ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

PROGRAMA DIGITAL PARA EL ANALISIS DE SISTEMAS NO LINEALES MEDIANTE EL TRAZO DE TRAYECTORIAS DE FASE HASTA EN TRES DIMENSIONES

por

ANGEL GEN PARRA

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO

EN ELECTRONICA Y CONTROL

ENERO DE 1992

CERTIFICO QUE EL PRESENTE TRABAJO HA SIDO REALIZADO EN SU TOTALIDAD POR EL SR. ANGEL GEN PARRA

Ing. Marco Barragán B.

DIRECTOR DE TESIS

Mi más sincero agradecimiento al Ing. Marco Barragán B. por la acertada dirección dada a esta Tesis, por el tiempo dedicado a ella, y por la confianza brindada.

A mi madre, a mi esposa, a todos aquellos
que de una u otra forma ,
han estado a mi lado y
me dieron su apoyo

CONTENIDO

INTRODUCCION

Desde el punto de vista de ingeniería hay muchas aplicaciones donde los sistemas no lineales ofrecen ventajas sobre los sistemas lineales. La deseabilidad de control no lineal para ciertas aplicaciones especificas dan lugar a una discusión comprensible sobre los principales problemas de análisis de este tipo de sistemas. Este análisis puede realizarse mediante la obtención de trayectorias de estado fase , Sin embargo para estos casos los métodos gráficos como el de isóclinas o los analíticos, como el de integración directa pierden precisión y aumentan en complejidad siendo su uso inadecuado. En su lugar se requiere recurrir a las ventajas de la computación para el manejo de datos y graficación tridimensional. Este método es particularmente útil si se trabaja con sistemas no lineales con tres variables de estado y restricciones en el rango de validez de estas ecuaciones. de estado.

Los objetivos del presente trabajo son :

Extender los conceptos relevantes del plano de fase al espacio tridimensional.

Obtener las trayectorias de fase a partir de un diagrama de bloques generalizado para sistemas con parte lineal de hasta tercer orden, parte no lineal característica, compensador y realimentación unitaria.

Obtener las trayectorias de fase a partir de un sistema de ecuaciones no lineales con tres variables de estado, aprovechando la graficación tridimensional, considerando restricciones a cualquier ecuación.

Generalizar el programa digital para los computadores compatibles con IBM.

No se pretende en este trabajo hacer un estudio completo y exhaustivo de la materia, sino una presentación general de la teoría concerniente al método del plano de fase solamente, y la solución de problemas relativos a el .

A continuación se delineará el contenido de la Tesis:

El primer capítulo se dedica al estudio de la representación de sistemas lineales y no lineales , y sus definiciones. Luego , se introducen los conceptos fundamentales para manejar los modelos de diagramas de bloques o funcion de transferencia , y de ecuaciones de estado. A continuación se presentan algunos de los criterios mas usuales para la definición de la calidad de los sistemas de control y los compensadores mas comúnmente usados.

El segundo capítulo se dedica a los conceptos fundamentales de espacio de estado y plano de fase. Se presenta un estudio de las diversas formas de construcción de trayectorias en el plano de fase, acompañado de ejemplos teóricos. Luego, se indican las características relevantes de las trayectorias de fase, las mismas que servirán posteriormente para el análisis de los resultados obtenidos.

El tercer capítulo contiene la descripción del programa. Se obtiene, en este, los modelos matemáticos de la parte lineal , la parte no lineal, el compensador y se desarrollan las ecuaciones a usarse en el método numérico de simulación digital a implementarse en el programa. Se presentan los algoritmos de los módulos que constituyen el programa. y finalmente se describe la parte fundamental del programa digital.

Los capítulos anteriores contienen la teoría necesaria para el manejo de este programa, y es en este capítulo en el que se desarrolla propiamente el objetivo de la Tesis.

El cuarto capítulo contiene ejemplos desarrollados , los mismos que a mas de probar la validez del programa implementado , sirven para ilustrar la utilización del mismo. Aquí se dan algunas conclusiones parciales resultantes del análisis de los ejemplos.

Finalmente se presenta algunas conclusiones que son producto del desarrollo de la tesis.

Se añade los apéndices que contienen el cambio de base para graficación tridimensional, el uso del programa auxiliar de impresión, y el listado del programa implementado.

CAPITULO I

TEORIA DE LOS SISTEMAS DE CONTROL

1.1 SISTEMAS LINEALES Y NO LINEALES

Los sistemas físicos son factibles de ser representados por ecuaciones matemáticas que describen la relación entre las variables que afectan al sistema y las variables del sistema que son afectadas. Los sistemas de control son sistemas físicos que involucran elementos almacenadores o disipadores de energía. Cuando un sistema posee tales elementos, las ecuaciones matemáticas que lo representan son ecuaciones diferenciales.

En los últimos años se ha dado un indudable empuje a la ingeniería de control de sistemas, por el uso de cada vez más poderosos equipos de computación, que permiten hacer análisis de ecuaciones más complejas que aquellas que eran manejadas con la teoria convencional.

La teoría convencional representaba los sistemas físicos aproximándolos a modelos lineales y de orden reducido, para poder aplicar las técnicas convencionales de análisis y diseño. Actualmente, se puede valer de los avances de los lenguajes de programación y de la teoría de control moderna para representar los sistemas físicos por sus modelos no lineales más exactos y realizar un análisis más real.

A continuación se dan las definiciones de estos dos tipos de sistemas: lineales y no lineales.

Los SISTEMAS LINEALES son aquellos en los que las ecuaciones del modelo son lineales. Una ecuación diferencial es lineal si todos los coeficientes son constantes o solo función de t.

La figura 1.1 muestra un sistema lineal en el que se considera la masa m y la constante k del resorte independientes de la fuerza x aplicada y del desplazamiento y.

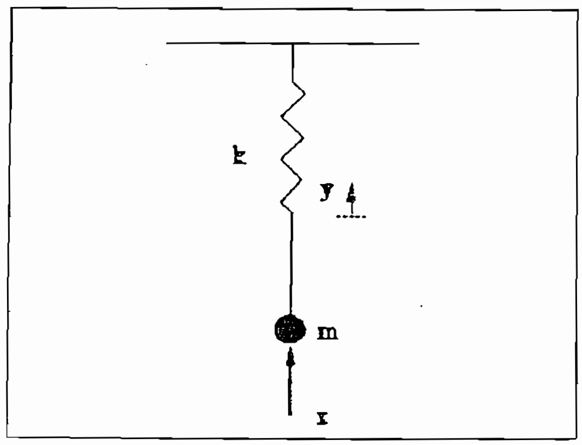


Figura 1.1. Ejemplo de Sistema Lineal

$$m\,y'' + k\,y = x \qquad (1.1)$$

Se observa que la ecuación diferencial que define al sistema tiene coeficientes constantes; por tanto, es una ecuación diferencial lineal.

El orden de la ecuación diferencial está determinado por el máximo orden de la derivada.

En forma general un sistema lineal de una entrada y una salida puede ser representado por una ecuación diferencial lineal de la forma:

$$a_0y^{(n)} + a_1y^{(n-1)} + \ldots + a_{n-1}y^{(1)} + a_ny = b_0x^{(m)} + b_1x^{(m-1)} + \ldots + b_{m-1}x^{(1)} + b_mx \qquad (1.2)$$

donde y es la salida, x es la entrada , $y^{(n)}$ representa la derivada n-ésima de la salida , $x^{(m)}$ representa la derivada m-ésima de la entrada y ai, bi son coeficientes constantes.

Como ya se anotó, un sistema físico de cualquier naturaleza : mecánico, eléctrico, etc. puede ser representado por un sistema lineal, despreciando las no linealidades inherentes y limitando la validez de ese modelo .

Los sistemas lineales, sin embargo de estar acompañados por una extensa teoría de análisis, sólo son aproximaciones válidas para una pequeña cantidad de sistemas físicos.

Para obtener resultados más correctos en el análisis de sistemas, se deben considerar sus no linealidades, esto es, se debe definir otro tipo de sistemas, los sistemas no lineales.

Los SISTEMAS NO LINEALES son aquellos representados por ecuaciones diferenciales no lineales. Una ecuación no lineal involucra funciones trigonométricas, logarítmicas, exponenciales o potenciales. Una ecuación diferencial recibe el nombre de no lineal si no es lineal.

El sistema de la figura 1.1 es ahora reproducido considerando la no linealidad inherente del resorte. En la figura 1.2 la constante k del resorte depende del desplazamiento y.

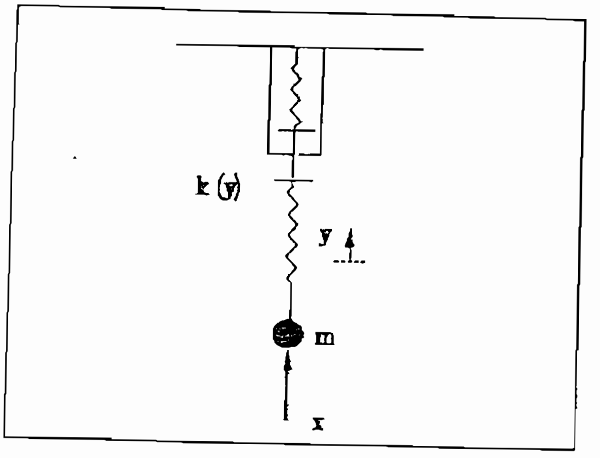


Figura 1.2. Ejemplo de Sistema No Lineal

$$m\, y'' + k(y)\, y = x \qquad (1.3)$$

La ecuación ahora debe ser expresada con sus coeficientes ya no constantes, sino como una función no lineal del desplazamiento, por tanto; es una ecuación no lineal, y el sistema es no lineal también.

En forma general, un sistema no lineal de una sola entrada y una sola salida puede ser representado por una ecuación diferencial no lineal de la forma:

$$\begin{aligned} a_0 y^{(n)} + a_1 y^{(n-1)} + \ldots + a_{n-1} y^{(1)} + a_n y = \\ b_0 x^{(m)} + b_1 x^{(m-1)} + \ldots + b_{m-1} x^{(1)} + b_m x + \\ f(y^{(n)}, y^{(n-1)}, \ldots, y^{(1)}, y, x^{(m)}, x^{(m-1)}, \ldots, x^{(1)}, x) \end{aligned} \qquad (1.4)$$

donde f es una función no lineal de la entrada, salida o alguna derivada de estas.

Es importante notar que el conjunto de los sistemas no lineales contiene al conjunto de sistemas lineales, puesto que, si en la ecuación que representa en general a los sistemas no lineales, la función f no lineal es cero, la ecuación se reduce a la ecuación que anteriormente se indicó que presentaba a los sistemas lineales en general.

Los sistemas a los que se harán mención aquí, lineales o no lineales son invariables en el tiempo por ello los coeficientes a_i, b_i son constantes independientes del tiempo.

1.2 MODELOS

Cualquiera de los sistemas mencionados, lineales o no lineales, representados por las ecuaciones diferenciales generales deben ser pasados a la presentación en Función de Transferencia o en Variables de Estado antes de aplicar cualquiera de las técnicas de análisis.

Si bien el concepto de la Función de Transferencia se aplica solamente a los sistemas lineales invariables en el tiempo, se puede extender a ciertos sistemas no lineales.

1.2.1 FUNCION DE TRANSFERENCIA

La Función de Trasferencia se define como la relación de la transformada de Laplace de la salida (función respuesta) a la transformada de Laplace de la entrada (función excitadora) bajo la suposición de que todas las condiciones iniciales son cero.

Sea el sistema lineal definido por la ecuación (1.1), se obtiene la Función de Transferencia, G(s), tomando la transformada de Laplace de ambos miembros de la ecuación bajo la suposición de que todas las condiciones iniciales son cero. Entonces,

$$G(s) = \frac{Y(s)}{X(s)} = \frac{1}{ms^2 + k} \qquad (1.5)$$

Si se procede de igual forma con la ecuación diferencial general para sistemas lineales, la ecuación (1.2), se obtiene la función general de transferencia para sistemas lineales de orden n-ésimo.

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n} \qquad (1.6)$$

Para sistemas no lineales el equivalente es la Función Descriptiva.

1.2.1.1 FUNCION DESCRIPTIVA

La Función Descriptiva, de un elemento no lineal o un sistema no lineal, se define como la relación compleja entre la componente armónica fundamental de la salida respecto de la entrada, cuando se aplica a la entrada una señal sinusoidal pura.

Considerando por ejemplo el elemento de ganancia no lineal descrito por la característica de transferencia de la figura 1.3, con la constante k del elemento como función de la variable x.

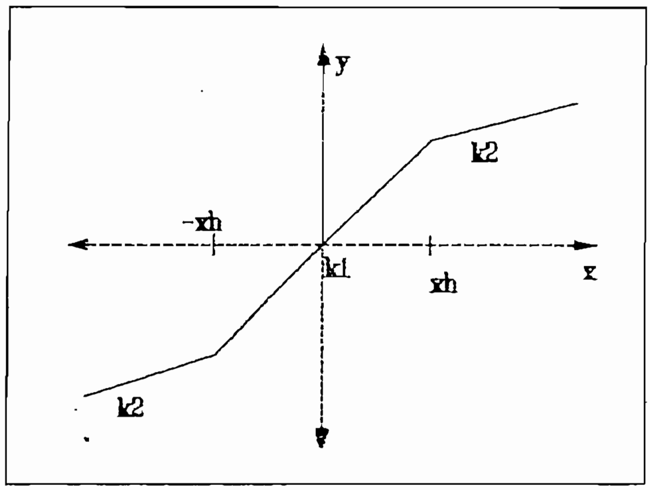


Figura 1.3 Característica de transferencia de un elemento con ganancia no lineal

La salida se relaciona con la entrada por la ecuación:

$$y = K(x)*x + c = \begin{cases} K_1*x + c1 & |x| < x_h \\ K_2*x + c2 & |x| > x_h \end{cases} \qquad (1.7)$$

La función descriptiva se obtiene hallando la componente fundamental de la salida , para la entrada sinusoidal $x(t) = X\ \mathrm{sen}(wt)$ al elemento no lineal.

Así, se puede expresar la salida y(t), cuando es periódica, por una serie de Fourier como sigue:

$$y(t) = A_o + (A_n \cos(nwt) + B_n \operatorname{sen}(nwt)) \qquad (1.8)$$

Si la alinealidad es simétrica, la componente fundamental de la salida es:

$$Y_1 = A_1 \cos(wt) + B_1 \operatorname{sen}(wt) \qquad (1.9)$$

siendo:

$$A_1 = \frac{1}{\pi} \int_0^{2\pi} y(t) \cos(wt)\, dwt$$

$$B1 = \frac{1}{\pi} \int_0^{2\pi} y(t) \operatorname{sen}(wt)\, dwt \qquad (1.10)$$

así, la Función Descriptiva, N, del elemento será :

$$N = \frac{Y_1}{X_1} \angle B_1 = \frac{A_1^2 + B_1^2}{X_1} \angle \tan^{-1}(A_1/B_1) \qquad (1.12)$$

Se ve que cuando O_1 es nulo N es real. Para el ejemplo usado del elemento de ganancia no lineal, se tiene:

$$N = K2 + \frac{2(K_1-K_2)}{\pi}(\operatorname{sen}^{-1}(H/X) + (H/X)\sqrt{1 - (H/X)^2}) \qquad (1.13)$$

Obtenidas las funciones de transferencia de los elementos lineales y las funciones descriptivas de los no lineales se puede representar a los sistemas en general mediante diagramas de bloques.

1.2.1.2 DIAGRAMAS DE BLOQUES

Un diagrama de bloques de un sistema, es la representación gráfica de las funciones realizadas por cada componente y del flujo de las señales del sistema. Está formado por bloques funcionales dentro de los cuales se anota las funciones de transferencia de las partes lineales del sistema o la función descriptiva de los elementos no lineales del sistema.

EJEMPLO 1.1.

Considérese el sistema eléctrico de la figura 1.4a, para obtener su representación en diagrama de bloques primeramente se obtienen las funciones de transferencias de los elementos que lo forman, en este caso la resistencia y el condensador, y se representan en bloques funcionales como se muestra en la figura 1.4.b. Finalmente se unen en un solo diagrama como se indica en la figura 1.4.c :

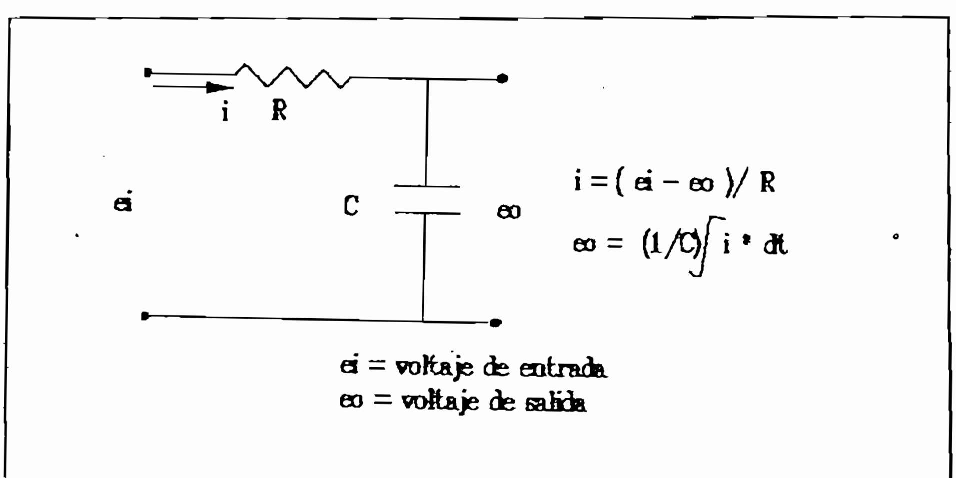


Figura 1.4.a Sistema eléctrico del ejemplo 1.1

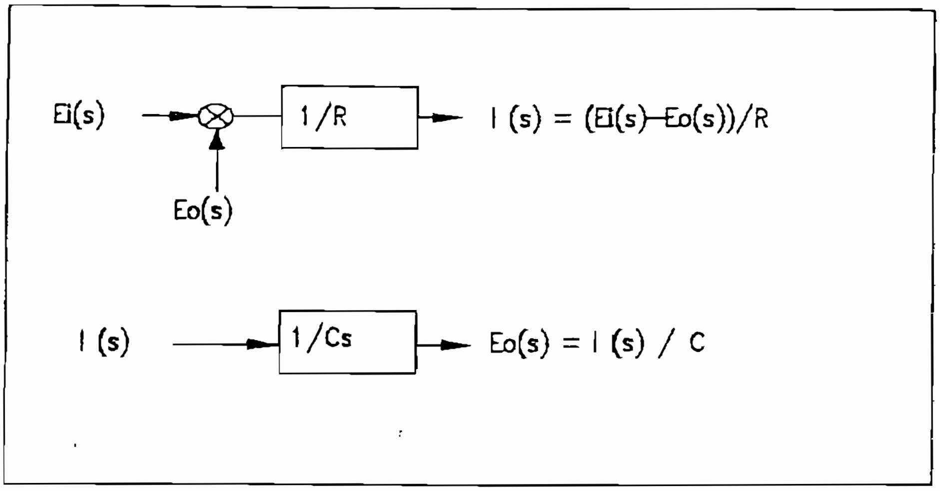


Figura 1.4.b Obtención de los bloques funcionales del ejemplo 1.1

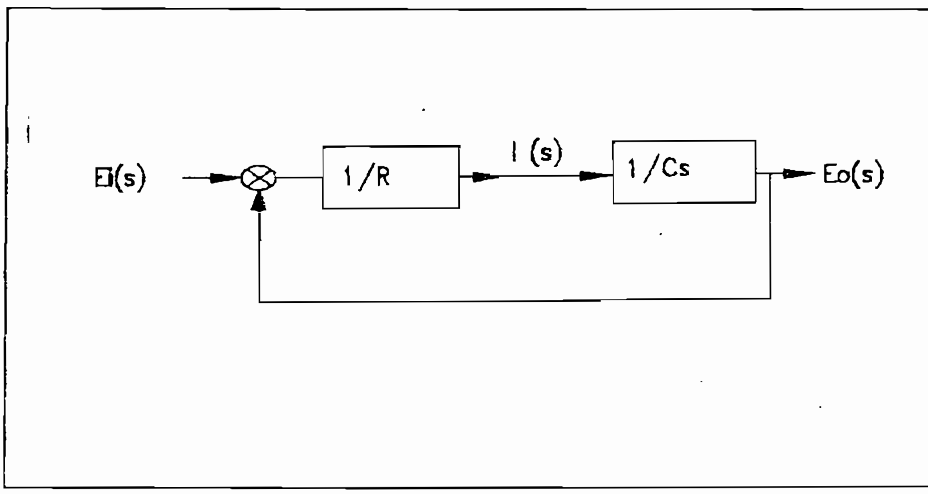


Figura 1.4.c Diagrama de bloques del ejemplo 1.1

Una vez obtenido el diagrama de bloques total de un sistema de control, se puede hallar la ecuación característica o la función de transferencia de lazo cerrado y aplicar las técnicas de análisis convencionales: lugar de las raíces, diagrama de Nyquist,etc., si el sistema es lineal o aproximarlo por tramos previamente si el sistema es no lineal.

Para el método desarrollado en este trabajo, se requiere llegar al diagrama de bloques simplificado general de lazo cerrado, que se muestra en la figura 1.5.

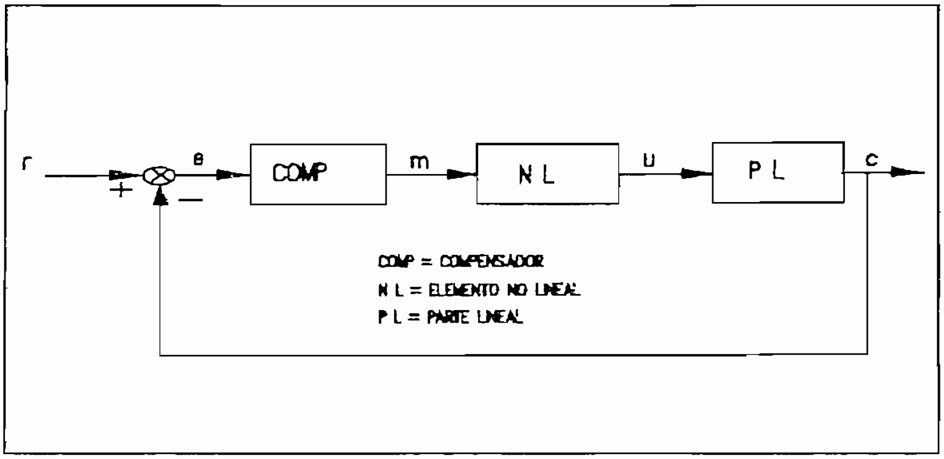
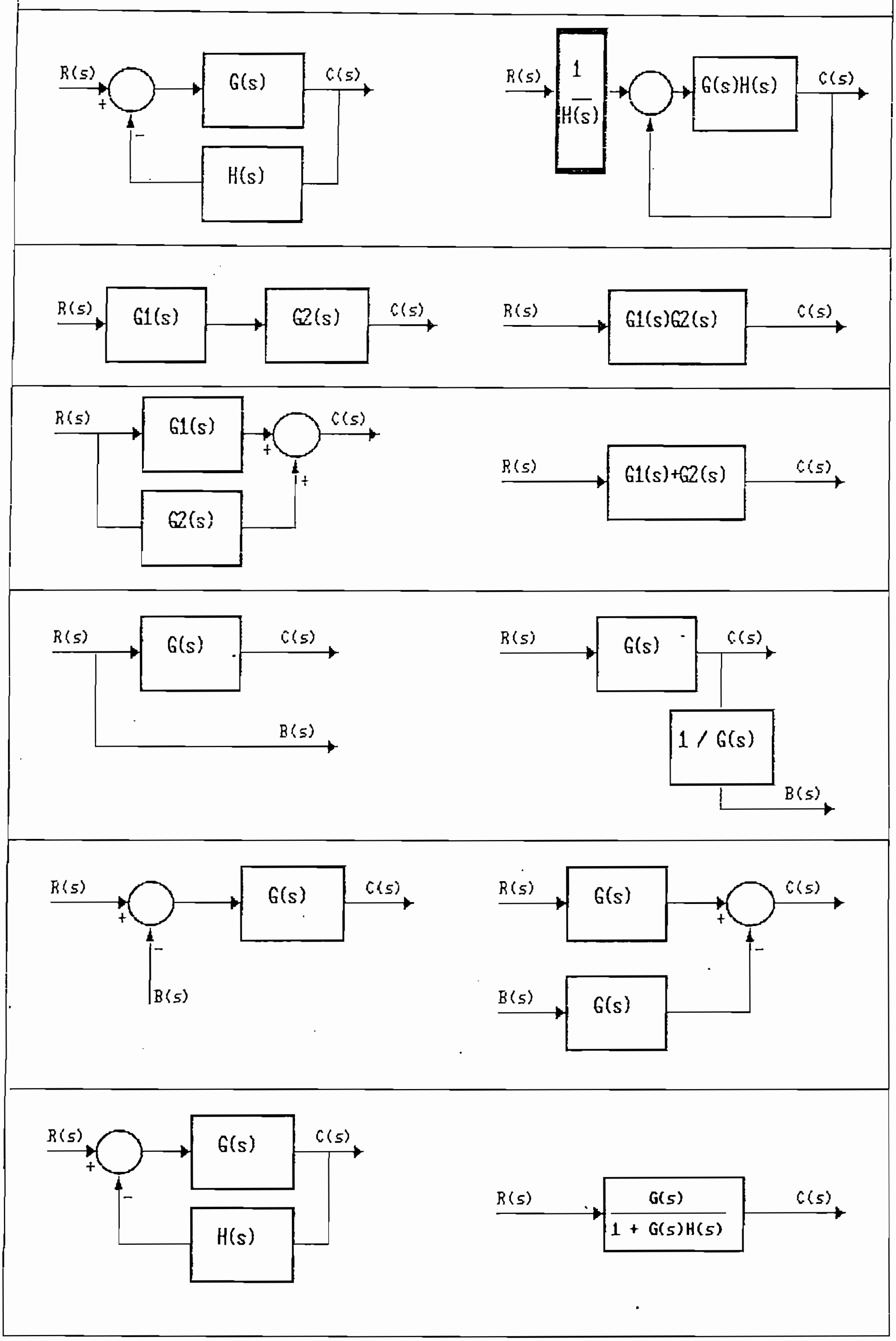


Figura 1.5 Diagrama de bloques general simplificado de un sistema de control no lineal con compensación.

En el diagrama de bloques de la figura 1.5 tenemos un bloque de compensación (COMP), una parte no lineal (NL) y una parte lineal (PL), y un detector de error. La entrada o punto de referencia es r, la salida del comparador es el error e, la salida del compensador es m, la salida del elemento o parte no lineal es u y la variable de salida del sistema de control es c.

La simplificación de un diagrama de bloques se puede realizar utilizando la tabla 1.1 de simplificaciones.

TABLA 1.1 SIMPLIFICACIONES



1.2.2 VARIABLES DE ESTADO

Otra forma de representar cualquier sistema, lineal o no lineal es expresarlo por las variables de estado.

Las ecuaciones de estado son las ecuaciones diferenciales que representan el sistema y que tienen como variables a las variables de estado del sistema.

Las variables de estado del sistema dinámico son el conjunto más pequeño de variables que determinan el estado del sistema. Si el sistema es de enésimo orden se necesitan al menos n variables $x1(t)$, ... , $xn(t)$ para describir totalmente el comportamiento del sistema. De esta manera, una vez dada la entrada para $t > to$ y si el estado inicial en $t = to$ está especificado, entonces el estado futuro del sistema queda totalmente determinado. Estas n variables constituyen un conjunto de variables de estado. Las variables de estado no han de ser necesariamente magnitudes físicamente medibles u observables.

Matemáticamente, puede considerarse a las n variables de estado como componentes de un vector $\underline{x}(t)$. Este vector se denomina vector de estado. Utilizando notación vectorial-matricial se puede expresar una ecuación diferencial de enésimo orden por una ecuación diferencial vectorial-matricial de primer orden, que se denomina ecuación de estado.

Como un ejemplo de obtención del modelo por variables de estado, sea la ecuación diferencial (1.2) que define un sistema lineal . Si la función excitadora no involucra términos derivativos se tendrá:

$$y^{(n)} + a_1 y^{(n-1)} + \ldots + a_{n-1} y' + a_n y = u \qquad (1.14)$$

Notando que el conocimiento de $y(0)$, ..., $y^{(n-1)}(0)$ juntamente con la entrada $u(t)$ para $t >= 0$ determinan totalmente el comportamiento futuro del sistema, se puede tomar a $y(t)$, ..., $y^{(n-1)}(t)$ como un conjunto de n variables de estado.

Se define:

$$\begin{aligned} x_1 &= y \\ x_2 &= y' \\ \ldots &= \ldots \\ x_n &= y^{(n-1)} + \ldots + y \end{aligned} \qquad (1.15)$$

Entonces se puede escribir la ecuación diferencial de enésimo orden por el siguiente conjunto de ecuaciones de primer orden.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\ldots\ldots\ldots \\ \dot{x}_n &= -a_n x_1 - \ldots - a_1 x_n + u \end{aligned} \qquad (1.16)$$

O como la siguiente ecuación vectorial - matricial de primer orden:

$$\underline{\dot{X}} = \underline{A}\ \underline{X} + \underline{B}\ u \qquad (1.17)$$

Obtenida la ecuación vectorial matricial de estado se puede aplicar los conceptos de control moderno con sus ventajas sobre las técnicas convencionales, tales como: optimización, etc.

1.3 CALIDAD DE LOS SISTEMAS LINEALES

A continuación se indicarán los criterios de medición de la calidad de los sistemas de control, y, luego, cómo influyen en esta medición las características del sistema de control.
Es importante notar que no existen respuestas absolutas a este problema. Puesto que, lo que se considera bueno para un proceso puede no ser satisfactorio para otro. Por ello, se consideran algunos de los criterios generales que pueden ser usados para la medición de la calidad de los sistemas de control.

1.3.1 DEFINICION DE LA CALIDAD

Considérese por un momento un sistema de control de un proceso. La razón por la que se requiere un sistema de control es porque la variable bajo control es una variable dinámica, que está cambiando por alguna influencia, y por ello pueden ser requeridas operaciones reguladoras.

No es posible para el sistema de control regular esta variable al punto de referencia exactamente. Y para que el sistema de control pueda generar una acción correctiva es necesario que , antes, se haya producido un cambio en la variable bajo control.

Entonces, dentro de las consideraciones sobre la calidad de los sistemas de control, debe aceptarse de antemano que no existe el "control perfecto", y que pueden ocurrir desviaciones inevitables de las variables respecto de sus valores óptimos.

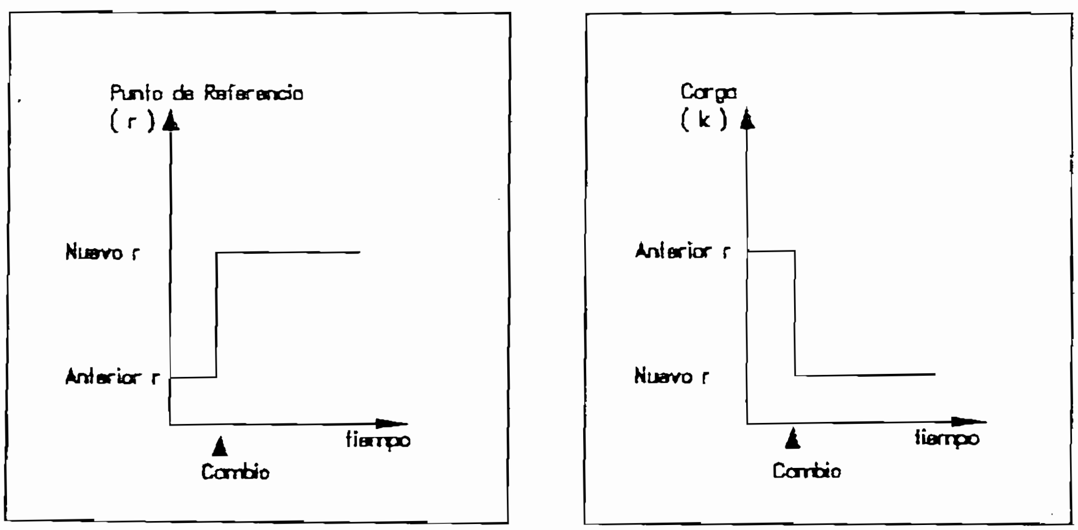
En consecuencia, una definición de la calidad está concebida de acuerdo con estas desviaciones, y su interpretación debe remitirse necesariamente al resultado final del control del proceso.

Se considerará un conjunto de medidas y criterios que permiten que un resultado pueda ser evaluado en términos de las características dinámicas del sistema específico y servir por tanto como medida de su calidad.

Para entender estas medidas, primeramente se define la calidad en términos del sistema de control de proceso.

1.3.1.1 PERTURBACIONES EN EL LAZO DE CONTROL

Puesto que interesa una medida de la calidad del control, se asumirá que existen las condiciones requeridas para la acción de control. Especificamente, se usan dos tipos de perturbaciones, y ambas suponen el cambio como una función paso o escalón, dentro del lazo de control. Un cambio como función paso en el punto de referencia, como se muestra en la figura 1.7a es un cambio instantáneo en el punto de referencia del sistema de control, desde su valor anterior a su nuevo valor. La segunda posible perturbación es una función paso como cambio en la carga del proceso, como es sugerido en la figura 1.7b también ocurre en un tiempo instantáneo. El cambio de la carga puede deberse a un cambio súbito en los parámetros del sistema que constituyen la carga del sistema. A fín de proveer de una medidas de la calidad, se evalúa como responde el sistema a cualquiera de estos tipos cambios súbitos. Para el presente trabajo se usa el cambio en el punto de referencia del sistema de control.



a) Función paso de cambio en el punto de referencia. Cambio intencional

b) Función paso de cambio en la carga. Cambio no intencional

Figura 1.7 Perturbaciones que pueden ocurrir en el lazo de control

1.3.1.2 ESTABILIDAD

La característica más importante en la definición de la calidad de los sistemas de control, es la referente a que este provea de una regulación estable de la variable dinámica.

Regulación estable significa que la variable dinámica no crece sin límite. En la figura 1.8. se muestran dos tipos de respuesta inestable. En el primer caso una perturbación causa que la variable dinámica crezca sin límite. En el otro caso, la variable dinámica empieza a realizar oscilaciones crecientes donde la amplitud se incrementa sin límite. En ambos casos un daño (tal como la destrucción u otro mal funcionamiento) termina eventualmente con el incremento.

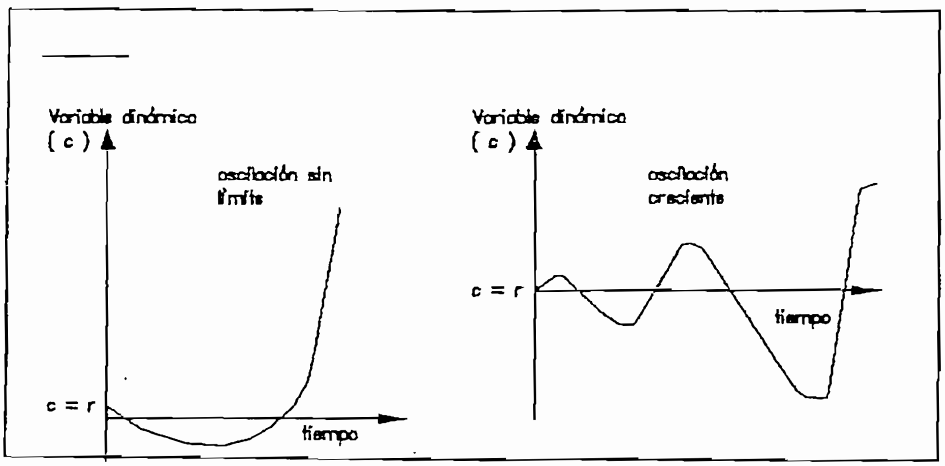


Figura 1.8 Inestabilidad en el sistema de control de proceso referida a un crecimiento no controlado de la variable dinámica.

Para dar una medida de la calidad a través de la estabilidad, se asume que la operación estable ha sido alcanzada. Nótese que una variable dinámica en algunos procesos puede ser estable y tener una oscilación constante. Este es el caso por ejemplo del control de dos posiciones, donde la variable dinámica oscila entre dos límites bajo condiciones normales de operación. Un cambio en los parámetros del sistema puede cambiar la frecuencia de oscilación; pero la amplitud de la oscilación continúa siendo la misma, por lo tanto, la variable está bajo control estable.

1.3.1.3 DESVIACION MINIMA

Si un sistema de control de proceso ha sido ajustado para regular una variable dinámica en un valor del punto de referencia, entonces una definición de la calidad obviamente involucra el límite al cual la variable dinámica se desvia desde el punto de referencia a causa de una perturbación. En el caso donde la perturbación corresponde a un cambio en el punto de referencia, entonces el límite puede considerarse como un sobretiro alto o bajo de la variable dinámica respecto del punto de referencia en su camino al alcance del nuevo punto de referencia. En general, se requiere minimizar cualquier desviación de la variable dinámica del valor de referencia.

1.3.1.4 DURACION MINIMA

Si ocurre una perturbación, puede concluirse que ocurrirá una desviación. Otra definición de calidad es el lapso de tiempo antes de que la variable dinámica regrese o adopte el valor del punto de referencia o al menos caiga dentro del límite aceptable de ese valor.

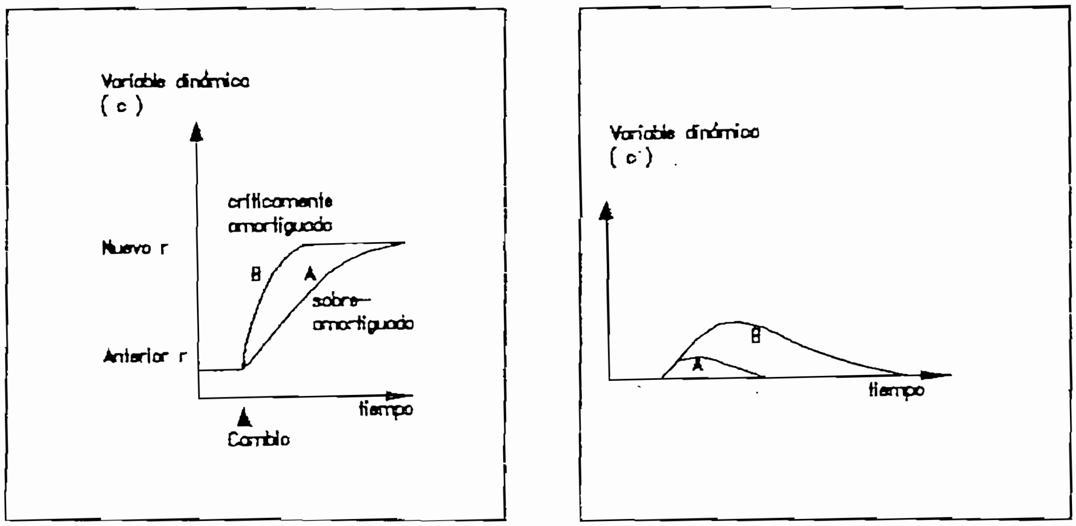
Entonces, por las definiciones anotadas, la calidad de un sistema de control de proceso está definida por la evaluación de la ESTABILIDAD, LA DESVIACION MINIMA, y LA DURACION MINIMA siguiendo a una perturbación, de la variable dinámica

1.3.2 MEDIDA DE LA CALIDAD

En general, no es suficiente decir sencillamente que se diseñará o ajustará el sistema de control para proveer de una operación estable, desviación mínima y duración mínima. Por ejemplo, el logro de la desviación mínima puede resultar en un alejamiento de la duración mínima, como usualmente sucede. También el resultado final de la variable dinámica puede favorecer un ajuste diferente del ajuste mínimo absoluto para proveer una operación más rápida, dentro de una aceptable degradación de las especificaciones del resultado. Para acomodarse a estas circunstancias, se distinguen a continuación algunas medidas de calidad por medio de las cuales se transmite el grado de aproximación lograda a los ideales.

Asumiendo que la operación estable ha sido alcanzada, hay tres respuestas posibles a una perturbación, que una variable dinámica puede presentar en un sistema de control de proceso.

La respuesta específica depende de la ganancia y los parámetros específicos del sistema y el proceso.



a) Reacción no oscilatoria a un cambio en el punto de referencia.

b) Reacción no oscilatoria a un cambio en la carga

Figura 1.9 Reacción del sistema de control, ajustada para no presentar oscilaciones.

Refiriendose a la figura 1.9a para un cambio en la carga y a la figura 1.9b para un cambio en el punto de referencia, se tienen las siguientes definiciones.

1.3.2.1 SISTEMA SOBREAMORTIGUADO

El sistema está sobreamortiguado en el caso A de la figura 1.9 La desviación alcanza el valor del punto de referencia suavemente (siguiendo un perturbación) sin oscilaciones. La duración no es mínima en este caso. Y la desviación misma usualmente no es mínima tampoco. Tal respuesta es segura, sin embargo, asegurando que no ocurre inestabilidades y ciertamente nunca ocurren desviaciones máximas.

1.3.2.2 SISTEMA CRITICAMENTE AMORTIGUADO

Un ajuste cuidadoso del sistema lleva a la curva B de la figura 1.9 . En este caso, la duración es mínima, de todos modos, la desviación puede ser larga. Esta es la respuesta óptima para una condición donde no es deseable sobretiros en un cambio del punto de referencia, o en general no se deseen oscilaciones.

1.3.3.3 SISTEMA SUBAMORTIGUADO

El resultado natural de ajustes adicionales en el sistema de control es una respuesta oscilatoria, donde la desviación realiza un número de oscilaciones alrededor del punto de referencia. Es posible que esta respuesta entregue desviación mínima y duración mínima en algunos casos. Tal respuesta es preferida cuando las oscilaciones pueden ser toleradas en el proceso.

Dos medidas especializadas de la calidad de los sistemas de control son usadas cuando ninguna de las condiciones anteriores son válidas para definir la medida de la calidad de los sistemas de control en el proceso. Estas medidas se indican a continuación

1.3.2.4 CUARTO DE AMPLITUD

Cuando un sistema de control tiene unas respuesta oscilatoria amortiguada para una perturbación, a veces se usa un criterio que no es, ni de desviación mínima ni de duración mínima. Esta medida de la calidad es hallada mediante ajuste del sistema de control hasta que la desviación debida a una perturbación sea tal que el pico de desviación es reducido a un cuarto de la amplitud de la desviación anterior, como se muestra en la figura 1.10

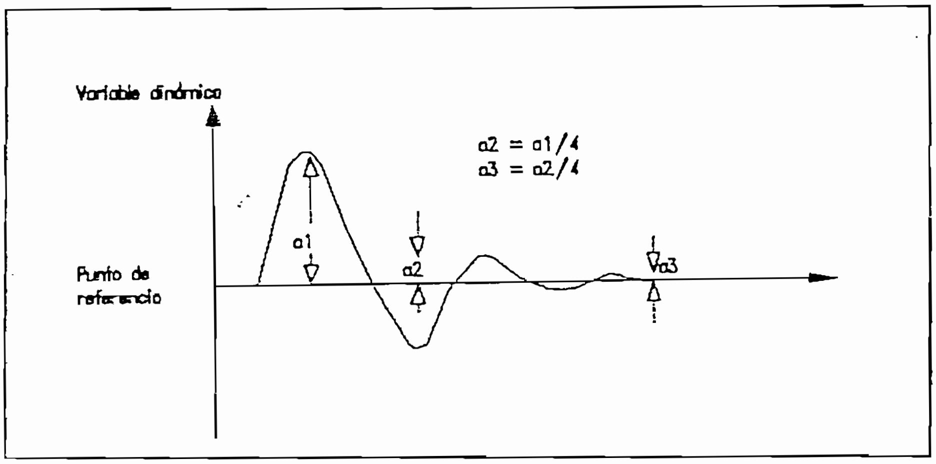


Figura 1.10 Respuesta oscilatoria de un sistema ajustado para tener en cada oscilación un cuarto de la amplitud de la oscilación anterior.

En este caso la magnitud actual de la desviación no esta incluída en la medida de la calidad ni tampoco lo está el tiempo entre cada pico de desviación. En este sentido, ni la magnitud ni la duración de la desviación están directamente incluidas en el criterio del cuarto de amplitud.

1.3.2.5 AREA MINIMA

En el caso de una respuesta oscilatoria o subamortiguada, lo más critico es a veces la combinación de la duración mínima y la desviación mínima, la cual debe ser minimizada. Por tanto, si ocurre una desviación mínima en un ajuste del sistema y una duración mínima en otro ajuste, entonces ninguno es óptimo. Un tipo de medida óptima en estos casos es minimizar el área neta de la desviación como función del tiempo. En la figura 1.11

esta área se muestra como la suma de las medias áreas bajo la curva.

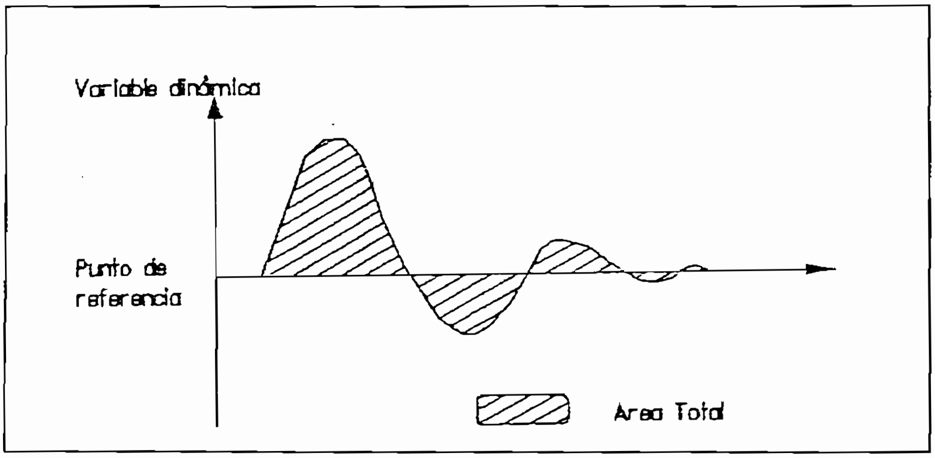


Figura 1.11 Area total de la respuesta oscilatoria a ajustarse al mínimo mediante la aplicación del criterio de la mínima area en el ajuste del sistema de control.

Analíticamente puede ser expresada como

$$A = \int (C - R) dt \tag{1.18}$$

donde

A = área de la desviación o error

C = valor medido

R = valor del punto de referencia

Otra forma de representar esto es usar una escala de porcentaje total, así.

$$Ap = \int Ep \cdot dt \tag{1.19}$$

donde

Ep = escala de error total en porcentaje

Ap = área como porcentaje-tiempo

Mediante la adopción de este criterio de medida de la calidad del sistema de control de proceso se toma los límites de la duración y de la desviación y se lleva su producto al mínimo.

1.4 CALIDAD DE LOS SISTEMAS NO LINEALES

Como se indicó anteriormente, el conjunto de los sistemas lineales está contenido en un conjunto mayor que es el de los sistemas no lineales. Por tanto, los criterios aplicados a los sistemas lineales son aplicables también a los sistemas no lineales, sin embargo debe anotarse que la estabilidad alcanzada en un punto de operación seleccionado no implica que el sistema no lineal sea estable si este punto de operación es cambiado. Esta característica propia de los sistemas no lineales, de que la estabilidad local de un punto no implique estabilidad global para todos los puntos hace preferir la medición de la calidad de los sistemas de control a través de las definiciones dadas para los sistemas lineales , pero obtenidas a partir de las trayectorias en el plano de fase o en el espacio tridimensional de estado.

Por lo anteriormente dicho, en el presente trabajo se utilizará las trayectorias de fase para aplicar los criterios de medición de calidad descritos anteriormente, tanto sobre los sistemas no lineales como los sistemas lineales. La figura 1.12 muestra una trayectoria de fase y las definiciones necesarias para la calificación de los sistemas de control.

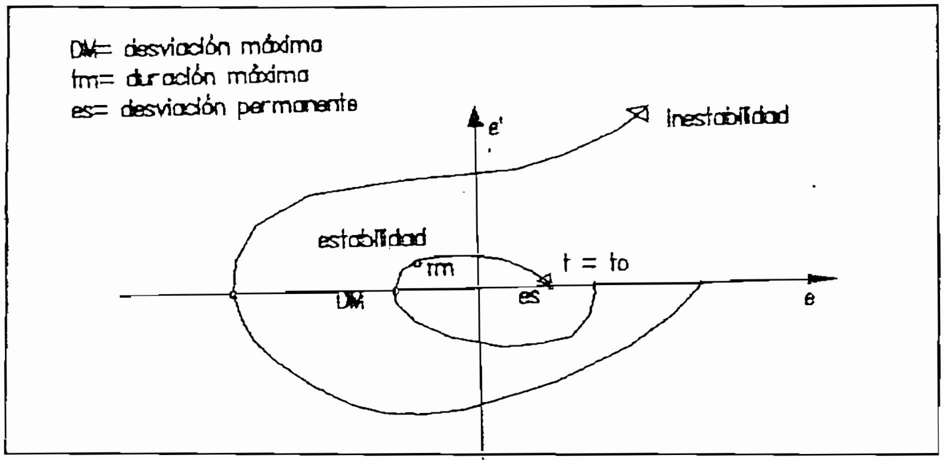


Figura 1.12. Trayectoria de fase con las definiciones para calificación de los sistemas de control.

En el siguiente capítulo se indican los métodos para la obtención de estas trayectorias, y las características relevantes de las mismas para los sistemas de hasta tercer orden.

1.5 COMPENSACIONES

1.5.1 PARA SISTEMAS LINEALES

En un sistema de control de lazo cerrado, como el indicado en la figura 1.13 , se compara el valor efectivo de la salida de la planta y se compara con el valor de la entrada que es el punto de referencia para el sistema de control. Se determina la desviación y se produce una señal de control que trata de reducir la desviación a cero o al menos a un valor menor al inicial. El bloque funcional, en el diagrama de bloques del sistema, que corrige la desviación se conoce como compensador, compensación o acción de control.

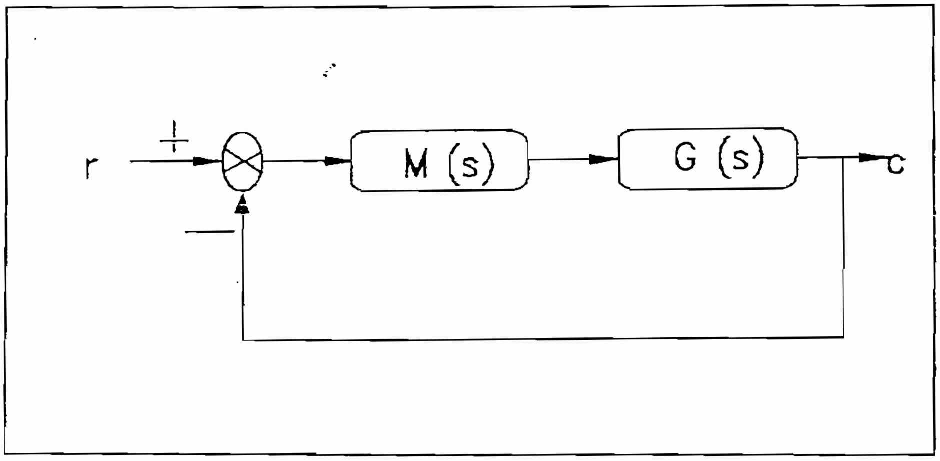


Figura 1.13 Diagrama de bloques de un sistema de control de lazo cerrado con un bloque funcional de compensación.

Las formas de compensación más conocidas y usadas son : proporcional, proporcional derivativa, proporcional integral y proporcional integral derivativa.

PROPORCIONAL

La relación entre la salida del controlador m(t) y la señal de error actuante e(t) es :

$$m(t) = Kp\ e(t) \tag{1.20}$$

o en el dominio de Laplace

$$M(s)/E(s) = Kp \tag{1.21}$$

donde Kp se denomina sensibilidad proporcional o ganancia. Esencialmente consiste de un amplificador con ganancia variable.

PROPORCIONAL DERIVATIVO.

Esta definido por la ecuación :

$$m(t) = Kp\ e(t) + Kd\ de(t)/dt \qquad (1.22)$$

o la función de transferencia

$$M(s)/E(s) = Kp\ (1 + Td\ s) \qquad (1.23)$$

siendo Kp la sensibilidad proporcional, Td el tiempo de derivativo. Se conoce tambien como control de velocidad, el valor de la salida es proporcional a la velocidad de variación de la señal de error actuante. Nunca se tiene la acción de control derivativa sola porque ésta actúa sólo mientras existe variación del error , sino que se la acompaña de un compensador proporcional para corregir el error en la respuesta estacionaria.

PROPORCIONAL INTEGRAL.

Queda definida por la ecuación :

$$m(t) = Kp\ e(t) + Kp/Ti \int e(t)\ dt \qquad (1.24)$$

o la función de transferencia

$$M(s)/E(s) = Kp(1 + 1/Tis) \quad (1.25)$$

donde Kp es la sensibilidad proporcional y Ti es el tiempo de integración.

La acción integral produce un polo en el origen a la función de transferencia global del sistema. Esto aumenta en uno el grado del sistema. Así, los sistemas de tercer orden se convertirian en sistemas de cuarto orden, por lo tanto, no se presenta, en este trabajo, como alternativa de compensación para estos casos.

1.5.2 PARA SISTEMAS NO LINEALES

Para los sistemas no lineales se puede usar también las compensaciones indicadas para los sistemas lineales. Se probará a continuación que la característica no lineal no es afectada por la compensación sino que a igual que en los sistemas lineales esta etapa traslada el gráfico de la respuesta inicial del sistema, cualquier sea el método de graficación usado.

Cuando se habló de los métodos gráficos de análisis de sistemas lineales se mencionó el diagrama de Nyquist o diagrama polar. En este diagrama el punto crítico es $-1 + j0$ y se obtiene de la ecuación característica

$$1 + G = 0 \tag{1.26}$$

$$G(w) = -1 + j0 \tag{1.27}$$

Introduciendo una etapa de ganancia no lineal la ecuación característica se transforma en

$$1 + N\,G = 0 \tag{1.28}$$

$$G(w) = -1/N \tag{1.29}$$

Los puntos críticos ahora son los puntos correspondientes al lugar $-1/N$.

Introduciendo además una etapa de compensación la ecuación característica se transforma en

$$1 + G_c\,N\,G = 0 \tag{1.30}$$

$$G_c(w)\,G(w) = -1/N \tag{1.31}$$

$$G_T(w) = -1/N \tag{1.32}$$

La introducción de la etapa de compensación a recorrido el lugar de $G(w)$ hasta el lugar de $G_T(w)$. Pero, el lugar $-1/N$ de puntos críticos sigue siendo el mismo haya o no compensación .

