ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

TESIS DE GRADO

"CONTROL MULTIVARIABLE EN TIEMPO REAL"

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN ELECTRONICA Y CONTROL

CESAR AUGUSTO SARMIENTO BRAVO

FEBRERO, 1991

Dedicatoria

A mis padres y hermanos quienes supieron apoyarme y alentarme en cada momento de mi vida.

Agradecimiento

Mi sincero agradecimiento al Ing. Patricio Burbano R. por su acertada dirección en la presente tesis.

Certifico que el presente trabajo ha sido desarrollado en su totalidad por el Sr. César Augusto Sarmiento Bravo.

Ing. Patricio Burbano R.

Director

CONTENIDO

		Pag.
1	INTRODUCCION	1
1.1	Introducción	1
1.2	Estrategias de control multivariable	5
1.3	Simulación y control en tiempo real	8
2	SIMULACION Y CONTROL DE SISTEMAS MULTIVARIABLES	10
2.1	Control multilazo tipo P.I.D.	10
2.2	Control multivariable mediante desacoplamiento	19
2.3	Regulador cuadrático lineal	34
2.4	Estructura del software desarrollado	51
2.5	Software para control multilazo	54
2.6	Software para desacoplamiento	60
2.7	Software para regulador cuadrático lineal	69
2.8	Software para gráficos	77
2.9	Rutinas de uso general	78
2.10	Software para tiempo real	79
2.11	Simulación mediante el programa CC	88
3	RESULTADOS DE SIMULACION Y CONTROL EN TIEMPO REAL	100

3.1 Resultados de simulación de los programas implementados 105
3.2 Resultados de control en tiempo real 129
3.3 Resultados de simulación del programa CC 143

4 CONCLUSIONES Y RECOMENDACIONES 149
4.1 Conclusiones 149
4.2 Recomendaciones 153

BIBLIGRAFIA 154

APENDICES
Apéndice A : Manual de uso A1
Apéndice B : Macros del programa CC B1
Apéndice C : Listado de software C1

1 INTRODUCCION

1.1 ESTRATEGIAS DE CONTROL MULTIVARIABLE

1.2 SIMULACION Y CONTROL EN TIEMPO REAL

1.-INTRODUCCION:

1.1 Introducción .-

Considerando el creciente avance del control moderno en general y del control digital en particular se hace necesario en nuestro medio el desarrollo de un paquete de SOFTWARE que permita utilizar los diversos algoritmos de control para su aplicación en casos reales, de tal modo que se puedan ensayar diversas estrategias, las mismas que se implementarían en forma práctica mediante el desarrollo de rutinas en tiempo real.

Por otro lado el presente trabajo tiene una utilización inmediata en la enseñanza de laboratorio de control moderno, control óptimo, control de procesos y en particular de control discreto; pero además su utilización bién puede estar orientada al campo industrial, pues se podrían ensayar diversos algoritmos de control para una aplicación del tipo indicado.

El presente trabajo de tesis tiene por objeto utilizar diversas estrategias de control las cuales serán aplicadas para sistemas multivariables ; las estrategias indicadas se reducen a algoritmos implementados en forma discreta ya que se desea realizar un control de índole digital.

El software desarrollado comprende dos módulos perfectamente diferenciados, el primer módulo tiene que ver con la parte correspondiente a SIMULACION , mientras que el segundo módulo tiene por objeto la implementación de técnicas de control en TIEMPO REAL. En lo referente a simulación se pone énfasis en la parte gráfica a fin de poder observar el comportamiento de las diferentes plantas sobre las cuales se va a trabajar; en lo correspondiente a tiempo real se realiza un programa base con el cual se ensayan las estrategias de control simuladas previamente en el módulo respectivo, para su aplicación directa en sistemas físicos multivariables.

Adicionalmente, en lo que a simulación se refiere se utilizará el paquete de control CC para explorar sus potencialidades en cuanto a sistemas multivariables, ya que estos sistemas constituyen el tema central del presente trabajo. Cabe indicar que el programa de control CC constituye un software especializado con un alto número de opciones de análisis, por lo que se utilizará apenas una pequeña parte de dichas opciones y en los aspectos de interés.

El trabajo que se pone a consideración consta de diversas tópicos los cuales han sido estructurados cláramente en varios capítulos:

- En el capítulo uno se realiza una breve introducción de los alcances y objetivos que se persiguen, así como de las diversas estrategias de control que existen y cuales de ellas han sido elegidas para ser desarrolladas tanto a nivel de simulación como en lo que a tiempo real se refiere.

- En el capitulo dos se realiza un análisis teórico de las diversas estrategias utilizadas, abordándose de este modo la base necesaria para el desarrollo del software respectivo. En este capitulo se presentan los diversos algoritmos y la estructura del software desarrollado. Por otro lado se revisan también las rutinas correspondientes a tiempo real en lo que a sistemas multivariables se refiere. Se indica también además los tópicos correspondientes al programa CC que han sido elegidos como parte integrante del presente trabajo de tesis.

- En el capitulo tres se indican los diversos resultados que se han obtenido tanto con el programa propio desarrollado así como con la utilización del paquete de control CC; en la parte de simulación se hace incapié en lo referente a gráficos. Los resultados de tiempo real se los puede apreciar en forma gráfica mediante la ayuda de un registrador.

- Como siguiente capitulo se tiene la parte correspondiente a conclusiones y recomendaciones que vienen a sintetizar el trabajo desarrollado; en este punto, se realizan los comentarios provenientes de la elaboración del trabajo en si y del análisis de los resultados que se han obtenido a través de la ejecución del tema en cuestión.

- Finalmente se tiene la parte correspondiente a la bibliografía utilizada y los diversos apéndices necesarios para una correcta complementación del presente tema de tesis.

Se debe indicar en forma complementaria que el programa se ha desarrollado en un computador IBM PS modelo 60 con monitor a color y utilizando el programa QUICK BASIC , versión 4.5 ; se ha usado también en lo referente a tiempo real el paquete QUICK 500 que es afín a la versión de quick basic indicada.

1.2 Estrategias de control multivariable.-

Diversos tópicos de la teoría de control tienen que ver con los sistemas multivariables, así se tiene: control moderno, control óptimo, control de procesos,etc. Es por esto que existen muchas estrategias de control que abordan el campo multivariable , las mismas que tienen un verdadero sustento teórico y pueden ser desarrolladas en forma discreta para realizar de esta manera un control de tipo digital multivariable.

Las estrategias en mención que conllevan a algoritmos de control de diversa índole son extensas y variadas, por lo que debe delimitarse el campo en el cual se va a trabajar. Para tener una idea más clara, es importante realizar un esquema de las diversas técnicas existentes, asi, se puede hacer referencia a la figura 1.1: técnicas de control multivariable, que nos presenta varios algoritmos los mismos que pueden ser ensayados tanto a nivel de simulación como en el campo de tiempo real.

De las técnicas en mención sa ha tomado para el desarrollo del presente trabajo tres técnicas a ser ensayadas: P.I.D. multilazo, desacoplamiento, y regulador cuadrático lineal; los dos primeros tópicos están incluidos en el problema de seguimiento, mientras que el tercero constituye un problema de regulación.

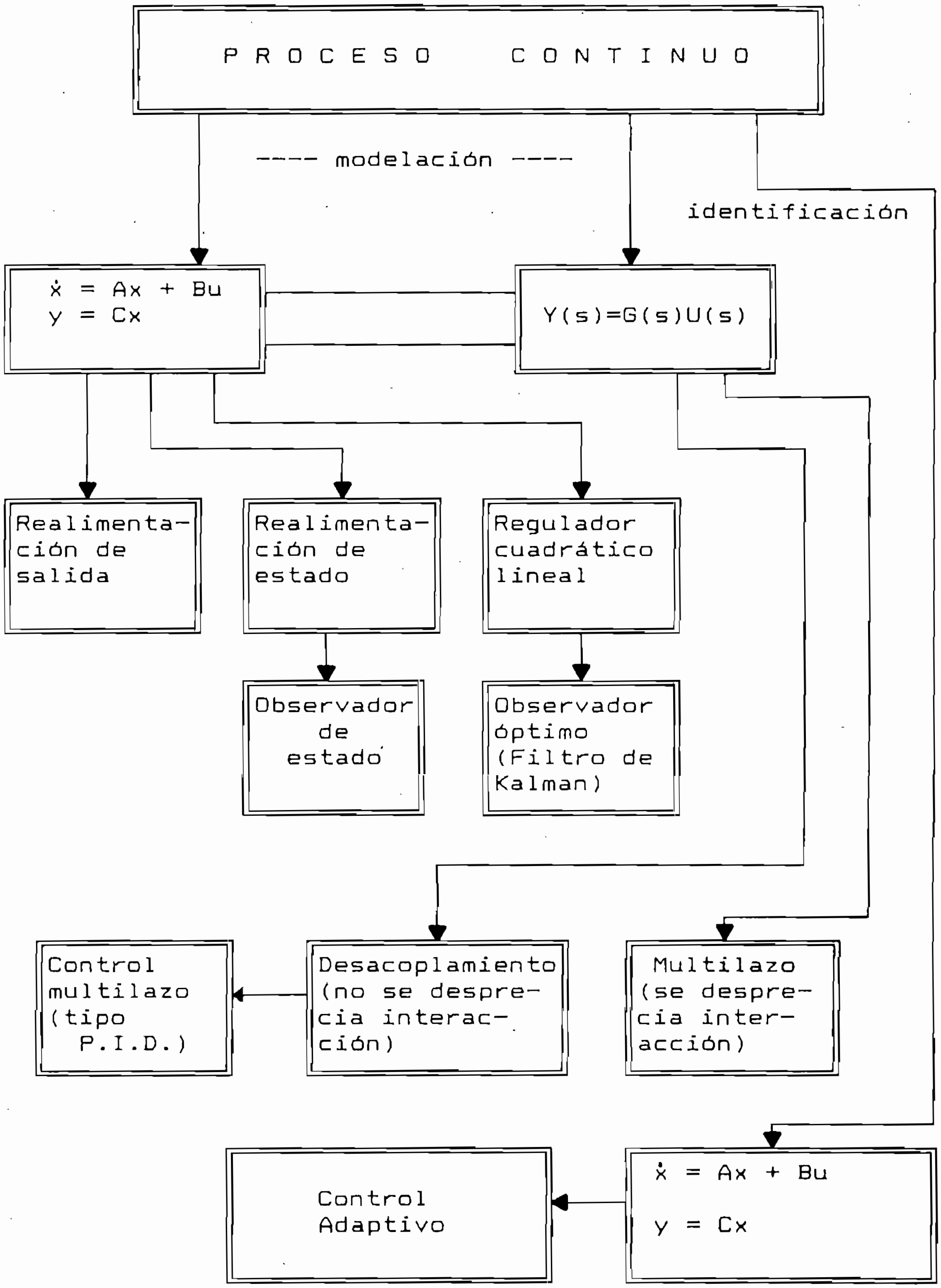


Fig. 1.1.- Técnicas de control multivariable

Se trata por lo tanto de desarrollar el software de las técnicas mencionadas para control multivariable de tal forma de tener un criterio referente a las ventajas y desventajas de las diversas estrategias ensayadas.

1.3 Simulación y control en tiempo real.-

Uno de los principales objetivos del software desarrollado es el soporte en la enseñanza del control moderno, control óptimo, etc. en una de sus aplicaciones. Para esto, primeramente se realizan programas propios que permitan la simulación mediante la utilización de las técnicas multivariables seleccionadas, esto es: P.I.D. multilazo, desacoplamiento y regulador cuadrático lineal. También en lo que a simulación se refiere, se exploran las potencialidades del paquete CC en lo correspondiente a sistemas multivariables, afines al presente trabajo.

Una vez realizado el software correspondiente a simulación, se aplican las técnicas ensayadas a control en tiempo real, mediante los resultados provenientes de la simulación y con la utilización del equipo de adquisición de datos y control KEITHLEY 500A y la ayuda del paquete de software QUICK 500 que permite el enlace entre el equipo mencionado y el computador bajo el esquema indicado en la figura 1.2: esquema de control en tiempo real.

Por otro lado cabe indicar que los cálculos realizados se los ejecuta en forma <u>off line</u> en la simulación, mientras que el control en sistemas físicos ya se lo implementa en forma <u>on line</u> , es decir que el computador forma parte del lazo de control,en un esquema de control digital directo (DDC).

Finalmente es importante mencionar que la principal ventaja del control digital, es que el algoritmo de control se realiza por software y si se desea modificar el tipo de control, solo se requiere cambiar el programa, según el algoritmo utilizado; es decir, se modifica el software y no el hardware.

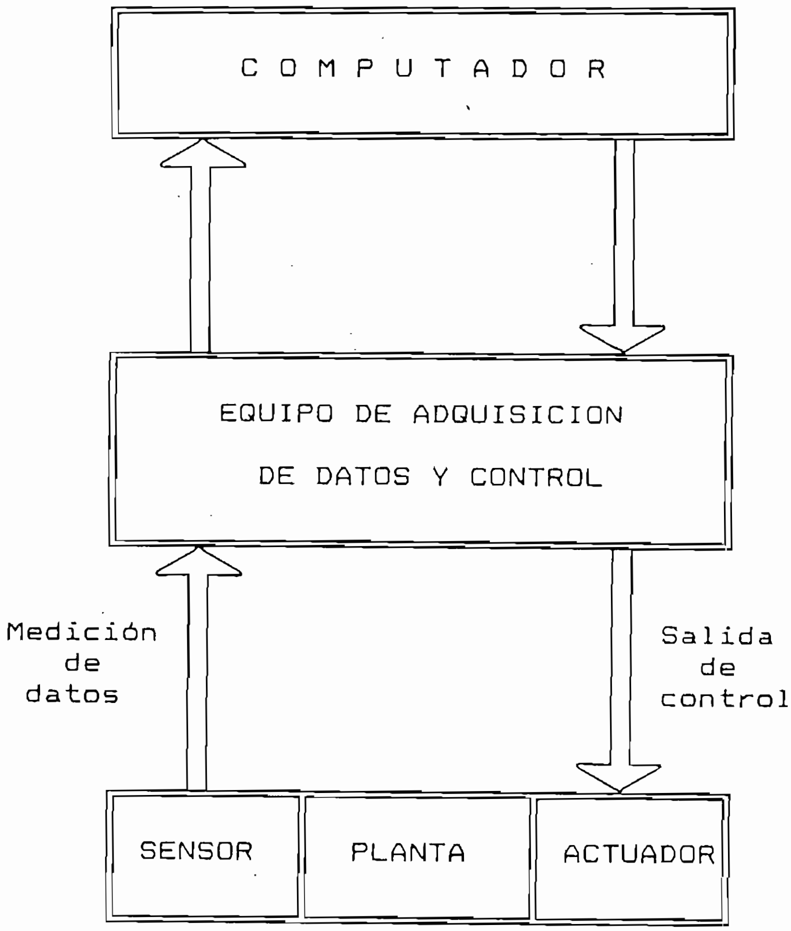


Fig. 1.2.- Esquema de control en tiempo real

2 SIMULACION Y CONTROL EN TIEMPO REAL

2.1 CONTROL MULTILAZO TIPO P.I.D.

2.2 CONTROL MULTIVARIABLE MEDIANTE DESACOPLAMIENTO

2.3 REGULADOR CUADRATICO LINEAL

2.4 ESTRUCTURA DEL SOFTWARE DESARROLLADO

2.5 SOFTWARE PARA CONTROL MULTILAZO

2.6 SOFTWARE PARA DESACOPLAMIENTO

2.7 SOFTWARE PARA REGULADOR CUADRATICO LINEAL

2.8 SOFTWARE PARA GRAFICOS

2.9 RUTINAS DE USO GENERAL

2.10 SOFTWARE PARA TIEMPO REAL

2.11 SIMULACION MEDIANTE EL PROGRAMA CC

2.- SIMULACION Y CONTROL DE SISTEMAS MULTIVARIABLES

En el presente capítulo se realiza un análisis de las diversas técnicas que van a ser ensayadas tanto a nivel de simulación como en tiempo real; las técnicas en mención son: P.I.D. multilazo, desacoplamiento y regulador cuadrático lineal.

2.1.- CONTROL MULTILAZO TIPO P.I.D.

Una de las diversas técnicas existentes en la ingeniería de control constituye el control multilazo el mismo que consiste en considerar una planta con el mismo número de entradas y salidas de tal forma que se asume que cada salida es independientemente controlada por la entrada correspondiente; lógicamente, esta consideración es una aproximación que permite obtener buenos resultados si el sistema tiene un débil acoplamiento o interacción ; en tal caso se supone la existencia de lazos independientes.

En forma práctica se utilizará el equipo de adquisición de datos y control KEITHLEY 500A por lo que debido a sus propias limitaciones se tendrá un máximo de cinco entradas y cinco salidas para poder implementar este tipo de control en tiempo real.

La figura 2.1. control multilazo de un sistema multivariable da una idea clara de los fines que se persiguen. En este caso se puede considerar a la planta con varias salidas y_1, y_2, y_3 y varias entradas u1 ,u2, u3 de tal forma que despreciando las interaciones (en el caso que sean débiles) se puede considerar varios lazos independientes.Cada uno de estos lazos deben ser

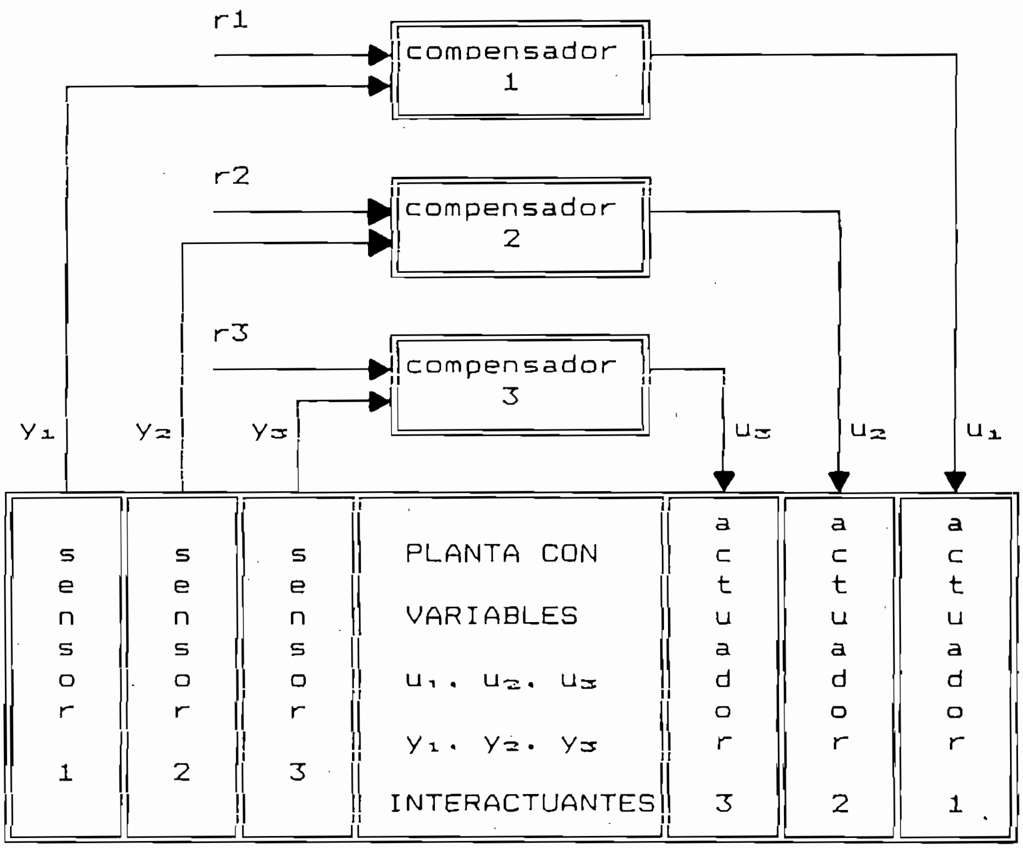


Fig. 2.1 Control multilazo de un sistema multivariable

controlados por lo que se puede ensayar una diversidad de compensadores que tienen por objeto obtener la respuesta deseada de acuerdo a diversos requerimientos. Se ha optado ensayar en cada uno de estos casos un control proporcional-integral-derivativo (P.I.D.), el mismo que será implementado en tiempo real mediante una previa discretización.

Un control tipo P.I.D. se lo tiene bajo el esquema de control que se presenta en la figura 2.2, control tipo P.I.D., con sus diferentes variables.

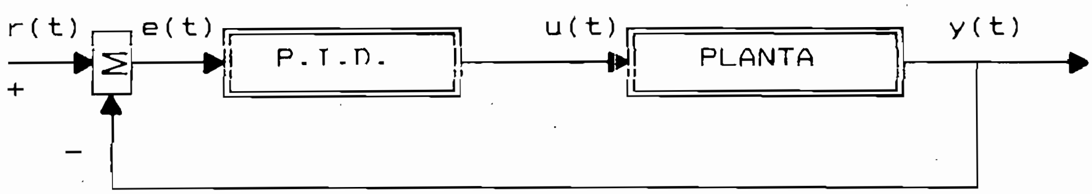


Fig. 2.2 Control tipo P.I.D.

El control proporcional-integral-derivativo (P.I.D.) actua sobre la señal de error e(t), la misma que constituye la diferencia entre la señal de referencia r(t) y la señal de salida de la planta y(t). El control proporcional, simplemente multiplica la señal de error por la ganancia proporcional K_P; el control integral multiplica la ganancia integral K_I por la integral del error y el control derivativo, genera una señal, la cual es proporcional al tiempo derivativo de la señal de error, aquí, se multiplica la derivada del error por la ganancia derivativa K_D. [KUO,B., 1980, pp. 509-514].

La función del control integral es proveer una acción que reduce el error en estado estable, mientras que la acción derivativa provee una acción anticipatoria que mejora la respuesta transitoria.

Lo indicado se puede expresar mediante la llamada ley de control, la misma que dá la relación entre el error e(t) y el control u(t) que constituyen respectivamente la entrada y la salida del controlador:

$$u(t) = K_P * e(t) + K_I * \int_0^t e(t) * dt + K_D * de(t)/dt \qquad [\ 2.1\ ]$$

Y en términos de Laplace se tiene:

$$U(s) = (K_P + K_I/s + K_D * s)E(s) \qquad [\ 2.2\ ]$$

El mismo principio del control P.I.D. puede ser aplicado al control digital; el control proporcional es el mismo que en el caso continuo , mientras que existen diferentes caminos de discretización del control integral y del derivativo para la ecuación 2.2 ; si se toma el método del trapecio para el primero y la definición matemática de derivada para el segundo se obtienen los siguientes resultados en la transformada Z:

$$U(z) = (K_P + (K_I * T/2) * (z+1)/(z-1) + (K_D/T) * (z-1)/z) * E(z) \qquad [\ 2.3\ ]$$

Donde: T = Periodo de muestreo

Reduciendo términos semejantes en la ecuación 2.3 y pasando al dominio del tiempo se obtiene la expresión de la ley de control en ecuación de diferencias:

$$u[kT] = u[(k-1)T] + b_o*e[kT] + b1*e[(k-1)T] + b2*e[(k-2)T] \qquad [\ 2.4\ ]$$

Donde:

$$b0 = KP + KI*T/2 + KD/T$$

$$b1 = KI*T/2 - KP - 2*KD/T$$

$$b2 = KD/T$$

En los párrafos precedentes se han desarrollado las expresiones matemáticas del compensador P.I.D. para el caso de sistemas univariables; por lo que debe resaltarse que para el caso multivariable puede ser aplicado este método de control en forma multilazo toda vez que el sistema tenga una débil interacción entre las entradas y las salidas; por otro lado cabe mencionar que al variar una referencia se tiene variaciones en todas las salidas.

Si en algún proceso se tiene lazos de control independientes univariables, es decir existen diferentes salidas sin acoplamiento o interacción, el caso por ejemplo de diferentes variables que corresponden a sistemas independientes, para este caso se aplica el control multilazo de dichos sistemas independientes con toda propiedad, esto es sin ninguna aproximación, según se muestra en la figura 2.3.

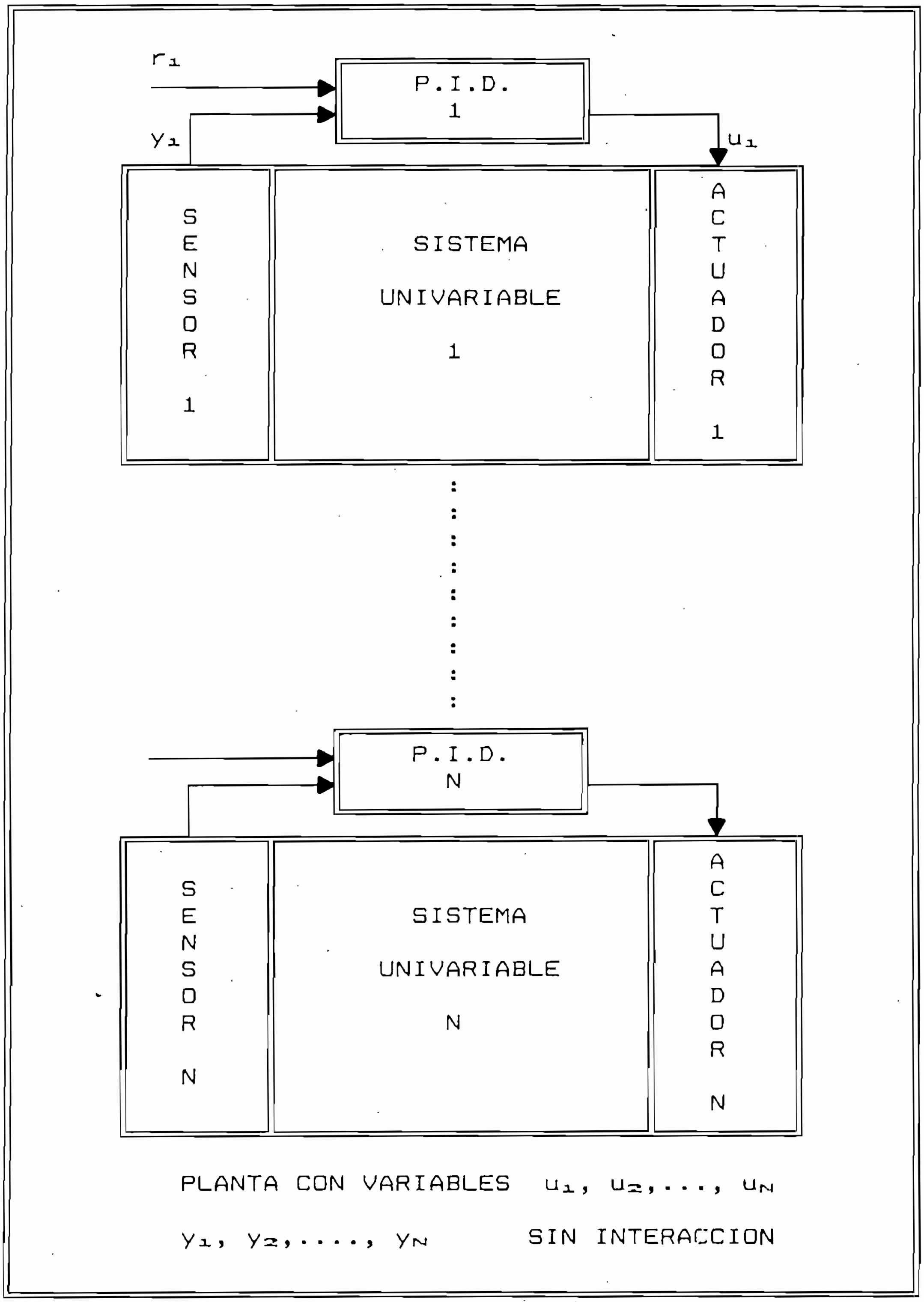


Fig. 2.3 Control multilazo de sistemas independientes univariables

A manera de ejemplo, se ilustra este esquema de control con un sistema multivariable cuya función de transferencia es:

$$G(s) = \begin{bmatrix} \dfrac{1}{2s+1} & 0 \\ \dfrac{0.2}{s+5} & \dfrac{1}{s+1} \end{bmatrix}$$

Este sistema puede ser esquematizado de la siguiente manera:

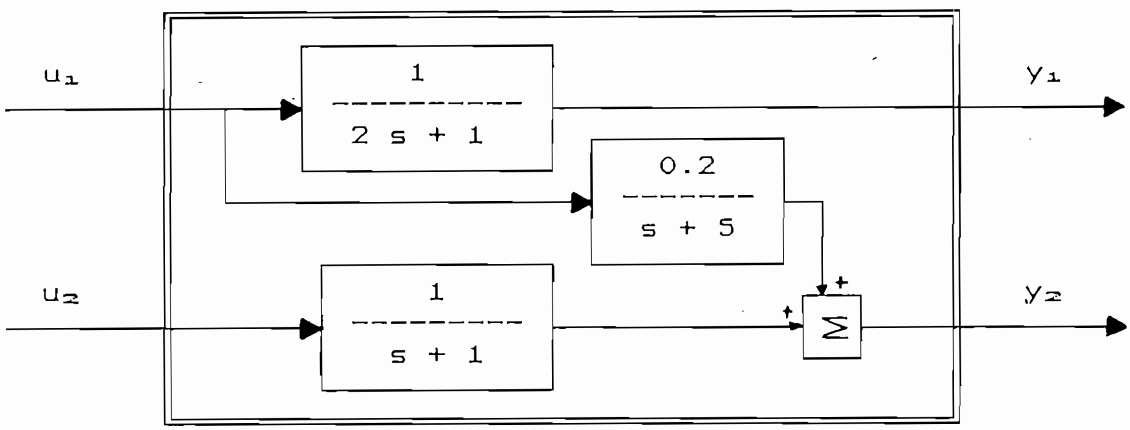


Fig. 2.4 Sistema multivariable débilmente acoplado

El sistema indicado posee un débil acoplamiento, por lo cual se realiza un control multilazo tipo P.I.D.; para esto, se procede a despreciar la interacción por lo que se considera los elementos de la diagonal de la matriz función de transferencia de lazo abierto para el diseño de los compensadores de cada lazo.

Para el efecto se consideran los siguientes lazos:

$$\text{Lazo 1:} \quad G_1(s) = \frac{1}{2\,s + 1} = \frac{0.5}{s + 0.5}$$

$$\text{Lazo 2:} \quad G_2(s) = \frac{1}{s + 1}$$

En este caso se realizará para el ejemplo una compensación tipo P.I., este compensador presenta la siguiente función de transferencia:

$$G_c(s) = K_p + \frac{K_I}{s} = \frac{(s\,K_p + K_I)}{s} = \frac{K_p}{s}\,(\,s + K_I/K_p\,)$$

Luego, para el compensador uno, se trata de minimizar la presencia del polo ubicado en -0.5, por lo cual se realiza la siguiente consideración de diseño: $K_p/K_I = 0.5$ que determina la siguiente función de transferencia de lazo abierto:

$$G_{1a}(s) = \frac{0.5\;K_p}{s}$$

que da como resultado la siguiente función de transferencia de lazo cerrado:

$$G_{1c}(s) = \frac{0.5\,K_p}{s + 0.5\,K_p}$$

Ahora bién, se desea que el sistema sea más rápido, por lo que se quiere tener un polo de lazo cerrado igual a −2; entonces se tiene el siguiente resultado para el compensador 1:

$$K_p = 4 \qquad\qquad K_I = 8$$

Siguiendo el mismo esquema de desarrollo que en el lazo 1 y minimizando el efecto del polo −1, se tiene:

$$G_{2a}(s) = \frac{K_p}{s} \qquad\qquad (K_p/K_I = 1)$$

$$G_{2c}(s) = \frac{K_p}{s + K_p}$$

Luego, se desea tener el polo de lazo cerrado en −3, con lo que se obtiene el siguiente resultado para el compensador dos:

$$K_p = 3 \qquad\qquad K_I = 3$$

Debe aclararse que en un caso real no existe la anulación de polos, sin embargo, el efecto de estos puede ser disminuido con un compensador que tienda a anularlos. Por otro lado, es importante mencionar que el cálculo de los compensadores es aproximado, ya que se ha despreciado la interacción, es por esto que en un caso real, se debe realizar reajustes en la implementación.

2 2.- CONTROL MULTIVARIABLE MEDIANTE DESACOPLAMIENTO

En forma general se tiene que un cambio de una entrada en un proceso multivariale, produce cambios en todas las salidas del proceso; lo que en muchos casos dificulta la aplicaión de diversos tipos de control. [LEIGH,E., 1985, p. 291]

En base a la consideración anterior, seria deseable que en un sistema, la variación de una señal de entrada determine un cambio únicamente en una salida, sin que las restantes salidas se vean afectadas; esto es lo que se conoce con el nombre de <u>desacoplamiento</u> y se desarrollará en los párrafos siguientes.

Se debe indicar que anteriormente se ha realizado una tesis completa de desacoplamiento,esta es:<u>Desacoplamiento para sistemas continuos en el tiempo</u> del Ing.Evelio Granizo; en la tesis mencionada se realiza un análisis de las técnicas existentes así como una demostración profunda de los diversos algoritmos correspondientes al tema de desacoplamiento. Sin embargo el objetivo de la presente tesis <u>Control multivariable en tiempo real</u> es ensayar una de las técnicas de desacoplamiento en <u>tiempo real</u> , por lo que, como el análisis teórico profundo ya se lo ha realizado en la tesis anterior mencionada, en la presente tesis se expone la base teórica necesaria para una clara comprensión del desarrollo del trabajo en cuestión,sin profundizar en la demostración de los diversos algoritmos.

Primeramente debe indicarse que para que un sistema pueda ser desacoplado se requiere indispensablemente que el número de entradas sea igual al número de salidas; si no se cumple esta condición, el sistema no podrá ser desacoplado.

Existen diversas técnicas de desacoplamiento, entre éstas, tres son las más utilizadas en el campo del control moderno:

- Desacoplamiento mediante matriz función de transferencia.
- Desacoplamiento mediante realimentación de estado.
- Desacoplamiento mediante realimentación de salida.

La primera técnica de desacoplamiento presenta una gran dificultad en la implementación computacional, principalmente al realizar la inversión de matrices literales en términos de Laplace. Haciendo referencia a las dos siguientes técnicas de desacoplamiento, puesto que generalmente el número de variables de estado es mayor que el número de variables de salida existe mayor campo de manipulación en la realimentación de estado que en la realimentación de salida. [GRANIZO,E., 1988, p. 18]

Para diseño de sistemas, el desacoplamiento por realimentación de estado es más aconsejable que el desacoplamiento por realimentación de salida, ya que por medio del primero se puede conseguir simultaneamente el desacoplamiento y estabilidad del sistema, mientras que en la segunda técnica se necesita prácticamente un precompensador para estabilizar dicho sistema.[GRANIZO,E.., 1988, p. 283]

El esquema de un sistema representado mediante variables de estado, se expresa de la siguiente manera:

$$\dot{x} = A\,x + B\,u \qquad [\,2.5\,]$$

$$y = C\,x + D\,u$$

Donde, considerando que se tiene igual número de entradas que de salidas, condición necesaria para desacoplamiento, se tiene:

x = vector de estado (n x 1)

y = vector de salida (m x 1)

u = vector de entrada (m x 1)

A = matriz que define el sistema (n x n)

B = matriz que define el sistema (n x m)

C = matriz que define el sistema (m x n)

D = matriz que define el sistema (m x m)

n = número de variables de estado

m = número de entradas = número de salidas

En el desarrollo posterior que se presenta se considerará que la matriz D=0 lo cual se satisface en sistemas reales.

En la técnica de desacoplamiento por realimentación de estado se tiene que la ley de control u se obtiene de la siguiente manera:

$$u = E\,r + F\,x \qquad [\,2.6\,]$$

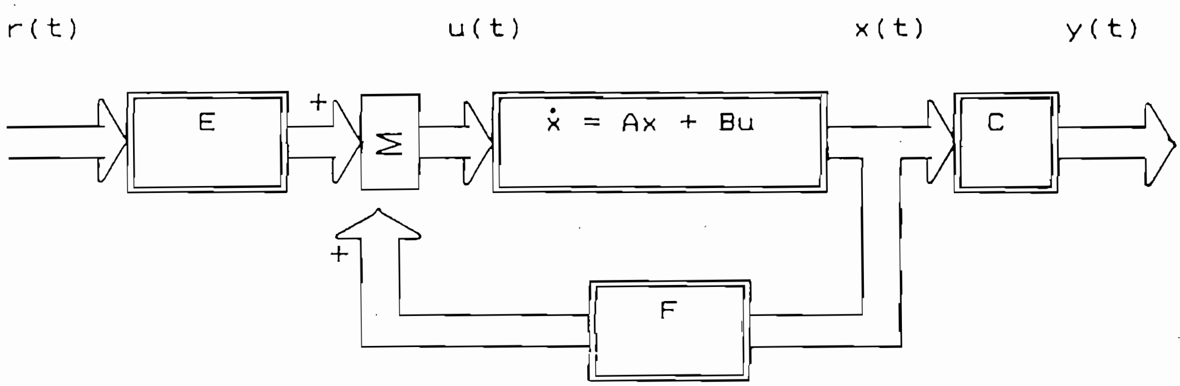
Donde:

r = vector de control (m x 1)

E = matriz de precompensación (m x m)

F = matriz de realimentación (m x n)

El esquema de realimentación de estado para desacoplamiento se muestra en la figura 2.5.



Fiq. 2.5 Desacoplamiento por realimentación de estado

Mediante la realimentación expuesta, se expresa el sistema en lazo cerrado de la siguiente forma:

$$\dot{x} = (\,A + B\,F\,)\,x + B\,E\,r \qquad [\,2.7\,]$$

$$y = C\,x$$

Luego, si se obtiene la correspondiente matriz función de transferencia de lazo cerrado se llega al siguiente resultado:

$$H(s) = C\ (\ s\ I - A - B\ F\ )^{-1}\ \ B\ E \qquad [\ 2.8\ ]$$

Entonces, para tener un sistema desacoplado, se requiere que la matriz función de transferencia de lazo cerrado H(s) sea diagonal y no singular (invertible); es por esto que la presente técnica de desacoplamiento tiene por objeto encontrar un par de matrices constantes F y E que permitan que se cumpla lo requerido para H(s).

En el desarrollo teórico de desacoplamiento por realimentación de estado , se recurre a la siguiente consideración: sean d_1, d2, ... , dm los enteros dados por:

$$d_i = \min \{\ k:\ Ci\ A^k\ B \neq 0\ \} \text{ para } k = 0,1,\ldots\ldots,(n-1)$$

o

$$di = (n-1) \text{ si } Ci\ A^k\ B = 0 \quad \text{para todo } k \in [0,(n-1)]$$

Donde: Ci = i-esima fila de la matriz C

Por lo tanto, un sistema multivariable puede ser desacoplado mediante realimentación de estado si y solamante si la matriz B^* de orden m x m es no singular ; esta matriz está definida de la siguiente manera:

$$B^* = \begin{bmatrix} C_1 \; A^{d_1} \; B \\ C_2 \; A^{d_2} \; B \\ \vdots \\ C_m \; A^{d_m} \; B \end{bmatrix} \qquad [\; 2.9 \;]$$

Haciendo referencia a la matriz B* debe indicarse lo siguiente: siendo G(s) la matriz función de transferencia del sistema original; si det G(s) = 0 , implica que det B* = 0 , por lo que el sistema con det G(s) = 0 es funcionalmente incontrolable y ninguna ley de control puede efectivamente desacoplar el sistema. Se dice que estos sistemas tienen un fuerte acoplamiento inherente [MUNRO,R., 1976, P. 3].

Si det G(s) $\neq$ 0, y det B* $\neq$ 0, el sistema tiene un acoplamiento no inherente: pero si det G(s) $\neq$ 0 y det B* = 0 . se dice que el sistema tiene un débil acoplamiento inherente. Un sistema con débil acoplamiento inherente no puede ser desacoplado mediente realimentación de estado, pero es posible determinar un precompensador dinámico y obtener un nuevo sistema con acoplamiento no inherente.[MUNRO.R.. 1976. P. 3]

Por otro lado se llega a determinar, que si el par F y E desacopla al sistema. entonces el rango de $Q^i(F)$ es uno para cualquier i; inversamente, si el rango de $Q^i(F)$ es uno para todo i y si B* es no singular, entonces el par F y E desacopla el sistema.[GRANIZO,E. .- 1988.- pp. 66-71]

A continuación se expone la matriz $Q^{1}(F)$:

$$Q^{1}(F) = \begin{bmatrix} C_i \ (A + BF)^{n-1} \ B \\ C_i \ (A + BF)^{n-2} \ B \\ \vdots \\ C_i \ (A \underset{0}{+} BF)^{d_1} \ B \end{bmatrix}$$

En conclusión, las condiciones más importantes para tener el conjunto de todos los pares de matrices F y E que desacoplan el sistema son: rango $[Q^{1}(F)] = 1$ para todas las matrices F ; y B^* es no singular para E.[GRANIZO,E. .- 1988.- P. 123].

En base a las consideraciones anteriores se tiene como resultado las matrices F y E que desacoplan al sistema, de la siguiente manera:

$$E = (B^*)^{-1} \qquad [\ 2.10\ ]$$

$$F = -(B^*)^{-1} A^*$$

Donde:

$$A^* = \begin{bmatrix} C_1 \ A^{d_1+1} \\ C_2 \ A^{d_2+1} \\ \vdots \\ C_m \ A^{d_m+1} \end{bmatrix} \qquad [\ 2.11\ ]$$

Ahora bién, el par indicado siempre desacoplará el sistema si det $B^* \neq 0$, pero puede producir inestabilidad o mala configuración de polos al presentarse modos inobservables . En tal caso, la matriz de realimentación puede ser definida por F' de tal forma que :

$$F' = F + F_o$$

Produciendo un set de polos deseados de lazo cerrado; con lo que el par (F', E) pertenece al conjunto de todos los pares (F , E) que desacoplan el sistema.

La nueva matriz F_o puede ser encontrada de la siguiente manera:

$$F_o = (B^*)^{-1} \sum_{K=0}^{\delta} M_K \; C \; A^k$$

Donde:

δ = max di y

Mk = diagonal $[mk^1 , mk^2 , \ldots\ldots , mk^m]$

Entonces, de esta manera se asignan los polos de lazo cerrado del nuevo sistema, mientras simultaneamente se desacopla el sistema. De esta forma $m + \sum_{i=1}^{m} di$ polos de lazo cerrado pueden ser modificados si se reasignan los valores de las matrices Mk.[MUNRO,N., 1976, p. 3]

Es así como se obtiene ya un proceso desacoplado, en que la variación de una entrada influye únicamente en una salida sin que las otras sean afectadas; pero, adicionalmente es conveniente realizar un control que nos permita llevar la variable de salida a valores y especificaciones deseadas, es por esto que en forma adicional en el presente trabajo de tesis se recurre a la utilización de compensadores tipo P.I.D. en forma multilazo, y aqui si se tiene un verdadero control multilazo en el que se encuentran eliminadas las interacciones entre multiples entradas y múltiples salidas.

El esquema que se presenta a continuación nos indica la forma de control utilizada:

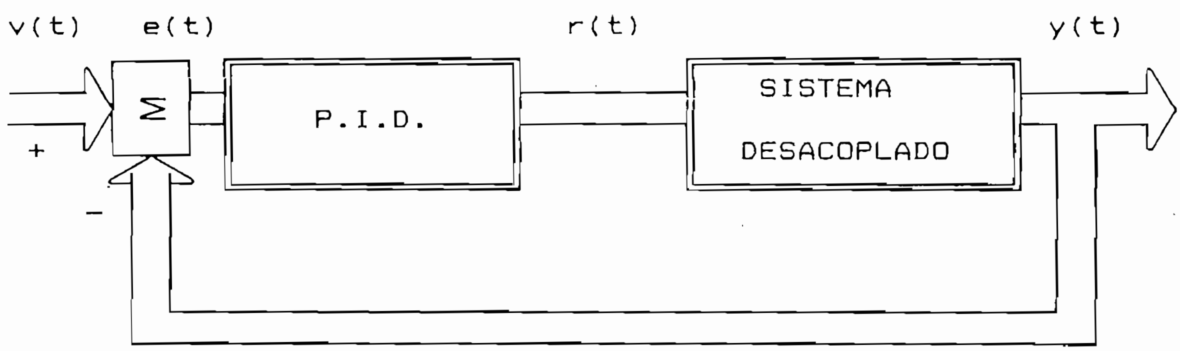


Fig. 2.6 Esquema de control del sistema desacoplado.

En términos generales, la técnica de control con desacoplamiento es de relativa importancia, puesto que permite diseñar compensadores en forma univariable de acuerdo a las diversos métodos de diseño que se tiene en el control clásico.

Adicionalmente cabe recalcar que la teoria de desacoplamiento se ha venido desarrollando para el caso continuo, sin embargo para la realización de la simulación y del control en tiempo real se ha procedido a la discretización de todo el sistema, y, las matrices F y E de realimentación y de precompensación respectivamente tal como se calcularon en el caso continuo, fueron utilizadas en forma discreta, obteniéndose los resultados esperados.

Para poder visualizar de una manera más objetiva la teoría desarrollada de desacoplamiento, se recurre al siguiente ejemplo:[GRANIZO E., 1988, p. 79-88]

Debe indicarse que las funciones de transferencia que se presentan en el siguiente ejemplo han sido obtenidas mediante la ayuda del programa de control CC.

Sea el sistema:

$$A = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

La matriz función de transferencia del sistema original no desacoplado se obtiene bajo la siguiente expresión:

$$H(s) = C\,(\,s\,I - A\,)^{-1}B$$

La matriz función de transferencia indicada se presenta a continuación:

$$H(s) = \frac{1}{(s+3)(s^2-s+1)} \begin{bmatrix} s^2 - s + 1 & 0 \\ 2\,(s-2) & (s-3) \end{bmatrix}$$

De acuerdo a la expresión matricial 2.9 y a la definición de d_i, se tiene :

$$B^* = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \quad , \text{ siendo } \quad d_1 = 0 \quad y \quad d_2 = 1$$

y se cumple además que el determinante de $B^* \neq 0$.

De esta forma, según la expresión matricial 2.11, se tiene el siguiente resultado:

$$A^* = \begin{bmatrix} 0 & 0 & 3 \\ -1 & 1 & 3 \end{bmatrix}$$

Una vez obtenidas las matrices A^* y B^* y aplicando las ecuaciones 2.10, se obtienen las siguientes matrices de precompensación y de realimentación respectivamente:

$$E = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 0 & -3 \\ 1 & -1 & 3 \end{bmatrix}$$

Luego, se procede a obtener la nueva matriz función de transferencia con las matrices de desacoplamiento indicadas, en base a la expresión:

$$H1(s) = C\ (\ s\ I - A - B\ F\ )^{-1}\ B\ E$$

se tiene:

$$G1(s) = \begin{bmatrix} \dfrac{1}{s} & 0 \\ 0 & \dfrac{1}{s^2} \end{bmatrix}$$

Luego, para poder localizar los polos del sistema mientras simultaneamente se desacopla el sistema, se procede a obtener F_o , y conociendo que $\delta = \max d_i = 1$, se tiene:

$$F_o = B^{*-1}\ [\ M_o\ C\ A^o + M_1\ C\ A^1\ ]$$

Para el efecto, se consideran la siguientes matrices diagonales de asignación de polos del sistema desacoplado:

$$M_o = \begin{bmatrix} m_{o1} & 0 \\ 0 & m_{o2} \end{bmatrix} \qquad M_1 = \begin{bmatrix} m_{11} & 0 \\ 0 & m_{12} \end{bmatrix}$$

Es así como se obtiene la siguiente matriz F_o:

$$F_o = \begin{bmatrix} 0 & 0 & m_{o1} + 3m_{11} \\ m_{o2} & m_{12} & m_{12} - 2m_{o1} - 6m_{11} \end{bmatrix}$$

Con lo que se logra finalmente obtener $F' = F + F_o$, que conjuntamente con E conforman el par de matrices de realimentación de estado y de precompensación que desacoplan el sistema.

$$F' = \begin{bmatrix} 0 & 0 & m_{o1} + 3m_{11} - 3 \\ m_{o2} + 1 & m_{12} - 1 & m_{12} - 2m_{o1} - 6m_{11} + 3 \end{bmatrix}$$

La matriz función de transferencia del sistema asi desacoplado, se obtiene de acuerdo a la siguiente expresión:

$$H2(s) = C\ (\ s\ I - A - B\ F'\ )^{-1}\ B\ E$$

que determina la matriz función de transferencia que se presenta a continuación:

$$H2(s) = \begin{bmatrix} \dfrac{1}{(s - m_{01} - 3m_{11})} & 0 \\ 0 & \dfrac{1}{(s^2 - m_{12}s - m_{02})} \end{bmatrix}$$

Se puede observar claramente que el proceso original ha sido desacoplado, pues los elementos fuera de la diagonal de la matriz función de transferencia final son iguales a cero.

Ahora bien, se desea asignar los polos del sistema desacoplado mediante la elección de los valores de las matrices M_k de tal modo que los subsistemas desacoplados posean un comportamiento de la respuesta de acuerdo a los requerimientos deseados.

