

**UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**CONTROL ADAPTIVO MULTIVARIABLE DE UNA
COLUMNA DE DESTILACIÓN CON MÚLTIPLES
TIEMPOS MUERTOS**

INFORME DE SUFICIENCIA

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO ELECTRÓNICO**

PRESENTADO POR:

OSCAR JACINTO NAVARRO RAMOS

**PROMOCIÓN
1977-2**

**LIMA – PERÚ
2002**

A mi Familia
Especialmente a mis Hijos

**CONTROL ADAPTIVO MULTIVARIABLE DE UNA
COLUMNA DE DESTILACIÓN CON MÚLTIPLES
TIEMPOS MUERTOS**

INDICE

PRÓLOGO	1
CAPÍTULO I	
FORMULACIÓN DEL PROBLEMA	2
1.1 Antecedentes	2
1.2 La Columna de Destilación	2
1.3 Procesos Multivariantes Cuadrados	6
1.4 El Sistema de Control Adaptivo	6
1.5 Especificaciones de Diseño	7
CAPÍTULO II	
MODELADO DEL PROCESO	8
2.1 Dinámica del Proceso	8
2.2 Dinámica de los Tiempos Muertos	9
2.3 Modelado Aproximado del Proceso	12
2.4 Determinación del Modelo del Proceso	13
CAPÍTULO III	
EL CONTROLADOR MULTIVARIABLE DISCRETO	25
3.1 El Controlador SISO Óptimo Proporcional – Integral	25
3.2 El Regulador MIMO Óptimo Proporcional – Integral	25

CAPÍTULO IV

EL OBSERVADOR DE ESTADOS MULTIVARIABLE 30

4.1 ¿Porqué se Requiere un Observador? 30

4.2 El Observador Óptimo Cuadrático Multivariable 30

CAPÍTULO V

EL ESTIMADOR DE PARÁMETROS MULTIVARIABLE 33

5.1 Consideraciones Generales 33

5.2 El Método de Identificación de Parámetros IMCRM 33

5.3 Algunas Notas sobre el Método IIMCRM 35

CAPÍTULO VI

EL SISTEMA DE CONTROL ADAPTIVO 37

6.1 Procedimiento de Diseño 37

6.2 Simulación del Sistema de Control Adaptivo 39

CONCLUSIONES 42

APÉNDICE: LISTADO DE PROGRAMAS 46

BIBLIOGRAFÍA 61

SUMARIO

El presente estudio se ocupa del diseño y la simulación de un sistema de control adaptivo multivariable de una columna de destilación empleada para fraccionar petróleo crudo. Este sistema de control comprende una ley de control óptima cuadrática multivariable discreta, un algoritmo recursivo de estimación de parámetros y un observador óptimo discreto de estados.

El proceso a ser controlado, la columna de destilación para fraccionar petróleo crudo, ha sido empleado ampliamente por la comunidad técnica y académica para experimentar diferentes estrategias de control. En la literatura especializada, esta columna es conocida como la columna fraccionadora de petróleo crudo de Shell.

El modelo dinámico del proceso posee múltiples tiempos muertos, los cuales de hecho complican la realización del diseño del sistema de control adaptivo discreto. La filosofía de diseño empleada en este estudio, consiste en determinar un modelo lineal del proceso, pero con la capacidad de capturar los efectos de los tiempos muertos de la forma e^{-Ds} . Para ello, los tiempos muertos considerados en dicho modelo, serán reemplazados por aproximaciones racionales de bajo orden.

El modelo de múltiples entradas y múltiples salidas del proceso, se transforma luego a su forma canónica observable, la que dá lugar a un conjunto de ecuaciones de diferencias de múltiples entradas y una sola salida, las cuales son convenientes para propósitos de identificación de parámetros y diseño del observador y controlador multivariables del sistema. Mediante simulación, se va a demostrar que la metodología de diseño empleada, conduce a un sistema de control adaptivo multivariable de buen rendimiento, incluso, cuando todos los tiempos muertos del sistema varían considerablemente.

PRÓLOGO

El presente estudio trata sobre el diseño y la simulación de un sistema de control adaptivo multivariable, apropiado para controlar procesos multivariables, incluso, con múltiples tiempos muertos. El proceso a controlar es una columna de destilación multivariable que posee múltiples tiempos muertos, la cual se emplea en el fraccionamiento de petróleo crudo. El sistema de control comprende una ley de control óptima, un estimador multivariable de parámetros y un observador de estados óptimo multivariable. Este estudio se desarrolla en ocho (8) capítulos:

El capítulo I presenta los antecedentes respecto al proceso a controlar, describe brevemente dicho proceso y el sistema de control adaptivo a diseñar, y presenta las especificaciones de diseño del sistema de control.

El capítulo II trata sobre el modelo dinámico del proceso con múltiples tiempos muertos. También presenta el modelado de los tiempos muertos usando funciones de transferencia racionales, y culmina formulando la representación canónica observable del modelo del proceso.

El capítulo III abarca el diseño de la ley de control óptima cuadrática multivariable, con característica proporcional-integral.

El capítulo IV desarrolla el diseño del observador multivariable discreto empleado para estimar los estados del proceso, los cuales en su gran mayoría son no medibles.

El capítulo V presenta el estimador recursivo de parámetros. La técnica empleada es la de los mínimos cuadrados mejorado.

El capítulo VI emplea los resultados de los capítulos II al V para diseñar y simular el sistema de control adaptivo multivariable en estudio.

A continuación se presentan algunas conclusiones derivadas del presente trabajo.

El anexo de este estudio contiene el listado de los programas fuentes empleados. Tales programas están escritos en código MATLAB.

CAPITULO I

FORMULACIÓN DEL PROBLEMA

1.1 Antecedentes

El modelo matemático que describe la dinámica de la columna de destilación para fraccionar aceite pesado, se presentó originalmente en la referencia [7] y ha sido usado extensiva para probar diferentes estrategias de control. Una de dichas estrategias, el control predictivo multivariable de dicha columna, se describe en la referencia [1]. Recientemente, teniendo como base el modelo dinámico en cuestión, se ha presentado un informe de suficiencia que trata sobre el control óptimo cuadrático multivariable de la columna de destilación. No se ha reportado, al menos en el Perú, estudios conducentes al control adaptivo con autosintonización (este es el tipo de control adaptivo que usaremos) del proceso en cuestión. El proceso a controlar, la columna de destilación, es bastante particular debido a la presencia de múltiples tiempos muertos. Por esta razón es que en este estudio, al igual que se hizo en el trabajo referido al sistema de control óptimo del proceso, trataremos este tema con la debida amplitud.

1.2 La Columna de Destilación

La figura 1.1 muestra la denominada columna columna fraccionadora de petróleo crudo de Shell, cuyo modelo matemático ha sido usada ampliamente para experimentar diferentes estrategias y algoritmos de control.

Sabemos que el petróleo crudo contiene varias sustancias tales como gasolina, kerosene, nafta, gasóleo, etc., las cuales se pueden aprovechar de formas diferentes. Estas sustancias se obtienen fraccionando (separando) el petróleo crudo en una columna de destilación. El proceso de fraccionamiento de petróleo se basa fundamentalmente en los diferentes niveles de volatilidad que poseen las sustancias contenidas en el petróleo. Estas volatilidades están vinculadas de manera directa con el punto de ebullición de dichas sustancias, las cuales son diferentes

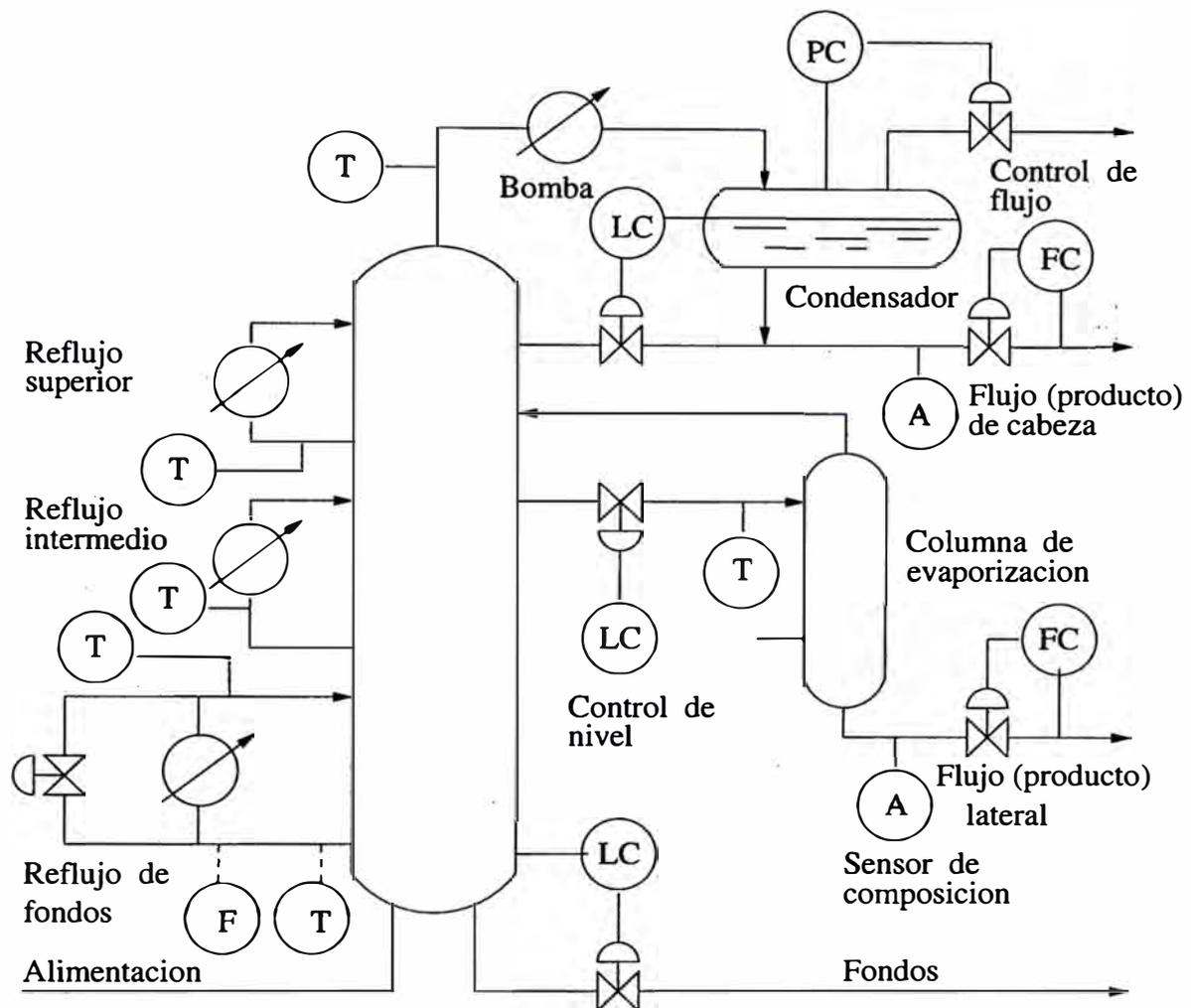


Figura 1.1: Columna de destilación para fraccionar petróleo crudo.

entre si. De esta forma, los productos más volátiles tienen un punto de ebullición más bajo que los productos menos volátiles.

la figura 1.1 muestra el esquema de la columna de destilación empleada en estudio. Podemos observar que el petróleo crudo ingresa por la parte inferior (línea de alimentación). Junto a esta línea de alimentación, se encuentra la línea de calefacción de la columna, la cual no aparece en la figura por simplicidad. Esta línea de calefacción proporciona todos los requerimientos caloríficos para la columna (principalmente el flujo de vapor necesario para producir el fraccionamiento). Esta columna muestra tres zonas bien demarcadas: superior, intermedia y de fondos. En la zona superior se encuentran los productos mas volátiles, mientras que en la zona intermedia, los menos volátiles. En la zona de fondos existe la mezcla de los productos volátiles de interés, para ser sometidos a fraccionamiento. En cada una de las zonas mencionadas, existen bombas de reflujo que tienen la finalidad de fraccionar nuevamente, mediante reciclaje, el producto semifraccionado. Esta operación permite aumentar la eficiencia de fraccionamiento de la columna.

En la parte superior de la columna emergen los productos más volátiles, los gases. Estos gases contienen vapores que se conducen a través de un condensador, el cual se emplea para recuperar parte del flujo de cabeza que se encuentra en estado líquido. La fuente principal del flujo de cabeza se alimenta por la parte lateral superior de la columna. Este flujo no pasa por el condensador.

Por la parte intermedia de la columna se obtiene el segundo producto de interés, el flujo lateral, el cual tiene menor volatilidad. Este producto se conduce a través de una pequeña torre de evaporización, para generar gases de mayor volatilidad. Estos gases se dirigen nuevamente a la columna para ser fraccionadas otra vez. Por consiguiente, la torre de evaporización mejora la pureza del producto lateral y permite también recuperar el producto de cabeza.

En la parte inferior de la columna encontramos la zona de fondos, que es básicamente en donde se encuentran el producto crudo en estado de volatilización. En esta zona se debe tener una temperatura constante para de esta manera asegurar una volatilización uniforme.

El control clásico de las columnas de destilación consiste en controlar independientemente cada variable controlada. Por ejemplo, el control de la composición del productos de cabeza se hace controlando el flujo de salida del mismo.

Del mismo modo, el control de la composición del producto lateral se realiza controlando el flujo de salida correspondiente, y, el control de la temperatura de fondos, se hace controlando el reflujo de fondos. Los controladores usados son normalmente los PID.

En el control clásico, y cuando la complejidad de una parte del proceso lo amerita, también se pueden emplear configuraciones multilazo, tales como cascada, relación de control, control de tres componentes, entre otras. Además de los lazos de control mencionados, con el fin de asegurar el óptimo funcionamiento de la columna, es necesario controlar en forma independiente los niveles de los productos fraccionados y acumulados tanto en el condensador de productos de cabeza como en la torre de evaporización (para los productos laterales). También es necesario controlar el nivel en la zona de fondos.

El control clásico de estas columnas de destilación puede perfeccionarse empleando estrategias de control multivariable. Es decir, una estrategia de control, cuyo algoritmo matemático tome en cuenta la interconectividad entre las entradas y las salidas seleccionadas, que realmente existen en la columna operando normalmente. La matriz de transferencia de una estrategia de control clásica, es diagonal. Esta matriz diagonal nos informa el carácter independiente de los controladores de la columna. Es decir, la fuerza de control u_1 controla la salida y_1 , la fuerza de control u_2 controla la salida y_2 , y así sucesivamente. La desventaja de esta estrategia es evidente: no se toma en cuenta la interrelación que realmente existe entre todas las entradas y todas las salidas. Por otra parte, la matriz de transferencia de una estrategia de control moderna, por lo general es no diagonal, debido a que se toma en cuenta las interrelaciones entre las variables de entrada y las variables de salida. Esta última es la estrategia que emplearemos en este estudio.

Las variables a controlar en la columna de destilación son: las composiciones del producto de cabeza y producto lateral (y_1 e y_2), los cuales se pueden medir empleando analizadores, y la temperatura de fondos (y_3). Las variables manipuladas (las fuerzas de control) son: el flujo de productos de cabeza (u_1), el flujo del producto lateral (u_2) y el reflujo de fondos (u_3).

1.3 Procesos Multivariables Cuadrados

Debemos enfatizar que el proceso en estudio es del tipo cuadrado. Es decir, en dicho proceso el número de entradas es igual al número de salidas. Bajo el punto de vista de control, un proceso puede ser controlado, siempre que el número de entradas independientes al sistema sea mayor o igual que el número de salidas. En esta situación, cuando el número de salidas es menor que el número de entradas, siempre es posible construir un sistema cuadrado que sea controlable. Basta con crear salidas ficticias para que el número de salidas iguale al número de entradas.

En cambio, si el número de entradas del sistema es menor que el número de salidas (estos sistemas se denominan subactuados), entonces dicho sistema en general es incontrolable. En esta situación no se pueden crear entradas ficticias. La filosofía de diseño de sistemas subactuados consiste en generar señales de control adicionales, las cuales pueden actuar, conforme a lo establecido en el diseño, simultáneamente o después que actúen las señales de control originales.

1.4 El Sistema de Control Adaptivo

Los sistemas de control adaptivo, ajustan su comportamiento a las cambiantes propiedades del proceso y de las señales que interactúan sobre dicho proceso. Por el contrario, en los sistemas de control fijos, la ley de control es invariable con el tiempo. La investigación sobre controladores adaptivos se ha centrado en dos grandes grupos: controladores adaptivos con modelo referencial y con autosintonización. En este trabajo se emplea la configuración de un sistema de control adaptivo con autosintonización mostrado en la figura 1.2 [9] (otras configuraciones se encuentran en [4], [2]). Tal sistema combina en su diseño un método de estimación de parámetros (el de los mínimos cuadrados recursivo mejorado), una técnica de estimación óptima de estados, una representación lineal del modelo del proceso, y una ley de control (el controlador óptimo proporcional-integral). El objetivo del control es determinar una fuerza de control, capaz de minimizar la diferencia entre la salida del proceso y la señal de referencia.

El sistema de control con autosintonización mostrado en la figura 1.2 opera como sigue: para cada tiempo de muestreo, el vector estimado de parámetros $\hat{\theta}$ se actualiza empleando los datos proporcionados por la entrada Uu del proceso y por la salida Yy del mismo. Luego, los elementos de $\hat{\theta}$ se usan para recuperar el

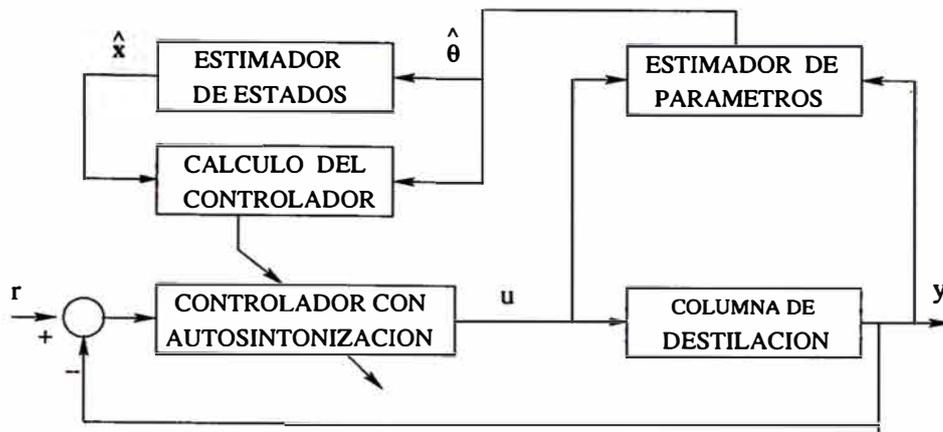


Figura 1.2: Configuración del sistema de control adaptivo con autosintonización.

modelo lineal del proceso, lo cual permite estimar el vector de estado del modelo del proceso \hat{x} . Tales resultados se usan luego para computar la ley de control u .

1.5 Especificaciones de Diseño

En este estudio se contempla el diseño de un sistema de control adaptivo multivariable, cuya configuración se muestra en la figura 1.2. Hemos mencionado en la primera sección de este capítulo que las variables a controlar en la columna de destilación son: las composiciones del producto de cabeza y producto lateral (y_1 e y_2) y la temperatura de fondos (y_3). Las variables manipuladas (las fuerzas de control) son: el flujo de productos de cabeza (u_1), el flujo del producto lateral (u_2) y el reflujo de fondos (u_3). El sistema de control adaptivo diseñado debe ser capaz de estabilizar las señales de salida y_1 , y_2 e y_3 en presencia de cambios tipo escalón de las señales referencia r_1 , r_2 y r_3 . Las señales de control u_1 , u_2 y u_3 no deben de sobrepasar el valor normalizado de ± 0.7 cuando ocurran los cambios tipo escalón de las señales de referencia. El tiempo de estabilización debe ser menor de 120 minutos para y_1 , y_2 e y_3 . El error en estado estable de las señales controladas debe ser nulo, con mínimo sobreimpulso.

