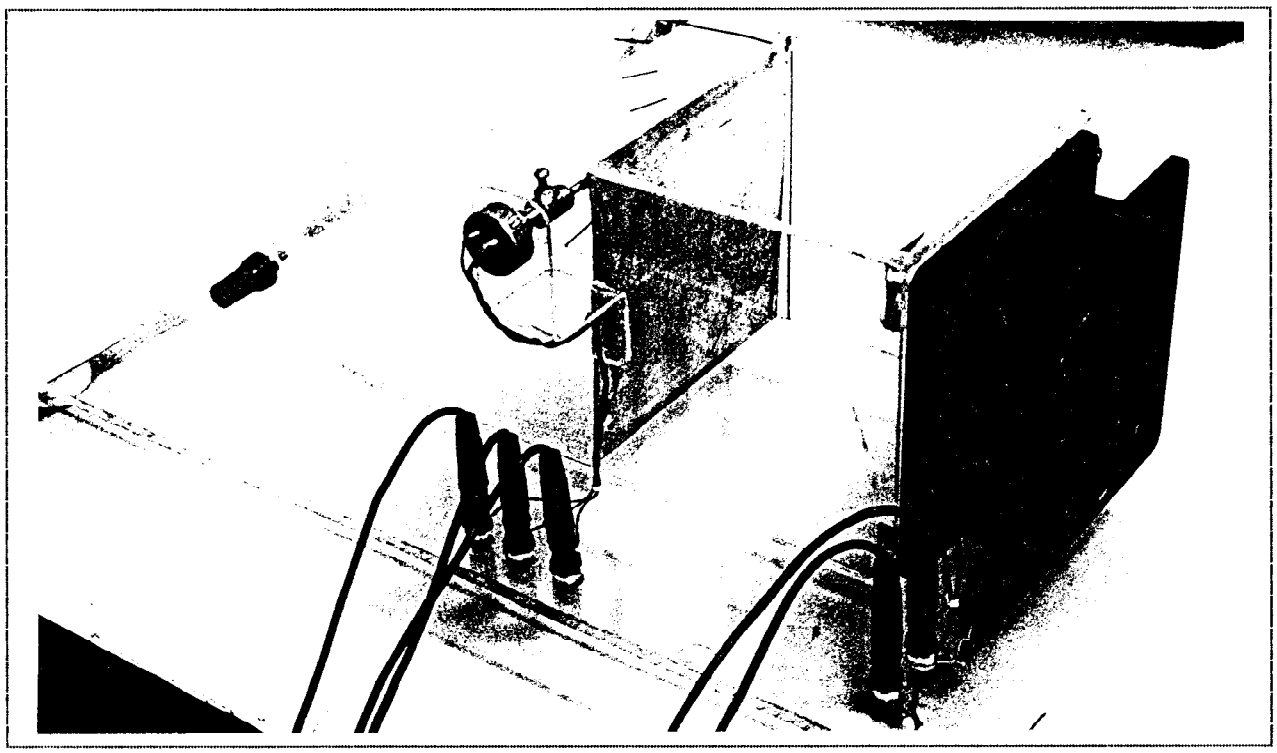


Fig 4.79. Planta piloto.

(figura 4.79):

- Ventilador de c.c.
- Conducto de canalización del aire.
- Placa pendular.

Se pretende controlar el ángulo que forma la placa con la vertical, mediante el flujo de aire que proviene del ventilador de velocidad variable. El ventilador es accionado por un motor de corriente continua con amplificador de potencia lineal y el ángulo de la placa es medido por un potenciómetro de bajo rozamiento.

La función de transferencia del motor-ventilador-placa, resulta de cuarto orden con dos polos complejos conjugados sobre el eje real en el plano S (en el límite de estabilidad), aunque suponiendo despreciable la inductancia del motor (como ocurre realmente), el sistema puede ser modelado con orden tres.

El desconocimiento de varios de los parámetros del sistema, así como el alto grado de no-linealidad que presenta hace adecuado el uso de un controlador adaptativo para su gobierno.

En la figura 4.80 se muestran las señales de control, consigna y salida durante la fase de arranque del sistema con control adaptativo en los casos en que el número de períodos utilizados en la estimación previa de parámetros sea de (a) 5 (b) 15. En ambos casos se ha utilizado un período de muestreo de 800 ms y se ha tomado