Así, se requiere que el subsistema 1 posea un polo en −4, mientras que el subsistema 2 posea un polo en −3 y otro en −5 ; en tal caso deben ser asignados los siguientes valores: $m_{01} = -1$; $m_{02} = -15$; $m_{11} = -1$ y $m_{12} = -8$ con lo que se tiene las siguientes matrices M_k:

$$M_0 = \begin{bmatrix} -1 & 0 \\ 0 & -15 \end{bmatrix} \qquad M_1 = \begin{bmatrix} -1 & 0 \\ 0 & -8 \end{bmatrix}$$

De este modo se tiene la siguiente matriz función de transferencia del sistema desacoplado:

$$H2\ (s) = \begin{bmatrix} \dfrac{1}{(s+4)} & 0 \\ 0 & \dfrac{1}{(s+3)(s+5)} \end{bmatrix}$$

2.3.- REGULADOR CUADRATICO LINEAL

La teoría de control moderno, con la ayuda del computador y métodos digitales, permite la relización de un sistema de funcionamiento óptimo de acuerdo a las especificaciones del criterio de funcionamiento utilizado, de tal forma que este se máximice o minimice.[D'AZZO, J., 1981, p. 525]

Este método, en contraste con las técnicas de diseño convencional se realiza bajo un extenso análisis matemático; por otro lado, mientras que el control óptimo resultante satisface el índice de funcionamiento, podría no satisfacer valores de M_p, t_s, t_p, etc. por lo cual debe existir un compromiso entre los dos tópicos.[D'AZZO, J., 1981, p. 525]

Un índice de funcionamiento o función de costo es un valor escalar que pondera los valores importantes del proceso; dicho índice, debe ser maximizado o minimizado de acuerdo a los fines que se persiguen, para de esta manera optimizar el proceso. El índice de funcionamiento tiene tres propiedades básicas: confiable, de fácil aplicación y selectivo. Debe ser confiable, porque dado una clase de sistema puede ser aplicado con seguridad; debe también ser de fácil aplicación y selectivo ya que el resultado tiene que ser un claro sistema óptimo.[D'AZZO, J., 1981, p. 525]

Existen diversos índices de funcionamiento, los cuales pueden ser aplicados de acuerdo a los requerímientos y fines que se persiguen; estos indices de funcionamiento (J) se establecen en base al error y al tiempo, entre ellos se tienen:

$$J_1 = \int_0^{\infty} e(t)*dt \qquad J_5 = \int_0^{\infty} t*|e(t)|*dt$$

$$J_2 = \int_0^{\infty} t*e(t)*dt \qquad J_6 = \int_0^{\infty} t*e^2(t)*dt$$

$$J_3 = \int_0^{\infty} e^2(t)*dt \qquad J_7 = \int_0^{\infty} t^2*e^2(t)*dt$$

$$J_4 = \int_0^{\infty} |e(t)|*dt \qquad J_8 = \int_0^{\infty} t^2*|e(t)|*dt$$

Los índices de funcionamiento indicados, determinan un funcionamiento óptimo basados en la optimización de parámetros, esto es que los valores de la planta son fijados de tal forma que satisfaga el índice de funcionamiento. Las variables del sistema pueden tener limitaciones en cuanto a sus valores máximos o mínimos como en el caso de la ley de control u.

Ante la dificultad de limitar estas variables en los parámetros de optimización, es necesario desarrollar otros métodos de optimización, de lo cual se encarga el control moderno.

En la teoría del control moderno el índice de funcionamiento se define en terminos de los véctores de estado y de control, x(t) y u(t) respectivamente, y las limitaciones de las variables se encuentran dadas en los pesos que se asignan a cada una de ellas.

En la presente tesis, en lo referente a control óptimo, se desarrollará un tipo de realimentación óptima de estado, regulador cuadrático lineal, en la que se tiene por objeto bajo ciertas condiciones, determinar la ley de control óptima $u^{o}(x,t)$, ver figura 2.7 que constituye la entrada del sistema y la cual puede transferir el sistema desde un estado inicial a un estado final, mientras existe una minimización del índice de funcionamiento.

El regulador cuadrático lineal, como su nombre lo indica es un problema de regulación en que se asume cero a la entrada del sistema y se obtiene la respuesta del mismo a partir de una condición inicial $[x(t_i)]$. En este regulador, mediante un criterio óptimo se realiza realimentación de estado bajo el esquema de la figura 2.7, realimentación óptima de estado.

La representación del sistema original en variables de estado es la siguiente:

$$\dot{x}(t) = A\,x(t) + B\,u(t) \qquad [\ 2.12\ ]$$

Donde:

x = vector continuo de variables de estado

u = vector continuo de control

A = matriz que define el sistema

B = matriz que define el sistema

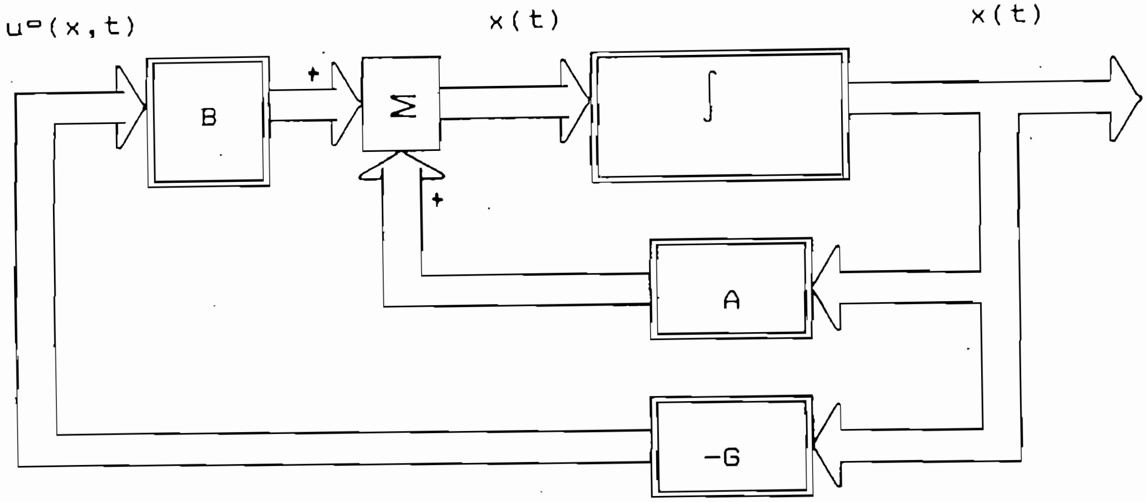


Fig. 2.7 Realimentación óptima de estado

Luego, se desea desarrollar un algoritmo que permita encontrar la ley de control óptima que será aplicada al sistema; el algoritmo indicado, se lo realiza desde un punto de vista discreto, por lo que se procede a la discretización del sistema representado en la ecuación 2.12, asi se tiene:

$$x[(k+1)T] = \Phi(T)\ x[kT] + \Theta[T]\ u[kT] \qquad [\ 2.13\ ]$$

Donde:

$$\Phi(T) = e^{AT}$$

$$\Theta(T) = \int_{0}^{T} e^{A(T-\tau)}\ B\ d\tau$$

Para sistemas fisicos, la entrada de control es siempre limitada. Por ejemplo, la amplitud de cada componente del vector de control puede ser limitada tal como:

$$|u_i(kT)| \leq U_i$$

donde cada U_i es una constante, y el subíndice i representa la i-ésima componente del vector.[PHILLIPS,C., 1984, p. 307]

Para el caso en que se desea limitar la energía del control se tiene:

$$u_i^2(kT) \leq M_i$$

donde cada M_i es una constante. La disponibilidad de una energía finita en el control puede ser representado mediante la siguiente función de costo:

$$u'(kT)\ R\ u(kT)$$

donde R es una matriz de peso; esta función es llamada una forma cuadrática. En muchos casos el control debe ser limitado y si satisface dicha condición es llamado control admiscible.[PHILLIPS,C., 1984, p. 307]

De esta forma se trata de encontrar la ley de control óptima $u^o(kT)$ para k = 0, 1, 2, ... , N-1 mediante el siguiente índice de funcionamiento cuadrático:

$$J_N = \tfrac{1}{2}[x'(NT)Sx(NT)] + \tfrac{1}{2}\sum_{K=0}^{N-1}[x'(kT)Qx(kT) + 2x'(kT)Mu(kT) + u'(kT)Ru(kT)] \qquad [\ 2.14\ ]$$

Donde: S = matriz simétrica semidefinida positiva

Q = matriz simétrica semidefinida positiva

R = matriz simétrica definida positiva

M = matriz de ponderación

Nota: El apóstrofe (′) indica la transpuesta de la matriz indicada.

Ahora, el problema que nos ocupa constituye en que dado el sistema discreto de la ecuación 2.13 , se debe encontrar la ley de control óptima que minimice el índice de funcionamiento cuadrático que se presenta en la ecuación 2.14; así, y considerando en forma implícita el periodo de muestreo T se tiene:

$$J_N = G[x(N)] + \sum_{K=0}^{N-1} F_k[x(k),u(k)] = \text{mínimo} \qquad [\ 2.15\ ]$$

Donde:

$$G[x(N)] = \tfrac{1}{2}\ x'(N)\ S\ x(N) \qquad [\ 2.16\ ]$$

$$F_k[x(k),u(k)] = \tfrac{1}{2}x'(k)Qx(k) + x'(k)Mu(k) + \tfrac{1}{2}u'(k)Ru(k) \qquad [\ 2.17\ ]$$

Sujeto al sistema:

$$x(k+1) = \Phi\ x(k) + \theta\ u(k) \qquad [\ 2.18\ ]$$

con $x(0)$ dado.

Nota: La resolución detallada de este algoritmo de control se presenta a continuación y ha sido tomado del libro "Digital Control Systems" de Benjamin Kuo,1980.

Siendo $J_{N-1}[x(i)]$ el índice de funcionamiento para el intervalo $[i,N]$, esto es, sobre los N-i intervalos de muestreo, se tiene:

$$J_{N-1}[x(i)] = G[x(N)] + \sum_{K=i}^{N-1} F_k[x(k),u(k)]$$

para i= 0, 1, 2, ..., N

Por lo que el mínimo valor de $J_{N-1}[x(i)]$ se representa mediante:

$$f_{N-1}[x(i)] = \min_{u(i)} J_{N-1}[x(i)]$$

Para i=N esta última ecuación representa el índice de funcionamiento terminal o función de costo terminal, así, se tiene:

$$f_0[x(N)] = G[x(N)] = \tfrac{1}{2}\, x'(N)\ S\ x(N) \qquad [\ 2.19\ ]$$

Luego, para i=N-1, se tiene el último intervalo de muestreo, aquí, el índice de funcionamiento es:

$$f_1[x(N-1)] = \min_{u(N-1)} J_1[x(N-1)]$$

$$= \min_{u(N-1)}\{G[x(N)] + F_{N-1}[x(N-1),u(N-1)]\} \qquad [\ 2.20\ ]$$

Sustituyendo, la ecuación 2.18 en la ecuación 2.16, se tiene:

$$G[x(N)] = \tfrac{1}{2}[\Phi x(N-1) + \Theta u(N-1)]'\ S\ [\Phi x(N-1) + \Theta u(N-1)]$$

Entonces:

$$G[x(N)] = \tfrac{1}{2}[x'(N-1)\Phi' + u'(N-1)\Theta']\ S\ [\Phi x(N-1) + \Theta u(N-1)] \qquad [\ 2.21\ ]$$

Por otro lado, si en 2.17 se considerá $k = N-1$, se tiene:

$$F_{N-1}[x(N-1),u(N-1)] = ½[x'(N-1)Qx(N-1) + 2x'(N-1)Mu(N-1) + u'(N-1)Ru(N-1)]$$

[2.22]

Ahora bién, reemplazando 2.21 y 2.22 en 2.20 se tiene:

$$f_1[x(N-1)] = \min_{u(N-1)} [½x'(N-1)(Q+\Phi'S\Phi)x(N-1) + x'(N-1)(M + ½\Phi'S\boldsymbol{\theta})u(N-1) + ½u'(N-1)\boldsymbol{\theta}'S\Phi x(N-1) + ½u'(N-1)(R+\boldsymbol{\theta}'S\boldsymbol{\theta})u(N-1)]$$

[2.23]

Entonces:

$$f_1[x(N-1)] = \min_{u(N-1)} J_1[x(N-1)]$$

Siendo J_1 un valor escalar, para obtener el mínimo valor de J_1, siendo u el parámetro de control se debe aplicar el concepto de derivada, así:

$$\partial J_1[x(N-1)] / \partial u(N-1) = 0$$

Con lo que se obtiene como resultado:

$$[(M + ½\Phi'S\boldsymbol{\theta})' + ½\boldsymbol{\theta}'S\Phi]\ x^o(N-1) + (R + \boldsymbol{\theta}'S\boldsymbol{\theta})\ u^o(N-1) = 0$$

Con lo que el control óptimo resultante es:

$$u^o(N-1) = -(R + \boldsymbol{\theta}'S\boldsymbol{\theta})^{-1}\ (M' + \boldsymbol{\theta}'S\Phi)\ x^o(N-1) \qquad [\ 2.24\ ]$$

Luego, sustituyendo 2.24 en 2.23, reemplazando el valor de $u(N-1)$ por su óptimo $u^o(N-1)$, se tiene:

$$f_1[x(N-1)] = \tfrac{1}{2}x'(N-1)\ [Q + \Phi'S\Phi - (M' + \Theta'S\Phi)'(R + \Theta'S\Theta)^{-1}(M' + \Theta'S\Phi)]\ x(N-1) \qquad [\ 2.25\ ]$$

Luego, realizando las siguientes definiciones:

$$K(N) = S \qquad [\ 2.26\ ]$$

$$K(N-1) = Q + \Phi'S\Phi - (M' + \Theta'S\Phi)'(R + \Theta'S\Theta)^{-1}(M' + \Theta'S\Phi)$$

se tiene la siguiente expresión:

$$K(k) = Q + \Phi'K(k+1)\Phi - (M' + \Theta'K(k+1)\Phi)'(R + \Theta'K(k+1)\Theta)^{-1}(M' + \Theta'K(k+1)\Phi) \qquad [\ 2.27\ ]$$

Esta última ecuación, 2.27, tiene la forma de las llamadas ecuaciones de Riccati.

Entonces, utilizando las definiciones 2.26, en 2.19 y 2.25, se tiene las siguientes expresiones:

$$f_0[x(N)] = \tfrac{1}{2}\ x'(N)\ K(N)\ x(N) \qquad [\ 2.28\ ]$$

$$f_1[x(N-1)] = \tfrac{1}{2}\ x'(N-1)\ K(N-1)\ x(N-1) \qquad [\ 2.29\ ]$$

Continuando con el proceso, se realiza el cálculo para $i=N-2$, esto es, el problema de optimización para los dos últimos intervalos; así se tiene:

$$f_2[x(N-2)] = \min_{u(N-1),u(N-2)} J_2[x(N-2)]$$

$$= \min_{u(N-1),u(N-2)} \{F_{N-2}[x(N-2),u(N-2)] + F_{N-1}[x(N-1),u(N-1)] + G[x(N)]\}$$

$$= \min_{u(N-2)} \{ F_{N-2}[x(N-2),u(N-2)] + f_1 [x(N-1)]\} \qquad [\ 2.30\ ]$$

Donde $f_1[x(N-1)]$ es el índice de funcionamiento en el último intervalo.

Aplicando los pasos de demostración explicados en el desarrollo precedente, se llega a la siguiente ecuación:

$$F_{N-2}[x(N-2),u(N-2)] = \tfrac{1}{2}x'(N-2)Qx(N-2) + x'(N-2)Mu(N-2) + \tfrac{1}{2}u'(N-2)Ru(N-2) \qquad [\ 2.31\ ]$$

Por otro lado, reemplazando la ecuación 2.18 en la ecuación 2.29, se tiene:

$$f_1[x(N-1)] = \tfrac{1}{2}[\Phi x(N-2) + \Theta u(N-2)]'K(N-1)[\Phi x(N-2) + \Theta u(N-2)] \qquad [\ 2.32\ ]$$

Luego reemplazando 2.31 y 2.32 en 2.30 y además derivando:

$$\partial J_2[x(N-2)]/\partial u(N-2) = 0$$

se puede determinar que el control óptimo es:

$$u^o(N-2) = -[R + \Theta'K(N-1)\Theta]^{-1} [M' + \Theta'K(N-1)\Phi] x^o(N-2)$$

y por lo tanto:

$$f_2[x(N-2)] = \tfrac{1}{2}x'(N-2)\{Q + \Phi'K(N-1)\Phi - [M' + \Theta'K(N-1)\Phi]'$$
$$[R + \Theta'K(N-1)\Theta]^{-1}[M' + \Theta'K(N-1)\Phi]\}x(N-2)$$

Por lo que conservando las definiciones previas de las ecuaciones 2.26, se tiene:

$$K(N-2) = Q + \Phi'K(N-1)\Phi - [M' + \Theta'K(N-1)\Phi]'[R + \Theta'K(N-1)\Theta]^{-1}$$
$$[M' + \Theta'K(N-1)\Phi]$$

Por lo que 2.30 puede ser descrito de la siguiente manera:

$$f_2[x(N-2)] = \tfrac{1}{2}x'(N-2)\, K(N-2)\, x(N-2)$$

Continuando con el proceso de inducción se puede encontrar una fórmula general que se presenta a continuación:

$$f_{N-i}[x(i)] = \tfrac{1}{2}\, x'(i)\, K(i)\, x(i) \qquad [\ 2.33\ ]$$

Donde:

$$K(i) = Q + \Phi'K(i+1)\Phi - [M' + \Theta'K(i+1)\Phi]'[R + \Theta'K(i+1)\Theta]^{-1}$$
$$[M' + \Theta'K(i+1)\Phi]$$

$$[\ 2.34\ ]$$

Con lo que se obtiene la siguiente ley óptima de control:

$$u^o(i) = -[R + \Theta' K(i+1)\Theta]^{-1} [M' + \Theta' K(i+1)\Phi]\, x^o(i) \qquad [\ 2.35\ ]$$

Lo cual puede ser descrito de la siguiente manera:

$$u^o(i) = - G(i)\, x^o(i) \qquad [\ 2.36\ ]$$

Donde:

$$G(i) = [R + \Theta' K(i+1)\Theta]^{-1} [M' + \Theta' K(i+1)\Phi] \qquad [\ 2.37\ ]$$

De este modo se ha llegado a la llamada ecuación de Riccati, 2.34 utilizando el principio de optimalidad; el método de resolución es conocido como programación dinámica.

Es interesante notar que el método de programación dinámica no requiere que R sea definida positiva, ya que R^{-1} no es encontrada; sin embargo, la matriz $[R + \Theta' K(i+1)\Theta]$ debe tener su inversa.

Para el caso de tiempo infinito, o infinito número de intervalos de muestreo, $N \dashrightarrow \infty$, se tiene que $K(k) \dashrightarrow K$, por lo que sustituyendo $K(k+1)$ y $K(k)$ por K en las ecuaciones 2.34 y 2.35 se tiene:

$$K = Q + \Phi' K \Phi - [M' + \Theta' K \Phi]' [R + \Theta' K \Theta]^{-1} [M' + \Theta' K\, \Phi] \qquad [\ 2.38\ ]$$

$$u^{o}(i) = - [R + \boldsymbol{\theta}'K\boldsymbol{\theta}]^{-1} [M' + \boldsymbol{\theta}'K\Phi] \, x^{o}(i) \qquad [\ 2.39\ ]$$

Que se puede expresar de la siguiente manera:

$$u^{o}(i) = - G \, u^{o}(i) \qquad [\ 2.40\ ]$$

Donde:

$$G = [R + \boldsymbol{\theta}'K\boldsymbol{\theta}]^{-1} [M' + \boldsymbol{\theta}'K\Phi] \qquad [\ 2.41\ ]$$

Solución de la ecuación discreta de Riccati:

En general la ecuación algebraica de Riccati (2.38) presenta mayor dificultad en la resolución que la ecuación de diferencias de Riccati, del tipo de las ecuaciones 2.27 o 2.34.

La ecuación de diferencias de Riccati es generalmente resuelta por medio de los siguientes métodos:

- Métodos numéricos computacionales.
- Métodos recursivos.
- Métodos de valores propios - vectores propios.

La programación dinámica presentada en forma precedente constituye un método recursivo y se conoce generalmente como solución recursiva de problemas de control óptimo.

Dado que:

$$G(i) = [R + \theta' K(i+1)\theta]^{-1} [M' + \theta' K(i+1)\Phi] \qquad [\ 2.42\ ]$$

es la ganancia de realimentación óptima de estado [$u^{o}(i) = -G(i)\ x^{o}(i)$] ; la ecuación de Riccati se puede expresar de la siguiente manera:

$$K(i) = Q + \Phi' K(i+1)\Phi - [M' + \theta' K(i+1)\Phi]'\ G(i) \qquad [\ 2.43\ ]$$

Entonces, empezando con la condición inicial $K(N) = S$, las ecuaciones 2.42 y 2.43 son resueltas por recursión en la dirección de adelante hacia atrás.

En el programa realizado se ha utilizado la matriz S igual a cero como condición inicial,de tal forma que la ganancia de realimentación tiene un valor inicial nulo, para finalmente adquirir un valor estable luego de algún número de iteraciones del proceso recursivo.

Adicionalmente es importante mencionar que esta técnica de control es una técnica de regulación por lo que con una condición inicial de los estados distinta de cero, se llegará finalmente a un valor igual a cero ; sin embargo, en el desarrollo del programa respectivo se ha realizado una adecuación que permita llevar el valor de los estados a ciertos niveles, mediante la inserción de un valor de offset que permita que los estados adquieran cierto valor final diferente de cero.

Para visualizar de una manera más objetiva los fines que se persiguen se presenta el siguiente ejemplo:[KUO,B., 1980, pp. 619-621]

Dado un proceso digital descrito por la siguiente ecuación de estado:

$$x(k+1) = \Phi x(k) + \theta u(k)$$

Donde:

$$\Phi = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} \qquad \theta = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

y dado el valor inicial del vector de estado $x(0)=[1 \quad 1]'$, se desea encontrar el control óptimo $u(k), k= 0, 1, \ldots, 7$ que satisfaga el siguiente índice de funcionamiento sea minimizado:

$$J_8 = \sum_{K=0}^{7} [x_1^2(k) + u^2(k)]$$

En este problema se identifica que $M = 0$, $R = 2$,

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \quad y \quad S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Sustituyendo estos parámetros en la ecuación 2.43, se tiene:

$$K(i) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} K(i+1) \left[\begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} G(i) \right]$$

La ley de control óptima es: $u^o(i) = -G(i)\, x^o(i)$

Donde, de la ecuación 2.42, se tiene:

$$G(i) = \left[2 + [0 \quad 1]\, K(i+1) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right]^{-1} [0 \quad 1]\, K(i+1) \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}$$

Si se parte de una condiciuón inicial:

$$K(8) = S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Entonces se puede resolver las ecuaciones anteriores recursivamente, obteniéndose los siguientes resultados:

$$K(7) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \qquad G(7) = [\ 0 \quad 0]$$

$$K(6) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \qquad G(6) = [\ 0 \quad 0]$$

$$K(5) = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \qquad G(5) = [\ -0.5 \quad 0.5]$$

$$K(4) = \begin{bmatrix} 3.2 & -0.8 \\ -0.8 & 3.2 \end{bmatrix} \qquad G(4) = [\ -0.6 \quad 0.4]$$

$$K(3) = \begin{bmatrix} 3.23 & -0.922 \\ -0.922 & 3.69 \end{bmatrix} \qquad G(3) = [\ -0.615 \quad 0.462]$$

$$K(2) = \begin{bmatrix} 3.297 & -0.973 \\ -0.973 & 3.729 \end{bmatrix} \qquad G(2) = [\ -0.651 \quad 0.481]$$

$$K(1) = \begin{bmatrix} 3.301 & -0.962 \\ -0.962 & 3.75 \end{bmatrix} \qquad G(1) = [\,-0.652 \quad 0.485\,]$$

$$K(0) = \begin{bmatrix} 3.305 & -0.97 \\ -0.97 & 3.777 \end{bmatrix} \qquad G(0) = [\,-0.6538 \quad 0.486\,]$$

El control óptimo y la trayectoria óptima de los estados puede ser calculada utilizando la descripción en ecuación de estado y la ecuación 2.36, sustituyendo la ganancia de realimentación óptima de estado.

Para altos valores de N, se puede obtener la ganancia de Riccati aprovechando el estado estable de la solución recursiva, esto es:

$$K = \begin{bmatrix} 3.308 & -0.972 \\ -0.972 & 3.780 \end{bmatrix}$$

Y la ganancia de realimentación óptima será:

$$G = [\,-0.654 \quad 0.486\,]$$

2.4.- ESTRUCTURA DEL SOFTWARE DESARROLLADO

El software realizado se encuentra estructurado en dos módulos perfectamente definidos, el primer módulo se refiere a simulación en base a los algoritmos desarrollados y el segundo módulo constituye la utilización de dichos algoritmos en tiempo real.

El programa realizado en el presente tema de tesis ha sido desarrollado en el lenguaje de programación QUICK BASIC, versión 4.5 ; además en lo que a tiempo real se refiere, se utiliza el software especializado QUICK 500 para manejar en forma adecuada el equipo de adquisición de datos KEITHLEY 500A.

El programa completo consta de dos programas ejecutables: MULTIPR.EXE que posee el software referente a simulación y MULTITR.EXE que contiene a las rutinas que permiten la aplicación del control en tiempo real.

El programa ejecutable MULTIPR.EXE dispone además de un menú principal en el cual el usuario tiene la posibilidad de elegir el realizar simulación o tiempo real; si se opta por la primera opción, se utilizan las rutinas presentes en el mismo programa; por el contrario, si se especifica la segunda opción,se realiza un llamado al programa MULTITR.EXE que permite trabajar en tiempo real. Al terminar la ejecución del segundo programa, se retorna al primero y en el lugar correspondiente al menú principal, lo cual posibilita el repetir cualquier opción o salir del programa.

A continuación se presentan las figuras 2.8 y 2.9 que indican el esquema del menú principal utilizado y la forma de ejecución del programa en tiempo real:

1) SIMULACION

2) TIEMPO REAL

3) ABANDONAR EL PROGRAMA

Fig. 2.8 Menú principal

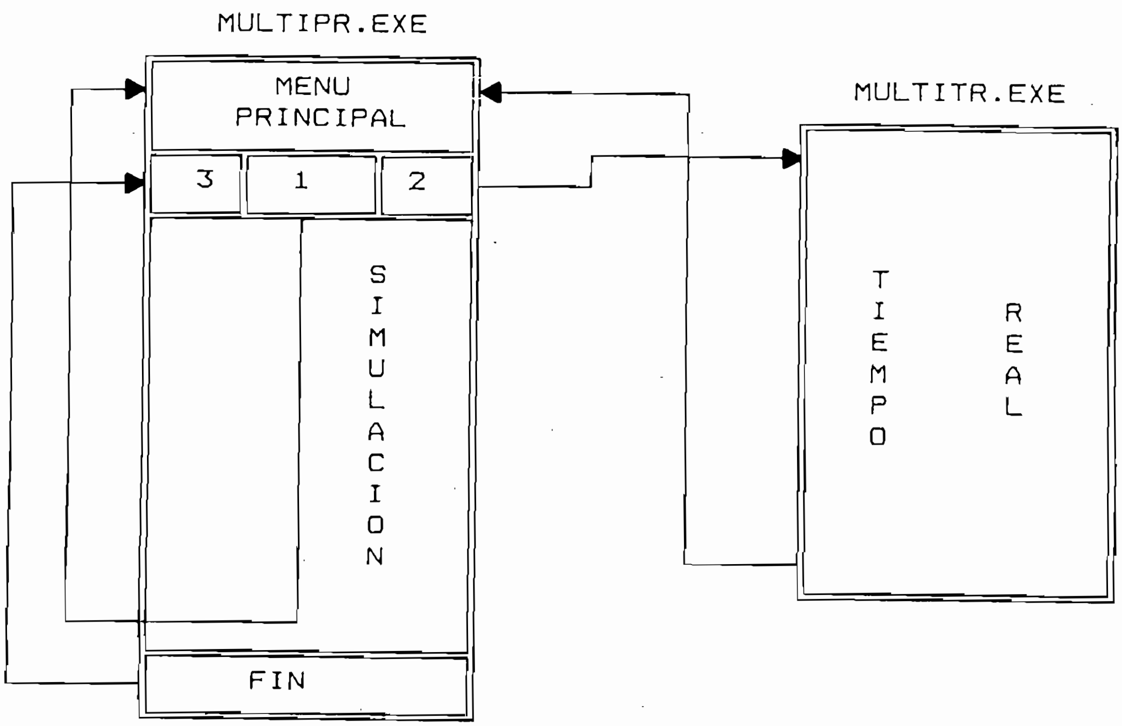


Fig. 2.9 Ejecución del programa en tiempo real

Por otro lado, una vez que se ha elegido trabajar en simulación o en tiempo real, en cada uno de estos casos se tiene otro menú que permite utilizar los siguientes subtópicos revisados en la parte teórica:

- P.I.D. multilazo
- Desacoplamiento
- Regulador cuadrático lineal

La especificación y descripción de cada uno de los algoritmos implementados se la realizará posteriormente, sin embargo cabe resaltar que se han tomado como variables enteras I - N y como variables reales de doble precisión A - H y O - Z para la definición de todo el programa.

Adicionalmente, cabe recalcar que los cálculos de los diversos algoritmos se realizan en forma off-line, y estos cálculos obtenidos fuera de linea se utilizan tanto en simulación como en tiempo real, requeriéndose el ingreso manual de dichos valores para ser ensayados en tiempo real.

Es importante mencionar que para poder ejecutar un programa con las librerias propias del Quick 500, que permitan menejar el equipo de adquisición de datos y control se debe anteponer BRUN al nombre del programa a ejecutarse; así, para utilizar el software desarrollado se tiene que especificar BRUN MULTIPR. Lo indicado, puede realizarse directamente a través de un archivo .BAT y es por esto que se ha creado el archivo **MPR.BAT**; en tal caso, para utilizar el programa realizado en la presente tesis, el usuario puede hacerlo únicamente con MPR.

2.5.- SOFTWARE PARA CONTROL MULTILAZO

En este numeral, se analizan las principales características en cuanto al software desarrollado para el caso de control multilazo y en lo que a simulación se refiere, pues lo referente a tiempo real será analizado posteriormente.

En términos generales se tiene el control multilazo para lazos totalmente independientes o para sistemas multivariables con débil acoplamiento, es por esto que se hace una diferencia entre estas dos posibildades en el programa realizado, lo cual se lo encuentra en un menu autoexplicativo que nos permite bifurcar para realizar el ingreso de datos ya sea en uno u otro sentido.

En el caso de lazos totalmente independientes se realiza el cálculo de las salidas en base a las ecuaciones de diferencias de cada lazo; por otro lado, en el caso de sistemas multivariables el cálculo respectivo se realiza en base a la descripción del sistema mediante variables de estado.

En forma adicional, cabe indicar que en lo correspondiente al ingreso de datos para compensación, se permite ya sea al ingreso de datos de un control tipo P.I.D. o cualquier otro tipo de control digital, en este último caso se debe mencionar que el compensador a utilizarse será ingresado en forma de ecuación de diferencias, para lo cual se dispone de un máximo de cinco instantes de retardo, que constituye una limitación sin mayor inconveniente, pues en forma práctica no se excede del valor indicado.

En esta parte del programa, se visualizan los resultados en forma gráfica, permitiendo además un despliegue de los valores en pantalla y la impresión de resultados totales en papel. Adicionalmente, se dispone de la posibilidad de almacenar valores en un archivo para su posterior graficación desde al programa LOTUS.

En la figura 2.10, al final de este numeral se puede apreciar con cierto detalle el contenido de la parte del programa en mención mediante la ayuda del diagrama de flujo propuesto.

Listado de las principales variables:

m = Número de lazos = número de entradas del sistema

m1 = Número de salidas del sistema

n = Orden del sistema

t = Periodo de muestreo

acon() = Matriz continua que define el sistema a() = Matriz discretizada que define el sistema

bcon() = Matriz continua que define el sistema

b() = Matriz discretizada que define el sistema

c() = Matriz que define el sistema

l = Número de iteracciones

yl1() = Matriz de salidas del sistema en sus diversos muestreos.

el1() = Matriz de errores del sistema en sus diversos muestreos.

ul1() = Matriz de leyes de control del sistema en sus diversos muestreos

rl1() = Vector de referencias de control

npidl = Variable entera que lleva la información de si se tiene control multilazo o multivariable.

gankp = Ganancia proporcional (K_P) del P.I.D.

ganki = Ganancia integral (K_I) del P.I.D.

gankd = Ganancia integral (K_D) del P.I.D.

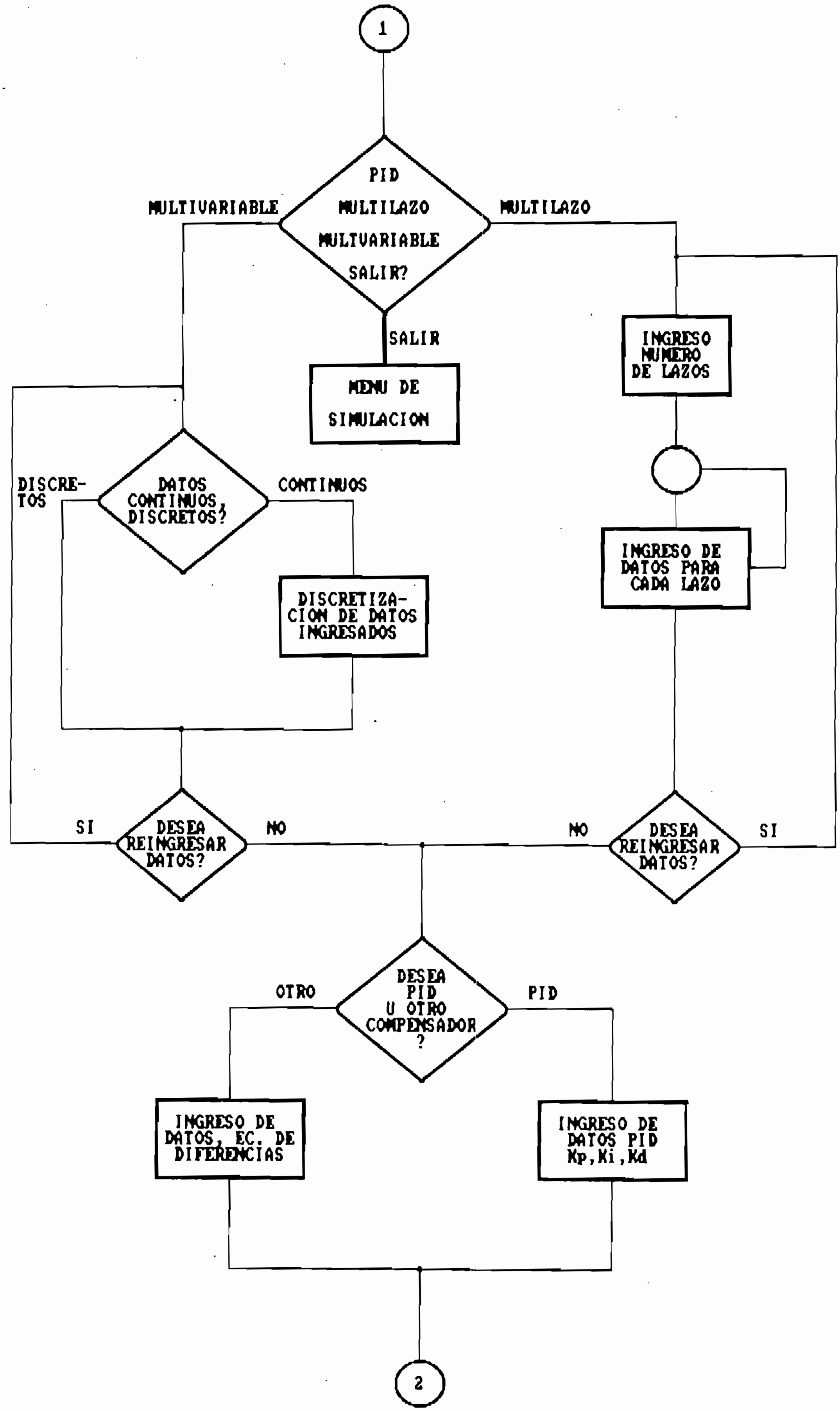


Fig. 2.10 (a) DIAGRAMA DE FLUJO.- CONTROL MULTILAZO

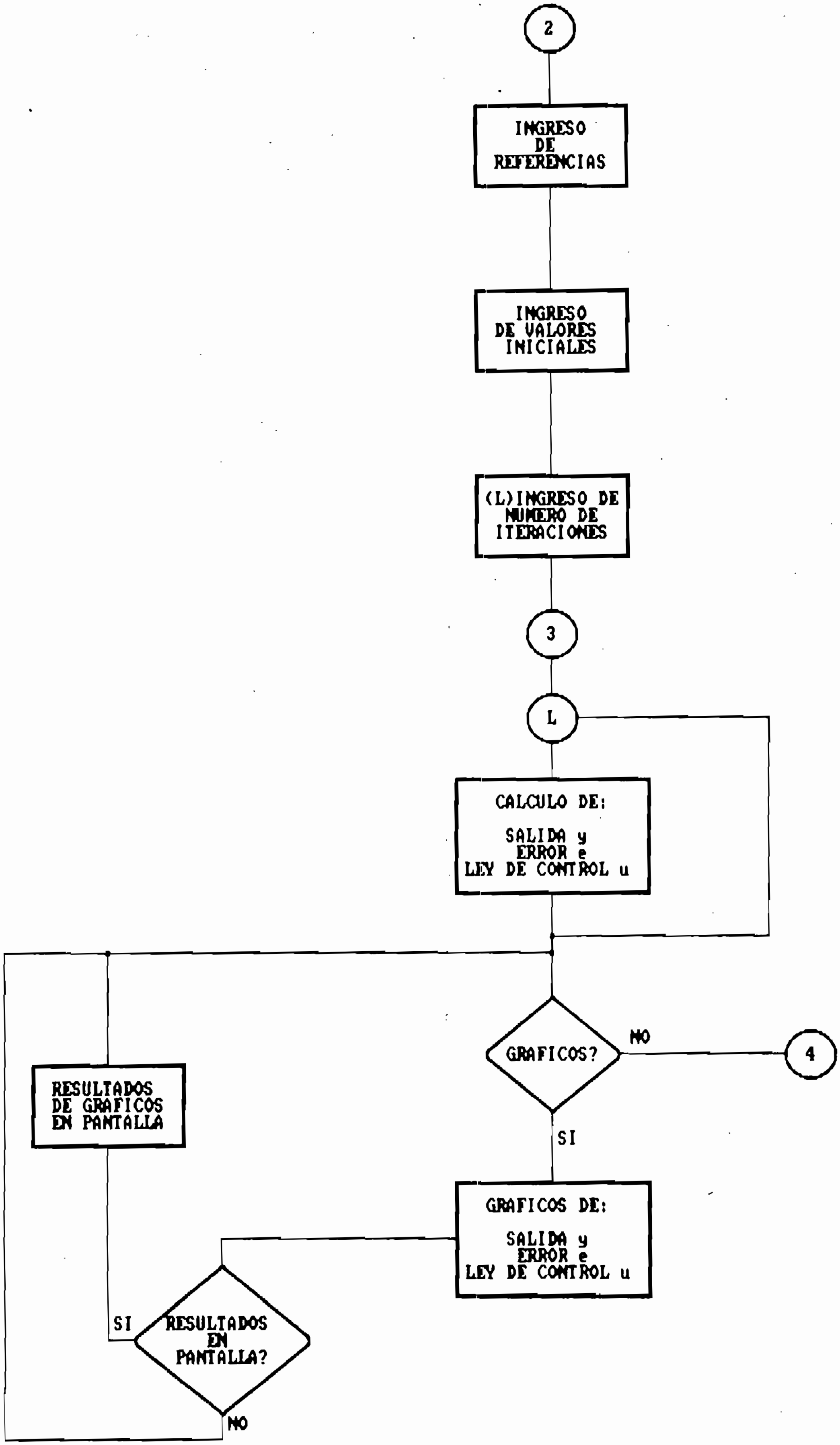


Fig. 2.10 (b) DIAGRAMA DE FLUJO.- CONTROL MULTILAZO

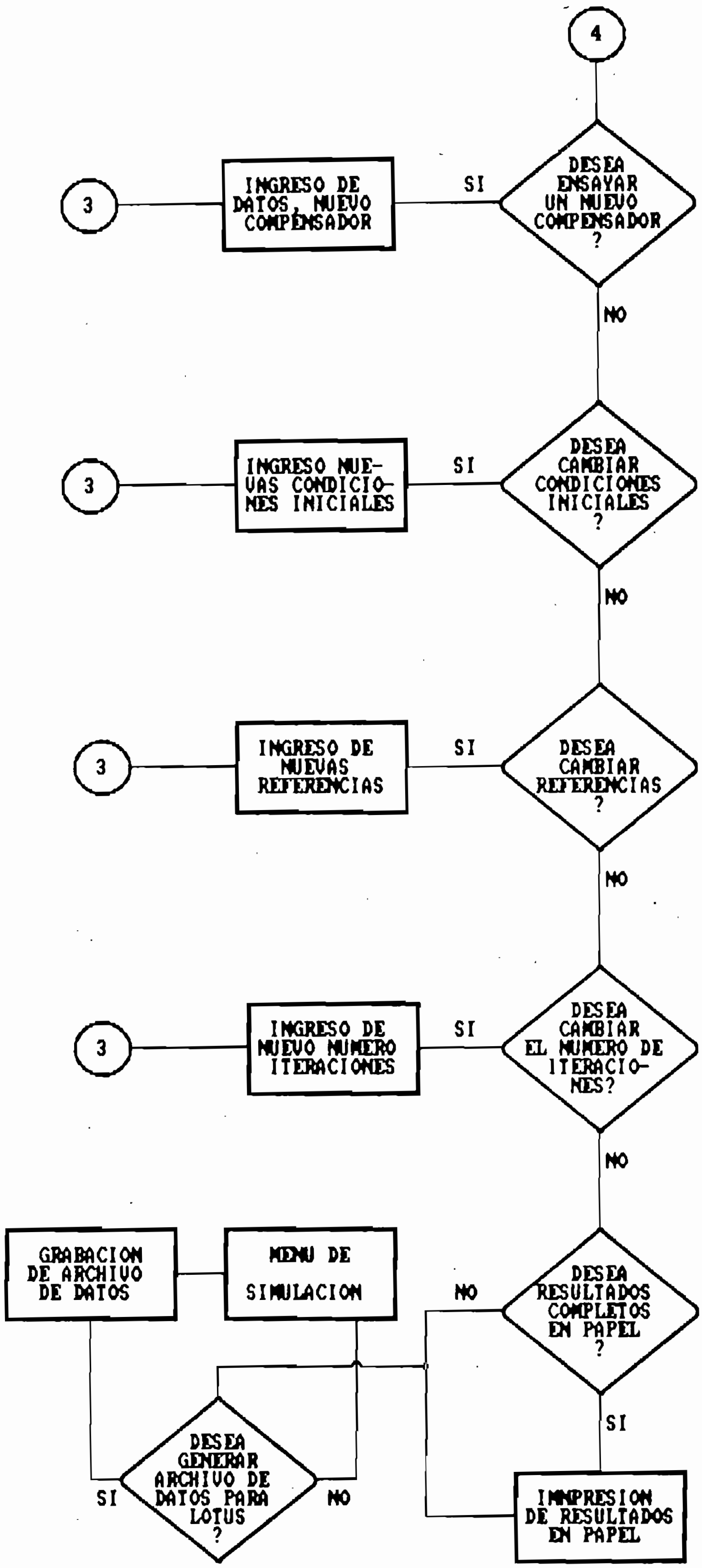


Fig. 2.10 (c) DIAGRAMA DE FLUJO.- CONTROL MULTILAZO

2.6.- SOFTWARE PARA DESACOPLAMIENTO

En cuanto al software desarrollado para desacoplamiento, se lo ha estructurado en dos partes; la primera, en la que se realiza el cálculo necesario que permite obtener el par de matrices que desacoplan el sistema y realizar la simulación gráfica de respuesta del sistema desacoplado al variar las entradas del mismo.

Una segunda parte de este software constituye el control de dicho sistema desacoplado en base a la implemetación de un compensador por cada par entrada-salida, ya que se ha logrado eliminar la interacción; esto es, se realiza el control de subsistemas univariables.

Al igual que en lo correspondiente a P.I.D. multilazo, en el software realizado para desacoplamiento se pone énfasis en la parte gráfica, para poder visualizar gráficamente las diversas respuestas del sistema. Por otro lado, se tiene la posibilidad de variar valores tales como referencias, valores iniciales de los estados, número de iteraciones requerido, compensadores en el caso del sistema desacoplado y controlado, etc.

El software realizado permite además obtener resultados completos en papel y por otro lado se dispone de una rutina que posibilita generar un archivo de datos que tiene por objetivo la posterior presentación en papel de los gráficos obtenidos en simulación, con la ayuda de la hoja electrónica LOTUS.

Listado de las principales variables:

n = Orden del sistema

m = Número de entradas

mn = Número de salidas

acon() = Matriz continua que define el sistema

bcon() = Matriz continua que define el sistema

c() = Matriz continua que define el sistema

a() = Matriz discretizada que define el sistema

b() = Matriz discretizada que define el sistema

t = Periodo de muestreo

l = Número de iteracciones

bbb() = Matriz B* mxm

bbbi() = Matriz inversa de B*

aaa() = Matriz A* mxn

e() = Matriz de precompensación

f() = Matriz de realimentación de estado

m#() = Matriz M_k de asignación de polos

fo() = Matriz resultante de la asignación de polos

f1() = Matriz de realimentación de estado con polos asignados.

u() = Entrada del sistema original

r() = Entrada del sistema desacoplado

x() = Variables de estado

y() = Variables de salida

v() = Referencias externas del sistema desacoplado

e11() = Valores de error del sistema desacoplado

gankp = Ganancia proporcional (K_P) del P.I.D.

ganki = Ganancia integral (K_I) del P.I.D.

gankd = Ganancia integral (K_D) del P.I.D.

Lo indicado en cuanto al software realizado para desacoplamiento, puede ser esquematizado mediante el diagrama de flujo que se presenta en la figura 2.11, en la que se da una clara idea de la estructura del software desarrollado.