CAPITULO II MODELADO DEL PROCESO

2.1 Dinámica del Proceso

El modelo dinámico que describe la dinámica de la columna de destilación para fraccionar petróleo pesado ha sido tomado de la referencia [1]. Este modelo dinámico, que considera tiempos muertos en su estructura, se describe como:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ y_3(s) \end{bmatrix} = \begin{bmatrix} \frac{K_{p11}e^{-D_{p11}s}}{1+T_{p11}s} & \frac{K_{p12}e^{-D_{p12}s}}{1+T_{p12}s} & \frac{K_{p13}e^{-D_{p13}s}}{1+T_{p13}s} \\ \frac{K_{p21}e^{-D_{p21}s}}{1+T_{p21}s} & \frac{K_{p22}e^{-D_{p22}s}}{1+T_{p22}s} & \frac{K_{p23}e^{-D_{p23}s}}{1+T_{p23}s} \\ \frac{K_{p31}e^{-D_{p31}s}}{1+T_{p31}s} & \frac{K_{p32}e^{-D_{p32}s}}{1+T_{p32}s} & \frac{K_{p33}}{1+T_{p33}s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ u_3(s) \end{bmatrix} \quad (2.1)$$

donde y_1 e y_2 son las composiciones de los productos de la parte superior y lateral de la columna (los cuales se miden con analizadores), y_3 es la temperatura del reflujo de fondos de la columna, la variable de control u_1 es el flujo superior, u_2 es el flujo lateral y u_3 es el reflujo de fondos. Los parámetros valorados del proceso son: $K_{p11} = 4.05$; $K_{p12} = 1.77$; $K_{p13} = 5.88$; $K_{p21} = 5.39$; $K_{p22} = 5.72$; $K_{p23} = 6.9$; $K_{p31} = 4.38$; $K_{p32} = 4.42$; $K_{p33} = 7.2$; $T_{p11} = 50$; $T_{p12} = 60$; $T_{p13} = 50$; $T_{p21} = 50$; $T_{p22} = 60$; $T_{p23} = 40$; $T_{p31} = 33$; $T_{p32} = 44$; $T_{p33} = 19$; $D_{p11} = 27$; $D_{p12} = 28$; $D_{p13} = 27$; $D_{p21} = 18$; $D_{p22} = 22$; $D_{p23} = 23$; $D_{p31} = 20$; $D_{p32} = 23$;

Este modelo matemático establece que la columna mostrada en la figura 1.1 posee tres variables que requieren ser controladas: la composición de los productos de la parte superior lateral de la columna, y la temperatura de fondos. Las señales de control correspondientes son: el flujo superior, el flujo lateral y el reflujo de fondos. Se supone que la línea de alimentación proporciona todos los requerimientos de calentamiento para la columna.

La respuesta a lazo abierto a una excitación escalón, del sistema en estudio, se obtiene ejecutando el programa `naval.m`, escrito en código MATLAB [3]. El listado de este programa se muestra en el Apéndice de este trabajo. La figura 2.1 muestra el resultado de la simulación. Cabe anotar que los listados de todos los

programas con extensión m desarrollados en este trabajo, se muestran también en el Apéndice.

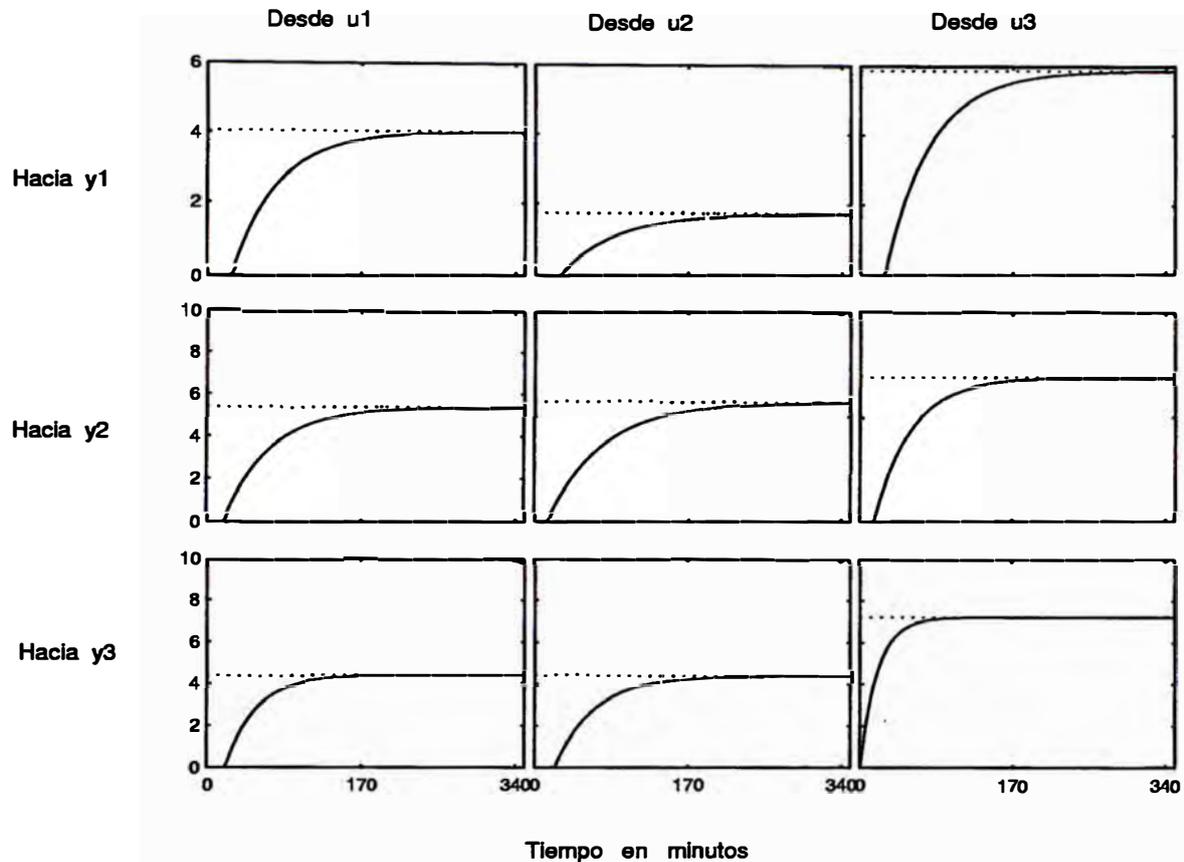


Figura 2.1: Respuesta a lazo abierto del proceso columna de destilación. Las entradas son escalones.

2.2 Dinámica de los Tiempos Muertos

La mayor parte de esta sección ha sido tomada de la referencia [8], con el permiso del autor para que sólo sea usado con fines académicos.

Es un hecho que todos los procesos industriales reales son no lineales y muchos de ellos poseen parámetros variables con el tiempo. Las curvas de datos obtenidas a partir de los sistemas reales mediante experimentación, sólo pueden representarse mediante complicados modelos matemáticos de orden n , donde n es un entero suficientemente grande. Por definición, el modelo de cualquier proceso dinámico no es el proceso actual, se puede entonces argumentar que siempre habrá errores de modelado. La experiencia dicta sin embargo, que puede ser posible tratar en forma efectiva con sistemas de considerable complejidad empleando modelos de orden $m < n$, donde m es un entero suficientemente pequeño.

Es también conocido que la presencia de tiempos muertos no despreciables en el proceso, pueden complicar el modelado y el diseño del sistema de control del proceso. Considere el sistema realimentado SISO (Single-Input-Single-Output) mostrado en la figura 2.2, donde $G(s)$ es una FT (función de transferencia) propia (el orden del polinomio del denominador es mayor o igual que el orden del polinomio del numerador) y D es el tiempo muerto. Para determinar la estabilidad de tal sistema, se requiere estimar el rango de valores de $D > 0$, tal que la relación $1 + G(s)e^{-sD} = 0$ (la ecuación característica del sistema) no posea ceros en el semiplano derecho del plano- s . Debido a la presencia de D , las raíces de tal ecuación son infinitas en número. Un método exacto para encontrar tales raíces requiere de un procedimiento gráfico o de una solución numérica. Evidentemente, la complejidad de la solución va a ser mayor en sistemas multivariables.

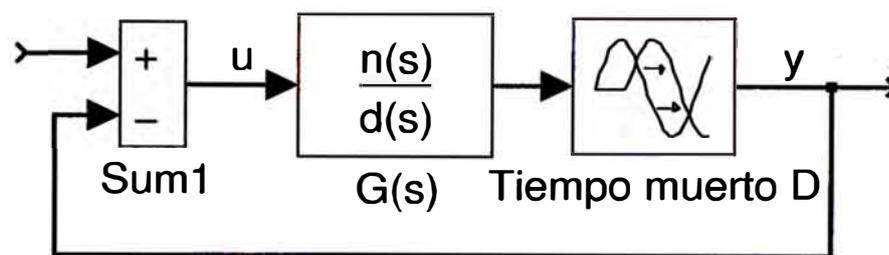


Figura 2.2: Simple sistema realimentado con la presencia del tiempo muerto D .

El problema principal que debemos enfrentar en el modelado de procesos con tiempos muertos, es encontrar un modelo del proceso que posea una estructura suficientemente simple, pero que tenga la capacidad de capturar las características significativas del proceso actual (con parámetros conocidos, o parcialmente conocidos) operando bajo la influencia de disturbios desconocidos. Por otra parte, para aplicaciones en el dominio del tiempo discreto, es necesario que el tiempo de computación del sistema de control sea menor que el tiempo de muestreo. Este requerimiento podría ser problemático de cumplir en algunas configuraciones de sistemas de control multivariable. Por consiguiente, la selección de un modelo del proceso que posea una simple estructura, es crucial.

A pesar de que el modelo seleccionado del proceso sea simple en su estructura, queda el problema de la inevitable presencia de tiempos muertos que no se pueden despreciar. En este trabajo, la filosofía general del método para modelar

sistemas multivariables que poseen múltiples tiempos muertos es aproximar el efecto de cada tiempo muerto, mediante una aproximación racional de orden dos. Este método permite obtener un modelo del proceso sin la presencia de tiempos muertos, pero con la capacidad de capturar significativamente las características dinámicas del proceso actual.

La idea de reemplazar la FT del tiempo muerto e^{-Ds} por una aproximación de bajo orden no es nueva. En varios trabajos dicha FT ha sido reemplazada empleando las aproximaciones de Padé, todo-polo, o pasa-todo de bajo orden. En [5], la aproximación de Padé de segundo orden se empleó para obtener representaciones de estado de la forma:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

en reemplazo de sistemas multivariables con múltiples tiempos muertos.

Con propósitos de comparación, describamos las aproximaciones. La FT $G_a(s)$ y la expresión de la fase $\phi_a(w)$ de una aproximación pasa-todo está dada por:

$$G_a(s) = \frac{(1 - sD/2n_a)^{n_a}}{(1 + sD/2n_a)^{n_a}} \quad (2.2)$$

$$\phi_a = -2n_a \arctan(wD/2n_a) \quad (2.3)$$

donde D es el tiempo muerto, n_a es el orden de la aproximación, $s = jw$ es el operador Laplaciano, y w es la frecuencia angular. La FT y la expresión de la fase de una aproximación de Padé de segundo orden está dada por:

$$G_{p2}(s) = \frac{1 - Ds/2 + (Ds)^2/12}{1 + Ds/2 + (Ds)^2/12} = \frac{num(s)}{den(s)} \quad (2.4)$$

mientras que la FT de una aproximación de Padé de tercer orden se describe como sigue:

$$G_{p3} = \frac{1 - Ds/2 + (Ds)^2/10 - (DS)^3/120}{1 + Ds/2 + (Ds)^2/10 + (DS)^3/120} = \frac{num(s)}{den(s)} \quad (2.5)$$

Observar que las aproximaciones de primer orden de Padé y pasa-todo son iguales. Todas las aproximaciones racionales poseen una ganancia unitaria, la cual es también la ganancia del tiempo muerto real. La figura 2.3 (que puede ser obtenida ejecutando el programa `tmuertos.m`) compara las respuestas en ángulo de las aproximaciones de segundo y tercer orden de Padé, y segundo orden pasa-todo con respecto a la fase real $-wD$ del tiempo muerto. La comparación se

ejecutó para ángulos entre 0 y 7 rad (i.e., entre 0 y 401°). Esta figura revela que:

1. Empleando aproximaciones de segundo orden de Padé o pasa-todo podemos acomodar retardos de fase de hasta 2 rad (i.e., 114°) que provienen de la FT e^{-sD} .
2. Incrementando el orden de la aproximación de Padé en uno, podemos acomodar retardos de fase de hasta 3.5 rad (i.e., 200°).

En este estudio, emplearemos aproximaciones de Padé de segundo orden para aproximar la FT e^{-sD} .

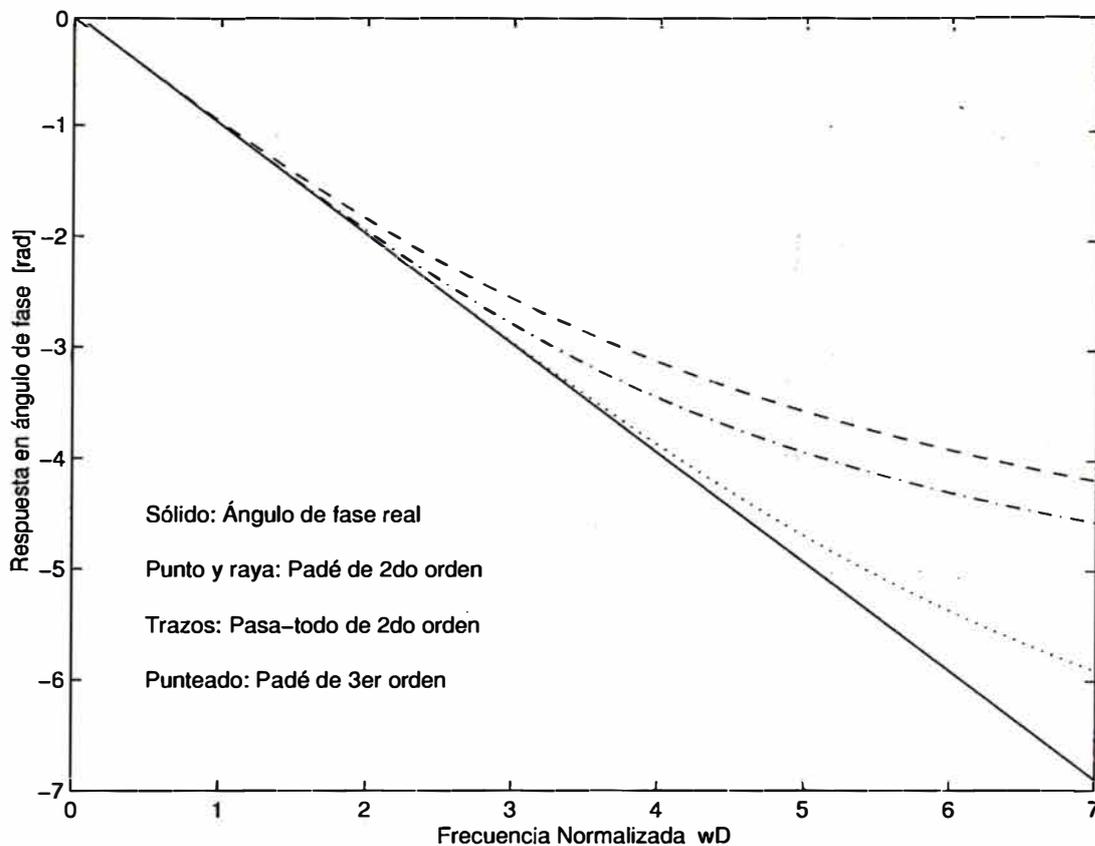


Figura 2.3: Comparación de las respuestas de fase de varias aproximaciones racionales de la FT e^{-sD} .

2.3 Modelado Aproximado del Proceso

Vimos que la matriz de transferencia del proceso dada en la ecuación (2.1), posee múltiples tiempos muertos de la forma e^{-Ds} . Para transformar esta matriz de transferencia en una ecuación de estado de la forma $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ (para así poder usar leyes de control y observadores de estado lineales), se requiere emplear

aproximaciones racionales de e^{-Ds} . Sabemos que si usamos una aproximación de Padé de segundo orden, la función de transferencia de e^{-Ds} es (ver (2.5)):

$$G_{p2}(s) = \frac{1 - Ds/2 + (Ds)^2/12}{1 + Ds/2 + (Ds)^2/12} = \frac{num(s)}{den(s)}$$

donde $num(s)$ es el polinomio del numerador mientras que $den(s)$ es el polinomio del denominador. Vale la pena preguntarnos qué tan efectiva es la FT $G_{p3}(s)$ para capturar los efectos del tiempo muerto real e^{-Ds} . La figura 2.4 muestra la respuesta del proceso columna de destilación con todos sus tiempos muertos reemplazados por aproximaciones racionales $G_{p3}(s)$. La figura 2.4 se obtiene ejecutando el programa `nava2.m`, cuyo listado se encuentra en el Apéndice. Si comparamos uno a uno las respuestas al escalón mostradas en las figuras 2.1 y 2.4, podemos afirmar que las respuestas al escalón del modelo aproximado del sistema columna de destilación (con los tiempos muertos reemplazados por aproximaciones de Padé de segundo orden), es bastante similar a las respuestas al escalón del modelo con tiempos muertos del sistema en estudio.

En adición a las repuestas al escalón, la figura 2.5 muestra las respuestas al impulso, tanto del proceso actual como del proceso aproximado. Dicha figura demuestra que las respuestas generadas por el proceso aproximado son muy similares a las respuestas generadas por el proceso original. La figura 2.5 se obtiene ejecutando el programa `impulso.m` (ver su listado en el Apéndice).

2.4 Determinación del Modelo del Proceso

El procedimiento para determinar la forma canónica observable del proceso se describe en la referencia [5]. Este procedimiento consiste en determinar primero el diagrama de bloques del modelo. Luego, se deduce la ecuación de estado continua del modelo, y se calcula su correspondiente ecuación de estado en el campo discreto. A continuación, partiendo de la descripción del modelo en el campo discreto, se determina la forma canónica controlable del modelo. Finalmente, la forma canónica observable, debe transformarse en una mínima representación MISO (Multiple-Input-Multiple-Output). Dicha forma canónica observable es la que se usa para diseñar el controlador multivariable del sistema de control adaptivo, así como también el observador y el estimador de parámetros multivariables de dicho sistema. Cada uno de estos pasos se detallan a continuación.

