

**ESCUELA POLITECNICA NACIONAL**

**FACULTAD DE INGENIERIA ELECTRICA**

**SIMULACION DIGITAL DE SISTEMAS  
DE CONTROL**

**Tesis previa a la obtención del Título de  
Ingeniero en Electrónica y Control**

**PABLO ALVARADO POLO**

**Marzo - 1988**

**QUITO - ECUADOR**



Certifico que bajo mi dirección  
la presente Tesis fue realizada  
en su totalidad por el señor  
PABLO ALVARADO POLO



~~Ing. Luis Barajas.~~

DIRECTOR

Agradezco a todas las personas que de una u otra manera colaboraron en la realización de este trabajo, de manera especial a quien lo dirigió Ingeniero Luis Barajas.

A mis hijos

## INDICE

|   | Pág. |
|---|------|
| INTRODUCCION  | 1    |
| CAPITULO I.   |      |
| ASPECTOS GENERALES  | 4    |
| 1.1 FUNDAMENTOS DE LA SIMULACION DE PROCESOS                  | 4    |
| 1.1.1 Definiciones  | 4    |
| Procesos  | 4    |
| Plantas   | 4    |
| Sistemas  | 4    |
| Perturbaciones  | 5    |
| Sistemas de control realimentado                              | 5    |
| Servomecanismos   | 5    |
| Sistemas de regulación automática                             | 6    |
| Sistemas de control de procesos                               | 6    |
| Sistemas de control de lazo cerrado                           | 7    |
| Sistemas de control de lazo abierto                           | 9    |
| Sistemas de control adaptados                                 | 10   |
| Sistemas de control con aprendizaje                           | 11   |
| Ejemplos de sistemas de control                               | 12   |
| 1.1.2 La naturaleza de la simulación                          | 14   |
| 1.1.3 Sistemas, modelos y simulación                          | 15   |
| 1.2 METODOLOGIA DE LA SIMULACION DIGITAL DE PROCESOS          | 20   |
| 1.2.1 Requerimientos generales de un sistema de control       | 20   |
| 1.2.2 Problemas básicos en el proyecto de sistemas de control | 20   |
| Análisis  | 21   |
| Proyecto  | 21   |
| Síntesis  | 22   |
| Enfoque básico del proyecto de sistemas de control            | 22   |

|   | Pág. |
|---|------|
| 1.2.3 Metodología para la simulación digital  | 23   |
| <br>  |      |
| CAPITULO II.  |      |
| MODELOS DE SIMULACION Y BASE  |      |
| MATEMATICA  | 32   |
| <br>  |      |
| 2.1 FUNCIONES ASOCIADAS A UN SISTEMA  | 32   |
| 2.1.1 Funciones lineales  | 33   |
| 2.1.2 Funciones no lineales   | 35   |
| Alinealidad limitador o de saturación   | 37   |
| Alinealidad comparador  | 37   |
| Alinealidad comparador con histéresis   | 38   |
| Alinealidad rectificador de media onda  | 39   |
| Alinealidad valor absoluto  | 39   |
| Alinealidad zona muerta   | 39   |
| Alinealidad de juego  | 39   |
| Alinealidades cuadrador y raíz cuadrada   | 40   |
| Alinealidades logarítmica y exponencial   | 40   |
| Alinealidades seno y coseno   | 40   |
| 2.1.3 Otras funciones   | 54   |
| Multiplicación y división entre variables   | 54   |
| Funciones lógicas   | 54   |
| Monoestable   | 55   |
| Interruptores analógicos  | 55   |
| Retardo en el tiempo  | 55   |
| 2.2 EL COMPUTADOR ANALOGICO: MODELOS DE SIMULACION                                  | 64   |
| 2.2.1 Simbología utilizada  | 64   |
| 2.2.2 Equipo de medida e impresión  | 66   |
| 2.2.3 Interconexiones en el panel de programación                                   | 67   |
| 2.2.4 Controles del computador  | 67   |
| 2.2.5 Diagramas de simulación analógica   | 68   |
| 2.3 TRATAMIENTO DEL MODELO DE SIMULACION ANALOGICA EN UN COMPUTADOR DIGITAL         | 70   |
| 2.3.1 Tratamiento discreto del modelo de simulación basado en el algoritmo de Euler | 71   |
| 2.4 ALGORITMO DE RUNGE-KUTTA  | 73   |

|   | Pág. |
|---|------|
| CAPITULO III.   |      |
| DESARROLLO DEL PROGRAMA                                       | 76   |
| 3.1 ENTRADA DE DATOS  | 81   |
| Número de salidas   | 84   |
| Matriz de incidencia  | 84   |
| Definición de los módulos                                     | 87   |
| Matriz PM   | 88   |
| Definición de los subíndices de las entradas                  | 88   |
| Datos de tiempo   | 88   |
| 3.2 PROGRAMA DE SIMULACION                                    | 89   |
| 3.2 PRESENTACION DE RESULTADOS                                | 101  |
| CHDIR   | 102  |
| REVISAR   | 102  |
| IMPRIMIR  | 102  |
| GRAFICAR  | 102  |
| PGRAFIC   | 102  |
| ALMACENAR   | 102  |
| OTROS   | 103  |
| OTROS/ CAMBIAR  | 103  |
| OTROS/ ALMACENAR  | 103  |
| OTROS/ MENU   | 103  |
| OTROS/ SALIR  | 103  |
| OTROS/ FIN  | 103  |
| <br>  |      |
| CAPITULO IV.  |      |
| EJEMPLOS, CONCLUSIONES,<br>RECOMENDACIONES                    | 105  |
| 4.1 EJEMPLOS DE SIMULACION                                    | 105  |
| 4.1.1 Ejemplo 1. Sistema lineal con retardo                   | 105  |
| 4.1.2 Ejemplo 2. Ecuación de Van der Pol                      | 110  |
| 4.1.3 Ejemplo 3. Sistema de control Si-No con His-<br>téresis | 113  |
| 4.1.4 Ejemplo 4. Sistema de control con ganancia<br>no lineal | 116  |
| 4.1.5 Ejemplo 5. Sistema lineal con perturbación              | 121  |

|  | Pág. |
|--|------|
| 4.1.6 Ejemplo 6. Circuito generador de funciones | 126  |
| 4.1.7 Ejemplo 7. Detector de secuencia           | 128  |
| 4.2 CONCLUSIONES                                 | 134  |
| 4.3 RECOMENDACIONES                              | 136  |
| <br>   |      |
| ANEXOS   |      |
| <br>   |      |
| ANEXO A. MANUAL DE UTILIZACION DEL PROGRAMA      | 138  |
| ANEXO B. OPCIONES ADICIONALES DE LOTUS 123       | 164  |



INTRODUCCION

## INTRODUCCION

A través de la historia, desde las primeras aplicaciones del control con realimentación, como los mecanismos regulados con flotador desarrollados en Grecia en el periodo 0 a 300 a.c., el objeto de controlar un sistema ha tenido un desarrollo fascinante. El primer trabajo significativo en control automático fué el regulador centrífugo de James Watt para el control de velocidad de una máquina de vapor en el siglo dieciocho. A partir de este momento, los pasos relevantes en el desarrollo de la teoría de control dedican a obtener métodos de análisis y síntesis que permitan estimar la estabilidad absoluta y relativa de un sistema sin tener que resolver las ecuaciones diferenciales que lo describen. Ejemplos de tales métodos son el lugar geométrico de las raíces, el análisis de respuesta de frecuencia, el análisis con la función descriptiva para sistemas no lineales, etc. Todas estas técnicas comprenden la llamada Teoría de Control Clásica. Durante este desarrollo las técnicas de análisis en el dominio del tiempo, quedan relegadas a un segundo plano ya que involucran serias dificultades matemáticas en la evaluación de las ecuaciones diferenciales que describen un sistema.

Como las plantas modernas con muchas entradas y salidas se van haciendo más y más complejas, la descripción de tales sistemas de control requieren de una gran cantidad de ecuaciones. La teoría de control clásica que dedica su estudio a sistemas lineales de entrada y salida única, se vuelve absolutamente impotente ante sistemas de múltiples entradas y salidas o que eventualmente incluyan alinealidades.

Desde aproximadamente 1960, se ha desarrollado la Teoría de Control Moderno para afrontar la complejidad creciente de las plantas y sus necesidades rigurosas en exactitud, peso y costo en aplicaciones industriales, militares, médicas, etc.

Con el advenimiento de poderosas computadoras analógicas, digitales o híbridas que pueden realizar complejos cálculos, el uso de las mismas en el proyecto de sistemas de control y en su control de operación se está convirtiendo ahora en práctica habitual.

Quizá el más efectivo de los métodos modernos de análisis y síntesis se basa en la simulación del sistema. Principalmente la simulación analógica, utilizando para ello computadores análogos-híbridos los cuales permiten obtener con gran precisión y velocidad las respuestas de un sistema. Sin embargo, debido a las dificultades de nuestro medio que imposibilitan disponer de dichos equipos en los lugares donde se los requiere, nos vemos obligados a descartar esta posibilidad. Por otro lado, dada la fácil disponibilidad de computadoras electrónicas digitales, el análisis y diseño de sistemas de control con la ayuda de estos equipos es la alternativa más conveniente.

En realidad, el uso de computadoras digitales para simulación de sistemas no es algo nuevo, pues existen gran cantidad de programas desarrollados para esta finalidad. Sin embargo, estos programas presentan la dificultad de que no están diseñados para aplicaciones generales, sino más bien para resolver problemas específicos, a más de que son de difícil adquisición por cuestiones de costo y comercialización. Ante estas dificultades, se presenta el problema de desarrollar un programa de propósitos generales que facilite la simulación de sistemas. Considerando las ventajas del computador analógico-híbrido en este sentido, este trabajo se dedica al desarrollo de un programa que permite "convertir" al computador digital en un poderoso computador analógico-híbrido. Así, los datos del sistema por analizar se introducen al computador teniendo como base un diagrama de simulación analógica, obteniéndose como resultados valores numéricos o gráficos de cualesquiera de los puntos de interés. Este es el objetivo fundamental de la tesis, el cual se cumple en gran medida.

El programa implementado para un computador personal IBM o compatible dispone de muchas de las bondades de un computador análogo-híbrido e inclusive supera las características de este último en lo referente a precisión, capacidad, entrada de datos y presentación de resultados. Comprende un total de 25 módulos o funciones matemáticas entre las que se cuentan funciones lineales, no lineales, lógicas y otras. La entrada de datos se basa en una codificación adecuada del diagrama de simulación, son infinitas las posibilidades de generación de funciones de entrada o prueba ya que utiliza líneas de programación BASIC para el propósito, el proceso de cálculo se cumple con celeridad gracias a que se desarrolla en un lenguaje compilado de alto nivel (GW-BASIC Compilado) y la presentación de resultados realizada en la hoja electrónica LOTUS 123 facilita el acceso a tablas de valores, resultados gráficos de alta calidad, análisis estadístico de las respuestas, operaciones adicionales entre resultados, etc. Los menús de ayuda presentes en todos los pasos del programa, facilitan enormemente su utilización.

Nuestro objetivo no se centra en el desarrollo de este programa ya que adicionalmente se presentan las bases teóricas necesarias (Capítulos I y II) que brindan una metodología alternativa para el análisis y diseño de sistemas, considerando a su modelo como un conjunto de módulos o funciones matemáticas. En el Capítulo III, se presenta la descripción del programa en forma detallada, se dá énfasis en lo que se refiere a la estructura de arreglos y archivos internamente generados. En el Capítulo IV se describen algunos ejemplos con la finalidad de demostrar las características del programa más que como problemas de control. Finalmente, los Anexos A y B presentan una guía completa sobre la instalación y la utilización del programa.

CAPITULO I

## CAPITULO I

### ASPECTOS GENERALES

Con el objeto de adentrarnos en el problema de la simulación de procesos presentaremos en este capítulo algunos de los conceptos fundamentales sobre procesos y sistemas, la naturaleza de la simulación y una metodología a seguirse para obtener la simulación de un sistema. En resumen, los conceptos descritos en este capítulo permitirán enmarcarnos correctamente dentro del problema general de la simulación y algunos de los métodos de enfrentarlo.

#### 1.1 FUNDAMENTOS DE LA SIMULACION DE PROCESOS

##### 1.1.1 Definiciones.

Empezaremos dando las definiciones de la terminología necesaria para describir un proceso.

**Procesos.** El diccionario Merriam-Webster define un proceso como una operación o desarrollo natural, progresivamente continuo, caracterizado por una serie de cambios graduales que llevan de un estado a otro de un modo relativamente fijo y que tienden a un determinado resultado final; o una operación artificial o voluntaria, progresivamente continua que consiste de una serie de acciones controladas o movimientos dirigidos sistemáticamente hacia determinado resultado o fin. Los procesos, por ejemplo, son químicos, económicos o biológicos.

**Plantas.** Una planta es un equipo, quizá simplemente un juego de piezas de una máquina funcionando juntas, cuyo objeto es realizar una operación determinada.

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

sistema no está limitado a los objetivos físicos. El concepto de sistema puede ser ampliado a fenómenos abstractos y dinámicos, como la economía. Por tanto, hay que interpretar el término *sistema* como referido a sistemas físicos, biológicos, económicos, etc.

**Perturbaciones.** Una perturbación es una señal que tiende a afectar adversamente el estados de un sistema. Si la perturbación se genera dentro del sistema, se denomina *interna*, mientras que una perturbación *externa* se genera fuera del sistema y constituye una entrada del mismo.

**Sistemas de control realimentado.** El control de realimentación es una operación que, en presencia de perturbaciones, tiende a reducir la diferencia entre la salida y la entrada de referencia de un sistema (o un estado deseado, arbitrariamente variado) y que lo hace sobre la base de esta diferencia. Aquí solamente se consideran perturbaciones a las no previsibles (es decir, las desconocidas de antemano), pues para las que pueden ser predichas o conocidas de antemano siempre se puede incluir una compensación dentro del sistema de modo que sean innecesarias las mediciones.

Un sistema de control realimentado es aquel que ejecuta la acción de control de realimentación midiendo la salida, comparandola con la referencia y utilizando la diferencia como parámetro de control, para mantener una relación preestablecida entre ambas.

Es de notar que los sistemas de control realimentado no están limitados al campo de la ingeniería, sino que se los puede encontrar en áreas ajenas a la misma, como la economía y la biología. Por ejemplo, el organismo humano, en un aspecto, es análogo a una intrincada planta química con una enorme variedad de operaciones unitarias. El control de procesos de esta red de transporte y reacciones químicas involucra una variedad de lazos de control. De hecho, el organismo humano es un sistema de control realimentado extremadamente complejo.

**Servomecanismos.** Un servomecanismo es un sistema de control realimentado en el cual la salida es alguna posición velocidad o aceleración mecánica. Por tanto, los términos *sistema de control de servomecanismo* o *de posición* (o *de velocidad*

o de aceleración) son sinónimos. Los servomecanismos son extremadamente usados en la industria moderna. Por ejemplo, el funcionamiento totalmente automatizado de máquinas herramienta, con su instrucción programada, son cumplidos por el uso de servomecanismos.

Sistemas de regulación automática. Un sistema de regulación automática es un sistema de control realimentado en el que la entrada de referencia o la salida deseada son o bien constantes o varían lentamente en el tiempo, y donde la tarea fundamental consiste en mantener la salida en el valor deseado a pesar de las perturbaciones presentes.

Un sistema de calefacción domiciliario en el que un termostato es el control, constituye un ejemplo de sistema de regulación automático. En este sistema, se compara el ajuste del termostato (la temperatura deseada) con la temperatura efectiva de la habitación. Una perturbación en este sistema la constituye un cambio en la temperatura exterior. El objetivo es mantener la temperatura deseada en la habitación a pesar de las variaciones en la temperatura exterior. Hay muchos otros ejemplos de sistemas de regulación automáticos, algunos de los cuales son el control automático de presión y de magnitudes eléctricas como corriente, tensión y frecuencia.

Sistemas de control de procesos. Un sistema de regulación automático en el que la salida es una variable como temperatura, presión, flujo, nivel de líquido o pH, se llama sistema de control de procesos. El control de procesos tiene amplia aplicación en la industria. Frecuentemente se usan en estos sistemas controles programados como el de temperatura de un horno de calentamiento en el que la temperatura del horno es controlada de acuerdo con un programa preestablecido. Por ejemplo, el programa preestablecido puede consistir en elevar la temperatura a determinado valor durante determinado intervalo de tiempo, y luego reducir a otra temperatura prefijada también durante un período de tiempo prefijado. En un programa como este, el punto de ajuste varía de acuerdo con un programa preestablecido. Entonces el control funciona para mantener la temperatura del horno cerca del punto fijado, variable. Se hace notar que la mayoría de los sistemas de control de



procesos incluyen servomecanismos como parte integral.

Sistemas de control de lazo cerrado. Un sistema de control de lazo cerrado es aquel en el que la señal de salida tiene efecto directo sobre la acción de control. Esto es, los sistemas de control de lazo cerrado son sistemas de control realimentado.

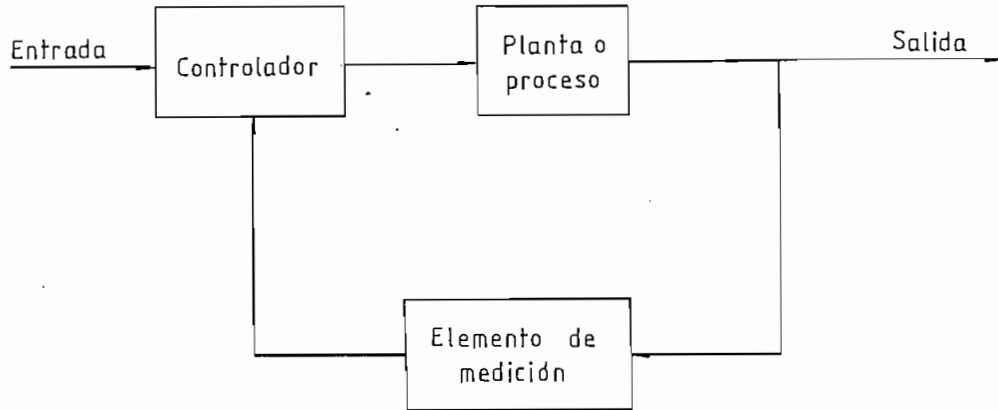


Fig. 1.1. Sistema de control de lazo cerrado.

La señal de error actuante, que es la diferencia entre la señal de entrada y la realimentación (que puede ser la señal de salida o una función de la señal de salida y sus derivadas), entra al detector o control de manera de reducir el error y llevar la salida del sistema al valor deseado. En otras palabras, el término lazo cerrado implica el uso de acción de realimentación para reducir el error del sistema. La Fig. 1.1 muestra la relación entrada-salida de un sistema de control de lazo cerrado. Una figura como ésta recibe el nombre de *diagrama de bloques*. Para ilustrar el concepto de sistemas de control de lazo cerrado, se considera el sistema térmico de la Fig. 1.2. Aquí actúa un ser humano como controlador. Su intención es mantener la temperatura del agua caliente a un valor determinado. El termómetro instalado en el caño de salida del agua caliente indica la temperatura efectiva. Esta temperatura es la salida del sistema. Si el operador observa el termómetro y descubre que la temperatura es superior a la deseada, reduce la entrada de vapor para bajar esa temperatura. Es bien posible que la temperatura llegue ahora a ser excesiva-

mente baja, en cuyo caso hará falta repetir la secuencia de operaciones en sentido contrario.

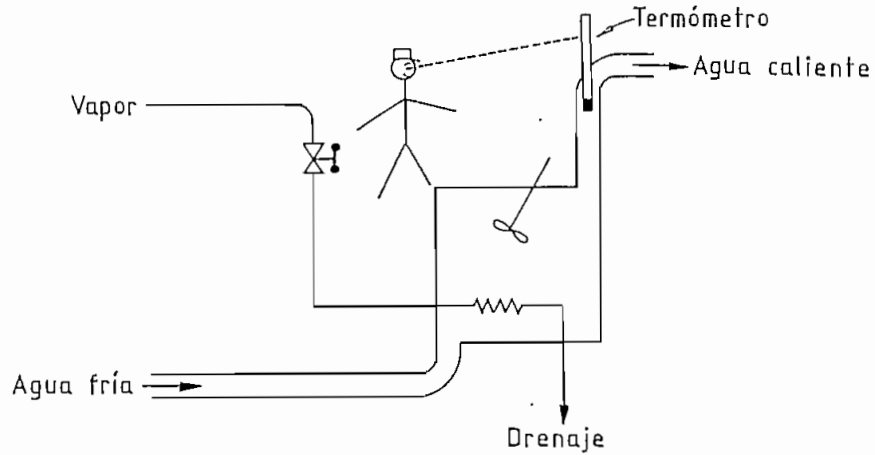


Fig. 1.2 Control de realimentación manual en un sistema térmico.

Esta acción de control está basada en la operación de lazo cerrado. Como tanto la realimentación de la salida (temperatura del agua) para comparación con la entrada de referencia, como la acción de control tienen lugar a través de los actos del operador, este es un sistema de control de lazo cerrado. Se podría denominar a este sistema de realimentación manual o de lazo cerrado manual.

Si se usa un controlador detector automático en reemplazo del operador humano, como se ve en la Fig. 1.3, el sistema de control se vuelve automático, es decir, un sistema de control de realimentación automático o de lazo cerrado automático. La posición del dial del controlador automático fija la temperatura deseada. La salida, la temperatura efectiva del agua caliente, detectada por el dispositivo de medición de temperatura, es comparada con la temperatura deseada para generar una señal de error que actúe corrigiendo. Al realizar esto, se convierte la temperatura de salida en las mismas unidades que la entrada (punto de ajuste) por su transductor (un transductor es un dispositivo que convierte una señal de una forma a otra). La señal de error producida en el controlador automático es amplificada y la salida del controlador es enviada a la válvula de provisión de vapor para corregir la temperatura que toma el agua. Si no hay error, no hace falta modificar la apertura de

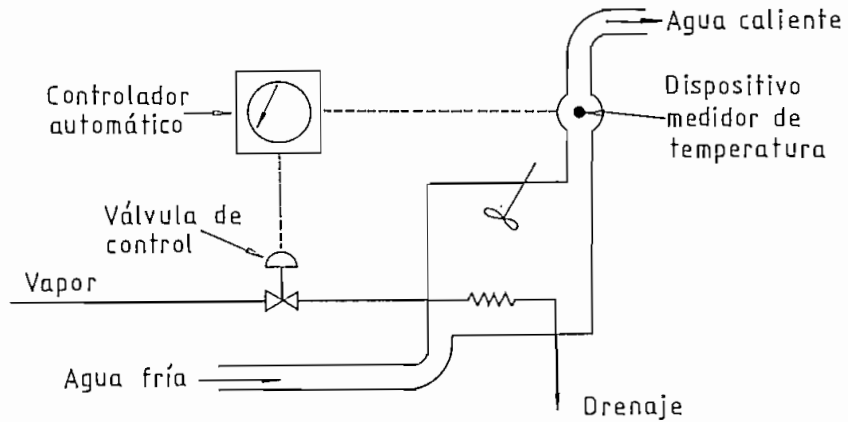


Fig. 1.3 Control de realimentación automática en un sistema térmico.

la válvula.

En los sistemas considerados aquí, las variaciones de temperatura ambiente, la temperatura del agua fría de entrada, etc.. pueden reputarse como perturbaciones externas.

Los sistemas de realimentación automática y manual citados anteriormente operan en forma similar. Los ojos del operador constituyen el análogo del dispositivo de medición de error; su mente, la del controlador automático y sus músculos el análogo del elemento actuante.

El control de un sistema complejo por un operador humano no es eficaz, por las muchas interrelaciones entre las diversas variables. Nótese que aún en un sistema simple, un detector automático elimina cualquier error humano de operación. Si se necesita control de alta precisión, el control debe ser automático.

Sistemas de control de lazo abierto. Los sistemas de control de lazo abierto son sistemas de control en los que la salida no tiene efecto sobre la acción de control. Es decir, en un sistemas de control de lazo abierto la salida ni se mide ni se realimenta para comparación con la entrada. La Fig. 1.4 muestra la relación entrada-salida de tal sistema. Un ejemplo práctico es la máquina de lavar. El remojo, lavado y enjuague en la máquina de lavar se cumplen sobre una base de tiempos. La máquina no mide la señal de salida, es decir, la limpieza de la ropa.

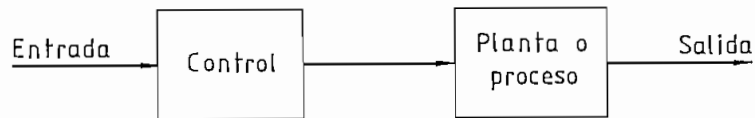


Fig. 1.4 Sistema de control de lazo abierto.

En un sistema de control de lazo abierto, no se compara la salida con la entrada de referencia. Por tanto, para cada entrada de referencia corresponde una condición de operación fijada. Así, la exactitud del sistema depende de la calibración (los sistemas de control de lazo abierto deben ser cuidadosamente calibrados y deben mantener esa calibración a lo largo del tiempo). En presencia de perturbaciones un sistema de control de lazo abierto no cumple su función asignada. Nótese que cualquier sistema de control que funciona sobre una base de tiempos es de lazo abierto. Por ejemplo, el control de tráfico por señales actuantes en función de tiempos es otro caso de control de lazo abierto.

Una ventaja del sistema de control de lazo cerrado sobre el de lazo abierto es que el uso de la realimentación hace al sistema en su respuesta, relativamente insensible a perturbaciones externas y a variaciones internas de parámetros del mismo. De este modo es posible utilizar componentes relativamente inexactos y económicos y lograr la exactitud de control requerida en determinada planta; mientras esto sería imposible en el caso de lazo abierto.

Desde el punto de vista de estabilidad, en el sistema de control de lazo abierto es más fácil de lograr, ya que la estabilidad no constituye un problema importante. Por otro lado, en los sistemas de lazo cerrado la estabilidad siempre constituye un problema de importancia, por la tendencia a sobre corregir errores, que puede producir oscilaciones de amplitud constante o variable.

Sistemas de control adaptados. Las características dinámicas de la mayoría de los sistemas de control no son constantes por diversas razones, como el deterioro de los com-

ponentes al transcurrir el tiempo o las modificaciones en parámetros o en el medio ambiente (por ejemplo, cambios en la masa y en las condiciones atmosféricas en un sistema de control de vuelo). Aunque en un sistema de control realimentado se atenúan los efectos de pequeños cambios en las características dinámicas, si las modificaciones en los parámetros del sistema y en el medio son significativas, un sistema para ser satisfactorio ha de tener la capacidad de adaptación. La adaptación implica la capacidad de autoajustarse o automodificarse de acuerdo con modificaciones imprevisibles del medio o estructura. Los sistemas de control que tienen algún grado de capacidad de adaptación, se denominan sistemas de control adaptados o adaptivos.

En un sistema de control adaptado, las características dinámicas deben estar identificadas en todo momento de manera que los parámetros de control o detección puedan ajustarse para mantener el funcionamiento óptimo. Este concepto tiene una buena dosis de atracción para el ingeniero diseñador de sistemas, ya que un sistema de control adaptivo, además de acomodarse a los cambios ambientales, también lo hace ante moderados errores de proyecto de ingeniería o incertidumbres y compensa la eventual falla de componentes menores del sistema, aumentando, por tanto, la confiabilidad de todo el sistema.

Sistemas de control con aprendizaje. Muchos sistemas de control que aparentemente son de lazo abierto pueden ser convertidos a sistemas de control de lazo cerrado si se considera un detector o control humano que compara la entrada y la salida y realiza las acciones correctivas basadas en la diferencia resultante, o error.

Si intentamos analizar sistemas de control de lazo cerrado con operación humana, encontramos el difícil problema de escribir ecuaciones que describen el comportamiento de un ser humano. En este caso uno de los muchos factores que lo complican es la capacidad de aprender del operador humano. A medida que éste va adquiriendo experiencia, se convierte en un mejor elemento de control, y esto debe ser considerado al analizar el sistema. Un sistema de control que incluye ciertas capacidades de aprender, similares a las que tiene un ser

humano recibe el nombre de sistema de control con aprendizaje.

Ejemplos de sistemas de control. Con el fin de ilustrar los diferentes tipos de procesos involucrados dentro de los sistemas de control, vamos a presentar varios ejemplos ilustrativos de sistemas de control de lazo cerrado.

- Sistemas de control de presión. Estos sistemas controlan la presión de un fluido en un proceso químico. Ejemplo de este sistema de control sería el control de presión de un horno controlado por la posición de un dispositivo regulador de escape

- Sistemas de control de velocidad. Cualquier sistema que esté dedicado al control de la velocidad lineal o angular de una determinada planta es llamado sistema de control de velocidad. Un ejemplo de este tipo de sistema es el regulador de Watt para máquinas de vapor (1).

- Sistemas de control numérico. El control numérico es un método de controlar sistemas usando números. Por ejemplo se puede controlar el movimiento de una cabeza cortante con la información binaria contenida en una cinta. En este tipo de control se convierten los valores numéricos simbólicos en valores físicos (dimensiones o cantidades) por medio de señales eléctricas (o de otro tipo), que se traducen en movimiento lineal o circular. Estas señales son: o bien digitales (pulsos) o analógicas (tensiones continuamente variables en el tiempo).