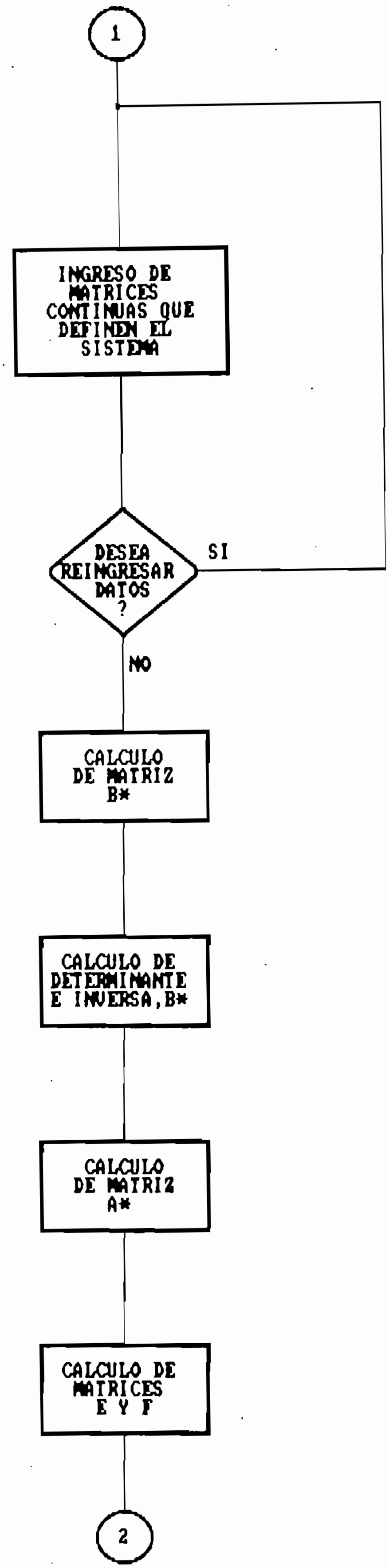


Fig. 2.11 (a) DIAGRAMA DE FLUJO.- DESACOPLAMIENTO

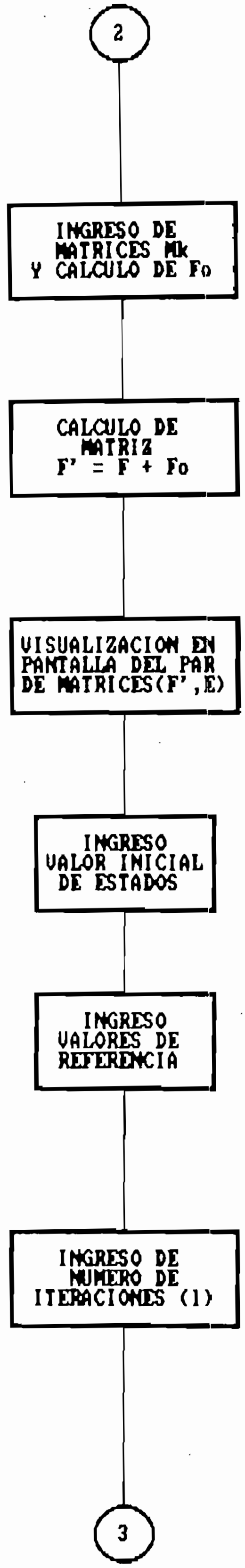


Fig. 2.11 (6) DIAGRAMA DE FLUJO.- DESACOPLAMIENTO

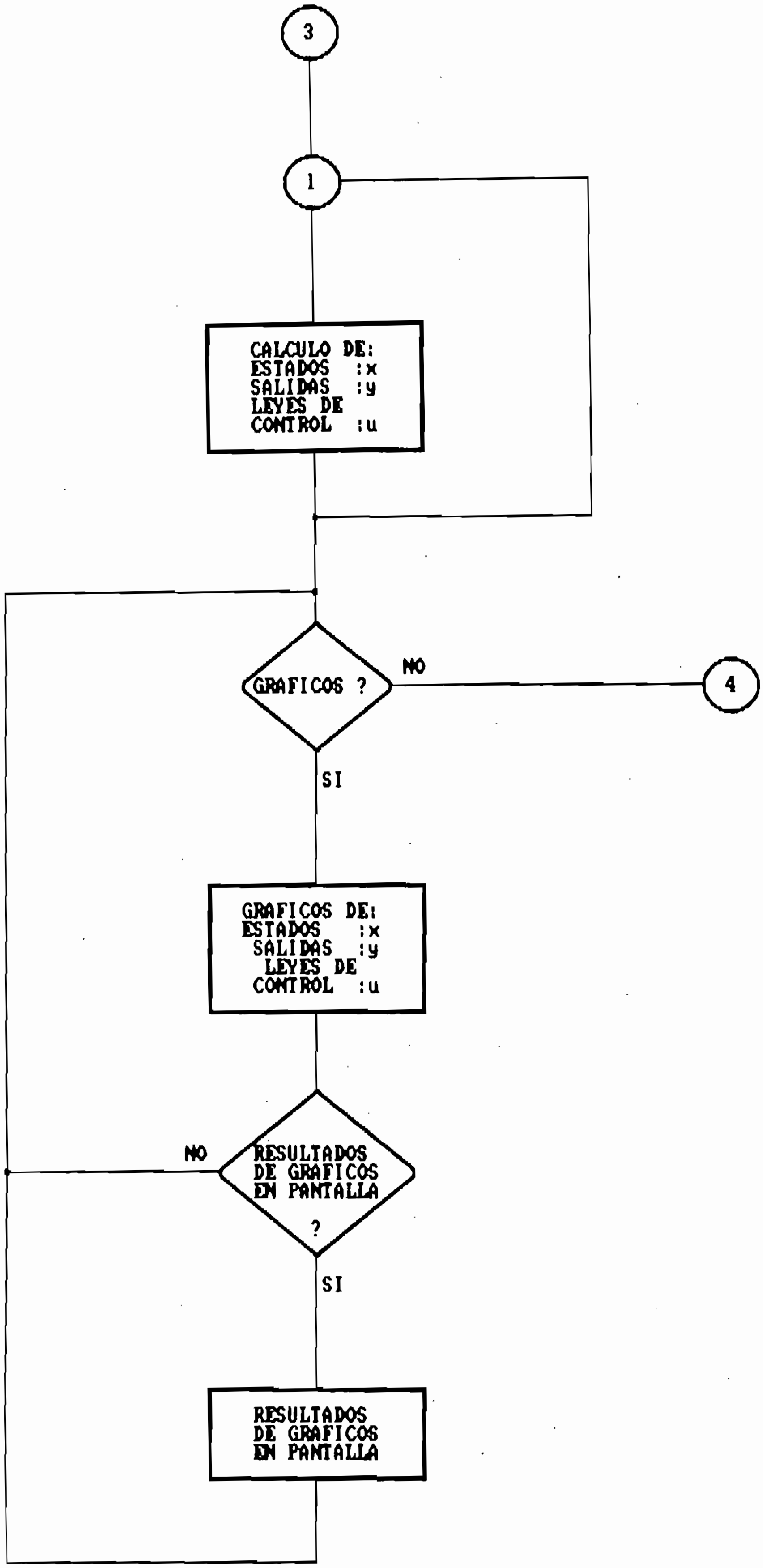


Fig. 2.11 (c) DIAGRAMA DE FLUJO.- DESACOPLAMIENTO

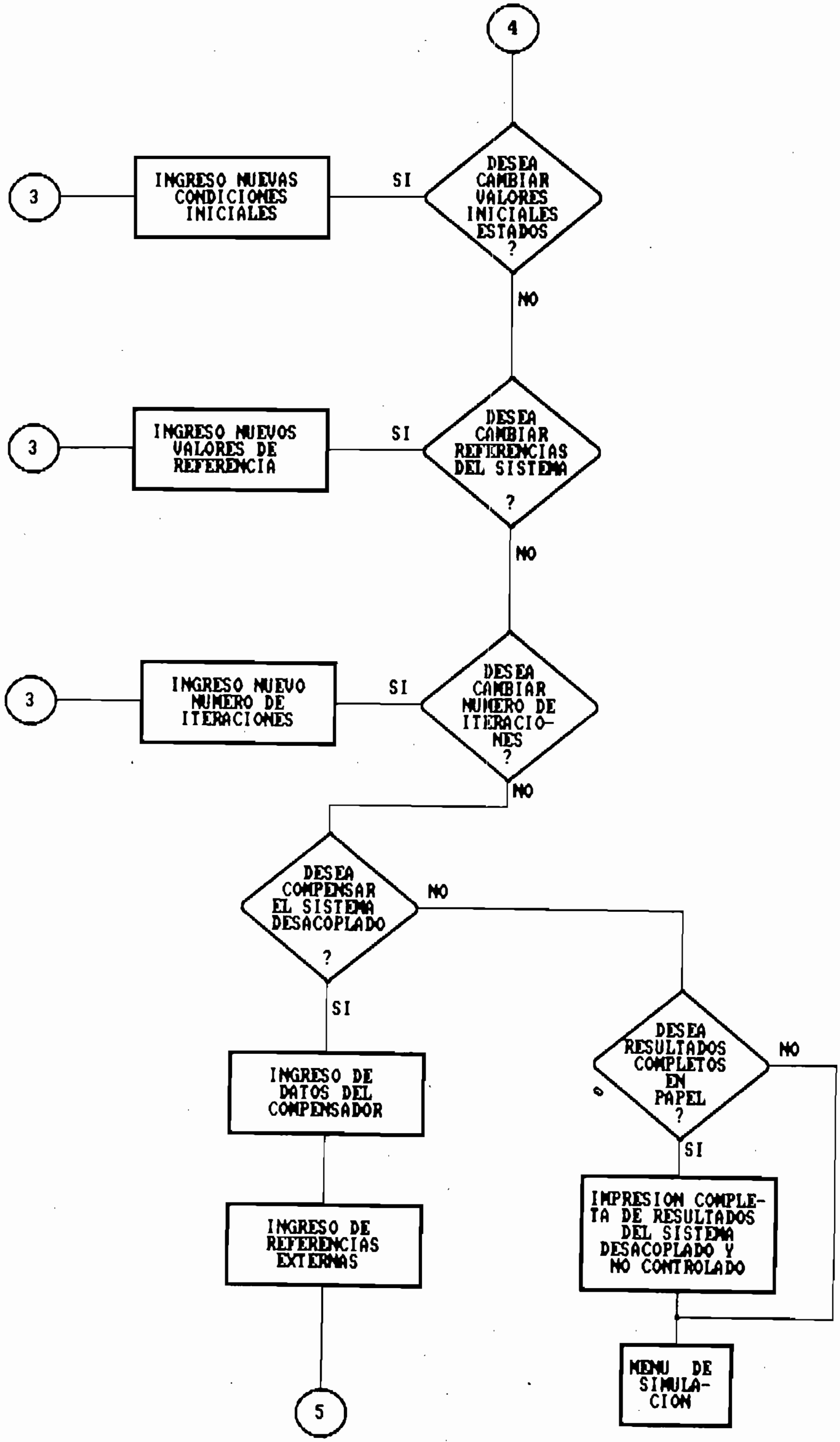


Fig. 2.11 (d) DIAGRAMA DE FLUJO .- DESACOPLAMIENTO

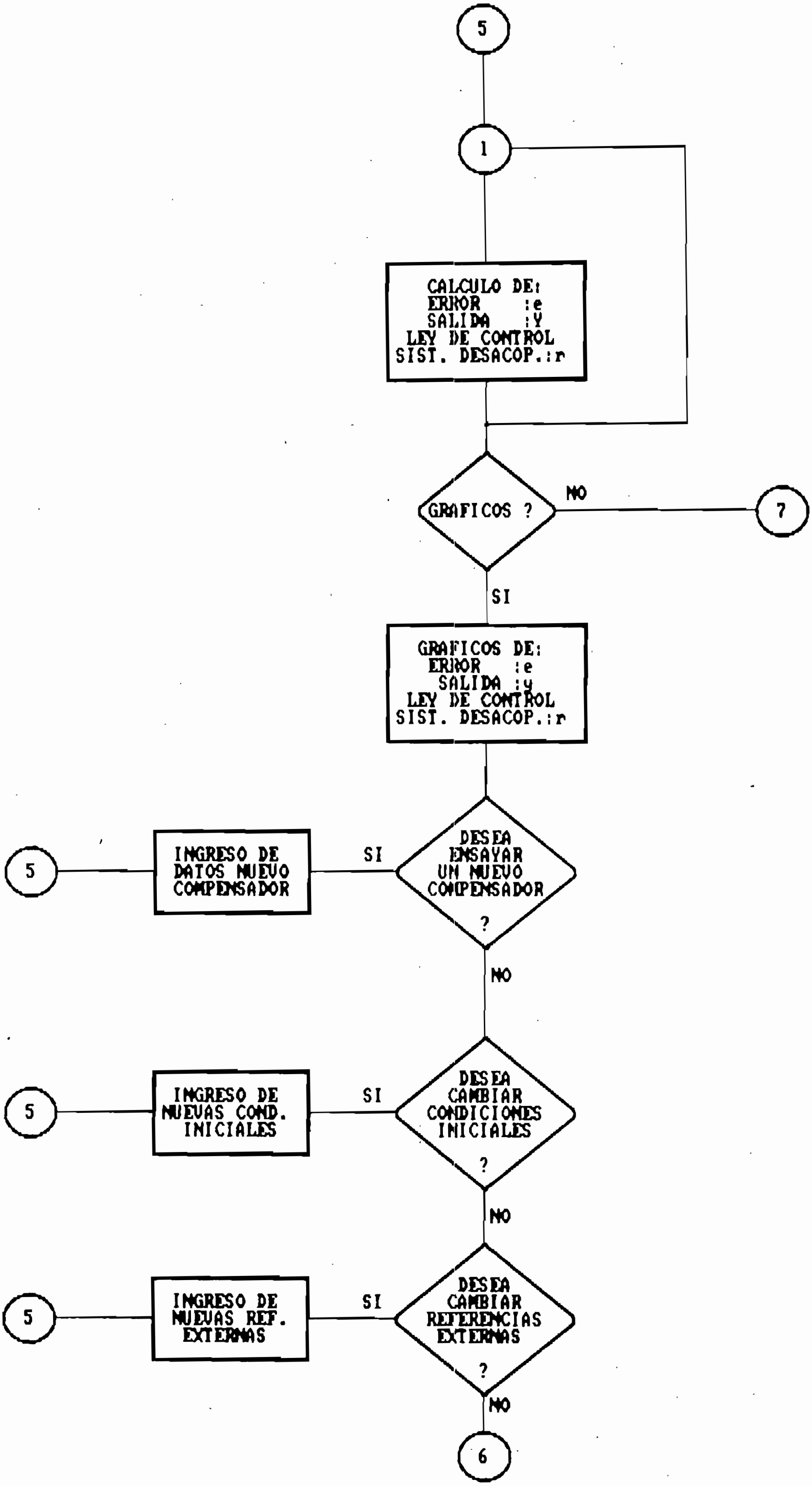


Fig. 2.11 (e) DIAGRAMA DE FLUJO.- DESACOPLAMIENTO

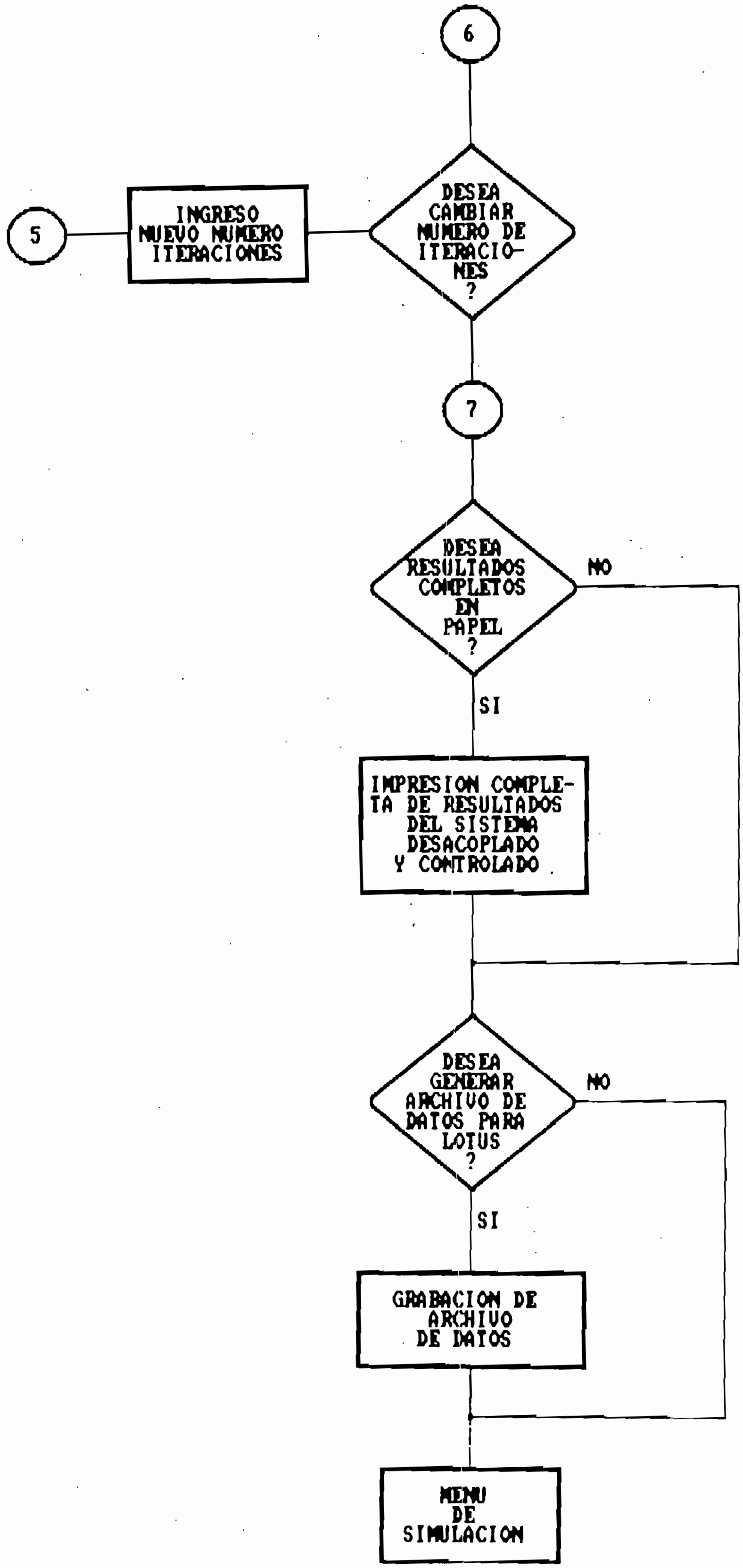


Fig. 2.11 (f) DIAGRAMA DE FLUJO.- DESACOPLAMIENTO

2.7.- SOFTWARE PARA REGULADOR CUADRATICO LINEAL

En el software correspondiente al regulador cuadrático lineal, se realiza primeramente el ingreso de datos tanto de las matrices que definen el sistema en variables de estado como de las matrices de ponderación que determinan los pesos que se dan a las diversas variables del sistema.

Con los valores indicados se procede al cálculo iterativo de la ganancia de realimentación de estado, que permitirá realizar el cálculo y la simulación gráfica de las diversas variables del sistema.

Al igual que en los casos anteriores, se pone énfasis en la parte gráfica; se tiene la posibilidad además de cambiar valores tales como condiciones iniciales de los estados, número de iteraciones en el cálculo, etc.

Cabe mencionar que el regulador cuadrático lineal constituye un problema de seguimiento, sin embargo se ha realizado una adecuación, la cual constituye en sumar un valor de offset al valor de la ley de control calculada por el algoritmo; este valor de offset puede ser modificado por programa de tal modo que pueda apreciarse diversas condiciones de operación.

Finalmente es importante mencionar que el presente software, dispone también de la posibilidad de imprimir resultados completos de la simulación en papel y además permite generar un archivo de datos para graficación posterior mediante el programa LOTUS.

Listado de las principales variables:

n = Orden del sistema

m = Número de entradas

acon() = Matriz continua que define el sistema

bcon() = Matriz continua que define el sistema

a() = Matriz discreta que define el sistema

b() = Matriz discreta que define el sistema

t = Periodo de muestreo

l = Número de iteracciones

q() = Matriz de ponderación simétrica definida positiva

r() = Matriz de ponderación simétrica definida positiva

m#() = Matriz de ponderación

offset() = Valores de offset

k#() = Matriz ganancia de Riccati

g() = Matriz de realimentación óptima de estado

x() = Variables de estado

u() = Leyes de control

En forma precedente se ha indicado en términos generales la estructura del software desarrollado; esto puede apreciarse con mayor detalle en el diagrama de flujo que se presenta en la figura 2.12.

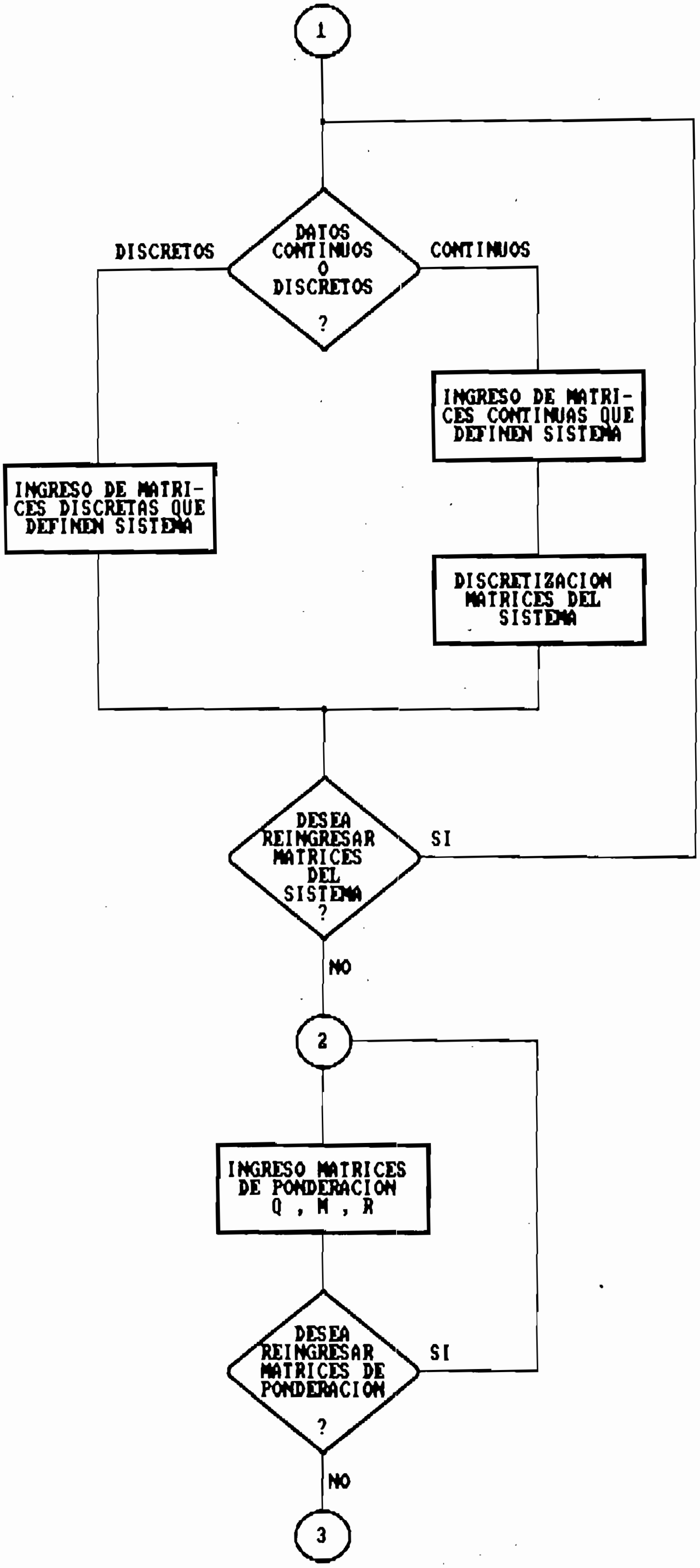


Fig. 2.12 (a) DIAGRAMA DE FLUJO.- CUADRATICO LINEAL

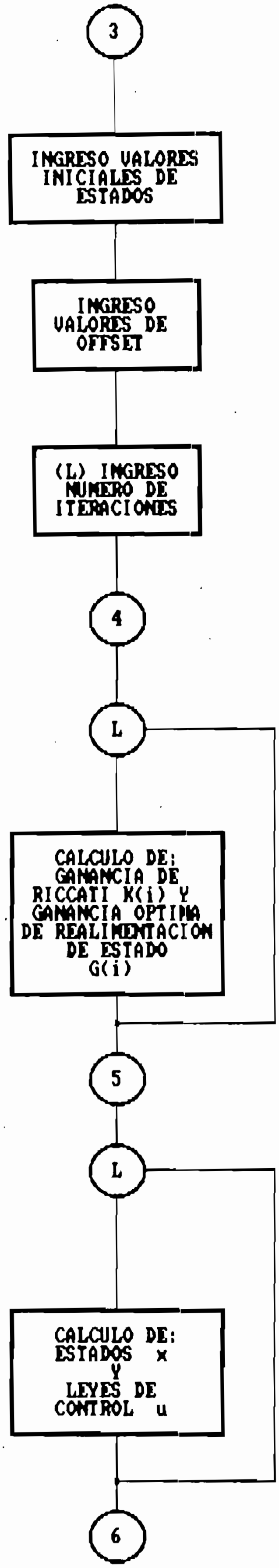


Fig. 2.12 (b) DIAGRAMA DE FLUJO.- REGULADOR CUADRATICO LINEAL

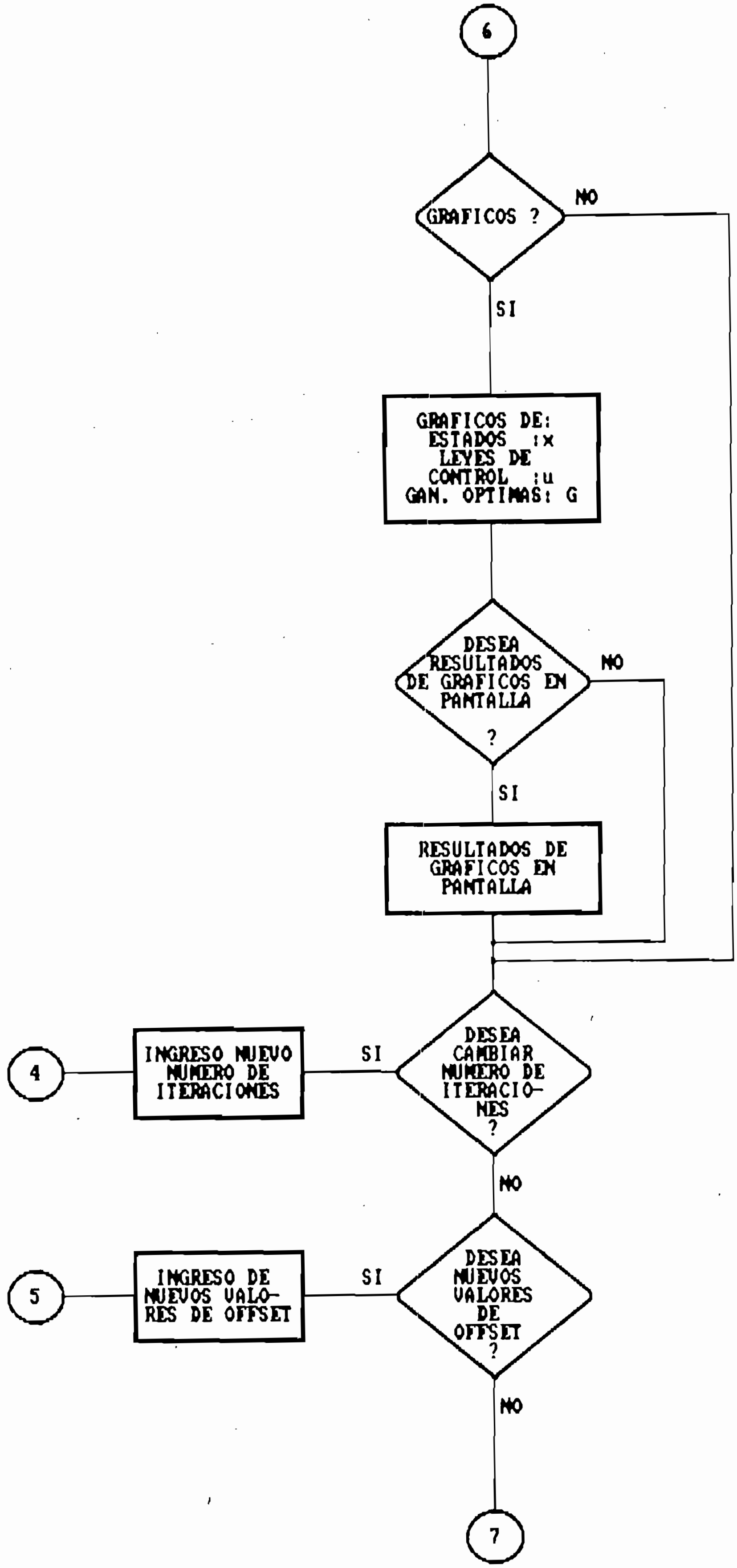


Fig. 2.12 (c) DIAGRAMA DE FLUJO.- REGULADOR CUADRATICO LINEAL

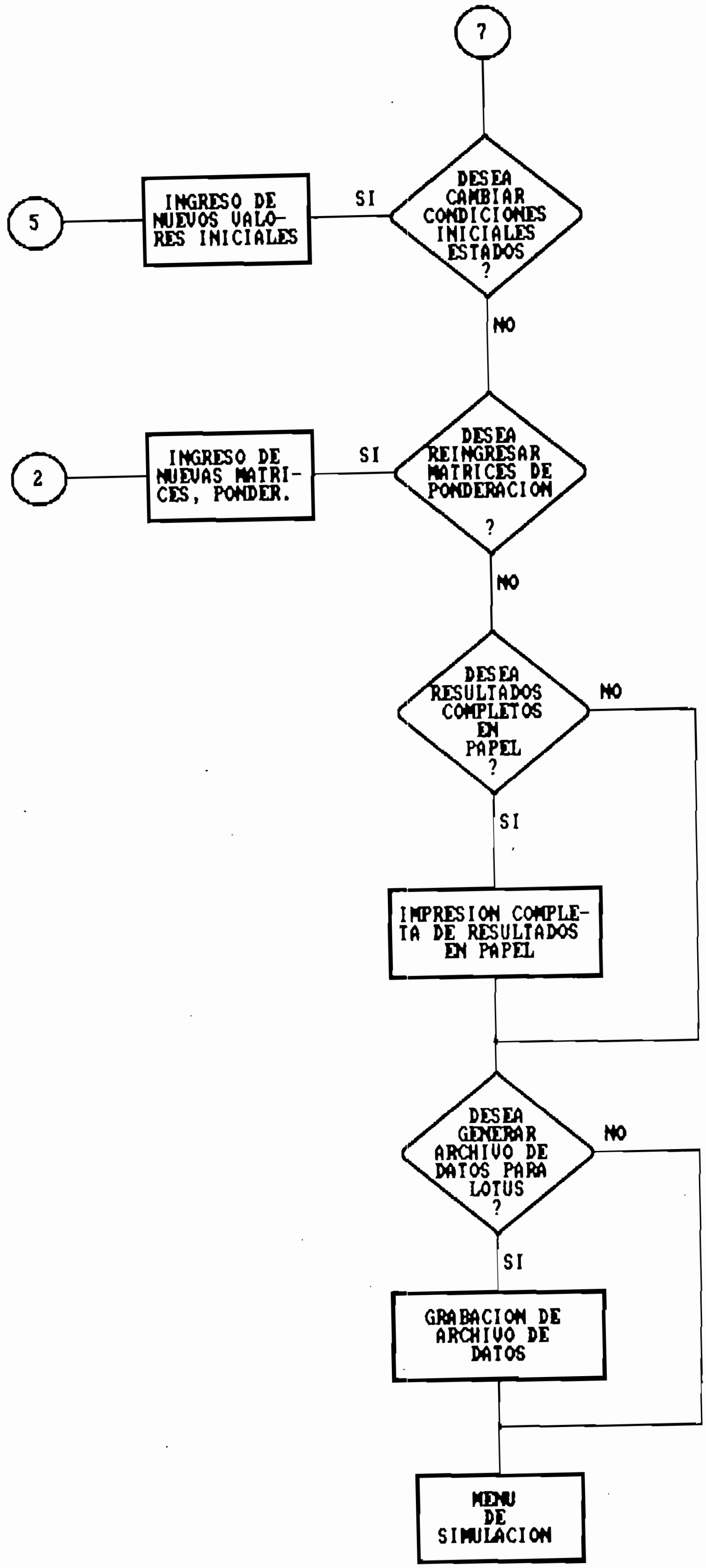


Fig. 2.12 (d) DIAGRAMA DE FLUJO.- REGULADOR CUADRATICO

Una de las partes constitutivas del diagrama de flujo de la figura 2.12 es el cálculo de la ganancia de Riccati y de la ganancia de realimentación óptima de estado, este cálculo consiste en un método recursivo de resolución de la ecuación discreta de Riccati y debido a su importancia se realiza ua descripción con cierto detalle de este tipo de resolución.

Primeramente se debe partir de una condición inicial de la ganancia de Riccati, con lo que se calculará iterativamente tanto la ganancia indicada como la de realimentación óptima de estado en un proceso de adelante hacia atrás hasta llegar al estado estable de solución; lo esquematizado puede se visualizado en la figura 2.13.

El cálculo iterativo mencionado se realiza en base a las ecuaciones 2.42 y 2.43 y se ejecuta el proceso de resolución a partir de un valor inicial para finalmente converger a un valor estable que da como resultado tanto la ganancia de Riccati estable como la ganancia óptima de realimentación estable, esta última es la ganancia que posteriormente será utilizada para simulación y para tiempo real. Debe recalcarse que el cálculo de la ganancia óptima de realimentación de estado constituye un cálculo tipo off-line, que será ingresado para ser ensayado en tiempo real.

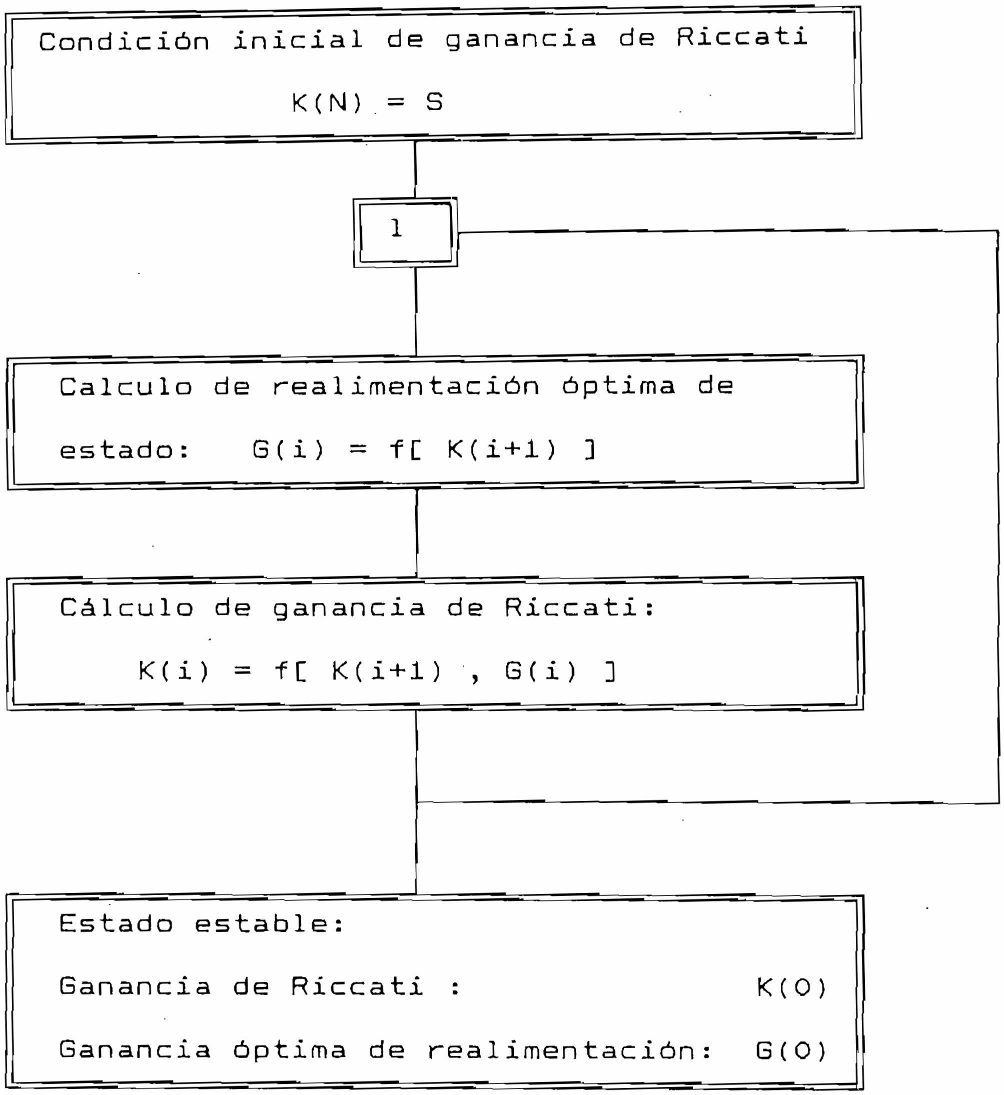


Fig. 2.13 Resolución iterativa de la ecuación de Riccati

2.8.- SOFTWARE PARA GRAFICOS

Como se indicó anteriormente, el programa de simulación tiene un claro énfasis en la parte gráfica, pues se tiene como criterio la visualización de los resultados mediante gráficos.

Para esto se ha realizado una subrutina de gráficos, para lo cual previamente se debe almacenar los datos a ser graficados en el vector: auxiliar(). La subrutina en mención (ventana), primeramente realiza el cálculo del valor máximo y mínimo del vector auxiliar(), de esta, con dichos valores puede dimensionar adecuadamente la ventana que posee las coordenadas lógicas para iniciar el gráfico.

Posteriormente se utiliza la llamada a otra subrutina que permite realizar marcas en cada uno de los ejes coordenados; además se presenta la numeración correspondiente a cada una de estas marcas. Finalmente, se realiza el gráfico mediante la unión del valor de las coordenadas correspondientes.

El software de gráficos podría resumirse de la siguiente manera:

- Cálculo de valor máximo y mínimo
- Creación de ventana y coordenadas lógicas
- Realización de marcas
- Numeración de marcas

- Gráfico

El programa en si se encuentra estructurado de tal modo que se pueda modificar fácilmente la variable que se desea graficar, mediante la elección de las opciones que se presentan en la posición superior de la ventana de gráficos.

Finalmente cabe indicar que al trabajarse en un monitor a color se ha elegido para los gráficos un fondo azul, siendo blanco el color del gráfico en si; se ha utilizado además la definición de SCREEN 9 que permite una buena resolución en la presentación gráfica.

2.9.- RUTINAS DE USO GENERAL

En este numeral, se presentan algunas rutinas que han sido utilizadas a lo largo del desarrollo del programa total; así se tiene:

Determatrix = Subrutina para obtener el determinante de la matriz especificada.

Discret = Subrutina que discretiza las matrices del sistema especificado en variables de estado.

Escalmatrix = Subrutina que multiplica un escalar por un vector.

Expomatrix = Subrutina que eleva una matriz al exponente especificado.

Invermatrix = Subrutina que invierte la matriz especificada.

Mulmatrix = Subrutina que multiplica dos matrices especificadas.

Negmatrix = Subrutina que obtiene el negativo de la matriz especificada.

Submatrix = Subrutina que realiza la resta de dos matrices especificadas.

Summatrix = Subrutina que realiza la suma de dos matrices especificadas.

Transmatrix = Subrutina que transpone la matriz especificada.

Retardo = Subrutina que realiza el retardo necesario para visualizar en pantalla un mensaje momentaneo.

2.10.- SOFTWARE PARA TIEMPO REAL

El software que permite trabajar en tiempo real, desarrollado en QUICK BASIC, versión 4.5 y con la ayuda de las librerías del QUICK 500, ha sido compilado en un programa ejecutable independiente con el nombre de MULTITR.EXE, el cual es llamado en base a un menú principal.

Primeramente se debe indicar que los algoritmos utilizados para tiempo real ocupan tiempo en su ejecución de cálculo, por lo cual debe ser considerado este tiempo para la determinación del periodo de muestreo que puede ser utilizado. Es así como, en el caso de los algoritmos implementados (P.I.D. multilazo, desacoplamiento y regulador cuadrático lineal) se ocupa un máximo de 113 ms., por lo que se ha optado por tomar un tiempo de 150 ms. como el tiempo suficiente para que el algoritmo sea ejecutado en su totalidad; en base a esta consideración, el periodo mínimo de muestreo que se puede utilizar es de 150 ms.

Lo anotado puede esquematizarse bajo el esquema de la figura 2.14 :

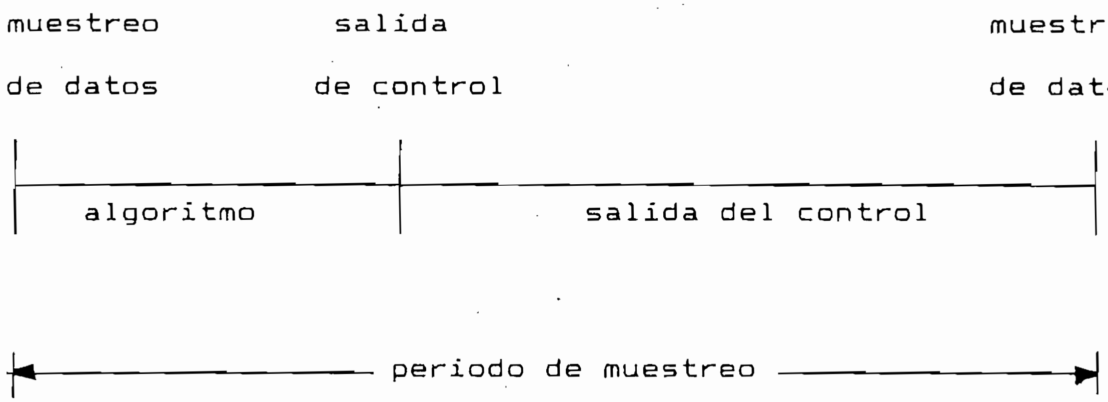


Fig. 2.14 Tareas en tiempo real

Debido al propio funcionamiento del soporte de software del QUICK 500 se ha optado por tener el muestreo de datos cada periodo de muestreo especificado, y la salida de la ley de control, cada fracción entera del mismo, este tiempo debe ser mayor o igual que el tiempo de duración del algoritmo, es decir 150 ms. en el presente caso.

El software desarrollado para tiempo real, primeramente se inicia con un chequeo de que el equipo de adquisición de datos se encuentre encendido, de no ser así, se genera una condición de error que no permite ingresar al software de tiempo real.

Una vez superada la condición indicada, se dispone de un menú de tiempo real en el cual se opta por una de las tres técnicas de control ensayadas, esto es: P.I.D. multilazo, desacoplamiento y regulador cuadrático lineal. Una vez realizada la elección se procede al ingreso de datos necesarios según el tipo de técnica de control elegida y calculada previamente en un esquema off-line.

Posteriormente, se setean las opciones de gráficos y se crean los arreglos necesarios para el almacenamiento tanto de los datos de entrada como de los de salida.

Una vez cumplido con lo anterior se procede al control en tiempo real, el cual básicamente consiste en muestrear datos, realizar el cálculo del respectivo algoritmo y sacar la ley de control necesaria que será ingresada al sistema multivariable especificado.

Lo indicado se puede apreciar con cierto detalle en el diagrama de flujo esquematizado en la figura 2.15.

Listado de las principales variables:

nu = Numero de entradas

m = Número de salidas

n = Número de variables de estado

t = Periodo de muestreo

gtr() = Ganancia óptima de realimenteción de estado

e() = Matriz de precompensación en desacoplamiento

f1() = Matriz de realimentación de estado en desacoplamiento.

ytr() = Vector de salida del sistema (valor medido)

utr() = Vector de control del sistema (valor calculado)

rtr() = Vector de referencia del control

e11tr() = Entrada del control P.I.D.

u11tr() = Salida del control P.I.D.

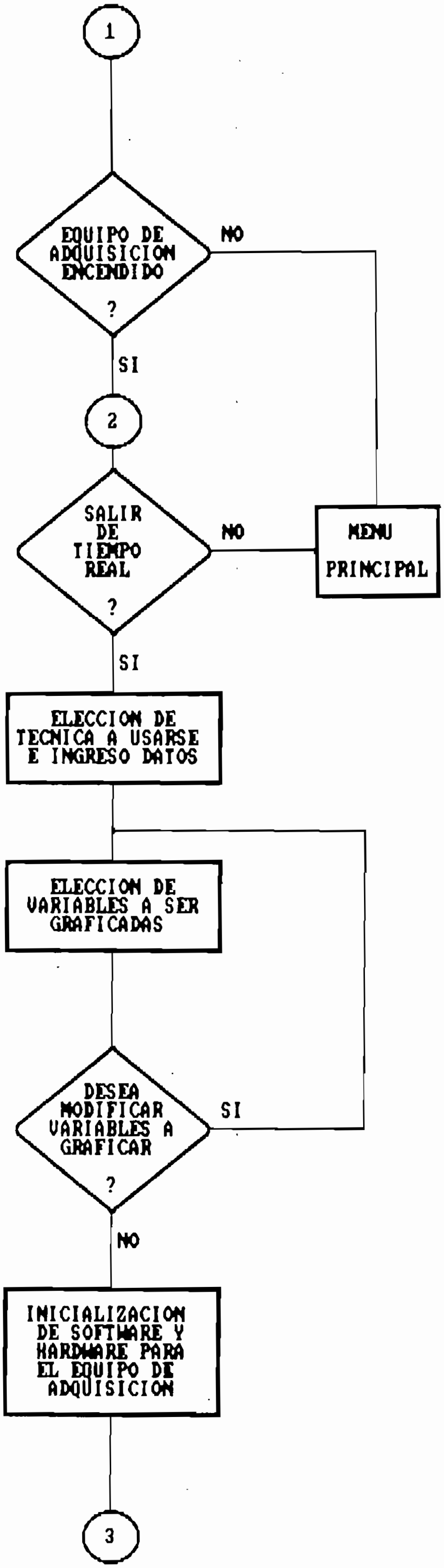


Fig. 2.15 (a) DIAGRAMA DE FLUJO.- CONTROL EN TIEMPO

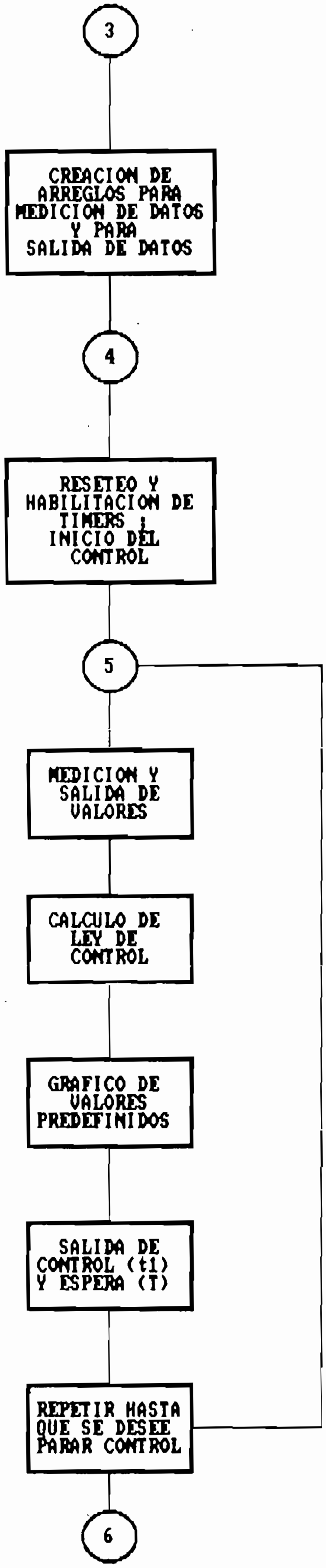


Fig. 2.15 (b) DIAGRAMA DE FLUJO.- CONTROL EN TIEMPO REAL

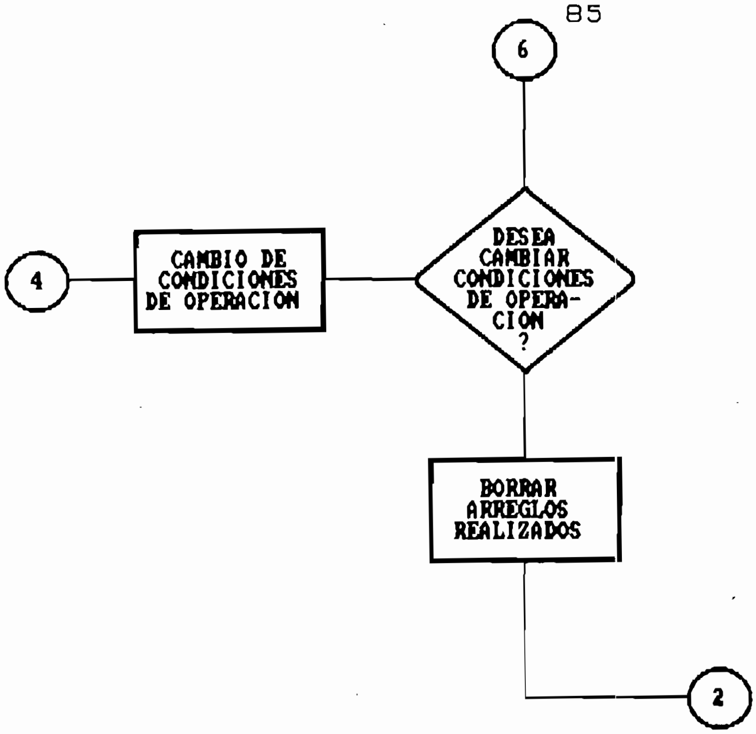
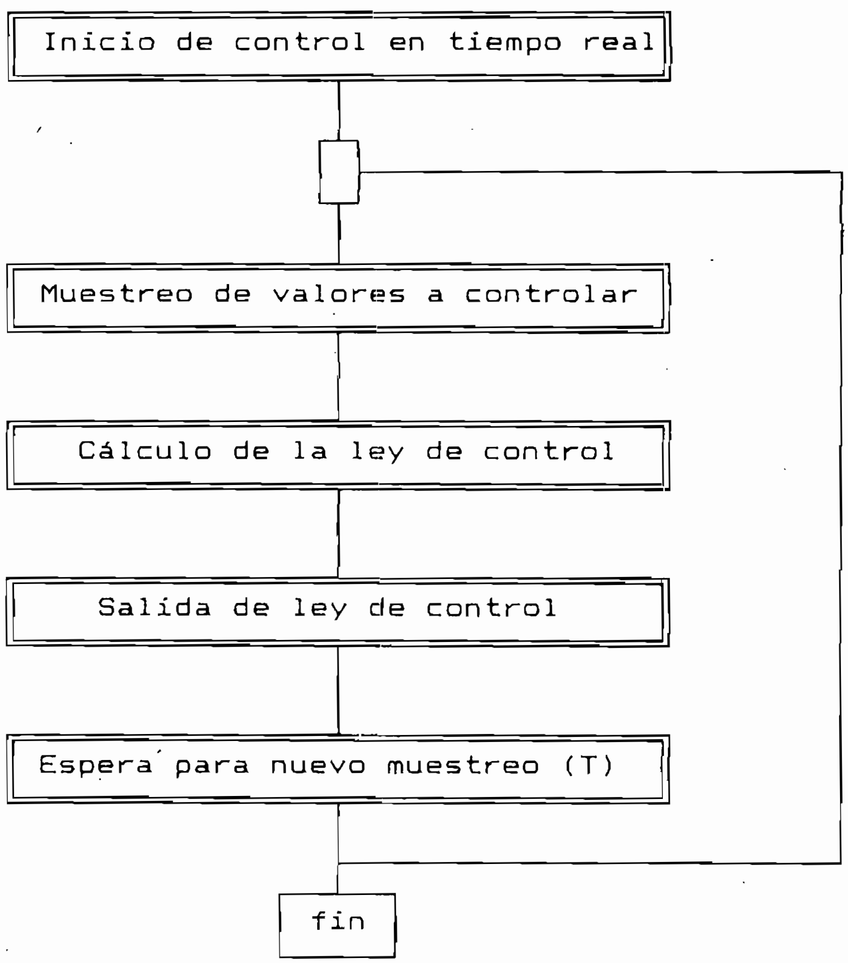


Fig. 2.15 (c) DIAGRAMA DE FLUJO.- CONTROL EN TIEMPO REAL

Dado que en este numeral se tratan tópicos en tiempo real, es importante señalar aspectos tales como muetreo de datos, salida de control y tiempo de espera, ilustrados en la figura 2.14, los mismos que están incluidos en el programa de tiempo real.

Lo expresado puede sintetizarse en el diagrama de flujo de la figura 2.16, el mismo que presenta en terminos generales el esquema seguido en un desarrollo para tiempo real.



Fiq. 2.16 Control en timepo real

Bajo el esquema indicado se ha desarrollado el programa referente a tiempo real para sistemas multivariables, debiendo indicarse que en vista de que el equipo de adquisición de datos y control KEITHLEY 500A dispone de ocho canales análogos de entrada y cinco de salida, y considerando que P.I.D. y desacoplamiento requieren igual número de entradas y salidas, se trabajará con sistemas multivariables de máximo cinco salidas y cinco entradas.

El diagrama de flujo presentado en la figura 2.15 da una clara idea de la estructura del programa desarrollado para tiempo real; el esquema en si, en forma general posee la estructura de un programa base para la realización de control multivariable en tiempo real mediante la implementación de nuevos algoritmos.

En tal caso debe tenerse presente que se tiene que utilizar los vectores ytr() y utr() como variables de medida de datos y salida de valores calculados respectivamente. Debe considerarse además: n como número de estados (si se utiliza este criterio), nu como número de entradas, y m como número de salidas del sistema a ser controlado.Por otro lado si el nuevo o nuevos algoritmos utilizados tienen un tiempo de duración mayor a 150 ms tiene que modificarse la variable entera tmintr1% a un valor mayor al tiempo de duración máximo del algoritmo utilizado.

En definitiva, bajo las consideraciones anotadas cabe recalcar que la implementación de nuevos algoritmos de control requieren en base al programa desarrollado, modificar únicamante el ingreso de los diversos datos propios para cada algoritmo y lógicamente el algoritmo en si, ya que la estructura general para tiempo real en sistemas multivariables es la presentada en el programa realizado. La modificación de las condiciones de operación, esto es, referencias, valores iniciales, compensadores, etc. debe realizarse de acuerdo al algoritmo utilizado.

2.11 .- SIMULACION MEDIANTE EL PROGRAMA CC

El programa CC constituye un paquete especializado de análisis que dispone de una gran cantidad de técnicas de control tanto clásico como moderno, por lo que en el presente trabajo únicamente se exploran algunas de sus potencialidades y específicamente en lo que a sistemas multivariables se refiere.

El software en mención se encuentra estructurado en base a niveles de comandos, cada uno de los cuales tiene por objetivo el análisis de diversas estrategias de control; entre los niveles indicados se tiene:

CC: Se realiza en forma general técnicas de control clásico.