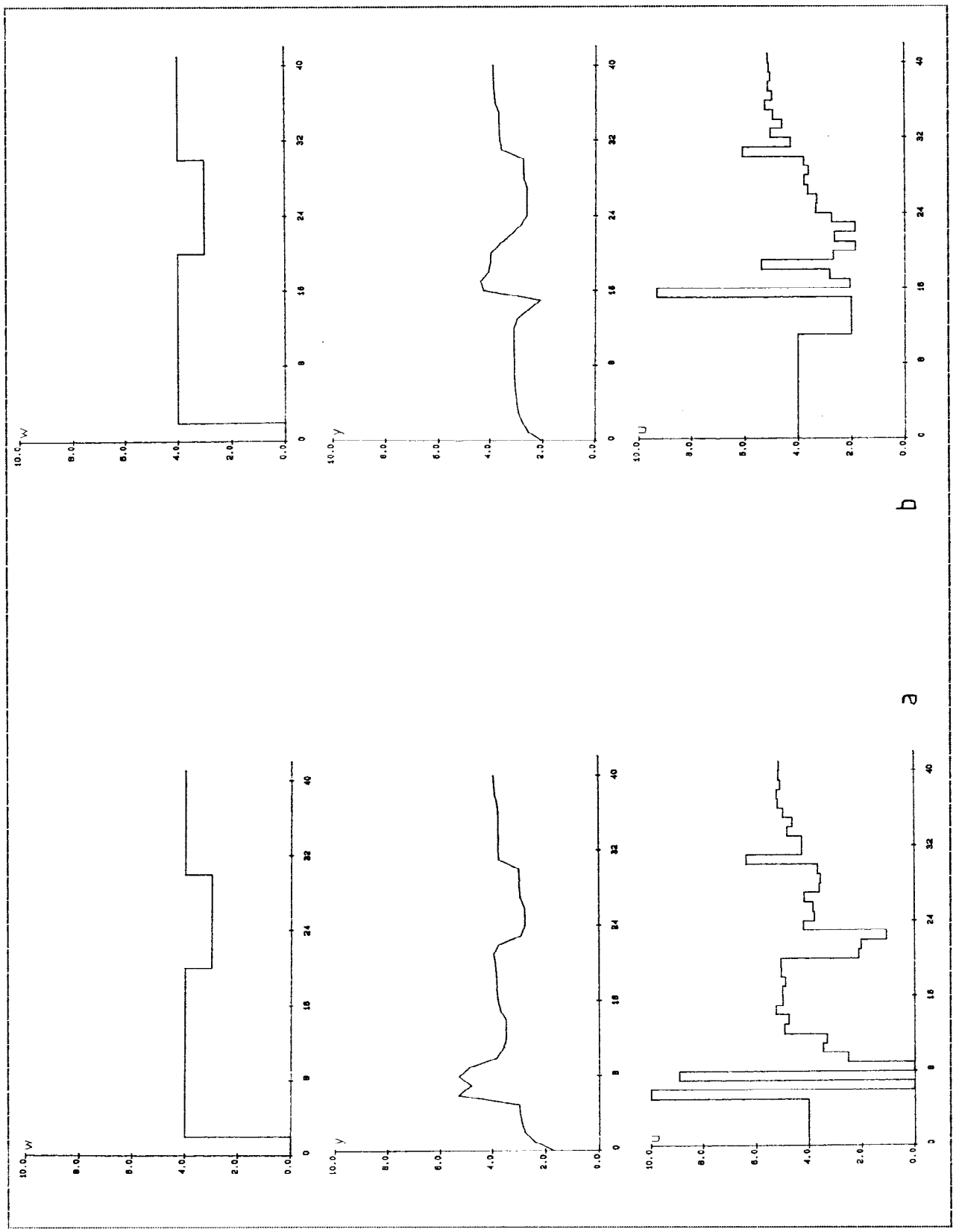


Fig 4.80. Fase de arranque con control adaptativo.

$\hat{m}$=3, d=2, τ límite =0.95, registro de generación de PRBS con 11 bits con amplitud de 1 V. y algoritmo de control de tiempo finito de orden incrementado.

El punto de funcionamiento inicial se ha tomado con U_{00}=3 V., lo que da para la señal de salida un valor cercano a Y_{00}=2 V., equivalente a un ángulo de unos 18° (10 V. procedentes del sensor de posición corresponden a 90° de la placa con la vertical).

Como se puede comprobar, en el caso (a) al finalizar la fase de estimación previa (con valores de U entre 2 y 4 V., las estimaciones de los parámetros aún no son lo bastante aproximadas y la sobreoscilación inicial que se produce en el siguimiento de la consigna es bastante elevada. En el caso (b) la señal de salida se aproxima mucho mejor a la consigna aunque, como se podrá comprobar en los casos siguientes, es posible mejorar esta fase incrementando aún más el número de períodos de muestreo de estimación previa (en adelante se tomarán 40).

Se muestran con las siguientes gráficas los problemas que se presentan para el control de la planta, dado el comportamiento altamente no lineal de la misma, cuando no se efectúa una correcta estimación de sus parámetros en las distintas zonas de funcionamiento que se indican en la figura 4.81. Como se ve, al pasar de la zona 1 a la 2 se incrementa considerablemente la pendiente de la curva entrada/salida, aunque ésta es similar en las zonas 1 y 3.

En la figura 4.82(a) se utiliza control adaptativo sin supervisión y con $\tau=1$ una vez finalizada la fase estimación previa de parámetros. Como se ve, al pasar de las zonas 1 ó 3 a la 2 se produce un comportamiento no deseado del sistema dado que no se puede seguir correctamente las variaciones de los parámetros del mismo.

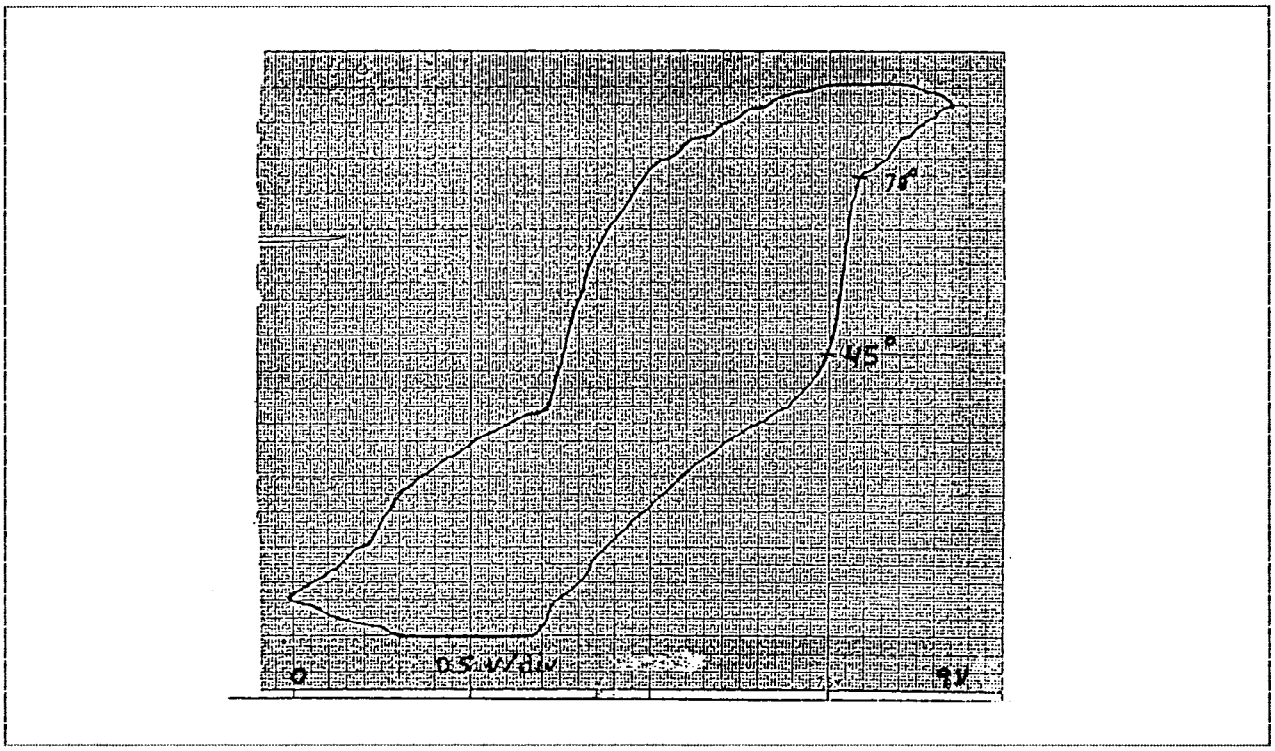


Fig 4.81. Zonas de funcionamiento de la planta piloto.

