INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA

REPORTE TECNICO

SIMULACION DE SISTEMAS DINAMICOS LINEALES Y CALCULOS MATRICIALES RELACIONADOS

PROGRAMA ALIN

OCTUBRE DE 1999
AUTOR: Salvador Saucedo Flores
SEPI-ESIME (Becario COFAA y DD)

R E S U M E N

Se presenta un programa que permite resolver numéricamente la ecuación de estado espacio para sistemas lineales e invariantes. Permite que las señales de control sean funciones del tiempo y/o del estado, por lo que el control puede ser no lineal y variante en el tiempo. Produce la gráfica de las dos primeras variables de estado contra el tiempo, la de plano de estado y la de salida y de control.

Genera dos archivos tipo ASCII con los resultados de la simulación. Se puede salvar en un archivo el sistema de ecuaciones, para su posterior ejecución. Permite edición de fórmulas y de valores e imprimir el sistema ya mencionado, ya sea en impresora o en un archivo tipo texto. Permite discretizar el modelo continuo y efectuar la simulación en forma discreta.

Calcula la matriz de realimentación de estado, así como el observador mínimo o el completo. Puede calcular el modelo de estado espacio a partir de la función de transferencia y viceversa. Cuenta con ayuda en línea. Versiones posteriores hallarán la matriz de transición de estado y se podrá efectuar análisis de observabilidad y de controlabilidad.

C O N T E N I D O

1 Introducción

El presente trabajo describe un método computacional para resolver numéricamente un sistema de ecuaciones diferenciales lineales e invariantes en el tiempo, de primer orden, a partir de condiciones iniciales conocidas. El programa del mismo se pone a disponibilidad de la comunidad de la ESIME para servir de herramienta auxiliar en diferentes asignaturas de las licenciaturas y del posgrado, como pueden ser: Métodos Numéricos, Teoría de Control, Circuitos Eléctricos, Simulación de Sistemas, etc.

El sistema de ecuaciones puede ser desde una ecuación diferencial hasta siete de ellas. Su mayor ventaja es resolver ecuaciones lineales, al mismo tiempo que proporciona resultados como la ecuación característica, las raíces del sistema en lazo abierto y la inversa de la matriz de estado.

Permite ver la gráfica de las dos primeras variables de estado, además de que permite salvar el modelo matemático del sistema simulado en un archivo para su posterior edición y simulación. Los datos de salida de las variables de estado, de salida y de control versus el tiempo se guardan en un archivo tipo ASCII, con el fin de que puedan ser impresos o procesados con software más avanzado.

El programa **ALIN** permite la edición de las funciones que constituyen el control, así como variar cómodamente los principales parámetros de integración, como son el paso inicial de integración, los tiempos inicial y final de la solución, las condiciones iniciales de las variables de estado y el error para controlar el ajuste del paso de integración.

Cuando sólo se usa una señal de control, la otra se puede usar para evaluar la solución analítica con la idea de que el estudiante corrobore la exactitud de la solución numérica.

El programa se presta para implementar la simulación de sistemas retroalimentados de control, pues cuenta con dos funciones para tal fin.

Se presentan ocho ejemplos de uso del programa con el fin de que sirvan de guía de uso, además de que representan ciertos resultados clásicos sobre la materia de Circuitos Eléctricos y de Teoría de Control IV, de ICA .

Versiones posteriores del programa permitirán resolver problemas con valor a la frontera, así como de optimización dinámica.

El programa está escrito enteramente en C de Borland.

2 Método del Espacio de Estado para Análisis de Sistemas

El estado de un sistema dinámico es el conjunto más pequeño de variables (variables de estado) tales que el conocimiento de estas variables en $t = t_0$, en combinación con el conocimiento de la entrada para $t \geq t_0$ determinan enteramente el comportamiento del sistema para $t \geq t_0$. Si se necesitan n variables de estado para definir completamente el comportamiento de un sistema, se puede tomar a esas n variables como las n componentes de un vector **x** que recibe el nombre de vector de estado.

Si asumimos que el sistema tiene r entradas $u_1(t)$, $u_2(t)$,...,$u_r(t)$ y que tiene m salidas $y_1(t)$, $y_2(t)$,...$y_m(t)$, además de que es lineal e invariante podemos representarlo mediante:

$$\boxed{\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}}$$

donde **A** se llama matriz de estado, **B** es la matriz de entrada, de n filas y r columas; **C** es la matriz de salida, de m filas y n columnas y **D** es la matriz de transmisión directa; **x** es el vector de estado, **u** es el vector de control, **y** es el vector de salida.

Una representación en el espacio de estado de ecuaciones diferenciales lineales de orden n en los que la función excitadora incluye términos derivativos

Dada la ecuación diferencial de un sistema lineal:

$$y^{(n)} + a_1 y^{(n-1)} + \ldots + a_n y = b_0 u^{(n)} + b_1 u^{(n-1)} + \ldots + b_n u$$

La matriz de estado **A** viene dada por:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \bullet & \bullet & \bullet & \dots & \bullet \\ \bullet & \bullet & \bullet & \dots & \bullet \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}$$

donde:

$$x_1 = y - b_0 u$$

$$x_2 = \dot{y} - b_0 \dot{u} - b_1 u$$

$$x_3 = \ddot{y} - b_0 \ddot{u} - b_1 \dot{u} - b_2 u$$

$$\vdots$$

$$x_n = y^{(n-1)} - b_0 u^{(n-1)} - b_1 u^{(n-2)} - \dots - b_{n-2} \dot{u} - b_{n-1} u$$

donde β_1, β_2, ...β_n se determinan mediante:

$$b_0 = b_0$$

$$b_1 = b_1 - a_1 b_0$$

$$b_2 = b_2 - a_1 b_1 - a_2 b_0$$

$$\vdots$$

$$b_n = b_n - a_1 b_{n-1} - \bullet\bullet\bullet - a_{n-1} b_1 - a_n b_0$$

Al elegir de esta manera a las variables de estado, resulta:

$$\dot{x}_1 = x_2 + b_1 u$$

$$\dot{x}_2 = x_3 + b_3 u$$

$$\vdots$$

$$\dot{x}_{n-1} = x_n + b_{n-1} u$$

$$\dot{x}_n = - a_n x_1 - a_{n-1} x_2 - \bullet\bullet\bullet - a_1 x_n + b_n u$$

El valor de la salida viene dado por:

$$y = [1\,0\,0 \cdots 0] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + b_0 u$$

Resumiendo las expresiones (3) a (6), resulta:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \bullet & \bullet & \bullet & \dots & 0 \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \bullet \\ \bullet \\ b_n \end{bmatrix} u$$

$$C = [1 \quad 0 \quad \cdots \quad 0], \quad D = b_0 = b_0$$

ALIN permite obtener (8) y (9) a partir de la función de transferencia.

3 Método de Integración Numérica Runge-Kutta-Fehlberg

En [6] y [7] se presenta el método de integración numérica de Runge-Kutta-Fehlberg que emplea **ALIN** para resolver (1). En esencia, el método es un Runge-Kutta de quinto orden con estimado del error, que sirve de criterio para cambiar el lapso de integración. El estimado del error se obtiene a costa de una complejidad mayor y de un trabajo computacional más elevado.

El programa ALIN implementa la solución mencionada arriba, pero ajusta el valor del paso de integración Ha usando el siguiente criterio, basado en el error actual de integración, Ea:

1) Si el error Ea es menor que Eo/4 duplicar el valor de Ha.
2) Si Ea es mayor que Eo*4 dividir por 2 el valor de Ha.
3) El mínimo valor de Ha permitido es Ho/16.
4) Si Ea rebasa el valor 330*Eo + 0.01*$|x_1|$ aborta el cálculo.

El programa **ALIN** no es tan estricto como para repetir el paso ya calculado, esto es si detecta un error mayor que Eo*4 reduce Ha a la mitad pero lo emplea para la siguiente iteración, sin corregir la ya calculada. Esto se hace para lograr más rapidez

El método de integración es bastante más pesado que el Runge-Kutta de cuarto orden tradicional, pero hay casos en que la precisión lograda lo hace preferible. Debido a lo extenso de los cálculos se recomienda correr el programa ALIN en una máquina 486 con procesador numérico, como mínimo.

4 Instructivo de Uso

INSTALACION El programa se instala desde un diskette a disco duro mediante la siguiente secuencia:

1) Crear un directorio en el disco C, dándole cualquier nombre, aunque se recomienda llamarle ALIN:

   C:>md alin ENTER
   C:>cd alin ENTER

2) Copiar desde A: (o B:) los programas y archivos que forman el paquete.

   C:\ALIN>copy a:*.* ENTER

EJECUCION Para llamar al programa ALIN se pueden usar dos maneras:

1) La forma más sencilla consiste en teclear el nombre del programa sin dar ningún parámetro en la línea de comandos:

   C:\ALIN>alin ENTER

En este caso se cargará el archivo llamado LINA.SIS, mismo que el usuario puede modificar para obtener otro modelo matemático, el cual puede ser ejecutado y salvado con otro nombre o con LINA.SIS, si así se desea.

2) Agregar después del nombre del ejecutable el nombre del archivo que contiene el sistema que se desea cargar inicialmente:

   C:\ALIN>alin NombreArch

En este caso el programa carga el sistema matemático contenido en el archivo especificado por NombreArch, mismo que puede ser editado, ejecutado y salvado con el mismo nombre o con otro nuevo, con lo que se pueden crear más sistemas a partir del original. Se recomienda usar la extensión **SIS** para designar a los archivos que contienen los modelos.

En ambos casos se presenta inicialmente la página de bienvenida, misma que ofrece una somera indicación de la capacidad del programa:

pantalla de presentación:

```
               Ing. en Control y Automatización de ESIME Zacatenco


                   Solución Numérica de un Sistema Lineal
              (Método Runge-Kutta-Fehlberg con paso ajustable)

   Del tipo dXi(t)/dt = Ai1*X1+Ai2*X2+..+Ain*Xn+Bi1*U1+Bi2*U2; i = 1,n; n<8

   Con salidas Yj(t) = Cj1*X1+Cj2*X2+..+Cjn*Xn+Dj1*U1+Dj2*U2; j = 1,2;  n<8

U1 = G1(X1,X2,..,Xn,Ki,t), U2 = G2(X1,X2,..,Xn,Ki,t) son variables de control

   Las Gs aceptan los operadores: +,-,*,/,^,(,) y las funciones: RAIZ,CUAD,
   ABS,SEN,COS,TAN,ASEN,ACOS,ATAN,LOG,LN,ALOG,EXP,ENT,SENH,COSH,TANH y SIG

      Calcula matriz de realimentación para ubicar polos de lazo cerrado
          (Calcula observador reducido ubicando los polos del mismo)
   Convierte al modelo de Estado Espacio a partir de la función de Tranferencia y viceversa
   Puede Discretizar el modelo continuo

                    Presionar una tecla para continuar.
```

Al presionar una tecla se presenta la pantalla de edición y comandos:

```
Memoria Vacante:  49K     Dar / para ver comandos (F1 = Ayuda) Forma   Autónomo
                 MATRICES Y FUNCIONES                        Xi(t)   Xi(0) Graf
A1i    0.00    1.00    0.00                                   1.000    1.00    1
A2i    0.00    0.00    1.00                                   -1.00   -1.00   -1
A3i   -1.00   -2.00   -3.00                                    0.00    0.00    1
A4i                                                           0.000            0
A5i                                                            0.00            2
A6i                                                           0.000            1
A7i                                                            0.00            1
Bi1    0.00    0.00    1.00                                  Y1(t)=            2
Bi2                                                          Y2(t)=            1
C1i                                                                            2
C2i                                                                           -1
G1     1.00                                                                    1
G2
Ho Lapso inicial de integración =      0.02     sistema:   lina.sis
Ha  Lapso actual de integración =      0.02
```

```
Ti Tiempo inicial de simulación =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulación =      0.00
Tf   Tiempo final de simulación =     20.00
Eo Error inicial de integración =   0.00002  Iteraciones =        0
Ea  Error actual de integración =   0.00002                        orden:  3

lina.sis
```

USO DE LAS TECLAS.

Una vez que se introduce la pantalla de presentación el usuario puede editar las constantes y las fórmulas mediante la introducción de nueva información directamente. También se puede ayudar usando las teclas siguientes:

Fin El cursor se posiciona en el valor del orden del sistema.

Inicio El cursor se pone en la primera constante (A_{11}).

RePág El cursor se coloca en la 1ra. condición inicial, $X_1(0)$.

AvPág El cursor se coloca en el lapso inicial, es decir Ho.

FLECHAS El cursor se mueve en la dirección correspondiente.

Supr Borra fórmula o valor bajo el cursor.

F1 o Alt-Y Permite ver la ayuda del programa **ALIN**.
F9 Recalcula las funciones si no se está en el modo autónomo.

F2 Edita la fórmula o el texto señalado por el cursor.

F3 Conmuta en ver fórmulas o ver valores.

F5 Genera la gráfica de plano de estado entre x_1 y x_2.

F8 Formatea la celda iluminada.

Alt-F2 Salva el sistema bajo el mismo nombre, sin confirmar.

Alt-D Permite ir al sistema operativo DOS. Retornar con **EXIT**.

Ctrl-F10 Abandona el programa sin salvar el modelo.

/ Acceso a los comandos principales. Exhibe en la línea 24 la lista de comandos. Esta línea se verá así:

```
Sistema,  Formato,  Borra,  Gráfica,  Calcula,  Inicia,  Edita,  Utiles,  Auto,  Termina
```

Al exhibirse dicha lista de comandos el usuario sólo debe oprimir la tecla correspondiente a la primera letra de comando para ejecutar dicho comando; por ejemplo, si presiona "i" o "I" ejecutará el comando **Inicia**.

Vale la pena aclarar desde este momento que algunos de los valores no pueden ser editados o borrados directamente por el usuario, pues ellos reflejan alguna condición que es producida por el algoritmo de integración, por lo que el usuario sólo puede alterar:

a) el orden del sistema (número de variables de estado).
b) Las 49 constantes para la matriz de estado A.
c) Las siete condiciones iniciales de las Xi.
d) Las catorce constantes para la matriz B.
e) Las 14 constantes para la matriz C y las 4 para la matriz D.
f) Los tiempos inicial y final, así como el lapso Ho.
g) El error de control, Eo.
h) Las funciones o valores para u_1 y para u_2.

Veamos ahora los comandos principales:

Sistema Este comando da lugar a cuatro sub comandos:

Cargar Permite cargar un nuevo sistema, pero antes pregunta al usuario si salva el sistema actual.

Salvar Permite salvar el sistema actual; con el mismo nombre de archivo o con otro diferente. Pregunta al usuario si desea sobreescribir un archivo ya existente.

Imprimir Imprime pantalla de trabajo en impresora o en un archivo ASCII. Pregunta al usuario si desea imprimir el borde.

Reiniciar Borra todas las constantes, cargando archivo NADA.SIS.

Discretiza Calcula el modelo discreto y pone o quita el modo discreto de cálculo.

Formato Formatea el valor bajo el cursor. Justifica a derecha o izquierda, puede poner comas y fija cantidad de decimales.

Borrar Tiene el mismo efecto que la tecla Supr, es decir borra la fórmula o el valor bajo el cursor (si es borrable).

Gráfica Exhibe la gráfica en video VGA de la variables de estado x1 y x2, contra el tiempo y la de y1 y u1 vs tiempo. Si el ratón está activo se verán los valores del tiempo, de x1 y x2, además de que el cursor del ratón funciona como vernier o apuntador. Usar este comando después de haber calculado la solución.