- Sistemas de control por computadora. En el caso de control por computadora se ingresa periódicamente al computador la información numérica sobre el estado del sistema. La computadora realiza entonces los complejos cálculos requeridos para determinar las condiciones óptimas de funcionamiento. A su vez, mediante adecuadas interfaces se envían los datos obtenidos por el computador a los diferentes actuadores del sistema para corregir los errores y mantener las salidas dentro del valor deseado. El control por computadora se utiliza para el control de complejos sistemas, los cuales involucran muchas variables a ser controladas las cuales están sujetas a varias perturbaciones.

- Sistemas de control de tráfico. Como se indicó anteriormente, el control de tráfico por medio de señales actuadas sobre una base de tiempos, constituye un sistema de control de lazo abierto. Sin embargo, si continuamente se mide la cantidad de automotores en espera en cada señal de tráfico en un área congestionada y esa información se lleva a una computadora central que controla las señales de tráfico, el sistema se convierte en un sistema de lazo cerrado. El movimiento de tráfico en redes es muy complejo porque la variación en volumen de tráfico depende mucho de la hora y día de la semana, así como de muchos otros factores. De hecho, minimizar el tiempo medio de espera es un problema de control muy complejo.

- Sistemas biológicos. Sea el caso de dos especies de bacterias competidoras, cuyas respectivas poblaciones son  $x_1$  y  $x_2$ . Son competidoras en el sentido de que consumen de la misma fuente de alimentación. Bajo ciertas condiciones, las poblaciones  $x_1$  y  $x_2$  varían en el tiempo de acuerdo con las llamadas ecuaciones de competencia de Volterra (2). Llevar, con la ayuda de una droga, dentro de un tiempo determinado a un mínimo la población  $x_1$ , manteniendo al mismo tiempo la población  $x_2$  en el máximo posible, presenta un interesante problema. Este es un ejemplo de sistema biológico al que se puede aplicar la teoría de control.

- Sistemas de control de inventario. Otro ejemplo de sistemas de control de lazo cerrado lo constituye la programación industrial del ritmo de producción y nivel de inventario. Se compara el nivel de inventario real, que es la salida del sistema, con el nivel de inventario deseado, que puede ir cambiando de acuerdo con el mercado, de cuando en cuando. Si hay cualquier diferencia entre el nivel de inventario real y el deseado, se ajusta el ritmo de producción de manera que la salida siempre esté en la vecindad del nivel deseado, que se elige para hacer máxima las utilidades.

- Sistemas empresariales. Un sistema empresarial puede consistir de muchos grupos. Cada tarea asignada a un grupo, representa un elemento dinámico del sistema. Para el funcionamiento correcto de tales sistemas, hay que establecer métodos de realimentación para el control de los logros de cada

grupo. El acoplamiento mutuo entre grupos funcionales debe ser mínimo para reducir atrazos inútiles en el sistema. Cuanto menor sea ese acoplamiento más suave será el flujo de señales y de materiales. El sistema empresarial constituye un ejemplo de lazo cerrado. Un buen diseño reduce el trabajo de dirección futuro. Nótese que en este sistema las perturbaciones son la falta de personal, interrupción de las comunicaciones, errores humanos, etc.

Los ejemplos descritos, dan una idea general de los diversos campos con los cuales se involucran los sistemas de control. Por tanto, recalcamos que los sistemas de control no están limitados al campo de la ingeniería sino que se relacionan con variedad de procesos económicos, biológicos, etc.

#### 1.1.2 La naturaleza de la simulación.

La simulación de procesos no es más que el estudio de las técnicas que nos permiten imitar, o simular, la operación de varias clases de fenómenos reales. Tratándose de estudiar científicamente dichos fenómenos debemos hacer algunas consideraciones a cerca del comportamiento de estos. Estas consideraciones, que usualmente toman la forma de relaciones matemáticas o lógicas, constituyen un modelo que se utilizará para determinar cual es el comportamiento del sistema.

Si las ecuaciones que componen el modelo son sencillas, es posible utilizar métodos matemáticos analíticos para obtener la información exacta sobre las variables de interés; este método de estudio es llamado *solución analítica*. Sin embargo, muchos de los sistemas reales dan como resultados modelos bastante complejos como para ser evaluados analíticamente, por lo tanto estos deben ser analizados por medio de la simulación. En una simulación, nosotros utilizaremos métodos numéricos para evaluar el modelo matemático del sistema en un intervalo de interés.

Como ejemplo del uso de la simulación, tomemos una firma manufacturera que está considerando el ampliar su planta con



la construcción de un gran edificio pero no tiene la certeza de que las ganancias en productividad justifiquen los costos de construcción. La respuesta a la pregunta planteada puede obtenerse por medio de simulación; así, se deberá simular la operación de la planta que actualmente existe, y la operación de esta cuando sea ampliada. Los resultados obtenidos serán en definitiva, los ingresos económicos de cada uno de los casos y la diferencia entre estos indicará la conveniencia o no de la ampliación de la planta. Notemos que dentro del modelo de simulación de la nueva planta se deberán considerar factores como el incremento de popularidad de la firma, factores que en definitiva incrementan los ingresos pero cuya cuantificación es muy difícil dentro del modelo. Estos factores constituyen un primer problema en el estudio de sistemas de gran escala, sin embargo en los recientes años se han desarrollado muchos algoritmos y lenguajes de simulación que proveen gran cantidad de opciones para el análisis de estos modelos de simulación. Un segundo problema con la simulación de sistemas complejos es la gran cantidad de memoria y tiempo de computador utilizado en la evaluación de estos. De otro lado, los recientes desarrollos de computadores de gran capacidad de memoria y alta velocidad de operación permiten disminuir considerablemente los costos de computación. Finalmente, debemos descartar la idea de que el problema de simulación no es más que un ejercicio de programación. A pesar de que la mayoría de los problemas de simulación se resuelven con la ayuda de lenguajes de programación y computadores, existen algoritmos sobre la metodología de la simulación que son completamente independientes de estos últimos.

### 1.1.3 Sistemas, modelos y simulación.

En la práctica, para el análisis de grandes y complejos sistemas debemos determinar cuales son los sujetos del sistema que pueden ser objeto de un estudio particular. Por ejemplo, si estamos estudiando un banco con el fin de determinar el número de cajeros que se necesitan para proveer de un adecuado servicio a los clientes quienes tan solo realizarán operaciones

de cobro o depósitos, el sistema puede ser definido como una parte del banco consistente de los cajeros y los clientes que esperan en línea para ser atendidos. Si de otro lado, el personal de oficina y las cajas de los depósitos de seguridad son incluidos, la definición de nuestro sistema como es obvio se incrementará. Definimos el estado de un sistema como una colección de variables necesarias para describir el mismo en un tiempo particular relativo a los objetivos de estudio. En el caso del banco, ejemplos de posibles variables de estado son: el número de cajeros, el número de clientes y el tiempo al cual el cliente llega al banco. Existen dos categorías de sistemas: discretos y continuos. Un sistema discreto es uno en el cual las variables de estado cambian solo un número finito de veces en un intervalo de tiempo. Un banco es un ejemplo de un sistema discreto ya que sus variables de estado, como por ejemplo el número de clientes, cambia solo cuando un nuevo cliente arriba al mismo. Un sistema es continuo cuando las variables de estado cambian continuamente con respecto al tiempo. Un aeroplano en movimiento es un ejemplo de un sistema continuo ya que sus variables de estado tales como posición o velocidad cambian en cada instante de tiempo. Pocos sistemas en la práctica son solamente continuos o discretos, pero basándose en los cambios predominantes de las variables de estado, usualmente es posible clasificar a un sistema dentro de estos dos grupos.

Algunas veces se desea estudiar a un sistema para observar las relaciones entre los diversos componentes del mismo o predecir el comportamiento bajo diferentes condiciones de operación. El estudio de un sistema en lo posible se hará experimentando sobre el mismo, sin embargo, la experimentación sobre un sistema en funcionamiento puede ser difícil debido a cuestiones de costo, poca efectividad o alteraciones drásticas en la operación del sistema. Considerando el ejemplo del banco, supongamos que para estudiar el efecto de reducir el número de cajeros efectivamente los reducimos temporalmente; esto podría causar un incremento significativo en el número de clientes alineados en espera de servicio lo cual afectaría fundamentalmente a las relaciones banco-cliente. Por la imposibili-

dad de experimentar sobre un sistema en funcionamiento o simplemente porque se está diseñando un nuevo sistema, utilizaremos un sistema analítico o un modelo del sistema para fines de estudio. Definiremos un *modelo* como una representación de un sistema desarrollado con el propósito de estudiar al mismo. El modelo debe ser suficientemente detallado o válido para permitir el análisis de respuesta bajo ciertas consideraciones que luego se aplicarán en la práctica.

Existen dos tipos de modelos de simulación: matemáticos y físicos. Un modelo matemático, describe al sistema como un conjunto de ecuaciones cuya solución indicará las características del mismo. Un modelo físico no es más que un modelo a escala del sistema, por ejemplo, el modelo a escala de un aeroplano que es probado en un túnel de viento. Dentro de los modelos matemáticos, distinguimos algunas clases tales como: estáticos o dinámicos, determinísticos o estocásticos y discretos o continuos. Un modelo de simulación estático corresponde a la representación de un sistema en un instante de tiempo, los modelos de simulación de Monte Carlo (3) son de este tipo. Un modelo de simulación dinámico es la representación de un sistema en un intervalo de tiempo, por ejemplo, el modelo del comportamiento de un máquina en un intervalo de tiempo durante el cual se le somete a perturbaciones. Un modelo de simulación es llamado determinístico si no contiene variables aleatorias. En un modelo determinístico, existe un único set de datos de salida para un determinado set de datos de entrada. De otro lado, un modelo de simulación es estocástico si contiene una o más variables aleatorias. Los datos de salida de un modelo estocástico son aleatorios lo que permite tener tan solo una estimación de las verdaderas características del sistema. En el modelo de simulación del banco, serán variables aleatorias la cantidad de clientes que ingresan al banco así como el tiempo al cual lo hacen; cualquiera de estas variables cumplirá con una determinada distribución de probabilidad. Por último, los modelos discretos o continuos serán consecuencia de sistemas discretos o continuos descritos anteriormente.

Una última clasificación de sistemas se basa en la forma

que tome el modelo matemático del mismo; así, un sistema se dice lineal si las ecuaciones de su modelo son lineales. Si el sistema es dinámico, las ecuaciones que lo describe serán ecuaciones diferenciales. Una ecuación diferencial es lineal si los coeficientes son constantes o funciones únicamente de la variable independiente. La propiedad más importante de los sistemas lineales es que se les puede aplicar el principio de superposición. El principio de superposición establece que la respuesta producida por la aplicación simultánea de dos funciones excitadoras distintas, es la suma de las dos respuestas individuales. Por tanto, para sistemas lineales se puede calcular la respuesta a diversas entradas, tratando una entrada por vez, y añadiendo o sumando los resultados. Este principio es el que nos permite construir complicadas soluciones a las ecuaciones diferenciales lineales, con la ayuda de soluciones simples.

Los sistemas dinámicos que son lineales y están constituidos por componentes concentrados e invariantes en el tiempo, pueden ser descritos por ecuaciones diferenciales lineales, invariantes en el tiempo. Estos sistemas reciben el nombre de lineales e invariantes en el tiempo (o lineales de coeficientes constantes). Los sistemas representados por ecuaciones diferenciales, cuyos coeficientes son funciones del tiempo, reciben el nombre de sistemas lineales variables en el tiempo. Ejemplo de un sistema variable en el tiempo es el sistema de control de un vehículo espacial (la masa del vehículo cambia debido al consumo de combustible y la fuerza de gravedad se modifica a medida que el vehículo se aleja de la tierra).

Sistemas no lineales son aquellos representados por ecuaciones no lineales. Ejemplos de ecuaciones no lineales son:

$$y = \text{sen } x$$

$$y = x^2$$

$$z = x^2 + y^2$$

(en la última ecuación  $z$  es función no lineal de  $x$  e  $y$ ).

Una ecuación diferencial recibe el nombre de no lineal si

no es lineal. Ejemplos de ecuaciones diferenciales no lineales son:

$$d^2x/dt^2 + (dx/dt)^2 + x = A \sin \omega t$$

$$d^2x/dt^2 + (x^2 - 1) dx/dt = 0.$$

$$d^2x/dt^2 + dx/dt + x + x^3 = 0$$

Aunque muchas relaciones físicas son frecuentemente representadas por ecuaciones lineales, en la mayor parte de los casos realmente las relaciones no son muy lineales. De hecho, un estudio cuidadoso de los sistemas físicos indica que aún los denominados sistemas lineales son realmente lineales solamente en restringidos rangos de operación. En la práctica, muchos sistemas electromecánicos, hidráulicos, neumáticos, etc., involucran relaciones no lineales entre las variables. Por ejemplo, la salida de un componente puede saturarse para niveles elevados de señal de entrada. Puede haber una franja o zona muerta que afecta a las señales pequeñas (la franja muerta de un componente es un pequeño rango de variaciones de entrada a las cuales el componente no es sensible). En algunos componentes puede haber alinealidades cuadráticas, por ejemplo los amortiguadores utilizados en sistemas físicos pueden ser lineales para operaciones de baja velocidad, pero volverse no lineales en velocidades altas y la fuerza de amortiguación volverse proporcional al cuadrado de la velocidad de funcionamiento.

La característica más importante de los sistemas no lineales es que no es aplicable el principio de superposición. En general, los procedimientos para hallar soluciones de problemas que involucran sistemas no lineales, son extremadamente complicados por métodos analíticos debiendo utilizar métodos numéricos, simulación analógica o introducir sistemas lineales "equivalentes"; para obtener la solución. Los sistemas lineales equivalentes son válidos solamente en un restringido rango de operación. Una vez que se ha aproximado un sistema no lineal con un modelo matemático lineal, se puede utilizar un cierto número de herramientas lineales para su análisis y diseño.

## 1.2 METODOLOGIA DE LA SIMULACION DIGITAL DE PROCESOS

Partiendo de los requerimientos generales de un sistema de control y de los problemas básicos que se presentan en el proyecto de sistemas de control, desarrollaremos una metodología para la simulación digital de procesos. Es decir, simulación utilizando métodos numéricos y un computador digital para obtener la solución de las ecuaciones que describen un sistema. De ahora en adelante, enfocaremos nuestro estudio hacia sistemas físicos sean estos dinámicos, continuos, discretos, determinísticos o estocásticos. En cualquiera de los casos los sistemas considerados pueden ser lineales o no lineales.

### 1.2.1 Requerimientos generales de un sistema de control.

Un proceso al que se le incorpora un sistema de control y se lo automatiza, debe ser en términos técnicos estable. Este es un requerimiento primario. Además de la estabilidad absoluta, un sistema de control debe tener una estabilidad relativa razonable; es decir, la velocidad de respuesta debe ser razonablemente rápida presentar un razonable amortiguamiento. Un sistema de control debe también poder reducir a cero, o a un valor tolerablemente pequeño, los errores. Cualquier sistema de control útil debe satisfacer estos requerimientos.

El requisito de estabilidad relativa razonable y el de una exactitud en régimen permanente tienden a ser incompatibles. Al proyectar un sistema de control, es, por tanto, necesario llegar a un compromiso eficaz entre ambos requisitos.

### 1.2.2 Problemas básicos en el proyecto de sistemas de control.

La Fig. 1.5 representa un diagrama de bloques de un sistema de control. El controlador produce señales de control basadas en las variables de referencia de entrada y las de salida. En la práctica también hay siempre algunas perturbaciones que actúan en la planta. Estas pueden ser de origen

interno o externo, y pueden ser causales o previsibles. El controlador debe tener en cuenta cualquier perturbación que afecte las variables de salida.

Para determinar la señal de control óptima, es necesario definir un índice de funcionamiento. Este índice es una medida cuantitativa del funcionamiento, midiendo la desviación respecto a un funcionamiento ideal. La especificación de la señal en el intervalo de tiempo de operación es denominada *ley de control*. Matemáticamente el problema básico de control es determinar la ley de control adecuada, sujeta a diversas restricciones de ingeniería y económicas, que minimice un índice de funcionamiento dado.

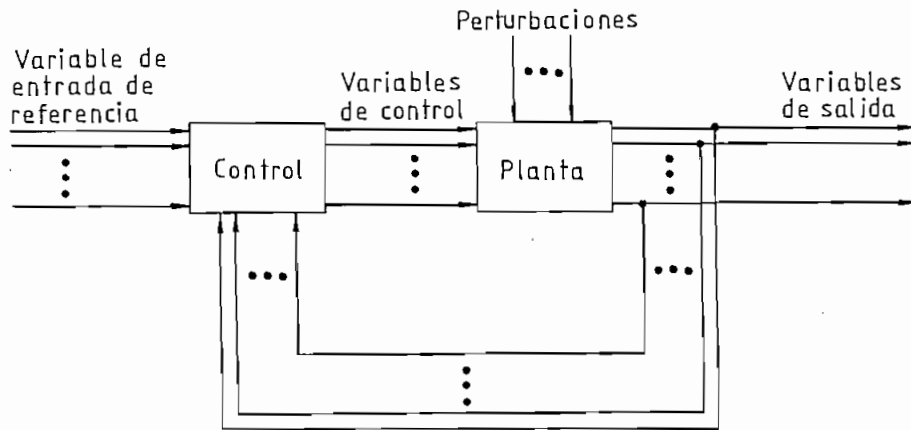


Fig. 1.5 Diagrama de bloques de un sistema de control.

Para sistemas relativamente simples se puede obtener la ley de control en forma analítica. Para sistemas complejos puede ser necesario generar la ley de control con una computadora digital trabajando en línea.

**Análisis.** Se entiende por análisis de un sistema de control, la investigación, bajo condiciones especificadas, del funcionamiento del sistema cuyo modelo matemático se conoce.

Como cualquier sistema consta de componentes, el análisis debe comenzar por una descripción matemática de cada componente. Una vez deducido el modelo matemático del sistema completo, el modo en que se realiza el análisis es independiente de si el sistema físicamente es neumático, eléctrico, mecánico, etc.

**Proyecto.** Proyectar un sistema significa hallar uno que

cumpla la tarea pedida. En general, el procedimiento de diseño o proyecto no es directo, sino que requiere de algunos tanteos.

**Síntesis.** Por síntesis se entiende encontrar por un procedimiento directo, un sistema que funcione de un modo especificado. Generalmente ese procedimiento es totalmente matemático desde el principio al final del proceso de diseño. Actualmente se dispone de procedimientos de síntesis para redes lineales y sistemas óptimos lineales.

**Enfoque básico del proyecto de sistemas de control.** El enfoque básico de cualquier sistema de control práctico, necesariamente involucra procedimientos de tanteo. Teóricamente es posible la síntesis de sistemas lineales de control, y el ingeniero de control puede determinar sistemáticamente los componentes necesarios para lograr el objetivo propuesto. En la práctica, sin embargo, el sistema puede quedar sujeto a muchas restricciones, o no ser lineal, y para estos casos hasta ahora no hay métodos de síntesis. Además, pueden no conocerse con precisión las características de los componentes. Así, siempre se hacen necesarios los procedimientos de tanteo.

Frecuentemente se encuentran en la práctica situaciones en que la planta está dada y el ingeniero de control ha de proyectar el resto del sistema de modo que el conjunto cumpla con las especificaciones dadas de cumplimiento de tareas. Las especificaciones han de ser interpretadas en términos matemáticos.

En muchos casos el proyecto de un sistema de control se cumple del siguiente modo: el ingeniero comienza el proceso de proyecto conociendo las especificaciones o el índice de funcionamiento, la dinámica de la planta dada y la dinámica de los componentes; esto último abarca los parámetros de diseño. El ingeniero de control aplica técnicas de control, si las hay, junto con otras técnicas, para establecer un modelo matemático del sistema.

Una vez formulado el problema de diseño en términos de este modelo, realiza un proyecto matemático que brinda la solución a la versión matemática del problema de diseño. En esta etapa es importante la simulación del modelo matemático en una computadora.



Tras completar el diseño matemático, el ingeniero de control simula el modelo en una computadora para verificar el comportamiento del sistema resultante en respuesta a varias señales y perturbaciones. Habitualmente, la configuración inicial del sistema no es satisfactoria, entonces hay que rediseñar el sistema y completar el correspondiente análisis. Nótese que en esta etapa es muy útil la teoría de control.

Como último paso se puede construir un prototipo del sistema físico. Se hace notar que este proceso de construcción del prototipo es el inverso del de modelado matemático. El prototipo es un sistema físico que representa con razonable error el modelo matemático. Una vez construido el prototipo, el ingeniero lo prueba para ver si es satisfactorio, o si no lo es. Si lo es, el diseño está completo. Si no, hay que modificar y probar nuevamente el prototipo. El proceso continúa hasta que el prototipo resulta completamente satisfactorio.

Finalmente, teniendo como base el prototipo, el ingeniero de control procederá a la implementación física del sistema el cual definitivamente tendrá el índice de funcionamiento deseado.

### 1.2.3 Metodología para la simulación digital.

Tratándose de simulación digital de sistemas, introducimos una metodología similar a la descrita en el numeral anterior, con la única variante de que utilizamos métodos numéricos y un computador digital para obtener la solución de las ecuaciones diferenciales que componen el modelo matemático del sistema. En resumen, los pasos a seguirse para resolver un problema de análisis o proyecto de sistemas mediante la simulación digital serán:

1. definir el sistema y sus componentes;
2. formular el modelo matemático bajo las suposiciones necesarias;
3. desarrollar el programa digital de simulación y obtener las soluciones para las variables de salida deseadas;
4. examinar las soluciones y las suposiciones; y entonces

5. analizar de nuevo si el sistema no cumple con el índice de funcionamiento deseado o diseñar si es que cumple.

Vamos a desarrollar un ejemplo de simulación digital para ilustrar el método antes descrito.

Los sistemas de control para mantener constante la tensión de una cinta de acero en una laminadora caliente, se conocen como reguladores de tensión. En la Fig. 1.6 se muestra un sistema típico. El regulador de tensión es un brazo de 2-3 pies de largo con un rodillo en el extremo y sube para presionar contra la cinta por medio de un motor. La velocidad típica de la cinta al pasar por el regulador de tensión es de 2000 pies/min. Se compara un voltaje proporcional a la posición del regulador de tensión con un voltaje de referencia y se integra, suponiéndose que un cambio en la posición del regulador de tensión es proporcional al cambio en la tensión de la cinta de acero. El error así obtenido se lo filtra y la señal resultante comanda a un grupo amplidina generador el cual a su vez maneja al motor.

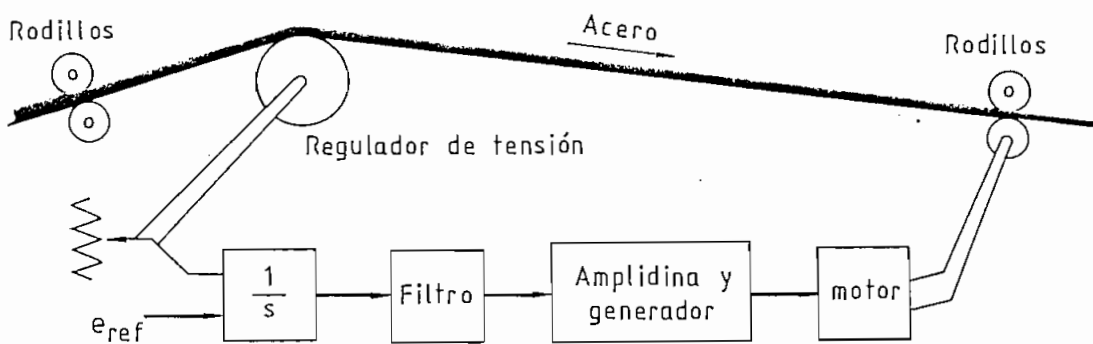


Fig 1.6 Sistema regulador de tensión de cinta de acero.

Se requiere calibrar la ganancia  $K_a$  de la amplidina para obtener una respuesta sobreamortiguada con un tiempo de establecimiento menor a 20 segundos, ante la presencia de un función escalón unitaria en la entrada.

El sistema y sus componentes. El sistema es dinámico y lineal. Debemos hacer un estudio de los diversos componentes que lo forman con el fin de determinar las ecuaciones que los describen. Este estudio se lo realizará en forma práctica, es decir, experimentando sobre el filtro, la amplidina, el generador y el motor para estimar los parámetros que completen la estructura matemática correspondiente a cada uno de estos componentes.

Asumimos que este trabajo de estimación o medida de los parámetros correspondientes a los componentes antes mencionados se ha realizado obteniéndose los siguientes modelos matemáticos descritos en forma de función de transferencia:

filtro;  $T_f(s) = (2s+1)/(0.1s+1)$  (1.1)

amplidina;  $T_a(s) = K_a/(s+1)$  (1.2)

generador;  $T_g(s) = 1/(s+1)$  (1.3)

motor;  $T_m(s) = 0.25/(s^2+s+0.25)$  (1.4)

El modelo matemático. Una vez examinado el sistema físico, el siguiente paso es obtener un modelo matemático adecuado para la simulación. Basados en las ecuaciones 1.1 a 1.4 podemos obtener el modelo del sistema en forma de diagrama de bloques tal como se muestra en la Fig. 1.7.

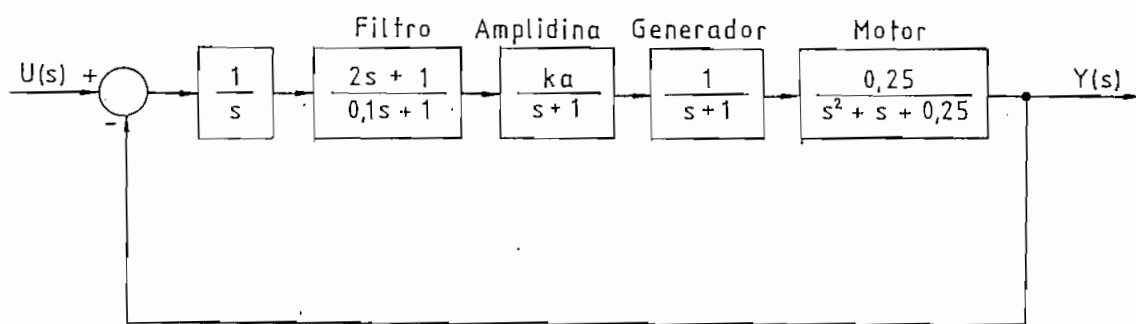


Fig. 1.7. Diagrama de bloques del regulador de tensión.

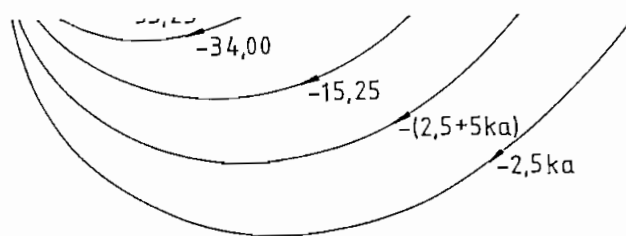


Fig. 1.8. Gráfica del flujo de señal de la ec. 1.5.

En este gráfico definimos las nuevas variables de estado  $X_1$  a  $X_6$  que nos ayudan a formular el modelo matemático definitivo como sigue

$$dx_1/dt = u - 13x_1 - 33.25x_2 - 34x_3 - 15.25x_4 - (2.5 + 5K_a)x_5 - 2.5K_a x_6 \quad (1.6)$$

$$\begin{aligned} dx_2/dt &= x_1 \\ dx_3/dt &= x_2 \\ dx_4/dt &= x_3 \\ dx_5/dt &= x_4 \\ dx_6/dt &= x_5 \end{aligned} \tag{1.6}$$

$$y = 5K_1 x_5 + 2.5K_1 x_6 \tag{1.7}$$

las ecuaciones 1.6 son llamadas ecuaciones de estado y la ecuación 1.7 es llamada ecuación de salida.

Notemos que el modelo matemático de un sistema puede tomar diferentes formas dependiendo basicamente del método que se vaya a emplear para realizar el análisis. Siempre que se utilice la simulación digital se debe obtener el modelo matemático del sistema en forma de ecuaciones de estado.

Programa digital de simulación y soluciones. Ahora estamos listos para obtener las soluciones de las ecuaciones 1.6 y 1.7 con la ayuda de un computador digital. En este punto, lo primero que debemos hacer es escoger el algoritmo para resolver las ecuaciones. Existen muchos algoritmos para resolver ecuaciones diferenciales, algoritmos como el de Euler, de Runge-Kutta, fórmulas multipaso, predictor-corrector, de Adams-Moulton, etc. (5); cada uno de estos algoritmos ofrece ventajas y desventajas frente a los otros, factores que deberán ser evaluados para escoger el algoritmo adecuado. No vamos a explicar detalladamente cuales son las características de cada uno de los posibles algoritmos que podríamos emplear, mas bien, tan solo con la finalidad de ilustrar la metodología de la simulación digital de procesos, escogeremos en este paso el algoritmo de Euler el cual permite obtener una rápida solución aunque con un error de paso apreciable, error que puede ser reducido si se toman intervalos de tiempo suficientemente pequeños como paso para la integración. Más adelante veremos que uno de los algoritmos de fácil aplicación, que presenta un error de paso relativamente pequeño y que más se ajusta a nuestras necesidades es el de Runge-Kutta de cuarto orden.