BUILD: Se utiliza para crear y cambiar funciones de transferencia.

MACRO: Usado para crear y editar macros.

MIMO: Trabaja con matriz función de transferencia en sistemas multivariables.

STATE: Contiene algoritmos para trabajar con sistemas definidos en el espacio de estado.

ZEROS: Calcula las raices de un polinomio.

MR: Permite trabajar con submultiplos del periodo de muestreo.

En forma general, el presente tema de tesis al trabajar con sistemas multivariables utiliza principalmente los niveles de comandos CC,STATE, MIMO y MACRO. El último nivel de comandos es utilizado para editar y crear diversos macros que serán llamados desde el nivel CC; los macros constituyen una lista de comandos almacenados en un archivo y ejecutados bajo la llamada a un simple comando de la siguiente manera: @nombre,variables de ingreso.

El nivel de comandos MACRO es esencialmente un editor de texto y como tal puede realizar funciones de busqueda, de borrado, de adición de lineas, etc.; los macros creados en este nivel se almacenan en formato ASCII, por lo cual pueden ser recuperados o creados y editados por otros editores de texto.

Los tópicos abordados mediante la utilización del programa CC se los realiza en base a un conjunto de comandos estructurados en macros, tanto en forma análoga como en forma digital; especificamente se desarrollan cuatro tópicos:

- Regulador cuadrático lineal, el mismo que se realiza en base a comandos de cálculo de la matriz óptima de realimentación, realimentación en si, simulación, gráficos, etc. Este macro se denomina como OPTIMOC y OPTIMOD para sistemas continuos y discretos respectivamente sin condiciones iniciales y, OPTIMOC2 y OPTIMOD2 con condiciones iniciales.

- Chequeo de controlabilidad y observabilidad de sistemas multivariables mediante el comando CONTROL.

- Obtención de la matriz función de transferencia en base a la definición de un sistema en variables de estado y obtención del determinante de la matriz función de transferencia, mediante el macro FUNTRANC para el caso continuo o FUNTRAND para el caso discreto.

- Adicionalmente se ha decidido realizar en forma complementaria un observador; para esto se ha utilizado la parte correspondiente al filtro de Kalman que se dispone en el programa CC y que abarca a sistemas multivariables, pero se tratará desde un punto de vista determinístico. Para el efecto se utilizan tres macros desarrollados, estos son: KALMAN, KALMUL y KALGRAF.

Lo indicado puede esquematizarse mediante la figura 2.17 que da una clara idea de la estructura de los macros desarrollados.

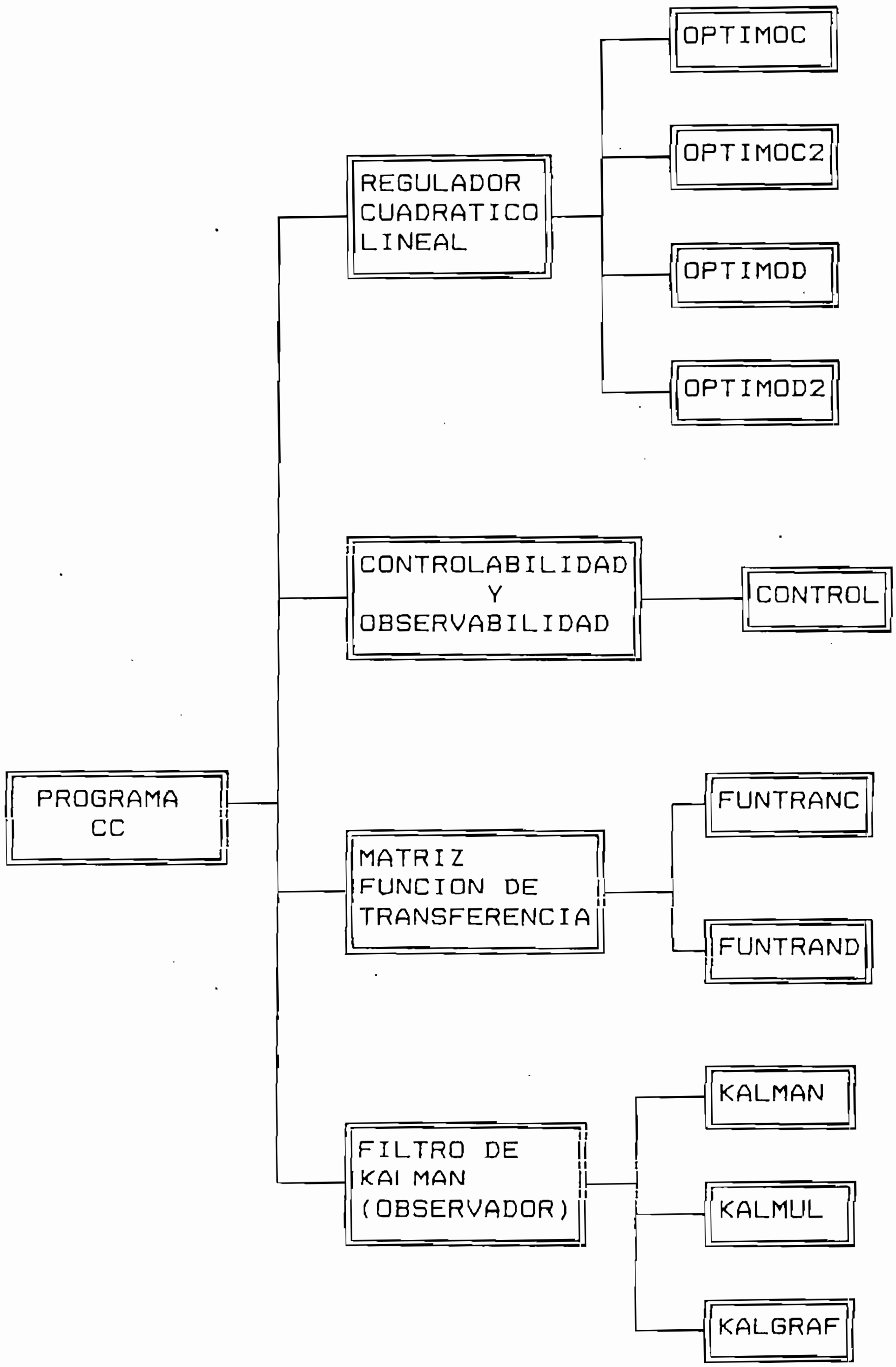


Fig. 2.17 Esquema de macros desarrollados en CC

En los macros realizados se trabaja a partir de la definición de un sistema en variables de estado, para lo cual deben ser ingresadas las matrices que definen el sistema mediante el comando PENTER e indicando primeramente el número de estados, número de entradas y número de salidas, definiendose de esta forma la matriz sistema. Adicionalmente deben ingresarse diversas matrices que intervienen en el cálculo, para esto se ha logrado determinar que se lo puede realizar tambien con el comando PENTER, pero especificando el número de estados igual a cero. PENTER pertenece al nivel de comandos STATE.

En los macros correspondientes a <u>regulador cuadrático lineal</u> primeramente antes de su ejecución deben ser ingresadas las siguientes matrices:

P1 = Matriz sistema (continua)

P11 = Matriz de ponderación Q simétrica semidefinida positiva

P22 = Matriz de ponderación R simétrica definida positiva

P44 = Condiciones iniciales (en OPTIMOC2 y OPOTIMOD2)

Adicionalmente para obtener el gráfico correspondiente se requiere simular el sistema, por lo que debe utilizarse el comando INPUT para generar el archivo <u>entrada</u>, que constituirá la entrada del sistema.

En base a las matrices especificadas se obtiene el gráfico correspondiente a más de las siguientes matrices:

P33 = Matriz óptima de realimentación de estado

P2 = Matriz equivalente del sistema realimentado

Adicionalmente se indica que el comando SIMULATION y DSIMULATION generan un archivo P_i.Y que permite la generación del gráfico correspondiente.

En forma general puede esquematizarse el macro para regulador cuadrático lineal, tanto para el caso continuo como para el caso discreto de acuerdo a las figuras 2.18 y 2.19.

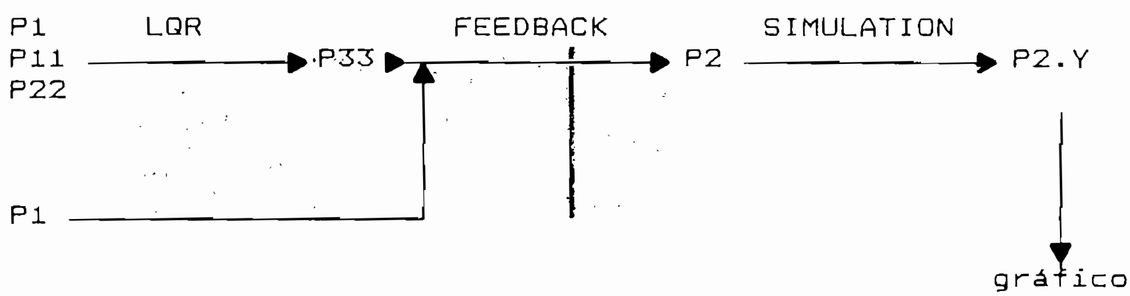


Fig. 2.18 Regulador cuadrático lineal continuo

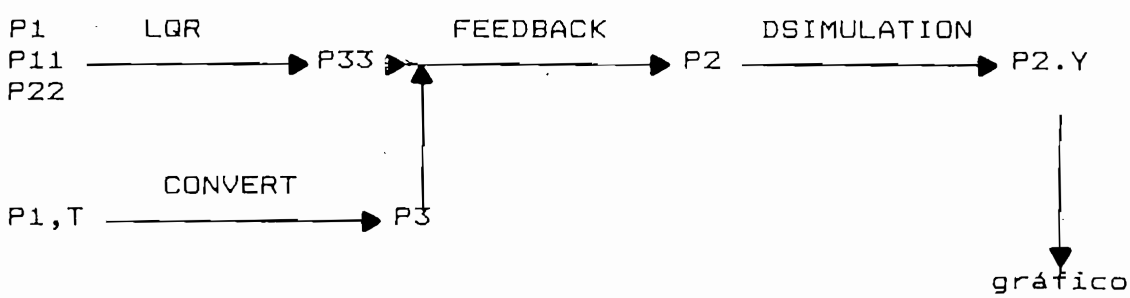


Fig. 2.19 Regulador cuadrático lineal discreto

La diferencia en trabajar o no con condiciones iniciales se presenta en el comando SIMULATION para el caso contínuo o DSIMULATION para el caso discreto.

Para el chequeo de controlabilidad y observabilidad únicamente es necesario utilizar el comando CONTROL y especificar la matriz sistema de la cual se desea realizar la operación indicada; esto es: CONTROL,P_1. Lo indicado se puede esquematizar con la figura 2.20.

$P_1 \xrightarrow{\text{CONTROL}}$ valores propios
observable ?
controlable ?

Fig. 2.20 Controlabilidad y observabilidad

Para obtener la matriz función de transferencia y su determinante en un sistema, a partir de la definición en variables de estado mediante la utilización de los macros desarrollados debe tenerse las siguientes matrices:

P1 = Matriz sistema continua (para FUNTRANC)

P3 = Matriz sistema discreta (para FUNTRAND)

El paso de P1 a P3 se realiza con el comando CONVERT, lo cual no ha sido incluido en el macro FUNTRAND para posibilitar que se pueda encontrar la matriz función de transferencia discreta de un sistema definido ya en forma discreta.

En los macros FUNTRANC y FUNTRAND se presentan las matrices función de transferencia y sus determinantes tanto en forma polinomial como en factores, como se indica en la figura 2.21.

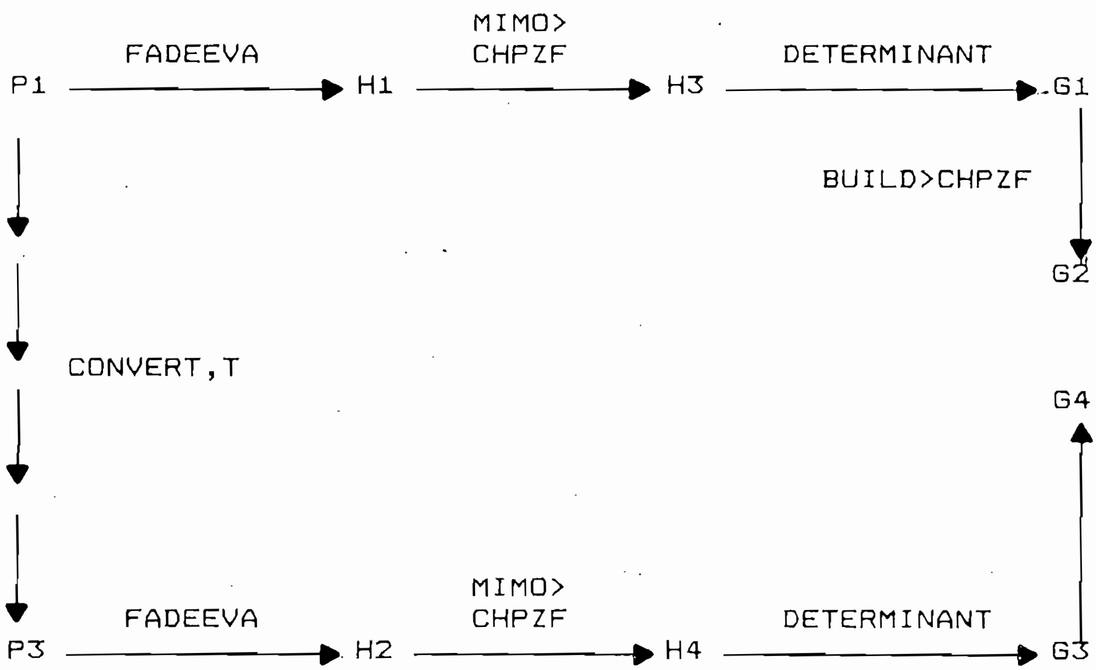


Fig. 2.21 Obtención de matriz función de transferencia

En el esquema indicado se tienen las siguientes definiciones:

H1 = Matriz función de transferencia continua H(s) presentada en forma polinomial.

H3 = Matriz función de transferencia continua H(s) presentada en factores.

H2 = Matriz función de transferencia discreta H(z) presentada en forma polinomial.

H4 = Matriz función de transferencia discreta H(z) presentada en factores.

G1 = Determinante de H(s) presentado en forma polinomial.

G2 = Determinante de H(s) presentado en factores.

G3 = Determinante de H(z) presentado en forma polinomial.

G4 = Determinante de H(z) presentado en factores.

Para la utilización del filtro de Kalman que presenta el programa CC, se debe considerar:

Sistema: $\dot{x} = A\,x + B\,u + \sigma$

$y = C\,x + \alpha$

Observador: $\dot{\hat{x}} = A\,\hat{x} + B\,u + H(y - \hat{y})$

$\hat{y} = C\,\hat{x}$

Covarianzas: $E\{\sigma(t)\,\sigma'(\tau)\} = \boldsymbol{\Theta}\,\delta(t - \tau)$; $\boldsymbol{\Theta} = \boldsymbol{\Theta}' \geq 0$

$E\{\alpha(t)\,\alpha'(\tau)\} = N\,\delta(t - \tau)$; $N = N' > 0$

Error: $E\{[x(t) - \hat{x}(t)]'[x(t) - \hat{x}(t)]\}$

Donde:

A = Matriz que define el sistema

B = Matriz que define el sistema

C = Matriz que define el sistema

σ = Ruido en la entrada

α = Ruido en la medición

$\boldsymbol{\Theta}$ = Matriz de covarianza

N = Matriz de covarianza

x = Vector de estado

$\hat{x}$ = Vector estimado de estado

y = Vector de salida

$\hat{y}$ = Vector estimado de salida

H = Ganancia de Kalman

Nota: (') denota la transpuesta de una matriz.

El propósito del observador es estimar el estado y el del filtro de Kalman es encontrar la ganancia del filtro H que minimice el error medio cuadrático de estimación. La solución es:

$$H = \Sigma\ C'\ N^{-1}$$

donde $\Sigma = \Sigma' \geq 0$ es la única solución de la ecuación algebraica de Riccati:

$$A\ \Sigma + \Sigma\ A' + \boldsymbol{\Theta} - \Sigma\ C'\ N^{-1}\ C\ \Sigma = 0$$

De acuerdo a lo anotado, se utiliza el macro KALMAN para calcular la ganancia del filtro para lo cual debe ser ingresado previamente:

P1 = Matriz sistema (continua)

P111 = Matriz de covarianza $\boldsymbol{\Theta}$

P222 = Matriz de covarianza N

Adicionalmente como variable del macro debe ingresarse un parámertro μ que cambia el tamaño de la covarianza del ruido en la medida.De esta forma el macro KALMAN da como resultado:

P4 = Ganancia del filtro de Kalman

Adicionalmente se ha considerado el error:

$$\tilde{x} = x - \hat{x}$$

Con lo que sustituyendo los valores x y $\hat{x}$ se llega al siguiente resultado:

$$\dot{\tilde{x}} = (A - H\ C)\ \tilde{x}$$

La obtención de la matriz (A - H C) puede realizarse a mano e ingresada mediante el comando ENTER en la matriz P9 como matriz A equivalente dentro de la matriz sistema P9, pero adicionalmente se ha creado el macro KALMUL que permite realizar el cálculo de (A - H C) para lo cual se debe ingresar previamente las matrices originales A y C en P5 y P6 respectivamente; obviamente P9 debe ser ingresado con el resultado de KALMUL que se tiene en P8 (P8 = A - H C).

Finalmente se tiene el macro KALGRAF que se encarga de la simulación y graficación del error $\tilde{x} = x - \hat{x}$.

Todo lo expresado puede resumirse en la figura 2.22.

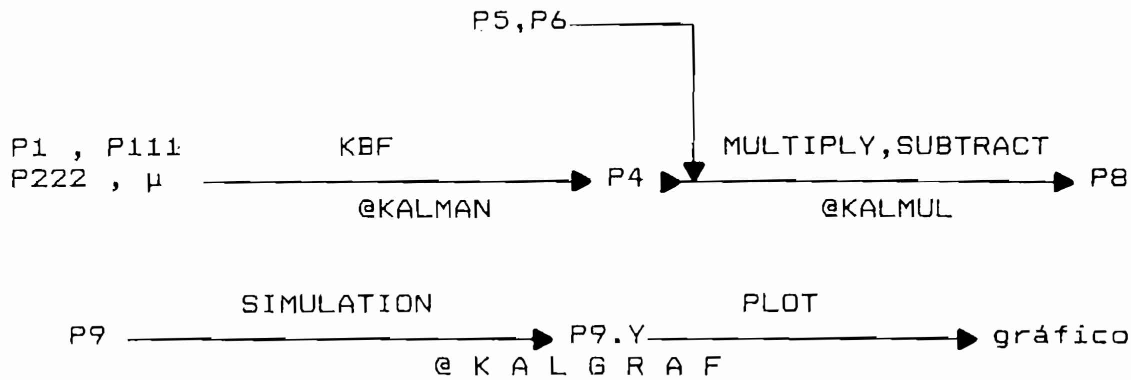


Fig. 2.22   Filtro de Kalman

3 RESULTADOS DE SIMULACION Y CONTROL EN TIEMPO REAL

3.1 RESULTADOS DE SIMULACION DE LOS PROGRAMAS IMPLEMENTADOS

3.2 RESULTADOS DE CONTROL EN TIEMPO REAL

3.- RESULTADOS DE SIMULACION Y CONTROL EN TIEMPO REAL

Para probar la validez del software desarrollado tanto a nivel de simulación como en tiempo real se presenta en este capítulo los resultados de la aplicación a sistemas físicos.

Para el efecto se ha escogido un circuito eléctrico de segundo orden en base a elementos almacenadores de energía (capacitores) y elementos disipadores (resistencias). El circuito mencionado será modelado tanto en variables de estado como en matriz función de transferencia; en el primer caso se realiza la modelación mediante la aplicación de las leyes físicas que gobiernan el comportamiento del sistema, por otro lado, en el segundo caso se aprovecha las facilidades que proporciona el software de control CC para obtener en forma numérica la matriz función de transferencia en base a la modelación en variables de estado.

Cabe recalcar que el cálculo de las diversas matrices que se requieren para el control del sistema de acuerdo a los algoritmos utilizados se lo realiza fuera de línea y dichos valores deben ser ingresados para tiempo real.

La figura 3.1 presenta el diagrama del circuito electrico utilizado en las diversas pruebas realizadas tanto en simulación como en tiempo real.

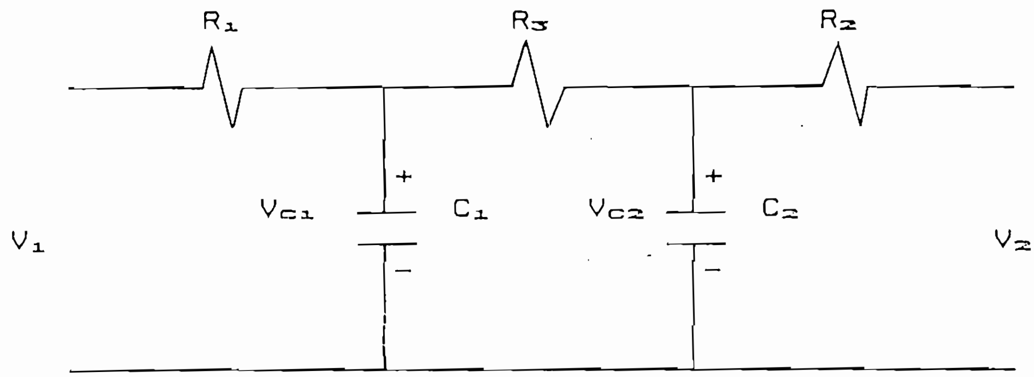


Fig. 3.1 Circuito eléctrico de prueba

En base a las leyes físicas que gobiernan el sistema utilizado se obtiene la modelación en variables de estado de la siguiente manera:

$$\dot{V}_c = A\ V_c + B\ V$$

$$Y = C\ V_c$$

Donde:

V_c = Vector de estado (voltajes en los condensadores)

V = Vector de voltajes de entrada del sistema

Y = Vector de salida del sistema

A = Matriz que define el sistema

B = Matriz que define el sistema

C = Matriz que define el sistema

En base a los parámetros del sistema, las variables mencionadas se definen de la siguiente manera:

$$V_c = \begin{bmatrix} V_{c1} \\ V_{c2} \end{bmatrix} \qquad V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \qquad Y = \begin{bmatrix} V_{c1} \\ V_{c2} \end{bmatrix}$$

$$A = \begin{bmatrix} -(1/C_1)(1/R_1 + 1/R_3) & 1/R_3C_1 \\ 1/R_3C_2 & -(1/C_2)(1/R_2 + 1/R_3) \end{bmatrix}$$

$$B = \begin{bmatrix} 1/R_1C_1 & 0 \\ 0 & 1/R_2C_2 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Luego, se procede a obtener la matriz función de transferencia del sistema en base a la aplicación de la siguiente ecuación:

$$H(s) = C\,(\,sI - A\,)^{-1}\,B$$

Con lo que se obtiene el siguiente resultado:

$$\begin{bmatrix} Y_{c1}(s) \\ Y_{c2}(s) \end{bmatrix} = \frac{1}{\Gamma} \begin{bmatrix} a_3(s - a_5) & a_2a_6 \\ a_3a_4 & a_6(s - a_1) \end{bmatrix} \begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix}$$

Donde:

$$\Gamma = (s - a_1)(s - a_5) - a_2 a_4$$

$$a_1 = -(1/C_1)(1/R_1 + 1/R_3)$$

$$a_2 = 1/R_3C_1$$

$$a_3 = 1/R1C_1$$

$$a_4 = 1/R_3C_2$$

$$a_5 = -(1/C_2)(1/R_2 + 1/R_3)$$

$$a_6 = 1/R_2C_2$$

Para poder trabajar con un sistema cuyas constantes de tiempo permitan visualizar en forma objetiva la utilidad y aplicación de los algoritmos desarrollados en base al equipo existente se han seleccionado los siguientes valores de condensadores y resistencias:

C_1 = 1000 uF $\quad$ R_1 = 10 kΩ $\quad$ R_3 = 100 kΩ

C_2 = 1000 uF $\quad$ R_2 = 15 kΩ

Con lo que se tienen las siguientes matrices de la modelación a variables de estado:

$$A = \begin{bmatrix} -0.11 & 0.01 \\ 0.01 & -0.076667 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.06667 \end{bmatrix}$$

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

El mismo sistema puede ser definido en términos de Laplace de la siguiente manera:

$$\begin{bmatrix} V_{c1}(s) \\ V_{c2}(s) \end{bmatrix} = \frac{1}{\Gamma_1} \begin{bmatrix} b1_1 & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix}$$

Donde:

$$\Gamma_1 = (s + 0.0739)(s + 0.1128)$$

$$b_{11} - 0.1(s + 0.076667)$$

$$b_{12} = 0.0006667$$

$$b_{21} = 0.001$$

$$b_{22} = 0.066667(s + 0.11)$$

Considerando además que se tiene una constante de tiempo mínima de 8.87 segundos, se utilizará un periodo de muestreo aproximadamente unas diez veces menor, por lo que se ha elegido un periodo de muestreo adecuado de 0.9 segundos.

3.1.- RESULTADOS DE SIMULACION DE LOS PROGRAMAS IMPLEMENTADOS

Para obtener resultados de los programas desarrollados se ha procedido a realizar algunos ejemplos con muchas variaciones, por lo que en este numeral se presentarán los más representativos, luego de una selección adecuada; es lógico que no se presenten todos los resultados obtenidos por razones de espacio.

Primeramente se ha procedido a utilizar el algoritmo de P.I.D. multilazo para lo cual se tomó tres lazos independientes que fueron controlados mediante la técnica de P.I.D.; los sistemas anotados presentan las siguientes funciones de transferencia de lazo abierto:

$$G1(s) = \frac{8}{(s+1)(s+4)}$$

$$G2(s) = \frac{1}{s(s+1)}$$

$$G3(s) = \frac{1}{(s+1)}$$

A continuación se presentan los resultados obtenidos que se han imprimido mediante una opción del programa:

P.I.D. MULTILAZO

L A Z O 1

Lazo 1 .- Coeficientes de retardo en y

Coeficiente de retardo en y(k- 1)= 1.88015
Coeficiente de retardo en y(k- 2)=-.8825
Coeficiente de retardo en y(k- 3)= 0
Coeficiente de retardo en y(k- 4)= 0
Coeficiente de retardo en y(k- 5)= 0

Lazo 1 .- Coeficientes de retardo en u

Coeficiente de retardo en u(k- 1)= .0024
Coeficiente de retardo en u(k- 2)= .0023
Coeficiente de retardo en u(k- 3)= 0
Coeficiente de retardo en u(k- 4)= 0
Coeficiente de retardo en u(k- 5)= 0

L A Z O 2

Lazo 2 .- Coeficientes de retardo en y

Coeficiente de retardo en y(k- 1)= 1.99005
Coeficiente de retardo en y(k- 2)=-.99005
Coeficiente de retardo en y(k- 3)= 0
Coeficiente de retardo en y(k- 4)= 0
Coeficiente de retardo en y(k- 5)= 0

Lazo 2 .- Coeficientes de retardo en u

Coeficiente de retardo en u(k- 1)= .0000498
Coeficiente de retardo en u(k- 2)= .0000497
Coeficiente de retardo en u(k- 3)= 0
Coeficiente de retardo en u(k- 4)= 0
Coeficiente de retardo en u(k- 5)= 0

L A Z O 3

Lazo 3 .- Coeficientes de retardo en y

Coeficiente de retardo en y(k- 1)= .99005
Coeficiente de retardo en y(k- 2)= 0
Coeficiente de retardo en y(k- 3)= 0
Coeficiente de retardo en y(k- 4)= 0
Coeficiente de retardo en y(k- 5)= 0

Lazo 3 .- Coeficientes de retardo en u

Coeficiente de retardo en u(k- 1)= 9.950000000000001D-03
Coeficiente de retardo en u(k- 2)= 0
Coeficiente de retardo en u(k- 3)= 0
Coeficiente de retardo en u(k- 4)= 0
Coeficiente de retardo en u(k- 5)= 0

COMPENSACION

Compensador 1

Ganancia proporcional .... Kp= 5.50000
Ganancia integral .... Ki= 5.00000
Ganancia derivativa .... Kd= 0.30000

Compensador 2

Ganancia proporcional .... Kp= 15.00000
Ganancia integral .... Ki= 0.00000
Ganancia derivativa .... Kd= 15.00000

Compensador 3

Ganancia proporcional .... Kp= 15.00000
Ganancia integral .... Ki= 15.00000
Ganancia derivativa .... Kd= 0.00000

P.I.D. MULTILAZO

Valores resultantes de la simulación:

Periodo de muestreo (s) .... T= .01
Número de iteraciones .... N= 200
Tiempo total del proceso (s) .... t= 2

Valores de referencia utilizados
referencia 1 = 1
referencia 2 = 1
referencia 3 = 2

Valores de salida

Salida y 1
Valor inicial .... Vo 0.00000
Valor final .... Vf 0.99063
Valor máximo .... Vmax 1.00327
Valor mínimo .... Vmin 0.00000

Salida y 2
Valor inicial .... Vo 0.00000
Valor final .... Vf 0.99995
Valor máximo .... Vmax 0.99995
Valor mínimo .... Vmin 0.00000

Salida y 3
Valor inicial .... Vo 0.00000
Valor final .... Vf 2.00000
Valor máximo .... Vmax 2.00000
Valor mínimo .... Vmin 0.00000

Valores de las leyes de control

Ley de control u 1
Valor inicial .... Vo 35.52500
Valor final .... Vf 0.49730
Valor máximo .... Vmax 35.52500
Valor mínimo .... Vmin -0.95431

Ley de control u 2
Valor inicial .... Vo 1515.00000
Valor final .... Vf 0.00000
Valor máximo .... Vmax 1515.00000
Valor mínimo .... Vmin -205.87723

Ley de control u 3
Valor inicial .... Vo 30.15000
Valor final .... Vf 2.00000
Valor máximo .... Vmax 30.15000
Valor mínimo .... Vmin 2.00000

Valores de error del sistema controlado

Error e 1
Valor inicial .... Vo 1.00000
Valor final .... Vf 0.00937
Valor máximo .... Vmax 1.00000
Valor mínimo .... Vmin -0.00327

Error e 2
Valor inicial .... Vo 1.00000
Valor final .... Vf 0.00005
Valor máximo .... Vmax 1.00000
Valor mínimo .... Vmin 0.00005

Error e 3
Valor inicial .... Vo 2.00000
Valor final .... Vf -0.00000
Valor máximo .... Vmax 2.00000
Valor mínimo .... Vmin -0.00000

Los resultados obtenidos son satisfactorios y para el control se partió de un diseño inicial, para luego proceder a realizar ajustes que determinan los resultados indicados.

Los gráficos de las principales variables del sistema han sido visualisados en pantalla, teniéndose la posibilidad de generar un archivo de datos para ser recuperados en LOTUS y poder obtener resultados gráficos en papel, sin embargo no se ha creido necesario utilizar esta opción.

Posteriormente se realizaron pruebas con la versión de P.I.D. multivariable, para lo cual se trabajó con el sistema que representa el circuito de la figura 3.1 y con los valores indicados al principio del presente capitulo, en los cuales desde luego no se ha tomado en cuenta la interacción.

Los resultados son los siguientes:

P.I.D. MULTIVARIABLE

SISTEMA ORIGINAL NO CONTROLADO

Orden del sistema : 2
Numero de entradas : 2
Numero de salidas : 2

Matriz continua A

-0.11000	0.01000
0.01000	-0.07667

Matriz continua B

0.10000	0.00000
0.00000	0.06667

Matriz continua C

1.00000	0.00000
0.00000	1.00000

COMPENSACION

Compensador 1

Ganancia proporcional .... Kp= 3.00000
Ganancia integral .... Ki= 0.20000
Ganancia derivativa .... Kd= 1.00000

Compensador 2

Ganancia proporcional .... Kp=   3.00000
Ganancia integral .... Ki=   0.20000
Ganancia derivativa .... Kd=   1.00000

P.I.D.   MULTIVARIABLE

Valores resultantes de la simulación:

Periodo de muestreo (s) .... T= .9
Número de iteraciones .... N= 200
Tiempo total del proceso (s) .... t= 180

Valores de referencia utilizados
referencia 1 = 1
referencia 2 = 2

Valores de salida

Salida   y 1
Valor inicial .... Vo   0.00000
Valor final .... Vf   1.00000
Valor máximo .... Vmax   1.00000
Valor mínimo .... Vmin   0.00000

Salida   y 2
Valor inicial .... Vo   0.00000
Valor final .... Vf   2.00000
Valor máximo .... Vmax   2.00000
Valor mínimo .... Vmin   0.00000

Valores de las leyes de control

Ley de control   u 1
Valor inicial .... Vo   4.20111
Valor final .... Vf   0.90000
Valor máximo .... Vmax   4.20111
Valor mínimo .... Vmin   0.89434

Ley de control   u 2
Valor inicial .... Vo   8.40222
Valor final .... Vf   2.15000
Valor máximo .... Vmax   8.40222
Valor mínimo .... Vmin   2.15000

Valores de error del sistema controlado

Error   e 1
Valor inicial .... Vo   1.00000
Valor final .... Vf   0.00000
Valor máximo .... Vmax   1.00000
Valor mínimo .... Vmin   0.00000

Error   e 2
Valor inicial .... Vo   2.00000
Valor final .... Vf   -0.00000
Valor máximo .... Vmax   2.00000
Valor mínimo .... Vmin   -0.00000

Los resultados aqui obtenidos son satisfactorios; se ha realizado el ajuste correspondiente de valores despreciando la interacción y considerando una ganancia integral que asegure que no existe error en estado estable, pero que tampoco lleve el control a la saturación, una ganancia proporcional relativamente alta para una respuesta rápida y una no muy significativa ganancia derivativa.

Como era de esperarse se puede apreciar de las pruebas realizadas que una variación en una de las entradas del sistema multivariable en cuestión afecta sobre la respuesta transitoria de la segunda salida, puesto que no se ha desacoplado el sistema.

Luego, se utilizó el mismo circuito de la figura 3.1 utilizado en la prueba anterior, pero en este caso se realiza el desacoplamiento del mismo, con lo que se obtienen los siguientes resultados:

```
DESACOPLAMIENTO

SISTEMA ORIGINAL NO DESACOPLADO

Orden del sistema   : 2
Numero de salidas   : 2
Numero de entradas  : 2

Matriz continua A

   -0.11000    0.01000
    0.01000   -0.07667
Matriz continua B

    0.10000    0.00000
    0.00000    0.06667
Matriz continua C

    1.00000    0.00000
    0.00000    1.00000
```

Matrices de asignación de polos Mk

Matriz M 0

```
 -0.25000    0.00000
  0.00000   -0.16000
```

MATRICES DE DESACOPLAMIENTO

Matriz de realimentación de estado F1

```
 -1.40000   -0.10000
 -0.15000   -1.25000
```

Matriz de preamplificación E

```
 10.00000    0.00000
  0.00000   15.00000
```

DESACOPLAMIENTO .- Sistema no controlado

Valores resultantes de la simulación:

```
Periodo de muestreo (s) ............. T= .9
Número de iteraciones ............... N= 100
Tiempo total del proceso (s) ......... t= 90
```

Valores de los estados

```
Estado         x 1
Valor inicial .................. Vo     0.00000
Valor final .................... Vf     4.00000
Valor máximo ................. Vmax     4.00004
Valor mínimo ................. Vmin     0.00000

Estado         x 2
Valor inicial .................. Vo     0.00000
Valor final .................... Vf    12.50000
Valor máximo ................. Vmax    12.50000
Valor mínimo ................. Vmin     0.00000
```

Valores de las leyes de control

```
Ley de control       u 1
Valor inicial .................. Vo    10.00000
Valor final .................... Vf     3.15000
Valor máximo ................. Vmax    10.00000
Valor mínimo ................. Vmin     3.15000

Ley de control       u 2
Valor inicial .................. Vo    30.00000
Valor final .................... Vf    13.77501
Valor máximo ................. Vmax    30.00000
Valor mínimo ................. Vmin    13.77501
```

Valores de salida del sistema desacoplado

Salida y 1
Valor inicial .... Vo 0.00000
Valor final .... Vf 4.00000
Valor máximo .... Vmax 4.00004
Valor mínimo .... Vmin 0.00000

Salida y 2
Valor inicial .... Vo 0.00000
Valor final .... Vf 12.50000
Valor máximo .... Vmax 12.50000
Valor mínimo .... Vmin 0.00000

El sistema, de esta manera ha sido desacoplado, pues la variación de una entrada no afecta mas que a la salida correspondiente y no a la otra salida. Sin embargo se ha podido observar que la salida difiere bastante de la referencia del nuevo sistema, por lo que se hace necesario la compensación del sistema ya desacoplado para lo cual se ha utilizado un compensador tipo P.I.D., con lo que se obtienen los siguientes resultados:

DESACOPLAMIENTO

SISTEMA ORIGINAL NO DESACOPLADO

Orden del sistema : 2
Numero de salidas : 2
Numero de entradas : 2

Matriz continua A

-0.11000	0.01000
0.01000	-0.07667

Matriz continua B

0.10000	0.00000
0.00000	0.06667

Matriz continua C

1.00000	0.00000
0.00000	1.00000

Matrices de asignación de polos Mk

Matriz M 0

```
   -0.25000    0.00000
    0.00000   -0.16000
```

MATRICES DE DESACOPLAMIENTO

Matriz de realimentación de estado F1

```
   -1.40000   -0.10000
   -0.15000   -1.25000
```

Matriz de preamplificación E

```
   10.00000    0.00000
    0.00000   15.00000
```

COMPENSACION

Compensador 1

```
Ganancia proporcional ......... Kp=    0.50000
Ganancia integral ............. Ki=    0.20000
Ganancia derivativa ........... Kd=    0.10000
```

Compensador 2

```
Ganancia proporcional ......... Kp=    0.50000
Ganancia integral ............. Ki=    0.10000
Ganancia derivativa ........... Kd=    0.10000
```

DESACOPLAMIENTO .- Sistema controlado

Valores resultantes de la simulación:

```
Periodo de muestreo (s) .............. T= .9
Número de iteraciones ................ N= 100
Tiempo total del proceso (s) ......... t= 90
```

Valores de referencia utilizados
referencia 1 = 1
referencia 2 = 2

Valores de salida

```
Salida         y 1
Valor inicial .................... Vo     0.00000
Valor final ...................... Vf     1.00000
Valor máximo ................... Vmax     1.05567
Valor mínimo ................... Vmin     0.00000
```

Salida	y 2		
Valor inicial	....	Vo	0.00000
Valor final	....	Vf	2.00000
Valor máximo	....	Vmax	2.04068
Valor mínimo	....	Vmin	0.00000

Valores de las leyes de control

Ley de control	u 1		
Valor inicial	....	Vo	7.01111
Valor final	....	Vf	0.90000
Valor máximo	....	Vmax	7.01111
Valor mínimo	....	Vmin	0.82133

Ley de control	u 2		
Valor inicial	....	Vo	19.68333
Valor final	....	Vf	2.15000
Valor máximo	....	Vmax	19.68333
Valor mínimo	....	Vmin	2.10899

Valores de error del sistema desacoplado y compensado

Error	e 1		
Valor inicial	....	Vo	1.00000
Valor final	....	Vf	-0.00000
Valor máximo	....	Vmax	1.00000
Valor mínimo	....	Vmin	-0.05567

Error	e 2		
Valor inicial	....	Vo	2.00000
Valor final	....	Vf	-0.00000
Valor máximo	....	Vmax	2.00000
Valor mínimo	....	Vmin	-0.04068

De esta manera el sistema desacoplado ha sido compensado satisfactoriamente en base a los parámetros del compensador utilizado, los cuales han sido ajustados en base a un diseño inicial.

Los resultados coinciden con lo esperado, lo cual puede ser visualizado mediante los gráficos realizados en LOTUS con el archivo de datos generado en el software desarrollado; los gráficos en mención se presentan en las figuras 3.2 y 3.3.

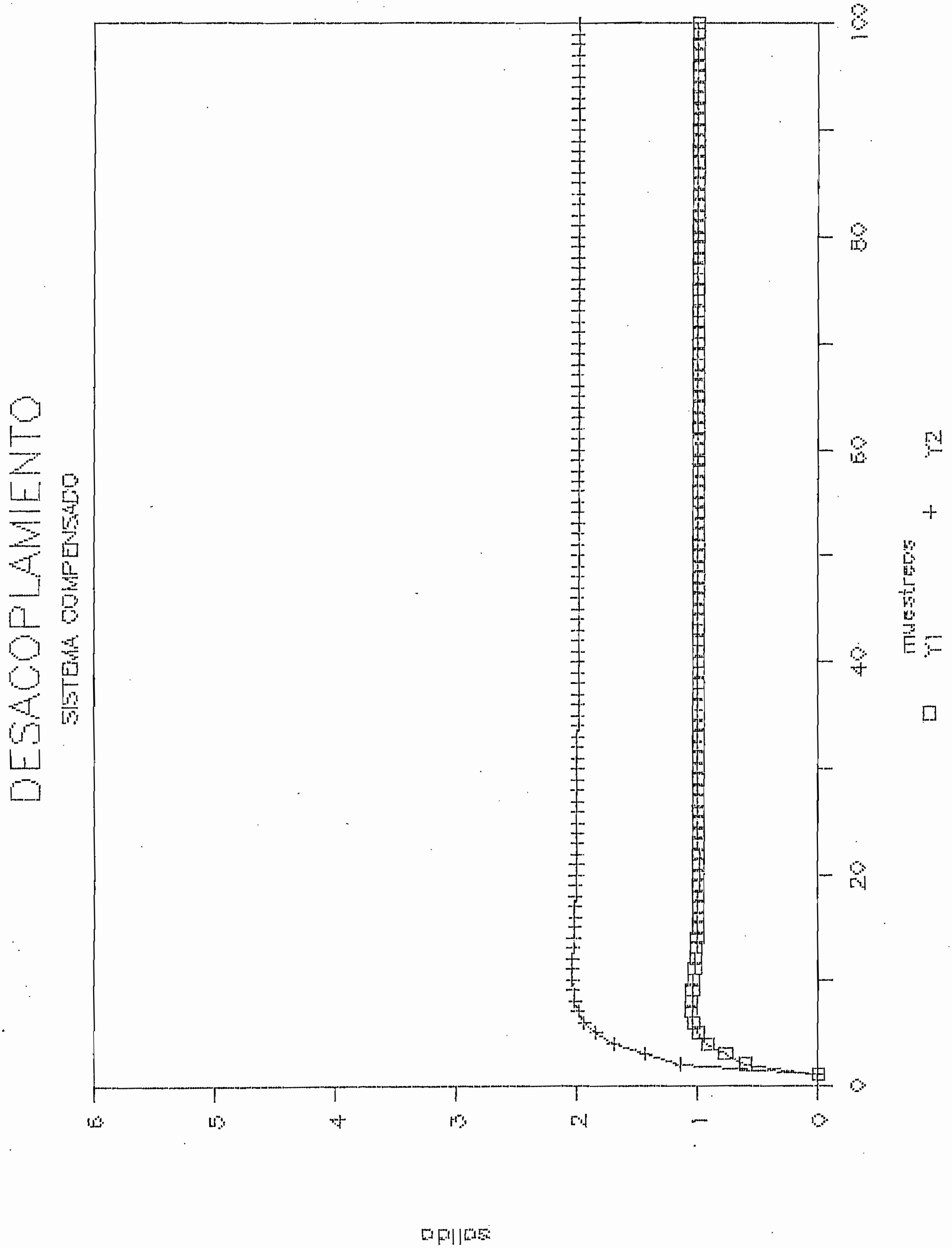


Fig. 3.2 SALIDA EN DESACOPLAMIENTO ($\gamma_1 = 1$, $\gamma_2 = 2$)

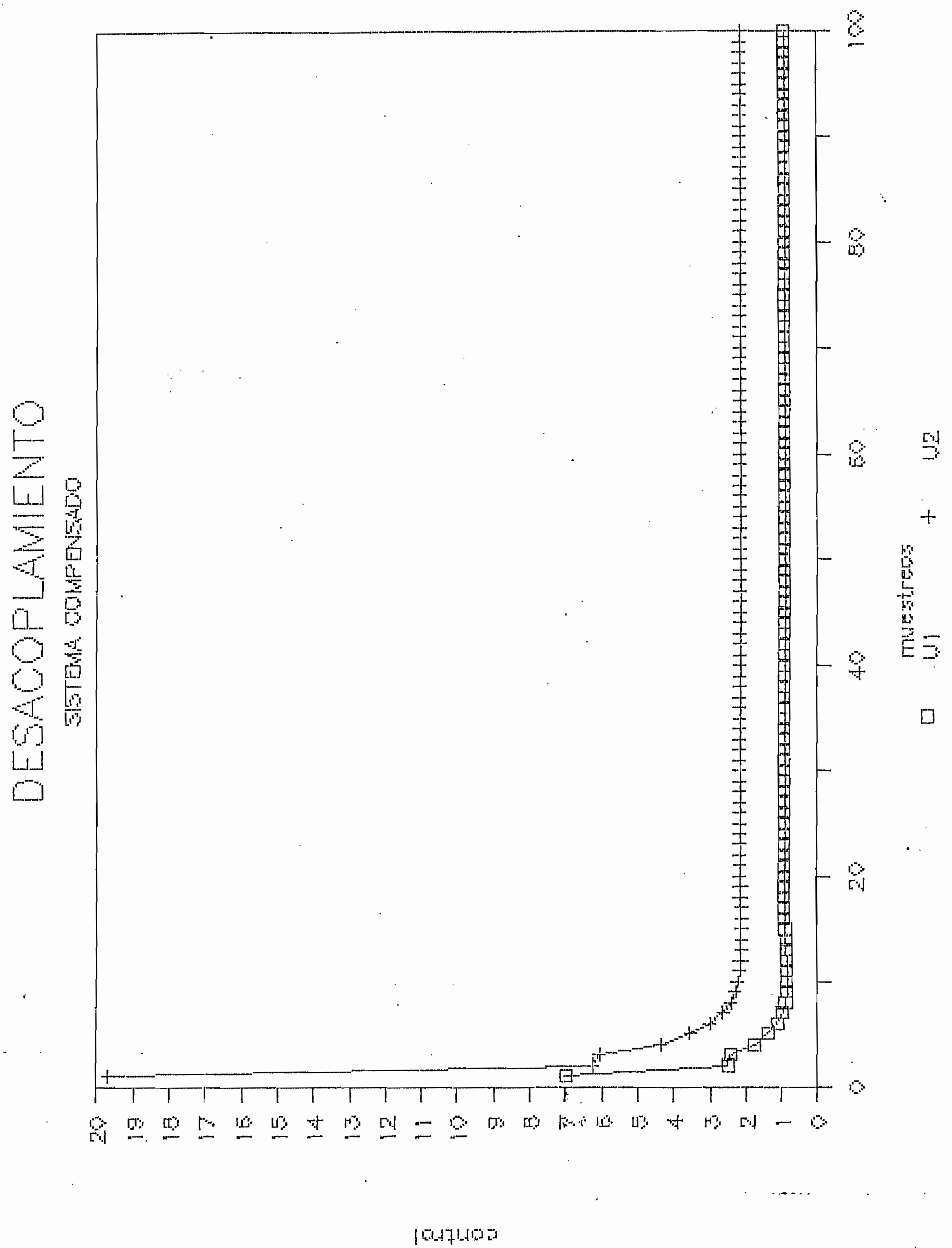


Fig. 3.3 CONTROLES EN DESACOPLAMIENTO ($\gamma_1 = 1$, $\gamma_2 = 2$)

Para tener clara la técnica de desacoplamiento, se ha procedido a cambiar una referencia del sistema desacoplado y controlado, utilizando los mismos compensadores; los resultados obtenidos se los presenta a continuación:

DESACOPLAMIENTO

SISTEMA ORIGINAL NO DESACOPLADO

Orden del sistema : 2
Numero de salidas : 2
Numero de entradas : 2

Matriz continua A

-0.11000	0.01000
0.01000	-0.07667

Matriz continua B

0.10000	0.00000
0.00000	0.06667

Matriz continua C

1.00000	0.00000
0.00000	1.00000

Matrices de asignación de polos Mk

Matriz M 0

-0.25000	0.00000
0.00000	-0.16000

MATRICES DE DESACOPLAMIENTO

Matriz de realimentación de estado F1

-1.40000	-0.10000
-0.15000	-1.25000

Matriz de preamplificación E

10.00000	0.00000
0.00000	15.00000

COMPENSACION

Compensador 1

Ganancia proporcional .... Kp= 0.50000
Ganancia integral .... Ki= 0.20000
Ganancia derivativa .... Kd= 0.10000

Compensador 2

Ganancia proporcional	Kp=	0.50000
Ganancia integral	Ki=	0.10000
Ganancia derivativa	Kd=	0.10000

DESACOPLAMIENTO .- Sistema controlado

Valores resultantes de la simulación:

Periodo de muestreo (s)	T= .9
Número de iteraciones	N= 100
Tiempo total del proceso (s)	t= 90

Valores de referencia utilizados
referencia 1 = 5
referencia 2 = 2

Valores de salida

Salida y 1

Valor inicial	Vo	0.00000
Valor final	Vf	5.00000
Valor máximo	Vmax	5.28380
Valor mínimo	Vmin	0.00000

Salida y 2

Valor inicial	Vo	0.00000
Valor final	Vf	2.00000
Valor máximo	Vmax	2.03807
Valor mínimo	Vmin	0.00000

Valores de las leyes de control

Ley de control u 1

Valor inicial	Vo	35.05556
Valor final	Vf	5.30000
Valor máximo	Vmax	35.05556
Valor mínimo	Vmin	4.91812

Ley de control u 2

Valor inicial	Vo	19.68333
Valor final	Vf	1.55000
Valor máximo	Vmax	19.68333
Valor mínimo	Vmin	1.50890

Valores de error del sistema desacoplado y compensado

Error e 1

Valor inicial	Vo	5.00000
Valor final	Vf	-0.00000
Valor máximo	Vmax	5.00000
Valor mínimo	Vmin	-0.28380

Error e 2

Valor inicial	Vo	2.00000
Valor final	Vf	-0.00000
Valor máximo	Vmax	2.00000
Valor mínimo	Vmin	-0.03807

Estos resultados pueden ser visualizados gráficamente en el gráfico que se presenta en la figura 3.4 el cual ha sido obtenido con la ayuda del programa LOTUS.