Calcula En su primera etapa presenta al usuario las matrices del sistema y los parámetros principales de la simulación; Halla el polinomio característico de A. Presenta la matriz inversa de A y las raíces del sistema en lazo abierto. Salva resultados en archivo ASCII REPORTE.DAT. A continuación se muestra el archivo REPORTE.DAT correspondiente a la pantalla de trabajo vista:

```
  u =  1.00  h = 0.020    Tfinal =  20.000

 El modelo viene dado por:

   0.000   1.000   0.000  x1     0.000
   0.000   0.000   1.000  x2 +   0.000 u
  -1.000  -2.000  -3.000  x3     1.000

cond. iniciales: x1:   1.000  x2:  -1.000  x3:   0.000

La salida es y1 = +0.000*x1 +0.000*x2  +0.000*x3

polinomio característico para matriz A:
  s^3   +3.000*s^2   +2.000*s^1   +1.000

la inversa de la matriz A es:
   -2.000   -3.000   -1.000
    1.000   -0.000   -0.000
   -0.000    1.000   -0.000
las raíces del sistema son:
p1 =   -0.3376   -   0.5623j
p2 =   -0.3376   +   0.5623j
p3 =   -2.3250
```

La segunda etapa de **Calcula** es opcional y usa el método Runge-Kutta-Fehlberg para obtener la solución del sistema de ecuaciones diferenciales. Envia al archivo SALIDA.PRN los valores de tiempo y de las variables de estado hasta un máximo de 500 veces. El archivo SALIDA:PRN es procesado por el comando **Gráfica** para hacer la gráfica ya mencionada. Mientras se está calculando, cualquier teclazo, permite ver el valor del estado, el error de integración actual, Ea y el número de iteraciones realizadas.

La tecla **ESC** aborta a la segunda etapa del comando **Calcula**.

Inicia Inicia el sistema para dejarlo listo para el comando Calcula, pues pone las condiciones iniciales en el vector de estado, así como el tiempo inicial en el tiempo actual. Inicia las variables que emplea el algoritmo de solución.

Editar Tiene el mismo efecto que F2, es decir edita la fórmula o el valor bajo el cursor.

Utiles Origina cinco sub comandos:

Recalcula calcula las fórmulas si no está activo el modo autónomo.

Fórmula visible Como F3, conmuta entre ver fórmula o ver valor.

Ackermann Calcula la matriz de realimentación de estado para ubicar los polos de lazo cerrado, mediante la fórmula de Ackermann. Asume un sistema con una señal de control solamente. Modifica la expresión para el control.

Obs. Reducido Calcula el observador mínimo y lo pone en el modelo, pidiendo los polos del observador. El usuario debe modificar, si así lo desea, el orden del sistema, tras el cálculo correspondiente. Asume que la única salida es $y(t) = x_1(t)$.

Completo Calcula el observador completo y lo pone en el modelo, pidiendo los polos del observador. El usuario debe modificar, si así lo desea, el orden del sistema, tras el cálculo correspondiente. Asume una única salida.

Auto Conmuta entre el modo autónomo y el modo manual. Se debe usar el modo (o forma) autónomo al calcular la solución.

ESCAPE Regresa al modo de inserción directa (borra línea 24).

Reglas para las fórmulas

Pueden ser de hasta 98 caracteres de largo. Cuando al terminar de teclear una fórmula ésta se exhibe en color magenta se indica que sí fue aceptada como tal. Sólo los primeros 8 caracteres de una fórmula son visibles directamente.

Las fórmulas admiten la referencia a las variables de estado con la letra "x" seguida de un dígito de 1 a 7 para designar a la variable de estado correspondiente. Con "k" seguido de un dígito se designan las constantes Ki.

Usar "t" para referenciar al tiempo actual Ta.

Se pueden aplicar los operadores y las funciones:

operadores:

+	suma.
-	resta o signo de negativo.
*	producto.
/	división.
^	potenciación.
(	abre paréntesis para destruir ambigüedad.
)	cierra paréntesis. Debe haber balance de paréntesis.

FUNCIONES:

RAIZ	Raíz cuadrada de un valor no negativo.
CUAD	Eleva su argumento al cuadrado.
ABS	Valor absoluto
SEN	Seno del argumento (en radianes)
COS	Coseno del argumento (en radianes).

TAN Tangente del argumento (en radianes).
ASEN Seno inverso, en radianes.
ACOS Coseno inverso, en radianes.
ATAN Tangente inversa, en radianes.
LOG Logaritmo base 10.
LN Logaritmo natural o neperiano.
ALOG Antilogaritmo base 10.
EXP Antilogaritmo natural o neperiano.
ENT Entero más cercano al argumento.
SENH Seno hiperbólico.
COSH Coseno hiperbólico.
TANH Tangente hiperbólica.
SIG Retorna +1 si argumento ≥ 0, -1 si no.

El argumento de una función debe ir entre paréntesis y se permite usar funciones compuestas o anidadas. Toda fórmula puede introducirse en mayúsculas o en minúsculas o mezclando ambas.

Se recomienda inhabilitar (mediante edición) temporalmente las fórmulas que no intervienen en los cálculos de la solución para que ésta sea más rápida.

Se puede insertar espacios para hacer más legible a la fórmula, aunque esto no siempre es recomendable. Las fórmulas se procesan mediante un analizador sintáctico, que constituye la base del programa **ALIN**.

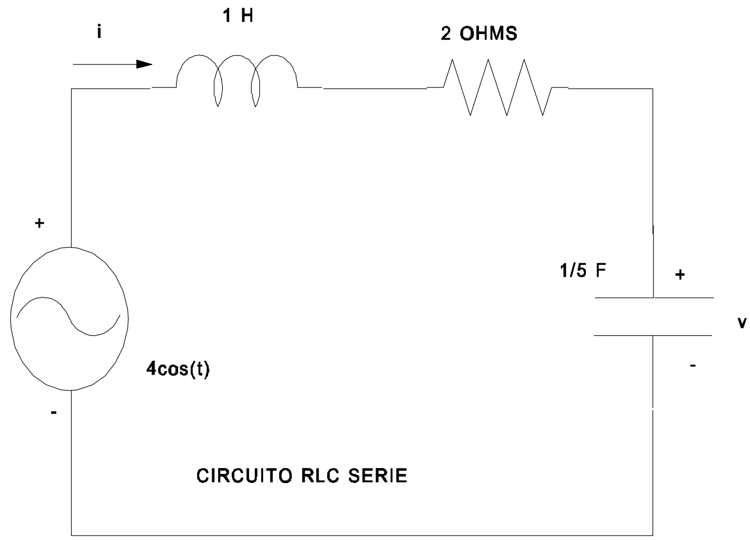
Constantes vacías El programa siempre asume que sólo una variable de control o de excitación está activa, es decir u_1. Para indicar que también u_2 está activa la constante B_{12} debe tener un valor, es decir, no estar vacía. Igualmente se asume que sólo se tiene la salida y_1 definida, a menos que la constante C_{21} no esté vacía, en cuyo caso también y_2 está definida. El resto de las constantes que estén vacías se evalúan como cero.

La inserción tiene un reductor automático por lo que si se desea introducir la constante 22/7, por ejemplo, se puede teclear justamente **22/7** y se tendrá el número flotante correspondiente.

Edición Permite editar valores y fórmulas usando las teclas **Inicio, Fin, flechas, Supr, retroceso, Insert, ENTER** y **ESC.** Al editar una fórmula muy larga sólo se exhiben 79 caracteres (en la línea 24) como máximo.

Las constantes Ki, de la columna de la extrema derecha, sí tienen cierto uso en la presente versión del programa, pues pueden servir para tener más capacidad de funciones.

5 Ejemplo 1: Circuito serie RLC (2do orden)



Si se tiene un circuito RLC serie excitado por una fuente de voltaje, ver página 308 de [3], tendremos la ecuación diferencial de segundo orden:

$$v'' + 2v' + 5v = 20\cos(t)$$

donde la fuente viene dada por:

$$v_g = 4\cos(t)V$$

Si hacemos $\mathbf{x_1 = y = v}$, la expresión para las matrices A, B y C resulta:

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 5 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = 0$$

Pasos:

1) Ejecutamos **ALIN** sin dar ningún nombre de archivo:

C:\ALIN>alin ENTER

La anterior hará que el archivo LINA.SIS se cargue.

2) Limpiar cualquier fórmula usando el comando **Sistema** seguido del subcomando **Reiniciar**, lo que hará que se cargue el archivo NADA.SIS que tiene sus fórmulas sin llenar.

3) Ir al lugar para A11 e insertar el valor:

0.00 ENTER

Poner en A12 1.0, en A21, -5.0 y en A22 poner -2.0; Insertar 6.0 en X1(0) y 10.0 en X2(0) para definir las condiciones iniciales. Poner los tiempos inicial y final en 0.0 y 5.0 respectivamente.

Se elige el valor de Ho, el lapso inicial de integración, igual a 0.01 y se selecciona el valor para Eo, el error para control del lapso como 0.00001.

4) Posicionarse en la esquina inferior derecha de la pantalla de acopio de datos para fijar el orden del modelo como 2.

5) Aunque el método no necesita la solución analítica, resulta didáctico incluirla, por lo que se escribirá la siguiente cadena:

EXP(-T)*(2*COS(2*T)+5*SEN(2*T))+4*COS(T)+2*SEN(T) ENTER

en la variable auxiliar G_2. Efectuar el comando **Inicia** para corroborar la solución concuerde con la condición inicial. Usar ahora el comando **Sistema** seguido de **Salvar** y archivar el sistema con el nombre HILBURN.SIS.

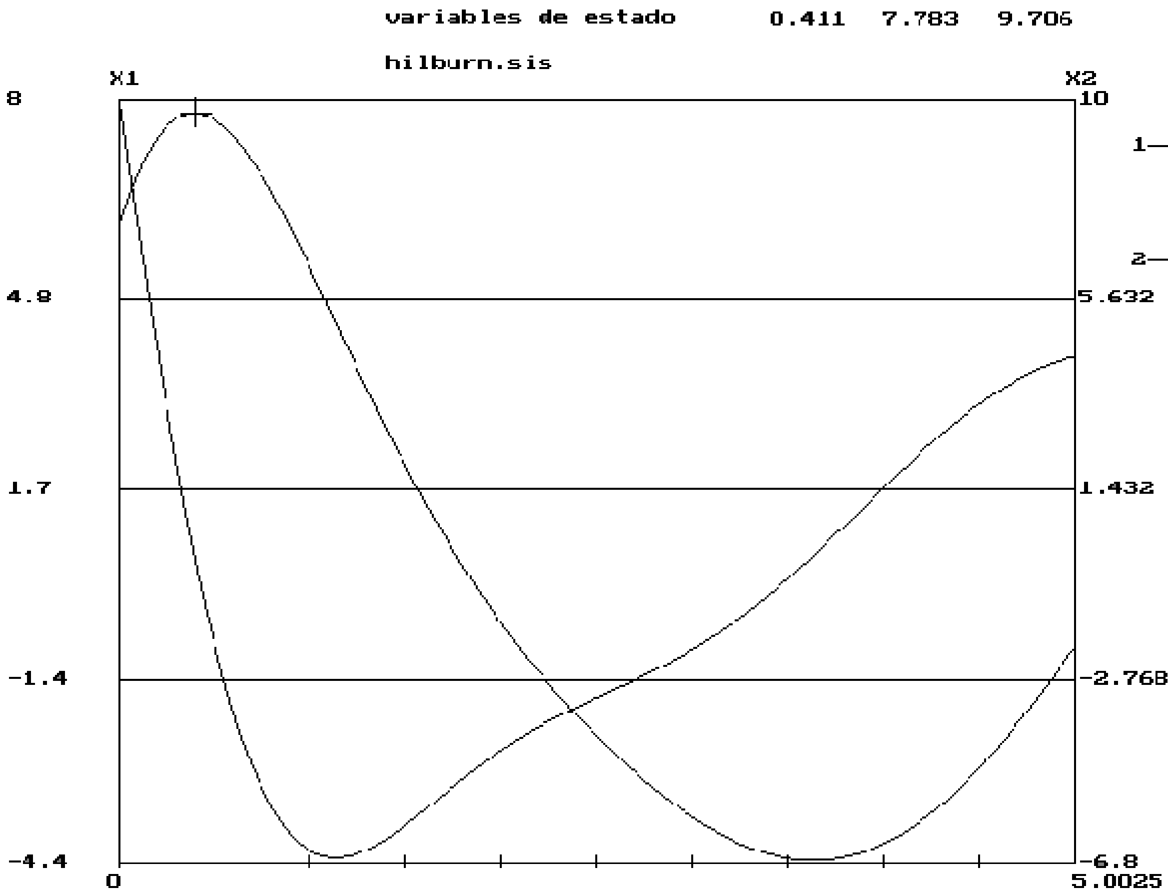
6) Desactivar, mediante edición, a G2 para obtener más velocidad y ejecutar el comando **Calcula** para obtener la solución. En la primera etapa se exhibirán en el video los resultados del análisis preliminar; y se salvarán en el archivo REPORTE.DAT, que aquí se anexa:

```
u =   4.000  h =   0.010  T final =   5.000   orden =   2

La matriz de estado A es:
   0.000    1.000
  -5.000   -2.000

La transpuesta de la matriz para el control, B', es:
   0.000    5.000

Condiciones iniciales:
   6.000   10.000

La matriz de salida, C, es:
   1.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000   2.000    5.000

la inversa de la matriz A es:
   -0.400   -0.200
    1.000    0.000

las raíces del sistema son:
 p1 =   -1.0000   -   2.0000j
 p2 =   -1.0000   +   2.0000j
```

ALIN usa el algoritmo de Newton para calcular las raíces, que en este caso son complejas. El algoritmo usado es capaz de calcular las raíces del sistema en lazo abierto, aún en el caso de que haya una o más parejas de raíces complejas.

Para determinar el polinomio característico, **ALIN** emplea el método de Fadeev-Leverrier.

Dar lugar a la segunda etapa del comando y observar como el tiempo se incrementa hasta arribar al valor final. Contrastar el valor final para x_1 contra el valor teórico dado por G_2. Ver la gráfica de la solución usando el comando **Gráfica**.



7) Usar ahora el mandato **Imprimir** para enviar la pantalla de trabajo al archivo HILBURN.PRN, que se exhibe a continuación:

```
                 MATRICES Y FUNCIONES                        Xi(t)   Xi(0) Gráf
A1i    0.00    1.00                                          -0.805    6.00    1
A2i   -5.00   -2.00                                            4.39   10.00   -1
A3i                                                            0.00            1
A4i                                                           0.000            0
A5i                                                            0.00            2
A6i                                                           0.000            1
A7i                                                            0.00            1
Bi1    0.00    5.00                                          Y1(t)=   -0.81    2
Bi2                                                          Y2(t)=            1
C1i    1.00    0.00                                            0.00    0.00    2
C2i                                                                           -1
G1 4*COS(T)                                                                    1
G2 EXP(-T)*(2*COS(2*T)+5*SEN(2*T))+4*COS(T)+2*SEN(T)_
Ho Lapso inicial de integracion =      0.01     sistema:   hilburn.sis
Ha  Lapso actual de integracion =      0.00
Ti Tiempo inicial de simulacion =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulacion =      5.00 OK tabla en SALIDA.PRN
Tf   Tiempo final de simulacion =      5.00
Eo Error inicial de integracion =   0.00001  Iteraciones =     1792
Ea  Error actual de integracion =   0.00001                        orden:  2
```

Notar el caracter "_" al final de la fórmula para G2, que se usa para desactivar temporalmente a la fórmula original.

8) Salir del programa con el comando **Termina,** salvando si se desea el modelo en HILBURN.SIS.