En el capitulo siguiente se indica el análisis de los sistemas de control utilizando las trayectorias de fase. Los sistemas analizados pueden ser lineales o no lineales con los tipos de compensación que aquí se han señalado.

CAPITULO II

ANALISIS EN EL PLANO DE FASE

2.1 METODOS DISPONIBLES

Como se indicó en el capítulo I, cualquier sistema, lineal o no lineal, puede ser representado mediante sus variables de estado (x_1, x_2, ..., x_n). En general, al sistema coordenado cuyos ejes son las variables de estado se lo conoce como ESPACIO DE ESTADO, y al valor que, en un instante cualquier ($t \geq 0$), tienen las variables de estado se lo conoce como ESTADO DEL SISTEMA y se representa como un punto del espacio de estado.

La curva que une los puntos que representan el estado del sistema a través del tiempo se la llama TRAYECTORIA. La trayectoria parte del estado inicial ($t = 0$) del sistema y llega hasta el estado final ($t \rightarrow \infty$).

En el caso del presente trabajo, para un análisis de los sistemas de tercer orden, se requieren tres variables de estado y un sistema tridimensional coordenado para graficación de las trayectorias de fase. Para un análisis de los sistemas de segundo orden se necesitan dos variables de estado y un sistema coordenado planar que se denomina comúnmente PLANO DE FASE .

El análisis de los sistemas de control en el plano de fase, involucra el trazo de trayectorias que representan la evolución del sistema, la dinámica del sistema, esto es, el estado del sistema en diferentes instantes de tiempo, partiendo de diferentes condiciones iniciales . A esto se denomina el RETRATO DE FASE .

Para la obtención del retrato de fase existen algunos métodos, que los describiremos a continuación .

2.1.1 SIMULACION ANALOGA.

Se requiere de un computador análogo, en el cual se implementa la función que describe el sistema, mediante arreglo de los sumadores, integradores, etc. del computador. Se puede realizar operaciones no lineales, como la multiplicación de dos variables. Se dispone en el computador circuitos electrónicos normalizados para simular alinealidades habitualmente halladas en los sistemas de control, tal como zona muerta, saturación, histérisis y fricción.

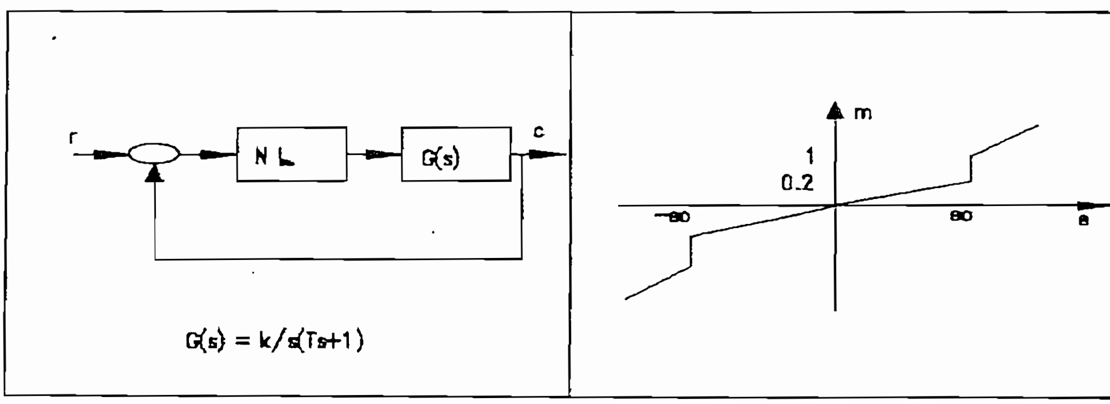
Actualmente se dispone de programas digitales que simulan a un computador análogo, aunque el tiempo de ejecución es mayor frente a este último.

La desventaja de usar el computador análogo para obtener el retrato de fase de un sistema de control, es que este método permite sólo el trazo de una trayectoria de fase por vez, y no es aplicable a sistemas de tercer orden . Se deben usar las técnicas de escalamiento en tiempo y magnitud para asegurar que las señales en los amplificadores permanezcan en un rango que permita la operación en la zona lineal de trabajo.

A continuación se presenta un ejemplo de obtención de la trayectoria de fase de un sistema mediante este método.

Ejemplo 2.1.[1]

El diagrama de la figura 2.1a es la representación en bloques de un sistema de segundo orden con un elemento de ganancia no lineal, cuya característica de transferencia se muestra en la figura 2.1b (este elemento se usa para sistemas sujetos a ruidos de baja amplitud y alta frecuencia).



a) Diagrama de bloques de un sistema de segundo orden con elemento no lineal
b) Característica de transferencia del elemento no lineal

Figura 2.1 Sistema de segundo orden con elemento de ganancia no lineal

La ecuación (2.3) que define el sistema se obtiene del diagrama de bloques y al característica de transferencia del elemento no lineal.

$$m = f(e)*e \qquad f(e) = \begin{cases} 1*e & |e| > e_o \\ K*e & |e| < e_o \end{cases} \tag{2.1}$$

$$c = \frac{K}{s(Ts+1)}*m \tag{2.2}$$

$$\begin{aligned} T*\ddot{c} + \dot{c} &= K*m \\ &= K*f(e)*e \\ &= K*f(e)*(r-c) \end{aligned}$$

$$\ddot{c} = [k*f(e)*(r-c) - \dot{c}]/T \tag{2.3}$$

La parte no lineal de ganancia no esta disponible como módulo, sin embargo se la genera con la ayuda de los otros módulos como se muestra en la figura 2.2 , que presenta el diagrama completo de simulación, con condiciones iniciales e(0)=0, è(0)=0 y entrada escalón.

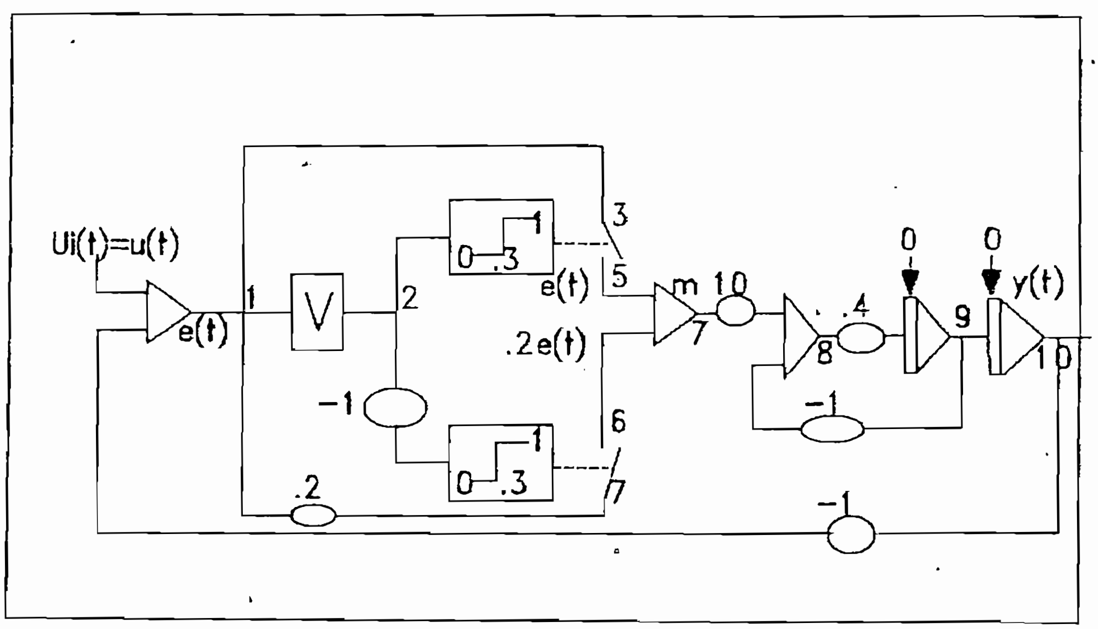


Figura 2.2 Diagrama de simulación del sistema del ejemplo 2.1

Como resultado gráfico se obtiene a través de un plotter la trayectoria de fase mostrada en la figura 2.3.

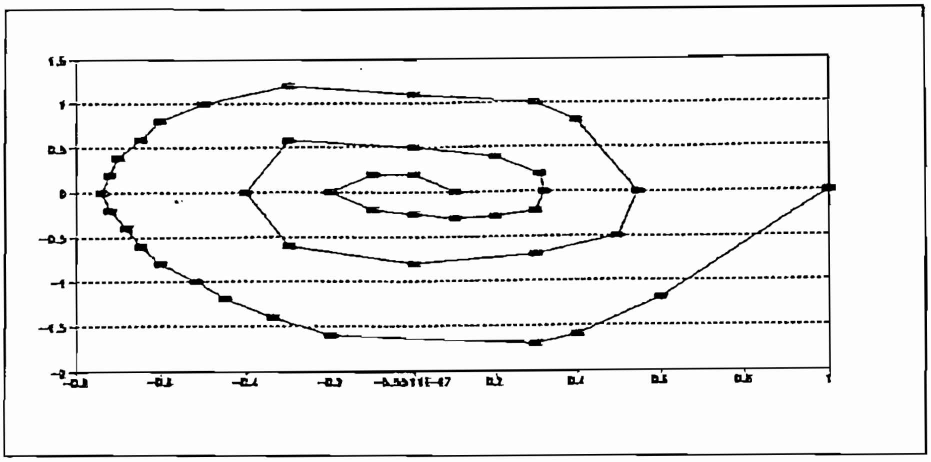


Figura 2.3 Trayectoria de fase obtenida por simulación análoga del sistema de control indicado en el ejemplo 2.1

2.1.2. SIMULACION DIGITAL

Se utiliza un método recursivo en un computador digital Las ecuaciones de estado son del tipo $\dot{x} = f(x,u)$, sin embargo para intervalos pequeños podemos decir que

$$\frac{dx}{dt} = f(x,u) = \frac{\Delta x}{\Delta t} \tag{2.4}$$

Entonces las nuevas ecuaciones de estado serán

$$\Delta x = f(x,u)\,\Delta t \tag{2.5}$$

y el estado del sistema se obtiene en base al estado anterior y el incremento de tiempo escogido.

$$\begin{array}{lll} x_1 = x_o + \Delta x_1 & \qquad & x_1 = f(x_o,u_o) * \Delta t \\ x_2 = x_1 + \Delta x_2 & & x_2 = f(x_1,u_1) * \Delta t \\ .. = ... & & .. = ... \\ x_{i+1} = x_i + \Delta x_{i+1} & & x_{i+1} = f(x_i,u_i) * \Delta t \end{array} \tag{2.6}$$

Este es el método que se utilizará en este trabajo y se indica en el siguiente capítulo.

2.1.3 INTEGRACION DIRECTA.

Se elimina la variable tiempo de las ecuaciones diferenciales que describen sistema, quedando únicamente las variables de fase. Se puede hacer separación de variables para integrar. Así,

$$\frac{dx_1}{dt} = f_1(x_1,x_2) \quad ; \quad \frac{dx_2}{dt} = f_2(x_1,x_2) \tag{2.7}$$

Entonces,

$$\frac{dx_2}{dx_1} = \frac{f_2(x_1,x_2)}{f_1(x_1,x_2)} \tag{2.8}$$

Se puede integrar la ecuación para obtener la ecuación siguiente que describe una trayectoria en el plano de fase.

$$x_2 = \emptyset(x_1) \tag{2.9}$$

Este método sólo es aplicable si la ecuación es integrable, lo que no sucede en general con los casos no lineales. Una vez obtenida la ecuación de la trayectoria de fase se pueden trazar directamente la relación entre x_1 y x_2 en el plano de fase.

Si la ecuación no es integrable, puede obtenerse las soluciones $x_1(t)$, $x_2(t)$ de las ecuaciones diferenciales originales y eliminar de estas soluciones la variable t . Se puede efectuar esto si la dificultad no es excesiva.

Ejemplo 2.2. [2]

Sea la siguiente ecuación (2.10) que define un sistema de segundo orden.

$$\ddot{x} = -M \tag{2.10}$$

donde M es una constante diferente de cero, y las condiciones iniciales son $x(0) = x_o$ y $x(0) = 0$

Se puede escribir la ecuación como

$$\ddot{x} = \frac{d^2x}{dt^2} = \frac{dx}{dx}\,\frac{d(\dot{x})}{dt} = \dot{x}\,\frac{d\dot{x}}{dx} = -M \tag{2.11}$$

Separando variables e integrando se obtiene la ecuación (2.12).

$$\dot{x}^2 = 2(x_o - x)\ M \tag{2.12}$$

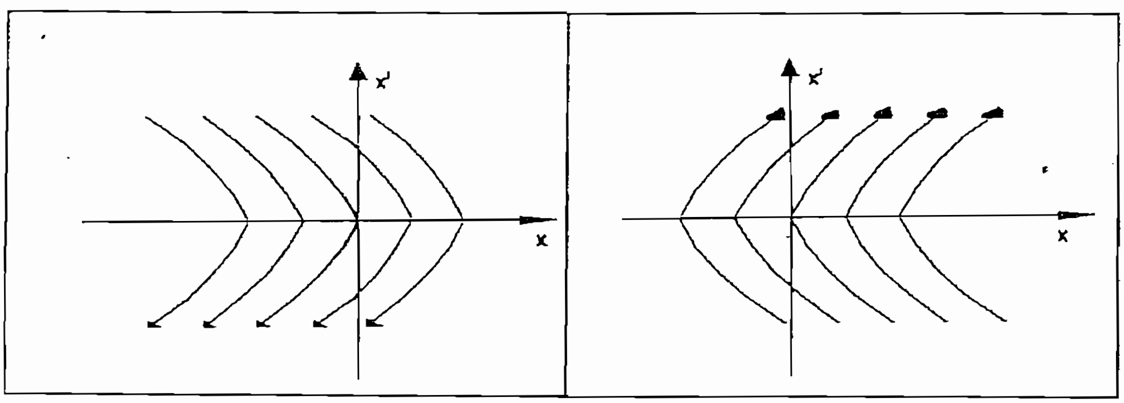
donde la constante de integración se determinó en base a las condiciones iniciales dadas. La ecuación (2.12) representa una trayectoria que pasa por el punto $(x_o,0)$.

Como se indicó anteriormente, también puede obtenerse esta ecuación hallando las soluciones $x = \phi_1(t)$, $x = \phi_2(t)$ y eliminando la variable t, de estas soluciones. Utilizando las condiciones iniciales dadas se encuentra que $\dot{x}(t)$ y $x(t)$ son :

$$\dot{x}(t) = -\,M\,t \tag{2.13}$$

$$x(t) = -\,1/2\;M\,t^2 - x_o \tag{2.14}$$

Eliminando la variable tiempo, t, de las ecuaciones (2.13) y (2.14) se llega a la ecuación (2.12), como se deseaba. En la figura 2.4 se tiene las trayectorias de fase correspondientes al sistema con $M = 1$ y $M = -1$.



a) Trayectoria obtenida para $M = 1$

b) Trayectoria obtenida para $M = -1$

Figura 2.4 Trayectorias de fase del sistema indicado en el ejemplo 2.2

2.1.4 METODO DE LAS ISOCLINAS.

Considerando la representación general de un sistema en variables de estado , para un sistema de segundo orden se tiene:

$$\dot{x}_1 = f_1(x_1,x_2) \tag{2.15}$$

$$\dot{x}_2 = f_2(x_1,x_2) \tag{2.16}$$

la cual puede ser reescrita como una ecuación en el plano de fase

$$\frac{dx_2}{dx_1} = \frac{f_2(x_1,x_2)}{f_1(x_1,x_2)} \tag{2.17}$$

Esta ecuación indica que cualquier trayectoria que pasa por un punto (x_1,x_2) tiene pendiente $f_2(x_1,x_2)/f_1(x_1,x_2)$ en ese punto. Los únicos puntos donde la pendiente es indefinida (f_2 y f_1 son ambas cero) son llamados PUNTOS SINGULARES.

Conociendo la pendiente de las trayectorias en los puntos por donde ellas pasan es posible determinar gráficamente las trayectorias . Si se asume que dx_2/dx_1 es igual a una constante m , entonces la ecuación (2.18)

$$m = \frac{f_2(x_1,x_2)}{f_1(x_1,x_2)} \tag{2.18}$$

define una línea , y toda trayectoria en el punto de cruce sobre esta línea tendrá siempre pendiente m. Tal línea es conocida con el nombre de ISOCLINA.

Mediante el dibujo de algunas isóclinas, usando varios valores de pendiente m y trazp de una serie de líneas paralelas guias sobre cada trayectoria (cada línea tendrá pendiente m), es posible trazar las trayectorias de fase.

Cada trayectoria es trazada de tal forma que en cada intersección con una isóclina la trayectoria tenga la pendiente de la línea guía de la isóclina . Las isóclinas y líneas guías para un sistema lineal se muestra en la siguiente figura 2.5.

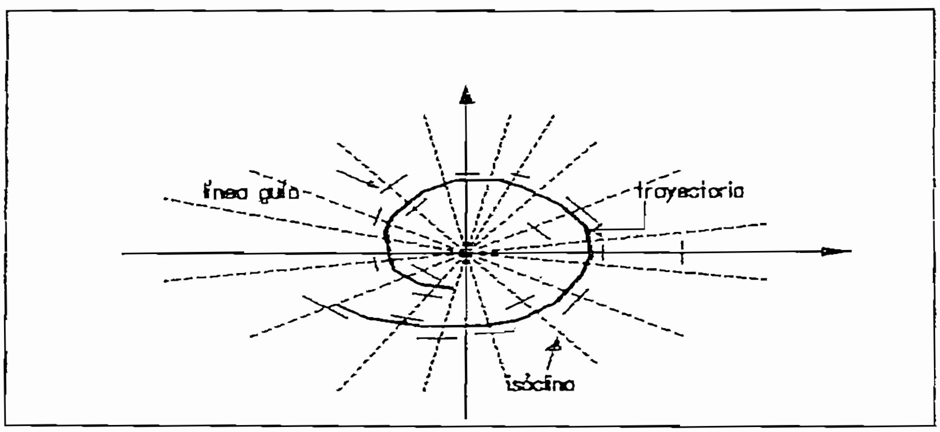


Figura 2.5 Isóclinas, líneas guías y trayectoria de fase para un sistema lineal.

Cabe anotar que las isóclinas no son líneas rectas en la generalidad de los casos, para un sistema de control cualquiera.

A continuación se indica un ejemplo de obtención del retrato de fase mediante este método.

Ejemplo 2.3 :

Considérese la función de transferencia total de un sistema de control lineal de segundo orden.

$$\frac{C(s)}{R(s)} = \frac{K}{As^2 + K} \qquad (2.19)$$

La correspondiente ecuación diferencial es

$$A*c(t) + K*c(t) = K*r() \qquad (2.20)$$

Tomando como variables de estado $x_1 = c(t)$ y $x_2 = c(t)$ el sistema puede ser expresado por ecuaciones diferenciales de primer orden.

$$x_2 = (K*r - K*x_1)/A \qquad (2.21)$$

$$x1 = x2 \qquad (2.22)$$

Dividiendo entre sí estas ecuaciones se llega a la ecuación en el plano de fase.

$$\frac{dx_2}{dx_1} = \frac{K(\ r - x_1\ )}{A*x_2} \qquad (2.23)$$

Si se grafica x_2 en función de x_1 se obtiene el retrato de fase del sistema de control, entonces la ecuación (2.23) es la pendiente de la trayectoria en el plano de fase. Para un valor m constante cualquiera de la pendiente se puede escribir.

$$m = K(\ r - x_1\ )/(A*x_2)$$

o

$$x_2 = (Kr/Am) -(K/Am)x_1 \qquad (2.24)$$

Esta es la ecuación de las isóclinas, que son las líneas que las trayectorias de fase atraviesan con igual pendiente.

Las trayectorias de fase pueden ser obtenidas a partir de las isóclinas. Asumiendo que la entrada r es una función paso unitaria, la ecuación (2.24) es la expresión matemática de líneas rectas que tienen pendiente $-K/Am$.

Todas las trayectorias que intersecten estas líneas tendrán la misma pendiente.

Las trayectorias y las líneas para este ejemplo son mostradas en la figura 2.6. Nótese que en el punto $x_2=c(t)=0$, $x_1=c'(t)=1$, que es solución de la ecuación diferencial (2.20) que define al sistema, se produce una indeterminación en la ecuación (2.17), por tanto, este es un punto singular.

El tipo de punto singular está en relación con la naturaleza y ubicación de las raíces de la ecuación característica del sistema, como se mostrará más adelante. La función de transferencia usada en el ejemplo corresponde a un sistema sin amortiguamiento.

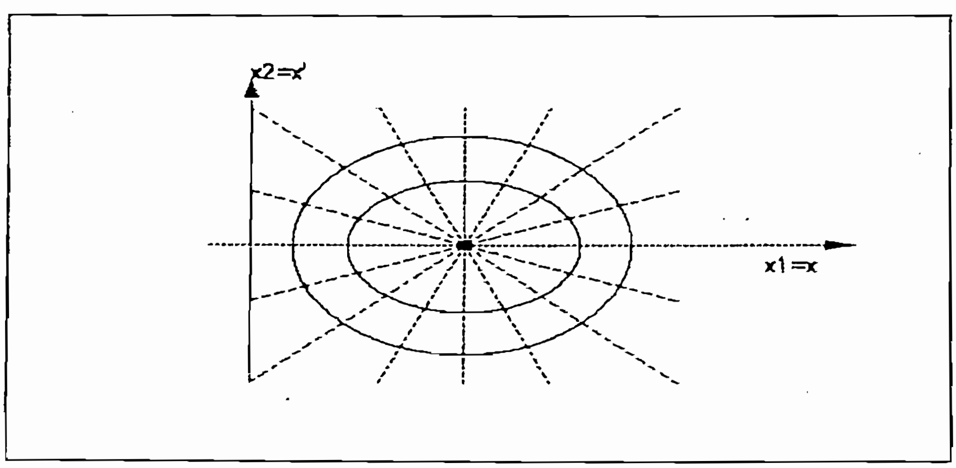


Figura 2.6 Isóclinas y trayectorias en el plano de fase correspondientes al sistema del ejemplo 2.3.

2.1.5 Método delta.

En el método delta, se obtiene la trayectoria de fase como una secuencias de arcos circulares cuyos centros se desplazan a lo largo del eje x. Se puede aplicar este método a las ecuaciones de la forma.

$$\ddot{x} = -f(\dot{x},x,t) \qquad (2.25)$$

donde $f(\dot{x},x,t)$ puede ser lineal o no lineal y como se indica, puede ser variable en el tiempo; pero debe ser continua y tener valor único .

Al aplicar este método, se modifica la ecuación anterior a la ecuación de la forma

$$\ddot{x} + w^2 x = - f(\dot{x},x,t) + w^2 x \qquad (2.26)$$

y se define la función delta como

$$\delta(\dot{x},x,t) = \frac{-f(\dot{x},x,t) + w^2 x}{w^2} \qquad (2.27)$$

El termino w^2x añadido a ambos miembros de la ecuación debe elegirse adecuadamente para que los valores de la función delta no sean ni demasiado pequeños ni demasiado elevados para el rango de los valores $\dot{x},x,t$ considerados.

Utilizando la función delta, la ecuación del sistema puede escribirse como

$$\ddot{x} + w^2 x = w^2 \, (\dot{x}, x, t) \qquad (2.28)$$

La función delta depende de las variables $\dot{x}, x, t$; sin embargo para pequeñas variaciones de esas variables se puede considerar a la función delta como constante en cada punto p_i de estado, pudiendo entonces en la vecindad del punto p_i de estado modificarse la ecuación como

$$\ddot{x} + w^2(x - \delta i) = 0 \qquad (2.29)$$

donde

$$\delta i = \text{constante} = \frac{-f(\dot{x}, x, t) - w^2 x}{w^2} \qquad (2.30)$$

La ecuación (2.30) es la de un movimiento armónico simple. Las trayectorias de este sistema son circunferencias con centros en $x = x_i$, $\dot{x}/w = 0$ en el plano $x - \dot{x}/w$. Entonces, para un pequeño incremento en la vecindad del estado $x = x_i$, $\dot{x}/w = \dot{x}_i/w_i$, $t = t_i$. La trayectoria es un arco de circunferencia, como se muestra en la figura 2.6.

centro : ($x = x_i$, $(\dot{x}/w) = 0$)

radio : $\sqrt{(\dot{x}_i/w_i)^2 - (x_i - \delta_i)^2}$

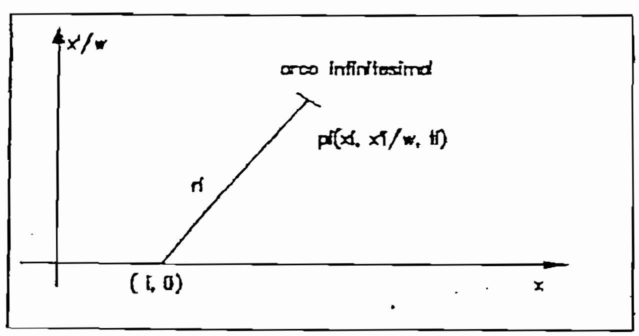


figura 2.7 Segmento de trayectoria obtenido con el método Delta

Es esencial utilizar el plano de fase normalizado y escalas iguales en los ejes , tal que el arco corresponda a una circunferencia con centro y radio constante.

Ejemplo 2.4 .[2]

Sea la ecuación (2.31) que define un sistema de segundo orden.

$$\frac{d^2x}{dt^2} + 2\xi\frac{wdx}{dt} + w^2x = 0 \qquad (2.31)$$

La escribimos nuevamente como la ecuación (2.32)

$$\frac{d^2x}{dt^2} + w^2x = -2\xi\frac{wdx}{dt} \qquad (2.32)$$

Si se hace

$$\frac{\dot{x}}{w} = \frac{dx}{dt}\frac{1}{w} = \frac{dx}{d\tau} = y$$

La ecuación (2.32) se convierte en

$$\frac{d^2x}{d\tau^2} + x = \delta \qquad (2.33)$$

donde $\delta = -2\xi y$.

Para $y = y1 = const$, la ecuación (2.33) indica que la trayectoria en el plano $x-(dx/d\tau)$ es una circunferencia cuyo centro está ubicado en $x=-2\xi y_1$, $y = 0$. Como la posición del centro depende de y_1, el centro de la circunferencia se desplaza a lo largo del eje x al variar y, para trazar la trayectoria de fase es conveniente trazar una línea $x = \delta$, es decir $\delta = -2\xi y$ en el plano $x - y$. En la figura 2.7 se ve el trazo de la trayectoria.

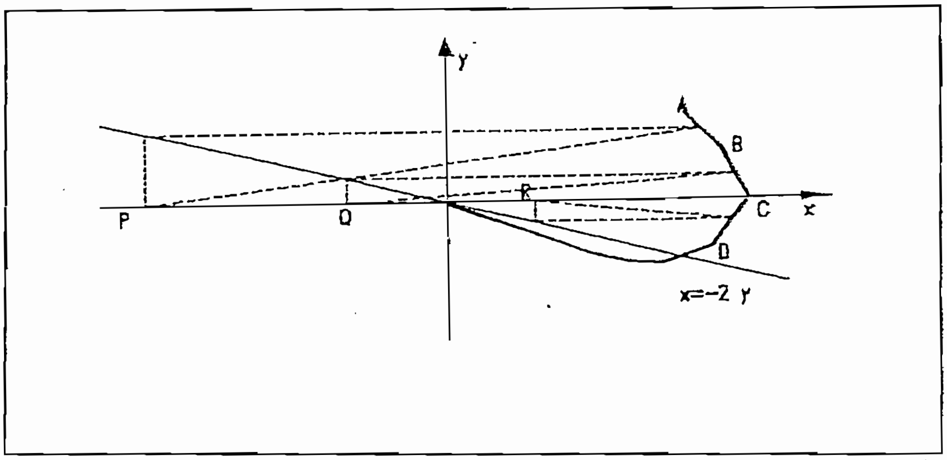


Figura 2.7 Diagramas que muestran el trazado de trayectorias por el método Delta para el sistema del ejemplo 2.4.

El método delta es un método general en el sentido que se pueden trazar las trayectorias en el plano de fase independientemente de si la ecuación del plano de fase representa un sistema lineal, no lineal, variable o invariable en el tiempo, con señal de entrada variable o invariable en el tiempo.

2.2 CARACTERISTICAS RELEVANTES DE LAS TRAYECTORIAS

Una trayectoria es una curva que ,se ha dicho, muestra la dinámica del sistema para una condición inicial determinada. El retrato de fase de un sistema es una familia de curvas que describen la respuesta del sistema para todas las condiciones iniciales posibles, normalmente cuando la referencia del sistema es constante. La respuesta del sistema se sabe depende de los parámetros de la planta en el caso de sistemas lineales; pero en el caso de sistemas no lineales, esta depende además de las condiciones iniciales .

Describiremos las caracteristicas relevantes de las trayectorias de fase partiendo del comportamiento conocido de las respuestas de los sistemas lineales de segundo orden a señales paso excitadoras.

Para los sistemas lineales, la característica de las trayectorias está ligada a las raices de la ecuación característica del sistema.

Sea la ecuación diferencial lineal de segundo orden :

$$\ddot{x} + a\dot{x} + bx = 0 \tag{2.34}$$

la naturaleza de la solución es determinada por las raíces de la ecuación caracteristica :

$$\lambda^2 + a\lambda + b = 0 \tag{2.35}$$

se supone que a y b son constantes diferentes de cero. La ubicación de las raices de esta ecuación, que se asume sean λ_1 y λ_2 , en el plano complejo determina la característica de la trayectoria. Se puede ver que se dan los seis casos siguientes respecto de la naturaleza y ubicación de λ_1 y λ_2 .

1ro. son reales, distintas y están en el semiplano derecho
2do. son reales, distintas y están en el semiplano izquierdo
3ro. son complejas conjugadas en el semiplano derecho
4to. son complejos conjugadas en el semiplano izquierdo
5to. son complejas conjugadas sobre el eje jw
6to. son reales , distintas, una en el semiplano izquierdo y otra en el semiplano derecho

Para los casos en que las dos o una de las raíces está en el semiplano derecho el sistema es inestable, y, por tanto su curva de respuesta es una trayectoria que empezando en cualquier condición inicial diverge hacia el infinito.

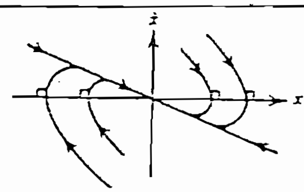
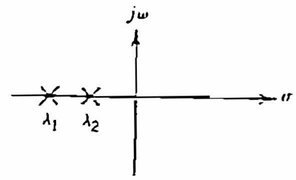
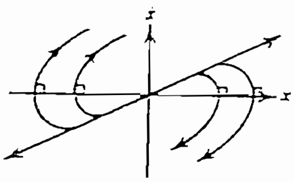
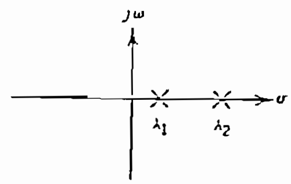
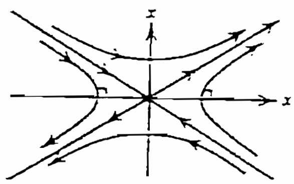
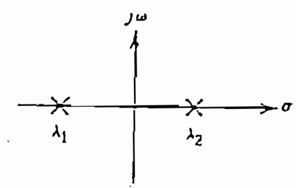
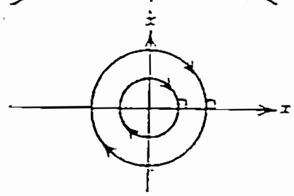
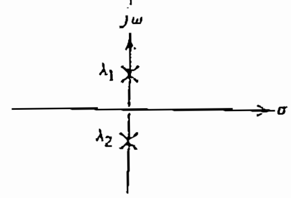
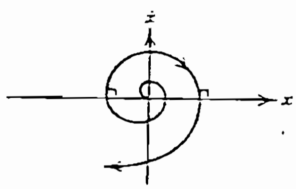
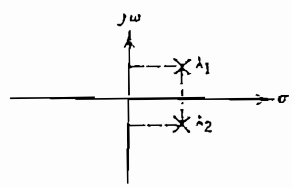
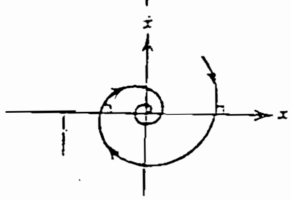
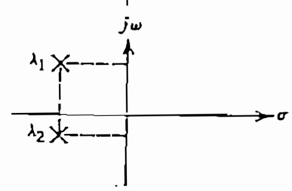
Para los casos en que las dos las raices están en el semiplano izquierdo el sistema es estable, y, por tanto su curva de respuesta es una trayectoria que empezando en cualquier condición inicial converge a un punto de equilibrio.

Para el caso en que las dos las raíces están sobre el eje jw del plano complejo el sistema es puramente oscilatorio,y , por tanto su curva de respuesta es una trayectoria cerrada sobre si misma, que empezando en cualquier condición inicial retorna a dicha condición después de cierto tiempo.

Para el caso en que las dos son reales, pero una de las raíces está en el semiplano derecho y otra está en el semiplano izquierdo el sistema presenta trayectorias que separan al plano de fase en sectores con comportamiento distinto, inestable en general.

Se indicó que las trayectorias son curvas que no se cortan, es decir por un punto del plano de fase pasa una y solo una trayectoria, excepto en los puntos singulares. La tabla 2.1 siguiente muestra los puntos singulares asociados a las seis posibles ubicaciones de las raices de la ecuación característica y los nombres con los que se los conoce.

TABLA 2.1: PUNTOS SINGULARES

GRÁFICAS DEL PLANO DE FASE	TIPO DE SINGULARIDAD	RAÍCES λ_1, λ_2 EN EL PLANO S
	NODO ESTABLE *(a)*	
	NODO INESTABLE *(b)*	
	PUNTO SILLA *(c)*	
	CENTRO *(d)*	
	FOCO INESTABLE *(e)*	
	FOCO ESTABLE *(f)*	

Muchas veces es conveniente una vez conocidas las raíces del sistema escribir las ecuaciones de estado en forma más simple, como se muestra a continuación.

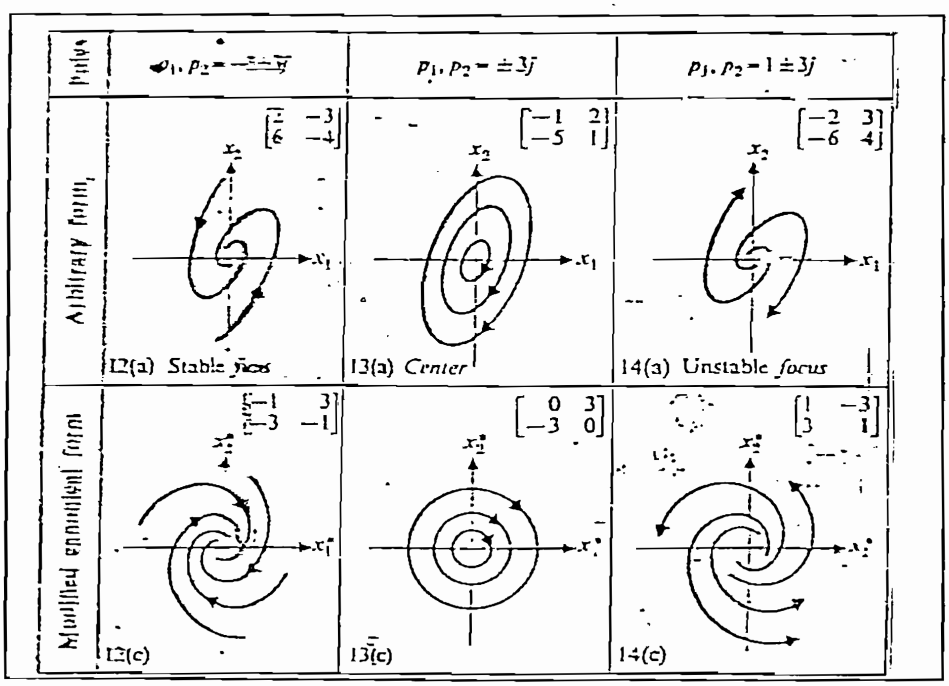
$$\begin{aligned} x_1 &= \lambda_1\, x_1 + 0\, x_2 \\ x_2 &= 0\, x_1 + \lambda_2\, x_2 \end{aligned} \qquad (2.37)$$

Que en forma matricial se escribe

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (2.38$$

Esto se refleja en el plano de fase como una alineación de los vectores propios con los ejes coordenados, lo que da mayor simetría al retrato de fase. La tabla 2.2 muestra ejemplos de sistemas de segundo orden obtenidos a partir de las ecuaciones de estado en esta forma y los obtenidos a partir de la forma original de dichas ecuaciones de estado.

TABLA 2.2: TRAYECTORIAS DE FASE EN EL PLANO



Si un punto singular de un sistema lineal es un punto de equilibrio estable, es fácil apreciar que para un sistema lineal su estabilidad local implica estabilidad global .

Los sistemas no lineales pueden tener uno o mas de los puntos singulares mostrados en la tabla 2.1, por ello, no es posible una clasificación de sus retratos de fase. Y la estabilidad local en un punto singular no asegura la estabilidad global del sistema no lineal.

La figura 2.8 muestra el retrato de fase de un sistema no lineal de segundo orden con dos puntos singulares : una silla de montar y un nodo estable. El sistema es estable solo en una región del plano de fase a pesar de que un punto singular presenta estabilidad local.

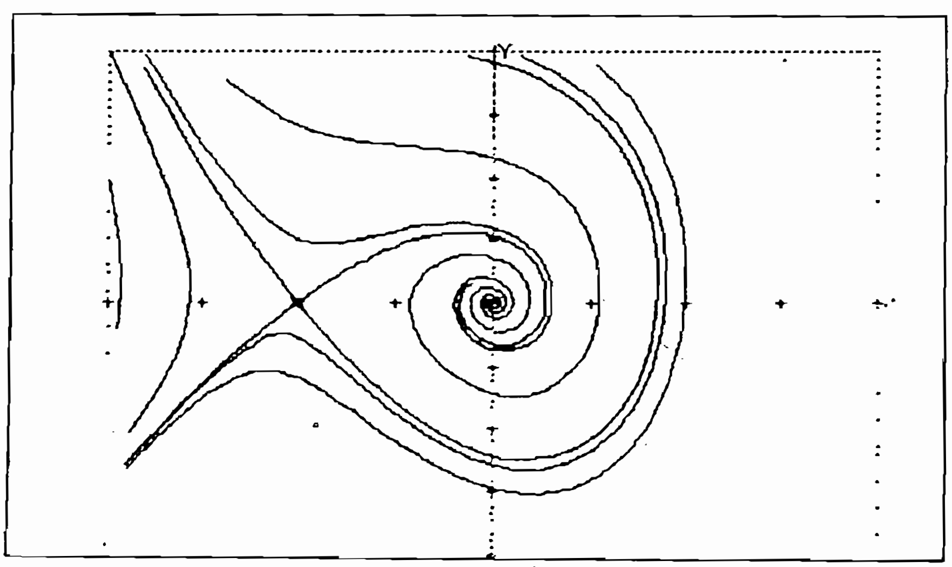
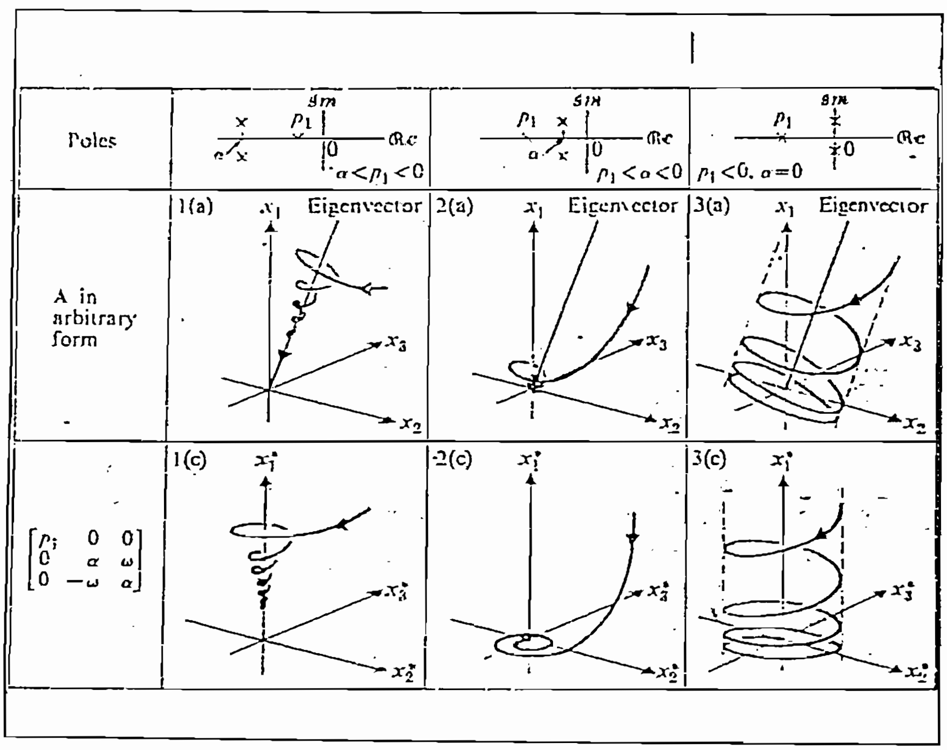


Figura 2.8 Retrato de fase de un sistema no lineal con dos puntos singulares: una silla de montar y un nodo estable.

La ecuación característica para un sistema de tercer orden es una ecuación cúbica, y por tanto, al menos uno de los valores propios debe ser real y al menos una línea recta aparecerá en el modelo de la trayectoria para un sistema de tercer orden (en un espacio de estado tridimensional). Algunos modelos de trayectorias son mostrados en la tabla 2.3 para sistemas de tercer orden con tres polos, un polo real p_1 y un par de polos complejos conjugados p_2 y p_3 .

La respuesta del sistema de control esta formado por dos modos, uno exponencial y otro oscilatorio. Dependiendo de cúal de los modos está más fuertemente amortiguado, las trayectorias seguirán los modelos 1, 2 ó 3 de la tabla 2.3.

TABLA 2.3 TRAYECTORIAS DE FASE TRIDIMENSIONALES



Si el modo oscilatorio es más pesadamente amortiguado que el modo exponencial, la trayectoria se aproximará al vector propio y seguirá al vector propio hasta el origen (tabla 2.3.1a) .

Por otro lado, si el amortiguamiento del modo oscilatorio no es mas grande que el modo exponencial, entonces con el incremento del tiempo la trayectoria de respuesta se acerca a una oscilación amortiguada que toma lugar en el plano a través de el origen (tabla 2.3.2a).

Las trayectorias se apoyan en un cilindro cuando el modo oscilatorio no tiene amortiguamiento (tabla 2.3.3a y 3c) .

Las trayectorias para la forma canónica están mostradas en la tabla 2.3, cuadros 1c, 2c y 3c.

No hay un método gráfico simple conocido para la determinación de las trayectorias tridimensionales. Aunque es posible obtener proyecciones sobre los planos coordenados, la trayectoria proyectadas pueden cruzarse y es a veces difícil dar una idea de la característica de la trayectoria tridimensional desde sus proyecciones. En el siguiente capítulo se usa simulación digital para el trazo de las trayectorias de sistemas de tercer orden.

Una descripción geométrica de trayectorias llega a ser imposible para sistemas de orden superior al tercero. Sin embargo los modos para estos polos desaparecen rápidamente y el resto de la respuesta puede ser investigada en un espacio de menor dimensión.

Resumiendo, para un sistema no lineal podemos tener tres tipos de comportamiento de la trayectoria de respuesta dependiendo de la condición inicial tomada en el espacio de estado . Si el sistema es inestable esta trayectoria diverge hacia el infinito. Si el sistema es estable la trayectoria converge hacia un punto finito del espacio de estado. Pero si el sistema presenta oscilaciones permanentes la trayectoria se cerrará sobre si misma, en cuyo caso la trayectoria se llamará CICLO LIMITE.

En el siguiente capítulo se muestra los algoritmos y el desarrollo de un programa para la obtención por computadora de las trayectorias de fase de sistemas de hasta tercer orden.

CAPITULO III

ALGORITMOS Y DESARROLLO DE PROGRAMAS

3.1 DESCRIPCION DEL SISTEMA

El Programa Digital para Trazo de Trayectorias de Fase hasta en Tres Dimensiones, denominado TTF, es un sistema de software destinado a ayudar en el análisis de sistemas de control , sean estos lineales o no lineales, de primero, segundo o tercer orden.

Como se indicó en el capítulo I, el análisis de estabilidad permite la elección adecuada de los parámetros del sistema . Para éste efecto el plano de fase da información suficiente de la respuesta del sistema de control. Sin embargo, como se observa en los ejemplos de los métodos de graficación para el trazo de trayectorias en el plano de fase, presentados en el capítulo II , el proceso de graficación manual es muy lento y de mucho cuidado para obtener resultados numéricos fiables; además, impide realizar cambios, rápidamente en los parámetros, que se pueden requerir dentro de un proceso normal de análisis o diseño. La facilidad que presenta el sistema de sofware para realizar el trazo de trayectorias de fase en forma rápida y sencilla hace que su implementación y conocimiento sea importante en la especialización de Control.

3.1.1 FACILIDADES Y ALCANCE

Este programa permite la obtención de las trayectorias fase para su posterior análisis . Puede manejar sistemas de primer o segundo orden y obtener las trayectorias en el plano de fase (x2-x1) o sistemas de tercer orden y obtener las trayectorias en el espacio de estado (x1-x2-x3).

Puede manejarse estos sistemas a partir de dos representaciones : el diagrama de bloques del sistema o el conjunto de ecuaciones de estado del mismo.

Los resultados a través de la impresora pueden ser las trayectorias en el plano de fase para sistemas de segundo orden, las trayectorias en el espacio de estado para sistemas de tercer orden y la característica de transferencia del elemento no lineal del sistema de control.

El programa puede manejar sistemas lineales o no lineales, puesto que, como se mostró en el capitulo I, el conjunto de sistemas no lineales contiene al subconjunto de sistemas lineales, los sistemas pueden ser de hasta tercer orden con con cualquier grado de potenciación e incluir funciones trigonométricas o exponenciales .

3.1.2 REQUERIMIENTOS DE HARDWARE

Se requiere trabajar con:

* Un computador IBM PC/XT/AT/PS o compatibles, con las

siguientes características :

- Sistema operativo 2.0 o más actualizado.
- dos drives de diskettes (360k/720k) o disco duro.
- mínimo 512 k de memoria RAM.
- tarjeta de gráficos CGA con monitor de colores o color sencillo, o tarjeta Hércules o el emulador respectivo y monitor monocromático.

* Para la impresión de resultados se necesita una impresora ; EPSON de matriz de puntos de la serie FX/MX/RX, IBM Graphics Printer o alguna equivalente a las descritas. Normalmente la impresora debe conectarse a un pórtico paralelo del computador.

3.1.3 MODULOS DEL PROGRAMA

Para un trabajo básico del programa se requiere por lo menos dos diskettes , el primero con los archivos del programa y el otro para guardar los resultados del trabajo con el programa. Los archivos esenciales que contiene el diskette de programa son :

1.- TTF.EXE que contiene el programa digital principal de trazo de trayectorias de fase para sistemas de hasta tres dimensiones.

2.- FT contiene información adicional para el programa TTF para el trazo de trayectoria a partir del diagrama de bloques del sistema.

3.- ECU contiene información adicional para el programa TTF para el trazo de trayectorias a partir de las ecuaciones de estado.

4.- PIZZAS.EXE contiene el programa principal para la impresión de resultados visualizados en la pantalla del computador, a través de la impresora.

5.- PZFASE contiene información adicional para el programa PIZZAS.EXE para la impresión de las trayectorias.

6.- PZGRAF contiene información adicional para el programa PIZZAS.EXE para la impresión de la característica de transferencia de la no linealidad.

7.- ERROR#.EJM, VELOZ#.EJM contiene información adicional para el programa TTF.EXE para visualización en pantalla de ejemplos de retratos de fase obtenidos con este programa.

8.- Puede ser necesario un emulador de tarjeta de gráficos (MSHERC, QBHERC, SIMCGA).

3.1.4 EJECUCION DE LOS DIFERENTES PROGRAMAS

3.1.4.1 CREACION / EDICION DE LAS TRAYECTORIAS

El programa que permite la creación /edición de trayectorias es el programa TTF.EXE. Los pasos necesarios para su ejecución son :

- Arrancar el computador con el sistema operativo.
- Dependiendo de la tarjeta de gráficos, ejecutar o no uno de los emuladores de pantalla.
- Si se está trabajando en dos diskettes se coloca el disco del programa en la unidad A, y el de aplicaciones en la unidad B. Si se está trabajando con

un disco duro es necesario cambiarse al directorio que contiene los programas TTF.

3.1.4.2 SALIDA A LA IMPRESORA.

Para obtener los resultados en la impresora puede procederse de la siguiente manera:

Desde el programa de trazo de trayectorias, luego de la creación/ edición de trayectorias se escoge, en el menú principal, la opción IMPRIMIR y se siguen las instrucciones que presenta el programa. Desde el programa de trazo de trayectorias luego de la edición en pantalla de la característica de transferencia no lineal, se activa el programa de impresion y se sigue las instrucciones que presenta el programa en la opción IMPRIMIR.

El programa PIZZAS.EXE es activado en ambos casos desde adentro del trabajo del TTF pulsando las teclas <Shift><PrScr>, este programa tiene su propio menú de opciones. Al escoger la opción de impresión el programa TTF instruye sobre las opciones que deben seleccionarse del menú del programa PIZZAS para obtener una impresión de los resultados en el formato adecuado.

El programa PIZZAS es un paquete desarrollado por la casa Microsoft, y sirve únicamente para capturar la pantalla presente y su envío a la impresora.

3.1.5 TRABAJO CON EL TTF

3.1.5.1 INICIALIZACION DEL PROGRAMA

Estando en el sistema operativo, se ingresa el diskette del TTF en el drive A y se teclea

A> TTF <ENTER>

Una vez ejecutado el comando en el computador, aparece en pantalla un despliegue de información general,y al presionar cualquier tecla cualquiera, esta información desaparece y en su lugar se presenta el menú principal del programa.

(1). Trazado de trayectorias a partir del diagrama de bloques del sistema.

(2). Trazado de trayectorias a partir de las ecuaciones de estado del sistema.

(3). Impresión de resultados

(4). Ejemplos.

(5). Cambios en el programa.

(6). Salir.

La selección de opciones se realiza por simple pulsación del número que las identifica.

El programa detecta los errores en la selección de opciones de este y los posteriores menús o en el ingreso de datos y da razón de ellos mediante una línea de mensajes . Se puede borrar el texto del error y continuar trabajando únicamente repitiendo la selección o el ingreso de datos en forma correcta.

Se procederá ahora a analizar cada opción que se presenta en el menú principal del programa TTF, debiendo indicarse que el programa entero se ha hecho de manera conversacional, es decir es el propio programa que requiere los datos de entrada y verifica si estos son coherentes con el tipo de dato.

3.1.5.2 TRAZO DE TRAYECTORIAS A PARTIR DEL DIAGRAMA DE BLOQUES DEL SISTEMA

Escogiendo la opción (1) del menú principal, se pasa a la representación en diagrama de bloques para el ingreso de datos del sistema de control.

El diagrama de bloques mostrado en la pantalla corresponde a un sistema de control con realimentación unitaria, etapa de compensación, elemento de ganancia no lineal y parte lineal de la planta, que es el sistema más general que puede manejar el programa, según la figura 3.1.

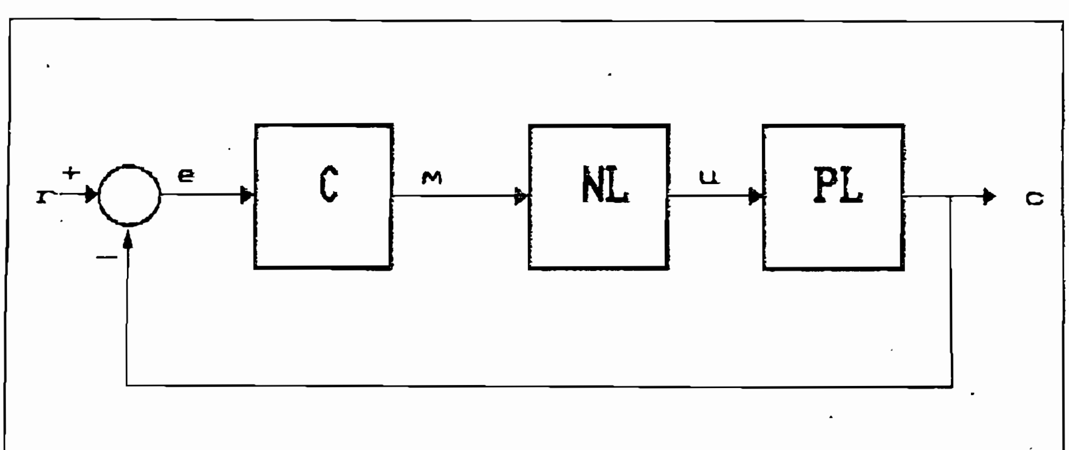


Figura 3.1 Diagrama de bloques presentado para el ingreso de los datos del sistema de control.

Los datos que se requieren ingresar, según la opción seleccionada, son los mostrados en la tabla 3.1. La selección de las opciones se realiza por simple pulsación del número que las identifica y el ingreso del dato requerido por el programa se realiza tecleando el dato numérico y presionando a continuación la tecla <ENTER>.

TABLA 3.1

Datos que deben ingresarse para cada etapa del sistema de control según la opción seleccionada en ella.

ETAPA	OPCION	DATO REQUERIDO
entrada	(1) escalón	amplitud del escalón (Resc)
	(2) rampa	pendiente de la rampa (Rrp)
compensación	(1) proporcional	valor de kp
	(2) proporcional derivativa	valor de kp y kd
	(3) proporcional integral	valor de kp y ki
	(4) proporcional derivativa integral	valor de kp de kd y ki
no linealidad	(1) si	coordenadas
	(2) no	--------
parte lineal	(1) 1er orden (2) 2do orden (3) 3er orden	valor de los coeficientes de la funcion de transferencia

El programa requiere también el número de trayectorias que se han de trazar y las condiciones iniciales de las variables de estado para cada trayectoria.

Ingresados todos los datos, se presenta en la pantalla las trayectorias de fase correspondientes a estos datos.

Se active o no el programa de impresión, a continuación el programa permite el cambio del valor de entrada de la señal para la obtención de nuevas trayectorias de fase sin introducir los datos del sistema nuevamente; este proceso continua hasta que se responda negativamente a la pregunta

< NUEVAS CONDICIONES INICIALES ? >

En este caso se retorna al menú principal del programa TTF.EXE

3.1.5.3 TRAZO DE TRAYECTORIAS DE FASE A PARTIR DE LAS ECUACIONES DE ESTADO.

Escogiendo las opción (2) del menú principal, se pasa a la representación en variables de estado para el ingreso de datos del sistema.