Del análisis de resultados presentados tanto numéricos como gráficos así como del propio uso del programa y su visualización completa en pantalla con una gran cantidad de opciones que permiten probar el sistema bajo diversas condiciones, se puede apreciar que la variación de una entrada solo afecta a la salida correspondiente, es decir que el desacoplamiento da efectivamente como resultado subsistemas univariables.

Por otro lado, se ha probado además que la variación de los parámetros de un compensador no afecta la salida no correspondiente, afianzándose aun más el criterio de desacoplamiento.

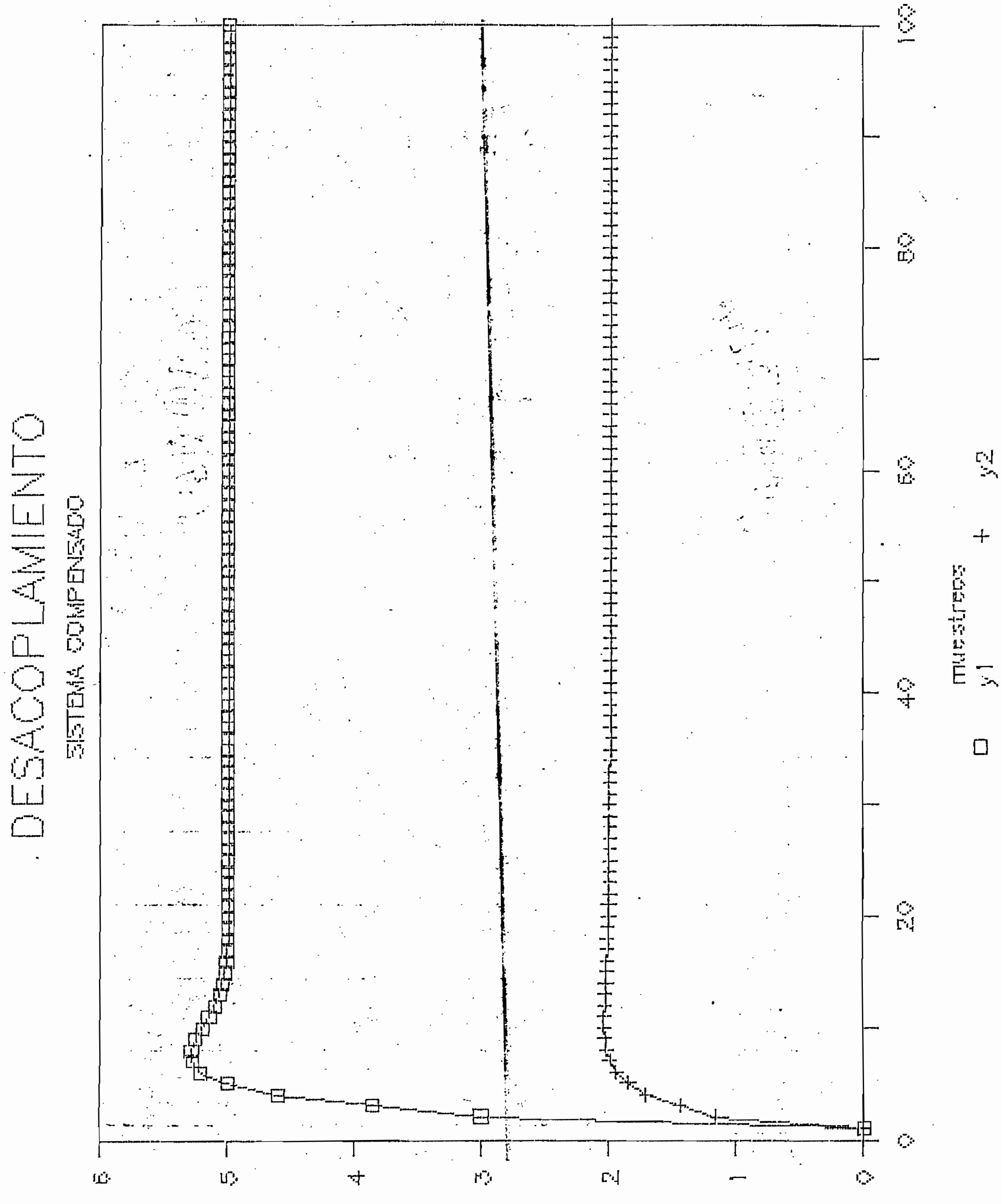


Fig. 3.4 SALIDAS EN DESACOPLAMIENTO (γ_1 = 5 , γ_2 = 2)

Por otro lado, se realizaron pruebas en cuanto a regulador cuadrático lineal se refiere y con la simulación igualmente del circuito de la figura 3.1 con lo que se obtienen los siguientes resultados:

REGULADOR CUADRATICO LINEAL

SISTEMA ORIGINAL NO REGULADO

Orden del sistema : 2
Numero de entradas : 2

Matriz continua A

-0.11000	0.01000
0.01000	-0.07667

Matriz continua B

0.10000	0.00000
0.00000	0.06667

MATRICES DE PONDERACION (CONTROL OPTIMO)

Matriz de ponderacion simetrica semidefinida (+) Q

50.00000	0.00000
0.00000	3.00000

Matriz de ponderacion M

0.00000	0.00000
0.00000	0.00000

Matriz de ponderacion simetrica definida (+) R

1.00000	0.00000
0.00000	1.00000

REGULADOR CUADRATICO LINEAL

Valores resultantes de la simulación:

Periodo de muestreo (s) .... T= .9
Número de iteraciones .... N= 150
Tiempo total del proceso (s) ...... t= 135

Matriz de ganancias de realimentación optima

4.37820	0.08488
0.04647	0.87416

```
Valores de offset utilizados
offset 1 = 2
offset 2 = 3

Valores de los estados

Estado        x 1
Valor inicial .................. Vo     0.00000
Valor final .................... Vf     0.36923
Valor máximo ................. Vmax     0.36923
Valor mínimo ................. Vmin     0.00000

Estado        x 2
Valor inicial .................. Vo     0.00000
Valor final .................... Vf     1.50098
Valor máximo ................. Vmax     1.50098
Valor mínimo ................. Vmin     0.00000

Valores de las leyes de control

Ley de control        u 1
Valor inicial .................. Vo     2.00000
Valor final .................... Vf     0.25605
Valor máximo ................. Vmax     2.00000
Valor mínimo ................. Vmin     0.25605

Ley de control        u 2
Valor inicial .................. Vo     3.00000
Valor final .................... Vf     1.67074
Valor máximo ................. Vmax     3.00000
Valor mínimo ................. Vmin     1.67074
```

Se ha obtenido resultados satisfactorios de regulación; las salidas no convergen a los valores de offset, lo cual es aceptable, pues se trata de un problema de regulación y no de seguimiento.

Los resultados presentados, pueden igualmente ser visualizados mediante los gráficos que se presentan en las figuras 3.5 y 3.6, los cuales han sido obtenidos mediante el programa LOTUS y en base al archivo de datos generado en el software desarrollado.

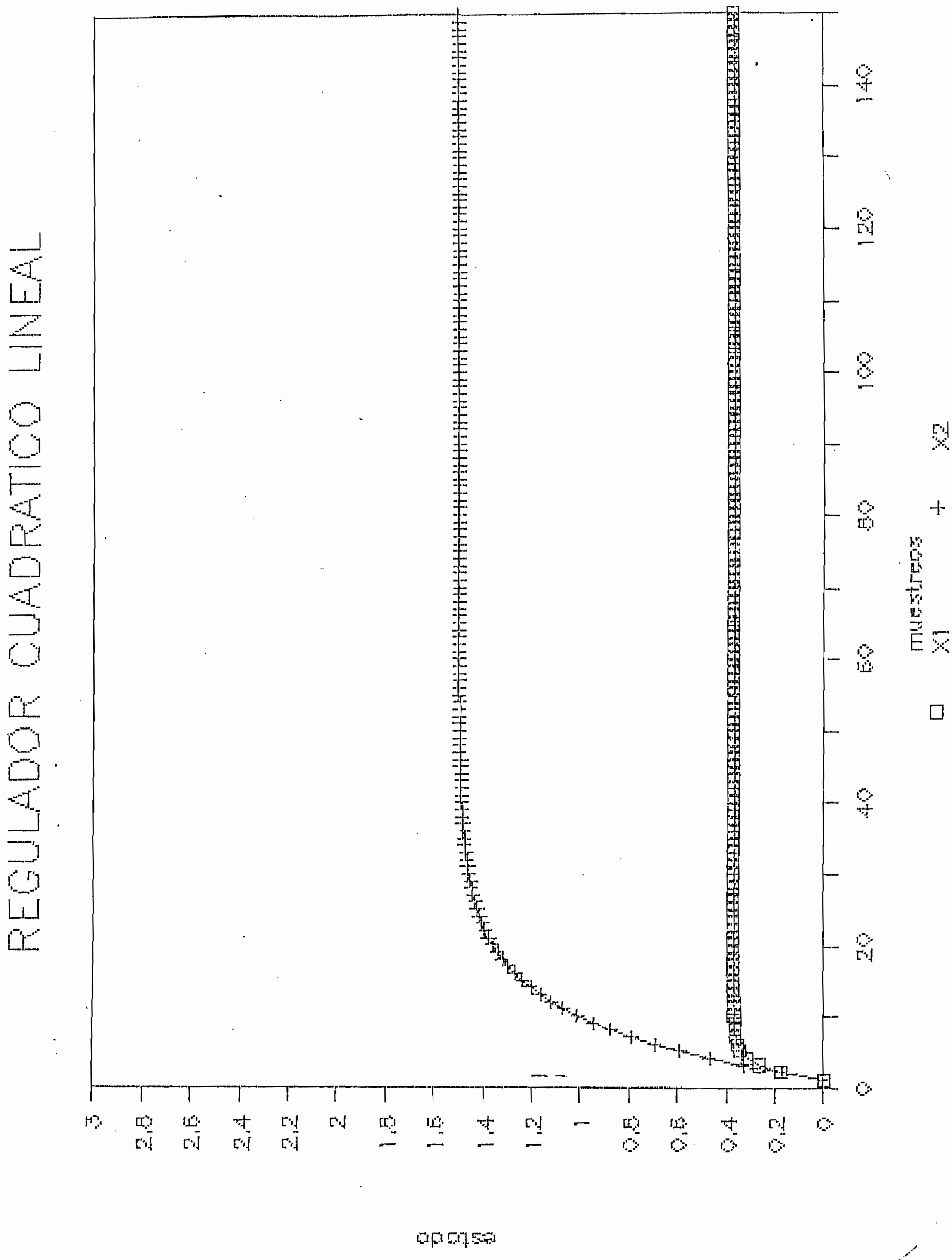


Fig. 3.5 SALIDAS EN REGULADOR CUADRATICO LINEAL (MAYOR PESO A LOS ESTADOS)

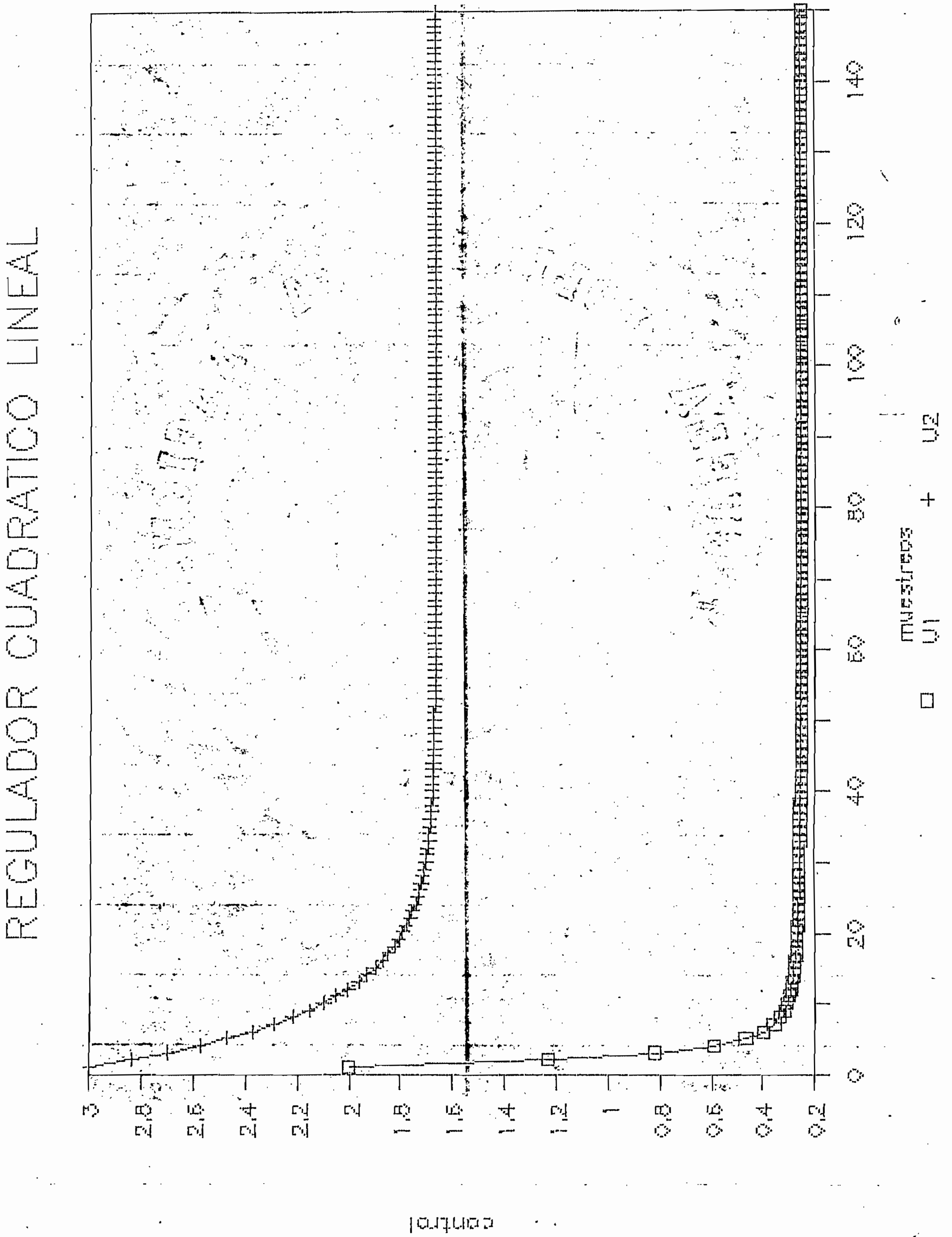


Fig. 3.6 CONTROLES EN REGULADOR CUADRATICO LINEAL
(MAYOR PESO A LOS ESTADOS)

Para tener una idea clara de la influencia de las matrices de ponderación en la respuesta del sistema se ha variado los pesos de ellas y a continuación se presenta otro de los resultados de los tantos realizados:

REGULADOR CUADRATICO LINEAL

SISTEMA ORIGINAL NO REGULADO

Orden del sistema : 2
Numero de entradas : 2

Matriz continua A

-0.11000	0.01000
0.01000	-0.07667

Matriz continua B

0.10000	0.00000
0.00000	0.06667

MATRICES DE PONDERACION (CONTROL OPTIMO)

Matriz de ponderacion simetrica semidefinida (+) Q

1.00000	0.00000
0.00000	1.00000

Matriz de ponderacion M

0.00000	0.00000
0.00000	0.00000

Matriz de ponderacion simetrica definida (+) R

50.00000	0.00000
0.00000	3.00000

REGULADOR CUADRATICO LINEAL

Valores resultantes de la simulación:

Periodo de muestreo (s) .... T= .9
Número de iteraciones .... N= 150
Tiempo total del proceso (s) ...... t= 135

Matriz de ganancias de realimentación optima

0.00870	0.00110
0.01215	0.13293

Valores de offset utilizados
offset 1 = 2
offset 2 = 3

Valores de los estados

Estado x 1		
Valor inicial	Vo	0.00000
Valor final	Vf	2.03198
Valor máximo	Vmax	2.03198
Valor mínimo	Vmin	0.00000
Estado x 2		
Valor inicial	Vo	0.00000
Valor final	Vf	2.55669
Valor máximo	Vmax	2.55669
Valor mínimo	Vmin	0.00000

Valores de las leyes de control

Ley de control u 1		
Valor inicial	Vo	2.00000
Valor final	Vf	1.97953
Valor máximo	Vmax	2.00000
Valor mínimo	Vmin	1.97953
Ley de control u 2		
Valor inicial	Vo	3.00000
Valor final	Vf	2.63545
Valor máximo	Vmax	3.00000
Valor mínimo	Vmin	2.63545

Los resultados obtenidos se pueden visualizar gráficamente en el gráfico de la figura 3.7. Del análisis de los resultados aquí presentados como de muchos resultados ensayados se llega a la conclusión que altos valores en la matriz Q y bajos valores en la matriz R, es decir un mayor peso a los estados, determina que los estados se estabilicen rápidamente pero difieren en gran porcentaje con el offset de entrada.

Por el contrario bajos valores de Q y altos valores de R,

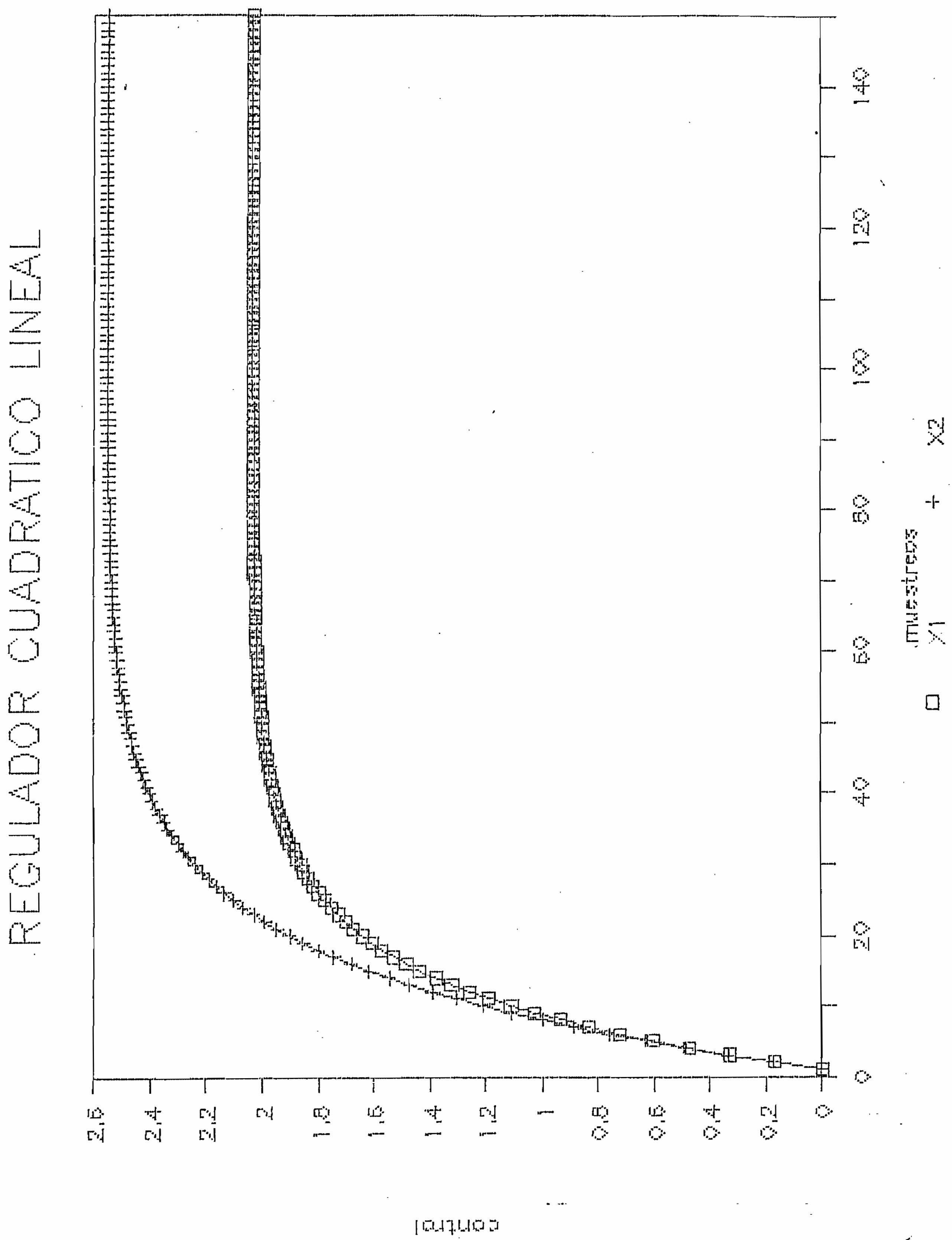


Fig. 3.7 SALIDAS EN REGULADOR CUADRATICO LINAL
(MAYOR PESO A LOS CONTROLES)

es decir, un mayor peso al control, determina que los estados tiendan al offset pero con menor rapidez. Se puede por lo tanto concluir que este caso se asemeja a un problema de seguimiento.

En forma general, se debe tener un compromiso entre los pesos que se asignan a cada una de las matrices de ponderación de tal modo que se obtengan resultados aceptables de acuerdo a los requerimientos.

3.2.- RESULTADOS DE CONTROL EN TIEMPO REAL

Para obtener los diversos resultados provenientes de la ejecución de los algoritmos desarrollados en tiempo real se ha utilizado el circuito de la figura 3.1, el cual ha sido implementado en forma física mediante la disposición de condensadores y resistencias y de acuerdo a los valores establecidos al principio del presenta capítulo.

Para el ingreso de los parámetros necesarios para las diversas técnicas de control se ha tomado como base los resultados provenientes de la simulación, obteniendose de este modo en forma general resultados satisfactorios.

Es importante mencionar que los gráficos que se presentan en este numeral tienen una escala vertical de un voltio por división, mientras que la escala horizontal es de seis segundos por división.

Inicialmente se realizaron las pruebas mediante la técnica de control **P.I.D. multilazo**, despreciendo de esta manera la interacción que se presenta en el sistema. Primeramente se muestra en la figura 3.8 las salidas y_1 y y_2 del sistema compensado mediante los parámetros de los compensadores obtenidos previamente en simulación; en este caso se ha utilizado como referencias los valores de 1 y 2 voltios.

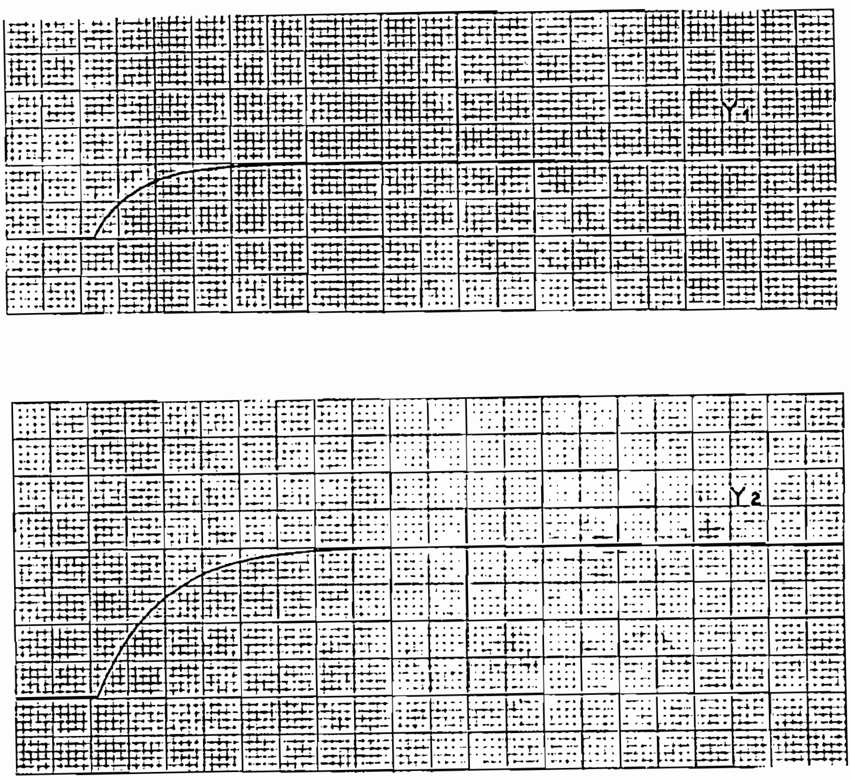


Fig. 3.8 Salidas del sistema .- P.I.D. multilazo

Luego se presentan los gráficos obtenidos de los controles que determinan las salidas de la figura 3.8. Los controles indicados se presentan en la figura 3.9.

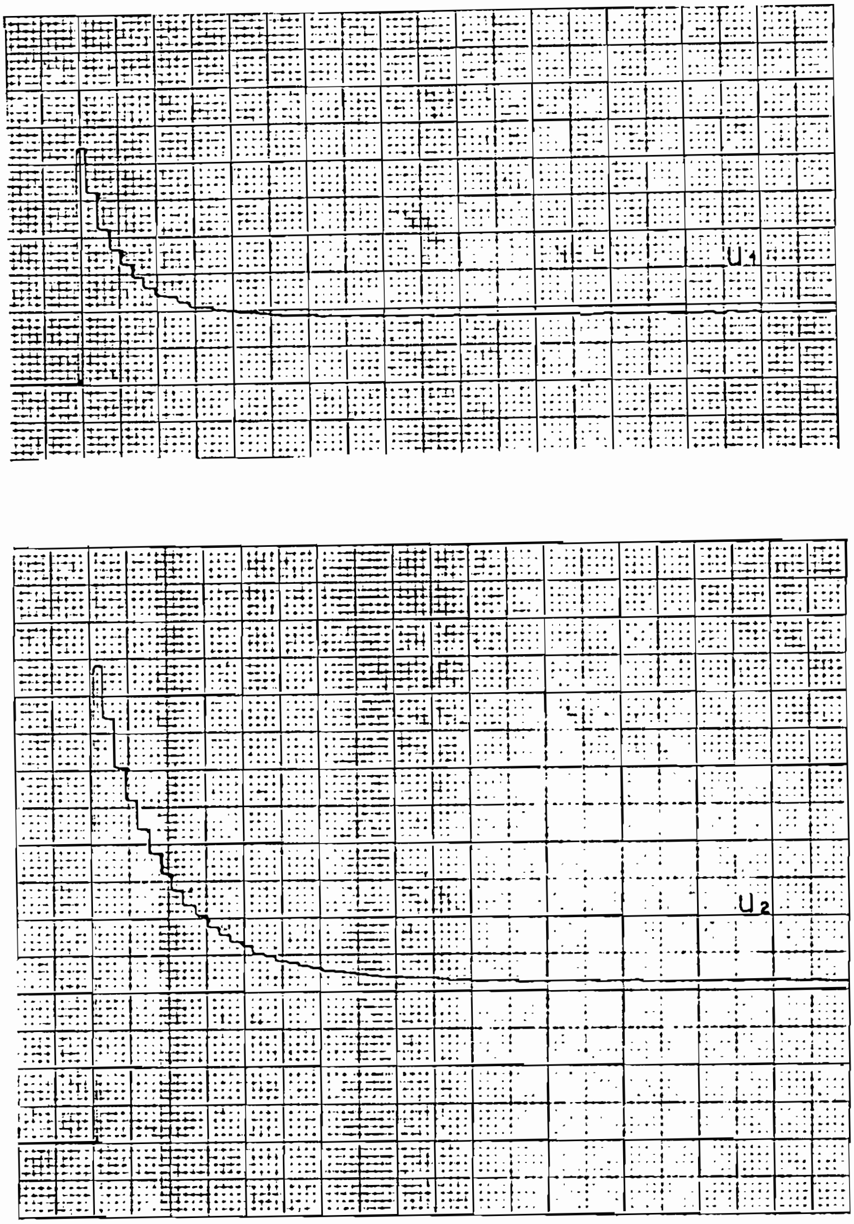


Fig. 3.9 Controles del sistema .- P.I.D. multilazo

Luego se ha realizado el cambio de las referencias del sistema a valores de 5 y 2 voltios, las variaciones que de este modo se obtienen en las salidas del sistema y_1 y y_2 pueden ser visualizadas en la figura 3.10.

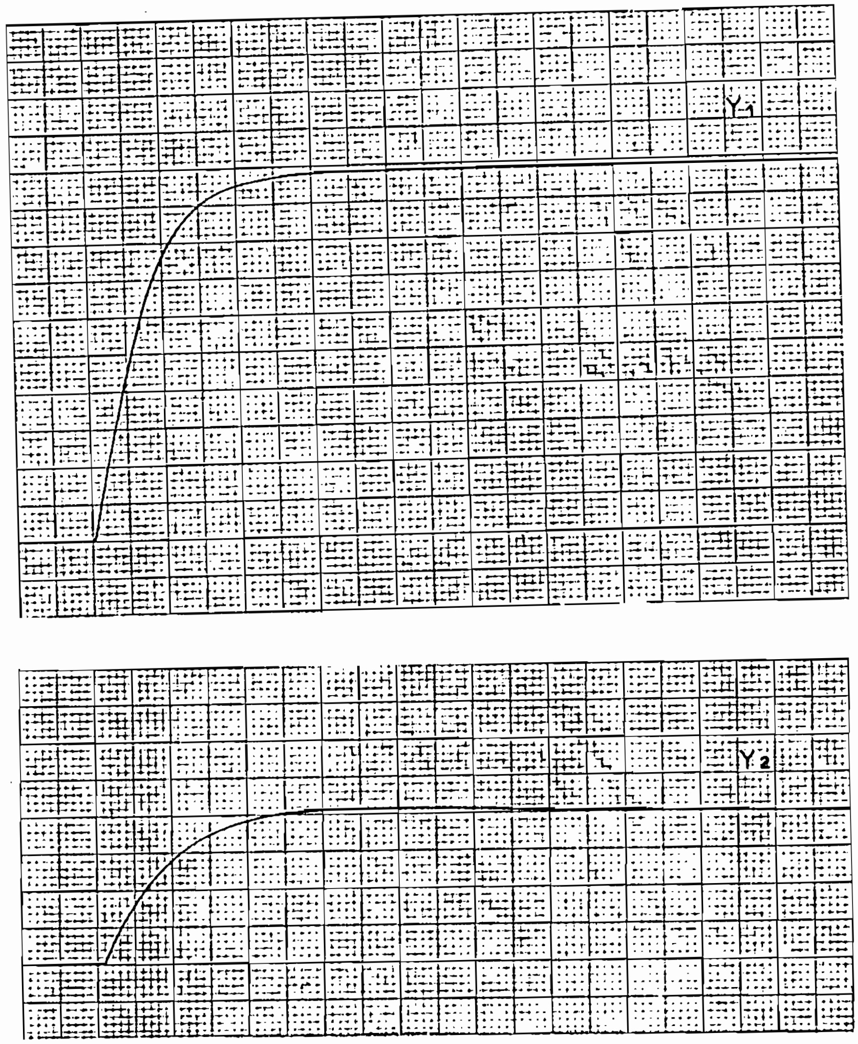


Fig. 3.10 Salidas del sistema .- P.I.D. multilazo

Realizando una comparación entre la salida y_2 tanto en la figura 3.8 como en la figura 3.10, se puede apreciar que la variación de la referencia uno de 1 a 5 voltios afecta a la salida y_2, pues se puede observar que en el segundo caso la salida indicada presenta un pequeño sobreimpulso, el cual no se aprecia en el primer caso.

Considerando las salidas de la figura 3.10, se procedió finalmente a realizar una perturbación fuerte en la salida y_1, como se puede observar en la figura 3.11 la respuesta y_2 se ve claramente afectada.

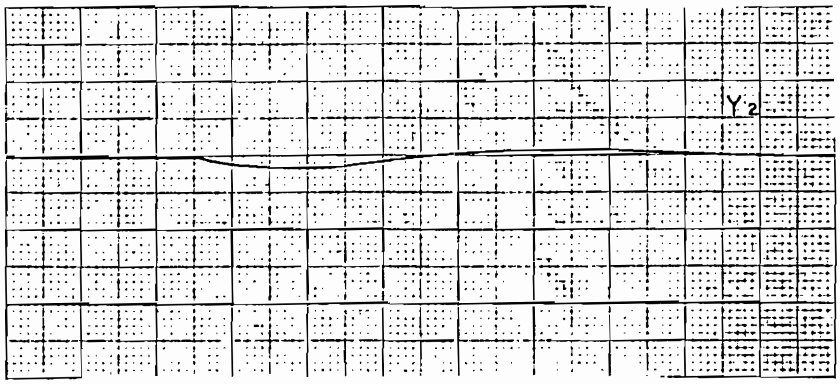


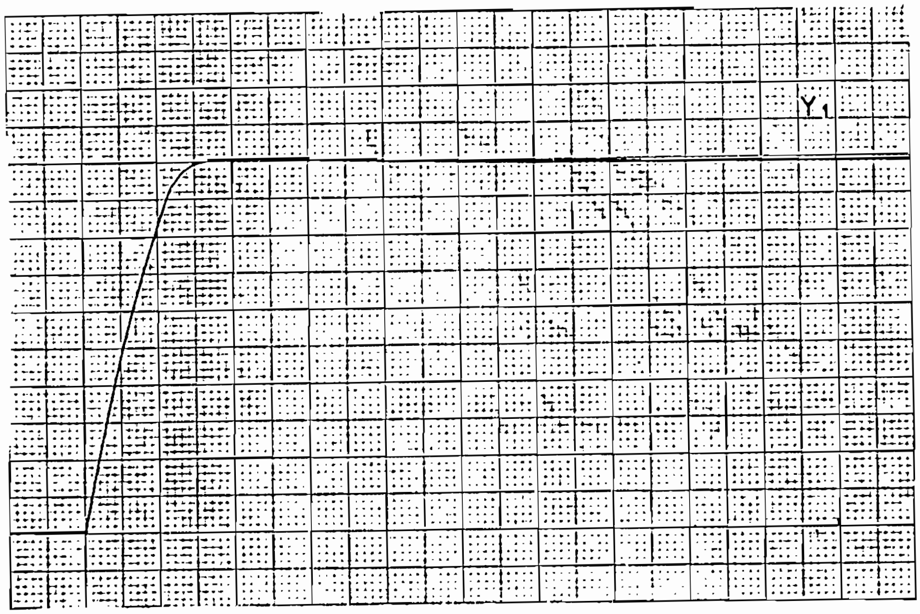
Fig. 3.11 Salida y_2 .- respuesta ante perturbación en y_1

Del análisis de las figuras presentadas en cuanto a la técnica P.I.D. multilazo se refiere se puede apreciar que este tipo de control resulta satisfactorio para sistemas con débil acoplamiento.

Posteriormente se procedió a realizar pruebas en tiempo real utilizando la técnica de **desacoplamiento**, de esta forma se presenta en la figura 3.12 las respuestas de las salidas del sistema desacoplado y compensado mediante compensadores tipo P.I.D. y utilizando las referencias de 5 y 2 voltios para el control. Los compensadores utilizados han sido ajustados en base a los resultado de simulación y se han utilizado los siguientes valores que determinan respuestas satisfactorias:

Subsistema 1 .- K_P = 0.5 ; K_I = 0.055 ; K_D = 0.1

Subsistema 2 .- K_P = 0.5 ; K_I = 0.05 ; K_D = 0.1



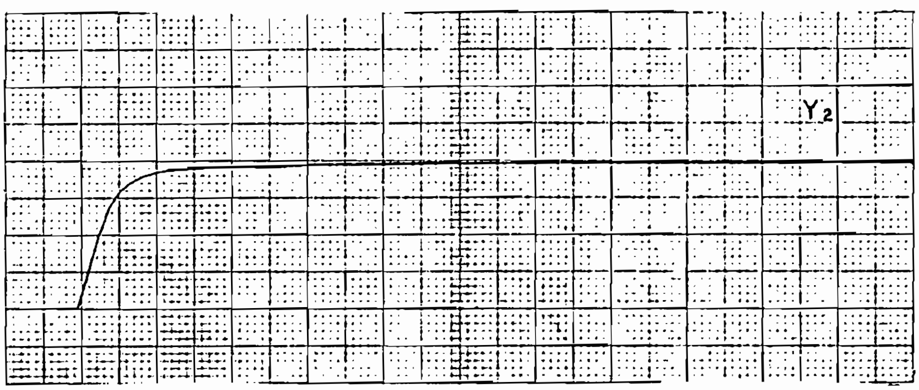


Fig. 3.12 Salidas del sistema desacoplado y compensado

Adicionalmente se han realizado pruebas que determinan que la variación en una referencia del sistema no afecta la salida no correspondiente del sistema desacoplado; por otro lado se ha perturbado fuertemente la salida y_1 observándose que la salida y_2 practicamente no se encuentra afectada; lo indicado puede apreciarse en la figura 3.13.

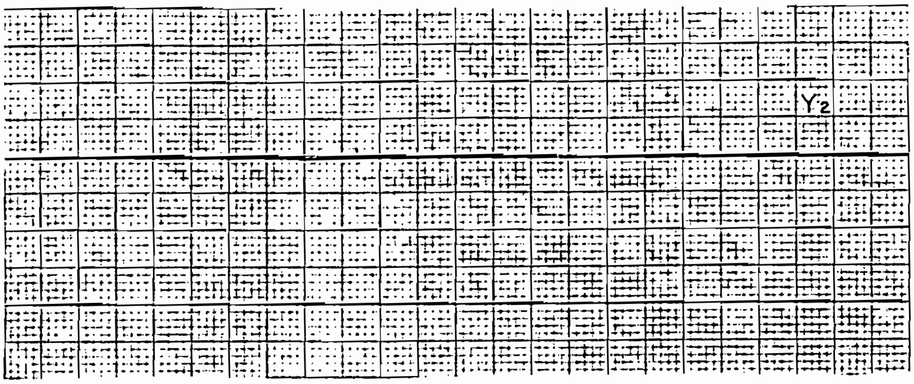
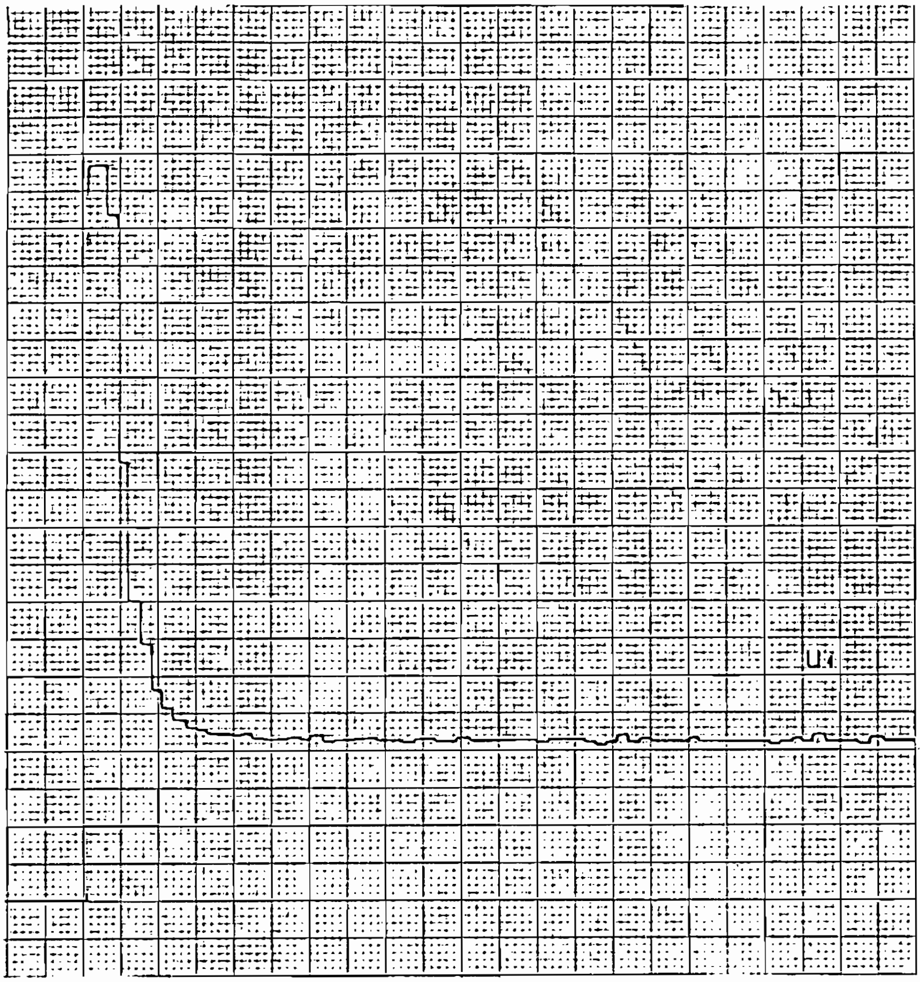


Fig. 3.13 Salida y_2 .- respuesta ante perturbación en y_1

De este modo se puede afianzar el conocimiento de la técnica de desacoplamiento, pues se observa la existenacia de dos subsistemas independientes, ya que la variación de una referencia, el cambio de un compensador y una perturbación fuerte no afecta a la salida no correspondiente.

Finalmente en lo que a desacoplamiento se refiere sa han obtenido los gráficos de los controles que determinan que el sistema desacoplado de como respuesta las salidas de la figura 3.12. Los controles en mención, u_1 y u_2 se los presenta en la figura 3.14.



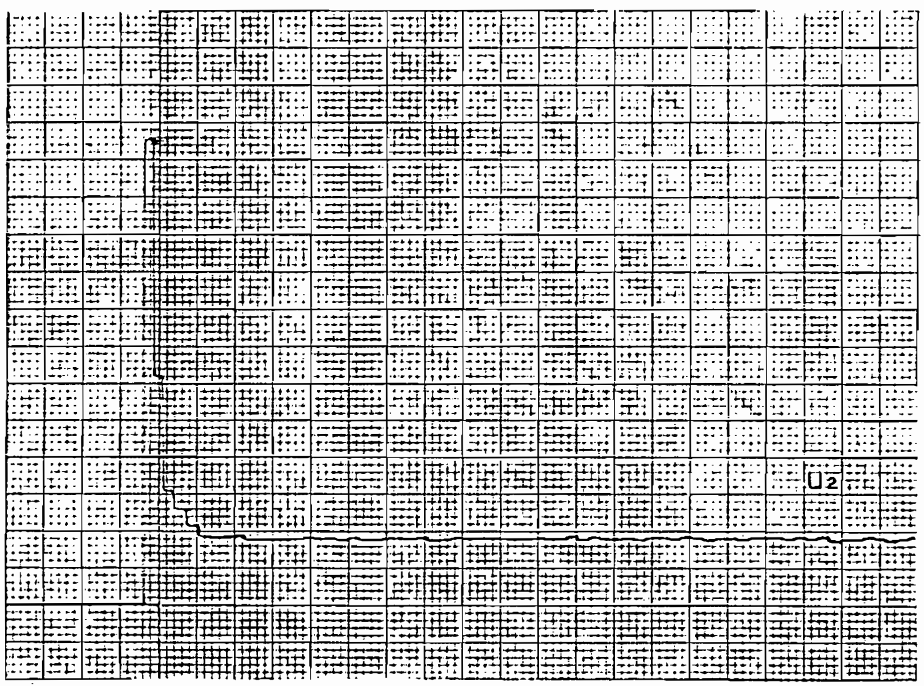


Fig. 3.14 Controles del sistema desacoplado

La última técnica utilizada en tiempo real en el presente tema de tesis es la de **regulador cuadrático lineal**; para esto se han utilizado los resultados provenientes de simulación en sus dos versiones; la primera en la que se da un alto peso a los estados y bajo peso al control, en esta caso se puede observar las salidas del sistema y_1 y y_2, las mismas que se muestran en la figura 3.15; y, con la utilización de 2 y 3 voltios como valores de offset.

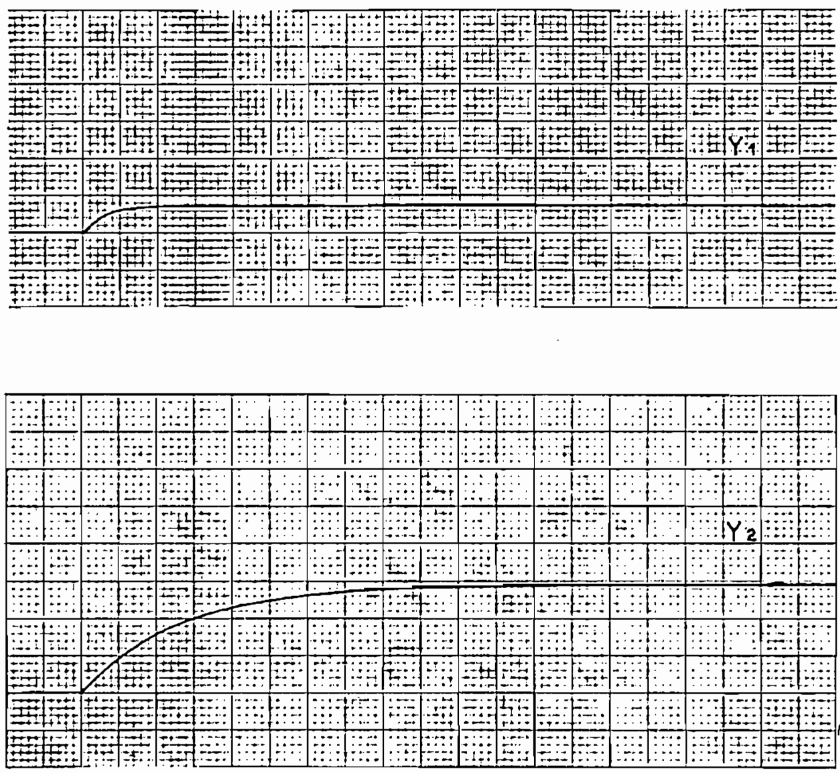


Fig.3.15 Salidas del sistema regulado .- regulador cuadrático lineal

Las salidas del sistema regulado mostradas en la figura 3.15 han sido obtenidas mediante los controles u_1 y u_2 que se muestran en la figura 3.16.

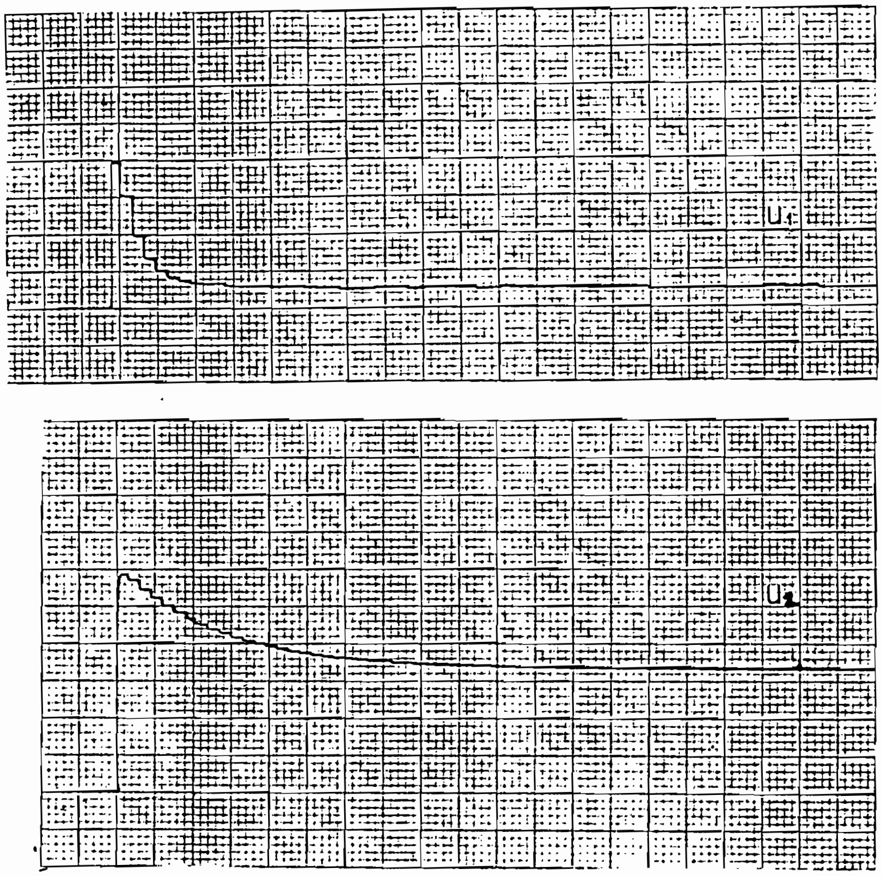


Fig. 3.16 Controles del sistema regulado .- regulador cuadrático lineal.

Posteriormente, se procedió a realizar una fuerte perturbación en la salida y_1 observándose que la salida y_2 no ha sido afectada, teniéndose de este modo un buen regulador; lo indicado puede apreciarse en la figura 3.17.

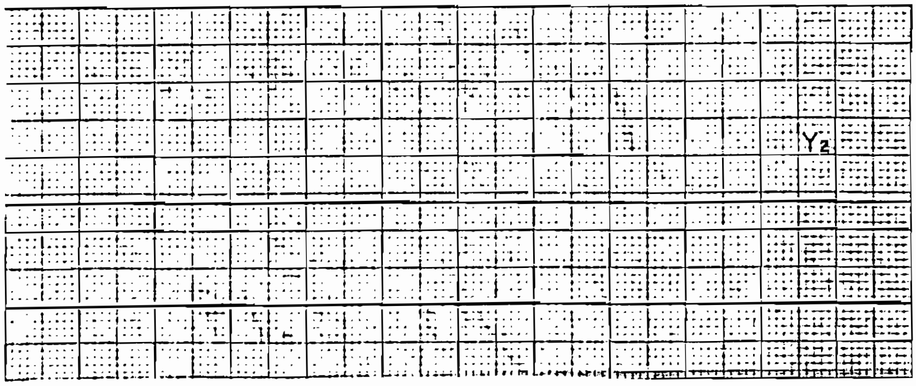
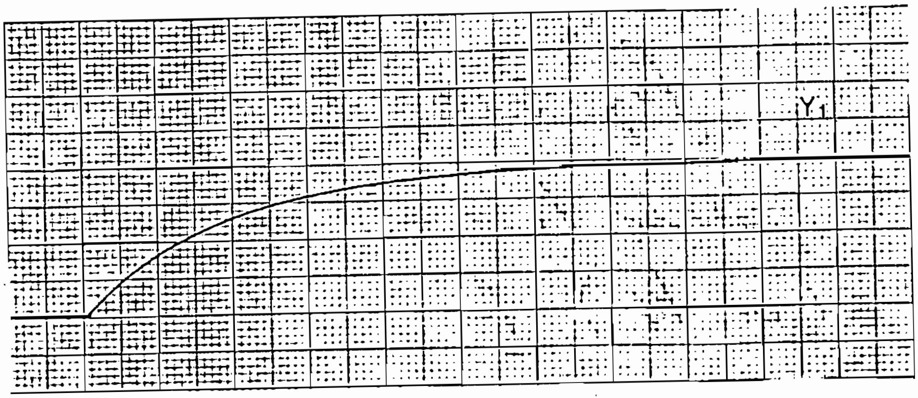


Fig. 3.17 Salida y_2 .- respuesta ante perturbación en y_1

Posteriormente, se repitió el proceso, para el caso en que se tiene bajo peso en los estados y alto peso en el control con los resultados de simulación; con los mismos valores de offset (2 y 3 voltios), se obtienen las respuestas de las salidas que se presentan en la figura 3.18.