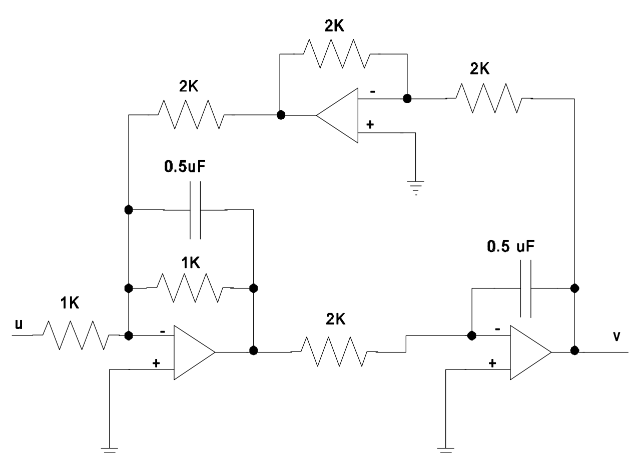
9) Examinar con un editor de texto al archivo SALIDA.PRN:

tiempo	x1	x2	salida 1	control 1
0.000	6.000	10.000	6.000	4.000
0.010	6.098	9.701	6.098	4.000
0.020	6.194	9.402	6.194	3.999
0.030	6.286	9.105	6.286	3.998
0.040	6.375	8.810	6.375	3.997
0.050	6.462	8.515	6.462	3.995
0.060	6.546	8.222	6.546	3.993
0.070	6.626	7.931	6.626	3.990
0.080	6.704	7.642	6.704	3.987
0.090	6.779	7.354	6.779	3.984
0.100	6.851	7.068	6.851	3.980
.....	.....	.....	.....	.....
4.800	-1.666	4.105	-1.666	0.349
4.810	-1.625	4.124	-1.625	0.389
4.820	-1.584	4.142	-1.584	0.429
4.830	-1.542	4.159	-1.542	0.469
4.840	-1.501	4.176	-1.501	0.508
4.850	-1.459	4.193	-1.459	0.548
4.860	-1.417	4.209	-1.417	0.588
4.870	-1.375	4.225	-1.375	0.627
4.880	-1.332	4.240	-1.332	0.667
4.890	-1.290	4.255	-1.290	0.706
4.900	-1.247	4.270	-1.247	0.745
4.910	-1.205	4.284	-1.205	0.785
4.920	-1.162	4.297	-1.162	0.824
4.930	-1.119	4.310	-1.119	0.863
4.940	-1.075	4.323	-1.075	0.902
4.950	-1.032	4.335	-1.032	0.941
4.960	-0.989	4.347	-0.989	0.980
4.970	-0.945	4.358	-0.945	1.018
4.980	-0.902	4.369	-0.902	1.057
4.990	-0.858	4.379	-0.858	1.095
5.000	-0.814	4.389	-0.814	1.134

6 Ejemplo 2: Circuito con amplificadores operacionales (2do orden)

Resolvamos ahora el ejercicio 9.37 de [3], que origina la ecuación diferencial de segundo orden:

$$y'' + 2 \bullet 10^3\, y' + 10^6\, y = -2 \bullet 10^6\, u(t)$$



cuyas raíces son reales y repetidas por lo que tenemos el **caso crítico amortiguado.** A continuación se presenta la solución numérica del caso no homogéneo:

Se define $\mathbf{x_1 = y}$ = salida de último amplificador y a $\mathbf{x_2}$ = a la salida del primer amplificador y si escalamos por mil para tener el tiempo en milisegundos:

$$\dot{x}_1 = -x_2$$

$$\dot{x}_2 = x_1 - 2x_2 - 2u$$

La solución analítica para (13) viene dada por:

$$y = -(10t + 10)e^{-t} + 10V \; ; \; t \text{ en ms}$$

cuya derivada es:

$$y' = 10t\,e^{-t} \quad t \text{ en ms}$$

Pasos:

1) Ejecutar ALIN dando el nombre del archivo para dicho sistema:

C:\ALIN>alin p937.sis

Se debe presentar el sistema de segundo orden que acompaña al programa ALIN:

```
                 MATRICES Y FUNCIONES                        Xi(t)   Xi(0) Gráf
A1i    0.00   -1.00                                           0.000    0.00    1
A2i    1.00   -2.00                                            0.00    0.00   -1
A3i                                                            0.00            1
A4i                                                           0.000            0
A5i                                                            0.00            2
A6i                                                           0.000            1
A7i                                                            0.00            1
Bi1    0.00   -2.00                                         Y1(t)=    0.00    2
Bi2                                                         Y2(t)=            1
C1i    1.00    0.00                                            0.00            2
C2i                                                                           -1
G1     5.00                                                                    1
G2 -(10+10*T)*EXP(-T)+10_
Ho Lapso inicial de integración =      0.01     sistema:   p937.sis
Ha  Lapso actual de integración =      0.01
Ti Tiempo inicial de simulación =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulación =      0.00
Tf   Tiempo final de simulación =      8.00
Eo Error inicial de integración =   0.00002  Iteraciones =        0
Ea  Error actual de integración =   0.00002                        orden:  2
```

Las condiciones iniciales son $x_1 = 0$ y $x_2 = 0$. La excitación es constante e igual a 5V.

2) Ejecutar el comando **Inicia** seguido de **Calcula**. Al terminar la segunda etapa, comparar los valores obtenidos con los teóricos. El análisis preliminar se tiene en el archivo REPORTE.DAT:

```
u =   5.000  h =   0.010  T final =   8.000   orden =   2

La matriz de estado A es:
   0.000   -1.000
   1.000   -2.000

La transpuesta de la matriz para el control, B', es:
   0.000   -2.000

Condiciones iniciales:
   0.000    0.000

La matriz de salida, C, es:
   1.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000   2.000    1.000

la inversa de la matriz A es:
   -2.000    1.000
   -1.000    0.000

las raíces del sistema son:
 p1 =   -0.9999
 p2 =   -1.0000
```

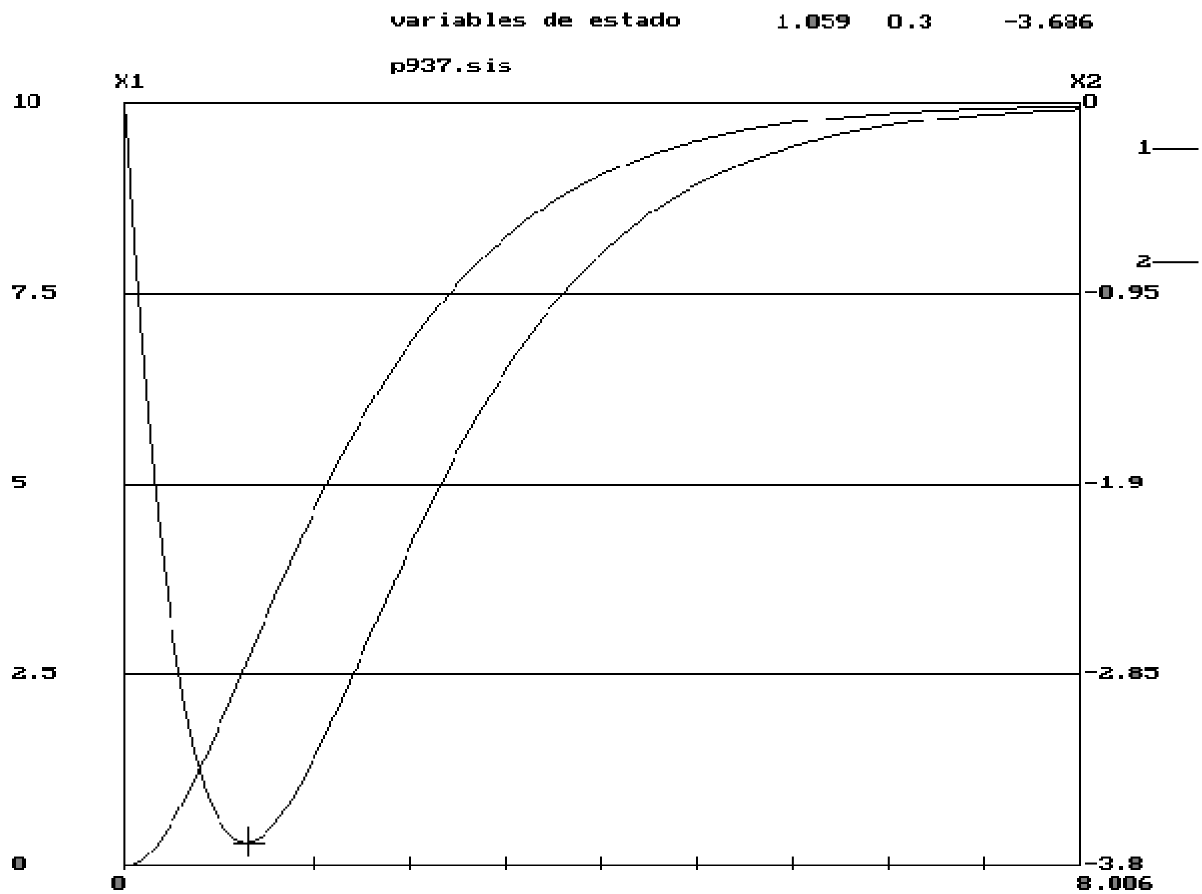
3) Ejecutar el comando **Gráfica** para ver las variables de estado contra el tiempo. Mover el cursor del ratón sobre una y otra curva para ver el efecto de vernier que tiene el ratón. Oprimir cualquier tacla para regresar a la pantalla alfanumérica:

4) Ejecutar **Salir** y examinar el archivo SALIDA.PRN con un editor:

```
 tiempo      x1       x2   salida
   0.000    0.000    0.000    0.000
   0.016    0.001   -0.157    0.001
   0.032    0.005   -0.310    0.005
   0.048    0.011   -0.457    0.011
   0.064    0.020   -0.600    0.020
   0.080    0.031   -0.738    0.031
   0.096    0.044   -0.872    0.044
   0.112    0.059   -1.001    0.059
   0.128    0.076   -1.125    0.076
   0.144    0.095   -1.246    0.095
   0.160    0.116   -1.363    0.116
   0.176    0.138   -1.475    0.138
   0.192    0.163   -1.584    0.163
   0.208    0.189   -1.688    0.189
   0.224    0.217   -1.789    0.217
   .....    .....   ......    .....
   7.664    9.959   -0.036    9.959
   7.680    9.960   -0.036    9.960
   7.696    9.960   -0.035    9.960
   7.712    9.961   -0.035    9.961
   7.728    9.961   -0.034    9.961
   7.744    9.962   -0.034    9.962
   7.760    9.962   -0.033    9.962
   7.776    9.963   -0.033    9.963
   7.792    9.963   -0.032    9.963
   7.808    9.964   -0.032    9.964
   7.824    9.964   -0.031    9.964
   7.840    9.965   -0.031    9.965
   7.856    9.965   -0.031    9.965
   7.872    9.966   -0.030    9.966
```

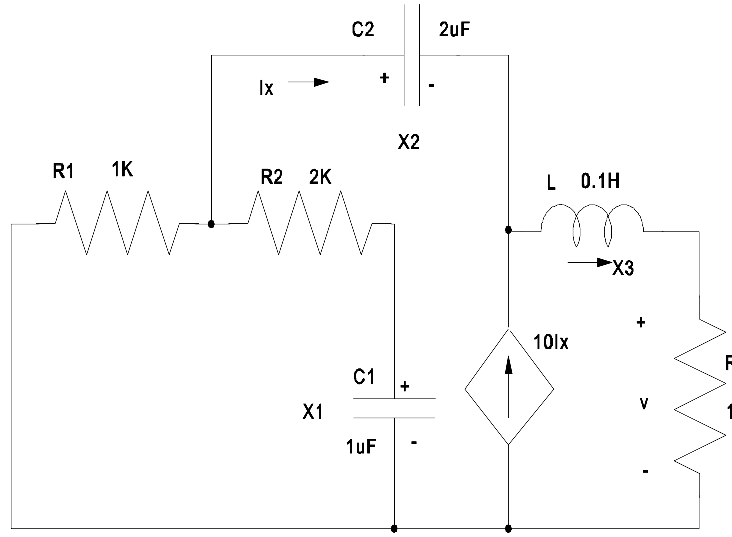
7.888	9.966	-0.030	9.966
7.904	9.967	-0.029	9.967
7.920	9.967	-0.029	9.967
7.936	9.968	-0.029	9.968
7.952	9.968	-0.028	9.968
7.968	9.969	-0.028	9.969
7.984	9.969	-0.027	9.969

5) Se presenta a continuación la gráfica del sistema. Notar el efecto de vernier que tiene el cursor del ratón.



7 Ejemplo 3: Circuito con una fuente dependiente (3er. orden)

Resolvamos ahora un sistema que origina una ecuación diferencial lineal de tercer orden, cuyos coeficientes no dependen del tiempo, tomada de [3].



Se define $\mathbf{x_1} =$ voltaje en el capacitor 1, $\mathbf{x_2} =$ voltaje en el capacitor 2 y $\mathbf{x_3}$ = corriente en el inductor, si además se escala el tiempo para que las unidades del mismo sean milisegundos, tendremos que:

$$\dot{x}_1 = -x_1/3 - x_3/0.033$$

$$\dot{x}_2 = x_3/0.022$$

$$\dot{x}_3 = 0.0033x_1 - 0.01x_2 - 10.606x_3$$

La solución numérica viene dada en [3] hecha con SPICE y reporta sólo el voltaje en R_3:

tiempo (ms)	V en R_3	tiempo (ms)	V en R_3
0.0	4.997	8.0	-0.0526
1.0	1.085	9.0	-0.0798
2.0	7.111	10.0	-0.0975
3.0	4.493	11.0	-0.1083
4.0	2.655	12.0	-0.1143
5.0	1.376	13.0	-0.1170
6.0	0.0493	14.0	-0.1175
7.0	-0.0115	15.0	-0.1164

Pasos:

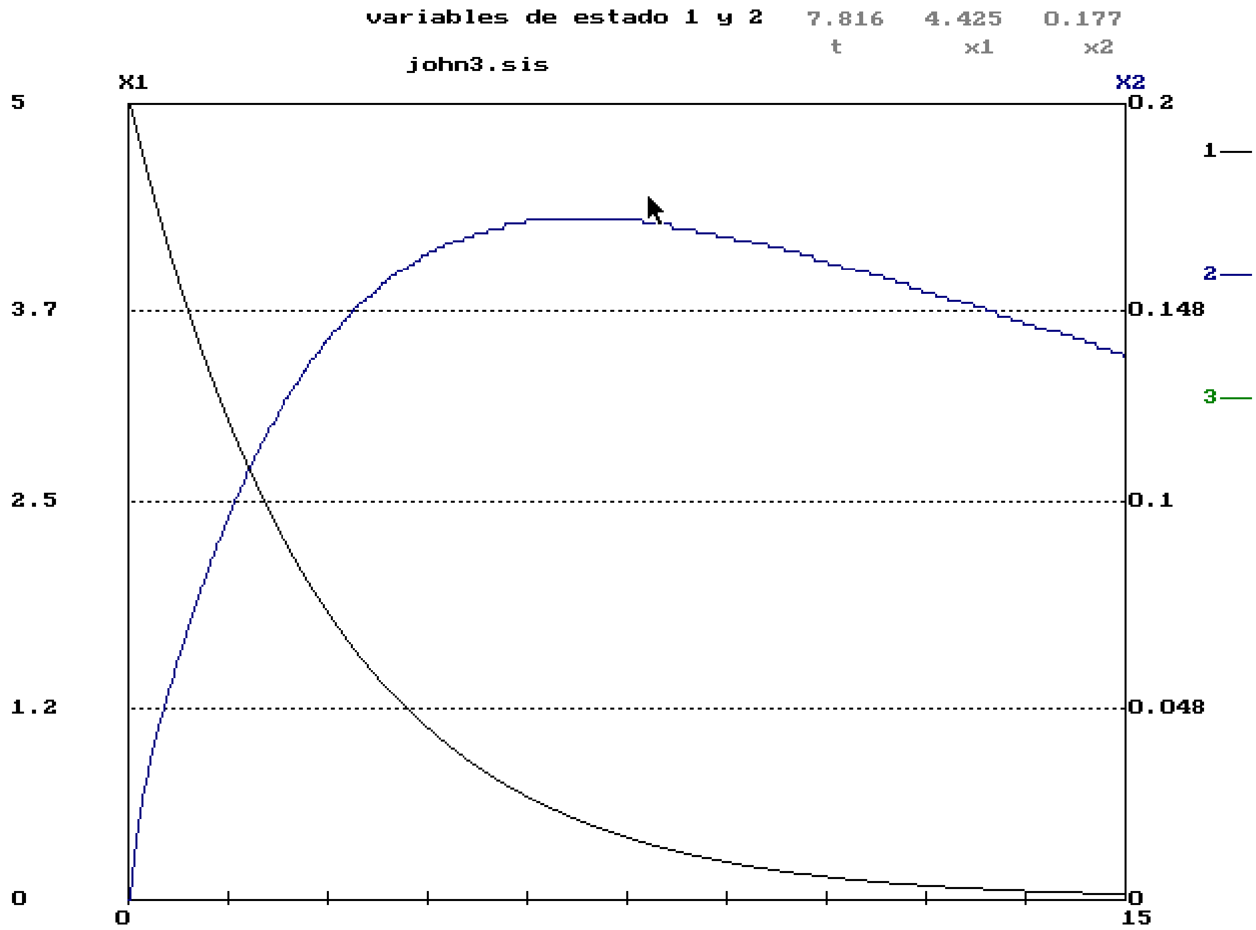
1) Ejecutar **ALIN** con el archivo que contiene el modelo visto:
C:\ALIN>alin john3.sis

2) Ejecutar el comando **Inicia** para poner las condiciones. Notar que las unidades de tiempo son milisegundos.