En el caso (b) se utiliza control adaptativo con supervisión y el método propuesto en el apartado 2.3.1.3 para que el factor de olvido se adecúe a la excitación existente, con lo que, gracias a la provocada en los momentos de cambios de consigna, se podrá efectuar un seguimiento correcto de las estimaciones de los parámetros cuando se pasa de una a otra zona de funcionamiento.

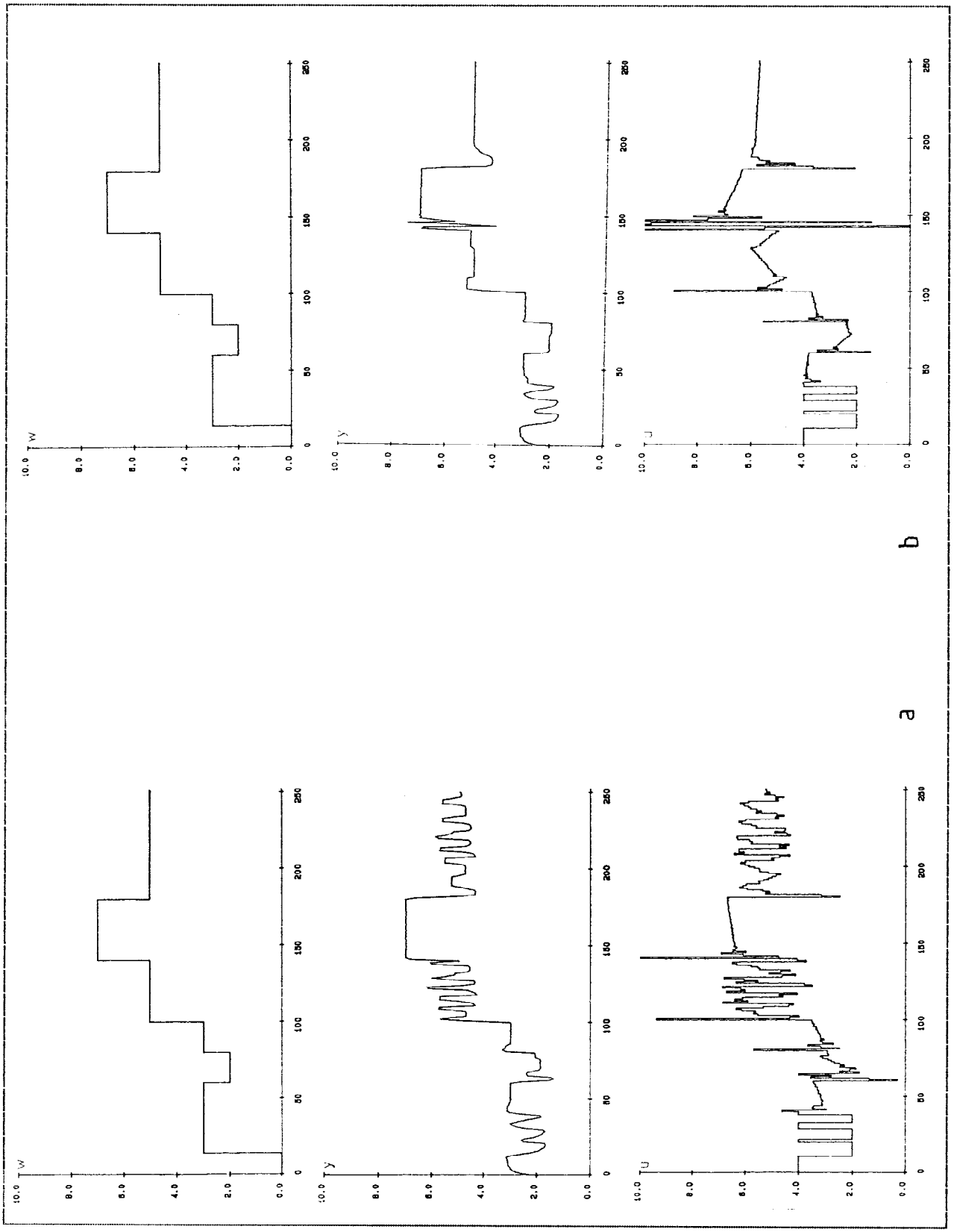


Fig 4.82. Control adaptativo de la planta piloto.

En la figura 4.83 (a) y (b) se muestra la diferencia de comportamiento del algoritmo de control adaptativo sin y con supervisión (incorporando filtrado de outliers), cuando aproximadamente entre k=100 y 140 se coloca una placa a pocos centímetros del ventilador con lo que se dificulta la entrada de aire al mismo. En k=170 se le envía una medida errónea al controlador (Y=10 V.) como si proviniera del sensor de posición para simular un outlier.

Como se ve el comportamiento en el caso (b) es mejor, aunque en los dos casos la señal de control aumenta, para mantener el correcto seguimiento de la consigna, cuando se simula una perturbación determinista con la placa.

Los efectos de la inclusión del filtrado analógico y/o digital cuando existen perturbaciones estocásticas se muestran en las figuras 4.84 (a) y (b). En el caso (a) se aplica una señal senoidal de frecuencia 0.62 Hz a partir de k=110 y de 1.2 Hz a partir de k=180. El filtro digital de segundo orden no actúa entre k=130 y k=150. Como la frecuencia de Nyquist es de:

$f_N=f_m/2 \qquad f_m=1/0.8=1.25 \text{ Hz} \qquad f_N=0.625 \text{ Hz}$

el filtro digital atenúa la señal de 0.62 Hz pero no es efectivo con la de 1.2 Hz.