El primer dato que se debe ingresar es el número de ecuaciones que representan al sistema. Esto porque aunque el sistema sea de primero, segundo o tercer orden no necesariamente tiene 1,2 o 3 ecuaciones respectivamente, sino mayor, esto es claro en los sistemas no lineales aproximados a sistemas lineales por tramos o a sistemas que tienen un elemento lineal por segmentos. Cabe anotar que no es que en la realidad las no linealidades van constituidas por segmentos de recta, aunque se procede así para una aproximación matemática de esta. En cuyo caso el plano de fase es dividido en sectores que responden a diferentes ecuaciones de estado. Por ello, el segundo dato que requiere el programa es el tipo de restricción que tiene la ecuación de estado a introducirse seguidamente, las restricciones pueden ser:

(0) ninguna

(1)	$x_1>a_1$	(5)	$x_2>b_1$	(A)	$x_3>c_1$
(2)	$x_1<a_2$	(6)	$x_2<b_2$	(B)	$x_3<c_3$
(3)	$a_3<x_1<a_4$	(7)	$b_3<x_3<b_4$	(C)	$c_3<x_3<c_4$
(4)	$a_5>x_1>a_6$	(8)	$b_5>x_2>b_6$	(D)	$c_5>x_3>c_6$

La selección de la restricción se realiza por simple pulsación del número que la identifica.

El modelo de la ecuación de estado es una función no lineal polinómica y trigonométrica, cuyo término general es de la forma:

$$\text{term gen} = \text{coeficiente} * x_1^{p1} * x_2^{p2} * x_3^{p3} * f_1(x_1) * f_2(x_2) * f_3(x_3)$$

Los datos que se deben ingresar son, para cada término, el coeficiente, los exponentes de las variables y las funciones trigonométricas o exponenciales si las hubiere.

El programa requiere también el número de trayectorias que se han de trazar y las condiciones iniciales de las variables de estado para cada trayectoria.

Ingresados todos los datos, el programa edita en pantalla las trayectorias de fase del sistema. A continuación de activar o no el programa de impresión, el programa presenta la opción de ingresar nuevas condiciones iniciales, para analizar el mismo sistema sin tener que ingresar nuevamente los datos. Este proceso continúa hasta que se conteste negativamente a la pregunta.

< NUEVAS CONDICIONES INICIALES ? >

En cuyo caso se puede ingresar los datos de un nuevo sistema o retornar al menú principal del programa TTF.EXE.

3.1.5.4 IMPRESION DE RESULTADOS Y EJEMPLOS.

Escogiendo la opción (3), se presenta en pantalla las instrucciones para manejar el programa de impresión, y los tipos de resultados a imprimir

[1] Trayectorias de fase
[2] Respuesta en el tiempo
[3] Datos numéricos

Con esta selección el programa muestra en la pantalla los resultados pedidos permitiendo, en los gráficos, cambiar el rango de visualización antes de imprimirlos.

También puede obtenerse el gráfico de alguna de las alinealidades del menú gráfico de 16 no linealidades típicas, el mismo que se presenta para la selección del tipo de no linealidad en la representación del sistema empleando diagrama de bloques, activando el programa de impresión desde ese punto, según las instrucciones antes indicadas en la opción (3) del menú principal.

3.1.5.6 FINALIZACION DEL PROGRAMA

Casi tan importante como conocer la forma empezar una sesión de trabajo, es como salir del programa. Siempre que se desee retornar al DOS o terminar la sesión de trabajo, debe terminarse de ingresar los datos que pida el programa hasta obtenerse cualquier gráfico, entonces se selecciona las opciones que permitan el retorno del programa al menú principal donde puede elegirse la opción [5] SALIR con lo cual definitivamente se sale al DOS. Es recomendable resetear el sistema para evitar que algún comando del programa altere el funcionamiento usual con otros programas.

3.2 METODO DE SOLUCION

Para mostrar el método de solución utilizado en este trabajo, se procede a un desarrollo matemático en base a las dos representaciones del sistema de control utilizadas en este trabajo, en diagrama de bloques y en variables de estado.

3.2.1 EN REPRESENTACION DE DIAGRAMA DE BLOQUE.

Para el trazo de trayectorias en el plano de fase a partir del diagrama de bloques, se han obtenido las ecuaciones de estado para el sistema más general. Este conjunto de ecuaciones involucran todas las alternativas que se presentan en cada etapa del sistema de control.

Estas ecuaciones se construyen partiendo de las ecuaciones que variables presentes.

Los bloques funcionales del diagrama de bloque general, que permite el ingreso de los datos al programa, y que se usarán para obtener las ecuaciones de estado son los siguiente:

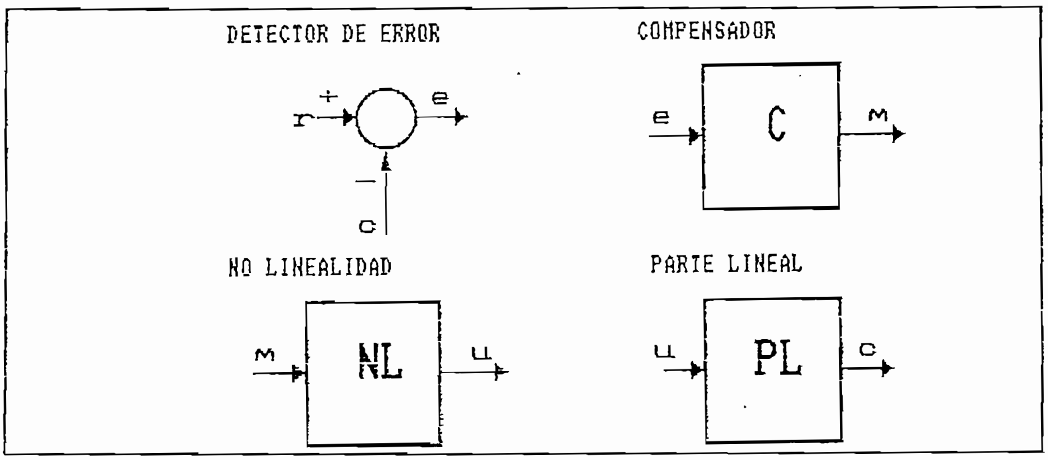


Figura 3.2 Bloques funcionales del Diagrama de bloques del sistema de control más general que puede manejar el programa

Las ecuaciones se obtienen como sigue:

3.2.1.1. PARTE LINEAL

Se tiene tres posibilidades, que la parte lineal tenga como función de transferencia una de primer orden, una de segundo orden o una de tercer orden. Se obtendrá en este punto las ecuaciones de estado para estos tres casos.

<u>Primer Grado</u>

$$u * \frac{b_o}{a_1 s + a_o} = c$$

$$b_o u = a_1 s c + a_o c \tag{3.1}$$

Segundo Grado

$$u * \frac{b_1 s + b_o}{a_2 s^2 + a_1 s + a_o} = c$$

$$b_1 s u + b_o u = a_2 s^2 c + a_1 s c + a_o c \tag{3.2}$$

Tercer Grado

$$u * \frac{b_2 s^2 + b_1 s + b_o}{a_3 s^3 + a_2 s^2 + a_1 s + a_o} = c$$

$$b_2 s^2 u + b_1 s u + b_o u = a_3 s^3 c + a_2 s^2 c + a_1 s c + a_o c \tag{3.3}$$

3.2.1.2 DETECTOR DE ERROR

Aqui se puede seleccionar como señal de entrada una función rampa o una función escalón. Estas dos posibilidades pueden expresarse como una función suma de rampa y escalón, en la representación matemática del sensor de error.

$$e = r - c$$
$$e = (R_{rp} * t + R_{esc}) - c \tag{3.4}$$

Las derivadas del error serán

$$\dot{e} = R_{rp} - \dot{c}$$
$$\ddot{e} = -\ddot{c} \tag{3.5}$$

3.2.1.3 NO LINEALIDAD

Al analizar este bloque, tenemos la posibilidad de que la no linealidad sea una de las 16 no linealidades aproximadas por segmentos, que presenta el menú del programa; una vez conocida se tiene la posibilidad de que el punto de operación se encuentre en cualquiera de los tramos lineales que forman la no linealidad. Para ello se indica con subindices a que no linealidad y a que tramo de ella corresponden los coeficientes de la ecuación.

$$u = k_1(\#nl,tr)*m + k_2*(\#nl,tr) \qquad (3.6)$$

Unicamente con el valor de m se conoce el tramo donde se encuentra el punto de operación en la característica de transferencia de la no linealidad, y se puede saber el valor de las constantes k_1 y k_2.

3.2.1.4 COMPENSACION

En este bloque se tiene la posibilidad de incluir cuatro tipos de compensación, proporcional, proporcional derivativa, proporcional integral y proporcional integral derivativa. Para la representación matemática, pueden resumirse en una sola ecuación, expresando la compensación como una combinación de las tres como sigue .

$$m = k_p*e + k_d*se + k_i*e/s \qquad (3.7)$$

Es importante recordar que la compensación integral aumenta en uno el orden del sistema.

3.2.1.5. ECUACIONES GENERALES

Enlazando los bloques y las respectivas ecuaciones se obtienen las siguientes representaciones de los sistemas de control en el dominio del tiempo.

Sistema Lineal de Tercer Orden con Compensación Proporcional Derivativa.

$$
\begin{aligned}
(b_2k_d + a_3)\dddot{e} = & -(b_2k_p + b_1k_d + a_2)\ddot{e} \\
& -(b_1k_p + b_ok_d + a_1)\dot{e} \\
& -(b_ok_p + a_o)e \\
& + a_1R_{rp} + a_o(R_{rp}*t + R_{esc})
\end{aligned}
\tag{3.8}
$$

Sistema Lineal de Segundo Orden con Compensación Proporcional Integral Derivativa

$$
\begin{aligned}
(b_1k_d + a_2)\dddot{e} = & -(b_1k_p + b_ok_d + a_1)\ddot{e} \\
& -(b_1k_i + b_ok_p + a_o)\dot{e} \\
& + b_ok_ie + a_oR_{rp}
\end{aligned}
\tag{3.9}
$$

Sistema Lineal de Segundo Orden con Compensación Proporcional Derivativa

$$
\begin{aligned}
(b_1k_d + a_2)\ddot{e} = & -(b_1k_p + b_ok_d + a_1)\dot{e} \\
& -(b_ok_p + a_o)e + a_1R_{rp} \\
& + a_o(R_{rp}*t + R_{esc})
\end{aligned}
\tag{3.10}
$$

Sistema Lineal de Primer Orden con Compensación Proporcional Integral Derivativa

$$(b_ok_d + a_1)\ddot{e} = -(b_ok_p + a_o)\dot{e} - b_ok_ie + a_oR_{rp}$$

Sistema Lineal de Primer Orden con Compensación Proporcional

Derivativa

$$(b_o k_d + a_1)\dot{e} = -(b_o k_p + a_o)e + a_o(R_{rp} * t + R_{ssc}) \qquad (3.11)$$

Sistema No Lineal de Tercer Orden con Compensación Proporcional Derivativa

$$\begin{aligned}(b_2 k_1 k_d + a_3)\dddot{e} = &-(b_2 k_1 k_p + b_1 k_1 k_d + a_2)\ddot{e} \\ &-(b_1 k_1 k_p + b_o k_1 k_d + a_1)\dot{e} \\ &-(b_o k_1 k_p + a_o)e \\ &-b_o k_2 + a_1 R_{rp} + a_o(R_{rp} * t + R_{ssc})\end{aligned} \qquad (3.12)$$

Sistema No Lineal de Segundo Orden con Compensación Proporcional Derivativa

$$\begin{aligned}(b_1 k_1 k_d + a_2)\ddot{e} = &-(b_1 k_1 k_p + b_o k_1 k_d + a_1)\dot{e} \\ &-(b_o k_1 k_p + a_o)e \\ &-b_o k_2 + a_1 R_{rp} + a_o(R_{rp} * t + R_{ssc})\end{aligned} \qquad (3.13)$$

Sistema No Lineal de Primer Orden con Compensación Proporcional Derivativa

$$\begin{aligned}(b_o k_1 k_d + a_1)\dot{e} = &-(b_o k_1 k_p + a_o)e \\ &-b_o k_2 + a_1 R_{rp} + a_o(R_{rp} * t + R_{ssc})\end{aligned} \qquad (3.14)$$

Tomando como variables de estado

$$\begin{aligned} x_1 &= e \\ x_2 &= \dot{e} \\ x_3 &= \ddot{e} \end{aligned} \qquad (3.15)$$

tenemos un grupo de ecuaciones diferenciales de primer grado, ecuaciones de estado, que representan a los sistemas de control .

Sistema Lineal de Tercer Orden con Compensación Proporcional

Derivativa.

$$\dot{x}_3 = [\ -(\ b_2k_p\ +b_1k_d\ +a_2\ )x_3$$
$$-(\ b_1k_p\ +b_ok_d\ +a_1\ )x_2$$
$$-(\ b_ok_p\ +a_o\ )x_1$$
$$+\ a_1R_{rp}\ +a_o(\ R_{rp}*t\ +R_{esc}\ )\ ]\ /\ (b_2k_d\ +a_3)$$
$$\dot{x}_2 = x_3$$
$$\dot{x}_1 = x_2 \qquad (3.16)$$

Sistema Lineal de Segundo Orden con Compensación Proporcional Integral Derivativa

$$\dot{x}_3 = [\ -(\ b_1k_p\ +b_ok_d\ +a_1\ )x_3$$
$$-(\ b_1k_i\ +b_ok_p\ +a_o\ )x_2$$
$$+\ b_ok_ix_1\ +a_oR_{rp}\ ]\ /\ (b_1k_d\ +a_2)$$
$$\dot{x}_2 = x_3$$
$$\dot{x}_1 = x_2 \qquad (3.17)$$

Sistema Lineal de Segundo Orden con Compensación Proporcional Derivativa

$$\dot{x}_2 = [\ -(\ b_1k_p\ +b_ok_d\ +a_1\ )x_2$$
$$-(\ b_ok_p\ +a_o\ )x_1$$
$$+\ a_1R_{rp}\ +a_o(R_{rp}*t\ +R_{esc}\ ]\ /\ (b_1k_d\ +a_2)$$
$$\dot{x}_1 = x_2 \qquad (3.17)$$

Sistema Lineal de Primer Orden con Compensación Proporcional Integral Derivativa

$$\dot{x}_2 = [\ -(\ b_ok_p\ +a_o\ )x_2\ -b_ok_ix_1\ +a_oR_{rp}\ ]\ /\ (b_ok_d\ +a_1)$$
$$\dot{x}_1 = x_2 \qquad (3.18)$$

Sistema Lineal de Primer Orden con Compensación Proporcional Derivativa

$$\dot{x}_1 = [\ -(\ b_ok_p\ +a_o\ )x_1\ +a_o(\ R_{rp}*t\ +R_{esc}\ )]\ /\ (b_ok_d\ +a_1)$$
$$(3.19)$$

Sistema No Lineal de Tercer Orden con Compensación Proporcional Derivativa

$$\dot{x}_3 = [\ -(\ b_2k_1k_p + b_1k_1k_d + a_2\ )x_3$$

$$-(\ b_1k_1k_p + b_ok_1k_d + a_1\ )x_2$$

$$-(\ b_ok_1k_p + a_o\ )x_1$$

$$-\ b_ok_2 + a_1R_{rp} + a_o(R_{rp}*t + R_{esc})\ ]\ /\ (b_2k_1k_d + a_3)$$

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

$$\dot{x}_1 = x_2 \qquad (3.20)$$

Sistema No Lineal de Segundo Orden con Compensación Proporcional Derivativa

$$\dot{x}_2 = [\ -(\ b_1k_1k_p + b_ok_1k_d + a_1\ )x_2$$

$$-(\ b_ok_1k_p + a_o\ )x_1$$

$$-\ b_ok_2 + a_1R_{rp} + a_o(R_{rp}*t + R_{esc})\ ]\ /\ (b_1k_1k_d + a_2)$$

$$\dot{x}_1 = x_2 \qquad (3.21)$$

Sistema No Lineal de Primer Orden con Compensación Proporcional Derivativa

$$\dot{x}_1 = [\ -(\ b_ok_1k_p + a_o\ )x_1$$

$$-\ b_ok_2 + a_1R_{rp} + a_o(R_{rp}*t + R_{esc})\ ]\ /\ (b_ok_1k_d + a_1) \qquad (3.22)$$

Este conjunto de ecuaciones se utilizarán para aplicar el método de simulación digital para la obtención de las trayectorias de fase.

3.2.2 EN REPRESENTACION DE ECUACIONES DE ESTADO

Para el trazo de trayectorias en el plano de fase a partir de la representación de variables de estado, se han obtenido las ecuaciones de estado para el sistema más general. Este conjunto de ecuaciones de estado involucran todas las alternativas que se presentan en sistemas de control no lineal autónomos de la forma.

$$\dot{x} = f(x) \tag{3.23}$$

La ecuación no lineal que se usará para obtener las trayectorias de fase mediante el programa ECU es la siguiente.

$$\dot{x} = term\#1 + \ldots + term\#n \tag{3.24}$$

donde cada término en general es de la forma

$$term\#i = coef*x_1\text{^}p_1*x_2\text{^}p_2*x_3\text{^}p_3*f_1(x_1)*f_2(x_2)*f_3(x_3) \tag{3.25}$$

La nomenclatura usada en esta ecuación se indicó anteriormente en el numeral 3.1.5.3

Como se indicó anteriormente, los sistemas no lineales pueden ser expresados como sistemas lineales por tramos, en cuyo caso las variables de estado estarán determinadas por diferentes sistemas de ecuaciones, en los diferentes sectores en que se haya dividido el espacio de estado, debido a las restricciones que serían, como se indicó anteriormente:

(0) ninguna

(1)	$x_1>a_1$	(5)	$x_2>b_1$	(A)	$x_3>c_1$
(2)	$x_1<a_2$	(6)	$x_2<b_2$	(B)	$x_3<c_2$
(3)	$a_3<x_1<a_4$	(7)	$b_3<x_2<b_4$	(C)	$c_3<x_3<c_4$
(4)	$a_5>x_1>a_6$	(8)	$b_5>x_2>b_6$	(D)	$c_5>x_3>c_6$

El programa prevé la existencia de mínimo una ecuación por cada variable de estado del sistema. Así un sistema de tres variables de estado tendrá un mínimo de tres ecuaciones de estado. El número máximo de ecuaciones es determinado por la combinación de restricciones, las cuales cubren cualquier sistema práctico.

3.2.3 SIMULACION DIGITAL

Como se indicó en los dos numerales anteriores, el cálculo de las trayectorias se hace a partir de ecuaciones de estado generales tanto partiendo del diagrama de bloques como de las variables de estado.

En ambos casos estas ecuaciones generales son de la forma

$$\dot{x} = f(x) \qquad (3.26)$$

Que se puede, para intervalos pequeños, escribir como

$$\frac{\Delta x}{\Delta t} = f(x) \qquad (3.27)$$

El método de simulación digital es un algoritmo recursivo como sigue.

$$x_{i+1} = x_i + \Delta x \qquad \text{con} \qquad \Delta x = f(x)\,\Delta t \qquad (3.28)$$

En general para tres variables de estado se tendrá

$$x1_{i+1} = x1_i + \Delta x1_i \quad ,\text{con} \quad \Delta x1_i = f1(x1_i, x2_i, x3_i)\,\Delta t$$

$$x2_{i+1} = x2_i + \Delta x2_i \quad ,\text{con} \quad \Delta x2_i = f2(x1_i, x2_i, x3_i)\,\Delta t$$

$$x3_{i+1} = x3_i + \Delta x3_i \quad ,\text{con} \quad \Delta x3_i = f3(x1_i, x2_i, x3_i)\,\Delta t$$

(3.29)

Cada nuevo punto (x_1, x_2, x_3) representa un nuevo estado alcanzado por las trayectorias de fase en el espacio de estado y debe ser almacenado para su graficación.

Sea (j) el número de trayectorias e (i) el número de puntos, se requiere entonces tres matrices $X_{j,i}$, $Y_{j,i}$, $Z_{j,i}$ para almacenar el retrato de fase completo del sistema.

Si el sistema es de primer o segundo orden la graficación se realiza directamente con las matrices $X_{j,i}$ $Y_{j,i}$.

Si el sistema es de tercer orden es necesario un cambio de base para dar la perspectiva espacial, entonces se calcula las nuevas coordenadas .[4]

$$X3D_{j,i} = -X_{j,i}*\cos(h) - Y_{j,i}*\sin(h) \qquad (3.30)$$

$$Y3D_{j,i} = -X_{j,i}*\sin(h)*\cos(v) + Y_{j,i}*\cos(h)*\cos(v) + Z_{j,i}*\sin(v)$$

donde V y H son los ángulos de rotación vertical y horizontal del sistema tridimensional respecto del punto de visualización.

La deducción de este cambio de base se indica en el anexo .

A fín de tener suficiente precisión en las trayectorias se ha escogido un intervalo de tiempo t = 0.1 . Los resultados obtenidos que se presentan en el siguiente capítulo muestra la validez de esta elección.

Una vez discutido como se ha resuelto matemáticamente el problema de obtención de las trayectorias, en el plano de fase y el espacio de estado tridimensional, para las dos formas de representación utilizadas en este trabajo, de los sistemas de control , ahora se va ha describir los algoritmos que indican la operación del programa y la de los diferentes módulos del mismo.

3.3. DESCRIPCION DE LOS ALGORITMOS

A continuación se presentan el algoritmo del programa principal TTF.EXE y los algoritmos de los módulos que contiene: FT, ECU, IMPRESION. Se presentan los diagramas de flujo para una mejor comprensión del modo de ingreso de los datos, la forma de procesarlos y la entrega de resultados.

3.3.1 ALGORITMO DEL PROGRAMA TTF.

El programa TTF.EXE y el programa PIZZAS.EXE se ejecutan desde un archivo de procesamiento por lotes, llamado TTF.BAT.
El programa PIZZAS.EXE, que se utilizará para la impresión de los resultados gráficos, se ejecuta primero y que queda residente en memoria RAM del computador para su activación posterior.
Seguidamente ejecuta el programa TTF.EXE.
Este es el programa que inicializa las variables a usarse.
En la pantalla presenta el menú principal de opciones.
Finalmente direcciona al módulo seleccionado o a la terminación del programa.
El.diagrama de flujo del programa se muestra en la figura 3.3

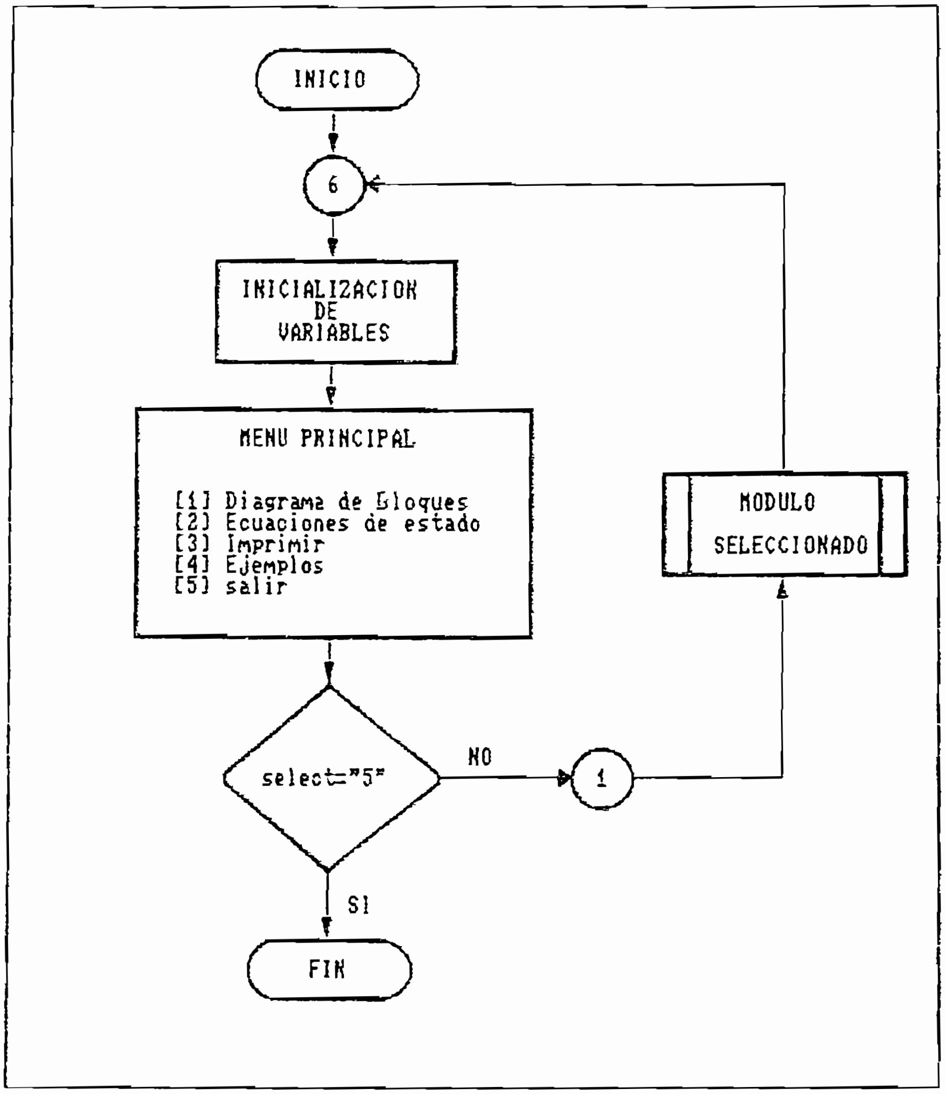


Figura 3.3 Diagrama de flujo del algoritmo del programa TTF.exe

3.3.2 ALGORITMO DEL MODULO FT

El módulo FT permite la obtención de trayectorias de fase para una entrada especifica, y condiciones iniciales conocidas , a partir de la representación del sistema en diagrama de bloques.

Primeramente se requiere los datos de la parte lineal del sistema, su función de transferencia, y los datos de la parte no lineal del sistema, su característica de transferencia.

A continuación se requiere los datos del bloque de compensación, el valor de las constantes Kp, kd y/o ki, los datos de la señal de entrada, escalón y/o rampa y los datos de las condiciones iniciales para el número de trayectorias a dibujarse.

Con estos datos se obtiene por simulación digital los incrementos de las variables de estado y el nuevos estados del sistema, hasta que se alcance un punto de variación mínima, un punto fuera del rango de graficación o un número máximo de puntos de la trayectoria.

Obtenido el retrato de fase con el número de trayectorias indicado, se permite el cambio de rango de visualización del resultado gráfico tantas veces como sea necesario.

Permite, a continuación, el cambio del valor de la entrada y de las condiciones iniciales, así como el cambio de datos del bloque de compensación, para la obtención de nuevas trayectorias para el mismo sistema.

No se requiere especificar el número de puntos, la trayectoria es calculada hasta que esta salga del rango de graficación, alcance un punto de equilibrio o el número de puntos que componen la trayectoria alcancen el límite.

El rango de graficación esta limitado por veinte veces el valor del error inicial. La variación mínima está limitada por la milésima parte del valor del error inicial. El máximo número de puntos se ha escogido de ciento cincuenta y el intervalo de tiempo para la simulación digital entre cada punto de la trayectoria se ha escogido de 0.1 unidades de tiempo. Estos valores pueden ser cambiados con la opción [5] del menú principal.

El diagrama de flujo del módulo FT se presenta en la figura 3.4

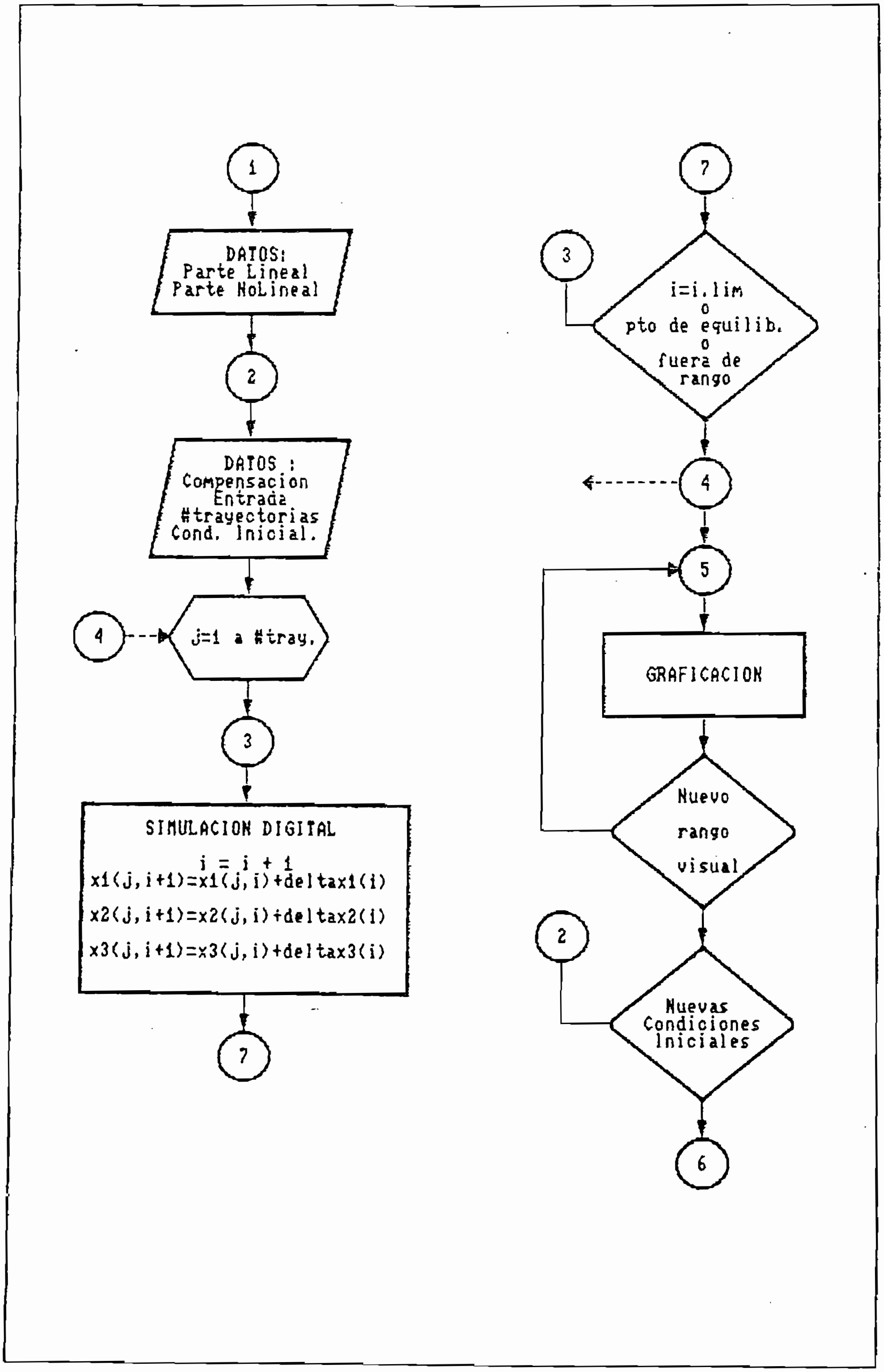


Figura 3.4 Diagrama de flujo del modulo FT

3.3.3 ALGORITMO DEL MODULO ECU

El módulo ECU permite la obtención de un retrato de fase formado por el número de trayectorias deseadas, partiendo de las ecuaciones de estado del sistema y de las condiciones iniciales indicadas al programa .

Se requiere el número de ecuaciones de estado del sistema. Los datos de las ecuaciones son el número de términos, los coeficientes , exponentes de las variables y funciones logaritmicas o exponenciales si las hubiere , en cada término. Y las condiciones o rangos de validez de cada una de las ecuaciones si las tienen.

Se requiere el número de trayectorias y las condiciones iniciales de las variables de estado para cada trayectoria.

Se aplica simulación digital para el calculo de cada nuevo estado del sistema, representado por un punto de la trayectoria de fase.

Obtenido el retrato de fase se permite el cambio de condiciones iniciales para la obtención de un nuevo retrato de fase del mismo sistema.

No se requiere especificar el número de puntos, la trayectoria es calculada hasta que esta salga del rango de graficación, alcance un punto de equilibrio o un punto de variación mínima para la graficación o el número de puntos que componen la trayectoria alcancen el límite.

El nUmero de puntos, el intervalo de tiempo y el número máximo de trayectorias también pueden ser cambiados.

El diagrama de flujo del módulo ECU se indica en la figura 3.5

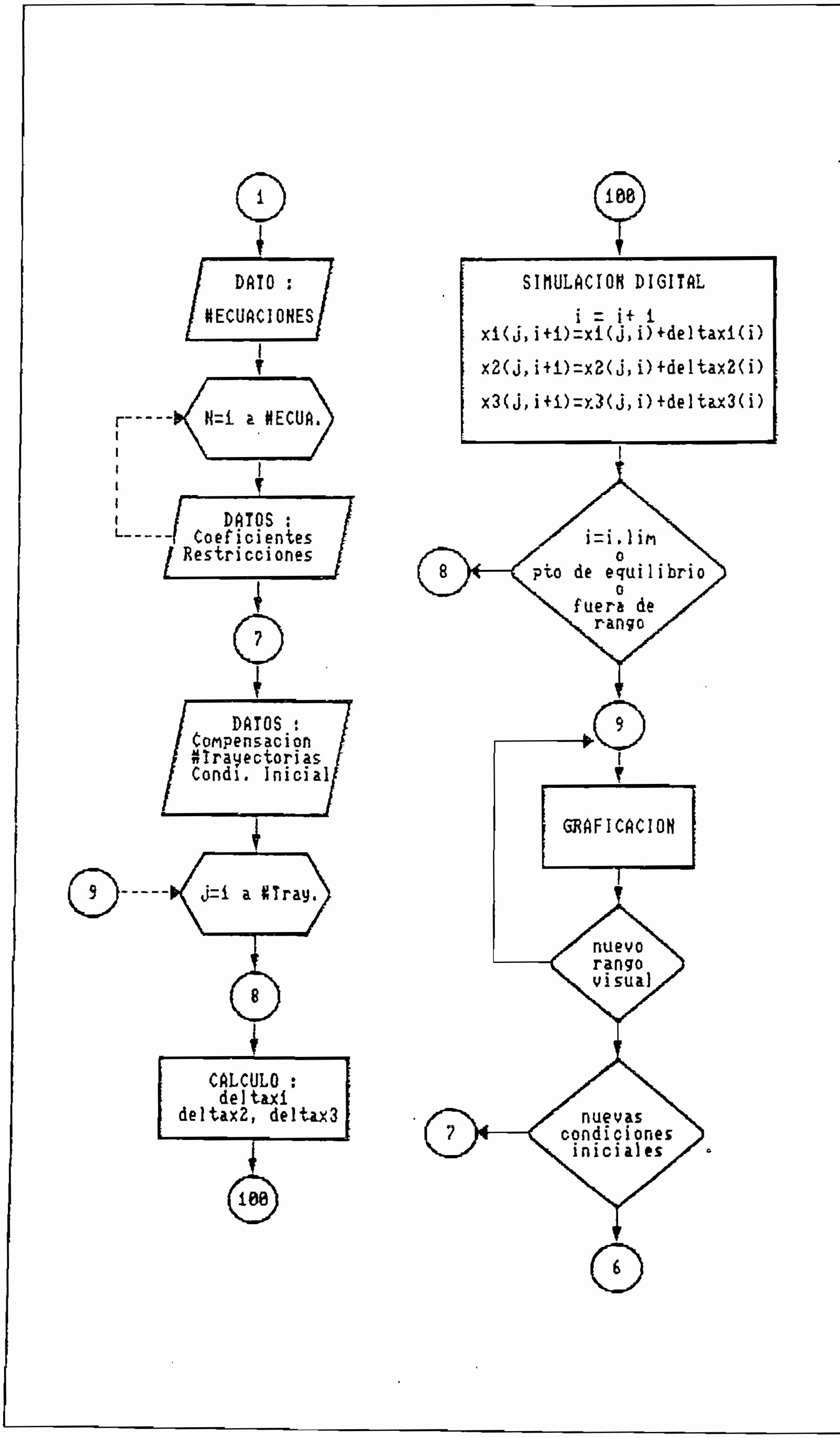


Figura 3.5 Diagrama de flujo del modulo ECU

3.3.4 ALGORITMO DEL MODULO DE IMPRESION

El módulo de impresión permite leer los datos de los puntos de las trayectorias, que previamente se calcularon.

Presenta las trayectorias de fase o la variación en el tiempo de las variables de estado, en la pantalla , o una lista de los puntos inicial , final y máximo de cada trayectoria.

Permite la obtención de estos resultados a través de la impresora. Eu diagrama de flujo del módulo IMPRESION se indica en la figura 3.6.

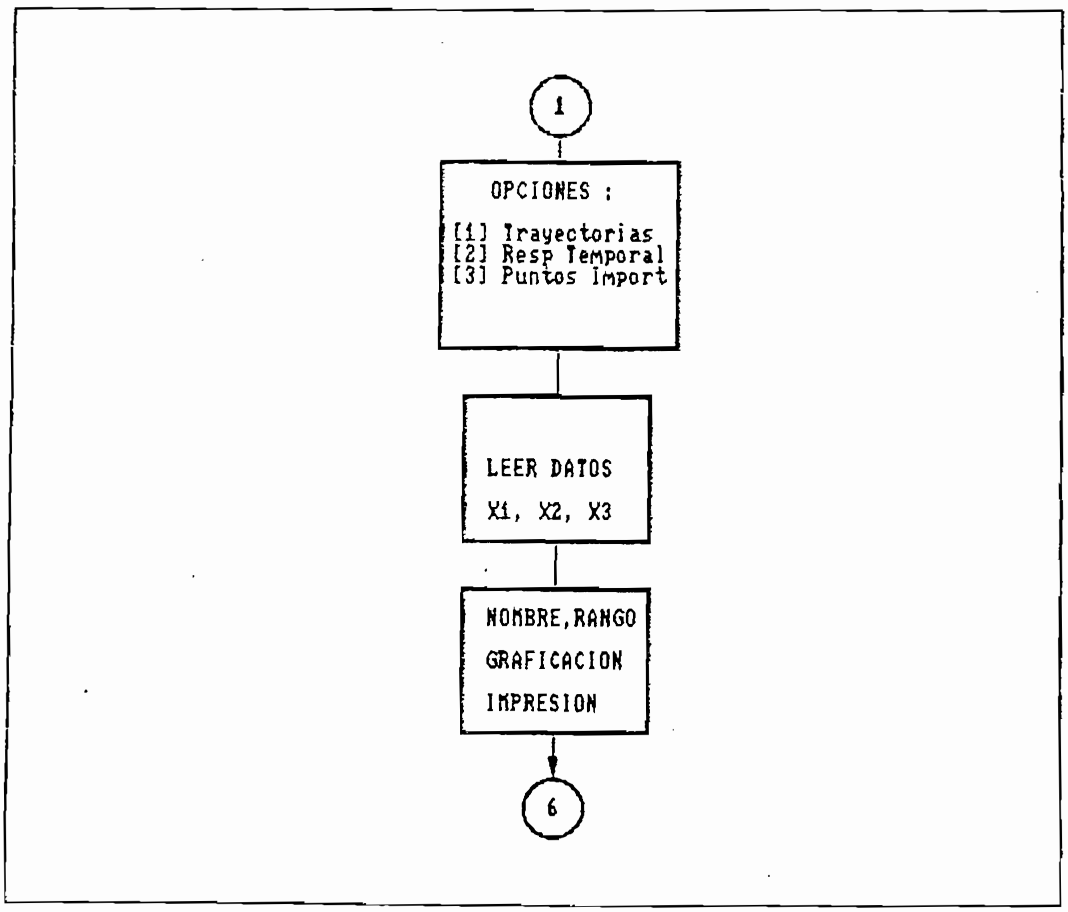


Figura 3.6 Diagrama de flujo del módulo IMPRESION

Con esto queda cubierta la teoría básica, sobre los sistemas de control, sobre la obtención del retrato de fase mediante el programa diseñado y sobre la forma como se diseño este programa para la simulación digital de las trayectorias tanto en el plano como en el espacio. Pudiendo procederse a la presentación en el siguiente capitulo de los resultados obtenidos con el uso de este programa como herramienta para el análisis de los sistemas de control.

CAPITULO IV

APLICACION DEL SOFTWARE

En el capítulo anterior se indicó la forma de trabajo y el fundamento teórico del programa; en este capítulo se presentan finalmente ejemplos de aplicación del software, en los que se prueba la validez de los resultados en comparación con lo teóricamente esperado, así como también la precisión y rapidez en el programa.

Primero se analizan teóricamente los sistemas , mediante sus modelos matemáticos y se determina el tipo de puntos singulares existentes de acuerdo a lo indicado en las tablas del capítulo segundo, y a continuación se presentan los resultados gráficos obtenidos con el programa TTF.

4.1 EJEMPLOS DE PRECISION Y RAPIDEZ EN EL PROGRAMA

4.1.2 SISTEMA LINEAL DE SEGUNDO ORDEN

Considérese un sistema lineal de control sin compensación y con realimentación unitaria, cuya función de transferencia de lazo abierto es de la forma :

$$G(s) = \frac{C(s)}{E(s)} = \frac{b_o}{s(a_2 s + a_1)} \qquad (4.1)$$

donde

$$a_2 = 1$$

$$a_1 = 2\xi w_n$$
$$b_o = w_n^2$$

La función de transferencia en lazo cerrado es, por tanto :

$$\frac{C(s)}{R(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}$$

La ecuación característica del sistema es :

$$a_2 \lambda^2 + a1 \lambda + b_o = 0 \tag{4.2}$$

Y sus raíces son :

$$\begin{aligned} \lambda_1 &= [-a_1 + SQR(a_1\char`^2 - 4a_2b_o)] \;/\; (2a_2) \\ \lambda_2 &= [-a_1 - SQR(a_1\char`^2 - 4a_2b_o)] \;/\; (2a_2) \end{aligned} \tag{4.3}$$

Para determinar la ubicación del punto singular se expresa el sistema mediante sus ecuaciones de estado.

Las relaciones entre las variables, para este sistema, considerando como entrada una señal compuesta, escalón (Resc) y rampa (Rrp), son:

$$e = R_{esc} + R_{rp} * t - c \tag{4.4a}$$

$$c = \frac{b_o}{a_2 s^2 + a_1 s} * e \tag{4.4b}$$

de donde

$$a_2 s^2 c + a_1 s c = b_o e$$

$$a_2 \ddot{c} + a_1 \dot{c} = b_o e \tag{4.5}$$

Si se reemplaza c de la ecuación (4.4a) en la ecuación (4.5) se tiene

$$-a_2 \ddot{e} - a_1 \dot{e} + a_1 R_{rp} = b_o e \tag{4.6}$$

Si se asignan variables de estado al error y su derivada

$$\begin{aligned} x_1 &= e \\ x_2 &= \dot{e} \end{aligned} \tag{4.7}$$

Las ecuaciones de estado que resultan son:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= [-a_1 x_2 - b_o x_1 + a_1 R_{rp}]/a_2 \end{aligned} \tag{4.8}$$

Igualando a cero las ecuaciones (4.8) se puede conocer la ubicación del punto singular.

$$x_2 = 0$$

$$a_1 * x_2 + b_0 * x_1 - a_1 R_{rp} = 0$$

Así el punto singular para este tipo de sistema está ubicado en el punto del plano de fase de coordenadas:

$$x_1 = a_1 R_{rp} / b_0$$
$$x_2 = 0 \qquad (4.9)$$

Con este análisis preliminar, se presentan a continuación ejemplos sobre los casos de estabilidad que pueden haber.

EJEMPLO 1 FOCO ESTABLE

Sea el sistema lineal de segundo orden con realimentación unitaria, cuya función de transferencia tiene valores $a_2 = 1$, $a_1 = 1$, $b_0 = 1$, tal que la función de transferencia en lazo abierto $G(s) = 1/s(s+1)$. Se analiza el sistema con señal escalón unitario y con señal rampa de pendiente unitaria.

El análisis teórico del sistema indica que las raices de la ecuación caracteristica según (4.3) son:

$$\lambda_1 = -.5 + j.866 \qquad \lambda_2 = -.5 + j.866$$

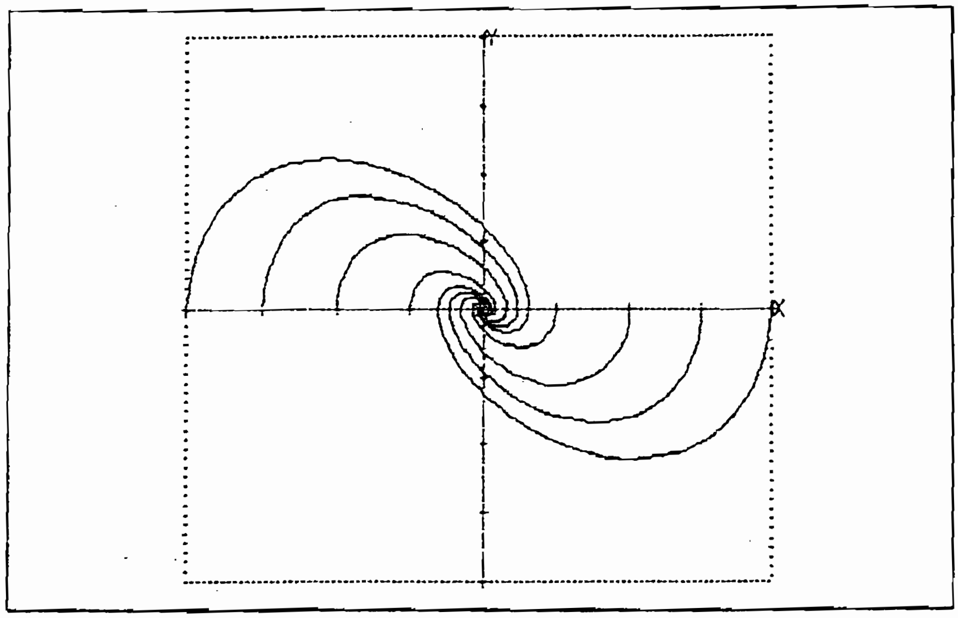
Según la tabla 2.1 la ubicación de las raíces, en el plano complejo, indica la existencia, en el plano de fase, de un punto singular del tipo FOCO ESTABLE.

La ubicación del punto singular según (4.9),en ausencia de señal o con entrada escalón es $x_2 = 0$, $x_1 = 0$, y con señal rampa de pendiente (Rrp) unitaria es $x_2 = 0$, $x_1 = 1$.

Las trayectorias de fase obtenidas para este sistema, con el programa TTF, se muestran en la figura 4.1

TRAYECTORIAS DE FASE

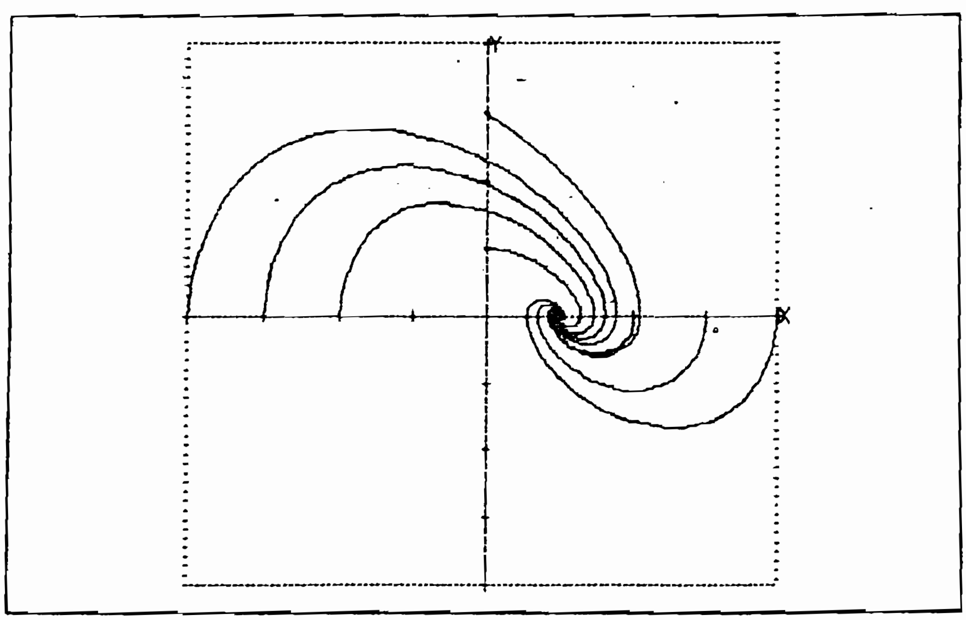
G(s)=1/s(s+1) Entrada ESCALON UNITARIO Pto Sing FOCO ESTABLE



1 division = 1 unidades

TRAYECTORIAS DE FASE

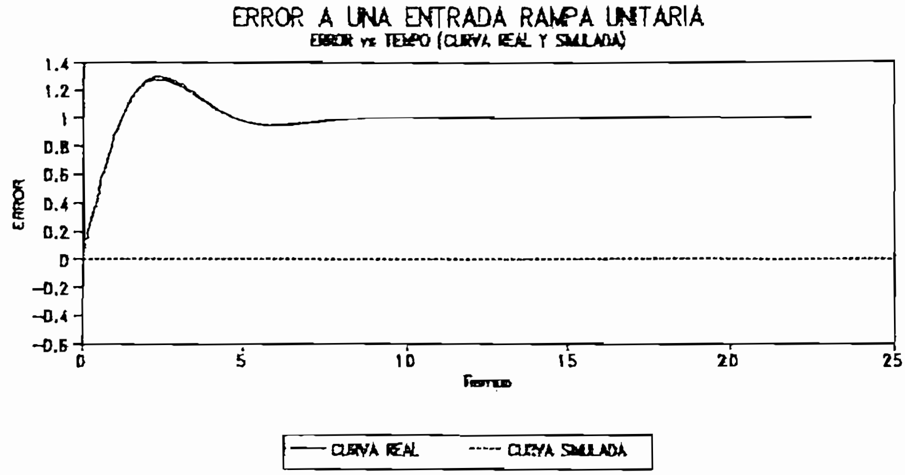
G(s)=1/s(s+1) Entrada RAMPA UNITARIA Pto Sing FOCO ESTABLE

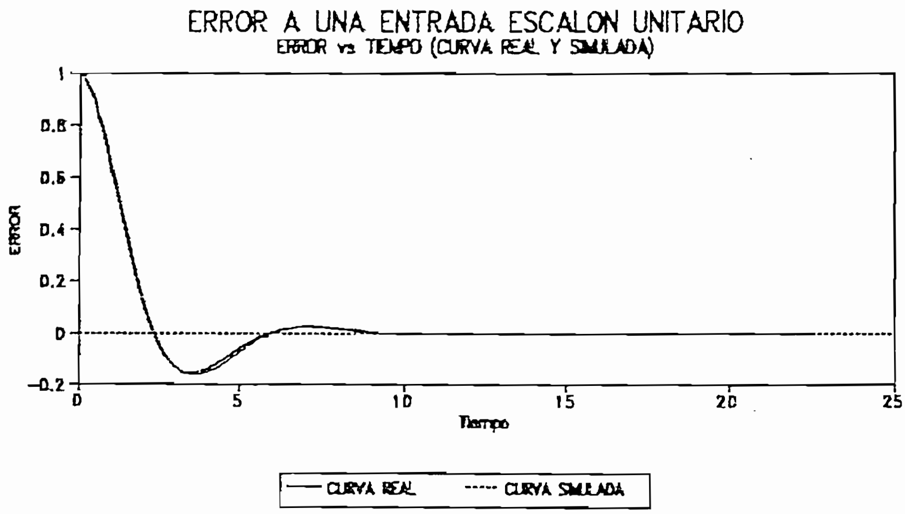


1 division = 1 unidades

Figura 4.1 Retratos de fase correspondientes al sistema del ejemplo 1

Figura 4.2 Respuesta en el tiempo correspondientes al sistema del ejemplo 1





	ENTRADA ESCALON			ENTRADA RAMPA		
TIEMPO	REAL	APROX	error	REAL	APROX	error
2.0	0.1506	0.1178	−0.03275	1.2687	1.2510	−0.0177
2.1	0.1100	0.0798	−0.03025	1.2817	1.2628	−0.0189
2.2	0.0723	0.0447	−0.02756	1.2908	1.2708	−0.0200
2.3	0.0374	0.0127	−0.02471	1.2963	1.2752	−0.0210
2.4	0.0055	−0.0162	−0.02175	1.2984	1.2765	−0.0219
2.5	−0.0234	−0.0421	−0.01872	1.2975	1.2749	−0.0226
2.6	−0.0493	−0.0649	−0.01565	1.2938	1.2707	−0.0231
2.7	−0.0723	−0.0849	−0.01258	1.2877	1.2642	−0.0235
2.8	−0.0924	−0.1019	−0.00955	1.2795	1.2557	−0.0238
2.9	−0.1097	−0.1163	−0.00658	1.2693	1.2455	−0.0238
3.0	−0.1244	−0.1281	−0.0037	1.2576	1.2339	−0.0237
3.1	−0.1364	−0.1374	−0.00095	1.2445	1.2211	−0.0235
3.2	−0.1460	−0.1444	0.001668	1.2304	1.2073	−0.0231
3.3	−0.1533	−0.1492	0.004124	1.2154	1.1929	−0.0225
3.4	−0.1585	−0.1521	0.006403	1.1998	1.1780	−0.0218

La validez del programa se muestra por la correspondencia del retrato de fase graficado y el tipo de punto singular y su ubicación teórica esperados.

El programa permite la obtención gráfica en la pantalla en un tiempo de 1 minuto y 41 segundos.

La precisión del gráfico está basada en el cálculo de cada nuevo estado del sistema usando un incremento de tiempo de 0.1 unidades de tiempo para el proceso de simulación digital.

La figura 4.2 muestra los gráficos de la respuesta temporal $e(t)$ obtenidos a partir de las expresiones reales (4.10) y (4.11), para entrada escalón y rampa respectivamente, y de los puntos calculados por el programa. Las condiciones iniciales son $e(0)=1$, $e(0)=0$ para entrada escalón unitario, y $e(0)=0$, $e(0)=1$ para entrada rampa unitaria.

$$e(t)= \exp(-t/2)*(\cos(0.866t) + \mathrm{sen}(0.8866t)/\mathrm{sqr}(3)) \qquad (4.10)$$

$$e(t)= 1-\exp(-t/2)*(\cos(0.866t) - \mathrm{sen}(0.8866t)/\mathrm{sqr}(3)) \qquad (4.10)$$

Se presentan también los datos correspondientes a los puntos obtenidos mediante el programa y los valores obtenidos a partir de la expresión de la solución real.