El algoritmo de Euler reza lo siguiente

Para hallar una solución aproximada de la ecuación diferencial

$$dy/dx = f(x,y)$$

$$y(a) = y_0$$

sobre un intervalo (a,b):

1. escoger un paso h adecuado

$$x_0 = a \quad x_k = a+kh \quad y(x_k) = y(a+kh)$$

2. generar aproximaciones  $y_k$  a  $y(x_k)$  a partir de la fórmula recurrente

$$y_{k+1} = y_k + hf(x_k, y_k)$$

el error de paso estimado para este método es del orden de  $h^2$ .

El listado del programa basic empleado para la simulación de este problema utilizando el método de Euler se muestra a continuación.

```
100 CLS
110 INPUT "Ka";KA
120 INPUT "t0";T0
130 INPUT "tf";TF
140 INPUT "h";H
150 INPUT "Intervalo entre impresos";DTI
160 OPEN "DATOS.PRN" FOR OUTPUT AS #1
170 NP=(TF-T0)/DTI:NPI=INT((TF-T0)/DTI)
180 X1=0:X2=0:X3=0:X4=0:X5=0:X6=0
190 FOR I=0 TO NP
200 T=I*H
210 Y=5*KA*X5+2.5*KA*X6
220 IF INT(I/NPI)*NPI=I THEN PRINT #1,T;Y
230 X1=X1+DT*(1-(13*X1+33.25*X2+34*X3+15.25*X4+(2.5+5*KA)*
X5+2.5*KA*X6))
240 X2=X2+H*X1
250 X3=X3+H*X2
260 X4=X4+H*X3
270 X5=X5+H*X4
280 X6=X6+H*X5
290 NEXT I
300 END
```

El programa permite definir los valores de  $K_a$ , h, y el intervalo de tiempo de interés. Se prueba con una entrada escalón unitaria. Los resultados son almacenados en ASCII en un archivo llamado DATOS.PRN para luego poder visualizarlos y analizarlos con la ayuda de la hoja electrónica LOTUS 123.

Análisis de las soluciones. Corriendo varias veces el

programa descrito anteriormente para diferentes valores de  $K_a$  y examinando las soluciones en cada caso se puede observar que el sistema cumple con las especificaciones deseadas para un valor de  $K_a$  de 0.1. Los valores de salida y el gráfico de estos en función del tiempo se muestran en la tabla 1.1 y en la Fig. 1.9 respectivamente.

Notemos que este último paso involucra un procedimiento de tanteo; así, damos un valor inicial a  $K_a$  corremos el programa para este valor, obtenemos la solución y observamos si esta cumple con el índice de funcionamiento deseado. En el caso de no cumplir, aplicamos en forma cualitativa algunos conceptos de teoría de control para ver si incrementamos o disminuimos el valor de  $K_a$  inicial; entonces estimamos el nuevo valor de  $K_a$  y repetimos el proceso hasta encontrar el valor adecuado.

#### REFERENCIAS

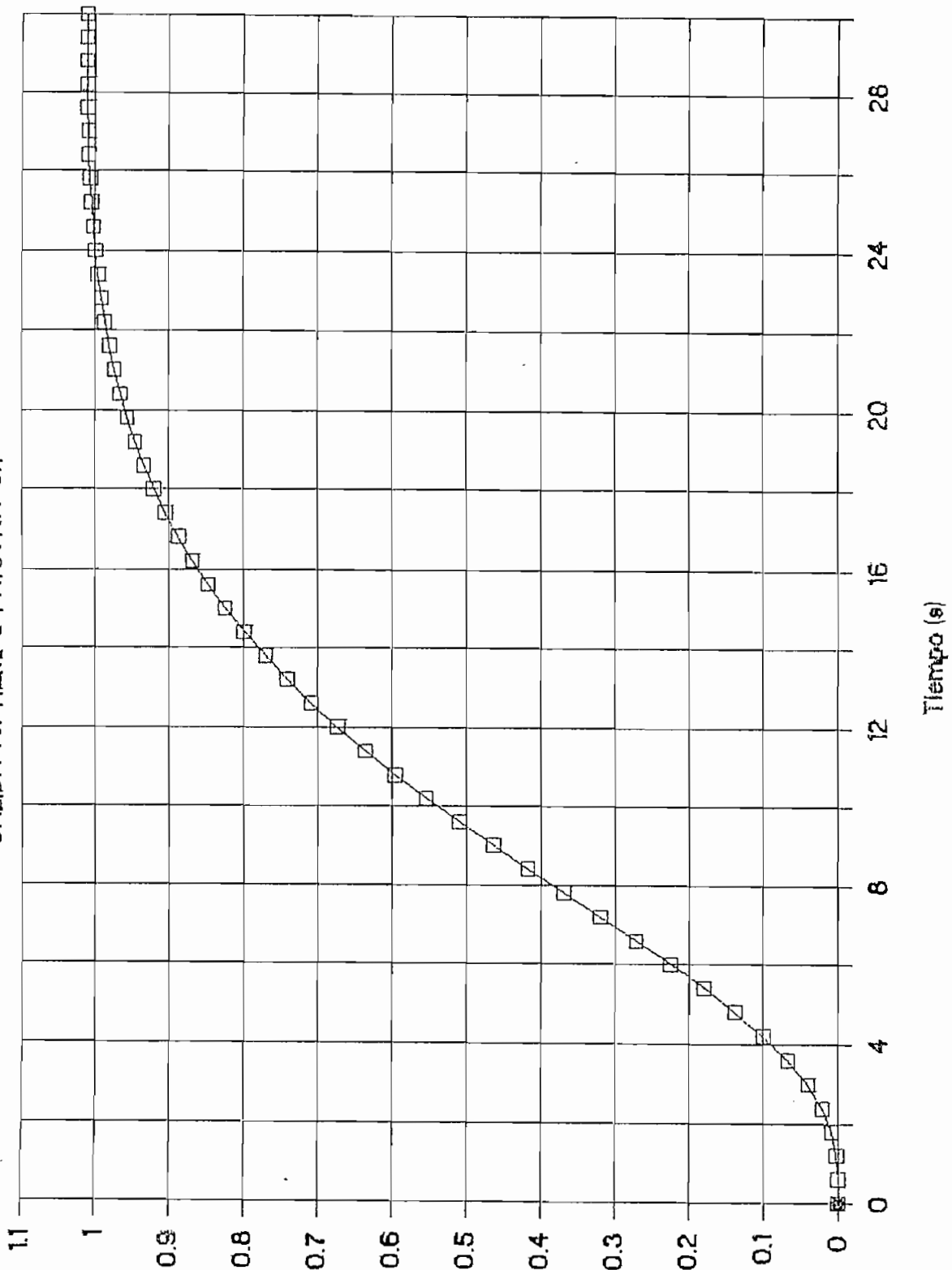
1. KATSUHIKO OGATA, *Ingeniería de Control Moderna*. Prentice-Hall International, 1974; pp. 9-10.
2. KATSUHIKO OGATA, *Ingeniería de Control Moderna*. Prentice-Hall International, 1974; pp. 12.
3. AVERILL M. LAW, y W. DAVID KELTON, *Simulation Modeling and Analysis*. McGraw-Hill, 1982; pp. 49.
4. RICHARD C. DORF, *Sistemas Automáticos de Control*. Fondo Educativo Interamericano S.A., 1978; pp. 50.
5. S. D. CONTE, y CARL de BOOR, *Análisis Numérico Elemental*. McGraw-Hill, 1972; pp. 339-396.

| Tiempo (s) | Salida   | Error % |
|------------|----------|---------|
| 0.0        | 0.000000 | 100.00% |
| 0.6        | 0.000119 | 99.99%  |
| 1.2        | 0.001853 | 99.81%  |
| 1.8        | 0.007888 | 99.21%  |
| 2.4        | 0.020235 | 97.98%  |
| 3.0        | 0.039735 | 96.03%  |
| 3.6        | 0.066271 | 93.37%  |
| 4.2        | 0.099121 | 90.09%  |
| 4.8        | 0.137251 | 86.27%  |
| 5.4        | 0.179520 | 82.05%  |
| 6.0        | 0.224803 | 77.52%  |
| 6.6        | 0.272061 | 72.79%  |
| 7.2        | 0.320372 | 67.96%  |
| 7.8        | 0.368943 | 63.11%  |
| 8.4        | 0.417105 | 58.29%  |
| 9.0        | 0.464308 | 53.57%  |
| 9.6        | 0.510110 | 48.99%  |
| 10.2       | 0.554162 | 44.58%  |
| 10.8       | 0.596203 | 40.38%  |
| 11.4       | 0.636040 | 36.40%  |
| 12.0       | 0.673546 | 32.65%  |
| 12.6       | 0.708646 | 29.14%  |
| 13.2       | 0.741310 | 25.87%  |
| 13.8       | 0.771544 | 22.85%  |
| 14.4       | 0.799385 | 20.06%  |
| 15.0       | 0.824896 | 17.51%  |
| 15.6       | 0.848158 | 15.18%  |
| 16.2       | 0.869268 | 13.07%  |
| 16.8       | 0.888331 | 11.17%  |
| 17.4       | 0.905463 | 9.45%   |
| 18.0       | 0.920785 | 7.92%   |
| 18.6       | 0.934417 | 6.56%   |
| 19.2       | 0.946482 | 5.35%   |
| 19.8       | 0.957101 | 4.29%   |
| 20.4       | 0.966393 | 3.36%   |
| 21.0       | 0.974472 | 2.55%   |
| 21.6       | 0.981449 | 1.86%   |
| 22.2       | 0.987427 | 1.26%   |
| 22.8       | 0.992508 | 0.75%   |
| 23.4       | 0.996783 | 0.32%   |
| 24.0       | 1.000340 | -0.03%  |
| 24.6       | 1.003261 | -0.33%  |
| 25.2       | 1.005620 | -0.56%  |
| 25.8       | 1.007487 | -0.75%  |
| 26.4       | 1.008924 | -0.89%  |
| 27.0       | 1.009990 | -1.00%  |
| 27.6       | 1.010736 | -1.07%  |
| 28.2       | 1.011209 | -1.12%  |
| 28.8       | 1.011452 | -1.15%  |
| 29.4       | 1.011502 | -1.15%  |
| 30.0       | 1.011393 | -1.14%  |

Tabla 1.1. Valores de salida del regulador de tensión para  $K_v=0.1$

# Fig. 1.9 REGULADOR DE TENSION

SALIDA vs. TIEMPO PARA KA=0.1



Y(t)



CAPITULO II

## CAPITULO II

### MODELOS DE SIMULACION Y BASE MATEMATICA

En este capítulo analizaremos el modelo matemático de un sistema con el fin de obtener un modelo adecuado de simulación; así, trataremos al modelo matemático como un conjunto de "bloques matemáticos", cada uno de ellos cumpliendo una determinada función. Una vez considerado de esta forma al sistema, analizaremos cuales son las funciones comunes asociadas al mismo; veremos cual es la forma de representar estas funciones en un diagrama de simulación analógica y su implementación en un computador analógico. Por último, analizaremos algunas formas de tratar al modelo de simulación analógica en forma discreta, es decir, considerando pequeños intervalos de tiempo y realizando las operaciones matemáticas pertinentes dentro de los mismos; esto último nos lleva al desarrollo de un algoritmo de simulación digital el cual lo basaremos en el método de Runge-Kutta de cuarto orden para la solución de ecuaciones diferenciales.

#### 2.1 FUNCIONES ASOCIADAS A UN SISTEMA

Dentro del método descrito en el Capítulo I para la simulación de sistemas los dos primeros pasos están dedicados a obtener la representación del sistema en forma matemática, es decir, implícita o explícitamente en forma de una o un conjunto de ecuaciones diferenciales para el caso de sistemas dinámicos que son de interés en nuestro estudio.

Partiremos de que se dispone del conjunto de ecuaciones que representan al sistema. Cualquiera que sea la forma de estas ecuaciones, es posible obtener un modelo general para cualquier sistema que esté representado por un conjunto de ecuaciones diferenciales de primer orden, a esta forma de representar un sistema se la llama *formulación con variables de*

estado. La forma general que toma el modelo de un sistema descrito mediante una formulación de variables de estado es:

$$dX/dt = AX+BU \quad (2.1)$$

$$Y = CX+DU \quad (2.2)$$

donde el vector  $X$  que contiene a las variables de estado  $x_1(t)$ ,  $x_2(t), \dots, x_n(t)$  describe totalmente el comportamiento del sistema dinámico. El vector  $U$  contiene a las funciones de entrada al sistema. El vector  $Y$  contiene a las salidas del sistema. Las matrices  $A$ ,  $B$ ,  $C$  y  $D$  son las matrices de coeficientes del sistema, estas matrices contendrán coeficientes constantes si el sistema es lineal invariante en el tiempo, los coeficientes serán funciones lineales del tiempo si es que se trata de un sistema lineal variable en el tiempo y contendrán funciones no lineales para sistemas no lineales. La ecuación 2.1 es llamada ecuación de estado mientras que la ecuación 2.2 es llamada ecuación de salida.

El método para modelar un sistema mediante variables de estado es sencillo si se parte de un diagrama de bloques o una función de transferencia y por medio de descomposición directa (utilizada en el ejemplo del capítulo anterior), descomposición en cascada o descomposición en paralelo (1) obtenemos un diagrama de flujo de señal. En este diagrama asignamos a cada nodo precedido de un factor  $s^{-1}$  una variable de estado, luego observando las relaciones entre las variables de estado, las entradas y las salidas se puede escribir fácilmente las ecuaciones de estado y salida.

Nuestro problema se centra ahora en la solución de las ecuaciones 2.1 y 2.2. Este problema lo podemos afrontar asumiendo que disponemos de un conjunto de "bloques o funciones matemáticas", las cuales las analizaremos a continuación.

### 2.1.1 Funciones lineales.

Sin importarnos cuales son las características de las

ecuaciones de estado que representan al sistema, podemos observar que implícita o explícitamente llevan involucradas al menos tres funciones operaciones como son: la integración de una variable, la suma de variables y la multiplicación de una variable por una constante. Como ejemplo consideremos un sencillo sistema de primer orden cuya función de transferencia está dada por

$$T(s) = Y(s)/U(s) = 1/(s+p) \quad (2.3)$$

la ecuación diferencial correspondiente será

$$dy/dt + py(t) = u(t) \quad (2.4)$$

que puede ser escrita como

$$\begin{aligned} e(t) &= u(t) + (-py(t)) \\ y(t) &= \int_{t_0}^t e(t) dt + y(t_0) \end{aligned} \quad (2.5)$$

observando la forma de las ecuaciones 2.5 podemos darnos cuenta que bastan tres operaciones para evaluar el valor de la salida  $y(t)$  en cualquier instante de tiempo si es que conocemos la entrada  $x(t)$  y la condición inicial  $y(t_0)$ . Así, para evaluar  $e(t)$  debemos multiplicar a  $y(t)$  por la constante  $-p$  y sumar el resultado a  $u(t)$ ; para evaluar  $y(t)$  deberemos integrar en forma continua la variable  $e(t)$ . Notemos que el procedimiento debe ser realizado en forma paralela, es decir, evaluando conjuntamente la variable  $e(t)$  y la salida  $y(t)$ .

Supongamos ahora que disponemos de tres funciones matemáticas que realizan las operaciones antes mencionadas podríamos entonces resolver cualquier problema de simulación de sistemas lineales invariantes en el tiempo tan solo con la obtención de un modelo matemático adecuado y la correcta utilización de estas funciones. En general si se trata de sistemas más complejos las funciones de integración, suma y multiplicación por una constante serán necesarias para obtener las respuestas de los mismos. Así, diremos que las funciones

descritas se encuentran asociadas a un sistema dinámico.

En la Fig. 2.1 introducimos una representación gráfica de estas operaciones, representación que posteriormente nos ayudará a obtener un diagrama de simulación para un sistema dado.

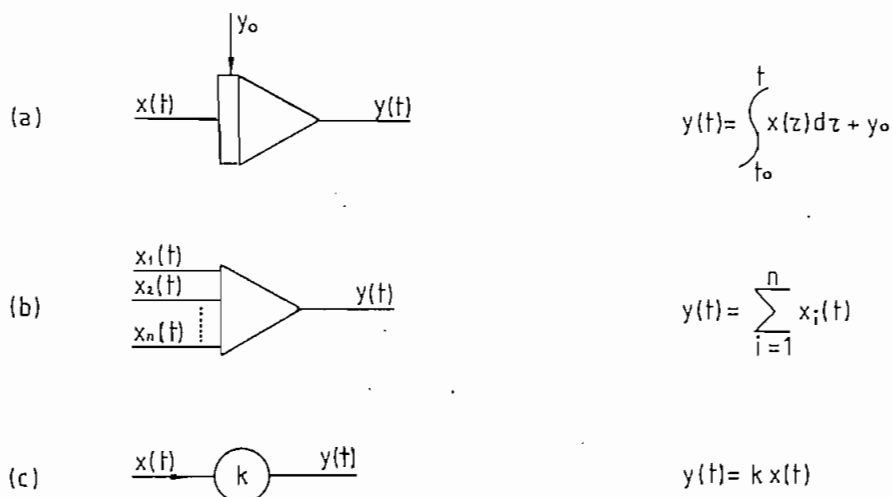


Fig. 2.1. Representaciones gráficas: a) Integrador, b) Sumador y c) multiplicación por una constante.

Por último, cabe anotar que las funciones descritas son lineales ya que cumplen con el principio de superposición.

### 2.1.2 Funciones no lineales.

Es un hecho conocido que muchas relaciones entre magnitudes físicas no son lineales aunque frecuentemente se aproximan por medio de ecuaciones lineales por simplificación matemática. Esta simplificación puede ser satisfactoria mientras las soluciones resultantes estén de acuerdo con los resultados experimentales. Una de las características más importantes de los sistemas no lineales es la dependencia en el comportamiento de respuesta del sistema, de la magnitud y tipo de entrada. Por ejemplo, un sistema no lineal puede comportarse en forma completamente distinta en respuesta a entradas escalón de diferentes valores.

En los sistemas de control reales se encuentran muchos tipos de alinealidades, y se las puede dividir en dos clases según sean inherentes al sistema o deliberadamente insertadas en el mismo.

Las alinealidades inherentes son inevitables en los sistemas de control. Algunos ejemplos de tales alinealidades son:

1. saturación
2. zona muerta
3. histéresis
4. juego
5. fricción estática, fricción de Coulomb y otras fricciones no lineales
6. resorte no lineal
7. comprensibilidad de fluido.

En términos generales, la presencia de estas alinealidades en el sistema de control, afecta adversamente el comportamiento del mismo. Por ejemplo, el juego puede producir inestabilidad en el sistema y la zona muerta a su vez puede producir error de régimen.

El otro tipo de alinealidades son las que se introducen intencionalmente en un sistema para mejorar el comportamiento del mismo o para simplificar su construcción, o por ambas causas. Un sistema no lineal adecuadamente diseñado para cumplir cierta función, frecuentemente es mejor desde un punto de vista económico, de espacio y de confiabilidad frente a los sistemas lineales diseñados para cumplir la misma tarea. El ejemplo más simple de un sistema no lineal intencional es un sistema encendido-apagado accionado por un relé. Se pueden encontrar otros ejemplos de sistemas de control óptimo que frecuentemente emplean controles alineales complicados. Debe notarse que aunque los elementos no lineales intencionales, pueden mejorar el comportamiento del sistema bajo ciertas condiciones de funcionamiento especificadas, en general degradan el comportamiento del sistema en otras condiciones de funcionamiento.

Matemáticamente, las matrices de las ecuaciones de estado que describen a un sistema dinámico no lineal involucrarán dentro de sus términos funciones no lineales que afectan a las variables de estado. La forma de afrontar estos problemas se basa en la utilización de las funciones lineales descritas en 2.1.1 y un conjunto de funciones no lineales comunmente asociadas a estos sistemas. Al no existir un método analítico general para el análisis de ecuaciones diferenciales no lineales, se procede a la utilización de computadoras digitales o analógicas para obtener la solución de estas ecuaciones.

A continuación describiremos algunas de las funciones no lineales que son comunes en los sistemas de control reales. Incluimos dentro de las representaciones gráficas de las diferentes alinealidades un diagrama de flujo que sirve de guía para el tratamiento o simulación de la alinealidad en un computador digital. Estos diagramas nos servirán cuando analicemos el tratamiento del modelo de simulación en un computador de este tipo.

Alinealidad limitador o de saturación. La curva característica típica entrada-salida para la alinealidad de saturación, corresponde a la mostrada en la Fig. 2.2. Para señales de entrada pequeñas la salida de un elemento de saturación es proporcional a la entrada. Para señales grandes de entrada (en valor absoluto), la salida permanece constante.

El fenómeno de saturación se presenta con mucha frecuencia en los sistemas físicos. La mayoría de sistemas de control de posición, velocidad, temperatura, etc. tienen componentes que pueden variar su comportamiento solo dentro de ciertos rangos; en sistemas eléctricos y electrónicos ejemplos de tales fenómenos son: la magnetización del hierro, la saturación de ciertos componentes electrónicos como transistores, amplificadores operacionales, etc. La alinealidad de saturación permite modelar estos componentes tanto en su región lineal como en sus límites.

Alinealidad comparador. También llamada de conexión-

desconexión o de dos posiciones, se la muestra en la Fig. 2.3. La salida de este elemento es, o bien una constante positiva o una constante negativa dependiendo si la entrada es mayor o menor que un valor referencial (XR).

La alinealidad comparador esta comunmente asociada a todos los sistemas de control de dos posiciones (on-off), ejemplos de tales sistemas se los encuentra principalmente en controles de temperatura. Esta alinealidad permite también modelar elementos eléctricos, electrónicos o mecánicos que dentro de un sistema se encuentran trabajando en régimen de conmutación.

Cabe anotar que en sistemas reales cuando la entrada a un elemento no lineal de estas características se encuentra dentro de valores muy próximos al valor de referencia la salida es por lo general impredecible, es decir, puede tomar cualquiera de los dos valores permisibles.

Alinealidad comparador con histéresis. La relación de transferencia que define este tipo de alinealidad se la muestra en la Fig. 2.4. En realidad, esta alinealidad corresponde a la combinación de dos comparadores que admiten una ventana dentro de la cual variaciones en la entrada no afectan a la salida.

La alinealidad comparador con histéresis puede tomar varias formas de acuerdo con el sentido de cambio y los valores permisibles en la salida, sin embargo, partiendo del modelo indicado en la Fig. 2.4 y con la ayuda de un adecuado escalamiento a la función de entrada (multiplicándola por una constante), se pueden obtener relaciones de transferencia diferentes a la indicada en la Fig. 2.4a. Como modelo para obtener la relación de transferencia de la Fig. 2.6a se consideró un circuito comparador con histéresis montado con un amplificador operacional tal como se muestra en la Fig. 2.5. De acuerdo con este circuito, en el instante de encendido  $t=t_0$  si la entrada se encuentra en un valor que cae dentro de la ventana, la salida puede tomar cualquiera de los dos valores permisibles.

La alinealidad comparador con histéresis es comunmente insertada en sistemas de control de dos posiciones con el fin de mejorar la estabilidad evitando las posibles oscilaciones de la



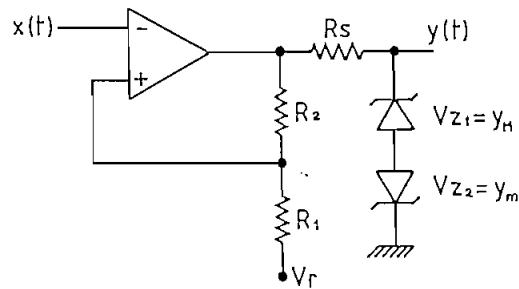


Fig. 2.5 Circuito comparador con histéresis.

salida alrededor del punto de trabajo. Se la encuentra también como característica inherente de muchos fenómenos físicos como la magnetización, variaciones de temperatura, etc.

Alinealidad rectificador de media onda. Las características de esta alinealidad se muestran en la Fig. 2.6. La salida es nula cuando la entrada es negativa y es igual a la entrada para valores positivos de esta.

Esta alinealidad permite modelar componentes físicos que admiten la circulación de fluidos en forma unidireccional tal es el caso de válvulas unidireccionales o diodos rectificadores cuando se trate de corriente eléctrica.

Alinealidad valor absoluto. Sus características se muestran en la Fig. 2.7. La salida de esta alinealidad es igual al valor absoluto de la entrada.

Esta alinealidad es característica de ciertos componentes o sistemas físicos que admiten solamente valores positivos o negativos de una variable.

Alinealidad zona muerta. A menudo se la denomina también alinealidad de umbral. En la Fig. 2.8 se muestran sus características. En un elemento con zona muerta no existe salida para entradas muy pequeñas.

Esta alinealidad permite modelar los pequeños errores de offset de elementos mecánicos, eléctricos, electrónicos, etc.

Alinealidad de juego. La relación de transferencia

entrada-salida de este tipo de alinealidad se presenta en la Fig. 2.9. Representa fundamentalmente el pequeño error de juego que pueden tener ciertos mecanismos al rededor de un punto de trabajo.

Este tipo de alinealidad se la encuentra básicamente en los sistemas de acoplamiento mecánico tales como engranajes, articulaciones, etc.

Igual que en el caso de los comparadores, se debe asumir un determinado valor de salida cuando el sistema es arrancado y la variable de entrada se encuentra en valores dentro de la zona donde existe juego.

Alinealidades cuadrador y raíz cuadrada. Mostradas en las Figs. 2.10 y 2.11, responden respectivamente al cuadrado y a la raíz cuadrada de la señal de entrada.

Encontramos la necesidad de esta alinealidad para modelar sistemas físicos no lineales cuyo modelo matemático incluye una o más funciones o variables elevadas al cuadrado o extraída su raíz. Ejemplos de estos modelos se los encuentra en sistemas de propagación de calor, fricción, oscilaciones mecánicas con resortes no lineales, etc.

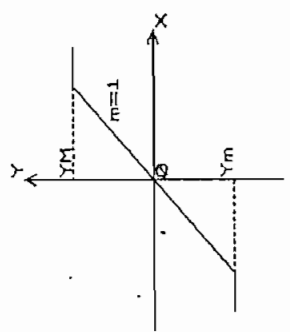
Alinealidades logarítmica y exponencial. Las características entrada-salida de estas alinealidades se muestran en las Figs. 2.12 y 2.13 respectivamente.

Es común encontrar este tipo de alinealidades en sistemas transductores empleados en la medición de magnitudes físicas como resistencia eléctrica, posición, etc. Se las emplea también para modelar con mayor exactitud la operación de juntas semiconductoras.

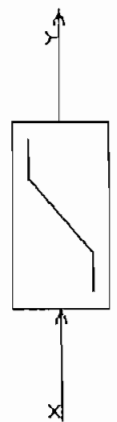
Alinealidades seno y coseno. Las funciones de transferencia entrada-salida de estas alinealidades se muestran en las Figs. 2.14 y 2.15 respectivamente.

Estas alinealidades son comunes en el modelado de sistemas convertidores eléctricos de corriente alterna a continua y en sistemas rotativos.

FIG 2.2 ALINEALIDAD LIMITADOR O DE SATURACION.  
DATOS:  $y_m$ ,  $y_M$ .