3) Ejecutar el comando **Calcula**. La primera etapa produce el archivo REPORTE.DAT, con resultados preliminares:

```
u =   0.000  h =   0.010  T final =  15.000   orden =   3

La matriz de estado A es:
  -0.333    0.000  -30.303
   0.000    0.000   45.455
   0.003   -0.010  -10.606

La transpuesta de la matriz para el control, B', es:
   0.000    0.000    0.000

Condiciones iniciales:
   5.000    0.000    0.005

La matriz de salida, C, es:
   0.000    0.000 1000.000

Coeficientes de ecuación característica en orden descendente:
1.000  10.939    4.091    0.152

la inversa de la matriz A es:
   -3.000   -2.000   -0.000
   -1.000  -23.999  -99.997
   -0.000    0.022   -0.000

las raíces del sistema son:
 p1 =   -0.0417
 p2 =   -0.3446
 p3 =  -10.5530
```

4) Usar el comando **Gráfica** para ver las curvas de la solución:



5) Usar el comando **Imprimir** para enviar al archivo JOHN3.PRN la pantalla de trabajo:

```
                 MATRICES Y FUNCIONES                        Xi(t)   Xi(0) Gráf
A1i   -0.33    0.00   -30.3                                   0.042    5.00    1
A2i    0.00    0.00    45.5                                    0.14    0.00   -1
A3i    0.00   -0.01  -10.61                                  -0.000   0.005    1
A4i                                                           0.000            0
A5i                    0.00                                    0.00            2
A6i                                                           0.000            1
A7i                                                            0.00            1
Bi1    0.00    0.00    0.00                                  Y1(t)=   -0.12    2
Bi2                                                          Y2(t)=            1
C1i    0.00         1000.00                                    0.00            2
C2i                                                                           -1
G1     0.00                                                                    1
G2
Ho Lapso inicial de integracion =      0.01     sistema:   JOHN3.SIS
Ha  Lapso actual de integracion =      0.02
Ti Tiempo inicial de simulacion =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulacion =     15.00 OK tabla en SALIDA.PRN
Tf   Tiempo final de simulacion =     15.00
Eo Error inicial de integracion =   0.00020  Iteraciones =     1000
Ea  Error actual de integracion =   0.00000                        orden:   3
```

6) Los valores contra el tiempo se pueden ver en el archivo SALIDA.PRN:

tiempo	x1	x2	x3	salida 1	control 1
0.000	5.000	0.000	0.005	5.000	0.000
0.030	4.946	0.006	0.004	4.074	0.000
0.060	4.894	0.011	0.003	3.391	0.000
0.090	4.842	0.015	0.003	2.886	0.000
0.120	4.792	0.019	0.003	2.512	0.000
0.150	4.742	0.022	0.002	2.234	0.000
0.180	4.693	0.025	0.002	2.025	0.000
0.210	4.644	0.028	0.002	1.868	0.000
0.240	4.597	0.030	0.002	1.748	0.000
0.270	4.549	0.032	0.002	1.656	0.000
0.300	4.503	0.035	0.002	1.585	0.000
0.330	4.457	0.037	0.002	1.528	0.000
0.360	4.411	0.039	0.001	1.482	0.000
0.390	4.366	0.041	0.001	1.444	0.000
0.420	4.321	0.043	0.001	1.412	0.000
0.450	4.277	0.045	0.001	1.384	0.000
0.480	4.233	0.047	0.001	1.359	0.000
0.510	4.190	0.048	0.001	1.337	0.000
0.540	4.147	0.050	0.001	1.317	0.000
0.570	4.104	0.052	0.001	1.298	0.000
0.600	4.062	0.054	0.001	1.281	0.000
.....	.....	.....	.....	.....	.....
14.400	0.048	0.140	-0.000	-0.117	0.000
14.430	0.048	0.140	-0.000	-0.117	0.000
14.460	0.048	0.140	-0.000	-0.117	0.000
14.490	0.047	0.140	-0.000	-0.117	0.000
14.520	0.047	0.140	-0.000	-0.117	0.000
14.550	0.047	0.139	-0.000	-0.117	0.000
14.580	0.046	0.139	-0.000	-0.117	0.000
14.610	0.046	0.139	-0.000	-0.117	0.000
14.640	0.045	0.139	-0.000	-0.117	0.000
14.670	0.045	0.139	-0.000	-0.117	0.000
14.700	0.045	0.139	-0.000	-0.117	0.000
14.730	0.044	0.139	-0.000	-0.117	0.000
14.760	0.044	0.138	-0.000	-0.117	0.000
14.790	0.044	0.138	-0.000	-0.117	0.000
14.820	0.043	0.138	-0.000	-0.117	0.000
14.850	0.043	0.138	-0.000	-0.117	0.000
14.880	0.043	0.138	-0.000	-0.117	0.000
14.910	0.042	0.138	-0.000	-0.117	0.000
14.940	0.042	0.137	-0.000	-0.116	0.000
14.970	0.042	0.137	-0.000	-0.116	0.000

8 Ejemplo 4: Sistema de control clásico

Veamos ahora un ejemplo tomado de [1] que consiste en encontrar la respuesta en el tiempo de un sistema con retroalimentación unitaria cuya función de transferencia en lazo abierto es:

$$G(s) = \frac{5(s+20)}{s(s+4.59)(s^2+3.41s+16.35)}$$

desarrollando numerador y denominador:

$$G(s) = \frac{5s + 100}{s^4 + 8\,s^3 + 32\,s^2 + 75s}$$

La solución analítica para dicho sistema viene dada por:

$$y(t) = 1 + \frac{3}{8}e^{-t}\cos 3t - \frac{17}{24}e^{-T}\,sen3t - \frac{11}{24}e^{-3t}\cos t - \frac{13}{8}e^{-3t}\,sent$$

Pasos:

1. Ejecutar **ALIN** junto con el archivo que tiene el sistema:

C:\ALIN>alin pa411.sis

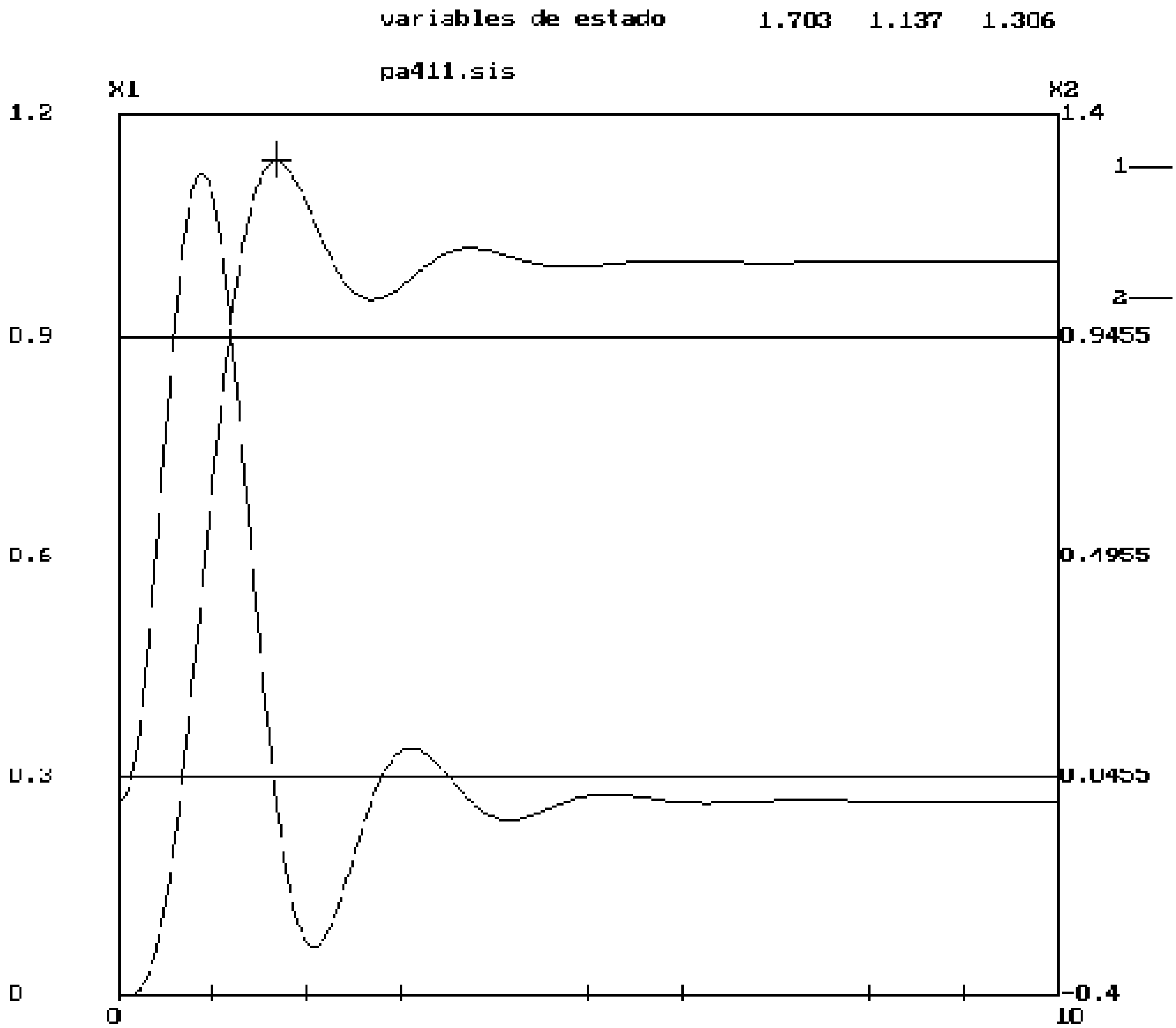
2. Ejecutar el comando **Inicia** para iniciar todo.

3. Ejecutar el comando **Calcula** que produce el archivo REPORTE.DAT:

```
u =   1.000  h =   0.020  T final =  10.000   orden =   4

La matriz de estado A es:
   0.000    1.000    0.000    0.000
   0.000    0.000    1.000    0.000
   0.000    0.000    0.000    1.000

   0.000  -75.000  -32.000   -8.000

La transpuesta de la matriz para el control, B', es:
   0.000    0.000    5.000   60.000

Condiciones iniciales:
   0.000    0.000    0.000    0.000


La matriz de salida, C, es:
   1.000    0.000    0.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000   8.000   32.000   75.000   -0.000

  det(A) = 0
las raíces del sistema son:
 p1 =    0.0000
 p2 =   -4.5880
 p3 =   -1.7060   -   3.6660j
 p4 =   -1.7060   +   3.6660j
```

Notar que los polos de lazo abierto, dados en REPORTE.DAT, son diferentes a los polos de lazo cerrado.

4. Usar comando **Gráfica** para ver la gráfica del sistema:



5. Usar **Imprimir** para producir el archivo PA411.prn:

```
                 MATRICES Y FUNCIONES                        Xi(t)   Xi(0) Gráf
A1i    0.00    1.00    0.00    0.00                           1.000    0.00    1
A2i    0.00    0.00    1.00    0.00                            0.00    0.00   -1
A3i    0.00    0.00    0.00    1.00                           -0.00    0.00    1
A4i    0.00  -75.00  -32.00   -8.00                           0.001    0.00    0
A5i                                                            0.00            2
A6i                                                           0.000            1
A7i                                                            0.00            1
Bi1    0.00    0.00    5.00   60.00                          Y1(t)=    1.00    2
Bi2                                                          Y2(t)=            1
C1i    1.00    0.00    0.00    0.00                            0.00            2
C2i                                                                           -1
G1     1-X1                                                                    1
G2 1+((9*COS(3*T)-17*SEN(3*T))*EXP(-T)-(33*COS(T)+39*SEN(T))*EXP(-3*T))/24_
Ho Lapso inicial de integracion =      0.02     sistema:   pa411.sis
Ha  Lapso actual de integracion =      0.00
Ti Tiempo inicial de simulacion =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulacion =     10.00 OK tabla en SALIDA.PRN
Tf   Tiempo final de simulacion =     10.00
Eo Error inicial de integracion =   0.00002  Iteraciones =     2724
Ea  Error actual de integracion =   0.00000                      orden:  4
```

6. Salir del programa y ver archivo SALIDA.PRN con un editor:

tiempo	x1	x2	x3	x4	salida 1	control 1
0.000	0.000	0.000	0.000	0.000	0.000	1.000
0.020	0.000	0.001	0.113	1.052	0.000	1.000
0.040	0.000	0.005	0.243	1.887	0.000	1.000
0.060	0.000	0.011	0.387	2.517	0.000	1.000
0.080	0.001	0.021	0.542	2.954	0.001	0.999
0.100	0.001	0.033	0.704	3.217	0.001	0.999
0.120	0.002	0.049	0.870	3.324	0.002	0.998
0.140	0.003	0.068	1.036	3.292	0.003	0.997
0.160	0.005	0.090	1.200	3.136	0.005	0.995
0.180	0.007	0.116	1.360	2.873	0.007	0.993
0.200	0.009	0.145	1.513	2.515	0.009	0.991
0.220	0.013	0.177	1.658	2.077	0.013	0.987
0.240	0.016	0.211	1.793	1.572	0.016	0.984
0.260	0.021	0.248	1.917	1.010	0.021	0.979
0.280	0.026	0.288	2.028	0.402	0.026	0.974
0.300	0.033	0.329	2.127	-0.240	0.033	0.967
0.320	0.040	0.373	2.212	-0.907	0.040	0.960
0.340	0.048	0.418	2.282	-1.591	0.048	0.952
0.360	0.056	0.464	2.338	-2.284	0.056	0.944
0.380	0.066	0.511	2.379	-2.978	0.066	0.934
0.400	0.077	0.559	2.405	-3.666	0.077	0.923
0.420	0.089	0.607	2.417	-4.342	0.089	0.911
0.440	0.101	0.656	2.414	-5.002	0.101	0.899
0.460	0.115	0.704	2.396	-5.638	0.115	0.885
0.480	0.129	0.751	2.365	-6.248	0.129	0.871
0.500	0.145	0.798	2.320	-6.828	0.145	0.855
.....	.....	.....	.....	.....	.....	......
9.820	1.000	0.000	-0.000	0.000	1.000	-0.000
9.840	1.000	0.000	-0.000	0.000	1.000	-0.000
9.860	1.000	0.000	-0.000	0.000	1.000	-0.000
9.880	1.000	0.000	-0.000	0.000	1.000	-0.000
9.900	1.000	0.000	-0.000	0.000	1.000	-0.000
9.920	1.000	0.000	-0.000	0.001	1.000	-0.000
9.940	1.000	0.000	-0.000	0.001	1.000	-0.000
9.960	1.000	0.000	-0.000	0.001	1.000	-0.000
9.980	1.000	0.000	-0.000	0.001	1.000	-0.000

9 Ejemplo 5: Control óptimo de un Servosistema

Consideremos el problema de traer al origen al sistema de segundo orden, dado adelante, en un tiempo mínimo, si el control es del tipo todo-nada (bang-bang). El ejemplo está tomado de [10].

