19
29

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO Y PROGRAMACION DE ALGORITMOS PARA EL ANALISIS Y SIMULACION DE SISTEMAS EN EL ESPACIO DE ESTADOS.

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A:

ARTEMIO ROBERTO FLORES RAMOS

DIRECTOR DE TESIS:
ING. F. RODRIGUEZ

México D. F. FALLA DE ORIGEN 1990

CONTENIDO.

I.- INTRODUCCION: ESPACIO DE ESTADOS.

ESPACIO DE ESTADOS

Un sistema es una combinación de componentes que actúan conjuntamente y cumplen determinado objetivo.

Dado un sistema para determinar la respuesta de salida presente y futura de este sistema,es suficiente conocer un conjunto de condiciones iniciales y las funciones de entrada del sistema, presentes y futuras. En esencia, esto implica una filosofía de estado del sistema.

El espacio n-dimensional ,cuyos ejes coordenados son X_1 , X_2 , X_3, . . . , X_n , se denominan espacio de estados.Se puede representar cualquier estado por un punto en el espacio de estados.

ANALISIS DEL ESPACIO DE ESTADOS

Un sistema puede tener múltiples entradas y múltiples salidas,donde estas pueden estar interrelacionadas en forma complicada.Para analizar tales sistemas mediante tecnicas modernas,es escencial reducir la complejidad de las expresiones matemáticas,así como también se es escencial recurrir al uso de computadoras para los cálculos necesarios en su análisis.

Desde este punto de vista de un sistema con múltiples entradas y múltiples salidas,el método del espacio de estados para el análisis de sistemas es el más adecuado.

El concepto de el método de Espacio de Estados se basa en la descripción de las ecuaciones del sistema en términos de n ecuaciones diferenciales o ecuaciones en diferencias de primer orden que pueden combinarse en una ecuación diferencial o ecuación en diferencias vectorial-matricial de primer orden.

ESPACIO DE ESTADOS.

Un sistema dinámico y de parámetros concentrados, puede ser descrito mediante ecuaciones diferenciales ordinarias,o en diferencias ordinarias,donde la variable independiente es el tiempo.

Utilizando la notación vectorial-matricial,se puede expresar una ecuación diferencial o ecuación en diferencias de orden -n por un sistema de ecuaciones diferenciales o en diferencias de primer orden,en forma matricial.

En este caso si n elementos del vector constituyen un juego de variables de estado, se denomina ecuación de estado a la ecuación diferencial o ecuación en diferencias vectorial-matricial.

Por otro lado un vector de estado es un vector que determina unívocamente el estado del sistema $X(t)$, para cualquier $t \geq t_o$, una vez especificada la entrada $U(t)$ para $t \geq t_o$.

ESTADO.

El estado de un sistema dinámico es el conjunto más pequeño de variables (llamadas variables de estado) tal que el conocimiento de esas variables en $t = t_o$, justamente con la entrada para $t \geq t_o$, determinan totalmente el comportamiento del sistema para cualquier tiempo $t \geq t_o$.

VARIABLES DE ESTADO.

Las variables de estado de un sistema dinamíco son el conjunto más pequeño de variables que determinan el estado del sistema dinámico si se necesitan al menos n variables $X_1(t), X_2(t), \ldots, X_n(t)$ para describir totalmente el comportamiento de un sistema dinámico, tal que una vez dada la entrada para $t \geq t_o$ y el estado inicial en $t = t_o$ está especificado, el estado futuro del sistema queda totalmente determinado, esas n variables $X_1, X_2, \ldots, X_n$ constituyen un conjunto de variables de estado.

VECTOR DE ESTADO.

Si n variables de estado son necesarias para describir completamente el comportamiento de un sistema dado, entonces esas n variables de estado pueden ser consideradas como las n componentes de un vector X, ese vector es llamado vector de estado.

DEFINICION: SISTEMA DINAMICO.

Un sistema dinámico es una relación matemática orientada en la cual.

(1) Existe un valor real de salida $y(t)$ para todo $t \geq to$ dado un valor real de entrada $u(t)$ para todo t.

(2) Las salidas $y(t)$ no dependen de las entradas $u(\tau)$ para $\tau > t$.

DEFINICION DE LINEALIDAD.

Dados cualquier par de números α, β; dos estados $X_1(t_0), X_2(t_0)$; dos señales de entrada $U_1(\tau), U_2(\tau)$ y sus correspondientes señales de salida $Y_1(\tau), Y_2(\tau)$ para $\tau \geq t_0$. Entonces un sistema es lineal si:

El estado $X_3(t_0) = \alpha \cdot X_1(t_0) + \beta \cdot X_2(t_0)$, la salida $Y_3(\tau) = \alpha \cdot Y_1(\tau) + \beta \cdot Y_2(\tau)$ y la entrada $U_3(\tau) = \alpha \cdot U_1(\tau) + \beta \cdot U_2(\tau)$ aparecen en la totalidad de pares entrada-salida que describen el comportamiento del sistema. Los dos $Y_3(\tau)$ y $X_3(\tau)$ corresponden al estado $X_3(t_0)$ y a la señal de salida $U_3(\tau)$.

ECUACIONES DEL ESPACIO DE ESTADOS EN TIEMPO CONTINUO.

Para sistemas lineales, tiempo continuo e invariante en el tiempo y de parámetros concentrados puede ser representado por la siguiente ecuación de estado y la ecuación de salida.

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t) \qquad (1)$$

$$Y(t) = C \cdot X(t) + D \cdot U(t) \qquad (2)$$

Donde

$X(t)$: Es el vector de estado y es de dimensión 'n'.

$Y(t)$: Es el vector de salida y es de dimensión 'm'.

$U(t)$: Es el vector de entrada y es de dimensión 'r'.

A : Matriz de estado de orden $(n \times n)$

B : Matriz de entrada de orden $(n \times r)$

C : Matriz de salida de orden $(m \times n)$

D : Matriz de orden $(m \times r)$

ECUACIONES DEL ESPACIO DE ESTADOS EN TIEMPO DISCRETO.

Para sistemas de tiempo discreto, lineales, invariantes con el tiempo y de parámetros concentrados la ecuación de estado y la ecuación de salida están dadas por

$$X(k+1) = G.X(k) + H.U(k) \qquad (3)$$

$$Y(k) = C \cdot X(k) + D.U(k) \qquad (4)$$

Donde

X(k) : Es el vevtor de estado y es de dimensión 'n'.

Y(k) : Es el vector de salida y es de dimensión 'm'.

U(k) : Es el vector de entrada y es de dimensión 'r'.

G : Matriz de estado de orden (n x n)

H : Matriz de entrada de orden (n x r)

C : Matriz de salida de orden (m x n)

D : Matriz de orden (m x r)

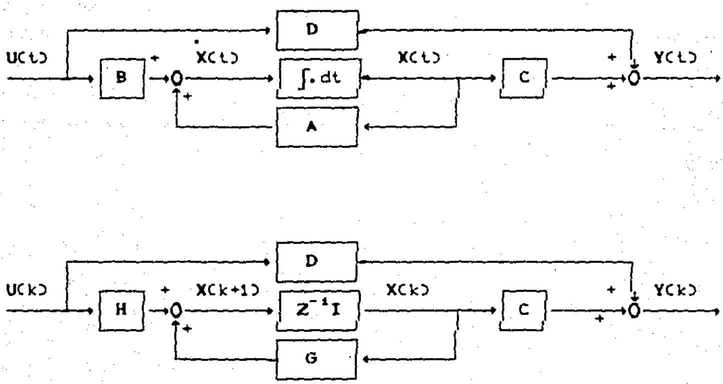


FIGURA 1:DIAGRAMA DE BLOQUES DE UN SISTEMA DE CONTROL LINEAL, INVARIANTE EN EL TIEMPO Y DE PARAMETROS CONCENTRADOS DE TIEMPO CONTINUO COMO DE TIEMPO DISCRETO RESPECTIVAMENTE REPRESENTADO EN ESPACIO DE ESTADOS, DEFINIDO POR LAS ECUACIONES DE ESTADOS.

(A) SOLUCION DE LA ECUACION NO-HOMOGENEA DE TIEMPO CONTINUO.

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t) \tag{5}$$

$$e^{-A \cdot t}\,\dot{X}(t) = e^{-A \cdot t}\left[A \cdot X(t) + B \cdot U(t) \right] \tag{6}$$

$$e^{-A \cdot t}\,\dot{X}(t) - e^{-A \cdot t} A \cdot X(t) = e^{-A \cdot t} B \cdot U(t) \tag{7}$$

$$\frac{d}{dt}\left[e^{-A \cdot t} X(t) \right] = e^{-A \cdot t} B \cdot U(t) \tag{8}$$

$$\int_0^t d\,(e^{-A \cdot t} X(t)) = \int_0^t e^{-A \cdot t} B \cdot U(t)\, dt$$

$$\left(e^{-A \cdot t} X(t) \right) \Big]_0^t = \int_0^t e^{-A \cdot \delta} B \cdot U(\delta) \cdot d\delta$$

$$e^{-A \cdot t} X(t) - e^{-A \cdot (0)} X(0) = \int_0^t e^{-A \cdot \delta} B \cdot U(\delta)\, d\delta \tag{9}$$

Donde

$$X(t) = e^{A \cdot t} X(0) + \int_0^t e^{A(t-\delta)} B \cdot U(\delta)\, d\delta \tag{10}$$

(B) SOLUCION DE LA ECUACION DE ESTADOS DE TIEMPO DISCRETO.

$$X(k+1) = G \cdot X(k) + H \cdot U(k) \tag{11}$$

Por recurrencia tenemos $k>0$

$$X(1) = G\cdot X(0) + H\cdot U(0) \tag{12}$$

$$X(2) = G\cdot X(1)+H\cdot U(1) = \tag{13}$$

$$= G(G\cdot X(0) + H\cdot U(0))+H\cdot U(1) =$$

$$= G^2X(0) + G\cdot H\cdot U(0) + H\cdot U(1)$$

$$X(3) = G\cdot X(2) + H\cdot U(2) = \tag{14}$$

$$= G(G^2X(0) + G\cdot H\cdot U(0) + H\cdot U(1)) + H\cdot U(2) =$$

$$= G^3X(0) + G^2\cdot H\cdot U(0) + G\cdot H\cdot U(1) + H\cdot U(2)$$

Para k

$$X(k) = G\cdot X(k-1) + H\cdot U(k-1) = \tag{15}$$

$$X(k)=G\left[G^{k-1}X(0)+G^{k-2}HU(0)+G^{k-3}HU(1)+\ldots+GHU(k-3)+HU(k-2)\right]+HU(k-1)$$

$$X(k) = G^kX(0)+G^{k-1}HU(0)+G^{k-2}HU(1)+\ldots+G^2HU(k-3)+GHU(k-2)+HU(k-1)$$

Donde

$$X(k) = G^kX(0) + \sum_{J=0}^{k-1} G^{k-J-1}H\cdot U(J) \qquad (k=1,2,3,\ldots) \tag{16}$$

ESTABILIDAD.

Un sistema lineal invariante en el tiempo se llama estable si para cualquier estado inicial finito $X(t_o)$ hay un número positivo M (que depende de $X(t_o)$) tal que

$$\|X(t)\| < M \qquad \text{para todo} \quad t\geq t_o$$

y

$$\lim_{t\to\infty} \|X(t)\| = 0$$

Por otro lado las condiciones de entrada nulas, la ecuación de transición de estado del sistema:

$$\dot{X} = A \cdot X$$

es

$$X(t) = e^{A(t-to)}X(to)$$

$$\|X(t)\| = \|e^{A(t-to)}X(to)\| \leq \|e^{A(t-to)}\|.\|X(to)\|$$

de la definición de estabilidad se requiere que

$$\|X(t)\| \quad o \quad \|\phi(t-to)\|.\|X(t_o)\| \quad \text{sea}$$

finita, con lo cual, si tal como postulamos $\| X(t_o) \|$ es finita, entonces $\|\phi(t-t_o)\|$ es tambien finita para $t>t_o$ analogamente se llega a la condición de que $\lim_{t\to\infty} \|e^{A(t-to)}\|=0$. Por otro lado la estabilidad de un sistema de Tiempo Discreto obtenido por la discretización de un sistema de Tiempo Continuo el correspondiente sistema es:

$$X((k+1)T) = G \cdot X(kT)$$

Donde

$$G = e^{A \cdot T}$$

si el sistema de Tiempo Continuo es asintoticamente estable, entonces

$$\lim_{n\to\infty} \|G^n\| \to 0$$

y el sistema discretizado es tambien asintoticamente estable. Esto es porque λi's son los valores caracteristicos de **A**, entonces los $e^{\lambda i.T}$'s son los modos naturales de G. $|e^{\lambda i.T}| < 1$ si $\lambda i.T$ es negativo.

REPRESENTACION EN EL ESPACIO DE ESTADOS DE SISTEMAS DE TIEMPO-DISCRETO.

(el caso de sistemas en Tiempo-Continuo se hace en paralelo).

Si tenemos el sistema

$$y(k) + a_1 y(k-1) + a_2 y(k-2) + \ldots + a_n y(k-n) =$$

$$= b_0 u(k) + b_1 u(k-1) + \quad + b_n u(k-n) \tag{17}$$

la ecuación puede ser escrita como función de transferencia.

$$G(Z) = \frac{Y(Z)}{U(Z)} = \frac{b_0 + b_1 Z^{-1} + \ldots + b_n Z^{-n}}{1 + a_1 Z^{-1} + \ldots + a_n Z^{-n}} \tag{18}$$

$$G(Z) = \frac{b_0 + b_1 Z^{-1} + \ldots + b_n Z^{-n} + a_1 b_0 Z^{-n} + a_2 b_0 Z^{-2} + \ldots - a_1 b_0 Z^{-1} - a_2 b_0 Z^{-2} - \ldots}{1 + a_1 Z^{-1} + \ldots + a_n Z^{-n}}$$

$$G(Z) = \frac{b_0 + a_1 b_0 Z^{-1} + a_2 b_0 Z^{-2} + \ldots + a_n b_0 Z^{-n} + (b_1 - a_1 b_0) Z^{-1} + \ldots + (b_0 - a_n b_0) Z^{-n}}{1 + a_1 Z^{-1} + \ldots + a_n Z^{-n}}$$

$$G(Z) = \frac{b_0 (1 + a_1 Z^{-1} + a_2 Z^{-2} + \ldots + a_n Z^{-n}) + (b_1 - a_1 b_0) Z^{-1} + \ldots + (b_n - a_n b_0) Z^{-n}}{(1 + a_1 Z^{-1} + a_2 Z^{-2} + \ldots + a_n Z^{-n})}$$

$$G(Z) = b_0 + \frac{(b_1 - a_1 b_0) Z^{-1} + (b_2 - a_2 b_0) Z^{-2} + \quad + (b_n - a_n b_0) Z^{-n}}{(1 + a_1 Z^{-1} + a_2 Z^{-2} + \ldots + a_n Z^{-n})} = \frac{Y(Z)}{U(Z)}$$

Entonces

$$Y(Z) = b_o U(Z) + [\ldots\ldots\ldots\ldots\ldots\ldots]\ U(Z) \qquad (19)$$

si definimos que :

$$Y(Z) = b_o U(Z) + \bar{Y}(Z) \qquad (20)$$

Donde :

$$\bar{Y}(Z) = [\ \ldots\ ]\ U(Z) \qquad (21)$$

$$\frac{\bar{Y}(Z)}{(b_1-a_1b_o)Z^{-1}+(b_2-a_2b_o)Z^{-2}+\ldots+(b_n-a_nb_o)Z^{-n}} = \frac{U(Z)}{1+a_1Z^{-1}+\ldots+a_nZ^{-n}} = Q(Z)$$

Entonces

$$Q(Z)(1+a_1Z^{-1}+a_2Z^{-2}+\ldots+a_nZ^{-n}) = U(Z) \qquad (22)$$

$$Q(Z) = -a_1Z^{-1}Q(Z)-a_2Z^{-2}Q(Z)-\ldots-a_nZ^{-n}Q(Z) + U(Z) \qquad (23)$$

$$\bar{Y}(Z)=(b_1-a_1b_o)Z^{-1}Q(Z)+(b_2-a_2b_o)Z^{-2}Q(Z)+\ldots+(b_n-a_nb_o)Z^{-n}Q(Z) \qquad (24)$$

Definiendo las variables de estado de esta ecuación.

$X_1(Z) = Z^{-n}Q(Z)$ obtenemos: $ZX_1(Z) = X_2(Z)$

$X_2(Z) = Z^{-n+1}Q(Z)$ obtenemos: $ZX_2(Z) = X_3(Z)$

$\vdots$

$X_n(Z) = Z^{-1}Q(Z)$ obtenemos: $ZX_{n-1}(Z) = X_n(Z)$

Entonces obtenemos que:

$$X_1(k+1) = X_2(k)$$

$$
\begin{aligned}
X_2(k+1) &= X_3(k) \\
&\vdots \\
X_{n-1}(k+1) &= X_n(k)
\end{aligned}
\qquad (25)
$$

de la ecuación (23) sustituyendo las relaciones anteriores.

$$ZX_n(Z) = -a_1 X_n(Z) - a_2 X_{n-1}(Z) - \ldots - a_n X_1(Z) + U(Z)$$

$$X_n(k+1) = -a_n X_1(k) - a_{n-1} X_2(k) - \ldots - a_1 X_n(k) + U(k) \qquad (26)$$

y de la ecuación (24) sustituyendo las relaciones tenemos

$$\overline{Y}(Z) = (b_1 - a_1 b_0) X_n(Z) + (b_2 - a_2 b_0) X_{n-1}(Z) + \ldots + (b_n - a_n) X_1(Z)$$

usando esta última ecuación, la ecuación (20) puede ser escrita en la forma

$$Y(k) = (b_n - a_n b_0) X_1(k) + (b_{n-1} - a_{n-1} b_0) X_2(k) + \ldots + (b_1 - a_1 b_0) X_n(k) + b_0 U(k) \qquad (27)$$

Combinando las ecuaciones (10) y (11) resulta la ecuación de estado.

$$
\begin{bmatrix} X_1(k+1) \\ X_2(k+1) \\ \vdots \\ X_{n-1}(k+1) \\ X_n(k+1) \end{bmatrix}
\begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \ldots & -a_1 \end{bmatrix}
\begin{bmatrix} X_1(k) \\ X_2(k) \\ \vdots \\ \\ X_n(k) \end{bmatrix}
+
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} U(k)
\qquad (28)
$$

y la ecuación de salida.de la ecuación (27)

$$Y(k) = \begin{bmatrix} b_n - a_n b_0 & b_{n-1} - a_{n-1} b_0 & \cdots & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} X_1(k) \\ X_2(k) \\ \vdots \\ X_n(k) \end{bmatrix} + b_0 \cdot U(k) \tag{29}$$

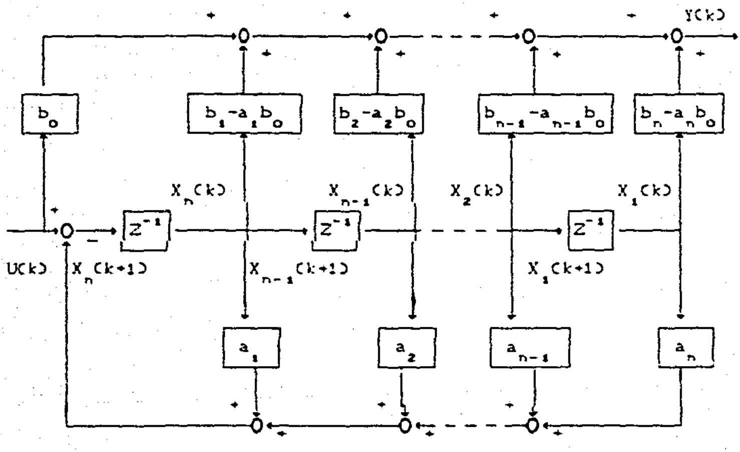


FIGURA 2 :DIAGRAMA DE BLOQUES QUE REPRESENTA EL SISTEMA DADO

POR LAS ECUACIONES (28) Y (29)

TRANSFORMACION A UN SISTEMA EQUIVALENTE.

Si tenemos el sistema lineal, tiempo discreto e invariante en el tiempo.

$$X(k+1) = G\cdot X(k) + H\cdot U(k) \tag{30}$$

$$Y(k) = C\cdot X(k) + D\cdot U(k) \tag{31}$$

Donde

$X(k)$: Vector de estado de dimension 'n'

$Y(k)$: Vector de salida de dimensión 'm'

$U(k)$: Vector de entrada de dimension 'r'

G : Matriz de estado de orden [nxn]

H : Matriz de entrada de orden [nxr]

C : Matriz de salida de orden [mxn]

D : Matriz de orden [mxr]

Definiendo un nuevo vector de estado

$$X(k) = P\cdot \hat{X}(k) \tag{32}$$

Donde P es cualquier matriz de orden [nxn].no-singular.

$$P\cdot \hat{X}(k+1) = G\cdot P\cdot \hat{X}(k) + H\cdot U(k) \tag{33}$$

$$\hat{X}(k+1) = P^{-1}(G\cdot P\cdot \hat{X}(k) + H\cdot U(k)) \tag{34}$$

$$\hat{X}(k+1) = (P^{-1}G\cdot P)\hat{X}(k) + (P^{-1}H)U(k) \tag{35}$$

$$Y(k) = (C\cdot P)\hat{X}(k) + D\cdot U(k) \tag{36}$$

Los valores característicos de **A** y los de $P^{-1}A \cdot P$ son idénticos es decir los polinomios característicos de

$$| \lambda I - A | = | \lambda I - P^{-1} A \cdot P |$$

son idénticos .

Demostración:

$$\begin{aligned} | \lambda I - P^{-1} A \cdot P | &= | \lambda P^{-1} P - P^{-1} A \cdot P | & (37) \\ &= | P^{-1} (\lambda I - A) P | \\ &= | P^{-1} | \; | \lambda I - A | \; | P | \\ &= | P^{-1} | \; | P | \; | \lambda I - A | \\ &= | P^{-1} P | \; | \lambda I - A | \\ &= | \lambda I - A | & (38) \end{aligned}$$

Algoritmo de Faddeev.

Si tenemos el sistema lineal, tiempo continuo e invariante con el tiempo

$$\dot{X}(t) = A \cdot X(t) \qquad (39)$$

Donde

$X(t)$: Vector de estado de dimensión 'n'

A : Matriz de estado de orden [nxn]

La solución es

$$X(t) = \phi(t) \cdot X(0) \qquad (40)$$

Donde $\phi(t)$ es la matriz de transición del sistema.

Tomando la transformada de Laplace de la ecuación

$$S \cdot X(S) - X(0) = A \cdot X(S) \qquad (41)$$

$$(SI - A) X(S) = X(0) \qquad (42)$$

$$X(S) = (SI-A)^{-1}X(0) \tag{43}$$

Si consideramos la ecuación de solución

$$X(t) = \phi(t) \cdot X(0) \tag{44}$$

vemos que

$$\phi(s) = (SI-A)^{-1} \tag{45}$$

donde

$$\phi(S) = L \{ \phi(t) \} \tag{46}$$

Como $\phi(s)$ es una función racional de S para sistemas lineales, invariantes en el tiempo, podemos escribir la ecuación como

$$\phi(S) = (SI-A)^{-1} = \frac{adj(SI-A)}{det(SI-A)} \tag{47}$$

reconociendo al $|SI-A|$ como la ecuación característica, tenemos.

$$det(SI-A) = d(S) = S^n + a_1 S^{n-1} + a_2 S^{n-2} + \ldots + a_n \tag{48}$$

$$adj(SI-A) = IS^{n-1} + B_1 S^{n-2} + B_2 S^{n-3} + \ldots + B_{n-2} S + B_{n-1}$$

de la ecuación

$$(SI-A)^{-1} = \frac{adj(SI-A)}{det(SI-A)}$$

$$I(det(SI-A)) = (SI-A)(adj(SI-A)) \tag{49}$$

y sustituyendo las anteriores ecuaciones tenemos

$$I(S^n + a_1 S^{n-1} + a_2 S^{n-2} + \ldots + a_n) = (SI-A)(IS^{n-1} + B_1 S^{n-2} + B_2 S^{n-3} + \ldots + B_{n-1}) \tag{50}$$

Igualando potencias iguales de S tenemos

$$B_1 = A - a_1 I \tag{51}$$

$$B_2 = A \cdot B_1 - a_2 I$$

$$\vdots$$

$$B_{n-1} = A \cdot B_{n-2} + a_{n-1} I$$

$$B_n = A \cdot B_{n-1} - a_n I = 0$$

El coeficiente a_1 es igual a la suma de las raíces de A y puede demostrarse que es igual al trazo de A[1];esto es

$$a_1 = - \mathrm{Tr}\ A \tag{52}$$

en forma semejante

$$a_2 = - \frac{1}{2} \mathrm{Tr}\ A \cdot B_1$$

$$a_3 = - \frac{1}{3} \mathrm{Tr}\ A \cdot B_2$$

$$\vdots$$

$$a_n = - \frac{1}{n} \mathrm{Tr}\ A \cdot B_{n-1}$$

II.- FORMULACION DEL PROBLEMA.

Se va a diseñar un programa de computadora que permita el simular (analizar) sistemas multivariables, además que éste podra ser utilizado para diseño de controladores de sistemas representados mediante las ecuaciones de estado.

1.-El método de análisis por espacio de estados es una de las técnicas mas útiles que se han creado para el análisis y diseño de sistemas multivariables.La teoría de espacio de estados es esencialmente un método que se analiza en el dominio del tiempo.
La descripción del espacio de estados proporciona un sistema estándar de ecuaciones que ofrecen economía en la notación, facilitan la entrada de datos en el procesamiento de computadoras.
Y este método es adecuado para su extensión a los sistemas no-lineales y de tiempo variable.En la teoria de espacio de estados se pueden analizar problemas lineales de orden alto.
Esta teoría se analiza como ya se dijo en el dominio del tiempo contra la teoría clasica o convencional la cual se trabaja en el dominio de la frecuencia compleja.lo cual nos da una ventaja utilizando este método , ya que los métodos clasicos o convencionales utilizados no son rigurosos ya que estos se basan

en procedimientos de tanteo . En especial, para sistemas multivariables y un alto grado de complejidad, los métodos clasicos puede que no resulten muy adecuados en su solución. Por otro lado la función de transferencia describe la relación entrada-salida en el dominio de la transformada de laplace o Z , sin embargo, el método de la transformación tiene la desventaja de que todas las condiciones iniciales del sistema se despresian.
Por lo tanto cuando uno esta interesado en una solución en el dominio del tiempo que dependa mucho de la historia pasada del sistema que se esta analizando, la función de transferencia no nos facilitará toda la información necesaria del sistema.

2.-Se generarán algoritmos para solucionar las ecuaciones de estado. Tanto las ecuaciones de estado en tiempo discreto, como de tiempo continuo.

3.-El programa también permite analizar:

— Controlabilidad de estado.
— Controlabilida de salida.
— Observabilidad.

4.-El programa permitirá análizar tanto el sistema original como el sistema realimentado.

5.-El programa también permitirá el diseño de compensadores (observador-controlador).

6.-De los análisis anteriores se podra hacer el diseño de observadores de orden completo como los observadores de orden reducido.

Auxiliandose para esto de gráficas para todas las respuestas,en donde la entrada para los sistemas sera de cualquier tipo de funciones tanto algebráicas como funciones trascedentales, como son seno,coseno,tangente y exponencial.

III.- ANALISIS Y GENERACION DE ALGORITMOS.

Para poder análizar los sistemas multivariables tanto de tiempo continuo y de tiempo discreto se daran las siguientes demostraciones,para sistemas representados en espacio de estados,como es la controlabilidad de estado completa,controlabilidad de la salida completa y de la observabilidad completa,para posteriormente hacer la generación de los algoritmos para todos estos casos,y analizarlos por la computadora.Las siguientes demostraciones para el análisis de los sistemas en el espacio de estados se harán de una manera formal. Además daremos la generación de los algoritmos para obtener la Discretización de un sistema en tiempo-continuo para pasarlo a tiempo discreto , ademas los algoritmos para obtener las respuestas libre y la respuesta forzada,con el procedimiento de graficación y el algoritmo de la obtención de los valores caracteristicos del sistema.

A.- SISTEMAS MULTIVARIABLES EN TIEMPO CONTINUO :

TEOREMA: CONTROLABILIDAD DE ESTADO.

Se dice que un sistema es controlable en el tiempo t_o si es posible transferir un sistema por medio de un vector de control no restringido, desde cualquier estado inicial $X(t_o)$ a cualquier otro estado en un intervalo de tiempo finito.

Si el sistema considerado no es controlable puede no existir solución a un problema de control óptimo.

DEMOSTRACION:

Considere el sistema de control lineal de tiempo continuo e invariante con el tiempo

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t) \qquad (1)$$

Donde

$X(t)$: Vector de estado de dimensión 'n'

$U(t)$: Vector de entrada o control de dimensión 'r'

A : Matriz de estado de orden [n x n]

B : Matriz de entrada de orden [n x r]

Asumimos que el estado final $X(t_f)=X(t_1)$ es un estado albitrario.

La solución de la ecuación es, $t_o = 0$, estado final $= X(t_1)$

$$X(t_1) = e^{A\cdot t_1}X(0) + \int_0^{t_1} e^{A(t_1-s)}B\cdot U(s)ds \qquad (2)$$

$$X(t_1)-e^{At_1}X(0) = \int_0^{t_1} e^{A(t_1-s)}B\cdot U(s)ds \qquad (3)$$

$$e^{-A\cdot t_1}\left[X(t_1)-e^{A\cdot t_1}X(0) \right] = e^{-A\cdot t_1}e^{A\cdot t_1} \int_0^{t_1} e^{-A\cdot s}B\cdot U(s)\ ds$$

$$e^{-A\cdot t_1}X(t_1)-X(0) = \int_0^{t_1} e^{-A\cdot s}B\cdot U(s)ds \qquad (4)$$

Por el teorema de Cayley-Hamilton

$$e^{-A\cdot s} = \sum_{k=0}^{p-1} \alpha_k(s)A^k \qquad \text{Donde} \quad p \leq n \qquad (5)$$

$$e^{-A\cdot t_1}X(t_1)-X(0) = \int_0^{t_1} \sum_{k=0}^{p-1} \alpha_k(s)A^kB\cdot U(s)ds \qquad (6)$$

$$e^{-A\cdot t_1}X(t_1)-X(0) = \sum_{k=0}^{p-1} A^kB\cdot\int_0^{t_1} \alpha_k(s)U(s)ds \qquad (7)$$

Si se hace

$$\int_0^{t_1} \alpha_k(s)\cdot U(s)ds = \beta_k = \begin{bmatrix} \beta_{k1} \\ \beta_{k2} \\ \vdots \\ \beta_{kr} \end{bmatrix} \qquad (8)$$

como $U(s)$-Vector de dimensión-r, entonces β_k-Vector de dimensión-r

$$e^{-A\cdot t_1}X(t_1)-X(0) = \sum_{k=0}^{p-1} A^k B\, \beta_k$$

$$e^{-A\cdot t_1}X(t_1)-X(0) = \sum_{k=0}^{p-1} A^k [B_1\ B_2 \ldots B_r] \begin{bmatrix} \beta_{k,1} \\ \beta_{k,2} \\ \vdots \\ \beta_{k,r} \end{bmatrix}$$

De esta última ecuación podemos ver que cualquier estado inicial X(0) este puede ser transferido a un estado albitrario $X(t_1) = X(t_{final})$, es decir, debe haber una combinación lineal de

$$[\ B_1\ \ B_2 \ldots\ AB_1\ \ AB_2 \ldots\ A^{p-1}B_{r-1}\ \ A^{p-1}B_r] =$$

$$= [\ B\ \ AB\ \ A^2B \ldots\ A^{p-1}B\ ]$$

Es decir el sistema es de estado completamente controlable si y solo si, la matriz es linealmente independiente, entonces existen n-vectores linealmente independientes en la matriz por consiguiente tienen un espacio de estado n-dimencional o si la matriz $[n\times n\cdot r]$ es de rango n.

TEOREMA: CONTROLABILIDAD DE LA SALIDA.

La salida es completamente controlable si es posible construir un vector de control no restringido U(t) que pueda tranferir cualquier salida inicial Y(to) a cualquier salida final $Y(t_f)$ en un intervalo de tiempo finito $t_o \le t \le t_1$, $t_1 = t_f$.

La controlabidad de estado completa no es necesaria ni suficiente para controlar la salida del sistema.

DEMOSTRACION:

Considerando el sistema lineal, tiempo continuo e invariante en el tiempo,

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t) \qquad (9)$$

$$Y(t) = C \cdot X(t) + D \cdot U(t) \qquad (10)$$

Donde:

X(t): Vector de estado de dimensión ''n''

Y(t): Vector de salida de dimensión ''m''

U(t): Vector de entrada de dimensión ''r''

A: Matriz de estado de orden n x n

B: Matriz de entrada de orden n x r

C: Matriz de salida de orden m x n

D: Matriz de orden m x r

DEMOSTRACION:

Si $t_o = 0$, y

deseamos llegar a un t final es decir a una Y(tfinal) = $Y(t_f)$

La solución de la ecuación es:

$$X(t_1) = e^{A\cdot t_1}X(0) + \int_0^{t_1} e^{A(t_1-\delta)}B\cdot U(\delta)d\delta \qquad (11)$$

$$Y(t_1) = C\cdot X(t) + D\cdot U(t)$$

$$Y(t_1) = C\cdot e^{At_1}X(0) + C\int_0^{t_1} e^{A(t_1-\delta)}B\cdot U(\delta)d\delta + D\cdot U(t)$$

$$Y(t_1)-C\cdot e^{At_1}X(0) = C\int_0^{t_1} e^{A(t_1-\delta)}B\cdot U(\delta)d\delta + D\cdot U(t) \qquad (12)$$

Por el teorema de Cayler-Hamilton $e^{-A\cdot t}$ puede ser escrita como

$$e^{-A\cdot\delta} = \sum_{k=0}^{p-1} \alpha_k(\delta)A^k$$

o

$$e^{-A(t_1-\delta)} = \sum_{k=0}^{p-1} \alpha_k(t_1-\delta)A^k \qquad \text{donde } p \le n$$

$$Y(t_1) - Ce^{At_1}X(0) = \int_0^{t_1} C\sum_{k=0}^{p-1} \alpha_k(t_1-\delta)A^kB\cdot U(\delta)d\delta + D\cdot U(t) \qquad (13)$$

$$Y(t_1) - C\cdot e^{A\cdot t_1}X(0) = \sum_{k=0}^{p-1} C\cdot A^kB\left[\int_0^{t_1} \alpha_k(t_1-\delta)U(\delta)d\delta\right] + D\cdot U(t) \qquad (14)$$

Si las integrales se reemplazan por vectores W_k-vector de dimensión-r porque $U(\delta)$-es un vector de dimensión-r, entonces

$$W_k = \int_0^{t_1} \alpha_k(t_1-\delta)U(\delta)d\delta = \begin{bmatrix} W_{k1} \\ W_{k2} \\ \vdots \\ W_{kr} \end{bmatrix} \qquad ; \ k = 1,2,\ldots,n-1 \qquad (15)$$

$$Y(t_1)-C\cdot e^{At_1}X(0) = \sum_{k=0}^{p-1} C\ A^k B\ W_k + D\ U(t) =$$

$$= C\cdot B\cdot W_0 + C\cdot A\cdot B\cdot W_1 + \ldots + C\cdot A^{p-1}B\cdot W_{p-1} + D\cdot U(t)$$

donde la podemos representar como.

$$Y(t_1)-Ce^{A\cdot t_1}X(0) = \begin{bmatrix} CB & CAB & CA^2B & \ldots & CA^{p-1}B & D \end{bmatrix} \begin{bmatrix} W_0(t) \\ W_1(t) \\ \vdots \\ W_{p-1}(t) \\ U(t) \end{bmatrix}$$

El vector de incógnitas W es de dimensión $(n+1)\cdot r$.

si se desea resolver para $W_i(t)$ $(i=0,1,2,3,\ldots n-1, U(t))$ deberan existir m vectores linealmente independientes de

$$\begin{bmatrix} CB & CAB & CA^2B \ldots & CA^{p-1}B & D \end{bmatrix} \tag{16}$$

Entonces el sistema descrito es de salida completamente controlable si y solo sí la matriz de $[m\times(n+1)r]$ es de rango = m.

TEOREMA :OBSERVABILIDAD.

Un sistema es observable en el tiempo t_0, si con el sistema en el estado $X(T_0)$, es posible determinar este estado partiendo de la observación de la salida durante un intervalo de tiempo finito. Es decir se dice que el sistema es completamente observable si se puede determinar todo estado inicial X(0), a partir de la observación de Y(t) en un intervalo de tiempo finito. El sistema es, por tanto, completamente observable si toda transición del estado puede afectar a todo elemento del vector de salida.

Este concepto es útil para resolver el problema de reconstruir variables de estado no medibles en el espacio mínimo de tiempo posible.

DEMOSTRACION:

Dado el sistema lineal, tiempo continuo e invariante en el tiempo.

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t) \tag{17}$$

$$Y(t) = C \cdot X(t) + D \cdot U(t) \tag{18}$$

Donde

X(t) : Vector de estado de dimensión ''n''

Y(t) : Vector de salida de dimensión ''m''

U(t) : Vector de entrada de dimensión ''r''

A : Matriz de estado de orden $n \times n$

B : Matriz de entrada de orden $n \times r$

C : Matriz de salida de orden $m \times n$

D : Matriz de orden $m \times r$

Debe considerarse al sistema como no forzado por lo siguiente.

su solución es:

$$X(t) = e^{A \cdot t} X(0) + \int_0^t e^{A(t-\delta)} B \cdot U(\delta) d\delta \tag{19}$$

$$Y(t) = C\cdot e^{A\cdot t}X(0) + \int_0^t C\cdot e^{A(t-\delta)}B\cdot U(\delta)d\delta + D\cdot U(t)$$

Como las matrices **A**,**B**,**C**,**D** son conocidas y también se conoce U(t) , el término integral en el segundo miembro de esta última ecuación y el tercer termino es una magnitud conocida.Por tanto se puede restar del valor observado de Y(t).Por otro lado se puede ver que por la def. esta se refiere básicamente a la relación entre el estado del sistema y la salida de éste,entonces las condiciones de observabilidad pueden encontrarse independientemente de la entrada aplicada al sistema,por simplicidad se considera nula ya que no se altera la definición de observabilidad.

Entonces se tiene el sistema de tiempo continuo ,lineal e invariante con el tiempo

$$\dot{X}(t) = A\cdot X(t) \qquad (20)$$

$$Y(t) = C\cdot X(t) \qquad (21)$$

La solución es:

$$X(t) = e^{A\cdot t}X(0) \qquad (22)$$

entonces

$$Y(t) = C\cdot e^{A\cdot t}X(0) \qquad (23)$$

Donde tenemos por el teorema de Cayley-Hamilton

$$e^{A\cdot t} = \sum_{k=0}^{p-1} \alpha_k(t)A^k \qquad ,\text{donde} \quad p \le n \qquad (24)$$

$$Y(t) = C\cdot \sum_{k=0}^{p-1} \alpha_k(t)A^k X(0)$$

$$Y(t) = \sum_{k=0}^{p-1} \alpha_k(t)C\cdot A^k X(0) \qquad (25)$$

$$Y(t) = \left[\alpha_0(t)C + \alpha_1(t)CA + \ldots + \alpha_{p-1}(t)CA^{p-1} \right]\cdot X(0)$$

matricialmente tenemos:

$$Y(t) = \left[\alpha_0(t) \quad \alpha_1(t) \quad \alpha_2(t) \ldots \alpha_{n-1}(t) \right] \left[\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix} [X(0)] \right] \qquad (26)$$

Entonces las variables de estado iniciales X(0) es decir $X_1(0)$, $X_2(0), \ldots X_n(0)$ pueden ser determinadas unicamente si y solo si la matriz de observabilidad siguiente tenga n vectores columna linealmente independiente o que tenga la matriz

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix}$$

Rango = n.

Como sabemos que el rango de una matriz es igual al rango de su transpuesta

$$R(A) = R(A^T)$$

entonces podemos poner su transpuesta

$$\left[C \quad CA \quad CA^2 \ldots CA^{p-1} \right]^T =$$

$$= \left[C^T \quad (CA)^T \quad (CA^2)^T \ \ldots \ (CA^{n-1})^T \right] =$$

$$= \left[C^T \quad A^TC^T \quad (A^T)^2C^T \ \ldots \ (A^T)^{n-1}C^T \right]$$

La condición necesaria y suficiente de observabilidad completa es que el rango de la matriz $[n \cdot n{\times}m]$ sea = n o que tenga n vectores columna linealmente independientes.

B.- SISTEMAS MULTIVARIABLES EN TIEMPO DISCRETO.

TEOREMA: CONTROLABILIDAD DE ESTADO.

El sistema es de estado controlable si existe una señal de control,constanle por secciones,$U(kT)$ definida en un intervalo finito de muestreo $0 \leq KT \leq nT$ tal que,partiendo de cualquier estado inicial,pueda tranferirse a un estado final X_f en n periodos de muestreo.

Demostración:

Sea el sistema de tiempo discreto,lineal e invariante en el tiempo

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (27)$$

donde:

$X(kT)$: Vector de estado de dimensión 'n''.

$U(kT)$: Vector de entrada de dimensión 'r'.

G : Matriz de estado de orden $[\, n \times n \,]$.

H : Matriz de entrada de orden $[\, n \times r \,]$.

T : Periodo de muestreo.

$U(kT)$ —— constante para $kT \leq t \leq (k+1)T$

Asumimos que el estado inicial es albitrario, así mismo el estado final X_f es un estado albitrario.

La solución de la ecuación es :

$$X(nT) = G^n X(0) + \sum_{j=0}^{n-1} G^{n-j-1} H \cdot U(jT) \tag{28}$$

$$X(nT) = G^n X(0) + G^{n-1} H \cdot U(0) + G^{n-2} H \cdot U(T) + \ldots + H \cdot U((n-1)T)$$

$$X(nT) - G^n X(0) = H \cdot U((n-1)T) + \ldots + G^{N-2} H \cdot U(T) + G^{n-1} H \cdot U(0)$$

Donde obtenemos que.

$$X(nT) - G^n X(0) = \begin{bmatrix} H & GH & G^2H & \ldots & G^{n-2}H & G^{n-1}H \end{bmatrix} \cdot \begin{bmatrix} U((n-1)T) \\ U((n-2)T) \\ \vdots \\ U(0) \end{bmatrix} \tag{29}$$

El sistema es de estado controlable completamente si y solo si los vectores :

$$\begin{bmatrix} H & GH & G^2H & \ldots & G^{n-2}H & G^{n-1}H \end{bmatrix}$$

son linealmente independientes de la matriz $[n \times n \cdot r]$, o el rango sea n.

TEOREMA : CONTROLABILIDAD DE LA SALIDA.

El sistema es controlable de salida si es posible construir una señal de control no restringida $U(kT)$ que pueda transferir cualquier salida inicial $Y(0)$ o $Y(kT)$ a cualquier salida final albitraria Y_f en un número n-finito de periodos de muestreo.

DEMOSTRACION:

Considerando el sistema de tiempo discreto, lineal e invariante en el tiempo

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (30)$$

$$Y(kT) = C \cdot X(kT) + D \cdot U(kT) \qquad (31)$$

Donde

$X(kT)$: Vector de estado de dimensión ''n''

$U(kT)$: Vector de entrada de dimensión ''r''

$Y(kT)$: Vector de salida de dimensión ''m''

G : Matriz de estado de dimensión $n \times n$

H : Matriz de entrada de dimensión $n \times r$

C : Matriz de salida de dimensión $m \times n$

D : Matriz m x r

La solución de la ecuación es:

$$X(nT) = G^n X(0) + \sum_{j=0}^{n-1} G^{n-j-1} H \cdot U(jT) \qquad (32)$$

$$Y(nT) = C \cdot X(nT) + D \cdot U(nT) \qquad (33)$$

$$Y(nT) = C \cdot G^n X(0) + \sum_{j=0}^{n-1} C \cdot G^{n-j-1} H \cdot U(jT) + D \cdot U(nT)$$

$$Y(nT) - C \cdot G^n X(0) = C \cdot G^{n-1} H \cdot U(0) + CG^{n-2} HU(T) + \ldots + CHU((n-1)T) + D \cdot U(nT)$$

$$Y(nT) - CG^n X(0) = \begin{bmatrix} D & CH & CGH & \ldots & CG^{n-1}H \end{bmatrix} \begin{bmatrix} U(nT) \\ U((n-1)T) \\ \vdots \\ U(0) \end{bmatrix} \qquad (34)$$

La condición necesaria y suficiente para que sea el sistema de Controlabilidad de salida completa es que la matriz m x (n+1)r

$$\begin{bmatrix} D & CH & CGH & \ldots & CG^{n-1}H \end{bmatrix}$$

sea de rango = m ,o que los vectores de esta matriz sean linealmente independientes.

TEOREMA : OBSERVABILIDAD.

Se dice que un sistema es observable en el tiempo to ,si con el sistema en el estado X(to),es posible determinar este estado partiendo de la observación de la salida durante un intervalo de tiempo finito.

El sistema es observable si,dada la salida Y(kT) para periodos de muestreo finitos,es posible determinar el vector de estado inicial X(0).

DEMOSTRACION:

Debe considerarse al sistema como no forzado por lo siguiente. Si el sistema es el siguiente en tiempo discreto,lineal e invariante en el tiempo

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (35)$$

$$Y(kT) = C \cdot X(kT) + D \cdot U(kT) \qquad (36)$$

Donde

X(kT) : Vector de estado de dimensión 'n'

U(kT) : Vector de entrada de dimensión 'r'

Y(kT) : Vector de salida de dimensión 'm'

G : Matriz de estado de orden n x n

H : Matriz de entrada de orden n x r

C : Matriz de salida m x n

D : Matriz m x r

T : Período de muestreo.

Entonces:

$$X(kT) = G^{k}\cdot X(0) + \sum_{j=0}^{k-1} G^{k-j-1} H\cdot U(jT) \qquad (37)$$

y

$$Y(kT) = C\cdot G^{k}X(0) + \sum_{j=0}^{k-1} C\cdot G^{k-j-1}H\cdot U(jT) + D\cdot U(kT) \qquad (38)$$

Como las matrices G,H,C y D son conocidas y también U(kT) es conocida,el segundo y tercer termino es una magnitud conocida.Por lo tanto,se le puede restar del valor observado de Y(kT).Esto implica que para investigar una condición necesaria y suficiente de observabilidad,basta con considerar el sistema siguiente.

Considerando el sistema lineal,tiempo discreto e invariante conel tiempo

$$X[(k+1)T] = G\cdot X(kT) \qquad (39)$$

$$Y(kT) = C\cdot X(kT) \qquad (40)$$

Donde

X(kT) : Vector de estado de dimensión ''n''

Y(kT) : Vector de salida de dimensión ''m''

G : Matriz de estado de orden n x n

C : Matriz de salida de orden m x n

La solución es:

$$X(kT) = G^{k}X(0) \qquad (41)$$

$$Y(kT) = C\cdot G^{k}X(0) \qquad (42)$$

Obsevabilidad completa significa que, dadas $y(0), y(t), \ldots y(nt)$, se puede determinar $X_1(0), X_2(0), \ldots, X_n(0)$. Para determinar n incógnitas se necesitan solamente n valores de $y(kT)$ por tanto, $N=n-1$, usamos n valores de $y(kt)$ ó $y(0), y(t), \ldots, y((n-1)t)$ para determinar $X_1(0), X_2(0), \ldots, X_n(0)$.

Para un sistema observable, dadas.

$$
\begin{aligned}
Y(0) &= C \cdot X(0) \\
Y(T) &= C \cdot G \cdot X(0) \\
&\vdots \\
Y((n-1)T) &= C \cdot G^{n-1} X(0)
\end{aligned}
\tag{43}
$$

Se deben poder determinar $X_1(0), X_2(0), \ldots, X_n(0)$.