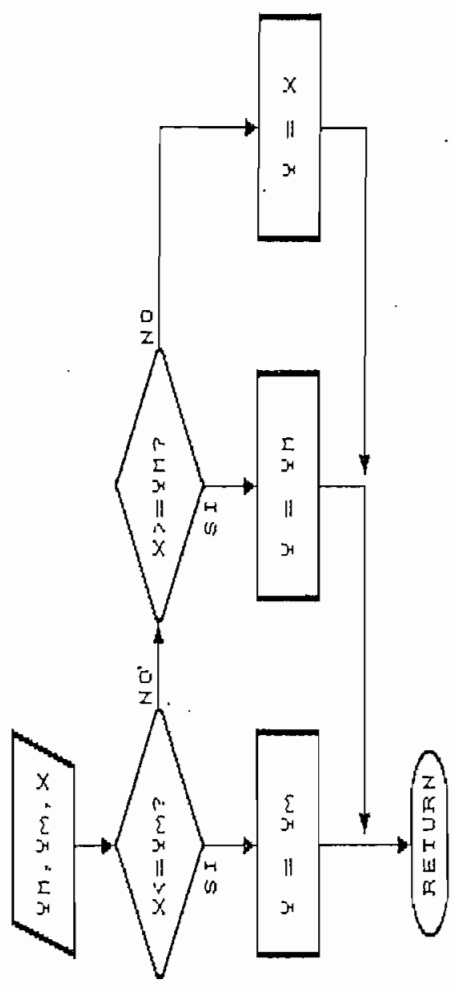
(a)



(b)

|           |      |                 |
|-----------|------|-----------------|
| $y = y_m$ | para | $x < y_m$       |
| $y = x$   | para | $y_m < x < y_M$ |
| $y = y_M$ | para | $x > y_M$       |

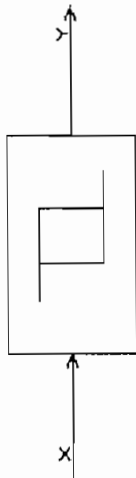
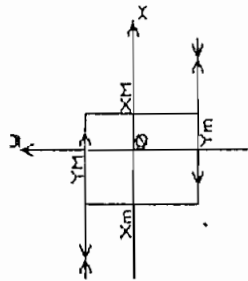
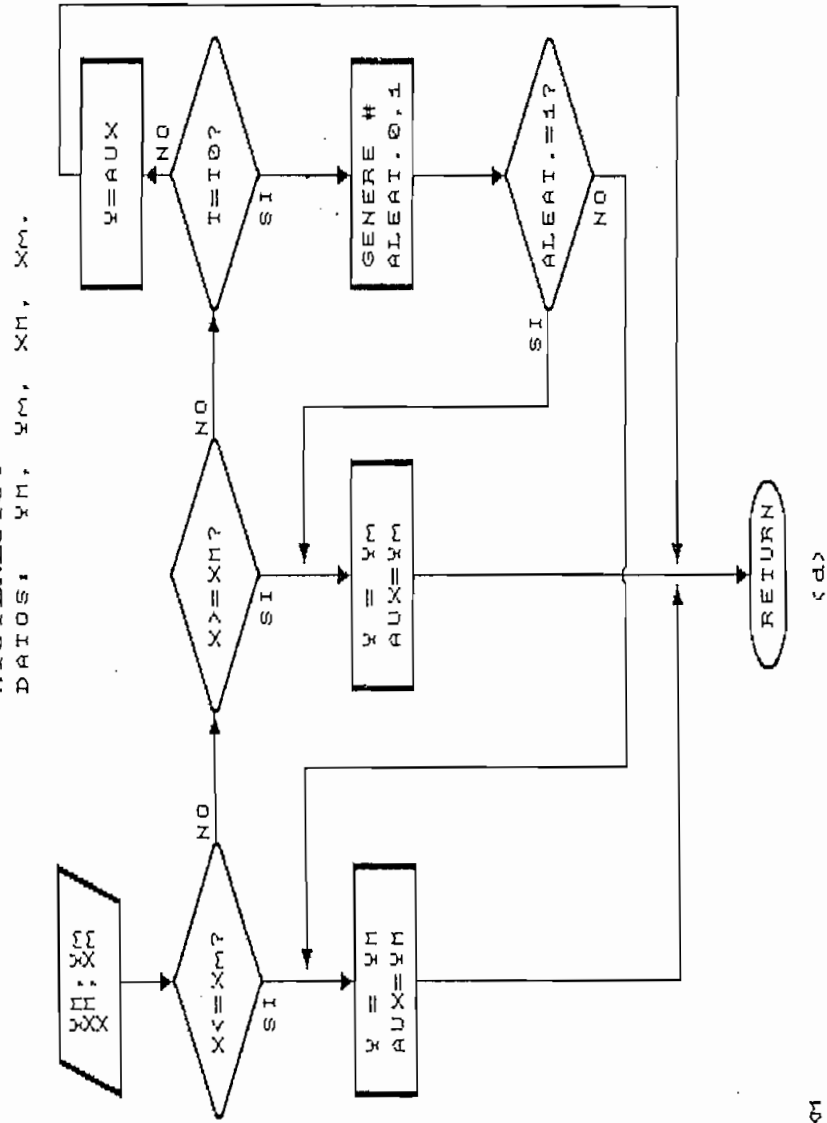
(c)



(d)

- a. FUNCION DE TRANSFERENCIA.
- b. REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c. ECUACIONES.
- d. DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2.4  
 ALINEALIDAD COMPARADOR CON  
 HISTERESIS.  
 DATOS: YM, YM, XM, XM.



CUANDO X CRECE DESDE X < XM

Y = YM para X < XM

Y = YM para X > XM

CUANDO X DRECRECE DESDE X > XM

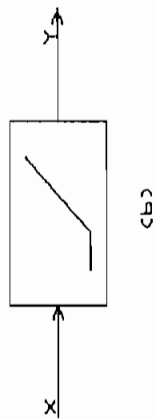
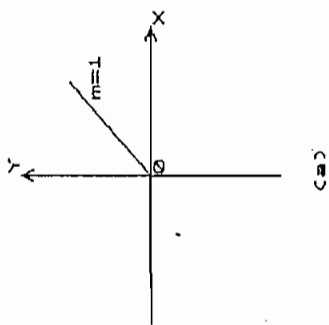
Y = Ym para X > XM

Y = YM para X < XM

(c)

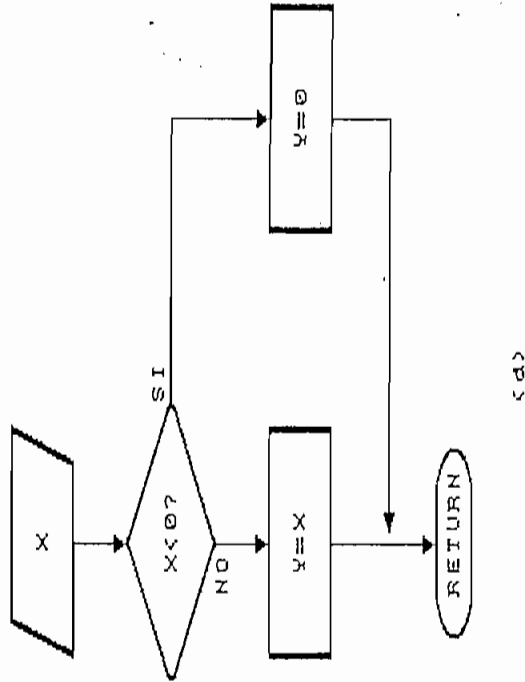
- a. - RELACION DE TRANSFERENCIA.
- b. - REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c. - ECUACIONES.
- d. - DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-6 ALINEALIDAD RECTIFICADOR DE MEDIA ONDA.



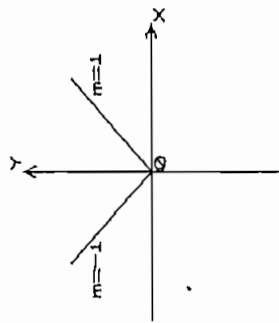
$Y=0$  para  $X < 0$   
 $Y=X$  para  $X > 0$

(c)

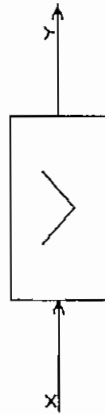


- a.- FUNCION DE TRANSFERENCIA.
- b.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c.- ECUACIONES.
- d.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2.7 ALINEALIDAD VALOR ABSOLUTO

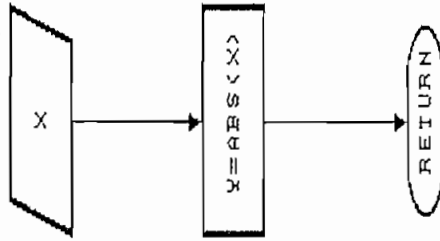


(a)



(b)

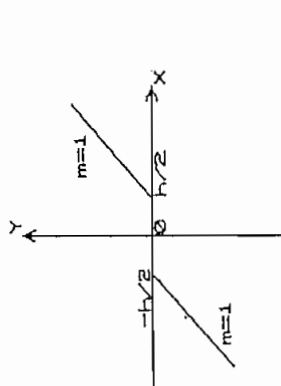
$Y = X$  Para  $X < 0$   
 $Y = -X$  Para  $X > 0$   
 (c)



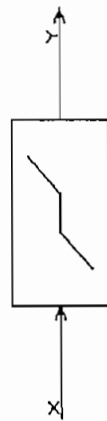
(d)

- a.- FUNCION DE TRANSFERENCIA.
- b.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c.- ECUACIONES.
- d.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2.8 ALINEALIDAD ZONA MUERIA  
DATOS: h.



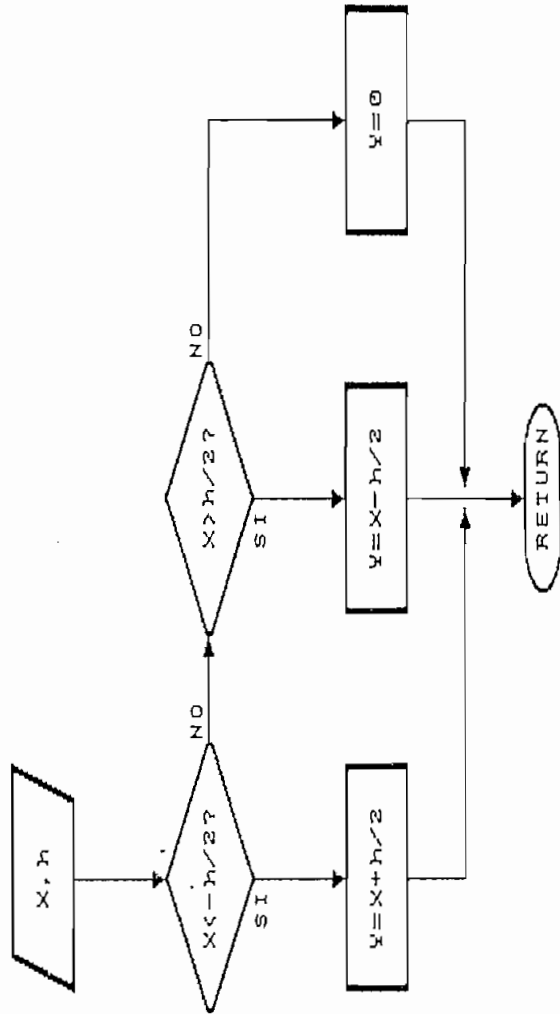
(a)



(b)

$Y = X + h/2$  para  $X < -h/2$   
 $Y = 0$  para  $-h/2 < X < h/2$   
 $Y = X - h/2$  para  $X > h/2$

(c)



(d)

- a.- FUNCION DE TRANSFERENCIA.
- b.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c.- ECUACIONES.
- d.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

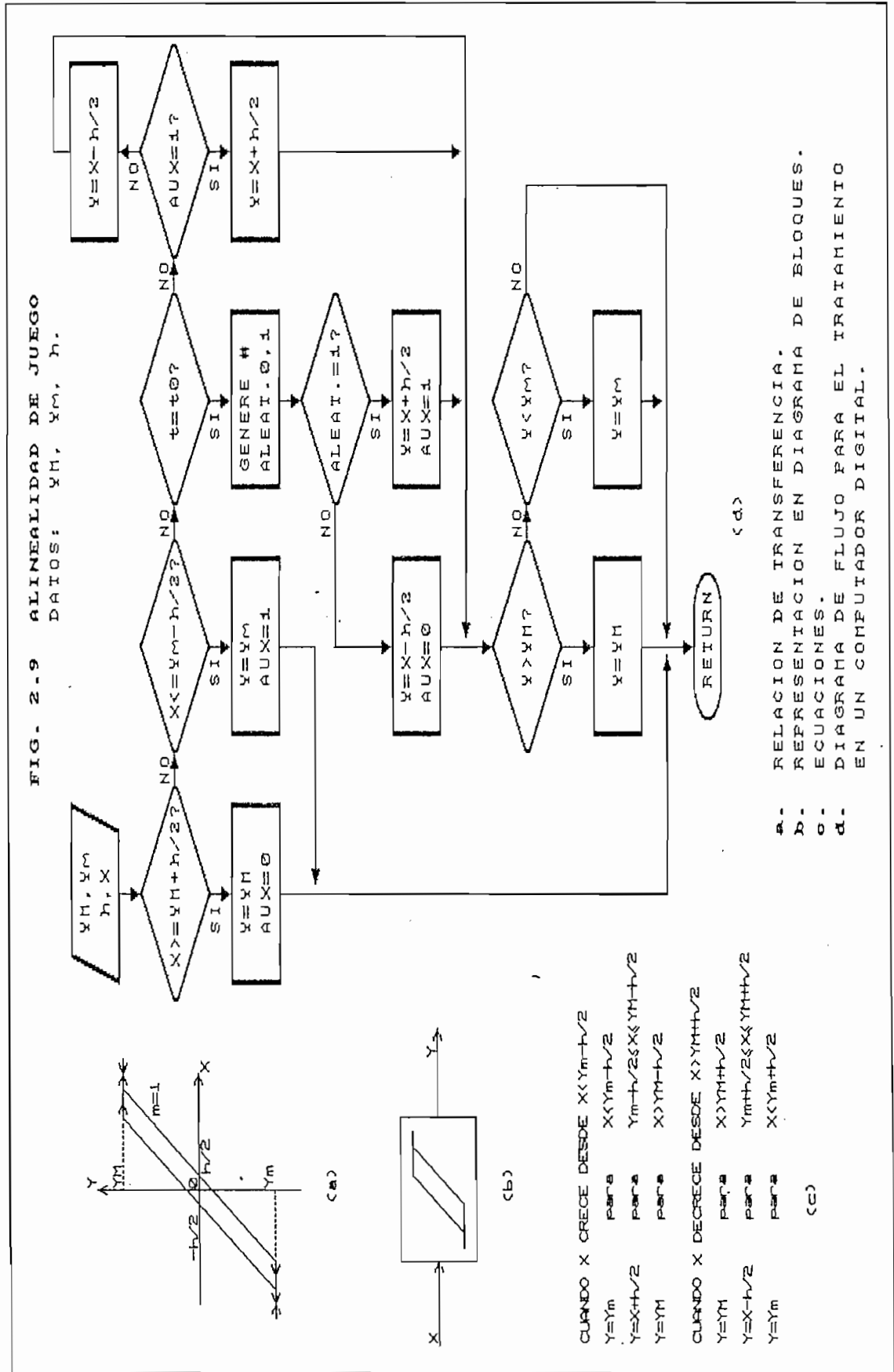
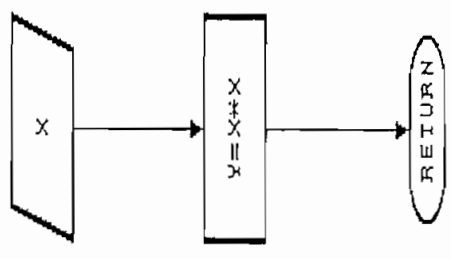
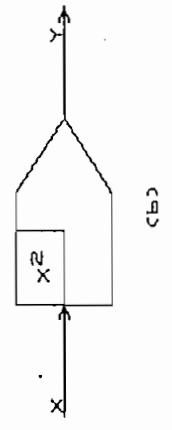
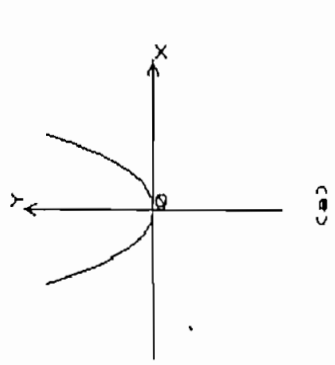




FIG 2-10 ALINEALIDAD CUADRADOR

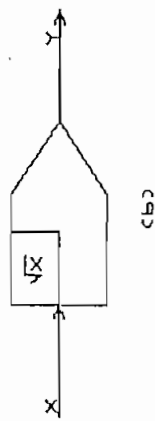
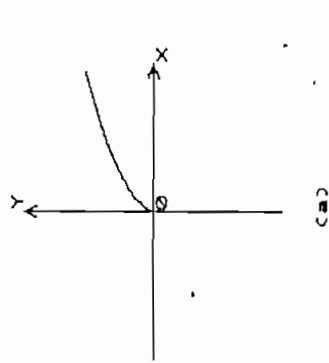


<d>

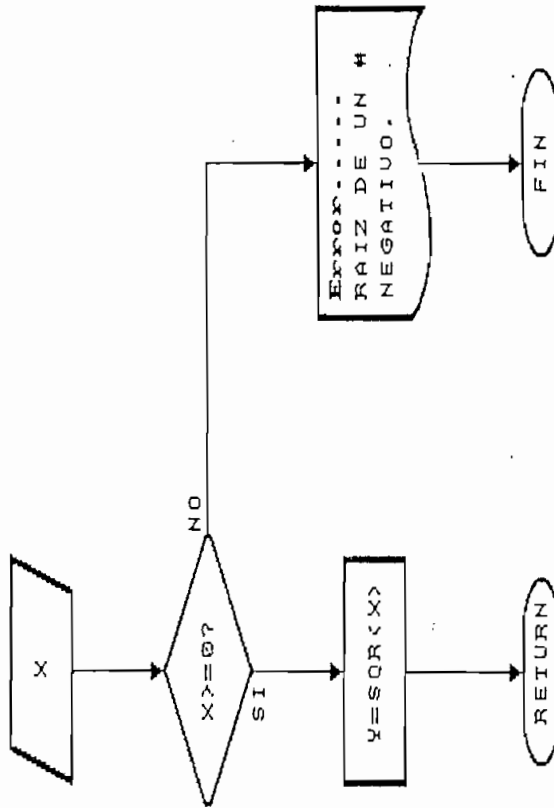
- a- FUNCION DE TRANSFERENCIA.
- b- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c- ECUACIONES.
- d- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

$Y = X^2$   
(c)

FIG 2-11 ALINEALIDAD RAIZ CUADRADA

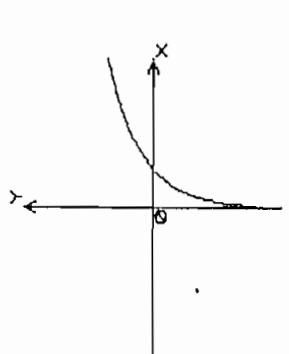


$Y = \sqrt{X}$  para  $X \geq 0$   
(c)

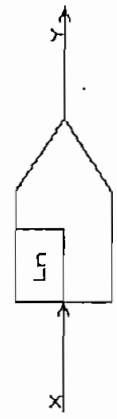


- a. FUNCION DE TRANSFERENCIA.
- b. REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c. ECUACIONES.
- d. DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-12 ALINEALIDAD LOGARITMO NATURAL



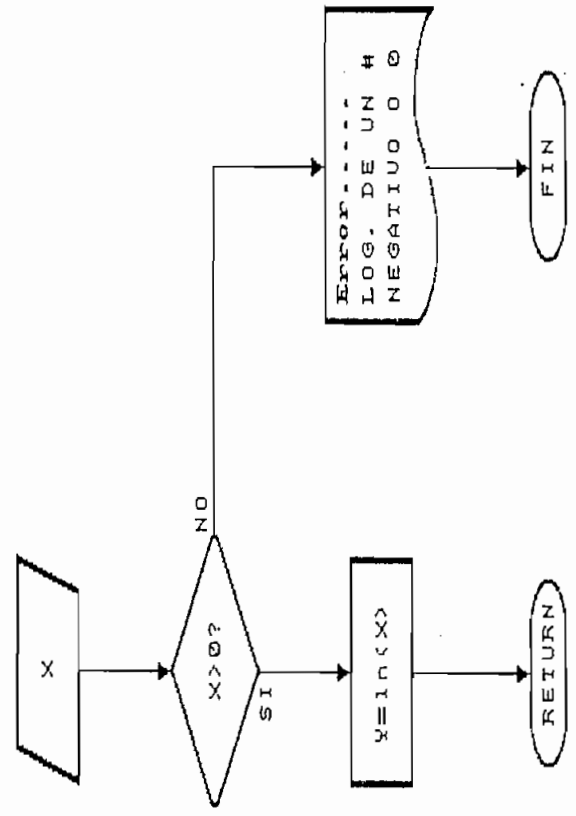
(a)



(b)

$Y = \ln(X)$  para  $X > 0$

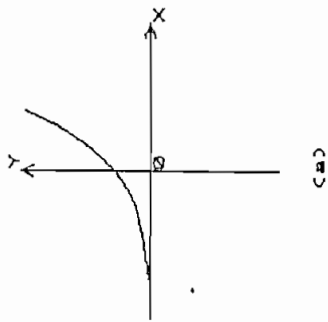
(c)



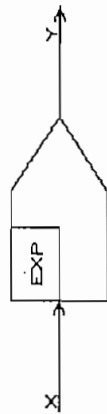
(d)

- a. FUNCION DE TRANSFERENCIA.
- b. REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c. ECUACIONES.
- d. DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-13 ALINEALIDAD EXPONENCIAL



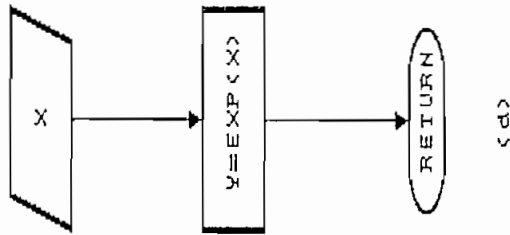
(a)



(b)

$$Y = e^X$$

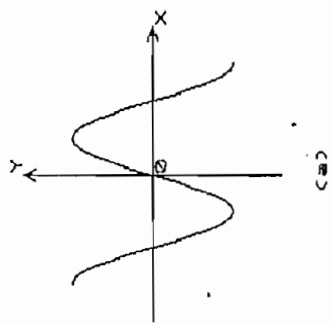
(c)



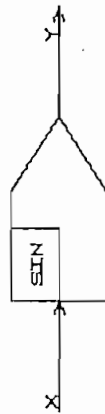
(d)

- a.- FUNCION DE TRANSFERENCIA.
- b.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c.- ECUACIONES.
- d.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-14 ALINEALIDAD SENO



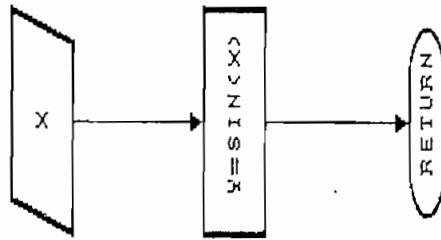
(a)



(b)

$$Y = \text{SEN}(X)$$

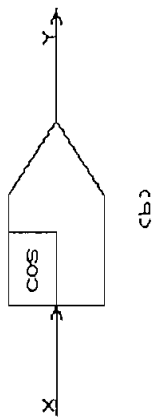
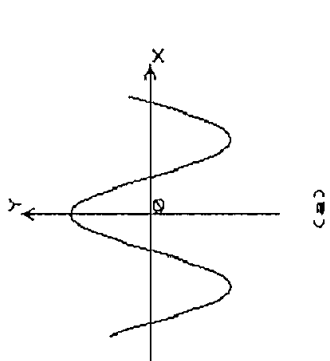
(c)



(d)

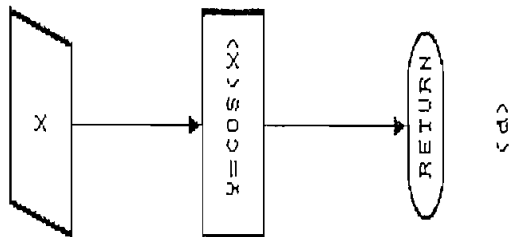
- a.- FUNCION DE TRANSFERENCIA.
- b.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- c.- ECUACIONES.
- d.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2.15 ALINEALIDAD COSENO



$Y = \cos(X)$

(c)



- \*: FUNCION DE TRANSFERENCIA.
- D: REPRESENTACION EN DIAGRAMA DE BLOQUES.
- G: ECUACIONES.
- d: DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

### 2.1.3 Otras funciones.

Existen otras funciones que están asociadas a los modelos matemáticos de los sistemas no lineales. Estas funciones adicionales permiten completar el grupo de "módulos matemáticos" necesarios para la simulación de cualquier sistema. A continuación describimos algunas de las funciones especiales comunes en el modelado de sistemas.

Multiplicación y división entre variables. En las Figs. 2.16a y 2.17a respectivamente se muestra la representación simbólica de estos módulos.

Es indudable que estas funciones son de mucha importancia en la simulación de modelos matemáticos no lineales cuyas ecuaciones involucran productos o cocientes entre variables. Muchos modelos correspondientes a sistemas físicos tales como sistemas de propagación de calor, sistemas con resortes no lineales, sistemas de propagación de ondas, etc. incluyen estas funciones.

Funciones lógicas En algunos sistemas se encuentran componentes lógicos los cuales, en su mayoría, se pueden simular en términos de las conectivas lógicas Y (AND) y NO (NOT), u O (OR) y NO. Para cubrir todas las posibilidades y tener cierta flexibilidad en el problema de simular estos componentes, podemos hacer uso de las compuertas lógicas AND, NAND, OR y NOR características de las cuales se muestran en la Fig. 2.18. Se incluye también un biestable del tipo SR como unidad básica de memoria. Las características de este último se muestran en la Fig. 2.19.

Tratándose de simulación de componentes lógicos podemos asumir como valores standar de los niveles lógicos de las señales a:

0L valores de la variable comprendidos entre 0 y 0.2u

1L valores de la variable comprendidos entre 0.8 y 1.0u

notemos que asumiendo estos niveles, tenemos un margen de error

aceptable para la simulación. El ajuste de estos niveles lógicos al valor lógico real presente en un determinado problema se lo hará multiplicando las variables por la constante adecuada.

**Monoestable.** Las características de entrada, salida en función del tiempo se muestran en la Fig. 2.20. Este componente no hace más que generar un pulso de amplitud  $T$  (amplitud programable) en su salida a partir de una señal de disparo en su entrada. Tanto la entrada como la salida de un monoestable corresponden a señales lógicas. Al igual que en el caso de las compuertas lógicas, el ajuste de estas señales a valores reales presentes en el problema de simulación se lo hará con el adecuado escalamiento en magnitud.

El monoestable permite simular componentes temporizados de un sistema. Ejemplos de tales componentes son: relés de tiempo, circuitos de disparo de tiristores, procesos secuenciales digitales, etc.

**Interruptores analógicos.** La Fig. 2.21a muestra la simbología utilizada para este componente. El trabajo que desempeña no es más que el de un interruptor cuya acción bistable es controlada por una señal lógica  $C$ ; así, la salida se conecta con la entrada para  $C=1L$  y permanece abierta para  $C=0L$ .

El interruptor análogo permite simular sistemas cuya operación cambia en función del estado en el que se encuentre o en función del tiempo. Ejemplos de este tipo de sistemas son: el problema del choque elástico de un cuerpo, sistemas de protección, etc.

**Retardo en el tiempo.** Las características entrada-salida de esta función se las muestra en la Fig. 2.22. Este componente permite retrasar en el tiempo una señal. Notemos que esta función es lineal.

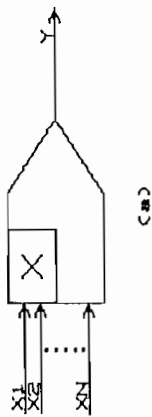
El efecto de retardar una señal en el tiempo va relacionado con componentes físicos que tienen la capacidad de almacenar energía. También se pueden producir retardos en sistemas de grandes dimensiones que tienen que transportar



fluidos.

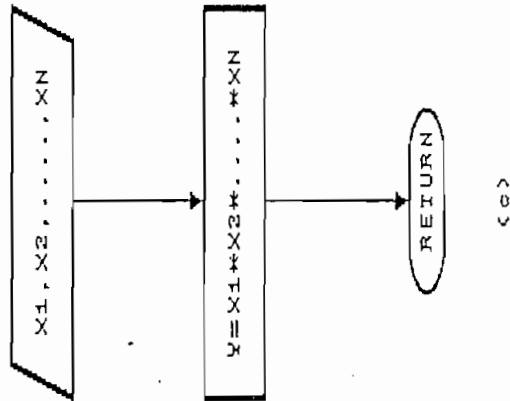
Si bien es verdad, se pueden presentar muchas otras funciones dentro del modelo de un sistema, no es menos cierto que con la ayuda de las funciones antes descritas y con combinaciones en cascada o paralelo de estas, es posible obtener muchas relaciones de transferencia o "módulos matemáticos" adicionales. En resumen, estamos seguros que con el grupo de funciones descritas es posible modelar una amplia gama de complicados sistemas.