La planta tiene raíces reales y viene dada por:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -0.5x_2 + u$$

El control óptimo, como función del estado, es:

$$u = -\,sig(x_1 + 2x_2 - 4sig(x_2)\ln(1 + 0.5|x_2|))$$

Pasos:

1. Llamar al programa agregando el nombre del archivo que contiene al sistema:

   C:\ALIN>alin bang.sis

2. Imprimir la pantalla de trabajo, usando el comando **Imprimir**:

```
                 MATRICES Y FUNCIONES                        Xi(t)   Xi(0) Gráf
A1i    0.00    1.00                                          -0.000    2.50    1
A2i    0.00   -0.50                                            0.01    1.00   -1
A3i                                                            0.00    0.00    1
A4i                                                           0.000            0
A5i                                                            0.00            2
A6i                                                           0.000            1
A7i                                                            0.00            1
Bi1    0.00    1.00                                          Y1(t)=   -0.00    2
Bi2                                                          Y2(t)=            1
A1i    1.00    0.00                                            0.00            2
C2i                                                                           -1
G1 -SIG(X1+2*X2-4*SIG(X2)*LN(1+0.5*ABS(X2)))_                               1
G2
Ho Lapso inicial de integración =     0.010     sistema:   bang.sis
Ha  Lapso actual de integración =     0.005
Ti Tiempo inicial de simulación =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulación =      5.00 OK tabla en SALIDA.PRN
Tf   Tiempo final de simulación =      5.00
Eo Error inicial de integración =   0.00002  Iteraciones =      756
Ea  Error actual de integración =   0.00000                       orden:  2
```

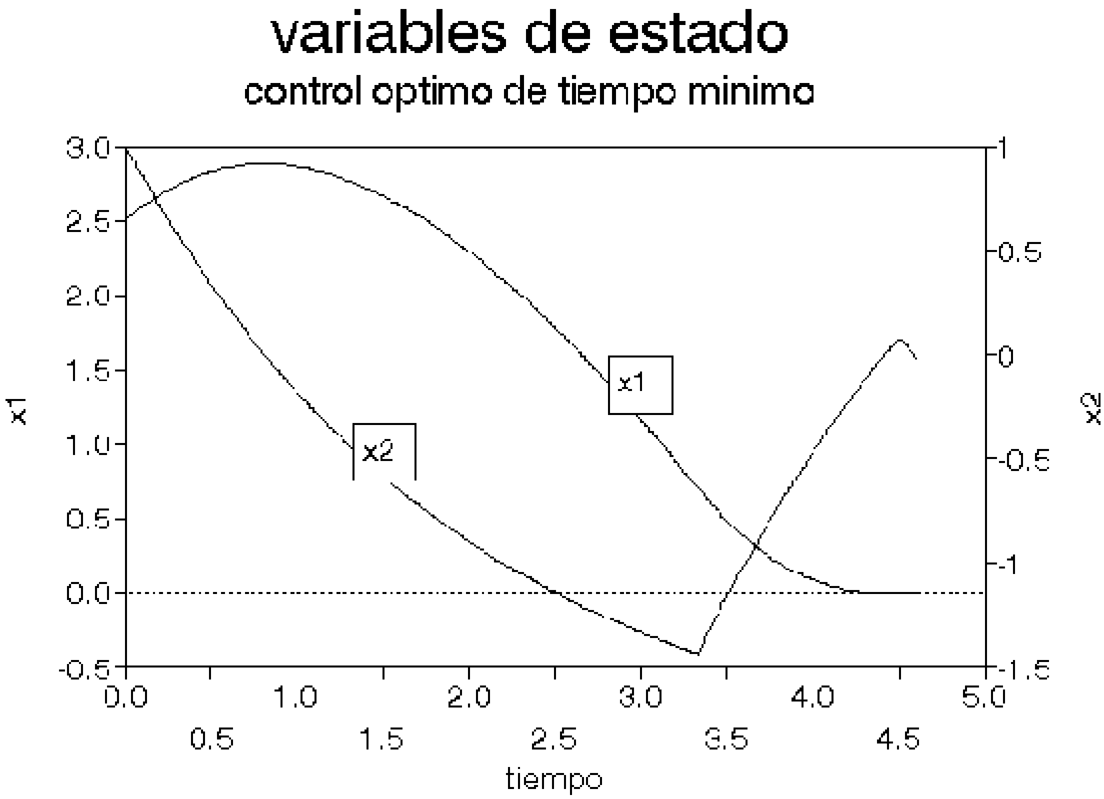
3. Editar la fórmula en G1 para reactivarla, y dar el comando **Inicia**, seguido de **Calcula,** lo que producirá el archivo REPORTE.DAT:

```
u =  -1.000  h =   0.010  T final =   5.000   orden =   2

La matriz de estado A es:
   0.000    1.000
   0.000   -0.500

La transpuesta de la matriz para el control, B', es:
   0.000    1.000

Condiciones iniciales:
   2.500    1.000

La matriz de salida, C, es:
   1.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000   0.500   -0.000

  det(A) = 0
las raíces del sistema son:
 p1 =    0.0000
 p2 =   -0.5000
```

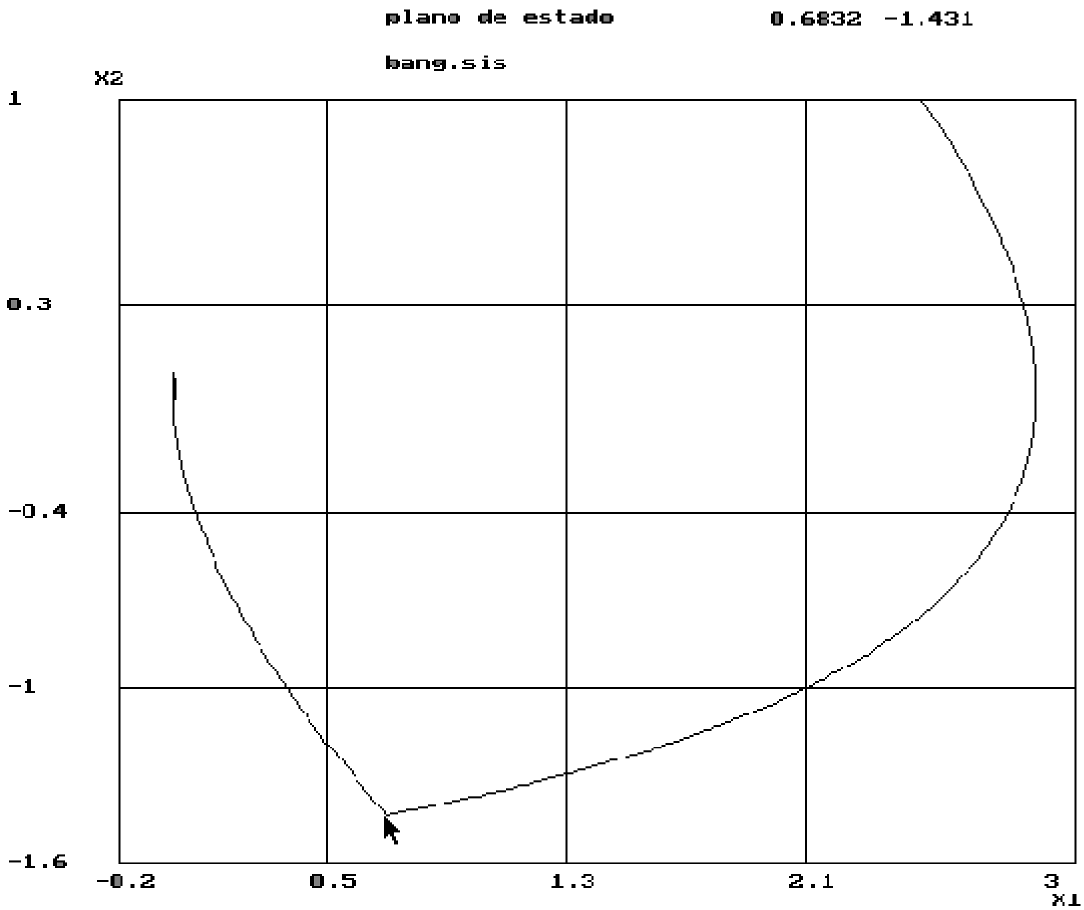
4. El archivo SALIDA.PRN se forma con los datos de salida:

tiempo	x1	x2	salida 1	control 1
0.000	2.500	1.000	2.500	-1.000
0.010	2.510	0.985	2.510	-1.000
0.020	2.520	0.970	2.520	-1.000
0.030	2.529	0.955	2.529	-1.000
0.040	2.539	0.941	2.539	-1.000
0.050	2.548	0.926	2.548	-1.000
0.060	2.557	0.911	2.557	-1.000
0.070	2.566	0.897	2.566	-1.000
0.080	2.575	0.882	2.575	-1.000
0.090	2.584	0.868	2.584	-1.000
0.100	2.592	0.854	2.592	-1.000
0.110	2.601	0.840	2.601	-1.000
0.120	2.609	0.825	2.609	-1.000
0.130	2.617	0.811	2.617	-1.000
0.140	2.625	0.797	2.625	-1.000
0.150	2.633	0.783	2.633	-1.000
0.160	2.641	0.769	2.641	-1.000
0.170	2.649	0.756	2.649	-1.000
0.180	2.656	0.742	2.656	-1.000
0.190	2.663	0.728	2.663	-1.000
0.200	2.671	0.715	2.671	-1.000
0.210	2.678	0.701	2.678	-1.000
0.220	2.685	0.688	2.685	-1.000
0.230	2.691	0.674	2.691	-1.000
0.240	2.698	0.661	2.698	-1.000
0.250	2.705	0.648	2.705	-1.000
0.260	2.711	0.634	2.711	-1.000
0.270	2.717	0.621	2.717	-1.000
0.280	2.723	0.608	2.723	-1.000
0.290	2.729	0.595	2.729	-1.000
0.300	2.735	0.582	2.735	-1.000
0.310	2.741	0.569	2.741	-1.000
0.320	2.747	0.557	2.747	-1.000
0.330	2.752	0.544	2.752	-1.000
0.340	2.757	0.531	2.757	-1.000
0.350	2.763	0.519	2.763	-1.000
0.360	2.768	0.506	2.768	-1.000
0.370	2.773	0.494	2.773	-1.000
0.380	2.778	0.481	2.778	-1.000
0.390	2.782	0.469	2.782	-1.000
0.400	2.787	0.456	2.787	-1.000
0.410	2.792	0.444	2.792	-1.000
0.420	2.796	0.432	2.796	-1.000
0.430	2.800	0.420	2.800	-1.000
0.440	2.804	0.408	2.804	-1.000
0.450	2.808	0.396	2.808	-1.000
0.460	2.812	0.384	2.812	-1.000
0.470	2.816	0.372	2.816	-1.000
0.480	2.820	0.360	2.820	-1.000
0.490	2.823	0.348	2.823	-1.000
0.500	2.826	0.337	2.826	-1.000
0.510	2.830	0.325	2.830	-1.000
0.520	2.833	0.314	2.833	-1.000
0.530	2.836	0.302	2.836	-1.000
0.540	2.839	0.291	2.839	-1.000
0.550	2.842	0.279	2.842	-1.000
0.560	2.845	0.268	2.845	-1.000
0.570	2.847	0.256	2.847	-1.000
0.580	2.850	0.245	2.850	-1.000
0.590	2.852	0.234	2.852	-1.000
0.600	2.854	0.223	2.854	-1.000
0.610	2.856	0.212	2.856	-1.000

....



4.200	0.018	-0.229	0.018	1.000
4.210	0.016	-0.217	0.016	1.000
4.220	0.014	-0.206	0.014	1.000
4.230	0.012	-0.195	0.012	1.000
4.240	0.010	-0.184	0.010	1.000
4.250	0.008	-0.174	0.008	1.000
4.260	0.006	-0.163	0.006	1.000
4.270	0.005	-0.152	0.005	1.000
4.280	0.003	-0.141	0.003	1.000
4.290	0.002	-0.131	0.002	1.000
4.300	0.001	-0.120	0.001	1.000
4.310	-0.001	-0.109	-0.001	1.000
4.320	-0.002	-0.099	-0.002	1.000
4.330	-0.003	-0.088	-0.003	1.000
4.340	-0.003	-0.078	-0.003	1.000
4.350	-0.004	-0.068	-0.004	1.000
4.360	-0.005	-0.057	-0.005	1.000
4.370	-0.005	-0.047	-0.005	1.000
4.380	-0.006	-0.037	-0.006	1.000
4.390	-0.006	-0.027	-0.006	1.000
4.400	-0.006	-0.017	-0.006	1.000
4.410	-0.006	-0.007	-0.006	1.000
4.420	-0.006	0.004	-0.006	1.000
4.430	-0.006	0.013	-0.006	1.000
4.440	-0.006	0.023	-0.006	1.000
4.450	-0.006	0.033	-0.006	1.000

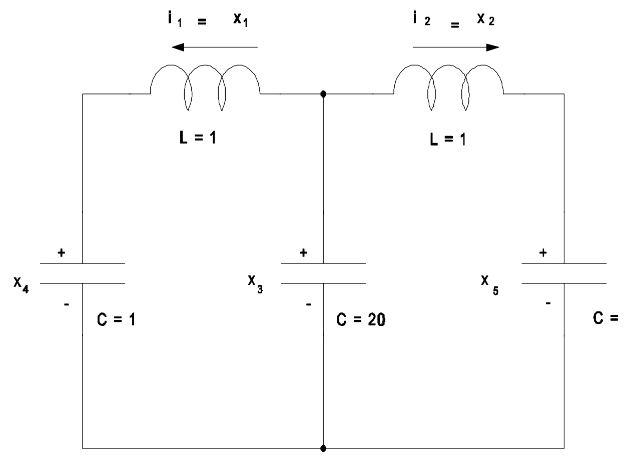


4.460	-0.005	0.043	-0.005	1.000
4.470	-0.005	0.053	-0.005	1.000
4.480	-0.004	0.062	-0.004	1.000
4.490	-0.004	0.072	-0.004	1.000
4.500	-0.003	0.077	-0.003	1.000
4.510	-0.002	0.074	-0.002	-1.000
4.520	-0.002	0.063	-0.002	-1.000
4.530	-0.001	0.053	-0.001	-1.000
4.540	-0.000	0.043	-0.000	-1.000
4.550	-0.000	0.033	-0.000	-1.000
4.560	0.000	0.023	0.000	-1.000
4.570	0.000	0.013	0.000	-1.000
4.580	0.000	0.003	0.000	-1.000
4.590	0.000	-0.007	0.000	-1.000
4.600	0.000	-0.017	0.000	-1.000
4.610	0.000	-0.012	0.000	1.000
4.620	-0.000	-0.002	-0.000	1.000
4.630	0.000	0.001	0.000	-1.000
4.640	0.000	-0.003	0.000	1.000
4.650	0.000	-0.002	0.000	-1.000
4.660	-0.000	-0.003	-0.000	1.000
4.670	-0.000	-0.000	-0.000	1.000
.....	.....	.....	.....	

5. Con la tecla F5 se puede generar la gráfica para el plano de fase y con QUATTRO la gráfica contra el tiempo de ambas variables de estado.

10. Ejemplo 6: Circuito eléctrico de quinto orden

Muchos ejemplos de sistemas físicos consisten de componentes casi idénticos acoplados débilmente entre sí. El enlace entre ambos sirve como un puente para transferir energía de un subsistema al otro y viceversa, una y otra vez. Presentamos un circuito eléctrico que consiste de dos redes eléctricas LC conectadas por un capacitor relativamente grande.