El error que se comete en la simulación es menor en ambos casos al 0.05 , lo que gráficamente es poco notorio y no influye en un criterio de análisis gráfico que pueda darse respecto al sistema, su estabilidad , rapidez o precisión.

EJEMPLO 2. CENTRO

Sea el sistema lineal de segundo orden con realimentación unitaria, cuya función de transferencia en lazo abierto es

$$G(s) = 1/s_2$$

El análisis teórico del sistema indica que las raices de la ecuación característica según (4.3) son:

$$\lambda_1 = +j \qquad \lambda_2 = -j$$

Según la tabla 2.1 la ubicación de las raíces, en el plano complejo, indica la existencia, en el plano de fase, de un punto singular del tipo CENTRO.

La ubicación del punto singular según (4.9) es $x_2 = 0$, $x_1 = 0$ para entrada escalón y para señal rampa unitaria.

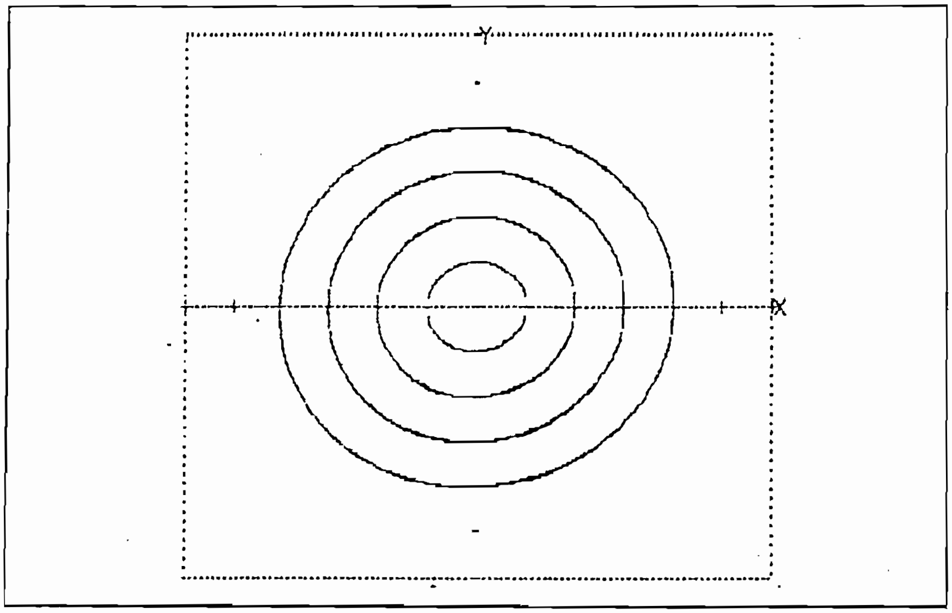
La rapidez del programa permite la obtención gráfica de las trayectorias de fase, en la pantalla, en un tiempo de 2 minutos y 5 segundos.

La precisión del programa está basada en el cálculo de cada nuevo estado del sistema usando un incremento de tiempo de 0.05 unidades de tiempo para el proceso de simulación digital.

Las trayectorias de fase obtenidas para este sistema, con el programa TTF, se muestran en la figura 4.3.

TRAYECTORIAS DE FASE

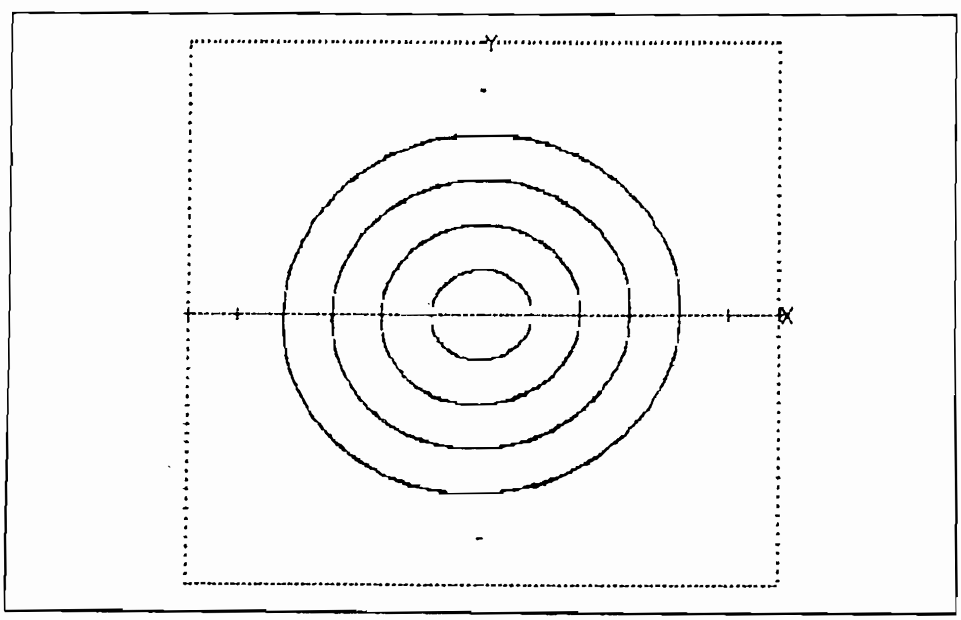
G(s) = 1/s^2   Entrada ESCALON UNITARIO   Pto Sing CENTRO



1 division = 1 unidades

TRAYECTORIAS DE FASE

G(s) = 1 / s^2   Entrada RAMPA UNITARIA   Pto Sing CENTRO



1 division = 1 unidades

Figura 4.3 Retratos de fase correspondientes al sistema del ejemplo 2

La correspondencia del retrato de fase graficado, el tipo de punto singular y su ubicación teórica esperados, tomando como variables de estado al error y su derivada , muestran la validez del programa como herramienta para el trazo de trayectorias de fase.

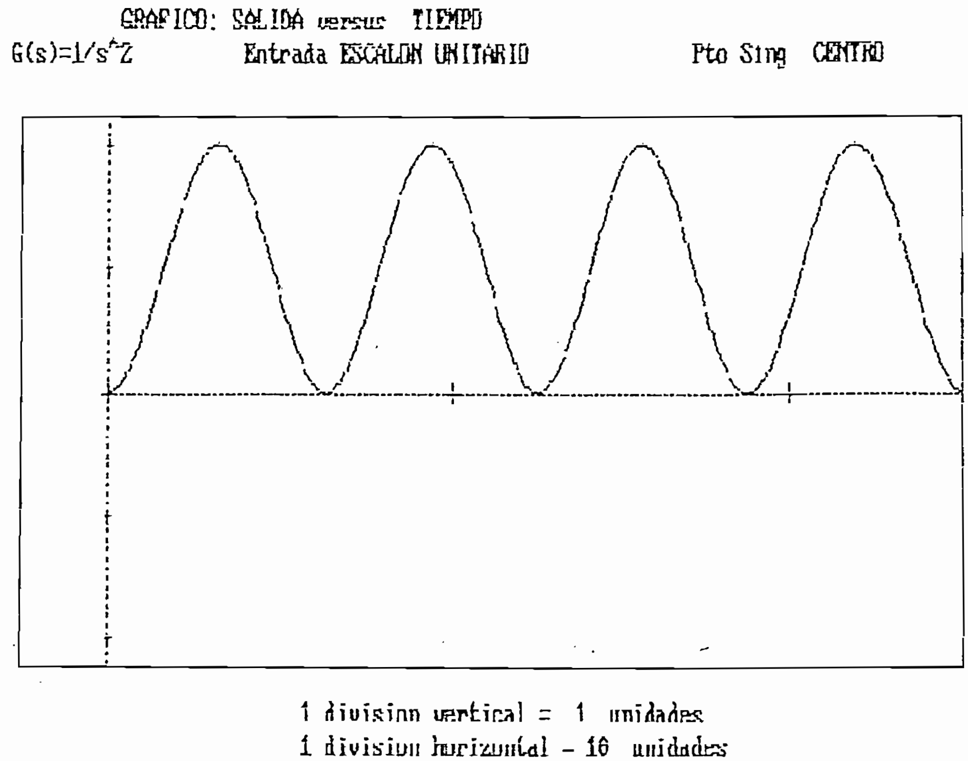
Es importante notar la ayuda que presta el programa al entregar también los gráficos de las respuestas temporales, como apoyo para el análisis de los sistemas de control.

La figura 4.4 presenta el comportamiento de las variables de estado del sistema en el tiempo para una de las trayectorias de fase .

Es importante también la posibilidad que presta el programa de grabar los datos y poder recuperarlos mediante cualquier hoja de cálculo. Esto permite relizar análisis o trabajos adicionales como la figura 4.5 que muestra el error de cálculo existente en la simulación.

Como se aprecia gráficamente el error de cálculo es inferior a 0.03 para entrada escalón, e inferior a 0.003 para entrada rampa. Esto significa que prácticamente la respuesta aproximada se superpone a la respuesta real.

Figura 4.4 Respuesta en el tiempo correspondientes al sistema del ejemplo 2



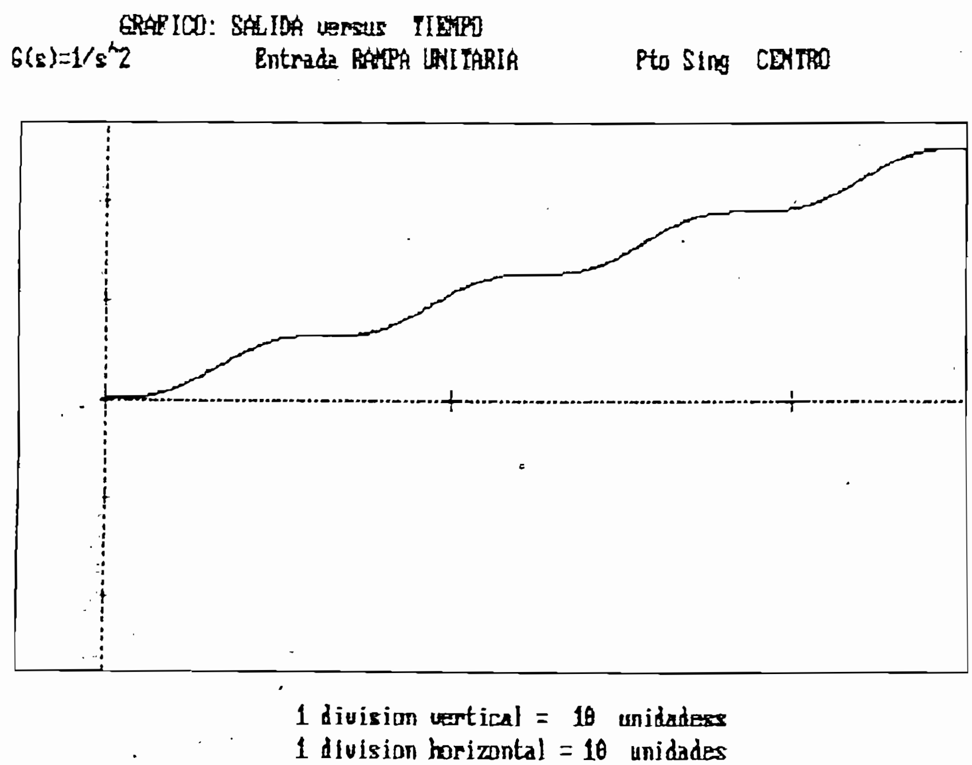
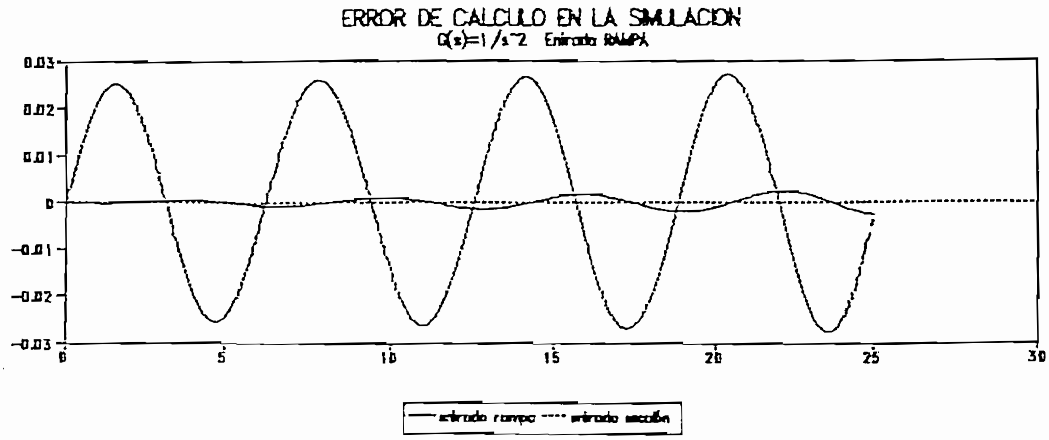


Figura 4.5 Error de cálculo existente en la simulación



	ENTRADA RAMPA UNITARIA			ENTRADA ESCALON UNITARIO		
	aprox	real		aprox	real	
tiempo	x1=e(t)	x1=sen(t)	error	x1=e(t)	x1=cos(t)	error
23.5000	-0.9985	-0.9981	0.0005	-0.0345	-0.0619	-0.0274
23.5500	-1.0003	-0.9999	0.0003	0.0155	-0.0119	-0.0275
23.6000	-0.9995	-0.9993	0.0002	0.0655	0.0380	-0.0274
23.6500	-0.9962	-0.9961	0.0001	0.1153	0.0879	-0.0274
23.7000	-0.9905	-0.9905	-0.0000	0.1648	0.1376	-0.0272
23.7500	-0.9822	-0.9824	-0.0002	0.2139	0.1869	-0.0270
23.8000	-0.9715	-0.9718	-0.0003	0.2625	0.2358	-0.0267
23.8500	-0.9584	-0.9588	-0.0004	0.3104	0.2841	-0.0263
23.9000	-0.9429	-0.9434	-0.0005	0.3576	0.3317	-0.0259
23.9500	-0.9250	-0.9256	-0.0007	0.4038	0.3784	-0.0254
24.0000	-0.9048	-0.9056	-0.0008	0.4491	0.4242	-0.0249
24.0500	-0.8823	-0.8832	-0.0009	0.4932	0.4689	-0.0243
24.1000	-0.8577	-0.8587	-0.0010	0.5361	0.5125	-0.0236
24.1500	-0.8309	-0.8320	-0.0011	0.5776	0.5547	-0.0229
24.2000	-0.8020	-0.8033	-0.0013	0.6177	0.5956	-0.0221
24.2500	-0.7711	-0.7725	-0.0014	0.6563	0.6350	-0.0212
24.3000	-0.7383	-0.7398	-0.0015	0.6932	0.6729	-0.0203
24.3500	-0.7036	-0.7052	-0.0016	0.7284	0.7090	-0.0194
24.4000	-0.6672	-0.6689	-0.0017	0.7617	0.7433	-0.0184
24.4500	-0.6291	-0.6309	-0.0018	0.7932	0.7758	-0.0173
24.5000	-0.5895	-0.5914	-0.0019	0.8227	0.8064	-0.0162
24.5500	-0.5483	-0.5503	-0.0020	0.8501	0.8350	-0.0151
24.6000	-0.5058	-0.5079	-0.0021	0.8754	0.8614	-0.0139
24.6500	-0.4621	-0.4642	-0.0021	0.8985	0.8857	-0.0127
24.7000	-0.4172	-0.4194	-0.0022	0.9193	0.9078	-0.0115
24.7500	-0.3712	-0.3735	-0.0023	0.9379	0.9276	-0.0102
24.8000	-0.3243	-0.3266	-0.0023	0.9541	0.9452	-0.0089
24.8500	-0.2766	-0.2790	-0.0024	0.9679	0.9603	-0.0076
24.9000	-0.2282	-0.2306	-0.0025	0.9793	0.9730	-0.0063
24.9500	-0.1792	-0.1817	-0.0025	0.9883	0.9833	-0.0050
25.0000	-0.1298	-0.1324	-0.0025	0.9948	0.9912	-0.0036

EJEMPLO 3 NODO ESTABLE

Sea el sistema lineal de segundo orden con realimentación unitaria, cuya función de transferencia e lazo abierto es

$$G(s) = 1 / s(s + 4)$$

el análisis teórico del sistema indica que las raíces de la ecuación característica según (4.3) son:

$$\lambda_1 = -3.73 \qquad \lambda_2 = -0.27$$

Según la tabla 2.1 la ubicación de las raíces, en el plano complejo, indica la existencia, en el plano de fase, de un punto singular del tipo NODO ESTABLE.

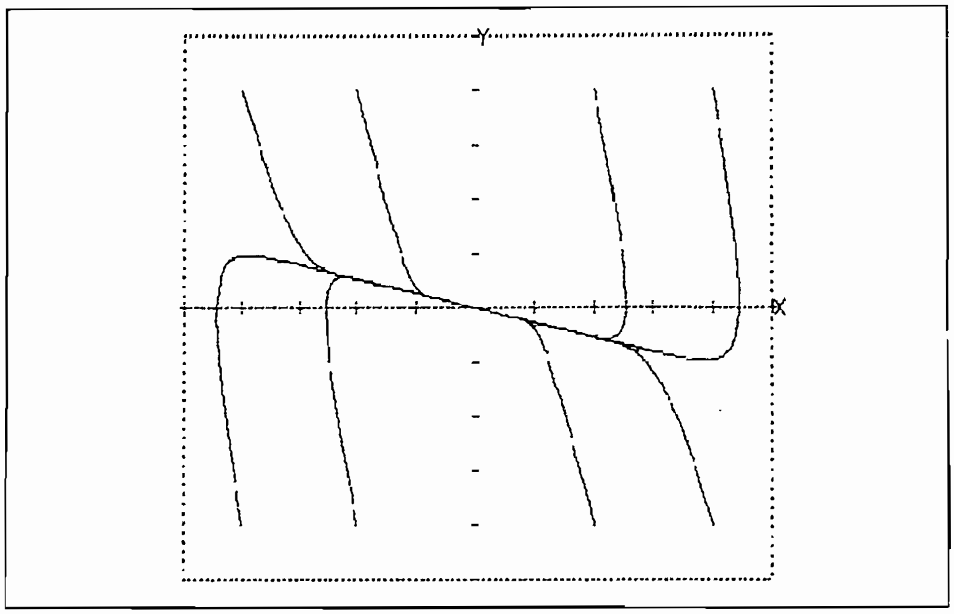
La ubicación del punto singular según (4.9) es $x_2 = 0$, $x_1 = 0$ para entrada escalón y $x_2 = 0$, $x_1 = 4$ para entrada rampa unitaria.

El programa permite la obtención gráfica en la pantalla en un tiempo de 1 minuto y 21 segundos.

Las trayectórias de fase obtenidas para este sistema, con el programa TTF, se muestran en la figura 4.6

TRAYECTORIAS DE FASE

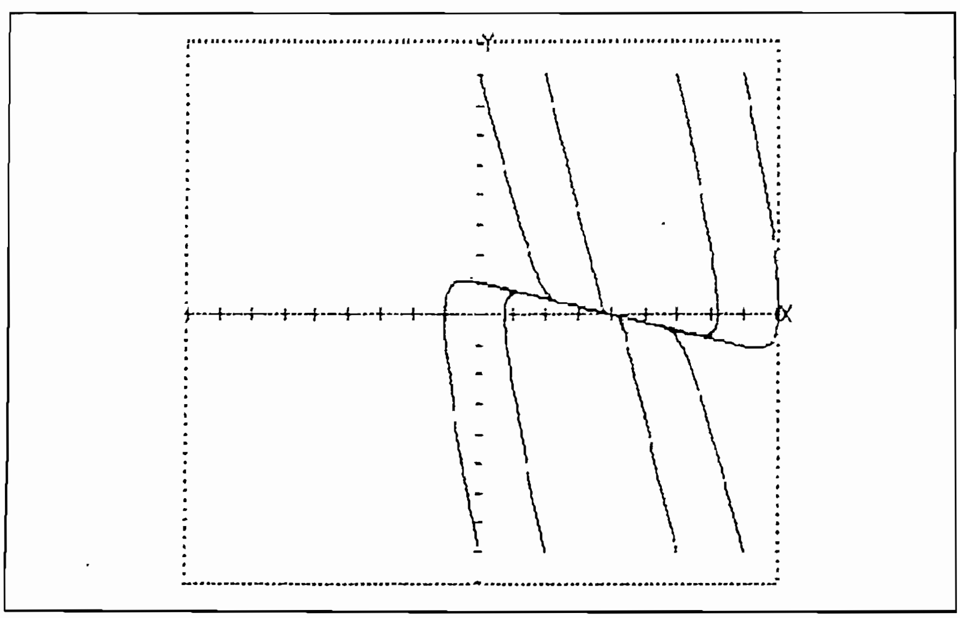
G(s)=1/s(s+4)   Entrada ESCALON UNITARIO   Pto Sing NODO ESTABLE



1 division = 1 unidades

TRAYECTORIAS DE FASE

G(s)=1/s(s+4)   Entrada RAMPA UNITARIA   Pto Sing NODO ESTABLE



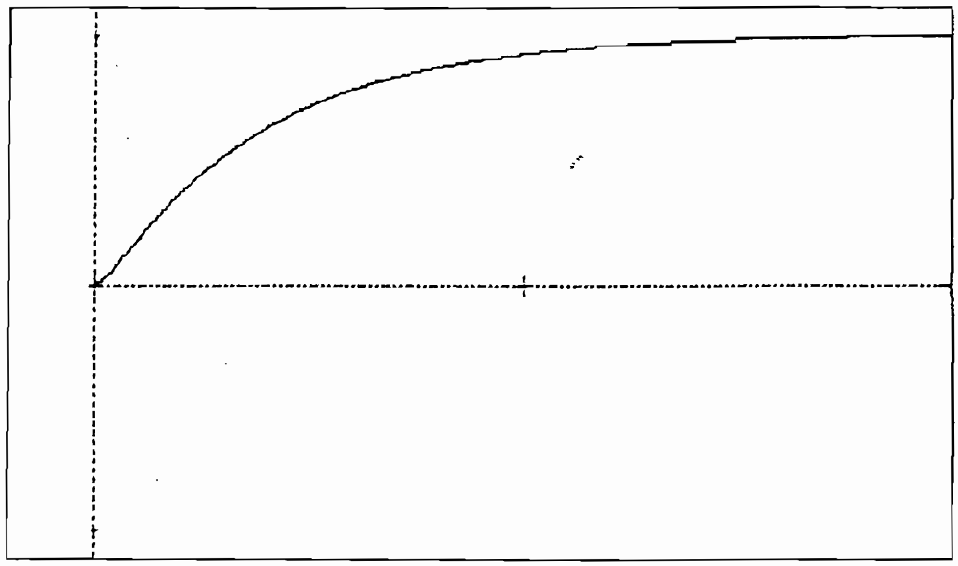
1 division = 1 unidades

Figura 4.6 Retratos de fase correspondientes al sistema del ejemplo 3

Figura 4.7 Respuesta en el tiempo correspondientes al sistema del ejemplo 3

GRAFICO: SALIDA versus TIEMPO

G(s)=1/s(s+4) Entrada ESCALON UNITARIO Pto Sing NODO ESTABLE

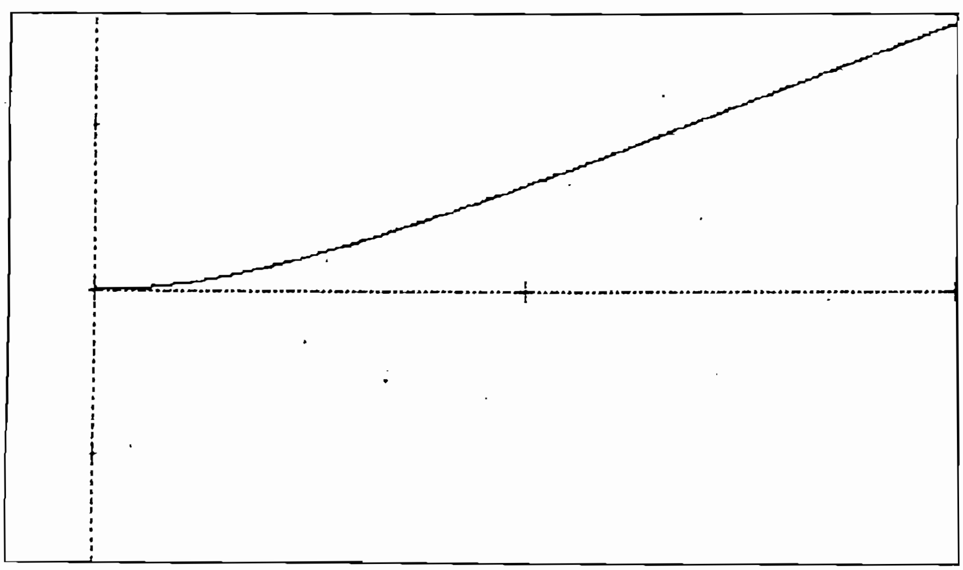


1 division vertical = 1 unidades

1 division horizontal = 10 unidades _

GRAFICO: SALIDA versus TIEMPO

G(s)=1/s(s+4) Entrada RAMPA UNITARIA Pto Sing NODO ESTABLE



1 division vertical = 10 unidades

1 division horizontal = 10 unidades _

La validez del programa se muestra por la correspondencia entre el retrato de fase graficado y el tipo de punto singular y su ubicación teórica esperados.

La figura 4.7 presenta el comportamiento de la salida del sistema en el tiempo para las condiciones iniciales $e(0)=1$, $e(0)=0$ con entrada escalón unitario, y condiciones iniciales $e(0)=0$, $e(0)=1$ con entrada rampa unitaria.

La precisión del resultado se fundamenta en el cálculo de cada nuevo estado del sistema usando un incremento de tiempo de 0.01 unidades de tiempo para el proceso de simulación digital.

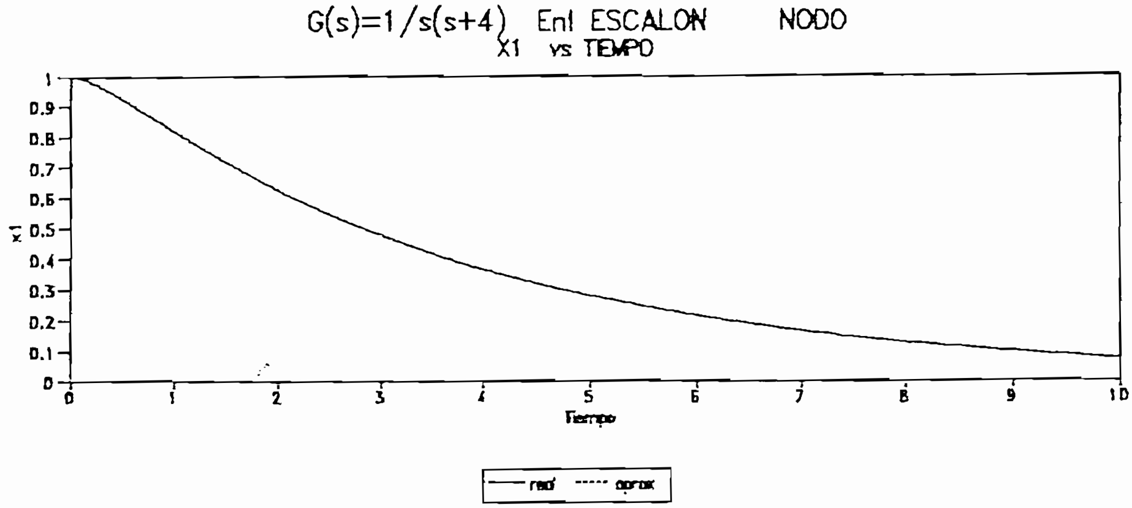
En la figura 4.8 se aprecia el error de cálculo existente en la simulación digital para el cálculo de x_1.

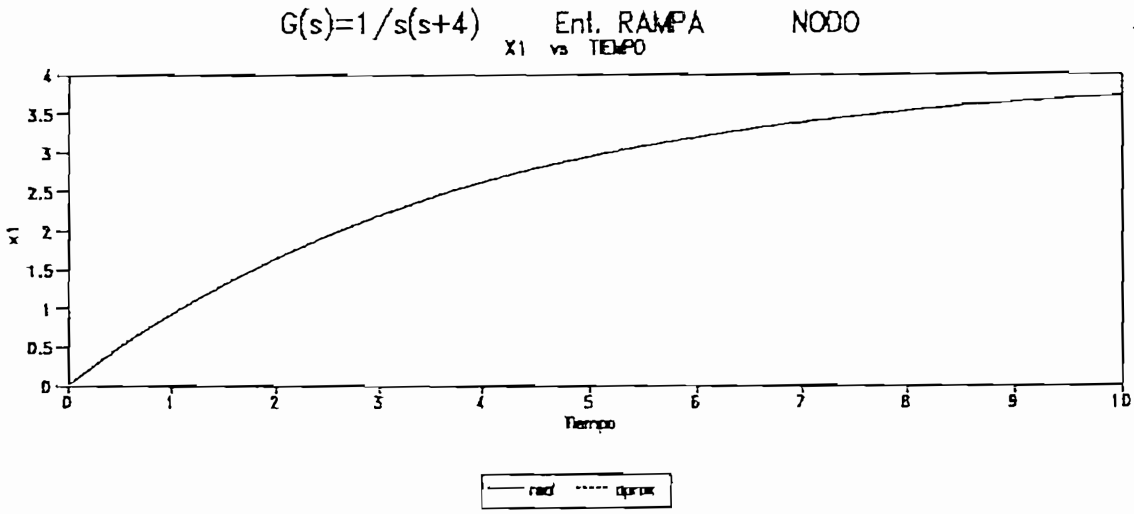
Para la obtención de este gráfico se usó los valores de x_1 (error del sistema de control), grabados con el TTF, y los valores obtenidos con las expresiones exactas del error, (4.12) y (4.13) para entrada escalón y rampa unitaria respectivamente.

$$e(t) = -\,0.077*\exp(-3.73*t) + 1.077*\exp(-0.27*t) \qquad (4.12)$$

$$\begin{aligned} e(t) = 4 &+ [(7-4*sqr(3))/(2*sqr(3)]*\exp(-(2+sqr(3)*t) \\ &- [(7-4*sqr(3))/(2sqr(3)]*\exp(-(2-sqr(3)*t) \qquad (4.13) \end{aligned}$$

Figura 4.8 Error en el cálculo de x_1 por simulación digital.





	ENTRADA ESCALON UNITARIO			ENTRADA RAMPA UNITARIA		
	real	aprox		real	aprox	
tiempo	x1	x1	error	x1	x1	error
0	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000
1	0.820312	0.820257	0.000055	0.924855	0.924255	0.000601
2	0.627576	0.628925	-0.001349	1.647311	1.645922	0.001389
3	0.479111	0.481319	-0.002208	2.200309	2.198504	0.001805
4	0.365744	0.368337	-0.002592	2.623333	2.621381	0.001952
5	0.279202	0.281874	-0.002673	2.946923	2.944994	0.001929
6	0.213137	0.215708	-0.002571	3.194452	3.192643	0.001809
7	0.162704	0.165074	-0.002369	3.383799	3.382159	0.001640
8	0.124205	0.126325	-0.002119	3.528639	3.527190	0.001449
9	0.094816	0.096672	-0.001856	3.639434	3.638176	0.001258
10	0.072380	0.073979	-0.001599	3.724186	3.723109	0.001077

Se observa que el resultado gráfico obtenido a partir del programa es el mismo que el obtenido a partir de la expresión real de la respuesta del sistema dada por (4.12) y (4.13), debido a que el mayor error introducido , menor a 0.003, no es observable en la escala del gráfico.

Como se indicó anteriormente el valor usado para el intervalo de tiempo puede ser cambiado en la opción CAMBIOS EN EL PROGRAMA . El valor máximo aceptado es 0.1 , con ello se asegura una precisión razonable en el trazo de las trayectorias de fase , como lo demuestra el ejemplo 1. No existe limite para el valor mínimo , de tal manera que la precisión en cálculo de cada nuevo estado del sistema sea tan buena como se requiera , siendo gráficamente suficiente y recomendable el valor de 0.01 como lo muestra el ejemplo 3. Debe recordarse que con un valor menor del intervalo de tiempo, para la simulación digital, se requiere un número mayor de puntos para obtener gráficos que muestren el comportamiento del sistema hasta que este alcanza un estado permanente.

EJEMPLO 4 FOCO INESTABLE

Sea el sistema lineal de segundo orden con realimentación unitaria, cuya función de transferencia en lazo abierto es

$$G(s)= 1/s(s-1)$$

Las raíces de la ecuación característica según (4.3) son:

$$\lambda_1 = .5 +.866j \qquad \lambda_2 = .5 +.866j$$

Según la tabla 2.1 la ubicación de las raíces indica que el punto singular es un FOCO INESTABLE.

La ubicación del punto singular según (4.9) es $x_2 = 0$, $x_1 = 0$ para entrada escalón unitario , y x2=0, x1=-1 para entrada rampa unitaria.

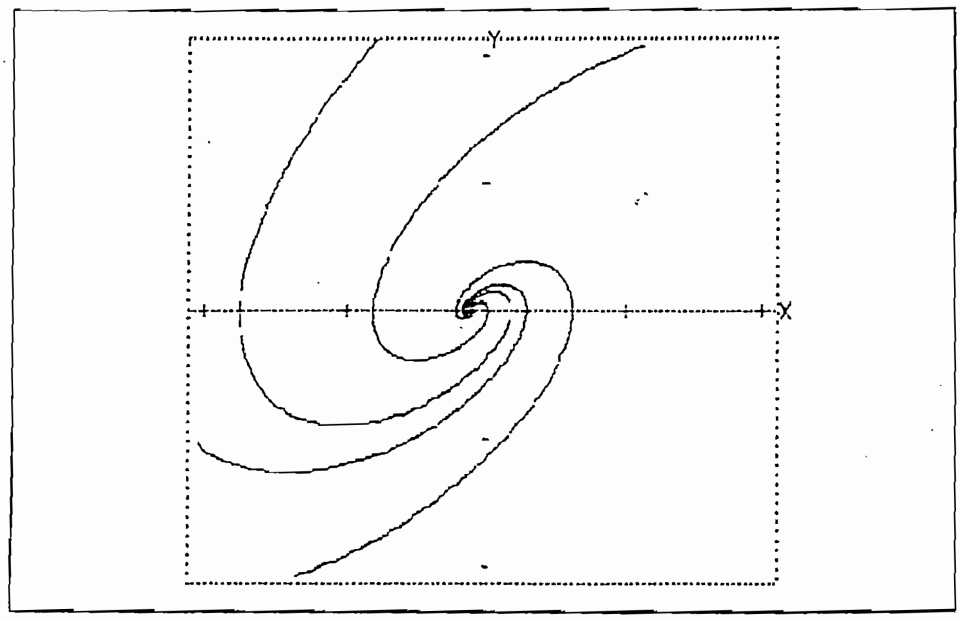
El programa permite la obtención gráfica en la pantalla en un tiempo de 1 minuto y 48 segundos

El cálculo de los puntos que forman las trayectorias se realiza hasta que el valor absoluto del nuevo estado del sistema sea mayor a veinte veces el valor de estado inicial dado, lo que da un rango razonable de graficación y permite decir con certeza si una trayectoria es o no divergente.

La precisión del gráfico en sistemas inestable puede disminuirse favoreciendo la velocidad de graficación de un mayor número de trayectorias utilizando un incremento de tiempo de 0.1 .

TRAYECTORIAS DE FASE

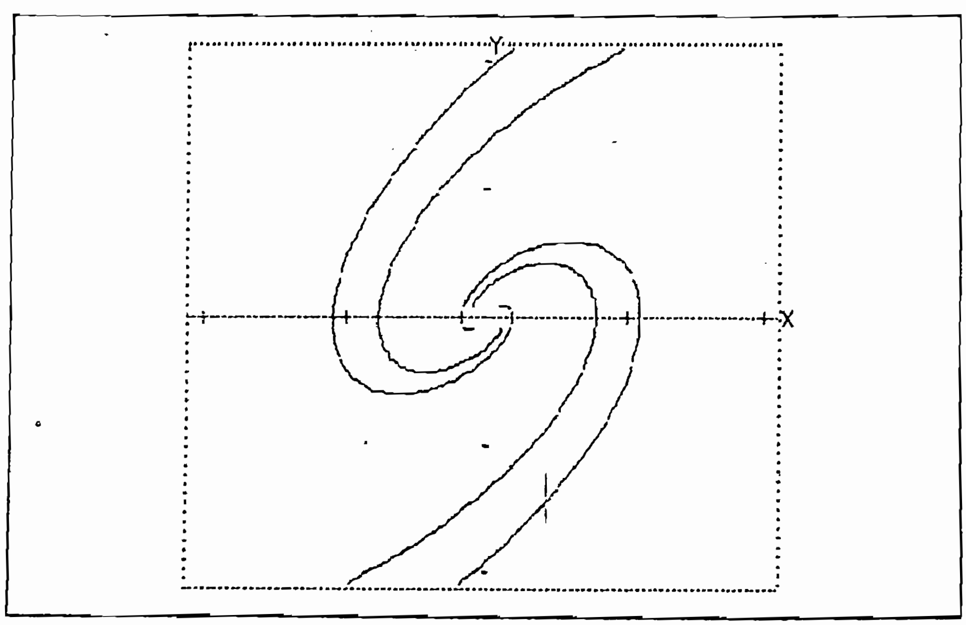
G(s)= 1/s(s-1) Entrada RAMPA UNITARIA Pto Sing FOCO INESTABLE



1 division = 10 unidades

TRAYECTORIAS DE FASE

G(s)=1/s(s-1) Entrada ESCALON UNITARIO Pto Sing FOCO INESTABLE



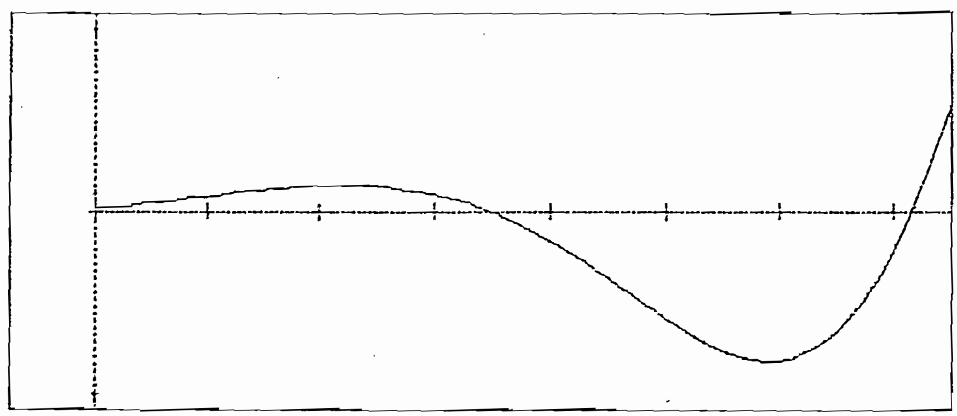
1 division = 1 unidades

Figura 4.9 Retratos de fase correspondientes al sistema del ejemplo 4

Figura 4.10 Respuesta en el tiempo correspondientes al sistema del ejemplo 4

RESPUESTA DE X1 EN EL TIEMPO

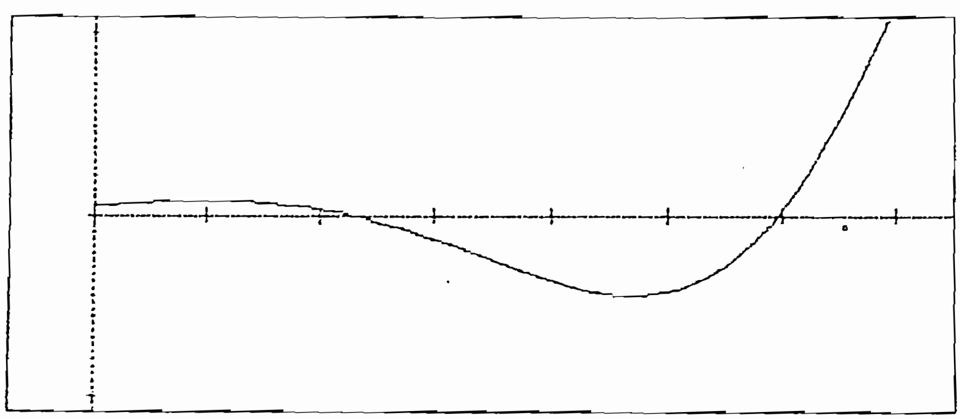
EJEMPLO4: G(s)=1/(s^2 -s +1) PTO. SING.= FOCO INESTABLE



1 division vertical = 18 unidades
1 division horizontal = 1 unidades

RESPUESTA DE X2 EN EL TIEMPO

EJEMPLO4: G(s)=1/(s^2 -s +1) PTO. SING.= FOCO INESTABLE



1 division vertical = 18 unidades
1 division horizontal = 1 unidades

Las trayectorias de fase obtenidas para este sistema, con el programa TTF, se muestran en la figura 4.9

La figura 4.10 presenta el comportamiento de las variables de estado, x1 y x2, del sistema, en el tiempo para una de las condiciones iniciales dadas en ausencia de señal.

La validez del programa se muestra por el retrato de fase obtenido y el tipo de punto singular y su ubicación teórica esperados.

EJEMPLO 5 NODO INESTABLE

Sea el sistema lineal de segundo orden con realimentación unitaria, cuya función de transferencia de lazo abierto es

$$G(s)= 1/s(s-2)$$

Las raices de la ecuación característica según (4.3) son:

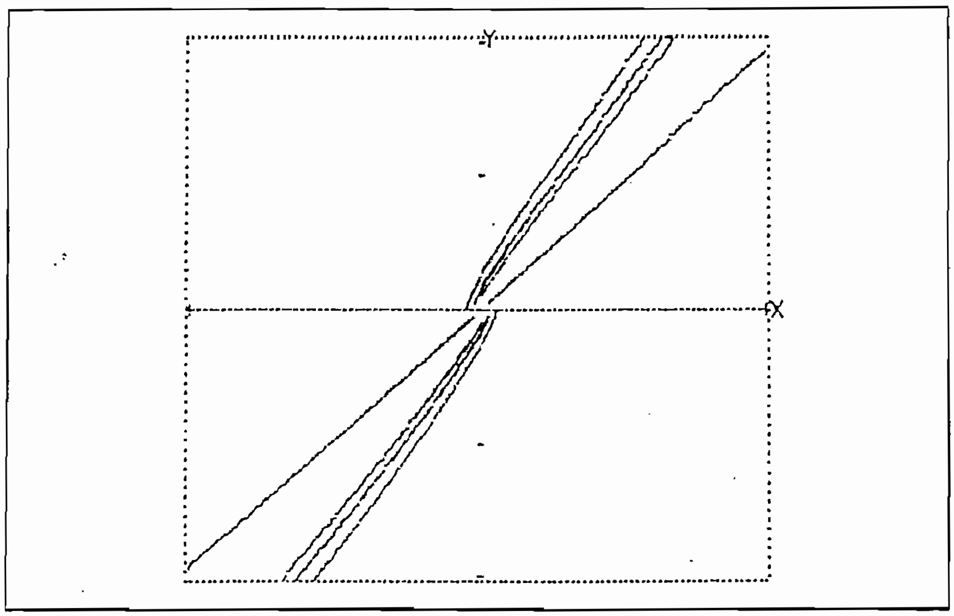
$$\lambda_1 = +1 \qquad\qquad \lambda_2 = +1$$

La ubicación de las raíces, en el plano complejo, indica la existencia, en el plano de fase, de un punto singular del tipo NODO INESTABLE.

La ubicación del punto singular según (4.9) es $x_2 = 0$, $x_1 = 0$ para entrada escalón unitario, y x2 = 0, x1 = -10 para una entrada rampa de pendiente 5.

TRAYECTORIAS DE FASE

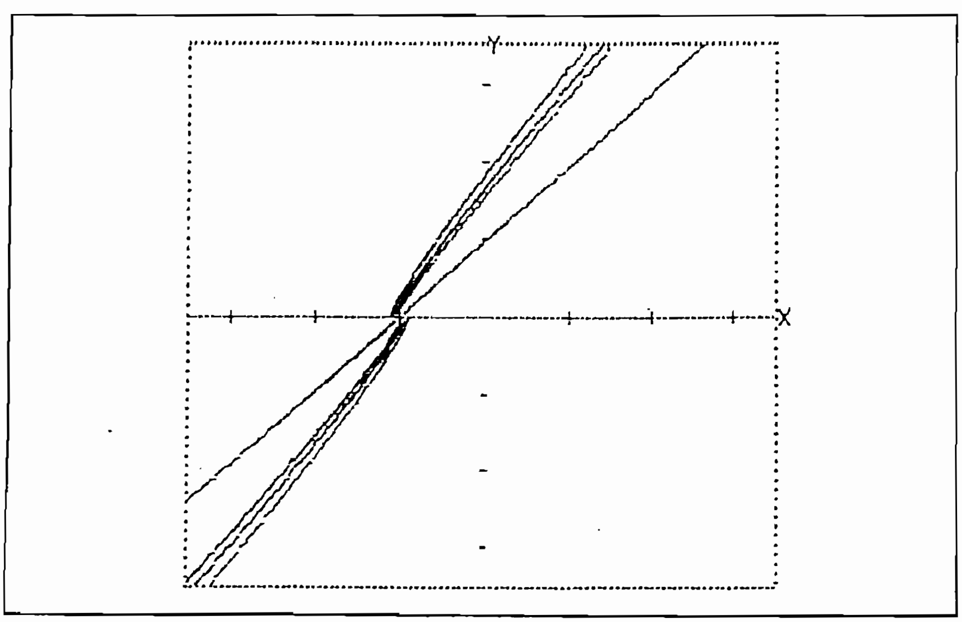
G(s) =1/s(s-2)   Entrada ESCALON UNITARIO   Pto Sing NODO INESTABLE



1 division = 10 unidades

TRAYECTORIAS DE FASE

G(s)=1/s(s-2)   Entrada RAMPA Pendiente=5   Pto Sing NODO INESTABLE



1 division = 10 unidades

Figura 4.11   Retratos de fase correspondientes al sistema del ejemplo 5

El programa permite la obtención gráfica en la pantalla en un tiempo de 2 minutos y 48 segundos.

El cálculo de los puntos que forman las trayectorias se realiza hasta que el valor absoluto del estado del sistema sea mayor que veinte veces el estado inicial del sistema, lo que permite asegurar el carácter divergente de la trayectoria trazada.

La validez del programa se muestra por la correspondencia del retrato de fase graficado y tipo de punto singular y su ubicación teórica esperados.

La figura 4.11 presenta el comportamiento de las variables de estado del sistema en el tiempo para una de las condiciones iniciales dadas. Como se indicó anteriormente el cálculo permite obtener gráficos que muestran el comportamiento del sistema hasta que este alcanza un estado permanente.

EJEMPLO 6 SILLA DE MONTAR

Sea el sistema lineal de segundo orden con realimentación unitaria, cuya función de transferencia de lazo abierto es

G(s= -1/s^2

Las raices de la ecuación característica según (4.3) son:

$$\lambda_1 = +1 \qquad\qquad \lambda_2 = -1$$

La ubicación de las raices, en el plano complejo, indica la existencia, en el plano de fase, de un punto singular del tipo SILLA

La ubicación del punto singular según (4.9) es:

$x_2 = 0$, $x_1 = 0$ independiente del tipo de señal de entrada que se use.

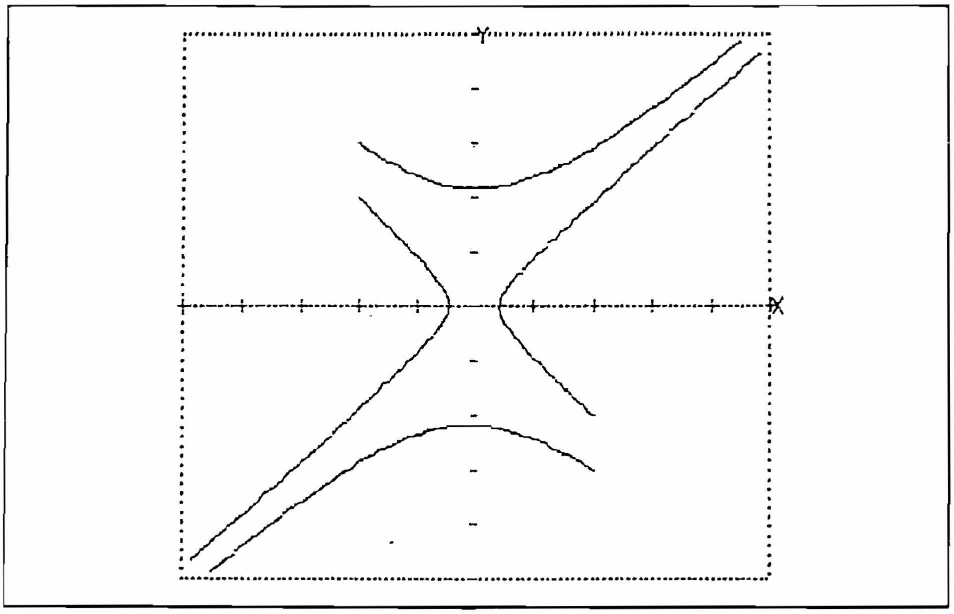
El programa permite la obtención gráfica en la pantalla en un tiempo de 2 minutos y 6 segundos.

La precisión del gráfico está basada en el cálculo de cada nuevo estado del sistema usando un incremento de tiempo de 0.1 unidades de tiempo para el proceso de simulación digital. Las trayectorias de fase obtenidas para este sistema, con el programa TTF, se muestran en la fig.4.12. En la fig. 4,13 se presenta el comportamiento de las variables de estado en el tiempo.

Se observa similitud entre el gráfico obtenido para entrada rampa y el obtenido para entrada escalón. Las respuestas temporales ratifican el comportamiento divergente de este sistema.

TRAYECTORIAS DE FASE

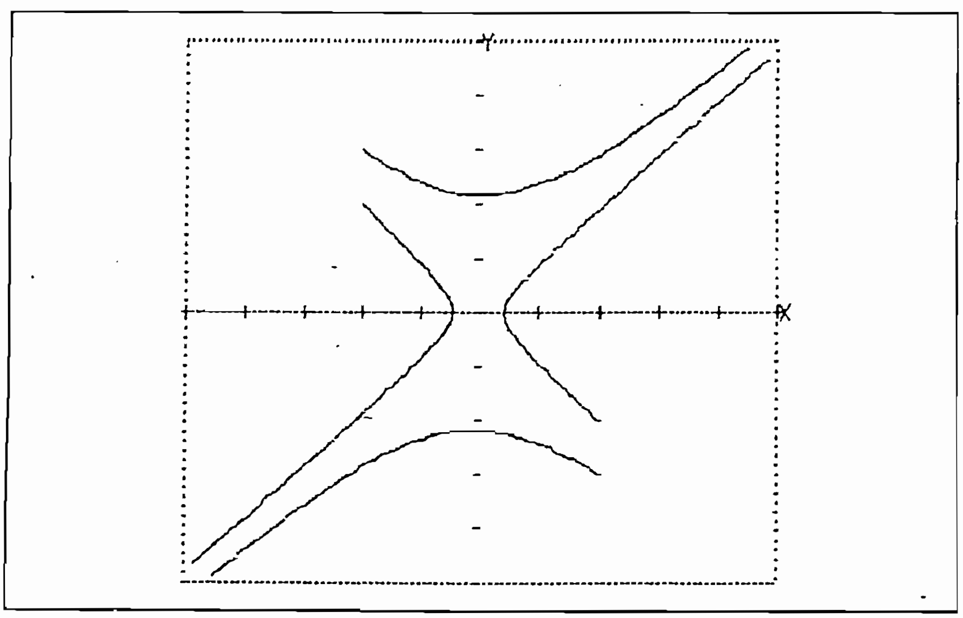
G(s)=-1/s^2 Entrada ESCALON UNITARIO Pto Sing SILLA



1 division = 1 unidades

TRAYECTORIAS DE FASE

G(s)=-1/s^2 Entrada RAMPA Pendiente=2 Pto Sing SILLA



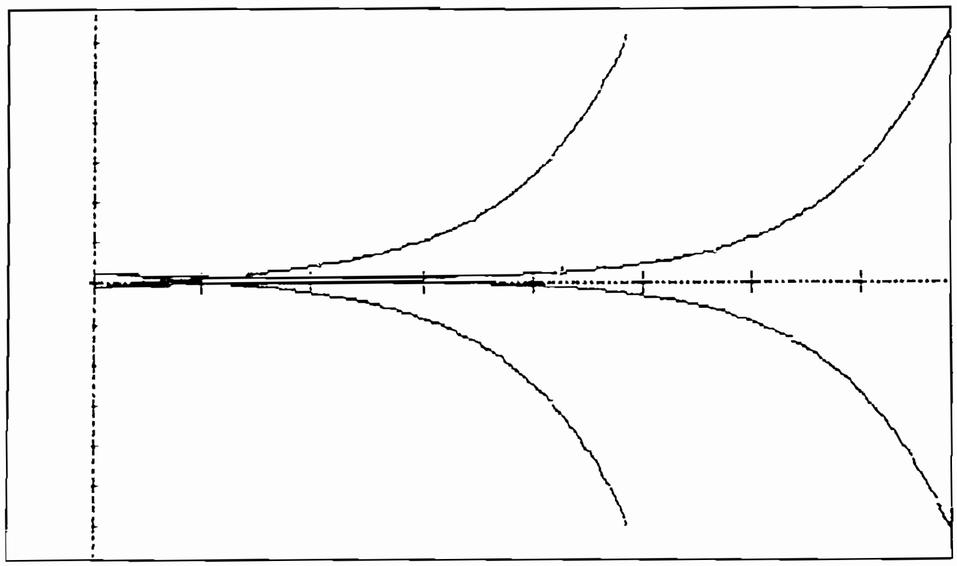
1 division = 1 unidades

Figura 4.12 Retratos de fase correspondientes al sistema del ejemplo 6

Figura 4.13 Respuesta en el tiempo correspondientes al sistema del ejemplo 6

GRAFICO: SALIDA versus TIEMPO

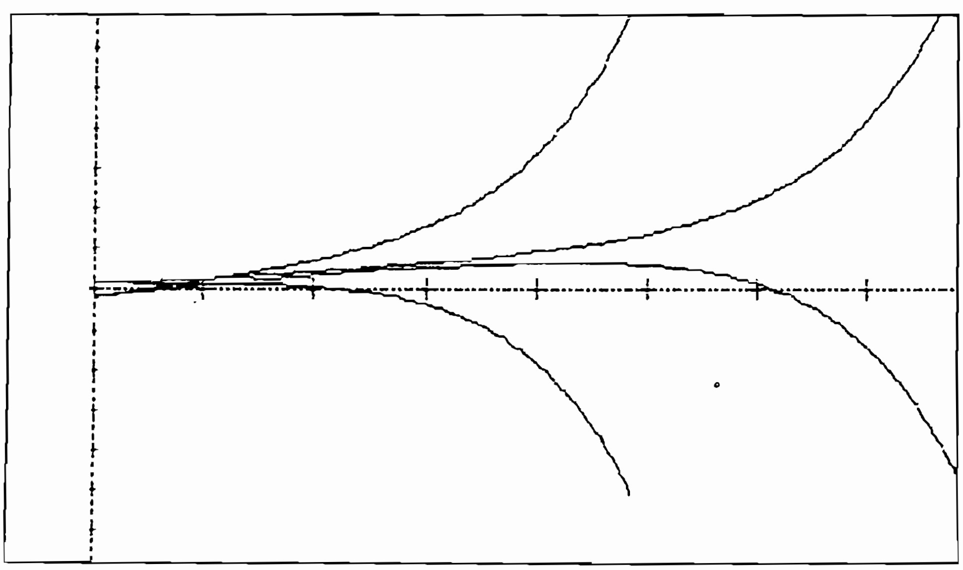
G(s)=-1/s^2 Entrada ESCALON UNITARIO Pto Sing SILLA



1 division vertical = 10 unidades
1 division horizontal = 1 unidades _

GRAFICO: SALIDA versus TIEMPO

G(s)=-1/s^2 Entrada RAMPA Pendiente=2 Pto Sing SILLA



1 division vertical = 10 unidades
1 division horizontal = 1 unidades _

La validez del programa se muestra por la correspondencia del retrato de fase graficado y tipo de punto singular y su ubicación teórica esperados.