En el caso (b) está activo el filtro analógico de Butterworth implementando físicamente en la placa de pruebas, a la que está

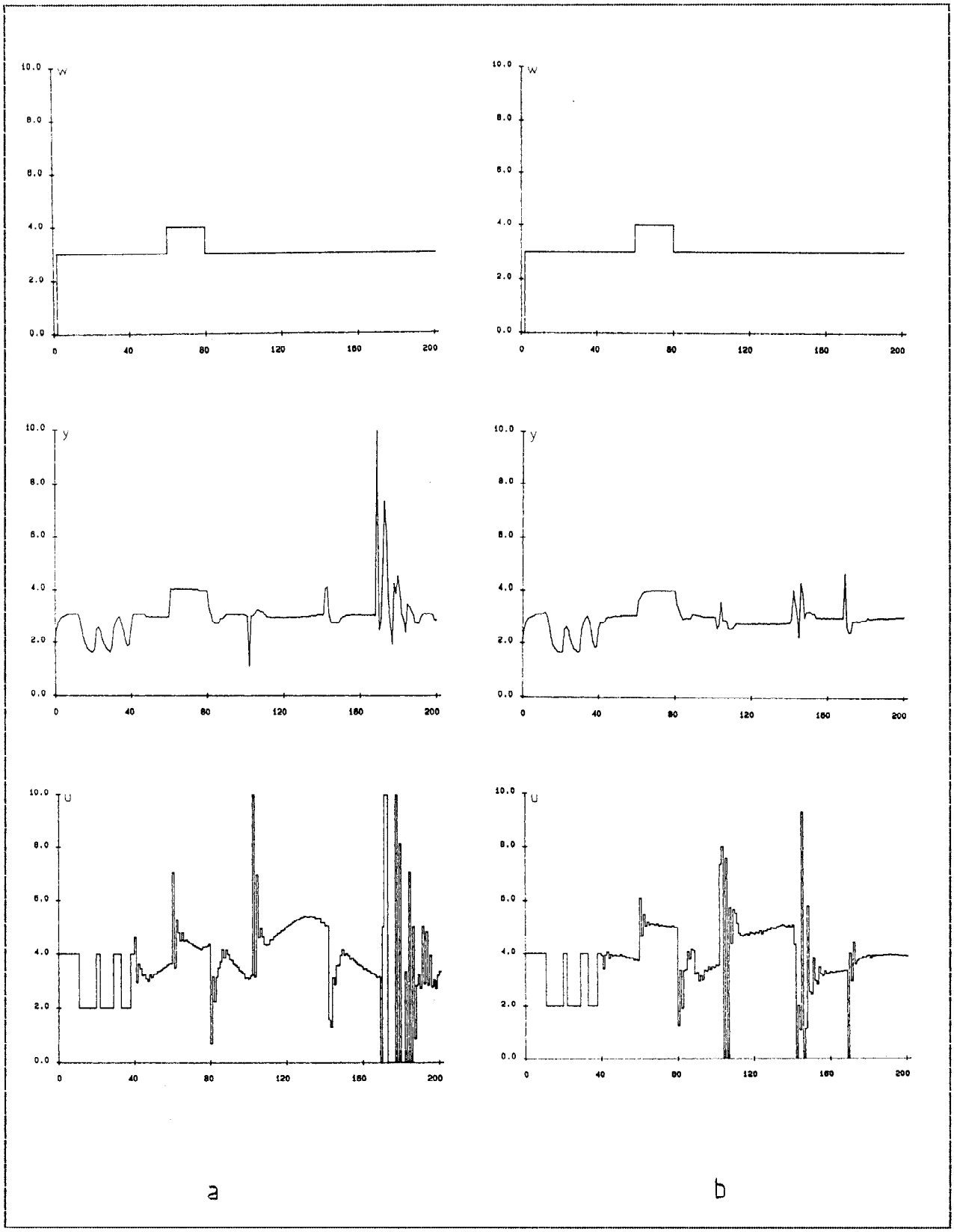


Fig 4.83. Comportamiento sin y con supervisión.

conectada la salida de la planta piloto, salvo entre k=80 y k=110. Se aplica una señal senoidal de frecuencia 8 Hz a partir de k=50.

En la figura 4.85 se puede comprobar la actuación de un controlador de respaldo, definido previamente, cuando la diferencia entre la señal de salida y de consigna supera un cierto valor especificado mediante menú. Los períodos en que el controlador de respaldo está activo se muestran en (c).

Se ha definido así mismo que la necesidad de activación continuada de este controlador pueda dar lugar a otro tipo de acción, como es el paso a modo Manual en la tercera activación (k=150) del controlador y hasta que el operador lo decida. Así éste actúa incrementando manualmente la señal de control hasta k=211 (d), en que conmuta a modo automático. Se observa así mismo que esta transición no se produce bruscamente, gracias al mecanismo software implementado para evitarlo.

4.8 RETARDO COMPUTACIONAL.

La forma de sincronización de entradas y salidas que se ha tomado en la implementación del sistema, es la mostrada en la figura 3.21 (b), es decir, se actualiza tan pronto como es posible la variable de control, no retardándola innecesariamente (caso (a)).

Esto implica que el tiempo entre conversiones A/D y D/A será variable, dependiente del tiempo de cálculo de los algoritmos que