FIG 2-16 MULTIPLICADOR



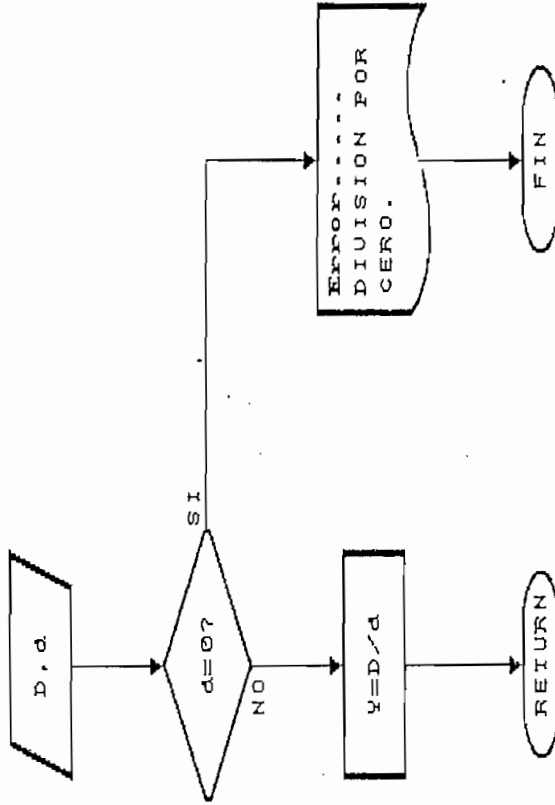
$$Y = X1 * X2 * \dots * XN$$

(b)

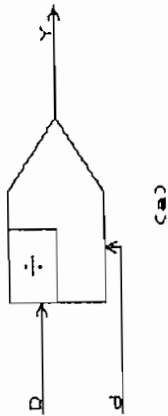


- a.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- b.- ECUACION.
- c.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-17 DIVISOR



(c)



$Y=D/d$  para  $d \neq 0$   
(Cb)

- a.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- b.- ECUACION.
- c.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

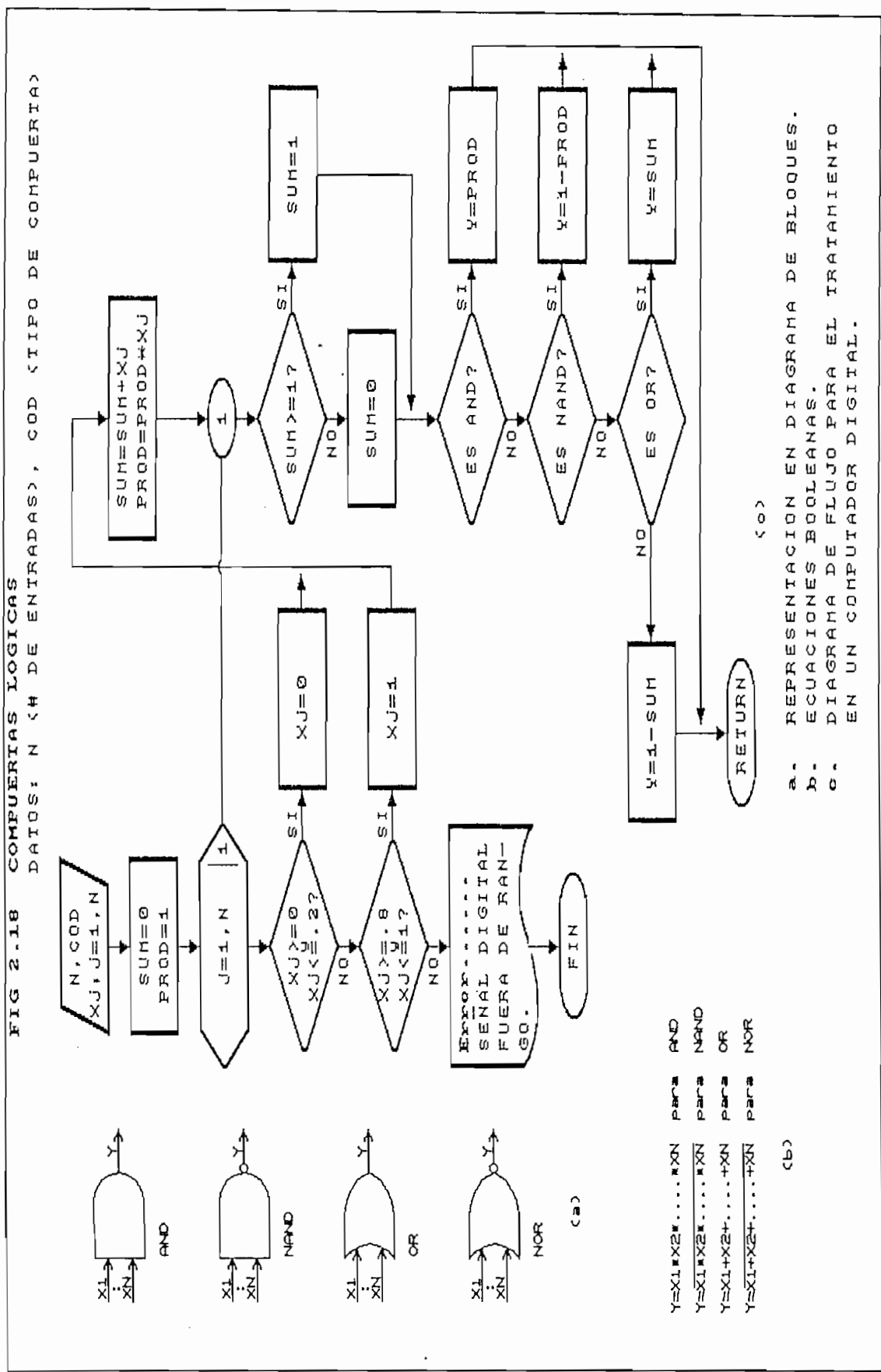
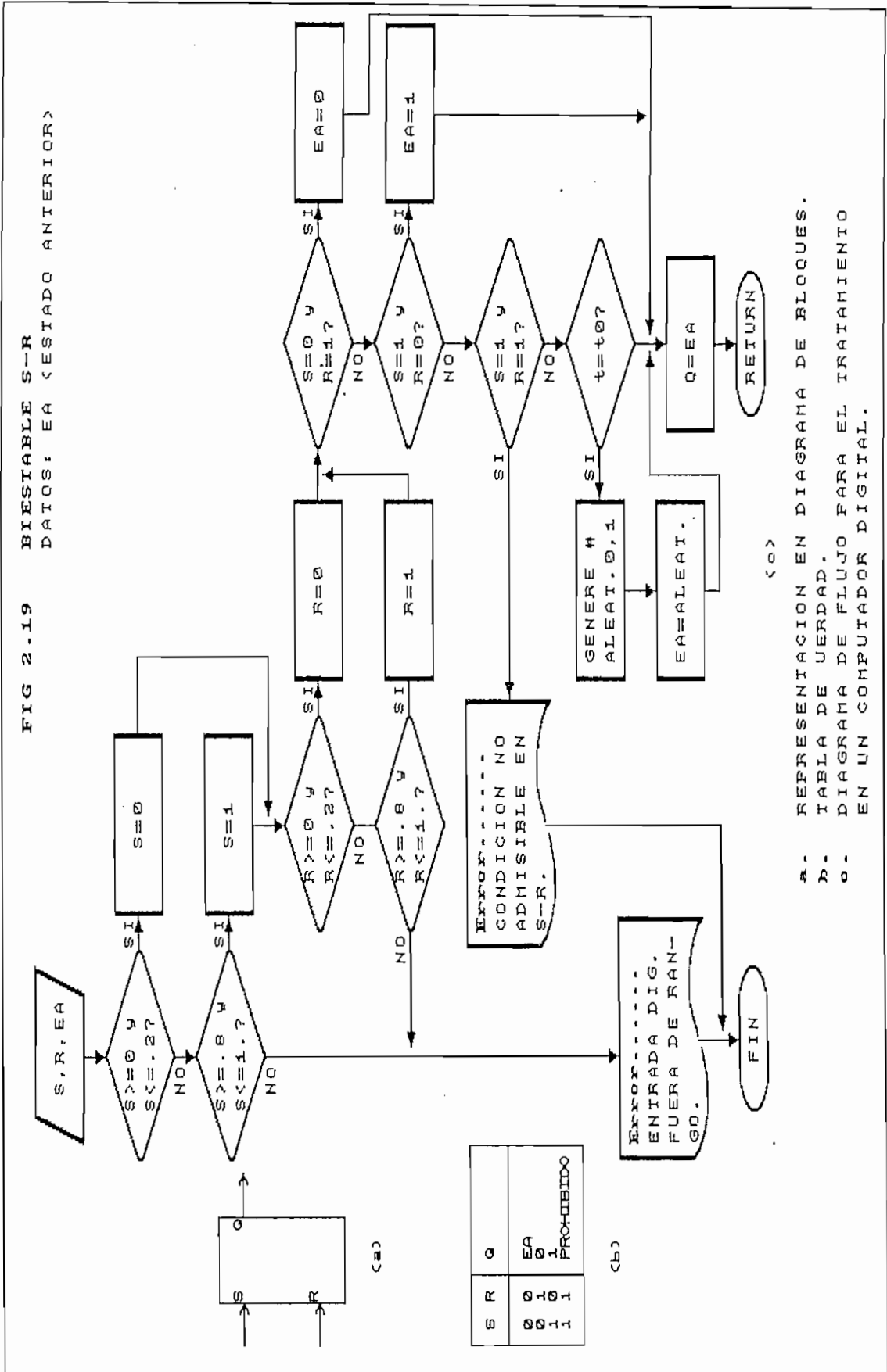


FIG 2.19 BIESIABLE S-R  
 DATOS: EA (ESTADO ANTERIOR)



| S | R | Q | EA |
|---|---|---|----|
| 0 | 0 | 0 | 0  |
| 0 | 1 | 0 | 1  |
| 1 | 0 | 1 | 0  |
| 1 | 1 | 1 | 1  |

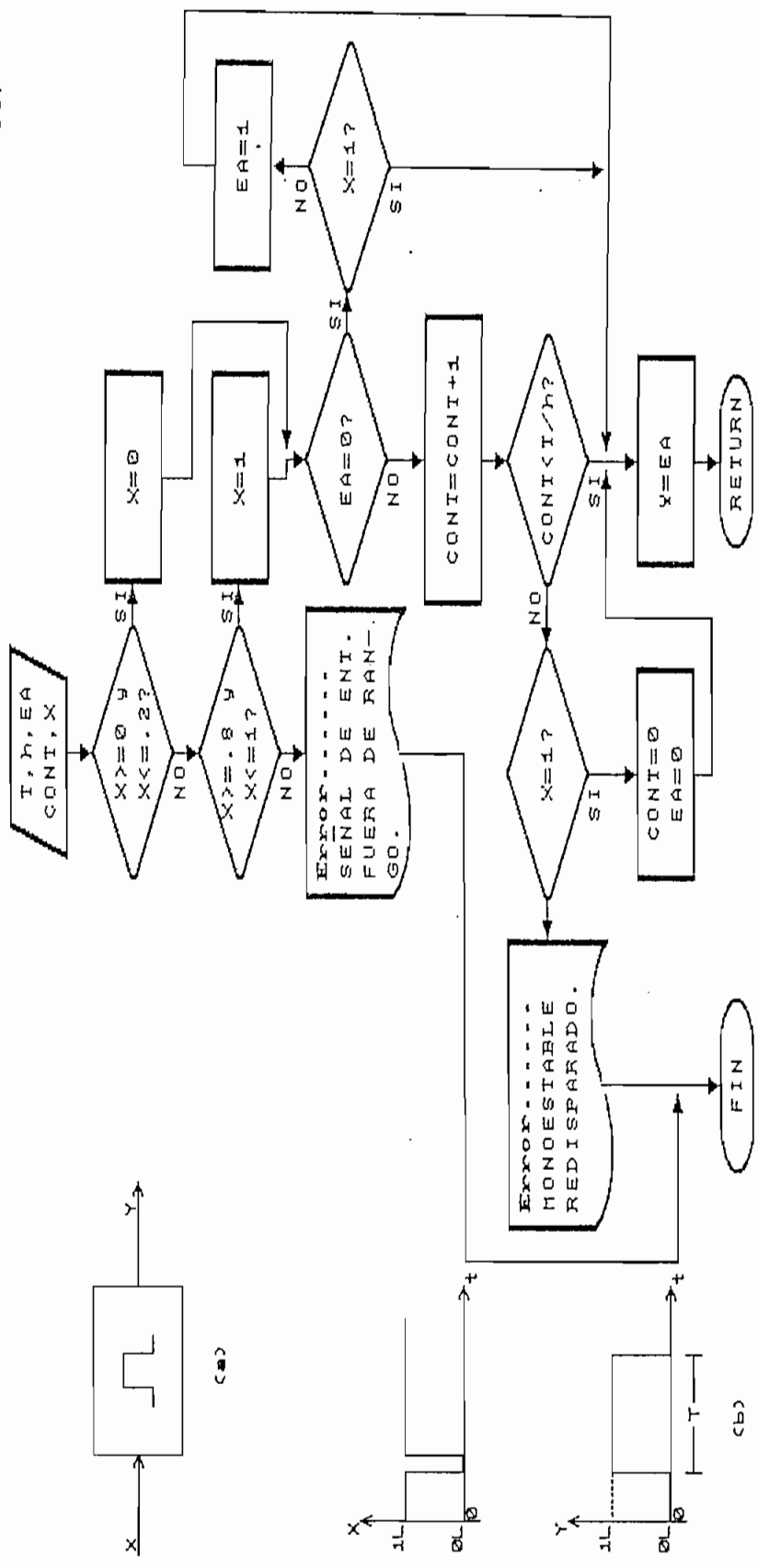
(b)

(a)

- a. REPRESENTACION EN DIAGRAMA DE BLOQUES.
- b. TABLA DE VERDAD.
- c. DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

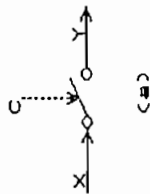
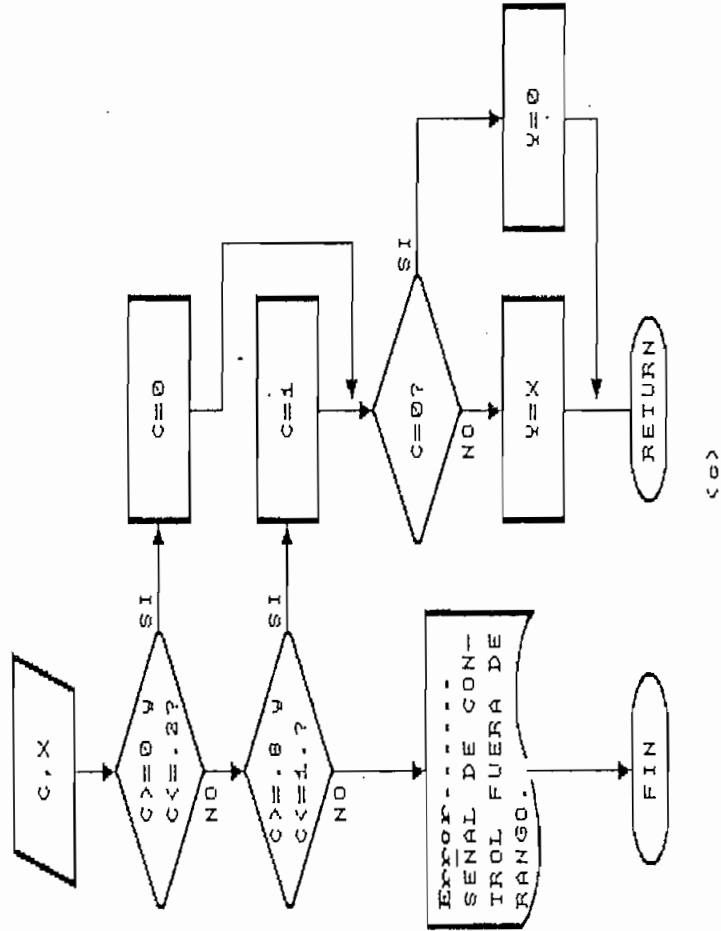
(c)

**FIG 2.20 MONOESTABLE**  
 DATOS: I, h (INTERVALO ENTRE CALCULOS)  
 EA (ESTADO ANTERIOR)  
 CONT (CONTADOR DE INTERVALOS)



- a. REPRESENTACION EN DIAGRAMA DE BLOQUES.
- b. GRAFICOS ENTRADA-SALIDA vs. TIEMPO.
- c. DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-21 SWITCH ANALOGO  
 DATOS: C (CENTRADA DE CONTROL)



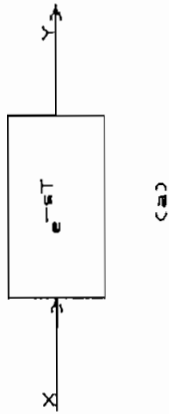
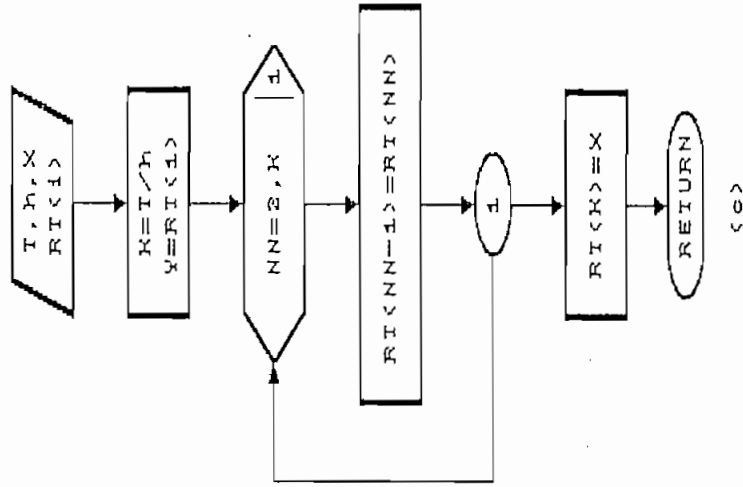
(a)

Y=0      Para      C=0L  
 Y=X      Para      C=1L

(b)

- a.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- b.- ECUACIONES.
- c.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.

FIG 2-22 RETARDO EN EL TIEMPO  
 DATOS: T, h <INTERVALO ENTRE CALCULOS>  
 RI<i>; i=1..T/h <VALORES ANI. DE X>



$Y(t) = X(t-T)$     para  $t \geq T$   
 $Y(t) = 0$         para  $t < T$   
 (b)

- a.- REPRESENTACION EN DIAGRAMA DE BLOQUES.
- b.- ECUACION.
- c.- DIAGRAMA DE FLUJO PARA EL TRATAMIENTO EN UN COMPUTADOR DIGITAL.



## 2.2 EL COMPUTADOR ANALOGICO: MODELOS DE SIMULACION

El computador análogo puede resolver ecuaciones diferenciales. En este tipo de máquinas, una ecuación diferencial es estudiada por medio de *elementos de computación* en los cuales las operaciones matemáticas involucradas en la ecuación, tales como adición, multiplicación, integración, etc., son realizadas considerando las variaciones continuas en el tiempo de las magnitudes físicas del problema. Tanto las entradas como las salidas pueden ser medidas y almacenadas en función del tiempo en forma continua. Usualmente se establece una correspondencia uno a uno entre las magnitudes físicas del problema y las salidas de los elementos de computación.

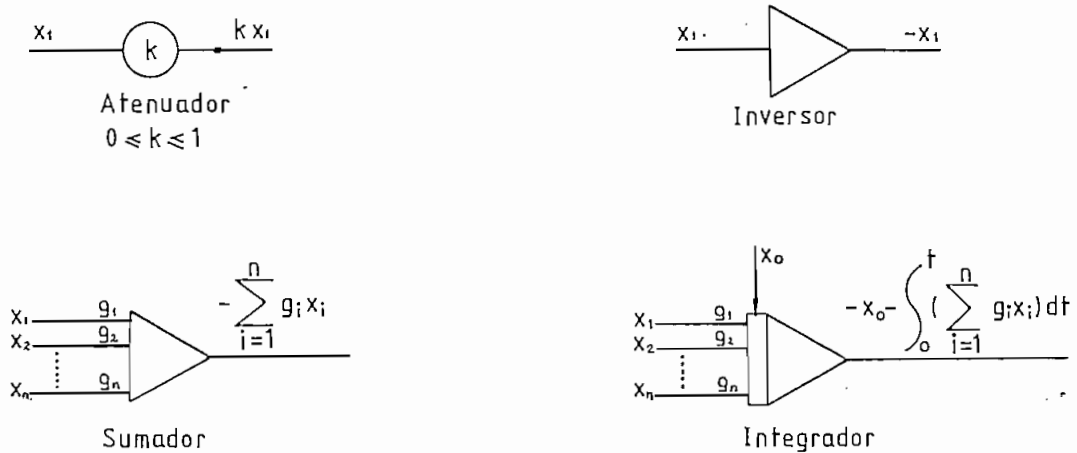
Un computador análogo utiliza bloques o módulos cuyas salidas son voltajes variables con el tiempo. Un computador de propósitos generales permite la interconexión entre los diferentes módulos, el control del proceso de solución y la presentación de las salidas. Debido a que el propósito general de un computador análogo es resolver ecuaciones diferenciales, también se lo suele llamar *analizador electrónico diferencial*.

### 2.2.1 Simbología utilizada.

La operación de un computador análogo se cumple en forma paralela, es decir, todos los módulos de computación operan simultáneamente. Un programa para un computador análogo es un diagrama que muestra como se van a interconectar los diferentes módulos, a este programa se lo llama también diagrama de simulación o modelo de simulación.

La simbología utilizada para representar los elementos de computación es similar a la ya descrita en el numeral anterior. Esta simbología es estandarizada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) y por el Simulation Councils, Inc. (2). Nosotros haremos uso de esta simbología para los diagramas de simulación. Se deberá, sin embargo, considerar que en diferentes textos y/o artículos sobre el tema, los autores suelen introducir simbologías diferentes.

La Fig. 2.23 muestra los símbolos utilizados para los diferentes elementos de computación lineales.



Notas: Todos los  $x_i$  son voltajes funciones del tiempo.

Todos los  $g_i$  son constantes mayores o iguales a cero.

Fig. 2.23 Elementos lineales de computación.

Notemos que:

1. Los sumadores y los integradores invierten la señal en su salida. En realidad se hace esto en los computadores análogos con el fin de facilitar la programación ya que no existe el producto por constantes negativas, sin embargo, nosotros consideraremos a los integradores y sumadores como no inversores (Fig. 2.1), esto lo hacemos en tanto que en lo posterior asumiremos que es posible la multiplicación de una variable por una constante negativa con lo que el problema de la simulación se facilita.

2. Todos los integradores necesitan una condición inicial  $x_0$  dado como un voltaje continuo, y la integración se realizará respecto del tiempo medido en segundos.

3. Los símbolos tienen formas similares a una "flecha", esto indicará la dirección del proceso. Si existe la posibilidad de

confusión, será necesario incluir una flecha para indicar el sentido, tal es el caso del producto por una constante.

4. Los  $g_i$  son llamados ganancias. Cuando  $g_i=1$  usualmente se omite el signo.

Los símbolos típicos de algunos módulos de computación no lineales se muestran en la Fig. 2.24.

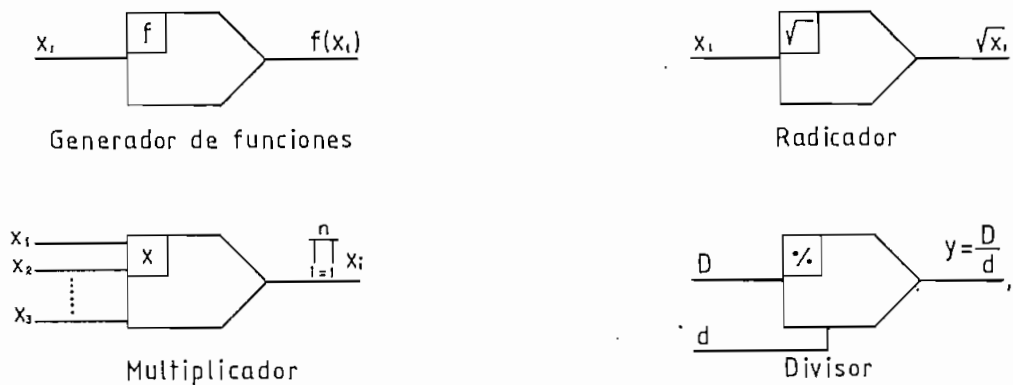


Fig. 2.24 Algunos símbolos de elementos no lineales.

Estos símbolos muestran las características generales en cuanto a la dirección y la función que cumplen, más de estos símbolos se pueden observar en las gráficas de las funciones descritas en el numeral anterior de este capítulo. En general esta simbología es utilizada para trabajar con el diagrama de simulación mas no para la interconexión física de los elementos. Por ejemplo, muchos computadores no tienen un elemento divisor, y el multiplicador es conectado de una forma especial para obtener un divisor.

### 2.2.2 Equipo de medida e impresión.

Para buenas presentaciones del voltaje de salida de cada uno de los componentes, se dispone de un equipo de medida en los computadores analógicos. No todos los equipos que describiremos están disponibles en los diferentes computadores. La descripción que haremos no está en ningún orden particular.

Voltímetro digital. Un voltímetro digital permite medir el voltaje estático en la salida de cualquier bloque.

Registrador de señales. Un registrador de señales, usualmente de 6 a 8 canales esta disponible para graficar los voltajes en función del tiempo.

Plotter. Esta versátil unidad permite graficar un voltaje en función de otro voltaje en un plano de coordenadas rectangulares.

Osciloscopio. Un osciloscopio de rayos catódicos es indispensable para observar voltajes que varían con velocidad.

Impresoras. Algunos computadores disponen de una impresora para registrar en forma numérica los valores de voltaje de las salidas de los módulos.

### 2.2.3 Interconexiones en el panel de programación.

Las entradas y las salidas de los módulos de computación son electricamente conectados en una localización central en la consola del computador llamado *panel de programación*. Generalmente este panel es removible y las conexiones se realizan con cables entre los contactos debidamente indicados en el panel. Para facilitar las interconexiones entre módulos se utiliza un código de colores para diferenciar las entradas y las salidas de los componentes.

Adicionalmente se dispone en una parte del panel de una variedad de terminaciones del equipo periférico eliminando la necesidad de grandes cables para la conexión de estos con el circuito simulador.

### 2.2.4 Controles del computador.

Los controles del computador proveen muchas opciones para hacer trabajar a un circuito de simulación en diversos modos de operación. Los más importantes de estos controles son:

Reset o condiciones iniciales (IC). Este modo se lo selecciona para poner los valores de las condiciones iniciales de los integradores.

Modo de operación. Arranca las integraciones de las entradas de todos los integradores. El tiempo de computación, inicialmente cero, es medido desde el instante en el que el computador es puesto en modo de operación. El equipo de medida también arrancará automáticamente desde el momento en que se pone el modo de operación.

Modo Hold. Detiene la integración. Los voltajes de salida de todos los módulos en el instante en el que el computador es puesto en hold se mantienen constantes. Este modo se selecciona para observar resultados intermedios dentro de un programa de simulación. El operador puede continuar en la ejecución regresando el computador al modo de operación.

Muchos computadores disponen de un sistema de control de reloj que permiten periódicamente pasar al computador entre los modos de Reset y de operación. Estos computadores reciben el nombre de *repetitivos*. La ventaja de esto radica en que soluciones obtenidas se pueden observar en un osciloscopio y analizar en forma instantánea el efecto de cambios en los parámetros del sistema.

Muchas sofisticadas opciones son también disponibles en algunos computadores. Individualmente algunos integradores pueden ser puestos en varios modos de operación por medio de arreglos lógicos. Los computadores que disponen de elementos lógicos se los llama computadores *análogos-híbridos*. Un computador con estas capacidades dispone generalmente de un panel lógico. Las salidas de los componentes lógicos corresponden a señales binarias de niveles lógicos 0L o 1L. Las funciones lógicas comunmente disponibles en un computador análogo/híbrido son las mostradas en 2.1.3.

Finalmente, existen computadores análogos que pueden ser enlazados a computadores digitales. El computador digital provee de mucha memoria y puede realizar muchos cálculos para el control del funcionamiento de los integradores análogos. A este tipo de computadores se los llama *computadores híbridos*.

#### 2.2.5 Diagramas de simulación analógica.

... puntos  $t_1$  y  $t_2$ . La fig. 2.26 muestra el diagrama final de computación en el cual el lazo final de realimentación ha sido cerrado. La solución  $y(t)$  es producida

continuamente en el proceso.

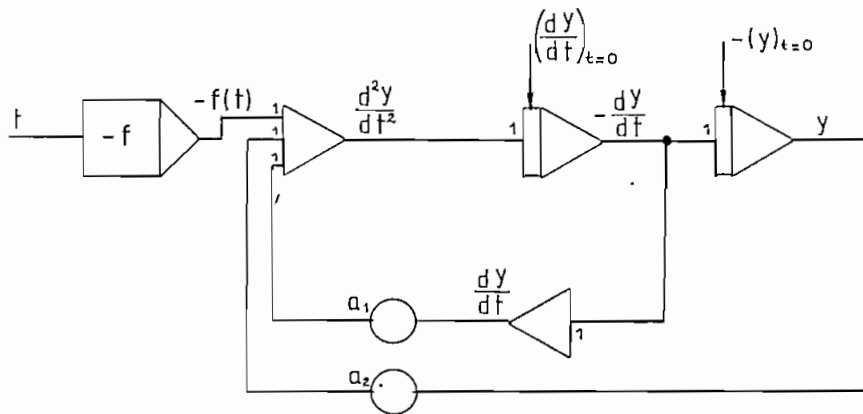


Fig. 2.26 Diagrama de simulación con el lazo cerrado.

Posteriormente veremos que se necesitan hacer algunas variantes al diagrama de simulación de la Fig. 2.26 cuando este vaya a ser tratado en un computador digital.

Siguiendo el método indicado en este ejemplo se puede obtener el diagrama de simulación analógica de cualquier modelo matemático sea este lineal o no lineal. En el caso de ser no lineal, la base para obtener el diagrama de simulación es la de determinar el número de las integraciones presentes en la ecuación (orden de la ecuación) y aplicar correctamente las funciones lineales o no lineales que afectan a las entradas de los integradores. En resumen, asumimos que a partir de este punto estamos en capacidad de obtener un diagrama de simulación como el de la Fig. 2.26 para cualquier problema. En adelante nos preocuparemos de como evaluar el modelo de simulación en un computador digital.

### 2.3 TRATAMIENTO DEL MODELO DE SIMULACION EN UN COMPUTADOR DIGITAL

Existen algunas formas de tratar el modelo de simulación en forma, discreta. Consideremos por ejemplo el método desarrollado por Lord Kelvin (3) aplicado al problema de la Fig. 2.25. Primero se genera una función en el punto Q en términos de  $y$ ,  $dy/dt$  y  $f(t)$  asumiendo una función arbitraria

$F_1(t)$  como alimentación al punto P. Usar la función generada en Q como una nueva función  $F_{i+1}(t)$  que se aplica en P. De esta forma, la secuencia de funciones  $F_i(t)$ ,  $i=1,2,\dots$ , cumple con la condición de que  $F_{i+1}-F_i$  tiende a cero a medida que  $i$  crece. Considerando un criterio adecuado de convergencia podemos encontrar las respuestas de todos los puntos del sistema en cualquier instante de tiempo. El proceso descrito es conocido como el método de Picard o método predictor-corrector para la integración de ecuaciones diferenciales. Notemos que este método aunque puede tener una precisión muy elevada (según el criterio de convergencia utilizado) se vuelve lento ya que para llegar a la convergencia hacen falta muchas iteraciones.