Se puede apreciar entonces que en este caso se tiende al valor de offset pero con un comportamiento más lento que en el caso anterior en el que se tiene un comportamiento rápido pero se difiere del valor de offset en gran porcentaje. Lo indicado puede apreciarse comparando las figuras 3.15 y 3.18.

Por lo tanto se debe tomar en cuenta los requerimientos que de desee en el sistema para de este modo dar una mayor ponderación ya sea a los estados o al control, teniéndose un compromiso entre ellos.



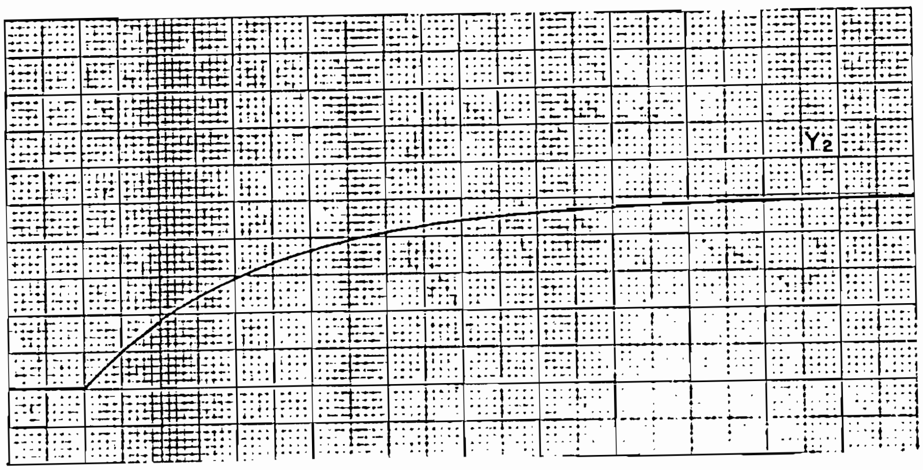
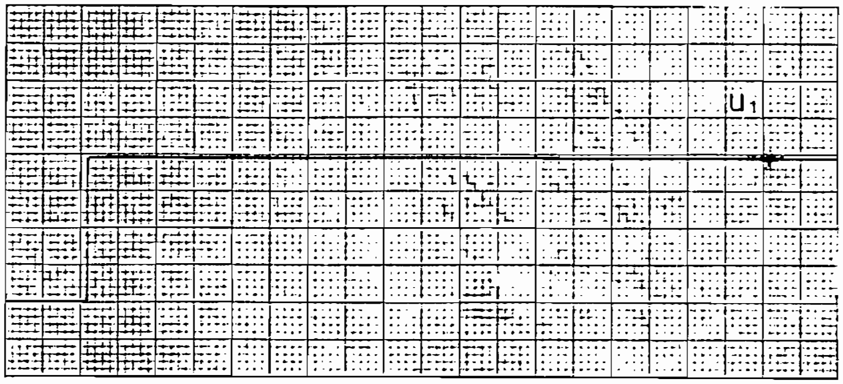


Fig. 3.18 Salidas del sistema regulado .- regulador cuadrático lineal

Adicionalmente, se presenta en la figura 3.19 los controles que determinan la obtención de las salidas de la figura 3.18.

Por otro lado, en este caso, se puede apreciar el cambio de la respuesta en la salida y_1 al realizar una fuerte perturbación en la salida y_2;lo indicado puede observarse en la figura 3.20, y si se compara con la figura 3.17 se puede concluir que el caso correspondiente a la figura 3.17 constituye un mejor regulador.



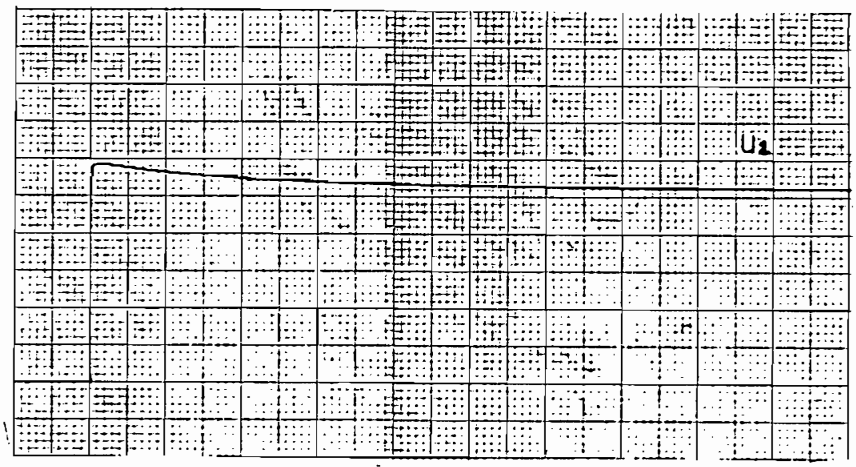


Fig. 3.19 Controles del sistema regulado .- regulador cuadrático lineal

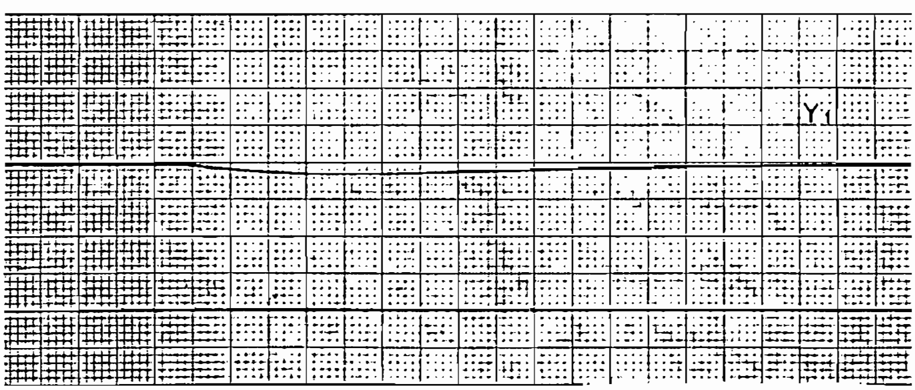


Fig. 3.20 Salida y_1 .- respuesta ante perturbación en y_2

Finalmente en lo que a regulador cuadrático lineal se refiere se realizó un cambio en el valor de offset correspondiente a la entrada uno del sistema con lo que se puede apreciar que en el caso de tener altos valores en la matriz de ponderación Q y bajos valores en la matriz de ponderación R, practicamente no se afecta a la salida y_2; esto puede observarse en la figura 3.21.

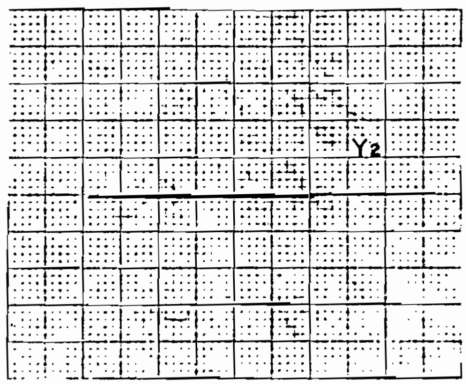


Fig. 3.21 Salida y_2 .- variación de offset 1

Por otro lado si se realiza el mismo procedimiento en el caso de tener bajos valores en la matriz de ponderación Q y altos valores en la matriz de ponderación R, si se produce una variación significativa en la salida y_2 al variar el valor de offset 1, como puede apreciarse en la figura 3.22.

Finalmente, de lo expuesto, se concluye que se dispone de una regulación óptima en al caso de tener altos valores de la matriz de ponderación Q y bajos valores de la matriz de ponderación R, con la salvedad de diferir de los valores de offset en un gran porcentaje; sin embargo este inconveniente puede superarse incrementando el offset utilizado hasta conseguir que la salida adquiera el valor estable deseado.

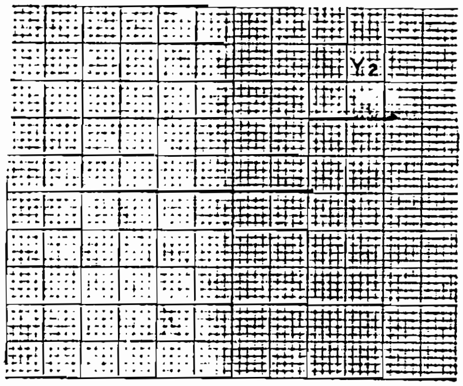


Fig. 3.22 Salida y_2 .- variación de offset 1

3.3.- RESULTADOS DE SIMULACION DEL PROGRAMA CC

En lo que a resultados del programa CC se refiere, primeramente se debe indicar que se ha trabajado con los valores y el circuito de la figura 3.1.

Se ha chequeado controlabilidad y observabilidad determinandose que el sistema es completamente controlable y observable; por otro lado se tiene la matriz función de transferencia del sistema modelado en variables de estado. La figura 3.23 presenta el resultado gráfico de regulador cuadrático lineal con alta ponderación de la matriz R; y, finalmente se tiene la figura 3.24 en la que se grafica el error $\tilde{x} = x - \hat{x}$, obtenido del filtro de Kalman.

$$H1(1,1) = \frac{.1s + 7.666667E-03}{s^2 + .1866667s + 8.333334E-03}$$

$$H1(1,2) = \frac{6.666667E-04}{s^2 + .1866667s + 8.333334E-03}$$

$$H1(2,1) = \frac{.001}{s^2 + .1866667s + 8.333334E-03}$$

$$H1(2,2) = \frac{6.666667E-02s + 7.333333E-03}{s^2 + .1866667s + 8.333334E-03}$$

```
STATE>pause
Hit any key to continue ♦
```

```
MIMO>FACTORES
MIMO>hdisplay,h3
 H3, #rows= 2 #columns= 2
 Hc is common multiplier of all elements


            1
 Hc = -------------------------------------
      ( s + .1127698)( s + 7.389685E-02)


 H3(1,1) =  .1( s + 7.666666E-02)


 H3(1,2) =  6.666667E-04


 H3(2,1) =  .001


 H3(2,2) =  6.666667E-02( s + .11)

MIMO>pause
Hit any key to continue
```

```
CC>DETERMINANTE DE MATRIZ FUNCION DE TRANSFERENCIA
CC>display,g1
```

$$G1(s) = \frac{6.666667E-03}{s^2 + .1866667s + 8.333334E-03}$$

```
CC>DETERMINANTE EN FACTORES
CC>display,g2
```

$$G2(s) = \frac{6.666667E-03}{(s + .1127699)(s + 7.389678E-02)}$$

```
CC>pause
Hit any key to continue
```

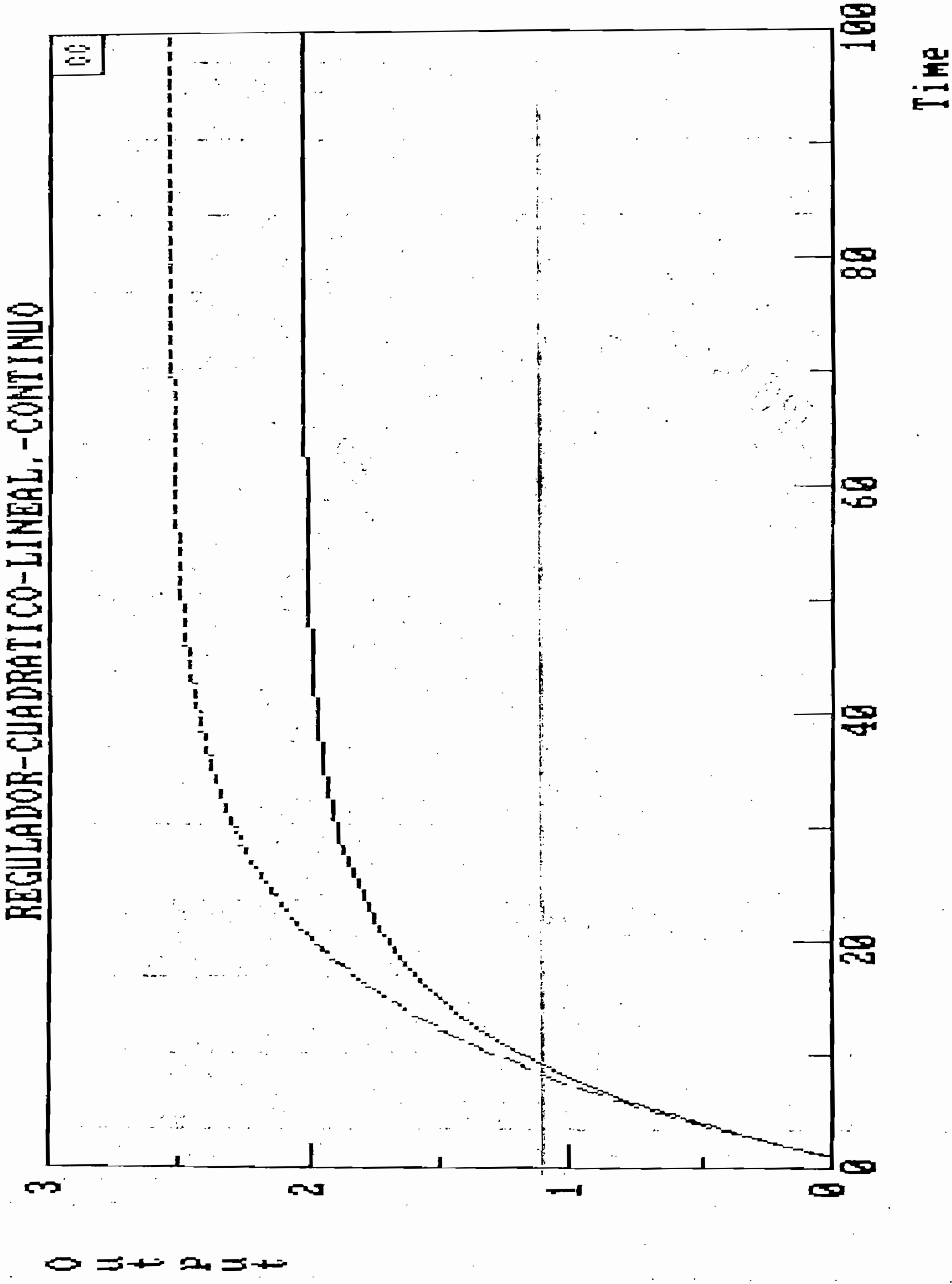


Fig. 3.23 REGULADOR CUADRATICO LINEAL .- PROGRAMA CC

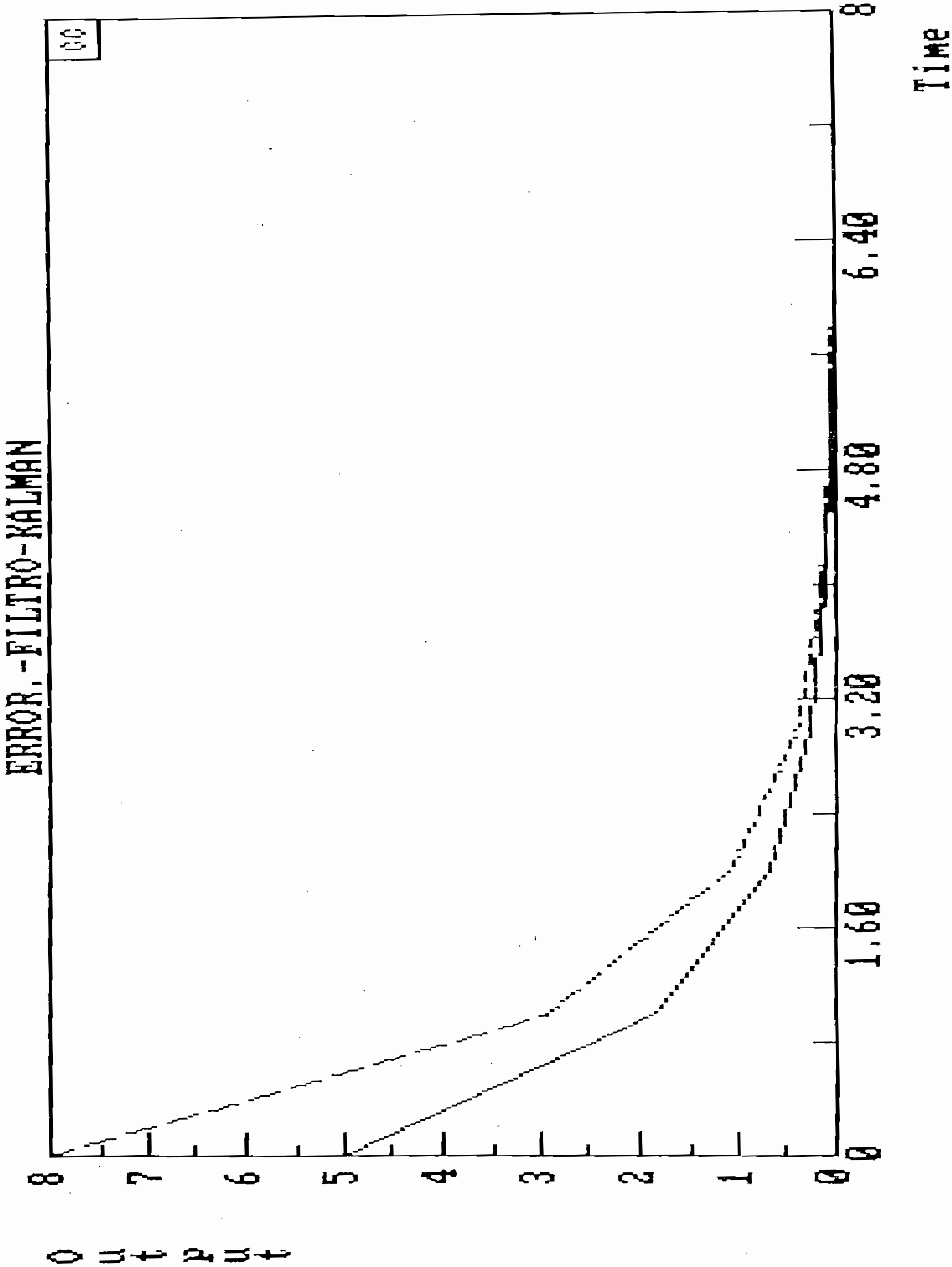


Fig. 3.24 FILTRO DE KALMAN.- PROGRAMA CC

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

4.2 RECOMENDACIONES

4.- CONCLUSIONES Y RECOMENDACIONES

4.1.- CONCLUSIONES

El programa desarrollado consta de dos partes perfectamente definidas: simulación y tiempo real; en cuanto a simulación el software en cuestión ofrece una gran versatilidad en la elección de los diversos tópicos y variación de condiciones de operación tales como referencias, valores iniciales, número de iteraciones, elección de compensadores e ingreso de sus parámetros, elección de gráficos a ser visualizados, despliege de resultados numéricos en pantalla, resultados completos en papel, generación de archivo de datos en formato DOS para una posterior graficación en papel, etc.

Los resultados de simulación son utilizados en forma eficiente en lo que a tiempo real se refiere y los resultados obtenidos tanto a nivel de simulación como en la implementación en tiempo real son enteramente satisfactorios.

El software, objeto del presente tema de tesis, ha sido desarrollado en un computador IBM, PS modelo 60 con monitor a color y en el lenguaje de programación QUICK BASIC, vesión 4.5; para trabajar en tiempo real se ha utilizado el equipo de adquisición de datos y control KEITHLEY 500A y el software específico correspondiente para su manejo QUICK

500. Todo el soporte utilizado tanto a nivel de hardware como de software presenta una gran versatilidad para este tipo de trabajos en tiempo real.

Debido a que la presente tesis "Control Multivariable en Tiempo Real" tiene una orientación pedagógica, se ha enfatizado en la presentación gráfica de las diversas variables del sistema, tanto en simulación como en tiempo real. Es por esto que en lo que a tiempo real se refiere, para especificar el periodo mínimo de muestreo que puede utilizarse se debió considerar a más del tiempo de duración de cada algoritmo, el tiempo que se require para la presentación gráfica punto a punto de las variables reales del sistema, es por esto que el mínimo periodo de muestreo es de 150 ms. Por otro lado debe indicarse que se podría trabajar con periodos de muetreo menores al indicado si se suprime la opción gráfica o bién si se explota mayormente el software de tiempo real, pero para el último caso se require de una tarjeta especial de la cual, actualmente no dispone el equipo de adquisición de datos y control con el que se ha trabajado.

La utilización de la técnica de control multilazo tipo P.I.D. da muy buenos resultados cuando se trabaja con sistemas multivariables con débil interacción; sin embargo su utilización no resulta adecuada en sistemas que presentan una interacción fuerte, en este caso, la técnica de desacoplamiento permite obtener subsistemas independientes en los cuales se puede ensayar la compensación tipo P.I.D. para lazos univariables independientes, de tal modo que se obtienen excelentes resultados principalmente en cuanto a perturbaciones.

Existen diversas técnicas de desacoplamiento, entre las cuales las más utilizadas son: por realimentación de estado y por realimentación de salida; en el presente trabajo de tesis se ha optado por realizar desacoplamiento por realimentación de estado y en esta técnica se puede asignar los polos del sistema desacoplado que permite definir la dinámica del sistema de acuerdo a los requerimientos deseados para el nuevo sistema o conjunto de subsistemas univariables.

Las técnicas multivariables de control y compensación tipo P.I.D. constituyen un problema de seguimiento. En la presente tesis se ha incluido un problema de regulación óptima en base a un criterio de función de costo cuadrático, por lo que se utiliza la técnica del regulador cuadrático lineal, en el cual, partiendo de una condición inicial de los estados, estos obtienen un valor final cero de acuerdo a un criterio óptimo. En este trabajo se ha realizado una variación a lo indicado al utilizar un offset que se suma al control calculado mediante un criterio óptimo.

En esta técnica la ponderación que se asigna a las diversas variables del sistema esta dada en base a tres matrices de ponderación: Q, M y R. La matriz M pondera la relación entre los estados y controles por lo que se utiliza como matriz igual a cero a fin de disminuir la interacción entrada-salida. La matriz Q pondera los estados del sistema y la matriz R pondera los controles del mismo.

Cuando la ponderación especificada por Q es alta, mientras que la ponderación especificada por R es baja se obtiene un excelente regulador óptimo, que permite que los estados adquieran su estado estable rápidamente y se disponga de un alto rechazo a las perturbaciones; sin embargo se tiene la desventaja que el valor estable de los estados difiere en gran porcentaje del offset especificado.

Por el contrario, si la ponderación asignada por Q es baja, mientras la ponderación de R es alta se puede obtener el valor estable de los estados igual al valor de offset especificado; sin embargo, se adquiere el estado estable con relativa lentitud y no se tiene un buen rechazo a las perturbaciones; se podría decir que este caso se asemeja a un problema de seguimiento.

Finalmente cabe indicar que de lo que se conoce se ha realizado por primera vez en nuestro medio un control digital con sistemas multivariables, lo cual resulta interesante toda vez que los resultados obtenidos han sido enteramente satisfactorios.

4.2.- RECOMENDACIONES

Una vez que se ha realizado control multivariable en tiempo real en base a los algoritmos implementados en la presente tesis, resultaría interesante desarrollar nuevos algoritmos de índole multivariable con el propósito de ser implementados en sistemas físicos reales, los mismos que podrían tener una utilidad talvés de tipo industrial.

Entre otros algoritmos se podría pensar en realizar por ejemplo desacoplamiento por realimentación de salida, observador óptimo de Kalman, técnicas en el dominio de la frecuencia, etc.

Dado que el software QUICK 500 es un software especializado para trabajar con el equipo de adquisición de datos y control KEITHLEY 500A, y en vista que dispone de muchas rutinas para tiempo real resultaría interesante desarrollar un estudio completo de las potencialidades del software especificado y sus posibles aplicaciones en los diversos campos de la ingeniería de control.

Desde el pundo de vista pedagógico sería recomendable utilizar los programas desarrollados a manera de prácticas que permitirían afianzar los conocimientos y contactar la realidad de la utilidad de la ingeniería de control en lo que se refiere a control moderno, control óptimo y control discreto básicamente.

BIBLIOGRAFIA:

1.- KUO BENJAMIN, "Digital Control Systems", Holt-Saunders International Editions, 1981.

2.- KUO BENJAMIN, "Discrete-Data Control Systems", Prentice-Hall International,1970.

3.- OGATA KATSUHIKO, "Ingeniería de Control Moderna", Prentice-Hall International, 1973.

4.- D'AZZO JHON y HOUPIS CONSTANTINE, "Linear Control System Analysis and Desing",MacGraw-Hill International Boock Company,1981.

5.- GRANIZO EVELIO, "Desacoplamiento para Sistemas Continuos en el Tiempo", Tesis E.P.N., 1988.

6.- MUNRO N., "Multivariable Control Theory 3.- Decoupling Theory", University of Manchester of Science and Thecnology, 1976.

7.- MUNRO N., "Multivariable Control Theory 3.- Non-Interacting Control", University of Manchester Institute of Science and Thecnology, 1974.

8.- KEITHLEY, "Quick 500 Data Adquisition and Control Software", 1988.

9.- THOMPSON PETER, "Computer-Aided Control System Desing", System Thecnologi Inc., 1985.

10.- MICROSOFT, "Microsoft Quick Basic", Microsoft Corporation, 1988.

11.- MICROSOFT, "Microsoft Quick Basic 4.0", Microsoft Corporation, 1987.

12.- IBM, "Disk operating system version 3.3", IBM Corporation, 1987.

13.- MUNRO N.- "Modern Approaches to Control System Desing", Peter Peregrinus Ltd., 1979.

APENDICES:

APENDICE A: MANUAL DE USO

APENDICE B: MACROS DEL PROGRAMA CC

APENDICE C: LISTADO DE SOFTWARE

MANUAL DE USO

El software desarrollado consta de dos programas ejecutables, multipr.exe y multitr.exe para simulación y tiempo real respectivamente; adicionalmente se ha realizado el archivo MPR.BAT que determina la ejecución del software total que se encuentra compilado, mediante la determinación del path correspondiente y la anteposición de grun que determina el trabajo en tiempo real; en tal circunstancia el programa se ejecuta con MPR únicamente.

Inicialmente se dispone de un menú principal bajo el siguiente esquema:

SIMULACION	(1)
TIEMPO REAL	(2)
ABANDONAR PROGRAMA	(3)

Si se selecciona la tercera posibilidad se abandona totalmente el programa; las posibilidades 1 y 2 permiten visualizar el menú de simulación o de tiempo real respectivamente.

Primeramente se abordarán los tópicos correspondientes al menú de simulación el cual se lo presenta a continuación:

P.I.D. DIGITAL	(1)
REGULADOR CUADRATICO LINEAL ...	(2)
DESACOPLAMIENTO	(3)
ABANDONAR PROGRAMA	(4)

La cuarta posibilidad permite regresar al menú principal y las tres primeras posibilidades permiten ejecutar la técnica sugerida de acuerdo a la selección realizada.

Si se selecciona la primera posibilidad, se ingresa al menú de P.I.D. digital el cual se presenta a continuación:

P.I.D. MULTILAZO	(1)
P.I.D. MULTIVARIABLE	(2)
ABANDONAR PROGRAMA	(3)

En este menú, si se selecciona la tercera posibilidad, se retorna al menú de simulación; por el contrario si se selecciona la posibilidad 1 se ingresa a la ejecución de la técnica elegida.

Por otro lado, la segunda posibilidad permite ingresar a otro menú que se lo presenta a continuación:

INGRESO DE DATOS DEL SISTEMA
En forma continua .... (1)
En forma discreta .... (2)
Abandonar programa .... (3)

Si se selecciona la tercera posibilidad se retorna al menú de P.I.D. digital, en caso contrario se opta por la ejecución de la técnica elegida mediante el ingreso de datos en forma continua o discreta según la selección de la posibilidad 1 o 2 respectivamente.

Ahora, si en el menú de simulación se selecciona la posibilidad 2, esto es, regulador cuadrático lineal, se ingresa a un menú exactamente igual al menú de selección de ingreso de datos en forma continua o discreta, de tal modo que si se selecciona la tercera posibilidad se retorna al menú de simulación; en caso contrario se procede a la ejecución de la técnica seleccionada mediante el ingreso de datos en forma continua o discreta (posibilidades 1 o 2 respectivamente).

Por otro lado, si en el menú de simulación se selecciona la tercera posibilidad, esto es desacoplamiento, se procede directamente a la ejecución de la técnica en cuestión mediante ingreso de datos continuos.

La ejecución de la técnica especificada abarca primeramente el ingreso de datos del sistema, el ingreso de parámetros de compensadores, ingreso de referencias, valores iniciales, número de iteraciones,etc. y el desarrollo en si del algoritmo de control utilizado; por otro lado se tiene opciones de selección de variables a graficar, reingreso de condiciones de operación, visualización de resultados numéricos en pantalla, resultados completos en papel, generación de archivo de datos en formato DOS para posterior graficación en papel, etc. Luego de la ejecución completa, se retorna al menú de simulación.

Retomando el menú principal, si se selecciona la posibilidad 2 se ingresa al menú de tiempo real, en el que se elige la técnica de control a implementarse en tiempo real; el menú mencionado se presenta a continuación:

P.I.D.	(1)
REGULADOR CUADRATICO LINEAL ..	(2)
DESACOPLAMIENTO	(3)
ABANDONAR TIEMPO REAL	(4)

Si se selecciona la cuarta posibilidad se retorna al menú principal; en caso contrario, se opta por una de las tres técnicas de control aquí desarrolladas.

En cada una de las técnicas se tiene en forma general primeramente el ingreso de datos tales como número de entradas, número de salidas, periodo de muestreo, parámetros de compensadores,referencias,etc.

Especificamente en desacoplamiento se ingresa además las matrices de precompensación y de realimentación de estado calculadas previamente en simulación en forma off-line. En regulador cuadrático lineal se ingresa la matriz óptima de realimentación de estado calculada en simulación.

Dado que el programa está desarrollado de una manera interactiva, una vez que se ha ingresado a cualquiera de las opciones, el programa solicita los distintos parámetros a ingresar, de tal manera que su utilización es directa y sencilla: desde luego, el usuario debe conocer la teoría de las técnicas a utilizar para que el ingreso de datos sea consistente y los resultados sean satisfactorios.

En todo caso, para una ejecución correcta se recomienda remitirse a los ejemplos ilustrativos de la teoría, a los diagrámas de flujo de los programas y a los ejemplos en los resultados.