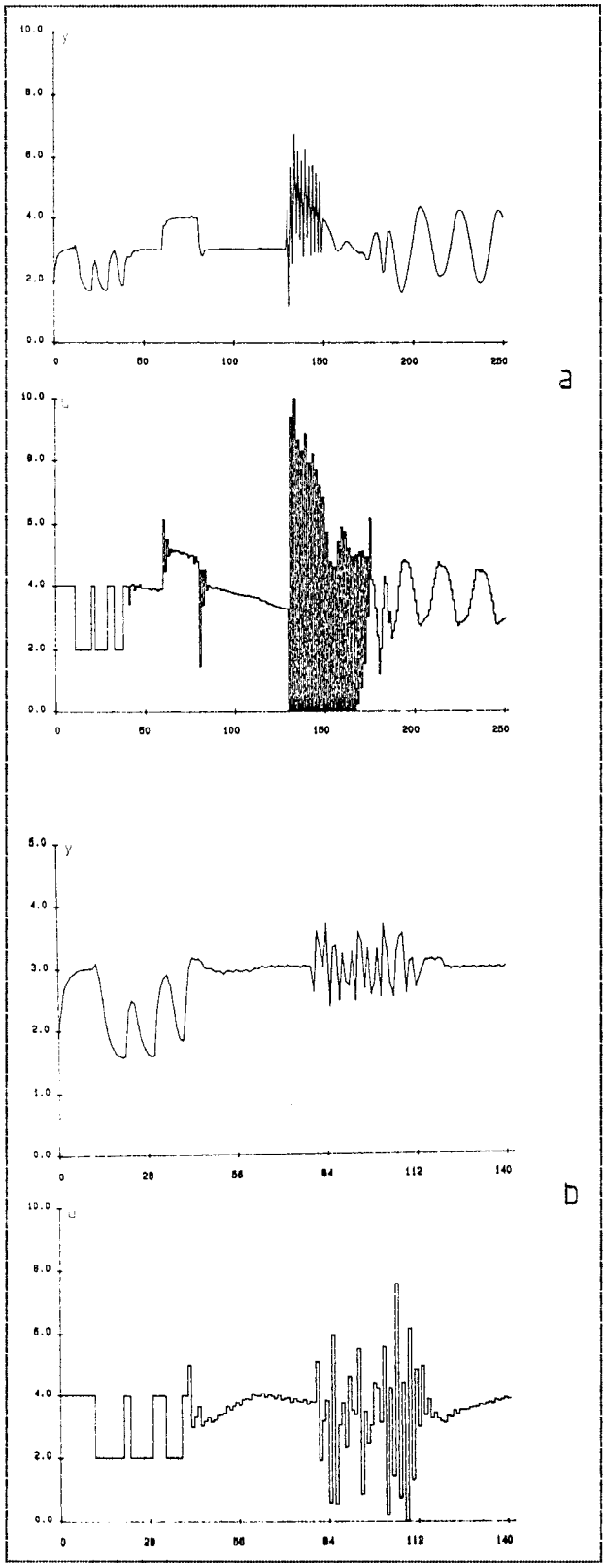


Fig 4.84. Inclusión de filtrado de perturbaciones.

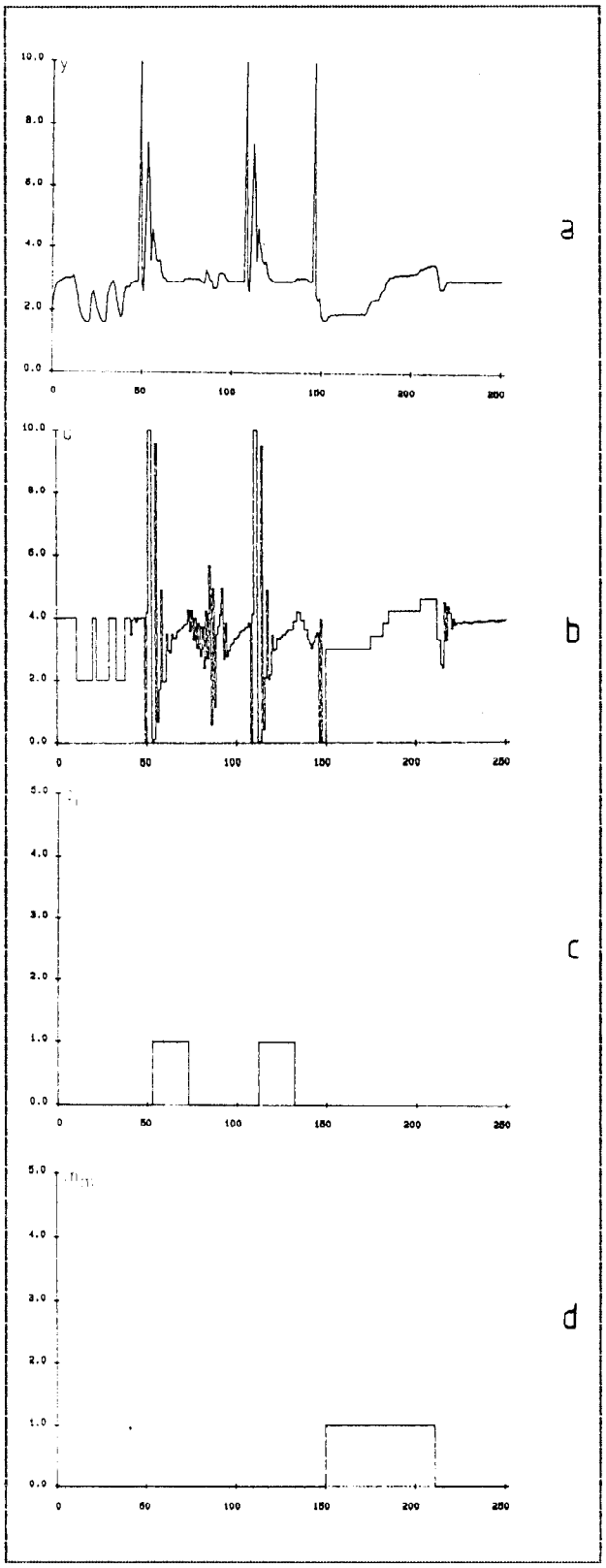


Fig 4.85. Actuación del controlador de respaldo.

se ejecutan, con lo que vendrá dado en función de los métodos de control, identificación o supervisión empleados, el orden del modelo del proceso, etc.

Como se puede deducir de las mediciones efectuadas, para el rango de aplicaciones en que se va autilizar el sistema desarrollado, lo anterior no afectará apreciablemente al funcionamiento del mismo, al ser el tiempo de cálculo muy pequeño en relación con el período de muestreo. Para ello, las operaciones aritméticas en coma flotante son llevadas a cabo por hardware, mediante coprocesador matemático.