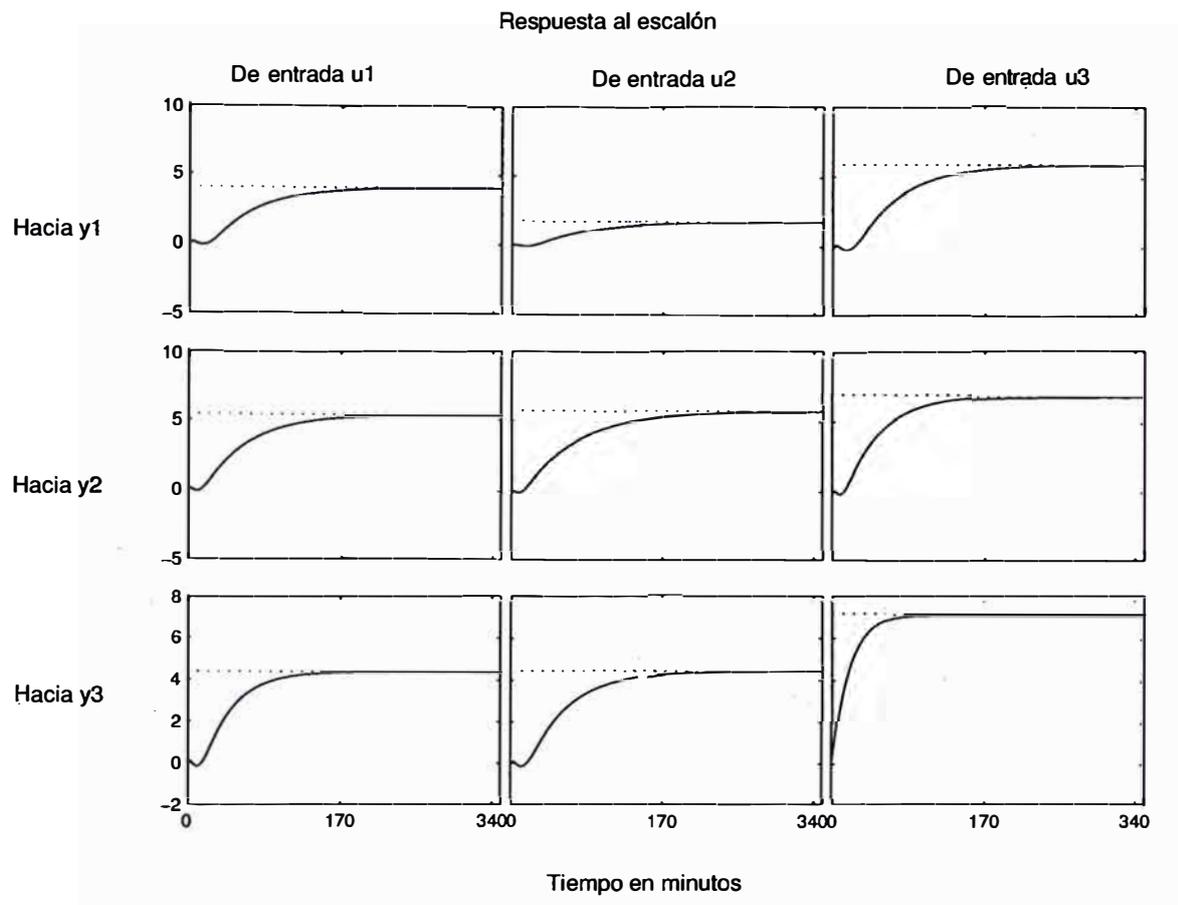


Figura 2.4: Respuesta a lazo abierto del proceso aproximado de la columna de destilación. Las señales excitadoras son escalones.

Diagrama de Bloques del Modelo y su Ecuación de Estado

El diagrama de bloques del proceso descrito por la ecuación 2.1 se muestra en la figura 2.6, mientras que el diagrama de bloques del modelo se muestra en la figura 2.7. Observar en esta última figura que el efecto de los 8 tiempos muertos del proceso está siendo capturado por tres aproximaciones de segundo orden, convenientemente ubicadas.

Para describir el modelo del proceso mostrado en la figura 2.7, se requieren quince variables de estado x_{ci} , $i = 1, \dots, 15$. La ecuación de estado del modelo se puede determinar fácilmente teniendo en cuenta que para x_{ci} , $i = 1, \dots, 9$, y para las combinaciones de los subíndices $(i,j,k)=(1,1,1)$, $(2,2,1)$, $(3,3,1)$, $(4,1,2)$, $(5,2,2)$, $(6,3,2)$, $(7,1,3)$, $(8,2,3)$, $(9,3,3)$ se cumple:

$$x_{ci} = \frac{K_{jk}}{T_{jk}s + 1} \quad \dot{x}_{ci} = -\frac{1}{T_{jk}}x_{ci} + \frac{K_{jk}}{T_{jk}}u_k \quad (2.6)$$

donde el subíndice c significa variable en el tiempo continuo. Las seis ecuaciones

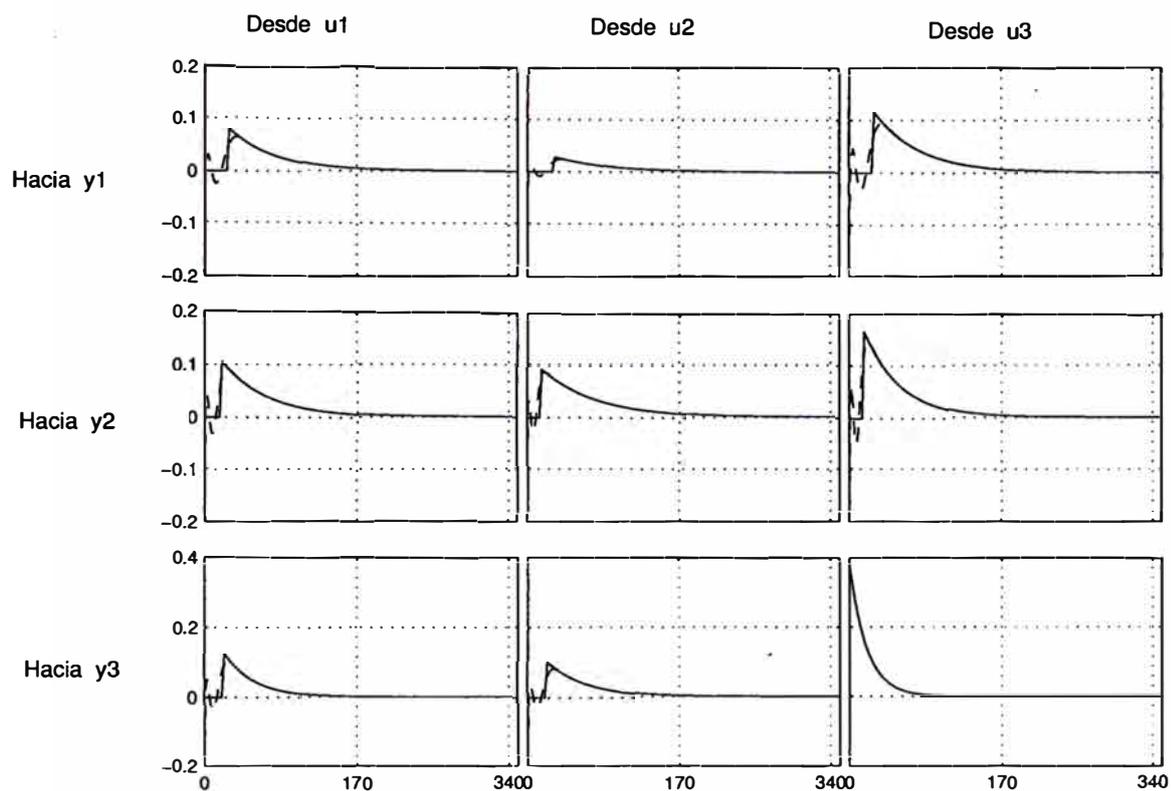


Figura 2.5: Respuesta al impulso del proceso columna de destilación (curvas sólidas) y de su modelo aproximado (curvas en trazos).

de estado restantes se generan para las siguientes combinaciones de los subíndices: $(s,r,q)=(1,0,1)$, $(3,2,2)$, $(5,4,3)$, de acuerdo a la estructura mostrada en la figura 2.8, como sigue:

$$\frac{x_{cls}}{w_q} = \frac{s^2 - a_{qq}s + b_{qq}}{s^2 + a_{qq}s + b_{qq}}$$

$$\ddot{x}_{cls} + a_{qq}\dot{x}_{cls} + b_{qq}x_{cls} = \ddot{w}_q - a_{qq}\dot{w}_q + b_{qq}w_q \quad (2.7)$$

Definiendo la siguiente ecuación de estado:

$$\dot{x}_{cls} = x_{clr} + \dot{w}_q \quad (s, r, q) = (1, 0, 1), (3, 2, 2), (5, 4, 3) \quad (2.8)$$

entonces:

$$\ddot{x}_{cls} = \dot{x}_{clr} + \ddot{w}_q \quad (2.9)$$

Reemplazando la ecuación (2.9) en (2.7) se obtiene:

$$\dot{x}_{clr} + a_{qq}\dot{x}_{cls} + b_{qq}x_{cls} = -a_{qq}\dot{w}_q + b_{qq}w_q$$

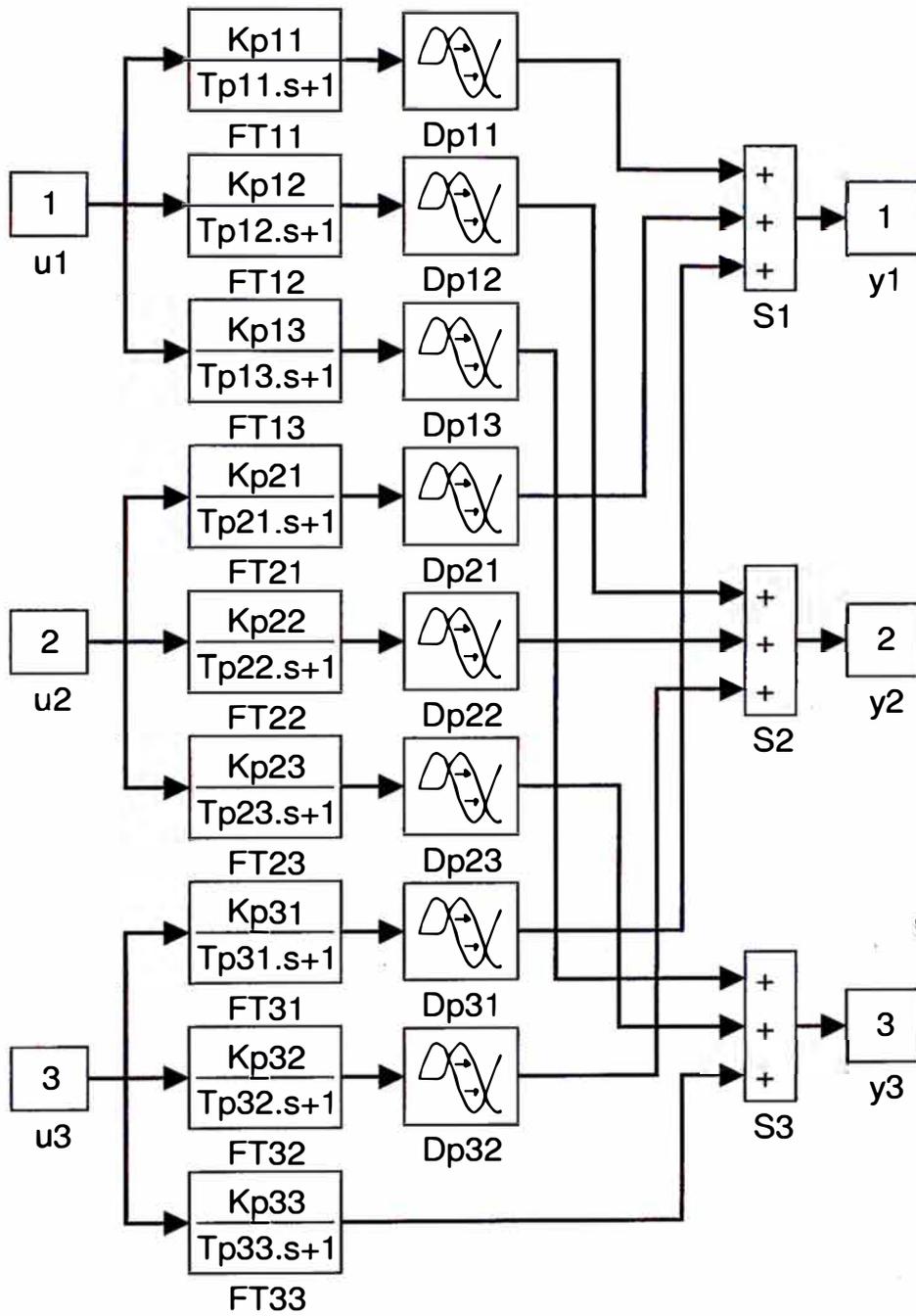


Figura 2.6: Diagrama de bloques del proceso.

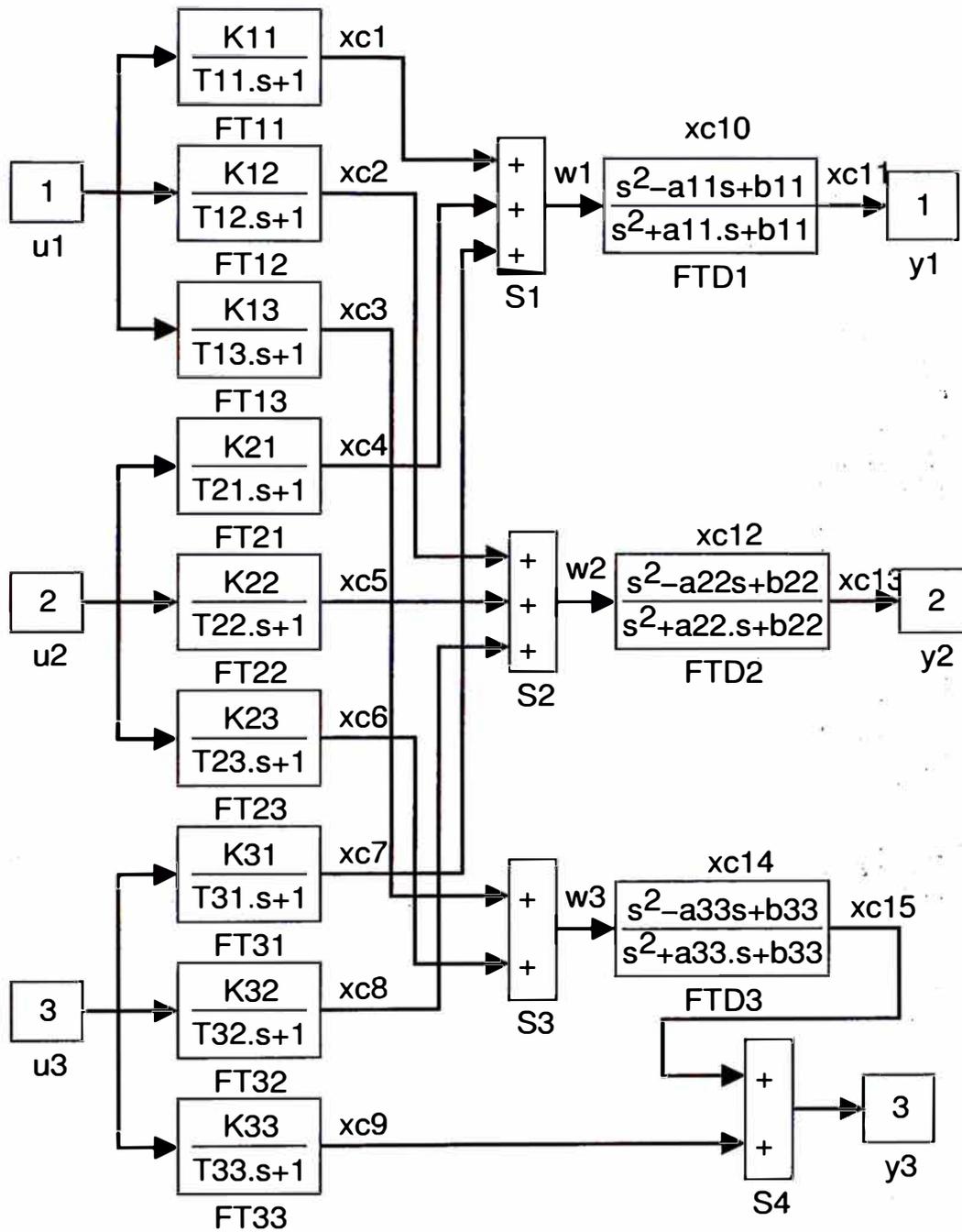


Figura 2.7: Diagrama de bloques del modelo.

Sustituyendo la relación (2.8) en esta última ecuación, y despejando se obtiene la siguiente ecuación de estado:

$$\dot{x}_{c1r} = -a_{qq}x_{c1r} - b_{qq}x_{c1s} + b_{qq}w_q - 2a_{qq}\dot{w}_q \quad (s, r, q) = (1, 0, 1), (3, 2, 2), (5, 4, 3) \quad (2.10)$$

Conviene recalcar que la ecuación (2.8) produce tres ecuaciones de estado. Las últimas tres ecuaciones de estado se derivan de la ecuación (2.10). Notar además en la figura 2.7 que:

$$w_1 = x_{c1} + x_{c4} + x_{c7} \quad w_2 = x_{c2} + x_{c5} + x_{c8} \quad w_3 = x_{c3} + x_{c6}$$

$$y_1 = x_{c11} \quad y_2 = x_{c13} \quad y_3 = x_{c15} + x_{c9}$$

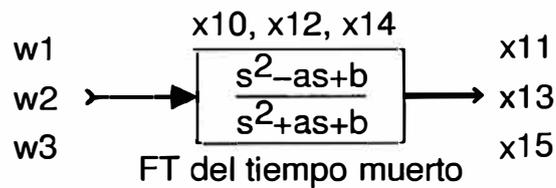


Figura 2.8: Aproximación de orden dos del tiempo muerto e^{-Ds} .

La ecuación de estado resultante es de la forma:

$$\dot{x}_c = A_c x_c + B_c u \quad y = C_c x_c + D_c u \quad (2.11)$$

donde los elementos no nulos A_{cij} de A_c poseen la forma:

$$A_{cii} = -\frac{1}{T_{ii}} \quad (i, i) = (1, 1), (2, 2), \dots, (9, 9)$$

$$A_{c10j} = b_{11} + \frac{2a_{11}}{T_{1q}} \quad (j, q) = (1, 1), (4, 2), (7, 3)$$

$$A_{c12j} = b_{22} + \frac{2a_{22}}{T_{2q}} \quad (j, q) = (2, 1), (5, 2), (8, 3)$$

$$A_{c14j} = b_{33} + \frac{2a_{33}}{T_{3q}} \quad (j, q) = (3, 1), (6, 2), (9, 3)$$

$$A_{c11j} = \frac{1}{T_{1q}} \quad (j, q) = (1, 1), (4, 2), (7, 3)$$

$$A_{c13j} = \frac{1}{T_{2q}} \quad (j, q) = (2, 1), (5, 2), (8, 3)$$

$$\begin{aligned}
A_{c15j} &= \frac{1}{T_{3q}} & (j, q) &= (3, 1), (6, 2) \\
A_{cii} &= -a_{qq} & (i, q) &= (10, 1), (12, 2), (14, 3) \\
A_{cij} &= -b_{qq} & (i, j, q) &= (10, 11, 1), (12, 13, 2), (14, 15, 3) \\
A_{cij} &= 1 & (i, j) &= (11, 10), (13, 12), (15, 14)
\end{aligned}$$

y las matrices \mathbf{B}_c y \mathbf{C}_c son:

$$\mathbf{B}_c = \begin{bmatrix}
\frac{K_{11}}{T_{11}} & 0 & 0 \\
\frac{K_{21}}{T_{21}} & 0 & 0 \\
\frac{K_{31}}{T_{31}} & 0 & 0 \\
0 & \frac{K_{12}}{T_{12}} & 0 \\
0 & \frac{K_{22}}{T_{22}} & 0 \\
0 & \frac{K_{32}}{T_{32}} & 0 \\
0 & 0 & \frac{K_{13}}{T_{13}} \\
0 & 0 & \frac{K_{23}}{T_{23}} \\
0 & 0 & \frac{K_{33}}{T_{33}} \\
-\frac{2a_{11}K_{11}}{T_{11}} & -\frac{2a_{11}K_{12}}{T_{12}} & -\frac{2a_{11}K_{13}}{T_{13}} \\
\frac{K_{11}}{T_{11}} & \frac{K_{12}}{T_{12}} & \frac{K_{13}}{T_{13}} \\
-\frac{2a_{22}K_{21}}{T_{21}} & -\frac{2a_{22}K_{22}}{T_{22}} & -\frac{2a_{22}K_{23}}{T_{23}} \\
\frac{K_{21}}{T_{11}} & \frac{K_{22}}{T_{22}} & \frac{K_{23}}{T_{23}} \\
-\frac{2a_{33}K_{31}}{T_{31}} & -\frac{2a_{33}K_{32}}{T_{32}} & -\frac{2a_{33}K_{33}}{T_{33}} \\
\frac{K_{31}}{T_{31}} & \frac{K_{32}}{T_{32}} & 0
\end{bmatrix}$$

$$\mathbf{C}_c = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$\mathbf{D}_c = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}$$