El modelo matemático para dicho circuito viene dado por:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -\frac{1}{20} & -\frac{1}{20} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Pasos:

1. Ejecutar el programa **ALIN** anexando el nombre del sistema:

   C:\ALIN>alin athans.sis

2. Imprimir la pantalla de trabajo (en archivo ATHANS.PRN):

```
               MATRICES Y FUNCIONES                       Xi(t)   Xi(0) Gráf
A1i    0.00    0.00    1.00   -1.00    0.00                6.406    0.00    1
A2i    0.00    0.00    1.00    0.00   -1.00                 1.85    0.00    2
A3i   -0.05   -0.05    0.00    0.00    0.00                 0.60    0.00    1
A4i    1.00    0.00    0.00    0.00    0.00                3.076   10.00    0
A5i    0.00    1.00    0.00    0.00    0.00                -4.99    0.00    2
A6i                                                        0.000            1
A7i                                                         0.00            1
Bi1    0.00    0.00    1.00                               Y1(t)=    6.72    2
Bi2                                                       Y2(t)=            1
C1i    1.00                                                                 2
C2i                                                                        -1
G1     0.00                                                                 1
G2
Ho Lapso inicial de integracion =     0.050     sistema:   athans.sis
Ha  Lapso actual de integracion =     0.006
Ti Tiempo inicial de simulacion =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulacion =    100.00 OK tabla en SALIDA.PRN
Tf   Tiempo final de simulacion =    100.00
```

```
Eo Error inicial de integracion =   0.00002  Iteraciones =    16491
Ea  Error actual de integracion =   0.00005                        orden:  5
```

3. Ejecutar **Inicia** seguido de **Calcula** para obtener el archivo REPORTE.DAT:

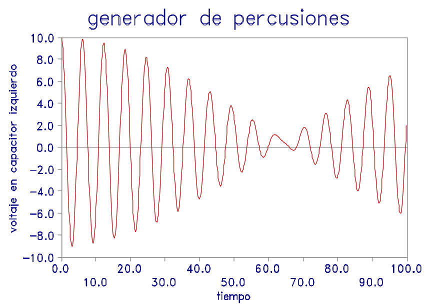
```
u =   0.000  h =   0.050  T final = 100.000   orden =   5

La matriz de estado A es:
   0.000    0.000    1.000   -1.000    0.000
   0.000    0.000    1.000    0.000   -1.000
  -0.050   -0.050    0.000    0.000    0.000
   1.000    0.000    0.000    0.000    0.000
   0.000    1.000    0.000    0.000    0.000

La transpuesta de la matriz para el control, B', es:
   0.000    0.000    1.000    0.000    0.000

Condiciones iniciales:
   0.000    0.000    0.000   10.000    0.000

La matriz de salida, C, es:
   1.000    0.000    0.000    0.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000  -0.000    2.100   -0.000    1.100   -0.000
  det(A) = 0
las raíces del sistema son:
 p1 =    0.0000
 p2 =    0.0000   -   1.0000j
 p3 =    0.0000   +   1.0000j
 p4         =   -1.0490j
 p5         =    1.0490j
```

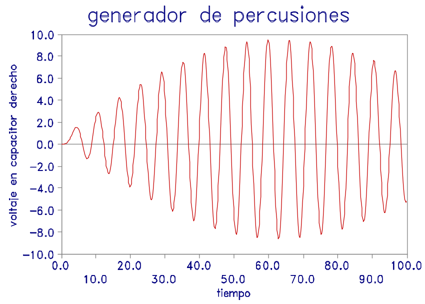
4. El archivo producido por la integración (SALIDA.PRN) se ve así:

tiempo	x1	x2	x3	x4	x5	salida 1
0.000	0.000	0.000	0.000	10.000	0.000	0.000
0.050	-0.499	0.000	0.001	9.984	0.000	-0.499
0.250	-2.471	0.001	0.016	9.684	0.000	-2.471
0.450	-4.339	0.008	0.050	8.999	0.001	-4.339
0.650	-6.025	0.022	0.102	7.958	0.004	-6.025
0.850	-7.458	0.048	0.169	6.604	0.011	-7.458
1.050	-8.578	0.086	0.249	4.994	0.024	-8.578
1.250	-9.340	0.138	0.338	3.195	0.046	-9.340
1.450	-9.710	0.203	0.432	1.283	0.080	-9.710
1.650	-9.675	0.278	0.527	-0.662	0.128	-9.675
1.850	-9.235	0.361	0.618	-2.559	0.192	-9.235
2.050	-8.410	0.447	0.703	-4.329	0.273	-8.410
2.250	-7.234	0.531	0.776	-5.898	0.370	-7.234
2.450	-5.757	0.607	0.836	-7.201	0.485	-5.757
2.650	-4.040	0.670	0.878	-8.182	0.613	-4.040
2.850	-2.156	0.712	0.903	-8.803	0.751	-2.156
3.050	-0.185	0.728	0.907	-9.036	0.896	-0.185
3.250	1.793	0.715	0.892	-8.873	1.040	1.793
3.450	3.694	0.668	0.857	-8.321	1.179	3.694
3.650	5.438	0.585	0.805	-7.404	1.305	5.438
3.850	6.952	0.468	0.737	-6.159	1.411	6.952
4.050	8.175	0.317	0.658	-4.640	1.490	8.175
4.250	9.054	0.136	0.569	-2.910	1.535	9.054
4.450	9.554	-0.068	0.475	-1.043	1.542	9.554
4.650	9.654	-0.288	0.380	0.885	1.507	9.654

4.850	9.351	-0.515	0.289	2.792	1.426	9.351
5.050	8.657	-0.739	0.205	4.598	1.300	8.657
5.250	7.603	-0.950	0.132	6.229	1.131	7.603
5.450	6.233	-1.135	0.073	7.617	0.922	6.233
5.650	4.605	-1.285	0.031	8.703	0.679	4.605
5.850	2.787	-1.391	0.007	9.444	0.411	2.787
6.050	0.856	-1.444	0.003	9.808	0.126	0.856
6.250	-1.107	-1.438	0.019	9.781	-0.163	-1.107
6.450	-3.021	-1.370	0.054	9.365	-0.445	-3.021
6.650	-4.805	-1.238	0.106	8.578	-0.707	-4.805
6.850	-6.385	-1.045	0.174	7.454	-0.936	-6.385
7.050	-7.695	-0.796	0.254	6.040	-1.121	-7.695
7.250	-8.681	-0.498	0.343	4.396	-1.251	-8.681
....	.....	.....	.....	.....	.....	.....
94.836	1.192	6.458	0.246	6.371	-1.300	1.192
95.036	-0.056	6.628	0.175	6.485	0.013	-0.056
95.236	-1.314	6.521	0.116	6.346	1.333	-1.314
95.436	-2.530	6.143	0.072	5.960	2.603	-2.530
95.636	-3.652	5.514	0.044	5.339	3.772	-3.652
95.836	-4.631	4.662	0.035	4.507	4.793	-4.631
96.036	-5.426	3.624	0.044	3.497	5.623	-5.426
96.236	-6.001	2.445	0.071	2.350	6.231	-6.001
96.436	-6.329	1.176	0.115	1.113	6.593	-6.329
96.636	-6.394	-0.128	0.173	-0.164	6.697	-6.394
96.836	-6.192	-1.415	0.244	-1.427	6.541	-6.192
97.035	-5.728	-2.629	0.324	-2.623	6.134	-5.728
97.235	-5.019	-3.722	0.410	-3.701	5.496	-5.019
97.435	-4.093	-4.649	0.498	-4.614	4.655	-4.093
97.635	-2.986	-5.372	0.584	-5.324	3.649	-2.986
97.835	-1.745	-5.865	0.664	-5.798	2.521	-1.745
98.035	-0.419	-6.109	0.735	-6.014	1.319	-0.419
98.235	0.937	-6.096	0.793	-5.962	0.094	0.937
98.435	2.264	-5.830	0.837	-5.640	-1.103	2.264
98.635	3.508	-5.325	0.864	-5.060	-2.222	3.508
98.835	4.616	-4.604	0.873	-4.244	-3.217	4.616
99.035	5.539	-3.699	0.864	-3.224	-4.050	5.539
99.235	6.237	-2.652	0.836	-2.042	-4.686	6.237
99.435	6.679	-1.506	0.792	-0.746	-5.102	6.679
99.635	6.845	-0.310	0.734	0.612	-5.283	6.845

5. Con la hoja de cálculo QUATTRO PRO o alguna similar se procesa el archivo SALIDA.PRN y se obtiene la gráfica para x_4 y x_5. Contrastar contra página 332 de [4].





11 Ejemplo 7: Cálculo de Matriz de Realimentación (2do orden)

Resolvamos ahora el ejercicio A10.5 de [1], que se origina la ecuación diferencial de segundo orden:

$$y'' + 3y' + 2y = 2u(t)$$

cuyas raíces son reales e iguales a -1 y -2 por lo que tenemos el **caso estable.** A continuación se presenta la solución numérica del caso en que la señal de control u(t) se genera con retroalimentación de estado:

Se define $\mathbf{x_1 = y}$ = salida del sistema y a $\mathbf{x_2}$ = a la primera derivada de y:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -2x_1 - 3x_2 + 2u$$

Si se eligen polos de lazo cerrado algo más rápidos que los originales, digamos en -3 y en -5, la matriz de ganancia para retroalimentar el estado es **K** = [6.5 2.5]; (calculada con **ALIN** por el método de Ackermann) lo que implica que la expresión para el control es:

$$u = -Kx = -6.5x_1 - 2.5x_2$$

El polinomio característico para el sistema deseado es:

$$s^2 + 8s + 15$$

Pasos:

1) Ejecutar **ALIN** dando el nombre del archivo para dicho sistema:

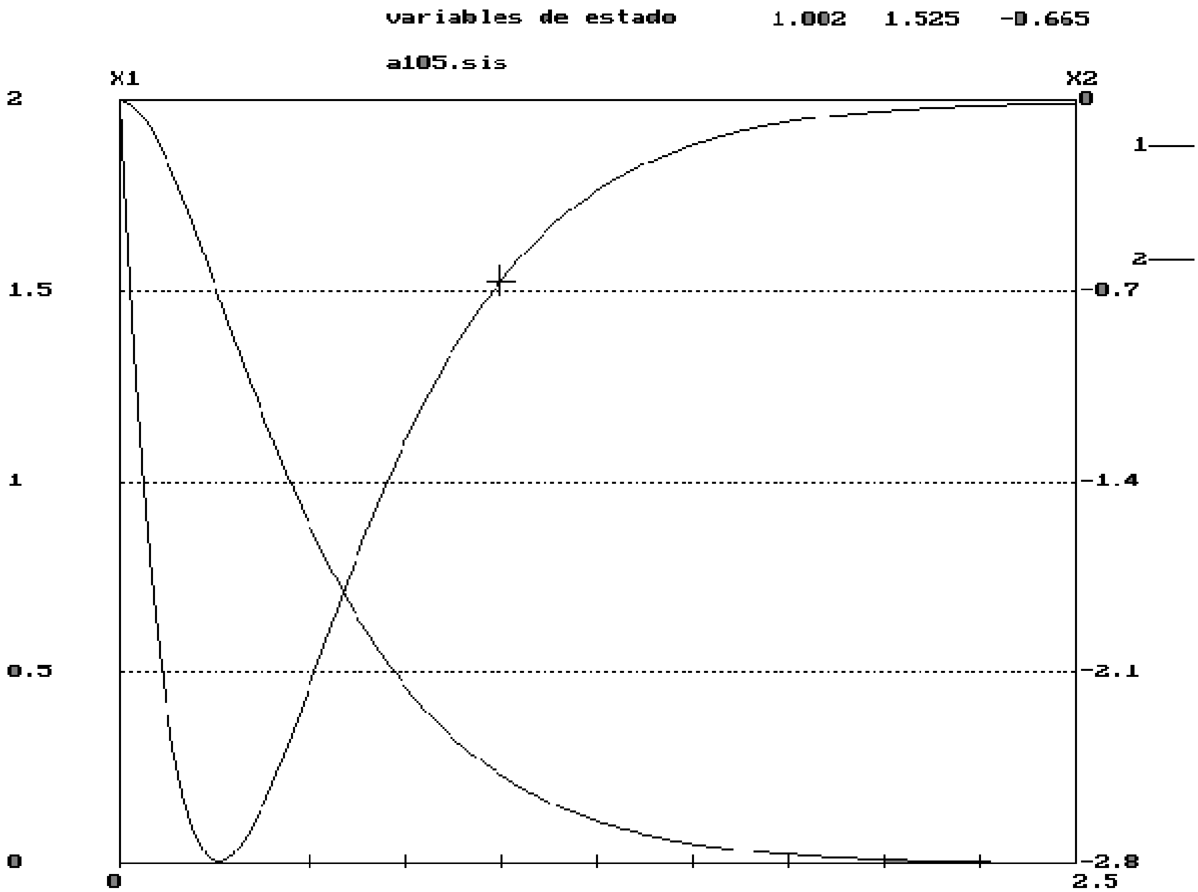
C:\ALIN>alin a105.sis

Las condiciones iniciales son x_1 = 2.0 y x_2 = 0. La excitación es nula (igual a 0). Obtener con F4 el modelo de estado espacio a partir de la función de transferencia.

2) Ejecutar el comando **Inicia** seguido de **Calcula**. Al terminar la segunda etapa, comparar los valores obtenidos con los teóricos. El análisis preliminar se tiene en el archivo REPORTE.DAT:

```
u = -13.000  h =   0.005  T final =   2.500   orden =   2

La matriz de estado A es:
   0.000    1.000
  -2.000   -3.000
La transpuesta de la matriz para el control, B', es:
   0.000    2.000

Condiciones iniciales:
   2.000    0.000

La matriz de salida, C, es:
   1.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000   3.000    2.000
la inversa de la matriz A es:
   -1.500   -0.500
    1.000    0.000
las raíces del sistema son:
 p1 =   -1.0000
 p2 =   -2.0000
```

ALIN usa el método de Ackermann para calcular la matriz **K**, de realimentación de estado del sistema y el método de Newton para el cálculo de las raíces de lazo abierto. Debe notarse que es muy fácil formar el modelo matemático para un sistema a partir de una función de transferencia dada.



3) Ejecutar el comando **Gráfica** para ver las variables de estado contra el tiempo. Mover el cursor del ratón sobre una y otra curva para ver el efecto de vernier que tiene el ratón. Oprimir cualquier tacla para regresar a la pantalla alfanumérica:

4) Ejecutar **Terminar** y, si se desea, examinar el archivo SALIDA.PRN con un editor.

12 Ejemplo 8: Observador Reducido y Acción Integral, con Realimentación de Estado

Veamos ahora un ejercicio similar el problema B-10-6 de [1], que se complementa con el ejemplo 10.6 de dicho texto que pide la determinación de la matriz de ganancia para formar el observador reducido del estado, pero le agregaremos aquí acción integral y además la retroalimentación de estado para formar la señal de control.

El sistema viene dado por:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{10}{(s+1)(s+2)(s+3)}$$

al representar (33) en variables de estado, resulta:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -6 & -11 & -6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 10\end{bmatrix}u$$

con C = [1 0 0]. Como se agregará un observador reducido mediante el subcomando **obs. mínimo** del comando **útiles**, el sistema aumenta su orden en dos:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ -6 & -11 & -6 & 0 & 0\\ 5 & 0 & 0 & -2 & 1\\ -100 & 0 & 0 & -20 & -6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 10\\ 0\\ 10\end{bmatrix}u$$

Con ayuda de [4] se puede calcular la inversa de la matriz T = MW:

$$T^{-1}=\begin{bmatrix}0 & 0 & 0 & -0.1\\ 0.1 & 0 & 0 & 0\\ 0 & 0.1 & 0 & 0\\ 0 & 0 & 0.1 & 0\end{bmatrix}$$

La matriz K_e, de ganancia del observador, especificando sus polos en -4 ±j4, es:

$$K_e=\begin{bmatrix}2\\ 9\end{bmatrix}$$

Al añadir la acción integral al sistema, éste queda de sexto orden:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ -6 & -11 & -6 & 0 & 0 & 0\\ 5 & 0 & 0 & -2 & 1 & 0\\ -100 & 0 & 0 & -20 & -6 & 0\\ -1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 10\\ 0\\ 10\\ 0\end{bmatrix}u+\begin{bmatrix}0\\ 0\\ 0\\ 0\\ 0\\ 1\end{bmatrix}r$$

Si se desean los polos de lazo cerrado en -6 y en -10 y un par complejo conjugado en -2 ±j3.464, **ALIN** nos permite, con el método de Ackermann, calcular la matriz K:

$$K = [\ 48.9989\ \_\ 12.8999\ \_\ 1.400\ \_\ -95.9958\ ]$$

Notar que el cálculo para la matriz de ganancia se hace sobre el sistema de tercer orden original con la acción integral, empleando el principio de la separación. El programa MATLAB arroja el mismo resultado.