En efecto, para un modelo del proceso de primer orden y utilizando control fijo con DBN, el tiempo necesario para efectuar la conversión A/D, realizar los cálculos previos a la conversión D/A y efectuar ésta, es de t_{F1}=1.5 ms. En el mismo caso, pero con control adaptativo será necesario t_{A1}=2.5 ms. En estas mediciones no se ha tenido en cuenta el tiempo necesario para la representación gráfica o numérica, dependiente del número de variables a representar, forma de efectuarla, etc., ni el empleado por los mecanismos de supervisión, dependiente de cuáles se utilicen.

Si se trabaja con modelos del proceso de orden superior, el tiempo empleado será mayor, por ejemplo para orden dos se tendrá t_{F2}=1.9 ms y t_{A2}=7.5 ms.

CAPITULO 5

CONCLUSIONES FINALES

5. CONCLUSIONES FINALES.

5.1. APORTACIONES DEL PRESENTE TRABAJO

En el presente trabajo se realiza un estudio de las técnicas de control adaptativo, los algoritmos necesarios para el diseño de controladores basados en las mismas y, en particular, los requerimientos necesarios para la implementación de los mismos en un sistema para trabajo en tiempo real, de forma que pueda ser ampliamente aplicado en la industria (en control, data logging, entrenamiento...) así como en la universidad (investigación, docencia ...).

El citado estudio se ha concretado en el desarrollo del sistema MASTER, en que se ha pretendido recoger la experiencia adquirida para subsanar las restricciones comúnmente encontradas en los paquetes de software de control adaptativo en cuanto a su aplicabilidad industrial, equiparándola a la de los controladores autónomos, sin perder flexibilidad ni perjudicar al resto de aplicaciones.

Así una aportación importante del presente trabajo consiste en la simplificación y automatización, en la medida de lo posible, de la fase de puesta en marcha del sistema, incorporando éste una fase de preidentificación que proporcione parámetros inicialmente necesarios. Se estudia así mismo la problemática presentada en el arranque del controlador adaptativo, viéndose la necesidad de

incorporar una estimación previa de parámetros en el mismo.

Se han definido así mismo los requerimientos a cumplir por un interface con el usuario adecuado, desarrollándose un sistema de menús arborescente para servir durante la fase de configuración así como un intérprete de comandos, mucho más apropiado para trabajar, de forma interactiva o batch, durante el funcionamiento en tiempo real.

Se definen en el capítulo 2 las funciones, y algoritmos apropiados para cumplirlas, de un módulo supervisor que permita el trabajo del sistema de forma continua (24 horas) y en condiciones "difíciles", por ejemplo con procesos de fase no-mínima, inestables, con cambios en la dinámica del mismo, etc. En particular se trata el caso de procesos con tiempo muerto desconocido o variante.

En el capítulo 3 se estudian detalladamente todos los requerimientos que debe cumplir un lenguaje de programación adecuado en la implementación de sistemas para trabajo en tiempo real, justificando la elección de Modula-2 en este caso.

Se estudia así mismo en el capítulo 3 el resto de requerimientos a tener en cuenta en la implementación, como son la necesidad de mecanismos anti-reset windup, de filtrado de perturbaciones y outliers ..., que se implementarán en el sistema desarrollado con los efectos que se muestran en el capítulo 4.

Es de resaltar que todas las opciones son seleccionables por software, sin necesidad de modificar el hardware. Esto es importante en el caso del filtrado analógico, en que se disponen mecanismos adicionales para no necesitar cambiar el filtro implementado físicamente cuando se varía el período de muestreo.

Para incrementar el rango de aplicabilidad industrial del sistema desarrollado se incluyen en el mismo los mecanismos adecuados para adquisición de datos vía DMA, gestión de alarmas o control manual. También se posibilita el uso del mismo para ajuste óptimo de los parámetros del controlador y aplicación de control fijo posteriormente. Así mismo la implementación está basada en un compatible PC, dotado del interface con el proceso adecuado, de forma que el sistema desarrollado será de bajo coste.

En el capítulo 4, en orden a evaluar la aplicabilidad industrial del sistema, se muestran los resultados de la aplicación del mismo a la placa de pruebas especialmente diseñada para este fin. Así mismo, el funcionamiento del sistema ha sido comprobado mediante la aplicación del mismo a una planta, altamente no lineal, conectada físicamente al computador, obteniéndose los resultados que se detallan en el apartado 4.7.

Es de resaltar que se ha desarrollado un sistema modular y flexible para posibilitar un rango de aplicaciones diverso, así como que el mismo haya dado lugar a la presentación de varias ponencias en congresos internacionales (López et al (1988), García et al

(1988)). Para evaluar su aplicabilidad en investigación o docencia se efectúa en el capítulo 4 un estudio comparativo de los resultados de su aplicación con varios algoritmos de control, apropiados para el seguimiento de una señal de consigna.

Estos algoritmos han sido definidos en el capitulo 2. En particular se han adaptado para el trabajo en tiempo real los métodos EPSAC de De Keyser et al. (permitiendo su aplicación sin una consigna preespecificada) y APR de Soh et al., en el que se han realizado varias aportaciones originales en la presente tesis como son:

- Transformación del método, basado en una minimización no lineal con restricciones en otro lineal, en ciertas condiciones.

- Definición y uso de un conjunto de regiones, de más a menos restrictivas, para la situación de los polos en cadena cerrada, en orden a obtener la menor en que el problema es factible.

- Uso del controlador robusto con estructura completa, con lo que se mejora el comportamiento del mismo.

5.2. SUGERENCIAS PARA FUTUROS TRABAJOS.

Como posibles desarrollos futuros de la presente tesis se proponen los siguientes:

- Implementación y verificación en el sistema desarrollado de

nuevos métodos de control, identificación y/o supervisión, sirviéndose para ello de las herramientas software y hardware existentes en el mismo.

- Aplicación del estudio y desarrollo realizados en la implementación de un sistema de control multivariable basado en las técnicas de control adaptativo, estudiando la problemática que se plantea.

- Implementación de un sistema distribuido de control adaptativo en tiempo real, que incorpore uno o varios DSPs (Procesadores Digitales de Señal), de forma que se aproveche la elevada velocidad de procesamiento de los mismos para la realización de diversas funciones de control, supervisión, etc., descargando de estas tareas al procesador principal.

- Aplicación de las técnicas de inteligencia artificial al sistema implementado, de forma que sumando la experiencia de especialistas en procesos, ingenieros de control, operadores, etc., se puedan emplear las técnicas de control adaptativo en tiempo real en el marco de un sistema experto.