Teniendo la descripción de estado en el tiempo continuo (ecuación (2.11)), su correspondiente descripción de estado en el tiempo discreto para un tiempo

de discretización de T_s minutos, se obtiene sin dificultad empleando el siguiente comando de MATLAB:

$$[A_d, B_d, C_d, D_d] = c2dm(A_c, B_c, C_c, D_c, T_s, 'zoh')$$

donde T_s es el tiempo de muestreo en minutos y 'zoh' significa que se está empleando el criterio 'zero-order-hold' para recuperar la discretización. Entonces, la descripción del modelo del proceso en el campo discreto toma la forma:

$$\mathbf{x}_d(k+1) = \mathbf{A}_d \mathbf{x}_d(k) + \mathbf{B}_d \mathbf{u}(k) \quad \mathbf{y}(k) = \mathbf{C}_d \mathbf{x}_d(k) + \mathbf{D}_d \mathbf{u}(k) \quad (2.12)$$

donde k es el tiempo discreto definido como $k = t/T_s$, y:

$$\mathbf{x}_d(k) = \begin{bmatrix} x_{d1}(k) \\ x_{d2}(k) \\ \vdots \\ x_{d15}(k) \end{bmatrix} \quad \mathbf{y}(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} \quad \mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

Forma Canónica Observable del Modelo

La forma canónica observable del modelo dada en la ecuación (2.12), requiere que dicho sistema sea completamente controlable y completamente observable. Este requisito será verificado más adelante, conjuntamente con el diseño del sistema de control adaptivo. La forma canónica observable del modelo es como sigue:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2.13)$$

Esta representación posee un mínimo número de parámetros a ser estimado, y su estructura es conocida, lo cual permite recuperar la ecuación (2.13) luego del proceso de estimación de parámetros. Para determinar las matrices \mathbf{A} , \mathbf{B} y \mathbf{C} , ya que $\mathbf{D} = \mathbf{D}_d = \mathbf{D}_c = \mathbf{0}$ en nuestro caso, se requiere determinar la matriz de transformación de estados \mathbf{T}_d para $n = 15$, sabiendo que \mathbf{C}_d posee 3 filas:

$$\mathbf{C}_d = \begin{bmatrix} c_{d1} \\ c_{d2} \\ c_{d3} \end{bmatrix}$$

Teniendo en cuenta que existen tres salidas y tres entradas, entonces podemos asumir que tal matriz \mathbf{T} toma la forma:

$$\mathbf{T} = \begin{bmatrix} \mathbf{c}_{d1} \\ \mathbf{c}_{d1}\mathbf{A}_d \\ \mathbf{c}_{d1}\mathbf{A}_d^2 \\ \vdots \\ \mathbf{c}_{d1}\mathbf{A}_d^{n_1-1} \\ \mathbf{c}_{d2} \\ \mathbf{c}_{d2}\mathbf{A}_d \\ \mathbf{c}_{d2}\mathbf{A}_d^2 \\ \vdots \\ \mathbf{c}_{d2}\mathbf{A}_d^{n_2-1} \\ \mathbf{c}_{d3} \\ \mathbf{c}_{d3}\mathbf{A}_d \\ \mathbf{c}_{d3}\mathbf{A}_d^2 \\ \vdots \\ \mathbf{c}_{d3}\mathbf{A}_d^{n_3-1} \end{bmatrix} \quad (2.14)$$

donde $n_1 = 5$; $n_2 = 5$, $n_3 = 5$ y $n_1 + n_2 + n_3 = n = 15$, para obtener una mínima representación [8]. Conocida la matriz T , entonces:

$$\mathbf{A} = \mathbf{T}\mathbf{A}_d\mathbf{T}^{-1} \quad \mathbf{B} = \mathbf{T}\mathbf{B}_d \quad \mathbf{C} = \mathbf{C}_d\mathbf{T}^{-1} \quad (2.15)$$

La matriz \mathbf{A} posee la siguiente estructura con $r = 3$:

$$\mathbf{A} = \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & A_{rr} \end{bmatrix} \quad \dim[A_{ij}] = n_i n_j \quad (2.16)$$

$$A_{ii} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ & & & & \alpha_{ii} \end{bmatrix} \quad i = 1, 2, 3$$

$$A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ & & & & \alpha_{ij} \end{bmatrix} \quad (i, j) = (2, 1), (3, 1), (3, 2)$$

Los vectores fila α_{ij} se determinan empleando la fórmula:

$$\alpha_{ij} = [a_{m_i m_{j-1}+1} \quad \cdots \quad a_{m_i m_j}] \quad j \leq i \quad (2.17)$$

donde:

$$m_0 = 0; m_i = \sum_{k=1}^i n_k \quad i = 1, 2, r = 3 \quad (2.18)$$

Por consiguiente:

$$m_0 = 0 \quad m_1 = n_1 = 5 \quad m_2 = n_1 + n_2 = 10 \quad m_3 = \sum_{i=1}^3 n_i = 15$$

$$\alpha_{11} = a_{5,1} \quad a_{5,2} \quad a_{5,3} \quad a_{5,4} \quad a_{5,5}$$

$$\alpha_{21} = a_{10,6} \quad a_{10,7} \quad a_{10,8} \quad a_{10,9} \quad a_{10,10}$$

$$\alpha_{22} = a_{10,1} \quad a_{10,2} \quad a_{10,3} \quad a_{10,4} \quad a_{10,5}$$

$$\alpha_{31} = a_{15,1} \quad a_{15,2} \quad a_{15,3} \quad a_{15,4} \quad a_{15,5}$$

$$\alpha_{32} = a_{15,6} \quad a_{15,7} \quad a_{15,8} \quad a_{15,9} \quad a_{15,10}$$

$$\alpha_{33} = a_{15,11} \quad a_{15,12} \quad a_{15,13} \quad a_{15,14} \quad a_{15,15}$$

Las ecuaciones de diferencias (la representación MISO) correspondiente a la mínima representación canónica observable tienen la forma:

$$y_i(k + n_i) = \sum_{j=1}^i \alpha_{ij} \eta_j + \sum_{j=0}^{n_i-1} \beta_j^i u(k + j) \quad i = 1, 2, 3 = r \quad (2.19)$$

donde:

$$\eta_j \begin{bmatrix} y_i(k) \\ y_i(k+1) \\ \vdots \\ y_i(k+n_j-1) \end{bmatrix} \quad u(k+j) = \begin{bmatrix} u_1(k+j) \\ u_2(k+j) \\ \vdots \\ u_r(k+j) \end{bmatrix} \quad r = 3$$

$$\beta_j^i = b_{m_i-j} - \sum_{k=1}^i \sum_{f=1}^{n_k-j-1} a_{m_i, m_k-f-j} \quad j = 0, \dots, n_i - 1 \quad (2.20)$$

Por consiguiente, operando obtenemos:

$$\begin{aligned}
 y_1(k+5) &= \alpha_{11}\eta_1 + \beta_0^1 u(k) + \beta_1^1 u(k+1) + \beta_2^1 u(k+2) + \beta_3^1 u(k+3) \\
 &\quad + \beta_4^1 u(k+4) \\
 y_2(k+5) &= \alpha_{21}\eta_1 + \alpha_{22}\eta_2 + \beta_0^2 u(k) + \beta_1^2 u(k+1) + \beta_2^2 u(k+2) + \beta_3^2 u(k+3) \\
 &\quad + \beta_4^2 u(k+4) \\
 y_3(k+5) &= \alpha_{31}\eta_1 + \alpha_{32}\eta_2 + \alpha_{33}\eta_3 + \beta_0^3 u(k) + \beta_1^3 u(k+1) + \beta_2^3 u(k+2) \\
 &\quad + \beta_3^3 u(k+3) + \beta_4^3 u(k+4)
 \end{aligned} \tag{2.21}$$

Los vectores fila β_j^i tienen la forma:

$$\begin{aligned}
 \beta_0^1 &= b_5 - a_{5,5}b_4 - a_{5,4}b_3 - a_{5,3}b_2 - a_{5,2}b_1 \\
 \beta_1^1 &= b_4 - a_{5,5}b_3 - a_{5,4}b_2 - a_{5,3}b_1 \\
 \beta_2^1 &= b_3 - a_{5,5}b_2 - a_{5,4}b_1 \\
 \beta_3^1 &= b_2 - a_{5,5}b_1 \\
 \beta_4^1 &= b_1 \\
 \beta_0^2 &= b_{10} - a_{10,5}b_4 - a_{10,4}b_3 - a_{10,3}b_2 - a_{10,2}b_1 \\
 &\quad - a_{10,10}b_9 - a_{10,9}b_8 - a_{10,8}b_7 - a_{10,7}b_6 \\
 \beta_1^2 &= b_9 - a_{10,5}b_3 - a_{10,4}b_2 - a_{10,3}b_1 - a_{10,10}b_8 - a_{10,9}b_7 - a_{10,8}b_6 \\
 \beta_2^2 &= b_8 - a_{10,5}b_2 - a_{10,4}b_1 - a_{10,10}b_7 - a_{10,9}b_6 \\
 \beta_3^2 &= b_7 - a_{10,5}b_1 - a_{10,10}b_6 \\
 \beta_4^2 &= b_6 \\
 \beta_0^3 &= b_{15} - a_{15,5}b_4 - a_{15,4}b_3 - a_{15,3}b_2 - a_{15,2}b_1 \\
 &\quad - a_{15,15}b_{14} - a_{15,14}b_{13} - a_{15,13}b_{12} - a_{15,12}b_{11} \\
 &\quad - a_{15,10}b_9 - a_{15,9}b_8 - a_{15,8}b_7 - a_{15,7}b_6 \\
 \beta_1^3 &= b_{14} - a_{15,5}b_3 - a_{15,4}b_2 - a_{15,3}b_1 \\
 &\quad - a_{15,10}b_8 - a_{15,9}b_7 - a_{15,8}b_6 \\
 &\quad - a_{15,15}b_{13} - a_{15,14}b_{12} - a_{15,13}b_{11} \\
 \beta_2^3 &= b_{13} - a_{15,5}b_2 - a_{15,4}b_1 - a_{15,10}b_7 - a_{15,9}b_6 - a_{15,15}b_{12} - a_{15,14}b_{11} \\
 \beta_3^3 &= b_{12} - a_{15,5}b_1 - a_{15,10}b_6 - a_{15,15}b_{11} \\
 \beta_{34}^3 &= b_{11}
 \end{aligned}$$

La matriz **C** calculada con la ecuación (2.15) es:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

CAPITULO III

EL CONTROLADOR MULTIVARIABLE DISCRETO

3.1 El Controlador SISO Óptimo Proporcional-Integral

Con el propósito de construir el sistema de control adaptivo con autosintonización descrito en la sección 1.4, debemos deducir la ley de control óptima cuadrática multivariable discreta estacionaria. Tal ley de control va a ser derivada por extensión de la ley de control SISO (Single-Input-Single-Output) correspondiente.

En este estudio usaremos emplea la configuración del regulador óptimo proporcional-integral mostrado en la figura 3.1 [5], [9],[6] en donde podemos notar la presencia de un integrador discreto, empleado para para eliminar los errores en estado estable. Es decir, para hacer que la diferencia entre la salida $y(k)$ y la referencia $r(k)$ sea nula en el estado estable. En esta configuración, por ahora, tanto la señal de salida, como la señal de control (así como también la señal de referencia) son escalares. Por ello el regulador de la figura 3.1 es del tipo SISO (Single-Input-Single-Output).

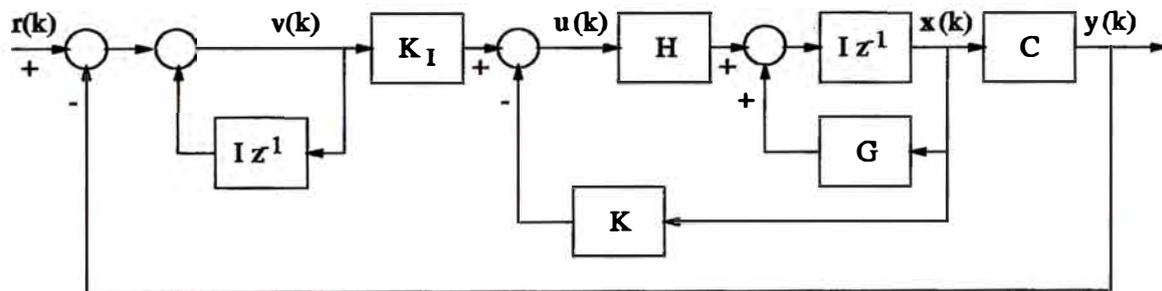


Figura 3.1: El regulador SISO óptimo proporcional-integral.

Las relaciones que gobiernan al regulador proporcional-integral se deducieron en el curso de Control Avanzado correspondiente al programa de titulación extraordinaria. Por claridad de la exposición, tales deducciones se presentan a

continuación. del curso De la figura 3.1 podemos escribir:

$$\mathbf{x}(k+1) = G\mathbf{x}(k) + Hu(k) \quad y(k) = C\mathbf{x}(k) \quad (3.1)$$

$$u(k) = -K\mathbf{x}(k) + K_I v(k) \quad (3.2)$$

donde $\mathbf{x}(k)$ es el vector de estado del sistema de orden n , $y(k)$ es la salida escalar controlada del sistema, $u(k)$ es la señal o fuerza de control, $v(k)$ es la señal de salida del integrador, G es la matriz de estado de orden $n \times n$, H es la matriz de control de orden $n \times 1$, C es la matriz de salida de orden $1 \times n$, K_I es la ganancia del integrador, y K es la matriz ganancia del controlador de orden $1 \times n$. La matriz de ganancia K posee la forma:

$$K = [K_1 \quad K_2 \quad \cdots \quad K_n] \quad (3.3)$$

Usando la figura 3.1 se formula la ecuación que describe al integrador discreto, como sigue:

$$v(k) = v(k-1) + r(k) - y(k) \quad (3.4)$$

En el tiempo $v(k+1)$ se tiene:

$$\begin{aligned} v(k+1) &= v(k) + r(k+1) - y(k+1) \\ &= v(k) + r(k+1) - C[G\mathbf{x}(k) + Hu(k)] \\ &= (1 - CHK_I)v(k) + (-CG + CHK)\mathbf{x}(k) + r(k+1) \end{aligned} \quad (3.5)$$

Utilizando (3.1) y (3.2), obtenemos:

$$\begin{aligned} \mathbf{x}(k+1) &= G\mathbf{x}(k) + H[-K\mathbf{x}(k) + K_I v(k)] \\ &= (G - HK)\mathbf{x}(k) + H K_I v(k) \end{aligned} \quad (3.6)$$

Las ecuaciones (3.5) y (3.6) producen:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k+1) \quad (3.7)$$

$$y(k) = [C \quad 0] \begin{bmatrix} \mathbf{x}(k) \\ v(k) \end{bmatrix} \quad (3.8)$$

En el estado estacionario (cuando $k \rightarrow \infty$), las variables $\mathbf{x}(k)$, $u(k)$ y $v(k)$ toman sus valores estacionarios $\mathbf{x}(\infty)$, $u(\infty)$ y $v(\infty)$, respectivamente, haciendo que la

ecuación (3.7) tome la forma:

$$\begin{bmatrix} \mathbf{x}(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} \mathbf{x}(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \quad (3.9)$$

Nosotros podemos definir:

$$\mathbf{x}(k) - \mathbf{x}(\infty) = \mathbf{x}_e(k) \quad (3.10)$$

$$v(k) - v(\infty) = v_e(k) \quad (3.11)$$

Restando (3.9) de (3.7), y luego usando las ecuaciones (3.10) y (3.11), obtendremos:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_e(k+1) \\ v_e(k+1) \end{bmatrix} &= \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ v_e(k) \end{bmatrix} \\ &= \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} \begin{bmatrix} -K & K_I \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ v_e(k) \end{bmatrix} \end{aligned} \quad (3.12)$$

La forma compacta de la ecuación (3.12) es:

$$\xi(k+1) = \tilde{G}\xi(k) + \tilde{H}w(k) \quad w(k) = -\tilde{K}\xi(k) \quad (3.13)$$

con:

$$\begin{aligned} \xi(k) &= \begin{bmatrix} \mathbf{x}_e(k) \\ v_e(k) \end{bmatrix} & \tilde{G}(k) &= \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix} \\ \tilde{H}(k) &= \begin{bmatrix} H \\ -CH \end{bmatrix} & \tilde{K}(k) &= \begin{bmatrix} K & -K_I \end{bmatrix} \end{aligned}$$

Tener en cuenta que el vector de estado $\mathbf{x}(k)$ del sistema original es de orden n , mientras que el vector de estado $\xi(k)$ del regulador SISO proporcional-integral es de orden $(n+1)$, debido a la presencia del integrador. Para resolver el problema del control óptimo cuadrático discreto estacionario, debemos minimizar la siguiente función de costo:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [\xi(k)^T \tilde{Q} \xi(k) + w^2(k) \tilde{R}] \quad (3.14)$$

Algunos métodos de minimización que pueden ser aplicados son, entre otros, los multiplicadores de Lagrange y el principio de optimalidad. En este estudio sólo nos vamos a referir a los resultados del proceso de minimización, el cual produce

las expresiones de la matriz de ganancia \tilde{K} del controlador y la ecuación de Riccati asociada; es decir:

$$\tilde{P} = \tilde{Q} + \tilde{G}^T \tilde{P} \tilde{G} - \tilde{G}^T \tilde{P} \tilde{H} [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (3.15)$$

$$\tilde{K} = [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (3.16)$$

donde \tilde{P} es una matriz simétrica definida positiva de dimensión $(n+1) \times (n+1)$ la cual es solución de la ecuación matricial de Riccati asociada.

3.2 El Regulador MIMO Óptimo Proporcional–Integral

La configuración del regulador óptimo proporcional–integral mostrado en la figura 3.1, también puede ser empleada para sistemas MIMO (Multiple–Input–Multiple–Output). Las ecuaciones que gobiernan la dinámica de un regulador MIMO proporcional–integral se pueden formular por extensión. Esto es, la ecuación de estado del sistema MIMO viene a ser (ver figura 3.1):

$$\mathbf{x}(k+1) = G\mathbf{x}(k) + H\mathbf{u}(k) \quad \mathbf{y}(k) = C\mathbf{x}(k) \quad (3.17)$$

$$\mathbf{u}(k) = -K\mathbf{x}(k) + K_I\mathbf{v}(k) \quad (3.18)$$

donde $\mathbf{x}(k)$ es el vector de estado del sistema de orden n , $\mathbf{y}(k)$ es el vector de salida de orden m , $\mathbf{u}(k)$ es el vector de control de orden m , $\mathbf{v}(k)$ es el vector de salida del integrador de orden m , G es la matriz de estado de orden $n \times n$, H es la matriz de control de orden $n \times m$, C es la matriz de salida de orden $m \times n$, K_I es la matriz de ganancia del integrador de orden $m \times m$, y K es la matriz ganancia del controlador de orden $m \times n$. La matriz de ganancia K posee ahora la forma:

$$K = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ \vdots & \vdots & \vdots & \\ K_{m1} & K_{m2} & \cdots & K_{mn} \end{bmatrix} \quad (3.19)$$

Del mismo modo, de la figura 3.1, podemos formular la ecuación que describe al vector integrador discreto:

$$\mathbf{v}(k) = \mathbf{v}(k-1) + \mathbf{r}(k) - \mathbf{y}(k) \quad (3.20)$$

donde \mathbf{r} es el vector de referencia de orden m . Operando como en el caso SISO, se puede demostrar que la forma compacta de la ecuación (3.12), pero para el

caso MIMO es (ver (3.21)):

$$\xi(k+1) = \tilde{G}\xi(k) + \tilde{H}\mathbf{w}(k) \quad \mathbf{w}(k) = -\tilde{K}\xi(k) \quad (3.21)$$

donde:

$$\xi(k) = \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{v}_e(k) \end{bmatrix} \quad \tilde{G}(k) = \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix}$$

$$\tilde{H}(k) = \begin{bmatrix} H \\ -CH \end{bmatrix} \quad \tilde{K}(k) = \begin{bmatrix} K & -K_I \end{bmatrix}$$

con:

$$\mathbf{x}(k) - \mathbf{x}(\infty) = \mathbf{x}_e(k)$$

$$\mathbf{v}(k) - \mathbf{v}(\infty) = \mathbf{v}_e(k)$$

La ecuación característica del sistema se deduce de (3.21) y resulta:

$$\det[zI - \tilde{G} + \tilde{H}\tilde{K}] = 0 \quad (3.22)$$

Por extensión, para resolver el problema del control óptimo cuadrático discreto estacionario, debemos minimizar la siguiente función de costo:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [\xi(k)^T \tilde{Q}\xi(k) + \mathbf{w}^T(k) \tilde{R}\mathbf{w}] \quad (3.23)$$

donde el vector de entrada \mathbf{w} es de orden m . El resultado del proceso de minimización, produce las expresiones de la matriz de ganancia \tilde{K} del controlador y la ecuación de Riccati asociada, es decir (comparar con las ecuaciones (3.15) y (3.16)):

$$\tilde{P} = \tilde{Q} + \tilde{G}^T \tilde{P} \tilde{G} - \tilde{G}^T \tilde{P} \tilde{H} [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (3.24)$$

$$\tilde{K} = [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (3.25)$$

donde \tilde{P} es una matriz simétrica definida positiva de dimensión $(n+1) \times (n+1)$, \tilde{Q} es una matriz simétrica semidefinida positiva de dimensión $(n+1) \times (n+1)$ y \tilde{R} es una matriz simétrica definida positiva de dimensión $m \times m$.