Notando que $Y(kT)$ es un vector m, las n ecuaciones simultáneas precedentes dan $n \cdot m$ ecuaciones, todas involucrando a $X_1(0), X_2(0), \ldots, X_n(0)$.

$$
\begin{bmatrix} Y(0) \\ Y(T) \\ \vdots \\ Y((n-1)T) \end{bmatrix} = \begin{bmatrix} C \\ CG \\ \vdots \\ CG^{n-1} \end{bmatrix} [X(0)]
\tag{44}
$$

Para obtener de estas $n \cdot m$ ecuaciones un conjunto único de soluciones $X_1(0), X_2(0), \ldots, X_n(0)$, hay que poder tener entre ellas n ecuaciones linealmente independientes de la siguiente matriz, es decir la matriz $[n \cdot m \times n]$

$$\begin{bmatrix} C \\ CG \\ \vdots \\ CG^{n-1} \end{bmatrix}$$

sea de rango = n.

Notando que el rango de la matriz anterior y el de su transpuesta conjugada son iguales.

$$\text{Rango}(\text{Matriz}) = \text{Rango}(\text{Matriz}^T)$$

se puede establecerse que la matriz

$$\begin{bmatrix} C & CG & \dots & CG^{n-1} \end{bmatrix}^T = \tag{45}$$

$$= \begin{bmatrix} C^T & (CG)^T \dots (CG^{n-1})^T \end{bmatrix} =$$

$$= \begin{bmatrix} C^T & G^TC^T & \dots & (G^T)^{n-1}C^T \end{bmatrix}$$

de observabilidad de que sea Rango = n o que tenga n vectores columna linealmente independientes.

GENERACION DE ALGORITMOS PARA LAS RESPUESTAS DEL SISTEMA.

Generación de algoritmos para el análisis de los sistemas en el espacio de estados.

OBSTENCION Y PROPIEDADES DE LA MATRIZ DE TRANSICION DE ESTADO.

$$\phi(t)=e^{At}=I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 +\dots+\frac{1}{k!}A^kt^k+\dots = \sum_{k=0}^{\infty}\frac{A^kt^k}{k!} \qquad (46)$$

$$[\phi(t)]^k=\phi(kt)=[e^{At}]^k = e^{kAt} =e^{A(kt)}=e^{At}e^{At}\ \dots\ e^{At}\ ;(k: terminos)$$

Evaluación de la serie de $G = e^{A\cdot T}$ en el programa:

El número de términos para la evaluación de la serie depende del criterio de convergencia E escogido para que no halla desbordamiento en los cálculos, se paran los cálculos cuando la suma de los terminos sea mayor a un E escogido, o dando un número de términos de la serie para su cálculo.

RESPUESTA LIBRE DEL SISTEMA EN TIEMPO CONTINUO.

ECUACION HOMOGENEA.

(Algoritmo Iterativo)

Si tenemos el sistema lineal, tiempo continuo e invariante en el tiempo:

$$\dot{X}(t) = A\cdot X(t) \qquad (47)$$

Donde

$X(t)$: Vector de estado de dimención 'n'

A : Matriz de estado de orden [nxn]

tenemos que la solución para un tiempo T,es:

$$X(T) = e^{A \cdot T} \cdot X(0) \tag{48}$$

para un tiempo $2 \cdot T$ es:

$$X(2T) = e^{2 \cdot AT} \cdot X(0) = e^{AT} \cdot (e^{AT} \cdot X(0)) = [e^{AT}] \cdot X(T)$$

en general para un tiempo kT es:

$$X(kT) = e^{k \cdot AT} \cdot X(0) = e^{AT} \cdot e^{(k-1)T} \cdot X(0) = e^{AT} \cdot X((k-1)T)$$

En general para otro tiempo $(k+1)T$ tenemos

$$X((k+1)T) = e^{(k+1)AT} \cdot X(0) \tag{49}$$

$$X((k+1)T) = e^{AT} \cdot [e^{kAT} X(0)] \tag{50}$$

$$X((k+1)T) = e^{AT} \cdot X(kT) \tag{51}$$

Donde esta última ecuación es la ecuación Discretizada.

Donde:

$$G = e^{AT} \tag{52}$$

se puede sustituir quedando la ecuación:

$$X((k+1)T) = G \cdot X(kT) \qquad (53)$$

DISCRETIZACION DE LA ECUACION DE ESPACIO DE ESTADOS DE TIEMPO CONTINUO A LA ECUACION DE ESTADO DE TIEMPO-DISCRETO.

Si tenemos el sistema lineal, tiempo continuo e invariante en el tiempo:

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t) \qquad (54)$$

$$Y(t) = C \cdot X(t) + D \cdot U(t) \qquad (55)$$

Donde

- $X(t)$: Vector de estado de dimensión 'n'
- $U(t)$: Vector de entrada de dimensión 'r'
- $Y(t)$: Vector de salida de dimensión 'm'
- A : Matriz de estado de orden [n × n]
- B : Matriz de entrada de orden [n × r]
- C : Matriz de salida de orden [m × n]
- D : Matriz de orden [m × r]

Donde tenemos que su solución es:

$$X(t) = e^{At}X(0) + \int_0^t e^{A(t-s)} B \cdot U(s)ds \qquad (56)$$

Si tomamos un incremento de tiempo, $T=\Delta t$

para un tiempo cualquiera tomamos un tiempo igual a kT

Entonces su solución general para un tiempo cualquiera es:

$$X(kT)=e^{A(kT)}X(0) + \int_0^{kT} e^{A(kT-s)}B\cdot U(s)ds \qquad (57)$$

Para otro tiempo posterior equiesparciado en $\Delta t = T$; tomamos un tiempo $= k\Delta t + \Delta t = (k+1)\Delta t = (k+1)T$; Entonces su solución general para un tiempo cualquiera es:

$$X((k+1)T) = e^{A((k+1)T)}X(0) + \int_0^{(k+1)T} e^{A((k+1)T-s)}B\cdot U(s)ds$$

$$X((k+1)T) = e^{A\cdot T}e^{A\cdot kT}X(0) + e^{A\cdot T}\int_0^{(k+1)T} e^{A(k\cdot T-s)}B\cdot U(s)ds$$

$$X((k+1)T) = e^{A\cdot T}\left[e^{A\cdot kT}X(0) + \int_0^{(k+1)T} e^{A(kT-s)}B\cdot U(s)ds\right] =$$

$$= e^{A\cdot T}\left[\underline{e^{A\cdot kT}X(0) + \int_0^{kT} e^{A(kT-s)}B.U(s)ds} + \int_{kT}^{(k+1)T} e^{A(kT-s)}BU(s)ds\right]$$

donde el termino subrayado es la solución General de X(kT)

Entonces

$$X((k+1)T) = e^{A\cdot T}\left[X(kT) + \int_{kT}^{(k+1)T} e^{A(kT-s)}B\cdot U(s)ds\right] \qquad (58)$$

$$X((k+1)T) = e^{A\cdot T}\cdot X(kT) + \int_{kT}^{(k+1)T} e^{A(kT+T-s)}B\cdot U(s)ds \qquad (59)$$

$$X((k+1)T) = e^{A\cdot T}X(kT) + \left[\int_{kT-kT}^{(k+1)T-kT} e^{A(kT+T-(s+kT))}BU(s+kT)ds \right]$$

donde, si para dos instantes de muestreo consecutivos tenemos que:

$$U(t) = U(kT) \quad \text{para} \quad kT \leq t < (k+1)T$$

entonces tenemos.

$$X[(k+1)T] = e^{A\cdot T}X(kT) + \left[\int_0^T e^{A(T-s)}B\cdot ds \right] U(kT)$$

Donde esta ecuación es del tipo discreta, donde.

$$G(T) = e^{A\cdot T} \tag{60}$$

y

$$H(T) = \left[\int_0^T e^{A(T-s)}Bds \right] \tag{61}$$

Enonces tenemos que:

$$X((k+1)T) = G(T)\cdot X(kT) + H(T)\cdot U(kT) \tag{62}$$

Donde $G(T)$ y $H(T)$ dependen del periodo de muestreo T, una vez fijado este período, G y H son matrices constantes.

Entonces:

$$X((k+1)T) = G\cdot X(kT) + H\cdot U(kT) \tag{63}$$

y la ecuación de salida

$$Y(kT) = C\cdot X(kT) + D\cdot U(kT)$$

Donde las matrices C y D son matrices constantes y no dependen del período de muestreo.

Las anteriores ecuaciones del sistema discretizado las podemos utilizar para obtener los puntos de las diferentes respuestas.como las respuestas totales del sistema,ya que es más rápido discretizar el sistema de tiempo-continuo y obtener iterativamente las respuestas que obtener las respuestas por la ecuación de la solución general del sistema de tiempo continuo. De hecho las respuestas totales del sistema utilizando la ecuación de la solución general del sistema en tiempo-continuo fue hecho el algoritmo y el programa,encontrando bastante tiempo-de-computadora.

Posteriormente bajo algunas condiciones haremos una fórmula más reducida y daremos algunos comentarios:

En general para convertir el sistema de ecuaciones de estado de tiempo continuo a un sistema de ecuaciones de estado de tiempo discreto,necesariamente tenemos que hacer una aproximación.
El período entre dos puntos es decir el incremento Δt nos conviene hacerlo muy pequeño.
Esto es por:

$$G(T) = G(\Delta t) \cong G(0) = e^{A(0)} = I \tag{64}$$

es aproximadamente la Matriz Identidad.
Para obtener una expresión más sencilla de la obtención de la matriz H.
si $\Delta t = T$,

$$H(\Delta t) = H(T) = \left[\int_0^T e^{A(T-t)} B \cdot dt \right] = \tag{65}$$

$$\left[\int_0^T e^{A(T-t)} B \cdot dt \right] = e^{A \cdot T} \int_0^T [\; e^{-A \cdot t} dt \;] \; B =$$

$$= e^{A \cdot T} \left[\int_0^T (I - At + \frac{A^2 t^2}{2!} - \frac{A^3 t^3}{3!} + \dots) dt \right] B \cdot =$$

$$= e^{A \cdot T} \left[IT - \frac{AT^2}{2} + \frac{A^2 T^3}{2! \cdot 3} - \frac{A^3 T^4}{3! \cdot 4} + \dots \right] B$$

$$= e^{A \cdot T} \left[IT - \frac{AT^2}{2!} + \frac{A^2 T^3}{3!} - \frac{A^3 T^4}{4!} + \dots \right] B \qquad (66)$$

Entonces la solución general del sistema es:

$$X[(k+1)T] = e^{A \cdot T} X(kT) + e^{A \cdot T} \left[IT - \frac{AT^2}{2!} + \frac{A^2 T^3}{3!} - \frac{A^3 T^4}{4!} + \dots \right] B \cdot U(kT)$$

La anterior ecuación es una solución general,pero si tenemos que A es no singular entonces puede simplificarse esta última ecuación, dando la siguiente relación matemática.

$$e^{-AT} = \left[I - AT + \frac{A^2 T^2}{2!} - \frac{A^3 T^3}{3!} + \frac{A^4 T^4}{4!} - \dots \right]$$

$$-e^{-AT} = \left[-I + AT - \frac{A^2 T^2}{2!} + \frac{A^3 T^3}{3!} - \frac{A^4 T^4}{4!} + \dots \right]$$

$$(-e^{-AT} + I) = (AT - \frac{A^2 T^2}{2!} + \frac{A^3 T^3}{3!} - \frac{A^4 T^4}{4!} + \dots)$$

$$(-e^{-AT} + I) = A \left[I \cdot T - \frac{AT^2}{2!} + \frac{A^2 T^3}{3!} - \frac{A^3 T^4}{4!} + \dots \right]$$

$$A^{-1}[-(e^{-AT}-I)] = A^{-1}A\left[I\cdot T - \frac{AT^2}{2!} + \frac{A^2T^3}{3!} - \frac{A^3T^4}{4!} + \dots\right]$$

$$-(A^{-1})(e^{-AT}-I) = \left[IT - \frac{AT^2}{2!} + \frac{A^2T^3}{3!} - \frac{A^3T^4}{4!} + \dots\right] \qquad (67)$$

Entonces de la solución general del sistema,obtenida anteriormente,tenemos que.

$$H(T) = \int_0^T e^{A(T-t)}Bdt = e^{AT}\left[IT - \frac{AT^2}{2!} + \frac{A^2T^3}{3!} - \frac{A^3T^4}{4!} + \dots\right]B \qquad (68)$$

Por lo tanto se puede expresar por medio de la relación de la siguiente manera

$$H(T) = e^{AT}\left[-(A^{-1})(e^{-AT} - I)\right]B \qquad (69)$$

$$H(T) = \left[-(A^{-1})\right]\left[e^{AT}e^{-AT} - e^{AT}\right]B \qquad (70)$$

$$H(T) = \left[-(A^{-1})\right]\left[I - e^{A\cdot T}\right]B \qquad (71)$$

$$H(T) = A^{-1}\left[-I + e^{A\cdot T}\right]B \qquad (72)$$

donde queda como:

$$H(T) = A^{-1}\left[e^{A\cdot T} - I\right]B \qquad (73)$$

Esta es la fórmula para la obtención de $H(T) = H(\Delta t)$,donde la matriz **A** del sistema tiene que ser no-singular.Entonces **H** se puede simplificar a la fórmula anterior para su cálculo en los análisis de los algoritmos del programa.

Por lo tanto podemos espresar la solución por medio de la relación matemática siguiente , mientras la matriz **A** del sistema sea no-singular.

$$X[(k+1)T] = \left[e^{\mathbf{A}\cdot T} \right] X(kT) + \left[\mathbf{A}^{-1} (e^{\mathbf{A}\cdot T} - I) B\cdot \right] U(kT) \tag{74}$$

RESPUESTA TOTAL DEL SISTEMA DE LA ECUACION NO-HOMOGENEA TIEMPO-CONTINUO.

Para obtener las respuestas totales del sistema de una ecuación de tiempo-continuo es mejor hacer primeramente la discretización del sistema,y de este modo obtener las respuestas totales del sistema iterativamente punto por punto de la ecuación de tiempo discreta, de hecho al programarlo por la solución general de tiempo-continuo para obtener las respuestas totales del sistema dicho algoritmo consumia bastante o exageradamente tiempo de maquina por lo que pensamos en dececharlo.De esta manera el programa cálcula las respuestas totales por la ecuación de tiempo discreta iterativamente.De esta forma podemos llegar a la ecuación de estado:

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \tag{75}$$

y la ecuación de salida:

$$Y(kT) = C \cdot X(kT) + D \cdot U(kT) \tag{76}$$

La cuál podemos obtener la solución recursivamente de los sistemas

IV.- TECNICAS COMPLEMANTARIAS.

Las técnicas complementarias que se análizaran en este capitulo serán las siguientes,para sistemas multivariables.

(1).- Realimentación de Estado.

(2).- Diseño del Observador.

(3).- Diseño del Observador de orden reducido.

DISEÑO POR ASIGNACION DE POLOS.

Esta técnica de diseño de la asignacion de polos para un determinado sistema,es el diseñar los polos de lazo-cerrado del sistema en cuestión,en este caso asumiremos que todas las variables de estado son medibles y ademas estan disponibles para realimentarse.

Puede demostrarse que si el sistema es controlable de estado completamente,entonces existe una matriz K que determinara un conjunto arbitrário de valores propios de (A-BK),es decir los polos de el sistema de lazo cerrado pueden ser asignados en cualquier localidad por medio de la realimentación de estado de una matriz de ganancia de realimentación de estado K,esta condición se puede demostrar o probar matemáticamente la cual se puede ver en algún libro sobre el tema.

Esto implica también que cualquier sistema inestable puede ser estabilizado por realimentación de estado si todos los estados son controlables.

Es importante notar que de aqui para delante haremos las demostraciones para sistemas de tiempo discreto ya que para los sistemas de tiempo-continuo las demostraciones se hacen en forma paralelamente a la discreta.

DISEÑO DE LA REALIMENTACION DE ESTADO O ASIGNACION DE POLOS. PARA SISTEMAS MULTIVARIABLES.

Si tenemos el sistema multivariable de tiempo discreto, lineal e invariante con el tiempo

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (1)$$

Donde

$X(kT)$: Vector de estado de dimensión 'n'

$U(kT)$: Vector de entrada de dimensión 'r'

G : Matriz de estado de orden $n \times n$

H : Matriz de entrada de orden $n \times r$

T : Período de muestreo.

Si la señal de control $U(k)$ es

$$U(kT) = -K \cdot X(kT) \qquad (2)$$

Donde K es la matriz de ganancia de Realimentación de estado[rxn].

Entonces el sistema se conbierte en.

$$X[(k+1)T] = G \cdot X(kT) + H \cdot (-K \cdot X(kT)) \qquad (3)$$

$$X[(k+1)T] = (G - H \cdot K)X(kT) \qquad (4)$$

Donde los valores caracteristicos de (G-H.K) son los polos de lazo cerrado diseñados, U_1 , U_2 , . . . , U_n .

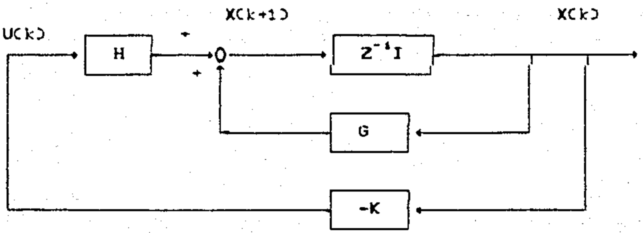


FIGURA 1: SISTEMA DE CONTROL CON REALIMENTACION DE ESTADO

$$U(k) = - K \cdot X(k)$$

REALIMENTACION DE ESTADO PARA SISTEMAS ESCALARES.

Si tenemos el sistema escalar de tiempo discreto, lineal e invariante con el tiempo

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (5)$$

Donde

$X(kT)$: Vector de estado de dimensión 'n'

$U(kT)$: Vector de entrada de dimensión '1' (escalar)

G : Matriz de estado de orden n x n

H : Matriz de de entrada de orden n x 1

T : Período de muestreo.

Donde el sistema es controlable de estado completamente. Y la realimentación de estado es:

$$U(kT) = -K \cdot X(kT) \qquad ;K \quad [1 \times n]$$

Para la determinación de la Matriz de Ganancia de Realimentación hay varios caminos para llegar a su determinación de varias fórmulas condensadas. Nosotros optamos por Ackermann's para su demostración y realización en el programa, la cual la fórmula de la determinación de la matriz de ganancia de realimentación K para sistemas de única entrada o escalar es llamada fórmula de Ackermann's.

Donde deseamos colocar los polos de lazo cerrado en.

$$Z = U_1 \ , \ Z = U_2 \ . \ . \ . \ . \ . \ , \ Z = U_n$$

Diseñando la ecuación caracteristica queda como.

$$|ZI-G+HK| = (Z-U_1)\,(Z-U_2)\;.\;.\;.\;(Z-U_n) \tag{6}$$

$$|ZI-(G-HK)| = Z^n + \alpha_1 Z^{n-1} + \alpha_2 Z^{n-2} + \ldots + \alpha_{n-1} Z + \alpha_n = 0$$

Si definimos a

$$\hat{G} = G - HK \tag{7}$$

Por el teorema de Cayley-Hamilton el estado de $\hat{G}$ debe satisfacer la ecuación caracteristica.

$$\hat{G}^{n} + \alpha_1 \hat{G}^{n-1} + \alpha_2 \hat{G}^{n-2} + \ldots + \alpha_{n-1}\hat{G} + \alpha_n I = \phi(\hat{G}) = 0 \tag{8}$$

utilizaremos la última ecuación para derivar la fórmula Ackermann's. Si consideramos las siguientes identidades.

$$I = I$$

$$\hat{G} = G-HK$$

$$\hat{G}^2 = (G-HK)^2 = G^2-GHK-HK\hat{G}$$

$$\vdots$$

$$\hat{G}^n = (G-HK)^n = G^n-G^{n-1}HK-\ldots-HK\hat{G}^{n-1}$$

Sustituyendo en la ecuación de Cayley-Hamilton, tenemos.

$$\alpha_n I + \alpha_{n-1}\hat{G} + \alpha_{n-2}\hat{G}^2 + \ldots + \hat{G}^n = \phi(\hat{G}) = 0 = \tag{9}$$

$$= \alpha_n I + \alpha_{n-1}G + \alpha_{n-2}G^2 + \ldots + G^n - \alpha_{n-1}HK - \alpha_{n-2}GHK -$$

$$- \alpha_{n-2}HK\hat{G} - \ldots - G^{n-1}HK - \ldots - HK\hat{G}^{n-1} \tag{10}$$

Entonces

$$\phi(\hat{G}) = 0 = \phi(G) - \alpha_{n-1}HK - \alpha_{n-2}GHK - \alpha_{n-2}HK\hat{G} - \ldots - HK\hat{G}^{n-1} - G^{n-1}HK$$

$$0 = \phi(G) - \begin{bmatrix} H & GH & \ldots & G^{n-1}H \end{bmatrix} \begin{bmatrix} \alpha_{n-1}K + \alpha_{n-2}K\hat{G} + \ldots + K\hat{G}^{n-1} \\ \alpha_{n-2}K + \alpha_{n-3}K\hat{G} + \ldots + K\hat{G}^{n-2} \\ \vdots \\ K \end{bmatrix} \tag{11}$$

$$\phi(G) = \begin{bmatrix} H & GH & \ldots & G^{n-1}H \end{bmatrix} \begin{bmatrix} \alpha_{n-1}K + \alpha_{n-2}K\hat{G} + \ldots + K\hat{G}^{n-1} \\ \alpha_{n-2}K + \alpha_{n-3}K\hat{G} + \ldots + K\hat{G}^{n-2} \\ \vdots \\ K \end{bmatrix} \tag{12}$$

Como el sistema es controlable de estado completamente, entonces

$\begin{bmatrix} H & GH & \ldots & G^{n-1}H \end{bmatrix}$ = Rango = (n) , existe la inversa.entonces la ecuación (12) puede ser modificada a la forma

$$\begin{bmatrix} \alpha_{n-1}K+\alpha_{n-2}K\hat{G}+\ldots+K\hat{G}^{n-1} \\ \alpha_{n-2}K+\alpha_{n-3}K\hat{G}+\ldots+K\hat{G}^{n-2} \\ \vdots \\ K \end{bmatrix} = \begin{bmatrix} H & GH & \ldots & G^{n-1}H \end{bmatrix}^{-1}\phi(G)$$

Premultiplicando los dos lados de la última ecuación por la matriz [0 0 0 . . . 1] , Entonces optenemos

$$[K] = \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \end{bmatrix}\begin{bmatrix} H & GH & \ldots & G^{n-1}H \end{bmatrix}^{-1}\phi(G) \qquad (13)$$

donde

$$\phi(G) = G^{n} + \alpha_{1}G^{n-1}+\ldots+\alpha_{n-1}G + \alpha_{n}I$$

DISEÑO POR ASIGNACION DE POLOS POR REALIMENTACION DE ESTADO PARA SISTENAS MULTIVARIABLES.

Para el diseño de la matriz de ganancia de realimentación de estado K [rxn],para sistemas de múltiples-entradas,también existen varios caminos para obtener la matriz K [rxn] como en el caso de sistemas de única-entrada, en este caso de sistemas de múltiples-entradas la obtención de la matriz de ganancia de realimentación de estado K [rxn] nosotros optamos por útilizar y demostrar uno de los varios métodos que existen.

ANALISIS DE ESPACIO DE ESTADOS DE REALIMENTACION DE ESTADO DE SISTEMAS MULTIVARIABLES.

Si tenemos el sistema en tiempo discreto, lineal e invariante con el tiempo

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (14)$$

Donde

$X(kT)$: Vector de estado y es de dimensión 'n'

$U(kT)$: Vector de entrada o de control de dimensión 'r'

G : Matriz de estado de orden $n \times n$

H : Matriz de entrada de orden $n \times r$

T : Período de muestreo.

El problema del diseño es hallar la matriz de realimentación K [rxn] si la señal de control es:

$$U(kT) = -K \cdot X(kT)$$

donde el sistema es de controlabilidad de estado.

primeramente nosotros construiremos un sistema de entrada única como sigue.

$$X[(k+1)T] = G\cdot X(kT) + \hat{H}\cdot U(kT) \qquad (15)$$

donde $T = \Delta t$

y la matriz $\hat{H}$ [nx1] sera definida como.

$$\hat{H} = H\cdot W \qquad (16)$$

Donde W es y debe ser una matriz [rx1] .La matriz W mostrada debe ser escogida para que el par (G , $\hat{H}$) sea controlable de estado. Entonces,nosotros podemos aplicar la realimentación de estado.

$$U(kT) = -\hat{K}\cdot X(kT) \qquad (17)$$

entonces

$$X[(k+1)T] = G\cdot X(kT) + \hat{H}\cdot U(kT) \qquad (18)$$

$$X[(k+1)T] = G\cdot X(kT) + \hat{H}\cdot(-\hat{K}\cdot X(kT)) \qquad (19)$$

$$X[(k+1)T) = G\cdot X(kT) - \hat{H}\cdot\hat{K}\cdot X(kT) \qquad (20)$$

$$X[(k+1)T) = (G-\hat{H}\cdot\hat{K})\cdot X(kT) \qquad (21)$$

donde $\hat{H} = H\cdot W$

Donde el par (G , $\hat{H}$) es controlable.

La asignación de los valores caracteristicos de

$$(G-\hat{H}\cdot\hat{K}) = (G-H\cdot W\cdot\hat{K}) \tag{22}$$

son los mismos valores característicos localizados que los de

$$(G-H.K) = (G-H\cdot W\cdot\hat{K}) \tag{23}$$

ya que la determinada K esta dada por

$$K = W.\hat{K} \qquad [r\times n].$$

Puesto que la ecuación.

$$X[(k+1)T] = G\cdot X(kT) - \hat{H}\cdot\hat{K}\cdot X(kT)$$

donde esta ecuación se combierte en.

$$X[(k+1)T] = G\cdot X(kT) - H\cdot(W\cdot\hat{K})\cdot X(kT) \tag{24}$$

Donde se ve que la matriz de realimentación esta dada por.

$$K = W\cdot\hat{K}$$

$$X[(k+1)T] = G\cdot X(kT) - H.K\cdot X(kT)$$

entonces las matrices deben ser iguales.

$$H\cdot K = \hat{H}\cdot\hat{K} \tag{25}$$

y la asignación de los valores característicos son iguales.

Se ve que en general para sistemas multivariables la matriz W no es única, y tiene que satisfacer la condición de (G ,H·W) sea completamente controlable de estado. Hemos reducido el problema matricial a un problema escalar, y $\hat{K}$ puede ser obtenida con las fórmulas del caso escalar. Por lo tanto la matriz K [r×n] no es única para sistemas multivariables.

Dado el sistema multivariable, lineal, de tiempo discreto e invariante con el tiempo.

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (26)$$

Donde

X(kT) : Vector de estado de dimensión 'n'

U(kT) : Vector de entrada o de control de dimensión 'r'

G : Matriz de estado de orden [n×n]

H : Matriz de entrada de orden [n×r]

T : Periodo de muestreo.

La principal cuestión es que si (G,H) es controlable de estado completamente y la matriz G es ciclica siempre existe un vector W tal que podemos tener (G,H·W) que sea controlable de estado completamente, escogiendo una matriz W.

Lo cual lo probaremos enseguida.

Si tenemos una realimentación de estado lineal de la forma.

$$U(kT) = r(kT) - K \cdot X(kT) \qquad (27)$$

donde K Matriz de orden [r × n]

Si G es no ciclica, introducimos.

$$U(kT) = W(kT) - K_1 \cdot X(kT) \qquad (28)$$

Entonces

$$X[(k+1)T] = G \cdot X(kT) + H \cdot (W(kT) - K_1 \cdot X(kT))$$

$$X[(k+1)T] = (G - H \cdot K_1) \cdot X(kT) + H \cdot W(kT) \tag{29}$$

Tal que si

$$\hat{G} \triangleq G - H \cdot K_1 \tag{30}$$

en

$$X[(k+1)T] = \hat{G} \cdot X(kT) + H \cdot W(kT) \tag{31}$$

es la matriz $\hat{G}$ es ciclica.

Entonces $(G\ ,\ H)$ es controlable, así como $(\hat{G}\ ,\ H)$. Por consiguiente existe un vector V (real) [rx1] tal que $(\hat{G}\ , H \cdot V)$ es controlable y además es ciclica la matriz $\hat{G}$. Nuevamente introducimos otra realimentación de estado.

$$W(kT) = r(kT) - K_2 \cdot X(kT) \tag{32}$$

como se muestra en la figura 2.

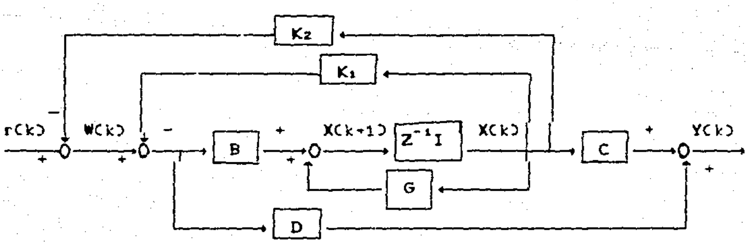


FIGURA 2

Con

$$K_2 = V \cdot K$$

donde **K** es una matriz $[1 \times n]$, entonces de.

$$X[(k+1)T] = (G - H \cdot K_1) X(kT) + H \cdot W(kT)$$

$$X[(k+1)T] = \hat{G} \cdot X(kT) + H \cdot W(kT) \qquad (33)$$

Donde podemos hacer a este sistema como.

$$X[(k+1)T] = \hat{G} \cdot X(kT) + (H \cdot V) \cdot W(kT) \qquad (34)$$

Donde $\hat{G}$ es ciclica y entonces $(\hat{G}, H \cdot V)$ es controlable de estado, donde los valores característicosde $(\hat{G} - H \cdot V \cdot K)$ pueden ser asignados arbitrariamente, por.

$$W(kT) = -K \cdot X(kT)$$

entonces.

$$X[(k+1)T] = \hat{G} \cdot X(kT) + (H \cdot V) W(kT) \qquad (35)$$

con realimentación.

$$X[(k+1)T] = \hat{G} \cdot X(kT) + (H \cdot V)(-K \cdot X(kT)) =$$

$$= (\hat{G} - H \cdot V \cdot K) X(kT) \qquad (36)$$

donde

$$K_2 = V \cdot K$$

entoces podemos hacer.

$$W(kT) = r(kT) + K_2 \cdot X(kT) \qquad (37)$$

$$X[(k+1)T] = \hat{G} \cdot X(kT) + H(r(kT) - K_2 \cdot X(kT)) =$$

$$= (\hat{G} - H \cdot K_2)X(kT) + H \cdot r(kT) \qquad (38)$$

$$X[(k+1)T] = (\hat{G} - H \cdot V \cdot K)X(kT) + H \cdot r(kT)$$

Por combinación de la realimentación de estado.

$$U(kT) = W(kT) - K_1 \cdot X(kT) \qquad (39)$$

y la realimentación de estado.

$$W(kT) = r(kT) - K_2 \cdot X(kT)$$

Entonces

$$U(kT) = r(kT) - K_2 \cdot X(kT) - K_1 \cdot X(kT) \qquad (40)$$

$$U(kT) = r(kT) - (K_1 + K_2)X(kT) \triangleq r(kT) - K \cdot X(kT) \qquad (41)$$

La cuál es la solución final.

DISEÑO DE OBSERVADORES O ESTIMADORES DE ESTADO.

Para aplicar físicamente la realimentación de estado, es necesario realimentar todas las variables de estado,desgraciadamente,en la práctica no todas las variables de estado son accesibles y podemos suponer que solamente las entradas y las salidas de cualquier sistema son medibles.

Un Observador de Estado o Estimador de Estado es un subsistema en el sistema de control que realiza la estimación de variables de estado a partir de las informaciones recibidas de las mediciones de las salidas y de las variables de control(entradas).

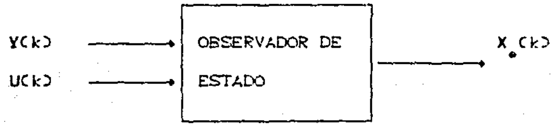


FIGURA 3 : DIAGRAMA DE UN OBSERVADOR DE ESTADO.

El diseño del observador de estado que análizaremos y utilizaremos en el programa sera digital es decir en sistemas de ecuacuones en tiempo discreto,ya que el análisis para el diseño del observador de sistemas de datos continuos se hace en paralelo.

En el análisis siguiente asumiremos que el estado actual del sistema a considerar X(kT) no puede ser medido directamente.Entonces el estado X(kT) debe ser estimado.

Un observador tiene una configuración con U(kT) y Y(kT) como entradas y debe ser diseñado que el estado observado $X_{\bullet}(kT)$ deba ser próximo como sea posible a el estado actual de $X(kT)$ o sea tendra la capacidad de minimizar el error entre $X(kT)$ y $X_e(kT)$ automáticamente, como no podemos comparar $X_e(kT)$ con $X(kT)$, una alternativa es, $Y_e(kT)$ compararla con $Y(kT)$, donde $Y_e(k)=C\cdot X_e(k)$.

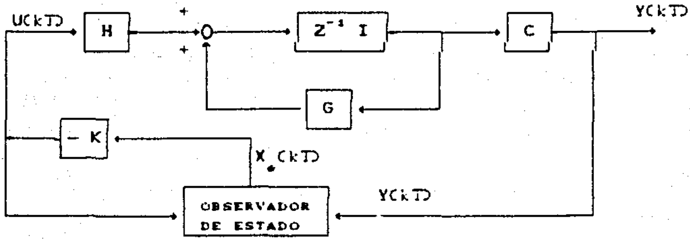


FIGURA 4 : SISTEMA REGULADOR CON UN OBSERVADOR DE ESTADO.

Un observador de estado puede ser diseñado si y solo si la condición de observabilidad es satisfecha la cual la demostración no la daremos aqui ya que es muy larga, la demostración se puede ver en algun libro sobre el tema.

Dado un sistema lineal, tiempo discreto e invariante con el tiempo

$$X(k+1) = G\cdot X(k) + H\cdot U(k) \qquad (42)$$

$$Y(k) = C \cdot X(k) \qquad (43)$$

Donde

$X(k)$: Vector de estado de dimensión 'n'

$Y(k)$: Vector de salida de dimensión 'm'

$U(k)$: Vector de entrada de dimensión 'r'

G : Matriz de estado de orden $n \times n$

H : Matriz de entrada de orden $n \times r$

C : Matriz de salida de orden $m \times n$

T : Período de muestreo.

Donde asumiremos que la matriz G es no singular.

El vector de estado $X(k)$ puede ser construido a partir de combinaciones lineales de la salida $Y(k)$, la entrada $U(k)$,y los valores diferencias pasados de estas variables, si el sistema es Observable. Como se demostrara.

DEMOSTRACION:

La ecuación (42) puede ser escrita como

$$X(k-1) = G^{-1}X(k) - G^{-1}H \cdot U(k-1) \qquad (k \geq 1) \qquad (44)$$

En general,

$$X(k-1) = G^{-1}X(k-n+1) - G^{-1}H \cdot U(k-n) \qquad (k \geq n) \qquad (45)$$

y de la ecuación (43) puede ser escrita como

$$Y(k-1) = C \cdot X(k-1) \qquad (k \geq 1)$$

sustituyendo la ecuaciones (44) en la ecuación (45) tenemos.

$$Y(k-1) = C \cdot G^{-1}X(k) - C \cdot G^{-1}H \cdot U(k-1) \qquad (k \geq 1)$$

y

$$Y(k-2) = C\cdot X(k-2) =$$

$$= CG^{-2}X(k)-CG^{-2}H\cdot U(k-1)-CG^{-1}H\cdot U(k-2)$$

Continuando o generalizando el proceso, tenemos.

$$Y(k-N) = C\cdot G^{-N}X(k) - \sum_{i=1}^{N} CG^{-N+i-1}H\cdot U(k-i) \qquad (k\geq N) \qquad (46)$$

Entonces obtenemos matricialmente.

$$\begin{bmatrix} Y(k-1) \\ Y(k-2) \\ Y(k-3) \\ \vdots \\ Y(k-N) \end{bmatrix} = \begin{bmatrix} CG^{-1} \\ CG^{-2} \\ CG^{-3} \\ \vdots \\ CG^{-N} \end{bmatrix} X(k) - \begin{bmatrix} CG^{-1}H & 0 & 0 & \cdots & 0 \\ CG^{-2}H & CG^{-1}H & 0 & \cdots & 0 \\ CG^{-3}H & CG^{-2}H & CG^{-1}H & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CG^{-N}H & CG^{-N+1} & CG^{-N+2} & \cdots & CG^{-1}H \end{bmatrix} \begin{bmatrix} U(k-1) \\ U(k-2) \\ U(k-3) \\ \vdots \\ U(k-N) \end{bmatrix}$$

La ecuación matricial representa n×m ecuaciones con n incognitas en el vector de estado X(k).

El vector de estado X(k) puede ser determinado si la matriz ($R(\text{Matriz}) = R(\text{Matriz}^T)$)

$$\begin{bmatrix} CG^{-1} & \vdots & CG^{-2} & \vdots & \cdots & \vdots & CG^{-N} \end{bmatrix}^T$$

es de rango n, dando y conociendo el control de entradas y de salidas en la ecuación anterior X(k). Ademas vemos que la matriz es equivalente o igual del criterio de observabilidad sobre el sistema dado. Donde las matrices son equivalentes

$$\left[CG^{-1} \mid CG^{-2} \mid \ldots \mid CG^{-N+1} \mid CG^{-N}\right]^{T} \quad = \quad \text{sea rango } n$$

$$= \quad \left[CG^{N-1} \mid CG^{N-2} \mid \ \ldots \ \mid CG^{1} \mid C\right]^{T} \quad \text{sea rango } n.$$

Para sistemas en tiempo continuo se tiene

El vector de estado X(t) puede ser construido a partir de combinaciones lineales de la salida Y(t) ,la entrada U(t), y las derivadas de estas variables si el sistema es observable.

DEMOSTRACION:

Dado un sistema lineal, tiempo continuo e invariante con el tiempo

$$\dot{X}(t) = A \cdot X(t) + B \cdot U(t)$$

$$Y(t) = C \cdot X(t)$$

Donde

X(t) : Vector de estado de dimensión 'n'

U(t) : Vector de entrada de dimensión 'r'

Y(t) : Vector de salida de dimensión 'm'

A : Matriz de estado de orden [nxn]

B : Matriz de entrada de orden [nxr]

C : Matriz de salida de orden [mxn]

Si el sistema es Observable entonces se tiene que la matriz de

$$\left[\ C^{T} \quad A^{T}C^{T} \quad (A^{T})^{2}C^{T} \ \ \ldots \ \ (A^{T})^{p-1} \quad C^{T}\ \right] \text{ sea } R=n$$

Se demostrara que los estados pueden expresarse como combinaciones lineales de la salida Y(t),la entrada U(t) y sus derivadas.

$$\dot{Y}(t) = C\cdot\dot{X}(t)$$

$$\dot{Y}(t) = C\cdot A\cdot X(t) + C\cdot B\cdot U(t)$$

$$\dot{Y}-C\cdot B\cdot U(t) = C\cdot A\cdot X(t)$$

$$\ddot{Y}(t)-C\cdot B\cdot\dot{U}(t) = C\cdot A\cdot\dot{X}(t)$$

$$\ddot{Y}(t)-C\cdot B\cdot\dot{U}(t) = C\cdot A\cdot A\cdot X(t) + C\cdot A\cdot B\cdot U(t)$$

$$\ddot{Y}(t)-C\cdot B\cdot\dot{U}(t)-C\cdot A\cdot B\cdot U(t) = C\cdot A^{2}\cdot X(t)$$

Derivando n veces se tiene

$$Y^{(n)}(t)-CBU^{(n-1)}(t)-CABU^{(n-2)}(t)-\ldots-CA^{(n-1)}BU(t) = C\cdot A^{(n-1)}X(t)$$

Obtenemos:

$$\begin{bmatrix} Y \\ Y-CBU \\ Y-CBU-CABU \\ \vdots \\ Y^{(n)}-CBU^{(n-1)}-CABU^{(n-2)}-\ldots-CA^{(n-1)}BU \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^{2} \\ \vdots \\ CA^{n-1} \end{bmatrix} X(t)$$

El vector de estado X(t) puede ser expresado en función de Y(t) , U(t) y sus derivadas,donde la matriz

(como tenemos que $R((\text{Matriz}))=R((\text{Matriz})^{T})$)

Entonces

$$\begin{bmatrix} C & CA & CA^2 & \ldots & CA^{n-1} \end{bmatrix}^T =$$

$$= \begin{bmatrix} C^T & A^TC^T & (A^T)^2C^T & \ldots & (A^T)^{n-1}C^T \end{bmatrix}$$

debera ser de rango n,es decir si es Observable el sistema.

OBSERVADOR DE ESTADO DE ORDEN COMPLETO.

En este tipo de observador de estado completo su orden de este es el mismo que tiene el sistema. Por otro lado si es posible, no es necesario, pero es conveniente que el observador de estado tenga las mismas matrices G y H como del sistema original.

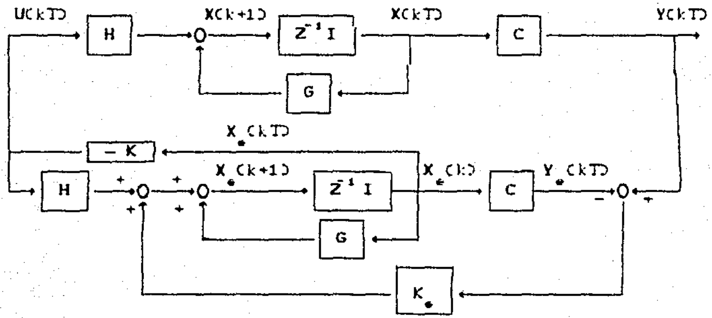


FIGURA 5 :SISTEMA DE CONTROL REALIMENTADO POR UN OBSERVADOR DE ESTADO.

En la figura 5 mostramos un observador de estado incorporado en el sistema. El cual el sistema es el siguiente, lineal, de tiempo discreto e invariante con el tiempo.

$$X[(k+1)T] = G \cdot X(kT) + H \cdot U(kT) \qquad (47)$$

$$Y(kT) = C \cdot X(kT) \qquad (48)$$

$$U(kT) = - K \cdot X(kT)$$

Donde

$X(kT)$: Vector de estado de dimensión 'n'

$U(kT)$: Vector de entrada o de control de dimensión 'r'

$Y(kT)$: Vector de salida de dimensión 'm'

G : Matriz de estado de orden [nxn]

H : Matriz de entrada de orden [nxr]

C : Matriz de salida de orden [mxn]

K : Matriz de Ganancia de Realimentación de estado de orden [rxn]

T : Período de muestreo.

Donde el sistema tiene que ser de observabilidad completa y el estado anterior $X(k)$ no esta disponible para medirlo directamente. Donde el observador de estado $X_e(kT)$ es usado para formar el vector de control $U(kT)$.

$$U(kT) = - K \cdot X_e(kT) \qquad (49)$$

De la figura 5 anterior tenemos que.

$$X_e[(k+1)T] = G \cdot X_e(kT) + H \cdot U(kT) + K_e \left[Y(kT) - Y_e(kT) \right] \qquad (50)$$

Donde K_e es la matriz de ganancia de realimentación del observador de dimensión (nxm), tenemos que.

$$X_e[(k+1)T] = G\cdot X_e(kT) + H\cdot U(kT) + K_e\left[Y(kT)-C\cdot X_e(kT)\right] \qquad (51)$$

$$X_e[(k+1)T] = G\cdot X_e(kT) - K_e C\cdot X_e(kT) + H\cdot U(kT) + K_e Y(kT)$$

$$X_e[(k+1)T] = (\ G-K_e C\ )X_e(kt) + H\cdot U(kT) + K_e\cdot Y(kT)$$

Como el error se define como,el cuál tendera a cero.

$$e(kT) = X(kT) - X_e(kT) \qquad (52)$$

Entonces

$$X[(k+1)T]-X_e[(k+1)T] = \qquad (53)$$

$$= G\cdot X(kT) + H\cdot U(kT) - (G-K_e\cdot C)X_e(kT) - H\cdot U(kT) - K_e\cdot Y(kT)$$

$$X[(k+1)T]-X_e[(k+1)T] = G\cdot X(kT) - G\cdot X_e(kT) + K_e\cdot C\cdot X_e(kT) - K_e\cdot Y(kT)$$

$$X[(k+1)T]-X_e[(k+1)T] = G\cdot X(kT) - G\cdot X_e(kT) + K_e\cdot C\cdot X_e(kT)-K_e\cdot C\cdot X(kT)$$

$$X[(k+1)T]-X_e[(k+1)T] = (G-K_e\cdot C)\left[X(kT)-X_e(kT)\right] \qquad (54)$$

$$e[(k+1)T] = (G-K_e\cdot C)\cdot e(kT) \qquad (55)$$

El comportamiento dinámico de la señal de error es determinada por los valores característicos de ($G - K_e\cdot C$).Ya que esta es una ecuación de estado de un sistema lineal.Si la matriz ($G - K_e C$) es una matriz estable el vector de error convergera a cero para para cualquier error inicial e(0).esto es $X_e(k)$ convergera a X(k) independientemente de los valores de X(0) y $X_e(0)$.Para obtener una respuesta rápida puede ser llevada a cabo si todos los valores caracteristicos de ($G - K_e C$) son elegidos para ser cero.

DISEÑO DEL OBSERVADOR DE SISTEMAS ESCALARES.

De nueva cuenta hay varios caminos para obtener la matriz de realimentación del observador K_e.Una forma de ahorrandonos muchas demostraciones es útilizando el principio de Dualidad de los sistemas,el cual lo veremos más adelante para sistemas multivariables .Nosotros escogimos el método de Ackermann's de los otros métodos para obtener la matriz de ganancia de realimentación para el observador K_e,en lo que sigue derivaremos el análisis o método para obtener K_e.

DISEÑO DEL OBSERVADOR DE ORDEN COMPLETO. PARA SISTEMAS ESCALARES.

Si tenemos el sistema de tiempo discreto,lineal e invariante con el tiempo.

$$X(k+1) = G\cdot X(k) + H\cdot U(k) \tag{56}$$

$$Y(k) = C\cdot X(k) \tag{57}$$

Donde

$X(k)$: Vector de estado de dimensión 'n'

$U(k)$: Vector de entrada o de control de dimensión 'r'

$Y(k)$: Vector de salida (escalar)

G : Matriz de estado no-singular de orden [n x n]

H : Matriz de salida de orden [n x r]

C : Matriz de salida de orden [1 x n]

T : Período de muestreo.

Se asume que es observable completamente,y el control es.

$$U(k) = -K\cdot X_e(k)$$

donde

K : Matriz de orden [r x n] .