MACROS EN EL PROGRAMA CC

```
*************************************************************
OPTIMOC.MAC
*************************************************************

cls
state
echo,
echo,SISTEMA A SER CONTROLADO
pdisplay,p1
pause
quit
cls
state
echo,MATRIZ DE PONDERACION SIMETRICA SEMIDEFINIDA POSITIVA Q
pdisplay,p11
echo,MATRIZ DE PONDERACION SIMETRICA DEFINIDA POSITIVA R
pdisplay,p22
lqr,1,p1,p11,p22,p33,1
pause
feedback,4,p1,p33,p2
quit
cls
state
echo,MATRIZ OPTIMA DE REALIMENTACION DE ESTADO
pdisplay,p33
echo,SISTEMA EQUIVALENTE REALIMENTADO
pdisplay,p2
pause
simulation,p2,5,entrada,1,100,1,1
plot,p2,all,auto,REGULADOR-CUADRATICO-LINEAL.-CONTINUO,1,0


*************************************************************
OPTIMOC2.MAC
*************************************************************

cls
state
echo,
echo,SISTEMA A SER CONTROLADO
pdisplay,p1
pause
quit
cls
state
echo,MATRIZ DE PONDERACION SIMETRICA SEMIDEFINIDA POSITIVA Q
pdisplay,p11
echo,MATRIZ DE PONDERACION SIMETRICA DEFINIDA POSITIVA R
pdisplay,p22
lqr,1,p1,p11,p22,p33,1
pause
feedback,4,p1,p33,p2
```

```
quit
cls
state
echo,MATRIZ OPTIMA DE REALIMENTACION DE ESTADO
pdisplay,p33
echo,SISTEMA EQUIVALENTE REALIMENTADO
pdisplay,p2
pause
simulation,p2,5,entrada,2,p44,100,1,1
plot,p2,all,auto,REGULADOR-CUADRATICO-LINEAL.-CONTINUO,1,0


************************************************************
OPTIMOD.MAC
************************************************************

cls
state
echo,
echo,SISTEMA A SER CONTROLADO
pdisplay,p1
pause
quit
cls
state
echo,MATRIZ DE PONDERACION SIMETRICA SEMIDEFINIDA POSITIVA Q
pdisplay,p11
echo,MATRIZ DE PONDERACION SIMETRICA DEFINIDA POSITIVA R
pdisplay,p22
lqr,1,p1,p11,p22,p33,1
pause
convert,p1,p3,1,&1
quit
cls
state
echo,MATRIZ OPTIMA DE REALIMENTACION DE ESTADO F
pdisplay,p33
echo,SISTEMA ORIGINAL DISCRETIZADO
pdisplay,p3
pause
quit
cls
state
echo,NOS ENCONTRAMOS EN MODO DIGITAL
feedback,4,p3,p33,p2
echo,SISTEMA EQUIVALENTE REALIMENTADO
pdisplay,p2
pause
dsimulation,p2,5,entrada,1,100,1
plot,p2,all,auto,REGULADOR-CUADRATICO-LINEAL.-DISCRETO,1,0
```

```
************************************************************
OPTIMOD2.MAC

************************************************************

cls
state
echo,
echo,SISTEMA A SER CONTROLADO
pdisplay,p1
pause
quit
cls
state
echo,MATRIZ DE PONDERACION SIMETRICA SEMIDEFINIDA POSITIVA Q
pdisplay,p11
echo,MATRIZ DE PONDERACION SIMETRICA DEFINIDA POSITIVA R
pdisplay,p22
lqr,1,p1,p11,p22,p33,1
pause
convert,p1,p3,1,&1
quit
cls
state
echo,MATRIZ OPTIMA DE REALIMENTACION DE ESTADO F
pdisplay,p33
echo,SISTEMA ORIGINAL DISCRETIZADO
pdisplay,p3
pause
quit
cls
state
echo,NOS ENCONTRAMOS EN MODO DIGITAL
feedback,4,p3,p33,p2
echo,SISTEMA EQUIVALENTE REALIMENTADO
pdisplay,p2
pause
dsimulation,p2,5,entrada,2,p44,100,1
plot,p2,all,auto,REGULADOR-CUADRATICO-LINEAL.-DISCRETO,1,0


************************************************************
FUNTRANC
************************************************************

cls
echo,MATRIZ FUNCION DE TRANSFERENCIA.-
echo,SISTEMA ORIGINAL
state
fadeeva,p1,h1
hdisplay,h1
pause
quit
mimo
```

```
chpzf,h1,h3
echo,MATRIZ FUNCION DE TRANSFERENCIA.-
echo,FACTORES
hdisplay,h3
pause
determinant,h1,g1
quit
build
chpzf,g1,g2
quit
cls
echo,DETERMINANTE DE MATRIZ FUNCION DE TRANSFERENCIA
display,g1
echo,DETERMINANTE EN FACTORES
display,g2
pause


***************************************************************
FUNTRAND.MAC
***************************************************************

dig,&1
cls
echo,MATRIZ FUNCION DE TRANSFERENCIA.-
echo,SITEMA ORIGINAL DISCRETO
state
fadeeva,p3,h2
hdisplay,h2
pause
quit
mimo
chpzf,h2,h4
echo,MATRIZ FUNCION DE TRANSFERENCIA.-
echo,FACTORES
hdisplay,h4
pause
determinant,h2,g3
quit
build
chpzf,g3,g4
quit
cls
echo,DETERMINANTE DE MATRIZ FUNCION DE TRANSFERENCIA
display,g3
echo,DETERMINANTE EN FACTORES
display,g4
pause
analog
```

```
*******************************************************
KALMAN.MAC
*******************************************************

cls
state
echo,SISTEMA ORIGINAL
pdisplay,p1
echo,COVARIANZA DE RUIDO DE ENTRADA
pdisplay,p111
pause
quit
cls
state
echo,COVARIANZA DE RUIDO EN MEDICION
pdisplay,p222
kbf,1,p1,p111,p222,p4,&1
echo,GANANCIA DE ESTIMACION
pdisplay,p4
pause
quit


******************************************************
KALMUL.MAC
******************************************************

cls
state
multiply,p4,p6,p7
subtract,p5,p7,p8
echo,MATRIZ A EQUIVALENTE DE ERROR
pdisplay,p8
quit

*******************************************************
KALGRAF.MAC
*******************************************************

state
simulation,p9,6,2,p44,100,1,1
plot,p9,all,auto,ERROR.-FILTRO-KALMAN,1,0
```

```
DECLARE SUB retardo ()
DECLARE SUB submatrix (n%, m%, A#(), b#(), c#())
DECLARE SUB transmatrix (n%, m%, A#(), c#())
DECLARE SUB discret (n%, nu%, t#, acon#(), bcon#())
DECLARE SUB determatrix (n%, A#(), d#)
DECLARE SUB expomatrix (n%, A#(), nexpo%, aaux#())
DECLARE SUB mulmatrix (n%, m%, l%, A#(), b#(), c#())
DECLARE SUB invermatrix (n%, b#(), bt#(), ntr%)
DECLARE SUB negmatrix (n%, m%, A#(), an#())
DECLARE SUB summatrix (n%, m%, A#(), b#(), c#())
DECLARE SUB maxmin (n%, aa#(), vmax#, vmin#, delta#)
DECLARE SUB escalmatrix (n%, m%, escalar#, A#(), b#())
DEFINT I-N
DEFDBL A-H, O-Z
COMMON SHARED vmin1
menu1:
COLOR 15, 1
CLS
COLOR 2
LOCATE 4, 16
PRINT "C O N T R O L   M U L T I V A R I A B L E "
LOCATE 6, 19
PRINT "     E N   T I E M P O   R E A L"
LOCATE 10, 15
PRINT "S I M U L A C I O N .....................(1)"
LOCATE 14, 15
PRINT "T I E M P O   R E A L .................. (2)"
LOCATE 18, 15
PRINT "A B A N D O N A R   P R O G R A M A .... (3)"
LOCATE 23, 45
INPUT "Ingrese opción deseada"; opcion

'--------------Selección de opcion requerida--------------
SELECT CASE opcion
   CASE 1
     GOTO menu2          'menu2 es el menu de simulación
     CASE 2
      SHELL "multitr"
      GOTO menu1
   CASE 3
      GOTO fin
   CASE ELSE
      GOTO menu1          'menu1 es el menu principal
END SELECT
menu2:
COLOR 2
CLS
LOCATE 4, 10
PRINT "P. I. D.  D I G I T A L ................................... (1)"
LOCATE 9, 10
PRINT "R E G U L A D O R  C U A D R A T I C O  L I N E A L ....... (2)"
LOCATE 14, 10
PRINT "D E S A C O P L A M I E N T O ............................ (3)"
LOCATE 19, 10
PRINT "A B A D O N A R  P R O G R A M A ......................... (4)"
LOCATE 23, 45
```

```
INPUT "Ingrese opción deseada"; opcion
SELECT CASE opcion
   CASE 1
     COLOR 15
     GOTO piddigital
   CASE 2
      GOTO menu3      'menu3 es el menu del regulador cuadrático lineal
   CASE 3
      GOTO desacoplamiento
   CASE 4
      GOTO menu1      'menu1 es el menu principal
   CASE ELSE
      GOTO menu2      'menu2 es el menu de selección de técnica
                      'de control a ser usada
END SELECT

menu3:
CLS
LOCATE 5, 10
PRINT "R E G U L A D O R   C U A D R A T I C O   L I N E A L"
LOCATE 9, 22
PRINT "INGRESO DE DATOS DEL SISTEMA"
LOCATE 13, 19
PRINT "En forma continua ............. (1)"
LOCATE 16, 19
PRINT "En forma discreta ............. (2)"
LOCATE 19, 19
PRINT "Abandonar programa ............ (3)"
LOCATE 23, 45
INPUT "Ingrese opción deseada"; opcion
SELECT CASE opcion
   CASE 1
     opcion111$ = " continuos "
     GOTO regcuadlineal
     CASE 2
     opcion111$ = " discretos "
     GOTO regcuadlineal
   CASE 3
     GOTO menu2       'menu2 es el menu de selección de técnica
                      'de control a ser usada
   CASE ELSE
     GOTO menu3      'menu3 es el menu del regulador cuadrático lineal
END SELECT
'***************************************************************************
'************************** DESACOPLAMIENTO ********************************
'***************************************************************************
desacoplamiento:

COLOR 15, 1
CLS
PRINT "DESACOPLAMIENTO POR REALIMENTACION DE ESTADO"
PRINT

opcion11$ = "nocontrol"
'---------------------------------------------------------------------------
'Esta primera parte del programa tiene que ver con el ingreso de datos
'del sistema a ser desacoplado
'---------------------------------------------------------------------------
```

```
INPUT "Orden del sistema"; n
INPUT "Numero de entradas"; m
INPUT "Numero de salidas"; mn
IF m <> mn THEN
  CLS
  BEEP
  COLOR 4, 7
  LOCATE 11, 15
  PRINT " Para desacoplar un sistema se requiere que el número de"
  LOCATE 12, 15
  PRINT " entradas sea igual al número de salidas ;en este caso  "
  LOCATE 13, 15
  PRINT " no es posible desacoplar el sistema                     "
  CALL retardo
  GOTO findesacop
END IF

REDIM A(n, n), b(n, m), c(m, n), acon(n, n), bcon(n, n)
'******************************
REDIM c11(m, 5), d11(m, 5)
REDIM vector(m), matriz1#(m, 3)
'******************************
PRINT
PRINT "Ingreso de datos del sistema a ser desacoplado;matrices A,B,C"
PRINT
PRINT "Ingreso de datos continuos .- matriz A"
  FOR i = 1 TO n
    FOR j = 1 TO n
      PRINT "A("; i; ","; j; ")="; : INPUT A(i, j)
      acon(i, j) = A(i, j)
    NEXT
  NEXT

PRINT "Ingreso de datos continuos .- matriz B"
  FOR i = 1 TO n
    FOR j = 1 TO m
      PRINT "B("; i; ","; j; ")="; : INPUT b(i, j)
      bcon(i, j) = b(i, j)
    NEXT
  NEXT

PRINT "Ingreso de datos continuos .- matriz C"
  FOR i = 1 TO m
    FOR j = 1 TO n
      PRINT "C("; i; ","; j; ")="; : INPUT c(i, j)
    NEXT
  NEXT
'*************************************************************
PRINT
INPUT "Periodo de muestreo utilizado, T"; t
CALL discret(n, m, t, A(), b())
'-------------------------------------------------------------
PRINT
INPUT "Desea reingresar los valores del sistema (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO desacoplamiento
END IF
'*************************************************************
```

```
PRINT
'-----------------------------------------------------------------------------
'A continuación se tiene el cálculo de la matriz B*
'-----------------------------------------------------------------------------
REDIM aaux(n, n), ab(n, m), cab(1, m), bbb(m, m), caux(1, n), dd%(m)
REDIM bbbi(m, m), f(m, n), e(m, m), caa(1, n), bba(m, n), bban(m, n)

FOR mm = 1 TO m
     FOR i = 1 TO n
       caux(1, i) = c(mm, i)
     NEXT
     k = 0
lazo1:
     CALL expomatrix(n, acon(), k, aaux())
     CALL mulmatrix(n, n, m, aaux(), bcon(), ab())
     CALL mulmatrix(1, n, m, caux(), ab(), cab())

     'El siguiente párrafo verifica si es o no diferente de cero el
     'producto matricial

     sumauno = 0
     FOR i = 1 TO m
       sumauno = sumauno + ABS(cab(1, i))
     NEXT
     IF sumauno = 0 THEN
        k = k + 1
         IF k <= n - 1 THEN
           GOTO lazo1
         ELSE
           k = n - 1
          GOSUB salida1
         END IF
     ELSE
        GOSUB salida1
     END IF
NEXT

'-----------------------------------------------------------------------------
'En la siguiente parte del programa se determina det B* y B*-1
'-----------------------------------------------------------------------------
CALL determatrix(m, bbb(), det)
IF det = 0 THEN
  CLS
  BEEP
  COLOR 4, 7
  LOCATE 11, 15
  PRINT "           La matriz B* es singular          "
  LOCATE 12, 15
  PRINT "El sistema no puede ser desacoplado directamente"
  CALL retardo
  COLOR 15, 1
  CLS
  GOTO menu2
ELSE
  CALL invermatrix(m, bbb(), bbbi(), ntr)
END IF
```

```
'------------------------------------------------------------------------
'A continuación se realiza el cálculo de la matriz A*
'------------------------------------------------------------------------
FOR mm = 1 TO m
   FOR i = 1 TO n
     caux(1, i) = c(mm, i)
   NEXT

   k = dd%(mm)
   CALL expomatrix(n, acon(), k + 1, aaux())
   CALL mulmatrix(1, n, n, caux(), aaux(), caa())

   FOR i = 1 TO n
     aaa(mm, i) = caa(1, i)
   NEXT
NEXT

'------------------------------------------------------------------------
'Inmediatamente se calculan las matrices F y E de realimentación de estad
'------------------------------------------------------------------------

'Cálculo de la matriz E
   FOR i = 1 TO m
     FOR j = 1 TO m
       e(i, j) = bbbi(i, j)
     NEXT
   NEXT

'Cálculo de la matriz F
   CALL mulmatrix(m, m, n, bbbi(), aaa(), bba())
   CALL negmatrix(m, n, bba(), bban())
   FOR i = 1 TO m
     FOR j = 1 TO n
       f(i, j) = bban(i, j)
     NEXT
   NEXT

'------------------------------------------------------------------------
'Este par desacoplaría al sistema si det(b*) es distinto de cero, pero
'puede producir inestabilidad o mala configuración de polos con mirar
'modos inobservables. En este caso, la matriz de realimentación puede ser
'modificada a F1=F+F0, produciendo un set de polos de lazo cerrado.
'A continuación se realiza el cálculo de la matriz de realimentación F1:
'------------------------------------------------------------------------

otrasM:
'---------------
'Cálculo de F0
'---------------
REDIM mca1#(m, n), m#(m, m), ca(m, n), mca#(m, n)
REDIM summca(m, n), f0(m, n), f1(m, n)
'Primeramente δ=max di=ndelta1
ndelta1 = dd%(1)
FOR i = 2 TO m
  IF dd%(i) > ndelta1 THEN
    ndelta1 = dd%(i)
  END IF
NEXT
```

```
'------------------------------------------------------------------
otrasM1:
'------------------------------------------------------------------
FOR i = 1 TO m
  FOR j = 1 TO n
    mca1#(i, j) = 0
  NEXT
NEXT
'------------------------------------------------------------------

CLS
PRINT "Ingresos de matrices M que determinan polos de lazo cerrado"
PRINT
REDIM maux#(m, m, ndelta1 + 1)
FOR k = 0 TO ndelta1

   FOR i = 1 TO m
     FOR j = 1 TO m
       PRINT "M"; k; "("; i; ","; j; ")=";
       INPUT m#(i, j)
       maux#(i, j, k + 1) = m#(i, j)
     NEXT
   NEXT
   PRINT
   CALL expomatrix(n, acon(), k, aaux())
   CALL mulmatrix(m, n, n, c(), aaux(), ca())
   CALL mulmatrix(m, m, n, m#(), ca(), mca#())
   CALL summatrix(m, n, mca1#(), mca#(), summca())

   FOR i = 1 TO m
     FOR j = 1 TO n
       mca1#(i, j) = mca#(i, j)
     NEXT
   NEXT
NEXT
'--------------------------------------------------------------------
PRINT
INPUT "Desea reingresar los valores de matrices Mk (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO otrasM1
END IF
'--------------------------------------------------------------------
CALL mulmatrix(m, m, n, bbbi(), summca(), f0())
CALL summatrix(m, n, f(), f0(), f1())

'Con lo que se tiene el par de desacoplamiento (F1,E)
'Se puede reasignar los polos deseados de lazo cerrado, reasignando
'las matrices Mk

CLS
PRINT
PRINT "Matriz de realimentacion de estado F1"
PRINT
FOR i = 1 TO m
  FOR j = 1 TO n
    PRINT "F1("; i; ","; j; ")=";
    PRINT USING "#####.######"; f1(i, j)
  NEXT
NEXT
```

```
PRINT
PRINT "Matriz de preamplificacion E"
PRINT
PRINT
FOR i = 1 TO m
  FOR j = 1 TO m
    PRINT "E("; i; ","; j; ")=";
    PRINT USING "#####.######"; e(i, j)
  NEXT
NEXT
PRINT
INPUT "Presione RETURN para continuar"; opcion$

'-----------------------------------------------------------------------------
'A continuación se realiza la simulación del sistema desacoplado, para lo
'cual las matrices del sistema han sido discretizadas:
'-----------------------------------------------------------------------------

REDIM xaux(n, 1), r(m, 1), er(m, 1), fx(m, 1), uaux(m, 1)
REDIM yaux(m, 1), bu(n, 1), ax(n, 1), v(m), x111(n)
CLS
PRINT
PRINT "Valores iniciales de los estados"

GOSUB valorinicial2

PRINT
PRINT "Valores de referencia"

FOR i = 1 TO m
  PRINT "r"; i; "=";
  INPUT r(i, 1)
NEXT
'-----------Se almacena en vectores:estados,salidas y leyes de control---
nuevomuestreo:

INPUT "Número de muestreos deseados"; l
REDIM x(n, 1, l), u(m, 1, l), y(m, 1, l), auxiliar(l)
'***************************
REDIM r22(m, 1), e11(m, 1)
'***************************
otrareferencia:

FOR k = 1 TO l
  IF k = 1 THEN
   FOR i = 1 TO n
      x(i, 1, k) = x111(i)
      xaux(i, 1) = x111(i)
   NEXT
  ELSE
   FOR i = 1 TO n
      x(i, 1, k) = xaux(i, 1)
   NEXT
  END IF
```

```
'*****************************
IF opcion11$ = "control" THEN
'*****************************
   CALL mulmatrix(m, n, 1, c(), xaux(), yaux())

   FOR j = 1 TO m

      '----------- cálculo del error -------------
      e11(j, k) = v(j) - yaux(j, 1)

      '------ cálculo de salida del compensador----
      r22(j, k) = 0
        FOR i = 1 TO 5
          IF k - i > 0 THEN
            r22(j, k) = r22(j, k) + c11(j, i) * r22(j, k - i)
          END IF
        NEXT
        FOR i = 1 TO 5
          IF k - i + 1 > 0 THEN
            r22(j, k) = r22(j, k) + d11(j, i) * e11(j, k - i + 1)
          END IF
        NEXT
        r(j, 1) = r22(j, k)

   NEXT
'****************
END IF
'****************

   CALL mulmatrix(m, m, 1, e(), r(), er())
   CALL mulmatrix(m, n, 1, f1(), xaux(), fx())
   CALL summatrix(m, 1, fx(), er(), uaux())

   FOR i = 1 TO m
     u(i, 1, k) = uaux(i, 1)
   NEXT

   CALL mulmatrix(m, n, 1, c(), xaux(), yaux())
   FOR i = 1 TO m
     y(i, 1, k) = yaux(i, 1)
   NEXT

   CALL mulmatrix(n, m, 1, b(), uaux(), bu())
   CALL mulmatrix(n, n, 1, A(), xaux(), ax())
   CALL summatrix(n, 1, ax(), bu(), xaux())

NEXT

'---------------------------------------------------------------------------
'A continuación se realiza la parte del programa correspondiente a
'gráficos, según tres opciones:estados,salidas,leyes de control.
'---------------------------------------------------------------------------
'*************************** importante ********************* *******
SCREEN 9
VIEW (44, 45)-(630, 300), 1, 1
'*********************************************************** *******
```

```
'*****************************
IF opcion11$ = "control" THEN
'*****************************
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
opcion22$ = "DESACOPLAMIENTO .- Sistema  controlado"

repita11:
     VIEW PRINT 1 TO 3
     CLS 2
     PRINT "Desea gráficos de salida (1), ley de control (2),"
     PRINT "error (3), abandonar opción de gráficos (4)";
     INPUT opcion
     IF opcion < 1 OR opcion > 4 THEN
       GOTO repita11
     END IF
     SELECT CASE opcion
        CASE 1
salida11:
           opcion33$ = "salida       y"
           CLS 0
           VIEW (45, 45)-(630, 300), 1, 1
           PRINT "Salida que desea graficar (valor desde 1 hasta"; m; ")";
           INPUT nsalida
           IF nsalida < 1 OR nsalida > m THEN
             GOTO salida11
           END IF
           FOR i = 1 TO l
             auxiliar(i) = y(nsalida, 1, i)
           NEXT
           nopcion = nsalida
           GOSUB ventana
           LINE (-1, v(nsalida))-(l + 2, v(nsalida)), 3, , &H8888
           GOSUB pantalla
           GOTO repita11

        CASE 2
leycontrol22:
           opcion33$ = "Ley de control       u"
           CLS 0
           VIEW (45, 45)-(630, 300), 1, 1
           PRINT "Ley de control que desea graficar (valor desde 1 hasta"
           "; m; ")";
           INPUT nentrada
           IF nentrada < 1 OR nentrada > m THEN
             GOTO leycontrol22
           END IF
           FOR i = 1 TO l
             auxiliar(i) = u(nentrada, 1, i)
           NEXT
           nopcion = nentrada
           GOSUB ventana
           GOSUB pantalla
           GOTO repita11
```

```
        CASE 3
error22:
            opcion33$ = "Error       e"
            CLS 0
            VIEW (45, 45)-(630, 300), 1, 1
            PRINT "Error que desea graficar (valor desde 1 hasta"; m; ")";
            INPUT nentrada
            IF nentrada < 1 OR nentrada > m THEN
              GOTO error22
            END IF
            FOR i = 1 TO l
              auxiliar(i) = e11(nentrada, i)
            NEXT
            nopcion = nentrada
            GOSUB ventana
            GOSUB pantalla
            GOTO repita11

        CASE 4
            CLS 0
            VIEW (45, 45)-(630, 300), 1, 1
     END SELECT

'************
ELSE
'************
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
opcion22$ = "DESACOPLAMIENTO .- Sistema no controlado"

repita1:
VIEW PRINT 1 TO 3
CLS 2
PRINT "Desea gráficos de estados(1),ley de control(2),"
PRINT "salidas del sistema desacoplado(3),abandonar programa (4)";
INPUT numero

IF numero < 1 OR numero > 4 THEN
'si no se especifica en forma adecuada se retorna al menu de gráficos
     GOTO repita1

ELSE
   '****REDIM auxiliar(l)
   IF numero = 1 THEN
   '-----------a continuación: grafico correspondientes a estados--------
estado:
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
   opcion33$ = "estado       x"
   CLS 0
   VIEW (45, 45)-(630, 300), 1, 1
     PRINT "estado que desea graficar (valor desde 1 hasta"; n; ")"; :
     INPUT nestado
        IF nestado < 1 OR nestado > n THEN
           GOTO estado
        END IF
        nopcion = nestado '¿¿¿¿¿¿¿¿¿¿¿¿
       FOR i = 1 TO l
         auxiliar(i) = x(nestado, 1, i)
       NEXT
```

```
      GOSUB ventana
      GOSUB pantalla   '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
      GOTO repital

    ELSEIF numero = 2 THEN
    '--------a continuación:grafico correspondiente a ley de control------
leycontrol:
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
    opcion33$ = "ley de control          u"
    CLS 0
    VIEW (45, 45)-(630, 300), 1, 1
      PRINT "Ley de control que desea graficar (valor desde 1 hasta"; m; ".
      INPUT nentrada
         IF nentrada < 1 OR nentrada > m THEN
            GOTO leycontrol
         END IF
         nopcion = nentrada    '¿¿¿¿¿¿¿¿
        FOR i = 1 TO l
          auxiliar(i) = u(nentrada, 1, i)
        NEXT
      GOSUB ventana
      GOSUB pantalla
      GOTO repital

    ELSEIF numero = 4 THEN

       CLS 0
       VIEW (45, 45)-(630, 300), 1, 1

    ELSE
    '---------a continuación:grafico correspondiente salidas del ---------
    '---------sistema realimentado-----------------------------------------
salidasistem:
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
    opcion33$ = "salida        y"
    CLS 0
    VIEW (45, 45)-(630, 300), 1, 1
             PRINT "salida que desea graficar (valor desde 1 hasta"; m; ")"
             INPUT nsalida
               IF nsalida < 1 OR nsalida > m THEN
                  GOTO salidasistem
               END IF
               nopcion = nsalida
               FOR i = 1 TO l
                 auxiliar(i) = y(nsalida, 1, i)
               NEXT
             GOSUB ventana
             LINE (-1, r(nsalida, 1))-(l + 2, r(nsalida, 1)), 3, , &H8888
             GOSUB pantalla
             GOTO repital

           END IF
END IF
'------------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea cambiar el valor inicial de los estados (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
```

```
   GOSUB valorinicial2

   GOTO otrareferencia
END IF
'--------------------------------------------------------------------------
opcion$ = "n"
PRINT "Desea cambiar el valor de las referencias del sistema desacoplado
(s/n)";
INPUT opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
   FOR i = 1 TO m
     PRINT "r"; i; "=";
     INPUT r(i, 1)
   NEXT
'--------------------------------------------------------------------------
  FOR i = 1 TO n                       'reactualiza el valor inicial de los
    xaux(i, 1) = x(i, 1, 1)            'estados para poder realizar la simu-
  NEXT                                 'lación con un nuevo # de muestreos.
'--------------------------------------------------------------------------
   GOTO otrareferencia
END IF
opcion$ = "n"
INPUT "Desea cambiar el número de iteraciones (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
'  SCREEN 0
'--------------------------------------------------------------------------
  FOR i = 1 TO n                       'reactualiza el valor inicial de los
    xaux(i, 1) = x(i, 1, 1)            'estados para poder realizar la simu-
  NEXT                                 'lación con un nuevo # de muestreos.
'--------------------------------------------------------------------------
  GOTO nuevomuestreo
END IF
'-------------------------
opcion1$ = "n"
INPUT "Desea ensayar otras matrices M en el cálculo de F0 (s/n)"; opcion1
IF opcion1$ = "s" OR opcion1$ = "S" THEN
  SCREEN 0
  COLOR 15, 1
  GOTO otrasM
END IF
'---------------------------
opcion$ = "n"
INPUT "Desea compensar el sistema desacoplado (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  opcion11$ = "control"
  SCREEN 0
  COLOR 15, 1
  GOSUB otrocomp1
  GOSUB referenciaext
  GOTO otrareferencia
END IF
'--------------------------------------------------------------------------
'--------------------------------------------------------------------------
'La siguiente parte del programa, permite imprimir en papel los diferent
'valores de las diversas matrices asi como los principales resultados de
'la simulación para el sistema desacoplado y no compensado
'--------------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea resultados completos en papel (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
```

```
  OPEN "lpt1:" FOR OUTPUT AS #1
     GOSUB desacopsist
     PRINT #1,
     PRINT #1, opcion22$
     PRINT #1,
     PRINT #1, "Valores resultantes de la simulación:"
     PRINT #1,
     PRINT #1, "Periodo de muestreo (s) .............. T="; t
     PRINT #1, "Número de iteraciones ............... N="; l
     PRINT #1, "Tiempo total del proceso (s) ......... t="; l * t
     PRINT #1,
     PRINT #1, "Valores de los estados"
     FOR i = 1 TO n
       PRINT #1,
       PRINT #1, "Estado       x"; i
       FOR j = 1 TO l
         auxiliar(j) = x(i, 1, j)
       NEXT
       GOSUB impresora
     NEXT

     PRINT #1,
     PRINT #1, "Valores de las leyes de control"
     FOR i = 1 TO m
       PRINT #1,
       PRINT #1, "Ley de control       u"; i
       FOR j = 1 TO l
         auxiliar(j) = u(i, 1, j)
       NEXT
       GOSUB impresora
     NEXT

     PRINT #1,
     PRINT #1, "Valores de salida del sistema desacoplado"
     FOR i = 1 TO m
       PRINT #1,
       PRINT #1, "Salida       y"; i
       FOR j = 1 TO l
         auxiliar(j) = y(i, 1, j)
       NEXT
       GOSUB impresora
     NEXT

  CLOSE #1

END IF
'-------------------------------------------------------------------------
'-------------------------------------------------------------------------


'********
END IF
'********
'*********************************
IF opcion11$ = "control" THEN
'*********************************
```

```
opcion$ = "n"
INPUT "Desea ensayar un nuevo compensador (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  SCREEN 0
  COLOR 15, 1
  GOSUB otrocomp1
  GOTO otrareferencia
END IF
'------------------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea ingresar nuevos valores iniciales (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOSUB valorinicial2
  GOTO otrareferencia
END IF
'------------------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea cambiar las referencias externas (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOSUB referenciaext
  GOTO otrareferencia
END IF
'------------------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea cambiar el número de iteraciones (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO nuevomuestreo
END IF
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
'La siguiente parte del programa, permite imprimir en papel los diferente
'valores de las diversas matrices asi como los principales resultados de
'la simulación para el sistema desacoplado y compensado
'------------------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea resultados completos en papel (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN

  OPEN "lpt1:" FOR OUTPUT AS #1
     GOSUB desacopsist
     PRINT #1,
     PRINT #1, opcion22$
     PRINT #1,
     PRINT #1, "Valores resultantes de la simulación:"
     PRINT #1,
     PRINT #1, "Periodo de muestreo (s) .............. T="; t
     PRINT #1, "Número de iteraciones ................ N="; l
     PRINT #1, "Tiempo total del proceso (s) ......... t="; l * t
     PRINT #1,
     PRINT #1, "Valores de referencia utilizados"
     FOR i = 1 TO m
       PRINT #1, "referencia"; i; "="; v(i)
     NEXT
     PRINT #1,
     PRINT #1, "Valores de salida"
```

```
    FOR i = 1 TO m
      PRINT #1,
      PRINT #1, "Salida        y"; i
      FOR j = 1 TO l
        auxiliar(j) = y(i, 1, j)
      NEXT
      GOSUB impresora
    NEXT

    PRINT #1,
    PRINT #1, "Valores de las leyes de control"
    FOR i = 1 TO m
      PRINT #1,
      PRINT #1, "Ley de control       u"; i
      FOR j = 1 TO l
        auxiliar(j) = u(i, 1, j)
      NEXT
      GOSUB impresora
    NEXT

    PRINT #1,
    PRINT #1, "Valores de error del sistema desacoplado y compensado"
    FOR i = 1 TO m
      PRINT #1,
      PRINT #1, "Error        e"; i
      FOR j = 1 TO l
        auxiliar(j) = e11(i, j)
      NEXT
      GOSUB impresora
    NEXT

  CLOSE #1

END IF
'-------------------------------------------------------------------------------
'-------------------------------------------------------------------------------


'*****************
END IF
'*****************
'*******************************************************************************
'*******************************************************************************
'La siguiente parte del programa nos da la posibilidad de poder almacenar
'los datos resultantes de la simulación para poder ser utilizados mediante
'las diversas opciones de gráficos que nos proporciona el paquete de hoja
'de cálculo LOTUS.(El archivo de datos será almacenado con extensión .PRN

CLS 2
opcion$ = "n"
PRINT "Desea crear archivo de datos de los resultados obtenidos para"
INPUT "graficación posterior en papel desde LOTUS  (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN  '================================
   INPUT "Path y nombre de archivo (sin extensión)"; opcion$
   OPEN opcion$ + ".prn" FOR OUTPUT AS #1
     IF opcion11$ = "control" THEN      '------------------------
       FOR i = 1 TO m
        PRINT #1, v(i)
       NEXT
```

```
      PRINT #1, 0;
      FOR j = 1 TO 3
       FOR i = 1 TO m
        PRINT #1, i;
       NEXT
      NEXT
      PRINT #1,
      FOR j = 1 TO l
        PRINT #1, j;
        FOR i = 1 TO m
          PRINT #1, y(i, 1, j);
        NEXT
        FOR i = 1 TO m
          PRINT #1, u(i, 1, j);
        NEXT
        FOR i = 1 TO m
          PRINT #1, USING "####.######"; e11(i, j);
        NEXT
        PRINT #1,
      NEXT
    ELSE                                          '------------------------
      PRINT #1, 0;
      FOR i = 1 TO n
        PRINT #1, i;
      NEXT
      FOR j = 1 TO 2
       FOR i = 1 TO m
        PRINT #1, i;
       NEXT
      NEXT
      PRINT #1,
      FOR j = 1 TO l
        PRINT #1, j;
        FOR i = 1 TO n
          PRINT #1, x(i, 1, j);
        NEXT
        FOR i = 1 TO m
          PRINT #1, u(i, 1, j);
        NEXT
        FOR i = 1 TO m
          PRINT #1, y(i, 1, j);
        NEXT
        PRINT #1,
      NEXT
    END IF                                        '------------------------
  CLOSE #1
END IF                                            '==============================

'************************************************************************
'************************************************************************


findesacop:
SCREEN 0
COLOR 15, 1
CLEAR
GOTO menu2
```

```
'*****************************************************************************
'************************** P.I.D.  DIGITAL ****************************
'*****************************************************************************

piddigital:
COLOR 15, 1
CLS
menu4:
CLS
COLOR 2
LOCATE 7, 13
PRINT "P. I. D.   M U L T I L A Z O ......................... (1)"
LOCATE 12, 13
PRINT "P. I. D.   M U L T I V A R I A B L E ............... (2)"
LOCATE 17, 13
PRINT "A B A N D O N A R   P R O G R A M A ............... (3)"

LOCATE 23, 45
INPUT "Ingrese opción deseada"; opcion
SELECT CASE opcion
   CASE 1
     npid1 = 1
     opcion22$ = "P.I.D.   MULTILAZO"
     GOTO multilazo
   CASE 2
     npid1 = 2
     opcion22$ = "P.I.D.   MULTIVARIABLE"
     GOTO multivariable
   CASE 3
     GOTO menu2     'Menú de selección de técnica a usarse
   CASE ELSE
     GOTO menu4     'Menú de P.I.D.
END SELECT

multilazo:
CLS
PRINT
LOCATE 8, 15
PRINT "Sistema de control discreto: Compensador P.I.D."
LOCATE 13, 28
PRINT "M U L T I L A Z O"
'
'Para control multilazo
'
LOCATE 18, 23
INPUT "Número de lazos a controlar"; m
COLOR 15, 1
'Se considera que el mayor número de retardos es de orden 5
REDIM a11(m, 5), b11(m, 5), c11(m, 5), d11(m, 5)
'¿¿¿¿¿¿¿¿¿¿¿¿
REDIM vector(m), matriz1#(m, 3)
'¿¿¿¿¿¿¿¿¿¿¿¿
FOR j = 1 TO m
   CLS
   LOCATE 3, 11
   PRINT "L A Z O"; j
   LOCATE 5
   PRINT "Orden de la salida en el lazo"; j; : INPUT nn1
```

```
    PRINT
    PRINT "Orden de la entrada en el lazo"; j; : INPUT nn2
    PRINT
    PRINT
    PRINT "Lazo"; j; ".- Coeficientes de retardo en y"
    PRINT
    FOR i = 1 TO nn1
      PRINT "Coeficiente de retardo en y(k-"; i; ")="; : INPUT a11(j, i)
    NEXT
    PRINT
    PRINT
    PRINT "Lazo"; j; ".- Coeficientes de retardo en u"
    PRINT
    FOR i = 1 TO nn2
      PRINT "Coeficiente de retardo en u(k-"; i; ")="; : INPUT b11(j, i)
    NEXT
NEXT
PRINT
INPUT "Periodo de muestro ................. T"; t
CLS
PRINT
'---------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea reingresar valores del sistema (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO multilazo
END IF
'---------------------------------------------------------------------
otrocomp:
GOSUB otrocomp1
REDIM r11(m), y111(m)


'A continuación se realiza el ingreso de las referencias del sistema

GOSUB referencia

GOSUB valorinicial1


'---------------------------------------------------------------------
PRINT
opcion$ = "n"
INPUT "Desea reingresar valores de control (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO otrocomp
  END IF
'---------------------------------------------------------------------
nuevamuest:

INPUT "Número de iteraciones deseado"; l

'A continuación se realiza el ingreso de las referencias del sistema

REDIM y11(m, l), u11(m, l), e11(m, l)
```

```
'-----------------------------------------------------------------------
'En el siguiente párrafo del programa se realiza el cálculo de los
'distintos valores de salida,error y leyes de control en l periodos
'de muestreo
'-----------------------------------------------------------------------
pid1:

FOR k = 1 TO l
   FOR j = 1 TO m

     '----------------- Cálculo de la salida y -------------

    IF k = 1 THEN
      y11(j, k) = y111(j)
    ELSE

      y11(j, k) = 0
      FOR i = 1 TO 5
        IF k - i > 0 THEN
          y11(j, k) = y11(j, k) + a11(j, i) * y11(j, k - i)
          y11(j, k) = y11(j, k) + b11(j, i) * u11(j, k - i)
        END IF
      NEXT
    END IF


     '----------------- Cálculo del error e -----------------
        e11(j, k) = r11(j) - y11(j, k)

      '------------- Cálculo de la ley de control u ----------
       u11(j, k) = 0
       FOR i = 1 TO 5
         IF k - i > 0 THEN
           u11(j, k) = u11(j, k) + c11(j, i) * u11(j, k - i)
         END IF
       NEXT
       FOR i = 1 TO 5
         IF k - i + 1 > 0 THEN
           u11(j, k) = u11(j, k) + d11(j, i) * e11(j, k - i + 1)
         END IF
       NEXT
   NEXT
NEXT
'**************************** importante ******************************
  SCREEN 9
  VIEW (45, 45)-(630, 300), 1, 1
'*********************************************************************

'---------------------------------------------------------------------
'A continuación se realiza las rutinas correspondientes a los distintos
'gráficos que desee visualizar el usuario del presente programa
'---------------------------------------------------------------------

repita:

VIEW PRINT 1 TO 3
CLS 2
PRINT "Desea gráficos de salida (1),ley de control (2),"
PRINT "error (3) ,abandonar opción de gráficos (4)";
INPUT numero
```

```
IF numero < 1 OR numero > 4 THEN
'si no se especifica en forma adecuada se retorna al menu de gráficos
     GOTO repita

ELSE
   REDIM auxiliar(l)

   IF numero = 1 THEN
otrasalida:
   opcion33$ = "Salida       y"
   CLS 0
   CLS 2
   VIEW (45, 45)-(630, 300), 1, 1
   PRINT "Salida que desea graficar (valor desde 1 hasta "; m; ")";
   INPUT nsalida
   IF nsalida < 1 OR nsalida > m THEN
      GOTO otrasalida
   END IF
   nopcion = nsalida

   '-----------a continuación: grafico correspondiente a salida----------
       FOR i = 1 TO l
         auxiliar(i) = y11(nsalida, i)
       NEXT
     GOSUB ventana
     LINE (-1, r11(nsalida))-(l + 2, r11(nsalida)), 3, , &H8888
     GOSUB pantalla
     GOTO repita

   ELSEIF numero = 2 THEN
   '--------a continuación:grafico correspondiente a ley de control------
nuevocontrol:
     opcion33$ = "Ley de control       u"
     CLS 0
     CLS 2
     VIEW (45, 45)-(630, 300), 1, 1
     PRINT "Ley de control que desea graficar (valor desde 1 hasta "; m;
     ")";
     INPUT ncontrol
     IF ncontrol < 1 OR ncontrol > m THEN
        GOTO nuevocontrol
     END IF
     nopcion = ncontrol
       FOR i = 1 TO l
         auxiliar(i) = u11(ncontrol, i)
       NEXT
     GOSUB ventana
     GOSUB pantalla
     GOTO repita

   ELSEIF numero = 4 THEN
   CLS 0
   VIEW (45, 45)-(630, 300), 1, 1

   ELSE
   '------------a continuación:grafico correspondiente al error----------
nuevoerror:
              opcion33$ = "Error       e"
              CLS 0
```

```
            CLS 2
            VIEW (45, 45)-(630, 300), 1, 1
            PRINT "Error que desea graficar (valor desde 1 hasta "; m; ")
            INPUT nerror
            IF nerror < 1 OR nerror > m THEN
              GOTO nuevoerror
            END IF
            nopcion = nerror

              FOR i = 1 TO 1
                auxiliar(i) = e11(nerror, i)
              NEXT
            GOSUB ventana
            GOSUB pantalla
            GOTO repita

          END IF
END IF
CLS 2
'-----------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea ensayar un nuevo compensador (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  SCREEN 0
  COLOR 15, 1
  SELECT CASE npid1
      CASE 1
         GOSUB otrocomp1
         GOTO pid1
      CASE 2
         GOSUB otrocomp1
         GOTO pid2
  END SELECT
END IF

'-----------------------------------------------------------------------
'-----------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea cambiar condiciones iniciales (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  SELECT CASE npid1
      CASE 1
        GOSUB valorinicial1
        GOTO pid1
      CASE 2
        GOSUB valorinicial2
        GOTO pid2
  END SELECT
END IF
'-----------------------------------------------------------------------

opcion$ = "n"
INPUT "Desea cambiar la referencia (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  SELECT CASE npid1
      CASE 1
        GOSUB referencia
        GOTO pid1
```

```
      CASE 2
        GOSUB referencia2
        GOTO pid2
  END SELECT
END IF
'-----------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea cambiar el número de muestreos (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  SELECT CASE npid1
      CASE 1
        GOTO nuevamuest
      CASE 2
        GOTO nuevamuest2
  END SELECT
END IF
'-----------------------------------------------------------------------
'La siguiente parte del programa, permite imprimir en papel los diferente
'valores de las diversas matrices asi como los principales resultados de
'la simulación
'-----------------------------------------------------------------------
'-----------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea resultados completos en papel (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN

  OPEN "lpt1:" FOR OUTPUT AS #1
     GOSUB pidsist
     PRINT #1,
     PRINT #1, opcion22$
     PRINT #1,
     PRINT #1, "Valores resultantes de la simulación:"
     PRINT #1,
     PRINT #1, "Periodo de muestreo (s) ................ T="; t
     PRINT #1, "Número de iteraciones .................. N="; l
     PRINT #1, "Tiempo total del proceso (s) ........... t="; l * t
     PRINT #1,
     PRINT #1, "Valores de referencia utilizados"
     FOR i = 1 TO m
       PRINT #1, "referencia"; i; "="; r11(i)
     NEXT
     PRINT #1,
     PRINT #1, "Valores de salida"
     FOR i = 1 TO m
       PRINT #1,
       PRINT #1, "Salida        y"; i
       FOR j = 1 TO l
         auxiliar(j) = y11(i, j)
       NEXT
       GOSUB impresora
     NEXT

     PRINT #1,
     PRINT #1, "Valores de las leyes de control"
     FOR i = 1 TO m
       PRINT #1,
       PRINT #1, "Ley de control        u"; i
       FOR j = 1 TO l
         auxiliar(j) = u11(i, j)
       NEXT
       GOSUB impresora
     NEXT
```

```
    PRINT #1,
    PRINT #1, "Valores de error del sistema controlado"
    FOR i = 1 TO m
      PRINT #1,
      PRINT #1, "Error       e"; i
      FOR j = 1 TO l
        auxiliar(j) = e11(i, j)
      NEXT
      GOSUB impresora
    NEXT

  CLOSE #1

END IF
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
'***************************************************************************
'***************************************************************************
'La siguiente parte del programa nos da la posibilidad de poder almacenar
'los datos resultantes de la simulación para poder ser utilizados mediante
'las diversas opciones de gráficos que nos proporciona el paquete de hoja
'de cálculo LOTUS.(El archivo de datos será almacenado con extensión .PRN

CLS 2
opcion$ = "n"
PRINT "Desea crear archivo de datos de los resultados obtenidos para"
INPUT "graficación posterior en papel desde LOTUS  (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN  '==============================
   INPUT "Path y nombre de archivo (sin extensión)"; opcion$
   OPEN opcion$ + ".prn" FOR OUTPUT AS #1
       FOR i = 1 TO m
        PRINT #1, r11(i)
       NEXT
       PRINT #1, 0;
       FOR j = 1 TO 3
        FOR i = 1 TO m
         PRINT #1, i;
        NEXT
       NEXT
       PRINT #1,
       FOR j = 1 TO l
         PRINT #1, j;
         FOR i = 1 TO m
           PRINT #1, y11(i, j);
         NEXT
         FOR i = 1 TO m
           PRINT #1, u11(i, j);
         NEXT
         FOR i = 1 TO m
           PRINT #1, USING "####.######"; e11(i, j);
         NEXT
         PRINT #1,
       NEXT
   CLOSE #1
END IF                                   '==============================

'***************************************************************************
'***************************************************************************
```

```
'-----------------------------------------------------------------------

finpid:
SCREEN 0
COLOR 15, 1
CLEAR
GOTO menu4
multivariable:
menu5:
CLS
COLOR 2
LOCATE 5, 18
PRINT "P. I. D.   M U L T I V A R I A B L E"
LOCATE 9, 22
PRINT "INGRESO DE DATOS DEL SISTEMA"
LOCATE 13, 19
PRINT "En forma continua ............. (1)"
LOCATE 16, 19
PRINT "En forma discreta ............. (2)"
LOCATE 19, 19
PRINT "Abandonar programa ............ (3)"
LOCATE 23, 45
INPUT "Ingrese opción deseada"; opcion
SELECT CASE opcion
   CASE 1
     opcion111$ = " continuos "
   CASE 2
     opcion111$ = " discretos "
   CASE 3
     GOTO menu4       'menu2 es el menu de selección de técnica
                      'de control a ser usada
   CASE ELSE
     GOTO menu5      'menu5 es el menu de P.I.D. multivariable
END SELECT
clave1:
COLOR 15, 1
CLS
PRINT
INPUT "Orden del sistema"; n
PRINT
INPUT "Número de entradas"; m
PRINT
INPUT "Número de salidas"; m1
PRINT

IF m <> m1 THEN
  CLS
  BEEP
  COLOR 4, 7
  LOCATE 11, 13
  PRINT "Este programa requiere que el número de entradas sea igual"
  LOCATE 12, 13
  PRINT "al número de salidas, lo cual no se cumple, pues se tiene "
  LOCATE 13, 13
  PRINT m; " entradas     y  "; m1; " salidas
  CALL retardo
  COLOR 15, 1
  GOTO menu5
END IF
```

```
REDIM A(n, n), b(n, m), c(m, n), acon(n, n), bcon(n, m)
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
REDIM vector(m), matriz1#(m, 3)
'¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
CLS
PRINT "Ingreso de datos del sistema; matrices A,B,C"
PRINT
PRINT "Ingreso de datos"; opcion111$; ".- matriz A"
PRINT
FOR i = 1 TO n
  FOR j = 1 TO n
    PRINT "A("; i; ","; j; ")="; : INPUT A(i, j)
    acon(i, j) = A(i, j)
  NEXT
NEXT
PRINT
PRINT "Ingreso de datos"; opcion111$; ".- matriz B"
PRINT
FOR i = 1 TO n
  FOR j = 1 TO m
    PRINT "B("; i; ","; j; ")="; : INPUT b(i, j)
    bcon(i, j) = b(i, j)
  NEXT
NEXT
PRINT
PRINT "Ingreso de datos"; opcion111$; ".- matriz C"
FOR i = 1 TO m
  FOR j = 1 TO n
    PRINT "C("; i; ","; j; ")="; : INPUT c(i, j)
  NEXT
NEXT
PRINT
INPUT "Periodo de muestreo ...................... T"; t
IF opcion111$ = " continuos " THEN
  CALL discret(n, m, t, A(), b())
END IF
'----------------------------------------------------------------------
PRINT
opcion$ = "n"
INPUT "Desea reingresar valores del sistema (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO clave1
END IF
'----------------------------------------------------------------------

REDIM a11(m, 5), b11(m, 5), c11(m, 5), d11(m, 5)
otrocomp2:
GOSUB otrocomp1

REDIM r11(m), x111(n)


'A continuación se realiza el ingreso de las referencias del sistema

GOSUB referencia2

GOSUB valorinicial2
```

```
PRINT
opcion$ = "n"
INPUT "Desea reingresar valores de control (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO otrocomp2
  END IF
'-----------------------------------------------------------------------
nuevamuest2:

INPUT "Número de iteraciones deseado"; l


'A continuación se realiza el ingreso de las referencias del sistema

REDIM y11(m, l), u11(m, l), e11(m, l), x11(n, l)'****, r11(m)
REDIM xaux(n, 1), yaux(m, 1), uaux(m, 1), xaux1(n, 1), bu(n, 1)

'--------------------------------------------------------------------------
'A continuación se realiza el cálculo de los valores de simulación para el
'control P.I.D. de un sistema multivariable
'--------------------------------------------------------------------------

pid2:

FOR k = 1 TO l

   IF k = 1 THEN
      FOR i = 1 TO n
         x11(i, k) = x111(i)
         xaux(i, 1) = x111(i)
      NEXT
   ELSE
      FOR i = 1 TO n
        x11(i, k) = xaux(i, 1)
      NEXT
   END IF

   CALL mulmatrix(m, n, 1, c(), xaux(), yaux())

   FOR j = 1 TO m
     '------------------- cálculo de la salida y -----------------------

      y11(j, k) = yaux(j, 1)

     '------------------- cálculo del error e ---------------------------

      e11(j, k) = r11(j) - y11(j, k)

     '----------------- cálculo de la ley de control u ------------------
      u11(j, k) = 0
      FOR i = 1 TO 5
        IF k - i > 0 THEN
           u11(j, k) = u11(j, k) + c11(j, i) * u11(j, k - i)
        END IF
      NEXT
      FOR i = 1 TO 5
        IF k - i + 1 > 0 THEN
          u11(j, k) = u11(j, k) + d11(j, i) * e11(j, k - i + 1)
        END IF
      NEXT
```

```
      uaux(j, 1) = u11(j, k)
    NEXT
    CALL mulmatrix(n, n, 1, A(), xaux(), xaux1())
    CALL mulmatrix(n, m, 1, b(), uaux(), bu())
    CALL summatrix(n, 1, xaux1(), bu(), xaux())
NEXT
'*************************** importante ******************************
  SCREEN 9
  VIEW (45, 45)-(630, 300), 1, 1
'*********************************************************************

GOTO repita


'**********************************************************************
'********************** REGULADOR CUADRATICO LINEAL ******************
'**********************************************************************
regcuadlineal:

COLOR 15, 1
CLS
PRINT "REGULADOR CUADRATICO LINEAL"
PRINT
INPUT "Orden del sistema"; n
PRINT
INPUT "Numero de entradas"; m
'----------------------------------------------------------------------
'luego se procede a dimensionar las respectivas matrices ha usarse
'----------------------------------------------------------------------
REDIM A(n, n), b(n, m), m#(n, m), q(n, n), r(m, m), acon(n, n), bcon(n, m
REDIM k1#(n, n)
REDIM mt#(m, n), bt(m, n), at(n, n), btk(m, n), btkb(m, m)
REDIM b1(m, m), b1i(m, m), btka(m, n), c1(m, n), g1(m, n), c1t(n, m)
REDIM c1tg(n, n), atk(n, n), atka(n, n), d(n, n), k0#(n, n), xaux(n, 1)
'
'******************************
REDIM offset(m, 1), x111(n)
'******************************
'
PRINT
PRINT "Ingreso de datos del sistema a ser realimentado,matrices A,B"
PRINT
PRINT "Ingreso de datos"; opcion111$; ".- matriz A"
PRINT
  FOR i = 1 TO n
    FOR j = 1 TO n
      PRINT "A("; i; ","; j; ")="; : INPUT A(i, j)
      acon(i, j) = A(i, j)
    NEXT
  NEXT
'
PRINT
PRINT "Ingreso de datos"; opcion111$; ".- matriz B"
PRINT
  FOR i = 1 TO n
    FOR j = 1 TO m
      PRINT "B("; i; ","; j; ")="; : INPUT b(i, j)
      bcon(i, j) = b(i, j)
    NEXT
  NEXT
```

```
PRINT
PRINT
INPUT "Periodo de muestreo"; t
'-----------------------------------------------------------------------
PRINT
opcion$ = "n"
INPUT "Desea reingresar valores del sistema (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO regcuadlineal
END IF
'-----------------------------------------------------------------------

IF opcion111$ = " continuos " THEN
  CALL discret(n, m, t, A(), b())
END IF

'
'*****************************
nuevasponderacion:
CLS
PRINT
PRINT "Ingreso de datos de las matrices de ponderación(Control Optimo)"
PRINT
PRINT "Ingreso de matriz de ponderacion simetrica semidefinida (+) Q"
PRINT
  FOR i = 1 TO n
    FOR j = 1 TO n
      PRINT "Q("; i; ","; j; ")="; : INPUT q(i, j)
    NEXT
  NEXT
'
PRINT
PRINT "Ingreso de matriz de ponderacion M"
PRINT
  FOR i = 1 TO n
    FOR j = 1 TO m
      PRINT "M("; i; ","; j; ")="; : INPUT m#(i, j)
    NEXT
  NEXT
'
PRINT
PRINT "Ingreso de matriz de ponderacion simetrica definida (+) R"
PRINT
  FOR i = 1 TO m
    FOR j = 1 TO m
      PRINT "R("; i; ","; j; ")="; : INPUT r(i, j)
    NEXT
  NEXT
'-----------------------------------------------------------------------
PRINT
opcion$ = "n"
INPUT "Desea reingresar valores de matrices de ponderación (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO nuevasponderacion
END IF
'-----------------------------------------------------------------------

PRINT
PRINT "Valores iniciales de los estados:"
PRINT
```

```
GOSUB valorinicial2

GOSUB nuevooffset

'=============================================================================
'En esta parte del programa se redimensionan las respectivas matrices
'
nuevaiterac:
INPUT "numero de iteraciones"; l
REDIM k#(n, n, l + 1), g(m, n, l)
  FOR i = 1 TO n
    FOR j = 1 TO n
      k#(i, j, l + 1) = 0
    NEXT
  NEXT
'-----------------------------------------------------------------------------
'a continuacion se realizara un calculo iterativo de las matrices K(i)
'y de la matriz de ganancia de realimentacion de estado G(i)
'-----------------------------------------------------------------------------
    CALL transmatrix(n, m, m#(), mt#())
    CALL transmatrix(n, m, b(), bt())
    CALL transmatrix(n, n, A(), at())
'
nuevaiterac2:
FOR k = l TO 1 STEP -1
      FOR i = 1 TO n
        FOR j = 1 TO n
          k1#(i, j) = k#(i, j, k + 1)
        NEXT
      NEXT
    CALL mulmatrix(m, n, n, bt(), k1#(), btk())
    CALL mulmatrix(m, n, m, btk(), b(), btkb())
    CALL summatrix(m, m, r(), btkb(), b1())
    CALL invermatrix(m, b1(), b1i(), ntr)
'-----------------------------------------------------------------------------
'A continuación se identifica que no existe inversa de la matriz
IF ntr = 0 THEN
   CLS
   BEEP
   COLOR 4, 7
   LOCATE 11, 27
   PRINT "Error en inversa       "
   LOCATE 12, 27
   PRINT "Modificar la matriz R"
   CALL retardo
   COLOR 15, 1
   CLS
   LOCATE 5
   PRINT "Ingreso de matriz simétrica definida (+) R"
   PRINT
   FOR i = 1 TO m
     FOR j = 1 TO m
        PRINT "R("; i; ","; j; ")="; : INPUT r(i, j)
     NEXT
   NEXT
   GOTO nuevaiterac2
END IF
```

```
'-------------------------------------------------------------------------
'-------------------------------------------------------------------------
'a continuacion se realizan las funciones del segundo término de G(i)
'-------------------------------------------------------------------------
    CALL mulmatrix(m, n, n, btk(), A(), btka())
    CALL summatrix(m, n, mt#(), btka(), c1())
    CALL mulmatrix(m, m, n, b1i(), c1(), g1())
      FOR i = 1 TO m
        FOR j = 1 TO n
          g(i, j, k) = g1(i, j)
        NEXT
      NEXT
    CALL transmatrix(m, n, c1(), c1t())
    CALL mulmatrix(n, m, n, c1t(), g1(), c1tg())
    CALL mulmatrix(n, n, n, at(), k1#(), atk())
    CALL mulmatrix(n, n, n, atk(), A(), atka())
    CALL summatrix(n, n, q(), atka(), d())
    CALL submatrix(n, n, d(), c1tg(), k0#())
      FOR i = 1 TO n
        FOR j = 1 TO n
          k#(i, j, k) = k0#(i, j)
        NEXT
      NEXT
NEXT
'
'-------------------------------------------------------------------------
'En las siguientes lineas del programa se detalla la forma en que se
'obtiene tanto los valores de los distintos estados como de la diferente
'o diferentes leyes de control de las que se dispone
'-------------------------------------------------------------------------
'
REDIM uaux(m, 1), bu(n, 1), ax(n, 1), gaux(m, n)',******* xaux(n, 1)
'*************************************
REDIM uaux1(m, 1)
'*************************************
REDIM x(n, 1, 1), u(m, 1, 1)

regulador1:

   FOR i = 1 TO m
     FOR j = 1 TO n
       gaux(i, j) = g(i, j, 1)
     NEXT
   NEXT

FOR k = 1 TO 1

    IF k = 1 THEN
      FOR i = 1 TO n
         x(i, 1, k) = x111(i)
         xaux(i, 1) = x111(i)
      NEXT
    ELSE
      FOR i = 1 TO n
         x(i, 1, k) = xaux(i, 1)
      NEXT
    END IF

   CALL mulmatrix(m, n, 1, gaux(), xaux(), uaux())
   CALL negmatrix(m, 1, uaux(), uaux1())
```

```
'***********************************************************************
   CALL summatrix(m, 1, offset(), uaux1(), uaux())
'***********************************************************************
     FOR i = 1 TO m
      u(i, 1, k) = uaux(i, 1)
     NEXT
   CALL mulmatrix(n, m, 1, b(), uaux(), bu())
   CALL mulmatrix(n, n, 1, A(), xaux(), ax())
   CALL summatrix(n, 1, ax(), bu(), xaux())
NEXT
'-----------------------------------------------------------------------
'a continuación se realiza las rutinas correspondientes a los distintos
'gráficos que desee visualizar el usuario del presente programa
'-----------------------------------------------------------------------
'
SCREEN 9
VIEW (45, 45)-(630, 300), 1, 1
repita2:
VIEW PRINT 1 TO 3
CLS 2
PRINT "Desea gráficos de estados(1),ley de control(2),"
PRINT "Ganancia de realimentación(3),abandonar programa (4)";
INPUT numero

IF numero < 1 OR numero > 4 THEN
'si no se especifica en forma adecuada se retorna al menu de gráficos
     GOTO repita2

ELSE
   opcion22$ = "REGULADOR CUADRATICO LINEAL"
   REDIM auxiliar(l)
   IF numero = 1 THEN
   '-----------a continuación: grafico correspondientes a estados--------