Nosotros no vamos a considerar un método de este tipo sino mas bien trataremos de desarrollar un método general aplicable a cualquier sistema. En realidad, el mayor problema que se presenta en el tratamiento discreto del modelo de simulación es el de la estimación de la salida de un integrador en el tiempo  $t_{k+1}$ , conociendo su entrada en el tiempo  $t_k$ . Es solamente en estos puntos donde debemos aplicar un criterio de integración como el de Euler o de Runge-Kutta ya que para todos los módulos del diagrama que no correspondan a integradores es factible evaluar su salida a partir de la(s) entrada(s) en cualquier instante de tiempo. El método que a continuación describiremos se basa en el algoritmo de Euler (sección 1.2.3) para la integración de ecuaciones diferenciales, mas adelante, cuando describamos el algoritmo de Runge-Kutta aplicaremos el mismo método, tan solo con la variante de que la estimación de la nueva salida de los integradores se basará en este algoritmo.

### 2.3.1 Tratamiento discreto del modelo de simulación basado en el algoritmo de Euler.

Partimos de las siguientes consideraciones:

1. Conocemos el diagrama de simulación y los valores numéricos de todos los parámetros involucrados en este.
2. El modelo es consecuencia de un sistema dinámico, es

decir, incluye integradores.

3. Conocemos las condiciones iniciales de todos los integradores y el intervalo de tiempo  $(t_0, t_f)$  dentro del cual se va a evaluar el modelo.

4. Las entradas externas del modelo están definidas para todo el intervalo de interés.

Basados en estas consideraciones hacemos:

1. En base al valor de las constantes de tiempo involucradas en el problema y el máximo error permisible determinamos un valor adecuado de  $h$  (intervalo de tiempo entre cálculos). Recordemos que el error de paso estimado para el método de Euler es del orden de  $h^2$ .

2. Para  $t=t_k=t_0+kh$  evaluamos cuales son las salidas de cada uno de los módulos del diagrama. Esto es fácil de realizar partiendo de que conocemos la salida de los integradores en  $t=t_k$ , y aplicando los algoritmos presentados en los diagramas de flujo para las diferentes funciones descritas en 2.1 que puedan estar presentes en el modelo. El procedimiento parte desde  $k=0$ , es decir  $t=t_0$ , en este instante el valor de la salida de un integrador es igual a su condición inicial.

3. Hasta ahora, conocemos el valor de las salidas de cada uno de los módulos en el tiempo  $t_k$  y considerando que las entradas de los integradores corresponden a salidas de otros módulos o entradas del sistema, entonces son conocidas estas señales en  $t_k$ . Para realizar el cálculo en el tiempo  $t_{k+1}=t_k+h$  realizamos la aproximación para las salidas de los integradores en este nuevo tiempo como sigue: sea  $y_1(t_k)$  el valor de la salida de un integrador en  $t_k$  y  $y_2(t_k)$  el valor de su entrada en  $t_k$  entonces

$$y_1(t_k+h) = y_1(t_k) + hy_2(t_k) \quad (2.8)$$

4. El procedimiento se repite para  $k=0,1,2,\dots,n$ , donde  $t_0+nk=t_f$ .

Notemos que el procedimiento descrito es bastante simple, como se puede ver en el ejemplo del capítulo anterior donde



hemos aplicado este método para un sistema relativamente complejo obteniendo resultados bastante confiables con un tiempo de computación pequeño y utilizando poca memoria del computador.

El problema del método radica en el error de paso, si bien es cierto que este error puede ser reducido utilizando un valor de  $h$  muy pequeño esto produciría que el cálculo se vuelva demasiado lento por lo cual deberemos aplicar otro algoritmo de integración de ecuaciones diferenciales que en tiempos relativamente pequeños nos de resultados confiables. En nuestro caso hemos escogido el algoritmo de Runge-Kutta de cuarto orden el cual permite tener un error de paso estimado del orden de  $h^4$ , error similar al que se obtiene con la utilización de otros algoritmos más complicados como son los de fórmulas multipaso o métodos predictor corrector. En el siguiente punto describiremos este algoritmo.

#### 2.4 ALGORITMO DE RUNGE-KUTTA

Como se mencionó antes, el método de Euler no es muy útil en problemas prácticos porque requiere de un tamaño de paso muy pequeño para una precisión razonable. Las fórmulas multipaso o métodos predictor corrector no son muy aplicables al problema que nos concierne debido a que requieren de un gran tiempo de compilación y/o gran capacidad de memoria. Los métodos de Runge-Kutta para la integración de ecuaciones diferenciales permiten obtener una alta precisión en los cálculos, calculando la función  $f(x,y)$  en puntos seleccionados de cada subintervalo y son de fácil aplicación. No vamos a presentar la deducción de las fórmulas de Runge-Kutta ya que no es de nuestro interés este problema matemático (4).

El más simple de los métodos de Runge-Kutta es el llamado de orden dos cuya forma es:

para resolver la ecuación  $dy/dx=f(x,y)$ ;  $y(x_0)=y_0$  en el intervalo  $(x_0, x_+)$ .

Escogemos un intervalo  $h$  adecuado tal que  $x_k = x_0 + kh$

Se aplica una fórmula de aproximación de la siguiente manera:

$$y_{k+1} = y_k + (k_1 + k_2) / 2 \tag{2.9}$$

donde

$$k_1 = hf(x_k, y_k)$$

$$k_2 = hf(x_k + h, y_k + k_1)$$

Este algoritmo se puede representar geoméricamente como en la Fig. 2.28. El método de Euler da un incremento  $P_1P_0 = hf(x_k, y_k)$  a  $y_k$ ;  $P_2P_0 = hf(x_k + h, y_k + hf(x_k, y_k))$  es otro incremento basado en la pendiente obtenida en  $x_{k+1}$ . Tomando el promedio de estos incrementos se llega a la fórmula 2.9.

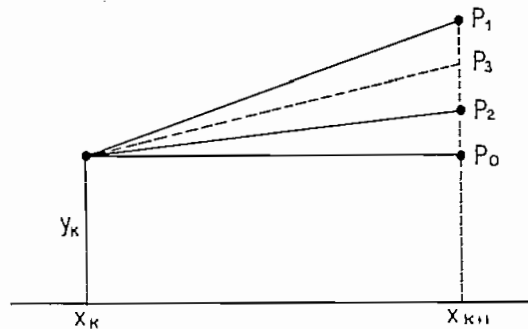


Fig. 2.28 Representación geométrica del algoritmo de Runge-Kutta de segundo orden.

El error de paso del algoritmo descrito es sumamente complejo, como es característica de los métodos de Runge-Kutta y constituye una de las características menos deseables de tales métodos ya que los estimativos de error de paso son muy difíciles de obtener. El error local de 2.9 es, sin embargo, de orden  $h^3$ , mientras que el del método de Euler es  $h^2$ . Podemos por tanto esperar que se pueda utilizar un tamaño más grande de paso con 2.9. El precio que pagamos por esto es que debemos evaluar la función  $f(x, y)$  dos veces por cada paso de integración. Las fórmulas del tipo de Runge-Kutta para cualquier orden se pueden deducir por métodos similares, sin embargo, las deducciones se vuelven excesivamente complicadas. La fórmula de este tipo que es más popular y más usada es la de cuarto orden que la enunciamos a continuación.

Para la ecuación  $dy/dx=f(x,y)$ ,  $y(x_0)=y_0$ , generar aproximaciones desde  $y_k$  a  $y_{k+1}$  para  $h$  fijo y para  $k=0,1,2,\dots$ , usando la fórmula de recurrencia

$$y_{k+1} = y_k + (k_1 + 2k_2 + 2k_3 + k_4) / 6 \quad (2.10)$$

donde

$$\begin{aligned} k_1 &= hf(x_k, y_k) \\ k_2 &= hf(x_k + h/2, y_k + k_1/2) \\ k_3 &= hf(x_k + h/2, y_k + k_2/2) \\ k_4 &= hf(x_k + h, y_k + k_3) \end{aligned}$$

El error local de truncamiento del algoritmo 2.10 es del orden de  $h^4$ . De nuevo el precio que pagamos por el error de discretización favorable es que se requieren cuatro evaluaciones funcionales por cada paso. Este precio puede ser considerable en tiempo de computador para aquellos problemas en los cuales la función  $f(x,y)$  es complicada. La fórmula 2.14 es muy usada en la práctica con considerable éxito. Tiene la ventaja importante de que se autoinicia, es decir, requiere solo el valor de  $y$  en un punto  $x=x_k$  para hallar  $y$  y  $y'$  en  $x=x_{k+1}$ .

Posteriormente, en el siguiente capítulo, veremos la aplicación de este método en el tratamiento digital del diagrama de simulación.

#### REFERENCIAS

1. ING. LUIS BARAJAS, ING. PATRICIO BURBANO, *Laboratorio de Sistemas de Control*. E.P.N., 1981; pp. 19.
2. SIMULATION Vol. 6, *Uniform Graphics for Simulation*. March 1966; Revised in vol. 9; Dec. 1967.
3. ARTHUR HAUSNER, *Analog and Analog/Hybrid Computer Programming*. Prentice-Hall International, 1971; pp. 10.
4. S. D. CONTE, y CARL de BOOR, *Análisis Numérico Elemental*. McGraw-Hill, 1972; pp. 357-361.

CAPITULO III

## CAPITULO III

### DESARROLLO DEL PROGRAMA

En este capítulo presentaremos el desarrollo del programa SIMULACION DIGITAL DE SISTEMAS DE CONTROL utilizado para el tratamiento del diagrama de simulación en un computador digital. En realidad, con este programa trataremos de "convertir" al conjunto computador digital-impresora en un poderoso computador análogo. Para la implementación del programa partiremos de los requerimientos básicos exigidos en este, los cuales serán considerados en base a las características de un computador análogo.

El programa desarrollado deberá ser lo más general posible en lo referente a la entrada de datos; así, el programa debe ser capaz de interpretar cualquier estructura de un diagrama de simulación sin cambiar la estructura del programa de entrada de datos.

Cuando se está realizando el diseño de un sistema siempre es necesario investigar cual es el comportamiento del mismo ante la variación de uno o más de sus parámetros. Este trabajo en un computador analógico es bastante fácil ya que tan solo basta cambiar los potenciómetros que representen a las constantes y visualizar los cambios en la respuesta del sistema simulado. En el computador digital esta opción debe estar disponible, es decir, será factible cambiar los valores de las constantes y funciones de entrada al sistema sin necesidad de introducir nuevamente la totalidad de los datos.

El computador analógico puede ser programado para generar una gran cantidad de funciones de entrada, si bien es cierto que la prueba de sistemas se realiza en base a perturbaciones de tipo impulso, escalón o rampa a veces puede requerirse de

entradas de otro tipo tales como ruido con determinada función de probabilidad o señales sinusoidales. El programa digital debe ser capaz de generar cualquier tipo de función de entrada, esto se consigue utilizando las funciones de biblioteca disponibles en el lenguaje de programación de alto nivel que se utilice para el desarrollo del programa.

En cuanto a la presentación de resultados, similar a como un computador análogo permite obtener resultados numéricos y gráficos de cualesquiera de los puntos del diagrama de simulación, en el computador digital se debe hacer lo mismo. En realidad, considerando las grandes ventajas que presenta un computador digital en este sentido es factible superar ampliamente las características de precisión y presentación de los resultados.

Cuando el diagrama de simulación es corrido en un computador analógico y se presentan fallas de programación tales como salidas interconectadas o fallas de ejecución como divisiones por cero, raíces de números negativos, etc., el computador da la indicación respecto del tipo de falla y de la ubicación de esta dentro del diagrama de simulación. En el caso del programa de simulación, este debe estar capacitado para realizar un continuo análisis tanto durante la entrada de datos como durante la ejecución para ubicar y presentar al usuario la indicación respectiva sobre posibles fallas que puedan presentarse tanto en la introducción de datos como en la ejecución.

Debido a que un computador analógico realiza la evaluación de un diagrama de simulación en forma paralela, es factible obtener la respuesta de un sistema en tiempos muy cortos de ejecución. Si bien es verdad, al trabajar con un computador digital en el cual las operaciones se realizan en forma serial no es posible igualar siquiera en velocidad a un computador análogo, trataremos sin embargo de que el tiempo de ejecución sea el menor posible. Esto se conseguirá utilizando para el desarrollo del programa un lenguaje compilado de alto nivel.

Los errores obtenidos al tratar el diagrama de simulación en un computador análogo estan en el orden del 1%, error que puede ser considerado despreciable en la gran mayoría de casos. En el caso del programa digital de simulación el error obtenido es consecuencia del método de integración utilizado, el paso de integración y del redondeo de números durante los cálculos, a pesar de que resulta bastante difícil estimar el verdadero error resultante de una evaluación, estamos seguros de que si escogemos un algoritmo de integración de alto orden (Runge-Kutta de cuarto orden) y un paso de integración suficientemente pequeño, se obtendrán errores menores al valor antes citado.

Debido a la amplia acogida que han tenido en nuestro medio los computadores personales de la firma IBM, el programa está desarrollado para trabajar en un computador de este tipo o uno compatible. Adicionalmente se tiene la opción para instalarlo en equipos con diferentes configuraciones.

En base a las consideraciones antes detalladas, a continuación presentaremos la estructura y el modo de operación del programa desarrollado.

El programa se compone fundamentalmente de tres bloques: la entrada de datos, la evaluación de los datos y la presentación de resultados. Cada uno de estos bloques es independiente de los otros teniendo como único lazo de unión el nombre del archivo de datos y resultados. El programa principal llamado PRES.BAS desarrollado en lenguaje GW-BASIC (1) es el que se encarga del encadenamiento entre los diferentes programas involucrados. En el diagrama de bloques de la Fig. 3.1 se muestra la estructura de este programa.

El acceso al programa PRES.BAS es directo desde el sistema tecleando las siglas SDSC (Simulación Digital de Sistemas de Control), el archivo de proceso por lotes SDSC.BAT se encarga de cargar el lenguaje GW-BASIC y correr el programa. El programa PRES.BAS da acceso al menú principal con las siguientes opciones.

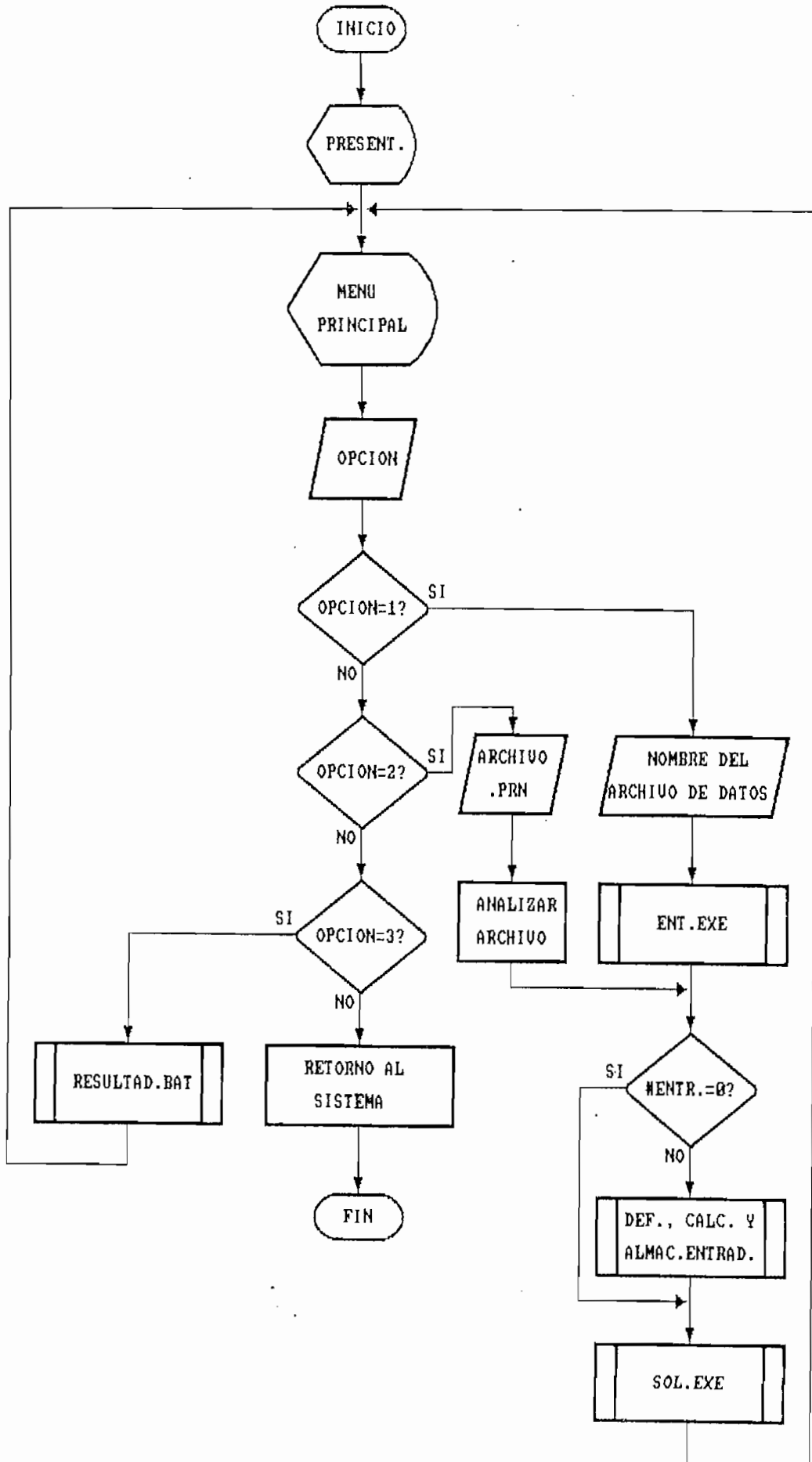


FIG. 3.1 Diagrama de bloques del programa PRES.BAS



1. Problema nuevo
2. Problema con datos cambiados
3. Revisar un archivo de datos
4. Terminar.

La opción 1. permite al usuario introducir los datos de la estructura de un diagrama de simulación para ser evaluado. Inmediatamente después de que se escoge esta opción, el programa pide por el nombre del archivo de datos. El archivo de datos es un archivo en ASCII que al final de la ejecución contiene los datos y los resultados del problema analizado. Como ya se dijo anteriormente, el nombre del archivo de datos (sin extensión) es el único vínculo de unión entre los diferentes programas que intervienen en el proceso de evaluación del diagrama, con este propósito, este nombre es almacenado en un archivo llamado NOMBRE.HLP que se crea en el directorio corriente. Posteriormente se ejecuta el llamado al programa ENT.EXE que es el programa que contiene los menús y las instrucciones necesarias para introducir los datos referentes a la estructura, valores de las constantes y valores de tiempo del diagrama de simulación. Este programa crea temporalmente un archivo de datos en ASCII de nombre el inicialmente indicado y de extensión .DAT. Terminada la ejecución del programa ENT.EXE, se definen las funciones de entrada del sistema (si es que existen) mediante la introducción de líneas de programa BASIC; así, el programa PRES.BAS se interrumpe luego de dar la indicación del formato requerido para la programación de las funciones de entrada. Esta forma de definir las entradas justifica el hecho de que el programa principal haya sido codificado con el lenguaje BASIC no compilado. Al arrancar nuevamente el programa PRES.BAS, se calculan los valores de las entradas en el intervalo de tiempo de interés y los resultados se almacenan temporalmente en un archivo de extensión .ENT.

Para la evaluación del diagrama se llama al programa SOL.EXE. Al final de la ejecución de este programa, los datos, valores de las entradas y valores de las salidas son almacenados en un archivo de extensión .PRN para luego ser analizados con la opción 3. del menú principal.

los programas ENT.EXE, SOL.EXE y RESULTAD.BAT.

### 3.1 ENTRADA DE DATOS

Los datos de un problema nuevo se ingresan al computador basándose en un diagrama de simulación previamente elaborado. El diagrama de simulación para ser introducido en el computador digital debe tener las siguientes características.

Ningún módulo invierte la señal, esto se lo hace debido a

que en el computador digital podemos definir el producto de una función por una constante negativa por tanto, para invertir una señal bastará multiplicarla por  $-1$ . En realidad esto simplifica mucho el diagrama de simulación.

Las constantes involucradas en el modelo de simulación tales como valores de los potenciómetros, condiciones iniciales, valores referenciales de los módulos no lineales, etc. pueden tener cualquier valor real, el único limitante es que no excedan en valor absoluto a los valores máximo y mínimo permisibles para variables en el computador ( $1.7E+38$ ,  $5.8E-39$ ).

El diagrama de simulación podrá contener solamente los módulos que se indican en la Tabla 3.1. El número de entradas que se indica para cada uno de los módulos no podrá ser excedido.

| CODIGO | MODULO                    | No. ENTRADAS |
|--------|---------------------------|--------------|
| 0      | Sumador                   | 2 o más      |
| 1      | Integrador                | 1            |
| 2      | Limitador                 | 1            |
| 3      | Comparador                | 1            |
| 4      | Comparador con histéresis | 1            |
| 5      | Rectificador 1/2 onda     | 1            |
| 6      | Valor absoluto            | 1            |
| 7      | Zona muerta               | 1            |
| 8      | Alinealidad de juego      | 1            |
| 9      | Cuadrador                 | 1            |
| 10     | Raíz Cuadrada             | 1            |
| 11     | Logaritmo natural         | 1            |
| 12     | Exponencial               | 1            |
| 13     | Seno                      | 1            |
| 14     | Coseno                    | 1            |
| 15     | Retardo en el tiempo      | 1            |
| 16     | Monoestable               | 1            |
| 17     | Multiplicador             | 2 o más      |
| 18     | Divisor                   | 2            |
| 19     | Switch análogo            | 2            |
| 20     | Compuerta AND             | 2 o más      |
| 21     | Compuerta NAND            | 2 o más      |
| 22     | Compuerta OR              | 2 o más      |
| 23     | Compuerta NOR             | 2 o más      |
| 24     | Biestable SR              | 2            |

Tabla 3.1 Módulos disponibles en el programa digital de simulación.

Todas las entradas que van a un módulo son afectadas por un potenciómetro antes de llegar al mismo, es decir, los potenciómetros no son tratados como módulos independientes sino que pasan a ser parte de los bloques indicados en la Tabla 3.1. Esto no sucede para las entradas externas del sistema, en este caso se considera que el potenciómetro que las afecta antes de llegar al módulo tiene un valor de 1.

Las entradas o perturbaciones externas pueden ser aplicadas solamente a módulos que acepten más de una entrada.

El programa está diseñado para el análisis de sistemas dinámicos, en otras palabras, se puede evaluar diagramas de simulación que tengan por lo menos un integrador. El programa no está capacitado para evaluar sistemas estáticos, ya que no se dispone de algoritmos para resolver sistemas de ecuaciones lineales, no lineales o trascendentes. En resumen, el diagrama de simulación deberá incluir por lo menos un integrador y no tener configuraciones que den como resultado sistemas de ecuaciones al relacionar las diferentes salidas del mismo.

Se consideran como salidas del sistema a cada una de las salidas de los módulos involucrados en el correspondiente diagrama de simulación. Para diferenciar estas salidas, se debe enumerar (sin importar el orden) cada uno de estos puntos.

Las entradas externas del sistema son llamadas  $u_i$   $i=1...l$ , en otras palabras, cada entrada externa deberá llevar el subíndice que la diferencie de las otras.

Una vez que se dispone del diagrama de simulación modificado según las características antes detalladas, es fácil introducir los datos requeridos por el programa ENT.EXE. Las características en detalle de la forma de introducir los datos y de la presentación de las diferentes pantallas para el efecto se pueden encontrar en el ANEXO A del presente trabajo. A continuación describiremos en líneas generales cual es la forma de operar del mencionado programa.

El programa ENT.EXE está desarrollado en lenguaje GW-BASIC y compilado. El acceso al mismo se lo hace desde el programa PRES.BAS al escoger la opción 1. del menú principal. El diagrama de flujo mostrado en la Fig. 3.2 indica la secuencia de operación de este programa.

Los datos requeridos por el programa son:

1. Número de salidas. La salida de cada uno de los módulos del diagrama se considera una salida del sistema. El valor se almacena en la variable entera m.

2. Matriz de incidencia. Es una matriz de ceros y unos que contiene la información sobre la topología del diagrama. Así, si la salida de un módulo está numerada con i, la salida de otro con j y la entrada del módulo cuya salida es j corresponde a la señal i, la matriz de incidencia tendrá un uno en el término ubicado en la fila i y la columna j. Si no existe ningún ramal directo entre estas dos salidas el término correspondiente será un cero.

Fudiendo haber enlaces directos de ida y regreso entre dos puntos del diagrama de simulación, el orden de la matriz de incidencia deberá ser por lo menos m por m, sin embargo, se aprovecha la última fila de la matriz (fila m+1) para definir la incidencia de las entradas, de tal forma que, si una entrada se aplica a un módulo cuya salida está numerada con j, el término de la fila m+1 y la columna j vale uno; de no existir ninguna entrada aplicada al módulo, este término valdrá cero.

La matriz de incidencia se almacena en el arreglo de variables enteras de dimensiones m+1 por m llamado MI.

Notemos que la forma de definir la MI evita errores de programación tales como la interconexión directa entre las salidas de dos módulos. Adicionalmente se puede verificar si las entradas externas están siendo aplicadas a módulos que aceptan más de una entrada, esto se puede hacer comprobando que exista por lo menos un uno en una columna que tiene un uno en su última fila, de no ser así, el programa pide nuevamente por la definición de la matriz.

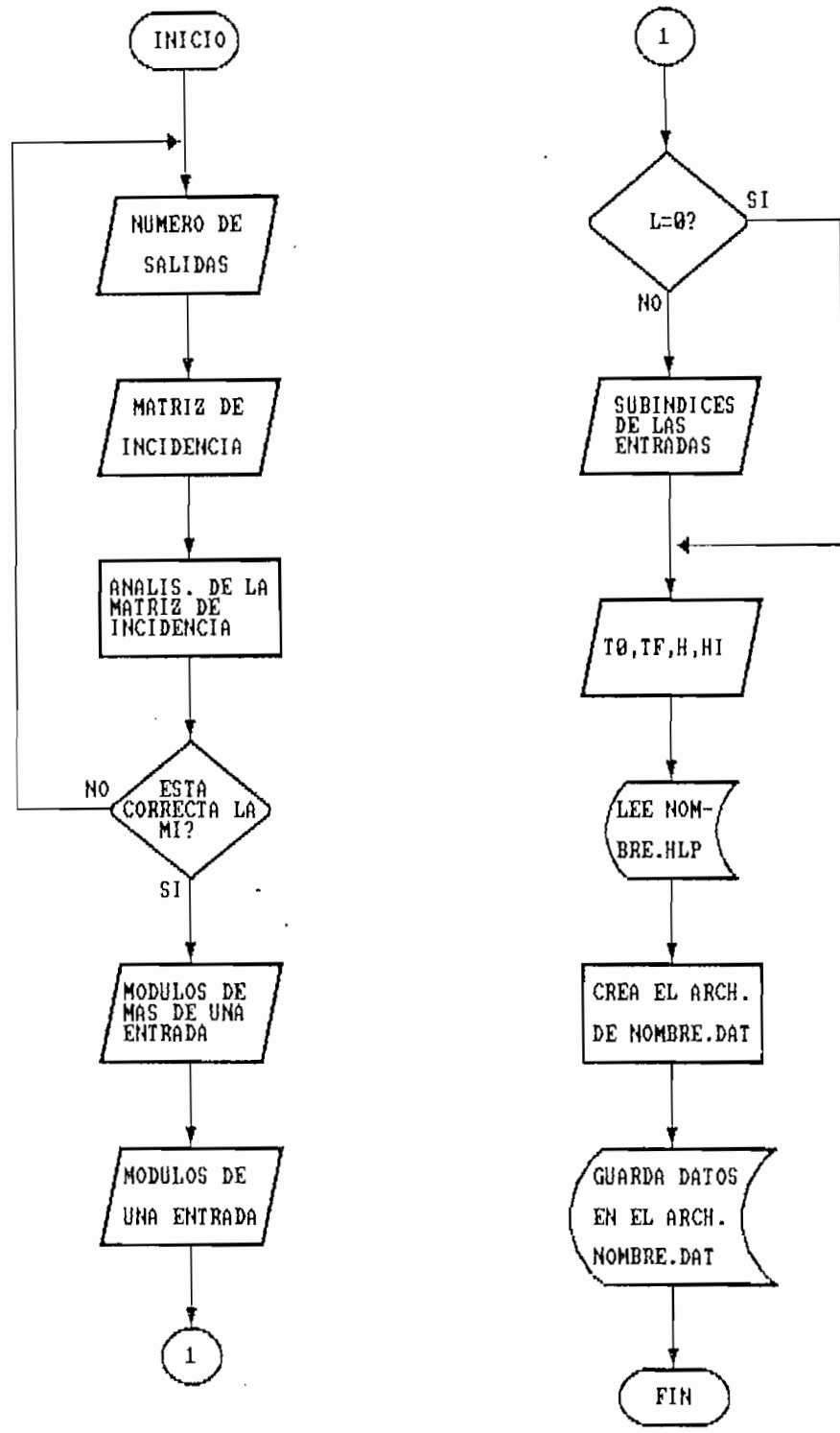
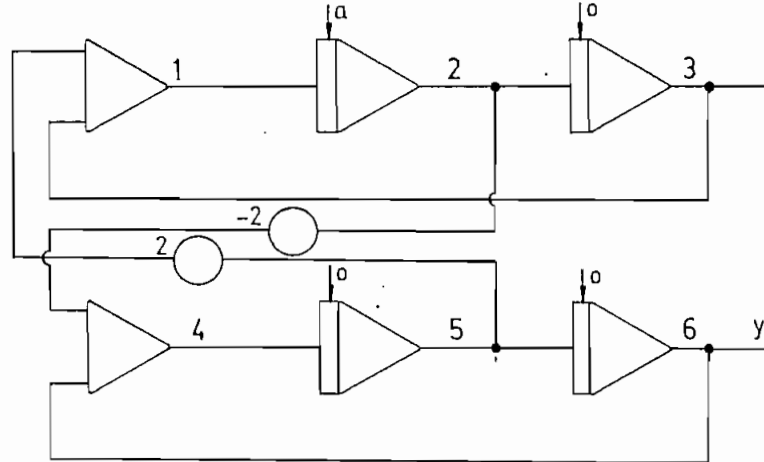


FIG. 3.2 DIAGRAMA DE FLUJO DEL PROGRAMA ENT.EXE

El diagrama de simulación, junto con una correcta definición de la matriz de incidencia son las partes fundamentales en la entrada de datos. Para ilustrar el método, presentamos los ejemplos de las Figs. 3.3 y 3.4.