REFERENCIAS

ARACIL, R. y A. JIMENEZ (1982) "Sistemas discretos de control", Departamento de publicaciones de la E.T.S.I.I. de Madrid.

ASTROM, K. J. y P. EYKHOFF (1971) "System identification-A survey", AUTOMATICA, Vol.7, pp.123-162.

ASTROM, K. J. y B. WITTENMARK (1980) "Self-tuning controllers based on pole-zero placement", IEE PROCEEDINGS, Vol. 127, Pt.D, No.3-MAY.

ASTROM, K. J. (1983) "Theory and applications of adaptive control-A survey", AUTOMATICA, Vol. 19, No.5, pp.471-486.

ASTROM, K. J. y B. WITTENMARK (1984) "Computer controlled systems, theory and desing", PRENTICE-HALL, information and system sciences series.

ASTROM, K. J. (1987) "Adaptive feedback control", IEE PROCEEDINGS, Vol.75, No.2, FEBRUARY-87.

BERGMANN, S. (1982) "Some improvements for the application of digital parameter-adaptive control algorithms", IEEE Conference on Applications of Adaptive and Multivariable Control, Hull, GB.

BERGMANN, S. (1983) "Digital parameter-adaptive control with microcomputers", Darmstadt. Tesis Doctoral.

BOITE, R. (1983) "On the quantization of low-level signals:the fixed-point case", Proc. Eur. Conf. Cct Theory and Design, Stuttgart.

BOOTH, A. D. (1951) "A signed binary multiplication technique", Q.J. Mech. Appl. Math, 4, 236.

CAVALCANTI, J. H. F., G. S. DEEP y J. G. A. LIRA (1988) "Real time executive for digital signal processor employed in process control", IECON'88, pp.397-402.

CLARKE, D. W. y P. J. GAWTHROP (1975) "Self-tuning controller", IEE PROCEEDINGS, Vol.122, No.9, pp.929-934.

CLARKE, D.W. y P.J. GAWTHROP (1981) "Implementation and application of microprocessor-based self-tuners", AUTOMATICA, Vol.17, No.1, pp.233-244.

DE KEYSER, R. M. C. y A. VAN CAUWENBERGHE (1979) "A self-tuning predictor as operator guide", 5th IFAC Symp. on Identification and System Parameter Estimation, Darmstadt.

DE KEYSER, R. M. C. y A. VAN CAUWENBERGHE (1981) "A Self-tuning multistep predictor application", AUTOMATICA, Vol.17, pp.167-174.

DE KEYSER, R. M. C. (1983) "Simple adaptive controller for systems

with fast parameter variations", ACI-83, pp.26-1/26-6.

DE KEYSER, R. M. C., Ph. G. A. VAN DE VELDE y F. A. G. DUMORTIER (1985) "A comparative study of self-adaptive long-range predictive control methods", 7th IFAC Symp. on Identification and System Parameter Estimation, York, U.K.

DE KEYSER, R. M. C. y A. R. VAN CAUWENBERGHE (1985) "Extended prediction self-adaptive control", 7th IFAC Symp. on Identification and System Parameter Estimation, York, pp.1255-1260.

DE KEYSER, R. M. C. (1986a) "Adaptive dead-time estimation", 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, Sweden, pp.299-213.

DE KEYSER, R. M. C. (1986b) "Personal computer training software for adaptive control", Journal A., Vol. 27. No.3, pp.155-161.

DE KEYSER, R. M. C., G. A. VAN DE VELDE y F. A. G. DUMORTIER (1988) "A Comparative study of self-adaptive long-range predictive control methods", AUTOMATICA, Vol.24, No.2, pp.149-163.

DE LA CRUZ, J. M., S. DORMIDO, J. M. GUILLEN y H. RUIPEREZ (1984) "Algoritmos adaptativos para control industrial con protecciones de seguridad", II Simposio Nacional sobre Automática en la Industria, Zaragoza.

DE LA PUENTE, J. A. y A. CRESPO (1984) "Programación sistemática para control industrial", II Simposio Nacional sobre Automática en la Industria, Zaragoza, pp.315-321.

DUFF, J., B. DWOLATZKY, S.P. SANOFF y P.E. WELLSTEAD (1985) "Hardware and software for self-tuning control", IEE CONTROL'85, pp.85-89.

ELMQVIST, M. y S. E. MATTSSON (1982) "Implementation of real-time facilities in Pascal", III IFAC/IFIP Symposium Software for Computer Control, pp.23-28.

EVANS, R. J. y X. XIANYA (1985) "Robust controller design", Int. J. Control, 41, pp.461-476.

FAVIER, G. (1987) "Computationally efficient adaptive identification algorithms", IEEE/ICASSP, Dallas, April 87.

FORTESCUE, T. R., L. KERSHENBAUM y B. E. YDSTIE (1981) "Implementation of self-tuning regulators with variable forgetting factors", AUTOMATICA, Vol.17, No.6, pp.831-835.

GARCIA, D. F., H. LOPEZ, A. DIEZ y G. OJEA (1987) "Aplicabilidad industrial de los controladores adaptativos, análisis comparativo de diversos algoritmos", Second World Basque Congress, Bilbao.

GARCIA, D., H. LOPEZ, J. TUYA y A. DIEZ. (1988) "Specification,

desing and Modula-2 implementation of a low cost industrial control system", IFAC/IFIP workshop on Real-Time Programming. Valencia.

GARCIA, D. (1988) "Control adaptativo de procesos monovariables y multivariables. Evaluación y diseño de controladores autoajustables de aplicación industrial generalizada". E.T.S.I.I. de Gijón. Tesis doctoral.

GERTLER, J. y SEDLAK, J. (1975) "Software for process control-A survey", AUTOMATICA, Vol.11, pp.613-625.

GOODWIN, G. C. y K. S. SIN (1984) "Adaptive filtering prediction and control", PRENTICE-HALL, information and system sciences series.

HAGGLUND, T. (1984) "Adaptive control of systems subjet to large parameter changes", 9th IFAC Word Congress, Budapest, Hungary.