estado2:
     opcion33$ = "Estado       x"
     CLS 0
     CLS 2
     VIEW (45, 45)-(630, 300), 1, 1
     PRINT "Estado que desea graficar (valor desde 1 hasta"; n; ")"; :
     INPUT nestado
        IF nestado < 1 OR nestado > n THEN
           GOTO estado2
        END IF
        nopcion = nestado
       FOR i = 1 TO l
         auxiliar(i) = x(nestado, 1, i)
       NEXT
     GOSUB ventana
     GOSUB pantalla
     GOTO repita2

   ELSEIF numero = 2 THEN
   '--------a continuación:grafico correspondiente a ley de control-------
leycontrol2:
   opcion33$ = "Ley de control        u"
   CLS 0
   CLS 2
   VIEW (45, 45)-(630, 300), 1, 1
```

```
      PRINT "Ley de control que desea graficar (valor desde 1 hasta"; m; "
      INPUT nentrada
         IF nentrada < 1 OR nentrada > m THEN
            GOTO leycontrol2
         END IF
         nopcion = nentrada
        FOR i = 1 TO l
          auxiliar(i) = u(nentrada, 1, i)
        NEXT
      GOSUB ventana
      GOSUB pantalla
      GOTO repita2

    ELSEIF numero = 4 THEN
      CLS 0
      VIEW (45, 45)-(630, 300), 1, 1

    ELSE
    '---------a continuación:grafico correspondiente a realimentacón ------
    '---------de estado-----------------------------------------------------
ganrealimen:
    opcion33$ = "Ganancia de realimentación     G"
    CLS 0
    CLS 2
    VIEW (45, 45)-(630, 300), 1, 1
             PRINT "Especificar ganancia relativa a estado y entrada"
             PRINT "Ganancia relativa a estado (valor desde 1 hasta "; n; "
             INPUT ncolumna
               IF ncolumna < 1 OR ncolumna > n THEN
                  GOTO ganrealimen
               END IF
             PRINT "Ganancia relativa a entrada (valor desde 1 hasta"; m; "
             INPUT nfila
               IF nfila < 1 OR nfila > m THEN
                  GOTO ganrealimen
               END IF
               nopcion = nfila * 10 + ncolumna
               FOR i = 1 TO l
                 auxiliar(i) = g(nfila, ncolumna, i)
               NEXT
             GOSUB ventana
             GOSUB pantalla
             GOTO repita2

           END IF
END IF
'-------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea cambiar el número de iteraciones (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOTO nuevaiterac
END IF

'--------------------------------------------------------------------
opcion$ = "n"
INPUT "Desea ingresar nuevos valores de offset (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOSUB nuevooffset
  GOTO regulador1
END IF
'----------------------------
```

```
opcion$ = "n"
INPUT "Desea ingresar nuevos valores iniciales de los estados (s/n)"
; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOSUB valorinicial2
  GOTO regulador1
END IF
'------------------------------------------------------------------------
opcion$ = "n"
PRINT "Desea ingresar nuevos valores de las matrices de ponderación (s/n)
INPUT opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
   SCREEN 0
   COLOR 15, 1
   GOTO nuevasponderacion
END IF
'------------------------------------------------------------------------
'------------------------------------------------------------------------
'La siguiente subrutina permite imprimir en papel los resultados totales
'proceso tales como las matrices del sistema , las matrices de ponderació
'y los resultados de la simulación

opcion$ = "n"
INPUT "Desea resultados completos en papel (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  OPEN "lpt1:" FOR OUTPUT AS #1
     GOSUB rqlsist
     PRINT #1,
     PRINT #1, opcion22$
     PRINT #1,
     PRINT #1, "Valores resultantes de la simulación:"
     PRINT #1,
     PRINT #1, "Periodo de muestreo (s) .......... T="; t
     PRINT #1, "Número de iteraciones ........... N="; l
     PRINT #1, "Tiempo total del proceso (s) ...... t="; l * t
     PRINT #1,
     PRINT #1, "Matriz de ganancias de realimentación optima"
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; g(i, j, 1);
       NEXT
       PRINT #1,
     NEXT

     PRINT #1,
     PRINT #1, "Valores de offset utilizados"
     FOR i = 1 TO m
       PRINT #1, "offset"; i; "="; offset(i, 1)
     NEXT
     PRINT #1,
     PRINT #1, "Valores de los estados"
     FOR i = 1 TO n
       PRINT #1,
       PRINT #1, "Estado       x"; i
       FOR j = 1 TO l
         auxiliar(j) = x(i, 1, j)
       NEXT
       GOSUB impresora
     NEXT
```

```
    PRINT #1,
    PRINT #1, "Valores de las leyes de control"
    FOR i = 1 TO m
      PRINT #1,
      PRINT #1, "Ley de control       u"; i
      FOR j = 1 TO l
        auxiliar(j) = u(i, 1, j)
      NEXT
      GOSUB impresora
    NEXT

  CLOSE #1
END IF
'************************************************************************
'************************************************************************
'La siguiente parte del programa nos da la posibilidad de poder almacena
'los datos resultantes de la simulación para poder ser utilizados median
'las diversas opciones de gráficos que nos proporciona el paquete de hoj
'de cálculo LOTUS.(El archivo de datos será almacenado con extensión .PR

CLS 2
opcion$ = "n"
PRINT "Desea crear archivo de datos de los resultados obtenidos para"
INPUT "graficación posterior en papel desde LOTUS  (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN  '=============================
   INPUT "Path y nombre de archivo (sin extensión)"; opcion$
   OPEN opcion$ + ".prn" FOR OUTPUT AS #1
       PRINT #1, 0;
        FOR i = 1 TO n
         PRINT #1, i;
        NEXT
        FOR i = 1 TO m
         PRINT #1, i;
        NEXT
       PRINT #1,
       FOR j = 1 TO l
         PRINT #1, j;
         FOR i = 1 TO m
           PRINT #1, x(i, 1, j);
         NEXT
         FOR i = 1 TO m
           PRINT #1, u(i, 1, j);
         NEXT
         PRINT #1,
       NEXT
   CLOSE #1
END IF                                       '=============================

'************************************************************************
'************************************************************************

SCREEN 0
COLOR 15, 1
CLEAR
GOTO menu2
fin:
COLOR 15, 0
CLS
END
```

```
'-----------------------------------------------------------------------------------
'
'==============
'---------la siguiente subrutina calcula la ventana adecuada que----------
'---------permitan graficar en forma convenientemente de acuerdo----------
'---------a los parámetros de window; además se inicializa en el----------
'---------valor inicial del  gráfico  correspondiente ; por otro----------
'---------lado, se dibujan los ejes en los respectivos  gráficos----------
ventana:
     SCREEN 0
     CALL maxmin(l, auxiliar(), vmax, vmin, delta)
     SCREEN 9
     VIEW (45, 45)-(630, 300), 1, 1
     WINDOW (-1, vmin - .1 * delta)-(l + 2, vmax + .1 * delta)
     GOSUB marcas
     GOSUB nummarcas
     '---------se dibujan los ejes---------
     LINE (-1, 0)-(l + 2, 0), , , &H8888
     LINE (0, vmax + .1 * delta)-(0, vmin - .1 * delta), , , &H8888
     '------------------------------------
     PSET (0, auxiliar(1))
       FOR k = 1 TO l
         LINE -(k - 1, auxiliar(k)), 15
       NEXT
RETURN
'===============
'subrutina que almacena los valores correspondientes de la matriz B*
'
salida1:
         dd%(mm) = k
         FOR i = 1 TO m
           bbb(mm, i) = cab(1, i)
         NEXT
         RETURN
'------------------------------------------------------------------------------------
'La siguientes rutinas del programa,tienen por objetivo, marcar los
'respectivos ejes de coordenadas
'------------------------------------------------------------------------------------
marcas:
       delta11 = vmin - .1 * delta
       delta12 = (l + 2) / 4
       PSET (0, delta11)
       LINE -STEP(0, .025 * delta)
       PSET (delta12, delta11)
       LINE -STEP(0, .025 * delta)
       PSET (2 * delta12, delta11)
       LINE -STEP(0, .025 * delta)
       PSET (3 * delta12, delta11)
       LINE -STEP(0, .025 * delta)
      ' PSET (-1,ABS(vmax)- ABS(vmin) + .2 * delta)
       PSET (-1, delta / 2 + vmin)
       LINE -STEP(.01 * (l + 3), 0)
RETURN

nummarcas:
          '---- Escala horizontal
         LOCATE 23, 5
         PRINT 0
```

```
        LOCATE 23, 23
        PRINT USING "####"; l / 4
        LOCATE 23, 41
        PRINT USING "####"; l / 2
        LOCATE 23, 59
        PRINT USING "####"; l * 3 / 4
        IF l / 100 < 1 THEN
          LOCATE 23, 77
        ELSE
          IF l / 100 < 10 THEN
            LOCATE 23, 76
          ELSE
            LOCATE 23, 75
          END IF
        END IF
        PRINT l
         '---- Escala vertical
         LOCATE 22
         PRINT USING "##.##"; vmin - .1 * delta
         LOCATE 13
         PRINT USING "##.##"; delta / 2 + vmin
         LOCATE 4
         PRINT USING "##.##"; vmax + .1 * delta
RETURN

otrocomp1:
FOR j = 1 TO m
otrocomp3:
   CLS
   LOCATE 3, 11
   PRINT "C O M P E N S A D O R "; j
   PRINT
   PRINT "Desea ensayar compensador P.I.D. (1)"
   PRINT "u otro compensador (2)";
   INPUT opcion
   '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
   vector(j) = opcion
   '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
   SELECT CASE opcion
      CASE 1
         PRINT
         PRINT "Parámetros del compensador P.I.D. #"; j
         PRINT
         INPUT "Ganancia proporcional................. Kp"; gankp
         INPUT "Ganancia integral .................... Ki"; ganki
         INPUT "Ganancia derivativa .................. Kd"; gankd
        '** INPUT "Periodo de muestreo ................... T"; t
        '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
         matriz1#(j, 1) = gankp
         matriz1#(j, 2) = ganki
         matriz1#(j, 3) = gankd
        '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
         c11(j, 1) = 1
         d11(j, 1) = gankp + (ganki * t) / 2 + gankd / t
         d11(j, 2) = (ganki * t) / 2 - gankp - (2 * gankd) / t
         d11(j, 3) = gankd / t
```

```
      CASE 2
         PRINT
         PRINT "Ecuación de diferencias del compensador"; j
         INPUT "Orden de la ley de control u"; nn11
         INPUT "Orden del error e"; nn22
         IF nn11 > nn22 THEN
          CLS
          BEEP
          COLOR 4, 7
          LOCATE 12, 15
          PRINT "Error,°numerador>°denominador en ecuación del compensador
          ; j
          CALL retardo
          COLOR 15, 1
          GOTO otrocomp3
         END IF
         PRINT
         PRINT "Compensador"; j; ".-Coeficientes de retardo en u"
         FOR i = 1 TO nn11
           PRINT "Coeficiente de retardo en u(k-"; i; ")="; :
           INPUT c11(j, i)
         NEXT
         PRINT
         FOR i = 1 TO nn22 + 1
           PRINT "Coeficiente de retardo en e(k-"; i - 1; ")="; :
           INPUT d11(j, i)
         NEXT
      CASE ELSE
         GOTO otrocomp3
      END SELECT
NEXT
RETURN

'-----------------------------------------------------------------------
referencia:
FOR i = 1 TO m
 PRINT "referencia"; i; : INPUT r11(i)
NEXT
RETURN

'----------------------------------------------------------------------
valorinicial1:
'A continuación, condiciones iniciales
FOR i = 1 TO m
  PRINT "y"; i; "(0)="; : INPUT y111(i)
NEXT
RETURN
'-----------------------------------------------------------------------
referencia2:
FOR i = 1 TO m
 PRINT "referencia"; i; : INPUT r11(i)
NEXT
RETURN
'-----------------------------------------------------
valorinicial2:
'A continuación, condiciones iniciales de los estados
FOR i = 1 TO n
  PRINT "x"; i; "(0)="; : INPUT x111(i)
NEXT
RETURN
'-----------------------------------------------------
```

```
nuevooffset:
PRINT "Valores de offset"
FOR i = 1 TO m
  PRINT "offset "; i; : INPUT offset(i, 1)
NEXT
RETURN
'-------------------------------------------------------------
referenciaext:
FOR ii = 1 TO m
  PRINT "ref. ext."; ii; : INPUT v(ii)
NEXT
RETURN
'-----------------------------------------------------------------------
resulpantalla:
SCREEN 0
COLOR 15, 1
CLS 0
LOCATE 3, 18
PRINT opcion22$
LOCATE 6, 18
PRINT "Valores de  "; opcion33$; nopcion
LOCATE 8, 18
PRINT "Valor inicial .................... Vo"; : PRINT USING "#####.#####"
auxiliar(1)
LOCATE 10, 18
PRINT "Valor final ...................... Vf"; : PRINT USING "#####.#####"
auxiliar (1)
LOCATE 12, 18
PRINT "Valor máximo .................. Vmax"; : PRINT USING "#####.#####"
vmax
LOCATE 14, 18
PRINT "Valor mínimo .................. Vmin"; : PRINT USING "#####.#####"
vmin1
LOCATE 16, 18
PRINT "Periodo de muestreo (s) ......... T"; : PRINT USING "#####.###"; t
LOCATE 18, 18
PRINT "Número de iteraciones .......... N"; : PRINT USING "#######"; l
LOCATE 20, 18
PRINT "Tiempo total (s) ................ t"; : PRINT USING "#####.##";
l * t
LOCATE 22, 18
opcion$ = "n"
INPUT "Presione RETURN para continuar "; opcion$
SCREEN 9
VIEW (45, 45)-(630, 300), 1, 1
RETURN
'-----------------------------------------------
pantalla:
VIEW PRINT 2 TO 2
opcion$ = "n"
INPUT "Desea resultados en pantalla (s/n)"; opcion$
IF opcion$ = "s" OR opcion$ = "S" THEN
  GOSUB resulpantalla
END IF
RETURN
'-----------------------------------------------------------------------
'La siguiente subrutina tiene por objeto imprimir en papel los valores
'de las matrices del sistema original como de las matrices M que per-
'miten la asignación de polos en desacoplamiento
```

```
desacopsist:
     PRINT #1,
     PRINT #1, "DESACOPLAMIENTO"
     PRINT #1,
     PRINT #1, "SISTEMA ORIGINAL NO DESACOPLADO"
     PRINT #1,
     PRINT #1, "Orden del sistema  :"; n
     PRINT #1, "Numero de salidas  :"; m
     PRINT #1, "Numero de entradas :"; m
     PRINT #1,
     PRINT #1, "Matriz continua A"
     PRINT #1,
     FOR i = 1 TO n
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; acon(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1, "Matriz continua B"
     PRINT #1,
     FOR i = 1 TO n
       FOR j = 1 TO m
         PRINT #1, USING "#####.#####"; bcon(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1, "Matriz continua C"
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; c(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1,
     PRINT #1, "Matrices de asignación de polos  Mk"
     PRINT #1,
     FOR k = 0 TO ndelta1
     PRINT #1, "Matriz   M"; k
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO m
         PRINT #1, USING "#####.#####"; maux#(i, j, k + 1);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1,
     NEXT
     PRINT #1,
     PRINT #1, "MATRICES DE DESACOPLAMIENTO"
     PRINT #1,
     PRINT #1, "Matriz  de realimentación de estado F1"
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; f1(i, j);
       NEXT
       PRINT #1,
     NEXT
```

```
     PRINT #1,
     PRINT #1, "Matriz  de preamplificación E"
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO m
         PRINT #1, USING "#####.#####"; e(i, j);
       NEXT
       PRINT #1,
     NEXT
       PRINT #1,
IF opcion11$ = "control" THEN

       PRINT #1, "COMPENSACION"
    FOR j = 1 TO m
       PRINT #1,
       PRINT #1, "Compensador"; j
       PRINT #1,
       IF vector(j) = 1 THEN
          PRINT #1, "Ganancia proporcional ......... Kp=";
          PRINT #1, USING "#####.#####"; matriz1#(j, 1)
          PRINT #1, "Ganancia integral ............. Ki=";
          PRINT #1, USING "#####.#####"; matriz1#(j, 2)
          PRINT #1, "Ganancia derivativa ........... Kd=";
          PRINT #1, USING "#####.#####"; matriz1#(j, 3)
       ELSEIF vector(j) = 2 THEN
          PRINT #1, "Coeficientes de retardo en u"
          PRINT #1,
          FOR i = 1 TO 5
            PRINT #1, "Coeficiente de retardo en u(k-"; i; ")=";
            PRINT #1, c11(j, i)
          NEXT
          PRINT #1,
          PRINT #1, "Coeficientes de retardo en e"
          PRINT #1,
          FOR i = 1 TO 5
            PRINT #1, "Coeficiente de retardo en e(k-"; i - 1; ")=";
            PRINT #1, d11(j, i)
          NEXT
       END IF
    NEXT
END IF

RETURN

'---------------------------------------------------------------------------
'La siguiente subrutina permite imprimir en papel los valores correspon-
'dientes al vector auxiliar() que contendrá los valores de los items
'deseados

impresora:

    CALL maxmin(l, auxiliar(), vmax, vmin, delta)
    PRINT #1, "Valor inicial ................... Vo";
    PRINT #1, USING "#####.#####"; auxiliar(1)
    PRINT #1, "Valor final ..................... Vf";
    PRINT #1, USING "#####.#####"; auxiliar(l)
    PRINT #1, "Valor máximo ................. Vmax";
    PRINT #1, USING "#####.#####"; vmax
    PRINT #1, "Valor mínimo ................. Vmin";
    PRINT #1, USING "#####.#####"; vmin1
```

```
'La siguiente rutina tiene por objeto imprimir en papel los valores de la
'matrices del sistema como de las diversas matrices de ponderación utili-
'zadas en el REGULADOR CUADRATICO LINEAL

rqlsist:
     PRINT #1,
     PRINT #1, "REGULADOR CUADRATICO LINEAL"
     PRINT #1,
     PRINT #1, "SISTEMA ORIGINAL NO REGULADO"
     PRINT #1,
     PRINT #1, "Orden del sistema  :"; n
     PRINT #1, "Numero de entradas  :"; m
     PRINT #1,
     IF opcion111$ = " continuos " THEN
        PRINT #1, "Matriz continua  A"
        PRINT #1,
        FOR i = 1 TO n
          FOR j = 1 TO n
            PRINT #1, USING "#####.#####"; acon(i, j);
          NEXT
          PRINT #1,
        NEXT
        PRINT #1, "Matriz continua B"
        PRINT #1,
        FOR i = 1 TO n
          FOR j = 1 TO m
            PRINT #1, USING "#####.#####"; bcon(i, j);
          NEXT
          PRINT #1,
        NEXT

     ELSEIF opcion111$ = " discretos " THEN

        PRINT #1, "Matriz discreta   A"
        PRINT #1,
        FOR i = 1 TO n
          FOR j = 1 TO n
            PRINT #1, USING "#####.#####"; A(i, j);
          NEXT
          PRINT #1,
        NEXT
        PRINT #1, "Matriz discreta   B"
        PRINT #1,
        FOR i = 1 TO n
          FOR j = 1 TO m
            PRINT #1, USING "#####.#####"; b(i, j);
          NEXT
          PRINT #1,
        NEXT

     END IF

     PRINT #1,
     PRINT #1, "MATRICES DE PONDERACION (CONTROL OPTIMO)"
     PRINT #1,
     PRINT #1, "Matriz de ponderacion simetrica semidefinida (+) Q"
     PRINT #1,
```

```
        FOR i = 1 TO n
          FOR j = 1 TO n
            PRINT #1, USING "#####.#####"; q(i, j);
          NEXT
          PRINT #1,
        NEXT
      PRINT #1,
      PRINT #1, "Matriz de ponderacion M"
      PRINT #1,
        FOR i = 1 TO n
          FOR j = 1 TO m
            PRINT #1, USING "#####.#####"; m#(i, j);
          NEXT
          PRINT #1,
        NEXT
      PRINT #1,
      PRINT #1, "Matriz de ponderacion simetrica definida (+) R"
      PRINT #1,
        FOR i = 1 TO m
          FOR j = 1 TO m
            PRINT #1, USING "#####.#####"; r(i, j);
         NEXT
         PRINT #1,
        NEXT
RETURN
'----------------------------------------------------------------------
'----------------------------------------------------------------------
'La siguiente subrutina tiene por objeto presentar en papel la impresión
'de los valores correspondientes al sistema a ser compensado
pidsist:
    PRINT #1,
    PRINT #1, opcion22$
    PRINT #1,
    IF npid1 = 1 THEN '========================================
      FOR j = 1 TO m
         PRINT #1,
         PRINT #1, "L A Z O"; j
         PRINT #1,
         PRINT #1, "Lazo"; j; ".- Coeficientes de retardo en y"
         PRINT #1,
          FOR i = 1 TO 5
            PRINT #1, "Coeficiente de retardo en y(k-"; i; ")=";
            PRINT #1, a11(j, i)
          NEXT
         PRINT #1,
         PRINT #1, "Lazo"; j; ".- Coeficientes de retardo en u"
         PRINT #1,
         FOR i = 1 TO 5
           PRINT #1, "Coeficiente de retardo en u(k-"; i; ")=";
           PRINT #1, b11(j, i)
         NEXT
      NEXT

    ELSEIF npid1 = 2 THEN '=========================

     PRINT #1,
     PRINT #1, "SISTEMA ORIGINAL NO CONTROLADO"
     PRINT #1,
```

```
  PRINT #1, "Orden del sistema  :"; n
  PRINT #1, "Numero de entradas :"; m
  PRINT #1, "Numero de salidas  :"; m
  PRINT #1,
  IF opcion111$ = " continuos " THEN '---------------
     PRINT #1, "Matriz continua   A"
     PRINT #1,
     FOR i = 1 TO n
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; acon(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1, "Matriz continua B"
     PRINT #1,
     FOR i = 1 TO n
       FOR j = 1 TO m
         PRINT #1, USING "#####.#####"; bcon(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1, "Matriz continua C"
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; c(i, j);
       NEXT
       PRINT #1,
     NEXT

  ELSEIF opcion111$ = " discretos " THEN '-----------------

     PRINT #1, "Matriz discreta   A"
     PRINT #1,
     FOR i = 1 TO n
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; A(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1, "Matriz discreta   B"
     PRINT #1,
     FOR i = 1 TO n
       FOR j = 1 TO m
         PRINT #1, USING "#####.#####"; b(i, j);
       NEXT
       PRINT #1,
     NEXT
     PRINT #1, "Matriz discreta   C"
     PRINT #1,
     FOR i = 1 TO m
       FOR j = 1 TO n
         PRINT #1, USING "#####.#####"; c(i, j);
       NEXT
       PRINT #1,
     NEXT

  END IF '------------------

END IF '=========================
```

```
      PRINT #1,
      PRINT #1, "COMPENSACION"
    FOR j = 1 TO m
      PRINT #1,
      PRINT #1, "Compensador"; j
      PRINT #1,
      IF vector(j) = 1 THEN
         PRINT #1, "Ganancia proporcional ......... Kp=";
         PRINT #1, USING "#####.#####"; matriz1#(j, 1)
         PRINT #1, "Ganancia integral ............ Ki=";
         PRINT #1, USING "#####.#####"; matriz1#(j, 2)
         PRINT #1, "Ganancia derivativa .......... Kd=";
         PRINT #1, USING "#####.#####"; matriz1#(j, 3)
      ELSEIF vector(j) = 2 THEN
         PRINT #1, "Coeficientes de retardo en u"
         PRINT #1,
         FOR i = 1 TO 5
           PRINT #1, "Coeficiente de retardo en u(k-"; i; ")=";
           PRINT #1, c11(j, i)
         NEXT
         PRINT #1,
         PRINT #1, "Coeficientes de retardo en e"
         PRINT #1,
         FOR i = 1 TO 5
           PRINT #1, "Coeficiente de retardo en e(k-"; i - 1; ")=";
           PRINT #1, d11(j, i)
         NEXT
      END IF
    NEXT

RETURN


DEFINT I-N
DEFDBL A-H, O-Z
SUB determatrix (n, A(), d)
REDIM ab(2 * n - 1, n), p(n), q(n)
FOR i = 1 TO n
  FOR j = 1 TO n
    ab(i, j) = A(i, j)
  NEXT
NEXT
IF n = 2 THEN
    d = A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1)
ELSE
   FOR i = n + 1 TO 2 * n - 1
     k = i - n
     FOR j = 1 TO n
       ab(i, j) = ab(k, j)
     NEXT
   NEXT
   k = 0
   FOR i = 1 TO n
     p(i) = 1
   NEXT
   FOR j = 1 TO n
     FOR i = 1 TO n
       p(i) = p(i) * ab(i + k, j)
     NEXT
     k = k + 1
```

```
    k = 0
    FOR i = 1 TO n
      q(i) = 1
    NEXT
    k = 0
    FOR kt = 1 TO n
      j = n + 1 - kt
      FOR i = 1 TO n
        q(i) = q(i) + ab(i + k, j)
      NEXT
      k = k + 1
    NEXT
    d = 0
    FOR i = 1 TO n
      d = d + p(i) - q(i)
    NEXT
END IF
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB discret (n, nu, t, acon(), bcon())
REDIM iden#(n, n), fi(n, n), fi1(n, n), fi2(n, n)
REDIM tfi(n, n), atfi(n, n), A(n, n), b(n, nu)
FOR i = 1 TO n
  FOR j = 1 TO n
     iden#(i, j) = 0
     fi(i, j) = 0
  NEXT
NEXT
FOR i = 1 TO n
  iden#(i, i) = 1
  fi(i, i) = 1
NEXT

'la subrutina escalmatrix, multiplica un escalar por una matriz
FOR i = 30 TO 2 STEP -1
   CALL escalmatrix(n, n, t / i, acon(), fi1())
   CALL mulmatrix(n, n, n, fi1(), fi(), fi2())
   CALL summatrix(n, n, iden#(), fi2(), fi())
NEXT
'-------------------------------------------

'Cálculo de A discreta:
'-------------------------------------------
CALL escalmatrix(n, n, t, fi(), tfi())
CALL mulmatrix(n, n, n, acon(), tfi(), atfi())
CALL summatrix(n, n, iden#(), atfi(), A())

'-------------------------------------------
'Cálculo de B discreta:
'-------------------------------------------
CALL mulmatrix(n, n, nu, tfi(), bcon(), b())

'-----------------------------------------
'salida da valores
'-----------------------------------------
FOR i = 1 TO n
  FOR j = 1 TO n
    acon(i, j) = A(i, j)
  NEXT
```

```
PRINT
FOR i = 1 TO n
  FOR j = 1 TO nu
    bcon(i, j) = b(i, j)
  NEXT
NEXT

END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB escalmatrix (n, m, escalar, A(), b())
FOR i = 1 TO n
  FOR j = 1 TO m
    b(i, j) = escalar * A(i, j)
  NEXT
NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB expomatrix (n, A(), nexpo, aaux())
'n=orden de la matriz a();nexpo=exponente de a()
REDIM aaux1(n, n)
IF nexpo = 0 THEN
    FOR i = 1 TO n
      FOR j = 1 TO n
        aaux(i, j) = 0
      NEXT
    NEXT
    FOR i = 1 TO n
      aaux(i, i) = 1
    NEXT
ELSEIF nexpo = 1 THEN
    FOR i = 1 TO n
      FOR j = 1 TO n
        aaux(i, j) = A(i, j)
      NEXT
    NEXT
ELSE
    FOR i = 1 TO n
      FOR j = 1 TO n
        aaux(i, j) = A(i, j)
      NEXT
    NEXT
    FOR i = 1 TO nexpo - 1
      CALL mulmatrix(n, n, n, aaux(), A(), aaux1())
    NEXT
    FOR i = 1 TO n
      FOR j = 1 TO n
        aaux(i, j) = aaux1(i, j)
      NEXT
    NEXT
END IF
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB invermatrix (n, b(), bt(), ntr)
```

```
REDIM h(n), k(n)
ntr = 1
l = 0
FOR k = 1 TO n
 IF b(k, k) = 0 THEN GOSUB 200
 g = 1 / b(k, k)
   FOR i = 1 TO n
     IF b(i, k) <> 0 THEN b(i, k) = b(i, k) * g
     NEXT i
     b(k, k) = g
       FOR j = 1 TO n
       IF j = k THEN 54
       g = b(k, j)
       b(k, j) = 0
       FOR i = 1 TO n
       IF g = 0 THEN 52
       b(i, j) = b(i, j) - g * b(i, k)
52     NEXT i
54     NEXT j
       NEXT k
       IF c$ = "s" THEN GOSUB 220
      FOR i = 1 TO n
         FOR j = 1 TO n
         bt(i, j) = b(i, j)
         NEXT
       NEXT
       GOTO 100
200  l = l + 1
     FOR j = k TO n
     IF b(k, j) <> 0 THEN 210
     NEXT j
    BEEP
    PRINT "matriz singular"
    ntr = 0
    GOTO 100
210 h(l) = j
    k(l) = k
    c$ = "s"
    FOR i = 1 TO n
      x = b(i, k)
      b(i, k) = b(i, j)
      b(i, j) = x
    NEXT i
    RETURN
220 FOR i = l TO 1 STEP -1
      FOR j = 1 TO n
      x = b(h(i), j)
      b(h(i), j) = b(k(i), j)
      b(k(i), j) = x
      NEXT
    NEXT
   RETURN
100
END SUB
```

```
DEFINT I-N
DEFDBL A-H, O-Z
SUB maxmin (n, aa(), vmax, vmin, delta)
  vmax = aa(1)
  vmin = aa(1)
    FOR i = 2 TO n
      IF aa(i) > vmax THEN
        vmax = aa(i)
      ELSEIF aa(i) < vmin THEN
        vmin = aa(i)
      ELSE
      END IF
    NEXT
     vmin1 = vmin
      IF vmin > 0 THEN
        vmin = 0
      END IF
    delta = ABS(vmax - vmin)
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB mulmatrix (n, m, l, A(), b(), c())
  FOR i = 1 TO n
    FOR j = 1 TO l
    r = 0
       FOR k = 1 TO m
       r = r + A(i, k) * b(k, j)
       NEXT
    c(i, j) = r
    NEXT
  NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB negmatrix (n, m, A(), an())
 FOR i = 1 TO n
   FOR j = 1 TO m
     an(i, j) = -A(i, j)
   NEXT
 NEXT
END SUB

SUB retardo
j = 0
FOR i = 1 TO 50000
j = j + 1
NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB submatrix (n, m, A(), b(), c())
   FOR i = 1 TO n
     FOR j = 1 TO m
     c(i, j) = A(i, j) - b(i, j)
     NEXT
   NEXT
END SUB
```

```
DEFINT I-N
DEFDBL A-H, O-Z
SUB summatrix (n, m, A(), b(), c())
  FOR i = 1 TO n
    FOR j = 1 TO m
    c(i, j) = A(i, j) + b(i, j)
    NEXT
  NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB transmatrix (n, m, A(), c())
   FOR i = 1 TO m
     FOR j = 1 TO n
     c(i, j) = A(j, i)
     NEXT
   NEXT
END SUB
```

```
DECLARE SUB retardo ()
DECLARE SUB negvector (n%, a#())
DECLARE SUB defgraf ()
DECLARE SUB graf (i%)
DECLARE SUB sumvector (n%, a#(), B#(), c#())
DECLARE SUB mulmatvector (n%, m%, a#(), B#(), c#())
DEFINT I-N
DEFDBL A-H, O-Z

COMMON SHARED utr, ytr, y$, u$, nopciontr3, nopciontr5
COMMON SHARED nopciontr2, nopciontr4, rtr()
COLOR 15, 1
CLS
'-----------------------------------------------------------------
'-----------------------------------------------------------------
'En la siguiente parte del programa se determina si el equipo de
'adquisición de datos y control está o no encendido; en caso de
'estar apagado, se genera una señal de error auditiva y visual
'hacia el usuario; si el equipo está encendido, se pasa a las res-
'pectivas rutinas de tiempo real
'-----------------------------------------------------------------
   DEF SEG = &HCFF9
   IF PEEK(&HB) = 255 THEN
      COLOR 4, 7
      BEEP
      LOCATE 12, 18
      PRINT "Error: Sistema de adquisición apagado";

      CALL retardo

      COLOR 15, 1
      DEF SEG
      SCREEN 0
      CLS
      CLEAR
      GOTO fintr
   END IF
'-----------------------------------------------------------------

CALL softinit                'Estas dos instruciones inicializan el
CALL init                    'software y hardware para tiempo real

menutr1:
COLOR 15, 1
CLS
COLOR 2
LOCATE 5, 16
PRINT "P.I.D. ..................................... (1)"
LOCATE 10, 16
PRINT "REGULADOR CUADRATICO LINEAL ................ (2)"
LOCATE 15, 16
PRINT "DESACOPLAMIENTO ............................ (3)"
LOCATE 20, 16
PRINT "ABANDONAR TIEMPO REAL ...................... (4)"
LOCATE 24, 37
INPUT "Ingrese opcion requerida"; nopciontr1
```

```
'------------------------------------------------------------------------
'La siguiente parte del programa tiene por objeto el ingreso de los
'datos de los diversos parámetros de control según la opción escogida
'------------------------------------------------------------------------

SELECT CASE nopciontr1
   CASE 1        '****************** P.I.D.  DIGITAL *******************
unotr:
     'Ingreso de datos para P.I.D.
     COLOR 15, 1
     CLS
     INPUT "Numero de entradas"; nu
     INPUT "Numero de salidas"; m
     IF nu <> m THEN
       CLS
       BEEP
       COLOR 4, 7
       LOCATE 11, 22
       PRINT "Se requiere en P.I.D. que el número"
       LOCATE 12, 22
       PRINT "de entradas sea igual al número de "
       LOCATE 13, 22
       PRINT "salidas"
       CALL retardo
       GOTO unotr
     END IF

     REDIM e11tr(m, 5), u11tr(m, 5)      'Se dimensiona para cálculo del
                                         'P.I.D. en el algoritmo

'------------------------------------------------------------------------
'------------------------------------------------------------------------
GOSUB resetear                'Resetea los valores almacenados para P.I.D.
'------------------------------------------------------------------------
'------------------------------------------------------------------------

     INPUT "Periodo de muestreo .......... T (seg.)"; t
     REDIM vector(m), matriz1#(m, 3), c11(m, 5), d11(m, 5)
     GOSUB otrocomp1
     REDIM rtr(m)
     PRINT

'------------------------------------------------------------------------
'------------------------------------------------------------------------
GOSUB referenciatr            'Setea valores de referencia
'------------------------------------------------------------------------
'------------------------------------------------------------------------
   opcion$ = "n"
   INPUT "Desea modificar los datos ingresados (s/n)"; opcion$
   IF opcion$ = "s" OR opcion$ = "S" THEN
     GOTO unotr
   END IF
'------------------------------------------------------------------------

   CASE 2        '*********** REGULADOR CUADRATICO LINEAL ***************
dostr:
     'Ingreso de datos de REGULADOR CUADRATICO LINEAL
     COLOR 15, 1
     CLS
```

```
    PRINT
    INPUT "Número de estados"; n
    PRINT
    INPUT "Número de entradas"; nu
    PRINT
    INPUT "Periodo de muestreo ........... T (seg.)"; t
    PRINT "Matriz óptima de realimentación de estado"
    PRINT
    REDIM gtr(nu, n), gytr(nu), rtr(nu)
    FOR i = 1 TO nu
      FOR j = 1 TO n
        PRINT "G óptimo("; i; ","; j; ")=";
        INPUT gtr(i, j)
      NEXT
    NEXT
    PRINT
    m = nu   'permite utilizar única subrutina de referencia
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
GOSUB referenciatr              'Setea valores de referencia
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
  opcion$ = "n"
  INPUT "Desea modificar los datos ingresados (s/n)"; opcion$
  IF opcion$ = "s" OR opcion$ = "S" THEN
    GOTO dostr
  END IF
'------------------------------------------------------------------------------

  CASE 3      '**************** DESACOPLAMIENTO ********************
trestr:
    'Ingreso de datos de DESACOPLAMIENTO
    COLOR 15, 1
    CLS
    PRINT
    INPUT "Número de estados"; n
    PRINT
    INPUT "Número de entradas"; nu
    PRINT
    INPUT "Número de salidas"; m
    PRINT

    REDIM e11tr(m, 5), u11tr(m, 5)      'Se dimensiona para cálculo del
    REDIM utr1(m)                       'P.I.D. en el algoritmo

'------------------------------------------------------------------------------
'------------------------------------------------------------------------------
GOSUB resetear           'Resetea los valores almacenados para P.I.D.
'------------------------------------------------------------------------------
'------------------------------------------------------------------------------

    INPUT "Periodo de muestreo ..................... T (seg.)"; t
     IF nu <> m THEN
       CLS
       BEEP
       COLOR 4, 7
       LOCATE 11, 22
       PRINT "Se requiere en desacoplamiento que"
       LOCATE 12, 22
       PRINT "el número de entradas sea igual al"
```

```
        LOCATE 13, 22
        PRINT "número de salidas                    "
        CALL retardo
        GOTO trestr
      END IF
    REDIM e(m, m), f1(m, n), rtr(m)
    REDIM vector(m), matriz1#(m, 3), c11(m, 5), d11(m, 5)
    PRINT
    PRINT "Matriz de preamplificación E"
    PRINT
    FOR i = 1 TO m
      FOR j = 1 TO m
        PRINT "E("; i; ","; j; "=";
        INPUT e(i, j)
      NEXT
    NEXT
    PRINT
    PRINT "Matriz de realimentación de estado"
    PRINT
    FOR i = 1 TO m
      FOR j = 1 TO n
        PRINT "F1("; i; ","; j; "=";
        INPUT f1(i, j)
      NEXT
    NEXT
    GOSUB otrocomp1
    PRINT
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
GOSUB referenciatr              'Setea valores de referencia
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------

    REDIM eutr1(m), f1ytr(m)      'Para el algoritmo de cálculo de u
'---------------------------------------------------------------------------
  opcion$ = "n"
  INPUT "Desea modificar los datos ingresados (s/n)"; opcion$
  IF opcion$ = "s" OR opcion$ = "S" THEN
    GOTO trestr
  END IF
'---------------------------------------------------------------------------

  CASE 4
    GOTO fintr           'finaliza trabajo en tiempo real
  CASE ELSE
    GOTO menutr1
END SELECT
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
'A continuación se realiza el seteo de los gráficos que irán en las dos
'ventanas de gráficos de acuerdo al deseo del usuario
'---------------------------------------------------------------------------
'---------------------------------------------------------------------------
cuatrotr:
CLS
LOCATE 10
grafico1:
INPUT "Gráfico en ventana 1.- Salida (1), Ley de control (2)"; nopciontr2
SELECT CASE nopciontr2
```

```
    CASE 1
      PRINT "Salida a graficar (valor desde 1 hasta "; m; ")";
      INPUT nopciontr3
      y$ = "y"
    CASE 2
      PRINT "Ley de control a graficar (valor desde 1 hasta "; nu; ")";
      INPUT nopciontr3
      y$ = "u"
    CASE ELSE
      GOTO grafico1
END SELECT
grafico2:
INPUT "Gráfico en ventana 2.- Salida (1), Ley de control (2)"; nopciontr4
SELECT CASE nopciontr4
    CASE 1
      PRINT "Salida a graficar (valor desde 1 hasta "; m; ")";
      INPUT nopciontr5
      u$ = "y"
    CASE 2
      PRINT "Ley de control a graficar (valor desde 1 hasta "; nu; ")";
      INPUT nopciontr5
      u$ = "u"
    CASE ELSE
      GOTO grafico2
END SELECT
'----------------------------------------------------------------------------
   PRINT
   opcion$ = "n"
   INPUT "Desea modificar opción de gráficos elegida (s/n)"; opcion$
   IF opcion$ = "s" OR opcion$ = "S" THEN
     GOTO cuatrotr
   END IF

'----------------------------------------------------------------------------
'----------------------------------------------------------------------------
'A continuación se realiza la parte correspondiente a tiempo real, en la
'cual se inicializa el equipo de adquisición de datos y se realiza el
'cálculo de las respectivas leyes de control
'----------------------------------------------------------------------------
'----------------------------------------------------------------------------
'****************************************************************************
'****************************************************************************


'Estas dos instrucciones inicializan el SOFTWARE y el HARDWARE del sistema
'QUICK500 respectivamente; debe necesariamente estar dentro del programa
'principal y en forma inicial al realizar la parte correspondiente a
'tiempo real


CALL softinit               'Inicialización del sistema para tiempo real
CALL init

'**************************************

'A continuación se setean los valores de muestreo tanto para salida
'como para entrada de los valores análogos
```

```
tmintr1% = 150
bintv% = INT(t * 1000)
uno% = INT(bintv% / tmintr1%)
bouttv% = bintv% / uno%       'bintv%/uno%  ms.
bintv% = uno% * bouttv%

DIM ytr(m), utr(nu)
'***************************************


'-------------------------------------------------------------------------
'-------------------------------------------------------------------------
'La siguiente parte del programa, crea los arreglos para la medida
'de datos desde los diversos canales análogos

'************************************************************************
'************************************************************************
'A continuación creará un arreglo de entrada para ocho canales
'a la vez
    ionl$ = "ANLG0 ANLG1 ANLG2 ANLG3 ANLG4"
    ionl4$ = "ANLG0 ANLG1 ANLG2 ANLG3 ANLG4"
    arn2$ = "entradas%"
    CALL anin(arn2$, 1!, ionl$, bintv%, -1, "nt", "taskstat")

'************************************************************************
'************************************************************************
'A continuación se creará un arreglo para cinco canales de salida
'a la vez

dep! = 1
arn1$ = "salidas%"
ionl$ = "ANOUT0 ANOUT1 ANOUT2 ANOUT3 ANOUT4"
ionl3$ = "ANOUT0 ANOUT1 ANOUT2 ANOUT3 ANOUT4"
CALL ARMAKE(arn1$, dep!, -1, ionl$)

FOR ii = 1 TO nu      '/////////////////////////////////////////////////.

nopcion = ii - 1

SELECT CASE nopcion
   CASE 0
     ionl$ = "ANOUT0"
   CASE 1
     ionl$ = "ANOUT1"
   CASE 2
     ionl$ = "ANOUT2"
   CASE 3
     ionl$ = "ANOUT3"
   CASE 4
     ionl$ = "ANOUT4"
END SELECT

CALL ARPUTVALF(arn1$, dep!, -1, ionl$, 0!, 0) 'asegura primera salida
                                               'en cero
NEXT

CALL ANOUT(arn1$, ionl3$, bouttv%, -1, "nt", "tarea2")

'Comando que se encarga de sacar los valores guardados en el area, una vez
'que se hayan habilitado las interrupciones
```

```
'*************************************************************************
'*************************************************************************

'En las instrucciones anteriores se setea el nombre del canal especificado
'el nombre del arreglo y su profundidad ; de tal modo de ubicar los datos
'previamente a su salida por el canal indicado.


'*************************************************************************
'*************************************************************************
repetirtr:

REDIM tim(0 TO 7) AS INTEGER, val1(0 TO 7) AS DOUBLE
tim(0) = 1      'Resetea el timer0 y habilita su conteo
t12 = 1

CALL inton(1, "mil")
CALL timerstart(tim(), "nt", "timer")

SCREEN 9
WIDTH 80, 43
CALL defgraf
'----------------------------------------------------------------
LOCATE 21, 12
PRINT "Presione barra espaciadora para terminar el control"
'----------------------------------------------------------------
iii = 0
'************
an$ = ""
WHILE an$ = ""             '*****************************inicio lazo
'----------------------------------------------------------------
LOCATE 22, 12
PRINT "Numero de muestreos="; t12
'----------------------------------------------------------------
CALL arlastp("entradas%", lp!)
CALL arlastp("salidas%", lp!)

FOR ii = 1 TO m
   nopcion = ii - 1
   SELECT CASE nopcion
       CASE 0
         ionl$ = "ANLG0"
       CASE 1
         ionl$ = "ANLG1"
       CASE 2
         ionl$ = "ANLG2"
       CASE 3
         ionl$ = "ANLG3"
       CASE 4
         ionl$ = "ANLG4"
   END SELECT

   FOR dep! = 1 TO 1          '****************///////////**********
      CALL argetvalf(arn2$, dep!, -1, ionl$, va!, 0)
   NEXT                       '****************///////////***********
   ytr(ii) = va!

NEXT
```

```
'----------------------------------------------------------------------
'----------------------------------------------------------------------
'----------------------------------------------------------------------
'A continuación se realiza el cálculo de las diversos valores de las res-
'pectivas leyes de control
'----------------------------------------------------------------------
'----------------------------------------------------------------------
SELECT CASE nopciontr1

    CASE 1      '***************** P.I.D.  DIGITAL ******************

      '----------------- Cálculo del error --------------------
      FOR j = 1 TO m
        e11tr(j, 5) = rtr(j) - ytr(j)
      NEXT

      '-------------- Cálculo de la ley de control u ---------------------
    FOR j = 1 TO m
      u11tr(j, 5) = 0
      FOR i = 4 TO 1 STEP -1
        u11tr(j, 5) = u11tr(j, 5) + c11(j, i) * u11tr(j, 5 - i)
      NEXT
      FOR i = 5 TO 1 STEP -1
        u11tr(j, 5) = u11tr(j, 5) + d11(j, i) * e11tr(j, 5 - i + 1)
      NEXT
    NEXT
      '----------- Recuperación de u en vector de salida ----------------
      FOR j = 1 TO m
        utr(j) = u11tr(j, 5)
        '---------- limitación de valores de salida debido a -----------
        '----- limitaciones del equipo de adquisición de datos y control -
        IF utr(j) > 10 THEN
           utr(j) = 10
        ELSEIF utr(j) < 0 THEN
           utr(j) = 0
        END IF
        '----------------------------------------------------------------
      NEXT
      '------------ Actualización de valores para nuevo cálculo ---------
      FOR j = 1 TO m
        FOR i = 1 TO 4
          e11tr(j, i) = e11tr(j, i + 1)
          u11tr(j, i) = u11tr(j, i + 1)
        NEXT
      NEXT

    CASE 2      '********** REGULADOR CUADRATICO LINEAL **************

      CALL mulmatvector(nu, n, gtr(), ytr(), gytr())
      CALL negvector(nu, gytr())
      CALL sumvector(nu, gytr(), rtr(), utr())

      FOR j = 1 TO nu
      '---------- limitación de valores de salida debido a -------------
      '----- limitaciones del equipo de adquisición de datos y control ---
      IF utr(j) > 10 THEN
         utr(j) = 10
      ELSEIF utr(j) < 0 THEN
         utr(j) = 0
      END IF
      '-----------------------------------------
```

```
    CASE 3      '**************** DESACOPLAMIENTO *******************

      '------------------ Cálculo del error -------------------
      FOR j = 1 TO m
        e11tr(j, 5) = rtr(j) - ytr(j)
      NEXT

      '-------------- Cálculo de la ley de control u -------------------
    FOR j = 1 TO m
      u11tr(j, 5) = 0
      FOR i = 4 TO 1 STEP -1
        u11tr(j, 5) = u11tr(j, 5) + c11(j, i) * u11tr(j, 5 - i)
      NEXT
      FOR i = 5 TO 1 STEP -1
        u11tr(j, 5) = u11tr(j, 5) + d11(j, i) * e11tr(j, 5 - i + 1)
      NEXT
    NEXT
      '------------ Recuperación de u en vector de salida --------------
      FOR j = 1 TO m
        utr1(j) = u11tr(j, 5)
      NEXT
      '------------ Actualización de valores para nuevo cálculo ---------
      FOR j = 1 TO m
        FOR i = 1 TO 4
          e11tr(j, i) = e11tr(j, i + 1)
          u11tr(j, i) = u11tr(j, i + 1)
        NEXT
      NEXT

      '----------- Luego de compensar, se realiza el cálculo de u -------
      CALL mulmatvector(m, m, e(), utr1(), eutr1())
      CALL mulmatvector(m, n, f1(), ytr(), f1ytr())
      CALL sumvector(m, eutr1(), f1ytr(), utr())

      FOR j = 1 TO m
      '----------- limitación de valores de salida debido a ------------
      '----- limitaciones del equipo de adquisición de datos y control --
      IF utr(j) > 10 THEN
         utr(j) = 10
      ELSEIF utr(j) < 0 THEN
         utr(j) = 0
      END IF
      '----------------------------------------------------------------
      NEXT

END SELECT

'--------------------------------------------------------------------
'--------------------------------------------------------------------
'A continuación se realiza la parte del programa correspondiente a la
'salida de los gráficos de acuerdo a la elección realizada previamente
'por el usuario.
'--------------------------------------------------------------------
'--------------------------------------------------------------------
IF nopciontr2 = 1 THEN
   ytr = ytr(nopciontr3)
ELSEIF nopciontr2 = 2 THEN
   ytr = utr(nopciontr3)
```

```
IF nopciontr4 = 1 THEN
   utr = ytr(nopciontr5)
ELSEIF nopciontr4 = 2 THEN
   utr = utr(nopciontr5)
END IF

IF iii > 580 THEN
  CALL defgraf
  iii = 0
END IF
CALL graf(iii)
iii = iii + 1

'///////////////////////////////////////////////////////////////////////////
'///////////////////////////////////////////////////////////////////////////

FOR ii = 1 TO nu        '//////////////////////////////////////////////////

nopcion = ii - 1

SELECT CASE nopcion
   CASE 0
     ionl$ = "ANOUT0"
   CASE 1
     ionl$ = "ANOUT1"
   CASE 2
     ionl$ = "ANOUT2"
   CASE 3
     ionl$ = "ANOUT3"
   CASE 4
     ionl$ = "ANOUT4"
END SELECT

d! = utr(ii)                'Se requiere real de precisión simple

IF d! > 10 THEN
  d! = 10
END IF

FOR dep! = 1 TO 1

            CALL ARPUTVALF(arn1$, dep!, -1, ionl$, d!, 0)

        'Comando que va colocando los valores calculados en el lugar del
        'area correspondiente

NEXT dep!

NEXT                   '//////////////////////////////////////////////////

DO
  CALL timerread(val1())
LOOP UNTIL val1(0) / bintv% > t12
t12 = INT(val1(0) / bintv%) + 1
```

```
'------------------------------------------------------------------------
'A continuación se determina que el equipo pueda permanecer controlando
'la planta en forma indefinida
IF t12 > INT(4294967295# / bintv% - 10) THEN
  tim(0) = 1
  t12 = 1
  CALL timerstart(tim(), "nt", "timer")
END IF
'------------------------------------------------------------------------

'//////////////////////////////////////////////////////////////////////////
'//////////////////////////////////////////////////////////////////////////

   an$ = INKEY$
WEND                    '********************************* fin lazo

CALL intoff                                   'Deshabilita interrupciones
LOCATE 22, 12
PRINT "                                                              "
'--------------------------------------------------------------------------
'--------------------------------------------------------------------------
dep! = 1
FOR ii = 1 TO nu        '/////////////////////////////////////////////////

nopcion = ii - 1        ' A continuación se asegura salidas en cero

SELECT CASE nopcion
   CASE 0
     ionl$ = "ANOUT0"
   CASE 1
     ionl$ = "ANOUT1"
   CASE 2
     ionl$ = "ANOUT2"
   CASE 3
     ionl$ = "ANOUT3"
   CASE 4
     ionl$ = "ANOUT4"
END SELECT

CALL ARPUTVALF(arn1$, dep!, -1, ionl$, 0!, 0) 'asegura salida
                                              'en cero
NEXT                       '//////////////////////////////////////////////

CALL inton(1, "mil")      'Los siguientes párrafos permiten la
ijk = 0                   'salida de un valor igual a cero en
FOR ij = 1 TO 10000       'todos los canales de salida , es por
ijk = ijk + 1             'esto que se da el retardo adecuado.
NEXT
CALL intoff

'--------------------------------------------------------------------------
'--------------------------------------------------------------------------
```

```
'------------------------------------------------------------------------
'------------------------------------------------------------------------
'A continuación se realiza la parte del programa que nos da la posibilidad
'de escoger nuevos compensadores o cambiar las referencias del sistema
'------------------------------------------------------------------------
IF nopciontr1 = 1 OR nopciontr1 = 3 THEN
'------------------------------------------------------------------------
   LOCATE 21, 12
   PRINT "                                                                  "
   LOCATE 21, 12
   INPUT "Desea ensayar un nuevo compensador (s/n)"; opcion$
   IF opcion$ = "s" OR opcion$ = "S" THEN
      SCREEN 0
      WIDTH 80, 25
      COLOR 15, 1
      CLS
      GOSUB otrocomp1
      GOSUB resetear
      GOTO repetirtr
   END IF
'------------------------------------------------------------------------
   LOCATE 21, 12
   PRINT "                                                         "
   LOCATE 21, 12
   INPUT "Desea cambiar las referencias del sistema (s/n)"; opcion$
   IF opcion$ = "s" OR opcion$ = "S" THEN
     GOSUB referenciatr
     GOSUB resetear
     GOTO repetirtr
   END IF
'------------------------------------------------------------------------
ELSEIF nopciontr1 = 2 THEN
'------------------------------------------------------------------------
   LOCATE 21, 12
   PRINT "                                                         "
   LOCATE 21, 12
   INPUT "Desea cambiar los valores de offset (s/n)"; opcion$
   IF opcion$ = "s" OR opcion$ = "S" THEN
     GOSUB referenciatr
     GOTO repetirtr
   END IF
'------------------------------------------------------------------------
END IF
'------------------------------------------------------------------------


CALL ardel("entradas%")                 'Se borran de memoria los arreglos
CALL ardel("salidas%")                 'que tienen que ver con tiempo real
CALL init
CALL softinit
CLEAR
SCREEN 0
WIDTH 80, 25
COLOR 15, 1
GOTO menutr1
fintr:
SCREEN 0
CLS
END
```

```
otrocomp1:
FOR j = 1 TO m
otrocomp3:
   CLS
   LOCATE 3, 11
   PRINT "C O M P E N S A D O R "; j
   PRINT
   PRINT "Desea ensayar compensador P.I.D. (1)"
   PRINT "u otro compensador (2)";
   INPUT opcion
   '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
   vector(j) = opcion
   '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
   SELECT CASE opcion
      CASE 1
         PRINT
         PRINT "Parámetros del compensador P.I.D. #"; j
         PRINT
         INPUT "Ganancia proporcional ................. Kp"; gankp
         INPUT "Ganancia integral ..................... Ki"; ganki
         INPUT "Ganancia derivativa ................... Kd"; gankd
        '** INPUT "Periodo de muestreo ................... T"; t
        '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
         matriz1#(j, 1) = gankp
         matriz1#(j, 2) = ganki
         matriz1#(j, 3) = gankd
        '¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿¿
         c11(j, 1) = 1
         d11(j, 1) = gankp + (ganki * t) / 2 + gankd / t
         d11(j, 2) = (ganki * t) / 2 - gankp - (2 * gankd) / t
         d11(j, 3) = gankd / t
      CASE 2
         PRINT
         PRINT "Ecuación de diferencias del compensador"; j
         INPUT "Orden de la ley de control u"; nn11
         INPUT "Orden del error e"; nn22
         IF nn11 > nn22 THEN
          PRINT "Error,°numerador>°denominador en ecuación del compensador
          ; j
         END IF
         PRINT
         PRINT "Compensador"; j; ".-Coeficientes de retardo en u"
         FOR i = 1 TO nn11
           PRINT "Coeficiente de retardo en u(k-"; i; ")="; :
           INPUT c11(j, i)
         NEXT
         PRINT
         FOR i = 1 TO nn22 + 1
           PRINT "Coeficiente de retardo en e(k-"; i - 1; ")="; :
           INPUT d11(j, i)
         NEXT
      CASE ELSE
         GOTO otrocomp3
      END SELECT
NEXT
RETURN
```

```
referenciatr:
  FOR i = 1 TO m
    LOCATE 21, 12
    PRINT "                                                          "
    IF nopciontr1 = 2 THEN
      LOCATE 21, 12
      PRINT "offset "; i; "=";
    ELSE
      LOCATE 21, 12
      PRINT "referencia "; i; "=";
    END IF
    INPUT rtr(i)
  NEXT
RETURN

resetear:
     FOR i = 1 TO m                         'Se asegura que los valores
       FOR j = 1 TO 5                       'iniciales del cálculo respec-
         e11tr(i, j) = 0                    'tivo son iguales a 0
         u11tr(i, j) = 0
       NEXT
     NEXT
RETURN


DEFINT I-N
DEFDBL A-H, O-Z
SUB defgraf
LINE (0, 10)-(639, 140), 1, BF
LINE (0, 10)-(639, 140), 2, B
'--------
LINE (40, 25)-(629, 25), 2, , &H8888
'--------
LINE (30, 125)-(629, 125), 7
LOCATE 17, 30
PRINT "TIEMPO"
LOCATE 5, 2
PRINT y$; nopciontr3
LINE (40, 20)-(40, 130), 7


LINE (0, 185)-(639, 315), 1, BF
LINE (0, 185)-(639, 315), 2, B
'---------
LINE (40, 200)-(629, 200), 2, , &H8888
'---------
LINE (30, 300)-(629, 300), 7
LOCATE 39, 30
PRINT "TIEMPO"
LOCATE 27, 2
PRINT u$; nopciontr5
LINE (40, 195)-(40, 305), 7

END SUB
```

```
DEFINT I-N
DEFDBL A-H, O-Z
SUB graf (i)
'-----------------------------------------------------
'A continuación se grafica referencia, dado el caso

IF nopciontr2 = 1 THEN
   rg = -10 * rtr(nopciontr3) + 125
   LINE (40, rg)-(629, rg), 5, , &H8888
END IF

IF ytr >= 10 THEN
   PSET (i + 40, 25), 4
ELSEIF ytr <= 0 THEN
   PSET (i + 40, 125), 4

ELSE
   YG = -10 * ytr + 125
   PSET (i + 40, YG), 14
END IF

'-----------------------------------------------------
'A continuación se grafica referencia, dado el caso

IF nopciontr4 = 1 THEN
   rg = -10 * rtr(nopciontr5) + 300
   LINE (40, rg)-(629, rg), 5, , &H8888
END IF

IF utr >= 10 THEN
   PSET (i + 40, 200), 4
ELSEIF utr <= 0 THEN
   PSET (i + 40, 300), 4
ELSE
   UG = -10 * utr + 300
   PSET (i + 40, UG), 14
END IF

END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB mulmatvector (n, m, a(), B(), c())
  FOR i = 1 TO n
    r = 0
    FOR j = 1 TO m
      r = r + a(i, j) * B(j)
    NEXT
    c(i) = r
  NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB negvector (n, a())
  FOR i = 1 TO n
    a(i) = -a(i)
  NEXT
END SUB
```

```
DEFINT I-N
DEFDBL A-H, O-Z
SUB retardo
j = 0
FOR i = 1 TO 32000 'retardo para visualización
   j = j + 1
   a = a + 5
   a = a - 5
NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB subvector (n, a(), B(), c())
  FOR i = 1 TO n
    c(i) = a(i) - B(i)
  NEXT
END SUB

DEFINT I-N
DEFDBL A-H, O-Z
SUB sumvector (n, a(), B(), c())
  FOR i = 1 TO n
    c(i) = a(i) + B(i)
  NEXT
END SUB
```