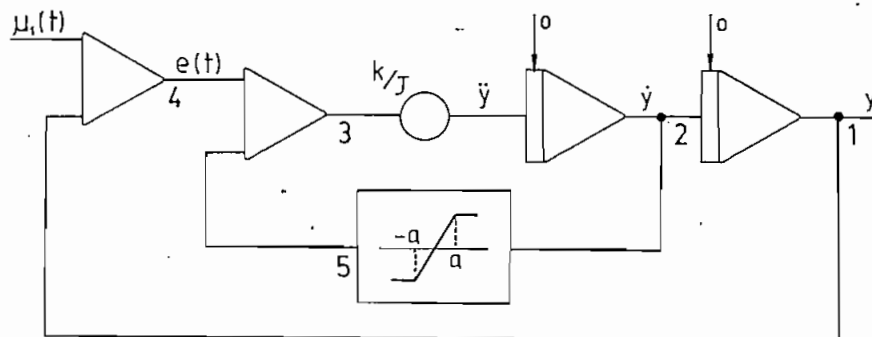
(a)

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Entradas --->

(b)

Fig. 3.3 a) Diagrama de simulación para generar la espiral de Arquímedes.  
b) Matriz de incidencia.



(a)

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |

Entradas --->

(b)

Fig. 3.4 a) Sistema de segundo orden con ganancia de realimentación no lineal.  
b) Matriz de incidencia.

3. Definición de los módulos. Se comienza definiendo los módulos que tienen más de una entrada y luego los de una sola entrada. En la pantalla se presentan los menús e indicaciones necesarias para una correcta definición.

Todos los datos referidos a los módulos de una o varias entradas que son solicitados por el programa, están de acuerdo con las definiciones de los mismos dadas en el capítulo 2, Figs. 2.1 a 2.22.

Los datos de los módulos presentes en el diagrama se almacenan en un arreglo de dos dimensiones de  $n_{mi}$  por 8 llamado D, donde  $n_{mi}$  es la variable entera que almacena el número de unos que tiene la MI sin incluir su última fila, es decir,  $n_{mi}$  es el número de ramales que tiene el diagrama de simulación. La estructura de una fila  $i$  de la matriz D es la siguiente:

|                             |                          |
|-----------------------------|--------------------------|
| $d_{i,1}$ = desde           | $d_{i,5}$ = referencia 1 |
| $d_{i,2}$ = hasta           | $d_{i,6}$ = referencia 2 |
| $d_{i,3}$ = código          | $d_{i,7}$ = referencia 3 |
| $d_{i,4}$ = potenc. entrada | $d_{i,8}$ = referencia 4 |

El primer término indica cual es el número de la salida donde inicia el ramal; el segundo término indica cual es el número de la salida del módulo o lo que es lo mismo el fin del ramal, el tercer dato lleva el código del módulo presente en el ramal, el valor de este código es el indicado en la Tabla 3.1, el cuarto dato contiene el valor del potenciómetro que afecta a la señal de entrada al módulo. Las referencias correspondientes al resto de datos tienen sentido para ciertos módulos. La Tabla 3.2 muestra lo que significan estas referencias.

| MODULO                    | REFERENCIA 1       | REFERENCIA 2      | REFERENCIA 3 | REFERENCIA 4 |
|---------------------------|--------------------|-------------------|--------------|--------------|
| INTEGRADOR                | Condición inicial  |                   |              |              |
| LIMITADOR                 | Máximo             | Mínimo            |              |              |
| COMPARADOR                | Referencia         | Máximo            | Mínimo       |              |
| COMPARADOR CON HISTERESIS | Referencia mínima  | Referencia máxima | Máximo       | Mínimo       |
| ZONA MUERTA               | Amplitud de la ZM  |                   |              |              |
| ALINEALIDAD DE JUEGO      | Amplitud del juego | Máximo            | Mínimo       |              |
| RETARDO EN EL TIEMPO      | Tiempo de retardo  |                   |              |              |
| MONOESTABLE               | Tiempo del pulso   |                   |              |              |
| DIVISOR                   | l=Dividendo        |                   |              |              |
| SW. ANALOGO               | l=Señal de control |                   |              |              |
| BIESTABLE                 | l=Set              |                   |              |              |

Tabla 3.2. Significado de los datos referenciales de la matriz D.

Los datos referenciales de la matriz D para los módulos no indicados en la Tabla 3.2 no tienen ningún significado y son iguales a cero.

4. Matriz PM. Con la matriz D construida, se crea otro arreglo de dimensión m por 4 y de nombre PM que nos servirá durante la evaluación de los datos y que contiene en su fila i cierta información sobre el módulo cuya salida es i. Inicialmente, la única información que introducimos a la matriz PM es la referente a los integradores presentes en el diagrama de simulación. Así, la estructura de una fila i de la matriz PM donde i es la salida de un integrador es la siguiente:

$pm_{i,1} = i$   
 $pm_{i,2} = pm_{i,4} =$  condición inicial del integrador  
 $pm_{i,3} =$  número asignado al estado correspondiente a la salida i.

Cuando tratemos sobre la estructura del programa de solución, explicaremos en detalle la composición de esta matriz. Adicionalmente, el número de integradores presentes en el diagrama se almacena en la variable entera n.

5. Definición de los subíndices de las entradas. Analizando la fila m+1 de la MI se puede determinar el número de entradas externas (dato que se almacena en la variable entera l) y los números correspondientes a las salidas de los módulos a los cuales van aplicadas. La definición de los subíndices de las entradas consiste en determinar el subíndice i de una entrada que va aplicada a un módulo cuya salida es j. Los datos se almacenan en un arreglo de enteros 1 por 2 denominado ME. La estructura de una fila i de la matriz ME es la siguiente:

$me_{i,1} =$  Subíndice de la entrada  
 $me_{i,2} =$  Número de la salida del módulo.

6. Datos de tiempo. Estos datos son requeridos al usuario con las respectivas indicaciones y se almacenan en las siguientes variables:



t0 = tiempo de iniciación del cálculo  
tf = tiempo de finalización del cálculo  
h = intervalo de tiempo entre cálculos  
hi = intervalo de tiempo entre resultados impresos.

Terminada la introducción de datos, el programa lee el nombre del archivo de datos que está almacenado en el archivo NOMBRE.HLP, creado en el directorio corriente y crea temporalmente un archivo con este nombre y con la extensión .DAT. La estructura del archivo de datos es la siguiente:

```
n  l  m  n1m1  ti  tf  h  hi
d1,1  d1,2.....d1,e
.      .      .
.      .      .
dn1m1,1 dn1m1,2.....dn1m1,e
me1,1  me1,2
.      .
.      .
me1,1  me1,2
pm1,1.....pm1,4
.      .
.      .
pmm,1.....pmm,4
```

Finalmente, se cierran todos los archivos y se regresa al programa principal en donde se pedirá por la programación de las funciones de entrada (en caso de que existan entradas) y se llamará al programa de solución, el cual lo describiremos en el siguiente punto.

### 3.2 PROGRAMA DE SIMULACION

Veamos inicialmente como aplicar el algoritmo de Runge-Kutta para la evaluación del diagrama de simulación. Hemos llamado n a la variable que contiene el número de integradores presentes en el diagrama, n corresponde también al número de estados del sistema o al orden de la mayor derivada de la ecuación diferencial que representa al mismo. Si a cada salida de un integrador le asignamos una variable de estado  $x_1(t)$ , y llamamos X al vector columna de dimensiones nx1 que contiene a todas las variables de estado (vector de estado), podemos es-

cribir la ecuación

$$dX/dt = f(t, X) \quad (3.1)$$

donde  $f$  representa el conjunto de funciones que determinan los valores de las entradas de los integradores en función de sus salidas y de las entradas externas del sistema. Notemos que  $f$  viene dada en forma implícita dentro del diagrama de simulación, es decir, las relaciones para las entradas de los integradores se encuentran establecidas en forma gráfica. Llamemos  $X_k = X(t_k)$  al vector de estado evaluado en  $t=t_k$ . El algoritmo de Runge-Kutta permite evaluar el vector de estado en  $t=t_{k+1}=t_k+h$  con las siguientes relaciones

$$\begin{aligned} X_{k+1} &= X_k + (K_1 + 2K_2 + 2K_3 + K_4) / 6 \\ K_1 &= hf(t_k, X_k) \\ K_2 &= hf(t_k + h/2, X_k + K_1/2) \\ K_3 &= hf(t_k + h/2, X_k + K_2/2) \\ K_4 &= hf(t_k + h, X_k + K_3) \end{aligned} \quad (3.2)$$

notemos que para determinar  $K_1$  basta evaluar el valor de las entradas de los integradores a partir del valor de sus salidas y el valor de las entradas externas y multiplicar el resultado por  $h$ . De igual forma, para evaluar  $K_2$  debemos calcular el valor de las entradas externas en  $t=t_k+h/2$ , evaluar el valor de las entradas de los integradores asignando al vector de estado el valor de  $X_k + K_1/2$  y multiplicar el resultado por  $h$ . Procedimientos similares permiten calcular  $K_3$  y  $K_4$ . En resumen, el algoritmo para calcular el valor de las salidas del diagrama de simulación en un tiempo  $t_{k+1}$  a partir de sus valores conocidos en  $t_k$  es:

1. Leemos el valor de las entradas externas del sistema (si es que existen) en  $t=t_k$ .
2. A partir de los valores conocidos de  $X_k$  y de las entradas externas, calculamos el valor de todas las salidas del diagrama en  $t_k$ .

3. Sabemos que las entradas de los integradores corresponden a salidas de otros módulos (no puede haber una entrada externa aplicada directamente a un integrador), por tanto, podemos asignar al vector  $K_1$  los valores de las entradas de los integradores multiplicadas por  $h$ .

4. Leemos el valor de las entradas externas del sistema en  $t=t_k+h/2$ .

5. Asignamos a las salidas de los integradores los valores correspondientes de  $x_k+k_1/2$ .

6. A partir de estos valores calculamos el valor de todas las salidas del diagrama.

7. Asignamos al vector  $K_2$  los valores de las entradas de los integradores multiplicadas por  $h$ .

8. Asignamos a las salidas de los integradores los valores correspondientes a  $x_k+k_2/2$ .

9. Calculamos el valor de todas las salidas del diagrama en base a estos nuevos valores.

10. Asignamos al vector  $K_3$  los valores de las entradas de los integradores multiplicadas por  $h$ .

11. Leemos el valor de las entradas externas del sistema en  $t=t_k+h$ .

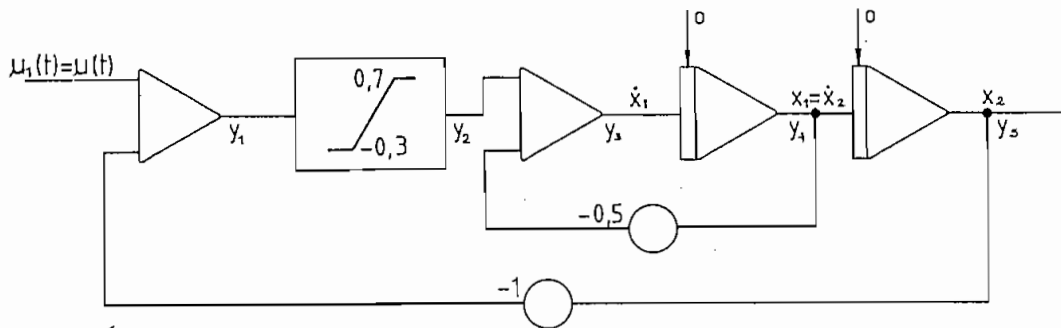
12. Asignamos a las salidas de los integradores los valores correspondientes de  $x_k+k_3$ .

13. Para estos nuevos valores, calculamos el valor de todas las salidas del diagrama.

14. Asignamos al vector  $K_4$  los valores de las entradas de los integradores multiplicadas por  $h$ .

15. Calculamos las salidas de los integradores en  $t_{k+1}$  con la ecuación  $X_{k+1} = X_k + (K_1 + 2K_2 + 2K_3 + K_4) / 6$ .

Para ilustrar el método, vamos a calcular el valor de las salidas del diagrama de simulación de la Fig. 3.5 para  $t=h$  a partir de las condiciones iniciales de los integradores y con una función escalón unitario en la entrada.



$$\dot{X} = AX + Bu$$

Fig. 3.5 Sistema no lineal de segundo orden.

Datos:

$$h = 0.1$$

$$t_0 = 0$$

$$X_0 = X(0) = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

1. Leemos el valor de la entrada  $u_1(t)$  para  $t=0$ .

$$u_1(0) = 1$$

2. Calculamos el valor de todas las salidas del diagrama en  $t=0$ .

$$y_1(0) = u_1(0) - x_2(0) = 1 - 0 = 1$$

$$y_2(0) = 0.7$$

$$y_3(0) = y_2(0) - 0.5x_1(0) = 0.7 - 0 = 0.7$$

$$y_4(0) = x_1(0) = 0$$

$$y_5(0) = x_2(0) = 0$$

Por efecto del limitador

3. Asignamos al vector  $K_1$  los valores de las entradas de los integradores multiplicadas por  $h$ .

$$K_1 = h \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = 0.1 \begin{bmatrix} 0.7 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0.07 \\ 0.0 \end{bmatrix}$$

4. Leemos el valor de la entrada para  $t=h/2$ .

$$u_1(h/2) = 1$$

5. Asignamos a las salidas de los integradores el valor de  $X_0 + K_1 / 2$ .

$$X = X_0 + K_1 / 2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.5 \begin{bmatrix} 0.07 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0.035 \\ 0.0 \end{bmatrix}$$

6. Calculamos el valor de las salidas del diagrama.

$$\begin{aligned}y_1 &= u_1(h/2) - x_2 = 1 - 0 = 1 \\y_2 &= 0.7 \\y_3 &= y_2 - 0.5x_1 = 0.7 - 0.0175 = 0.6825 \\y_4 &= x_1 = 0.035 \\y_5 &= x_2 = 0\end{aligned}$$

7. Asignamos a  $K_2$  los valores de las entradas de los integradores multiplicadas por  $h$ .

$$K_2 = h \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = 0.1 \begin{bmatrix} 0.6825 \\ 0.035 \end{bmatrix} = \begin{bmatrix} 0.06825 \\ 0.0035 \end{bmatrix}$$

8. Asignamos a las salidas de los integradores el valor de  $X_0 + K_2/2$ .

$$X = X_0 + K_2/2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.5 \begin{bmatrix} 0.06825 \\ 0.0035 \end{bmatrix} = \begin{bmatrix} 0.034125 \\ 0.00175 \end{bmatrix}$$

9. Calculamos el valor de todas las salidas del diagrama.

$$\begin{aligned}y_1 &= u_1(h/2) - x_2 = 1 - 0.00175 = 0.99825 \\y_2 &= 0.7 \\y_3 &= y_2 - 0.5x_1 = 0.7 - 0.0170625 = 0.6829375 \\y_4 &= x_1 = 0.034125 \\y_5 &= x_2 = 0.00175\end{aligned}$$

10. Asignamos a  $K_3$  los valores de las entradas de los integradores multiplicadas por  $h$ .

$$K_3 = h \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = 0.1 \begin{bmatrix} 0.6829375 \\ 0.034125 \end{bmatrix} = \begin{bmatrix} 0.06829375 \\ 0.0034125 \end{bmatrix}$$

11. Leemos el valor de la entrada en  $t=h$ .

$$u_1(h) = 1$$

12. Asignamos a las salidas de los integradores el valor de  $X_0 + K_3$ .

$$X = X_0 + K_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.06829375 \\ 0.0034125 \end{bmatrix} = \begin{bmatrix} 0.06829375 \\ 0.0034125 \end{bmatrix}$$

13. Calculamos el valor de todas las salidas del diagrama.

$$\begin{aligned}y_1 &= u_1(h) - x_2 = 1 - 0.0034125 \\y_2 &= 0.7 \\y_3 &= y_2 - 0.5x_1 = 0.7 - 0.034146875 = 0.665853125 \\y_4 &= x_1 = 0.06829375 \\y_5 &= x_2 = 0.0034125\end{aligned}$$

14. Asignamos a  $K_4$  los valores de las salidas de los integradores multiplicadas por  $h$ .

$$K_4 = h \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = 0.1 \begin{bmatrix} 0.665853125 \\ 0.06829375 \end{bmatrix} = \begin{bmatrix} 0.0665853125 \\ 0.006829375 \end{bmatrix}$$

15. Calculamos las salidas de los integradores en  $t=h$ .

$$X(h) = X_1 = X(0) + (K_1 + 2K_2 + 2K_3 + K_4) / 6$$

$$X(h) = \begin{bmatrix} 0.06828 \\ 0.003442 \end{bmatrix}$$

Luego los valores de las salidas del diagrama en  $t=h$  son

$$\begin{aligned} y_1(h) &= u_1(h) - x_2(h) = 1 - 0.02859 = 0.97141 \\ y_2(h) &= 0.7 \\ y_3(h) &= y_2(h) - 0.5x_1(h) = 0.7 - 0.03414 = 0.66586 \\ y_4(h) &= x_1(h) = 0.06828 \\ y_5(h) &= x_2(h) = 0.003442 \end{aligned}$$

La evaluación del diagrama de simulación se ejecuta en base al algoritmo descrito con la ayuda del programa SOL.EXE que lo describiremos a continuación.

El programa SOL.EXE está desarrollado en lenguaje GW-BASIC y luego compilado. El llamado al mismo es directo desde el programa principal PRES.BAS, una vez que la estructura del diagrama de simulación y sus entradas externas hayan sido definidas. El programa presenta en la pantalla la indicación de que se está calculando las salidas y procede a la lectura de los datos contenidos en el archivo de extensión .DAT. Inmediatamente después, se borra del disco el archivo de datos y se crea un archivo ASCII de extensión .PRN que será el que en definitiva contenga los datos y resultados del problema. Se procede inicialmente a guardar los datos del problema en el archivo PRN. Adicionalmente se calcula en número de intervalos de cálculo y el número de intervalos de impresión que se los almacena en las variables enteras  $nic$  y  $nii$  respectivamente.

A partir de este momento, la evaluación del diagrama de simulación se fundamenta en el tratamiento de los valores contenidos en tres arreglos. El primero es el arreglo llamado PM,

ya definido en el programa ENT.EXE pero que para el efecto sus dimensiones han sido ampliadas a  $m \times 8$ , el segundo y tercer arreglos corresponden a la matriz de datos D y la matriz de entradas ME respectivamente. La matriz PM contiene en su fila  $i$  toda la información necesaria sobre el estado de la salida  $i$  del diagrama de simulación. Así, el valor de  $pm_{i,1}$  será uno si es que la salida  $i$  del diagrama de simulación ya ha sido calculada en el instante de tiempo en que nos encontremos trabajando y cero en caso contrario,  $pm_{i,2}$  contiene el valor numérico de la salida  $i$  al final de un cálculo; el resto de datos de la fila  $i$  tienen diferente significado dependiendo del tipo de módulo cuya salida sea  $i$ .

El programa SOL.EXE tiene dos partes. La primera que se describe en el diagrama de flujo de la Fig. 3.6 que corresponde al tratamiento de los integradores y la segunda que se muestra en el diagrama de flujo de la Fig. 3.7 correspondiente a una subrutina que calcula el valor de todas las salidas del diagrama en el instante  $t=t_k$  a partir de los valores conocidos de las salidas de los integradores en el mismo instante.

La subrutina de cálculo de las salidas de los integradores ejecuta el algoritmo descrito anteriormente y almacena los valores de tiempo, entradas externas y salidas en el archivo PRN. La estructura de la fila  $i$  del arreglo PM, donde  $i$  corresponde a la salida de un integrador es la siguiente:

$pm_{i,1} = 1$ , se asume que la salida de un integrador es siempre conocida en este programa.  
 $pm_{i,2}$  = valor de la salida de un integrador que varía al pasar de  $t_k$  a  $t_{k+1}$ .  
 $pm_{i,3}$  = número de estado asignado a la salida del integrador.  
 $pm_{i,4}$  = valor fijo de la salida del integrador en  $t_k$ .  
 $pm_{i,5}$  = valor de la entrada del integrador (derivada).  
 $pm_{i,6}$  = valor correspondiente del vector  $K_1$ .  
 $pm_{i,7}$  = valor correspondiente del vector  $K_2$ .  
 $pm_{i,8}$  = valor correspondiente del vector  $K_3$ .

La columna dos del arreglo PM contiene los valores de  $X_k$ ,  $X_k+K_1/2$ ,  $X_k+K_2/2$  y  $X_k+K_3$  necesarios para el cálculo de  $X_{k+1}$ . La salida de un integrador tiene asignada en su fila correspon-

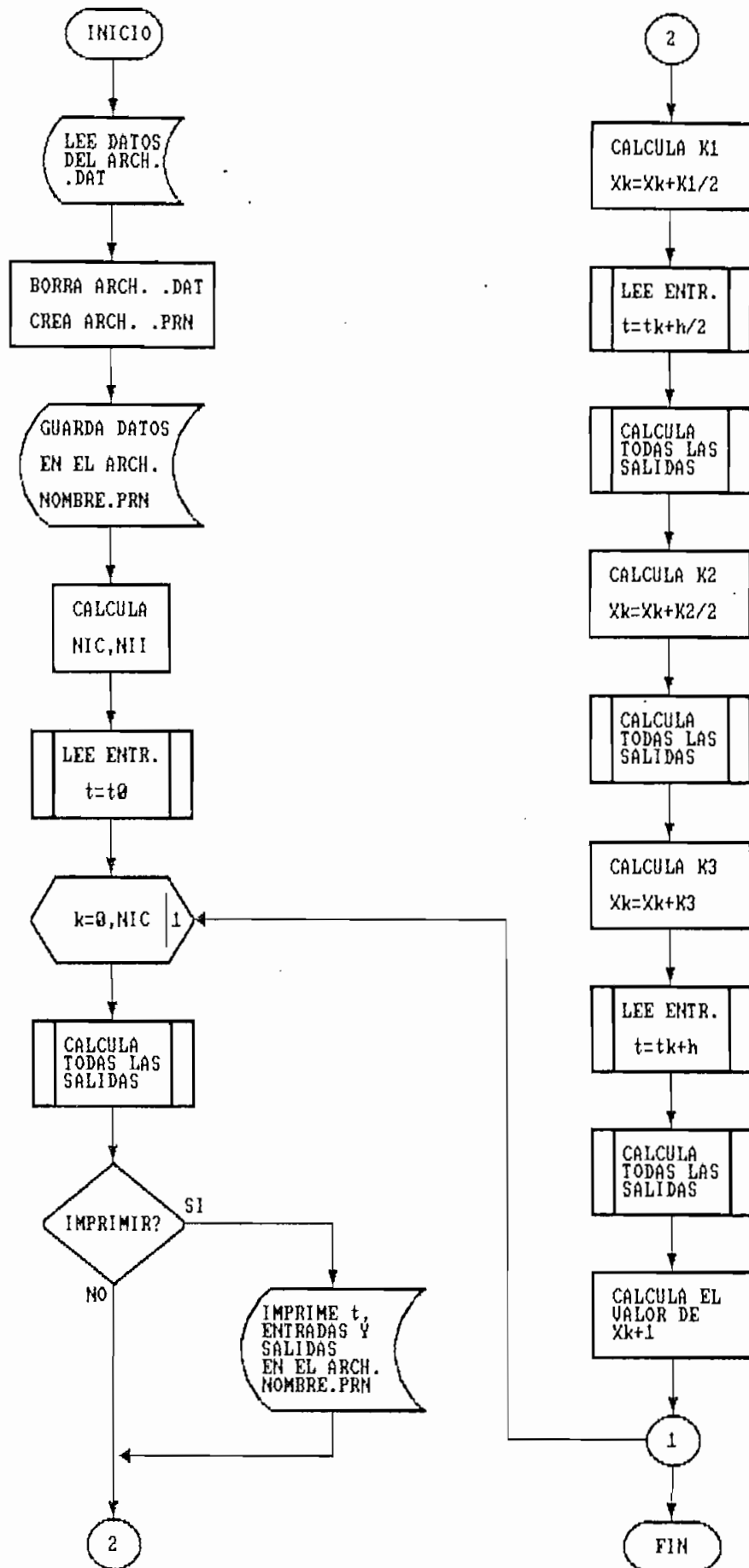


FIG. 3.6 DIAGRAMA DE FLUJO DEL PROGRAMA SOL.EXE



diente y en la columna tres un número diferente de cero. Los valores de  $K_4/h$  se tienen almacenados en la columna cinco de PM al terminar el último paso de integración.

La subrutina que calcula el valor de todas las salidas del diagrama de simulación a partir de los valores conocidos de las salidas de los integradores (Fig. 3.7), trabaja de la siguiente manera:

realiza un barrido de todas las salidas del diagrama;

si una salida  $i$  ya ha sido calculada ( $pm_{i,1}=1$ ), se continúa con la siguiente;

si una salida no ha sido calculada ( $pm_{i,1}=0$ ), se investiga si es que las entradas al módulo son conocidas;

si todas las entradas son conocidas, se evalúa la salida del módulo en base al tipo de este, según el diagrama de flujo correspondiente (Figs. 2.1 a 2.22), teniendo como datos los valores de las entradas al módulo y los valores referenciales contenidos en la matriz D. El resultado de la salida  $i$  calculada se almacena en  $pm_{i,2}$ , se asigna uno a  $pm_{i,1}$  y si el valor calculado corresponde a la entrada de un integrador se lo asigna a  $pm_{i,3}$ ;

si las entradas al módulo no son conocidas se continúa con una nueva salida;

el proceso termina cuando el número de salidas calculadas es igual al número total de salidas del diagrama o cuando durante una nueva corrida no se ha podido evaluar una salida adicional. Este último caso se presenta cuando la estructura del diagrama de simulación lleva a relaciones trascendentes entre dos o más variables, de ser así, el programa es abortado con la respectiva indicación en la pantalla.

Los diagramas de flujo mostrados en las Figs. 2.1 a 2.22

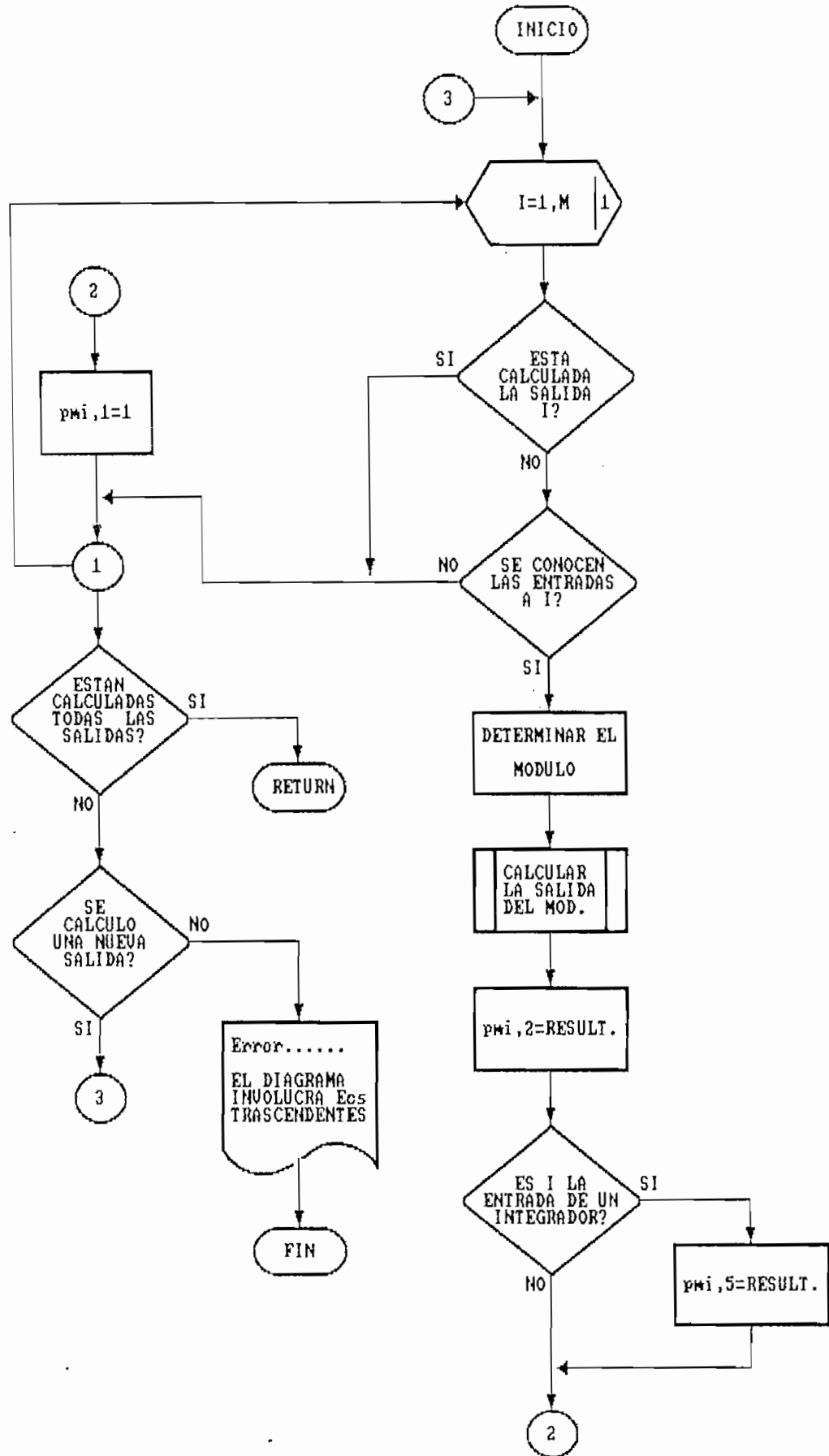


FIG. 3.7 DIAGRAMA DE FLUJO DE LA SUBROUTINA DE CALCULO DE TODAS LAS SALIDAS

muestran la forma en que se evalúan los diferentes módulos en el computador. Algunos módulos son fáciles de calcular pero otros presentan cierta dificultad, para estos últimos vamos a aclarar la forma de tratarlos.