Si la configuración es la misma que la de la figura 5 ,la dinámica del observador de estado esta dada por la ecuación,como se vio.

$$X_e(k+1) = (G-K_e\cdot C)X_e(k)+H\cdot U(k)+K_e\cdot C\cdot X(k)$$

Definiendo una transformación a un sistema equivalente demostrado anteriormente.

$$X(k) = Q\cdot E(k) \tag{61}$$

donde $E(k)$ es un nuevo vector de estado de dimensión 'n', donde Q es cualquier matriz de orden [n x n] no singular

Entonces

$$X(k+1) = G\cdot X(k) + H\cdot U(k) \tag{62}$$

$$Q^{-1}[Q\cdot E(k+1)] = Q^{-1}[G\cdot Q\cdot E(k)+H\cdot U(k)] \tag{63}$$

$$E(k+1) = (Q^{-1}G\cdot Q)E(k) + (Q^{-1}H)U(k) \tag{64}$$

y

$$Y(k) = C\cdot X(k) = (C\cdot Q)E(k) \tag{65}$$

y ademas definimos:

$$X_e(k) = Q\cdot E_e(k) \tag{66}$$

Sustituyendola en la ecuación del observador de estado, tenemos.

$$X_e(k+1) = G\cdot X_e(k)+H\cdot U(k) + K_e[Y(k)-Y_e(k)] \tag{67}$$

$$X_e(k+1) = (G-K_eC)X_e(k)+H\cdot U(k)+K_eC\cdot X(k)$$

$$Q\cdot E_e(k+1) = (G-K_eC)Q\cdot E_e(k)+H\cdot U(k)+K_eC\cdot Q\cdot E(k) \tag{68}$$

$$E_e(k+1) = [Q^{-1}(G-K_eC)Q]E_e(k)+(Q^{-1}H)U(k)+(Q^{-1}K_eCQ)E(k)$$

Restando las dos ecuaciones obtenidas anteriormente, tenemos.

$$E(k+1)-Ee(k+1) = (Q^{-1}GQ)E(k) + (Q^{-1}H)U(k) - Q^{-1}(G-KeC)QEe(k)-$$
$$-(Q^{-1}H)U(k) - (Q^{-1}KeCQ)E(k) \tag{69}$$

$$E(k+1)-E_e(k+1) = (Q^{-1}GQ)E(k) - (Q^{-1}GQ)E_e(k) + (Q^{-1}K_eCQ)E_e(k) -$$
$$- (Q^{-1}K_eCQ)E(k)$$

$$E(k+1)-E_e(k+1) = (Q^{-1}G\cdot Q-Q^{-1}K_eC\cdot Q)\cdot[E(k)-E_e(k)] \tag{70}$$

Definiendo el error como:

$$e(k) = E(k) - E_{\bullet}(k) \tag{71}$$

entonces se convierte.

$$e(k+1) = (Q^{-1}G \cdot Q - Q^{-1}K_{\bullet}C \cdot Q) \, e(k) =$$

$$= Q^{-1} (G - K_{\bullet}C) \, Q \cdot e(k) \tag{72}$$

La ecuación caracteristica para la dinamica del error esta dada por.

$$|ZI - Q^{-1}G \cdot Q + Q^{-1}K_{\bullet}C \cdot Q| = 0$$

donde definimos que

$$\hat{G} = Q^{-1}GQ \tag{73}$$

$$\hat{K}_{\bullet} = Q^{-1}K_{\bullet} \tag{74}$$

$$\hat{C} = CQ$$

Entonces

$$|ZI - \hat{G} + \hat{K}_{\bullet}\hat{C}| = 0 \tag{75}$$

la cual puede ser modificada a

$$\hat{G} - \hat{K}_{\bullet}\hat{C} = (\hat{G} - \hat{K}_{\bullet}\hat{C})^{T} =$$

$$= \hat{G}^{T} - (\hat{K}_{\bullet}\hat{C})^{T} =$$

$$= \hat{G}^{T} - \hat{C}^{T} \cdot \hat{K}_{\bullet}^{T}$$

Entonces

$$|ZI-\hat{G}+K_{e}\hat{C}| = |ZI-\hat{G}^{*}+\hat{C}^{*}\hat{K}_{e}^{*}| = 0 \qquad (76)$$

si definimos que

$$G_{o}^{*} = \hat{G}^{*}-\hat{C}^{*}\hat{K}_{e}^{*} \qquad (77)$$

Del teorema Cayley-Hamilton que el estado G_{o}^{*} debe satisfacer su propia ecuación caracteristica.

$$(G_{o}^{*})^{n}+\alpha_{1}(G_{o}^{*})^{n-1}+\ldots+\alpha_{n-1}G_{o}^{*}+\alpha_{n}I = \phi(G_{o}^{*}) = 0 \qquad (78)$$

considerando las identidades.

$I = I$

$$G_{o}^{*} = \hat{G}^{*}-\hat{C}^{*}\hat{K}_{e}^{*} \qquad (79)$$

$$(G_{o}^{*})^{2} = (\hat{G}^{*}-\hat{C}^{*}\hat{K}_{e}^{*})^{2} = (\hat{G}^{*})^{2}-\hat{G}^{*}\hat{C}^{*}\hat{K}^{*}-\hat{C}^{*}\hat{K}_{e}G_{o}^{*} \qquad (80)$$

$$(G_{o}^{*})^{3} = (\hat{G}^{*}-\hat{C}^{*}\hat{K}_{e}^{*})^{3} = (\hat{G}^{*})^{3}-(\hat{G}^{*})^{2}\hat{C}^{*}\hat{K}_{e}^{*}-\hat{G}^{*}\hat{C}^{*}\hat{K}_{e}^{*}G_{o}^{*}-\hat{C}^{*}\hat{K}_{e}^{*}(G_{o}^{*})^{2}$$

$$\vdots$$

$$(G_{o}^{*})^{n} = (\hat{G}^{*}-\hat{C}^{*}\hat{K}_{e}^{*})^{n} = (\hat{G}^{*})^{n}-(\hat{G}^{*})^{n-1}\hat{C}^{*}\hat{K}_{e}^{*}-\ldots-\hat{C}^{*}\hat{K}_{e}^{*}(G_{o}^{*})^{n-1}$$

Sustituyendo las relaciones en la ecuación caracteristica.

$$\alpha_n I + \alpha_{n-1} G_o^{*} + \alpha_{n-2} (G_o^{*})^2 + \ldots + (G_o^{*})^n$$

$$= \alpha_n I + \alpha_{n-1} \hat{G}^{*} + \alpha_{n-2} (\hat{G}^{*})^2 + \ldots + (\hat{G}^{*})^n - \alpha_{n-1} \hat{C}^{*} \hat{K}_{*}^{*} -$$

$$- \alpha_{n-2} \hat{G}^{*} \hat{C}^{*} \hat{K}_{*}^{*} - \alpha_{n-2} \hat{C}^{*} \hat{K}_{*}^{*} G_o^{*} - \ldots - (\hat{G}^{*})^{n-1} \hat{C}^{*} \hat{K}_{*}^{*} - \ldots -$$

$$- \hat{C}^{*} \hat{K}_{*}^{*} (G_o^{*})^{n-1} = 0 \qquad (B1)$$

la cuál puede escribirse como

$$0 = \phi(\hat{G}^{*}) - \left[\hat{C}^{*} \,\vdots\, \hat{G}^{*}\hat{C}^{*} \,\vdots\, \ldots \,\vdots\, (\hat{G}^{*})^{n-1}\hat{C}^{*} \right] \begin{bmatrix} \alpha_{n-1}\hat{K}_{*}^{*} + \alpha_{n-2}\hat{K}_{*}^{*}G_o + \ldots + \hat{K}_{*}^{*}(G_o)^{n-1} \\ \alpha_{n-2}\hat{K}_{*}^{*} + \alpha_{n-3}\hat{K}_{*}^{*}G_o + \ldots + \hat{K}_{*}^{*}(G_o)^{n-2} \\ \vdots \\ \hat{K}_{*}^{*} \end{bmatrix} \qquad (B2)$$

$$\phi(\hat{G}^{*}) = \left[\hat{C}^{*} \,\vdots\, \hat{G}^{*}\hat{C}^{*} \,\vdots\, \ldots \,\vdots\, (\hat{G}^{*})^{n-1}\hat{C}^{*} \right] \begin{bmatrix} \alpha_{n-1}\hat{K}_{*}^{*} + \alpha_{n-2}\hat{K}_{*}^{*}G_o + \ldots + \hat{K}_{*}^{*}(G_o)^{n-1} \\ \alpha_{n-2} \ldots \equiv \ldots \\ \hat{K}_{*}^{*} \end{bmatrix} \qquad (B3)$$

Notando que

$$\left[\hat{C}^{*} \,\vdots\, \hat{G}^{*}\hat{C}^{*} \,\vdots\, (\hat{G}^{*})^2\hat{C}^{*} \,\vdots\, \ldots \,\vdots\, (\hat{G}^{*})^{n-1}\hat{C}^{*} \right] = \qquad (B4)$$

$$= [(CQ)^{*} \,\vdots\, (Q^{-1}GQ)^{*}(CQ)^{*} \,\vdots\, [(Q^{-1}GQ)^{*}]^2 (CQ)^{*} \,\vdots\, \ldots \,\vdots\, [(Q^{-1}GQ)^{*}]^{n-1}(CQ)^{*}] =$$

$$= Q^{*}C^{*} \,\vdots\, (GQ)^{*}Q^{*^{-1}}Q^{*}C^{*} \,\vdots\, (GQ)^{*^{2}}[(Q^{*})^{-1}]^2 Q^{*}C^{*} \,\vdots\, \ldots \,\vdots\, [(GQ)^{*}]^{n-1}(Q^{*^{-1}})^{n-1} Q^{*}C^{*}$$

$$=[\,Q^*C^* \mid Q^*G^*C^* \mid (G^2Q^2)^*(Q^*)^{-2}Q^*C^* \mid \dots \mid (G^{n-1}Q^{n-1})^*(Q^*)^{-n+1}Q^*C^*\,] =$$

$$=[\,Q^*C^* \mid Q^*G^*C^* \mid (Q^2)^*(G^2)^*(Q^*)^{-1}C^* \mid \dots \mid (Q^*)^{n-1}(G^*)^{n-1}(Q^*)^{-n+2}C^*\,] =$$

$$=[\,Q^*C^* \mid Q^*G^*C^* \mid Q^*Q^*(G^*)^2(Q^*)^{-1}C^* \mid \dots \mid Q^*(Q^*)^{n-2}(G^*)^{n-1}(Q^*)^{-n+2}C^*\,] =$$

$$= [\,Q^*C^* \mid Q^*G^*C^* \mid Q^*(G^*)^2C^* \mid \dots \mid Q^*(G^*)^{n-1}C^*\,] = \qquad (85)$$

$$= Q^* \left[\, C^* \mid G^*C^* \mid (G^*)^2C^* \mid \dots \mid (G^*)^{n-1}C^* \,\right] \qquad (86)$$

De aqui vemos que la matriz del lado izquierda es la matriz de observabilidad completa del sistema, entonces el rango=n. Por lo tanto la matriz del lado izquierdo tiene inversa. Premúltiplicando ambos lados de la ecuación siguiente obtenida en (83)

$$\phi(\hat{G}^*) = \left[\, \hat{C}^* \mid \hat{G}^*\hat{C}^* \mid \dots \mid (\hat{G}^*)^{n-1}\hat{C}^* \,\right] \begin{bmatrix} \alpha_{n-1}\hat{K}_*^* + \alpha_{n-2}\hat{K}_*^*Go + \dots + \hat{K}_*^*(Go)^{n-1} \\ \alpha_{n-2}\hat{K}_*^* + \alpha_{n-3}\hat{K}_*^*Go + \dots + \hat{K}_*^*(Go)^{n-2} \\ \vdots \\ \hat{K}_*^* \end{bmatrix}$$

por la ecuación siguiente.

$$\begin{bmatrix} 0 & 0 & . & . & . & 0 & 1 \end{bmatrix} \left[\, \hat{C}^* \mid \hat{G}^*\hat{C}^* \mid \dots \mid (\hat{G}^*)^{n-1}\hat{C}^* \,\right]^{-1}$$

Obtendremos.

$$[0 \quad 0 \quad . \quad . \quad . \quad 0 \quad 1][\, \hat{C}^* \mid \hat{G}^*\hat{C}^* \mid \dots \mid (\hat{G}^*)^{n-1}\hat{C}^* \,]^{-1}\phi(\hat{G}^*) = \hat{K}_*^*$$

Sustituyendo la ecuación (86) en esta última ecuación, tenemos que

$$[0 \quad 0 \; . \; . \; . \; 0 \quad 1](Q^{*}[C^{*} \mid G^{*}C^{*} \mid (G^{*})^{2}C^{*} \mid \; . \; . \; . \mid (G^{*})^{n-1}C^{*}])^{-1} \phi(\hat{G}^{*}) = \hat{K}_{e}^{*}$$

$$[0 \quad 0 \; . \; . \; . \; 0 \quad 1][C^{*} \mid G^{*}C^{*} \mid \; . \; . \; . \mid (G^{*})^{n-1}C^{*}]^{-1} (Q^{*})^{-1} \phi(\hat{G}^{*}) = \hat{K}_{e}^{*}$$

refiriendonos a la relación demostrada anteriormente.

$$Q \cdot \phi(\hat{G}^{*}) \cdot Q^{-1} = \phi(G^{*}) \tag{87}$$

$$(Q^{*})^{-1} \phi(\hat{G}^{*}) \cdot Q^{*} = \phi(G^{*}) \tag{88}$$

$$[0 \quad 0 \; . \; . \; . \; 0 \quad 1][C^{*} \mid G^{*}C^{*} \mid \; . \; . \; . \mid (G^{*})^{n-1}C^{*}]^{-1} \phi(G^{*})(Q^{*})^{-1} = \hat{K}_{e}^{*}$$

$$[0 \quad 0 \; \ldots \; 0 \; 1][C^{*} \mid G^{*}C^{*} \mid \ldots \mid (G^{*})^{n-1}C^{*}]^{-1} \phi(G^{*})(Q^{*})^{-1}(Q^{*}) = K_{e}^{*} Q^{*^{-1}} Q^{*}$$

$$\Big[[0 \quad 0 \; . \; . \; . \; 0 \quad 1][C^{*} \mid G^{*}C^{*} \mid \ldots \mid (G^{*})^{n-1}C^{*}]^{-1} \phi(G^{*})\Big]^{*} = \Big[K_{e}^{*}\Big]^{*}$$

$$K_{e} = [\phi(G^{*})]^{*}([C^{*} \mid G^{*}C^{*} \mid \ldots \mid (G^{*})^{n-1}C^{*}]^{-1})^{*} ([0 \quad 0 \; . \; . \; . \; 0 \quad 1])^{*} \tag{89}$$

$$K_{e} = \phi(G) \cdot \begin{bmatrix} (C^{*})^{*} \\ (G^{*}C^{*})^{*} \\ \vdots \\ [(G^{*})^{n-1}C^{*}]^{*} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{90}$$

$$K_e = \phi(G) \cdot \begin{bmatrix} C \\ C \cdot G \\ \vdots \\ C \cdot G^{n-2} \\ C \cdot G^{n-1} \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{91}$$

La ecuación anterior es la solución final. De la matriz de ganancia de realimentación del observador K_e.

DISEÑO DEL OBSERVADOR DE ORDEN COMPLETO.

SISTEMAS MULTIVARIABLES

Obtendremos el método para encontrar la matriz de ganancia de realimentación para el observador K_e, tanto para sistemas escalares como para sistemas multivariables.

Si tenemos el sistema de multivariable , donde el sistema es

de observabilidad completa. En tiempo discreto, lineal e invariante con el tiempo

$$X(k+1) = G \cdot X(k) + H \cdot U(k) \tag{92}$$

$$Y(k) = C \cdot X(k) \tag{93}$$

donde

$X(k)$: Vector de Estado de dimensión 'n'

$U(k)$: Vector de entrada o de control de dimensión 'r'

$Y(k)$: Vector de Salida de dimensión 'm'

G : Matriz de estado de orden $[n \times n]$

H : Matriz de entrada de orden $[n \times r]$

C : Matriz de salida de orden $[m \times n]$.

T : Período de muestreo.

La ecuación del observador de estado esta dado por:

$$X_{\bullet}(k+1) = G \cdot X_{\bullet}(k) + H \cdot U(k) + K_{\bullet}\left[Y(k) - Y_{\bullet}(k)\right]$$

$$X_{\bullet}(k+1) = (G-K_{\bullet}C)X_{\bullet}(k) + H \cdot U(k) + K_{\bullet} \cdot Y(k) \qquad (94)$$

y la dinámica del error es

$$e(k+1) = (G-K_{\bullet}C)\ e(k). \qquad (95)$$

donde asumimos que es de observabilidad completa. Para el diseño del observador en la colocación arbitraria de los valores de característicos $(G-K_{\bullet}C)$ esto es posible, si es de observabilidad completa. Para obtener la matriz de ganancia de realimentación del del observador $K_{\bullet}$ se nota que los valores característicos de

$$(G-K_{\bullet}C) \quad \text{y los de} \quad (G-K_{\bullet}C)^{T} = G^{T}-C^{T}K_{\bullet}^{\ T}$$

son los mismos, utilizando el principio de dualidad, la condición de observabilidad completa para el sistema original es el mismo que la condición de controlabilidad de estado completa para el sistema mostrado como.

$$X(k+1) = G^T \cdot X(k) + C^T \cdot U(k)$$

es controlable de estado completamente si la matriz siguiente es de Rango

$$\left[C^T \quad G^T C^T \quad . \; . \; . \quad (G^T)^{P-1} C^T\right]$$

es igual a n.

la anterior ecuación se puede ver que es la condición de observabilidad completa de el sistema definido por las ecuaciones originales.

$$X(k+1) = G \cdot X(k) + H \cdot U(k) \qquad (96)$$

$$Y(k) = C \cdot X(k) \qquad (97)$$

El sistema es observable completamente si la matriz siguiente es de Rango

$$\left[C^T \quad G^T C^T \quad . \; . \; . \quad (G^T)^{P-1} C^T\right] = n$$

Entonces podemos ver que es facil considerar el sistema definido por, usando el principio de dualidad.

$$Z(k+1) = G^{*} \cdot Z(k) + C^{*} \cdot V(k) \qquad (98)$$

$$S(K) = H^{*} \cdot Z(K) \qquad (99)$$

La condición de controlabilidad de estado completa es:

$$\text{Rango}\left[C^T \quad G^T C^T \quad . \; . \; . \quad (G^T)^{P-1} C^T\right] = n.$$

la cuál es la misma que la condición de observabilidad completa del sistema original.

De aqui vemos que es más facil considerar este sistema último en lugar del sistema original para obtener la matriz K_e.Esto es porque la matriz

$$(G^T-C^TK)$$

y la matriz

$$(G-K_eC)$$

tienen los mismos valores propios,lo cual lo demostraremos.

Esto es porque para sistemas físicos todos los valores característicos complejos de la matriz $(G-K_eC)$ ocurren en pares complejos conjugados.Los valores característicos de $(G-K_eC)$ son los mismos que los de

$$(G-K_eC)^* = (G^*-(K_eC)^*) = (G^*-C^*K_e^*). \qquad (100)$$

Esto es fácilmente demostrable advirtiendo que si λ es un valor propio de (G^*-C^*K) ,su complejo conjugado λ^* tambien lo es. Entonces λ^* cumple la ecuación caracteristica $\det(\lambda^*I-G^*+C^*K)=0$ tomando el complejo conjugado se tiene que $\det(\lambda I-G+K^*C)=0$ y sabemos que el determinante es invariante por transposición.

De aqui vemos que la ecuación caracteristica para el diseño es.

$$|ZI-G+K_eC| = Z^n+\alpha_1Z^{n-1}+\dots+\alpha_{n-1}Z+\alpha_n = 0 \qquad (101)$$

y la ecuación anterior es equivalente a su sistema dual.

$$|ZI-G^{*}+C^{*}K_{e}^{*}| = Z^{n}+\alpha_{1}Z^{n-1}+\ldots+\alpha_{n-1}Z+\alpha_{n} = 0$$

Entonces la matriz de ganancia de realimentación para el observador K_e puede ser determinada útilizando su sistema dual de su sistema original y útilizando cualquier método para la obtención de la matriz de ganancia de realimentación de estado K. De aqui podemos ver que igual que en la realimentación de estado la K_e no es única la matriz. Entonces los valores caracteristicos de ($G-K_eC$) son los mismos como los elegidos de ($G^{*}-C^{*}K$). y estos estan relacionados a la matriz K por la ecuación de

$$K_{e} = K^{*}$$

como las matrices que se manejan son reales, entonces la relación es

$$K_{e} = K^{T}$$

LOS EFECTOS DE LA ADICION DE EL OBSERVADOR AL SISTEMA DE CONTROL.

Considerando el sistema de Controlabilidad de Estado Completa y Observabilidad completa dado por las ecuaciones.

Sistema de tiempo discreto, lineal e invariante con el tiempo.

$$X(k+1) = G\cdot X(k) + H\cdot U(k) \qquad (102)$$

$$Y(k) = C\cdot X(k) \qquad (103)$$

Donde

X(k) : Vector de estado de dimensión 'n'

U(k) : Vector de entrada o control de dimensión 'r'

Y(k) : Vector de salida de dimensión 'm'

G : Matriz de estado de orden [n×n]

H : Matriz de entrada de orden [n×r]

C : Matriz de salida de orden [m×n]

T : Período de muestreo

El control usado para modificar el funcionamiento de la planta con un observador es.

$$U(k) = -K\cdot X_{\bullet}(k) - r(k) \qquad (104)$$

de la relación definida

$$e(k) = X(k)-X_{\bullet}(k) \qquad (105)$$

$$U(k) = -K\cdot(X(k)-e(k)) - r(k) \qquad (106)$$

y la ecuación

$$X(k+1) = G\cdot X(k) + H\cdot U(k)$$

$$X(k+1) = G\cdot X(k) + H\cdot\left[-K(X(k)-e(k)) - r(k)\right]$$

$$X(k+1) = (G-H\cdot K) X(k) + H\cdot K\cdot e(k) - H\cdot r(k)$$

Del observador se tienen las ecuaciones

$$X_{\bullet}(k+1) = G\cdot X_{\bullet}(k) + H\cdot U(k) +K_{\bullet}(Y(k)-Y_{\bullet}(k))$$

$$Y_{\bullet}(k) = C\cdot X_{\bullet}(k) \qquad (107)$$

y la ecuación del error observado esta dada como.

$$e(k+1) = (G-K_{\bullet}\cdot C)e(k) \qquad (108)$$

Donde sabemos que el sistema de lazo cerrado consiste de la interconexión de la planta y del observador y el controlador de la ecuación

$$U(k) = -K\cdot X_{\bullet}(k)-r(k) = -K(X(k)-e(k))-r(k) \qquad (109)$$

Y este sistema es representado por las ecuaciones.

$$Xe(k+1) = (G-H\cdot K)X(k) + H\cdot K\cdot e(k) -H\cdot r(k)$$

$$e(k+1) = (G-K_{\bullet}C)e(k) \qquad (110)$$

La cual puede ser representada en forma matricial como.

$$\begin{bmatrix} X(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} G-H\cdot K & H\cdot K \\ 0 & G-K_e C \end{bmatrix} \begin{bmatrix} X(k) \\ e(k) \end{bmatrix} - \begin{bmatrix} H \\ 0 \end{bmatrix} \begin{bmatrix} r(k) \end{bmatrix}$$

Esta ecuación describe la dinámica del sistema de control realimentado por un observador de estado. Donde la ecuación caracteristica para el sistema es.

$$| ZI-(G-H\cdot K)|\cdot| ZI-(G-K_e C) | = 0$$

Donde los valores caracteristicos de

$$(G-H\cdot K)$$

y los de

$$(G-K_e C)$$

pueden ser asignados independientemente por la selección apropiada de las matrices K y K_e . Está es una importante propiedad que permite al observador y el control ser diseñados separadamente y combinarlos justamente hacia la forma del sistema de control realimentado por el observador de estado.

OBSERVADORES DE ORDEN REDUCIDO.

Los observadores mostrados anteriormente son diseñados y reconstruidos con todas las variables de estado. Como en la práctica solo algunas de las variables de estado pueden ser medidas exactamente,por consiguiente estas mediciones de las variables de estado no necesitan ser estimadas.

Un observador que estima menos de n variables de estado, donde n es la dimensión de el vector de estado ,esté es llamado un observador de orden reducido. Si el orden del observador de orden reducido es el mínimo posible,el observador de orden- reducido es llamado observador de orden-mínimo.

Teniendo un sistema en variables de estado el vector de estado $X(k)$ es un vector de dimensión n y el vector de salida $Y(k)$ es un vector de dimensión m las cuales pueden ser medidas. Entonces las m variables de salida son combinaciones lineales de las variables de estado. De este modo podemos ver que las m variables de estado no necesariamente deben ser estimadas.

De está manera podemos decir que necesitamos solo estimar en el sistema llamado observador ó estimador $n-m$ variables de estado. De esta manera el sistema llamado observador de orden reducido se conbierte en un observador de orden $(n-m)$,y como el orden $(n-m)$ que se puede tener es el mínimo entonces el observador o estimador es de orden-mínimo.

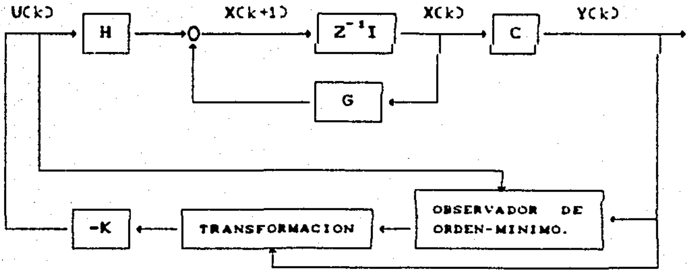


FIGURA 6 : SISTEMA DE CONTROL REALIMENTADO CON OBSERVADOR DE ESTADO,CON UN OBSERVADOR DE ORDEN-MINIMO.

DISEÑO DEL OBSERVADOR DE ESTADO DE ORDEN O DIMENSION MINIMA.

Si consideramos la siguiente ecuación de dimensión-n. De tiempo discreto,lineal e invariante con el tiempo

$$X(k+1) = G \cdot X(k) + H \cdot U(k) \tag{111}$$

$$Y(k) = C \cdot X(k) \tag{112}$$

Donde:

$X(k)$: Vector de estado de dimensión 'n'

$U(k)$: Vector de entrada o de control de dimensión 'r'

$Y(k)$: Vector de salida de dimensión 'm'

G : Matriz de estado de orden [n x n]

H : Matriz de entrada de orden [n x r]

C : Matriz de salida de orden [m x n]

Asumimos que la matriz C tiene rango-completo, esto es rango(C)= m.

Difinimos

$$P \stackrel{\Delta}{=} \begin{bmatrix} C \\ R \end{bmatrix} \tag{113}$$

donde R es una matriz [(n-m)xn] y es enteramente albitraria, mientras P sea no-singular. Obteniendo la inversa de P como.

$$Q \stackrel{\Delta}{=} P^{-1} \stackrel{\Delta}{=} [\; Q_1 \;|\; Q_2 \;] \tag{114}$$

donde Q_1 y Q_2 son matrices de orden [nxm] y [nx(n-m)].

Claramente tenemos que

$$I_n = P \cdot Q = P \cdot P^{-1} = \tag{115}$$

$$= \begin{bmatrix} C \\ R \end{bmatrix} [\, Q_1 \quad Q_2 \,] = \begin{bmatrix} C \cdot Q_1 & C \cdot Q_2 \\ R \cdot Q_1 & R \cdot Q_2 \end{bmatrix} = \begin{bmatrix} I_q & 0 \\ 0 & I_{n-q} \end{bmatrix}$$

transformando la ecuación dinámica que representa al sistema, por la transformación equivalente.

$$\hat{X}(k) = P \cdot X(k) \tag{116}$$

$$X(k) = P^{-1} \cdot \hat{X}(k) \tag{117}$$

$$P^{-1} \cdot \hat{X}(K+1) = G \cdot P^{-1} \cdot \hat{X}(K) + H \cdot U(k) \tag{118}$$

$$Y(k) = C \cdot P^{-1} \cdot \hat{X}(K) = \tag{119}$$

$$= C \cdot \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \cdot \hat{X}(k) = \tag{120}$$

$$= \begin{bmatrix} C \cdot Q_1 & C \cdot Q_2 \end{bmatrix} \cdot \hat{X}(k)$$

$$Y(k) = \begin{bmatrix} I_q & 0 \end{bmatrix} \cdot \hat{X}(k) \tag{121}$$

$$\hat{X}(k+1) = P \cdot G \cdot P^{-1} \cdot \hat{X}(k) + P \cdot B \cdot U(k)$$

$$Y(k) = C \cdot P^{-1} \cdot \hat{X}(k) = \tag{122}$$

$$= C \cdot Q \cdot \hat{X}(k) =$$

$$= [\ I \quad 0\ ] \cdot \hat{X}(k) \tag{123}$$

donde obtenemos

$$\begin{bmatrix} \hat{X}_1(k+1) \\ \hat{X}_2(k+1) \end{bmatrix} = \begin{bmatrix} \hat{G}_{11} & \hat{G}_{12} \\ \hat{G}_{21} & \hat{G}_{22} \end{bmatrix} \begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \end{bmatrix} + \begin{bmatrix} \hat{H}_1 \\ \hat{H}_2 \end{bmatrix} \cdot U(k) \qquad (124)$$

$$Y(k) = [\; I_m \;\vdots\; 0 \;] \begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \end{bmatrix} = \hat{X}_1(k) \qquad (125)$$

donde $\hat{X}_1(k)$ consiste de los primeros m elementos de $\hat{X}(k)$ y $\hat{X}_2(k)$ es el resto de $\hat{X}(k)$.

$\hat{G}_{11}$: Matriz de orden $[m\times m]$

$\hat{G}_{12}$: Matriz de orden $[m\times(n-m)]$

$\hat{G}_{21}$: Matriz de orden $[(n-m)\times m]$

$\hat{G}_{22}$: Matriz de orden $[(n-m)\times(n-m)]$

$\hat{H}_1$: Matriz de orden $[m\times r]$

$\hat{H}_2$: Matriz de orden $[(n-m)\times r]$.

Podemos ver de

$$Y(k) = [\ I_m \quad 0\ ]\ \hat{X}(k) = \hat{X}_1(k)$$

que

$$Y(k) = \hat{X}_1(k) \tag{126}$$

Aqui sólo el último elemento n-m de $\hat{X}(k)$ necesita ser estimado. De aqui podemos concluir que consecuentemente necesitamos solo un observador de estado de dimensión (n-m) más que un observador de dimensión (n), y $X_1(k)$ es la porción de el vector de estado que puede ser directamente medida. $X_1(k)$ es un vector de dimensión-m.

Entonces la ecuación matricial anterior la podemos escribir como.

$$\hat{X}_1(k+1) = \hat{G}_{11} \cdot \hat{X}_1(k) + \hat{G}_{12} \cdot \hat{X}_2(k) + \hat{H}_1 \cdot U(k)$$

$$\hat{X}_1(k+1) - \hat{G}_{11} \hat{X}_1(k) - \hat{H}_1 \cdot U(k) = \hat{G}_{12} \cdot \hat{X}_2(k) \tag{127}$$

esta ecuación actua como ecuación de salida, y describe la dinámica de la porción medible de estado. Y además la ecuación donde describe la dinámica de la porción no-medibles de estado es.

$$X_2(k+1) = \hat{G}_{22} \cdot \hat{X}_2(k) + [\hat{G}_{21} \cdot \hat{X}_1(k) + \hat{H}_2 \cdot U(k)] \tag{128}$$

de aqui podemos ver que se el par (G,C) en la ecuación original o el equivalente el par $(\hat{G},\hat{C})$ en la ecuación transformada es observable, entonces el par $(\hat{G}_{22}, \hat{G}_{12})$ es observable, consecuentemente existe un observador de estado de dimensión (n-m) de $\hat{X}_2(k)$. La ecuación para el observador de orden-completo es

$$X(k+1) = G \cdot X(k) + H \cdot U(k) \tag{129}$$

y la ecuación de estado para el observador de orden-mínimo es

$$\hat{X}_2(k+1) = \hat{G}_{22} \cdot \hat{X}_2(k) + [\hat{G}_{21} \cdot \hat{X}_1(k) + \hat{H}_2 \cdot U(k)] \tag{130}$$

y la ecuación de salida para el observador de orden-completo es

$$Y(k) = C \cdot X(k) \tag{131}$$

y la ecuación de salida para el observador de orden-mínimo es

$$\hat{X}_1(k+1) - \hat{G}_{11} \cdot \hat{X}_1(k) - \hat{H}_1 \cdot U(k) = \hat{G}_{12} \cdot \hat{X}_2(k) \tag{132}$$

El diseño de el observador de orden-mínimo puede ser construido por sustituciones dadas en las ecuaciones anteriores y en la ecuación del observador de orden-completo, la cuál es.

$$X_{\bullet}(k+1) = (G-K_{\bullet} \cdot C)X_{\bullet}(k) + H \cdot U(k) + K_{\bullet} \cdot Y(k) \tag{133}$$

y la lista de substituciones son:

Observador de estado de orden-completo.	Observador de estado de orden-mínimo
$X_{\bullet}(k)$	$\hat{X}_{2\bullet}(k)$
G	$\hat{G}_{22}$
C	$\hat{G}_{12}$
$H\cdot U(k)$	$[\hat{G}_{21}\hat{X}_{1}(k) + \hat{H}_{2}U(k)]$
$K_{\bullet}[n{\times}m]$	$K_{\bullet}[(n-m){\times}m]$
$Y(k)$	$[\hat{X}_{1}(k+1) - \hat{G}_{11}\hat{X}_{1}(k) - \hat{H}_{1}U(k)]$;

Tomando las sustituciones de la tabla de relación, obtenemos.

$$\hat{X}_{2\bullet}(k+1) = (\hat{G}_{22} - \hat{K}_{\bullet}\hat{G}_{12})\hat{X}_{2\bullet}(k)+[\hat{G}_{21}\cdot\hat{X}_{1}(k)+\hat{H}_{2}\cdot U(k)] +$$

$$+ \hat{K}_{\bullet}[\hat{X}_{1}(k+1)-\hat{G}_{11}\cdot\hat{X}_{1}(k) - \hat{H}_{1}\cdot U(k)] \qquad (134)$$

donde la matriz de realimentación $\hat{K}_{\bullet}$ es una matriz$[(n-m){\times}m]$, y la anterior ecuación define el observador-mínimo. Y $X_{1}(k)$ es ala porción de el vector de estado que puede ser directamente medido, $X_{1}(k)$ es un vector de dimensión -m.

Usando

$$Y(k) = \hat{X}_{1}(k) \qquad (135)$$

y sustituyendola

$$\hat{X}_{2e}(k+1) = (\hat{G}_{22}-\hat{K}_e\cdot\hat{G}_{12})\hat{X}_{2e}(k) + \hat{G}_{21}\cdot Y(k) + \hat{H}_2\cdot U(k) +$$

$$+ \hat{K}_e\cdot Y(k+1) - \hat{K}_e\cdot\hat{G}_{11}\cdot Y(k) - \hat{K}_e\cdot\hat{H}_1\cdot U(k) \qquad (136)$$

$$\hat{X}_{2e}(k+1) = (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})\hat{X}_{2e}(k) + \hat{K}_e\cdot Y(k+1) +$$

$$+(\hat{G}_{21} - \hat{K}_e\cdot\hat{G}_{11})Y(k) + (\hat{H}_2 - \hat{K}_e\cdot\hat{H}_1)U(k) \qquad (137)$$

$$\hat{X}_{2e}(k+1) - \hat{K}_e\cdot Y(k+1) = (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})\hat{X}_{2e}(k) +$$

$$+ (\hat{G}_{21} - \hat{K}_e\cdot\hat{G}_{11})Y(k) + (\hat{H}_2 - \hat{K}_e\cdot\hat{H}_1)U(k) \qquad (138)$$

$$\hat{X}_{2e}(k+1) - \hat{K}_e\cdot Y(k+1) = (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})[\hat{X}_{2e}(k) - \hat{K}_e\cdot Y(k)] +$$

$$+ (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})\hat{K}_e\cdot Y(k) + (\hat{G}_{21} - \hat{K}_e\cdot\hat{G}_{11})Y(k) + (\hat{H}_2 - \hat{K}_e\cdot\hat{H}_1)U(k)$$

$$\hat{X}_{2e}(k+1) - \hat{K}_e\cdot Y(k+1) = (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})[\hat{X}_{2e}(k) - \hat{K}_e\cdot Y(k)] +$$

$$+ [(\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})\hat{K}_e + \hat{G}_{21} - \hat{K}_e\cdot\hat{G}_{11})Y(k) + (\hat{H}_2 - \hat{K}_e\cdot\hat{H}_1)U(k)$$

Si definimos

$$X_2(K) - K_e\cdot Y(k) = X_2(k) - K_e\cdot X_1(k) = N(k) \qquad (139)$$

y

$$X_{2e}(k) - K_e\cdot Y(k) = X_{2e}(k) - K_e\cdot X_1(k) = N_e(k) \qquad (140)$$

donde obtenemos, sustituyendo.

$$N_e(k+1) = (\hat{G}_{22} - \hat{K}_e \cdot \hat{G}_{12})N_e(k) + [(\hat{G}_{22} - \hat{K}_e \cdot \hat{G}_{12})\hat{K}_e + \hat{G}_{21} - \hat{K}_e \cdot \hat{G}_{11}]Y(k) +$$

$$+ \ (\hat{H}_2 - \hat{K}_e \cdot \hat{H}_1)U(k) \qquad (141)$$

La ecuación $N_e(k)$ y esta última ecuación define la dinámica de el observador de orden-mínimo, y la ecuación de error del observador, se define como.

$$e(k) = N(k) - N_e(k) = \hat{X}_2(k) - \hat{X}_{2e}(k) \qquad (142)$$

Restando la ecuación (134) $\hat{X}_{2e}(k+1)$ de la

ecuación (128) $\hat{X}_2(k+1)$, obtenemos

$$\hat{X}_2(k+1) - \hat{X}_{2e}(k+1) = \hat{G}_{22} \cdot \hat{X}_2(k) - \hat{G}_{22} \cdot \hat{X}_{2e}(k) + \hat{K}_e \cdot \hat{G}_{12} \cdot \hat{X}_{2e}(k) -$$

$$- \hat{K}_e \cdot \hat{X}_1(k+1) + \hat{K}_e \cdot \hat{G}_{11} \cdot \hat{X}_1(k) + \hat{K}_e \cdot \hat{H}_1 \cdot U(k)$$

$$\hat{X}_2(k+1) - \hat{X}_{2e}(k+1) = \hat{G}_{22}[\hat{X}_2(k) - \hat{X}_{2e}(k)] + \hat{K}_e \cdot \hat{G}_{12} \cdot \hat{X}_{2e}(k) -$$

$$- \hat{K}_e[\hat{X}_1(k+1) - \hat{G}_{11} \cdot \hat{X}_1(k) - \hat{H}_1 \cdot U(k)] \qquad (143)$$

sustituyendo la ecuación

$$\hat{X}_1(k+1) - \hat{G}_{11}\cdot\hat{X}_1(k) - \hat{H}_1\cdot U(k) = \hat{G}_{12}\cdot\hat{X}_2(k) \qquad (144)$$

$$\hat{X}_2(k+1)-\hat{X}_{2e}(k+1) = \hat{G}_{22}\left[\hat{X}_2(k)-\hat{X}_{2e}(k)\right]+\hat{K}_e\cdot\hat{G}_{12}\cdot\hat{X}_{2e}(k)-\hat{K}_e\cdot\hat{G}_{12}\cdot\hat{X}_2(k)$$

$$\hat{X}_2(k+1) - \hat{X}_{2e}(k+1) = (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})\left[\hat{X}_2(k) - \hat{X}_{2e}(k)\right]$$

La cuál puede ser escrita como sigue.

$$e(k+1) = (\hat{G}_{22} - \hat{K}_e\cdot\hat{G}_{12})\, e(k) \qquad (145)$$

donde está es la ecuación de error del observador, $e(k)$ vector$[n-m]$ donde la ecuación característica para el observador de orden-mínimo es

$$\left| ZI - \hat{G}_{22} + \hat{K}_e\cdot\hat{G}_{12} \right| = 0 \qquad (146)$$

Donde puede ser determinada la matriz de ganancia de realimentación del observador $\hat{K}_e$ $[(n-m) \times m]$ (MATRIZ), pues los eigenvalores de $(\hat{G}_{22}-\hat{K}_e\cdot\hat{G}_{12})$ pueden ser asignados arbitrariamente con las técnicas vistas anteriormente para observadores.

Ahora combinamos.

$$\hat{X}_1(k) = Y(k) = \hat{X}_{1e} \qquad (147)$$

con

$$\hat{X}_{2e}(k) - \hat{K}_e\cdot Y(k) = N_e(k) \qquad (148)$$

$$\hat{X}_{2e}(k) = N_e(k) + \hat{K}_e \cdot Y(k) \tag{149}$$

entonces

$$X_e(k) = \begin{bmatrix} \hat{X}_{1e}(k) \\ \hat{X}_{2e}(k) \end{bmatrix} = \begin{bmatrix} Y(k) \\ \hat{K}_e \cdot Y(k) + N_e(k) \end{bmatrix} \tag{150}$$

entonces de

$$\hat{X}(k) = P \cdot X(k)$$

tenemos que

$$X(k) = P^{-1}\hat{X}(k) = Q \cdot \hat{X}(k) \qquad \text{o}$$

$$X_e(k) = Q \cdot X_e(k) = [Q_1 \quad Q_2] \begin{bmatrix} Y(K) \\ \hat{K}_e \cdot Y(k) + N_e(k) \end{bmatrix} \tag{151}$$

$$X_e(k) = [Q_1 \quad Q_2] \begin{bmatrix} I_m & 0 \\ \hat{K}_e & I_{n-m} \end{bmatrix} \begin{bmatrix} Y(k) \\ N_e(k) \end{bmatrix} \tag{152}$$

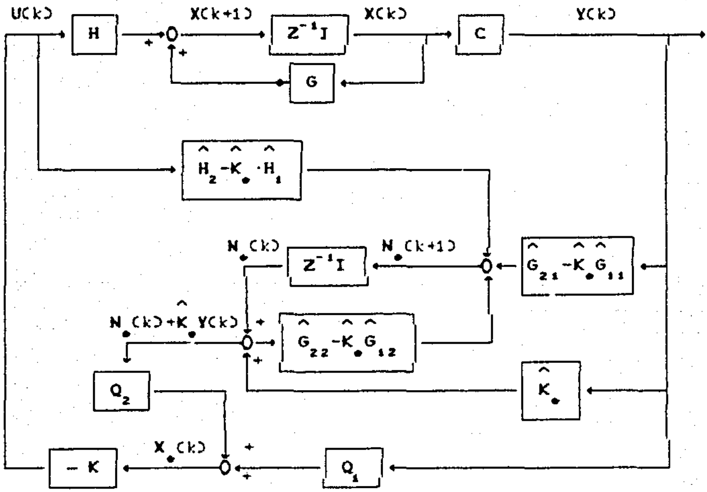


FIGURA 7 : CONTROL POR REALIMENTACION DE ESTADO,DONDE EL ESTADO REALIMENTADO CONSISTE DE LAS MEDICIONES DE UNA PORCION DEL ESTADO Y OTRA PORCION POR OBSERVACION DE EL ESTADO OBTENIDO POR UN OBSERVADOR DE ORDEN-MINIMO(n-m).

PROGRAMACION DE ALGORITMOS.

PROGRAMA : ESPACIO DE ESTADOS.

LENGUAJE : TURBO PASCAL VERSION 3.0

EQUIPO : P C . X T . A T.

ECES.PAS

LLAMAR A PROGRAMA : ECES.COM

PROGRAMA HECHO POR : ARTEMIO ROBERTO FLORES RAMOS.

ORGANIZACION FUNCIONAL.

Descripción del módulo ECES.

En este módulo principal se obtienen los datos de organización del sistema a traves del procedure LECTURA el cuál leera los datos del sistema en la forma de ecuación de estados.para obtener las gráficas del las diferentes respuestas es necesario que se den los datos de las gráficas asi como las entradas del sistema llamando para esto a METERDATOS en donde si es la primera vez entrada a MENEIGEN.para la determinación del tipo de diagnóstico o de salida que se quiera obtener del sistema en cuestión o que se produsca un nuevo diseño(dependiendo del diagnóstico) se podra entrar o transferir a los distintos módulos del programa como son : LISRESLIB, LOSARREG, NGRAFICAS, CONTROESTA, CONTROSAL, OBSERVAB, REALIMEN, OBSERVADOR, OBSERMINIMO, BAIRSTOW. Los cuales se expricaran a continuación cada uno de ellos. De esté modo bajo este proceso se podran ejecutar diferentes organizaciones de diferentes sistemas en la fórma de espacios de estados, en tiempo continuo como de tiempo discreto.

Constantes del módulo ECES.

NN: Valor constante que representa la dimención máxima que tendra el espacio de estados.

NNNN: Representa la dimención máxima de algunas matrices.

PMAX: Representa la dimensión de los arreglos de los puntos de las respuestas del sistema que se podran almacenar en el arreglo.

Variables del módulo ECES.

A, G: Representa la matriz de estado A o G del sistema, respectivamente.

B, H: Representa la matriz de entrada B ó H del sistema, respectivamente.

C: Representa la matriz C de salida del sistema.

D: Representa la matriz D de transmición directa del sistema.

AEXP: Representa el valor de la matriz e^{AT}.

CONEST: Representa la matriz de controlabilidad de estado.

KREALIMULT: Representa la matriz de ganacia de realimentación de estado.

KESTIMMULT: Representa la matriz de ganancia de realimentación para el observador K_e para sistemas de múltiples salidas.

N: Representa el número de variables de estado del sistema.

P: Representa el número de puntos para cálcular la respuesta del sistema.

NUM: Representa el número de terminos de la serie e^{AT} para ser cálculados.

R: Representa el número de entradas del sistema.

M: Representa el número de salidas del sistema.

CASO: Representa variable de control.

T: Representa variable del tiempo del sistema.

TI: Representa el tiempo inicial del sistema.

INC: Representa el incremento del tiempo,es decir el intervalo de un punto a otro punto en las respuestas del sistema.

DET: Representa el valor del determinante de alguna matriz.

SUMA: Representa una variable temporal.

EIGMIN: Representa el valor del eigenvalor mínimo del sistema.

X: Representa una matriz para almacenar las respuestas libres del sistema.

AREA: Representa matriz en donde se almacenan las respuestas totales del sistema.

PY: Representa vector de almacenamiento temporal para mandar a gráficar las respuestas de cada una de las variables del sistema.

TIPOSIS: Representa el caracter del tipo de sistema, de tiempo continuo o de tiempo discreto.

PAUSA: Representa variable de control para pausas de pantalla.

SI: Representa variable de contestación de estrada.

CADENN: Representa una cadena de caracteres en donde se guardan las funciones de las entradas del sistema.

XI: Representa el vector de las condiciones iniciales X(to).

RAICESR: Representa un vector donde se guarda la parte real de las raíces del sistema.

RAICESI: Representa un vector donde se guarda la parte imaginaria de las raíces del sistema.

VERD: Representa variable booleana de control para constestaciones.

BAN: Representa variable de control tipo bandera.

PRIERRENT, PRIMPAS: Representan variables de control.

Descripción del módulo IMPRIME.

En este módulo su objetivo es escribir matrices.

I, J: Representan variables de iteraciones.

Descripción del módulo GRAF.

En este módulo se gráfican las respuestas del sistema en una resulución de pantalla de trescientos puntos.

Variables del módulo GRAF.

DATOS: Representa el vector donde se tienen los puntos de respuesta

NUM: Representa el punto donde se quiere que se empiese a gráficar.

XMAX: Representa el número de puntos de la respuesta.

X: Representa el valor que toma el punto de la absiza para ser gráficado el punto en la resolución de la pantalla.

Y: Representa el valor que toma la ordenada del punto a gráficarse.

T: Representa contador de iteraciónes.

INC: Representa el incremento de un punto a otro punto para la resulución de la pantalla.

XXX: Representa el número máximo de los puntos a gráficarse.

TX: Representa las veces de contador para los diferentes puntos a ser gráficados en la pantalla.

RESS: Representa al número máximo de puntos que se tienen para cálcular el incremento.

TA: Representa indice del array de datos ,para obtener los datos salteados dependiente del número de puntos.

A: Representa la ordenada del punto a gráficarse sobre pantalla.