Como se indicó anteriormente el cálculo permite obtener gráficos que muestran el comportamiento del sistema hasta que este alcanza un estado permanente o su divergencia es evidente.

En resumen, recordando que, en estos sistemas, en la función de transferencia tenemos

$$a_2 = 1$$
$$a_1 = 2\xi wn$$
$$a_3 = wn\text{^}2$$

La figura 4.1 indica la existencia de un Foco estable, tal como se espera para los valores de frecuencia w_n^2 positivos y los valores de coeficiente de amortiguamiento positivos menores que uno. La figura 4.2 muestra el error esperado con oscilaciones alrededor de su valor final, tomando valores positivos y negativos.

La figura 4.3 indica la existencia de un Centro, tal como se espera para los valores de frecuencia w_n^2 positivos y coeficiente de amortiguamiento igual a cero. La figura 4.4. muestra la salida del sistema con sus oscilaciones mantenidas.

La figura 4.6 indica la existencia de un Nodo estable , tal como se espera para los valores de frecuencia w_n^2 positivos y coeficiente de amortiguamiento positivos igual o mayores que uno. La figura 4.7 presenta la salida del sistema tal como se teóricamente debe ser en estos casos, sin oscilaciones.

La figura 4.9 indica la existencia de un Foco inestable, tal como se espera para los valores de frecuencia w_n^2 positivos y los valores de coeficiente de amortiguamiento negativos mayores que menos uno (-1). La figura 4.10 presenta la salida del sistema, continuamente creciente.

La figura 4.11 indica la existencia de un Nodo inestable, tal como se espera para los valores de frecuencia w_n^2 positivos y los valores de coeficiente de amortiguamiento negativos iguales o menores que menos uno (-1).

La figura 4.12 indica la existencia de una Silla , tal como se espera para los valores de frecuencia w_n^2 negativos y coeficiente de amortiguamiento igual a cero.

En suma, los valores de frecuencia y amortiguamiento están en correspondencia a las respuestas con el programa TTF.

4.1.2 EJEMPLOS DE SISTEMA NO LINEAL DE SEGUNDO ORDEN

Siguiendo el análisis teórico realizado para los sistemas lineales y apoyándose en la técnica de linealización se puede determinar el tipo de puntos singulares y su ubicación en el plano de fase para los sistemas no lineales.

EJEMPLO 1 SILLA DE MONTAR Y FOCO ESTABLE

Sea el sistema definido por la ecuación diferencial no lineal de segundo orden:

$$\ddot{x} + 0.5\,\dot{x} + 2\,x + x\text{^}2 = 0 \qquad (4.10)$$

Tomando como variables de estado a:

$$x_1 = x$$

$$x_2 = \dot{x}$$

se tiene como ecuaciones de estado:

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

$$\dot{x}_2 = -2x_1 - 0.5x_2 - x1\text{^}2 \qquad (4.11)$$

Igualando a cero las ecuaciones para encontrar los puntos singulares se obtiene:

$$x_2 = 0$$

$$x_1(x_1 + 2) = 0$$

Existe, por tanto, dos puntos singulares.

$$x_1 = 0, \quad x_2 = 0$$

y

$$x_1 = -2, \quad x_2 = 0$$

Linealizando la ecuación en las proximidades de cada uno de estos puntos se determina el tipo de punto singular existente. Linealizando en las cercanías del punto $x_1=0$, $x_2=0$ se tiene:

$$\ddot{x} + 0.5\dot{x} + 2x = 0 \qquad (4.12)$$

cuya ecuación caracteristica es :

$$\lambda^2 + 0.5\,\lambda + 2 = 0 \qquad (4.13)$$

y las raices son:

$$\lambda_1 =-0.25 +j\ 1.39$$

$$\lambda_2 =-0.25 +j\ 1.39$$

Según la tabla 2.1 esta ubicación determina la existencia de un punto singular del tipo FOCO ESTABLE.

En la vecindad del punto $x_1=-2$, $x_2=0$ previa a la linealización se hace el cambio de variable,

$$y = x + 2 \qquad (4.14)$$

para escribir la ecuación como:

$$\ddot{y} + 0.5\dot{y} - 2y + y^{\wedge 2} = 0 \qquad (4.15)$$

Linealizando en la cercanias del punto x1=-2 , x2 =0, esto es y_1 =0 , y_2 =0 por el cambio de variable se tiene:

$$\ddot{y} + 0.5\ \dot{y} - 2 = 0 \qquad (4.16)$$

cuya ecuación caracteristica es :

$$\lambda^{\wedge 2} + 0.5\,\lambda - 2 = 0 \qquad (4.17)$$

y sus raices son:

$$\lambda_1 = 1.19$$

$$\lambda_2 = -1.69$$

De acuerdo a la tabla 2.1 el punto singular en el punto $x_1=-2$, x_2 =0 es un punto SILLA.

Al ingresar los datos de las ecuaciones de estado del sistema al programa TTF se obtuvieron las trayectorias mostradas en la figura 4.13

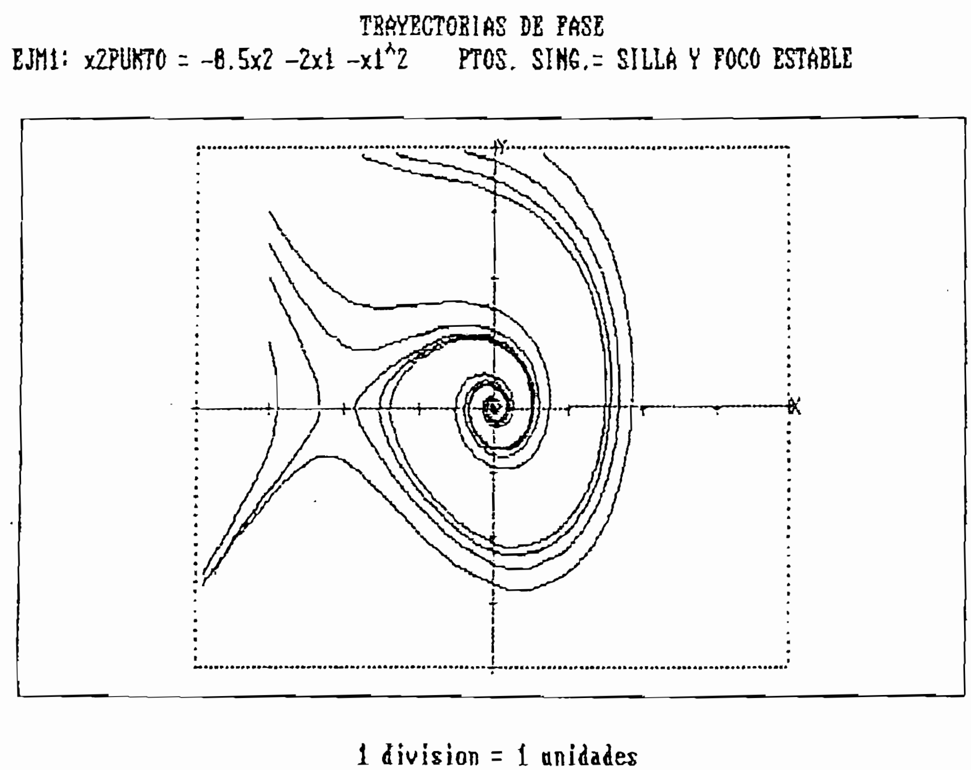


Figura 4.13 Retratos de fase correspondientes al sistema del ejemplo 1

La validez del programa se muestra por la correspondencia del retrato de fase graficado y tipo de punto singular y su ubicación teórica esperados.

Los siguientes son los datos correspondientes a los puntos iniciales, finales y máximos últimos de una trayectoria graficada.

TRAYECTORIA #1

ESTADO INICIAL:

$X_1=-3$ $X_2=6$

ESTADO FINAL :

$X_1 =-8.37$ $X_2=-12.67$

TIEMPO = 5.7

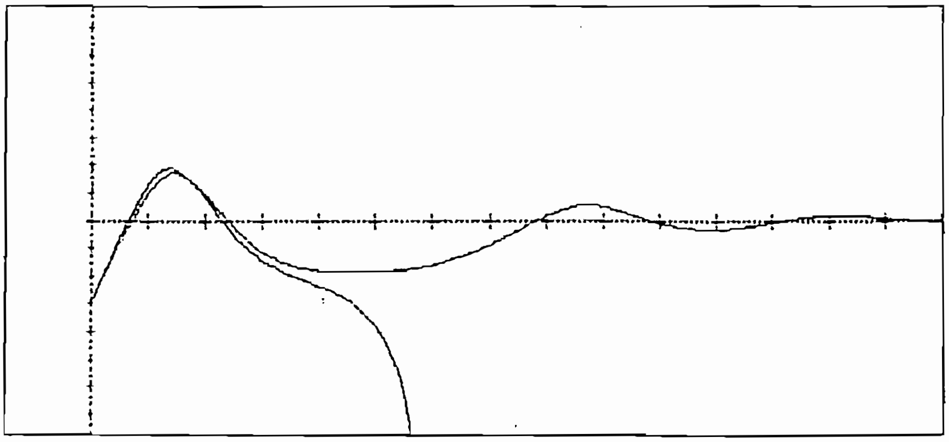
VALORES MAXIMOS ALCANZADOS

$X_1=1.86$ $t = 1.4$

$X_2=6$ $t = 0$

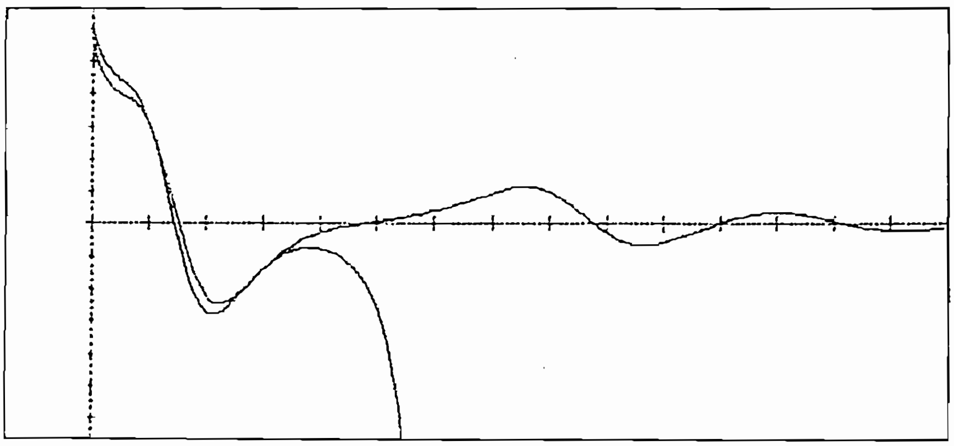
La figura 4.14 presenta el comportamiento de las variables de estado del sistema en el tiempo para una de las condiciones iniciales dadas. Como se indicó anteriormente el cálculo permite obtener gráficos que muestran el comportamiento del sistema hasta que este alcanza un estado permanente.

RESPUESTA DE X1 EN EL TIEMPO
EJM1: x2PUNTO = -0.5x2 -2x1 -x1^2 PTOS. SING.= SILLA Y POCO ESTABLE



1 division vertical = 1 unidades
1 division horizontal = 1 unidades

RESPUESTA DE X2 EN EL TIEMPO
EJM1: x2PUNTO = -0.5x2 -2x1 -x1^2 PTOS. SING.= SILLA Y POCO ESTABLE



1 division vertical = 1 unidades
1 division horizontal = 1 unidades

Figura 4.14 Respuesta en el tiempo correspondientes al sistema del ejemplo 1

EJEMPLO 2 SISTEMA CON ELEMENTO SEGMENTO LINEAL.

Considérese un sistema no lineal de control sin compensación y con realimentación unitaria, cuya función de transferencia de la parte lineal es de la forma :

$$G(s) = \frac{k}{a_2 s^2 + a_1 s + a_0}$$

donde

$$a_2 = 1$$
$$a_1 = 1$$
$$a_0 = 0$$

y como parte no lineal tiene un relé con zona muerta, con salida −1 a entradas negativas y +1 a entradas positivas ,y una zona muerta para entradas entre −1 y 1.

Se obtendrá el retrato de fase asumiendo ausencia de señal de entrada.

Según el análisis teórico realizado para el ejemplo 1 de sistemas lineales de segundo orden la función de transferencia determina que en ausencia de no linealidad exista un punto singular en el origen de tipo nodo estable . La inclusión de un elemento de característica segmento lineal provoca el desdoblamiento de las ecuaciones de estado .

Las ecuaciones que relacionan las variables del sistema son:

$$e = r - c \qquad (4.18)$$

$$\begin{array}{lll} u = 1 & \text{si} & e > 1 \\ u = 0 & \text{si} & -1 < e < 1 \\ u = -1 & \text{si} & e < -1 \end{array} \qquad (4.19)$$

$$c = \frac{1}{s(s+1)} * u \qquad (4.20)$$

De la ecuación (4.20) :

$$s^2 c + sc = u$$

$$\ddot{c} + \dot{c} = u$$

Si se reemplaza c de la ecuación (4.18) :

$$-\ddot{e} - \dot{e} = u \qquad (4.21)$$

Si se toma como variables de estado al error y su derivadas ecuaciones de estado serán:

a) cuando $e > 1$

$$-\ddot{e} - \dot{e} = 1$$

$$\ddot{e} + \dot{e} + 1 = 0 \qquad (4.22)$$

en variables de estado:

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

$$\dot{x}_2 = -x_2 - 1 \qquad (4.23)$$

b) cuando $-1 < e < 1$

$$-\ddot{e} - \dot{e} = 0$$

$$\ddot{e} + \dot{e} = 0 \qquad (4.24)$$

en variables de estado :

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

$$\dot{x}_2 = -x_2 \qquad (4.25)$$

c) cuando $e < -1$

$$-\ddot{e} - \dot{e} = -1$$

$$\ddot{e} + \dot{e} - 1 = 0 \qquad (4.26)$$

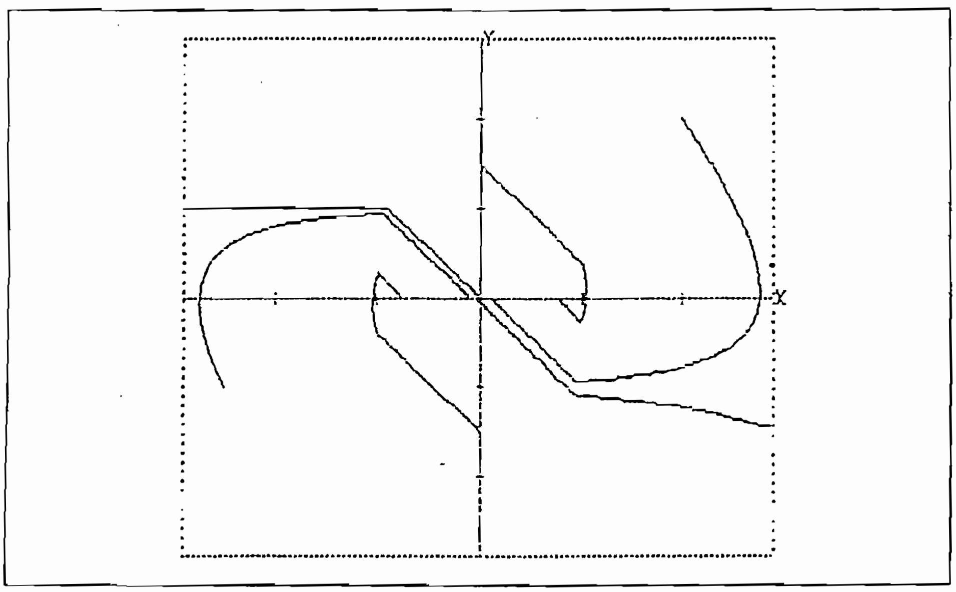
en variables de estado :

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

$$\dot{x}_2 = -x_2 + 1 \qquad (4.27)$$

Con las ecuaciones (4.23), (4.25), (4.27) el plano de fase queda dividido en tres regiones. Igualando a cero el grupo de ecuaciones correspondientes a la región $-1 < e < 1$ se tiene que cualquier punto en esta región con coordenadas $x_2 = 0$, x_1 cualquiera , es un punto de equilibrio del sistema.

EJEMPLO3: G(s)=4/s(s +1)   RELE salidas -1 y 1   ZONA MUERTA desde -1 a 1
PTO. SING.= FOCO ESTABLE



1 division = 1 unidades

Figura 4.15 Retratos de fase correspondientes al sistema del ejemplo 2

La validez del programa se muestra por la correspondencia del retrato de fase graficado y tipo de punto singular y su ubicación teórica esperados.

Los siguientes son los datos correspondientes a los puntos iniciales, finales y máximos últimos de una trayectoria graficada.

TRAYECTORIA #1

ESTADO INICIAL:

$X_1 = 2$ $X_2 = 2$

ESTADO FINAL :

$X_1 = .123$ $X_2 = -2.61$ E-02

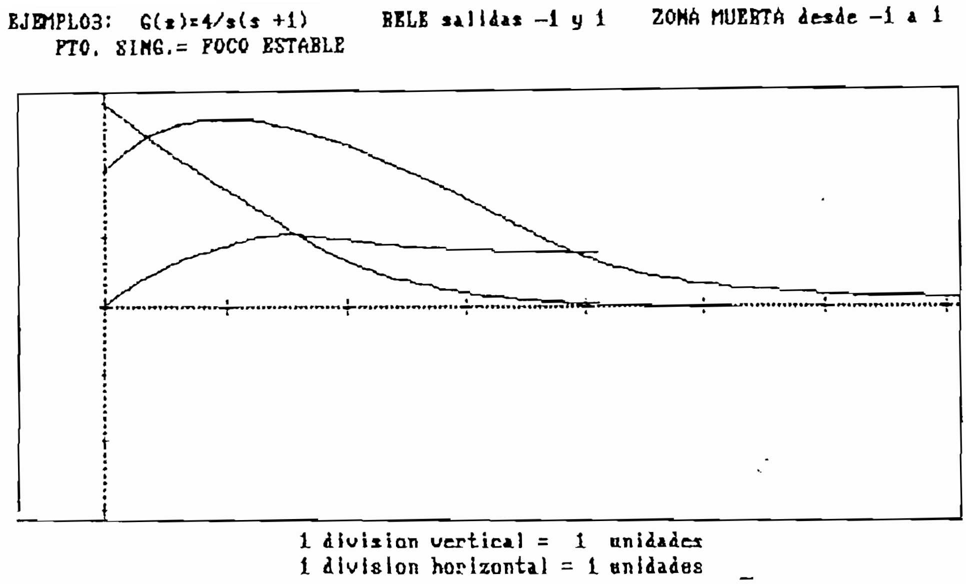
TIEMPO = 7.1

VALORES MAXIMOS ALCANZADOS

$X_1 = 2.76$ $t = 1$

$X_2 = 2$ $t = 0$

La figura 4.16 presenta el comportamiento de las variables de estado del sistema en el tiempo para una de las condiciones iniciales dadas. Como se indicó anteriormente el cálculo permite obtener gráficos que muestran el comportamiento del sistema hasta que este alcanza un estado permanente.



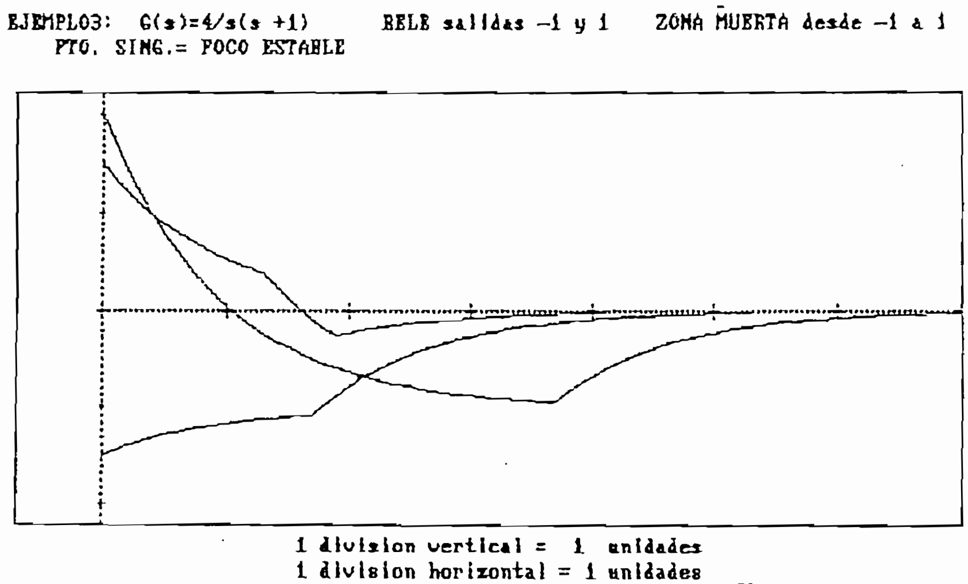


Figura 4.16 Respuesta en el tiempo correspondientes al sistema del ejemplo 2

EJEMPLO 3   PUNTOS SINGULARES PERIODICOS

Un sistema físico de péndulo simple sin rozamiento puede ser expresado matematicamente por la ecuación:

$$x = (-g/l) \sin(x)$$

donde x es el ángulo medido en sentido antihorario desde la vertical, g es la gravedad y l es la longitud del brazo del péndulo .

Tomando como variables de estado a:

$$x_1 = x$$

$$x_2 = \dot{x}$$

las ecuaciones de estado del sistema son:

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

$$\dot{x}_2 = -(g/l) \sin(x_1)$$

Igualando las ecuaciones a cero se determina la existencia de puntos periodicos singulares:

$$x_2 = 0$$

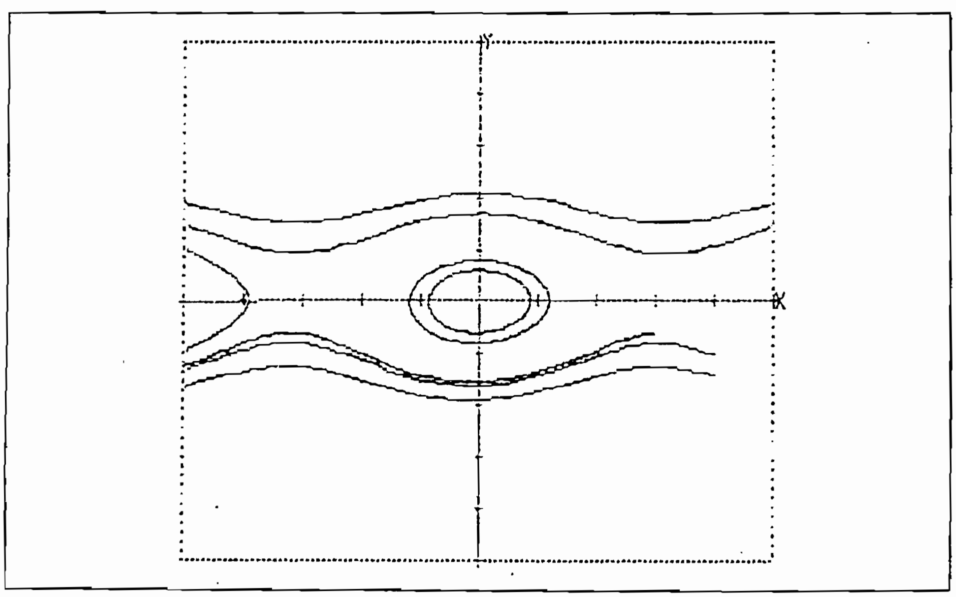
$$x_1 = 0 \pm k* \text{pi} \qquad \text{pi}=3.1416,\ k=0,1,2,..$$

Físicamente el punto de interés es el ubicado en el origen del plano de fase. Este punto está encerrado entre trayectorias que salen de −pi y llegan a pi tomando valores positivos de x_2, y que salen de pi y llegan a −pi tomando valores negativos de x_2. Los puntos (−pi,0) y (pi,0) corresponde a sillas de montar.

El retrato de fase correspondiente al ejemplo se muestra en la figura 4.18

TRAYECTORIAS DE FASE

EJEMPLO2: x2PUNTO = -.1sin(x1)   PTO. SING.= CENTROS PERIODICOS



1 division = 1 unidades

Figura 4.17 Retrato de fase correspondiente al ejemplo 3

EJEMPLO 4: PUNTOS SINGULARES PERIODICOS [3]

Un sistema de potencia consistente en dos generadores sincrónicos ,bajo las usuales simplificaciones , después de una falla es descrito por las ecuaciones:

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

$$\dot{x}_2 = -.7143x_2 +.234 -.234\cos(x_1) -.597\sin(x_1)$$

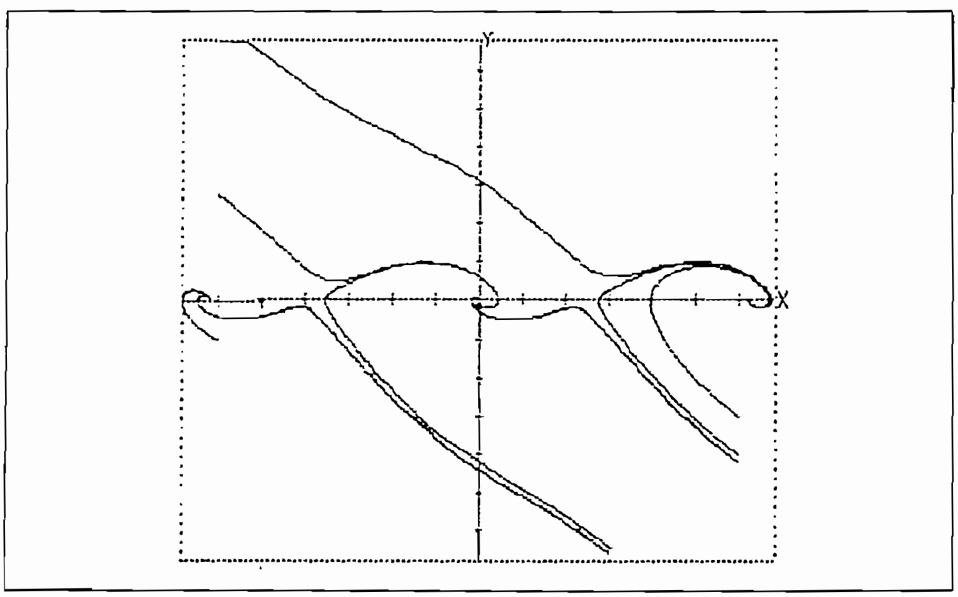
Igualando estas ecuaciones a cero se puede obtener la ubicación de los puntos singulares del sistema. Puesto que las ecuaciones involucran funciones trigonométricas, la presencia de puntos singulares se repetirá periódicamente.

Los puntos singulares de interés son , igualando a cero las ecuaciones, (0,0) y (2.39,0).

Utilizando el programa TTF se puede obtener gráficamente una aproximación de la región de estabilidad del sistema.

Mediante el trazo de unas pocas trayectorias, en la vecindad de los puntos singulares, se llega a determinar que en el origen se tiene un NODO ESTABLE y que el segundo punto singular corresponde a una SILLA DE MONTAR. Estos dos tipos de singularidades se repiten en forma periódica, y es de interés limitar mediante trayectorias la zona de estabilidad. La existencia de una silla de montar involucra mejorar la precisión del cálculo para determinar las separatrices de esta singularidad, para ello es útil la opción 5 del menú principal del programa TTF, que permite el cambio del intervalo de tiempo usado para la simulación digital, el numero de puntos máximo para el calculo y el número de trayectorias máximo a graficarse. En la figura 4.18 se muestra el plano de fase dividido por trayectorias que delimitan la zona de atracción o estabilidad del sistema.

TRAYECTORIAS DE FASE



1 division = 1 unidades

Figura 4.18. Retrato de fase del sistema de potencia del ejemplo 3.

Para el cálculo de los puntos de las trayectorias se usó la opción 5 , cambios en el programa, y se asignó el valor de 0.01 al intervalo de tiempo para la simulación digital.

4.2 EJEMPLO DE SISTEMAS HASTA EN TRES DIMENSIONES.

En esta parte se indica como obtener el retrato de fase de sistemas de tercer orden lineales y no lineales.

4.2.1 EJEMPLO DE SISTEMA LINEAL DE TERCER ORDEN

Considérese un sistema lineal de control sin compensación y con realimentación unitaria, cuya función de transferencia es de la forma :

$$G(s) = \frac{k}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{4.28}$$

donde

$a_3 = 1$
$a_2 = 1-\alpha-w_n$, siendo $s = -\alpha - jw_n$ un polo del sistema
$a_1 = 1-\alpha+w_n$, siendo $s = -\alpha + jw_n$ un polo del sistema
$a_0 = p_1$, siendo $s = -p_1$ un polo del sistema

tal que el sistema puede ser expresado por la ecuación de estado:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 \\ 0 & -\alpha & +wn \\ 0 & -wn & -\alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{4.29}$$

Puede determinarse el tipo de punto singular existente si se conoce la ubicación de los polos del sistema en el plano complejo, según la tabla 2.3 indicada en el capítulo segundo.

Los polos del sistema en función de los coeficientes a_3, a_2, a_1, a_0 serán:

$$s = -a_0$$

$$s = -[(a_1 + a_2 - 2)/2] + j\ [(a_1 - a_2)/2]$$

$$s = -[(a_1 + a_2 - 2)/2] - j\ [(a_1 - a_2)/2] \qquad (4.30)$$

Puede conocerse la ubicación de los puntos singulares en el plano de fase igualando las ecuaciones de estado a cero para hallar los puntos donde se produce una indeterminación.

Las ecuaciones que relacionan las variables del sistema son:

$$e = r - c$$

$$c = \frac{1}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} * e \qquad (4.31)$$

de la ecuación (4.29) :

$$a_3 s^3 c + a_2 s^2 c + a_1 s c + a_0 c = e \qquad (4.32)$$

reemplazando c de la ecuación del error :

$$-a_3 s^3 e - a_2 s^2 e - a_1 s e - a_0 e + a_0 r = 0 \qquad (4.33)$$

Tomando como variables de estado al error , y sus derivadas se tiene:

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

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

$$\dot{x}_3 = -[a_2 x_3 + a_1 x_2 + a_0 x_1 - a_0 r]/a_3 \qquad (4.32)$$

El sistema analizado puede ser expresado por la ecuación vectorial (4.29) o el sistema se ecuaciones (4.32). De cualquiera de ellas puede obtenerse la ubicación de los puntos singulares.

Usando las ecuaciones (4.32) la ubicación es:

$$x_2 = 0$$
$$x_3 = 0$$
$$a_2x_3 + a_1x_2 + a_0x_1 + a_0r = 0$$
o
$$x1 = - r$$

Es decir existe un único punto singular en el espacio de fase ubicado en el punto (-r,0,0). En ausencia de señal de entrada el punto singular se encuentra en el origen

EJEMPLO 1 CILINDRO O CENTRO

Sea un sistema de tercer orden con los valores k=1, a_3= 1, a_2=0, a_1=2 , a_0=-1. Se analizará su comportamiento en ausencia de señal de entrada.

Según las ecuaciones (4.28) los polos del sistema son:

$$s = -1$$

$$s = +j$$

$$s = -j$$

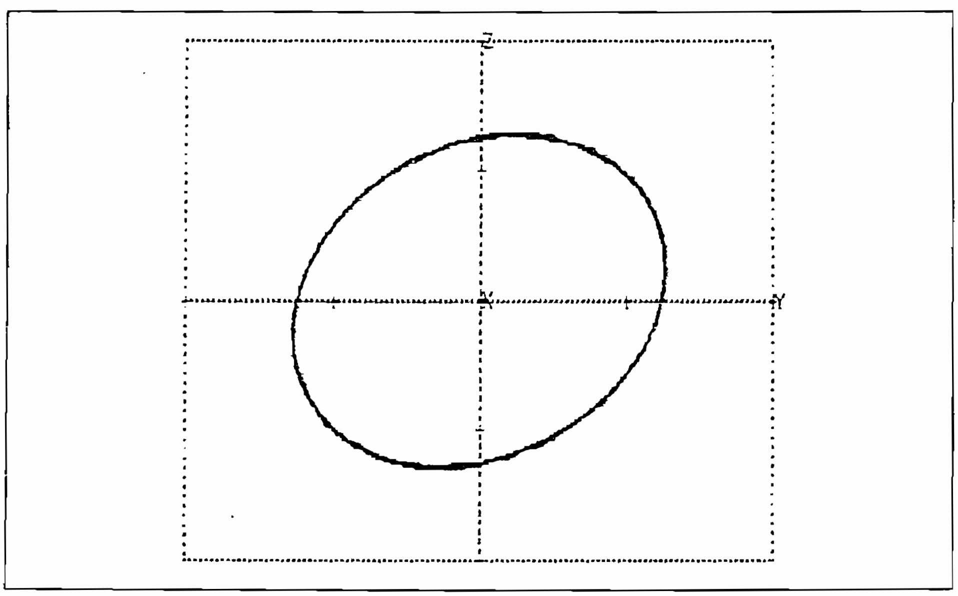
Según la tabla 2.3 del segundo capítulo, esta ubicación determina la existencia de un punto singular tipo cilindro o centro paralelo al eje de x_1. Se lo designa con estos nombres por la forma que adopta cualquier trayectoria vista en tres dimensiones o en su proyección sobre el plano x3-x2

Según las ecuaciones (4.32), igualadas acero, la ubicación de este punto singular es el origen del espacio de fase (0,0,0).

$$x_1 = 0$$
$$x_2 = 0$$
$$x_3 = 0$$

La trayectoria obtenida con el programa TTF luego de ingresar los datos de su representación en diagrama de bloques es la mostrada en la figura 4.19 , 4.20 y 4.21

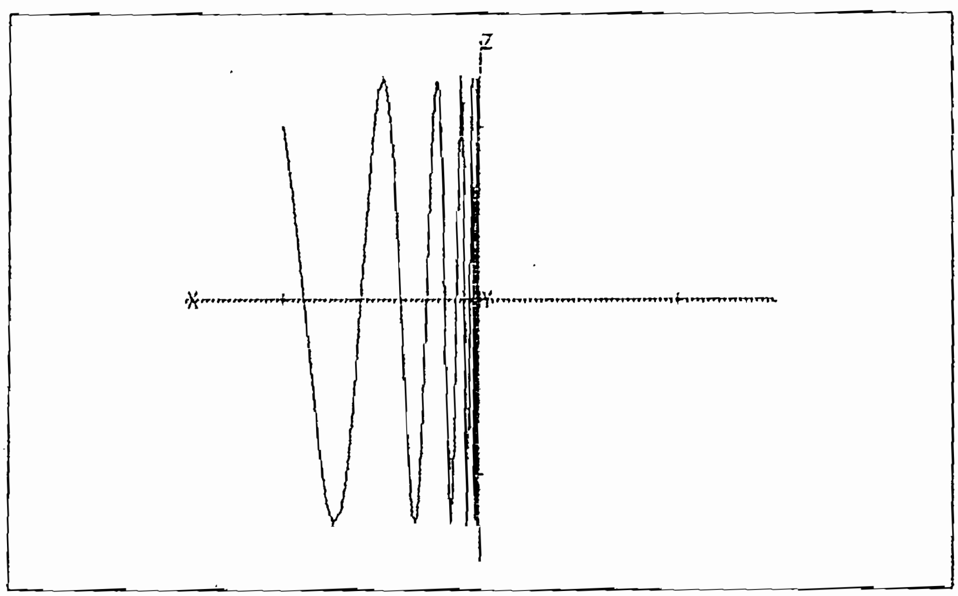
EJEMPLO1: G(s)=1/(s^3 +2s -1)　　PTO. SING.= CENTRO INESTABLE paralelo al eje x1 (X)



1 division = 1 unidades

Figura 4.19　Trayectoria de fase correspondientes al sistema del ejemplo 1. Vista del plano ZY (X_3-X_2).

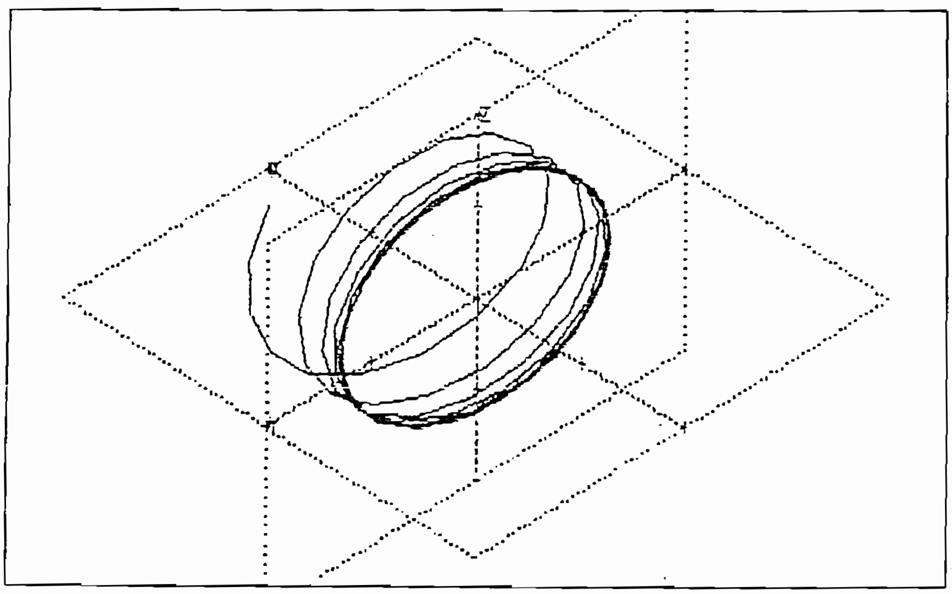
EJEMPLO1: $G(s)=1/(s^3 +2s -1)$ PTO. SING.= CENTRO INESTABLE paralelo al eje x1 (X)



1 division = 1 unidades

Figura 4.20 Trayectoria de fase correspondientes al sistema del ejemplo 1. Vista del plano XY (X_1-X_2).

EJEMPLO1: G(s)=1/(s^3 +2s -1)   PTO. SING.= CENTRO INESTABLE paralelo al eje x1 (X)



1 division = 1 unidades

Figura 4.21 Trayectorias de fase correspondientes al sistema del ejemplo 1. Vista tridimensional.

La validez del programa se muestra por la correspondencia del retrato de fase graficado y el tipo de punto singular y su ubicación teórica esperados.

Los siguientes son los datos correspondientes a los puntos iniciales, finales y máximos últimos de la trayectoria graficada.

TRAYECTORIA #1

ESTADO INICIAL:

X_1= 1 X_2= 1 X_3= 1

ESTADO FINAL :

X_1 = 4.55E-04 X_2= -1.29 X_3= -.217

TIEMPO = 15

VALORES MAXIMOS ALCANZADOS

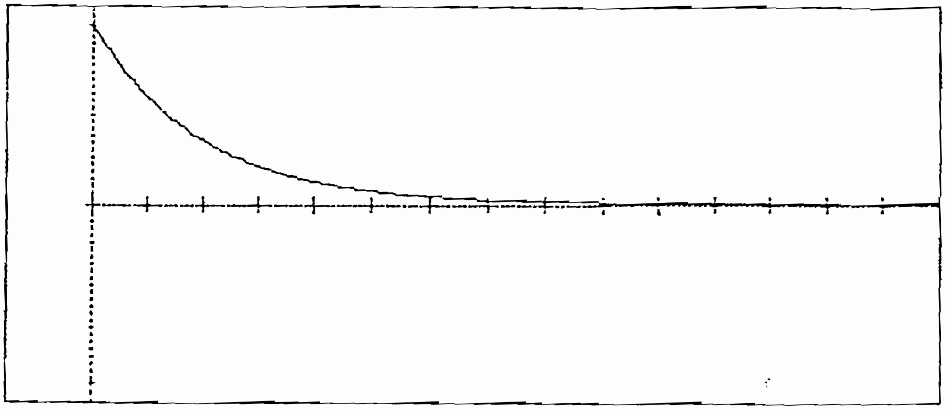
X_1= 1 t = 0

X_2= 1.29 t = 11.1

X_3= 1.29 t = 9.2

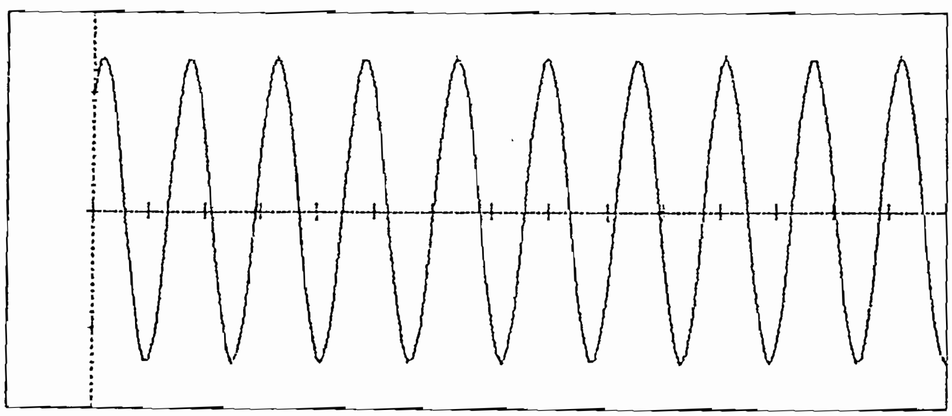
La figura 4.22 presenta el comportamiento de las variables de estado del sistema en el tiempo para una de las condiciones iniciales dadas.

EJEMPLO1: G(s)=1/(s^3 +2s -1) PTO. SING.= CENTRO INESTABLE paralelo al eje x1 (X)



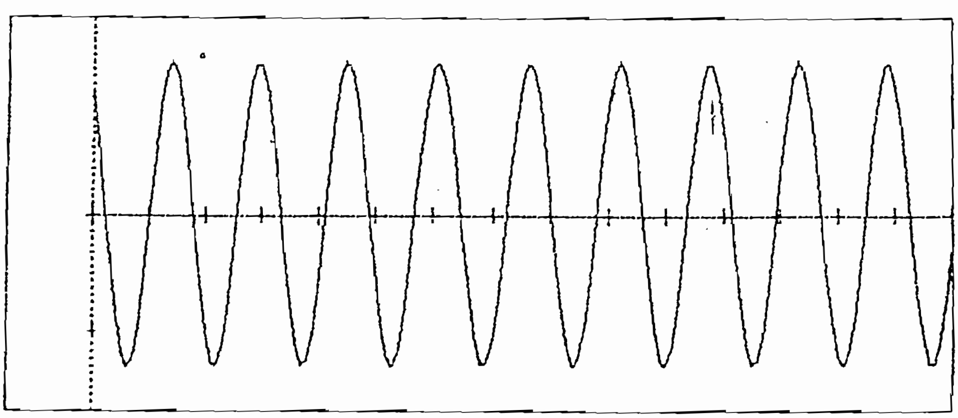
1 division vertical = 1 unidades
1 division horizontal = 1 unidades

EJEMPLO1: G(s)=1/(s^3 +2s -1) PTO. SING.= CENTRO INESTABLE paralelo al eje x1 (X)



1 division vertical = 1 unidades
1 division horizontal = 1 unidades

EJEMPLO1: G(s)=1/(s^3 +2s -1) PTO. SING.= CENTRO INESTABLE paralelo al eje x1 (X)



1 division vertical = 1 unidades
1 division horizontal = 1 unidades

EJEMPLO 2 TORNILLO O FOCO

Sea el sistema de tercer orden con función de transferencia

$$G(s) = \frac{k}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \qquad (4.28)$$

donde

$a_3 = 1$
$a_2 = 1-\alpha-w_n$, siendo $s = -\alpha - j w_n$ un polo del sistema
$a_1 = 1-\alpha+w_n$, siendo $s = -\alpha + j w_n$ un polo del sistema
$a_0 = p_1$, siendo $s = -p_1$ un polo del sistema

tal que el sistema puede ser expresado por la ecuación de estado:

$$\begin{matrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{matrix} = \begin{bmatrix} -p_1 & 0 & 0 \\ 0 & -\alpha & +wn \\ 0 & -wn & -\alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad (4.29)$$

Sean los polos determinados por la función de transferencia

$s = -.5$
$s = -.1 + j\ 4$
$s = -.1 + j\ 4$

Según la ubicación de los polos en el plano complejo se determina la existencia de un punto singular de tipo FOCO ESTABLE o TORNILLO. Se designa con estos nombres por la foma que adquiere cualquier trayectoria en su proyección sobre el plano x_3-x_2 o en su vista tridimensional. La ubicación del mismo según las ecuaciones (4.32) es el punto (0,0,0) en ausencia de señal.

La trayectoria obtenida luego de ingresar los datos al programa TTF es la indicada en las figuras 4.23, 4.24 y 4.25

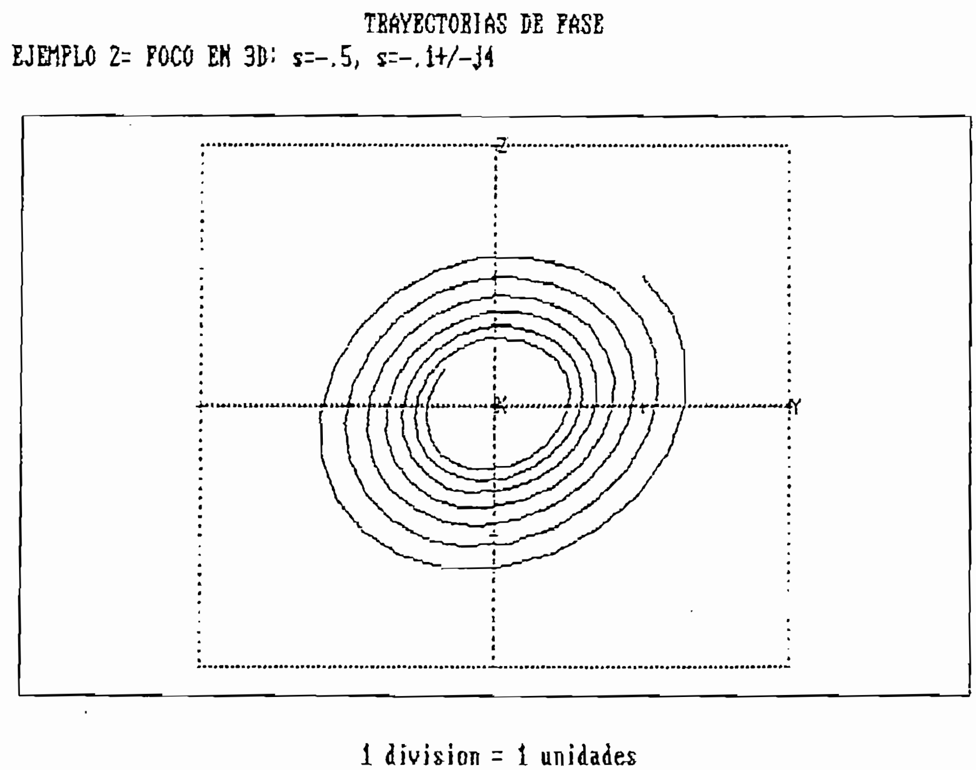


Figura 4.23 Trayectoria de fase correspondientes al sistema del ejemplo 1. Vista del plano ZY (X_3-X_2).

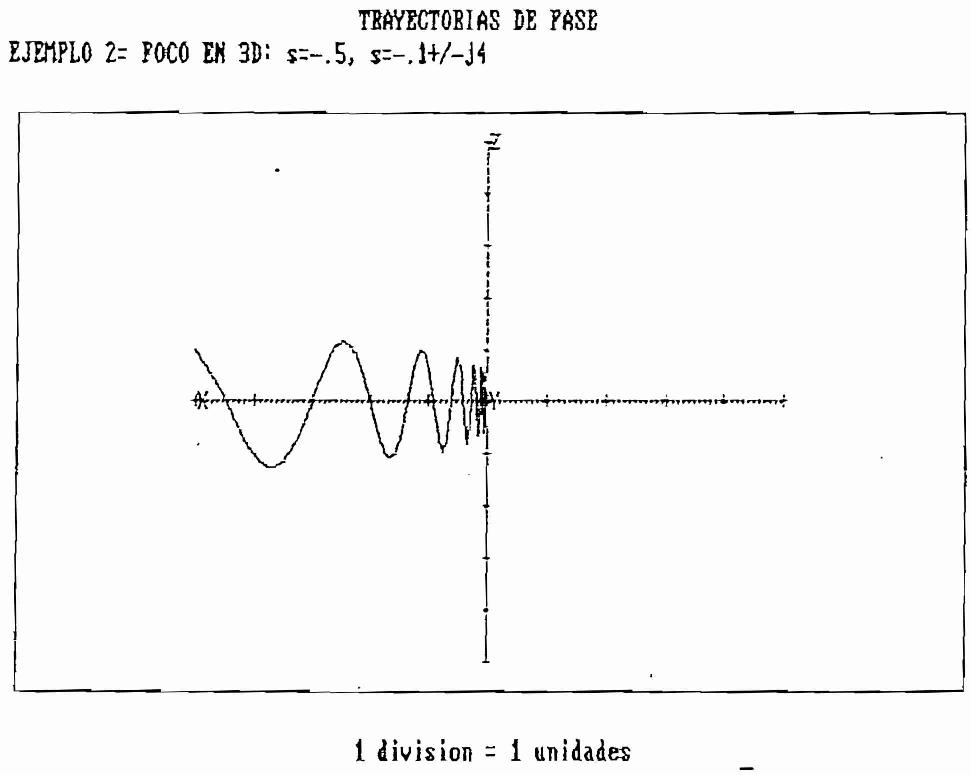


Figura 4.24 Trayectoria de fase correspondientes al sistema del ejemplo 1. Vista del plano XY (X_1-X_2).

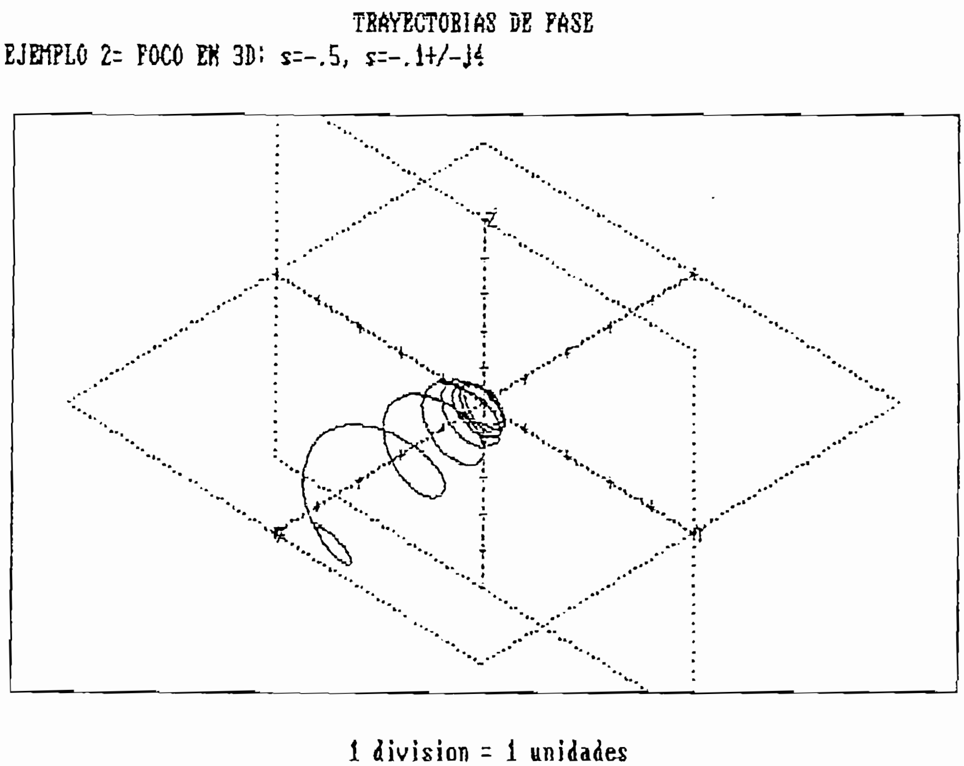


Figura 4.25 Trayectorias de fase correspondientes al sistema del ejemplo 1. Vista tridimensional.

La validez del programa se muestra por la correspondencia del retrato de fase graficado y tipo del punto singular y su ubicación teórica esperados.

Los siguientes son los datos correspondientes a los puntos iniciales, finales y máximos últimos de la trayectoria graficada.