Pasos:

1. Ejecutar **ALIN** junto con el archivo que tiene el sistema:

C:\ALIN>alin b106.sis

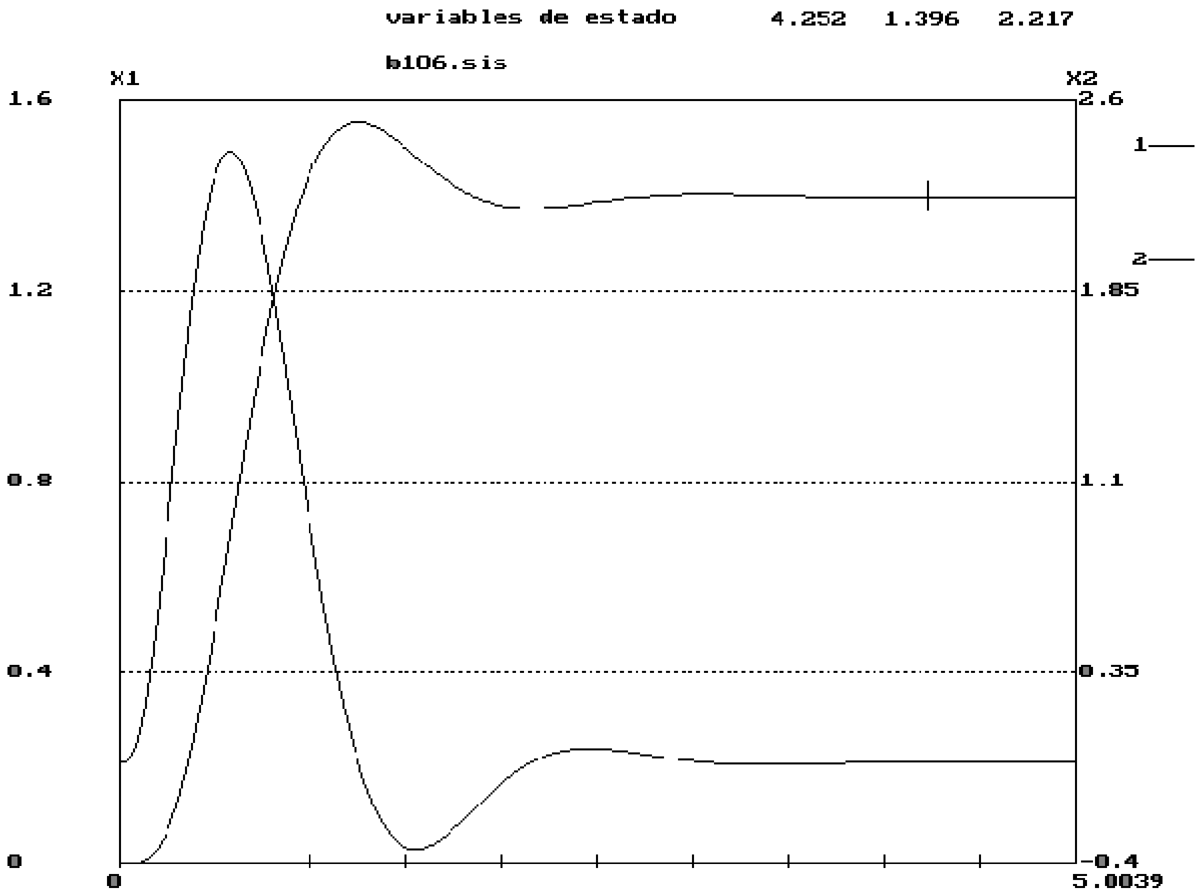
2. Reactivar la fórmula para U1 y ejecutar el comando **Inicia** para iniciar todo.

3. Ejecutar el comando **Calcula** que produce el archivo REPORTE.DAT:

```
u =   0.000  h =   0.005  T final =   5.000   orden =   6
La matriz de estado A es:
   0.000    1.000    0.000    0.000    0.000    0.000
   0.000    0.000    1.000    0.000    0.000    0.000
  -6.000  -11.000   -6.000    0.000    0.000    0.000
   5.000    0.000    0.000   -2.000    1.000    0.000
-100.000    0.000    0.000  -20.000   -6.000    0.000
  -1.000    0.000    0.000    0.000    0.000    0.000

La transpuesta de la matriz para el control, B', es:
   0.000    0.000   10.000    0.000   10.000    0.000
   0.000    0.000    0.000    0.000    0.000    1.000

Condiciones iniciales:
   0.000    0.000    0.000    0.000    0.000    0.000

La matriz de salida, C, es:
   1.000    0.000    0.000    0.000    0.000    0.000

Coeficientes de ecuación característica en orden descendente:
1.000  14.000   91.000  286.000  400.000  192.000   -0.000

  det(A) = 0
las raíces del sistema son:
 p1 =    0.0000
 p2 =   -1.0000
 p3 =   -2.0000
 p4 =   -3.0000
 p5 =   -4.0000   -   4.0000j
 p6 =   -4.0000   +   4.0000j
```

Notar que los polos de lazo abierto, dados en REPORTE.DAT, son diferentes a los polos de lazo cerrado. Notar que **ALIN** reporta los polos del observador y de la acción integral añadida.

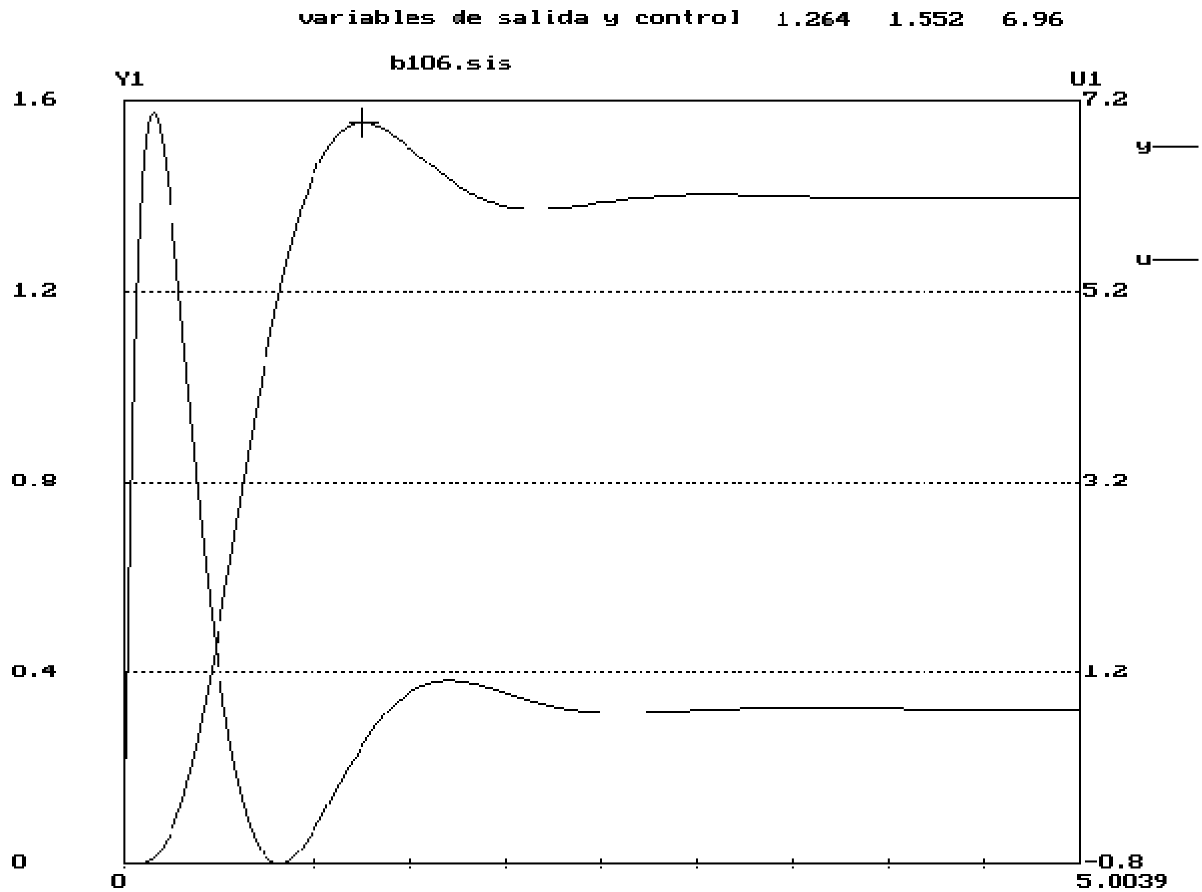
4. Usar comando **Gráfica** para ver la gráfica del sistema. Observar que debido al efecto del observador el sobretiro negativo es mayor que el primer sobretiro.



5. Usar **Imprimir** para producir el archivo B106.PRN, donde la función de retroalimentación en función del estado se muestra desactivada, para poder apreciarla; Notar que la entrada de escalón es de magnitud 1.4:

```
                 matrices y funciones                         Xi(t)    Xi(0)   Ki
A1i    0.00    1.00    0.00    0.00    0.00    0.00           1.400    0.00    1
A2i    0.00    0.00    1.00    0.00    0.00    0.00           -0.00    0.00   -1
A3i   -6.00  -11.00   -6.00    0.00    0.00    0.00           -0.00    0.00    1
A4i    5.00    0.00    0.00   -2.00    1.00    0.00          -2.800    0.00    0
A5i -100.00    0.00    0.00  -20.00   -6.00    0.00          -12.60    0.00    2
A6i   -1.00    0.00    0.00    0.00    0.00    0.00           0.723    0.00    2
A7i                                                            0.00            9
Bi1    0.00    0.00   10.00    0.00   10.00    0.00          Y1(t)=    1.40    2
Bi2    0.00    0.00    0.00    0.00    0.00    1.00          Y2(t)=            1
C1i    1.00    0.00    0.00    0.00    0.00    0.00            0.00            2
C2i                                                                           -1
G1 -48.9989*X1-12.8999*(X4+2*X1)-1.4*(X5+9*X1)+95.9958*X6_                     1
G2     1.40
Ho Lapso inicial de integración =     0.005     sistema:   b106.sis
Ha  Lapso actual de integración =     0.001
Ti Tiempo inicial de simulación =      0.00       Autor:   S. Saucedo
Ta  Tiempo actual de simulación =      5.00 OK tabla en SALIDA.PRN
Tf   Tiempo final de simulación =      5.00
Eo Error inicial de integración =   0.00002  Iteraciones =     3214
Ea  Error actual de integración =   0.00000                       orden:  6
```

6. Usar tecla F5 para generar la gráfica del plano de estado del sistema o el comando **gráfica** para ver variables de salida y de control:



APENDICE A BIBLIOGRAFIA

[1] **Ogata, Katsuhiko** Ingeniería de Control Moderna, Prentice Hall, 1993.

[2] **Wiberg, D. M.** Espacio de Estado y Sistemas Lineales Schaum-McGraw-Hill, 1973.

[3] **Johnson, D. E. et al** Análisis Básico de Circuitos Eléctricos , Prentice-Hall, 1991.

[4] **Athans, Michel et al** Systems, Networks, & Computation, McGraw-Hill, 1974.

[5] **Constantinides, Alkis** Applied Numerical Methods with Personal Computers, McGraw-Hill IE, 1987.

[6] **Chapra, S.C. y Canale R.P.** "Métodos Numéricos para Ingenieros; Con Aplicaciones en Computadoras Personales", McGraw-Hill,1988.

[7] **Saucedo, F. S.** "Programa EDON: Reporte Técnico", ESIME-IPN, 1995.

[8] **Pappas, C.H. y Murray W.H.** "Manual de Borland C++ VERSION 3.1", OborneMcGraw-Hill, 1993.

[9] **Borland** "C++ Library Reference VERSION 4.0", 1993.

[10] **Sage, A. P.** Optimum Systems Control, Prenctice Hall, 1968.

[11] **Petkov,P. H. et al** Computational Methods for Linear Control Systems, Prentice-Hall, 1991.

APENDICE A METODO DE NEWTON PARA ENCONTRAR LAS RAICES DE UN POLINOMIO

Veremos ahora el método de Newton, explicado en [5], que permite calcular las raíces complejas de un polinomio, además de las raíces reales; incluyendo los casos en que hay más de una pareja de raíces complejas. Las partes real e imaginaria de las raíces se tratan de manera separada y la búsqueda es conducida en la parte real y en la parte imaginaria de modo simultáneo, usando el método de Newton.

Considerar el polinomio:

$$a_N\, x^N + a_{N-1}\, x^{N-1} + \cdots + a_1 x + a_0 = 0$$

que puede escribirse como:

$$f(x) = \sum_{n=0}^{N} a_n x^n = 0$$

La raíz compleja que satisface (41) es del tipo:

$$x = a + b\, i$$

sustituyendo (42) en (41) resulta en una ecuación que contiene términos reales e imaginarios:

$$f(x) = m(a\ ,b\ ) + n(a\ ,b\ )i$$

Los términos reales μ(α,β) están dados por la ecuación:

$$m(a\ ,b\ ) = \sum_{n=0}^{N} a_n a_n = 0$$

y los términos imaginarios por:

$$n(a\ ,b\ ) = \sum_{n=0}^{N} a_n b_n = 0$$

Las expresiones para α_n y β_n son como sigue:

$$n = 0 \quad \begin{cases} a_0 = 1.0 \\ b_0 = 0.0 \end{cases} \qquad n > 0 \quad \begin{cases} a_n = a\, a_{n-1} - b\, b_{n-1} \\ b_n = a\, b_{n-1} + b\, a_{n-1} \end{cases}$$

Las ecuaciones (44) y (45) constituyen dos ecuaciones no lineales simultáneas en las incógnitas α y β. El vector de corrección para α y β viene dado por:

$$da = -\frac{\left[m \frac{\partial n}{\partial b} - n \frac{\partial m}{\partial b} \right]}{\left[\frac{\partial m}{\partial a}\frac{\partial n}{\partial b} - \frac{\partial m}{\partial b}\frac{\partial n}{\partial a} \right]} \qquad db = -\frac{\left[n \frac{\partial m}{\partial a} - m \frac{\partial n}{\partial a} \right]}{\left[\frac{\partial m}{\partial a}\frac{\partial n}{\partial b} - \frac{\partial m}{\partial b}\frac{\partial n}{\partial a} \right]}$$

Si usamos las ecuaciones de Cauchy-Riemann:

$$\frac{\partial m}{\partial a} = \frac{\partial n}{\partial b} \;\; y \;\; \frac{\partial m}{\partial b} = -\frac{\partial n}{\partial a}$$

las ecuaciones (48) y (49) se reducen a:

$$da = \frac{n \frac{\partial m}{\partial b} - m \frac{\partial m}{\partial a}}{\left(\frac{\partial m}{\partial a} \right)^2 + \left(\frac{\partial m}{\partial b} \right)^2} \qquad db = -\frac{m \frac{\partial m}{\partial b} + n \frac{\partial m}{\partial a}}{\left(\frac{\partial m}{\partial a} \right)^2 + \left(\frac{\partial m}{\partial b} \right)^2}$$

Las derivadas parciales se evalúan diferenciando las ecuaciones (44) y (45):

$$\frac{\partial m}{\partial a} = \sum_{n=1}^{N} n a_n \, a_{n-1} \qquad \frac{\partial m}{\partial b} = -\sum_{n=1}^{N} n a_n \, b_{n-1}$$

Finalmente, los valores corregidos de α y β se calculan mediante:

$$a^{(i+1)} = a^{(i)} + d\, a^{(i)} \qquad b^{(i+1)} = b^{(i)} + d\, b^{(i)}$$

donde el superíndice (i) denota el número de iteración. Se necesita un valor inicial para α y β para iniciar el proceso iterativo. En el programa cuyo listado se da a continuación aplica un máximo de 500 iteraciones para hallar cada raíz y un máximo de cinco diferentes juegos de valores iniciales para α y β. El polinomio se reduce, mediante división sintética, cuando una raíz (real o compleja) es hallada. Las iteraciones finales para cada raíz se realizan sobre el polinomio original en vez del polinomio reducido para evitar los errores acumulados en el polinomio.

El programa para implementar el método descrito está dado en [5], en lenguaje BASIC.