RANGO: Representa el rango que se tiene del punto máximo al punto mínimo para poder normalizar los puntos a gráficarse.

NORM: Representa la norma,para normalizar las ordenadas de los puntos a gráficarse sobre la resolución de 300 de pantalla.

Z: Representa variable temporal.

YMIN: Representa la ordenada mínima de los puntos.

YMAX: Representa la ordenada máxima de los puntos.

B, TIPO: Representan variables de control.

PSEUDOCODIGO.

```
PROCEDURE GRAF;
BEGIN
   (obtener la mínima y máxima ordenada de todos los puntos);
   (inicialización de variables: B ← 0);
   (mandar a pantalla los datos de las respuestas);
```

```
   IF (valor mínimo ordenada→VMO > 0) THEN (VMO ← cero);END;
   (obtener el rango de la función a gráficar);
   (Normalizar rango con respecto a la resolución de pantalla);
   (mandar a gráficar los ejes sobre la pantalla);
   (inicializar:XXX←XMAX;TX←0;XMAX←XMAX-NUM;NUM←NUM+1;TA←NUM);
   IF (núm.puntos a gráficar→XMAX > a 300) THEN (XXX←NUM+299);END;
   FOR T=NUM TO (XXX) DO
      (control para obtener los puntos a gráficar no mayor a)
      (trescientos los cuales se tomaran esquiesparciados);
      (A ← (DATOS[T]-(Valor mínimo de la ordenada);
      (control de desbordamiento en los cálculos);
      (Y ← TRUNCAR(A);
      (INCR ← RESS DIV XMAX;incremento punto a punto de resuluc.);
      (TX  ←TX+1  ;    X  ← ((TX-1)*INCR+15:normalización  de  los)
      (incrementos dependiendo de los puntos que se tienen);
      (gráficar los puntos con la abciza X y la ordenada 190-Y);
   END for;
END GRAF;
```

Descripción del módulo EXPA.

En este módulo se cálcula la matriz de transición del sistema por medio de la serie exp(A·T), la serie se cálcula dependiendo en determinado momento de cuantos términos de la serie se obtienen para tener una buena aproximación, si la serie no converge por consiguiente la serie no convergera entonces para que no se produsca desbordamiento los cálculos se detendran.

Constantes del módulo EXPA.

EEP: Representa el valor máximo tipo real, para que en el caso de desbordamiento de los cálculos se paren los cálculos de la serie

Variables del módulo EXPA.

EXP: Representa la matriz de solución de la serie exp(A·T).

A: Representa la matriz A del sistema.

T: Representa la variable T, el cuál se pasara su valor.

N: Representa el número de variables de estado.

NUM: Representa el número de terminos para el cálculo de la serie.

I, J, K, KK: Representan contadores de iteraciones.

SUMA: Representan variable temporal.

RFAC: Representa la variable para el cálculo de los terminos del reciproco del factorial y las potencias del tiempo.

AMAX: Representa el termino máximo de las potencias matriciales de A de cada iteración.

IDEN: Representa la matriz identidad.

AMULT: Representa las potencias de la matriz A.

APORA: Representa la matriz A por alguna potencia de la matriz A.

PSEUDOCODIGO.

```
PROCEDURE EXPA;
BEGIN
   (inicializar matriz identidad IDEN,RFAC ← 1.AMULT y EXP← IDEN);
   (inicializar KK ← 1.AMAX ← 1);
   WHILE ( |amax|<max núm.)y(T<>cero)y(KK<(itera.max.) DO
```

```
      (obtención de las potencias de A ;APORA ← AMULT*A);
      (asignación de AMULT←APORA; RFAC ←(RFAC*T)/KK);
      (control para prevenir si habra desbordamiento);
      (incrementar en uno la variable KK);
      (suma de los terminos de la serie:EXP ← EXP+AMULT*RAFC );
   END while;
   IF (hay desbordamiento en los cálculos) THEN (mensaje) END if;
END EXPA;
```

Descripción del módulo NGRAFICAS.

En este procedimiento se manda a gráficar las n-gráficas de la ecuación de estado, variable por variable.

FUN: Representa una matriz donde se tendran los puntos cálculados de las respuestas del sistema variable por variable de estado.

L: Representa el número del punto donde se empieza a gráficar.

NOM: Representa el número de variables a gráficar.

I, J: Representan contadores de iteraciones.

PSEUDOCODIGO.

```
PROCEDURE NGRAFICAS;
BEGIN
   FOR i=1 TO (núm.de variables de estado o de salida) DO
      (vector PY ← (los puntos del arreglo de una columna de la)
      (respuesta);              (llamar a GRAF(pasando PY));
   END for;
END NGRAFICAS;
```

Decripción del módulo LECTURA.

En este procedure se leé la ecuación de estado.es decir las matrices A,B,C,D o G,H,C,D dependiendo del sistema y las variables de control del sistema.

Variables del procedure LECTURA.

I,K,J: Representan contadores de iteraciones.

BAN: Representa variable de control tipo bandera.

SI: Representa variable de control tipo entrada.

Descripción del módulo LEVERRIER.

En este módulo se cálculara la matriz de transición y por supuesto el polinomio de la ecuación característica del sistema.por medio de un algoritmo computacional desarrollado por Leverrier y por Faddeev.

Variables del procedure LEVERRIER.

A: Representa la matriz **A** o **G** del sistema de la ecuación de estado.

N: Representa el número de variables de estado.

DS: Representa los terminos D_i de las variables S^i del polinomio del DET(SI-A) o d(s).el cuál la salida DS es un vector.

I,J,K,II: Representan contadores de iteraciones.

TRA: Representa la traza de la matriz A del sistema.

DK: Representa el termino D_i de la variable S^i del polinomio.

ABK: Representa la matriz $A \cdot B_i$ del algoritmo de Leverrier.

IDEN: Representa la matriz identidad.

BB: Representa la matriz Bi del algoritmo de Leverrier.

PSEUDOCODIGO.

```
PROCEDURE LEVERRIER;
   PROCEDURE TRAZA;
   BEGIN
      (se obtiene la multiplicación de las matrices A*B);
      (se obtiene la traza de la matriz (A*B)  );
   END TRAZA;
BEGIN
   (inicializa matriz identidad IDEN, BB ← IDEN);
   FOR ii=1 TO (N) DO
      (K ← ii);
      (llamar a TRAZA(entra A y BB;sale TRA,ABK));
      (DK ← TRA/K ; DS[K] ← DK);
      (se obtiene: BB ← ABK-DK*IDEN fórmula algoritmo);
   END for;
END LEVERRIER;
```

Descripción del módulo DETERM.

En este módulo se determina el determinante de una matriz de cualquier orden por el método de desarrollo por cofactores, el cuál el algoritmo es recursivo.

Variables de la función DETERM.

AD: Representa la matriz de entrada para obtener su determinante.

ORDEN: Representa el orden de la matriz de entrada.

I, J, KK, III: Representan contadores de iteraciones.

SUMA: Representa variable temporal.

MENOR: Representa la matriz del menor elemento $a_{i,j}$ de un determinante de orden n, a la matriz de orden n-1 que se obtiene al suprimir en la matriz de orden n el renglón i y la columna j.

BAN: Representa variable de control tipo bandera.

PSEUDOCODIGO.

```
FUNCTION DETERM(AD matriz entrada):TIPO REALL;
   FUNCTION POT:TIPO REAL;
   BEGIN
      (inicializar T←-1;TR←-1);
      FOR iii=(potencia) TO 2 DEC -1 DO TR ← TR*T;END for;POT=TR;
   END POT;
BEGIN
   IF (orden matriz=1) THEN (determinante←elemento único);END if;
   IF (orden de la matriz≥ 2) THEN
      (inicializar SUMA ← cero);
      FOR J=1 TO (orden) DO
         FOR I=1 TO (orden - 1) DO
            (BAN ← false);
            FOR K=1 TO (orden - 1) DO
```

```
            (KK ← K);
            IF (K=J) ó (BAN=verdadero) THEN
               (BAN ← verdadero);  (KK ← K + 1);
            END if;
            MENOR[I,K] ← AD[I+1,KK];
         END for;
      END for;
      SUMA ← SUMA + POT(1+J)* AD[1,J]* DETERM(MENOR,ORDEN-1);
    END for;
    (DETERM ← SUMA);
  END if;
END DETERM;
```

Descripción del módulo CONTROESTA.

En este módulo se obtiene la matriz de controlabilidad de estado del sistema.

Variables del procedure CONTROESTA.

A,B: Representan las matrices del sistema de ec. de estado A o G y B o H.

N: Representa el número de variables de estado del sistema.

R: Representa el número de entradas del sistema.

CCESTA: Representa la matriz de controlabilidad de estado.

V: Representa variable de control tipo salida.

TIPO: Representa variable de control de entrada.

APOT: Representa la matriz de las potencias de la matriz A.

AAPOT: Representa la matriz temporal de la matriz APOT.

APOTB: Representa la múltiplicación de las matrices AAPOT por B.

COESTA: Representa la matriz de controlabilidad de estado.

COESTAT: Representa la matriz transpuesta de la matriz de controlabilidad de estado.

TERM: Representa la variable de dimensión de la matriz de controlabilidad cuando esta matriz se esta fórmando.

I, J, K, L, II: Representan variables de contadores de iteraciones.

RAN: Representa la variable del rango de la matriz COESTA.

SUMA: Representa variable temporal de almacenamiento.

PSEUDOCODIGO.

```
PROCEDURE CONTROESTA;
BEGIN
   (obtención de la matriz identidad APOT)
   (asignación de la matriz B al principio de COESTA ← B)
   (asignación TERM ← R)
   FOR ii=1 TO (n-1) DO
      (multiplicación de las matrices APOT*A)
      (aignación APOT ← APOT*A)
      (multiplicación de las matrices APOT*A)
      (asignación CONESTA[i,k] ← APOT*B;donde k=TERM+j)
      (TERM ← TERM+R)
   END for;
   (obtención de la matriz COESTA^T)
   (aignación de CCESTA ← COESTA)
```

(multiplicación de las matrices COESTA*COESTA^T)

IF (llamar a DET(COESTA*COESTA^T) <> cero THEN (EXITO);

ELSE (no es el sistema de controlabilidad de estado completa);

END if;

END CONTROESTA;

Descripción del módulo CONTROSAL.

En este módulo se determina la matriz de controlabilidad de salida del sistema, como respuesta da si esta matriz es de rango = m.

Variables del módulo CONTROSAL.

A, B, C, D: Representan las matrices de la ec. de estado A o G .B o H, C y D.

N, R, M: Representan las mismas dimensiones que las del sistema.

I, J, K, II: Representan contadores de iteraciones.

TERM: Representa la dimensión de la matriz de controlabilidad de salida, cuando la matriz se esta formando.

RAN: Representa la variable donde se tendra el rango de la matriz de controlabilidad de la salida.

SUMA: Representa variable de almacenamiento temporal.

CONSA: Representa la matriz de controlabilidad de la salida.

CONSAT: Representa la matriz transpuesta de la matriz CONSA.

CAAA: Representa la matriz temporal de múltiplicación de matrices.

PSEUDOCODOGO.

```
PROCEDURE CONTROSAL;
BEGIN
   (multiplicación de las matrices C*B → CONSA[i,j];TERM ← R)
   FOR II=1 TO (n-1) DO
      (multiplicación de las matrices C*A) ; (C ← C*A)
      (multiplicación de las matrices C*B)
      (CONSA[i,k] ← C*B ; donde k=TERM+J)
      (TERM ← TERM+R)
   END for;
   (CONSA[i,k] ← D[i,j]; donde k=TERM+J)
   (obtener la matriz CONSA^T)
   (multiplicación de las matrices CONSA*CONSA^T)
   IF (llamar a DET(CONSA*CONSA^T)) <> cero THEN   EXITO;
      ELSE (no es de controlabilidad de salida completa)
   END if;
END CONTROSAL;
```

Descripción del módulo OBSERVAB.

En este módulo se obtiene la matriz de observabilidad del sistema.

Variables del procedure OBSEVAB.

A: Representa la matriz A o G del sistema de la ec. de estados.

AT: Representa la matriz C del sistema de la ec de estados en el momento de la entrada,despues será la matriz A^T.

N, M: Representan el número de variables de estado y el número de salidas del sistema de ecuaciones de estado.

V: Representa una variable de control de salida la cuál es verdadera si es del tipo de observabilidad completamente el sistema.

TIPO: Representa variable de control de salida.

I, J, K, II: Representan variables de contadores de iteraciones.

TERM: Representa la dimensión de la matriz de observabilidad cuando esta matriz se esta formando.

RAN: Representa el rango de la matriz de observabilidad.

SUMA: Representa una variable temporal.

CT: Representa la matriz transpuesta de C del sistema.

AAC: Representa una matriz temporal de la múltipl. de $^{T}A*C$.

OBS: Representa la matriz de observabilidad.

OBST: Representa la matriz transpuesta de la matriz OBS.

PSEUDOCODIGO.

```
Procedure OBSERVAB;
BEGIN
   (Inicializa TERM ← M)
   (Inicializa matrices,identidad y transpuestas AT y CT)
   (Asignación OBS[n,m] ← CT[n,m])
   FOR II=1 TO Variables de estado menos uno DO
         (Multiplicación de las matrices A*A^T → A)
         (Multiplicación de las matrices A*C^T→ OBS[reng,col+TERM])
         (TERM ← TERM+R)
   END for;
   (Obtención de OBS^T )
```

```
   (Multiplicación de matrices OBS*OBS^T )
   IF El det(OBS*OBS^T) <> Cero THEN (Exito.)
      ELSE  (el sistema no es de observabilidad completa)
   END if;
END OBSERVAB;
```

Descripción del módulo EXPRESION.

En este módulo se evaluan las funciones de las entradas del sistema de ecuaciones de estado,cuyo vector de entradas son cadenas de caracteres y la variable independiente es la variable tiempo,la cuál la salida es el resultado de esta función evaluada en el tiempo determinado,donde la función puede ser expresada en terminos algebraicos utilizando la suma (+),resta(-),múltiplicación (*),divición (/),las funciones unarias (+),(-) y la función potenciación (^) y puede contener funciones trascendentes como las funciones seno (sen(x)),coseno (cos(x)) y exponencial (exp(x)), donde la x puede representar la variable tiempo ó otra función, donde no se tiene límite de niveles de profundidad.Solo en las funciones es significativa la primera letra,las dos primeras letras o las tres primeras letras,como variable del tiempo acepta cualquier letra o conjunto de letras como una sola variable,es decir aceptaria como variable la T,TIEMPO,t,tiempo,en los números se acepta cualquier número real.

El algoritmo tiene seis niveles de profundidad,los espacios en blanco los ignora,algunas rutinas de este algoritmo son mutuamente recursivas.

Variables del módulo EXPRESION.

CADENA: Representa la cadena de caracteres de la función del vector de entradas del sistema.

TIEMPO: Representa la variable independiente de la función el cuál es el tiempo.

RESULT: Representa la variable de salida de los resultados de los procedures, siendo esta variable la que tendra el resultado final

TOKEN: Representa variable tipo cadena de caracteres la cuál representara un token de la cadena en donde se encuentra toda la función.

TIPOTOKEN: Representa una variable definida previamente por la variable llamada TIPO, el cuál tendra de que tipo es el token que se tiene en ese momento, la cuál la variable TIPOTOKEN podra tener el valor de número, delimitador ó de variable.

TT: Representa la variable para apuntar a los carácter de la cadena de la función en forma acendentemente, uno por uno.

Descripción del procedure OBTIENETOKEN.

En este módulo se desaparta o se obtiene un token de la función analizada además obteniendose de que tipo (número, delimitador ó variable) es el token, desechando los blancos en médio de los carácteres es decir no los toma en cuenta, donde delimitador podra ser + , - , / , * , ^ , = , (,) , sen , cos , exp.

Variables del módulo OBTIENETOKEN.

CTEMP: Representa una cadena de tipo temporal.

Descripción de los procedures EXPREC, ATOMON2, ATOMON3, ATOMON4, ATTOMON5CERO, ATOMON6CERO y ATOMOCERO.

En estos procedures se analiza y se evalua la función completa, donde la mayor parte de estos procedures se llaman entre si recursivamente,con el grado de profundidad que requiera la función en cuestión.Los detalles del algoritmo se dan en el pseudocadigo del módulo EXPRESION.

Variables de los procedures anteriores.

RESULT: Representa variable de salida de los diferentes procedures el cuál al final es el resultado total de la función a evaluar.

OP: Representa variable de tipo carácter,la cuál funciona como control de las evaluaciones de las operaciones que se tienen que hacer,tomando la operación que se realizara.

GUARDA: Representa la variable de salida de los procedures el cuál sera el resultado de la evaluación de la función.

PSEUDOCODIGO.

FUNCTION ESLETRA;Boolean;

(Verifica si el carácter enviado es una letra)

FUNCTION ESDIGITO;Boolean;

(Verifica si el carácter enviado es digito)

FUNCTION ESDELIMIT;Boolean;

(Verifica si el carácter enviado es un delimitador)

```
FUNCTION ESBLANCO;Boolean;
   (Verifica si el carácter enviado es un blanco o si es una nüeva
    linea o tab,es afirmativo devuelve verdadero,contrario falso)


PROCEDURE EXPRESION.
PROCEDURE OBTIENETOKEN;
BEGIN
   WHILE (si es BLANCO el token de la cadena) DO
      (Incrementa en la función(cadena) un token)
   END while;
   IF (fin de cadena) THEN TOKEN ← '$'; END if;
   IF token es DELIMITADOR  THEN
        (tipotoken ← delimitador)
       (token ← token de la cadena;incrementa un token de cadena)
   ELSIF (verificar si el token es alguna función) THEN
           (tipo token ← delimitador)
           (token ← token cadena;incrementa un token de cadena);
      ELSIF (token es variable) THEN
              (tomar las letras que corresponden a la variable)
               (tipo token ← variable);
         ELSIF (token es DIGITO)  THEN
                   (toma  todos  los  digitos  de  los  token  que
                    corresponden al número)
           END if;
END OBTIENETOKEN;
```

```
PROCEDURE ERRORES;
   (manda a escribir tipo del error)
FUNCTION POTENC;
   (obtiene la potencia de un número real;A^B=EXP(B*Ln(A)) )
FUNCTION VALORVAR;
   (toma la variable su valor que le corresponde en ese momento de
    tiempo)
PROCEDURE ARITM;
   (obtiene las operaciones aritméticas de dos números)


PROCEDURE EXPREC.
BEGIN
   (llama a oBTIENETOKEN;se obtiene un token de la cadena)
   IF (longitud del token <> cero) THEN
      (llama a ATOMONN2(sale result);
      ELSE   RESULT ← 0;
   END if;
END EXPREC;


PROCEDURE ATOMON2;
BEGIN
   (llama a ATOMON3(sale result);
   (operación ← primer token)
   WHILE (op es suma o resta) DO
      (llama a OBTIENETOKEN)
      (llama a ATOMON3(obtiene guarda))
```

```
      (llama a ARITM(entra result y guarda,sale nuevo result))
      (operación ← token primero)
   END while;
END ATOMON2;

PROCEDURE ATOMON3;
BEGIN
   (llama a ATOMON4(obtiene result))
   (op ← primer token)
   WHILE (op es multiplicación o divición) DO
      (llama a OBTIENETOKEN;llama a ATOMON4(obtiene guarda))
      (llama a ARITM(obtiene result))
      (op ← primer token)
   END while;
END ATOMON3;

PROCEDURE ATOMON4;
BEGIN
   (llama a ATOMON5CERO(obtiene result)
   IF (primer token = operación potenciación) THEN
      (llama a OBTIENETOKEN)
      (llama a ATOMON4(obtiene guarda) ;llama ARITM(obt result))
   END if;
END ATOMON4;

PROCEDURE ATOMON5CERO;
```

```
BEGIN
   IF (tipo token es función únaria o seno coseno o exp) THEN
      (llama a OPTIENETOKEN);
   END if;
   (llama a ATOMON6CERO(obtiene result))
   (se obtiene resiltado dependiendo si es función únaria,seno,
   coseno,exp,de otro modo no hace nada)
END ATOMON5CERO;

PROCEDURE ATOMON6CERO;
BEGIN
   IF (token es un parentesis y el tipo es delimitador) THEN
      (llama a OBTIENETOKEN ; llama a ATOMON2(obtiene resultado))
      IF (el token acaba con parentesis) THEN
         (llama a ERRORES)
      END if;
      (llama a OBTIENETOKEN)
   ELSE
      (llama a ATOMOCERO(obtiene resultado))
   END if;
END ATOMON6CERO;

PROCEDURE ATOMOCERO;
BEGIN
   IF (el tipo token es un número) THEN
      (se combierte una cadena de token en su número real equiv.)
```

```
  ELSIF (si el tipo token es variable) THEN
       (llama a VALORVAR(token),el cuál pone el valor que toma
       el tiempo en la variable en ese punto → resultado)
     ELSE      (llama a ERRORES)
    END if;
    (llama a OBTIENETOKEN)
END ATOMOCERO;

BEGIN
   (inicializa la cadena en el primer token ← 1)
   (llama a EXPREC(Result);se obtiene resultado en result)
END EXPRESION;
```

Descripción del módulo INVERSA.

En este módulo se obtiene la matriz inversa de cualquier matriz cuadrada y se obtiene el determinante de dicha matriz.

Variables del módulo INVERSA.

A1: Representa la matriz de entrada y al final tendra la matriz transformada durante el proceso, obteniendo la inversa.

N: Representa la variable de que dimensión es la matriz.

DET: Representa el determinante de la matriz.

PIV: Representa el vector donde se tendran los elementos del renglon en donde se encuentra el pivote.

REN: Representa el renglon donde se encuentra el pivote.

COL: Representa la columna donde se encuentra el pivote.

I, J, R, RR, M: Representan contadores de iteraciones.

MAX: Representa al máximo elemento entre dos elementos de la matriz

PIVOTE: Representa al elemento mayor que se escogio como el pivote.

AA: Representa la matriz para poder intecambiar columnas en el proceso de inversión de la matriz.

```
PROCEDURE INVERSA;
BEGIN
   FOR R=1 TO (dimensión de la matriz) DO
      (se busca elemento pivote en la matriz;PIVOTE ← elem.selec.)
      IF (si el pivote no es cero) THEN
         IF (pivote no esta en la diagonal principal) THEN
            (cambiar el pivote en la diagonal principal)
         END if;
         (dividir el renglón del pivote entre el pivote;se reducen
         los elementos de renglones diferentes al del pivote)
      ELSE  (el determinante sera cero)
      END if;
      IF (pivote no es cero) THEN
         (se intercambian las columnas de la matriz)
      ELSE  (determinante es igual a cero)
      END if;
   END for;
END INVERSA;
```

Descripción del módulo DATOS.

En este módulo se dan las variables de datos de control del programa, como serian las variables para el proceso de gráficación, del tiempo de respuesta, número de puntos a gráficar, número de terminos para el cálculo de la serie $\exp^{AT}$, sedan las funciones de entrada del sistema de ec. de estado y el vector de condiciones iniciales del sistema.

Variables del módulo DATOS.

P: Representa la variable que tendra el número de puntos de las respuestas del sistema a gráficar.

TI: Representa el tiempo de inicio del sistema.

INCREM: Representa el incremento de tiempo entre dos puntos(TR/P).

TR: Representa hasta que tiempo se quiere sea cálculada las respuestas del sistema.

I: Representa variable de control de iteraciones.

BAN: Representa variable de control, representa a una bandera.

Descripción del módulo RESPLIBRE.

En este módulo se obtienen los cálculos para obtener las respuestas libres del sistema de ec. de estado.

Variables del módulo RESPLIBRE.

G: Representa la matriz exp(AT) o la matriz G.

N: Representa al número de variables del sistema.

P: Representa el número de puntos de las diferentes respuestas.

TI: Representa al tiempo de inicio del sistema.

INCREM: Representa al incremento del tiempo entre dos puntos de las respuestas.

I, J, II, K: Representan contadores de iteraciones.

XX: Representa a un vector temporal del algoritmo.

XXI: Representa al vector que tendra el valor de las respuestas en cada punto calculado.

PSEUDOCODIGO.

```
PROCEDURE RESPLIBRE;
BEGIN
   (inicializar las matrices XXI ← XI ; X[i,0] ← XXI)
   FOR ii=1 TO (puntos de la respuesta)  DO
      (cálculo de la respuesta libre del sistema por medio de la
       fórmula recursiva del sistema en fórma discretizada)
   END for;
END RESPLIBRE;
```

Descripción del procedure ENTRADA.

En este módulo se manda a evaluar las funciones del vector de entrada, una por una, con el valor que entra del tiempo.

Variables del módulo ENTRADA.

I: Representa variable de contador de iteraciones.

RESULTADO: Representa el valor del resultado de la evaluación de la función.

PSEUDOCODIGO.

```
PROCEDURE ENTRADA;
BEGIN
  FOR I=1 TO (número de variables de entrada) DO
     (llamar a EXPRESION,para el cálculo del valor de las
      funciones de las entradas,pasando la variable independiente
      que es el tiempo,dependiendo en que punto se esta)
  END for;
END ENTRADA;
```

Descripción del módulo INTEGEXPB.

En este módulo se hacen los cálculos de la integral de la función de las matrices EXP(AT) * B,es decir se hacen los cálculos para obtener la matriz H por la fórmula general de convertir el sistema de tiempo-continuo a un sistema de tiempo-discreto,donde se tendra que obtener H,pora posteriormente poder obtener las respuestas del sistema.Donde este algoritmo funciona como un camino para obtener H cuando no se puede obtener H por la fórmula.Ya que el número de puntos para que cálculemos la integral de la función los podemos cálcular como nosotros querramos,depende del intervalo de integración,entonces podemos poner el número de puntos que nosotros queremos para el obtener el cálculo de la integral más aproximada a la real.

Variables del módulo INTEGEXPB.

EXPB: Representa la matriz H, es decir es el valor de la integral de la matriz exp(AT) por la matriz B. La cuál se tendra al final.

A: Representa la matriz de entrada A del sistema de ec. de estado.

B: Representa la matriz de entrada B del sistema de ec. de estado

N: Representa el número de variables de estado del sistema.

R: Representa el número de entradas del sistema de ec. de estado.

PP: Representa el número de puntos que se tomaron para hacer el cálculo de la integración en el intervalo de incremento que se cálculo de un punto a otro punto.

TI: Representa el tiempo inicial para las respuestas.

INC: Representa el incremento de tiempo del punto a otro punto, para el cálculo de las respuestas.

I, J, K, II, KK: Representan contadores de iteraciones.

PA: Representa variable de control del número de puntos.

TPUN, TE: Representa variable de control para el algoritmo.

INCREM: Representa al incremento de un punto a otro punto en el cálculo de la integración de la función exp(AT)*B en el intervalo de tiempo INC.

SUMPUN: Representa matriz que guarda las sumas de la integración.

PRIPUN: Representa el Cálculo del primer punto del algoritmo.

ULTPUN: Representa el cálculo del último punto del algoritmo.

EXPMULTB: Representa la matriz de la múltiplicación de exp(AT)*B.

PSEUDOCODIGO.

```
PROCEDURE INTEGEXPB;
PROCEDURE FUNEXPB;
BEGIN
   (llama a EXPA;en donde se cálcula e^AT pasando el tiempo)
   (se multiplican las matrices [e^AT][B])
END FUNEXPB;
BEGIN
   (se obtiene el nuevo incremento de integración para el cálculo
    de la integración númerica entre un primer incremento de un
    punto inicial a otro.el cuál se cálcula del incremento entre
    los puntos de integración que se escogierón para llevar a cabo
    la fórmula de integración del método trapecial,ya que podemos
    escoger el número de puntos este método es bueno.)
   (inicializar variables;TPUN,TE,PA y matrices)
   FOR i=1 TO (punto anterior del punto de integración) DO
      (sumar el incremento de integración al punto TE)
     (llamar a FUNEXPB(pasando el tiempo del punto adonde se hace)
      (el cálculo y TE=tiempo del primer punto de integración)
     (suma de las matrices obtenidas en el proceso de cada pasada)
   END for;
   (obtención del primer punto de la fórmula)
   (obtención del último punto de la fórmula)
   (obtención de la suma de las tres matrices anteriores)
END INTEGEXPB;
```

Descripción del procedure FUNEXPB.

En este procedure se cálculan las diferentes múltiplicaciones de las matrices EXP(AT) por la matriz B.

Variables del procedure FUNEXPB;

TPUN: Representa a un punto en un determinado tiempo.

TE: Representa al tiempo inicial de el sistema.

EXPMULTB: Representa la matriz de salida la cuál es EXP(AT)*B.

I,J,K: Representan contadores de iteraciones.

SUMA: Representa una variable temporal.

AEXPTX: Representa la matriz EXP(AT).

Descripción del módulo FORMINTEG.

En este módulo se obtiene la matriz H igual que en el procedure INTEGEEXPB, pero en este caso se útiliza la fórmula que se demostro en los capitulos anteriores, cuando la matriz A es no-singular entonces podemos simplificar la integración por una fórmula matematica más exacta que si útilizaramos la integración númerica.

Variables del módulo FORMINTEG.

HH: Representa la matriz de salida H.

GG,B: Representan las matrices G y B del sistema respectivamente.

N,R: Representan variables de estado y salida del sistema.

INVA: Representa la matriz inversa de A.

AA: Representa la múltiplicación de matrices $A^{-1}*G$.

PSEUDOCODIGO.

```
PROCEDURE FORMINTEG.
BEGIN
   (llama a INVERSA(de la matriz A ; INVA ← A^-1)
   (suma de las matrices (G - I) )
   (llamar a PROD(multiplicación de INVA*(G-I) → AA )
   (llamar a PROD(multiplicación de AA*B → HH = H ))
END FORMINTEG;
```

Descripción del módulo MENEIGEN.

En este módulo se obtiene el menor eigenvalor del sistema. Donde el método que se útiliza es el método de aproximaciones sucesivas.

Variables del módulo meneigen.

AA: Representa la matriz A o G del sistema.

MENVCAR: Representa el menor eigenvalores del sistema.

C: Representa al vector del producto de $A*X_i^{(k)}$ del algoritmo.

XA: Representa a los valores iniciales del vector característico que en el proceso se transforma en $X_1^{(k)}$.

V: Representa el valor caracteristico de mayor valor absoluto.

DET: Representa el determinante.

I,J: Representan contadores de iteraciones.

ITER: Representa al número de iteraciones.

CONT: Representa contador de control del algoritmo.

NITER: Representa al número maximo de iteraciones.

AUX: Representa bandera de control.

PSEUDOCODIGO.

```
PROCEDURE MENEIGEN;
BEGIN
   IF (determinante de A o G  <> cero) THEN llamar a INVERSA(A);
   ELSE   (bandera de que el determinante es cero);     END if;
   (control de cuantas iteraciones dependiendo del grado del s.)
   (inicializar banderas,contadores y datos del algoritmo)
   WHILE (contador de iteraciones≤ITER MAX) y (DER<>0) y (AUX) DO
      (algoritmo para encontrar el menor valor caracteristico)
      IF (no-convergio el método) THEN (sigue);
      ELSE (bandera=AUX ← false)  ;              END if;
   END while;
   IF (convergio el método) THEN (menor valor caracteristico←1);
   ELSE (menor valor caracteristico ← ABS(( 1/Vmin)*5)  ;
   END if;
END MENEIGEN;
```

Descripción del módulo RESFORITER.

En este módulo se obtienen los puntos de las respuestas forzadas de el sistema,por medio de la discretización de el sistema de tiempo-continuo y obteniendo los cálculos iterativamente.

Variables del módulo RESFORITER.

GGG: Representa la matriz G discretizada del sistema.

HHH: Representa la matriz H discretizada del sistema.

TI: Representa el tiempo inicial de la respuesta del sistema.

INNC: Representa al valor del incremento de los puntos.

AREA: Representa la matriz en donde se tienen las respuestas forzadas de las variables de estado.

U: Representa al vector de las entradas del sistema.

EXPBU: Representa al vector de la multiplicación H*U.

XXK: Representa al vector donde se tiene el anterior cálculo de la fórmula de iteración de las respuestas forzada.

XK: Representa al vector donde se tiene la multiplicación G*XXK.

I,J,K,II: Representan contadores de iteraciones.

SUMA: Representa variable temporal.

PSEUDOCODIGO.

```
PROCEDURE RESFORITER;
BEGIN
   (asigna las variables iniciales de X en la variable XXK)
   (asigna T tiempo ← el incremento de inicio ; AREA ← X inicial)
   FOR II=1 TO (número de puntos) DO
      (llama a ENTRADA;cálcula las funciones de la entrada en el
      tiempo que le corresponde en ese punto)
      (multipli. las matrices H*U(entradas) → EXPBU)
      (G*XXK → XK ;valor anterior de la ec.iterativa XXK)
      (sumar las matrices XK+EXPBU → AREA     ;asigna XXK ← AREA)
      (incrementar el tiempo)
   END for;
END RESFORITER;
```

Descripción del módulo BAIRSTOW.

En este módulo se obtienen los eigenvalores del sistema de ec. de variables de estado,los eigenvalores se obtienen en números complejos,teniendo el polinomio por el algoritmo de Leverrier,de aquí el método de Bairstow permite factorizar un polinomio de segundo grada a partir de un polinomio p(x) de grado n,obteniendo tanto raíces reales como raíces complejas del polinomio p(x).

Variables del módulo BAIRSTOW.

AAAAA: Representa la matriz A o G del sistema.

GRAD: Representa al grado del polinomio.

RAICESR: Representa al vector de las raíces con la parte real.

RAICESI: Representa al vector de raíces con la parte imaginaria.

ITER: Representa al número máximo de iteraciones.

CONT: Representa contador de iteraciones.

M: Representa la variable decrementador del grado del polinomio.

NN: Representa el grado del polinomio actualizado cuyas raíces se están buscando en ese momento.

NM1,IM1: Representa variable de control del método Bairstow.

NEWN: Representa al nuevo polinomio cuando se reduce el polinomio.

P: Representa a un coeficiente (p) de X^1 de la ec. cuadratica.

Q: Representa a un coeficiente (Q) de X^0 de la ec. cuadratica.

RAD: Representa a la raiz del descriminante.

CMB: Representa a una fórmula del método de Bairstow.

DENUM: Representa a una variable del método.

DELP,DELQ: Representa al incremento P y al de Q respectivamente.

RADTR: Representa al descriminante de la ec. cuadratica.

A: Representa al vector de coeficientes del polinomio.

B: Representa,vector de coeficientes del polinomio de grado NC-2.

C: Representa,vector de coeficientes del polinomio de grado NC-4.

RTREA: Representa al vector de la parte real de las raices.

RTIMA: Representa al vector de la parte imaginaria de las raices.

POLEVER: Representa un vector donde se tienen los coeficientes del polinomio característico.

SALIRSE: Representa variable de control,bandera de control.

PSEUDOCODIGO.

```
PROCEDURE BAIRSTOW;
BEGIN
   llama a LEVERRIER(obt.del polinomio de la ec.caract.de A);
   (cambio de signo de los exponentes del pol.de leverrier);
   (inicialización de variables de control del algoritmo);
   WHILE (iteraciones≤maxI.)y(grado actualizado pol)y(sali) DO
      (obtención de los coeficientes de B y C del algoritmo);
      (obtimización de los valores de P y Q);
      (resolver el sistema de ec.para determinar DELP y DELQ);
      (evaluación de valores incrementados de P y Q);
      IF (no convergio el metodo)y(iterac≥maxI.) THEN
         (cada ves que entra obtiene núevos valores iniciales);
         (contador de iteraciónes ← cero;inicializa);
      ELSE
```

```
      IF (no-convergio el proceso)y(iter<maxI.) THEN
         (incrementar el número de iteraciones en uno);
      ELSE
         IF (convergio el proceso) THEN
            (se obtienen las raíces del factor cuadrático);
            (reduce el orden del polinomio y cambio de
            coeficientes de los polinomios A y B);(iteracio←0);
           (dar nuevos datos iniciales a P y Q );
         END if;
    END if;
  END while;
  IF (grado del polinomio último es = dos) THEN
     (se obtienen el par de raices del último factor cuadrático);
  END if;
  IF (grado del polinomio último es = uno) THEN
     (se obtiene la raiz del factor lineal);
  END if;  (cambio de vectores para las raices);
END BAIRSTOW;
```

Descripción del módulo MULTPOLIN.

En este módulo se multiplican polinomios de diferentes grados, para obtener un polinomio al final único,ordenado de sus potencias de este último polinomio.

Variables del módulo MULTPOLIN.

P1: Representa a un apuntador en donde apunta al primer polinomío.

P2: Representa a un apuntador en donde apunta al segundo polinomio.

POL: Representa a un apuntador, en donde apunta al polinomio suma.

P, SIGP, SIGP1, PP1, SIGP2: Representa apuntadores del algoritmo.

PP: Representa al vector de apuntadores en donde se encuentran los polinomios resultantes de las multiplicaciones al multiplicar un polinomio por un coeficiente o término de otro polinomio.

NPOL: Representa contador de polinomios de control del algoritmo.

Descripción del procedure BUSTER.

En este procedure se buscan o se obtienen los terminos de los polinomios que son de igual potencia del que se busca.

Variables del procedure BUSTER.

POL: Representa al apuntador en donde apunta al polinomio en donde se llevara la buscara.

EXP: Representa al coeficiente del polinomio del exponente que se busca del polinomio.

P, PANT: Representa apuntadores del algoritmo.

BUSCA: Representa variable de control del algoritmo.

Descripción del procedure SUMPOL.

En este procedure se obtiene la suma de los polinomios que se obtienen al estar multiplicando polinomios.

Variables del procedure SUMPOL.

POLL: Representa un apuntador, al inicio apunta al primer polinomio de entrada que se sumara y a la salida apuntara al polinomio resultante de la suma.

POL2: Representa a un apuntador en donde apuntara al segundo polinomio que se sumara.

P, PP, PANT, PPANT: Representa apuntadores para el algoritmo.

PRIM: Variable de control del algoritmo.

PSEUDOCODIGO.

```
PROCEDURE MULTPOLIN;
PROCEDURE BUSTER(pasa apunt del polinomio y el exp);
BEGIN
   (inicializa las variables de control y apuntadores)
   WHILE (si el apuntador apunta a un dato) DO
      IF (apuntador apunta al exponencial buscado) DO EXITO ;
      ELSE  (sigue buscando,recorre apuntadores pol.);  END if;
   END while;
   IF (no se encontro el exp.buscado) THEN (mensaje sin exito);
     ELSE (manda control en donde se  encuentra  por  apuntador);
   END if;
END BUSTER;

PROCEDURE SUMPOL(pasa apuntadores de pil.1 y pol.2);
BEGIN
   (p ← pol.1)
   WHILE (no sea llegado al final del p) DO
```

```
      llamar a BUSTER(pasando a pol.2 y de p (el exp)
      IF (exito en BUSTER) THEN
         (sumar los coeficientes) ; (y pol.2 siguiente);
      END if;
      (pasar el apuntador al termino siguiente de p)
   END while;
   (poner los terminos restantes del pol.2 en el pol de p);
   (el polinomio total esta en pol.1);
END SUMPOL;
BEGIN
   WHILE (no es el final del polinomio 2) DO
      (se crea un nuevo polinomio[n] y control de apuntadores para
      los polinomios que serán creados);
      WHILE (no es el final del polinomio 1) DO
         (multiplicación de los coeficientes de los 2 polinomios);
         (suma de los dos exponentes de los dos polinomios);
         IF (el polinom.1 no es el final) THEN
            (control para obtener otro registro del polinomio[n]
         ELSE (fin polinomio[n] a tierra);
         END if;
      END while;
      (siguiente registro del apuntador del polinomio.2);
   END while;
   IF (numero de polinomios creados = tres) THEN
      llama a SUMPOL(pasa a pol[2] y pol[3] → pol[2]); END if;
   llama a SUMPOL(pasa a pol[1] y pol[2] → pol[1]; pol ← pol[1];
```

END MULTPOLIN;

Descripcion del modulo creapolin.

En este modulo se crea el polinomio para el diseño del sistema de la ecuacion caracteristica, dando solamente los eigenvalores del nuevo sistema que se quiera, los eigenvalores podran ser números complejos conjugados ya que los sistemas físicos tienen esa característica.

Variables del módulo CREAPOLIN.

N: Representa al orden del polinomio.

ALFAS: Representa un vector donde se tendran los coeficientes del polinomio que se creara.

NUMCOM: Representa al número de raíces pares complejas conjugadas.

NUMREAL: Representa al número de raíces reales del polinomio.

I: Representa contador de iteraciones.

NP: Representa el contador de polinomios que se crearan en el proc.

NUMRAIR: Representa el número de raíces de parte real solamente.

MARCA: Representa variable de control para el compilador pascal.

POL: Representa un vector de apuntadores de polinomios.

R: Representa la parte real leida de la raiz compleja.

Z: Representa la parte imaginaria de la raiz compleja.

A: Representa el coeficiente de X^1 del polinomio cuadratico.

B: Representa el coeficiente de X^0 del polinomio cuadratico.

P, SIG, POLTOT: Representan apuntadores de control del algiritmo.

PSEUDOCODIGO.

```
PROCEDURE CREAPOLIN;
BEGIN
   (leer 'número de raíces complejas')';
   IF (número raíces complejas > 0) THEN
      FOR i=1 TO (número raíces complejas) DO
         (leer raíces complejas:parte real y parte imaginaria);
        (creación de los polinomios independientes para cada raiz)
      END for;
   END if;
   (cálculo del núm.raices Reales←(var.de estado)-2*(núm.R.Compl.)
   IF ((núm.Raices Reales) > cero)   THEN
      FOR i=1 TO (número Raíces Reales) DO
         (leer las raíces reales);
         (creación de los polinomios,uno para cada raiz);
      END for;
   END if;
   FOR i=1 TO (número de polinomios creados) DO
      llamar a MULTPOLIN(pasando los polinomios creados);
   END for;
   (pasar los coeficiestes del polinomio a un vector);
END CREAPOLIN;
```

Descripción del módulo ECCARACT.

En este módulo se obtiene ¢(*) ó ¢(●) por medio del teorema de Cayley-Hamilton que **A** o **G** satisface su propia ecuación característica,la cuál se usara para obtener la matriz de realimentación de estado K y ademas Ke.

A: Representa la matriz A o G del sistema de ec. de estado.

N: Representa el número de variables de estado.

FIA: Representa la matriz donde se tendra al final la matriz ϕ([illegible]) ó la matriz ϕ([illegible]) dependiendo del tipo de sistema.

ALFAS: Representa al vector de coeficientes del polinomio del diseño del sistema de la ec.caracteristica de los eigenvalores.

I,J,K,KK: Representan contadores de iteraciones.

SUMA: Representa variable temporal.

APORA: Representa a las multiplicación de matrices A^n por A.

TERMINO: Representa a la suma de los terminos matriciales para la obtención de la ec. caracteristica para obtener $\phi(A)$ ó $\phi(G)$.

IDEN: Representa la matriz identidad.

APORAT: Representa la matriz transpuesta de APORA.

PSEUDOCODIGO.

```
PROCEDURE ECCARACT;
BEGIN
   (obtención de la matriz identidad)
   (obtención del primer termino del teorema de Cayley Hamilton)
   FOR kk=1 TO (número de variables de estado) DO
      (obtención de las potencias de la matriz A)
      IF (no es el último termino) THEN
         (obtención del siguiente termino del teorema Cayley H);
         (obtención de la suma de los terminos anteriores);
```

ELSE (última potencia de A^n+los anteriores terminos)

END if;

END for;

END ECCARACT;

Descripción del módulo KREALIMENT.

En este módulo se crea la matriz de ganancia de realimentación de estado K de dimensión [1xn] para sistemas de única entrada.

Variables del módulo KREALIMENT.

N: Representa al número de variables de estado.

FIA: Representa la matriz ¢(*) o ¢(*) dependiendo del sistema.

CONEST: Representa la matriz de controlabilidad de estado.

KREALIM: Representa la matriz de ganancia de realimentación K.

TEMPO: Representa matriz temporal de CONEST*FIA.

VECTOR: Representa un vector de la fórmula para obtener K.

I,J,K: Represemta contadores de iteraciones.

DET: Representa al determinante de la matriz CONEST.

SUMA: Representa variable temporal.

PSEUDOCODIGO.

```
PROCEDURE KREALIMENT;
BEGIN
   (inicializar vectores,para la fórmula de Ackermann's);
   (llamar a INVERSA   ;(multiplicación de CONEST*¢(A) → TEMPO);
   (multiplicación de TEMPO*(Vector de la fórmula) → K );
```

END KREALIMENT;

Descripción del módulo KOBSERV.

En este módulo se obtiene el vector de ganancia de realimentación del observador Ke [nx1],para sistemas de única salida, para el diseño del observador.

Variables del módulo KOBSERV.

A: Representa la matriz A o G del sistema de ec. de estado.

C: Representa la matriz C del sistema.

FIA: Representa la matriz $\phi(A)$ o $\phi(G)$ del sistema.

N: Representa el número de variables de estado.

KESTIM: Representa el vector K_e del diseño del observador[nx1].

TEMPA, AA, SUMA: Representa una matriz temporal.

OBS: Rep. matriz de observabilidad de sistemas de única-salida.

APORA: Representa matriz para obtener las potencias de la matriz A.

I, J, K: Representan contadores de iteraciones.

DET: Representa al determinante de la matriz OBS,no se útiliza.

VECTOR: Representa un vector de la fórmula para obtener la matriz de realimentación del observador K_e.

PSEUDOCODIGO.

```
PROCEDURE KOBSERV;
BEGIN
   (obtención de la matriz de observabilidad completa);
   (llamar INVERSA);
```

(inicialización del vector de la fórmula de **A.**);

(multiplicación de ϕ(A) * (matriz de observabilidad → TEMPA);

(multiplicación de TEMPA*(vector de la fórmula → K(estimada));

END KOBSERV;

Descripción de la función PSEUALEA.

En esta función se obtiene un número en forma pseualeatoria, para la formación de matrices.

Constantes de la función PSEUALEA.

MULT,INCRE,MODULO: Representan números constantes para la fórmula.

Variables de la función PSEUALEA.

SEMILLA: Representa variable de salida del número pseualeatoriamente, pasando un número anterior como semilla.

Descripción del módulo BUSCABPORW.

En este módulo se busca la matriz **W** del desarrollo expuesto para encontrar la K matriz de realimentación de estado para sistemas de múltiples entradas, para que el núevo sistema que se forme del par (A.BW) sea controlable de estado completamente.

Variables del módulo BUSCABPORW.

AA,BB: Rep. matrices del sistema A o G y B o H respectivamente.

ROM: Representa el número de R-entradas ó la M-salidas del sistema, dependiendo si es el sistema original o su dual.

N: Representa el número de variables de estado.

W: Representa la matriz **W** del algoritmo.

BBW: Representa la matriz B*W del algoritmo.

I, J, K, CONTADOR: Representan contadores de iteraciónes.

SEMILLA: Número semilla para obtener el número pseudoaleatorio.

SUMA: Representa variable temporal.

VER: Representa de control de respuesta de CONTROESTA.

PSEUDOCODIGO.

```
PROCEDURE BUSCABPORW;
BEGIN
   (inicializa variables del algoritmo y contadores);
   WHILE ((matriz controlabilidad estado rango=N)y(iter<Cont)) DO
      (llamar a PSEUALEA(obteniendo la matriz W);
      (multiplicación de B*W);
      (llamar a CONTROESTA(obteniendo si el rango=N → ver);
   END while;
END BUSCABPORW;
```

Descripción del módulo REALIMEN.

En este módulo se controla todo el método para la realimentación, en principio si es posible obtener la matriz de ganancia de realimentación de estado K [rxn]. Tanto para sistemas de única-entrada como de sistemas de múltiples-salidas.

Variables del módulo REALIMEN.

A,B: Representan las matrices A o G y B o H del sistema de la ec. de estado respectivamente.