HANSELMANN, M. y W. LOGES (1984) "Implementation of very fast state-space controllers using digital signal processors" 9th IFAC Wld Congr.

HANSELMANN, M. (1987) "Implementation of digital controllers-A survey", AUTOMATICA. Vol.23, No.1, pp.7-32.

HSIA, T. C. (1977) "System identification, least-squares methods", LEXINGTON BOOKS.

ISERMAN, R. (1980) "Practical aspects of process identification", AUTOMATICA, Vol.16, pp.575-587.

ISERMANN, R. (1981) "Digital control systems", SPRINGER-VERLAG.

ISERMANN, R. (1982) "Parameter adaptive control algorims-A tutorial", AUTOMATICA, Vol.18, No.5, pp.513-528.

ISERMANN, R. y K.-H. LACHMANN (1982) "On the development and implementation of parameter-adaptive controllers", IFAC-SOCOCO'82, Madrid.

ISERMANN, R. y K.-H. LACHANN (1985) "Parameter-adptive control with configuration aids and supervision functions", AUTOMATICA, Vol.21, No.6, pp.625-638.

ISERMAN, R. (1985) "Parameter-adaptive control systems-A review on methods and application", 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, York, UK, pp.373-378.

JONES, R. W. y R. P. D. BAKER (1986) "Self-tuning control:towards industrial viability", IFAC Low Cost Automation, Valencia.

KUCERA, V. (1979) "Discrete linear control. The polynomial equation approach", JOHN WILEY & SONS.

KUO, B. C. (1980) "Digital control systems", HOLT, RINEHART AND WINSTON, series in electrical and computer engineering.

KURZ,H. (1979) "Digital parameter-adaptive control of processes with unknown constant or time-varying dead time", 5th IFAC Symposium on Identification and System Parameter Estimation. Darmstadt, pp.1187-1193.

KURZ, H., R. ISERMANN y R. SCHUMANN (1980) "Experimental comparison and application of various parameter-adaptive control algorithms", AUTOMATICA, Vol.16, pp.117-133.

KURZ, H. y W. GOEDECKE (1981) "Digital paramater-adaptive control of processes with unknown dead time", AUTOMATICA, Vol.17, No.1, pp.245-252.

LEE, K. B. y LIM, K. G. (1988) "Control-calc: A versarile control and data acquisition software package", IECON'88, pp.907-911.

LEMMENS, W. J. M. y VAN DEN BOOM, A. J. W.(1979) "Interactive computer programs for education and research: A survey", AUTOMATICA Vol.15, pp.113-121.

LOPEZ, H., D.GARCIA, A. DIEZ, y G. OJEA. (1988) "SCM: A modular system for real time control", International conference on industrial Electronics,Control and Instrumentation (IECON'88). Singapur.

MAIER. G. (1982) "Modeb: A real time operating system kernel written in the high level programming language Modula-2", III IFAC/ IFIP Symposium Software for Computer Control, pp.15-21.

M'SAAD, M. ,M. DUQUE y D. LANDAU (1986) "On the applicability of adaptive control", 2nd IFAC workshop on adaptive control and signal processing, Lund, Sweden.

OJEA, G. (1981) "Diseño automático de sistemas de control multivariables, influencia de la estructura y del modelo", E.T.S.I.I. de Madrid. Tesis Doctoral.

PETERKA, V. (1984) "Predictor-based self-tuning control", AUTOMATICA, Vol.20, pp.39-50.

PLESSMANN, M. W. y H. ZOLLER (1986) "High-level-language programming in the field of process control", IECON'86, pp.665-670.

POPESCU, Th. y V. SIMA (1893) "Experiments with an adaptive control package", ACI-83, pp.21-34/21-39.

QUEVEDO, J. (1986) "Control adaptativo de procesos industriales", Mundo Electrónico, No.163.

SAELID, S. y B. FOSS (1983) "Adaptive controllers with a vector variable forgetting factors", 2nd Conference on Decision and Control, San Antonio, Texas, pp.1488-1494.

SAELID, S., O. EGELAND y B. FOSS (1985) "A solution to the blow-up problem in adaptive controllers", MODELLING, IDENTIFICATION AND CONTROL, Vol.6, No.1, pp.39-56.

SANZ R. y A. OLLERO (1984) "Conjunto de programas para el diseño de controladores autoajustables", II Simposio nacional sobre Automática en la Industria. Zaragoza.

SCHUMANN, R., K.-H LACHMANN y R. ISERMANN (1981) "Towards aplicability of parameter-adaptive control algorithms", 8th IFAC World Congress, Kyoto, Japan.

SIMA V., Th. D. POPESCU (1982) "ADCON-Adaptive control package", IFAC-SOCOCO'82, Madrid, pp.195-201.

SODERSTROM, T. (1977) "On model structure testing in system identification, INT. J. CONTROL, Vol.26, No.1, pp.1-18.

SOH, Y. C., R. J. EVANS, I. R. PETERSEN y R. E. BETZ (1987) "Robust pole assignment", AUTOMATICA, Vol.23, No.5 pp. 601-610

TUFFS, P. S. (1985) "Self-tuning control:algorithms and applications", Oxford University, OUEL Report No.1567/85, D.Phil Thesis.

TUFFS, P.S. y D. W. CLARKE (1985) "FAUST: A software package for self-tuning control", IEE CONTROL'85, pp.186-191.

VAN CAUWENBERGHE, A. R. y R. M. C. DE KEYSER (1985) "Self-adaptive long-range predictive control", American Control Conference, Vol II, pp.1155-1160.

WASER, S. H. y M. J. FLYNN (1982) "Introduction to arithmetic for digital systems designers", CBS College Publishing, New York.

WIRTH, N. (1985) "Programming in Modula-2", SPRINGER-VERLAG.

WITTENMARK, B. y K. J. ASTROM (1982) "Implementation aspects of adaptive controllers and their influence on robustness", 21st CDC, Orlando.

WITTENMARK, B. y K. J. ASTROM (1984) "Practical issues in the implementation of self-tuning control", AUTOMATICA, Vol.20, No.5, pp.595-605.

WONG, K. Y., M. M. BAYOUMI y S. NUYAN (1983) "Comparative study of some estimation algorithms for the self tuning control of time varying systems, ACI-83, pp.18-12/18-18.