APENDICE B LISTADO PARCIAL DEL PROGRAMA

Se ofrece a continuación el listado del módulo NEWTON.C que aplica el algoritmo explicado arriba.

```
// modulo NEWTON.C que calcula las raíces de un polinomio

#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <graphics.h>
```

```
#include <conio.h>
#include <math.h>
#include <dos.h>

//Prototipado de funciones:
void imprime(int N);
void busca(int N);

extern int IFIT;
extern float XCOF[9];
extern float COF[9],RaizR[9],RaizI[9];
extern FILE *datos;
int   newton(int M, float palfa[7])
 {
  int i,j,k,N;
  if(M <= 0 || M > 7) return 0;
  N=M;
  for(i=M;i>0;i--)
    XCOF[i] = palfa[M-i];
  XCOF[M+1] = 1.0;
  IFIT = 0;     N = M;
  busca(N);
  imprime(N);
  return 1;
  }
void imprime(int M)
 {
 int i;
 float yy;
     printf("\nlas raíces del sistema son:");
     fprintf(datos,"\nlas raíces del sistema son:");
     for(i=1;i<M+1;i++) { printf("\n"); fprintf(datos,"\n");
     yy = 1000;
     while(fabs(RaizR[i]*yy) > 32000) yy = yy/10;
     if(RaizR[i] != 0) {
     while(fabs(RaizR[i]*yy) < 1000) yy = 10*yy;
     RaizR[i] = floor(RaizR[i]*yy+0.5)/yy; }
     yy = 1000;
     while(fabs(RaizI[i]*yy) > 32000) yy = yy/10;
     if(RaizI[i] != 0) {
            while(fabs(RaizI[i]*yy) <  1000) yy = 10*yy;
            RaizI[i] = floor(RaizI[i]*yy + 0.5)/yy; }
     if(RaizI[i] == 0)
            {printf("p%i = %9.4f",i,RaizR[i]);
            fprintf(datos," p%i = %9.4f",i,RaizR[i]); continue;}
     if(RaizR[i] == 0)
            {printf("p%i          = %9.4fj",i,RaizI[i]);
            fprintf(datos," p%i         = %9.4fj",i,RaizI[i]); continue;}
     if(RaizI[i] > 0 )
            {printf("p%i = %9.4f   +%9.4fj", i, RaizR[i],fabs(RaizI[i]));
            fprintf(datos," p%i = %9.4f   +%9.4fj", i, RaizR[i],fabs(RaizI[i])); continue;}
     printf("p%i = %9.4f   -%9.4fj",i,RaizR[i],fabs(RaizI[i]));
     fprintf(datos," p%i = %9.4f   -%9.4fj",i,RaizR[i],fabs(RaizI[i]));
       }

       }
void busca(int NN)
 {
  int NX,NXX,MT,KJ1,IN,ContI,FI,N2,TEMPN;
  float UY,UX,YY,XX,YT,YT2,VV,UU,YPR,XPR,XO,YO,SumaC,XT,XT2;
  float Alfa,TEMP1,TEMP,DX,DY;
```

```
  int L,i,N;
  N = NN;
  NX = N;
  NXX = N+1;
  N2 = 1;
  KJ1 = N + 1;
  for(L=1;L<KJ1+1;L++) {
  MT = KJ1 - L + 1;
  COF[MT] = XCOF[L]; }
// Pone valores iniciales
  while(N>0) {
   XO = 0.005;
   YO = 0.01;
// contador de valor inicial en cero
  IN = 0;
bucle:
  XX = XO;
//Incrementa valores iniciales y contador
  XO = -10 * YO;
  YO = -10*XX;
//Fija  X y Y al valor actual
  XX = XO;
  YY = YO;
  IN++;
  goto evalua;
final:
  IFIT = 1;
  XPR = XX;
  YPR = YY;
//Evalua polinomios y derivadas
evalua:
  for(ContI=0;ContI<501;ContI++) {
  UX = 0;   UY = 0;   VV = 0;   YT = 0;   XT = 1;
  UU = COF[N+1];
  if(UU == 0) goto cero;
  for(i=1;i<N+1;i++)
   {
      L = N - i + 1;
      TEMP = COF[L];
      XT2 = XX*XT - YY*YT;   YT2 = XX*YT  + YY*XT;
      UU = UU + TEMP*XT2;  VV = VV + TEMP*YT2;
      FI = i;
      UX = UX + FI*XT*TEMP;
      UY = UY - FI*YT*TEMP;   XT = XT2;   YT = YT2;
   }
  SumaC = UX*UX + UY*UY;
  if(SumaC == 0) goto checa;
  DX = (VV * UY - UU * UX) / SumaC;
  XX = XX + DX;
  DY = -(UU * UY + VV * UX) / SumaC;
  YY = YY + DY;
  if(fabs(DY) + fabs(DX) - 0.00001 < 0) goto cerca;
// aumenta contador de iteraciones
  } //for(ContI..
  if( IFIT != 0) //
   goto cerca;
  if(IN < 5) goto bucle;
  else {printf("no se pudo hallar una raiz"); return;}
cerca:
  for(L =1;L<NXX+1;L++)
```

```
   {
     MT = KJ1 - L +1;
     TEMP1 = XCOF[MT]; XCOF[MT] = COF[L]; COF[L]=TEMP1;
   }
     TEMPN = N; N = NX; NX = TEMPN;
     if(IFIT == 0) goto final;
     else goto hallo;
checa:
     if(IFIT == 0) goto bucle;
     XX = XPR;
     YY = YPR;
hallo:
     IFIT = 0;
     if(fabs(YY) - 0.0001*fabs(XX) < 0) goto real;
     Alfa = XX + XX;
     SumaC = XX*XX + YY*YY;
     N -= 2;
     goto divide;
cero:
     XX = 0;   NX--;  NXX--;
real:
     YY = 0;  SumaC = 0;  Alfa = XX;  N--;
divide:
     COF[2] = COF[2] + Alfa*COF[1];
     for(L=2;L<N+1;L++)
             COF[L+1] +=  Alfa * COF[L] - SumaC * COF[L-1];
pone:
    RaizI[N2] = YY;
    RaizR[N2] = XX;
    N2++;
    if(SumaC != 0) {
      YY = -YY;
      SumaC = 0;
      goto pone; }
             } //while(N>0)
       }
```

Además de este módulo el programa cuenta con los siguientes módulos:

ALIN.C Módulo principal que procesa las teclas.
MANDO.C Módulo que procesa los comandos que invoca la tecla "/".

TOKEN.C Módulo que contiene el analizador sintático (tokeniza).
RECIBE.C Módulo que acepta la entrada del usuario. Tiene el editor.
MUESTRA.C Módulo que contiene las rutinas que refrescan la pantalla.
VARIOS.C Módulo que contiene diversas rutinas utilitarias.
ALIN.H Módulo que contiene letreros, definiciones y prototipos.
RKA.C Módulo que tiene el método Runge-Kutta-Fehlberg para integrar, el algoritmo que aplica el método de Ackermann, el que gráfica, procesa el movimiento del ratón y tiene el algoritmo de Fadeev-Leverrier.

APENDICE C FORMAS CANONICAS DE ESTADO ESPACIO

Una representación en el espacio de estado en la forma canónica controlable de ecuaciones diferenciales lineales de orden n en los que la función excitadora incluye términos derivativos

Dada la ecuación diferencial de un sistema lineal:

$$y^{(n)} + a_1 y^{(n-1)} + \ldots + a_n y = b_0 u^{(n)} + b_1 u^{(n-1)} + \ldots + b_n u$$

La matriz de estado **A** viene dada por:

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \bullet & \bullet & \bullet & \ldots & \bullet \\ \bullet & \bullet & \bullet & \ldots & \bullet \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \ldots & -a_1 \end{bmatrix}$$

donde:

$$\begin{aligned} X_1(s) &= Q(s) \\ X_2(s) &= sQ(s) \\ X_3(s) &= s^2 Q(s) \\ &\vdots \\ X_n(s) &= s^{(n-1)} Q(s) \end{aligned}$$

y Q(s), X_1(s), X_2(s), ...X_n(s) se determinan mediante:

$$Q(s) = \frac{U(s)}{s^n + a_1 s^{n-1} + \bullet\bullet + a_{n-1} s + a_n}$$

$$sX_1(s) = X_2(s)$$

$$sX_2(s) = X_3(s)$$

$$\bullet\bullet\bullet$$

$$sX_{n-1}(s) = X_n(s)$$

$$sX_n(s) = -a_1 X_n(s) - a_2 X_{n-1}(s) \bullet\bullet - a_n X_1 + U(s)$$

Al elegir de esta manera a las variables de estado, resulta:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_3$$
$$\vdots$$
$$\dot{x}_{n-1} = x_n$$
$$\dot{x}_n = -a_n x_1 - a_{n-1} x_2 - \bullet\bullet\bullet - a_1 x_n + u$$

El valor de la salida viene dado por:

$$y = \left(b_n - a_n b_0 \vdots \quad b_{n-1} - a_{n-1} b_0 \vdots \quad \ldots \quad \vdots b_1 - a_1 b_0 \right) \begin{bmatrix} x_1 \\ x_2 \\ \bullet \\ \bullet \\ x_n \end{bmatrix} + b_0 u$$

Resumiendo las expresiones (57) a (59), resulta:

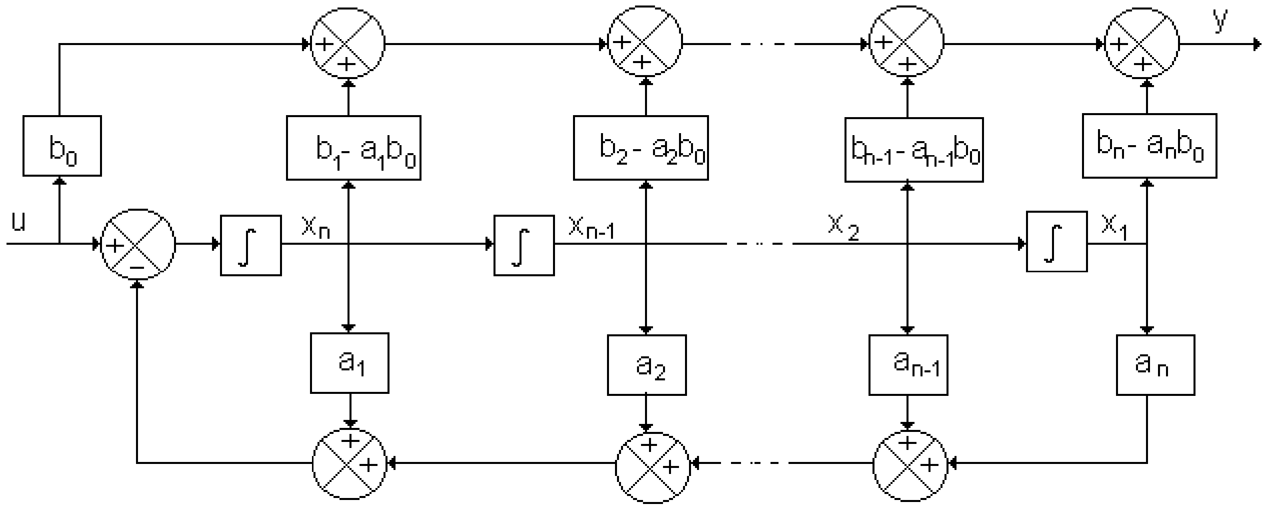
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \bullet & \bullet & \bullet & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \ldots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \bullet \\ \bullet \\ 1 \end{bmatrix} u$$

$$C = \left(b_n - a_n b_0 \vdots \quad b_{n-1} - a_{n-1} b_0 \vdots \quad \bullet \quad \bullet \quad \vdots b_1 - a_1 b_0 \right), \quad D = b_0$$

ALIN permite obtener (61) y (62) a partir de la función de transferencia o bien de los polos y ceros de la misma, mediante CTRL-F4 y Mayús-F4, respectivamente.

La siguiente figura presenta el diagrama de bloques de la representación dada por (61) y (62) (Forma canónica controlable) llamada de programación directa.

La forma canónica controlable es importante al estudiar el método para la ubicación de polos de lazo cerrado mediante la realimentación de estado.



Forma Canónica Controlable

Forma Canónica Observable

Al elegir de esta manera a las variables de estado, resulta:

$$\dot{x}_1 = - a_n x_n + (b_n - a_n b_0)u$$
$$\dot{x}_2 = x_1 - a_{n-1} x_n + (b_{n-1} - a_{n-1} b_0)u$$
$$\vdots$$
$$\dot{x}_{n-1} = x_{n-2} - a_2 x_n + (b_2 - a_2 b_0)u$$
$$\dot{x}_n = x_{n-1} - a_1 x_n + (b_1 - a_1 b_0)u$$

El valor de la salida viene dado por:

$$y = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + b_0 u$$

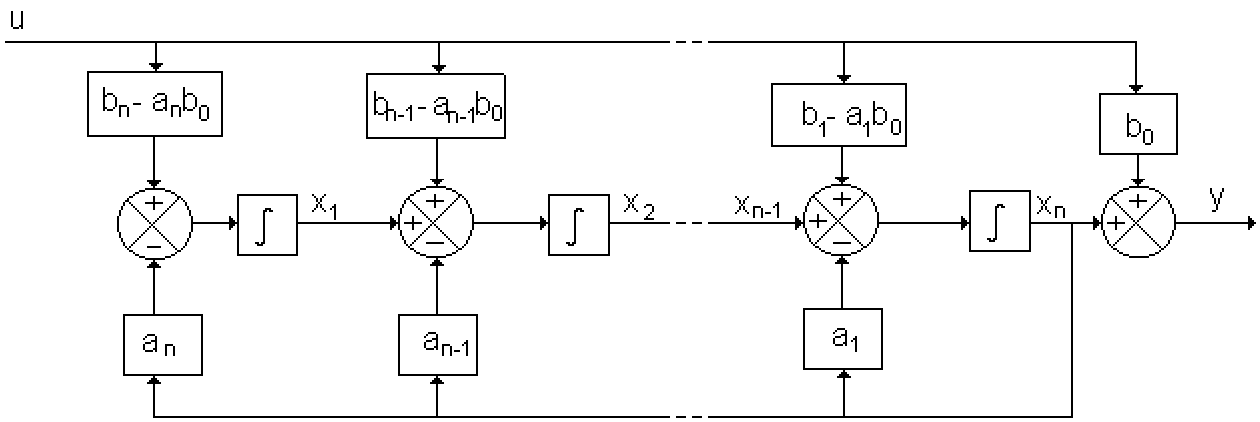
Resumiendo las expresiones (62) a (64), resulta:

$$C = \begin{bmatrix} 0 & 0 & \bullet & \bullet & \bullet & 1 \end{bmatrix}, \quad D = b_0$$

La siguiente figura presenta el diagrama de bloques de la representación dada por (65) y (66) (Forma canónica observable) llamada de programación anidada.

La forma canónica observable es importante al estudiar el método para la ubicación de polos del observador de estado. **ALIN** permite formar esta representación mediante la combinación ALT-F4

.



Forma Canónica Observable

APENDICE D Archivo AYUDALIN.HLP (para ayuda en línea con tecla F1)

```
Información de Soporte de ALIN
====================================
TECLADO

Tecla "/":
    Se utiliza para activar el menú.

Tecla "F1": ó teclas "Alt-Y":
    Se utiliza para ver esta ayuda.

Tecla "F2":
    Para editar la celda iluminada.

Tecla "F3":
    Conmuta modo de exhibir fórmulas

Tecla "F4": ó "Ctrl-F4" ó "Alt-F4"
   Obtiene modelos de estado espacio
   a partir de función de transfer.

Teclas "Mayús-F4":
   Obtiene modelo de estado espacio
   a partir de polos y ceros de fdt

Tecla "F5":
   Hace plano de estado tras cálculo
   Si procede hace gráfica de y2, u2

Tecla "F7":
    Se usa para ver la ayuda gráfica

Tecla "F8":
    Para formatear celda iluminada.

Tecla "F9"
   A partir de la forma general de
   edo-espacio calcula f. de transf.

Teclas "Alt-F9" ó "Ctrl-F9" ó "↑-F9"
   A partir de formas canónicas de
   edo-espacio calcula f de transf.

Teclas "alt-F2"
    Salva sistema, sin confirmación.

Teclas "alt-D"
    Ir a sistema operativo DOS.
```

```
Teclas "CTRL-F10":
    Para terminar ALIN, sin salvar.

Tecla "Flecha arriba":
    - Permite el desplazamiento
    hacia arriba en el área de la
    captura de datos del modelo.

Tecla "Flecha abajo":
    - Permite el desplazamiento
    hacia abajo en el área de la
    captura de datos del modelo.

Tecla "Flecha izquierda":
    - Permite el desplazamiento
    hacia la izquierda en el área de
    captura de datos.

Tecla "Flecha derecha":
    - Permite el desplazamiento
    hacia la derecha en el área de
    captura de datos del modelo.

Tecla "Inicio":
    Coloca el cursor en el origen,
    esto es, para poder editar A11.

Tecla "Fin":
    Coloca el cursor en orden del
    sistema.

Tecla "Re. Pág.":
    Coloca el cursor en CI, X1(0).

Tecla "Av. Pág.":
    Coloca el cursor en ho inicial .
```