N: Representa el número de variables de estado.

R: Representa el número de entradas del sistema.

KREAMULT: Representa la matriz obtenida de la ganancia de realimentación de estado K de [rxn], de sistemas de múltiples-entradas.

FIA: Representa la matríz $\phi(A)$ ó $\phi(G)$.

NUEVACONEST: Representa la matriz de controlabilidad de estado para sitemas de entradas-múltiples, en este caso es una matriz de controlabilidad de estado de un sistema núevo.

BBW: Representa la matriz BW del algoritmo para sistemas de múltiples-entradas.

CONEST: Representa la matriz de controlabilidad de estado de sistemas de única-entrada.

SUMA: Representa una variable temporal.

W: Representa un vector útilizado en el algoritmo para sistemas de múltiples-entradas.

ALFASS: Representa al vector de coeficientes del polinomio del diseño del sistema de la ec. característica de los eigenvalores.

KTESTA: Representa un vector K testada ($\bar{K}$) de el algoritmo del sistema de múltiples-entradas, para obtener la K [rxn].

KREALIM: Representa al vector de ganancia de realimentación de estado para sistemas de única-entrada.

VERD: Representa una variable de control del algoritmo.

PSEUDOCODIGO.

```
PROCEDURE REALIMEN(entrada datos del sistema);
BEGIN
   llamar a CONTROESTA(obteniendo rango es N → verdadero);
   IF (verdadero) THEN
      IF (el sistema es de entrada única) THEN
         (llama a CREAPOLIN(obteniendo el polinomio C.);
         (llama a ECCARACT(obteniendo ¢(A o G) → FIA);
         (llama a KREALIMENT(obteniendo matriz de realim.E. =K);
      ELSE
         llama a BUSCABPORW(obteniendo la matriz BW);
         llama a CONTROESTA(de(A,BW) → verdadero);
         IF (verdadero) THEN
            (llama a CREAPOLIN(obteniendo polinomio C.);
            (llama a ECCARACT(obteniendo ¢(A o G) → FIA);
            (llama a KREALIMENT(obteniendo K̄ );
            (multiplicación de W por K̄ obteniendo → K(escribe K));
         ELSE
            (mensaje no se puede hacer la colocación de polos);
         END if;
      END if;
   ELSE
      (mensaje no se puede hacer la colocación de polos);
   END if;
END REALIMEN;
```

Descripción del módulo OBSERVADOR.

En este módulo se obtiene el diseño del observador completo, obteniendose para esto la matriz de ganancia de realimentación para el observador Ke (K estimada). Tanto para sistemas de única-salida como para sistemas de múltiples-salidas.

Variables del módulo OBSERVADOR.

A: Representa la matriz **A** o **G** del sistema para el cuál queremos diseñar su observador.

B: Representa la matriz B o **H** del sistema para el cuál queremos diseñar su observador.

N: Representa el número de variables de estado.

M: Representa el número de salidas del sistema.

KESTIMMUL: Representa la matriz de ganancia de realimentación del observador, para sistemas de única-salida y de múltiples-salidas.

ATRANP: Representa la matriz A^T .

CTRANP: Representa la matriz transpuesta de la matriz C del sistema

NUEVACONEST: Representa la matriz de controlabilidad de estado de un nuevo sistema obtenido por el algoritmo de múltiples-salidas.

FIA: Representa la matriz $\phi(A)$ o $\phi(G)$.

BBW: Matriz BW o HB del algoritmo de múltiples-entradas.

KREAMULT: Representa la matriz K del sistema Dual.

W: Representa un vector **W** del algoritmo del Sistema multivariable.

ALFASS: Representa al vector de coeficientes del polinomio del diseño del sistema de la ec. característica de los eigenvalores.

KTESTATRAN: Representa la matriz $\bar{K}^*$ del algoritmo para obtener la

Ke para sistemas de múltiples entradas múltiples-salidas.

VERD: Representa una variable de control del algoritmo.

PSEUDOCODIGO.

```
PROCEDURE OBSERVADOR;
BEGIN
   (llama a OBSERVAB(obtiene obsevabilidad verdadera → ver);
   IF (ver) THEN
      IF (sistema de salida única) THEN
         (llama a CREAPOLIN(obtiene el polinomio C.);
         (llama a ECCARACT(obtiene φ(A o G) → FIA);
         (llama a KOBSERV(obtiene K(estimada);(escribe Ke);
      ELSE
         (obtener la transpuesta de las matrices A y de C del S.);
         (llama a BUSCABPORW(obtiene la matriz BW);
         (llama a CONTROESTA(de (A^T,BBW)=ver);
         IF (ver) THEN
            llama a CREAPOLIN(obtiene el polinomio C);
            (llama a ECCARACT(obtiene la matriz φ(A^T) =FIA);
            (llama a KREALIMENT(obtención de K̄^T );
            (multiplicación de W por K̄^T );(obtención de K^T(T)=K);
            (escribir K_e);
         ELSE
            (mensaje no se puede encontrar K_e múlt.salidas);
      END if;
   ELSE
```

(mensaje no puede encontrar K_e);

END if;

END OBSERVADOR;

Descripción del módulo OBSERMINIMO.

En este módulo se da todo el control para el diseño del observador mínimo dado en los capitulos anteriores,como sería la transformación del sistema,obteniendo la matriz K del núevo sistema y la obtención de la matriz de realimentación del observador K_e para el diseño del observador de orden mínimo.

Variables del módulo de OBSERMINIMO.

A,B,C: Representan matrices **A,B,C** o **G,H,C** del sistema de ec. estado.

N: Representa el número de variables de estado.

R,M: Representan el número de entradas,salidas del sistema respect.

RAN: Representa el rango de una matriz.

I,J: Representan contadores de iteraciones.

SEMILLA: Representa el número(semilla) del número pseualeatorio.

CONTADOR: Representa un contador de control del algoritmo.

DET: Representa el determinante de una matriz.

BAND: Representa variable de control(bandera).

P: Representa la matriz P del algoritmo del observador-mínimo.

PINV: Representa la matriz inversa de P del algoritmo.

PA: Representa la matriz P·A de la transformación del sistema.

PAQ: Representa la matriz PAQ del algoritmo del observador mínimo.

PB: Representa la matriz P·B de transformación del sistema.

CQ: Representa la matriz C·Q de transformación del sistema.

ABB: Representa la matriz $\hat{A}_{22}$ del algoritmo del observador mínimo.

AAB: Representa la matriz $\hat{A}_{12}$ del algoritmo del observador mínimo.

PSEUDOCODIGO.

```
PROCEDURE OBSERMINIMO(se pasan datos del sistema);
BEGIN
   (inicialización de banderas,contadores y de v.de control);
   IF (M.salidas es menor a N.variables de estado) THEN
      (verificar si la matriz C esta en la fórma [IM 0]);
      IF (C no esta en la fórma [IM  0]) THEN
         REPEAT    (crear la matriz de transformación  llamando a)
            (PSEUALEA y llamando a INVERSA de la matriz creada);
         UNTIL (encontramos la matriz de transformación+det);
         IF (encontramos la m.de transformación) THEN
            (transformación sistema con la matriz de transform.);
         END if;
         (mensaje de la parte Q1 y Q2);
      ELSE
         (asignaciónes correspondientes a otras variables del s.);
      END if;
      (obtener y asignar las matrices A22 y A12);
      (llamar a REALIMEN(para obtener la K nuevo sistema);
      (llamar a OBSERVADOR(para obtener la Ke nuevo sistema);
   ELSE
```

```
      (mensaje:observador-mínimo no es realizable);
   END if;
END OBSERMINIMO;
```

Descripción del módulo METERDATOS.

En este módulo se controla la obtención de los datos de entrada. Además llamando a procedures para la discretización,y para la obtención de los cálculos de las respuestas libre y forzada.Por último se cálcula la respuesta total de Y(t) o Y(k) del sistema.

PSEUDOCODIGO.

```
PROCEDURE METERDATOS;
BEGIN
   llamar a DATOS(obtener datos del sistema);
   IF (si el sistema es de tiempo-continuo) THEN
      (llamar a EXPA(para obtener la matriz G);
      IF (llamar a DETERM(matriz A) <> cero) THEN
         (llamar a FORMINTEG(cálculo por fórmula de la matriz H);
      ELSE
         (dependiendo del incremento de los puntos de la respuesta
          se escogen determinados puntos para hacer la integración
          númerica para obtener la matriz H);
          (llamar a INTEGEXPB(para obtener la matriz H);
      END if;
```

ELSE

(asignación de las matrices G ← A ; H ← B);

END if;

(llamar a RESPLIBRE y RESFORITER(respuestas libre y forzada);

(obtención de la respuesta total Y(t o k) ← C·X(t) + D·U(t));

END METERDATOS;

Descripción del módulo DISESIST.

En este módulo se obtienen los cálculos para obtener el nuevo sistema diseñado por la realimentación del estado,destruyendo al anterior sistema y dejando el sistema diseñado por realimentación de estado.

Variables del módulo DISESIST.

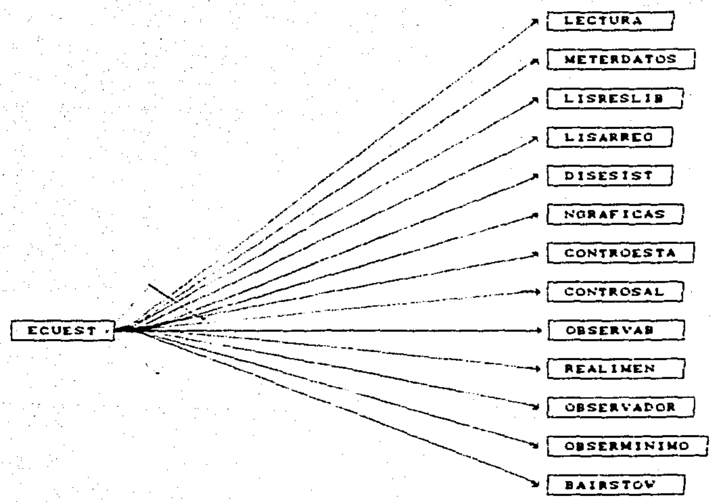
I,J,K:Representan contadores de iteraciones.

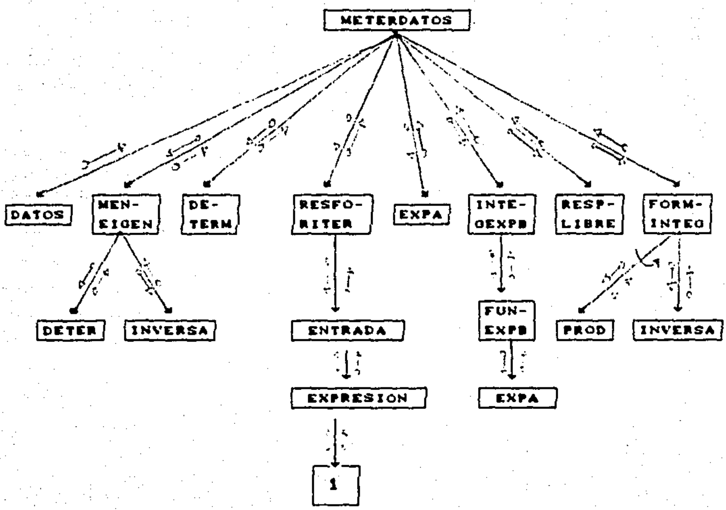
SUMA:Representa una variable temporal.

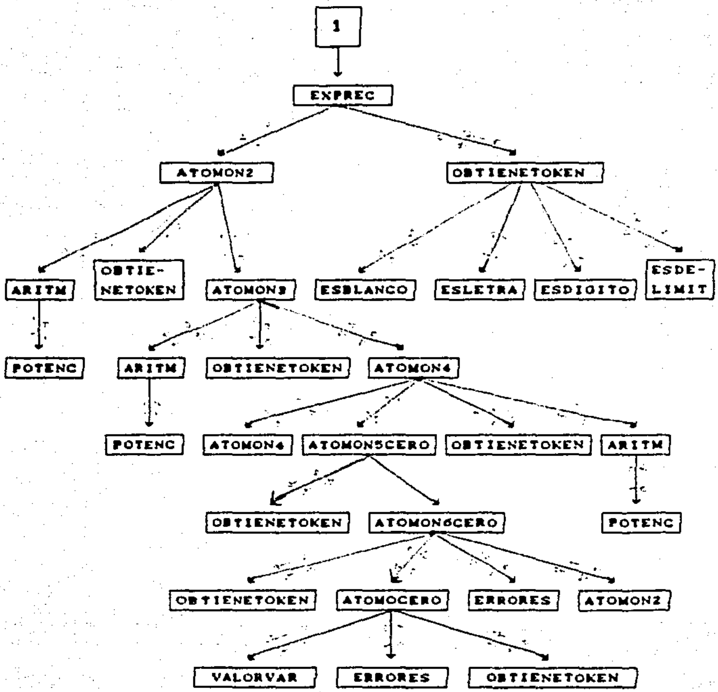
DIAGRAMAS DE FLUJO DE DATOS.

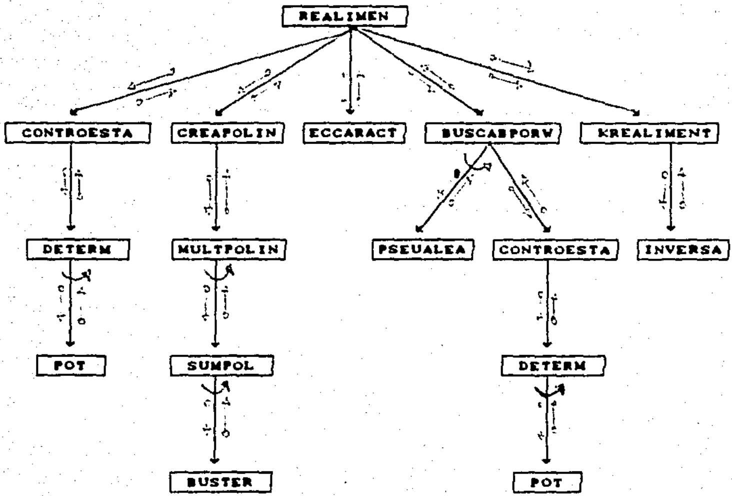
PROGRAMA TESIS ESPACIO DE ESTADOS.

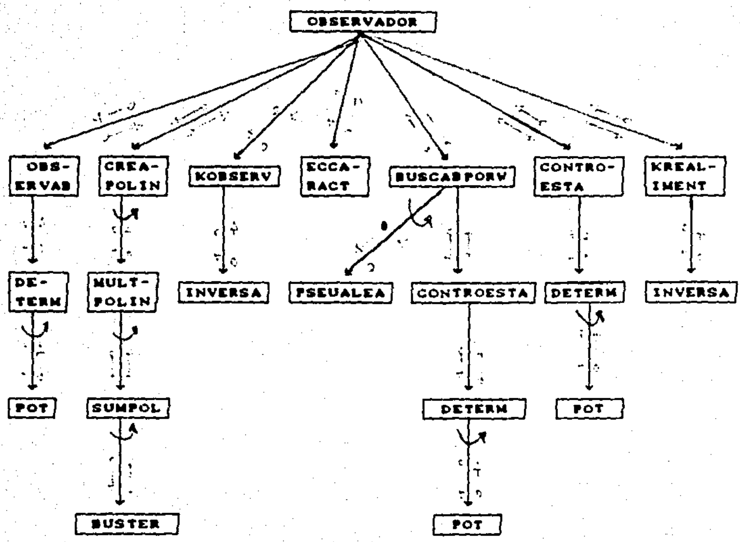
DIAGRAMA DEL MODULO PRINCIPAL DEL PROGRAMA ECES.PAS.

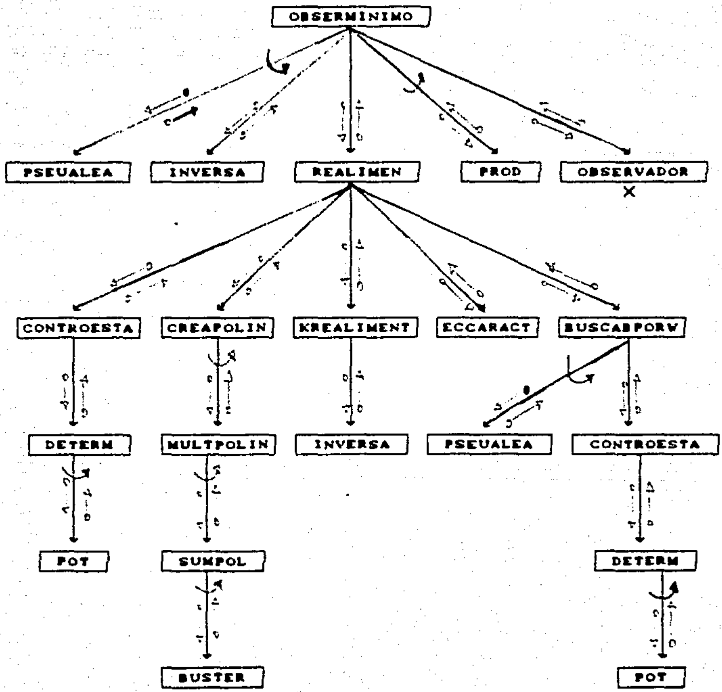


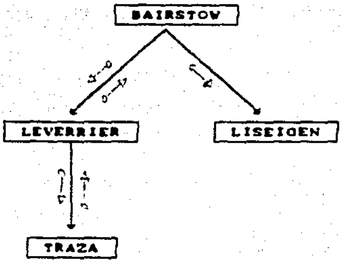

METERDATOS
DATOS
MEN-EIGEN
DE-TERM
RESFO-RITER
EXPA
INTE-GEXPB
RESP-LIBRE
FORM-INTEG
DETER
INVERSA
ENTRADA
FUN-EXPB
PROD
INVERSA
EXPRESION
EXPA
1

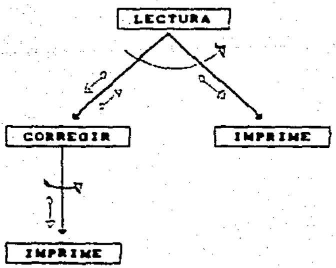

1
EXPREC
ATOMON2
OBTIENETOKEN
ARITM
OBTIE-
NETOKEN
ATOMON3
ESBLANCO
ESLETRA
ESDIGITO
ESDE-
LIMIT
POTENC
ARITM
OBTIENETOKEN
ATOMON4
POTENC
ATOMON4
ATOMON5CERO
OBTIENETOKEN
ARITM
OBTIENETOKEN
ATOMON6CERO
POTENC
OBTIENETOKEN
ATOMOCERO
ERRORES
ATOMON2
VALORVAR
ERRORES
OBTIENETOKEN


REALIMEN
CONTROESTA
CREAPOLIN
ECCARACT
BUSCASPORW
KREALIMENT
DETERM
MULTPOLIN
PSEUALEA
CONTROESTA
INVERSA
POT
SUMPOL
DETERM
BUSTER
POT


OBSERVADOR
OBS-ERVAB
CREA-POLIN
KOBSERV
ECCA-RACT
BUSCABPORV
CONTRO-ESTA
KREAL-IMENT
DE-TERM
MULT-POLIN
INVERSA
PSEUALEA
CONTROESTA
DETERM
INVERSA
POT
SUMPOL
DETERM
POT
BUSTER
POT


OBSERMINIMO
PSEUALEA
INVERSA
REALIMEN
PROD
OBSERVADOR
CONTROESTA
CREAPOLIN
KREALIMENT
ECCARACT
BUSCABPORW
DETERM
MULTPOLIN
INVERSA
PSEUALEA
CONTROESTA
POT
SUMPOL
DETERM
BUSTER
POT