TRAYECTORIA #1

ESTADO INICIAL:

$X_1 = 5$ $X_2 = 1$ $X_3 = 1$

ESTADO FINAL :

$X_1 = 2.388E-02$ $X_2 = -.342$ $X_3 = -.258$

TIEMPO = 15

VALORES MAXIMOS ALCANZADOS

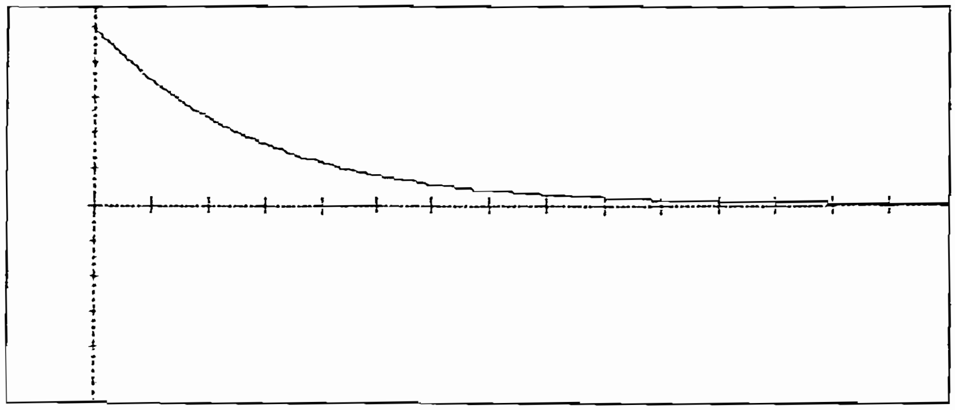
$X_1 = 5$ $t = 0$

$X_2 = 1.289$ $t = .2$

$X_3 = 1.1497$ $t = 2$

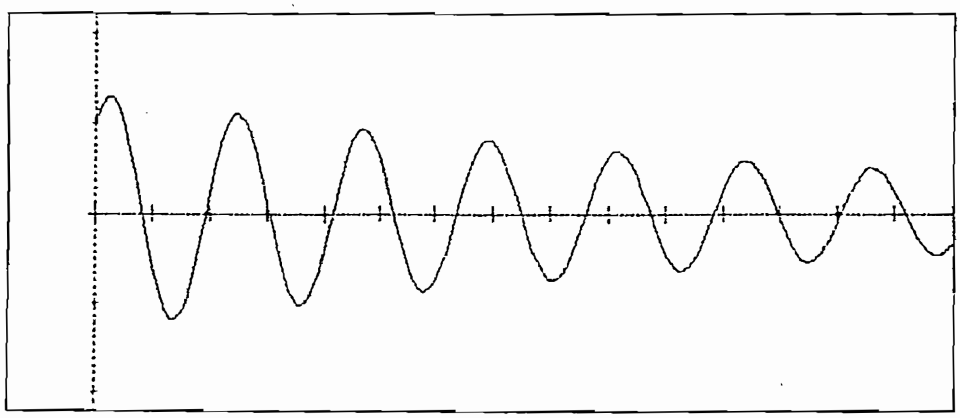
La figura 4.23 presenta el comportamiento de las variables de estado del sistema en el tiempo para una de las condiciones iniciales dadas. Como se indicó anteriormente el cálculo permite obtener también gráficos que muestran el comportamiento del sistema en el tiempo.

RESPUESTA DE X1 EN EL TIEMPO
EJEMPLO 2= FOCO EN 3D: s=-.5, s=-.1+/-j4



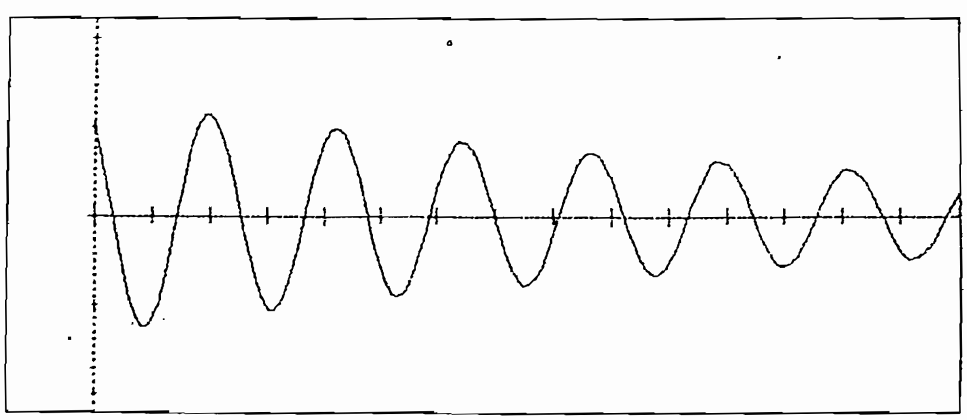
1 division vertical = 1 unidades
1 division horizontal = 1 unidades

RESPUESTA DE X2 EN EL TIEMPO
EJEMPLO 2= FOCO EN 3D: s=-.5, s=-.1+/-j4



1 division vertical = 1 unidades
1 division horizontal = 1 unidades

RESPUESTA DE X3 EN EL TIEMPO
EJEMPLO 2= FOCO EN 3D: s=-.5, s=-.1+/-j4



1 division vertical = 1 unidades
1 division horizontal = 1 unidades

4.2.2 EJEMPLOS DE SISTEMAS NO LINEALES

Para el análisis de los sistemas no lineales se utiliza la representación en diagramas de bloques, si el sistema puede ser representado incluyendo un elemento segmento lineal , o la representación de ecuaciones de estado , si el sistema puede ser representado incluyendo un término trigonométrico , exponencial , logarítmico o potenciación de las variables de estado.

EJEMPLO 1: CICLO LIMITE

La inclusión de un elemento de control ON/OFF puede provocar el aparecimiento de ciclos limites, definidos en el segundo capítulo. El sistema que se analiza a continuación es de este caso.

Sea un sistema de tercer orden cuya función de transferencia tiene los valores:

$$k = 10$$

$$a_3 = 1$$

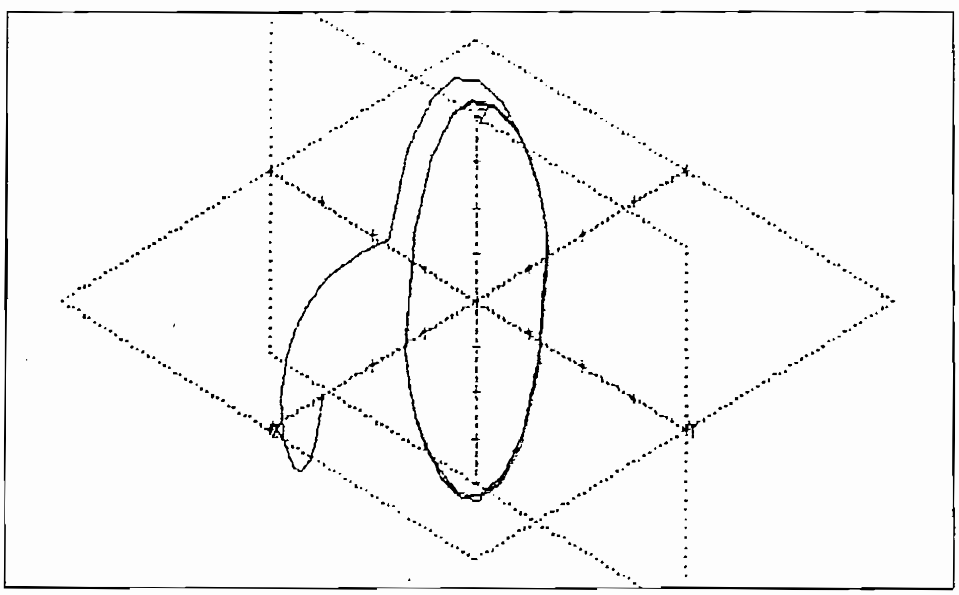
$$a_2 = 3$$

$$a_1 = 6$$

y como elemento no lineal un controlador ON/OFF con salida 1 para errores positivos y salida -1 para señales de error negativas.

La existencia del ciclo limite es fácilmente determinada con unas pocas trayectorias. En la figura 2.26 se muestra una trayectoria que alcanza el ciclo limite.

TRAYECTORIAS DE FASE



1 division = 1 unidades

Figura 2.26 Trayectoria que alcanza un ciclo limite en el sistema del ejemplo 1

CONCLUSIONES

El análisis gráfico del plano de fase, que se ha limitado tradicionalmente al análisis de sistemas de segundo orden, se ha extendido en este trabajo a los sistemas de tercer orden y se ha obtenido su retrato de fase en el espacio de estado tridimensional.

Como producto final de este trabajo se presenta un paquete de software desarrollado en forma estructurada mediante módulos, que servirá de ayuda para las materias de ingeniería de control, especialmente en el estudio de sistemas no lineales.

El programa permite la obtención de trayectorias de fase a partir de un diagrama de bloques general para sistemas de control, con parte lineal de hasta tercer orden, parte no lineal característica, compensador y realimentación unitaria. También permite, el programa, la obtención de trayectorias de fase a partir de un sistema de ecuaciones de estado, hasta con tres variables de estado.

Se ha generalizado el programa digital para que pueda ser usado en los computadores compatibles con IBM, en forma independiente del tipo de tarjeta de gráficos que estos posean.

Se ha visto que los resultados que se obtienen son de resolución y precisión aceptable para el trabajo de análisis, a igual que el tiempo requerido para la obtención de estos.

El poder de este programa se amplia con las opciones adicionales de obtención y graficación de las respuestas temporales de las variables de estado y el almacenamiento de datos para posterior uso, pudiendo recuperarse estos con cualquier programa de hoja electrónica para realizar cualquier otro tipo de análisis de los mismos.

El método numérico empleado, en el programa, para el cálculo de los puntos que forman las trayectorias, se basa en la simulación digital que, aunque es una técnica aproximada , sirve perfectamente bien para este objetivo; puesto que ofrece la posibilidad de variar el intervalo de tiempo y el número de puntos de la manera más conveniente Hay una relación inversamente proporcional entre estos dos parámetros. A menor intervalo de tiempo se necesitarán más puntos para obtener una trayectoria completa, y viceversa. Las combinaciones posibles de estos dos parámetros están limitadas únicamente por la memoria disponible en el computador. Para ello puede disminuirse el número máximo de trayectorias a calcularse. Así dentro de los límites de graficación se puede hacer la exactitud de un diagrama de plano de fase tan buena como se desee.

La representación para el ingreso de datos al programa es ventajosa, puesto que, siempre es factible transformar cualquier sistema de control no lineal a una de las dos representaciones generales, manejadas por el programa: diagrama de bloques o ecuaciones de estado. Por tanto , aunque se deben hacer las transformaciones previas, esto no representa una restricción y posibilita que la familiarización con el programa sea más fácil.

Según los ejemplos realizados, todas las afirmaciones teóricas, en cuanto al plano de fase y las características relevantes de las trayectorias , se comprueban en la práctica. Quedando por tanto , con total validez las diferentes formas de análisis de las características del sistema que puedan hacerse mediante el uso de este programa.

Para la mayoría de ejemplos resueltos ha sido suficiente un intervalo de tiempo de 0.1 segundos y un máximo de 150 puntos calculados por trayectoria. Este valor del intervalo de tiempo está relacionado a las constantes de tiempo del sistema analizado, debiendo ser menor siempre a la menor de estas constantes para no perder precisión.

En este trabajo se ha presentado el análisis gráfico asociado a sistemas no lineales de primero, segundo y tercer orden con gráficos de hasta tres dimensiones todo el análisis ha sido efectuado en el plano de fase con el error y su derivada como coordenadas para los sistemas de primero y segundo orden en el plano de fase, y el error, su primera y segunda derivada. para los sistemas de tercer orden en el espacio de fase tridimensional. Se pueden utilizar otras variables como coordenadas; pero en ese caso el plano de fase o espacio de fase varia consecuentemente.

Se a visto que el comportamiento en régimen permanente de la trayectoria de un sistema no lineal, cuando el tiempo t tiende a infinito, a de ser una de las tres posibilidades siguientes, para una condición inicial dada.

1.- La trayectoria tiende a un punto de equilibrio estable, en este caso el sistema no lineal converge .

2.- La trayectoria tiende al infinito, en este caso el sistema diverge.

3.- La trayectoria tiende a un ciclo limite , en este caso el sistema oscila permanentemente.

Los sistemas con no linealidad dependiente de la señal de entrada a ella, pueden aproximarse a sistemas segmento lineales. Con ello se divide el plano de fase en diversas subregiones, correspondiendo cada una de ellas a un funcionamiento lineal individual. Al funcionamiento en cada subregión le corresponde o acompaña un punto singular, aunque este puede no estar, necesariamente, ubicado dentro de esta subregión en particular.

De los ejemplos realizados se concluye que la señal de entrada y la característica de transferencia del elemento de ganancia no lineal determinan el comportamiento del sistema en régimen permanente. .

Finalmente, puesto que los métodos de análisis para el espacio de estado, indispensable en la teoría de control moderno, son extensiones del plano de fase que se aplican al análisis de sistemas dinámicos en un espacio n-dimensional ,y al ser éste un programa que facilita el análisis de sistemas no lineales , en el espacio tridimensional, se espera que sirva de base y de herramienta, para futuros trabajos dentro del campo de control moderno y el estudio de sistemas no lineales.

ANEXO A

GRAFICACION TRIDIMENSIONAL

La figura a.1 muestra el sistema tridimensional móvil a usarse, y su posición original respecto al plano estático de la pantalla sobre la cual se obtendrán los gráficos.

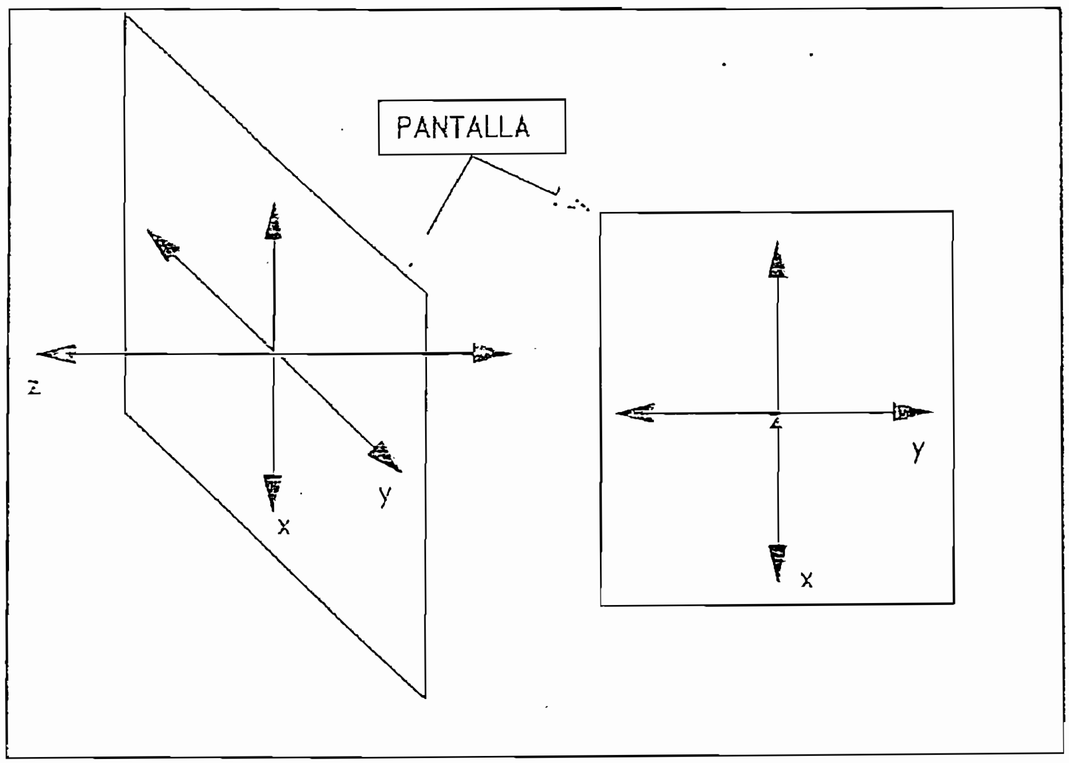


Figura a.1 Sistema de ejes tridimensional y su posición inicial respecto al plano de la pantalla

La figura a.2 muestra el sistema tridimensional con un giro de "h" grados del plano x-y alrededor del eje z.

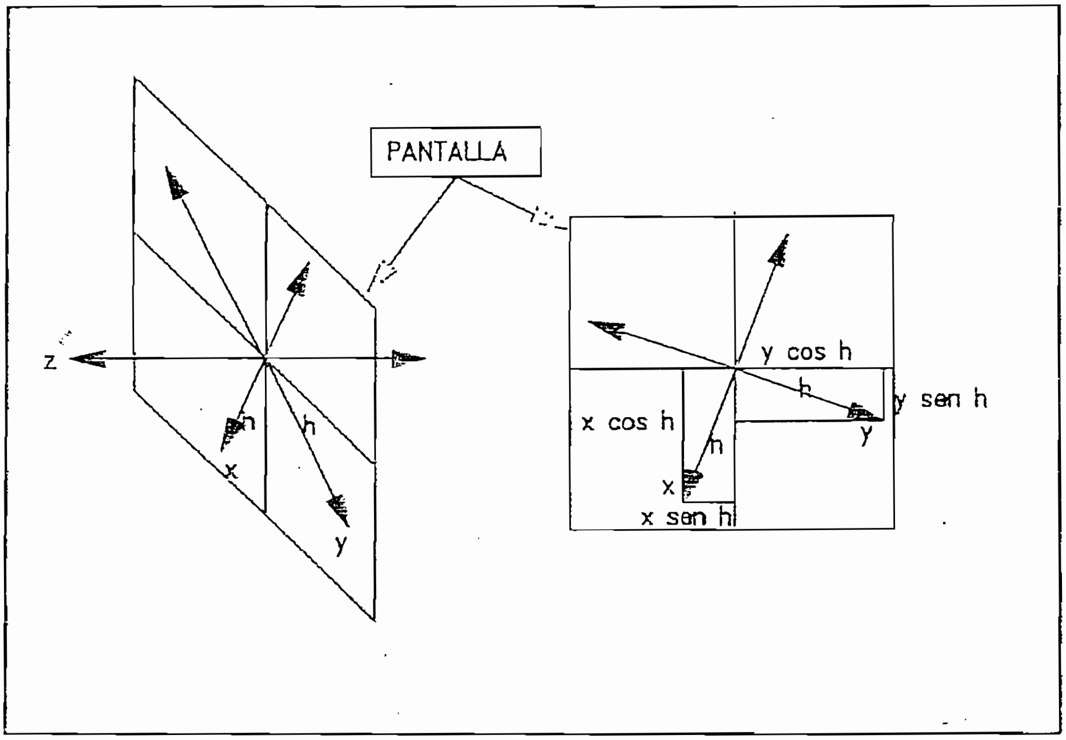


Figura a.2 Sistema de ejes tridimensional y su posición respecto al plano de la pantalla, al girar "h" grados el plano x-y .

Las coordenadas, de cualquier punto del sistema tridimensional móvil, en los ejes estáticos de la pantalla serán :

$$xh = -x \sin(h) + y \cos(h)$$

$$yh = -x \cos(h) - y \sin(h)$$

La figura a.3 muestra el sistema tridimensional con un giro de "v" grados del eje z en direccion perpendicular al plano de la pantalla, desde la posición de la figura a.2 .

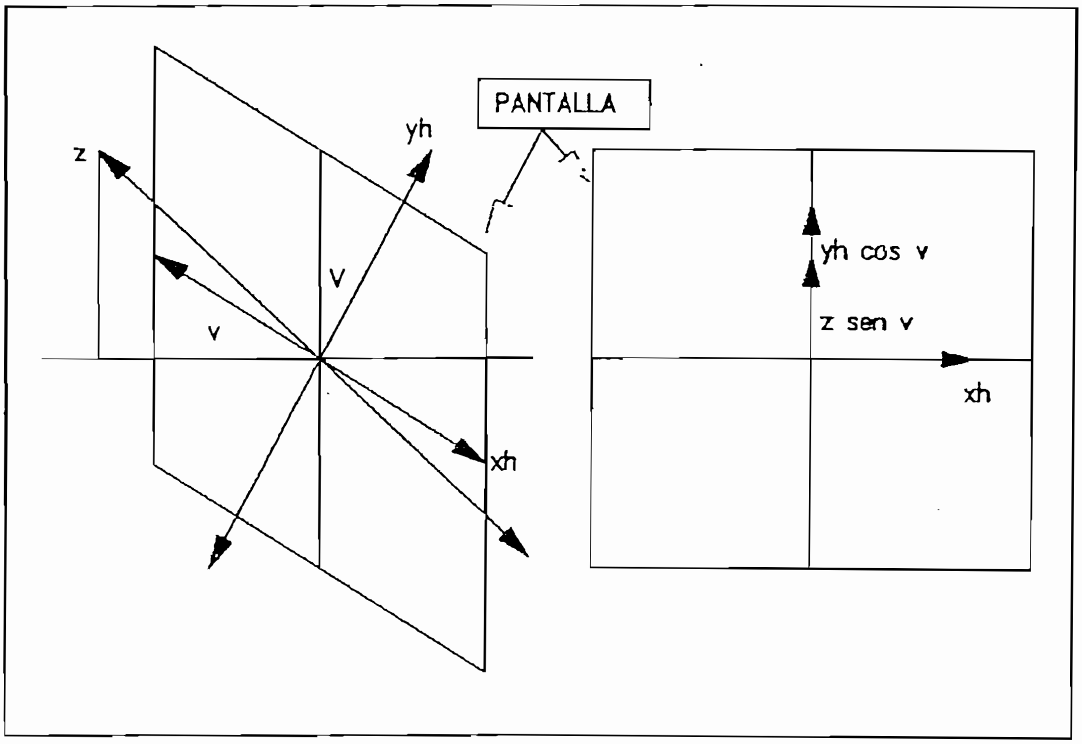


Figura a.3 Sistema de ejes tridimensional y su posición respecto al plano de la pantalla al girar "v" grados el eje z

Las coordenadas, de cualquier punto del sistema tridimensional móvil, en el plano estático de la pantalla serán :

$$xv = xh$$

$$yv = z \sin(v) - yh \cos(v)$$

En suma cualquier punto del sistema de ejes tridimensional, luego de un giro de "h" grados del plno x-y alrededor del eje z y de "v" grados del eje z en dirección perpendicular a plano de la pantalla tendrá las coordenadas X3D y Y3D en el plano estático de la pantalla de graficación.

$$X3D = -x \sin(h) + y \cos(h)$$

$$Y3D = x \cos(h)\cos(v) + y \sin(h)\cos(v) + z \sin(v)$$

ANEXO B

COMANDOS DEL PIZZAS

El programa PIZZAS permite la impresión a través de una impresora de cualquier información gráfica o textual presente en la pantalla.

En el presente trabajo se lo utiliza por sus ventajas en cuanto a rapidez de impresión , que es muy superior a cualquier otra rutina diseñada en trabajos anteriores , y especialmente por su flexibilidad en cuanto a tamaños de impresión y posibilidad de ampliación , reducción y selección de partes o sectores de gráfico.

los principales comandos del programa y su uso son presentados a continuación en el orden que aparecen en el menú del programa.

PRINT.

Permite el envio de la información de la pantalla a la impresora o a un archivo.

COLOR

Permite asignar colores de impresión a cada uno de los colores presentes en la pantalla.

DIVIDE

Permite atrapar en un rectángulo el sector de la pantalla que se desea imprimir

WITDH

Permite la fijación del ancho en que se desea salga impreso el gráfico.

HIGHT

Permite la fijación de la altura en que se desea salga impreso el gráfico.

INDENT

Permite fijar el espacio desde el borde derecho del papel hasta el borde derecho del gráfico.

SKIP

Permite fijar el espacio desde el tope superior de la hoja hasta el borde superior del gráfico.

FORM

Permite fijar las dimensiones del papel y su avance o no hasta el fin de papel despúes de la impresión.

SMOOTH

Permite activar la opción de continuidad en el trazo para la impresión de líneas.

DENSITY

Permite la selección de la calidad de impresión , tiene cuatro niveles: bajo ,medio, alto y super.

SETTINGS

Permite el setear todas las caracteristicas de impresión, grabar estas características o llamar un archivo que cargue estas características al programa.

QUIT

Permite la salida a la forma usual del programa que se estaba usando el momento de activar el PIZZAS.

ANEXO C

PROGRAMA DIGITAL TTF

'MODULO   TTF

```
'Módulo principal contiene información para direccionar a los diferentes subprogramas
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

COMMON SHARED iswprin, itotalipr, itotalpy, itotalcol, itotalfilas, rw, icl, ipr, py, pantallahabilitacion
COMMON SHARED deltat, imarpuntos, imartray
ON ERROR GOTO correccion
CLEAR , , 2500
mini$ = MID$(TIME$, 4, 2): sgi$ = MID$(TIME$, 7, 2)
deltat = .05: imartray = 4: imarpuntos = 500
DIM X(imartray, imarpuntos), y(imartray, imarpuntos), z(imartray, imarpuntos), puntomar(imartray), puntofin(imartray),
mp(imartray), tp(imartray)
DIM seleccion$(1)
HabilitacionDePantalla 3, itotalipr, itotalpy, itotalcol, itotalfilas, rw, cl, ipr, py, pantallahabilitacion
T$ = "U60 L20 U20 e10 R60 nd20 g10 nl60 D20 ne10 L10 nd, c1$, c2$, c3$, c4$, c5$, c6$, c7$, c8$
T$ = "U60 L20 U2050 110 D60 ne10 L20"
F$ = "U80 e10 R60 nd20 g10 nl60 D20 ne10 L30 nd10 110 D20 e10 R20 nd20 g10 nl20 D20 ne10 L10 nd10  110 D20 ne10 L20"
DRAW "c5" + "BD40 BL90" + t$ + "BR80" + t$ + "BR60" + F$
VIEW PRINT 1 * rw TO 5 * rw:PRINT
PRINT "            PROGRAMA DIGITAL PARA ANALISIS DE SISTEMAS NO LINEALES":PRINT
PRINT "   MEDIANTE EL TRAZO DE TRAYECTORIAS DE FASE HASTA EN TRES DIMENSIONES"
DO: A$ = INKEY$: LOOP UNTIL A$ <> ""
DO: VIEW: VIEW PRINT: CLS
HabilitacionDePantalla 0, itotalipr, itotalpy, itotalcol, itotalfilas, rw, cl, ipr, py, pantallahabilitacion
VIEW PRINT 10 * rw TO 20 * rw
PRINT TAB(25); "         MENU PRINCIPAL":PRINT
PRINT TAB(25); "[1]     DIAGRAMA DE BLOQUES "
PRINT TAB(25); "[2]     ECUACIONES DE ESTADO"
PRINT TAB(25); "[3]     IMPRIMIR "
PRINT TAB(25); "[4]     EJEMPLOS "
PRINT TAB(25); "[5]     CAMBIOS EN EL PROGRAMA"
PRINT TAB(25); "[6]     SALIR "
VIEW PRINT 18 * rw TO 19 * rw
pulsedato 33, "SELECCION =", "123456", seleccion$: CLS
CALL HabilitacionDePantalla(iswprin, itotalipr, itotalpy, itotalcol, itotalfilas, rw, cl, ipr, py, pantallahabilitacion)
SELECT CASE seleccion$
CASE "1":DO: CLS
        CAJA0$ = "NH4 NG4 U8 R40 D16 L40 U8 BR40 R40"
        CAJA$ = CAJA0$ + CAJA0$ + CAJA0$
        bomba$ = "l2 h3 u4 e3 r4 f3 d4 g3 l2"
        mas$ = "bl10 bu5 nu3 nd3 nl3 nr3 br10 bd5 bd10 nr2 nl2 bu10"
        flecha$ = "BL200 BU80 R40 NH4 NG4 " + mas$ + "BR16 R40 "
        BLOQUE$ = flecha$ + CAJA$ + "NR40 D32 L288 U28 NG4 NF4 " + bomba$ + "bu4 nh3 ne3 nf3 ng3"
        DRAW "C7" + BLOQUE$
        r$ = "nu2 nd3 e2 r2 f2"
        co$ = "u3 e2 r2 f2 bd4 g2 l2 h2 u3"
        ft$ = "nd6  nr3 u2 r5 bd8 br8 u8 nl3 nr3 bd4"
        c$ = "u1 e2 r3 f1 bd3 g1 l3 h2 u1"
        nl$ = "bd5 u8 f8 u8 br5 d8 r5 bu5"
        DRAW "C2" + "bl60" + r$ + "br120" + co$ + "br75" + nl$ + "br60" + ft$ + "br100" + c$
        VIEW PRINT 15 * rw TO 18 * rw
        PRINT TAB(25); "[1]     SISTEMAS LINEALES "
        PRINT TAB(25); "[2]     SISTEMAS NO LINEALES "
        PRINT TAB(25); "[3]     SALIR "
        VIEW PRINT 19 * rw TO 20 * rw
        pulsedato 33, "SELECCION =", "123", seleccion$: CLS
        SELECT CASE seleccion$
        CASE "1":   CALL SistemasLinealesDB(4, pantallahabilitacion, deltat, imarpuntos, imartray, ordensist, X(), y(), z(),
                    puntomar(), puntofin(), tray, Resc, Rrp, iswFT)
        CASE "2":   CALL SistemasNoLinealesDB(4, pantallahabilitacion, deltat, imarpuntos, imartray, ordensist, X(), y(),
                    z(), puntomar(), puntofin(), tray, Resc, Rrp, iswFT)
        CASE "3"
        END SELECT
        LOOP UNTIL seleccion$ = "3"
CASE "2":DO
       VIEW PRINT 15 * rw TO 18 * rw
       PRINT TAB(20); "REPRESENTACION EN ECUACIONES DE ESTADO ":PRINT
       PRINT TAB(25); "[1]     SISTEMAS LINEALES "
```

```
        PRINT TAB(25); "[2]     SISTEMAS NO LINEALES "
        PRINT TAB(25); "[3]     SALIR "
        VIEW PRINT 19 * rw TO 20 * rw
        pulsedato 33, "SELECCION =", "123", seleccion$: CLS
        SELECT CASE seleccion$
        CASE "1":   CALL EcuacionesDeEstadoL(pantallahabilitacion, deltat, imaxpuntos, imaxtray, ordensist, X(), y(), z(),
                    puntomax(), puntofin(), tray)
                    Resc = 0: Rrp = 0
        CASE "2":   CALL EcuacionesDeEstadoNL(pantallahabilitacion, deltat, imaxpuntos, imaxtray, ordensist, X(), y(), z(),
                    puntomax(), puntofin(), tray)
                    Resc = 0: Rrp = 0
        CASE "3"
        END SELECT
        LOOP UNTIL seleccion$ = "3"
CASE "3": CALL Imprimir(2, pantallahabilitacion, nombregraf$, deltat, ordensist, X(), y(), z(), puntomax(), puntofin(), tray,
          mp(), tp(), Resc, Rrp, iswFT)
          VIEW: VIEW PRINT: CLS
          iswprin = 0
CASE "4": CALL Ejemplos(pantallahabilitacion, nombregraf$, ordensist, X(), y(), z(), puntomax(), puntofin(), tray)
CASE "5": CALL Cambios(pantallahabilitacion, deltat, imaxpuntos, imaxtray)
          REDIM  X(imaxtray,  imaxpuntos),   y(imaxtray,  imaxpuntos),  z(imaxtray,  imaxpuntos),  puntomax(imaxtray),
          puntofin(imaxtray)
CASE "6": CALL Grabar(pantallahabilitacion, deltat, imaxtray, imaxpuntos, nombregraf$, ordensist,  X(), y(), z(), puntomax(),
          puntofin(), tray)
          pulsedato 30, "ESTA SEGURO de SALIR (S/N)?", "SNsn", R$
          IF R$ = "S" OR R$ = "s" THEN EXIT DO
END SELECT
LOOP
VIEW: VIEW PRINT: SCREEN 0: CLS
minf$ = MID$(TIME$, 4, 2): sgf$ = MID$(TIME$, 7, 2)
mini = VAL(mini$): sgi = VAL(sgi$):minf = VAL(minf$): sgf = VAL(sgf$)
PRINT "duracion"; minf - mini; "minutos:"; sgf - sgi; "segundos"
CLEAR , , 1598
END
correccion:
SELECT CASE ERR
CASE 5: pantallahabilitacion = false: RESUME NEXT
CASE 70: INPUT "Quitar la protección del diskette y pulsar una tecla"; z: RESUME
CASE 76: INPUT "CAMBIAR, UNIDAD O VIA DE ACCESO NO VALIDA"; z: RESUME
CASE ELSE: PRINT ERR:INPUT z:RESUME NEXT
END SELECT


                  SUB Cambios (pantallahabilitacion, deltat, imaxpuntos, imaxtray)

ON ERROR GOTO correccion
VIEW PRINT: CLS
PRINT TAB(25); "El programa esta basado en el algoritmo :":PRINT
PRINT TAB(30); "deltax(i) = F(x(i))*deltat":PRINT
PRINT TAB(30); "x(i+1) = x(i) + deltax(i)":PRINT:PRINT
PRINT "deltat es el intervalo de tiempo entre punto y punto de la trayectoria de fase,"
PRINT SPC(7); "mientras más pequeño sea este más preciso es el gráfico ":PRINT
PRINT "i  representa un punto de la trayectoria. imaxpuntos representa el máximo número    de puntos calculados para graficar
una trayectoria, mientras mayor es este      más completo es el gráfico de la trayectoria"
PRINT:PRINT TAB(30); "VALORES ACTUALES":PRINT
PRINT TAB(30); "deltat    ="; deltat
PRINT TAB(30); "maxpuntos ="; imaxpuntos
PRINT TAB(30); "maxtray   ="; imaxtray
pulsedato 25, "CAMBIAR ESTOS VALORES (S/N)", "SsnN", A$
IF A$ = "s" OR A$ = "S" THEN
   CLS
   indato 30, "NUMERO DE TRAYECTORIAS (maxtray)", maxtray, 1, 20
   imaxtray = INT(maxtray)
   SELECT CASE imaxtray
         CASE 1: indato 30, "NUMERO DE  PUNTOS (maxpuntos)", maxpuntos, 10, 2000
         CASE 2 TO 4: indato 30, "NUMERO DE  PUNTOS (maxpuntos)", maxpuntos, 10, 1000
         CASE 5 TO 9: indato 30, "NUMERO DE  PUNTOS (maxpuntos)", maxpuntos, 10, 500
         CASE 10 TO 20: indato 30, "NUMERO DE  PUNTOS (maxpuntos)", maxpuntos, 10, 100
   END SELECT
```

```
   imarpuntos = INT(marpuntos)
   indato 30, 'INTERVALO DE TIEMPO (deltat)', deltat, 0, .1
END IF
END SUB


SUB Grabar        (pantallahabilitacion, deltat,  imartray, imarpuntos, nombregraf$, ordensist, X(),  y(), z(), puntomar(),
                   puntofin(), tray)

ON ERROR GOTO correccion
DO
pulsedato 30, 'GRABAR LOS DATOS ULTIMOS (S/N)?', 'SNsn', A$
IF A$ = 'S' OR A$ = 's' THEN INPUT 'nombre del archivo (d:\subd\nom)'; nombre$
pulsedato 30, 'ESTA SEGURO (S/N)?', 'SNsn', B$
LOOP UNTIL B$ = 'S' OR B$ = 's'
IF A$ = 'S' OR A$ = 's' THEN
   OPEN nombre$ FOR OUTPUT AS #1
   WRITE #1, deltat, imarpuntos, imartray, ordensist, tray
   FOR j = 1 TO tray
      WRITE #1, j, puntomar(j), puntofin(j)
      FOR i = 0 TO puntofin(j)
      WRITE #1, X(j, i), y(j, i), z(j, i)
   NEXT i:NEXT j
   CLOSE #1
END IF
END SUB



SUB HabilitacionDePantalla (iswprin, itotalipr, itotalpy, itotalcol, itotalfilas, rw, cl, ipr, py, pantallahabilitacion)

ON ERROR GOTO correccion
CONST false = 0, TRUE = NOT false
IF iswprin = 3 THEN i = 13 else i=10
DO
pantallahabilitacion = TRUE
i = i - 1
SELECT CASE i
        CASE 2: SCREEN 2        'color
        CASE 3: SCREEN 3        'mono
        CASE 8: SCREEN 8        'color
        CASE 9: SCREEN 9        'color
        CASE 10: SCREEN 10      'mono
        CASE 11: SCREEN 11      'mono
        CASE 12: SCREEN 12      'color
        CASE ELSE: pantallahabilitacion = false
END SELECT
LOOP WHILE pantallahabilitacion = false
SELECT CASE i
CASE 2: itotalipr = 640: itotalpy = 200: itotalcol = 80: itotalfilas = 25
        ipr = 8: py = 8: rw = 1: cl = 1
CASE 3: itotalipr = 720: itotalpy = 335: itotalcol = 80: itotalfilas = 25
        ipr = 9: py = 13.3: rw = 1: cl = 1
CASE 8: WIDTH 80, 25:COLOR 3
        itotalipr = 640: itotalpy = 200: itotalcol = 80: itotalfilas = 25
        ipr = 8: py = 8: rw = 1: cl = 1
CASE 9: WIDTH 80, 25:COLOR 3
        itotalipr = 640: itotalpy = 350: itotalcol = 80: itotalfilas = 25
        ipr = 8: py = 14: rw = 1: cl = 1
CASE 10: WIDTH 80, 25
         itotalipr = 640: itotalpy = 350: itotalcol = 80: itotalfilas = 25
         ipr = 8: py = 14: rw = 1: cl = 1
CASE 11: WIDTH 80, 60:
         itotalipr = 640: itotalpy = 480: itotalcol = 80: itotalfilas = 25
         ipr = 8: py = 14: rw = 2.5: icl = 1
CASE 12: WIDTH 80, 60:COLOR 3
         itotalipr = 640: itotalpy = 480: itotalcol = 80: itotalfilas = 25
         ipr = 8: py = 14: rw = 2.45: icl = 1
CASE ELSE
```

```
END SELECT
END SUB


          SUB indato (col, leyenda$, A, MIN, MAX)

ON ERROR GOTO correccion
CONST false = 0: valido = false
DO
PRINT TAB(col); leyenda$;
INPUT A$
texto$ = ""
longtexto = LEN(A$)
filtroletra$ = "+-.0123456789"
FOR i = 1 TO longtexto
        c$ = MID$(A$, i, 1)
        IF INSTR(filtroletra$, c$) <> 0 THEN
                texto$ = texto$ + c$
        END IF
NEXT i
A = VAL(texto$)
IF MIN <> 0 AND MAX <> 0 THEN
          IF MIN <= A AND A <= MAX THEN
          valido = NOT false
          ELSE
          PRINT TAB(col); "DATO NO VALIDO ------ INGRESE OTRO DATO"
          PRINT TAB(col); "MAYOR o IGUAL QUE "; MIN; "Y MENOR o IGUAL QUE "; MAX
          valido = false
          END IF
ELSE
          valido = NOT false
END IF
LOOP UNTIL valido
END SUB


          SUB pulsedato (col, leyenda$, opcion$, A$)

ON ERROR GOTO correccion
CONST false = 0: valido = false
DO : PRINT TAB(col); leyenda$
     DO : A$ = INKEY$ : LOOP UNTIL A$ <> ""
     IF INSTR(opcion$, A$) <> 0 THEN valido = NOT false
LOOP UNTIL valido
END SUB
```