La realimentación de estados empleada en la ley de control requiere que todos los estados sean disponibles. Por esta razón es necesario diseñar un observador de estados multivariable, tema que se cubre en el siguiente capítulo.

CAPÍTULO IV

EL OBSERVADOR DE ESTADOS MULTIVARIABLE

4.1 ¿Porqué se Requiere un Observador?

Un observador de estados es un algoritmo que nos permite estimar los estados del sistema a partir de algunas mediciones. La experiencia dicta que en implementaciones reales de sistemas de control, sólo unas cuantas variables del vector de estado $\mathbf{x}(k)$ del proceso pueden ser medidas en forma directa. En el caso que nos ocupa, por ejemplo, sólo se miden las variables de salida y_1 , y_2 e y_3 de las 33 (o 25, según el caso) que conforman el vector de estados.

El vector de estados estimado, al cual denominaremos $\hat{\mathbf{x}}(k)$, es necesario debido a que la ley de control multivariable que estamos empleando requiere la realimentación de los estados del proceso en la forma: $\mathbf{u} = -\mathbf{K}\mathbf{x}$. El diseño de un observador de estados implica determinar su matriz de ganancia \mathbf{K}_e , la cual puede ser calculada a partir de la minimización de una función de costo cuadrática. El procedimiento es similar al seguido para determinar la matriz de ganancia del controlador de realimentación de estados (capítulo III). El diagrama de bloques del observador de estados de orden completo se ilustra en la figura 4.1. Este observador es de orden completo porque estima todos los estados del sistema, en contraposición a otros observadores de orden reducido. El observador que nos ocupa emplea las mediciones de la salida vectorial $\mathbf{y}(k)$ de orden m y de la señal de control vectorial $\mathbf{u}(k)$ de orden m .

4.2 El Observador Óptimo Cuadrático Multivariable

La deducción de la ecuación del observador en estudio, es directa a partir de la figura 4.1, donde:

$$\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) \quad (4.1)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (4.2)$$

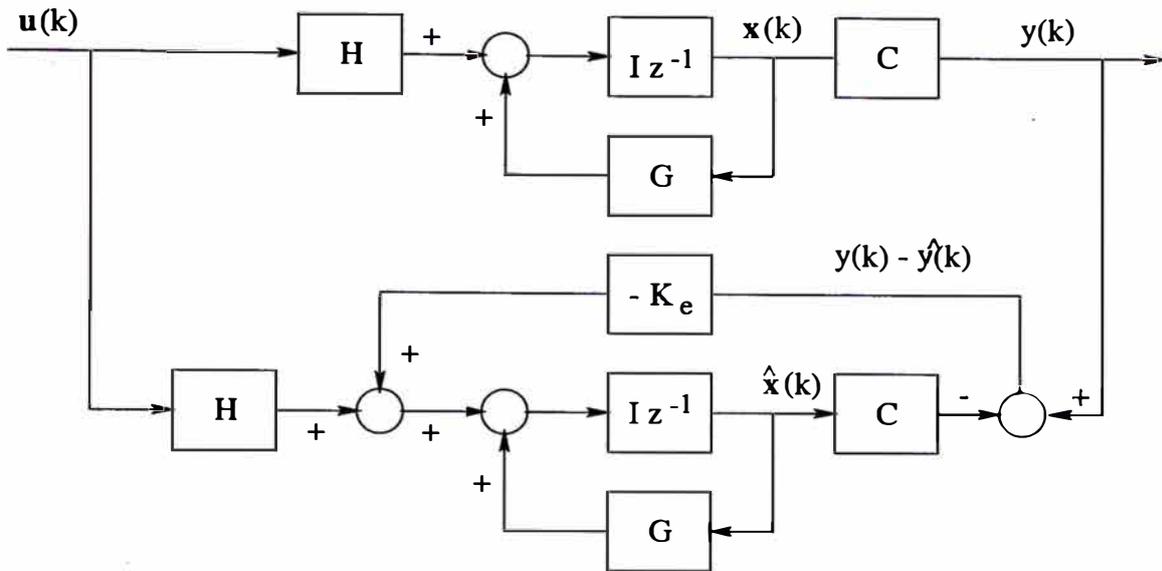


Figura 4.1: Diagrama de bloques de un observador de estados de orden completo

Entonces, la ecuación del observador discreto toma la forma:

$$\hat{\mathbf{x}}(k+1) = G\hat{\mathbf{x}}(k) + H\mathbf{u}(k) + K_e[\mathbf{y}(k) - C\hat{\mathbf{x}}(k)] \quad (4.3)$$

donde $\hat{\mathbf{x}}(k)$ es el vector de estado estimado de dimensión n , $\hat{\mathbf{y}}(k)$ representa el vector de salida estimado de dimensión m , K_e es la matriz de ganancia de realimentación del observador con dimensión $n \times m$ y C es la matriz de salida de dimensión $m \times n$. Reemplazando la ecuación (4.2) en (4.3) y restando la ecuación resultante de (4.1), dá como resultado la ecuación del error del observador:

$$\mathbf{e}(k+1) = [G - K_e C]\mathbf{e}(k); \quad \mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \quad (4.4)$$

De (4.4), se puede aseverar que la ecuación característica del observador es:

$$\det[zI - G + K_e C] = 0 \quad (4.5)$$

en donde la matriz K_e debe seleccionarse de modo tal que el error tienda a cero con una velocidad adecuada. Este observador tendrá comportamiento estable si es que las raíces de su ecuación característica se posicionan dentro del círculo unitario.

Nosotros calcularemos la matriz K_e empleando el concepto de dualidad. Dado que el determinante de una matriz y el de su transpuesta son iguales, podemos modificar la forma de la ecuación (4.5) como sigue:

$$\det[zI - G + K_e C] = \det[(zI - G + K_e C)^T] = \det[zI - G^T + C^T K_e^T] \quad (4.6)$$

La comparación de la ecuación característica del controlador de estados (3.22) con la ecuación (4.6), nos induce a efectuar las siguientes modificaciones:

$$G \rightarrow G^T; \quad H \rightarrow C^T; \quad K \rightarrow K_e^T \quad (4.7)$$

Usando dichas modificaciones en las estructuras de la ecuación de estado del proceso $\mathbf{x}(k+1) = G\mathbf{x}(k) + H\mathbf{u}(k)$, de su ecuación de salida $\mathbf{y}(k) = C\mathbf{x}(k)$ y de su ley de control $\mathbf{u}(k) = -K\mathbf{x}(k)$, obtendremos la siguiente ecuación de estado:

$$\alpha(k+1) = G^T\alpha(k) + C^T\beta(k) \quad (4.8)$$

con una ley de control con estructura:

$$\beta(k) = -K_e^T\alpha(k) \quad (4.9)$$

Empleando las ecuaciones (4.8) y (4.9) en la función de costo siguiente:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [\alpha^T(k)Q_e\alpha(k) + \beta^T(k)R_e\beta(k)] \quad (4.10)$$

Esto significa que la correspondiente ecuación de Riccati toma la forma:

$$P_e = Q_e + GP_eG^T - GP_eC^T[R_e + CP_eC^T]^{-1}CP_eG^T \quad (4.11)$$

donde la matriz de ganancia K_e viene dada por:

$$K_e = [R_e + CP_eC^T]^{-1}CP_eG^T \quad (4.12)$$

Para computar P_e , es útil la siguiente ecuación recursiva:

$$P_e(k+1) = Q_e + GP_e(k)G^T - GP_e(k)C^T[R_e + CP_e(k)C^T]^{-1}CP_e(k)G^T \quad (4.13)$$

CAPITULO V

IDENTIFICACIÓN DE PARÁMETROS

5.1 Consideraciones Generales

Dado que los parámetros K_{pij} , T_{pij} y D_{pij} del modelo descrito en (2.1) pueden ser desconocidos y tal vez, variantes con el tiempo, tales parámetros tienen que ser estimados indirectamente a partir del del modelo del proceso. Es decir, el proceso de estimación va a producir una ecuación de estado estimada, que tenga la capacidad de capturar las características del proceso original y reemplazarlo. Esta estimación se puede realizar empleando la técnica de los mínimos cuadrados recursivo. Esta técnica trabaja en el dominio discreto.

5.2 El Método de Identificación de Parámetros IMCRM

Para estimar los parámetros del modelo del sistema usaremos a las señales $u(t)$ e $y(t)$ como los agentes que proporcionan los datos al algoritmo de estimación. La descripción ilustrada en la ecuación (2.21) tiene que ser reordenada como:

$$y_1(k) = \psi_1^T(k)\hat{\theta}_1(k) \quad y_2(k) = \psi_2^T(k)\hat{\theta}_2(k) \quad y_3(k) = \psi_3^T(k)\hat{\theta}_3(k) \quad (5.1)$$

donde el vector de información ψ_i , $i = 1,2,3=r$, contiene los valores presentes y pasados de la entrada $u_i(k)$ y de la salida $y_i(k)$. La ecuación (5.1) nos informa que debemos efectuar tres estimaciones de parámetros dentro del mismo software de simulación. Los parámetros en juego podrían estimarse empleando el algoritmo básico de identificación (o estimación) de parámetros de los mínimos cuadrados recursivo (MCR). Sin embargo, tal algoritmo puede presentar problemas potenciales de carácter numérico que podrían afectar el diseño final del controlador adaptivo con ubicación de polos. Por tal motivo, en este estudio se emplea el método de identificación de los mínimos cuadrados recursivo mejorado (MCRM) propuesto en [10] y empleado satisfactoriamente en [9]. La forma recursiva del algoritmo de identificación de parámetros es necesaria, ya que la estimación de

parámetros se debe ejecutar “on-line” (en línea). El método MCRM puede implementarse para una salida característica $y_i(k)$, en la forma siguiente:

- 1) Obtener el vector de parámetros inicial $\hat{\theta}(0)$ empleando los valores de los parámetros del modelo del sistema.
- 2) La matriz de covarianza inicial P puede tener la forma $P = \alpha I$, donde I es la matriz identidad y $\alpha \gg 1$.
- 3) Tomar nuevas mediciones de $y(t)$ y de $u(t)$ en el proceso, para obtener una nueva ecuación de la forma dada en (5.1).
- 4) Computar: $\rho(t) = \max(1, \|\psi(t)\|)$; $\psi_n = \psi/\rho(t)$.
- 5) Determinar la matriz $N(t)$ aplicando factorización Cholesky en:

$$N(t)N(t)^T = P(t)$$

- 6) Determinar la matriz diagonal de escalamiento $S(t)$ con elementos s_{jj} . Tal matriz minimiza el número condicional del producto $S(t)N(t)$, siempre que $s_{jj} = 1/n_{jj}$, donde cada n_{jj} se obtiene tomando el valor absoluto de la suma de los elementos de la fila j de $N(t)$.
- 7) Computar:

$$P_S(t) = S(t)P(t)S(t)$$

$$\psi_{nS} = [S(t)]^{-1}\psi_n$$

$$r(t) = 1 + \psi_{nS}^T(t)P_S(t-1)\psi_{nS}(t)$$

$$\lambda(t) = 1 - \frac{1}{2} \left[r(t) - \sqrt{r^2(t) - \frac{4\|P_S(t-1)\psi_{nS}(t)\|^2}{tr P_S(t-1)}} \right]$$

$$e_n(t) = Y(k)/\rho(t) - \psi_n^T(t) \hat{\theta}(t-1)$$

$$j(t) = [\psi_{nS}^T(t+1)P_S(t)\psi_{nS}(t+1) + \lambda(t)]$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + S^{-1}(k)P_S(t)\psi_{nS}(k)e_n(t+1)/j(t)$$

$$H_{nS}(k+1) = P_S(t)\psi_{nS}(t+1) \times [\psi_{nS}^T(t+1)P_S(t)\psi_{nS}(t+1) + \lambda(t)]^{-1}$$

$$P_S(t+1) = [I - H_{nS}(t+1)\psi_{nS}^T(t+1)]P_S(t)/\lambda(t)$$

$$tt = \text{abs} \left[\frac{\max \text{ eig}(P_S)}{\min \text{ eig}(P_S)} \right]$$

- 8) Implementar el criterio para detener la estimación y encontrar la nueva matriz de escalamiento N_{new} como sigue: sabiendo que tt es el número condicionante de P_S , si $tt \leq \tau$ (cota inferior), parar la estimación; en otro

caso, si $tt \geq \kappa$ (cota superior), determinar $N_{new}(t)$ (a partir de la relación $P_S = N_{new}N_{new}^T$), calcular los elementos n_{newjj} (valor absoluto de la suma de los elementos de las filas de $N_{new}(t)$), determinar la matriz diagonal S_{new} con elementos $s_{newjj} = 1/n_{newjj}$, y calcular la matriz $P_{news} = S_{new}P_S S_{new}$. Finalmente, actualizar: $P = P_{news}$.

5.3 Algunas Notas sobre el Método IMCRM

Sobre el punto (1) del algoritmo del método IMCRM cabe hacer la recomendación siguiente. El algoritmo muy bien puede trabajar a partir de un vector inicial de parámetros nulo. En trabajos “off-line” (fuera de línea), por ejemplo: la identificación de los los parámetros de un motor, se puede emplear tal opción. Sin embargo, para implementar sistemas de control adaptivo, como en nuestro caso, es preferible que el vector inicial de parámetros contenga los valores más cercanos a sus valores reales, si es que se dispone de información al respecto. A manera de ejemplo, si no se tuviera información alguna de los parámetros de un motor que es parte del sistema de control adaptivo, entonces los parámetros conocidos de otro motor pueden ser usados como parámetros iniciales. En cualquier otro caso, es recomendable emplear valores diferentes de cero en el vector inicial de parámetros.

Respecto al punto (2), es decir, que la matriz de covarianza inicial P puede tener la forma $P = \alpha I$, donde I es la matriz identidad y $\alpha \gg 1$, significa que α puede ser un valor 500, 800, 1000 o mayor. En realidad, el parámetro α hace que el algoritmo de identificación converja más rápido. Sin embargo, si dicho parámetro resulta muy grande, entonces puede traer consigo problemas numéricos. Por ello se recomienda hacer las simulaciones del caso empleando valores típicos de α , con el propósito de seleccionar su valor más apropiado

El punto (4) tiene bastante significancia. En realidad, la operación:

$$\rho(t) = \max(1, \|\psi(t)\|) \quad \psi_n = \psi/\rho(t)$$

normaliza el valor numérico de los parámetros. Para el algoritmo de identificación esto es de suma importancia, debido a que la normalización evita que el algoritmo trabaje con números de diferente ordenes de magnitud.

Cualquier algoritmo de identificación recursiva requiere que la señal de excitación (la señal de entrada) del algoritmo sea abundante y persistente. La razón

es obvia: estos algoritmos requieren de data para operar. Si embargo, no siempre se dispone de excitación persistente, es decir, de datos suficientes en cada muestra para la operación normal del algoritmo. Una forma para hacer que el algoritmo de identificación sea más robusto en la presencia de excitación muy pobre (no persistente), es hacer un escalamiento de los datos. Para ello, necesitamos un filtro de raíz cuadrada, tal como se ilustra en el punto (5), para determinar una matriz de pre-escalamiento. Así, aplicando factorización Cholesky en la relación:

$$N(t)N(t)^T = P(t)$$

obtenemos la matriz de pre-escalamiento $N(t)$. Esta matriz $N(t)$ permite determinar la matriz de escalamiento definida en el punto (6). Este punto se explica por sí mismo.

Una cosa es tener la presencia de excitación pobre. Otra es no tener data disponible en el algoritmo de identificación. En el caso de ausencia de datos, la intuición dicta que es preferible detener el algoritmo de identificación y emplear los datos de la muestra anterior (tener en cuenta que el algoritmo se computa para cada muestra o tiempo de muestreo). Una forma de saber si los datos disponibles son o no suficientes para que el algoritmo opere sin problemas, es determinando el número condicionante actual de la matriz P y compararlo con los valores máximo y mínimo referenciales de dicho condicionante. Este procedimiento se describe en el punto (8).

CAPITULO VI

EL SISTEMA DE CONTROL ADAPTIVO

6.1 Procedimiento de Diseño

En la sección 1.4 mencionamos que los sistemas de control adaptivo ajustan su comportamiento a las cambiantes propiedades del proceso controlado y de las señales que interactúan sobre dicho proceso. Específicamente, en este trabajo se emplea la configuración de un sistema de control adaptivo con autosintonización mostrado en la figura 1.2, cuya operación se describe también en la sección 1.4. Como dicho sistema combina en su diseño un método de estimación de parámetros, una técnica de estimación óptima de estados, una representación lineal del modelo del proceso, y un controlador óptimo proporcional-integral, entonces el procedimiento de diseño debe de involucrar todos esos componentes, tal como se describe a continuación.

Formulación del Problema

En este estudio se diseña de un sistema de control adaptivo multivariable con autosintonización, cuya configuración se muestra en la figura 1.2. El proceso a controlar es la columna de destilación descrita en la sección 1.2. Las variables a controlar en dicha columna son: las composiciones del producto de cabeza y producto lateral (y_1 e y_2) y la temperatura de fondos (y_3). Las variables manipuladas (las fuerzas de control) son: el flujo de productos de cabeza (u_1), el flujo del producto lateral (u_2) y el reflujo de fondos (u_3). El objetivo de control consiste en estabilizar las señales de salida y_1 , y_2 e y_3 en presencia de cambios tipo escalón de las señales referencia r_1 , r_2 y r_3 . Las señales de control u_1 , u_2 y u_3 no deben de sobrepasar el valor normalizado de ± 0.7 cuando ocurran los cambios tipo escalón de las señales de referencia. El tiempo de estabilización debe ser menor de 110 minutos para y_1 , y_2 e y_3 . El error en estado estable de las señales controladas debe ser nulo, con mínimo sobreimpulso.

Modelado del Proceso Columna de Destilación

El modelado dinámico del proceso columna de destilación se realizó en la sección 2.1. El diagrama de bloques de la columna se muestra en la figura 2.6.