BAIRSTOW
LEVERRIER
LISEIGEN
TRAZA


LECTURA
CORREGIR
IMPRIME
IMPRIME

```
PROGRAM ECUEST(INPUT,OUTPUT,LST);

(****************************************************************************)
(*                                                                          *)
(*                                                                          *)
(*       U .    N .    A .    M .                                           *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*    F A C U L T A D   D E   I N G E N I E R I A                           *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*    I N G E N I E R I A   E N   C O M P U T A C I O N                     *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*    T E S I S :  E S P A C I O   D E   E S T A D O S                      *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*        PROGRAMA HECHO POR : ARTEMIO ROBERTO FLORES RAMOS                 *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*                                                                          *)
(*                                           MODIFICACION:15-OCTUBRE-1989   *)
(****************************************************************************)

CONST
   NN=10;        NNNN=20;
   PMAX=300;
TYPE
   MATRIZ=ARRAY[1..NN,1..NN]OF REAL;
   PUNTOS=ARRAY[1..NN,0..PMAX]OF REAL;
   RES=ARRAY[0..PMAX]OF REAL;
   SUMAS=ARRAY[1..NN]OF REAL;
   CADE=STRING[20];
   ENLACE=^TERMINO;
   TERMINO=RECORD
              COEF:REAL;
              EXP:INTEGER;
              SIGUIENTE:ENLACE;
           END;
VAR
   A,B,C,D,AEXP,G,H,CONEST,KREALIMULT,KESTIMMULT,A1,B1,C1:MATRIZ;
   N,P,K,NUM,R,M,FUNINTG,CASO:INTEGER;
   T,TI,INC,DET,DET2,SUMA,EIGMIN:REAL;
```

```
      X,AREA,SOLY:PUNTOS;
      PY:RES;
      TIPOSIS,PAUSA,TIP,SI:CHAR;
      CADENN:ARRAY[1..NN]OF CADE;
      XI,UU,RAICESR,RAICESI,KESTIMADA:SUMAS;
      VERD,BAN,PRIERRENT,PRIMPAS,PERR:BOOLEAN;

 PROCEDURE IMPRIME(MMATRIZ:CHAR;N,R,M:BYTE);
   VAR    I,J:INTEGER;
  BEGIN
      CASE MMATRIZ OF
         'A','a','G','g':BEGIN     WRITELN('  MATRIZ [A]  ');
                              FOR I:=1 TO N  DO      BEGIN
                                    FOR J:=1 TO N DO WRITE(A[I,J],' ');WRITELN;
                              END;
                         END;
         'B','b','H','h':BEGIN      WRITELN('  MATRIZ [B]   ');
                              FOR I:=1 TO N DO    BEGIN
                                 FOR J:=1 TO R DO WRITE(B[I,J],' ');WRITELN;
                              END;
                           END;
         'C','c':BEGIN   WRITELN('  MATRIZ [C]  ');
                     FOR I:=1 TO M DO    BEGIN
                        FOR J:=1 TO N DO WRITE(C[I,J],' '); WRITELN;
                        END;
                 END;
         'D','d':BEGIN   WRITELN('   MATRIZ [D]  ');
                    FOR I:=1 TO M DO    BEGIN
                          FOR J:=1 TO R DO  WRITE(D[I,J],' '); WRITELN;
                       END;
                 END;
          ELSE  WRITELN(' NO ENCONTRADA LA MATRIZ ');
   END;
 END;

  PROCEDURE LISEIGEN(RAICESR,RAICESI:SUMAS);
  VAR    I:INTEGER;
  BEGIN
     GOTOXY(5,15);WRITELN(' LOS EIGENVALORES DEL SISTEMA SON : ');WRITELN;
     FOR I:=1 TO N DO
        WRITELN(' RAIZ(real) ',RAICESR[I],' RAIZ(imag) ',RAICESI[I]);
  END;

PROCEDURE LISARREG(RESPXTOTAL:PUNTOS;N,PUNMAX:INTEGER;TTT:BYTE);
VAR    I,J:INTEGER;
BEGIN
   IF TTT=1 THEN WRITELN('  LIS. RESPUESTA TOTAL ------>  X ( t ) :   ')
   ELSE WRITELN('        LIS. RESPUESTA TOTAL ------>  Y ( t ) :   ');
   FOR I:=0 TO PUNMAX  DO    BEGIN
```

```
        FOR J:=1 TO N DO     WRITE(RESPXTOTAL[J,I],' ');    WRITELN
   END; (* for *)            READ(PAUSA);
END;

  PROCEDURE LISRESLIB(XLIB:PUNTOS;N,PUNMAX:INTEGER);
  VAR   I,J:INTEGER;
  BEGIN
     WRITELN('  LIS. RESPUESTA LIBRE  ------>  X ( t ) :    ');
     FOR I:=0 TO PUNMAX DO   BEGIN
          FOR J:=1 TO N DO    WRITE(XLIB[J,I],' ');   WRITELN;
     END;  (* for *)                  READ(PAUSA);
  END;
  PROCEDURE VERMATRIZ;
  VAR   SI:CHAR;
 BEGIN
    REPEAT  WRITELN('VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]    ');
       REPEAT  DELLINE;READ(SI);
          UNTIL  SI IN ['A','B','C','D','a','b','c','d','N','n'];WRITELN;
       IF NOT((SI='N') OR (SI='n'))   THEN   IMPRIME(SI,N,R,M)
    UNTIL SI IN ['N','n'];
 END;
 PROCEDURE CORREGIR(TIPO:CHAR;N,R,M:INTEGER);
   VAR
     CCO,RRE:INTEGER; ELEM:REAL;  SI:CHAR;  BAN:BOOLEAN;
 BEGIN
    REPEAT
       {$I-}   RRE:=0;   CCO:=0;
       WRITELN('DAR LAS COORDENADAS DEL ELEMENTO A CAMBIAR  ');
       REPEAT
          DELLINE;  WRITE(' RENGLON = '); READ(RRE); BAN:=(IORESULT=0);
       UNTIL BAN AND (RRE>0);
       WRITELN;
       REPEAT
          DELLINE;WRITE('  COLUMNA = ');READ(CCO); BAN:=(IORESULT=0);
       UNTIL BAN AND (CCO>0);
       {$I+}      WRITELN;
       CASE TIPO OF
          'A','a':IF (RRE<=N) AND  (CCO<=N) THEN
                   BEGIN
                    {$I-}  TIPO:='A';
                    REPEAT
                       DELLINE;  WRITELN;
                       WRITE('   VALOR DE  ',TIPO,'[',RRE,',',CCO,'] =   ');
                       READ(A[RRE,CCO]); BAN:=(IORESULT=0);
                    UNTIL BAN;
                    {$I+}    WRITELN;
                   END
                   ELSE
                      WRITELN('   NO EXISTE ESE ELEMENTO EN LA MATRIZ ',TIPO);
```

```
           'B','b':IF  (RRE<=N) AND  (CCO<=R) THEN
                   BEGIN
                    {$I-}          TIPO:='B';
                    REPEAT
                       DELLINE;   WRITELN;
                       WRITE('  VALOR DE  ',TIPO,'[',RRE,',',CCO,'] =     ');
                       READ(B[RRE,CCO]); BAN:=(IORESULT=0);
                    UNTIL BAN;
                    {$I+}     WRITELN;
                   END
                   ELSE
                      WRITELN('   NO EXISTE ESE ELEMENTO EN LA MATRIZ  ',TIPO);
           'C','c':IF  (RRE<=M) AND  (CCO<=N) THEN
                   BEGIN
                    {$I-}          TIPO:='C';
                    REPEAT
                       DELLINE;   WRITELN;
                       WRITE('  VALOR DE  ',TIPO,'[',RRE,',',CCO,'] =     ');
                       READ(C[RRE,CCO]); BAN:=(IORESULT=0);
                    UNTIL BAN;
                    {$I+}      WRITELN;
                   END
                   ELSE
                      WRITELN('   NO EXISTE ESE ELEMENTO EN LA MATRIZ ',TIPO);
           'D','d':IF  (RRE<=M) AND  (CCO<=R)  THEN
                   BEGIN
                    {$I-}         TIPO:='D';
                    REPEAT
                       DELLINE;   WRITELN;
                       WRITE('  VALOR DE ',TIPO,'[',RRE,',',CCO,'] =     ');
                       READ(D[RRE,CCO]); BAN:=(IORESULT=0);
                    UNTIL BAN;
                    {$I+}         WRITELN;
                   END
                   ELSE
                     WRITELN('   NO EXISTE ESE ELEMENTO EN LA MATRIZ  ',TIPO);
         ELSE WRITE('  NO SE ENCUENTRA ESA MATRIZ  ');
      END;
      VERMATRIZ;
      WRITELN('CORREGIR ALGUNA OTRA MATRIZ [A],[B],[C],[D],[N--NO]    ');
      REPEAT  DELLINE;   READ(TIPO);
      UNTIL  TIPO IN ['A','B','C','D','a','b','c','d','N','n'];WRITELN;
   UNTIL  (TIPO='N') OR (TIPO='n');
END;

PROCEDURE GRAF(DATOS:RES:NUM,XMAX:INTEGER;TIPO:BYTE);
   VAR
      X,Y,T,INCR,IA,XXX,TX,RESS,TA:INTEGER;
      A,RANGO,NORM,Z,YMIN,YMAX:REAL;    B:BYTE;
```

```
BEGIN
   YMIN:=DATOS[NUM];      YMAX:=DATOS[NUM]:
   FOR K:=NUM TO XMAX DO
      BEGIN
         IF DATOS[K] < YMIN   THEN    YMIN:=DATOS[K];
         IF DATOS[K] > YMAX   THEN    YMAX:=DATOS[K];
      END;
   B:=0; GRAPHMODE: PALETTE(1): GOTOXY(1,23);    WRITE(YMIN:4:3);
   GOTOXY(1,1):      WRITE(YMAX:4:3);
   GOTOXY(1,12);    WRITE((YMAX+YMIN)/2:4:3);
   GOTOXY(1,25):    WRITE(NUM);
   GOTOXY(37,25):   WRITE(XMAX):
   GOTOXY(17.25):   WRITE(TRUNC(XMAX+NUM)/2:4:0);
   IF YMIN > 0   THEN    YMIN:=0:
   RANGO:=YMAX-YMIN;
   IF RANGO = 0 THEN RANGO:=1:  NORM:=190 / RANGO ;
   FOR T:=5 TO 30 DO PLOT(T*10,0.1);
   FOR T:=5 TO 30 DO PLOT(T*10,95,1):
   FOR T:=1 TO 19 DO PLOT(150,T*10,1):
   FOR T:=1 TO 19 DO PLOT(315,T*10,1);
   DRAW(15,0,15,190,3);
   DRAW(0,190,300,190,3);     XXX:=XMAX;      TX:=0;   XMAX:=XMAX-NUM;
   NUM:=NUM+1;   TA:=NUM:
   IF XMAX > 300    THEN XXX:=NUM+299:
   FOR T:=NUM TO XXX   DO
      BEGIN
         IF XMAX<=300    THEN
            BEGIN
               A:=DATOS[T]-YMIN;RESS:=300:
            END;
         IF (XMAX > 300) AND (XMAX <= 600)    THEN
            BEGIN
               TA:=TA+2: A:=DATOS[TA]-YMIN:    RESS:=600;
            END:
         IF (XMAX>600) AND (XMAX<=900)    THEN
            BEGIN
               TA:=TA+3: A:=DATOS[TA]-YMIN:    RESS:=900:
            END;
         IF A < 1.0E-20    THEN     A:=0;
         IF A > 1.0E+20    THEN     A:=0:
         Z:=A*NORM;       A:=ABS(Z);
         IF A < -32766.0    THEN     A:=-325.7:
         IF A > 32765.0     THEN     A:=320.8:
         Y:=TRUNC(A);
         INCR:=RESS DIV XMAX ;
         TX:=TX+1;   X:=((TX-1)*INCR+15);
         PLOT(X,190-Y,2);
      END;
   END;
```

```
PROCEDURE EXPA(VAR EXP:MATRIZ;A:MATRIZ;T:REAL;N,NUM:INTEGER);
CONST
   EEP=1E-30;
VAR
   I,J,K,KK:INTEGER;
   SUMA,RFAC,AMAX:REAL;
   IDEN,AMULT,APORA:MATRIZ;
BEGIN
   FOR I:=1 TO N DO
      FOR J:=1 TO N DO    IDEN[I,J]:=0.0;
   FOR I:=1 TO N DO    IDEN[I,I]:=1;
   RFAC:=1 ;
   FOR I:=1 TO N DO
      FOR J:=1 TO N DO
         BEGIN
            AMULT[I,J]:=IDEN[I,J];
            EXP[I,J]:=IDEN[I,J];
         END;
   KK:=1;      AMAX:=1;
   WHILE (ABS(AMAX) < 1E35)  AND (T <> 0) AND (KK < (NUM+20))   DO
      BEGIN
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO
               BEGIN
                  SUMA:=0;
                  FOR K:=1 TO N DO
                     SUMA:=SUMA+AMULT[I,K]*A[K,J];
                  APORA[I,J]:=SUMA;
               END;
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO
               AMULT[I,J]:=APORA[I,J];
         RFAC:=(RFAC*T)/KK;
         AMAX:=APORA[1,1];
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO
               IF ABS(APORA[I,J]) > ABS(AMAX)   THEN
                  AMAX:=APORA[I,J];
         KK:=KK+1;
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO
               EXP[I,J]:=EXP[I,J]+(AMULT[I,J]*RFAC);
      END;
END;

  PROCEDURE NGRAFICAS(FUN:PUNTOS;L,NOM:INTEGER;TIPO:BYTE);
  VAR
```

```
      I,J:INTEGER;
   BEGIN
      FOR I:=1 TO NOM DO
         BEGIN
            FOR J:=0 TO P DO   PY[J]:=FUN[I,J];
            CLRSCR;    GRAF(PY,L,P,1);    READ(PAUSA);
         END;
   END;

PROCEDURE LECTURA;
   VAR
      I,K,J:BYTE;     BAN:BOOLEAN;    SI:CHAR;
BEGIN
   CLRSCR; N:=0; R:=0; M:=0;
   FOR I:=1 TO NN DO
      FOR J:=1 TO NN DO
         BEGIN
            A[I,J]:=99999E5; B[I,J]:=0; C[I,J]:=0; D[I,J]:=0;
         END;
   {$I-}
   REPEAT
      GOTOXY(6,3); CLREOL;
      WRITELN('ESPACIO DE ESTADOS : SISTEMAS MULTIVARIABLES');GOTOXY(3,5);
      WRITE('N:NUMERO DE VARIABLES DE ESTADO DEL SISTEMA = ');READLN(N);
      BAN:=(IORESULT=0);
   UNTIL BAN AND (N>0)  AND (N<=NN);
   REPEAT
      GOTOXY(3,7); CLREOL;
      WRITE('R:NUMERO DE ENTRADAS DEL SISTEMA  = ');READLN(R);
      BAN:=(IORESULT=0);
   UNTIL BAN AND (R>0) AND (R<=NN) ;
   REPEAT
      GOTOXY(3,9); CLREOL;
      WRITE('M:NUMERO DE SALIDAS DEL SISTEMA =  ');READLN(M);
      BAN:=(IORESULT=0);
   UNTIL BAN AND (M>0) AND (M<=NN) ;
   REPEAT   GOTOXY(3,11);;CLREOL; TIPOSIS:='C';
      WRITE('  SISTEMA T. CONTINUO=C o T.DISCRETO=D   ');READLN(TIPOSIS);
   UNTIL  TIPOSIS IN ['C','c','D','d'];
   WRITELN;  WRITELN('DAR LA MATRIZ   [A]  ');
   FOR I:=1 TO N DO
      BEGIN
         FOR K:=1 TO N DO
            BEGIN
               REPEAT
                  WRITE('  A[',I,',',K,'] = ');READ(A[I,K]);
                  BAN:=(IORESULT=0);
               UNTIL BAN AND (A[I,K]<>99999E5);
            END;
```

```
      WRITELN;
    END;
 WRITELN;
 WRITELN('DAR LA MATRIZ   [B] ');
 FOR I:=1 TO N  DO
    BEGIN
       FOR K:=1 TO R DO
          BEGIN
             REPEAT
                WRITE('  B[',I,',',K,'] =  ');READ(B[I,K]);
                BAN:=(IORESULT=0);
             UNTIL BAN;
          END;
       WRITELN;
    END;
 WRITELN;
 WRITELN('DAR LA MATRIZ  [C]' );
 FOR I:=1  TO M DO
    BEGIN
       FOR K:=1 TO N DO
          BEGIN
             REPEAT
                WRITE('  C[',I,',',K,']  =  ');READ(C[I,K]);
                BAN:=(IORESULT=0);
             UNTIL  BAN;
          END;
       WRITELN;
    END;
 WRITELN;  WRITELN('DAR LA MATRIZ   [D]  ');
 FOR I:=1  TO M DO
    BEGIN
       FOR K:=1 TO R DO
          BEGIN
             REPEAT
                WRITE('  D[',I,',',K,']  =  ');READ(D[I,K]);
                BAN:=(IORESULT=0);
             UNTIL BAN;
          END;
       WRITELN;
    END;
 {$I+}
 WRITELN('CORREGIR ALGUNA MATRIZ [A],[B],[C],[D],[N--No]   ');
 REPEAT DELLINE;  READ(SI);
 UNTIL   SI IN ['A','B','C','D','a','b','c','d','N','n'];WRITELN;
 IF NOT((SI='N')OR(SI='n'))    THEN
    CORREGIR(SI,N,R,M);
 VERMATRIZ;
END;
```

```
PROCEDURE LEVERRIER(A:MATRIZ;N:INTEGER;VAR DS:SUMAS);
VAR
   I,J,K,II:INTEGER;   TRA,DK:REAL;
   ABK,IDEN,BB:MATRIZ;   DDDS:SUMAS;
PROCEDURE TRAZA(A,B1:MATRIZ;VAR TR:REAL;VAR AB:MATRIZ);
VAR
   II,JJ,KK:INTEGER;      SUMA:REAL;
BEGIN
   FOR II:=1 TO N DO
      FOR JJ:=1 TO N DO
         BEGIN
            SUMA:=0;
            FOR KK:=1 TO N DO     SUMA:=SUMA+A[II,KK]*B1[KK,JJ];
            AB[II,JJ]:=SUMA;
         END;
   TR:=0;
   FOR II:=1 TO N DO  TR:=TR+AB[II,II];
END;
BEGIN
   FOR I:=1 TO N DO
      FOR J:=1 TO N DO
         BEGIN
            BB[I,J]:=0;     IDEN[I,J]:=0;
         END;
   FOR I:=1 TO N DO
      BEGIN
         BB[I,I]:=1;     IDEN[I,I]:=1;
      END;
   FOR II:=1 TO N DO
      BEGIN
         K:=II;    TRAZA(A,BB,TRA,ABK);
         DK:=TRA/K;     DS[K]:=DK;
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO
               BB[I,J]:=ABK[I,J]-DK*IDEN[I,J];
      END;
END;

FUNCTION DETERM(AD:MATRIZ;ORDEN:INTEGER):REAL;
 VAR
    I,J,KK:INTEGER;   SUMA:REAL;  MENOR:MATRIZ;  BAN:BOOLEAN;
 FUNCTION POT(JJ:INTEGER):REAL;
 VAR
    III:INTEGER;
    TR,T:REAL;
 BEGIN
    T:=-1;   TR:=-1;
    FOR III:=JJ  DOWNTO 2 DO
       TR:=TR*T;
```

```
    POT:=TR;
 END;
  BEGIN
     IF ORDEN=1 THEN
        DETERM:=AD[1,1];
     IF ORDEN >= 2 THEN
        BEGIN
           SUMA:=0;
           FOR J:=1 TO ORDEN DO
              BEGIN
                 FOR I:=1 TO (ORDEN-1) DO
                    BEGIN
                       BAN:=FALSE;
                       FOR K:=1 TO ORDEN-1 DO
                          BEGIN
                             KK:=K;
                             IF (K=J) OR (BAN=TRUE)  THEN
                                BEGIN
                                   BAN:=TRUE;
                                   KK:=K+1;
                                END;
                             MENOR[I,K]:=AD[I+1,KK];
                          END;
                    END;
                 SUMA:=SUMA+POT(1+J)*AD[1,J]*DETERM(MENOR,ORDEN-1);
              END;
           DETERM:=SUMA;
        END;
  END;

PROCEDURE CONTROESTA(A,B:MATRIZ;N,R:INTEGER;VAR CCESTA:MATRIZ;VAR V:BOOLEAN;
                                                                 TIPO:BYTE);
 VAR
    APOT,AAPOT.APOTB:MATRIZ;
    COESTA.COESTAT:ARRAY[1..NNNN,1..NNNN]OF REAL;
    TERM,I,J,K,L,II,RAN:INTEGER;  SUMA:REAL;
 BEGIN
    FOR I:=1 TO N DO
       FOR J:=1 TO N DO    APOT[I,J]:=0;
    FOR I:=1 TO N DO    APOT[I,I]:=1;
    FOR I:=1 TO N DO
       FOR J:=1 TO R DO  COESTA[I,J]:=B[I.J]:
    TERM:=R;
    FOR II:=1 TO (N-1)  DO
       BEGIN
          FOR I:=1 TO N DO
             FOR J:=1 TO N DO
                BEGIN
                   SUMA:=0;
```

```
                FOR K:=1 TO N DO
                    SUMA:=SUMA+APOT[I,K]*A[K,J];
                AAPOT[I,J]:=SUMA;
            END;
        FOR I:=1 TO N DO
            FOR J:=1 TO N DO
                APOT[I,J]:=AAPOT[I,J];
        FOR I:=1 TO N DO
            FOR J:=1 TO R DO
                BEGIN
                    SUMA:=0;
                    FOR K:=1 TO N DO
                        SUMA:=SUMA+APOT[I,K]*B[K,J];
                    APOTB[I,J]:=SUMA;
                END;
        FOR I:=1 TO N DO
            FOR J:=1 TO R  DO
                BEGIN
                    K:=TERM+J;    COESTA[I,K]:=APOTB[I,J];
                END;
        TERM:=TERM+R;
    END;
FOR I:=1 TO N DO
    FOR J:=1 TO (N*R)  DO
        COESTAT[J,I]:=COESTA[I,J];
FOR I:=1 TO N DO
    FOR J:=1 TO N DO
        CCESTA[I,J]:=COESTA[I,J];
FOR I:=1 TO N DO
    FOR J:=1 TO N DO
        BEGIN
            SUMA:=0;
            FOR K:=1 TO (N*R) DO
                SUMA:=SUMA+COESTA[I,K]*COESTAT[K,J];
            APOTB[I,J]:=SUMA;
        END;            V:=FALSE;
IF (DETERM(APOTB,N) <> 0)    THEN
    BEGIN
        IF TIPO = 1 THEN     BEGIN   GOTOXY(5,15);
          WRITELN('  EL SISTEMA ES CONTROLABLE DE ESTADO ');READ(PAUSA);
        END;
        V:=TRUE;
    END
ELSE
    IF TIPO = 1 THEN
        BEGIN   GOTOXY(5,15);
            WRITELN('    EL SISTEMA NO ES CONTROLABLE DE ESTADO   ');
            WRITELN('                  EL RANGO < ',N);    READ(PAUSA);
        END;
```

```
END;

PROCEDURE CONTROSAL(A,B,C,D:MATRIZ;N,R,M:INTEGER);
 VAR
    I,J,K,II,TERM,RAN:INTEGER;     SUMA:REAL;
    CONSA,CONSAT:ARRAY[1..NNNN,1..NNNN]OF REAL;       CAAA:MATRIZ;
 BEGIN
    FOR I:=1 TO M DO
        FOR J:=1 TO R DO
            BEGIN
                SUMA:=0;
                FOR K:=1 TO N DO
                    SUMA:=SUMA+C[I,K]*B[K,J];
                CONSA[I,J]:=SUMA;
            END;
    TERM:=R;
    FOR II:=1 TO (N-1) DO
        BEGIN
            FOR I:=1 TO  M DO
                FOR J:=1 TO N DO
                    BEGIN
                        SUMA:=0;
                        FOR K:=1 TO N DO
                            SUMA:=SUMA+C[I,K]*A[K,J];
                        CAAA[I,J]:=SUMA;
                    END;
            FOR I:=1 TO M DO
                FOR J:=1 TO N DO
                    C[I,J]:=CAAA[I,J];
            FOR I:=1 TO M DO
                FOR J:=1 TO R DO
                    BEGIN
                        SUMA:=0;
                        FOR K:=1 TO N DO
                            SUMA:=SUMA+C[I,K]*B[K,J];
                        CAAA[I,J]:=SUMA;
                    END;
            FOR I:=1 TO M DO
                FOR J:=1 TO R DO
                    BEGIN
                        K:=TERM+J;  CONSA[I,K]:=CAAA[I,J];
                    END;
            TERM:=TERM+R;
        END;
        FOR I:=1 TO M DO
            FOR J:=1 TO R DO
                BEGIN
                    K:=TERM+J;  CONSA[I,K]:=D[I,J];
                END;
```

```
      FOR I:=1 TO M DO
         FOR J:=1 TO (N+1)*R DO
            CONSAT[J,I]:=CONSA[I,J];
      FOR I:=1 TO M DO
         FOR J:=1 TO M DO
            BEGIN
               SUMA:=0;
               FOR K:=1 TO  ((N+1)*R)    DO
                  SUMA:=SUMA+CONSA[I,K]*CONSAT[K,J];
               CAAA[I,J]:=SUMA;
            END;GOTOXY(5,15);
      IF (DETERM(CAAA,M) <> 0) THEN
         WRITELN('       EL SISTEMA ES CONTROLABLE DE LA SALIDA ')
      ELSE
         WRITELN('       EL SISTEMA NO ES CONTROLABLE DE LA SALIDA ');
      READ(PAUSA);
END;

PROCEDURE OBSERVAB(A,AT:MATRIZ;N,M:INTEGER;VAR V:BOOLEAN;TIPO:BYTE);
 VAR
     I,J,K,II,TERM,RAN:INTEGER;   SUMA:REAL;    CT,AAC:MATRIZ;
     OBS,OBST:ARRAY[1..NNNN,1..NNNN]OF REAL;
BEGIN
   FOR I:=1 TO M DO
      FOR J:=1 TO N DO      CT[J,I]:=AT[I,J];
   FOR I:=1 TO N DO
      FOR J:=1 TO N DO      AT[J,I]:=A[I,J];
   FOR I:=1 TO N DO
      FOR J:=1 TO M DO      OBS[I,J]:=CT[I,J];
   FOR I:=1 TO N DO
      FOR J:=1 TO N DO      A[I,J]:=0;
   FOR I:=1 TO N DO         A[I,I]:=1;  TERM:=M;
   FOR II:=1 TO N-1 DO
      BEGIN
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO
               BEGIN
                  SUMA:=0;
                  FOR K:=1 TO N DO
                     SUMA:=SUMA+A[I,K]*AT[K,J];
                  AAC[I,J]:=SUMA;
               END;
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO  A[I,J]:=AAC[I,J];
         FOR I:=1 TO N DO
            FOR J:=1 TO M DO
               BEGIN
                  SUMA:=0;
                  FOR K:=1 TO N DO
```

```
                    SUMA:=SUMA+A[I.K]*C[K.J];
                  AAC[I,J]:=SUMA;
                END;
            FOR I:=1 TO N DO
              FOR J:=1 TO M DO
                BEGIN
                  K:=TERM+J;
                  OBS[I.K]:=AAC[I.J];
                END;
            TERM:=TERM+M;
        END;
        FOR I:=1 TO N DO
          FOR J:=1 TO (M*N) DO
            OBST[J.I]:=OBS[I.J];
        FOR I:=1 TO N DO
          FOR J:=1 TO N DO
            BEGIN
              SUMA:=0;
              FOR K:=1 TO M*N DO
                SUMA:=SUMA+OBS[I,K]*OBST[K.J];
              AAC[I,J]:=SUMA;
            END; V:=FALSE;
        IF (DETERM(AAC,N) <> 0)    THEN
          BEGIN
            IF TIPO = 1 THEN          BEGIN   GOTOXY(5.15);
             WRITELN('     EL SISTEMA ES OBSERVABLE '):READ(PAUSA);
            END;
            V:=TRUE;
          END
        ELSE
          IF TIPO = 1 THEN
            BEGIN  GOTOXY(5.15):
              WRITELN('         EL SISTEMA NO ES OBSERVABLE ');
              WRITELN('                  EL RANGO .  ',N):READ(PAUSA);
            END;
      END;

    FUNCTION ESLETRA(CARACT:CHAR):BOOLEAN;
      BEGIN
        ESLETRA:=(UPCASE(CARACT)>='A')AND(UPCASE(CARACT)<='Z');
      END;

    FUNCTION ESDIGITO(C:CHAR):BOOLEAN;
      BEGIN
        ESDIGITO:=(C>='0') AND (C<='9');
      END;
    FUNCTION ESDELIMIT(C:CHAR):BOOLEAN;
      BEGIN
        IF POS(C,'+-/*%^=()$') <> 0    THEN
```

```
          ESDELIMIT:=TRUE
       ELSE
          ESDELIMIT:=FALSE;
    END;

FUNCTION ESBLANCO(C:CHAR):BOOLEAN;
    BEGIN
       ESBLANCO:=(C=' ')OR(C=CHR(9))OR(C=CHR(13));
    END;

PROCEDURE EXPRESION(CADENA:CADE;TIEMPO:REAL;VAR RESULT:REAL);
TYPE
    TIPO=(NUMERO,DELIMITADOR,VARIABLE);
VAR
    TOKEN:CADE;    TIPOTOKEN:TIPO;   TT,CCC:INTEGER;
PROCEDURE OBTIENETOKEN;
BEGIN
    TOKEN:='';
    WHILE (ESBLANCO(CADENA[TT]))  DO
       TT:=TT+1;
    IF CADENA[TT]='#'    THEN
       TOKEN:='$';
    IF POS(CADENA[TT],'+-*/% =()') <> 0    THEN
       BEGIN
          TIPOTOKEN:=DELIMITADOR;
          TOKEN:=CADENA[TT];
          TT:=TT+1;
       END
    ELSE
       IF POS(CADENA[TT],'SCE') <> 0    THEN
          BEGIN
             REPEAT
                TIPOTOKEN:=DELIMITADOR;
                TOKEN:=CONCAT(TOKEN,CADENA[TT]);
                TT:=TT+1;
             UNTIL (NOT(POS(CADENA[TT],'ENOSXP') <> 0))
          END
       ELSE
          IF ESLETRA(CADENA[TT])    THEN
             BEGIN
                WHILE (NOT ESDELIMIT(CADENA[TT]))  DO
                   BEGIN
                      TOKEN:=CONCAT(TOKEN,CADENA[TT]); TT:=TT+1;
                   END;
                TIPOTOKEN:=VARIABLE;
             END
          ELSE
             IF ESDIGITO(CADENA[TT])    THEN
                BEGIN
```

```
                    WHILE (NOT ESDELIMIT(CADENA[TT]))    DO
                       BEGIN
                          TOKEN:=CONCAT(TOKEN,CADENA[TT]);  TT:=TT+1;
                          TIPOTOKEN:=NUMERO;
                       END;
                 END;
   END;
PROCEDURE ERRORES(I:INTEGER);
   BEGIN
      IF PERR THEN
         CASE I OF
            1:WRITELN('                    ERROR DE SINTAXIS');
            2:WRITELN('          PARENTESIS NO EQUILIBRADOS');
         END;
      P:=0;    PERR:=FALSE;
   END;

FUNCTION POTENC(A,B:REAL):REAL;
 BEGIN
    IF B=0   THEN    POTENC:=1
             ELSE    POTENC:=EXP(B*LN(A));
 END;

FUNCTION VALORVAR(C:CADE):REAL;
   BEGIN
      VALORVAR:=TIEMPO;
   END;

PROCEDURE ARITM(OPER:CHAR;VAR RESULT,OPERA:REAL);
   BEGIN
      CASE OPER OF
          '+':RESULT:=RESULT+OPERA;
          '-':RESULT:=RESULT-OPERA;
          '/':RESULT:=RESULT/OPERA;
          '^':RESULT:=POTENC(RESULT,OPERA);
      END;
   END;

PROCEDURE ATOMON2(VAR RESULT:REAL);FORWARD;
PROCEDURE ATOMON3(VAR RESULT:REAL);FORWARD;
PROCEDURE ATOMON4(VAR RESULT:REAL);FORWARD;
PROCEDURE ATOMON5CERO(VAR RESULT:REAL);FORWARD;
PROCEDURE ATOMON6CERO(VAR RESULT:REAL);FORWARD;
PROCEDURE ATOMOCERO(VAR RESULT:REAL);FORWARD;

PROCEDURE EXPREC(VAR RESULT:REAL);
   BEGIN
      OBTIENETOKEN;
      IF LENGTH(TOKEN) <> 0  THEN
```

```
        ATOMON2(RESULT)
     ELSE
        RESULT:=0;
  END;

PROCEDURE ATOMON2;
VAR
  OP:CHAR;    GUARDA:REAL;
BEGIN
  ATOMON3(RESULT);
  OP:=TOKEN[1];
  WHILE ((OP='+') OR (OP='-'))    DO
     BEGIN                 (* OPERACION SUMA Y RESTA      *)
        OBTIENETOKEN;  ATOMON3(GUARDA);
        ARITM(OP,RESULT,GUARDA);  OP:=TOKEN[1];
     END;
END;

PROCEDURE ATOMON3;
VAR
  OP:CHAR;        GUARDA:REAL;
BEGIN
  ATOMON4(RESULT);          (* OPERACION MULT  Y DIV  *)
  OP:=TOKEN[1];
  WHILE ((OP='*') OR (OP='/'))    DO
    BEGIN
      OBTIENETOKEN;ATOMON4(GUARDA);ARITM(OP,RESULT,GUARDA);OP:=TOKEN[1];
    END;
END;

PROCEDURE ATOMON4;
VAR
  GUARDA:REAL;
BEGIN
  ATOMON5CERO(RESULT);
  IF TOKEN[1]='^'    THEN                (* OPERACION POTENCIACION   *)
     BEGIN
        OBTIENETOKEN;     ATOMON4(GUARDA);
        ARITM('^',RESULT,GUARDA);
     END;
END;

PROCEDURE ATOMON5CERO;
  VAR
     OP:CHAR;
BEGIN
  OP:=' ';                  (*  FUNCION UNITARIA    *)
  IF ((TIPOTOKEN=DELIMITADOR)AND((TOKEN[1]='+')OR(TOKEN[1]='-')OR
      (TOKEN[1]='S')OR(TOKEN[1]='C')OR(TOKEN[1]='E')))    THEN
```

```
         BEGIN
            OP:=TOKEN[1]:
            OBTIENETOKEN;
         END:
    ATOMON6CERO(RESULT);
    CASE OP OF       (*  ENTRA SI ES FUNCION UNITARIA    *)
       '-':RESULT:=-RESULT;
       'S':RESULT:=SIN(RESULT):
       'C':RESULT:=COS(RESULT):
       'E':RESULT:=EXP(RESULT):
    END;
END;

PROCEDURE ATOMON6CERO:
    BEGIN              (*   EMPIEZA OTRO MODULO    *)
       IF (TOKEN[1]='(')AND(TIPOTOKEN=DELIMITADOR)   THEN
          BEGIN
             OBTIENETOKEN:
             ATOMON2(RESULT);
             IF TOKEN[1] <> ')'    THEN
                ERRORES(2):
             OBTIENETOKEN;
          END
       ELSE
          ATOMOCERO(RESULT):
    END:

PROCEDURE ATOMOCERO:
    BEGIN
       IF TIPOTOKEN=NUMERO   THEN
          VAL(TOKEN.RESULT.CCC)    (* OBTIENE EL NUMERO REAL  *)
       ELSE
          IF TIPOTOKEN = VARIABLE   THEN
             RESULT:=VALORVAR(TOKEN)   (* PONE EL VALOR QUE TOMA EL TIEMPO*)
          ELSE                         (*   EN ESE TIEMPO     *)
             ERRORES(1):
       OBTIENETOKEN:
    END:
   BEGIN
      TT:=1:  EXPREC(RESULT);
   END:

PROCEDURE INVERSA(VAR A1:MATRIZ;N:INTEGER:VAR DET:REAL):
  VAR
     PIV:ARRAY[1..20]OF INTEGER;
     REN,COL,I,J,R,RR,M:INTEGER;   MAX,PIVOTE:REAL;
     AA:ARRAY[1..20,1..2]OF INTEGER;
BEGIN
   DET:=1;
```

```
FOR I:=1 TO N DO   PIV[I]:=0;   (* BUSCA EL PIVOTE DE LA MATRIZ  *)
FOR R:=1 TO N DO
  BEGIN
    MAX:=0;
    FOR I:=1 TO N DO
      IF PIV[I] <> 1 THEN
        FOR J:=1 TO N DO
          BEGIN
            IF PIV[J] <> 1 THEN
            IF ABS(MAX) <= ABS(A1[I,J]) THEN
              BEGIN
                REN:=I; COL:=J; MAX:=A1[I,J];
              END;
          END;
    PIVOTE:=A1[REN,COL];
    IF ABS(PIVOTE) >= 0.00000001  THEN
      BEGIN
        DET:=DET*PIVOTE; AA[R,1]:=REN; AA[R,2]:=COL; PIV[COL]:=1;
        IF REN <> COL THEN
          BEGIN
            DET:=-DET;
            FOR J:=1 TO N DO
              BEGIN
                MAX:=A1[REN,J];A1[REN,J]:=A1[COL,J];A1[COL,J]:=MAX;
              END;
          END;
        A1[COL,COL]:=1;
        FOR J:=1 TO N DO   A1[COL,J]:=A1[COL,J]/PIVOTE;
        FOR I:=1 TO N DO
          IF I <> COL  THEN
            BEGIN
              MAX:=A1[I,COL];   A1[I,COL]:=0;
              FOR J:=1 TO N DO
                A1[I,J]:=A1[I,J]-MAX*A1[COL,J];
            END;
      END
    ELSE
      DET:=0;
  END;
IF ABS(PIVOTE) >= 0.00000001 THEN
  BEGIN
    FOR RR:=1 TO N DO
      BEGIN
        I:=N-RR+1;
        IF AA[I,1] <> AA[I,2]  THEN
          BEGIN
            REN:=AA[I,1]; COL:=AA[I,2];
            FOR I:=1 TO N DO
              BEGIN
```

```
                            MAX:=A1[I.REN]:  A1[I,REN]:=A1[I.COL];
                            A1[I,COL]:=MAX:
                         END;
                    END:
               END;
            END
         ELSE          DET:=0:
      END;

PROCEDURE DATOS(VAR P:INTEGER:VAR TI.INCREM:REAL):
VAR
   TR:REAL;      I:INTEGER;      BAN:BOOLEAN:
BEGIN     {$I-}
   REPEAT
      GOTOXY(3,4):    CLREOL;    TI:=0;
      WRITE('TIEMPO DE INICIO T(0) = 0 --> ');READ(TI):WRITELN('   -->  '.TI:8)
      BAN:=(IORESULT=0):
   UNTIL    BAN AND (TI>=0)     :
   REPEAT
      GOTOXY(3.5):    CLREOL;       TR:=ABS(EIGMIN*5):
      WRITE('TIEMPO DE RESPUESTA =      '.TR:8.'    -->   ');READ(TR):
      WRITELN('    --> '.TR:8):    BAN:=(IORESULT=0):
   UNTIL BAN AND (TR>0):
   REPEAT
      GOTOXY(3,7):    CLREOL:
      WRITE('NUMERO DE PUNTOS A CALCULAR   =   '):READLN(P):BAN:=(IORESULT=0):
   UNTIL  BAN AND (P>0) ;
   REPEAT
      GOTOXY(3,9):    CLREOL;
      WRITE('NUMERO DE TERMINOS DE LA SERIE + 20 =   '):READLN(NUM);
      BAN:=(IORESULT=0);
   UNTIL    BAN AND (NUM>0):
   WRITELN;
   WRITELN('  CONDICIONES INICIALES DEL VECTOR   X(T(0))    '):
   FOR I:=1 TO N DO
      BEGIN
         REPEAT
            WRITE('   X[',I,'] (T(0)) =    '):READLN(XI[I]):BAN:=(IORESULT=0);
         UNTIL   BAN:
      END:
   {$I+}
   WRITELN('DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA'):
   WRITELN('SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION ');
   WRITELN('  ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.'):
   FOR I:=1 TO P DO
      BEGIN
         WRITE('  U[',I,']    (t)   =    '):READLN(CADENN[I]):
         CADENN[I]:=CONCAT(CADENN[I],'$'):
      END:
```

```
 INCREM:=TR/P;    T:=0    ; WRITELN(' EL INCREMENTO DEL TIEMPO -->  ',INCREM);
END;

PROCEDURE  RESPLIBRE(G:MATRIZ;N,P:INTEGER:TI,INCREM:REAL);
TYPE
   VECTOR=ARRAY[1..NN]OF REAL;
VAR
   I,J,II,K:INTEGER;
   XX,XXI:VECTOR;
BEGIN
   FOR I:=1 TO N DO   XXI[I]:=XI[I];
   FOR I:=1 TO N DO   X[I,0]:=XXI[I];
   FOR II:=1 TO P DO
      BEGIN
         FOR I:=1 TO N DO   XX[I]:=0;
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO   XX[I]:=XX[I]+G[I,J]*XXI[J];
         FOR K:=1 TO N DO
            BEGIN
               XXI[K]:=XX[K];   X[K,II]:=XX[K];
            END;
      END;
END;

PROCEDURE PROD(AA,BB:MATRIZ;N1,N2,N3:INTEGER;VAR CC:MATRIZ);
VAR
   I,J,K:INTEGER;    S:REAL;
BEGIN
   FOR I:=1 TO N1 DO
      FOR J:=1 TO N2  DO
         BEGIN
            S:=0;
            FOR K:=1 TO N3 DO
               S:=S+AA[I,K]*BB[K,J];
            CC[I,J]:=S;
         END;
END;

PROCEDURE ENTRADA(VAR U:SUMAS;X:REAL);
   VAR
      I:INTEGER;    RESULTADO:REAL;
   BEGIN
      FOR I:=1 TO R DO
         BEGIN
            EXPRESION(CADENA[I],X,RESULTADO);  U[I]:=RESULTADO;
         END;
   END;

PROCEDURE INTEGEXPB(VAR EXPB:MATRIZ;A,B:MATRIZ;N,R,PP:INTEGER;TI,INC:REAL);
```

```
VAR
   I,J,K,II,PA,KK:INTEGER:   TPUN.TE.INCREM:REAL:
   SUMPUN,PRIPUN,ULTPUN,EXPMULTB:MATRIZ:
PROCEDURE FUNEXPB(TPUN,TE:REAL:VAR EXPMULTB:MATRIZ):
   VAR
      I.J.K:INTEGER:    SUMA:REAL:
      AEXPTX:MATRIZ:
   BEGIN
      EXPA(AEXPTX.A,TPUN-TE.N.1000+NUM):
      FOR I:=1 TO N DO
         FOR J:=1 TO R DO
            BEGIN
               SUMA:=0:
               FOR K:=1 TO N DO
                  SUMA:=SUMA+AEXPTX[I.K]*B[K.J]:
               EXPMULTB[I.J]:=SUMA:
            END;
   END;
BEGIN
   INCREM:=INC/PP:     TPUN:=INC:     TE:=TI:      PA:=PP-1:
   FOR I:=1 TO N DO
      FOR J:=1 TO R DO   SUMPUN[I.J]:=0;
   FOR I:=1 TO PA DO
      BEGIN
         TE:=TE+INCREM:
         FUNEXPB(TPUN.TE.EXPMULTB):
         FOR K:=1 TO N DO
            FOR KK:=1 TO R DO
               SUMPUN[K.KK]:=SUMPUN[K,KK]+EXPMULTB[K.KK]:
      END:
      FUNEXPB(TPUN.TI,EXPMULTB):
      FOR I:=1 TO N DO
         FOR J:=1 TO R DO   PRIPUN[I,J]:=EXPMULTB[I,J]:
      FUNEXPB(TPUN,TPUN,EXPMULTB):
      FOR I:=1 TO N DO
         FOR J:=1 TO R DO   ULTPUN[I J]:=EXPMULTB[I,J];
      FOR I:=1 TO N DO
         FOR J:=1 TO R DO
            EXPB[I.J]:=(PRIPUN[I.J]+ULTPUN[I.J]-(2*SUMPUN[I.J]))*INCREM/2:
END;

  PROCEDURE FORMINTEG(VAR HH:MATRIZ:GG.B:MATRIZ:N.R:INTEGER):
  VAR
     K:INTEGER:     INVA.AA:MATRIZ:
  BEGIN
     INVA:=A:    INVERSA(INVA.N.DET):
     FOR K:=1 TO N DO    GG[K.K]:=GG[K.K]-1:
     PROD(INVA.GG.N.N.N,AA):
```

```
     PROD(AA,B,N,R,K,INC);
  END;

 PROCEDURE MENEIGEN(AA:MATRIZ;N:INTEGER;VAR MENVCAR:REAL);
  VAR    C,XA:SUMAS; V,DET:REAL; I,J,ITER,CONT,NITER:INTEGER;AUX:BOOLEAN;
 BEGIN
    IF N<3 THEN NITER:=500 ELSE IF N>2 THEN NITER:=80 ELSE NITER:=0;
    IF NITER<>0 THEN IF DETERM(A,N) <> 0 THEN INVERSA(AA,N,DET) ELSE DET:=0;
    V:=0;CONT:=0; ITER:=0; AUX:=TRUE;
    FOR I:=1 TO N DO XA[I]:=0.1;
    WHILE (ITER < NITER) AND (DET <> 0) AND (AUX) DO
      BEGIN
         FOR I:=1 TO N DO
            BEGIN
               C[I]:=0;
               FOR J:=1 TO N DO   C[I]:=C[I]+AA[I,J]*XA[J];
            END;
         ITER:=ITER+1;    MENVCAR:=C[1];
         FOR I:=2 TO N DO
            IF ABS(MENVCAR) < ABS(C[I]) THEN    MENVCAR:=C[I];
         IF   MENVCAR <> 0 THEN   FOR I:=1 TO N DO XA[I]:=C[I]/MENVCAR
           ELSE  BEGIN
                     CONT:=CONT+1;
                     CASE CONT OF
                        1:FOR I:=1 TO N DO   XA[I]:=-0.1;
                        2:FOR I:=1 TO N DO   XA[I]:=-0.5;
                        3:FOR I:=1 TO N DO   XA[I]:=-10;
                        4:FOR I:=1 TO N DO   XA[I]:=20;
                        5:FOR I:=1 TO N DO   XA[I]:=-20;
                        6:FOR I:=1 TO N DO   XA[I]:=-5;
                        ELSE AUX:=FALSE;
                     END;
                 END;
         IF ABS(MENVCAR-V) > 0.00001 THEN    V:=MENVCAR
             ELSE  AUX:=FALSE;
      END;
  IF (ITER >= NITER) OR (DET = 0)    THEN      MENVCAR:=0
  ELSE MENVCAR:=ABS((1/V)*5);
END;

 PROCEDURE RESPORITER(GGG,HHH:MATRIZ;TI,INC:REAL;VAR AREA:PUNTOS);
    VAR
       U,EXPSU,XXK,XX:SUMAS;
       I,J,K,II:INTEGER;       SUMA,T:REAL;
 BEGIN
    FOR I:=1 TO N DO   XXK[I]:=X[I];          T:=INC;
    FOR I:=1 TO N DO   AREA[I,0]:=X[I];
    FOR II:=1 TO P DO
       BEGIN
```

```
          ENTRADA(U,T);
          FOR I:=1 TO N DO
             BEGIN
                SUMA:=0;
                FOR K:=1 TO R DO
                   SUMA:=SUMA+HHH[I,K]*U[K];
                EXPBU[I]:=SUMA;
             END;
          FOR I:=1 TO N DO
             BEGIN
                SUMA:=0;
                FOR K:=1 TO N DO
                   SUMA:=SUMA+GGG[I,K]*XX[K];
                XX[I]:=SUMA;
             END;
          FOR J:=1 TO N DO    AREA[J,II]:=XX[J]+EXPBU[J];
          FOR K:=1 TO N DO    XX[K]:=AREA[K,II];            T:=T+INC;
       END;
 END;

PROCEDURE BAIRSTOW(AAAAA:MATRIZ;GRAD:INTEGER;VAR RAICESR,RAICESI:SUMAS);
CONST
   ER=1E-9;
VAR
   ITER,CONT,I,J,M,NN,NM1,JM1,NEWN:INTEGER;
   P,Q,RAD,CMB,DENOM,DELP,DELQ,RAITR:REAL;
   A,B,C,RTREA,RTIMA,POLEVER:SUMAS;  SALIRSE:BOOLEAN;
BEGIN
   (*    LLAMA POLINOMIO DE LA MATRIZ A   *)
   LEVERRIER(AAAAA,GRAD,POLEVER); SALIRSE:=TRUE;
   FOR I:=1 TO GRAD DO
      A[I]:=-(POLEVER[I]);
   NUM:=0; M:=0;ITER:=500;CONT:=0; P:=0;Q:=0;NN:=GRAD-2*M;J:=1;
   WHILE (CONT<=ITER) AND (NN>2) AND SALIRSE DO
      BEGIN
         B[1]:=A[1]-P;
         B[2]:=A[2]-P*B[1]-Q;
         FOR I:=3 TO NN DO
            B[I]:=A[I]-P*B[I-1]-Q*B[I-2];
         NM1:=NN-1;
         C[1]:=B[1]-P;
         C[2]:=B[2]-P*C[1]-Q;
         C[3]:=B[3]-P*C[2]-Q*C[1];
         IF (NN > 3)  THEN
            FOR I:=3 TO NM1 DO
               C[I]:=B[I]-P*C[I-1]-Q*C[I-2];
         CMB:=C[NM1]-B[NM1];
         (*  RESUELVE EL SISTEMA DE ECUACIONES    *)
         IF (NN <> 3)   THEN
```

```
    DENOM:=(C[NN-2]*C[NN-2])-CMB*C[NN-3]
ELSE
    DENOM:=(C[NN-2]*C[NN-2])-CMB;
IF (DENOM <> 0 )   THEN
    BEGIN
       IF (NN <> 3)     THEN
          DELP:=(B[NN-1]*C[NN-2]-B[NN]*C[NN-3])/DENOM
       ELSE
          DELP:=(B[NN-1]*C[NN-2]-B[NN])/DENOM;
       DELQ:=(B[NN]*C[NN-2]-B[NN-1]*CMB)/DENOM;
       P:=P+DELP;
       Q:=Q+DELQ;
    END;
IF ((ABS(DELP) > ER) OR  ABS(DELQ) > ER)) AND (CONT>=ITER) THEN
    BEGIN
       NUM:=NUM+1;
       CASE  NUM  OF
          1:BEGIN   P:=-2;  Q:=2;   CONT:=0;    END;
          2:BEGIN   P:=2;   Q:=-2;  CONT:=0;    END;
          3:BEGIN   P:=5;   Q:=5;   CONT:=0;    END;
          4:BEGIN   P:=-5;  Q:=-5;  CONT:=0;    END;
          5:BEGIN   P:=20;  Q:=20;  CONT:=0;    END;
          6:BEGIN   P:=-50;  Q:=50;  CONT:=0;   END;
          7:BEGIN   P:=50;   Q:=-50;  CONT:=0;  END;
          8:BEGIN   P:=-100;  Q:=100;  CONT:=0;  END;
          9:BEGIN   P:=100;   Q:=100;  CONT:=0;  END;
          ELSE  BEGIN GOTOXY(5,5):WRITE('NO SE LLEGO A ENCONTRAR ');
                      WRITELN('LAS RAICES');JM1:=J-1;  SALIRSE:=FALSE;
                END;
        END;
     END
  ELSE
  BEGIN
     IF ((ABS(DELP)>ER) OR (ABS(DELQ)>ER)) AND (CONT <  ITER) THEN
        CONT:=CONT+1
     (* SE OBTIENEN RAICES CORRESPONDIENTES AL FACTOR CUADRATICO *)
     ELSE
  IF ((ABS(DELP) < ER) AND (ABS(DELQ) < ER))    THEN
     BEGIN
        RADTR:=P*P-4*Q;
        (*  RAICES COMPLEJAS CONJUGADAS  *)
        IF RADTR < 0    THEN
           BEGIN
              RAD:=SQRT(-RADTR);
              RTREA[J]:=(-P)/2;
              RTREA[J+1]:=-P/2;
              RTIMA[J]:=RAD/2;
              RTIMA[J+1]:=-RAD/2;
           END;
```

```
               (*  RAICES REALES REPETIDAS  *)
               IF RADTR = 0   THEN
                  BEGIN
                     RTREA[J]:=-P/2;
                     RTREA[J+1]:=-P/2;
                     RTIMA[J]:=0;  RTIMA[J+1]:=0;
                  END;
               IF RADTR > 0   THEN
                  BEGIN  (* RAICES REALES DIFERENTES  *)
                     RAD:=SQRT(RADTR);
                     RTREA[J]:=(-P+RAD)/2;
                     RTREA[J+1]:=(-P-RAD)/2;
                     RTIMA[J]:=0;   RTIMA[J+1]:=0;
                  END;
   (* REDUCIR ORDEN DEL POLINOMIO Y CAMBIAR COEFICIENTES  *)
               M:=M+1;  J:=J+2;  NUM:=0;
               P:=0;  Q:=0;   NEWN:=NN-2*M;
               FOR I:=1 TO NEWN DO
                  A[I]:=B[I];
               CONT:=0;   NN:=GRAD-2*M;
            END;
         END;
      END;
  (* CALCULA EL PAR DE RAICES DE LA ULTIMA EC. CUADRATICA  *)
  IF NN = 2   THEN
      BEGIN
         RADTR:=A[1]*A[1]-4*A[2];
         IF  RADTR < 0   THEN
            BEGIN
               RAD:=SQRT(-RADTR);
               RTREA[J]:=-A[1]/2;  RTREA[J+1]:=-A[1]/2;
               RTIMA[J]:=RAD/2;   RTIMA[J+1]:=-RAD/2;
            END
         ELSE
            IF RADTR = 0   THEN
               BEGIN
                  RTREA[J]:=-A[1]/2;  RTREA[J+1]:=-A[1]/2;
                  RTIMA[J]:=0;  RTIMA[J+1]:=0;
               END
            ELSE
               BEGIN
                  RAD:=SQRT(RADTR);
                  RTREA[J]:=(-A[1]+RAD)/2; RTREA[J+1]:=(-A[1]-RAD)/2;
                  RTIMA[J]:=0; RTIMA[J+1]:=0;
               END;
      END;
  IF NN = 1  THEN  (*  RAIZ FACTORIAL LINEAL  *)
      BEGIN
         RTREA[J]:=-A[1];  RTIMA[J]:=0;
```

```
      END;
   FOR I:=1 TO GRAD DO   BEGIN
         RAICESR[I]:=RTREA[I];   RAICESI[I]:=RTIMA[I]; END;(* DEL FOR *)
   IF SALIRSE THEN LISEIGEN(RAICESR,RAICESI);   READ(PAUSA);
END;


  PROCEDURE MULTPOLIN(P1,P2:ENLACE;VAR POL:ENLACE);
  VAR
     P,SIGP,SIGP1,PP1,SIGP2:ENLACE;
     PP:ARRAY[1..3]OF ENLACE;
     NPOL:INTEGER;
  FUNCTION BUSTER(POL:ENLACE;EXP:INTEGER):ENLACE;
  VAR
     P,PANT:ENLACE;
     BUSCA:BOOLEAN;
  BEGIN
     BUSCA:=TRUE;    P:=POL;
     WHILE (P <> NIL ) AND BUSCA   DO
        BEGIN
           IF EXP = P^.EXP   THEN
              BUSCA:=FALSE
           ELSE
              BEGIN
                 PANT:=P^.SIGUIENTE;  P:=PANT;
              END;
        END;
     IF BUSCA THEN   P:=NIL;
     BUSTER:=P;
  END;

  PROCEDURE SUMPOL(VAR POL1:ENLACE;POL2:ENLACE);
  VAR
     P,PP,PANT,PPANT:ENLACE;    PRIM:BOOLEAN;
 BEGIN
    P:=POL1;
    PANT:=NIL;
    WHILE P <> NIL DO
       BEGIN
          PP:=BUSTER(POL2,P^.EXP);
          IF PP <> NIL   THEN
             BEGIN
                P^.COEF:=P^.COEF+PP^.COEF;
                POL2:=PP^.SIGUIENTE;
             END;
          PANT:=P;
          P:=PANT^.SIGUIENTE;
       END;
    (* PONE EL RESTO DEL POL2 EN EL POLINOMIO 1    *)
```

```
    IF POL2 <> NIL  THEN
       BEGIN
          PANT^.SIGUIENTE:=POL2;
       END;
 END;

BEGIN
   PP1:=P1;         NPOL:=0;
   WHILE (P2 <> NIL)  DO
      BEGIN
         NEW(P); NPOL:=NPOL+1; PP[NPOL]:=P;     P1:=PP1;
         WHILE P1 <> NIL   DO
            BEGIN
               P^.COEF:=(P1^.COEF)*(P2^.COEF);
               P^.EXP:=(P1^.EXP)+(P2^.EXP);
               SIGP1:=P1^.SIGUIENTE;     P1:=SIGP1;
               IF P1 <> NIL  THEN
                  BEGIN
                     NEW(SIGP); P^.SIGUIENTE:=SIGP;  P:=SIGP;
                  END
               ELSE
                  P^.SIGUIENTE:=NIL;
            END;
         SIGP2:=P2^.SIGUIENTE;
         P2:=SIGP2;
      END;
   IF  NPOL = 3  THEN
      SUMPOL(PP[2],PP[3]);
   SUMPOL(PP[1],PP[2]);
   POL:=PP[1];
END;

PROCEDURE CREAPOLIN(N:INTEGER;VAR ALFAS:SUMAS);
VAR
   NUMCOM,NUMREAL,I,NP,NUMRAIR:INTEGER;   MARCA:^INTEGER;
   POL:ARRAY[1..N]OF ENLACE; R,Z,A,B:REAL;
   P,SIG,POLTOT:ENLACE;
BEGIN
   WRITELN(' CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE  ');
{$I-}  REPEAT DELLINE;READ(NUMCOM); BAN:=(IORESULT=0)
   UNTIL BAN AND (NUMCOM >= 0) AND (NUMCOM <= N DIV 2);
{$I+}   NP:=0; (* CONTADOR DE POLINOMIOS SE CREAN  *) MARK(MARCA);WRITELN;
   IF NUMCOM > 0  THEN
      BEGIN
         FOR I:=1 TO NUMCOM DO
            BEGIN
               WRITE('P(REAL)  ');READ(R);WRITE('  P(IMAG)  ');READLN(Z);
               WRITELN('P(REAL) ',R:5:4,' P(IMAG) ',-Z);
               NP:=NP+1;  B:=Z*Z+R*R;  A:=-2*R; NEW(SIG); P:=SIG;
```

```
              P^.COEF:=B;   P^.EXP:=0;  P^.SIGUIENTE:=NIL;
              NEW(SIG);   SIG^.SIGUIENTE:=P;   P:=SIG;  P^.COEF:=A;
              P^.EXP:=1;   NEW(SIG);  SIG^.SIGUIENTE:=P;  P:=SIG;
              P^.COEF:=1;    P^.EXP:=2;     POL[NP]:=P;
           END;
     END;
   NUMRAIR:=N-2*NUMCON;
   IF  NUMRAIR > 0   THEN
      FOR I:=1 TO NUMRAIR DO
         BEGIN
            WRITE('DAR RAIZ REAL =    ');    READLN(A);
            NEW(SIG);  P:=SIG;P^.COEF:=-A;  P^.EXP:=0;  P^.SIGUIENTE:=NIL;
            NEW(SIG);   SIG^.SIGUIENTE:=P;    P:=SIG;
            P^.COEF:=1;  P^.EXP:=1;  NP:=NP+1;   POL[NP]:=P;
         END;
   (* MULTIPLICACION DE NP DE POLINOMIOS   *)
   IF NP > 1  THEN
      FOR I:=2 TO NP DO
         BEGIN
            MULTPOLIN(POL[1],POL[I],POLTOT);  POL[1]:=POLTOT;
         END;
   POLTOT:=POL[1];
   FOR I:=1 TO N DO
      BEGIN
         P:=POLTOT^.SIGUIENTE;POLTOT:=P;ALFAS[I]:=P^.COEF;
      END;
   RELEASE(MARCA);   WRITELN;
END;

 PROCEDURE ECCARACT(A:MATRIZ;N:INTEGER;VAR FIA:MATRIZ;ALFAS:SUMAS);
 VAR
    I,J,K,KK,NN:INTEGER;    SUMA:REAL;
    APORA,TERMINO,IDEN,APORAT:MATRIZ;
 BEGIN
    FOR I:=1 TO N DO
       FOR J:=1 TO N DO     IDEN[I,J]:=0;
    FOR I:=1 TO N DO    IDEN[I,I]:=1;
    FOR I:=1 TO N DO
       FOR J:=1 TO N DO
          BEGIN
             APORA[I,J]:=IDEN[I,J];  TERMINO[I,J]:=ALFAS[N]*IDEN[I,J];
          END;
    NN:=N;
    FOR KK:=1 TO N DO
       BEGIN
          NN:=NN-1;
          FOR I:=1 TO N DO
             FOR J:=1 TO N DO
                BEGIN
```

```
                    SUMA:=0;
                    FOR K:=1 TO N DO
                        SUMA:=SUMA+APORA[I,K]*A[K,J];
                    APORAT[I,J]:=SUMA;
                END;
            APORA:=APORAT;
            IF KK <> N  THEN
                BEGIN
                    FOR I:=1 TO N DO
                        FOR J:=1 TO N DO
                            TERMINO[I,J]:=TERMINO[I,J]+ALFAS[NN]*APORA[I,J];
                END
            ELSE
                BEGIN
                    FOR I:=1 TO N DO
                        FOR J:=1 TO N DO
                            TERMINO[I,J]:=TERMINO[I,J]+APORA[I,J];
                END;
        END;
    FIA:=TERMINO;
END;

PROCEDURE KREALIMENT(N:INTEGER;FIA:MATRIZ;CONEST:MATRIZ;VAR KREALIM:SUMAS
VAR
    TEMPO:MATRIZ;   VECTOR:SUMAS;    I,J,K:INTEGER;    DET,SUMA:REAL;
BEGIN
    FOR I:=1 TO N-1 DO    VECTOR[I]:=0;
    VECTOR[N]:=1;     INVERSA(CONEST,N,DET);
    FOR I:=1 TO N DO
        FOR J:=1 TO N DO
            BEGIN
                SUMA:=0;
                FOR K:=1 TO N DO
                    SUMA:=SUMA+CONEST[I,K]*FIA[K,J];
                TEMPO[I,J]:=SUMA;
            END;
    FOR J:=1 TO N DO
        BEGIN
            SUMA:=0;
            FOR K:=1 TO N DO
                SUMA:=SUMA+VECTOR[K]*TEMPO[K,J];
            KREALIM[J]:=SUMA;
        END;
END;

FUNCTION PSEUALEA(VAR SEMILLA:INTEGER):REAL;
CONST
    MULT=374;  INCRE=197;    MODULO=997;
BEGIN
```

```
    PSEUALEA:=SEMILLA/100;SEMILLA:=(MULT*SEMILLA+INCRE)MOD (MODULO);
END;

PROCEDURE BUSCABPORW(AA,BB:MATRIZ;ROM,N:INTEGER;VAR W:SUMAS;VAR BBW:MATRIZ);
VAR
    I,J,K,SEMILLA,CONTADOR:INTEGER;
    CUAL:MATRIZ;  SUMA:REAL;  VER:BOOLEAN;
BEGIN
    SEMILLA:=ROM*1;        CONTADOR:=0;  VER:=FALSE;
    WHILE NOT(VER) AND (CONTADOR < 30000)  DO
        BEGIN
            FOR I:=1 TO ROM DO
                W[I]:=PSEUALEA(SEMILLA);

            FOR I:=1 TO N DO
                BEGIN
                    SUMA:=0;
                    FOR K:=1 TO ROM DO   SUMA:=SUMA+BB[I,K]*W[K];
                    BBW[I,1]:=SUMA;
                END;
            CONTROESTA(AA,BBW,N,1,CUAL,VER,2); CONTADOR:=CONTADOR+1;
        END;
END;

PROCEDURE REALIMEN(A,B:MATRIZ;N,R:INTEGER;VAR  KREAMULT:MATRIZ);
VAR
    FIA,NUEVACONEST,BBW,CONEST:MATRIZ;
    I,J:INTEGER;SUMA:REAL;W,ALFASS,KTESTA,KREALIM:SUMAS;VERD:BOOLEAN;
BEGIN
    WRITELN('              DISEÑO DE REALIMENTACION DE ESTADO ');
    WRITELN('       Asignacion de Polos'); CONTROESTA(A,B,N,R,CONEST,VERD,2);
    IF VERD   THEN
        BEGIN
            WRITELN(' EL SISTEMA ES CONTROLABLE DE ESTADO ');
            IF R = 1   THEN
                BEGIN
                    CREAPOLIN(N,ALFASS);   ECCARACT(A,N,FIA,ALFASS);
                    KREALIMENT(N,FIA,CONEST,KREALIM);
                    WRITELN(' LA MATRIZ DE GANANCIA DE REALIMENTACION');
                    WRITELN(' K  DE ENTRADA-UNICA  ');WRITE(' K[1 ',N,']=  ');
                    FOR I:=1 TO N DO   WRITE(KREALIM[I]);
                    FOR I:=1 TO N DO   KREALIMULT[1,I]:=KREALIM[I];
                END
            ELSE
                BEGIN
                    BUSCABPORW(A,B,R,N,W,BBW);
                    CONTROESTA(A,BBW,N,1,NUEVACONEST,VERD,2);
                    IF VERD  THEN
                        BEGIN
```

```
                     CREAPOLIN(N,ALFASS);   ECCARACT(A,N,FIA,ALFASS);
                     KREALIMENT(N,FIA,NUEVACONEST,KTESTA);
                     FOR I:=1 TO R DO
                        FOR J:=1 TO N DO
                           BEGIN
                              SUMA:=0; SUMA:=SUMA+W[I]*KTESTA[J];
                              KREAMULT[I,J]:=SUMA;
                           END;
                     WRITELN(' LA MATRIZ DE GANANCIA DE REALIMENTACION');
                     WRITELN('   K DE ENTRADAS MULTIPLES ');
                     FOR I:=1 TO R DO
                       BEGIN
                         FOR J:=1 TO N DO WRITE(KREAMULT[I,J],'  ');WRITELN;
                       END;
                  END
               ELSE
                  WRITELN('NO SE PUEDE HACER LA REALIMENTACION EN MUL');
            END
         END
      ELSE
         BEGIN
            WRITELN('EL SISTEMA NO SE PUEDE REALIMENTAR,NO ES ');
            WRITELN('CONTROLABLE DE ESTADO ');
         END;
 READ(PAUSA);
 END;

PROCEDURE OBSERVADOR(A,C:MATRIZ;N,M:INTEGER;VAR KESTIMADA:SUMAS;
                                   VAR KESTIMMUL:MATRIZ;LL:BYTE);
VAR
   ATRANP,CTRANP,NUEVACONEST,FIA,BBW,KREAMULT:MATRIZ;
   W,ALFASS,KTESTATRAN:SUMAS;     I,J:INTEGER;     VERD:BOOLEAN;
BEGIN
IF LL=1 THEN WRITELN('DISEÑO DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   OBSERVAB(A,C,N,M,VERD,2);
   IF VERD  THEN
      BEGIN
         WRITELN('    EL SISTEMA ES OBSERVABLE ');
         FOR I:=1 TO N DO
            FOR J:=1 TO N DO    ATRANP[J,I]:=A[I,J];
         FOR I:=1 TO M DO
            FOR J:=1 TO N DO    CTRANP[J,I]:=C[I,J];
         IF M = 1    THEN
            BEGIN
               CREAPOLIN(N,ALFASS);    ECCARACT(ATRANP,N,FIA,ALFASS);
               CONTROESTA(ATRANP,CTRANP,N,M,CONEST,VERD,2);
               KREALIMENT(N,FIA,CONEST,KESTIMADA);
               FOR I:=1 TO N DO    KESTIMMUL[I,1]:=KESTIMADA[I];
               WRITELN('LA MATRIZ K ESTIMADA ES   ');
```

```
            FOR I:=1 TO N DO    WRITELN(' K[',I,']',KESTIMMUL[I,1]);
          END
        ELSE
          BEGIN
            BUSCADPORW(ATRANP,CTRANP,M,N,W,BBW);
            CONTROESTA(ATRANP,BBW,N,1,NUEVACONEST,VERD,2);
            IF VERD   THEN
              BEGIN
                CREAPOLIN(N,ALFASS);   COCARACT(ATRANP,N,FIA,ALFASS);
                KREALIMENT(N,FIA,NUEVACONEST,KTESTATRAN);
                FOR I:=1  TO N DO
                  FOR J:=1  TO N DO
                    BEGIN
                      SUMA:=0;   SUMA:=SUMA+W[I]*KTESTATRAN[J];
                      KREAMULT[I,J]:=SUMA;
                      KESTIMMUL[J,I]:=KREAMULT[I,J];
                    END;
                WRITELN('LA MATRIZ    Ke ( K - ESTIMADA )     ');
                FOR I:=1  TO N DO
                  BEGIN
                    FOR J:=1 TO M DO    WRITE(KESTIMMUL[I,J]);
                    WRITELN;
                  END;
              END
            ELSE
              WRITELN('  NO SE PUEDE HACER EL OBSERVADOR DE SAL-MULT.');
          END
        END
    ELSE
        WRITELN(' EL SISTEMA NO ES OBSERVABLE ');
  READ(PAUSA);
  END;

PROCEDURE OBSERMINIMO(A,B,C:MATRIZ;N,R,M:INTEGER);
VAR
   RAN,I,J,SEMILLA,CONTADOR:INTEGER; DET:REAL;  BAND:BOOLEAN;
   P,PINV,PA,PAQ,PB,CC,ABE,AAE:MATRIZ;
BEGIN
   WRITELN('                DISEÑO DEL OBSERVADOR DE DIMENSION REDUCIDA ');
   SEMILLA:=1;    CONTADOR:=0;    BAND:=TRUE;WRITELN;WRITELN;
   IF  (M < N)    THEN
     BEGIN
       FOR I:=1 TO M DO
         BEGIN
           IF C[I,I] <> 1    THEN    BAND:=FALSE;
           FOR J:=1 TO N DO
             IF I <> J  THEN
               IF C[I,J] <> 0 THEN   BAND:=FALSE;
          END;
```

```
        IF BAND = FALSE    THEN
           BEGIN
              FOR I:=1 TO M DO
                 FOR J:=1 TO N DO    P[I,J]:=C[I,J];
              REPEAT
                 FOR I:=M+1 TO N DO
                    FOR J:=1 TO N DO    P[I,J]:=PSEUALEA(SEMILLA);
                 PINV:=P; INVERSA(PINV,N,DET); CONTADOR:=CONTADOR+1;
              UNTIL (DET <> 0) OR (CONTADOR > 1000);
              IF DET <> 0    THEN
                 BEGIN
                    PROD(C,PINV,M,N,N,CO); PROD(P,A,N,N,N,PA);
                    PROD(PA,PINV,N,N,N,PAO); PROD(P,B,N,R,N,PB);
                    WRITELN(' LA MATRIZ DE TRANS Q');WRITELN(' Q(1) ES = ');
                    FOR I:=1 TO N DO
                       BEGIN
                          FOR J:=1 TO M DO  WRITE(PINV[I,J],' ');WRITELN;
                       END;
                    WRITELN('   Q(2) ES = ');
                    FOR I:=1 TO N DO
                       BEGIN
                          FOR J:=M+1 TO N DO  WRITE(PINV[I,J],' ');WRITELN
                       END;
                 END;
           END
        ELSE
           BEGIN
              PAO:=A;  PB:=B;   CO:=C;
           END;
      IF (CONTADOR < 999)  THEN
       BEGIN
        FOR I:=1 TO N-M DO
           FOR J:=1 TO N-M DO    ABB[I,J]:=PAO[I+M,J+M];
        FOR I:=1 TO M DO
           FOR J:=1 TO N-M DO    AAB[I,J]:=PAO[I,J+M];
        WRITELN(' PARA EL SISTEMA TRANSFORMADO (su realimentacion es)');
        REALIMEN(PAO,PB,N,R,(REALIMULT); WRITELN;WRITELN;
        WRITELN('   LA MATRIZ L ( estimada ) DEL OBSERVADOR MINIMO ES  ');
        OBSERVADOR(ABB,AAB,N-M,M,(ESTIMADA,(ESTIMMULT,I)
       END
      ELSE BEGIN  WRITE(' NO SE ENCONTRO LA MATRIZ DE ');
                  WRITELN('TRANSFORMACION Q '); READ(PAUSA);        END
    END
  ELSE    BEGIN
  WRITELN('DISE&O OBSERVADOR-MINIMO NO SE PUEDE HACER ');READ(PAUSA);END;
END;


PROCEDURE METERDATOS(PRIMPAS:BOOLEAN);
```

```
VAR I,J,II,K:INTEGER; T,SUMA:REAL;   U:SUMAS;
BEGIN
   IF PRIMPAS THEN  MENEIGEN(A,N,EIGMIN);
   CLRSCR; DATOS(P,TI,INC); PRIERRENT:=TRUE; WRITE(' ESPERE POR FAVOR   ');
   IF (TIPOSIS='C') OR (TIPOSIS='c')    THEN
      BEGIN
         EXPA(G,A,INC,N,NUM);  (* CALCULO DE G *)
         IF (DETERM(A,N) <> 0)    THEN   (* CALCULO DE H *)
            FORMINTEG(H,G,B,N,R)
         ELSE
            BEGIN
               IF INC < 0.25 THEN FUNINTG:=21
               ELSE IF INC < 0.5 THEN FUNINTG:=30
               ELSE IF INC < 1    THEN FUNINTG:=51
               ELSE IF INC < 2    THEN FUNINTG:=60
               ELSE IF INC < 3    THEN FUNINTG:=75
               ELSE FUNINTG:=99;
               INTEGEXPB(H,A,B,N,R,FUNINTG,TI,INC);
            END;
      END
   ELSE
      BEGIN
         G:=A;    H:=B;
      END;
   FOR I:=1 TO N DO
      FOR J:=1 TO P+1 DO    AREA[I,J]:=0;
   RESPLIBRE(G,N,P,TI,INC);  RESFORITER(G,H,TI,INC,AREA);
   T:=0;
   FOR II:=0 TO P DO
      BEGIN
         FOR I:=1  TO M DO
            BEGIN
               SUMA:=0;
               FOR K:=1 TO N DO
                  SUMA:=SUMA+C[I,K]*AREA[K,II];
               SOLY[I,II]:=SUMA;
            END;
         ENTRADA(U,T) ;     T:=T+INC;
         FOR I:=1  TO M DO
            BEGIN
               SUMA:=0;
               FOR K:=1 TO R DO
                  SUMA:=SUMA+D[I,K]*U[K];
               SOLY[I,II]:=SOLY[I,II]+SUMA;
            END;
      END;  WRITELN('---------->  CALCULOS : TERMINADOS'); READLN(PAUSA);
END;

PROCEDURE DISESIST(TIPO:BYTE);
```