'MODULO F1

```
'Información para el cálculo de trayectorias a partir del diagrama de bloques
'****************************************************************************

SUB Modelonl        (pantallahabilitacion, isw, cx0, cx1,  cx2, cx3, cx4, cx5, cx6,  cx7, cx8, cx9, cy0,  cy1, cy2, cy3, cy4,
                    cy5, cy6, cy7, cy8, cy9, cx, cy, k1, k2)

ON ERROR GOTO correccion
DIM recta AS STRING * 3
SELECT CASE isw
CASE 1001
        IF cx <= cx1 THEN isw = 11: recta = "r01"
        IF (cx1 < cx) AND (cx <= cx2) THEN recta = "r12"
        IF (cx2 < cx) AND (cx <= cx4) THEN recta = "r24"
        IF (cx4 < cx) AND (cx <= cx5) THEN recta = "r45"
        IF (cx5 < cx) AND (cx <= cx7) THEN recta = "r57"
        IF (cx7 < cx) AND (cx <= cx8) THEN recta = "r78"
        IF cx8 < cx THEN recta = "r89": isw = 1100
CASE 11
        IF cx <= cx1 THEN recta = "r01"
        IF (cx1 < cx) AND (cx <= cx3) THEN recta = "r13"
        IF (cx3 < cx) AND (cx <= cx4) THEN recta = "r34"
        IF (cx4 < cx) AND (cx <= cx5) THEN recta = "r45": isw = 1001
        IF (cx5 < cx) AND (cx <= cx7) THEN recta = "r57": isw = 1001
        IF (cx7 < cx) AND (cx <= cx8) THEN recta = "r78": isw = 1001
        IF cx8 < cx THEN recta = "r89": isw = 1100
CASE 1100
        IF cx < cx1 THEN recta = "r01": isw = 11
        IF (cx1 <= cx) AND (cx < cx2) THEN recta = "r12": isw = 1001
        IF (cx2 <= cx) AND (cx < cx4) THEN recta = "r24": isw = 1001
        IF (cx4 <= cx) AND (cx < cx5) THEN recta = "r45": isw = 1001
        IF (cx5 <= cx) AND (cx < cx6) THEN recta = "r56"
        IF cx6 <= cx AND cx < cx8 THEN recta = "r68"
        IF cx8 <= cx THEN recta = "r89"
CASE ELSE
PRINT "mala sincronizacion de isw": INPUT z
END SELECT
SELECT CASE recta
        CASE "r01"
        CALL PuntoDeRecta(cx, cy, cx0, cy0, cx1, cy1, k1, k2)
        CASE "r12"
        CALL PuntoDeRecta(cx, cy, cx1, cy1, cx2, cy2, k1, k2)
        CASE "r13"
        CALL PuntoDeRecta(cx, cy, cx1, cy1, cx3, cy3, k1, k2)
        CASE "r24"
        CALL PuntoDeRecta(cx, cy, cx2, cy2, cx4, cy4, k1, k2)
        CASE "r34"
        CALL PuntoDeRecta(cx, cy, cx3, cy3, cx4, cy4, k1, k2)
        CASE "r45"
        CALL PuntoDeRecta(cx, cy, cx4, cy4, cx5, cy5, k1, k2)
        CASE "r56"
        CALL PuntoDeRecta(cx, cy, cx5, cy5, cx6, cy6, k1, k2)
        CASE "r57"
        CALL PuntoDeRecta(cx, cy, cx5, cy5, cx7, cy7, k1, k2)
        CASE "r68"
        CALL PuntoDeRecta(cx, cy, cx6, cy6, cx8, cy8, k1, k2)
        CASE "r78"
        CALL PuntoDeRecta(cx, cy, cx7, cy7, cx8, cy8, k1, k2)
        CASE "r89"
        CALL PuntoDeRecta(cx, cy, cx8, cy8, cx9, cy9, k1, k2)
        CASE ELSE
        PRINT "mala logica de programacion"
END SELECT
END SUB
```

```
        SUB PuntoDeRecta (cx, cy, a1, b1, a2, b2, k1, k2)

ON ERROR GOTO correccion
IF (a1 = a2) OR (b1 = b2) THEN
cy = b1: k1 = 0: k2 = b1
ELSE
m = (b1 - b2) / (a1 - a2): B = (a1 * b2 - a2 * b1) / (a1 - a2)
k1 = m: k2 = B
cy = m * cx + B
END IF
END SUB


SUB quitado        (iswprin, pantallahabilitacion, DELTAT, imaxpuntos, imaxtray, ordensist, x(), y(), z(), puntomax(),
                   puntofin(), tray, Resc, Rrp, b2, b1, b0, a3, a2, a1, a0, cx0, cx1, cx2, cx3, cx4, cx5, cx6, cx7, cx8,
                   cx9, cy0, cy1, cy2, cy3, cy4, cy5,cy6, cy7, cy8, cy9)

ON ERROR GOTO correccion
DO
VIEW: CLS
DO
CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion, c1$,
c2$, c3$, c4$, c5$, c6$, c7$, c8$)
VIEW PRINT 10 * rw TO 19 * rw
PRINT TAB(30); "COMPENSACION": PRINT
PRINT TAB(30); "m(s) = kp.e +kd.se ": PRINT
PRINT TAB(25); "[1] PROPROCIONAL"
PRINT TAB(25); "[2] PROPORCIONHAL DERIVATIVA"
PRINT TAB(25); "[3] NINGUNA": PRINT
pulsedato 30, "SELECION =", "123", COMPENSACION$: CLS 2
SELECT CASE COMPENSACION$
CASE "1": indato 30, "constante proporcional", kp, 0, 0
CASE "2": indato 30, "constante proporcional", kp, 0, 0
        indato 30, "constante derivativa", kd, 0, 0: CLS
CASE "3": kd = 0: kp = 1
END SELECT
IF kp = 0 THEN kp = 1
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
'CALCULO DE LAS TRAYECTORIAS
DO
PRINT TAB(32); "ENTRADA AL SISTEMA": PRINT
PRINT TAB(25); "[1] ESCALON"
PRINT TAB(25); "[2] RAMPA"
PRINT TAB(25); "[3] ESCALON Y RAMPA"
PRINT TAB(25); "[4] SIN SEÑAL DE ENTRADA": PRINT
pulsedato 30, "SELECCION =", "1234", ENTRADA$: CLS 2
SELECT CASE ENTRADA$
CASE "1": indato 30, "AMPLITUD DEL ESCALON", Resc, 0, 0
          Rrp = 0
CASE "2": indato 30, "PENDIENTE DE LA RAMPA", Rrp, 0, 0: CLS 2
          Resc = 0
CASE "3": indato 30, "AMPLITUD DEL ESCALON", Resc, 0, 0
          indato 30, "PENDIENTE DE LA RAMPA", Rrp, 0, 0: CLS 2
CASE "4": Resc = 0: Rrp = 0
END SELECT
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
DO
VIEW PRINT 18 * rw TO 24 * rw: CLS : PRINT TAB(30); "CONDICIONES INICIALES"
VIEW PRINT 20 * rw TO 25 * rw: indato 30, "numero de trayectorias", tray, 1, imaxtray: CLS 2
REDIM x(tray, imaxpuntos), y(tray, imaxpuntos), z(tray, imaxpuntos), puntomax(tray), puntofin(tray)
FOR j = 1 TO tray
   CLS 2: PRINT TAB(15 * j MOD 75); "trayectoria"; j
```

```
   indato 30, 'x1 error', x(j, 0), 0, 0
     IF ordensist > 1 THEN : indato 25, 'x2 derivada del error', y(j, 0), 0, 0
     IF ordensist = 3 THEN : indato 25, 'x3  2da. derivada del error', z(j, 0), 0, 0
NEXT j
CLS 2: pulsedato 25, 'LOS DATOS SON CORRECTOS (S/N)', 'sSNn', A$: CLS 2
IF A$ = 'S' OR A$ = 's' THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
PuntoCentro = 0
FOR j = 1 TO tray
   IF PuntoCentro < ABS(x(j, 0)) THEN PuntoCentro = ABS(x(j, 0))
   IF PuntoCentro < ABS(y(j, 0)) THEN PuntoCentro = ABS(y(j, 0))
   IF PuntoCentro < ABS(z(j, 0)) THEN PuntoCentro = ABS(z(j, 0))
NEXT j
IF PuntoCentro = 0 THEN PuntoCentro = 1
PuntoEqui = DELTAT * PuntoCentro / 10000
PuntoFuera = 20 * PuntoCentro
FOR j = 1 TO tray
VIEW PRINT: CLS
VIEW PRINT 4 * rw TO 10 * rw: CLS 2
PRINT TAB(15 * j MOD 75); 'TRAYECTORIA #'; j
PRINT TAB(9); 'Limite Maximo= '; PuntoFuera, 'Variacion Minima = '; PuntoEqui
PRINT '_______________________________________________________________'
PRINT
PRINT TAB(3); 'punto #'; SPC(10); 'X1'; SPC(15); 'X2'; SPC(15); 'X3'
PRINT '_______________________________________________________________'
'COLOR VAL(c3$)
 p = 0: t = 0
 x1 = x(j, 0): puntomax(j) = x1
 x2 = y(j, 0)
 x3 = z(j, 0)
VIEW PRINT 12 * rw TO 13 * rw
LOCATE 12 * rw, 5: PRINT p
LOCATE 12 * rw, 19: PRINT x1
LOCATE 12 * rw, 36: PRINT x2
LOCATE 12 * rw, 53: PRINT x3
 equilibrio = false
 fueraderango = false
isw = 1001: t = 0
DO
t = t + DELTAT
m = kp * x1 + kd * x2
CALL Modulonl(pantallahabilitacion, isw, cx0, cx1, cx2, cx3, cx4, cx5, cx6, cx7, cx8, cx9, cy0, cy1, cy2, cy3, cy4, cy5, cy6,
cy7, cy8, cy9, e, m, k1, k2)
SELECT CASE ordensist
CASE 3: x3punto =   (-(a2 + b1 * k1 * kd) * x3 - (a1 + b0 * k1 * kd + b1 * k1 * kp) * x2 - (a0 + b0 * k1 * kp) * x1 - b0 * k2
                    + a1 * Rrp + a0 * (Rrp * t + Resc)) / (a3 + b2 * k1 * kd)
        x3 = x3 + x3punto * DELTAT: z(j, p + 1) = x3
        x2punto = x3
        x2 = x2 + x2punto * DELTAT: y(j, p + 1) = x2
        x1punto = x2
        x1 = x1 + x1punto * DELTAT: x(j, p + 1) = x1
CASE 2: x2punto =   (-(a1 + b0 * k1 * kd + b1 * k1 * kp) * x2 - (a0 + b0 * k1 * kp) * x1 - b0 * k2 + a1 * Rrp + a0 * (Rrp * t
                    + Resc)) / (a2 + b1 * k1 * kd)
        x2 = x2 + x2punto * DELTAT: y(j, p + 1) = x2
        x1punto = x2
        x1 = x1 + x1punto * DELTAT: x(j, p + 1) = x1
CASE 1: x1punto = (-(a0 + b0 * k1 * kp) * x1 - b0 * k2 + a1 * Rrp + a0 * (Rrp * t + Resc)) / (a1 + b0 * k1 * kd)
        x1 = x1 + x1punto * DELTAT: x(j, p + 1) = x1
        x2 = x1punto: y(j, p + 1) = x2
END SELECT
   p = p + 1: puntofin(j) = p
VIEW PRINT 12 * rw TO 13 * rw
LOCATE 12 * rw, 5: PRINT p
LOCATE 12 * rw, 19: PRINT x1
LOCATE 12 * rw, 36: PRINT x2
LOCATE 12 * rw, 53: PRINT x3
   IF ABS(x1) > puntomax(j) THEN puntomax(j) = ABS(x1)
   IF puntomax(j) >= PuntoFuera THEN
   fueraderango = NOT false
```

```
    PRINT TAB(15); " DIVERGE A FUERA DE RANGO DE GRAFICACION"
    END IF
    IF ABS(x(j, p - 1) - x(j, p)) <= PuntoEqui AND ABS(y(j, p - 1) - y(j, p)) <= PuntoEqui AND ABS(z(j,  p - 1) - z(j, p)) <=
         PuntoEqui THEN
    equilibrio = NOT false
    PRINT TAB(5); "CONVERGE A UN PUNTO DE VARIACION MINIMA DE GRAFICACION"
    END IF
    LOOP UNTIL equilibrio OR fueraderango OR p = imaxpuntos
    'COLOR VAL(c6$)
NEXT j
CALL presente(3, pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), tray)
CALL  HabilitacionDePantalla(4, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion, c1$, c2$,
c3$, c4$, c5$, c6$, c7$, c8$)
VIEW: CLS
pulsedato 25, "NUEVAS CONDICIONES INICIALES (S/N)", "SsNn", A$
LOOP UNTIL A$ = "N" OR A$ = "n"

END SUB



     SUB SistemasLinealesDB   (iswprin, pantallahabilitacion, DELTAT, imaxpuntos, imaxtray, ordensist, x(), y(), z(),
                              puntomax(), puntofin(), tray, Resc, Rrp, iswFT)

ON ERROR GOTO correccion
DO: CLS
CONST false = 0
CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion)
CAJAO$ = "NH4 NG4 U8 R40 D16 L40 U8 BR40 R40"
CAJA$ = CAJAO$ + CAJAO$
bomba$ = "l2 h3 u4 e3 r4 f3 d4 g3 l2"
mas$ = "bl10 bu5 nu3 nd3 nl3 nr3 br10 bd5 bd10 nr2 nl2 bu10"
emu$ = "bu5 nr3 h1 u2 e1 r2 f1 d1 l2 br90 bd3 u3 nh1 e1 r2 f1 nd3 e1 r2 f1 d3 bl110 bd4"
flecha$ = "BL150 BU80 R40 NH4 NG4 " + mas$ + "BR16 R10 " + emu$ + "R40 "
BLOQUE$ = flecha$ + CAJA$ + "NR40 D32 L208 U28 NG4 NF4 " + bomba$ + "bu4 nh3 ne3 nf3 ng3"
DRAW "C7" + BLOQUE$
r$ = "nu2 nd3 e2 r2 f2"
co$ = "u3 e2 r2 f2 bd4 g2 l2 h2 u3"
ft$ = "nd6  nr3 u2 r5 bd6 br8 u8 nl3 nr3 bd4"
c$ = "nl e2 r3 f1 bd3 g1 l3 h2 u1"
DRAW "C5" + "bl60" + r$ + "br120" + co$ + "br75" + ft$ + "br100" + c$
VIEW PRINT 12 * rw TO 16 * rw
PRINT TAB(30); "[1] NUEVOS DATOS"
PRINT TAB(30); "[2] SALIR":PRINT
VIEW PRINT 15 * rw TO 16 * rw
pulsedato 35, "SELECCION =", "12", SIGUIENTE$
IF SIGUIENTE$ = "1" THEN
iswFT = 1
DO
VIEW PRINT 12 * rw TO 17 * rw: CLS 2
indato 30, "ORDEN DEL SISTEMA", ordensist, 1, 3: CLS 2
PRINT TAB(30); "FUNCION DE TRANSFERENCIA":PRINT
PRINT TAB(30); "       b2s^2 +b1s + b0     "
PRINT TAB(30); "FT = ---------------------"
PRINT TAB(30); "     a3s^3 +a2s^2 +a1s +a0"
VIEW PRINT 17 * rw TO 24 * rw
SELECT CASE ordensist
 CASE 1:DO
        indato 40, "b0", b0, 0, 0
        LOOP UNTIL b0 <> 0
        DO
        indato 40, "a1", a1, 0, 0
        LOOP UNTIL a1 <> 0
        indato 40, "a0", a0, 0, 0
        tipo$ = "1pd"
 CASE 2:DO
        indato 40, "b1", b1, 0, 0
        indato 40, "b0", b0, 0, 0
        LOOP UNTIL b0 <> 0 OR b1 <> 0
```

```
        DO
        indato 40, "a2", a2, 0, 0
        LOOP UNTIL a2 <> 0
        indato 40, "a1", a1, 0, 0
        indato 40, "a0", a0, 0, 0
        tipo$ = "2pd"
 CASE 3:DO
        indato 40, "b2", b2, 0, 0
        indato 40, "b1", b1, 0, 0
        indato 40, "b0", b0, 0, 0
        LOOP UNTIL b2 <> 0 OR b1 <> 0 OR b0 <> 0
        DO
        indato 40, "a3", a3, 0, 0
        LOOP UNTIL a3 <> 0
        indato 40, "a2", a2, 0, 0
        indato 40, "a1", a1, 0, 0
        indato 40, "a0", a0, 0, 0
        tipo$ = "3pd"
END SELECT
CLS 2: VIEW PRINT 17 + rw TO 18 + rw: CLS 2
pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
DO
'CALCULO DE TRAYECTORIAS CON CONDICIONES INICIALES INCLUIDO COMPENSACION
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
DO
VIEW PRINT 12 + rw TO 24 + rw
CLS 2: VIEW PRINT 12 + rw TO 13 + rw
pulsedato 30, "COMPENSACION (S/N)", "SsNn", A$: CLS 2
IF A$ = "s" OR A$ = "S" THEN
VIEW PRINT 12 + rw TO 24 + rw: CLS 2
PRINT TAB(30); "m(s) = kp.e +kd.se + ki.e/s"
PRINT
SELECT CASE ordensist
 CASE 3:CLS 2: PRINT TAB(30); "m(s) = kp.e +kd.se "
        PRINT
        PRINT TAB(30); "[1] PROPORCIONAL"
        PRINT TAB(30); "[2] PROPORCIONAL DERIVADA"
        pulsedato 30, "", "12", z$: CLS 2
        indato 27, "CONSTANTE PROPORCIONAL Kp", kp, 0, 0
        IF z$ = "2" THEN indato 27, "CONSTANTE DERIVATIVA Kd", kd, 0, 0
        tipo$ = "3pd"
 CASE ELSE
        PRINT TAB(30); "[1] PROPORCIONAL"
        PRINT TAB(30); "[2] PROPORCIONAL DERIVADA"
        PRINT TAB(30); "[3] PROPORCIONAL INTEGRAL "
        PRINT TAB(30); "[4] PROPORCIONAL INTEGRAL DERIVADA"
        pulsedato 30, "", "1234", z$: CLS 2
        indato 27, "CONSTANTE PROPORCIONAL Kp", kp, 0, 0
        IF z$ = "2" OR z$ = "4" THEN indato 27, "CONSTANTE DERIVATIVA Kd", kd, 0, 0
        IF z$ = "3" OR z$ = "4" THEN indato 27, "CONSTANTE DE INTEGRACION Ki", ki, 0, 0
        IF ki = 0 AND ordensist = 1 THEN tipo$ = "1pd"
        IF ki <> 0 AND ordensist = 1 THEN tipo$ = "1pdi"
        IF ki = 0 AND ordensist = 2 THEN tipo$ = "2pd"
        IF ki <> 0 AND ordensist = 2 THEN tipo$ = "2pdi"
END SELECT
ELSE
kp = 1: kd = 0: ki = 0
END IF
IF kp = 0 THEN kp = 1
CLS 2: VIEW PRINT 12 + rw TO 13 + rw: CLS 2
pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
'CALCULO DE LAS TRAYECTORIAS
''''''''''''''''''''''''''''''''''''''''''''
DO
VIEW PRINT 12 + rw TO 24 + rw: CLS 2
```

```
PRINT TAB(32); "ENTRADA AL SISTEMA":PRINT
PRINT TAB(25); "[1] ESCALON"
PRINT TAB(25); "[2] RAMPA"
PRINT TAB(25); "[3] ESCALON Y RAMPA"
PRINT TAB(25); "[4] SIN SEÑAL DE ENTRADA":PRINT
VIEW PRINT 19 * rw TO 20 * rw: CLS 2
pulsedato 30, "SELECCION =", "1234", ENTRADA$: CLS 2
VIEW PRINT 12 * rw TO 20 * rw: CLS 2
VIEW PRINT 12 * rw TO 15 * rw: CLS 2
SELECT CASE ENTRADA$
CASE "1": indato 30, "AMPLITUD DEL ESCALON", Resc, 0, 0
          Rrp = 0
CASE "2": indato 30, "PENDIENTE DE LA RAMPA", Rrp, 0, 0: CLS 2
          Resc = 0
CASE "3": indato 30, "AMPLITUD DEL ESCALON", Resc, 0, 0
          indato 30, "PENDIENTE DE LA RAMPA", Rrp, 0, 0: CLS 2
CASE "4": Rrp = 0: Resc = 0
END SELECT
CLS 2: VIEW PRINT 12 * rw TO 13 * rw: CLS 2
pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
DO
VIEW PRINT 12 * rw TO 16 * rw: CLS 2
indato 30, "NUMERO DE TRAYECTORIAS", tray, 1, imaxtray: CLS 2
FOR j = 1 TO tray
   PRINT TAB(30); "CONDICIONES INICIALES"
   VIEW PRINT 13 * rw TO 16 * rw: CLS 2
   PRINT TAB(5 * j MOD 60); "TRAYECTORIA #"; j
   indato 30, "error inicial", x(j, 0), 0, 0
   IF ordensist > 1 THEN indato 25, "derivada del error ", y(j, 0), 0, 0
   IF ordensist = 3 THEN indato 25, "2 da. derivada del error ", z(j, 0), 0, 0:CLS 2
NEXT j
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
PuntoCentro = 0
FOR j = 1 TO tray
   IF PuntoCentro < ABS(x(j, 0)) THEN PuntoCentro = ABS(x(j, 0))
   IF PuntoCentro < ABS(y(j, 0)) THEN PuntoCentro = ABS(y(j, 0))
   IF PuntoCentro < ABS(z(j, 0)) THEN PuntoCentro = ABS(z(j, 0))
NEXT j
IF PuntoCentro = 0 THEN PuntoCentro = 1
PuntoEqui = DELTAT * PuntoCentro / 10000
PuntoFuera = 20 * PuntoCentro
FOR j = 1 TO tray
VIEW PRINT 4 * rw TO 10 * rw: CLS
PRINT TAB(15 * j MOD 75); "TRAYECTORIA #"; j
PRINT TAB(3); "Limite Maximo= "; PuntoFuera; SPC(10); "Variacion Minima = "; PuntoEqui
PRINT "________________________________________________________________"
PRINT
PRINT TAB(3); "punto #"; SPC(10); "X1"; SPC(15); "X2"; SPC(15); "X3"
PRINT "________________________________________________________________"
 'COLOR VAL(c3$)
 p = 0: t = 0
 x1 = x(j, 0): puntomax(j) = x1
 x2 = y(j, 0)
 x3 = z(j, 0)
VIEW PRINT 12 * rw TO 13 * rw
LOCATE 12 * rw, 5: PRINT p
LOCATE 12 * rw, 19: PRINT x1
LOCATE 12 * rw, 36: PRINT x2
LOCATE 12 * rw, 53: PRINT x3
equilibrio = false: fueraderango = false
DO
t = t + DELTAT
SELECT CASE tipo$
       CASE "3pd":         x3punto = (-(b2 * kp + b1 * kd + a2) * x3 - (b1 * kp + b0 * kd + a1) * x2 - (b0 * kp + a0) * x1
                                + a1 * Rrp + a0 * (Rrp * t + Resc)) / (b2 * kd + a3)
```

```
        CASE '2pdi':        r3punto = (-(b1 * kp + b0 * kd + a1) * r3 - (b1 * ki + b0 * kp + a0) * r2 - b0 * ki * r1 + a0 *
                                 Rrp) / (b1 * kd + a2)
        CASE '2pd':         r2punto = (-(b1 * kp + b0 * kd +  a1) * r2 - (b0 * kp + a0) * r1 +  a1 * Rrp + a0 * (Rrp *  t +
                                 Resc)) / (b1 * kd + a2)
        CASE '1pdi':        r2punto = (-(b0 * kp + a0) * r2 - b0 * ki * r1 + a0 * Rrp) / (b0 * kd + a1)
        CASE '1pd':         r1punto = (-(b0 * kp + a0) * r1 + a1 * Rrp + a0 * (Rrp * t + Resc)) / (b0 * kd + a1)
END SELECT
SELECT CASE tipo$
CASE '3pd', '2pdi': r3 = r3 + r3punto * DELTAT: z(j, p + 1) = r3
                    r2punto = r3
                    r2 = r2 + r2punto * DELTAT: y(j, p + 1) = r2
                    r1punto = r2
                    r1 = r1 + r1punto * DELTAT: x(j, p + 1) = r1
CASE '2pd', '1pdi': r3 = r2punto: z(j, p + 1) = r3
                    r2 = r2 + r2punto * DELTAT: y(j, p + 1) = r2
                    r1punto = r2
                    r1 = r1 + r1punto * DELTAT: x(j, p + 1) = r1
CASE '1pd':         r2 = r1punto: y(j, p + 1) = r2
                    r1 = r1 + r1punto * DELTAT: x(j, p + 1) = r1
END SELECT
p = p + 1: puntofin(j) = p
VIEW PRINT 12 * rw TO 13 * rw
LOCATE 12 * rw, 5: PRINT p
LOCATE 12 * rw, 19: PRINT r1
LOCATE 12 * rw, 36: PRINT r2
LOCATE 12 * rw, 53: PRINT r3
    IF ABS(r1) > puntomax(j) THEN puntomax(j) = ABS(r1)
    IF puntomax(j) >= PuntoFuera THEN
    fueraderango = NOT false
    PRINT TAB(15); 'DIVERGE A FUERA DEL RANGO DE GRAFICACION'
    PRINT
    END IF
    IF ABS(x(j, p - 1) - x(j, p)) <= PuntoEqui AND ABS(y(j, p - 1) -  y(j, p)) <= PuntoEqui AND ABS(z(j, p - 1) - z(j, p)) <=
         PuntoEqui THEN
    equilibrio = NOT false
    PRINT ; TAB(27); 'CONVERGE A UN PUNTO DE EQUILIBRIO': PRINT
    END IF
    LOOP UNTIL equilibrio OR fueraderango OR p = imaxpuntos: PRINT
NEXT j
IF tipo$ = '2pdi' THEN ordensist = 3
CALL presente(3, pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), tray)
CALL HabilitacionDePantalla(4, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion)
IF tipo$ = '2pdi' THEN ordensist = 2
VIEW: CLS: VIEW PRINT 12 * rw TO 13 * rw
pulsedato 25, 'NUEVAS CONDICIONES INICIALES (S/N)', 'SsNn', A$: CLS
LOOP UNTIL A$ = 'n' OR A$ = 'N'
END IF
LOOP UNTIL SIGUIENTE$ = '2'
IF tipo$ = '2pdi' THEN ordensist = 3
END SUB


SUB SistemasNoLinealesDB       (iswprin, pantallahabilitacion, DELTAT, imaxpuntos, imaxtray, ordensist, x(), y(), z(),
                               puntomax(), puntofin(), tray, Resc, Rrp, iswFT)

ON ERROR GOTO correccion
DO: CLS
CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion, c1$,
c2$, c3$, c4$, c5$, c6$, c7$, c8$)
CAJA0$ = 'NH4 NG4 U8 R40 D16 L40 U8 BR40 R40'
CAJA$ = CAJA0$ + CAJA0$ + CAJA0$
bomba$ = 'l2 h3 u4 e3 r4 f3 d4 g3 l2'
mas$ = 'bl10 bu5 nu3 nd3 nl3 nr3 br10 bd5 bd10 nr2 nl2 bu10'
enu$ = 'bu5 nr3 hl u2 el r2 f1 d1 l2 bd3 br90 u3 nh1 el r2 f1 nd3 el r2 f1 d3 br70 bu1 nu3 f1 r2 el nf1 u3 bd4 bl173 bd4'
flecha$ = 'BL200 BU80 R40 NH4 NG4 ' + mas$ + 'BR16 R10 ' + enu$ + ' r30 '
BLOQUE$ = flecha$ + CAJA$ + 'NR40 D32 L288 U28 NG4 NF4 ' + bomba$ + 'bu4 nh3 ne3 nf3 ng3'
DRAW 'C7' + BLOQUE$
r$ = 'nu2 nd3 e2 r2 f2'
co$ = 'u3 e2 r2 f2 bd4 g2 l2 h2 u3'
```

```
ft$ = "nd6  mr3 u2 r5 bd8 br8 u8 nl3 nr3 bd4"
c$ = "ul e2 r3 fl bd3 gl l3 h2 ul"
nl$ = "bd5 u8 f8 u8 br5 d8 r5 bu5"
DRAW "C5" + "bl60" + r$ + "br120" + co$ + "br75" + nl$ + "br60" + ft$ + "br100" + c$
VIEW PRINT 12 + rw TO 20 + rw: CLS 2
PRINT TAB(30); "[1] NUEVOS DATOS"
PRINT TAB(30); "[2] SALIR":PRINT
VIEW PRINT 15 + rw TO 16 + rw
pulsedato 35, "SELECCION", "12", SIGUIENTE$
IF SIGUIENTE$ = "1" THEN
iswFT = 1
DO
VIEW PRINT 12 + rw TO 24 + rw: CLS 2
PRINT TAB(30); "       b2s^2 +b1s + b0     "
PRINT TAB(30); "FT = ----------------------"
PRINT TAB(30); "      a3s^3 +a2s^2 +a1s +a0 ":PRINT
indato 10, "ORDEN DE LA FUNCION DE TRANSFERENCIA DE LA PARTE LINEAL", ordensist, 1, 3
VIEW PRINT 17 + rw TO 24 + rw: CLS 2
SELECT CASE ordensist
CASE 3: DO
        indato 30, "b2=", b2, 0, 0
        indato 30, "b1=", b1, 0, 0
        indato 30, "b0=", b0, 0, 0
        LOOP UNTIL b2 <> 0 OR b1 <> 0 OR b0 <> 0
        DO
        indato 30, "a3=", a3, 0, 0
        LOOP UNTIL a3 <> 0
        indato 30, "a2=", a2, 0, 0
        indato 30, "a1=", a1, 0, 0
        indato 30, "a0=", a0, 0, 0
CASE 2: DO
        indato 30, "b1=", b1, 0, 0
        indato 30, "b0=", b0, 0, 0
        LOOP UNTIL b1 <> 0 OR b0 <> 0
        DO
        indato 30, "a2=", a2, 0, 0
        LOOP UNTIL a2 <> 0
        indato 30, "a1=", a1, 0, 0
        indato 30, "a0=", a0, 0, 0
CASE 1: DO
        indato 30, "b0=", b0, 0, 0
        LOOP UNTIL b0 <> 0
        DO
        indato 30, "a1=", a1, 0, 0
        LOOP UNTIL a1 <> 0
        indato 30, "a0=", a0, 0, 0
END SELECT
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
CALL znolin(2, pantallahabilitacion, c1$, c2$, c3$, c4$, c5$, c6$, c7$, c8$, cx0, cx1, cx2, cx3, cx4, cx5, cx6, cx7, cx8,
cx9, cy0, cy1, cy2, cy3, cy4, cy5, cy6, cy7, cy8, cy9)
CALL quitado(4, pantallahabilitacion, DELTAT, imaxpuntos, imaxtray, ordensist, x(), y(), z(), puntomax(), puntofin(), itray,
Resc, Krp, b2, b1, b0, a3, a2, a1, a0, cx0, cx1, cx2, cx3, cx4, cx5, cx6, cx7, cx8, cx9, cy0, cy1, cy2, cy3, cy4, cy5,
cy6,cy7, cy8, cy9)
END IF
LOOP UNTIL SIGUIENTE$ = "2"
END SUB


SUB znolin          (iswprin, pantallahabilitacion, c1$, c2$, c3$, c4$, c5$, c6$, c7$, c8$, cx0, cx1, cx2, cx3, cx4, cx5,
                    cx6, cx7, cx8, cx9, cy0, cy1, cy2, cy3, cy4, cy5, cy6, cy7, cy8, cy9)

ON ERROR GOTO correccion
CONST false = 0
DO
seleccion = false
HabilitacionDePantalla iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, icl, ipx, py, pantallahabilitacion
```

```
CLS: p = 0
limvert = itotalpy / 2: pasopy = itotalpy / 2 - py
limhorz = 3 * itotalipr / 4: pasoipr = itotalipr / 4
DO
FOR j = 0 TO limvert STEP pasopy
FOR i = 0 TO limhorz STEP pasoipr
VIEW (i + ipr, j + 2 * py + 2)-(i + pasoipr - ipr, j + pasopy - 2 * py), , 1
WINDOW (-7, 7)-(7, -7)
LINE (0, 7)-(0, -7), , , &H8888
LINE (7, 0)-(-7, 0), , , &H8888
p = (p + 1) MOD 17
SELECT CASE p
        CASE 1: LINE (-5, -5)-(0, -5): LINE -(0, 5): LINE -(5, 5)
        CASE 2: LINE (-5, -5)-(-1, -5): LINE -(-1, 0): LINE -(1, 0): LINE -(1, 5): LINE -(5, 5)
        CASE 3: LINE (5, 6)-(0, 5): LINE -(0, -5): LINE -(-5, -6)
        CASE 4: LINE (5, 6)-(1, 5): LINE -(1, 0): LINE -(-1, 0): LINE -(-1, -5): LINE -(-5, -6)
        CASE 5: LINE (5, 5)-(2.5, 5): LINE -(-2.5, -5): LINE -(-5, -5)
        CASE 6: LINE (5, 5)-(2.5, 5): LINE -(1, 0): LINE -(-1, 0): LINE -(-2.5, -5): LINE -(-5, -5)
        CASE 7: LINE (-5, -6)-(-2.5, -5): LINE -(2.5, 5): LINE -(5, 6)
        CASE 8: LINE (-5, -6)-(-2.5, -5): LINE -(-1, 0): LINE -(1, 0): LINE -(2.5, 5): LINE -(5, 6)
        CASE 9: LINE (-5, -5)-(1, -5): LINE -(1, 5): LINE -(5, 5): LINE -(-1, 5): LINE -(-1, -5)
        CASE 10:    LINE (-5, -5)-(-1, -5): LINE -(-1,  0): LINE -(2, 0): LINE -(2, 5): LINE -(5, 5): LINE -(1, 5): LINE -(1,
                    0): LINE -(-2, 0): LINE -(-2, -5)
        CASE 11: LINE (-5, -5)-(-.5, -5): LINE -(2, 5): LINE -(5, 5): LINE -(.5, 5): LINE -(-2, -5)
        CASE 12:    LINE (-5, -5)-(-2.5, -5): LINE -(-1, 0): LINE -(2, 0): LINE -(3.5, 5): LINE -(5, 5): LINE -(2.5, 5): LINE
                    -(1, 0): LINE -(-2, 0): LINE -(-3.5, -5)
        CASE 13: LINE (-4, -6)-(-2, -5): LINE -(2, -3): LINE -(2, 5): LINE -(4, 6): LINE -(-2, 3): LINE -(-2, -5)
        CASE 14:    LINE (-5,  -6)-(-3, -5): LINE -(-2, -4): LINE  -(-1, 0): LINE -(2,  0): LINE -(3, 5):  LINE -(5, 6): LINE
                    -(2, 4): LINE -(1, 0): LINE -(-2, 0): LINE -(-3, -5)
        CASE 15: LINE (-6, -6)-(-3, -4): LINE -(-2, -1): LINE -(2, 1): LINE -(3, 4): LINE -(6, 6)
        CASE 16:    LINE (-6, -6)-(-3,  -4): LINE -(-2, -1): LINE  -(2, 1): LINE -(3,  4): LINE -(6,  6): LINE -(3, 4):  LINE
                    -(1.5, 3): LINE -(.6, .3): LINE -(-2, -1): LINE -(-3, -4): LINE -(-1.5, -3): LINE -(-.6, -.3)
        CASE ELSE
END SELECT
IF p = 8 THEN
        VIEW PRINT 1 * rw TO 3 * rw
        PRINT TAB(37); "RELES"
        PRINT "[1] ideal"; SPC(11); "[2] con Z.M"; SPC(9); "[3] con Pend"; SPC(8); "[4] con Z.M y Pend"
        VIEW PRINT 12 * rw TO 15 * rw
        PRINT TAB(34); "SATURACION"
        PRINT "[5] ideal"; SPC(11); "[6] con Z.M"; SPC(9); "[7] con Pend"; SPC(8); "[8] con Z.M y Pend"
        VIEW PRINT 23 * rw TO 24 * rw
        pulsedato 22, "SELECCIONE CON UN NUMERO  (M = MAS)", "12345678Mm", n1$
        IF n1$ <> "m" AND n1$ <> "M" THEN seleccion = NOT false
        VIEW: CLS
END IF
IF p = 16 THEN
        VIEW PRINT 1 * rw TO 3 * rw
        PRINT TAB(34); "HISTERISIS"
        PRINT "[1] ideal"; SPC(10); "[2] con Z.M"; SPC(8); "[3] con Pend"; SPC(7); "[4] con Z.M y Pend"
        VIEW PRINT 12 * rw TO 15 * rw
        PRINT TAB(27); "SATURACION CON HISTERISIS"
        PRINT "[5] ideal"; SPC(10); "[6] con Z.M"; SPC(8); "[7] con Pend"; SPC(7); "[8] con Z.M y Pend"
        VIEW PRINT 23 * rw TO 24 * rw
        pulsedato 22, "SELECCIONE CON UN NUMERO  (M = MAS)", "12345678Mm", n1$
        IF n1$ <> "m" AND n1$ <> "M" THEN seleccion = NOT false
        VIEW: CLS : p = 0: n1$ = "1" + n1$
END IF
IF seleccion THEN EXIT FOR
NEXT i
IF seleccion THEN EXIT FOR
NEXT j
LOOP UNTIL seleccion = NOT false
CALL zvalores(2, pantallahabilitacion,  n1$, cx0, cx1, cx2, cx3, cx4, cx5, cx6,  cx7, cx8, cx9, cy0, cy1, cy2, cy3, cy4, cy5,
cy6, cy7, cy8, cy9)
CLS
VIEW PRINT
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
```

```
LOOP UNTIL CORRECTO
END SUB


SUB ivalores        (iswprin, pantallahabilitacion, al$, cr0, cr1, cr2, cr3, cr4, cr5, cr6, cr7, cr8, cr9, cy0, cy1, cy2,
                    cy3, cy4, cy5, cy6, cy7, cy8, cy9)

ON ERROR GOTO correccion
VIEW: CLS
  cero$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 u5 l5 d5 r5 bf3"
   uno$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 u5 ng2 bd5 bf3"
   dos$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 l5 u2 r5 u3 l5 bd5 br5 bf3"
  tres$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 nl5 u3 nl5 u2 nl5 bd5 bf3"
cuatro$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 u5 bl5 d3 r5 d2 bf3"
 cinco$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 bu5 l5 d2 r5 d3 nl5 bf3"
  seis$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 bu5 l5 d2 nr5 d3 r5 nu3 bf3"
 siete$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 u5 nl5 d2 nl3 d3 bf3"
  ocho$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 u5 l5 d2 nr5 d3 r5 bf3"
 nueve$ = "bh3 bl15 u7 r4 f1 d2 g1 l4 d3 br15 u5 l5 d3 r5 d2 nl5 bf3"
CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, icl, ipx, py, pantallahabilitacion,
c1$, c2$, c3$, c4$, c5$, c6$, c7$, c8$)
VIEW (1, itotalpy - 4 * py)-(itotalipx / 4, itotalpy - py), 1, 1
VIEW (3 * ipx, 3 * py)-(itotalipx - ipx, itotalpy - 4.5 * py), , 1
WINDOW (-7, 7)-(7, -7)
LINE (0, 6)-(0, -6), , , &H8888
LINE (6, 0)-(-6, 0), , , &H8888
VIEW PRINT 1 * rw TO 3 * rw
PRINT TAB(20); "CARACTERISTICA DE TRANSFERENCIA"
PRINT TAB(20); "    DEL ELEMENTO NO LINEAL     "
VIEW PRINT 23 * rw TO 23 * rw
SELECT CASE al$
       CASE "1": LINE (-5, -5)-(0, -5): DRAW uno$
                 LINE -(0, 5): DRAW dos$
                 LINE -(5, 5)
                 CLS 2: indato 0, "y1=", cy0, 0, 0
                 CLS 2: indato 0, "y2=", cy9, 0, 0
                 cr0 = -1: cr1 = 0: cr2 = 0: cr3 = 0: cr4 = 0
                 cr5 = 0: cr6 = 0: cr7 = 0: cr8 = 0: cr9 = 1
                 cy0 = cy0: cy1 = cy0: cy2 = 0: cy3 = cy0: cy4 = 0
                 cy5 = 0: cy6 = cy9: cy7 = 0: cy8 = cy9: cy9 = cy9
       CASE "2":  LINE (-5, -5)-(-1, -5): DRAW uno$: LINE -(-1, 0): DRAW dos$: LINE -(1, 0): DRAW tres$: LINE -(1, 5): DRAW
                  cuatro$: LINE -(5, 5)
                 CLS 2: indato 0, "y1", cy0, 0, 0
                 CLS 2: indato 0, "x2", cr1, 0, 0
                 CLS 2: indato 0, "x3", cr8, 0, 0
                 CLS 2: indato 0, "y4", cy9, 0, 0
                 cr0 = 2 * cr1: cr1 = cr1: cr2 = cr1: cr3 = cr1: cr4 = cr1
                 cr5 = cr8: cr6 = cr8: cr7 = cr8: cr8 = cr8: cr9 = 2 * cr8
                 cy0 = cy0: cy1 = cy0: cy2 = 0: cy3 = cy0: cy4 = 0
                 cy5 = 0: cy6 = cy9: cy7 = 0: cy8 = cy9: cy9 = cy9
       CASE "3": PSET (-5, -6): DRAW uno$
                 LINE -(0, -5): DRAW dos$
                 LINE -(0, 5): DRAW tres$
                 LINE -(5, 6): DRAW cuatro$
                 CLS 2: indato 0, "x1", cr0, 0, 0
                 CLS 2: indato 0, "y1", cy0, 0, 0
                 CLS 2: indato 0, "y2", cy1, 0, 0
                 CLS 2: indato 0, "y3", cy8, 0, 0
                 CLS 2: indato 0, "x4", cr9, 0, 0
                 CLS 2: indato 0, "y4", cy9, 0, 0
                 cr0 = cr0: cr1 = 0: cr2 = 0: cr3 = 0: cr4 = 0
                 cr5 = 0: cr6 = 0: cr7 = 0: cr8 = 0: cr9 = cr9
                 cy0 = cy0: cy1 = cy1: cy2 = 0: cy3 = cy1: cy4 = 0
                 cy5 = 0: cy6 = cy8: cy7 = 0: cy8 = cy8: cy9 = cy9
       CASE "4": LINE (-5, -6)-(-5, -6): DRAW uno$
                 LINE -(-1, -5): DRAW dos$
                 LINE -(-1, 0): DRAW tres$
                 LINE -(1, 0): DRAW cuatro$
```

```
          LINE -(1, 5): DRAW cinco$
          LINE -(5, 6): DRAW seis$
          CLS 2: indato 0, "x1", cx0, 0, 0
          CLS 2: indato 0, "y1", cy0, 0, 0
          CLS 2: indato 0, "y2", cy1, 0, 0
          CLS 2: indato 0, "x3", cx1, 0, 0
          CLS 2: indato 0, "x4", cx8, 0, 0
          CLS 2: indato 0, "y5", cy8, 0, 0
          CLS 2: indato 0, "x6", cx9, 0, 0
          CLS 2: indato 0, "y6", cy9, 0, 0
          cx0 = cx0: cx1 = cx1: cx2 = cx1: cx3 = cx1: cx4 = cx1
          cx5 = cx8: cx6 = cx8: cx7 = cx8: cx8 = cx8: cx9 = cx9
          cy0 = cy0: cy1 = cy1: cy2 = 0: cy3 = cy1: cy4 = 0
          cy5 = 0: cy6 = cy8: cy7 = 0: cy8 = cy8: cy9 = cy9
CASE "5": LINE (-5, -5)-(-2.5, -5): DRAW uno$
          LINE -(2.5, 5): DRAW dos$
          LINE -(5, 5)
          CLS 2: indato 0, "x1", cx1, 0, 0
          CLS 2: indato 0, "y1", cy1, 0, 0
          CLS 2: indato 0, "x2", cx8, 0, 0
          CLS 2: indato 0, "y2", cy8, 0, 0
          cx0 = 2 * cx1: cx1 = cx1: cx2 = cx1: cx3 = cx1: cx4 = cx1
          cx5 = cx8: cx6 = cx8: cx7 = cx8: cx8 = cx8: cx9 = 2 * cx8
          cy0 = cy1: cy1 = cy1: cy2 = cy1: cy3 = cy1: cy4 = cy1
          cy5 = cy8: cy6 = cy8: cy7 = cy8: cy8 = cy8: cy9 = cy8
CASE "6": LINE (-5, -5)-(-2.5, -5): DRAW uno$
          LINE -(-1, 0): DRAW dos$
          LINE -(1, 0): DRAW tres$
          LINE -(2.5, 5): DRAW cuatro$
          LINE -(5, 5)
          CLS 2: indato 0, "x1", cx1, 0, 0
          CLS 2: indato 0, "y1", cy1, 0, 0
          CLS 2: indato 0, "x2", cx2, 0, 0
          CLS 2: indato 0, "x3", cx5, 0, 0
          CLS 2: indato 0, "x4", cx8, 0, 0
          CLS 2: indato 0, "y4", cy8, 0, 0
          cx0 = 2 * cx1: cx1 = cx1: cx2 = cx2: cx3 = cx1: cx4 = cx2
          cx5 = cx5: cx6 = cx8: cx7 = cx5: cx8 = cx8: cx9 = 2 * cx8
          cy0 = cy1: cy1 = cy1: cy2 = 0: cy3 = cy1: cy4 = 0
          cy5 = 0: cy6 = cy8: cy7 = 0: cy8 = cy8: cy9 = cy8
CASE "7": PSET (-5, -6): DRAW uno$
          LINE -(-2.5, -5): DRAW dos$
          LINE -(2.5, 5): DRAW tres$
          LINE -(5, 6): DRAW cuatro$
          CLS 2: indato 0, "x1", cx0, 0, 0
          CLS 2: indato 0, "y1", cy0, 0, 0
          CLS 2: indato 0, "x2", cx1, 0, 0
          CLS 2: indato 0, "y2", cy1, 0, 0
          CLS 2: indato 0, "x3", cx8, 0, 0
          CLS 2: indato 0, "y3", cy8, 0, 0
          CLS 2: indato 0, "x4", cx9, 0, 0
          CLS 2: indato 0, "y4", cy9, 0, 0
          cx0 = cx0: cx1 = cx1: cx2 = cx1: cx3 = cx1: cx4 = cx1
          cx5 = cx8: cx6 = cx8: cx7 = cx8: cx8 = cx8: cx9 = cx9
          cy0 = cy0: cy1 = cy1: cy2 = cy1: cy3 = cy1: cy4 = cy1
          cy5 = cy8: cy6 = cy8: cy7 = cy8: cy8 = cy8: cy9 = cy9
CASE "8": LINE (-5, -6)-(-5, -6): DRAW uno$
          LINE -(-2.5, -5): DRAW dos$
          LINE -(-1, 0): DRAW tres$
          LINE -(1, 0): DRAW cuatro$
          LINE -(2.5, 5): DRAW cinco$
          LINE -(5, 6): DRAW seis$
          CLS 2: indato 0, "x1", cx0, 0, 0
          CLS 2: indato 0, "y1", cy0, 0, 0
          CLS 2: indato 0, "x2", cx1, 0, 0
          CLS 2: indato 0, "y2", cy1, 0, 0
          CLS 2: indato 0, "x3", cx2, 0, 0
          CLS 2: indato 0, "x4", cx5, 0, 0
          CLS 2: indato 0, "x5", cx8, 0, 0
```

```
          cx5 = cx5: cx6 = cx6: cx7 = cx7: cx8 = cx8: cx9 = 2 * cx8
          cy0 = cy1: cy1 = cy1: cy2 = 0: cy3 = cy1: cy4 = 0
          cy5 = 0: cy6 = cy8: cy7 = 0: cy8 = cy8: cy9 = cy8
CASE '15': LINE (-4, -6)-(-4, -6): DRAW uno$
           LINE -(-2, -5): DRAW dos$
           LINE -(2, -3): DRAW tres$
           LINE -(2, 5): DRAW cinco$
           LINE -(4, 6): DRAW seis$:
           LINE -(-2, 3): DRAW cuatro$
           LINE -(-2, -5)
          CLS 2: indato 0, 'x1', cx0, 0, 0
          CLS 2: indato 0, 'y1', cy0, 0, 0
          CLS 2: indato 0, 'x2', cx1, 0, 0
          CLS 2: indato 0, 'y2', cy1, 0, 0
          CLS 2: indato 0, 'x3', cx3, 0, 0
          CLS 2: indato 0, 'y3', cy3, 0, 0
          CLS 2: indato 0, 'y4', cy2, 0, 0
          CLS 2: indato 0, 'y5', cy4, 0, 0
          CLS 2: indato 0, 'x6', cx5, 0, 0
          CLS 2: indato 0, 'y6', cy5, 0, 0
          cx0 = cx0: cx1 = cx1: cx2 = cx1: cx3 = cx3: cx4 = cx3
          cx5 = cx5: cx6 = cx5: cx7 = cx5: cx8 = cx5: cx9 = cx5
          cy0 = cy0: cy1 = cy1: cy2 = cy2: cy3 = cy3: cy4 = cy4
          cy5 = cy5: cy6 = cy5: cy7 = cy5: cy8 = cy5: cy9 = cy5
CASE '16': LINE (-5, -6)-(-5, -6): DRAW cero$
           LINE -(-3, -5): DRAW uno$
           LINE -(-2, -4.5): DRAW dos$
           LINE -(-1, 0): DRAW cuatro$
           LINE -(2, 0): DRAW seis$
           LINE -(3, 5): DRAW ocho$
           LINE -(5, 6): DRAW nueve$
           LINE -(2, 4.5): DRAW siete$
           LINE -(1, 0): DRAW cinco$
           LINE -(-2, 0): DRAW tres$: LINE -(-3, -5)
          CLS 2: indato 0, 'x0', cx0, 0, 0
          CLS 2: indato 0, 'y0', cy0, 0, 0
          CLS 2: indato 0, 'x1', cx1, 0, 0
          CLS 2: indato 0, 'y1', cy1, 0, 0
          CLS 2: indato 0, 'x2', cx3, 0, 0
          CLS 2: indato 0, 'y2', cy3, 0, 0
          CLS 2: indato 0, 'x3', cx2, 0, 0
          CLS 2: indato 0, 'x4', cx4, 0, 0
          CLS 2: indato 0, 'x5', cx5, 0, 0
          CLS 2: indato 0, 'x6', cx7, 0, 0
          CLS 2: indato 0, 'x7', cx6, 0, 0
          CLS 2: indato 0, 'y7', cy6, 0, 0
          CLS 2: indato 0, 'x8', cx8, 0, 0
          CLS 2: indato 0, 'y8', cy8, 0, 0
          CLS 2: indato 0, 'x9', cx9, 0, 0
          CLS 2: indato 0, 'y9', cy9, 0, 0
          cx0 = cx0: cx1 = cx1: cx2 = cx2: cx3 = cx3: cx4 = cx4
          cx5 = cx5: cx6 = cx6: cx7 = cx7: cx8 = cx8: cx9 = cx9
          cy0 = cy0: cy1 = cy1: cy2 = 0: cy3 = cy1: cy4 = 0
          cy5 = 0: cy6 = cy6: cy7 = 0: cy8 = cy8: cy9 = cy9
CASE '17': LINE (-6, -6)-(-6, -6): DRAW uno$
           LINE -(-3, -4): DRAW dos$
           LINE -(-2, -1): DRAW tres$
           LINE -(2, 1): DRAW cuatro$
           LINE -(3, 4): DRAW cinco$
           LINE -(6, 6): DRAW seis$
          CLS 2: indato 0, 'x1', cx0, 0, 0
          CLS 2: indato 0, 'y1', cy0, 0, 0
          CLS 2: indato 0, 'x2', cx1, 0, 0
          CLS 2: indato 0, 'y2', cy1, 0, 0
          CLS 2: indato 0, 'x3', cx2, 0, 0
          CLS 2: indato 0, 'y3', cy2, 0, 0
          CLS 2: indato 0, 'x4', cx5, 0, 0
          CLS 2: indato 0, 'y4', cy5, 0, 0
          CLS 2: indato 0, 'x5', cx6, 0, 0
```

```
            CLS 2: indato 0, 'y5', cy6, 0, 0
            CLS 2: indato 0, 'x6', cx9, 0, 0
            CLS 2: indato 0, 'y6', cy9, 0, 0
            cx0 = cx0: cx1 = cx1: cx2 = cx2: cx3 = cx1: cx4 = cx2
            cx5 = cx5: cx6 = cx6: cx7 = cx5: cx8 = cx6: cx9 = cx9
            cy0 = cy0: cy1 = cy1: cy2 = cy2: cy3 = cy1: cy4 = cy2
            cy5 = cy5: cy6 = cy6: cy7 = cy5: cy8 = cy6: cy9 = cy9
      CASE '18': LINE (-6, -6)-(-6, -6): DRAW cero$
            LINE -(-3, -4): DRAW uno$
            LINE -(-2, -1): DRAW tres$
            LINE -(2, 1): DRAW seis$
            LINE -(3, 4): DRAW ocho$
            LINE -(6, 6): DRAW nueve$
            LINE -(3, 4): LINE -(1.5, 3): DRAW siete$
            LINE -(.6, .3): DRAW cinco$
            LINE -(-2, -1): LINE -(-3, -4)
            LINE -(-1.5, -3): DRAW dos$
            LINE -(-.6, -.3): DRAW cuatro$
            CLS 2: indato 0, 'x0', cx0, 0, 0
            CLS 2: indato 0, 'y0', cy0, 0, 0
            CLS 2: indato 0, 'x1', cx1, 0, 0
            CLS 2: indato 0, 'y1', cy1, 0, 0
            CLS 2: indato 0, 'x2', cx2, 0, 0
            CLS 2: indato 0, 'y2', cy2, 0, 0
            CLS 2: indato 0, 'x3', cx3, 0, 0
            CLS 2: indato 0, 'y3', cy3, 0, 0
            CLS 2: indato 0, 'x4', cx4, 0, 0
            CLS 2: indato 0, 'y4', cy4, 0, 0
            CLS 2: indato 0, 'x5', cx5, 0, 0
            CLS 2: indato 0, 'y5', cy5, 0, 0
            CLS 2: indato 0, 'x6', cx6, 0, 0
            CLS 2: indato 0, 'y6', cy6, 0, 0
            CLS 2: indato 0, 'x7', cx7, 0, 0
            CLS 2: indato 0, 'y7', cy7, 0, 0
            CLS 2: indato 0, 'x8', cx8, 0, 0
            CLS 2: indato 0, 'y8', cy8, 0, 0
            CLS 2: indato 0, 'x9', cx9, 0, 0
            CLS 2: indato 0, 'y9', cy9, 0, 0
            cx0 = cx0: cx1 = cx1: cx2 = cx2: cx3 = cx3: cx4 = cx4
            cx5 = cx5: cx6 = cx6: cx7 = cx7: cx8 = cx8: cx9 = cx9
            cy0 = cy0: cy1 = cy1: cy2 = cy2: cy3 = cy3: cy4 = cy4
            cy5 = cy5: cy6 = cy6: cy7 = cy7: cy8 = cy8: cy9 = cy9
CASE ELSE
END SELECT
END SUB
```

```
                                   'MODULO ECU

'Cálculo de trayectorias a partir de las ecuaciones de estado
'*****************************************************************************
```

```
SUB EcuacionesDeEstadoL          (pantallahabilitacion, DELTAT, imarpuntos, imartray, ordensist, x(), y(), z(), puntomar(),
                                  puntofin(), tray)

ON ERROR GOTO correccion
DIM coef(3, 4)
DO: CLS
CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, RW, cl, ipx, py, pantallahabilitacion) VIEW
PRINT 12 * RW TO 18 * RW
PRINT TAB(29); "SISTEMAS LINEALES":PRINT
PRINT TAB(30); "[1] NUEVOS DATOS"
PRINT TAB(30); "[2] SALIR":PRINT
pulsedato 29, "SELECCION =", "12", SIGUIENTE$: CLS 2
IF SIGUIENTE$ = "1" THEN
iswFT = 0
CONST false = 0: TRUE = NOT false
PuntoCentro = 0: equilibrio = false: fueraderango = false
itotalECUAX1 = 1: itotalecuax2 = 1: itotalecuax3 = 1
NUMECUACIONES = 0
indato 30, "ORDEN DEL SISTEMA ", ordensist, 1, 3: CLS 2
   SELECT CASE ordensist
      CASE 1:  itotalecuax2 = 0: itotalecuax3 = 0
      CASE 2:  itotalecuax3 = 0
      CASE ELSE
  END SELECT
FOR i = 1 TO ordensist
   SELECT CASE i
      CASE 1: varest$ = "x1"
      CASE 2: varest$ = "x2"
      CASE 3: varest$ = "x3"
   END SELECT
      DO
      VIEW PRINT 15 * RW TO 24 * RW: CLS 2
      PRINT TAB(25); "ECUACION  #"; i; "  para  "; varest$; "punto": PRINT
      SELECT CASE ordensist
         CASE 1: PRINT TAB(20); "x1punto = a11 * x1  +  b1"
              VIEW PRINT 18 * RW TO 24 * RW: CLS 2
              indato 30, "coeficiente  a11 =", coef(1, 1), 0, 0
              indato 30, "coeficiente   b1 =", coef(1, 4), 0, 0: CLS 2
        CASE 2: IF varest$ = "x1" THEN
              PRINT TAB(20); "x1punto = a11 * x1  +  a12 * x2  +  b1"
              VIEW PRINT 18 * RW TO 24 * RW: CLS 2
                 indato 30, "coeficiente  a11=", coef(1, 1), 0, 0
                 indato 30, "coeficiente  a12=", coef(1, 2), 0, 0
                 indato 30, "coeficiente   b1=", coef(1, 4), 0, 0: CLS 2
            ELSE
              PRINT TAB(20); "x2punto = a21 * x1  +  a22 * x2  +  b2"
              VIEW PRINT 18 * RW TO 24 * RW: CLS 2
                 indato 30, "coeficiente   a21=", coef(2, 1), 0, 0
                 indato 30, "coeficiente   a22=", coef(2, 2), 0, 0
                 indato 30, "coeficiente    b2=", coef(2, 4), 0, 0: CLS 2
           END IF
        CASE 3
           SELECT CASE varest$
              CASE "x1"
                    PRINT TAB(20); "x1punto = a11 * x1  +  a12 * x2  +  a13 * x3  +  b1"
                    VIEW PRINT 17 * RW TO 24 * RW: CLS 2
                    indato 30, "coeficiente   a11=", coef(1, 1), 0, 0
                    indato 30, "coeficiente   a12=", coef(1, 2), 0, 0
                    indato 30, "coeficiente   a13=", coef(1, 3), 0, 0
                    indato 30, "coeficiente    b1=", coef(1, 4), 0, 0: CLS 2
              CASE "x2"
                    PRINT TAB(20); "x2punto = a21 * x1  +  a22 * x2  +  a23 * x3  +  b2"
                    VIEW PRINT 17 * RW TO 24 * RW: CLS 2
                    indato 30, "coeficiente   a21=", coef(2, 1), 0, 0
                    indato 30, "coeficiente   a22=", coef(2, 2), 0, 0
                    indato 30, "coeficiente   a23=", coef(2, 3), 0, 0
                    indato 30, "coeficiente    b2=", coef(2, 4), 0, 0: CLS 2
```

```
            CASE 'x3'
                PRINT TAB(20); "x3punto = a31 * x1  +  a32 * x2  +  a33 * x3  +  b3"
                VIEW PRINT 17 * RW TO 24 * RW: CLS 2
                indato 30, "coeficiente  a31=", coef(3, 1), 0, 0
                indato 30, "coeficiente  a32=", coef(3, 2), 0, 0
                indato 30, "coeficiente  a33=", coef(3, 3), 0, 0
                indato 30, "coeficiente   b3=", coef(3, 4), 0, 0: CLS 2
          END SELECT
      END SELECT
      CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
      IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
      LOOP UNTIL CORRECTO
NEXT i
  DO
VIEW PRINT 12 * RW TO 24 * RW: CLS 2: PRINT TAB(30); "CONDICIONES INICIALES"
DO
VIEW PRINT 14 * RW TO 24 * RW: indato 30, "numero de trayectorias", tray, 1, imaxtray: CLS 2
FOR j = 1 TO tray
     CLS 2: PRINT TAB(15 * j MOD 75); "trayectoria"; j
     indato 30, "x1", x(j, 0), 0, 0
     IF ordensist > 1 THEN :  indato 30, "x2", y(j, 0), 0, 0
     IF ordensist = 3 THEN :  indato 30, "x3", z(j, 0), 0, 0
NEXT j
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
   PuntoCentro = 0
        FOR j = 1 TO tray
        IF PuntoCentro < ABS(x(j, 0)) THEN PuntoCentro = ABS(x(j, 0))
        IF PuntoCentro < ABS(y(j, 0)) THEN PuntoCentro = ABS(y(j, 0))
        IF PuntoCentro < ABS(z(j, 0)) THEN PuntoCentro = ABS(z(j, 0))
        NEXT j
   IF PuntoCentro = 0 THEN PuntoCentro = 1
   PuntoEqui = DELTAT * PuntoCentro / 10000
   PuntoFuera = 20 * PuntoCentro
   FOR j = 1 TO tray
    VIEW PRINT: CLS
    VIEW PRINT 4 * RW TO 10 * RW: CLS 2
    PRINT TAB(15 * j MOD 75); "TRAYECTORIA #"; j
    PRINT TAB(9); "Limite Maximo= "; PuntoFuera, "Variacion Minima = "; PuntoEqui
    PRINT "______________________________________________________________"
    PRINT
    PRINT TAB(3); "punto #"; SPC(10); "X1"; SPC(15); "X2"; SPC(15); "X3"
    PRINT "______________________________________________________________"
    'COLOR VAL(c3$)
    p = 0
    X1 = x(j, 0): puntomar(j) = X1
    X2 = y(j, 0)
    X3 = z(j, 0)
    VIEW PRINT 12 * RW TO 13 * RW
    LOCATE 12 * RW, 5: PRINT p
    LOCATE 12 * RW, 19: PRINT X1
    LOCATE 12 * RW, 36: PRINT X2
    LOCATE 12 * RW, 53: PRINT X3
    equilibrio = false
    fueraderango = false
    DO
     x3p = coef(3, 1) * X1 + coef(3, 2) * X2 + coef(3, 3) * X3 + coef(3, 4)
     z(j, p + 1) = z(j, p) + x3p * DELTAT: X3 = z(j, p + 1)
     x2p = coef(2, 1) * X1 + coef(2, 2) * X2 + coef(2, 3) * X3 + coef(2, 4)
     y(j, p + 1) = y(j, p) + x2p * DELTAT: X2 = y(j, p + 1)
     x1p = coef(1, 1) * X1 + coef(1, 2) * X2 + coef(1, 3) * X3 + coef(1, 4)
     x(j, p + 1) = x(j, p) + x1p * DELTAT: X1 = x(j, p + 1)
     IF ordensist = 1 THEN y(j, p + 1) = x1p
        VIEW PRINT 12 * RW TO 13 * RW
        LOCATE 12 * RW, 5: PRINT p
        LOCATE 12 * RW, 19: PRINT X1
        LOCATE 12 * RW, 36: PRINT X2
        LOCATE 12 * RW, 53: PRINT X3
```

```
    IF ABS(X1) > puntomax(j) THEN puntomax(j) = ABS(X1)
    IF puntomax(j) >= PuntoFuera THEN
    fueraderango = NOT false
    PRINT TAB(10); ' DIVERGE A UN PUNTO FUERA DEL RANGO DE GRAFICACION'
    END IF
    p = p + 1: puntofin(j) = p
    IF ABS(x(j, p - 1) - x(j, p)) <= PuntoEqui AND  ABS(y(j, p - 1) - y(j, p)) <= PuntoEqui AND ABS(z(j, p - 1) - z(j, p)) <=
         PuntoEqui THEN
    equilibrio = NOT false
    PRINT TAB(20); 'CONVERGE A UN PUNTO DE EQUILIBRIO'
    END IF
    LOOP UNTIL equilibrio OR fueraderango OR p = imaxpuntos
   NEXT j
   CALL presente(3, pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), tray)
   CALL HabilitacionDePantalla(4, itotalipx, itotalpy, itotalcol, itotalfilas, RW, cl, ipx, py, pantallahabilitacion)
   VIEW: CLS
   pulsedato 30, 'NUEVAS CONDICIONES INICIALES (S/N)', 'SsNn', A$
  LOOP UNTIL A$ = 'N' OR A$ = 'n'
 END IF
LOOP UNTIL SIGUIENTE$ = '2'
END SUB



SUB EcuacionesDeEstadoNL      (pantallahabilitacion, DELTAT, imaxpuntos, imaxtray, ordensist, x(), y(), z(), puntomax(),
                              puntofin(), tray)

DIM itotalsumx1(10), itotalsumx2(10), itotalsumx3(10)
DIM coef1(10, 10), coef2(10, 10), coef3(10, 10)
DIM p11(10, 10), p12(10, 10), p13(10, 10)
DIM p21(10, 10), p22(10, 10), p23(10, 10)
DIM p31(10, 10), p32(10, 10), p33(10, 10)
DIM f11(10, 10), f12(10, 10), f13(10, 10), f21(10, 10), f22(10, 10), f23(10, 10), f31(10, 10), f32(10, 10), f33(10, 10)
DIM a1(10, 3), a2(10, 3), a3(10, 3), a4(10, 3), a5(10, 3), a6(10, 3)
DIM b1(10, 3), b2(10, 3), b3(10, 3), b4(10, 3), b5(10, 3), b6(10, 3)
DIM c1(10, 3), c2(10, 3), c3(10, 3), c4(10, 3), c5(10, 3), c6(10, 3)
DIM restrx1(10, 3), restrx2(10, 3), restrx3(10, 3)
ON ERROR GOTO correccion
DO: CLS
CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, RW, cl, ipx, py, pantallahabilitacion)
VIEW PRINT 12 * RW TO 16 * RW
PRINT TAB(27); 'SISTEMAS NO LINEALES':PRINT
PRINT TAB(30); '[1] NUEVOS DATOS'
PRINT TAB(30); '[2] SALIR':PRINT
pulsedato 35, 'SELECCION', '12', SIGUIENTE$: CLS 2
IF SIGUIENTE$ = '1' THEN
iswFT = 0
DO
  CONST false = 0: TRUE = NOT false
  PuntoCentro = 0: equilibrio = false: fueraderango = false
  itotalECUAX1 = 1: itotalecuax2 = 1: itotalecuax3 = 1
  NUMECUACIONES = 0
  indato 30, 'ORDEN DEL SISTEMA ', ordensist, 1, 3: CLS 2
  SELECT CASE ordensist
      CASE 1:  itotalecuax2 = 0: itotalecuax3 = 0
               co3 = NOT false: co2 = NOT false
      CASE 2:  itotalecuax3 = 0: co3 = NOT false
      CASE ELSE
  END SELECT
  indato 25, 'numero de ecuaciones del sistema ', NUMECUACIONES, ordensist, 15: CLS 2
  IF NUMECUACIONES > ordensist THEN
   SELECT CASE ordensist
      CASE 1: itotalECUAX1 = NUMECUACIONES
      CASE 2: indato 25, 'numero de ecuaciones para x1 punto', itotalECUAX1, 1, (NUMECUACIONES - 1): CLS 2
              itotalecuax2 = NUMECUACIONES - itotalECUAX1
      CASE 3: indato 25, 'numero de ecuaciones para x1 punto', itotalECUAX1, 1, (NUMECUACIONES - 2): CLS 2
              indato 25, 'numero de ecuaciones para x2 punto', itotalecuax2, 1, (NUMECUACIONES - itotalECUAX1 - 1): CLS 2
              itotalecuax3 = NUMECUACIONES - itotalECUAX1 - itotalecuax2
   END SELECT
```

```
  ELSE
   co1 = NOT false: co2 = NOT false: co3 = NOT false
  END IF
  CLS 2
  PRINT TAB(25); 'x1punto'; SPC(5); itotalECUAX1;
  IF itotalECUAX1 = 1 THEN PRINT 'ecuacion'
  IF itotalECUAX1 <> 1 THEN PRINT 'ecuaciones'
  IF itotalecuax2 <> 0 THEN
       PRINT TAB(25); 'x2punto'; SPC(5); itotalecuax2;
       IF itotalecuax2 = 1 THEN PRINT 'ecuacion'
       IF itotalecuax2 > 1 THEN PRINT 'ecuaciones'
  END IF
  IF itotalecuax3 <> 0 THEN
       PRINT TAB(25); 'x3punto'; SPC(5); itotalecuax3;
       IF itotalecuax3 = 1 THEN PRINT 'ecuacion'
       IF itotalecuax3 > 1 THEN PRINT 'ecuaciones'
  END IF
  PRINT:pulsedato 25, 'LOS DATOS SON CORRECTOS (S/N)', 'sSNn', A$: CLS 2
  IF A$ = 'S' OR A$ = 's' THEN CORRECTO = NOT false ELSE CORRECTO = false
 LOOP UNTIL CORRECTO
 FOR i = 1 TO ordensist
    SELECT CASE i
       CASE 1: NUMECUACIONES = itotalECUAX1: varest$ = 'x1'
       CASE 2: NUMECUACIONES = itotalecuax2: varest$ = 'x2'
       CASE 3: NUMECUACIONES = itotalecuax3: varest$ = 'x3'
    END SELECT
    FOR j = 1 TO NUMECUACIONES
        DO
        VIEW PRINT 14 * RW TO 24 * RW: CLS 2
        PRINT TAB(25); 'ECUACION  #'; j; '  para  '; varest$; 'punto'
        PRINT
        PRINT TAB(20); 'xpunto = sumando #1 + ... + sumando #n'
        VIEW PRINT 17 * RW TO 24 * RW: CLS 2
        indato 25, 'número de términos o sumandos = ', numsumandos, 1, 15: CLS 2
        SELECT CASE ordensist
        CASE 1:PRINT TAB(24); 'sumando = coef * (x1)^p1 * f1'
               FOR l = 1 TO numsumandos
                   VIEW PRINT 21 * RW TO 22 * RW: CLS
                   PRINT TAB(15 * l MOD 75); 'sumando #'; l
                   VIEW PRINT 22 * RW TO 24 * RW
                   indato 30, 'coeficiente =', coef1(j, l), 0, 0: CLS 2
                   indato 30, 'exponente p1 =', p11(j, l), 0, 0: CLS 2
                   PRINT TAB(5); 'FUNCION DE X1: [0] ninguna  [1] SENx1  [2]  COSx1  [3] TANx1  '
                   indato 30, 'funcion f1 =', f11(j, l), 0, 3: CLS 2
               NEXT l
               itotalsumx1(j) = numsumandos
        CASE 2:PRINT TAB(30); 'sumando = coef * (x1)^p1 * (x2)^p2 * f1 * f2'
               IF varest$ = 'x1' THEN
                 FOR l = 1 TO numsumandos
                      VIEW PRINT 21 * RW TO 22 * RW: CLS 2
                      PRINT TAB(15 * l MOD 75); 'sumando #'; l
                      VIEW PRINT 22 * RW TO 24 * RW
                      indato 30, 'coeficiente =', coef1(j, l), 0, 0: CLS 2
                      indato 30, 'exponente p1 =', p11(j, l), 0, 0: CLS 2
                      indato 30, 'exponente p2 =', p12(j, l), 0, 0: CLS 2
                      PRINT TAB(5); 'FUNCION DE X1: [0] ninguna  [1] SENx1  [2]  COSx1  [3] TANx1  '
                      indato 30, 'funcion f1 =', f11(j, l), 0, 3: CLS 2
                      PRINT TAB(5); 'FUNCION DE X2: [0] ninguna  [1] SENx2  [2]  COSx2  [3] TANx2  '
                      indato 30, 'funcion f2 =', f12(j, l), 0, 3: CLS 2
                 NEXT l
                 itotalsumx1(j) = numsumandos
               ELSE
                 FOR l = 1 TO numsumandos
                     VIEW PRINT 21 * RW TO 22 * RW: CLS 2
                     PRINT TAB(15 * l MOD 75); 'sumando #'; l
                     VIEW PRINT 22 * RW TO 24 * RW: indato 30, 'coeficiente=', coef2(j, l), 0, 0: CLS 2
                     indato 30, 'exponente p1 =', p21(j, l), 0, 0: CLS 2
                     indato 30, 'exponente p2 =', p22(j, l), 0, 0: CLS 2
                     PRINT TAB(5); 'FUNCION DE X1: [0] ninguna  [1] SENx1  [2]  COSx1  [3] TANx1  '
```

```
            indato 30, "funcion f1 =", f21(j, l), 0, 3: CLS 2
            PRINT TAB(5); "FUNCION DE X2: [0] ninguna  [1] SENx2  [2]  COSx2  [3] TANx2  "
            indato 30, "funcion f2 =", f22(j, l), 0, 3: CLS 2
         NEXT l
         itotalsumx2(j) = numsumandos
       END IF
 CASE 3:PRINT TAB(20); "sumando = coef * (x1)^p1 * (x2)^p2 * (x3)^p3 * f1 * f2 * f3"
       SELECT CASE varest$
       CASE "x1": FOR l = 1 TO numsumandos
                  VIEW PRINT 21 * RW TO 22 * RW: CLS 2
                  PRINT TAB(15 * l MOD 75); "sumando #"; l
                  VIEW PRINT 22 * RW TO 24 * RW: indato 30, "coeficiente =", coef1(j, l), 0, 0: CLS 2
                  indato 30, "exponente p1 =", p11(j, l), 0, 0: CLS 2
                  indato 30, "exponente p2 =", p12(j, l), 0, 0: CLS 2
                  indato 30, "exponente p3 =", p13(j, l), 0, 0: CLS 2
                  PRINT TAB(5); "FUNCION DE X1: [0] ninguna  [1] SENx1  [2]  COSx1  [3] TANx1  "
                  indato 30, "funcion f1 =", f11(j, l), 0, 3: CLS 2
                  PRINT TAB(5); "FUNCION DE X2: [0] ninguna  [1] SENx2  [2]  COSx2  [3] TANx2  "
                  indato 30, "funcion f2 =", f12(j, l), 0, 3: CLS 2
                  PRINT TAB(5); "FUNCION DE X3: [0] ninguna  [1] SENx3  [2]  COSx3  [3] TANx3  "
                  indato 30, "funcion f3 =", f13(j, l), 0, 3: CLS 2
                  NEXT l
                  itotalsumx1(j) = numsumandos
       CASE "x2": FOR l = 1 TO numsumandos
                  VIEW PRINT 21 * RW TO 22 * RW: CLS 2
                  PRINT TAB(15 * l MOD 75); "sumando #"; l
                  VIEW PRINT 22 * RW TO 24 * RW: indato 30, "coeficiente =", coef2(j, l), 0, 0: CLS 2
                  indato 30, "exponente p1 =", p21(j, l), 0, 0: CLS 2
                  indato 30, "exponente p2 =", p22(j, l), 0, 0: CLS 2
                  indato 30, "exponente p3 =", p23(j, l), 0, 0: CLS 2
                  PRINT TAB(5); "FUNCION DE X1: [0] ninguna  [1] SENx1  [2]  COSx1  [3] TANx1  "
                  indato 30, "funcion f1 =", f21(j, l), 0, 3: CLS 2
                  PRINT TAB(5); "FUNCION DE X2: [0] ninguna  [1] SENx2  [2]  COSx2  [3] TANx2  "
                  indato 30, "funcion f2 =", f22(j, l), 0, 3: CLS 2
                  PRINT TAB(5); "FUNCION DE X3: [0] ninguna  [1] SENx3  [2]  COSx3  [3] TANx3  "
                  indato 30, "funcion f3 =", f23(j, l), 0, 3: CLS 2
                  NEXT l
                  itotalsumx2(j) = numsumandos
       CASE "x3": FOR l = 1 TO numsumandos
                  VIEW PRINT 21 * RW TO 22 * RW: CLS 2
                  PRINT TAB(15 * l MOD 75); "sumando #"; l
                  VIEW PRINT 22 * RW TO 24 * RW: indato 30, "coeficiente =", coef3(j, l), 0, 0: CLS 2
                  indato 30, "exponente p1=", p31(j, l), 0, 0: CLS 2
                  indato 30, "exponente p2=", p32(j, l), 0, 0: CLS 2
                  indato 30, "exponente p3=", p33(j, l), 0, 0: CLS 2
                  PRINT TAB(5); "FUNCION DE X1: [0] ninguna  [1] SENx1  [2]  COSx1  [3] TANx1  "
                  indato 30, "funcion f1=", f31(j, l), 0, 3: CLS 2
                  PRINT TAB(5); "FUNCION DE X2: [0] ninguna  [1] SENx2  [2]  COSx2  [3] TANx2  "
                  indato 30, "funcion f2=", f32(j, l), 0, 3: CLS 2
                  PRINT TAB(5); "FUNCION DE X3: [0] ninguna  [1] SENx3  [2]  COSx3  [3] TANx3  "
                  indato 30, "funcion f3=", f33(j, l), 0, 3: CLS 2
                  NEXT l
                  itotalsumx3(j) = numsumandos
      END SELECT
END SELECT
CLS 2: pulsedato 25, "LOS DATOS SON CORRECTOS (S/N)", "sSNn", A$: CLS 2
IF A$ = "S" OR A$ = "s" THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
IF NUMECUACIONES > 1 THEN
DO
VIEW PRINT 17 * RW TO 24 * RW: CLS 2
SELECT CASE ordensist
CASE 1: PRINT TAB(30); "restriccion = condicion a x1"
CASE 2: PRINT TAB(25); "restriccion = condicion a x1 AND condicion a x2"
CASE 3: PRINT TAB(20); "restriccion = condicion a x1 AND condicion a x2 AND condicion a x3"
END SELECT
  VIEW PRINT 19 * RW TO 24 * RW: CLS : PRINT "condicion a x1"
  PRINT "[1] x1 > a1       [2] x1 < a2     [3] a3 < x1 < a4        [4] a5 > x1 > a6"
  VIEW PRINT 23 * RW TO 24 * RW: PRINT TAB(30); : INPUT "condicion numero ="; restrx1(j, i): CLS 2
```

```
            CASE 2: indato 30, 'a2=', a2(j, i), 0, 0: CLS 2
            CASE 3: indato 30, 'a3=', a3(j, i), 0, 0: indato 30, 'a4=', a4(j, i), 0, 0: CLS 2
            CASE 4: indato 30, 'a5=', a5(j, i), 0, 0: indato 30, 'a6=', a6(j, i), 0, 0: CLS 2
         END SELECT
         IF ordensist > 1 THEN
            VIEW PRINT 19 * RW TO 24 * RW: CLS : PRINT 'condicion a x2'
            PRINT '[5] x2 > b1      [6] x2 < b2      [7] b3 < x2 < b4       [8] b5 > x2 > b6       [0] ninguna'
            VIEW PRINT 23 * RW TO 24 * RW: indato 30, 'numero de condicion=', restrx2(j, i), 0, 0: CLS 2
            SELECT CASE restrx2(j, i)
              CASE 0
              CASE 5: indato 30, 'b1=', b1(j, i), 0, 0: CLS 2
              CASE 6: indato 30, 'b2=', b2(j, i), 0, 0: CLS 2
              CASE 7: indato 30, 'b3=', b3(j, i), 0, 0: indato 30, 'b4=', b4(j, i), 0, 0: CLS 2
              CASE 8: indato 30, 'b5=', b5(j, i), 0, 0: indato 30, 'b6=', b6(j, i), 0, 0: CLS 2
            END SELECT
         END IF
         IF ordensist = 3 THEN
            VIEW PRINT 19 * RW TO 24 * RW: CLS : PRINT 'condicion x3'
            PRINT '[9] x3 > c1      [10] x3 < c2      [11] c3 < x3 < c4       [12] c5 > x3 > c6       [0] ninguna'
            VIEW PRINT 23 * RW TO 24 * RW: indato 30, 'numero de condicion=', restrx3(j, i), 0, 0: CLS 2
                 SELECT CASE restrx3(j, i)
                 CASE 9: indato 30, 'c1 =', c1(j, i), 0, 0: CLS 2
                 CASE 10: indato 30, 'c2 =', c2(j, i), 0, 0: CLS 2
                 CASE 11: indato 30, 'c3 =', c3(j, i), 0, 0: indato 30, 'c4 =', c4(j, i), 0, 0: CLS 2
                 CASE 12: indato 30, 'c5 =', c5(j, i), 0, 0: indato 30, 'c6 =', c6(j, i), 0, 0: CLS 2
                 END SELECT
         END IF
         CLS 2: pulsedato 25, 'LOS DATOS SON CORRECTOS (S/N)', 'sSNn', A$: CLS 2
         IF A$ = 'S' OR A$ = 's' THEN CORRECTO = NOT false ELSE CORRECTO = false
         LOOP UNTIL CORRECTO
    END IF
    NEXT j
NEXT i
DO
VIEW PRINT 12 * RW TO 24 * RW: CLS 2: PRINT TAB(30); 'CONDICIONES INICIALES'
DO
  VIEW PRINT 14 * RW TO 24 * RW: indato 30, 'numero de trayectorias', tray, 1, imaxtray: CLS 2
  FOR j = 1 TO tray
     CLS : PRINT TAB(15 * j MOD 75); 'trayectoria'; j
     indato 30, 'x1', x(j, 0), 0, 0
     IF ordensist > 1 THEN :  indato 30, 'x2', y(j, 0), 0, 0
     IF ordensist = 3 THEN :  indato 30, 'x3', z(j, 0), 0, 0
  NEXT j
  CLS 2: pulsedato 25, 'LOS DATOS SON CORRECTOS (S/N)', 'sSNn', A$: CLS 2
  IF A$ = 'S' OR A$ = 's' THEN CORRECTO = NOT false ELSE CORRECTO = false
LOOP UNTIL CORRECTO
   PuntoCentro = 0
FOR j = 1 TO tray
  IF PuntoCentro < ABS(x(j, 0)) THEN PuntoCentro = ABS(x(j, 0))
  IF PuntoCentro < ABS(y(j, 0)) THEN PuntoCentro = ABS(y(j, 0))
  IF PuntoCentro < ABS(z(j, 0)) THEN PuntoCentro = ABS(z(j, 0))
NEXT j
IF PuntoCentro = 0 THEN PuntoCentro = 1
PuntoEqui = DELTAT * PuntoCentro / 10000
PuntoFuera = 20 * PuntoCentro
FOR j = 1 TO tray
VIEW PRINT: CLS
VIEW PRINT 4 * RW TO 10 * RW: CLS 2
PRINT TAB(15 * j MOD 75); 'TRAYECTORIA #'; j
PRINT TAB(9); 'Limite Maximo= '; PuntoFuera, 'Variacion Minima = '; PuntoEqui
PRINT '______________________________________________________________________'
PRINT
PRINT TAB(3); 'punto #'; SPC(10); 'X1'; SPC(15); 'X2'; SPC(15); 'X3'
PRINT '______________________________________________________________________'
 p = 0
 X1 = x(j, 0): puntomax(j) = X1
 X2 = y(j, 0)
 X3 = z(j, 0)
VIEW PRINT 12 * RW TO 13 * RW
```

```
 X3 = z(j, 0)
VIEW PRINT 12 * RW TO 13 * RW
LOCATE 12 * RW, 5: PRINT p
LOCATE 12 * RW, 19: PRINT X1
LOCATE 12 * RW, 36: PRINT X2
LOCATE 12 * RW, 53: PRINT X3
 equilibrio = false
 fueraderango = false
 DO
 FOR k = ordensist TO 1 STEP -1
 SELECT CASE k
        CASE 3: nume = itotalecuax3
        CASE 2: nume = itotalecuax2
        CASE 1: nume = itotalECUAX1
 END SELECT
 IF nume > 1 THEN
    FOR i = 1 TO nume
    IF ordensist = 3 THEN
    SELECT CASE restrz3(i, k)
    CASE 9: IF z(j, p) >= c1(i, k) THEN PRINT "z="; z(j, p); ">"; c1(i, k); "cumple": co3 = NOT false
    CASE 10: IF z(j, p) <= c2(i, k) THEN PRINT "z="; z(j, p); "<"; c2(i, k); "cumple": co3 = NOT false
    CASE 11: IF z(j, p)  >= c3(i, k) AND z(j, p)  <= c4(i, k) THEN PRINT  "z="; z(j, p); ">";  c1(i, k); "z="; z(j, p);  ">";
         c1(i, k); "cumple": co3 = NOT false
    CASE 12: IF z(j, p) <= c5(i, k) OR z(j, p) >= c6(i, k) THEN PRINT "z="; z(j, p); ">"; c1(i, k); "z="; z(j, p); ">"; c1(i,
         k); "cumple": co3 = NOT false
    END SELECT
 END IF
 IF ordensist > 2 OR ordensist = 2 THEN
    SELECT CASE restrz2(i, k)
    CASE 5: IF y(j, p) >= b1(i, k) THEN PRINT "y="; y(j, p); ">"; b1(i, k); "cumple": co2 = NOT false
    CASE 6: IF y(j, p) <= b2(i, k) THEN PRINT "y="; y(j, p); "<"; b2(i, k); "cumple": co2 = NOT false
    CASE 7: IF y(j, p)  >= b3(i, k) AND y(j,  p) <= b4(i, k) THEN PRINT  "y="; y(j, p); ">"; b1(i, k); "cumple":  PRINT "y=";
         y(j, p); ">"; b1(i, k); "cumple": co2 = NOT false
    CASE 8: IF y(j, p) <= b5(i, k) OR y(j, p) >= b5(i, k) THEN PRINT "y="; y(j, p); ">"; b1(i, k); "cumple": PRINT "y="; y(j,
         p); ">"; b1(i, k); "cumple": co2 = NOT false
    CASE 0: co2 = NOT false
    END SELECT
 END IF
 SELECT CASE restrz1(i, k)
 CASE 1: IF x(j, p) >= a1(i, k) THEN PRINT "x="; x(j, p); ">"; a1(i, k); "cumple": co1 = NOT false
 CASE 2: IF x(j, p) <= a2(i, k) THEN PRINT "x="; x(j, p); "<"; a2(i, k); "cumple": co1 = NOT false
 CASE 3: IF x(j, p) >= a3(i, k) AND x(j, p) <= a4(i, k) THEN PRINT a3(i, k); "<"; "x="; x(j, p); "<"; a4(i, k); "cumple": co1
         = NOT false
 CASE 4: IF x(j,  p) <= a5(i, k) OR x(j, p) >= a6(i, k) THEN PRINT a5(i, k); ">"; "x="; x(j, p); ">"; a6(i, k); "cumple": co1
         = NOT false
 CASE 0: co1 = NOT false
 END SELECT
 IF co1 AND co2 AND co3 THEN condicionitotal = NOT false
 IF condicionitotal THEN EXIT FOR
 NEXT i
 ELSE
 i = 1
 condicionitotal = NOT false
 END IF
 IF condicionitotal THEN
    SELECT CASE k
           CASE 3:x3p = 0
                  FOR l = 1 TO itotalsumx3(i)
                     SELECT CASE f31(i, l)
                     CASE 1: f1 = SIN(X1)
                     CASE 2: f1 = COS(X1)
                     CASE 3: f1 = TAN(X1)
                     CASE 0: f1 = 1
                     END SELECT
                     SELECT CASE f32(i, l)
                     CASE 1: f2 = SIN(X2)
                     CASE 2: f2 = COS(X2)
                     CASE 3: f2 = TAN(X2)
                     CASE 0: f2 = 1
```

```
                  END SELECT
                  SELECT CASE f33(i, l)
                  CASE 1: f3 = SIN(X3)
                  CASE 2: f3 = COS(X3)
                  CASE 3: f3 = TAN(X3)
                  CASE 0: f3 = 1
                  END SELECT
                  r3p = r3p + coef3(i, l) * X1 ^ p31(i, l) * X2 ^ p32(i, l) * X3 ^ p33(i, l) * f1 * f2 * f3
                NEXT l
                  z(j, p + 1) = z(j, p) + r3p * DELTAT: X3 = z(j, p + 1)
           CASE 2:r2p = 0
                FOR l = 1 TO itotalsumr2(i)
                  SELECT CASE f21(i, l)
                  CASE 1: f1 = SIN(X1)
                  CASE 2: f1 = COS(X1)
                  CASE 3: f1 = TAN(X1)
                  CASE 0: f1 = 1
                  END SELECT
                  SELECT CASE f12(i, l)
                  CASE 1: f2 = SIN(X2)
                  CASE 2: f2 = COS(X2)
                  CASE 3: f2 = TAN(X2)
                  CASE 0: f2 = 1
                  END SELECT
                  SELECT CASE f13(i, l)
                  CASE 1: f3 = SIN(X3)
                  CASE 2: f3 = COS(X3)
                  CASE 3: f3 = TAN(X3)
                  CASE 0: f2 = 1
                  END SELECT
                  r2p = r2p + coef2(i, l) * X1 ^ p21(i, l) * X2 ^ p22(i, l) * X3 ^ p23(i, l) * f1 * f2 * f3
                NEXT l
                  y(j, p + 1) = y(j, p) + r2p * DELTAT: X2 = y(j, p + 1)
           CASE 1:r1p = 0
                FOR l = 1 TO itotalsumr1(i)
                  SELECT CASE f11(i, l)
                  CASE 1: f1 = SIN(X1)
                  CASE 2: f1 = COS(X1)
                  CASE 3: f1 = TAN(X1)
                  CASE 0: f1 = 1
                  END SELECT
                  SELECT CASE f12(i, l)
                  CASE 1: f2 = SIN(X2)
                  CASE 2: f2 = COS(X2)
                  CASE 3: f2 = TAN(X2)
                  CASE 0: f2 = 1
                  END SELECT
                  SELECT CASE f13(i, l)
                  CASE 1: f3 = SIN(X3)
                  CASE 2: f3 = COS(X3)
                  CASE 3: f3 = TAN(X3)
                  CASE 0: f3 = 1
                  END SELECT
                  r1p = r1p + coef1(i, l) * X1 ^ p11(i, l) * X2 ^ p12(i, l) * X3 ^ p13(i, l) * f1 * f2 * f3
                NEXT l
                  x(j, p + 1) = x(j, p) + r1p * DELTAT: X1 = x(j, p + 1)
                IF ordenasist = 1 THEN
                  y(j, p + 1) = r1p
                END IF
              END SELECT
        ELSE
        PRINT "ojo problema en la logica de restricciones": STOP
        END IF
     NEXT k
  VIEW PRINT 12 * RW TO 13 * RW
  LOCATE 12 * RW, 5: PRINT p
  LOCATE 12 * RW, 19: PRINT X1
  LOCATE 12 * RW, 36: PRINT X2
  LOCATE 12 * RW, 53: PRINT X3
```

```
    IF ABS(X1) > puntomax(j) THEN puntomax(j) = ABS(X1)
    IF puntomax(j) >= PuntoFuera THEN
    fueraderango = NOT false
    PRINT TAB(10); ' DIVERGE A UN PUNTO FUERA DEL RANGO DE GRAFICACION'
    END IF
    p = p + 1: puntofin(j) = p
    IF ABS(x(j, p - 1) - x(j, p)) <= PuntoEqui AND  ABS(y(j, p - 1) - y(j, p)) <= PuntoEqui AND ABS(z(j, p - 1) - z(j, p)) <=
         PuntoEqui THEN
    equilibrio = NOT false
    PRINT TAB(20); 'CONVERGE A UN PUNTO DE EQUILIBRIO'
    END IF
    IF ordensist = 1 THEN co1 = false
    IF ordensist = 2 THEN co1 = false: co2 = false
    IF ordensist = 3 THEN co1 = false: co2 = false: co3 = false
    condicionitotal = false
    LOOP UNTIL equilibrio OR fueraderango OR p = imaxpuntos
NEXT j
CALL presente(3, pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), tray)
CALL HabilitacionDePantalla(4, itotalipx, itotalpy, itotalcol, itotalfilas, RW, cl, ipx, py, pantallahabilitacion)
VIEW: CLS
pulsedato 30, 'NUEVAS CONDICIONES INICIALES (S/N)', 'SsNn', A$
LOOP UNTIL A$ = 'N' OR A$ = 'n'
END IF
LOOP UNTIL SIGUIENTE$ = '2'
END SUB
```

```
                                        'MODULO TRAY

'Información para la presentación o impresión de resultados
'**************************************************************


SUB BuscaDatos (pantallahabilitacion, deltat, ordensist, x(), y(), z(), puntomax(), puntofin(), tray, mp(), Resc, Rrp)

REDIM Mipx(tray), Mpy(tray), MPZ(tray), Tipx(tray), Tpy(tray), TPZ(tray), mp(tray), tp(tray)
ON ERROR GOTO correccion
FOR j = 1 TO tray
   FOR i = 0 TO puntofin(j)
      IF Mipx(j) < x(j, i) THEN Mipx(j) = x(j, i): Tipx(j) = deltat * i
      IF Mpy(j) < y(j, i) THEN Mpy(j) = y(j, i): Tpy(j) = deltat * i
      IF MPZ(j) < z(j, i) THEN MPZ(j) = z(j, i): TPZ(j) = deltat * i
NEXT i:NEXT j
VIEW PRINT 10 * rw TO 23 * rw: CLS
FOR j = 1 TO tray
  PRINT 'TRAYECTORIA #'; j; SPC(20); puntofin(j); ' PUNTOS'
  PRINT 'PUNTO INICIAL'
  PRINT 'X1 = '; x(j, 0); SPC(10); 'X2 = '; y(j, 0); SPC(10); 'X3 = '; z(j, 0)
  PRINT 'PUNTO FINAL'
  PRINT 'X1 = '; x(j, puntofin(j)); SPC(10); 'X2 = '; y(j, puntofin(j)); SPC(10); 'X3 = '; z(j, puntofin(j))
  PRINT 'PUNTOS MAXIMOS'
```

```
    IF Resc <> 0 OR Rrp <> 0 THEN PRINT "Maximo sobreimpulso de la salida ="; mp(j); "t ="; tp(j)
    PRINT "X1 = "; Xipr(j); SPC(10); "t ="; Tipr(j)
    PRINT "X2 = "; Mpy(j); SPC(10); "t ="; Tpy(j)
    PRINT "X3 = "; MPZ(j); SPC(10); "t ="; TPZ(j)
    indato 25, "PULSE <ENTER> PARA SEGUIR", c, 0, 0: CLS
  NEXT j
  pulsedato 25, "PRESENTAR LISTADO DE PUNTOS (S/N)", "snSN", LISTA$
  IF LISTA$ = "s" OR LISTA$ = "S" THEN
  FOR j = 1 TO tray
     PRINT "trayectoria #"; j
     indato 25, "NUMERO DE PUNTOS A PRESENTAR", nptos, 0, puntofin(j)
     IF nptos <> 0 THEN
     VIEW PRINT: CLS
     PRINT SPC(5); "t              x1         x2              x3"
     i = 0
     DO
     PRINT SPC(5);    (CINT(i * 100) / 100) * deltat; SPC(5); CINT(x(j, i) * 1000) / 1000; SPC(5); CINT(y(j, i) * 1000) / 1000;
                       SPC(5); CINT(z(j, i) * 1000) / 1000
     i = i + INT(puntofin(j) / nptos)
     LOOP UNTIL i >= puntofin(j)
     indato 25, "", c, 0, 0
     END IF
  NEXT j
  END IF
  END SUB



  SUB Ejemplos (pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), tray)

  ON ERROR GOTO correccion
  HabilitacionDePantalla iswprin, itotalipr, itotalpy, itotalcol, itotalfilas, rw, cl, ipr, py, pantallahabilitacion
  imaxpuntos = 150
  VIEW PRINT 8 * rw TO 14 * rw: CLS
  PRINT TAB(20); "[1]  SISTEMAS LINEALES 2do orden"
  PRINT TAB(20); "[2]  SISTEMAS NO LINEALES"
  PRINT TAB(20); "[3]  SISTEMAS LINEALES 3er orden"
  PRINT TAB(20); "[4]  SISTEMAS NO LINEALES 3er orden"
  PRINT TAB(20); "[5]  RESULTADOS GRABADOS"
  VIEW PRINT 14 * rw TO 15 * rw
  pulsedato 20, "SELECCION =", "12345", seleccion$
  SELECT CASE seleccion$
  CASE "1":  VIEW PRINT 10 * rw TO 17 * rw: CLS
         PRINT TAB(30); "EJEMPLO 1= FOCO ESTABLE"
         PRINT TAB(30); "EJEMPLO 2= CENTRO"
         PRINT TAB(30); "EJEMPLO 3= NODO ESTABLE"
         PRINT TAB(30); "EJEMPLO 4= FOCO INESTABLE"
         PRINT TAB(30); "EJEMPLO 5= NODO INESTABLE"
         PRINT TAB(30); "EJEMPLO 6= SILLA DE MONTAR"
         VIEW PRINT 17 * rw TO 18 * rw
         pulsedato 30, "SELECCION=", "123456", sistemalineal$
         SELECT CASE sistemalineal$
         CASE "1": nombre$ = "b:\sl2fe.ejm": wn2 = 1: eps = .5: wn = 1         'wn2 > 0, 0 < eps < 1
                        nombregraf$ = "EJEMPLO1:  G(s)=1/(s^2 +s +1)             PTO. SING.= FOCO ESTABLE"
         CASE "2": nombre$ = "b:\sl2c.ejm": wn2 = 1: eps = 0: wn = 1           'wn2 > 0, eps = 0
                        nombregraf$ = "EJEMPLO2:  G(s)=1/(s^2 +1)               PTO. SING.= CENTRO"
         CASE "3": nombre$ = "b:\sl2ne.ejm": wn2 = 1: eps = 1.2: wn = 1        'wn2 > 0, eps >= 1
                        nombregraf$ = "EJEMPLO3:  G(s)=1/(s^2 +2.4s +1)          PTO. SING.= NODO ESTABLE"
         CASE "4": nombre$ = "b:\sl2fi.ejm": wn2 = 1: eps = -.5: wn = 1        'wn2 > 0, -1 < eps < 0
                        nombregraf$ = "EJEMPLO4:  G(s)=1/(s^2 -s +1)             PTO. SING.= FOCO INESTABLE"
         CASE "5": nombre$ = "b:\sl2ni.ejm": wn2 = 1: eps = -1: wn = 1         'wn2 > 0, eps <= -1
                        nombregraf$ = "EJEMPLO5:  G(s)=1/(s^2 -2s +1)            PTO. SING.= NODO ESTABLE"
         CASE "6": nombre$ = "b:\sl2sm.ejm": wn2 = -1: eps = 0: wn = 1         'wn2 < 0, eps = 0
                        nombregraf$ = "EJEMPLO6:  G(s)=1/(s^2 -1)               PTO. SING.= FOCO ESTABLE"
         CASE ELSE
         END SELECT
  CASE "2": VIEW PRINT 10 * rw TO 17 * rw: CLS
         PRINT TAB(30); "EJEMPLO 1= SILLA DE MONTAR Y FOCO ESTABLE"
         PRINT TAB(30); "EJEMPLO 2= CENTROS PERIODICOS"
```

```
        PRINT TAB(30); "EJEMPLO 3= CON RELE Y ZONA MUERTA , ESTABLE"
        VIEW PRINT 17 + rw TO 18 + rw
        pulsedato 30, "SELECCION=", "123", sistemanolineal$
        SELECT CASE sistemanolineal$
        CASE "1": nombre$ = "b:\snl2smfe.eje"
                        nombregraf$ = "EJM1: x2PUNTO = -0.5x2 -2x1 -x1^2    PTOS. SING.= SILLA Y FOCO ESTABLE"
        CASE "2": nombre$ = "b:\snl2cp.eje"
                        nombregraf$ = "EJEMPLO2:  x2PUNTO = -.1sin(x1)          PTO. SING.= CENTROS PERIODICOS"
        CASE "3": nombre$ = "b:\snl2RZM.eje"
                        nombregraf$ =   "EJEMPLO3:  G(s)=4/s(s +1)       RELE salidas -1 y 1    ZONA MUERTA desde -1 a 1
                                        PTO. SING.= FOCO ESTABLE"
        END SELECT
 CASE "3": VIEW PRINT 10 + rw TO 17 + rw: CLS
        PRINT TAB(30); "EJEMPLO 1= CENTRO EN 3D"
        PRINT TAB(30); "EJEMPLO 2= FOCO EN 3D"
        PRINT TAB(30); "EJEMPLO 3= NODO EN 3D"
        VIEW PRINT 17 + rw TO 18 + rw
        pulsedato 30, "SELECCION=", "123456", sistemalineal3D$
        SELECT CASE sistemalineal3D$
        CASE "1": nombre$ = "b:\centro3d.eje"
                        nombregraf$ =   "EJEMPLO1:  G(s)=1/(s^3 +2s -1)          PTO. SING.= CENTRO INESTABLE paralelo al
                                        eje x1 (X)"
        CASE "2": nombre$ = "b:\foco3d.eje"
                        nombregraf$ = "EJEMPLO 2= FOCO EN 3D: s=-.5, s=-.1+/-j4"
        CASE "3": nombre$ = "b:\nodo3d.eje"
                        nombregraf$ = "EJEMPLO 3= NODO EN 3D: s=-.1, s=-.5+/-j4"
        END SELECT
 CASE "4": VIEW PRINT 10 + rw TO 17 + rw: CLS
        PRINT TAB(30); "EJEMPLO 1= CENTRO EN 3D"
        VIEW PRINT 17 + rw TO 18 + rw
        pulsedato 30, "SELECCION=", "123456", sistemanolineal3D$
        SELECT CASE sistemanolineal3D$
        CASE "1": nombre$ = "b:\ciclo3d.eje"
        END SELECT
 CASE "5": INPUT "     nombre del archivo (d:\path\nombre.ext)"; nombre$
 END SELECT
        OPEN nombre$ FOR INPUT AS #1
        INPUT #1, deltat, imaxpuntos, imaxtray, ordensist, tray
        REDIM x(tray, imaxpuntos), y(tray, imaxpuntos), z(tray, imaxpuntos)
        FOR j = 1 TO tray
        INPUT #1, j, puntomax(j), puntofin(j)
        FOR i = 0 TO puntofin(j)
        INPUT #1, x(j, i), y(j, i), z(j, i)
        NEXT i
        NEXT j
        CLOSE #1
 CALL presente(iswprin, pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), tray)
 END SUB


 SUB imprimir       (iswprim, pantallahabilitacion, nombregraf$,  deltat, ordensist, x(), y(),  z(), puntomax(), puntofin(),
                    tray, mp(), tp(), Kesc, Krp, iswFT)

 ON ERROR GOTO correccion
 CALL HabilitacionDePantalla(iswprim, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion)
 VIEW PRINT 5 + rw TO 24 + rw: CLS
 IF (x(1, 0) + y(1, 0) + z(1, 0) = 0) AND (x(1, 10) + y(1, 10) + z(1, 10) = 0) THEN
 PRINT "                        INGRESE LOS DATOS DEL SISTEMA"
 PRINT "                    EN REPRESENTACION DE DIAGRAMA DE BLOQUES"
 PRINT "                           O ECUACIONES DE ESTADO"
 PRINT "                  CON LAS OPCIONES [1] Y [2] DEL MENU PRINCIPAL":PRINT
 pulsedato 20, "PULSE <S> PARA RETORNAR AL MENU PRINCIPAL", "Ss", A$: CLS
 ELSE
 DO
 CALL HabilitacionDePantalla(iswprim, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion)
 VIEW PRINT: CLS
 IF iswprim = 2 THEN
 PRINT "                   SELECCIONE EL TIPO DE GRAFICO A IMPRIMIR"
```

```
PRINT "                    UNA VEZ PRESENTE EL GRAFICO EN LA PANTALLA"
PRINT "                     SI DESEA CAMBIE EL RANGO DE GRAFICACION":PRINT
PRINT "                           PULSE SIMULTANEAMENTE"
PRINT "                        LAS TECLAS  <Shift> <PrtSc>"
PRINT "              DEL MENU QUE SE PRESENTA DEL PROGRAMA PIZZAS SELECIONE"
PRINT "                             {Print} {Printer}":PRINT
PRINT "           FINALMENTE ESCOGA LA OPCION {quit} PARA RETORNAR AL PROGRAMA TTF":PRINT
PRINT "                         PULSE <ENTER> PARA SEGUIR":PRINT:PRINT:PRINT
END IF
PRINT TAB(20); "[1]  GRAFICO DE TRAYECTORIAS DE FASE"
PRINT TAB(20); "[2]  GRAFICO DE RESPUESTA EN EL TIEMPO"
PRINT TAB(20); "[3]  CARACTERISTICAS DEL SISTEMA"
PRINT TAB(20); "[4]  SALIR":PRINT
pulsedato 25, "SELECCION =", "1234", sgraf$: CLS
IF tray > 1 AND sgraf$ <> "4" THEN
indato 25, "NUMERO DE CURVAS A IMPRIMIRSE ", newtray, 1, tray
ELSE
newtray = 1
END IF
SELECT CASE sgraf$
CASE "1":CALL presente(iswprin, pantallahabilitacion, nombregraf$, ordensist, x(), y(), z(), puntomax(), puntofin(), newtray)
       VIEW: VIEW PRINT: CLS
CASE "2":CALL       Tiempo(iswprin, pantallahabilitacion, deltat, nombregraf$, x(), y(), z(), puntomax(), puntofin(),
                newtray, mp(), tp(), Resc, Rrp, iswFT)
CASE "3":CALL       BuscaDatos(pantallahabilitacion, deltat, ordensist, x(), y(), z(), puntomax(), puntofin(), newtray,
                mp(), Resc, Rrp)
CASE "4"
END SELECT
LOOP UNTIL sgraf$ = "4"
END IF
END SUB


    SUB presente (iswprin, pantallahabilitacion, nombregraf$, ordensist, X(), y(), z(), puntomax(), puntofin(), tray)

ON ERROR GOTO correccion
HabilitacionDePantalla iswprin, itotalipr, itotalpy, itotalcol, itotalfilas, rw, icl, ipr, py, pantallahabilitacion
FOR j = 1 TO tray
   IF maxpunto < puntofin(j) THEN maxpunto = puntofin(j)
NEXT j
REDIM X3D(tray, maxpunto), Y3D(tray, maxpunto)
CONST PI = 3.141592653589#
numerodepresentaciones = 0
cruz$ = "NU2 ND2 NL2 NR2"
EXIS$ = "br5 ne3 nf3 ng3 nh3"
YE$ = "br5 ne3 nh3 nd3"
ZETA$ = "br5 e3 l6 r6 g6 r6"
VIEW: CLS
IF ordensist = 3 THEN va = 45: ha = 45
IF ordensist <> 3 THEN va = 0: ha = -90
VIEW (ipr, 3.5 * py)-(itotalipr - ipr, itotalpy - 3.5 * py), , 1
VIEW PRINT 23 * rw TO 24 * rw
DO
V = va * PI / 180
h = ha * PI / 180
CLS 2: PRINT TAB(30); "GRAFICO EN PROCESO"
IF ordensist = 3 THEN
       VIEW PRINT 23 * rw TO 24 * rw
       FOR j = 1 TO tray
       FOR i = 0 TO puntofin(j)
       X3D(j, i) = -X(j, i) * SIN(h) + y(j, i) * COS(h)
       Y3D(j, i) = -X(j, i) * COS(V) * COS(h) - y(j, i) * COS(V) * SIN(h) + z(j, i) * SIN(V)
       NEXT i
       NEXT j
END IF
IF numerodepresentaciones <> 0 THEN
       xm = 1.6 * m: ym = 1.1 * m
ELSE
```

```
        m = 0
        FOR j = 1 TO tray
        IF m < puntomax(j) THEN m = puntomax(j)
        NEXT j
        IF m = 0 THEN CLS 2: PRINT TAB(30); "NO SE ENCONTRO GRAFICO": m = 1
        xm = 1.6 * m: ym = 1.1 * m
        numerodepresentaciones = 1
END IF
        WINDOW (-xm, ym)-(xm, -ym)
CLS
DENTRO = 0
IF ordensist = 3 THEN
FOR j = 1 TO tray
        IF ABS(X3D(j, 0)) <= m AND ABS(Y3D(j, 0)) <= m THEN
        PSET (X3D(j, 0), Y3D(j, 0))
        DENTRO = 1
        END IF
FOR i = 1 TO puntofin(j)
        IF ABS(X3D(j, i)) <= m AND ABS(Y3D(j, i)) <= m THEN
                IF DENTRO = 1 THEN
                IF X(j, i) > 0 THEN DONDE = 5 ELSE DONDE = 2
                LINE -(X3D(j, i), Y3D(j, i)), DONDE
                ELSE
                  PSET (X3D(j, i), Y3D(j, i))
                  DENTRO = 1
                END IF
        ELSE
                DENTRO = 0
        END IF
NEXT i
DENTRO = 0
NEXT j
ELSE
FOR j = 1 TO tray
        IF ABS(X(j, 0)) <= m AND ABS(y(j, 0)) <= m THEN
        PSET (X(j, 0), y(j, 0))
        DENTRO = 1
        END IF
FOR i = 1 TO puntofin(j)
        IF ABS(X(j, i)) <= m AND ABS(y(j, i)) <= m THEN
                IF DENTRO = 1 THEN
                IF X(j, i) > 0 THEN DONDE = 5 ELSE DONDE = 2
                  LINE -(X(j, i), y(j, i)), DONDE
                ELSE
                  PSET (X(j, i), y(j, i))
                  DENTRO = 1
                END IF
        ELSE
                DENTRO = 0
        END IF
NEXT i
DENTRO = 0
NEXT j
END IF
limgraf = 1 * m
xy3xm = -(limgraf) * SIN(h) + (-limgraf) * COS(h)
xy3ym = -(limgraf) * COS(V) * COS(h) - (-limgraf) * COS(V) * SIN(h) + (0) * SIN(V)
xy3xd = -(-limgraf) * SIN(h) + (limgraf) * COS(h)
xy3yd = -(-limgraf) * COS(V) * COS(h) - (limgraf) * COS(V) * SIN(h) + (0) * SIN(V)
xy3xl = -(-limgraf) * SIN(h) + (-limgraf) * COS(h)
xy3yl = -(-limgraf) * COS(V) * COS(h) - (-limgraf) * COS(V) * SIN(h) + (0) * SIN(V)
xy3xr = -(limgraf) * SIN(h) + (limgraf) * COS(h)
xy3yr = -(limgraf) * COS(V) * COS(h) - (limgraf) * COS(V) * SIN(h) + (0) * SIN(V)
zy3xm = -(0) * SIN(h) + (-limgraf) * COS(h)
zy3ym = -(0) * COS(V) * COS(h) - (-limgraf) * COS(V) * SIN(h) + (limgraf) * SIN(V)
zy3xd = -(0) * SIN(h) + (limgraf) * COS(h)
zy3yd = -(0) * COS(V) * COS(h) - (limgraf) * COS(V) * SIN(h) + (-limgraf) * SIN(V)
zy3xl = -(0) * SIN(h) + (-limgraf) * COS(h)
zy3yl = -(0) * COS(V) * COS(h) - (-limgraf) * COS(V) * SIN(h) + (-limgraf) * SIN(V)
```

```
zy3zr = -(0) * SIN(h) + (limgraf) * COS(h)
zy3yr = -(0) * COS(V) * COS(h) - (limgraf) * COS(V) * SIN(h) + (limgraf) * SIN(V)
x3xi = -(-limgraf) * SIN(h) + 0 * COS(h)
x3yi = -(-limgraf) * COS(V) * COS(h) - (0) * COS(V) * SIN(h)
x3xd = -(limgraf) * SIN(h) + 0 * COS(h)
x3yd = -(limgraf) * COS(V) * COS(h) - (0) * COS(V) * SIN(h) + 0 * SIN(V)
y3xi = -(0) * SIN(h) + (-limgraf) * COS(h)
y3yi = -(0) * COS(V) * COS(h) - (-limgraf) * COS(V) * SIN(h)
y3xd = -(0) * SIN(h) + (limgraf) * COS(h)
y3yd = -(0) * COS(V) * COS(h) - (limgraf) * COS(V) * SIN(h) + (0) * SIN(V)
z3xi = -(0) * SIN(h) + (0) * COS(h)
z3yi = -(0) * COS(V) * COS(h) - (0) * COS(V) * SIN(h) + (-limgraf) * SIN(V)
z3xd = -(0) * SIN(h) + (0) * COS(h)
z3yd = -(0) * COS(V) * COS(h) - (0) * COS(V) * SIN(h) + (limgraf) * SIN(V)
IF limgraf <= .01 THEN paso = .001
IF limgraf > .01 AND limgraf <= .1 THEN paso = .01
IF limgraf > .1 AND limgraf <= 1 THEN paso = .1
IF limgraf > 1 AND limgraf <= 10 THEN paso = 1
IF limgraf > 10 AND limgraf <= 100 THEN paso = 10
IF limgraf > 100 AND limgraf <= 1000 THEN paso = 100
IF limgraf > 1000 AND limgraf <= 10000 THEN paso = 1000
IF limgraf > 10000 THEN paso = 10000
LINE (0, 0)-(x3xi, x3yi), , , &HCCCC
LINE (0, 0)-(x3xd, x3yd), , , &HCCCC
DRAW EXIS$
FOR i = 0 TO -(limgraf) STEP -paso
          XESC = i * SIN(h)
          YESC = i * COS(V) * COS(h)
          PSET (XESC, YESC)
          DRAW cruz$
NEXT i
FOR i = 0 TO (limgraf) STEP paso
          XESC = i * SIN(h)
          YESC = i * COS(V) * COS(h)
          PSET (XESC, YESC)
          DRAW cruz$
NEXT i
LINE (0, 0)-(y3xi, y3yi), , , &HCCCC
LINE (0, 0)-(y3xd, y3yd), , , &HCCCC
DRAW YE$
FOR i = 0 TO -(limgraf) STEP -paso
          XESC = i * COS(h)
          YESC = -i * COS(V) * SIN(h)
          PSET (XESC, YESC)
          DRAW cruz$
NEXT i
FOR i = 0 TO (limgraf) STEP paso
          XESC = i * COS(h)
          YESC = -i * COS(V) * SIN(h)
          PSET (XESC, YESC)
          DRAW cruz$
NEXT i
IF ordensist = 3 THEN
   LINE (0, 0)-(z3xi, z3yi), , , &HCCCC
   LINE (0, 0)-(z3xd, z3yd), , , &HCCCC
   DRAW ZETA$
   FOR i = -0 TO -(limgraf) STEP -paso
      YESC = i * SIN(V)
      PSET (0, YESC)
      DRAW "l2 r4 l2"
   NEXT i
   FOR i = 0 TO (limgraf) STEP paso
      YESC = i * SIN(V)
      PSET (0, YESC)
      DRAW "l2 r4 l2"
   NEXT i
END IF
LINE (zy3xl, zy3yl)-(zy3xd, zy3yd), , , &H8888
LINE -(zy3zr, zy3yr), , , &H8888
```

```
 LINE -(xy3xu, xy3yu), , , &H8888
 LINE -(xy3xl, xy3yl), , , &H8888
 LINE (zy3xl, zy3yl)-(zy3xd, zy3yd), , , &H8888
 LINE -(zy3xr, zy3yr), , , &H8888
 LINE -(zy3xu, zy3yu), , , &H8888
 LINE -(zy3xl, zy3yl), , , &H8888
        VIEW PRINT 1 * rw TO 2 * rw
        IF iswprin <> 3 THEN
        IF nombregraf$ = "" THEN
        CLS 2: PRINT "NOMBRE O REFERENCIA PARA EL GRAFICO (max. 76 letras)"
        INPUT nombregraf$: CLS 2
        ELSE
        CLS 2: pulsedato 0, "NUEVO NOMBRE PARA EL GRAFICO (S/N)", "sSnN", A$: CLS 2
        IF A$ = "S" OR A$ = "s" THEN
        PRINT "NOMBRE O REFERENCIA PARA EL GRAFICO (max. 76 letras)"
        INPUT nombregraf$: CLS 2
        END IF
        END IF
        END IF
        VIEW PRINT 1 * rw TO 3 * rw
        PRINT TAB(30); "TRAYECTORIAS DE FASE "
        PRINT nombregraf$
 VIEW PRINT 23 * rw TO 24 * rw
 PRINT TAB(30); "1 division ="; paso; "unidades"
 VIEW PRINT 24 * rw TO 24 * rw
 indato 20, "Maximo punto de visualizacion (0 SALIR)=", m, 0, 0: CLS 2
 IF m <> 0 AND ordensist = 3 THEN
 VIEW PRINT 23 * rw TO 24 * rw
 CLS 2
 indato 30, "ANGULO VERTICAL", va, -360, 360: CLS 2
 indato 30, "ANGULO HORIZONTAL", ha, -360, 360: CLS 2
 END IF
 VIEW PRINT 23 * rw TO 24 * rw
 LOOP UNTIL m = 0
 END SUB


         SUB Tiempo (iswprin, pantallahabilitacion, deltat, nombregraf$, x(), y(), z(), puntomax(), puntofin(), tray, mp(),
         tp(), Resc, Rrp, iswFT)

 ON ERROR GOTO correccion
 CALL HabilitacionDePantalla(iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacio)
 DO
 VIEW PRINT 10 * rw TO 18 * rw: CLS
 IF iswFT = 1 THEN
 PRINT TAB(30); "[1] error vs. tiempo"
 PRINT TAB(30); "[2] derivada del error vs. tiempo"
 PRINT TAB(30); "[3] segunda derivada del error vs. tiempo"
 PRINT TAB(30); "[4] salida vs. tiempo"
 PRINT TAB(30); "[5] SALIR":PRINT
 VIEW PRINT 19 * rw TO 20 * rw: CLS 2
 pulsedato 30, "seleccion =", "12345", selecciongraf$: CLS
 ELSE
 PRINT TAB(30); "[1] x1  vs.  tiempo"
 PRINT TAB(30); "[2] x2  vs. tiempo"
 PRINT TAB(30); "[3] x3  vs. tiempo"
 PRINT TAB(30); "[4] SALIR":PRINT
 VIEW PRINT 14 * rw TO 16 * rw: CLS 2
 pulsedato 30, "seleccion =", "1234", selecciongraf$: CLS
 IF selecciongraf$ = "4" THEN selecciongraf$ = "5"
 END IF
 IF selecciongraf$ <> "5" THEN
 m = 0
 DO
 CLS
 IF numerodepasadas = 0 THEN
       FOR j = 1 TO tray
       IF m < puntomax(j) THEN m = puntomax(j)
       IF t < puntofin(j) THEN t = puntofin(j)
```

```
        NEXT j
        IF m = 0 THEN m = 1
        t = deltat * t
        IF t = 0 THEN t = 1
        numerodepasadas = 1
        ELSE
 END IF
        HabilitacionDePantalla iswprin, itotalipx, itotalpy, itotalcol, itotalfilas, rw, cl, ipx, py, pantallahabilitacion
        VIEW (ipx, 3.5 * py)-(itotalipx - ipx, itotalpy - 5 * py), , 1
        WINDOW (-.1 * t, 1.1 * m)-(t, -1.1 * m)
        limgraf = 1.1 * m
        IF limgraf <= .01 THEN paso = .001
        IF limgraf > .01 AND limgraf <= .1 THEN paso = .01
        IF limgraf > .1 AND limgraf <= 1 THEN paso = .1
        IF limgraf > 1 AND limgraf <= 10 THEN paso = 1
        IF limgraf > 10 AND limgraf <= 100 THEN paso = 10
        IF limgraf > 100 AND limgraf <= 1000 THEN paso = 100
        IF limgraf > 1000 AND limgraf <= 10000 THEN paso = 1000
        IF limgraf > 10000 THEN paso = 10000
        FOR i = 0 TO -(limgraf) STEP -paso
        YESC = i
        PSET (0, YESC)
        DRAW "nu2 nd2 nr2 nl2"
        NEXT i
        FOR i = 0 TO (limgraf) STEP paso
        YESC = i
        PSET (0, YESC)
        DRAW "nu2 nd2 nr2 nl2"
        NEXT i
        LINE (0, -limgraf)-(0, limgraf), , , &HCCCC
        LINE (0, 0)-(t, 0), , , &HCCCC
        IF t <= .01 THEN pasot = .001
        IF t > .01 AND t <= .1 THEN pasot = .01
        IF t > .1 AND t <= 1 THEN pasot = .1
        IF t > 1 AND t <= 10 THEN pasot = 1
        IF t > 10 AND t <= 100 THEN pasot = 10
        IF t > 100 AND t <= 1000 THEN pasot = 100
        IF t > 1000 AND t <= 10000 THEN pasot = 1000
        IF t > 10000 THEN pasot = 10000
        FOR i = 0 TO t STEP pasot
        PSET (i, 0)
        DRAW "nu4 nd4 nr4 nl4"
        NEXT i
        VIEW PRINT 1 * rw TO 2 * rw: CLS 2
        IF nombregraf$ = "" THEN
        PRINT "NOMBRE O REFERENCIA PARA EL GRAFICO (max. 76 letras)"
        LINE INPUT nombregraf$: CLS 2
        ELSE
        pulsedato 0, "NUEVO NOMBRE PARA EL GRAFICO (S/N)", "sSnN", CAMGRA$: CLS 2
        IF CAMGRA$ = "S" OR CAMGRA$ = "s" THEN
        PRINT "NOMBRE O REFERENCIA PARA EL GRAFICO (max. 76 letras)"
        LINE INPUT nombregraf$: CLS 2
        END IF
        END IF
SELECT CASE selecciongraf$
CASE "1":IF iswFT <> 1 THEN
        VARIABLE$ = "X1"
        ELSE
        VARIABLE$ = "error"
        RESPUESTA$ = "ERROR vs TIEMPO"
        END IF
        FOR j = 1 TO tray
        PSET (0, x(j, 0))
        FOR i = 0 TO puntofin(j)
        LINE -(deltat * i, x(j, i)), 7
        NEXT i
        NEXT j
CASE "2":IF iswFT <> 1 THEN
        VARIABLE$ = "X2"
```

```
        ELSE
        VARIABLE$ = "derivada del error"
        RESPUESTA$ = "DERIVADA del ERROR vs TIEMPO"
        END IF
        FOR j = 1 TO tray
        PSET (0, y(j, 0))
        FOR i = 0 TO puntofin(j)
        LINE -(deltat * i, y(j, i)), 7
        NEXT i
        NEXT j
 CASE "3":IF iswFT <> 1 THEN
        VARIABLE$ = "X3"
        ELSE
        VARIABLE$ = "2da. derivada del error"
        RESPUESTA$ = "2da. DERIVADA del ERROR vs TIEMPO"
        END IF
        FOR j = 1 TO tray
        PSET (0, z(j, 0))
        FOR i = 0 TO puntofin(j)
        LINE -(deltat * i, z(j, i)), 7
        NEXT i
        NEXT j
 CASE "4":REDIM mp(tray)
        VARIABLE$ = "SALIDA"
        RESPUESTA$ = "SALIDA vs TIEMPO"
        FOR j = 1 TO tray
        mp(j) = 0
        PSET (0, Resc - x(j, 0))
        FOR i = 0 TO puntofin(j)
        salida = Resc + Rrp * deltat * i - x(j, i)
        LINE -(deltat * i, salida), 7
        IF mp(j) < salida THEN mp(j) = salida: tp(j) = deltat * i
        NEXT i
        NEXT j
CASE "5": GRAFICO = 1
END SELECT
        VIEW PRINT 1 * rw TO 3 * rw
        PRINT TAB(10); "GRAFICO: "; VARIABLE$; " versus  TIEMPO"
        PRINT nombregraf$
        VIEW PRINT 22 * rw TO 24 * rw
        PRINT TAB(25); "1 division vertical = "; paso; " unidades"
        PRINT TAB(25); "1 division horizontal ="; pasot; " unidades"
        VIEW PRINT 24 * rw TO 24 * rw
        indato 20, "maximo punto de visualizacion vertical =", m, 0, 0: CLS 2
        IF m <> 0 THEN indato 20, "maximo punto de visualizacion horizontal =", t, 0, 0: CLS 2
LOOP UNTIL m = 0
        numerodepasadas = 0
        VIEW: CLS
        VIEW PRINT: CLS
END IF
LOOP UNTIL selecciongraf$ = "5"
END SUB
```