Determinación del Modelo del Proceso

El modelo del proceso empleado en este trabajo se muestra en la figura 2.7. La ecuación de estado de este modelo se determinó en la sección 2.4. La ecuación de estado del modelo en el tiempo discreto está dada por la ecuación (2.12):

$$\mathbf{x}_d(k+1) = \mathbf{A}_d \mathbf{x}_d(k) + \mathbf{B}_d \mathbf{u}(k) \quad \mathbf{y}(k) = \mathbf{C}_d \mathbf{x}_d(k) + \mathbf{D}_d \mathbf{u}(k)$$

Forma Canónica Observable del Modelo

La forma canónica observable del modelo se determinó también en la sección 2.4. La ecuación de estado de la forma canónica observable se muestra en la ecuación 2.13:

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k) \quad \mathbf{y}(k) = \mathbf{C} \mathbf{x}(k) + \mathbf{D} \mathbf{u}(k)$$

Mínima Representación MISO del Modelo

En la sección 2.4 también se determinó las ecuaciones que representan el conjunto de ecuaciones de diferencias para mínima representación, empleando el un número mínimo de parámetros. La ecuación (2.21) muestra tal representación. Esta representación es muy importante por dos buenas razones. La primera, es que ya posee la forma empleada en un algoritmo de estimación de parámetros recursivo discreto. La segunda razón, es que permite recobrar la forma canónica observable del sistema, luego del proceso de estimación de parámetros. El método de estimación empleado se menciona en el siguiente párrafo.

Estimación de los Parámetros del Modelo

Este punto se trató en el Capítulo V. El método de estimación a emplear, es de los mínimos cuadrados recursivo mejorado (MCRM), el cual es una versión mejorada del método clásico de los mínimos cuadrados. Este método se describe en detalle en la sección 5.2.

Diseño del Observador de Estados Multivariable

El observador óptimo cuadrático multivariable se diseñó en la sección 4.2. Básicamente, la ecuación del observador es (ver ecuación (4.3)):

$$\hat{\mathbf{x}}(k+1) = G\hat{\mathbf{x}}(k) + H\mathbf{u}(k) + K_e[\mathbf{y}(k) - C\hat{\mathbf{x}}(k)]$$

Diseño del Controlador Multivariable

La ley de control proporcional-integral que forma parte de la configuración del sistema de control adaptivo con autosintonización, fue determinada en la sección 3.2. Esta ley de control está dada por la ecuación (3.12).

Simulación del Sistema de Control Adaptivo

Esta parte se trata en la siguiente sección. Las partes correspondientes al hardware del sistema para implementación en tiempo real, al software de control, y, a las pruebas de funcionamiento, no se cubren en este trabajo.

6.2 Simulación del Sistema de Control Adaptivo

El programa `nava4.m` escrito en código MATLAB [3], cuyo listado se muestra en el Apéndice, resuelve el problema del control adaptivo con autosintonización para la columna de destilación. Se ha considerado que todos los tiempos muertos varían de un valor mínimo a un valor máximo. El resultado de la simulación se muestra en las figuras 6.1, 6.2, 6.3 y 6.4. Estos resultados serán discutidos en las Conclusiones de este estudio.

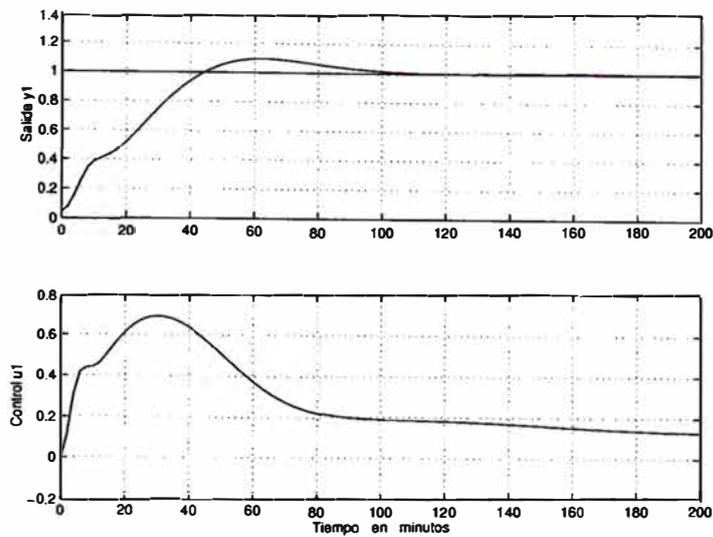


Figura 6.1: Entrada u_1 y salida controlada y_1 para el sistema de control adaptivo de la columna de destilación.

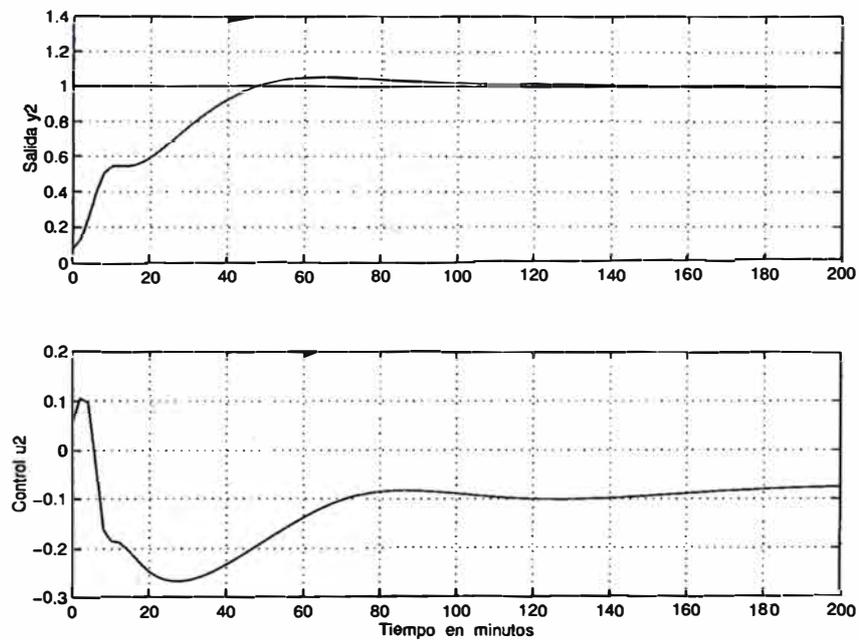


Figura 6.2: Entrada u_2 y salida controlada y_2 para el sistema de control adaptivo de la columna de destilación.

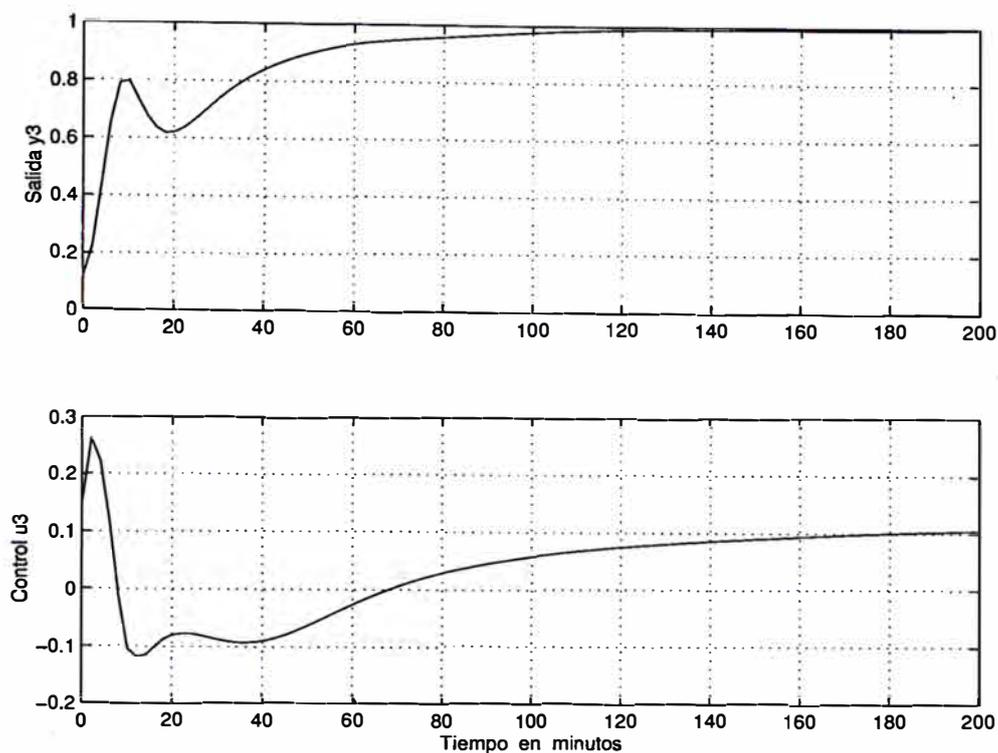


Figura 6.3: Entrada u_3 y salida controlada y_3 para el sistema de control adaptivo de la columna de destilación.

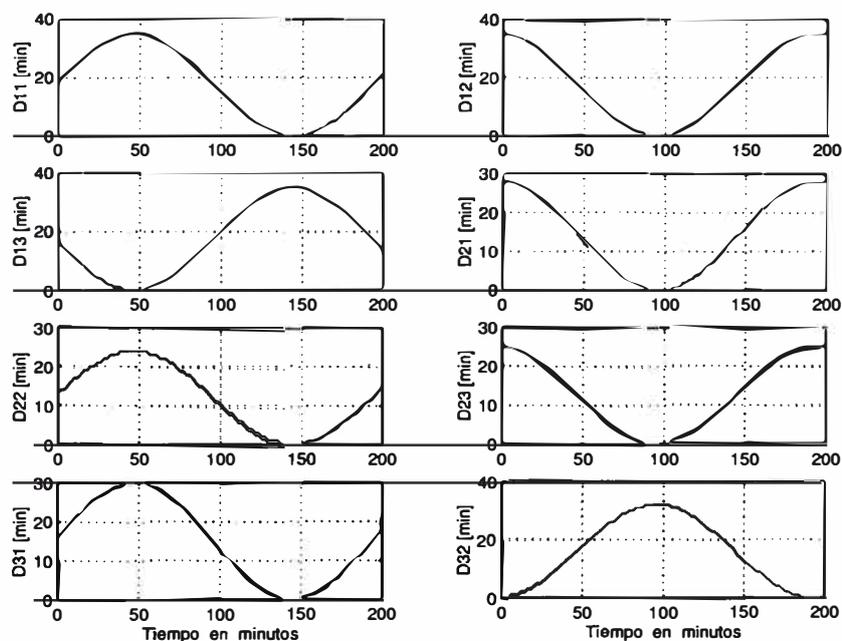


Figura 6.4: Variación en el tiempo de los tiempos muertos del proceso columna de destilación.

CONCLUSIONES

Conclusiones Generales

En este estudio se ha propuesto un procedimiento de diseño de un sistema de control adaptivo multivariable discreto y con autosintonización, el cual permite controlar procesos multivariables que poseen múltiples tiempos muertos. Estos tiempos muertos pueden ser o no variantes con el tiempo.

Específicamente, en este trabajo se emplea la configuración de un sistema de control adaptivo con autosintonización mostrado en la figura 1.2. Dicho sistema combina en su diseño: el método de los mínimos cuadrados recursivo mejorado para la estimación de los parámetros, el observador óptimo cuadrático multivariable para estimar los estados, la forma canónica observable del modelo del proceso, y el controlador óptimo proporcional-integral multivariable.

El proceso controlado es la columna de destilación, cuyo modelo dinámico posee múltiples tiempos muertos $e^{-D_{ij}}$, los cuales complican tremendamente el diseño del sistema de control adaptivo discreto. La filosofía de diseño empleada en este estudio, precisamente para evitar la presencia de los tiempos muertos en el modelo dinámico, consiste en determinar un modelo lineal del proceso, con la capacidad de capturar los efectos de los tiempos muertos. Para ello, aquellos tiempos muertos considerados en dicho modelo, se reemplazan por aproximaciones racionales de segundo orden. Otras aplicaciones podrían requerir aproximaciones racionales de mayor orden, mas no el caso que nos ocupa.

El modelo de múltiples entradas y múltiples salidas del proceso, se transforma luego a su forma canónica observable, la que da lugar a un conjunto de ecuaciones de diferencias de múltiples entradas y una sola salida, las cuales son convenientes para propósitos de identificación de parámetros y diseño del observador y controlador multivariables del sistema.

El sistema de control adaptivo diseñado, tal como lo comprueban las simulaciones del caso, cumple con las especificaciones de diseño planteadas, Así, las señales de control u_1 (flujo de productos de cabeza), u_2 (el flujo del producto

lateral) u_3 (el reflujo de fondos) no sobrepasan el valor normalizado de ± 0.7 cuando ocurran los cambios tipo escalón de las señales de referencia r_1 , r_2 y r_3 . Del mismo modo, el tiempo de estabilización es menor de 110 minutos para las salidas y_1 (composición del producto de cabeza), y_2 (composición del producto lateral) e y_3 (temperatura de fondos) son menores de 110 min, incluso cuando todos los tiempos muertos están variano en el tiempo. También se ha logrado que el error en estado estable de las señales controladas sea nulo y con mínimo sobreimpulso.

El tiempo de muestreo empleado es de dos minutos. Este tiempo es suficiente para poder implementar en tiempo real el sistema de control óptimo discreto diseñado. Es decir, para poder realizar en menos de dos minutos todos los cálculos requeridos para diseñar el sistema de control adaptivo.

Para demostrar la robustez con respecto a variaciones del tiempo muerto, la simulación del sistema de control adaptivo se realizó para la situación en que todos los tiempos muertos varían de un valor máximo a un valor mínimo, tal como se observa en la figura 6.4. En las figuras 6.1, 6.2 y 6.3, se observa que las salidas controladas satisfacen las especificaciones de diseño a pesar de la presencia de tiempos muertos variando simultáneamente.

Sobre la Implementación del Sistema de Control

Todas las configuraciones para implementar en tiempo real el sistema de control adaptivo de la columna de destilación, necesitan desarrollar el software de control ad-hoc. Tener en cuenta las leyes de control avanzado, aún no están comercializados en masa, como sí lo están los controladores del tipo PID. La configuración a la que me voy a referir, ha sido empleada con éxito en las aplicaciones de la referencia [6] y se muestra en la figura 6.5. La implementación del sistema de control adaptivo multivariable de la columna de destilación parte de la suposición de que dicho proceso ya está controlado en forma clásica, y cuenta con el equipamiento siguiente:

- 1) Válvula de control, transmisor de flujo y controlador para el lazo de control de producto de cabeza.
- 2) Válvula de control, transmisor de flujo y controlador para el lazo de control de producto lateral.

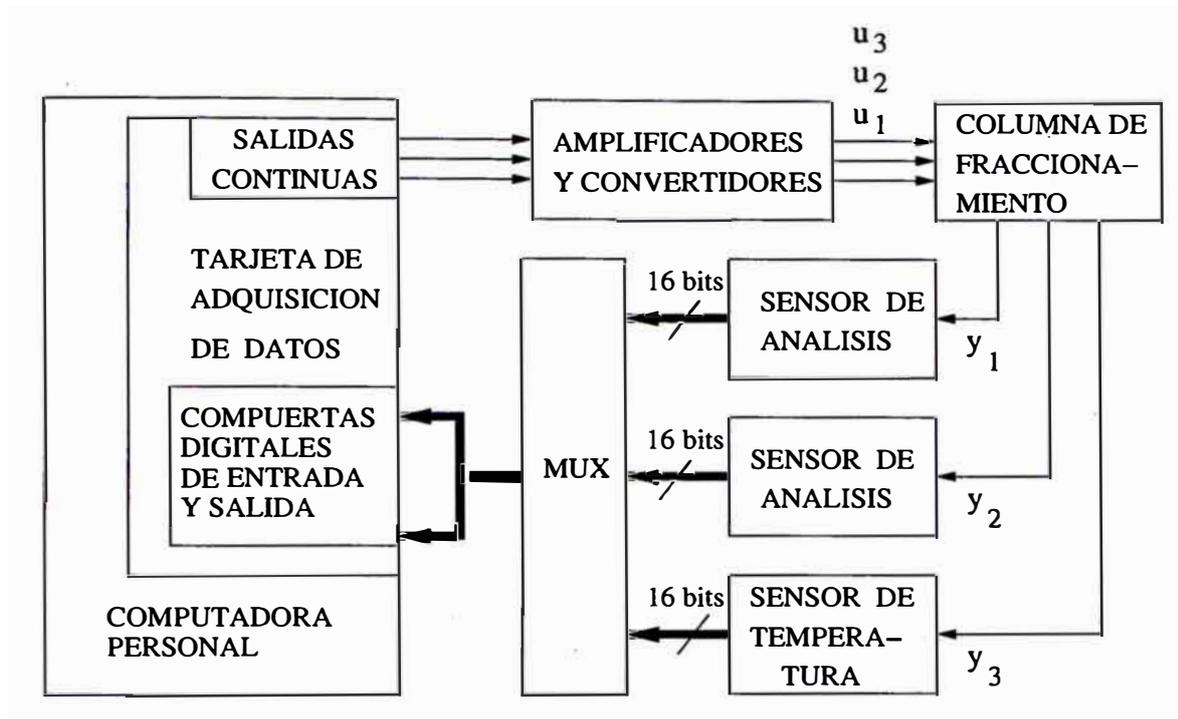


Figura 6.5: Implementación del sistema de control adaptivo para la columna de destilación.

- 3) Válvula de control, transmisor de temperatura, transmisor de flujo (para medición) y controlador para el lazo de control de temperatura de fondos.
- 4) Válvula de control, transmisor de nivel y controlador para el lazo de control del nivel de fondos.
- 5) Válvula de control, transmisor de nivel y controlador para el lazo de control del nivel de la columna de evaporización.
- 6) Válvula de control, transmisor de nivel y controlador para el lazo de control del nivel del condensador.

La implementación de los lazos del sistema de control adaptivo requiere la adquisición de equipos medidores (analizadores) en línea de las concentraciones de los productos de cabeza (y_1) y lateral (y_1). Para el procesamiento de las señales, asumiendo que los lazos de control existentes son del tipo neumático, se requiere además: un convertidor de presión a corriente para la variable y_3 (temperatura de fondos) y tres convertidores de corriente a presión para las señales de control u_1 , u_2 y u_3). Las señales de referencia se originan en nuestro caso en la PC.

El costo de implementación del sistema de control adaptivo, partiendo de una instalación clásica existente resulta entonces:

- 1) Una PC Industrial: US \$ 3000

- 2) Dos analizadores de concentración de petróleo: US \$ 6000
 - 3) Cuatro convertidores de señal: US \$ 2400
 - 4) Desarrollo del software: US \$ 3000
 - 5) Costos de instalación y de puesta en operación: US \$ 2000
- Gran total: US \$ 16400

Finalmente, deseo mencionar que este estudio sirva como punto inicial para desarrollar en un futuro cercano, otros trabajos relacionados con el tema, como puede ser, por ejemplo, el diseño de la columna de destilación empleando control adaptivo con modelo referencial.