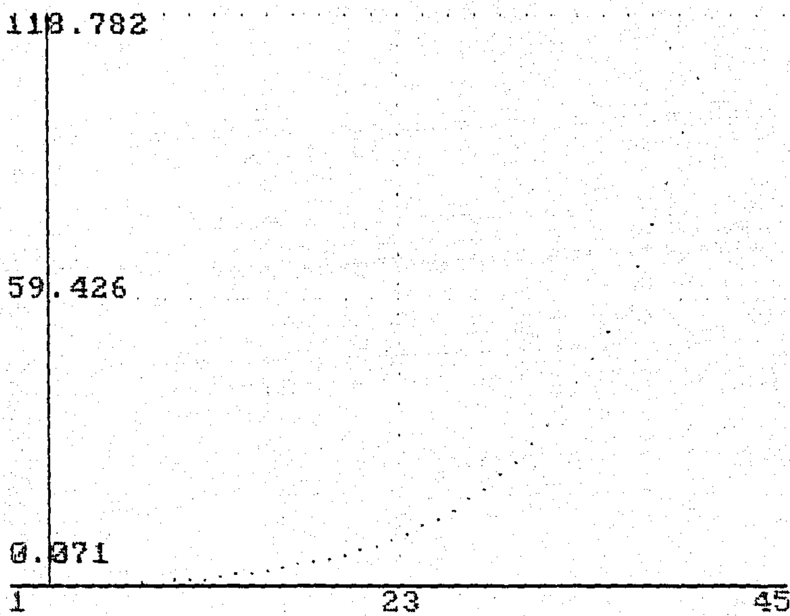
```
 VAR    I,J,K:INTEGER;    SUMA:REAL;
BEGIN
 IF TIPO = 1 THEN
    BEGIN
       A1:=A; B1:=B; C1:=C;
       GOTOXY(5,15);WRITE('  DISEÑO DEL SISTEMA REALIMENTADO (listo) ');
       FOR I:=1 TO N DO
          FOR J:=1 TO N DO
             BEGIN
                SUMA:=0;
                FOR K:=1 TO R DO
                   SUMA:=SUMA+B[I,K]*REALIMULT[K,J];
                A[I,J]:=A[I,J]-SUMA;
             END;
    END
 ELSE BEGIN    A:=A1;B:=B1;C:=C1; GOTOXY(5,15);
               WRITE(' SISTEMA ANTERIOR (listo) ');  END; READLN(PAUSA);
END;
  (*   P R O G R A M A    P R I N C I P A L   *)
BEGIN          SI:='S';
   WHILE (SI = 'S') OR (SI = 's')  DO
      BEGIN
         CLRSCR; LECTURA; P:=1;  PRIMPAS:=TRUE;
         WHILE (SI ='S') OR (SI ='s')    DO
            BEGIN
               CLRSCR;
               WRITELN(' ':30,' 0: METER DATOS ');
               WRITELN(' ':30,' 1: VER MATRICES DEL SISTEMA ACTUAL');
               WRITELN(' ':30,' 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)');
               WRITELN(' ':30,' 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)');
               WRITELN(' ':30,' 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)');
               WRITELN(' ':30,' 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)');
               WRITELN(' ':30,' 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)');
               WRITELN(' ':30,' 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)');
               WRITELN(' ':30,' 8: VERIFICACION CONTROLABILIDAD DE ESTADO ')
               WRITELN(' ':30,' 9: VERIF.CONTROLABILIDAD DE LA SALIDA ');
               WRITELN(' ':30,'10: VERIFICACION OBSEVABILIDAD  ');
               WRITELN(' ':30,'11: REALIMENTACION DEL ESTADO ');
               WRITELN(' ':30,'12: OBSERVADOR   ');
               WRITELN(' ':30,'13: DISEÑO DEL OBSERVADOR MINIMO DE ESTADO ')
               WRITELN(' ':30,'14: VALORES CARACTERISTICOS DEL SISTEMA  ');
               WRITELN(' ':30,'15: EC. EST.-->SIST. DISEÑADO REALIMENTACION'
               WRITELN(' ':30,'16: VOLVER AL SISTEMA ANTERIOR');
               WRITELN(' ':30,'17: SALIRSE  ');
               {$I-}
               REPEAT          GOTOXY(10,22);
                  CLREOL;WRITE('Opcion --> ');READ(CASO);BAN:=(IORESULT=0);
               UNTIL ((BAN) AND (CASO >= 0) AND (CASO <18));
               {$I+}
```

```
            CLRSCR;
            CASE CASO OF
                0:BEGIN METERDATOS(PRIMPAS);PRIMPAS:=FALSE;PERR:=TRUE;END;
                1:VERMATRIZ;
                2:LISRESLIB(X,N,P);
                3:LISARREG(AREA,N,P,1);
                4:LISARREG(SOLY,M,P,2);
                5:NGRAFICAS(X,0,N,1);
                6:NGRAFICAS(AREA,1,N,2);
                7:NGRAFICAS(SOLY,1,M,2);
                8:CONTROESTA(A,B,N,R,CONEST,VERD,1);
                9:CONTROSAL(A,B,C,D,N,R,M);
               10:OBSERVAB(A,C,N,M,VERD,1);
               11:REALIMEN(A,B,N,R,KREALIMULT);
               12:OBSERVADOR(A,C,N,M,KESTIMADA,KESTIMMULT,1);
               13:OBSERMINIMO(A,B,C,N,R,M);
               14:BAIRSTOW(A,N,RAICESR,RAICESI);
               15:BEGIN PRIMPAS:=TRUE; DISESIST(1); P:=0;    END;
               16:BEGIN PRIMPAS:=TRUE; DISESIST(2); P:=0;    END;
               17:SI:='N';
               ELSE    SI:='N';
          END;
       END;
     GOTOXY(5,19);WRITELN('DISEÑO DE OTRO SISTEMA    S/N');
     REPEAT  DELLINE;  READ(SI);
     UNTIL SI IN ['S','s','N','n'];
   END;
END.
```

1.- ESPACIO DE ESTADOS : SISTEMAS MULTIVARIABLES

```
 N:NUMERO DE VARIABLES DE ESTADO DEL SISTEMA = 4

 R:NUMERO DE ENTRADAS DEL SISTEMA  = 2

 M:NUMERO DE SALIDAS DEL SISTEMA =  2

   SISTEMA T. CONTINUO=C o T.DISCRETO=D   C

DAR LA MATRIZ    [A]
  A[1,1] = .4  A[1,2] = .7  A[1,3] = .9  A[1,4] = .1
  A[2,1] = 0  A[2,2] = .3  A[2,3] = .5  A[2,4] = .2
  A[3,1] = .7  A[3,2] = .6  A[3,3] = .3  A[3,4] = .4
  A[4,1] = .2  A[4,2] = .2  A[4,3] = .1  A[4,4] = .06

DAR LA MATRIZ    [B]
  B[1,1] =  .6  B[1,2] =  .4
  B[2,1] =  .3  B[2,2] =  .21
  B[3,1] =  .45  B[3,2] =  .6
  B[4,1] =  .4  B[4,2] =  .89

DAR LA MATRIZ  [C]
  C[1,1]  =  .8  C[1,2]  =  .7  C[1,3]  =  .9  C[1,4]  =  .98
  C[2,1]  =  .3  C[2,2]  =  .12  C[2,3]  =  1.6  C[2,4]  =  0




VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
D
   MATRIZ [D]
  2.0000000000E-01   3.0000000000E-01
  5.0000000000E-01   1.0000000000E-01
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

```
TIEMPO DE INICIO T(0) = 0 -->    --> 0.00E+00
TIEMPO DE RESPUESTA =    0.00E+00   -->  3   --> 3.00E+00

NUMERO DE PUNTOS A CALCULAR  =  45

NUMERO DE TERMINOS DE LA SERIE + 20 =  1

CONDICIONES INICIALES DEL VECTOR  X(T(0))
X[1] (T(0)) =  0
X[2] (T(0)) =  0
X[3] (T(0)) =  0
X[4] (T(0)) =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o ver.quela.
U[1]   (t)  =   1
U[2]   (t)  =   1
EL INCREMENTO DEL TIEMPO -->     6.66666666E-02
ESPERE POR FAVOR   ---------->  CALCULOS : TERMINADOS


                    0: METER DATOS
                    1: VER MATRICES DEL SISTEMA ACTUAL
                    2: PUNTOS RESPUESTA LIBRE X(t)-X(t)
                    3: PUNTOS RESPUESTA TOTAL X(t)-X(t)
                    4: PUNTOS RESPUESTA TOTAL Y(t)-Y(t)
                    5: GRAFICAS RESPUESTA LIBRE X(t)-X(t)
                    6: GRAFICAS RESPUESTA TOTAL X(t)-X(t)
                    7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(t)
                    8: VERIF.CONTROLABILIDAD DE ESTADO COMPLETA
                    9: VERIF.CONTROLABILIDAD DE SALIDA COMPLETA
                   10: VERIF. OBSERVABILIDAD COMPLETA
                   11: REALIMENTACION DEL ESTADO
                   12: OBSERVADOR
                   13: DISEÑO DEL OBSERVADOR MINIMO DE ESTADO
                   14: EIGENVALORES DEL SISTEMA
                   15: EC. EST.--.SIST. DISEÑADO REALIMENTACION
                   16: VOLVER AL SISTEMA ANTERIOR
                   17: SALIRSE


     Opcion -->  6
```



LAS SIGUIENTES GRAFICAS SON PARECIDAS .

```
          DISEÑO DE REALIMENTACION DE ESTADO
      Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
1
P(REAL) -1  P(IMAG) 4
P(REAL) -1.0000 P(IMAG)  -4.0000000000E+00
DAR RAIZ REAL =  -2
DAR RAIZ REAL =  -1

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
 -6.0779559241E-02  -1.3549591145E-01   2.9125360762E-01  -1.2214355820E-01
 -2.8718341741E+01  -6.4021818159E+01   1.3761734377E+02  -5.7711886247E+01
```

```
DISEÑO DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0

LA MATRIZ   Ke ( K - ESTIMADA )
  1.3889426502E-03  6.5627540224E-01
  9.4927723579E-04  4.4653349391E-01
  1.3581008255E-03  6.4170268729E-01
  4.0959463088E-04  1.9353046209E-01
```

```
 4.6571679621E+00  -2.1229566661E+00
 6.4366187276E-01  -9.3688943831E-01
-9.2149363334E-01   1.0934033953E+00
-1.1277427645E-01   4.2665162289E-01
   Q(2) ES =
-9.8502284060E-01  -3.6576255827E-01
-1.1120305056E-02  -1.2952563634E-01
 1.6677580549E-01   7.8295127401E-02
 1.4272491972E-01   1.4057419490E-01
 PARA EL SISTEMA TRANSFORMADO (su realimentacion es)
          DISEÑO DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
2
P(REAL) 0  P(IMAG) 3
P(REAL) 0.0000 P(IMAG)  -3.0000000000E+00
P(REAL) -1  P(IMAG) 2
P(REAL) -1.0000 P(IMAG)  -2.0000000000E+00

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
 -2.0915249193E+00   1.3767861352E+00   3.5085869724E-01   1.5852556835E-01
 EB.9927BBA4E7E+02TF064BDEDE34ESTADO2   1.6578073445E+02   7.4903331281E+01
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
2
P(REAL) 0  P(IMAG) 3
P(REAL) 0.0000 P(IMAG)  -3.0000000000E+00
P(REAL) -1  P(IMAG) 2
P(REAL) -1.0000 P(IMAG)  -2.0000000000E+00

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
 -2.0915249193E+00   1.3767861352E+00   3.5085869724E-01   1.5852556885E-01
 -9.8824552437E+02   6.5653144887E+01   1.6578073445E+02   7.4903331281E+01


   LA MATRIZ ( estimada ) DEL OBSERVADOR MINIMO ES
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL = 3
DAR RAIZ REAL = 4

LA MATRIZ   Ke ( K - ESTIMADA )
  4.0540429213E-03  1.9244757528E+00
 -4.9927162365E-03 -2.3075594217E+00
```

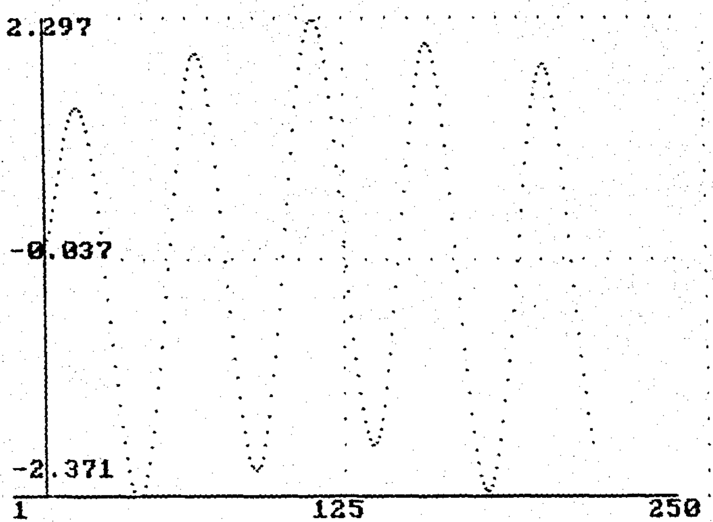
```
DISEÑO DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
    EL SISTEMA ES OBSERVABLE COMPLETAMENTE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
1
P(REAL) -0.1 P(IMAG) 0.2
P(REAL) -0.1000 P(IMAG)  -2.0000000000E-01
DAR RAIZ REAL =  0
DAR RAIZ REAL =  -0.1

LA MATRIZ   Ke ( K - ESTIMADA )
   1.7248669356E-03  8.145562707E-01
   1.0113114165E-02  4.7784464425E-01
   1.6165564355E-03  7.9691209076E-01
   5.3547150013E-04  2.5310185386E-01
```

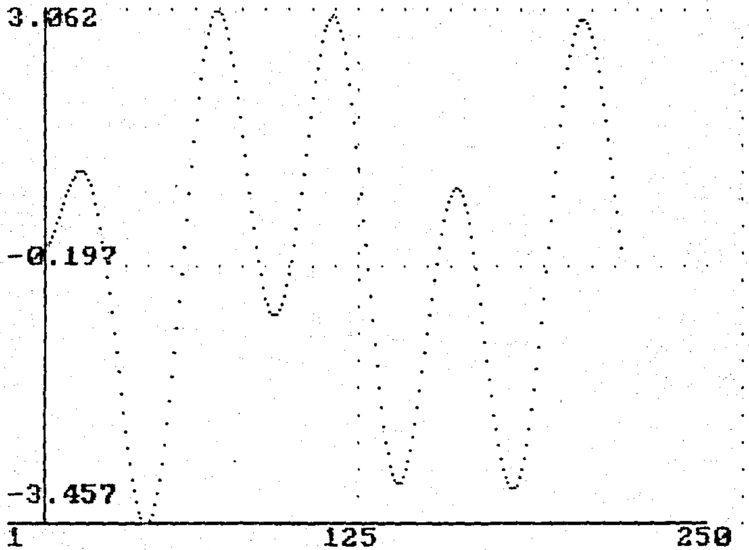
```
          DISE&O DE REALIMENTACION DE ESTADO
      Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
2
P(REAL) 0  P(IMAG) 3
P(REAL) 0.0000 P(IMAG)  -3.0000000000E+00
P(REAL) -1  P(IMAG) 2
P(REAL) -1.0000 P(IMAG)  -2.0000000000E+00

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
 -3.1871043487E-01  -1.8319313150E-01   5.5282628888E-01  -1.7926270681E-01
 -1.5059068048E+02  -8.6558754632E+01   2.6121042150E+02  -8.4229128968E+01
```

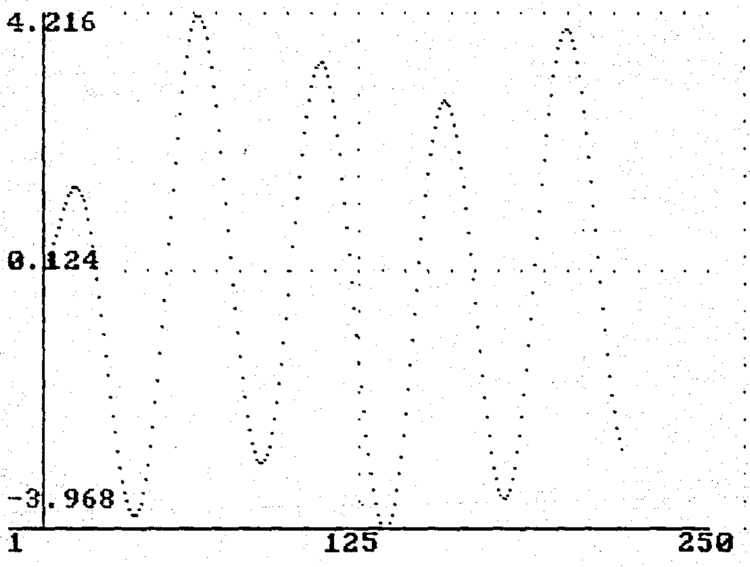
```
       DISE&O DEL SISTEMA REALIMENTADO (listo)
```

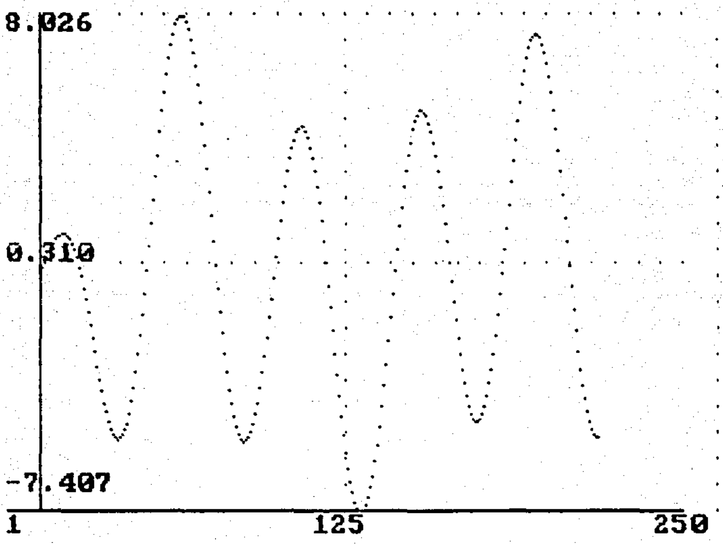
```
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
A
  MATRIZ [A]
  6.0627493452E+01   3.5433417731E+01  -1.0391586437E+02   3.3895609211E+01
  3.1719656031E+01   1.8532296412E+01  -5.4520036401E+01   1.8541595895E+01
  9.1197827982E+01   5.2617689688E+01  -1.5647502473E+02   5.1017695599E+01
  1.3445316980E+02   7.7310568875E+01  -2.3259840565E+02   7.5115229864E+01
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
B
  MATRIZ [B]
  6.0000000000E-01   4.0000000000E-01
  3.0000000000E-01   2.1000000000E-01
  4.5000000000E-01   6.0000000000E-01
  4.0000000000E-01   8.9000000000E-01
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

```
TIEMPO DE INICIO T(0) = 0 -->    --> 0.00E+00
TIEMPO DE RESPUESTA =     0.00E+00   -->  10    --> 1.00E+01

NUMERO DE PUNTOS A CALCULAR  =  250

NUMERO DE TERMINOS DE LA SERIE + 20 =  1

CONDICIONES INICIALES DEL VECTOR  X(T(0))
X[1]  (T(0)) =   0
X[2]  (T(0)) =   0
X[3]  (T(0)) =   0
X[4]  (T(0)) =   0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
U[1]    (t)   =    COS(T)
U[2]    (t)   =    SEN(T/2)
EL INCREMENTO DEL TIEMPO -->     4.0000000000E-02
ESPERE POR FAVOR    --------->  CALCULOS : TERMINADOS


                          0: METER DATOS
                          1: VER MATRICES DEL SISTEMA ACTUAL
                          2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                          3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                          4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                          5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                          6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                          7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                          8: VERIFICACION CONTROLABILIDAD DE ESTADO
                          9: VERIF.CONTROLABILIDAD DE LA SALIDA
                         10: VERIFICACION OBSEVABILIDAD
                         11: REALIMENTACION DEL ESTADO
                         12: OBSERVADOR
                         13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
                         14: VALORES CARACTERISTICOS DEL SISTEMA
                         15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
                         16: VOLVER AL SISTEMA ANTERIOR
                         17: SALIRSE


        Opcion -->  6
```


2.297
-0.037
-2.371
1
125
250


3.062
-0.197
-3.457
1
125
250


4.216
0.124
-3.968
1
125
250



8.026
0.310
-7.407
1
125
250

LIS. RESPUESTA TOTAL -------> X (t) :

0.0000000000E+00	0.0000000000E+00	0.0000000000E+00	0.0000000000E+00
3.5122318941E-02	1.8172945300E-02	3.4255334479E-02	3.8720653220E-02
9.0801954867E-02	4.8509902153E-02	9.8009516490E-02	1.1670403931E-01
1.6472362353E-01	9.0679105874E-02	1.8710460366E-01	2.2547743408E-01
2.5428105512E-01	1.4395692482E-01	2.9701001535E-01	3.5638760890E-01
3.5662765413E-01	2.0725441943E-01	4.2291177479E-01	5.0076552485E-01
4.6672971555E-01	2.7914968267E-01	5.5980669245E-01	6.5008591126E-01
5.8742140813E-01	3.5792530041E-01	7.0259671775E-01	7.9611956659E-01
7.0946071941E-01	4.4161044637E-01	8.4618248700E-01	9.3107635388E-01
8.3158555126E-01	5.2802623178E-01	9.8555479087E-01	1.0477370231E+00
9.5056916059E-01	6.1493456752E-01	1.1158826464E+00	1.1395721758E+00
1.0632741564E+00	6.9958866518E-01	1.2325967465E+00	1.2008468930E+00
1.1667042912E+00	7.7978496422E-01	1.3314671345E+00	1.2267097671E+00
1.2580532239E+00	8.5291542555E-01	1.4086740483E+00	1.2130553188E+00
1.3347502758E+00	9.1651937696E-01	1.4608709861E+00	1.1576290273E+00
1.3945004616E+00	9.6823408276E-01	1.4852391642E+00	1.0579644590E+00
1.4353217394E+00	1.0058432293E+00	1.4795326736E+00	9.1350224353E-01
1.4555754961E+00	1.0273226376E+00	1.4421137779E+00	7.2454090739E-01
1.4539919659E+00	1.0308822860E+00	1.3719779436E+00	4.9242983718E-01
1.4296895595E+00	1.0150043447E+00	1.2697683462E+00	2.1953489941E-01
1.3821879731E+00	9.7847629456E-01	1.1327797484E+00	-9.0812518857E-02
1.3114149351E+00	9.2041884042E-01	9.6495180276E-01	-4.3438202384E-01
1.2177065407E+00	8.4030813059E-01	7.6685198266E-01	-8.0612509912E-01
1.1018012200E+00	7.3799196695E-01		

1.1018012200E+00	7.3799196695E-01	5.4064849646E-01	-1.2002689077E+00
9.6482747288E-01	6.1069977111E-01	2.8707268145E-01	-1.6104214409E+00
8.0828559598E-01	4.6804616978E-01	1.5378510130E-02	-2.0296862619E+00
6.3402371160E-01	3.0202815996E-01	-2.7672103277E-01	-2.4507849785E+00
4.4420845044E-01	1.1701591258E-01	-5.8310483152E-01	-2.8661854714E+00
2.4129103399E-01	-8.5262630290E-02	-8.9931580360E-01	-3.2682338468E+00
2.7968450847E-02	-3.0274311611E-01	-1.2206322599E+00	-3.6492680330E+00
-1.9285827832E-01	-5.3305176201E-01	-1.5421445389E+00	-4.0018509365E+00
-4.1812950104E-01	-7.7353753139E-01	-1.8588341004E+00	-4.3187010924E+00
-6.4467447092E-01	-1.0213090105E+00	-2.1656549082E+00	-4.5930187942E+00
-8.6926058284E-01	-1.2732754048E+00	-2.4576148106E+00	-4.8185057685E+00
-1.0886435242E+00	-1.5261910405E+00	-2.7298563063E+00	-4.9894965660E-00
-1.2996175998E+00	-1.7767026725E+00	-2.9777352625E+00	-5.1010599737E+00
-1.4990654083E+00	-2.0213988834E+00	-3.1968964233E+00	-5.1490889109E+00
-1.6840063394E+00	-2.2568608157E+00	-3.3830445710E+00	-5.1303774503E+00
-1.8516428218E+00	-2.4797134641E+00	-3.5335102704E+00	-5.0426838021E+00
-1.9994039838E+00	-2.6866767445E+00	-3.6443092058E+00	-4.8847783133E+00
-2.1249858730E+00	-2.8746155620E+00	-3.7131942251E+00	-4.6564757632E+00
-2.2263877067E+00	-3.0405891141E+00	-3.7391993095E+00	-4.3595514705E+00
-2.3019436122E+00	-3.1815916922E+00	-3.7179748090E+00	-3.9932409710E+00
-2.3503493848E+00	-3.2961052823E+00	-3.6519134159E+00	-3.5632232712E+00
-2.3706838732E+00	-3.3811283112E+00	-3.5396664952E+00	-3.0725879274E+00
-2.3624246740E+00	-3.4352149446E+00	-3.3821505084E+00	-2.5262864399E+00
-2.3254579077E+00	-3.4570034070E+00	-3.1805434409E+00	-1.9301686862E+00
-2.2600819295E+00	-3.4455398670E+00	-2.	

-2.2600819295E+00	-3.4455398670E+00	-2.9367712822E+00	-1.2909053374E+00
-2.1670049213E+00	-3.4002965135E+00	-2.6533847457E+00	-6.1589740969E-01
-2.0473364001E+00	-3.3211835316E+00	-2.3335265705E+00	8.6825707595E-02
-1.9025727653E+00	-3.2085547795E+00	-1.9808898795E+00	8.0871823426E-01
-1.7345770968E+00	-3.0632070568E+00	-1.5996682020E+00	1.5408383331E+00
-1.5455534972E+00	-2.8863729542E+00	-1.1944978905E+00	2.2739750133E+00
-1.3380163549E+00	-2.6797073632E+00	-7.7039377260E-01	2.9987797232E+00
-1.1147549738E+00	-2.4452678229E+00	-3.3267898065E-01	3.7059006843E+00
-8.7879408757E-01	-2.1854889688E+00	1.1309001757E-01	4.3861179763E+00
-6.3335083347E-01	-1.9031514362E+00	5.6120208800E-01	5.0304773751E+00
-3.8178881326E-01	-1.6013456527E+00	1.0058726499E+00	5.6204209573E+00
-1.2756991080E-01	-1.2834210306E+00	1.4413262574E+00	6.1779125396E+00
1.2579543059E-01	-9.5299113509E-01	1.8616793332E+ 0	6.6655560885E+00
3.7479874641E-01	-6.1378546916E-01	2.2620218447E+00	7.0967053428E+00
6.1598416808E-01	-2.6969855998E-01	2.6364967240E+00	7.4355630185E+00
8.4599838095E-01	7.5312922604E-02	2.9803758707E+00	7.7072681170E+00
1.0616391936E+00	4.1727425788E-01	3.2891316193E+00	7.8979700285E+00
1.2599020112E+00	7.5224719948E-01	3.5587026248E+00	8.0048883184E+00
1.4380235334E+00	1.0763841490E+00	3.7855531968E+00	8.0263572905E+00
1.5935220393E+00	1.3859811978E+00	3.9667252076E+00	7.9618546415E+00
1.7242336698E+00	1.6775292708E+00	4.0998818095E+00	7.8120137565E+00
1.8283441779E+00	1.9477626343E+00	4.1833420127E+00	7.5786194292E+00
1.9044156795E+00	2.1937040644E+00	4.2161077039E+00	7.2645870330E+00
1.9514080129E+00	2.4127060229E+00	4.1978764224E+00	6.8739254131E+00
1.9686943919E+00	2.6024872411E+00	4.1290501487E+00	6.4
⋮	⋮	⋮	⋮
1.8351269944E+00	2.3086648153E+00	3.9694649265E+00	6.5210508369E+00
1.8615981365E+00	2.4944021170E+00	3.9198811139E+00	6.1059737948E+00
1.8584528137E+00	2.6505322986E+00	3.8211473347E+00	5.6240125373E+00
1.8258049943E+00	2.7754937411E+00	3.6748394931E+00	5.0820430860E+00
1.7641973735E+00	2.8681621445E+00	3.4822059666E+00	4.4973726673E+00
1.6745934537E+00	2.9278656187E+00	3.2491345420E+00	3.8481337993E+00
1.5583635674E+00	2.9543932903E+00	2.9761116687E+00	3.1730680010E+00
1.4172650398E+00	2.9479972986E+00	2.6681711300E+00	2.4714007922E+00
1.2534167813E+00	2.9093981551E+00	2.3298363566E+00	1.7527097820E+00
1.0692686712E+00	2.8397235349E+00	1.9660551188E+00	1.0267877423E+00
8.6756617601E-01	2.7405906623E+00	1.5821282359E+00	3.0350263263E-01
6.5131071039E-01	2.6139825453E+00	1.1836332289E+00	-4.0734340974E-01
4.2371631106E-01	2.4622684035E+00	7.7634401292E-01	-1.0961540895E+00
1.9816324763E-01	2.2881587145E+00	3.6614778782E-01	-1.7506759691E+00
-5.1850762903E-02	2.0946653847E+00	-4.1039681458E-02	-2.3711305123E+00
-2.9276103572E-01	1.8850576168E+00	-4.3905917354E-01	-2.9403392587E+00
-5.3098736524E-01	1.6628141113E+00	-8.2309228490E-01	-3.4538403286E+00
-7.6298569983E-01	1.4315722911E+00	-1.1867436577E+00	-3.9049945866E+00
-9.8529930049E-01	1.1950752790E+00	-1.5251199790E+00	-4.2880799365E+00
-1.1946066475E+00	9.5711739351E-01	-1.8334046138E+00	-4.5983723961E+00
-1.3877793769E+00	7.2148894655E-01	-2.1072268011E+00	-4.8322127846E+00
-1.5619075567E+00	4.9192119954E-01	-2.3427244142E+00	-4.9870580678E+00
-1.7143616520E+00	2.7203184980E-01	-2.5365993869E+00	-5.0615166236E+00
-1.8428205747E+00	6.5273089070E-02	-2.6861650050E+00	-5.0553669231E+00

```
 0: METER DATOS
 1: VER MATRICES DEL SISTEMA ACTUAL
 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
 8: VERIFICACION CONTROLABILIDAD DE ESTADO
 9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE

Opcion -->  14


LOS EIGENVALORES DEL SISTEMA SON :

RAIZ(real)  -9.9999997303E-01 RAIZ(imag)   2.0000000511E+00
RAIZ(real)  -9.9999997303E-01 RAIZ(imag)  -2.0000000511E+00
RAIZ(real)  -2.69046722268E-08 RAIZ(imag)   2.9999999624E+00
RAIZ(real)  -2.69046722268E-08 RAIZ(imag)  -2.9999999624E+00
```

```
     ESPACIO  DE  ESTADOS : SISTEMAS MULTIVARIABLES

  N:NUMERO DE VARIABLES DE ESTADO DEL SISTEMA = 6

  R:NUMERO DE ENTRADAS DEL SISTEMA  = 3

  M:NUMERO DE SALIDAS DEL SISTEMA =  2

    SISTEMA T. CONTINUO=C o T.DISCRETO=D   D

DAR LA MATRIZ   [A]
  A[1.1] = .3  A[1,2] = .7  A[1.3] = .9  A[1,4] = 0  A[1.5] = .1  A[1.6] = 0
  A[2,1] = .1  A[2,2] = 0  A[2,3] = 0  A[2,4] = .2  A[2.5] = .5  A[2,6] = .1
  A[3.1] = 0  A[3.2] = 0  A[3.3] = .1  A[3.4] = .4  A[3.5] = .5  A[3.6] = 0
  A[4.1] = .2  A[4.2] = 0  A[4.3] = 0  A[4.4] = .2  A[4.5] = 0  A[4,6] = .1
  A[5.1] = .2  A[5.2] = 0  A[5.3] = .2  A[5.4] = .3  A[5.5] = 0  A[5,6] = .2
  A[6,1] = .3  A[6.2] = .5  A[6,3] = .1  A[6.4] = .1  A[6.5] = 0  A[6.6] = 0

DAR LA MATRIZ   [B]
  B[1,1] =  .3  B[1,2] =  .5  B[1,3] =  .1
  B[2.1] =  .8  B[2.2] =  .098  B[2,3] =  .2
  B[3,1] =  .4  B[3,2] =  .5  B[3,3] =  .6
  B[4,1] =  .3  B[4,2] =  .1  B[4,3] =  .5
  B[5,1] =  .7  B[5,2] =  .8  B[5.3] =  0
  B[6,1] =  0  B[6.2] =  .2  B[6,3] =  .6




VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
C
  MATRIZ [C]
  2.0000000000E-01   4.5000000000E-01   9.0000000000E-02   0.0000000000E+00   1
0000000000E-01  2.0000000000E-01
  3.0000000000E-01   4.0000000000E-01   9.0000000000E-02   0.0000000000E+00   0
0000000000E+00  2.0000000000E-01
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
D
   MATRIZ [D]
  0.0000000000E+00   0.0000000000E+00   0.0000000000E+00
  0.0000000000E+00   0.0000000000E+00   0.0000000000E+00
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

```
          DISE&O DEL OBSERVADOR DE DIMENSION REDUCIDA


 LA MATRIZ DE TRANS Q
 Q(1) ES =
 -1.0934061355E+01   1.0086664045E+01
 -2.2257401479E+01   2.0872478148E+01
  8.0773012938E+01  -7.0469035085E-01
 -6.8408864776E+00   5.2972684037E+00
  1.0194639384E+01  -1.0349575029E+01
  2.4568039168E+01  -2.0163886576E+01
   Q(2) ES =
 -5.3636484611E-03   1.7013883412E-02   7.0858799545E-02  -6.9702721829E-02
  1.0200512958E-01   2.9204757354E-01   1.2016048119E-01  -4.4957150264E-02
 -1.7721876060E-01  -8.7859840001E-01  -4.7244245014E-01   1.9738249948E-01
  1.1458284745E-01   7.4487172310E-02   8.8876893230E-02   3.0195302154E-02
 -5.6366213249E-02  -1.2900990336E-01   1.0778558952E-02  -4.7224146696E-02
 -1.1621634419E-01  -2.1424669219E-01  -1.3401005913E-01   1.0564625851E-01
 PARA EL SISTEMA TRANSFORMADO (su realimentacion es)
          DISE&O DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
```


```
  1.1458284745E-01   7.4487172310E-02   8.8876893230E-02   3.0195302154E-02
 -5.6366213249E-02  -1.2900990336E-01   1.0778558952E-02  -4.7224146696E-02
 -1.1621634419E-01  -2.1424669219E-01  -1.3401005913E-01   1.0564625851E-01
 PARA EL SISTEMA TRANSFORMADO (su realimentacion es)
          DISE&O DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
```

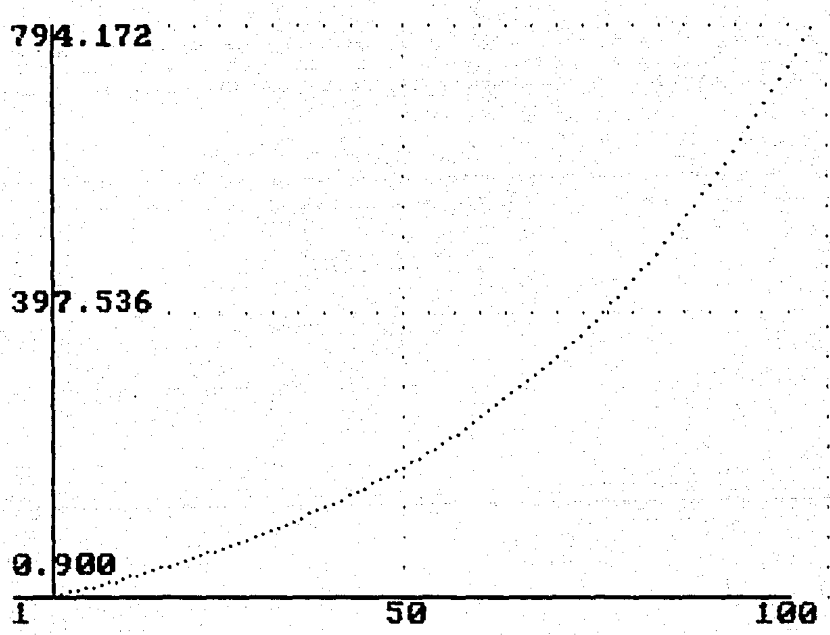

```
2
P(REAL) 0.2  P(IMAG) 0.5
P(REAL) 0.2000 P(IMAG)  -5.0000000000E-01
P(REAL) 0.3  P(IMAG) 0.6
P(REAL) 0.3000 P(IMAG)  -6.0000000000E-01
DAR RAIZ REAL =  0.2
DAR RAIZ REAL =  0.3

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
 -6.5778028767E-01   5.9937036965E-01   3.0879956481E-03   7.5123609047E-03
9453285843E-03  -8.5398772881E-04
 -7.2794351835E+01   6.6330320907E-01   3.4173818506E-01   8.3136794012E-01
2594969666E-01  -9.4507975322E-02
  1.7211917527E+02  -1.5683524672E+02  -8.0802552792E-01  -1.9657344367E+00
7069431290E-01   2.2346012237E-01
```

LA MATRIZ DE GANANCIA DE REALIMENTACION
K DE ENTRADAS MULTIPLES
-6.5778028767E-01   5.9937036965E-01   3.0879956481E-03   7.5123609047E-03   2.
9453285843E-03   -8.5398772881E-04
-7.2794351835E+01   6.6330320907E+01   3.4173818506E-01   8.3136794012E-01   3.
2594969666E-01   -9.4507975322E-02
1.7211917527E+02   -1.5683524672E+02   -8.0802552792E-01   -1.9657344367E+00   -7.
7069431290E-01   2.2346012237E-01

LA MATRIZ K (estimada) DEL OBSERVADOR MINIMO ES
EL SISTEMA ES OBSERVABLE .
CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL = 0
DAR RAIZ REAL = 0
DAR RAIZ REAL = 0
DAR RAIZ REAL = 0

LA MATRIZ Ke (K - ESTIMADA)
1.3470755117E-02 6.3649317926E+00
-3.7881690013E-02 -1.7899098531E+01
3.4829458539E-02 1.6456919160E+01
-3.1013684589E-02 -1.4653965963E+01

```
TIEMPO DE INICIO T(0) = 0 -->    --> 0.00E+00
TIEMPO DE RESPUESTA =     0.00E+00   -->  40    --> 4.00E+01

NUMERO DE PUNTOS A CALCULAR  =  100

NUMERO DE TERMINOS DE LA SERIE + 20 =  1

CONDICIONES INICIALES DEL VECTOR  X(T(0))
X[1]  (T(0))  =  0
X[2]  (T(0))  =  0
X[3]  (T(0))  =  0
X[4]  (T(0))  =  0
X[5]  (T(0))  =  0
X[6]  (T(0))  =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
U[1]    (t)   =   1
U[2]    (t)   =   1
U[3]    (t)   =   1
EL INCREMENTO DEL TIEMPO -->    4.0000000000E-01
ESPERE POR FAVOR   --------->  CALCULOS : TERMINADOS

                     0: METER DATOS
                     1: VER MATRICES DEL SISTEMA ACTUAL
                     2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                     3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                     4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                     5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                     6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                     7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                     8: VERIFICACION CONTROLABILIDAD DE ESTADO
                     9: VERIF.CONTROLABILIDAD DE LA SALIDA
                    10: VERIFICACION OBSEVABILIDAD
                    11: REALIMENTACION DEL ESTADO
                    12: OBSERVADOR
                    13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
                    14: VALORES CARACTERISTICOS DEL SISTEMA
                    15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
                    16: VOLVER AL SISTEMA ANTERIOR
                    17: SALIRSE


     Opcion -->  6
```



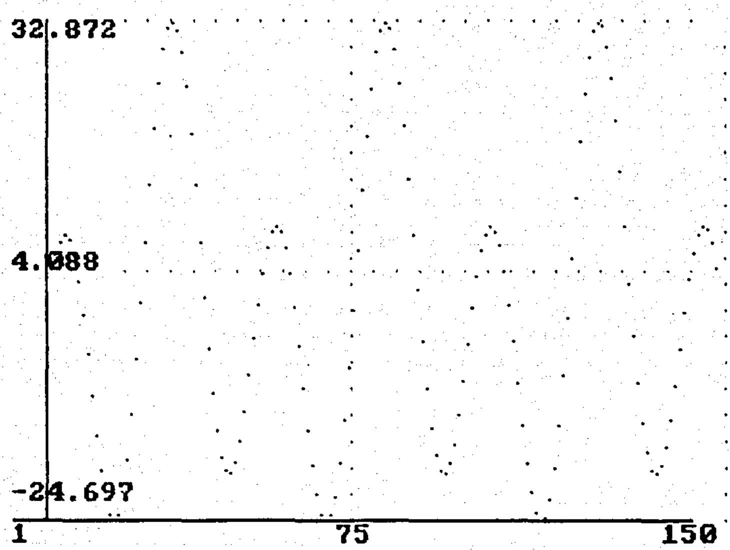
LAS SIGUIENTES GRAFICAS FUERON INESTABLES.

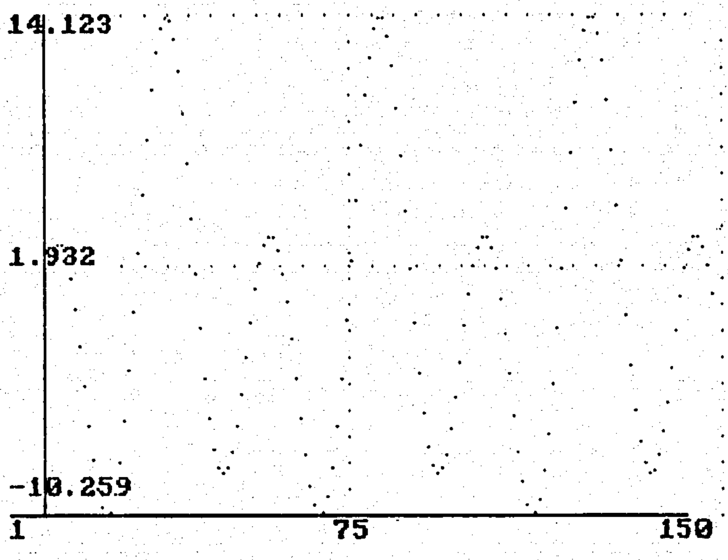
```
DISE&O DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0

LA MATRIZ   Ke ( K - ESTIMADA )
  3.1188259367E-03  1.4736452551E+00
  2.9428577167E-04  1.3905002711E-01
 -8.6360418645E-04 -4.0805297810E-01
  1.1660786674E-03  5.5097217037E-01
  1.5023667610E-03  7.0986829456E-01
  1.4591602201E-03  6.8945320400E-01
```

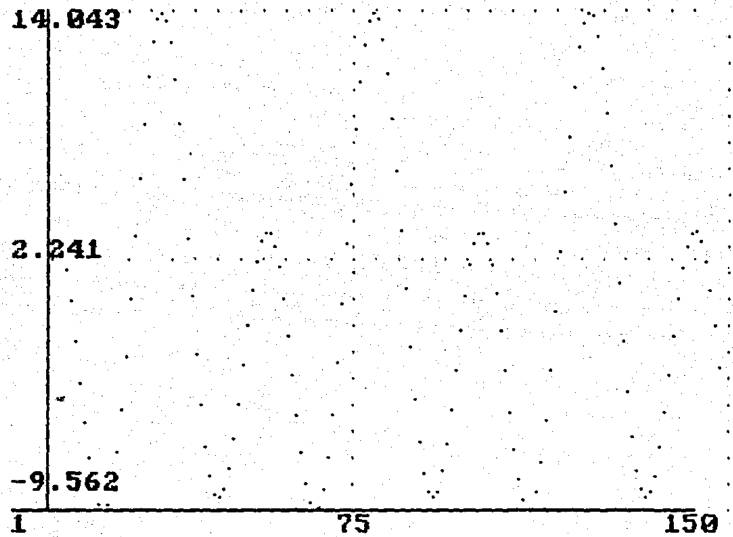
```
DISE&O DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
2
P(REAL) .02  P(IMAG) 0.1
P(REAL) 0.0200 P(IMAG)  -1.0000000000E-01
P(REAL) 0.03  P(IMAG) 0.02
P(REAL) 0.0300 P(IMAG)  -2.0000000000E-02
DAR RAIZ REAL =  0.02
DAR RAIZ REAL =  0.01

LA MATRIZ   Ke ( K - ESTIMADA )
  1.7101175262E-03  8.0803053115E-01
  7.3983965005E-04  3.4957423465E-01
 -1.0132509682E-03 -4.7876108249E-01
  1.6093327693E-03  7.6040973349E-01
  1.4631058593E-03  6.9131751852E-01
  1.3740187278E-03  6.4922384888E-01
```

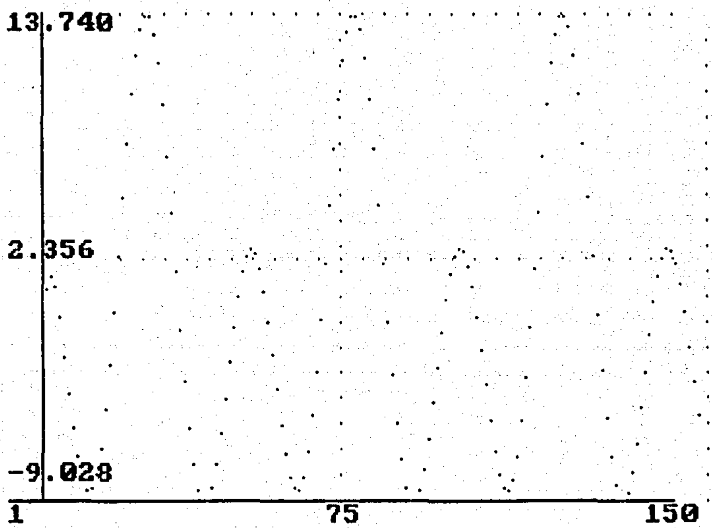
```
          DISEÑO DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
2
P(REAL) .2  P(IMAG) .5
P(REAL) 0.2000 P(IMAG)  -5.0000000000E-01
P(REAL) .3  P(IMAG) .4
P(REAL) 0.3000 P(IMAG)  -4.0000000000E-01
DAR RAIZ REAL =   .4
DAR RAIZ REAL =   .8

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
  3.6579234096E-03   2.3655462085E-02  -2.5031657755E-03   9.1042300523E-03  -1.
7642674639E-02  -1.4098468222E-02
  4.0481019067E-01   2.6178711374E+00  -2.7701701249E-01   1.0075347925E+00  -1.
9524559934E+00  -1.5602304832E+00
 -9.5715662552E-01  -6.1898459123E+00   6.5499504459E-01  -2.3822735303E+00   4.
6164998638E+00   3.6890991847E+00




      DISEÑO DEL SISTEMA REALIMENTADO (listo)



VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
A
  MATRIZ [A]
  1.9221319020E-01   2.9523838866E-03   9.7375995152E-01  -2.6827131221E-01   6.
1987081269E-01   4.1543486361E-01
  2.4883358769E-01   9.6249344132E-01  -1.0184880907E-01   5.7043291237E-01  -2.
1784514570E-01  -4.7363847501E-01
  3.7042571061E-01   2.3955097936E+00  -1.5348725420E-01   1.3219550300E+00  -1.
2866148517E+00  -1.4277048819E+00
  7.3271557922E-01   3.4450237950E+00  -3.6454437577E-01   1.5258793699E+00  -2.
5693615166E+00  -1.9532069221E+00
 -1.2640869892E-01  -2.1108557334E+00   4.2336582603E-01  -5.1240079500E-01   1.
5743146669E+00   1.4580533143E+00
  7.9331937168E-01   3.6903332199E+00  -2.3759362426E-01   1.3278571597E+00  -2.
3794087196E+00  -1.9014134142E+00
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