REFERENCIAS

[1] BARAJAS, L. y ALVARADO, P., "Simulación Digital de Sistemas de Control", Anales de las Jornadas en Ingeniería Eleéctrica y Electrónica, Escuela Politécnica Nacional, Julio de 1990.

[2] OGATA, K., "Ingeniería de Control Moderna", Prentice Hall Internacional, 1975.

[3] JOCIC, L. B., "Planar Regions of Attraction", IEEE Transactions on Automatic Control, Junio de 1982.

[4] FLORES, C. y CALDERON, CH., "Gráficos Tridimensionales", Anales de las Jornadas en Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Junio de 1989.

BIBLIOGRAFIA

1. BARAJAS, L. y ALVARADO, P., "Simulación Digital de Sistemas de Control", Anales de las Jornadas en Ingeniería Eleéctrica y Electrónica, Escuela Politécnica Nacional, Julio de 1990.

2. DISTEFANO III, S. W., "Retroalimentación y Sistemas de Control", McGraw Hill México, 1978.

3. DORF, R. C., "Sistemas Automáticos de Control", Fondo Educativo Latinoamericano, 1978.

4. EVELEIGH, V. W., "Introduction to Control Sistem Design", McGraw Hill Book Company, 1972.

5. FLORES, C. y CALDERON, CH., "Gráficos Tridimensionales", Anales de las Jornadas en Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Junio de 1989.

6. MICROSOFT, "QBasic, Basic Language Reference", Microsoft Corp., 1985

7. MICROSOFT, "QBasic, Programming in Basic", Microsoft Corp., 1987

8. OGATA, K., "Ingeniería de Control Moderna", Prentice Hall Internacional, 1975.

9. RAVEN, F. H., "Automatic Control Enginering", McGraw Hill Kogakusha, 1975.