APÉNDICE

LISTADO DE PROGRAMAS

Programas del Capítulo II

```

% naval.m RESPUESTA AL ESCALON DEL SISTEMA COLUMNA DE
%           DESTILACION PARA FRACCIONAR PETROLEO
clear all

num= {4.05, 1.77, 5.88 ;...
      5.39 ,5.72 , 6.9 ;...
      4.38 ,4.42, 7.2};
den= {[50 1], [60 1] , [50 1];...
      [50 1], [60 1] , [40 1];...
      [33 1], [44 1] , [19 1]};
tf0=tf(num,den); %funcion de transferencia sin retardo
tf0.iodelaymatrix=[27, 28, 27; 18 ,14 ,15; 20 ,22 ,0];

step(tf0); % EDITAR (TRADUCIR LAS PALABRAS) DE LA FIGURA CREADA
           % ANTES DE GUARDARLA
print -deps -f naval

% tmuertos.m RESPUESTAS DE FASE DE APROX. RACIONALES DE exp(-Ds)
clear all

D = 1;
c = sqrt(3)/3; N=70;
for k=1:N
    w = 0.1*k - 0.1;
    pa(k) = angle(1-j*w*D/2+(j*w*D)^2/12) - ...
            angle(1+j*w*D/2+(j*w*D)^2/12);          % PADE 2do ORDEN
    pb(k) = -4*atan(w*D/4);                          % PASA TODO
    pc(k) = angle(1-j*w*D/2+(j*w*D)^2/10-(j*w*D)^3/120) - ...
            angle(1+j*w*D/2+(j*w*D)^2/10+(j*w*D)^3/120); % PADE 3er ORDEN
    r(k) = -w*D;                                     % TIEMPO MUERTO REAL
end

jj=linspace(0,N/10,N);
plot(jj,pb,'--',jj,pa,'-.',jj,pc,':',jj,r);
xlabel('Frecuencia Normalizada wD');
ylabel('Respuesta en ángulo de fase [rad]');

```

```

text(0.5,-4.5,'Sólido: Ángulo de fase real');
text(0.5,-5.0,'Punto y raya: Padé de 2do orden');
text(0.5,-5.5,'Trazos: Pasa-todo de 2do orden');
text(0.5,-6.0,'Punteado: Padé de 3er orden');

print -deps -f tmueertos

% nava2.m RESPUESTA AL ESCALON DEL SISTEMA APROXIMADO COLUMNA DE
%           DESTILACION PARA FRACCIONAR PETROLEO
clear all
clc

% MODELADO DE LOS TIEMPOS MUERTOS USANDO PADE DE TERCER ORDEN
ord=2; D=20; % D=14;
[ndelay11,ddelay11]=pade(2*D-1,ord);
[ndelay12,ddelay12]=pade(2*D,ord);
[ndelay13,ddelay13]=pade(2*D-1,ord);
[ndelay21,ddelay21]=pade(D+4,ord);
[ndelay22,ddelay22]=pade(D,ord);
[ndelay23,ddelay23]=pade(D+1,ord);
[ndelay31,ddelay31]=pade(D+6,ord);
[ndelay32,ddelay32]=pade(D+8,ord);
[ndelay33,ddelay33]=pade(0,ord);

% CALCULO DEL NUMERADOR Y DENOMINADOR DE LA MATRIZ DE TRANSFERENCIA
num1= {4.05*ndelay11, 1.77*ndelay12, 5.88*ndelay13 ;...
       5.39*ndelay21 ,5.72*ndelay22 , 6.9*ndelay23 ;...
       4.38*ndelay31 ,4.42*ndelay32, 7.2*ndelay33};
den1= {conv(ddelay11,[50 1]),conv(ddelay12, [60 1]),...
       conv(ddelay13,[50 1]); conv(ddelay21,[50 1]),...
       conv(ddelay22,[60 1]),conv(ddelay23,[40 1]);...
       conv(ddelay31,[33 1]), conv(ddelay32,[44 1]),...
       conv(ddelay33,[19 1])};
tf1=tf(num1,den1);

step(tf1) % EDITAR (TRADUCIR LAS PALABRAS) LA FIGURA
          % ANTES DE GUARDARLA
print -deps -f nava2

% impulso.m COMPARACION DE LAS RESPUESTAS AL IMPULSO DE LOS SISTEMAS
%           COLUMNA DE DESTILACION CON TIEMPO MUERTO Y APROXIMADO
%           (REEMPLAZANDO LOS TIEMPOS MUERTOS CON PADE DE TERCER ORDEN)

clear all;
clc;

% FUNCION DE TRANSFERENCIA DEL SISTEMA CON TIEMPOS MUERTOS
num= {4.05, 1.77, 5.88 ;...
      5.39 ,5.72 , 6.9 ;...

```

```

    4.38 ,4.42, 7.2};
den= {[50 1], [60 1] , [50 1];...
      [50 1], [60 1] , [40 1];...
      [33 1], [44 1] , [19 1]};
tf0=tf(num,den); % FUNCION DE TRANSFERENCIA SIN TIEMPOS MUERTOS

% SE AÑADEN LOS TIEMPOS MUERTOS
tf0.iodelaymatrix=[27, 28, 27; 18 ,14 ,15; 20 ,22 ,0];

% SISTEMA COLUMNA DE DESTILACION APROXIMADO (REEMPLAZANDO LOS
% TIEMPOS MUERTOS CON APROXIMACION DE PADE DE TERCER ORDEN)

ord=3;
[ndelay11,ddelay11]=pade(27,ord);
[ndelay12,ddelay12]=pade(28,ord);
[ndelay13,ddelay13]=pade(27,ord);
[ndelay21,ddelay21]=pade(18,ord);
[ndelay22,ddelay22]=pade(14,ord);
[ndelay23,ddelay23]=pade(15,ord);
[ndelay31,ddelay31]=pade(20,ord);
[ndelay32,ddelay32]=pade(22,ord);
[ndelay33,ddelay33]=pade(0,ord);
% MATRIZ DE TRANSFERENCIA APROXIMADA
num1= {4.05*ndelay11, 1.77*ndelay12, 5.88*ndelay13 ;...
      5.39*ndelay21 ,5.72*ndelay22 , 6.9*ndelay23 ;...
      4.38*ndelay31 ,4.42*ndelay32, 7.2*ndelay33};
den1= {conv(ddelay11,[50 1]),conv(ddelay12, [60 1]),...
      conv(ddelay13,[50 1]);...
      conv(ddelay21,[50 1]), conv(ddelay22,[60 1]),...
      conv(ddelay23,[40 1]);...
      conv(ddelay31,[33 1]), conv(ddelay32,[44 1]),...
      conv(ddelay33,[19 1])};

tf1=tf(num1,den1);

% GRAFICOS

grid on;impulse(tf0,'-',tf1,'--');

print -deps -f impulso

```

Programa del Capítulo V

```

% nava4.m CONTROL ADAPTIVO DE LA COLUMNA DE DESTILACION CON TIEMPOS MUERTOS.
clear all

% ECUACION DE ESTADO DEL MODELO DE LA COLUMNA DE DESTILACION

```

```

T11=50; T12=60; T13=50; T21=50; T22=60; T23=40; T31=33; T32=44; T33=19;
K11=4.05; K12=1.77; K13=5.88; K21=5.39; K22=5.72; K23=6.9;
K31=4.38; K32=4.42; K33=7.2;
D11=27; D12=28; D13=27; D21=18; D22=14; D23=15; D31=20; D32=22;
% D11=7; D12=8; D13=7; D21=8; D22=4; D23=5; D31=5; D32=5;
% D11=35; D12=35; D13=35; D21=28; D22=24; D23=25; D31=30; D32=32;
a11=D12/2; b11=D12^2/12; a22=D21/2; b22=D21^2/12; a33=D32/2; b33=D32^2/12;
Am=[-1/T11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 -1/T21 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 -1/T31 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 -1/T12 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 -1/T22 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 -1/T32 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 -1/T13 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 -1/T23 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 -1/T33 0 0 0 0 0 0 0
      b11+2*a11/T11 0 0 b11+2*a11/T12 0 0 b11+2*a11/T13 0 0 -a11 -b11 0 0 0 0
      -1/T11 0 0 -1/T12 0 0 -1/T13 0 0 1 0 0 0 0 0
      0 b22+2*a22/T21 0 0 b22+2*a22/T22 0 0 b22+2*a22/T23 0 0 0 -a22 -b22 0 0
      0 -1/T21 0 0 -1/T22 0 0 -1/T23 0 0 0 1 0 0 0
      0 0 b33+2*a33/T31 0 0 b33+2*a33/T32 0 0 b33+2*a33/T33 0 0 0 0 -a33 -b33
      0 0 -1/T31 0 0 -1/T32 0 0 0 0 0 0 1 0];
%      0 0 -1/T31 0 0 -1/T32 0 0 -1/T33 0 0 0 0 1 0];

Bm=[K11/T11 0 0
      K21/T21 0 0
      K31/T31 0 0
      0 K12/T12 0
      0 K22/T22 0
      0 K32/T32 0
      0 0 K13/T13
      0 0 K23/T23
      0 0 K33/T33
      -2*a11*K11/T11 -2*a11*K12/T12 -2*a11*K13/T13
      K11/T11 K12/T12 K13/T13
      -2*a22*K21/T21 -2*a22*K22/T22 -2*a22*K23/T23
      K21/T21 K22/T22 K23/T23
      -2*a33*K31/T31 -2*a33*K32/T32 -2*a33*K33/T33
      K31/T31 K32/T32 0];

Cm=[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 1];

Dm=[0 0 0
      0 0 0
      0 0 0];
n=15; % ORDEN DEL MODELO
r=3; % No DE SALIDAS DEL PROCESO

```

```

p=3; % No DE ENTRADAS AL PROCESO
N = 100; % No DE ITERACIONES EN EL BUCLE DE CONTROL
% MATRICES DE PONDERACION DEL CONTROLADOR
Q=0.5*eye(n+r);
R=50*eye(r);
% MATRICES DE PONDERACION DEL OBSERVADOR
Qo=0.5*eye(n);
Ro=0.5*eye(r);

% DESCRIPCION DISCRETA DEL MODELO
Ts=2; % TIEMPO DE MUESTREO EN MINUTOS
[Ad,Bd,Cd,Dd] = c2dm(Am,Bm,Cm,Dm,Ts,'zoh');
rcoc=rank(ctrb(Am,Bm)); % rcoc=15, COMPL. CONTR. Y OBSV.
robc=rank(observ(Am,Cm)); % robc=15, COMPL. CONTR. Y OBSV.
rcod=rank(ctrb(Ad,Bd)); % rcod=15, COMPL. CONTR. Y OBSV.
robd=rank(observ(Ad,Cd)); % robd=15, COMPL. CONTR. Y OBSV.

% MATRIZ DE OBSERVABILIDAD
Od=[Cd
    Cd*Ad
    Cd*Ad^2
    Cd*Ad^3
    Cd*Ad^4
    Cd*Ad^5
    Cd*Ad^6
    Cd*Ad^7
    Cd*Ad^8
    Cd*Ad^9
    Cd*Ad^10
    Cd*Ad^11
    Cd*Ad^12
    Cd*Ad^13
    Cd*Ad^(n-1)]; % (n-1)=14 EN ESTE CASO
i=1;
for jj=1:n
    Od1(jj,:)=Od(i,:);
    Od2(jj,:)=Od(i+1,:);
    Od3(jj,:)=Od(i+1,:);
    i=i+r;
if (i>3*n) end;
end
% [svd(Od1)  svd(Od2)  svd(Od3)]; % SE OBTIENE:
%   5.7050    5.6731    5.6731
%   1.0000    1.0000    1.0000
%   0.0362    0.0772    0.0772
%   0.0001    0.0019    0.0019
%   0.0000    0.0003    0.0003
%   0.0000    0.0000    0.0000
%   0.0000    0.0000    0.0000

```

```

% 0.0000 0.0000 0.0000
% 0.0000 0.0000 0.0000
% 0.0000 0.0000 0.0000
% 0.0000 0.0000 0.0000
% 0 0.0000 0.0000
% 0 0 0
% 0 0 0
% 0 0 0
n1=5; n2=5; n3=5; % n1+n2+n3=n
cd1=Cd(1,:);
cd2=Cd(2,:);
cd3=Cd(3,:);
T=[cd1
    cd1*Ad
    cd1*Ad^2
    cd1*Ad^3
    cd1*Ad^(n1-1)
    cd2
    cd2*Ad
    cd2*Ad^2
    cd2*Ad^3
    cd2*Ad^(n2-1)
    cd3
    cd3*Ad
    cd3*Ad^2
    cd3*Ad^3
    cd3*Ad^(n3-1)];
% FORMA CANONICA OBSERVABLE
A=T*Ad*inv(T);
B=T*Bd;
C=Cd*inv(T);
rco=rank(ctrb(A,B)); % rco=15, COMPL. CONTR. Y OBSV.
rob=rank(obsv(A,C)); % rob=15, COMPL. CONTR. Y OBSV.

% m0=0; m1=n1=5; m2=n1+n2=10; m3=n1+n2+n3=15
m0=0; m1=5; m2=10; m3=15;
% i=1: cij=1 para j=1+m0=1; c11=1; cij=0 EN OTRO CASO
% i=2: cij=1 para j=1+m1=6; c26=1; cij=0 EN OTRO CASO
% i=3=r: cij=1 para j=1+m2=11; c311=1; cij=0 EN OTRO CASO
C=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 1 0 0 0 0];
% A=[A11 Z Z
% A21 A22 Z
% A31 A32 A33]; % A33=Arr; r=3;
Z=zeros(5);
% dim(A11)=n1Xn1=3X3; dim(A21)=n2Xn1=2X3; dim(A22)=n2Xn2=2X2, etc.

% alpha11=[a_(m1,m0+1) a_(m1,m0+2) a_(m1,m0+3) a_(m1,m0+4) a_(m1,m1)]

```

```

alpha11=[A(5,1) A(5,2) A(5,3) A(5,4) A(5,5)];
A11=[0 1 0 0 0
     0 0 1 0 0
     0 0 0 1 0
     0 0 0 0 1
     alpha11];
alpha21=[A(10,1) A(10,2) A(10,3) A(10,4) A(10,5)];
A21=[0 0 0 0 0
     0 0 0 0 0
     0 0 0 0 0
     0 0 0 0 0
     alpha21];
alpha22=[A(10,6) A(10,7) A(10,8) A(10,9) A(10,10)];
A22=[0 1 0 0 0
     0 0 1 0 0
     0 0 0 1 0
     0 0 0 0 1
     alpha22];
alpha31=[A(15,6) A(15,2) A(15,3) A(15,4) A(15,5)];
A31=[0 0 0 0 0
     0 0 0 0 0
     0 0 0 0 0
     0 0 0 0 0
     alpha31];
alpha32=[A(15,6) A(15,6) A(15,7) A(15,8) A(15,9)];
A32=[0 0 0 0 0
     0 0 0 0 0
     0 0 0 0 0
     0 0 0 0 0
     alpha32];
alpha33=[A(15,11) A(15,12) A(15,13) A(15,14) A(15,15)];
A33=[0 1 0 0 0
     0 0 1 0 0
     0 0 0 1 0
     0 0 0 0 1
     alpha33];
A=[A11  Z  Z
   A21 A22  Z
   A31 A32 A33];

% CALCULO DE LOS PARAMETROS betai_j <> betai_j, i=1,2,3=r, j=0,...,ni-1
% i=1, ni=n1=5, j=0,1,2,3,4=n1-1
beta1_0=B(5,:)-A(5,5)*B(4,:)-A(5,4)*B(3,:)-A(5,3)*B(2,:)-A(5,2)*B(1,:);
beta1_1=B(4,:)-A(5,5)*B(3,:)-A(5,4)*B(2,:)-A(5,3)*B(1,:);
beta1_2=B(3,:)-A(5,5)*B(2,:)-A(5,4)*B(1,:);
beta1_3=B(2,:)-A(5,5)*B(1,:);
beta1_4=B(1,:);
% i=2, ni=n2=5, j=0,1,2,3,4=n2-1
beta2_0=B(10,:) ...

```

```

        -A(10,5)*B(4,:) -A(10,4)*B(3:)-A(10,3)*B(2:)-A(10,2)*B(1,:);
        -A(10,10)*B(9:)-A(10,9)*B(8:)-A(10,8)*B(7:)-A(10,7)*B(6:);
beta2_1=B(9,:) ...
        -A(10,5)*B(3:)-A(10,4)*B(2:)-A(10,3)*B(1:);
        -A(10,10)*B(8:)-A(10,9)*B(7:)-A(10,8)*B(6:);
beta2_2=B(8,:) ...
        -A(10,5)*B(2:)-A(10,4)*B(1:)-A(10,10)*B(7:)-A(10,9)*B(6:);
beta2_3=B(7:)-A(10,5)*B(1:)-A(10,10)*B(6:);
beta2_4=B(6:);
% i=3, ni=n3=5, j=0,1,2,3,4=n3-1
beta3_0=B(15,:) ...
        -A(15,5)*B(4,:) -A(15,4)*B(3:)-A(15,3)*B(2:)-A(15,2)*B(1:);
        -A(15,15)*B(14:)-A(15,14)*B(13:)-A(15,13)*B(12:)-A(15,12)*B(11:);
        -A(15,10)*B(9:)-A(15,9)*B(8:)-A(15,8)*B(7:)-A(15,7)*B(6:);
beta3_1=B(14,:) ...
        -A(15,5)*B(3:)-A(15,4)*B(2:)-A(15,3)*B(1:); ...
        -A(15,10)*B(8:)-A(15,9)*B(7:)-A(15,8)*B(6:); ...
        -A(15,15)*B(13:)-A(15,14)*B(12:)-A(15,13)*B(11:);
beta3_2=B(13,:) ...
        -A(15,5)*B(2:)-A(15,4)*B(1:); ...
        -A(15,10)*B(7:)-A(15,9)*B(6:); ...
        -A(15,15)*B(12:)-A(15,14)*B(11:);
beta3_3=B(12:)-A(15,5)*B(1:)-A(15,10)*B(6:)-A(15,15)*B(11:);
beta3_4=B(11:);

% ECUACIONES MISO (MULTIPLE-INPUT-SINGLE-OUTPUT)

% y1(k+5)=alpha11*eta1+beta1_0*u(k)+beta1_1*u(k+1)+beta1_2*u(k+2)
%           +beta1_3*u(k+3)+beta1_4*u(k+4)
% eta1=[y1(k);y1(k+1);y1(k+2);y1(k+3);y1(k+4)];
% u(k+j)=[u1(k+j);u2(k+j);u3(k+j)]; j=0,1,2,3,4=n1-1

% y2(k+5)=alpha21*eta1+alpha22*eta2+beta2_0*u(k)+beta2_1*u(k+1)
%           +beta2_2*u(k+2)+beta2_3*u(k+3)+beta2_4*u(k+4)
% eta1=[y1(k);y1(k+1);y1(k+2);y1(k+3);y1(k+4)];
% eta2=[y2(k);y2(k+1);y2(k+2);y2(k+3);y2(k+4)];
% u(k+j)=[u1(k+j);u2(k+j);u3(k+j)]; j=0,1,2,3,4=n2-1

% y3(k+5)=alpha31*eta1+alpha32*eta2+alpha33*eta3+beta3_0*u(k)+beta3_1*u(k+1)
%           +beta3_2*u(k+2)+beta3_3*u(k+3)+beta3_4*u(k+4)
% eta1=[y1(k);y1(k+1);y1(k+2);y1(k+3);y1(k+4)];
% eta2=[y2(k);y2(k+1);y2(k+2);y2(k+3);y2(k+4)];
% eta3=[y3(k);y3(k+1);y3(k+2);y3(k+3);y3(k+4)];
% u(k+j)=[u1(k+j);u2(k+j);u3(k+j)]; j=0,1,2,3,4=n3-1

% CONDICIONES INICIALES
NN1 = 20; NN2=25; NN3=30; % ORDEN DE LOS VECTORES DE PARAMETROS
th1i=0.1*[alpha11 beta1_0 beta1_1 beta1_2 beta1_3 beta1_4]';
th2i=0.1*[alpha21 alpha22 beta2_0 beta2_1 beta2_2 beta2_3 beta2_4]';