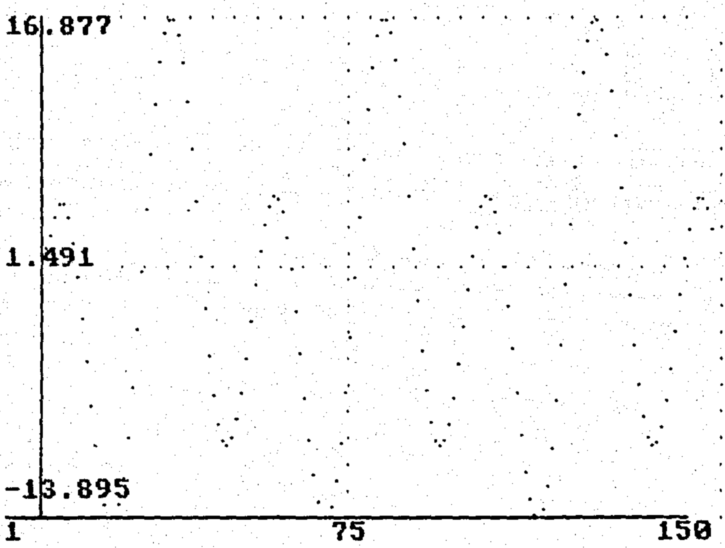
```
TIEMPO DE INICIO T(0) = 0 -->   --> 0.00E+00
TIEMPO DE RESPUESTA =    0.00E+00   -->  40   --> 4.00E+01

NUMERO DE PUNTOS A CALCULAR  =  150

NUMERO DE TERMINOS DE LA SERIE + 20 =  1

CONDICIONES INICIALES DEL VECTOR  X(T(0))
X[1] (T(0)) =  0
X[2] (T(0)) =  0
X[3] (T(0)) =  0
X[4] (T(0)) =  0
X[5] (T(0)) =  0
X[6] (T(0)) =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
  ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
  U[1]    (t)   =    COS(T)
  U[2]    (t)   =    COS(T/2)
  U[3]    (t)   =    SEN(T)
 EL INCREMENTO DEL TIEMPO -->     2.6666666667E-01
 ESPERE POR FAVOR    --------->  CALCULOS : TERMINADOS

                 0: METER DATOS
                 1: VER MATRICES DEL SISTEMA ACTUAL
                 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                 8: VERIFICACION CONTROLABILIDAD DE ESTADO
                 9: VERIF.CONTROLABILIDAD DE LA SALIDA
                10: VERIFICACION OBSEVABILIDAD
                11: REALIMENTACION DEL ESTADO
                12: OBSERVADOR
                13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
                14: VALORES CARACTERISTICOS DEL SISTEMA
                15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
                16: VOLVER AL SISTEMA ANTERIOR
                17: SALIRSE


     Opcion -->  6
```

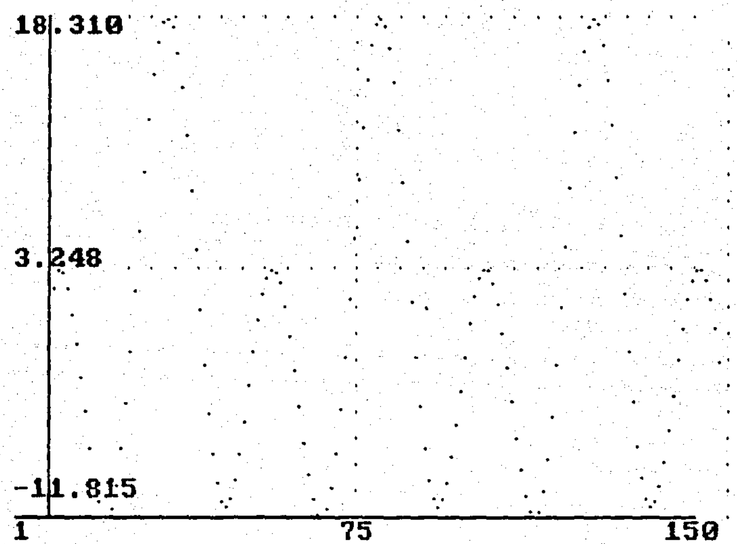

32.872
4.088
-24.697
1
75
150


14.123
1.932
-10.259
1
75
150


14.043
2.241
-9.562
1
75
150


13.740
2.356
-9.028
1
75
150


16.877
1.491
-13.895
1
75
150


18.310
3.248
-11.815
1
75
150

```
ESPACIO  DE  ESTADOS : SISTEMAS MULTIVARIABLES

N:NUMERO DE VARIABLES DE ESTADO DEL SISTEMA = 4

R:NUMERO DE ENTRADAS DEL SISTEMA  = 4

M:NUMERO DE SALIDAS DEL SISTEMA =  2

  SISTEMA T. CONTINUO=C o T.DISCRETO=D   C

DAR LA MATRIZ   [A]
  A[1,1] = .2  A[1,2] = .5  A[1,3] = 1  A[1,4] = 0
  A[2,1] = 2  A[2,2] = .98  A[2,3] = .3  A[2,4] = .67
  A[3,1] = .32  A[3,2] = 1  A[3,3] = 3  A[3,4] = 2
  A[4,1] = .6  A[4,2] = .8  A[4,3] = 1.2  A[4,4] = 1.6

DAR LA MATRIZ   [B]
  B[1,1] =  .3  B[1,2] =  .6  B[1,3] =  1  B[1,4] =  2
  B[2,1] =  .4  B[2,2] =  .9  B[2,3] =  0  B[2,4] =  .2
  B[3,1] =  .3  B[3,2] =  1.5  B[3,3] =  1.23  B[3,4] =  2
  B[4,1] =  .21  B[4,2] =  1.43  B[4,3] =  1.2  B[4,4] =  0

DAR LA MATRIZ  [C]
  C[1,1]  =

VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
C
  MATRIZ [C]
  1.0000000000E+00   4.0000000000E-01   6.0000000000E-01   9.0000000000E-01
  3.0000000000E-01   5.0000000000E-01   6.0000000000E-01   2.0000000000E+00
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
D
   MATRIZ [D]
  0.0000000000E+00   0.0000000000E+00   0.0000000000E+00   0.0000000000E+00
  0.0000000000E+00   0.0000000000E+00   0.0000000000E+00   0.0000000000E+00
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

```
-1.3922790162E-02 -6.5785183516E+00
```

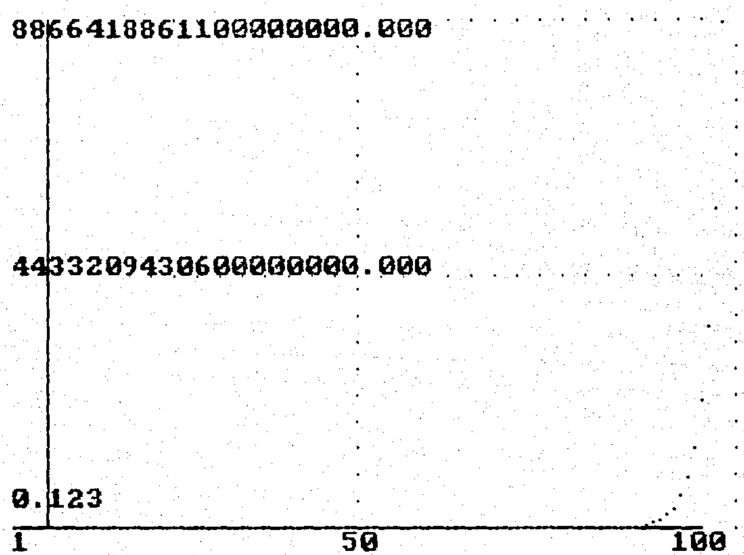
DISE&O DEL OBSERVADOR DE DIMENSION REDUCIDA

```
 LA MATRIZ DE TRANS Q
 Q(1) ES =
  1.2542084002E+00  -7.2943513410E-01
  3.6905664239E-01  -9.8731712439E-01
 -4.5295508671E-01   1.0719451144E+00
 -1.4450889462E-01   5.3466101688E-01
   Q(2) ES =
  2.7432653053E-02   5.2429686066E-02
  3.4580559319E-01   1.1223464855E-01
 -2.5528855666E-01  -1.9694770286E-01
 -1.3979729256E-02   2.3161195811E-02
 PARA EL SISTEMA TRANSFORMADO (su realimentacion es)
          DISE&O DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE


1
P(REAL) -2  P(IMAG) 3
P(REAL) -2.0000 P(IMAG)  -3.0000000000E+00
DAR RAIZ REAL =  -2
DAR RAIZ REAL =  -4

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
  8.9594522755E-01  -5.1415124205E-01   7.5090373671E-02   8.0463925857E-02
  1.5813433266E+02  -9.0747694221E+01   1.3253450953E+01   1.4201889214E+01
  2.7550315747E+01  -1.5810150693E+01   2.3090289904E+00   2.4742657201E+00
 -1.3080800322E+02   7.5066081359E+01  -1.0963194556E+01  -1.1747733175E+01


   LA MATRIZ K ( estimada ) DEL OBSERVADOR MINIMO ES
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0

LA MATRIZ   Ke ( K - ESTIMADA )
  2.7545266662E-03  1.3015138498E+00
 -1.3922790162E-02 -6.5785183516E+00
```

```
LOS EIGENVALORES DEL SISTEMA SON :

RAIZ(real)    4.5389791168E+00 RAIZ(imag)    0.0000000000E+00
RAIZ(real)   -3.5260260924E-01 RAIZ(imag)    0.0000000000E+00
RAIZ(real)    7.9681174614E-01 RAIZ(imag)    4.0254064402E-01
RAIZ(real)    7.9681174614E-01 RAIZ(imag)   -4.0254064402E-01




 TIEMPO DE INICIO T(0) = 0 -->     --> 0.00E+00
 TIEMPO DE RESPUESTA =      0.00E+00    -->  10    --> 1.00E+01

 NUMERO DE PUNTOS A CALCULAR  =  100

 NUMERO DE TERMINOS DE LA SERIE + 20 =  1

 CONDICIONES INICIALES DEL VECTOR  X(T(0))
  X[1] (T(0)) =  0
  X[2] (T(0)) =  0
  X[3] (T(0)) =  0
  X[4] (T(0)) =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
 ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
 U[1]    (t)   =   COS(T)
 U[2]    (t)   =   COS(T/2)
 U[3]    (t)   =   SEN(T)
 U[4]    (t)   =   SEN(T/3)
EL INCREMENTO DEL TIEMPO -->     1.0000000000E-01
ESPERE POR FAVOR   --------->  CALCULOS : TERMINADOS

                         0: METER DATOS
                         1: VER MATRICES DEL SISTEMA ACTUAL
                         2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                         3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                         4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                         5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                         6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                         7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                         8: VERIFICACION CONTROLABILIDAD DE ESTADO
                         9: VERIF.CONTROLABILIDAD DE LA SALIDA
                        10: VERIFICACION OBSEVABILIDAD
                        11: REALIMENTACION DEL ESTADO
                        12: OBSERVADOR
                        13: DISE%O DEL OBSERVADOR MINIMO DE ESTADO
                        14: VALORES CARACTERISTICOS DEL SISTEMA
                        15: EC. EST.-->SIST. DISE%ADO REALIMENTACION
                        16: VOLVER AL SISTEMA ANTERIOR
                        17: SALIRSE
```



```
      Opcion -->  6
```

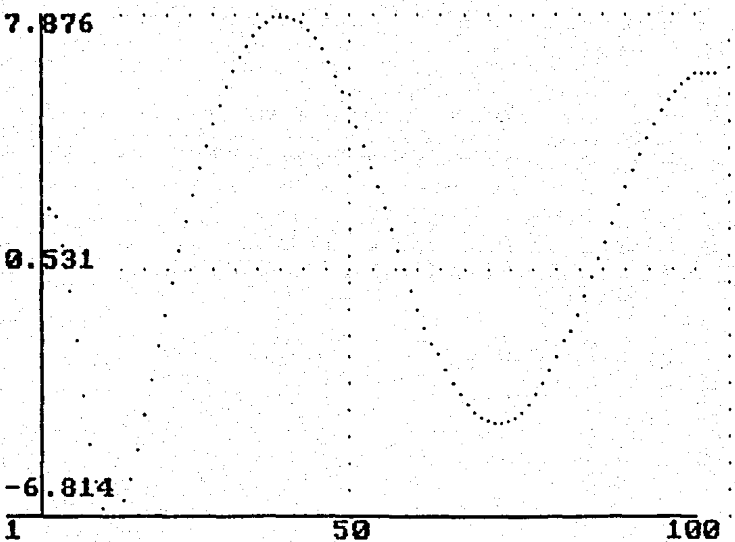

88664188611000000.000
44332094306000000.000
0.123
1
50
100

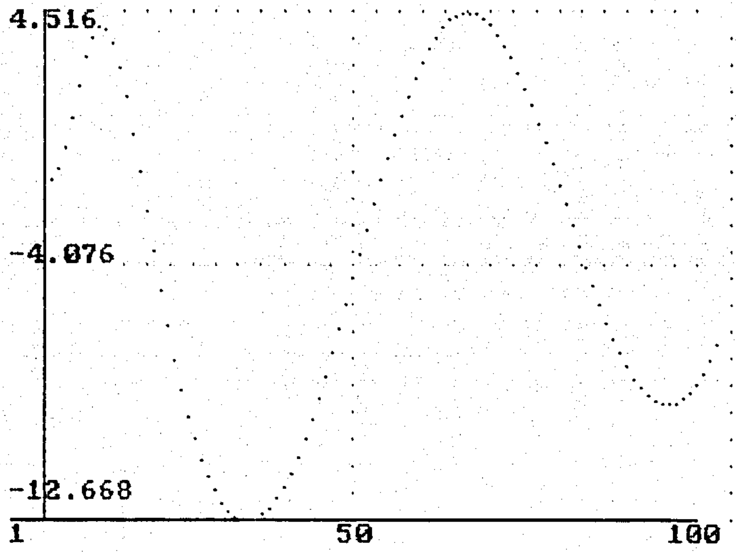
```
DISE&O DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0

LA MATRIZ   Ke ( K - ESTIMADA )
  1.9448305378E-03  9.1693242912E-01
  1.7643673532E-03  8.3366357437E-01
  7.1123980926E-03  3.3606080987E+00
  3.2394604145E-03  1.5306450458E+00



          DISE&O DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
1
P(REAL) -2  P(IMAG) 3
P(REAL) -2.0000 P(IMAG)  -3.0000000000E+00
DAR RAIZ REAL =  -2.5
DAR RAIZ REAL =  -3

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
  6.4762358209E-01   7.3786730515E-02  -1.2433009308E-01   3.7115024280E-01
  1.1430556224E+02   1.3023357936E+01  -2.1944261429E+01   6.5508017854E+01
  1.9914425149E+01   2.2689419633E+00  -3.8231503623E+00   1.1412869966E+01
 -9.4553042986E+01  -1.0772862655E+01   1.8152193590E+01  -5.4187935446E+01



      DISE&O DEL SISTEMA REALIMENTADO (listo)
```

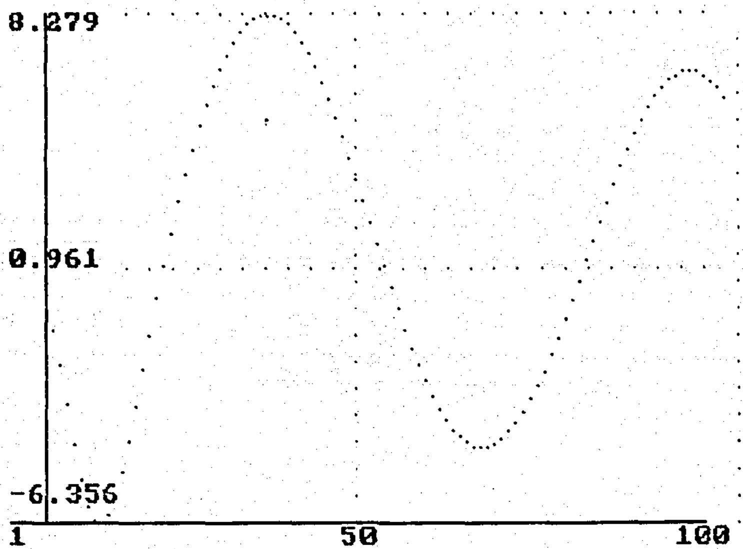


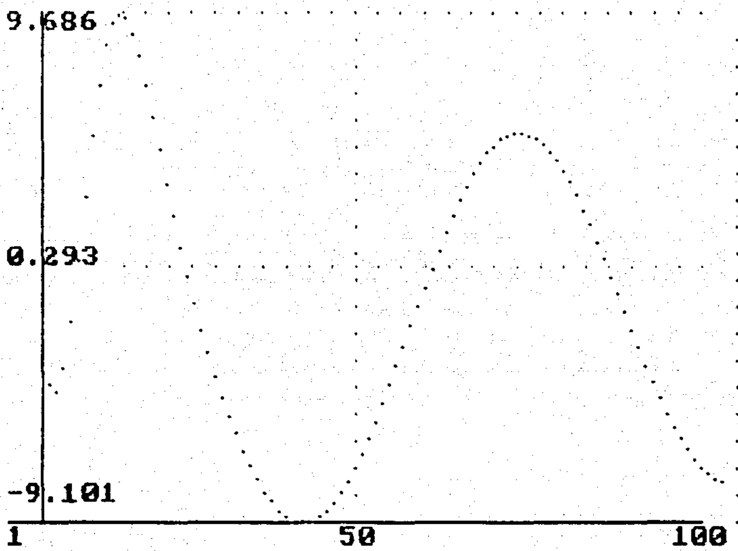
```
     LOS EIGENVALORES DEL SISTEMA SON :

 RAIZ(real)  -2.4999976061E+00 RAIZ(imag)   0.0000000000E+00
 RAIZ(real)  -3.0000021095E+00 RAIZ(imag)   0.0000000000E+00
 RAIZ(real)  -2.0000000438E+00 RAIZ(imag)   3.0000001633E+00
 RAIZ(real)  -2.0000000438E+00 RAIZ(imag)  -3.0000001633E+00
```

```
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
A
  MATRIZ [A]
  1.0061403640E+02   1.1940632566E+01  -1.8277380933E+01   5.7546845146E+01
 -8.2223446851E+01  -8.6159643034E+00   1.6469128605E+01  -4.7598089076E+01
 -6.7212873961E+00   1.9775377252E-01   4.3517789370E+00  -2.0353310145E+00
 -1.8689026513E+02  -2.0561627418E+01   3.7194183598E+01  -1.0584985104E+02
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]


  TIEMPO DE INICIO T(0) = 0 -->    --> 0.00E+00
  TIEMPO DE RESPUESTA =    0.00E+00   -->  10   --> 1.00E+01

  NUMERO DE PUNTOS A CALCULAR  =  100

  NUMERO DE TERMINOS DE LA SERIE + 20 =  1

  CONDICIONES INICIALES DEL VECTOR  X(T(0))
  X[1]  (T(0)) =  0
  X[2]  (T(0)) =  0
  X[3]  (T(0)) =  0
  X[4]  (T(0)) =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
  ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
  U[1]    (t)   =    COS(T)
  U[2]    (t)   =    COS(T/2)
  U[3]    (t)   =    SEN(T)
  U[4]    (t)   =    SEN(T/3)
 EL INCREMENTO DEL TIEMPO -->     1.0000000000E-01
 ESPERE POR FAVOR    --------->  CALCULOS : TERMINADOS
                                 0: METER DATOS
                                 1: VER MATRICES DEL SISTEMA ACTUAL
                                 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                                 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                                 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                                 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                                 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                                 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                                 8: VERIFICACION CONTROLABILIDAD DE ESTADO
                                 9: VERIF.CONTROLABILIDAD DE LA SALIDA
                                10: VERIFICACION OBSEVABILIDAD
                                11: REALIMENTACION DEL ESTADO
                                12: OBSERVADOR
                                13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
                                14: VALORES CARACTERISTICOS DEL SISTEMA
                                15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
                                16: VOLVER AL SISTEMA ANTERIOR
                                17: SALIRSE


         Opcion -->  6
```

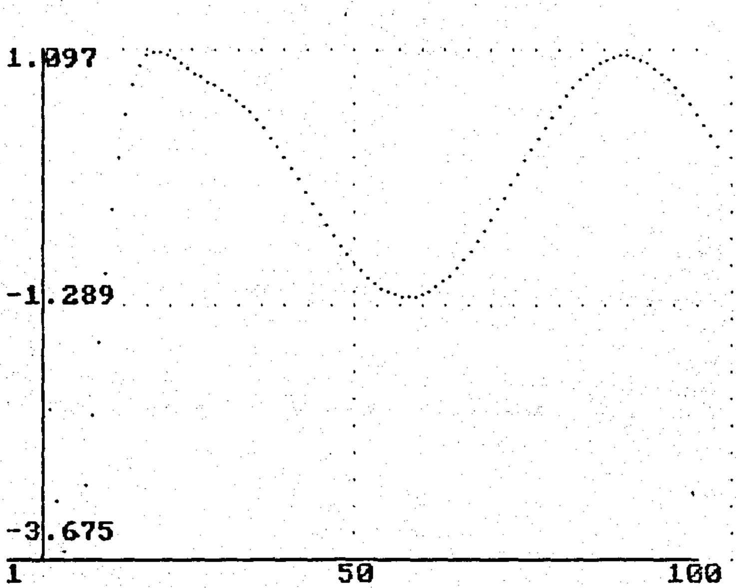

7.876
0.531
-6.814
1
50
100

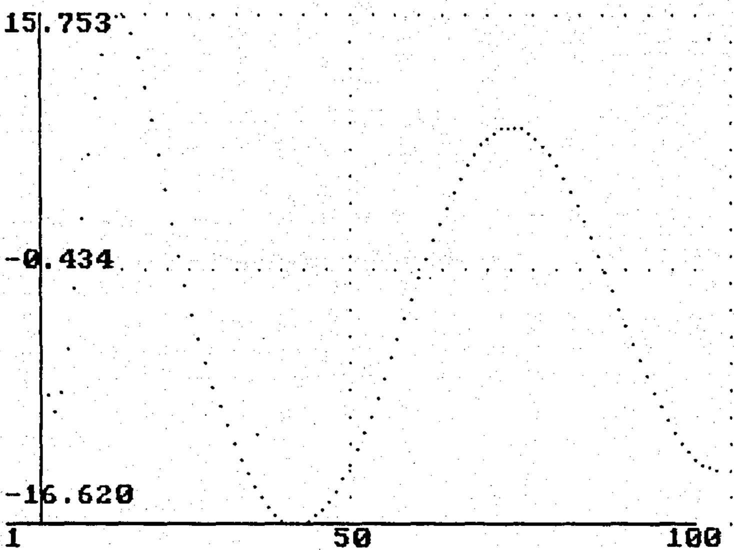

4.516
-4.076
-12.668
1
50
100


8.279
0.961
-6.356
1
50
100


9.686
0.293
-9.101
1
50
100

```
 0: METER DATOS
 1: VER MATRICES DEL SISTEMA ACTUAL
 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
 8: VERIFICACION CONTROLABILIDAD DE ESTADO
 9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISEÑO DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.--->SIST. DISEÑADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE


Opcion -->  7
```


1.097
-1.289
-3.675
1
50
100


15.753
-0.434
-16.620
1
50
100

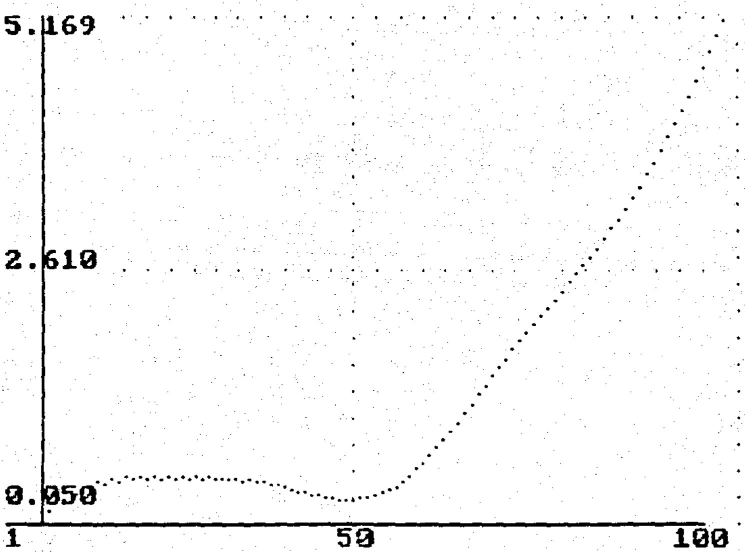
```
    ESPACIO DE ESTADOS : SISTEMAS MULTIVARIABLES

  N:NUMERO DE VARIABLES DE ESTADO DEL SISTEMA = 4

  R:NUMERO DE ENTRADAS DEL SISTEMA  = 2

  M:NUMERO DE SALIDAS DEL SISTEMA =  4

   SISTEMA T. CONTINUO=C o T.DISCRETO=D   D

DAR LA MATRIZ   [A]
  A[1,1] = -0.5  A[1,2] = 0.5  A[1,3] = 0  A[1,4] = 0
  A[2,1] = 0.6  A[2,2] = -0.3  A[2,3] = 0.1  A[2,4] = 0
  A[3,1] = 0.2  A[3,2] = 0  A[3,3] = 0.6  A[3,4] = 0.9
  A[4,1] = 0.1  A[4,2] = 0  A[4,3] = .5  A[4,4] = 0

DAR LA MATRIZ   [B]
  B[1,1] =  0  B[1,2] =  .5
  B[2,1] =  .2  B[2,2] =  0
  B[3,1] =  0.3  B[3,2] =  0
  B[4,1] =  0.3  B[4,2] =  0.5

DAR LA MATRIZ  [C]
  C[1,1]  =

VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
C
  MATRIZ [C]
  5.0000000000E-01   2.0000000000E-01   4.0000000000E-01   1.0000000000E-01
  0.0000000000E+00   0.0000000000E+00   1.0000000000E-01   3.0000000000E-01
  1.0000000000E-01   0.0000000000E+00   0.0000000000E+00   2.0000000000E-01
  2.0000000000E-01   0.0000000000E+00   3.0000000000E-01   0.0000000000E+00
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
D
   MATRIZ [D]
  2.0000000000E-01   4.0000000000E-01
  5.0000000000E-01   1.0000000000E-01
  3.0000000000E-01   4.0000000000E-02
  3.0000000000E-01   0.0000000000E+00
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

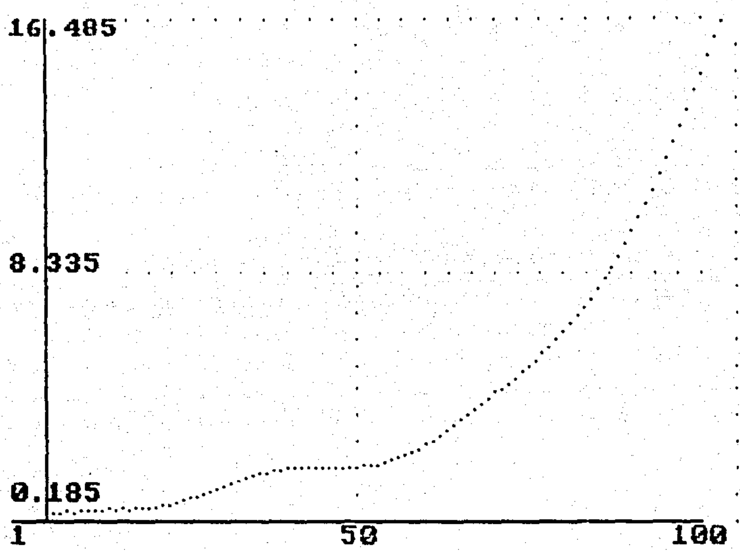
```
 0: METER DATOS
 1: VER MATRICES DEL SISTEMA ACTUAL
 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
 8: VERIFICACION CONTROLABILIDAD DE ESTADO
 9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISEÑO DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.-->SIST. DISEÑADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE
```

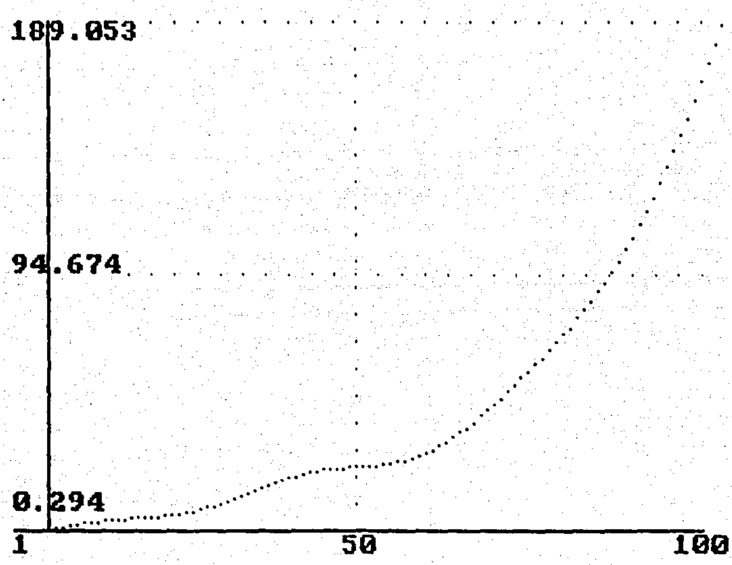
```
Opcion -->  14
```

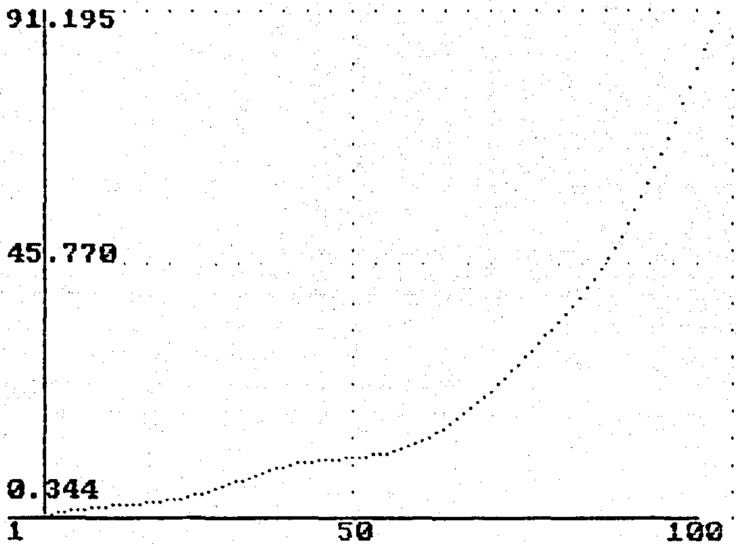
```
LOS EIGENVALORES DEL SISTEMA SON :

RAIZ(real)   1.4630771881E-01 RAIZ(imag)   0.0000000000E+00
RAIZ(real)  -4.3450544487E-01 RAIZ(imag)   0.0000000000E+00
RAIZ(real)   1.0405700494E+00 RAIZ(imag)   0.0000000000E+00
RAIZ(real)  -9.5237232338E-01 RAIZ(imag)   0.0000000000E+00
```

```
TIEMPO DE INICIO T(0) = 0 -->    --> 0.00E+00
TIEMPO DE RESPUESTA =     0.00E+00   -->  20   --> 2.00E+01

NUMERO DE PUNTOS A CALCULAR  =  100

NUMERO DE TERMINOS DE LA SERIE + 20 =  1

CONDICIONES INICIALES DEL VECTOR  X(T(0))
X[1] (T(0)) =  0
X[2] (T(0)) =  0
X[3] (T(0)) =  0
X[4] (T(0)) =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
 ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
 U[1]    (t)   =    COS(T)
 U[2]    (t)   =    SEN(T/2)
EL INCREMENTO DEL TIEMPO -->     2.0000000000E-01
ESPERE POR FAVOR    --------->  CALCULOS : TERMINADOS


                         0: METER DATOS
                         1: VER MATRICES DEL SISTEMA ACTUAL
                         2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                         3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                         4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                         5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                         6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                         7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                         8: VERIFICACION CONTROLABILIDAD DE ESTADO
                         9: VERIF.CONTROLABILIDAD DE LA SALIDA
                        10: VERIFICACION OBSEVABILIDAD
                        11: REALIMENTACION DEL ESTADO
                        12: OBSERVADOR
                        13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
                        14: VALORES CARACTERISTICOS DEL SISTEMA
                        15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
                        16: VOLVER AL SISTEMA ANTERIOR
                        17: SALIRSE


      Opcion -->  6
```


5.169
2.610
0.050
1
50
100


16.485
8.335
0.185
1
50
100


189.053
94.674
0.294
1
50
100


91.195
45.770
0.344
1
50
100

```
DISE&O DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0
DAR RAIZ REAL =  0

LA MATRIZ   Ke ( K - ESTIMADA )
  3.3467631053E-02  5.9070368809E+00  1.0291296549E+00 -4.8862741337E+00
 -6.6243989283E-03 -1.1692064109E+00 -2.0370026705E-01  9.6716224354E-01
  2.0395089954E-01  3.5997333769E+01  6.2714901609E+00 -2.9776831333E+01
  9.9842374610E-02  1.7622179154E+01  3.0701530254E+00 -1.4576986722E+01


          DISE&O DE REALIMENTACION DE ESTADO
     Asignacion de polos
 EL SISTEMA ES CONTROLABLE DE ESTADO
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
1
P(REAL) 0.3  P(IMAG) 0.5
P(REAL) 0.3000 P(IMAG)  -5.0000000000E-01
DAR RAIZ REAL =  0.2
DAR RAIZ REAL =  0.1

 LA MATRIZ DE GANANCIA DE REALIMENTACION
  K DE ENTRADAS MULTIPLES
 -8.6159884001E-03  6.8363664468E-03  -9.2065508248E-04  3.9502692148E-03
 -4.0710545190E+00  3.2301831461E+00  -4.3500952647E-01  1.8665022040E+00


DISE&O DEL OBSERVADOR DE ESTADO DE DIMENCION COMPLETA
   EL SISTEMA ES OBSERVABLE
 CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
2
P(REAL) 0.5  P(IMAG) 0.5
P(REAL) 0.5000 P(IMAG)  -5.0000000000E-01
P(REAL) 0.2  P(IMAG) 0.3
P(REAL) 0.2000 P(IMAG)  -3.0000000000E-01

LA MATRIZ   Ke ( K - ESTIMADA )
  1.1831218266E-01  2.0882100240E+01  3.6380996169E+00 -1.7273578669E+01
 -9.0796793161E-02 -1.6025633993E+01 -2.7920013897E+00  1.3256331802E+01
  5.0772192695E-02  8.9612920106E+00  1.5612449254E+00 -7.4127401334E+00
  4.6575895291E-02  8.2206455139E+00  1.4322097802E+00 -6.8000807125E+00
```

```
0: METER DATOS
1: VER MATRICES DEL SISTEMA ACTUAL
2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
8: VERIFICACION CONTROLABILIDAD DE ESTADO
9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE


Opcion -->  11


DISE&O DE REALIMENTACION DE ESTADO
Asignacion de polos
EL SISTEMA ES CONTROLABLE DE ESTADO
CUANTAS RAICES PARES COMPLEJAS CONJUGADAS QUIERE
1
P(REAL) 0.5  P(IMAG) 0.5
P(REAL) 0.5000 P(IMAG)  -5.0000000000E-01
DAR RAIZ REAL =  -0.2
DAR RAIZ REAL =  -0.6

LA MATRIZ DE GANANCIA DE REALIMENTACION
K DE ENTRADAS MULTIPLES
-3.0416756675E-03   1.0545427981E-03   1.6886865670E-03   1.3438104499E-03
-1.4371917529E+00   4.9827147208E-01   7.9790440290E-01   6.3495043758E-01
```

```
 0: METER DATOS
 1: VER MATRICES DEL SISTEMA ACTUAL
 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
 8: VERIFICACION CONTROLABILIDAD DE ESTADO
 9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE
```

```
Opcion -->  15
```

```
DISE&O DEL SISTEMA REALIMENTADO (listo)
```

```
 0: METER DATOS
 1: VER MATRICES DEL SISTEMA ACTUAL
 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
 8: VERIFICACION CONTROLABILIDAD DE ESTADO
 9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE
```

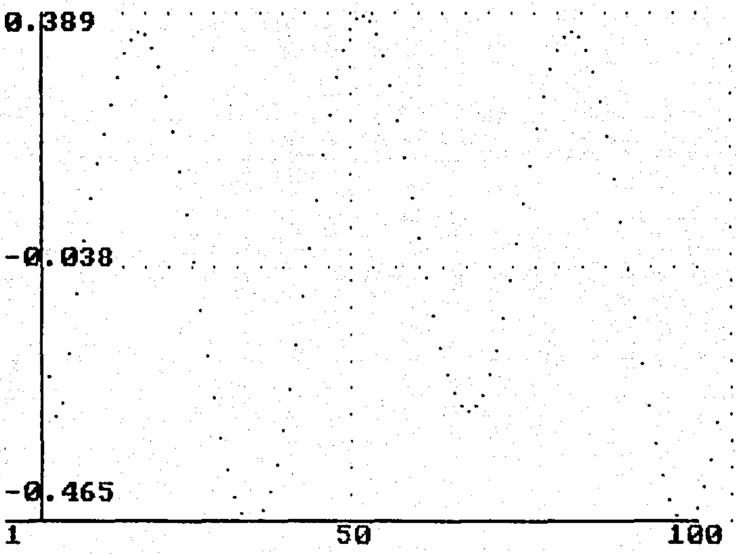


```
Opcion -->  1
```

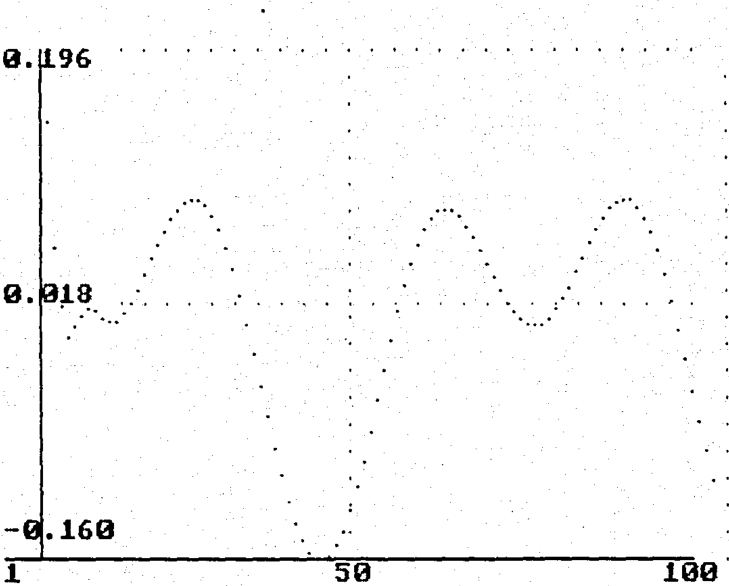
```
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
A
  MATRIZ [A]
  2.1859587646E-01   2.5086426396E-01  -3.9895220145E-01  -3.1747521879E-01
  6.0060833513E-01  -3.0021090856E-01   9.9662262686E-02  -2.6876208998E-04
  2.0091250270E-01  -3.1636283942E-04   5.9949339403E-01   8.9959685686E-01
  8.1950837916E-01  -2.4945209888E-01   1.0054119258E-01  -3.1787836192E-01
VER MATRIZ  [A] - [B] - [C] - [D] - [N-No]
```

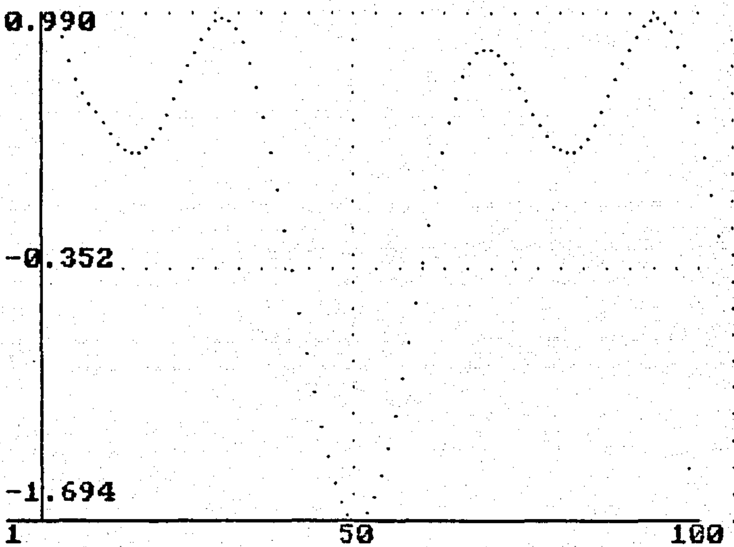
```
 0: METER DATOS
 1: VER MATRICES DEL SISTEMA ACTUAL
 2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
 3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
 4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
 5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
 6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
 7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
 8: VERIFICACION CONTROLABILIDAD DE ESTADO
 9: VERIF.CONTROLABILIDAD DE LA SALIDA
10: VERIFICACION OBSEVABILIDAD
11: REALIMENTACION DEL ESTADO
12: OBSERVADOR
13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
14: VALORES CARACTERISTICOS DEL SISTEMA
15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
16: VOLVER AL SISTEMA ANTERIOR
17: SALIRSE


      Opcion -->  14


   LOS EIGENVALORES DEL SISTEMA SON :

RAIZ(real)  -1.9999999958E-01 RAIZ(imag)   0.0000000000E+00
RAIZ(real)  -6.0000000021E-01 RAIZ(imag)   0.0000000000E+00
RAIZ(real)   5.0000000018E-01 RAIZ(imag)   5.0000000010E-01
RAIZ(real)   5.0000000018E-01 RAIZ(imag)  -5.0000000010E-01
```

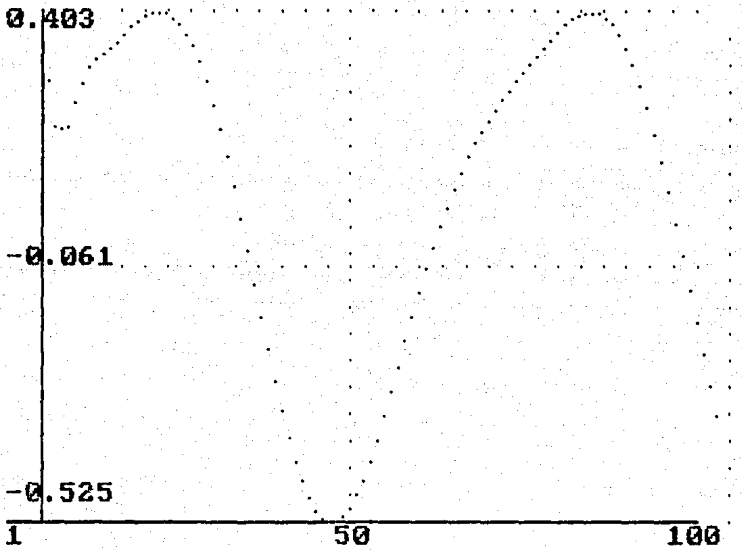
```
  TIEMPO DE INICIO T(0) = 0 -->    --> 0.00E+00
  TIEMPO DE RESPUESTA =     0.00E+00   -->  20   --> 2.00E+01

  NUMERO DE PUNTOS A CALCULAR  =  100

  NUMERO DE TERMINOS DE LA SERIE + 20 =  20

  CONDICIONES INICIALES DEL VECTOR  X(T(0))
  X[1] (T(0)) =  0
  X[2] (T(0)) =  0
  X[3] (T(0)) =  0
  X[4] (T(0)) =  0
DAR LAS ENTRADAS AL SISTEMA EN UNA EXPRESION MATEMATICA
SEN(X)-COS(X)-EXP(X):+:-:*:/:^:Donde X = FUNCION
  ESCALON = CONSTANTE:RAMPA = T: TIEMPO = T o var.cualq.
  U[1]    (t)   =   COS(T)
  U[2]    (t)   =   SEN(T/2)
 EL INCREMENTO DEL TIEMPO -->     2.0000000000E-01
 ESPERE POR FAVOR    --------->  CALCULOS : TERMINADOS


                                  0: METER DATOS
                                  1: VER MATRICES DEL SISTEMA ACTUAL
                                  2: PUNTOS RESPUESTA LIBRE X(t)-X(k)
                                  3: PUNTOS RESPUESTA TOTAL X(t)-X(k)
                                  4: PUNTOS RESPUESTA TOTAL Y(t)-Y(k)
                                  5: GRAFICAS RESPUESTA LIBRE X(t)-X(k)
                                  6: GRAFICAS RESPUESTA TOTAL X(t)-X(k)
                                  7: GRAFICAS RESPUESTA TOTAL Y(t)-Y(k)
                                  8: VERIFICACION CONTROLABILIDAD DE ESTADO
                                  9: VERIF.CONTROLABILIDAD DE LA SALIDA
                                 10: VERIFICACION OBSEVABILIDAD
                                 11: REALIMENTACION DEL ESTADO
                                 12: OBSERVADOR
                                 13: DISE&O DEL OBSERVADOR MINIMO DE ESTADO
                                 14: VALORES CARACTERISTICOS DEL SISTEMA
                                 15: EC. EST.-->SIST. DISE&ADO REALIMENTACION
                                 16: VOLVER AL SISTEMA ANTERIOR
                                 17: SALIRSE


         Opcion -->  6
```



0.389
-0.038
-0.465
1
50
100


0.196
0.018
-0.160
1
50
100


0.990
-0.352
-1.694
1
50
100



0.403
-0.061
-0.525
1
50
100

BIBLIOGRAFIA.

-CHEN C.T., Linear System Theory and Design;Holt,Rinehart and Winston,Inc,1984.

-D'AZZO J.J.;HOUPIS C.H., 3rd. Edition , Linear Control System Analysis and Design:Conventional and Modern . Mc. GRAW-HILL , 1988.

-DORF R.C., Time Domain Analysis and Design of Control Systems, Reading,Mass.,Addison-Wesley Inc.,1965.

-DORF R.C., Modern Control Systems,Addison-Wesley,Massachusets 1974.

-FRIEDLAND B.,Control System Design,An introduction to state-space methods, Mc. Graw-Hill,Inc, 1987.

-HOSTETTER G.H;SAVANT C.J;STEFANI R.T., Sistemas de Control, Interamericana,México, 1987.

-KAILATH T., Linear Systems,Prentice-Hall,Inc.,1980.

-KUO B.C., Automatic Control Systems,Fifth Edition;Prentice-Hall, Inc, 1987.

-KUO B.C., Digital Control Systems;Holt,Rinehart and Winston,Inc, 1980.

-LAGO G.;BENNINGFIELD L.M.,Teoría de Sistemas y Circuitos;Limusa, México, 1984.

-LUTHE R;OLIVERA A;SCHUTZ F., Métodos Numéricos;Limusa,México, 1978.

-MASTASCUSA E.J., Computer-Assisted Network and System Analysis; John-Wiley and Sons,Inc, 1988.

-OGATA K., Discrete-Time Control Systems, Prentice-Hall, Inc, 1987.

-OGATA K., Ingeniería de Control Moderna,Prentice-Hall,México, 1986.

-PHILLIPS C.L.;HARBOR R.D., Feedback Control Systems,Prentice-HALL,Inc, 1988.

-RODRIGUEZ R.F.J., Sistemas Dinámicos,Facultad de Ingeniería;UNAM, 1986.

-ROSENBROCK H.H., State-Space and Multivariable Theory.T.Nelson and Sons LTD, 1970.

-SCHILLING R.J.;LEE H., Engineering-Analysis,A Vector Space Approach., John-Wiley and Sons, 1988.

-SCHILDT H., Turbo Pascal Avanzado,Mc. Graw-Hill,Inc, 1987.

-TORRES J.;CZITRON V., Métodos Para la Solución de Problemas con Computadora Digital,Representaciones y Servicios de Ingeniería, México,1980.

-WIBERG D.M., Espacios de estado y Sistemas Lineales, Mc.Graw-Hill,México, 1973.