```

```

th3i=0.1*[alpha31 alpha32 alpha33 beta3_0 beta3_1 beta3_2 beta3_3 beta3_4]';
alfa = 1000;
P1i = alfa*eye(NN1); P2i = alfa*eye(NN2); P3i = alfa*eye(NN3);
y1p1=0; y1p2=0; y1p3=0; y1p4=0; y1p5=0;
y2p1=0; y2p2=0; y2p3=0; y2p4=0; y2p5=0;
y3p1=0; y3p2=0; y3p3=0; y3p4=0; y3p5=0;
u1p1=0; u1p2=0; u1p3=0; u1p4=0; u1p5=0;
u2p1=0; u2p2=0; u2p3=0; u2p4=0; u2p5=0;
u3p1=0; u3p2=0; u3p3=0; u3p4=0; u3p5=0;
y = [0;0;0]; u = [0.1;0.1;0.1];
x = zeros(n,1); xe = x; % VECTOR DE ESTADO INICIAL
v = [0;0;0]; % ACCION INTEGRAL INICIAL
ref = [1;1;1]; % REFERENCIA
X=0.5*zeros(9,1);
% BUCLE DE CONTROL *****
for k = 1:N
D11=round(17.5+17.5*sin(6.5*k/N)); d11(k)=D11;
D12=round(17.5+17.5*cos(6.5*k/N)); d12(k)=D12;
D13=round(17.5+17.5*cos(6.5*k/N+pi/2)); d13(k)=D13;
D21=round(14+14*cos(6.5*k/N)); d21(k)=D21;
D22=round(12+12*sin(6.5*k/N)); d22(k)=D22;
D23=round(12.5+12.5*sin(6.5*k/N+pi/2)); d23(k)=D23;
D31=round(15+15*cos(6.5*k/N+pi*3/2)); d31(k)=D31;
D32=round(16+16*sin(6.5*k/N+pi*3/2)); d32(k)=D32;

% ECUACION DE ESTADO DEL PROCESO (CON TIEMPOS MUERTOS)
Gc=tf({K11 K12 K13;K21 K22 K23;K31 K32 K33},...
{[T11 1] [T12 1] [T13 1];[T21 1] [T22 1] [T23 1];[T31 1] [T32 1] [T33 1]},...
'iodelay',[D11 D12 D13;D21 D22 D23;D31 D32 0]);
ss1=ss(Gc);
Ac=ss1.a; Bc=ss1.b; Cc=ss1.c; Dc=ss1.d;

% MODELO DISCRETO DE ss1 (DEL PROCESO)
[G H Cp Dp]=c2dm(Ac,Bc,Cc,Dc,Ts);
X=G*X+H*u; % MODELO DEL PROCESO
y=Cp*X+Dp*u;
y1(k)=y(1); y2(k)=y(2); y3(k)=y(3);

% ESTIMACION DE PARAMETROS (METODO MCRM)
% y1(k+5)=alpha11*eta1+beta1_0*u(k)+beta1_1*u(k+1)+beta1_2*u(k+2)
% +beta1_3*u(k+3)+beta1_4*u(k+4)
% eta1=[y1(k);y1(k+1);y1(k+2);y1(k+3);y1(k+4)];
% u(k+j)=[u1(k+j);u2(k+j);u3(k+j)]; j=0,1,2,3,4=n1-1

psi1=[y1p5 y1p4 y1p3 y1p2 y1p1 ...
u1p5 u2p5 u3p5 u1p4 u2p4 u3p4 u1p3 u2p3 u3p3 u1p2 u2p2 u3p2 u1p1 u2p1 u3p1]';
rho1 = max(1,norm(psi1));
psin1 = psi1/rho1;
Nn1 = chol(P1i'); % Nn1'*Nn1 = P1i => Nn1*Nn1' = P1i'

```

```

S1 = inv(diag(Nn1*ones(NN1,1),0));
Ps1 = S1*Pli*S1;
psins1 = inv(S1)*psin1;
rt1 = 1 + psins1'*Ps1*psins1;
lamb1 = 1 - (rt1-sqrt(rt1^2-4*norm(Ps1*psins1)^2/trace(Ps1)))/2;
e1 = y1(k)/rho1 - psin1'*th1i;
j1 = psins1'*Ps1*psins1 + lamb1;
th1 = th1i + e1*inv(S1)*Ps1*psins1/j1; aa(k)=th1(3);
Hns1 = Ps1*psins1/j1;
Ps1 = (Ps1 - Hns1*psins1'*Ps1)/lamb1;
tt1 = abs(max(eig(Ps1))/min(eig(Ps1)) );
cmax = 100; cmin = 15;
if tt1 <= cmin,
    Pli = Ps1; th1i = th1;
elseif tt1 >= cmax,
    Nnew1 = chol(Ps1'); % Nnew1*Nnew1' = Ps1'
    Snew1 = inv(diag(Nnew1*ones(NN1,1),0));
    Psnew1 = Snew1*Ps1*Snew1;
    Pli = Psnew1; th1i = th1;
end

% y2(k+5)=alpha21*eta1+alpha22*eta2+beta2_0*u(k)+beta2_1*u(k+1)
%          +beta2_2*u(k+2)+beta2_3*u(k+3)+beta2_4*u(k+4)
% eta1=[y1(k);y1(k+1);y1(k+2);y1(k+3);y1(k+4)];
% eta2=[y2(k);y2(k+1);y2(k+2);y2(k+3);y2(k+4)];
% u(k+j)=[u1(k+j);u2(k+j);u3(k+j)]; j=0,1,2,3,4=n2-1

psi2=[y1p5 y1p4 y1p3 y1p2 y1p1 y2p5 y2p4 y2p3 y2p2 y2p1
u1p5 u2p5 u3p5 u1p4 u2p4 u3p4 u1p3 u2p3 u3p3 u1p2 u2p2 u3p2 u1p1 u2p1 u3p1]';
rho2 = max(1,norm(psi2));
psin2 = psi2/rho2;
Nn2 = chol(P2i'); % Nn2'*Nn2 = Pi2 => Nn2*Nn2' = P2i'
S2 = inv(diag(Nn2*ones(NN2,1),0));
Ps2 = S2*P2i*S2;
psins2 = inv(S2)*psin2;
rt2 = 1 + psins2'*Ps2*psins2;
lamb2 = 1 - (rt2-sqrt(rt2^2-4*norm(Ps2*psins2)^2/trace(Ps2)))/2;
e2 = y2(k)/rho2 - psin2'*th2i;
j2 = psins2'*Ps2*psins2 + lamb2;
th2 = th2i + e2*inv(S2)*Ps2*psins2/j2;
Hns2 = Ps2*psins2/j2;
Ps2 = (Ps2 - Hns2*psins2'*Ps2)/lamb2;
tt2 = abs(max(eig(Ps2))/min(eig(Ps2)) );
cmax = 100; cmin = 15;
if tt2 <= cmin,
    P2i = Ps2; th2i = th2;
elseif tt2 >= cmax,
    Nnew2 = chol(Ps2'); % Nnew2*Nnew2' = Ps2'
    Snew2 = inv(diag(Nnew2*ones(NN2,1),0));

```

```

    Psnew2 = Snew2*Ps2*Snew2;
    P2i = Psnew2; th2i = th2;
end

% y3(k+5)=alpha31*eta1+alpha32*eta2+alpha33*eta3+beta3_0*u(k)+beta3_1*u(k+1)
%           +beta3_2*u(k+2)+beta3_3*u(k+3)+beta3_4*u(k+4)
% eta1=[y1(k);y1(k+1);y1(k+2);y1(k+3);y1(k+4)];
% eta2=[y2(k);y2(k+1);y2(k+2);y2(k+3);y2(k+4)];
% eta3=[y3(k);y3(k+1);y3(k+2);y3(k+3);y3(k+4)];
% u(k+j)=[u1(k+j);u2(k+j);u3(k+j)]; j=0,1,2,3,4=n3-1

psi3=[y1p5 y1p4 y1p3 y1p2 y1p1 y2p5 y2p4 y2p3 y2p2 y2p1
      y3p5 y3p4 y3p3 y3p2 y3p1
u1p5 u2p5 u3p5 u1p4 u2p4 u3p4 u1p3 u2p3 u3p3 u1p2 u2p2 u3p2 u1p1 u2p1 u3p1]';
rho3 = max(1,norm(psi3));
psin3 = psi3/rho3;
Nn3 = chol(P3i'); % Nn3'*Nn3 = Pi3 => Nn3*Nn3' = P3i'
S3 = inv(diag(Nn3*ones(NN3,1),0));
Ps3 = S3*P3i*S3;
psins3 = inv(S3)*psin3;
rt3 = 1 + psins3'*Ps3*psins3;
lamb3 = 1 - (rt3-sqrt(rt3^2-4*norm(Ps3*psins3)^2/trace(Ps3)))/2;
e3 = y3(k)/rho3 - psin3'*th3i;
j3 = psins3'*Ps3*psins3 + lamb3;
th3 = th3i + e3*inv(S3)*Ps3*psins3/j3;
Hns3 = Ps3*psins3/j3;
Ps3 = (Ps3 - Hns3*psins3'*Ps3)/lamb3;
tt3 = abs(max(eig(Ps3))/min(eig(Ps3))) );
cmax = 100; cmin = 15;
if tt3 <= cmin,
    P3i = Ps3; th3i = th3;
elseif tt3 >= cmax,
    Nnew3 = chol(Ps3'); % Nnew3*Nnew3' = Ps3'
    Snew3 = inv(diag(Nnew3*ones(NN3,1),0));
    Psnew3 = Snew3*Ps3*Snew3;
    P3i = Psnew3; th3i = th3;
end

% RECUPERANDO LA ECUACION DE ESTADO ESTIMADA x(k+1)=AE*x(k)+BE*u(k)
% th1=[alphae11 betae1_0 betae1_1 betae1_2 betae1_3 betae1_4]';
% th2=[alphae21 alphae22 betae2_0 betae2_1 betae2_2 betae2_3 betae2_4]';
% th3=[alphae31 alphae32 alphae33 betae3_0 betae3_1 betae3_2 ...
%           betae3_3 betae3_4]';

alphae11=th1(1:5)';
alphae21=th2(1:5)'; alphae22=th2(6:10)';
alphae31=th3(1:5)'; alphae32=th3(6:10)'; alphae33=th3(11:15)';
AE11=[0 1 0 0 0
      0 0 1 0 0

```

```

        0 0 0 1 0
        0 0 0 0 1
        alphae11];
AE21=[0 0 0 0 0
       0 0 0 0 0
       0 0 0 0 0
       0 0 0 0 0
       alphae21];
AE22=[0 1 0 0 0
       0 0 1 0 0
       0 0 0 1 0
       0 0 0 0 1
       alphae22];
AE31=[0 0 0 0 0
       0 0 0 0 0
       0 0 0 0 0
       0 0 0 0 0
       alphae31];
AE32=[0 0 0 0 0
       0 0 0 0 0
       0 0 0 0 0
       0 0 0 0 0
       alphae32];
AE33=[0 1 0 0 0
       0 0 1 0 0
       0 0 0 1 0
       0 0 0 0 1
       alphae33];
AE=[AE11    Z    Z
     AE21 AE22    Z
     AE31 AE32 AE33];
% th1=[alphae11 betae1_0 betae1_1 betae1_2 betae1_3 betae1_4]';
% th2=[alphae21 alphae22 betae2_0 betae2_1 betae2_2 betae2_3 betae2_4]';
% th3=[alphae31 alphae32 alphae33 betae3_0 betae3_1 betae3_2 ...
%      betae3_3 betae3_4]';
betae1_0=th1(6:8)';
betae1_1=th1(9:11)';
betae1_2=th1(12:14)';
betae1_3=th1(15:17)';
betae1_4=th1(18:20)';
betae2_0=th2(11:13)';
betae2_1=th2(14:16)';
betae2_2=th2(17:19)';
betae2_3=th2(20:22)';
betae2_4=th2(23:25)';
betae3_0=th3(16:18)';
betae3_1=th3(19:21)';
betae3_2=th3(22:24)';
betae3_3=th3(25:27)';

```

```

betae3_4=th3(28:30)';

BE(1,:)=betae1_4;
BE(2,:)=betae1_3+AE(5,5)*BE(1,:);
BE(3,:)=betae1_2+AE(5,5)*BE(2,:)+AE(5,4)*BE(1,:);
BE(4,:)=betae1_1+AE(5,5)*BE(3,:)+AE(5,4)*BE(2,:)+AE(5,3)*BE(1,:);
BE(5,:)=betae1_0+AE(5,5)*BE(4,:)+AE(5,4)*BE(3,:) ...
        +AE(5,3)*BE(2,:)+AE(5,2)*BE(1,:);
BE(6,:)=betae2_4;
BE(7,:)=betae2_3+AE(10,5)*BE(1,:) +AE(10,10)*BE(6,:);
BE(8,:)=betae2_2+AE(10,5)*BE(2,:) +AE(10,4)*BE(1,:) ...
        +AE(10,10)*BE(7,:)+AE(10,9)*BE(6,:);
BE(9,:)=betae2_1+AE(10,5)*BE(3,:)+AE(10,4)*BE(2,:)+AE(10,3)*BE(1,:)
        +AE(10,10)*BE(8,:)+AE(10,9)*BE(7,:)+AE(10,8)*BE(6,:);
BE(10,:)=betae2_0 ...
        +AE(10,5)*BE(4,:)+AE(10,4)*BE(3,:)+AE(10,3)*BE(2,:)+AE(10,2)*BE(1,:)
        +AE(10,10)*BE(9,:)+AE(10,9)*BE(8,:)+AE(10,8)*BE(7,:)+AE(10,7)*BE(6,:);
BE(11,:)=betae3_4;
BE(12,:)=betae3_3+AE(15,5)*BE(1,:)+AE(15,10)*BE(6,:)+AE(15,15)*BE(11,:);
BE(13,:)=betae3_2+AE(15,5)*BE(2,:) +AE(15,4)*BE(1,:) ...
        +AE(15,10)*BE(7,:) +AE(15,9)*BE(6,:) ...
        +AE(15,15)*BE(12,:)+AE(15,14)*BE(11,:);
BE(14,:)=betae3_1+AE(15,5)*BE(3,:) +AE(15,4)*BE(2,:)+AE(15,3)*BE(1,:)
        +AE(15,10)*BE(8,:) +AE(15,9)*BE(7,:)+AE(15,8)*BE(6,:)
        +AE(15,15)*BE(13,:)+AE(15,14)*BE(12,:)+AE(15,13)*BE(11,:);
BE(15,:)=betae3_0 ...
+AE(15,5)*BE(4,:) +AE(15,4)*BE(3,:)+AE(15,3)*BE(2,:)+AE(15,2)*BE(1,:) ...
+AE(15,15)*BE(14,:)+AE(15,14)*BE(13,:)+AE(15,13)*BE(12,:)+AE(15,12)*BE(11,:)
+AE(15,10)*BE(9,:)+AE(15,9)*BE(8,:)+AE(15,8)*BE(7,:)+AE(15,7)*BE(6,:);

% CALCULO DE LA GANACIA Ko DEL OBSERVADOR
% EQUACION DE RICATTI
% Po = diag(0,3);
% for i = 1:20
%   Po = Qo + GE*Po*GE' - GE'*Po*CE'*inv(Ro+CE*Po*CE')*CE*Po*GE';
% end
% Ko = inv(Ro+CE*Po*CE')*CE*Po*GE';
Ko = dlqe(AE,eye(n),C,Qo,Ro);

% ESTIMACION DE ESTADOS
xe = AE*xe + BE*u + Ko*(y-C*xe);

% CALCULO DE LA GANANCIA Ka DEL CONTROLADOR
Aa = [AE zeros(n,r);-C*AE eye(r)];
Ba = [BE;-C*BE];

% EQUACION DE RICATTI
% Pa = diag(0,4);
% for i = 1:20

```

```

% Pa = Qa + Ga'*Pa*Ga - Ga'*Pa*Ha*inv(Ra+Ha'*Pa*Ha)*Ha'*Pa*Ga;
% end
% Ka = inv(Ra+Ha'*Pa*Ha)*Ha'*Pa*Ga;
Ka = dlqr(Aa,Ba,Q,R);
K = Ka(:,1:n); % GANANCIA OPTIMA K PROPORCIONAL
KI= -Ka(:,n+1:n+r); % GANANCIA OPTIMA KI INTEGRAL
% CALCULO DE LA LEY DE CONTROL
v = v + ref - y; % ACCION INTEGRAL
r1(k)=ref(1); r2(k)=ref(2); r3(k)=ref(3);
u = - K*x + KI*v;
u1(k)=u(1); u2(k)=u(2); u3(k)=u(3);
% if k >= 1
%   r=[0.1 3]';
% end
% if k > MM/5
%   r=[0.2 1]';
% end
% if k > 3*MM/5
%   r=[0.3 2]';
% end

% UPDATE
y1p1=y1(k); y1p2=y1p1; y1p3=y1p2; y1p4=y1p3; y1p5=y1p4;
y2p1=y2(k); y2p2=y2p1; y2p3=y2p2; y2p4=y2p3; y2p5=y2p4;
y3p1=y3(k); y3p2=y3p1; y3p3=y3p2; y3p4=y3p3; y3p5=y3p4;
u1p1=u1(k); u1p2=u1p1; u1p3=u1p2; u1p4=u1p3; u1p5=u1p4;
u2p1=u2(k); u2p2=u2p1; u2p3=u2p2; u2p4=u2p3; u2p5=u2p4;
u3p1=u3(k); u3p2=u3p1; u3p3=u3p2; u3p4=u3p3; u3p5=u3p4;

end % FIN DEL BUCLE *****

% GRAFICOS
t = linspace(0,Ts*N,N);
figure(1)
subplot(2,1,1)
plot(t,y1,'r',t,r1,'b'); ylabel('Salida y1');
grid;
subplot(2,1,2)
plot(t,u1,'b'); ylabel('Control u1'); grid;
xlabel('Tiempo en minutos')
print -deps -f nava4y1u1

figure(2)
subplot(2,1,1)
plot(t,y2,'r',t,r2,'b'); ylabel('Salida y2'); grid;
subplot(2,1,2)
plot(t,u2,'b'); ylabel('Control u2'); grid;
xlabel('Tiempo en minutos')
print -deps -f nava4y2u2

```

BIBLIOGRAFÍA

- [1] Camacho, Eduardo F. y Bordons, Carlos. *Model Predictive Control*. Springer-Verlag Berlin Heidelberg New York, 1999.
- [2] Isermann, Rolf. *Digital Control System, Vol.: Stochastic Control, Multi-variable Control, Adaptive Control, Applications*. Springer-Verlag Second Edition, 1991.
- [3] MathWorks, Inc. *MATLAB Reference Guide*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.
- [4] Narendra, Kumpa S. y Annaswamy, Anuradha M. *Stable Adaptive Systems*. Prentice Hall, 1989.
- [5] Ogata, Katushito. *Designing Linear Control Systems with MATLAB*. Prentice Hall Englewood Cliffs New Jersey, 1994.
- [6] Ogata, Katushito. *Sistemas de Control en Tiempo Discreto*. Prentice Hall Hispanoamericana S.A. México, Nueva York Segunda Edición, 1996.
- [7] Prett, D. M. y Morari, M. *Shell Process Control Workshop*. Butterworths, 1987.
- [8] Rojas-Moreno, Arturo. *Design of Self-Tuning Controllers for Processes Having Multiple Unknown Time-Varying Delays*. PhD dissertation, Utah State University, Logan UT, June 1995.
- [9] Rojas-Moreno, Arturo. *Control Avanzado-Diseño y Aplicaciones en Tiempo Real*. Publicación Independiente, 2001.
- [10] Sripada, N. Rao y Fisher, D. Grant. "Improved Least Squares Identification," *Int. J. Control*, vol. 46, no. 6, 1889-1913 (Dic 1987).

```
figure(3)
subplot(2,1,1)
plot(t,y3,'r',t,r3,'b'); ylabel('Salida y3'); grid;
subplot(2,1,2)
plot(t,u3,'b'); ylabel('Control u3'); grid;
xlabel('Tiempo en minutos')
print -deps -f nava4y3u3
```

```
figure(4)
subplot(4,2,1)
plot(t,d11); ylabel('D11 [min]'); grid;
subplot(4,2,2)
plot(t,d12); ylabel('D12 [min]'); grid;
subplot(4,2,3)
plot(t,d13); ylabel('D13 [min]'); grid;
subplot(4,2,4)
plot(t,d21); ylabel('D21 [min]'); grid;
subplot(4,2,5)
plot(t,d22); ylabel('D22 [min]'); grid;
subplot(4,2,6)
plot(t,d23); ylabel('D23 [min]'); grid;
subplot(4,2,7)
plot(t,d31); ylabel('D31 [min]'); grid;
xlabel('Tiempo en minutos')
subplot(4,2,8)
plot(t,d32); ylabel('D32 [min]'); grid;
xlabel('Tiempo en minutos')
print -deps -f tdelays
```