La salida de una alinealidad comparador Fig. 2.3, puede tomar cualquier valor cuando su entrada es igual al valor de referencia. En el computador digital, la decisión sobre el valor que tome la salida se lo hace generando un número aleatorio de valor 0 o 1; si el aleatorio es 1, se toma el valor máximo en la salida caso contrario se toma el mínimo.

Para la alinealidad comparador con histéresis Fig. 2.4, si estamos en el inicio del cálculo  $t=t_0$  y la entrada cae dentro de la ventana de histéresis, la salida puede tomar cualesquiera de los dos valores permisibles. La decisión sobre el valor que se tomará en la salida se lo hace en forma idéntica al caso del comparador. Cuando estemos evaluando el valor de la salida en  $t=t_k$  y el valor de la entrada cae dentro de la ventana de histéresis, la salida debe permanecer sin alterarse para lo cual debemos disponer del valor de la misma en  $t=t_{k-1}$ . Este dato se almacena en la  $pm_{i,j}$  donde  $i$  es el número asignado a la salida del módulo.

Lo mismo que sucede para el comparador con histéresis puede darse para la alinealidad de juego Fig. 2.9. La forma de afrontar el problema es idéntica en ambos casos.

Para la alinealidad raíz cuadrada Fig. 2.11, si la entrada es un número negativo, se presenta en la pantalla una indicación de error y el proceso es abortado.

De igual forma que en el caso anterior, para la alinealidad logaritmo natural Fig. 2.12, si la entrada es un número negativo o cero el proceso es abortado con la respectiva indicación de error.

Para el módulo de división Fig. 2.19, si el dividendo es

igual a cero, el proceso es abandonado indicando el error.

Para las compuertas lógicas y el biestable SR Figs. 2.18 y 2.19 respectivamente, se analiza que todas las entradas tengan los valores adecuados (0 a 0.2 para cero lógico y 0.8 a 1 para uno lógico), de no ser así, el programa es abandonado con la respectiva indicación de error. Adicionalmente, el programa es también abortado cuando se produce la condición prohibida en el biestable ( $S=1L$  y  $R=1L$ ). En  $pm_{1,7}$  se almacena el resultado anterior del biestable cuya salida es  $i$ . La decisión sobre el valor inicial del biestable cuando se arranca con  $S=0$  y  $R=0$  se la toma en base a la generación de un aleatorio de valor 0 o 1.

En el caso del monoestable Fig. 2.22, la señal de entrada (disparo) debe tener niveles lógicos y no puede estar en cero cuando se termine el período del monoestable, de producirse cualesquiera de estos dos errores durante la ejecución el programa termina con las indicaciones respectivas de error. El monoestable produce en su salida un pulso de amplitud  $T$  a partir del momento de disparo. El tiempo  $T$  se mide en base al número de intervalos de cálculo transcurridos desde el disparo, este número se almacena en  $pm_{1,6}$ . El estado anterior del monoestable cuya salida es  $i$  se almacena en  $pm_{1,7}$ .

La señal de control del switch análogo Fig. 2.21, debe ser digital. Si los valores de la entrada de control están fuera de los niveles lógicos el programa es abortado indicando el error respectivo.

La forma de tratar el retardo en el tiempo Fig. 2.22 es en base a la creación de un vector de retardo llamado  $RT_i$  donde  $i$  es el número asignado a la salida del monoestable. El vector  $RT$  contiene los valores de la entrada al módulo durante el intervalo de tiempo de retardo.

Terminada la evaluación del diagrama de simulación se borra el archivo que contiene el valor de las entradas (archivo .ENT), se cierra el archivo de datos y resultados y se regresa

al menú principal.

La estructura final del archivo de datos y resultados es la siguiente:

```
l
m
t0
tf
h
h^

d1,1      d1,2.....d1,e
.
.
dn1m1,1  dn1m1,2.....dn1m1,e

me1,1     me1,2
.
.
me1,1     me1,2

t0  u1(t0).....u1(t0)  y1(t0).....ym(t0)
t1  u1(t1).....u1(t1)  y1(t1).....ym(t1)
.
.
.
tf  u1(tf).....u1(tf)  y1(tf).....ym(tf)
```

### 3.3 PRESENTACION DE RESULTADOS

El archivo del tipo .PRN creado en la ejecución del programa SOL.EXE puede ser analizado mediante la opción 3. del menú principal. Al escoger esta opción se corre el archivo de proceso por lotes RESULTAD.BAT. Este archivo permite el acceso a la hoja electrónica LOTUS 123, es en esta hoja donde se analizarán los resultados. Se escoge LOTUS para la presentación de resultados debido a la versatilidad que este programa presenta para trabajar con tablas de números, producir gráficos en pantalla e impresos, imprimir rangos de valores, etc. a más de que presenta la facilidad de acoplarse a otro tipo de programas (BASIC en nuestro caso). Seria muy difícil superar las cualidades de LOTUS desarrollando un programa de presentación de resultados en un lenguaje de alto nivel como BASIC, PASCAL o ADA.

Para revisar un archivo de datos, directamente en la hoja es carga el programa MENU.WK1. La hoja MENU.WK1 contiene las instrucciones que generan los macro comandos para facilitar el tratamiento de las tablas de datos y resultados. Inmediatamente después de que se carga el programa MENU, se corre el macro de nombre \0 el cual presenta en la línea de estado de la hoja un menú propio del programa. Las opciones disponibles en este menú de resultados son:

CHDIR permite cambiar el directorio corriente. Esta opción se la utiliza para indicar a la hoja cual es el directorio en el cual se encuentra el archivo de datos y resultados.

REVISAR carga el archivo de datos y resultados (.PRN) en la hoja y los etiqueta para facilitar la revisión.

IMPRIMIR para imprimir un rango determinado de la hoja. Al escoger esta opción, el usuario deberá introducir los datos del rango y los márgenes para la impresión.

GRAFICAR con esta opción se puede crear gráficos en la pantalla. Los datos que son requeridos por la hoja son: el rango de valores del eje x, hasta seis rangos de valores para el eje y, título, subtítulo, leyendas para los ejes y leyendas para los diferentes rangos de y. El resto de opciones se escoge automáticamente con el macro correspondiente. El gráfico resultante es del tipo XY con reticulado y ajuste automático de escalas.

PGRAFIC permite almacenar un gráfico para impresión. Al pedir por esta opción, el programa solicita el nombre del archivo donde se almacenará el gráfico, este archivo es creado por la hoja con la extensión .PIC. El archivo .PIC se puede imprimir posteriormente con el programa PrintGraph de LOTUS.

ALMACENAR se escoge para almacenar en el disco la hoja de trabajo como un archivo de LOTUS (.WK1). El macro dedicado a este trabajo solicitará el nombre del archivo con el cual se

almacena la hoja.

OTROS es el nombre de un submenú que contiene algunas opciones adicionales.

OTROS/ CAMBIAR permite cambiar los datos de las constantes involucradas en el diagrama de simulación, el cursor se mueve únicamente por las casillas que contienen datos que pueden ser cambiados. Los datos factibles de cambiar son: tiempo inicial de cálculo, tiempo final de cálculo, intervalo entre cálculos, intervalo entre resultados impresos, valores de los potenciómetros de entrada y valores de las referencias de los diferentes módulos involucrados (Tabla 3.2).

OTROS/ ALMACENAR los datos cambiados dentro de la hoja deben ser almacenados en el disco para luego evaluarse con la opción 2. del menú principal del programa, esto se puede hacer con este comando del menú de resultados. Luego de escoger esta opción, el programa pide por el nombre del archivo donde se almacenarán los nuevos datos. El archivo de datos creado por la hoja es de extensión .PRN y su estructura es idéntica a la del archivo .DAT creado por el programa ENT.EXE.

OTROS/ MENU permite el regreso al menú inicial de resultados.

OTROS/ SALIR para salir del menú de resultados y entrar al menú de LOTUS. Esta opción es requerida cuando se desea tener una mayor capacidad para el análisis de resultados mediante la utilización de las opciones adicionales de las que dispone LOTUS.

OTROS/ FIN permite salir de la hoja de trabajo para imprimir gráficos o retornar al menú principal.

Adicionalmente, presionando la tecla Esc se puede salir del menú de resultados y trabajar en la hoja con los comandos de LOTUS. El retorno al menú de resultados se lo hace ejecu-

tando el macro \A para lo cual se debe presionar simultáneamente las teclas Alt y A. Cualquier error cometido dentro de la ejecución de un macro puede ser eliminado presionando simultáneamente las teclas Ctrl y Break y luego la tecla Esc.

Notemos que las opciones del menú de resultados cubren la mayor parte de operaciones requeridas para el análisis de los mismos, sin embargo, si es que se requiere algún tratamiento especial de los resultados, se puede utilizar las diferentes opciones de la hoja para lo cual se deberá conocer con mayor profundidad al programa LOTUS. En el Anexo B del presente trabajo se presenta la descripción de algunas operaciones adicionales que se pueden realizar en la hoja.

Al salir de la hoja de trabajo, LOTUS presenta tres opciones: 1-2-3 permite retornar a la hoja, PrintGraph para imprimir gráficos y Exit para salir al menú principal.

#### REFERENCIAS

1. NCR PERSONAL COMPUTER, *GWT™-Basic*. NCR Corporation, 1984.
2. LOTUS DEVELOPMENT CORPORATION, *1-2-3*. Lotus Development Corporation, 1985.



CAPITULO IV

## CAPITULO IV

### EJEMPLOS, CONCLUSIONES, RECOMENDACIONES

#### 4.1 EJEMPLOS DE SIMULACION

En este punto, presentaremos algunos ejemplos con el fin de demostrar el poder del programa de simulación. Así, encontraremos las respuestas de sistemas conocidos todos sus parámetros sin involucrarnos en el problema de diseño. Para cada uno de los ejemplos exhibiremos la descripción del problema, el diagrama de bloques, el diagrama de simulación y los resultados. No presentaremos la forma de introducir los datos por cuanto en el Anexo A del presente trabajo se detallan estas operaciones.

##### 4.1.1 Ejemplo 1. Sistema lineal con retardo.

La red principal de control de una planta de energía nuclear incluye un atraso de tiempo debido al tiempo necesario para trasportar el fluido desde el reactor hasta el punto de medición Fig. 4.1.

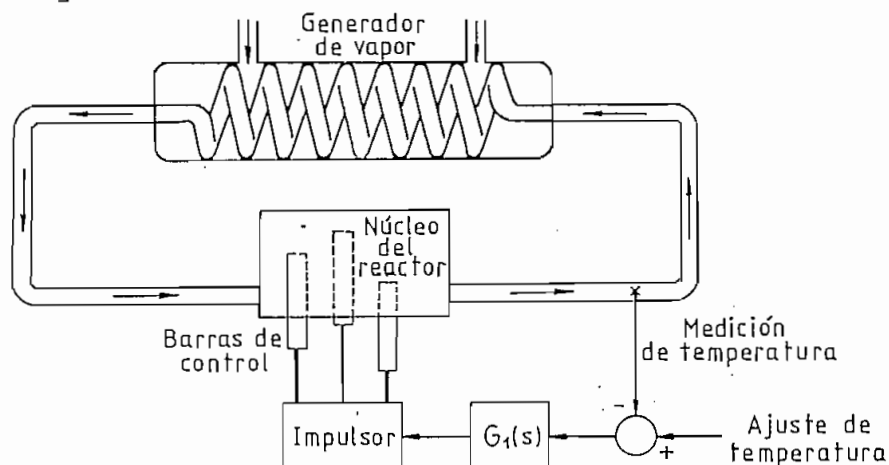


Fig. 4.1 Planta de energía nuclear.

El sistema contiene un controlador proporcional integral cuya

función de transferencia es

$$G_1(s) = 0.5 + 1.5/s$$

la función de transferencia del reactor y del atraso de tiempo es

$$G(s) = e^{-0.5s} / (0.25s + 1)$$

Averiguemos cual es el comportamiento del sistema ante una perturbación escalón unitario en el ajuste de temperatura.

Los diagramas de bloques y de simulación para el sistema se muestran en las Figs. 4.2 y 4.3 respectivamente.

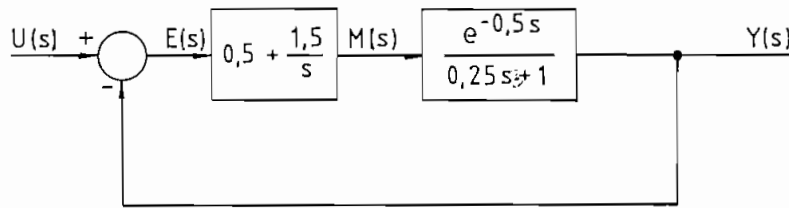


Fig. 4.2 Diagrama de bloques de la planta de energía nuclear.

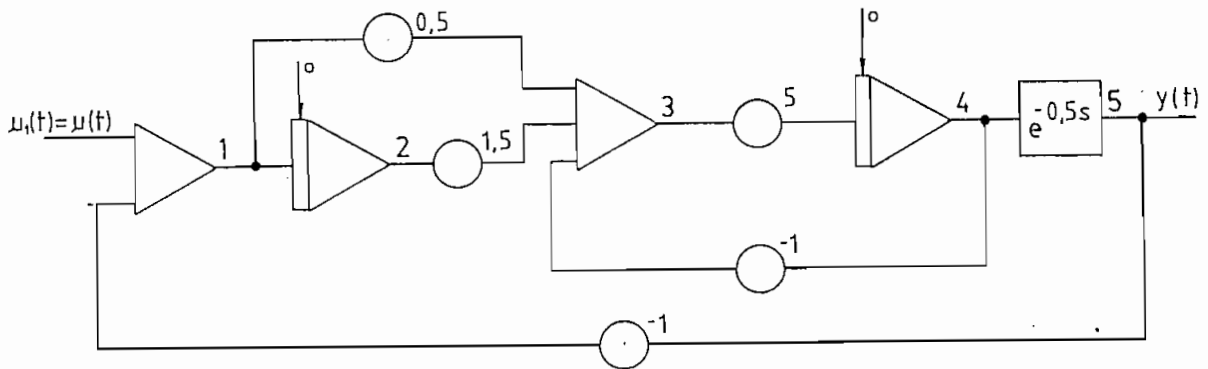


Fig. 4.3 Diagrama de simulación de la planta de energía nuclear.

El diagrama de simulación está desarrollado de forma que permite observar el comportamiento del error del sistema  $y_1$ , de su salida retardada  $y_5$  y de la salida sin retardo  $y_4$ .

Los siguientes datos adicionales son requeridos para la simulación

$t_0 = 0$   
 $t_f = 6$   
 $h = 0.1$   
Intervalo entre impresos = 0.1

Los resultados numéricos obtenidos se muestran en la tabla 4.1. En la Fig. 4.4 se muestran los gráficos de la salida, la salida sin retardo y el error en función del tiempo.

A partir de los resultados numéricos podemos hacer las siguientes comprobaciones.

Se ve claramente el efecto del retardo en la salida del sistema. Se cumple que  $y_3(t) = y_4(t - 0.5)$ .

La salida  $y_1(t)$  del diagrama de simulación (error del sistema) es igual a la entrada menos la salida.  $y_1(t) = 1 - y_3(t)$ .

Se cumple que  $y_3(t) = 0.5y_1(t) + 1.5y_2(t) - y_4(t)$ .

Para comprobar el funcionamiento de la integración, calculemos en forma analítica la respuesta no retardada del sistema para  $0 < t < 0.5$ . Para este intervalo de tiempo no existe efecto de la salida sobre el error del sistema de forma que este se comporta como un sistema de lazo abierto. El error es igual a la entrada  $u(t) = 1$ . En el diagrama de bloques de la Fig. 4.2 se cumplen las siguientes relaciones

$$\begin{aligned} e(t) &= 1 \\ m(t) &= 0.5e(t) + 1.5 \int_0^t e(t) dt = 0.5 + 1.5t \\ 0.2d\bar{y}/dt + \bar{y} &= m(t); \quad \bar{y}(0) = 0 \end{aligned} \quad (4.1)$$

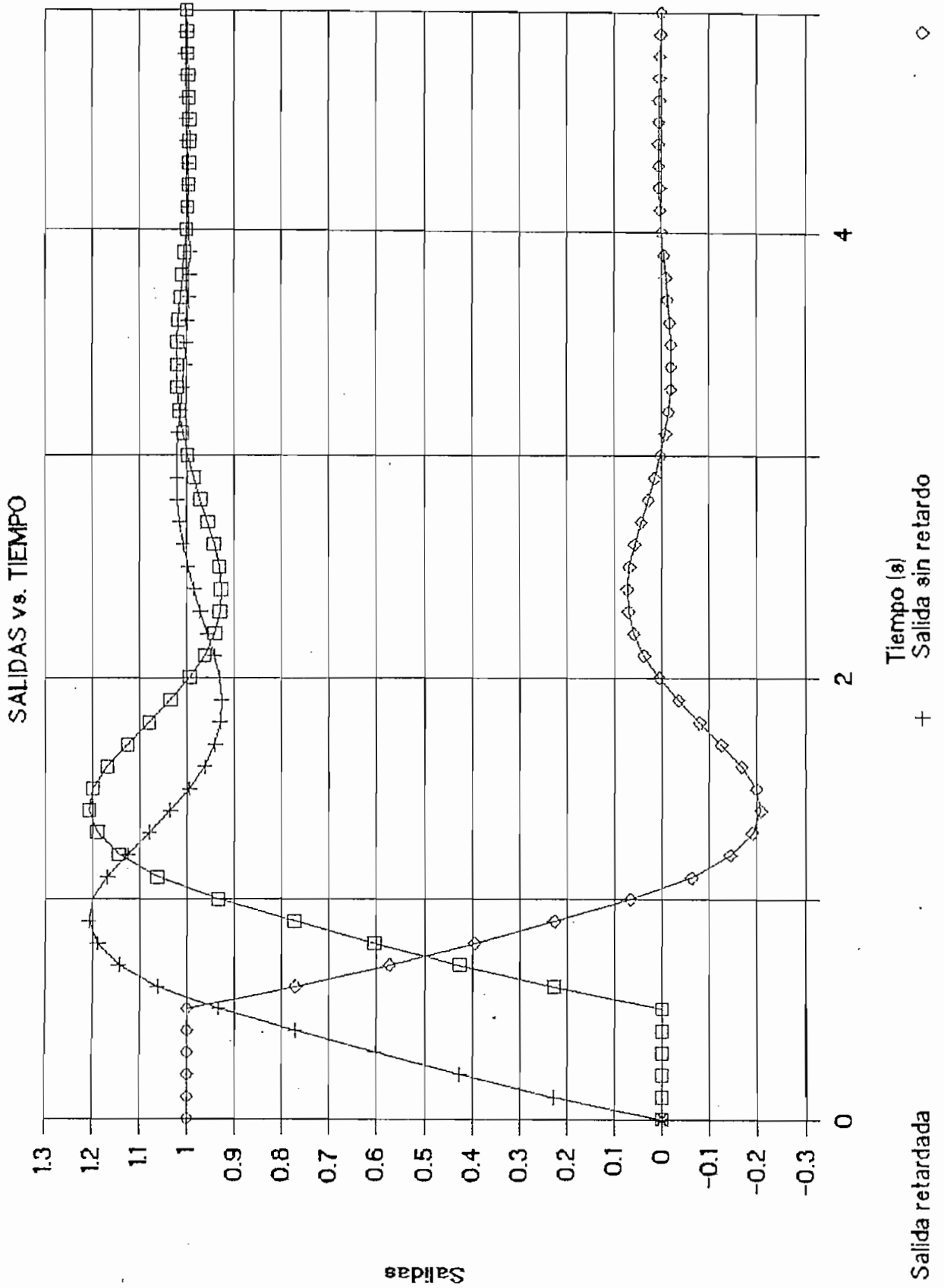
donde  $\bar{y}(t)$  es la salida sin retardo. La solución de 4.1 es

$$\bar{y}(t) = 0.2 + 1.5t - 0.2e^{-5t} \quad (4.2)$$

| TIEMPO | SALIDAS:   |            |            |            |            |
|--------|------------|------------|------------|------------|------------|
|        | 1          | 2          | 3          | 4          | 5          |
| 0.0    | 1.00000000 | 0.00000000 | 0.50000000 | 0.00000000 | 0.00000000 |
| 0.1    | 1.00000000 | 0.10000000 | 0.4213541  | 0.2286458  | 0.00000000 |
| 0.2    | 1.00000000 | 0.20000000 | 0.3736342  | 0.4263659  | 0.00000000 |
| 0.3    | 1.00000000 | 0.30000000 | 0.3446791  | 0.6053210  | 0.00000000 |
| 0.4    | 1.00000000 | 0.40000000 | 0.3271099  | 0.7728901  | 0.00000000 |
| 0.5    | 1.00000000 | 0.50000000 | 0.3164495  | 0.9335505  | 0.00000000 |
| 0.6    | 0.7713541  | 0.5882292  | 0.2057103  | 1.0623110  | 0.2286458  |
| 0.7    | 0.5736342  | 0.6552732  | 0.1255732  | 1.1441540  | 0.4263659  |
| 0.8    | 0.3946790  | 0.7035642  | 0.0621361  | 1.1905500  | 0.6053210  |
| 0.9    | 0.2271099  | 0.7345780  | 0.0076982  | 1.2077240  | 0.7728901  |
| 1.0    | 0.0664495  | 0.7492101  | -0.0419668 | 1.1990070  | 0.9335505  |
| 1.1    | -0.0623106 | 0.7489413  | -0.0761066 | 1.1683630  | 1.0623110  |
| 1.2    | -0.1441537 | 0.7382761  | -0.0906708 | 1.1260080  | 1.1441540  |
| 1.3    | -0.1905499 | 0.7212721  | -0.0930163 | 1.0796500  | 1.1905500  |
| 1.4    | -0.2077237 | 0.7011291  | -0.0865836 | 1.0344150  | 1.2077240  |
| 1.5    | -0.1990067 | 0.6805843  | -0.0728847 | 0.9942578  | 1.1990070  |
| 1.6    | -0.1683631 | 0.6620641  | -0.0535563 | 0.9624708  | 1.1683630  |
| 1.7    | -0.1260083 | 0.6472801  | -0.0329317 | 0.9408477  | 1.1260080  |
| 1.8    | -0.0796496 | 0.6369851  | -0.0136349 | 0.9292879  | 1.0796500  |
| 1.9    | -0.0344154 | 0.6313076  | 0.0029968  | 0.9267569  | 1.0344150  |
| 2.0    | 0.0057422  | 0.6299306  | 0.0160778  | 0.9316892  | 0.9942578  |
| 2.1    | 0.0375292  | 0.6321707  | 0.0249084  | 0.9421122  | 0.9624708  |
| 2.2    | 0.0591523  | 0.6370890  | 0.0293611  | 0.9558485  | 0.9408477  |
| 2.3    | 0.0707122  | 0.6436618  | 0.0300149  | 0.9708339  | 0.9292879  |
| 2.4    | 0.0732431  | 0.6509284  | 0.0276548  | 0.9853594  | 0.9267569  |
| 2.5    | 0.0683108  | 0.6580604  | 0.0231203  | 0.9981257  | 0.9316892  |
| 2.6    | 0.0578878  | 0.6644065  | 0.0172906  | 1.0082630  | 0.9421122  |
| 2.7    | 0.0441515  | 0.6695270  | 0.0110228  | 1.0153430  | 0.9558485  |
| 2.8    | 0.0291661  | 0.6731958  | 0.0050410  | 1.0193360  | 0.9708339  |
| 2.9    | 0.0146406  | 0.6753766  | -0.0001360 | 1.0205210  | 0.9853594  |
| 3.0    | 0.0016743  | 0.6761838  | -0.0041786 | 1.0193910  | 0.9981257  |
| 3.1    | -0.0062630 | 0.6758403  | -0.0069309 | 1.0165600  | 1.0082630  |
| 3.2    | -0.0153434 | 0.6746342  | -0.0083972 | 1.0126770  | 1.0153430  |
| 3.3    | -0.0193359 | 0.6728756  | -0.0087109 | 1.0083560  | 1.0193360  |
| 3.4    | -0.0205212 | 0.6708614  | -0.0080910 | 1.0041230  | 1.0205210  |
| 3.5    | -0.0193914 | 0.6688490  | -0.0067999 | 1.0003780  | 1.0193910  |
| 3.6    | -0.0165600 | 0.6670400  | -0.0051092 | 0.9973893  | 1.0165600  |
| 3.7    | -0.0126768 | 0.6655721  | -0.0032727 | 0.9952924  | 1.0126770  |
| 3.8    | -0.0083563 | 0.6645190  | -0.0015042 | 0.9941046  | 1.0083560  |
| 3.9    | -0.0041225 | 0.6638976  | 0.0000358  | 0.9937493  | 1.0041230  |
| 4.0    | -0.0003777 | 0.6636779  | 0.0012437  | 0.9940842  | 1.0003780  |
| 4.1    | 0.0026107  | 0.6637965  | 0.0020711  | 0.9949290  | 0.9973893  |
| 4.2    | 0.0047076  | 0.6641700  | 0.0025175  | 0.9960913  | 0.9952924  |
| 4.3    | 0.0056954  | 0.6647074  | 0.0026201  | 0.9973887  | 0.9941046  |
| 4.4    | 0.0062507  | 0.6653211  | 0.0024425  | 0.9986644  | 0.9937493  |
| 4.5    | 0.0059158  | 0.6659344  | 0.0020622  | 0.9997973  | 0.9940842  |
| 4.6    | 0.0050710  | 0.6664872  | 0.0015601  | 1.0007060  | 0.9949290  |
| 4.7    | 0.0039088  | 0.6669381  | 0.0010117  | 1.0013500  | 0.9960913  |
| 4.8    | 0.0026113  | 0.6672645  | 0.0004812  | 1.0017210  | 0.9973887  |
| 4.9    | 0.0013356  | 0.6674612  | 0.0000172  | 1.0018420  | 0.9986644  |
| 5.0    | 0.0002027  | 0.6675365  | -0.0003489 | 1.0017550  | 0.9997973  |

Tabla 3.1 Resultados del Ejemplo 1.

# Fig. 4.4. PLANTA DE ENERGIA NUCLEAR



Aplicando el retardo en el tiempo, la salida  $y(t)$  para  $0 < t < 1$  será

$$\begin{aligned}
 y(t) &= 0 && \text{para } 0 < t < 0.5 \\
 y(t) &= 0.2 + 1.5(t - 0.5) - 0.2e^{-5(t - 0.5)} && \text{para } 0.5 < t < 1 \quad (4.3)
 \end{aligned}$$

La tabla 4.2 muestra un cuadro comparativo entre los resultados analíticos y los obtenidos con el programa.

| TIEMPO | ANALITICO | PROGRAMA  | ERROR% |
|--------|-----------|-----------|--------|
| 0.0    | 0.0000000 | 0.0000000 | 0.000% |
| 0.2    | 0.0000000 | 0.0000000 | 0.000% |
| 0.4    | 0.0000000 | 0.0000000 | 0.000% |
| 0.6    | 0.2286939 | 0.2286458 | 0.021% |
| 0.8    | 0.6053740 | 0.6053210 | 0.009% |
| 1.0    | 0.9335830 | 0.9335505 | 0.003% |

Tabla 4.2 Comparación entre resultados del Ejemplo 1.

Como se puede ver el error es despreciable y en cualquier caso menor a 0.0001 que era el valor esperado. En resumen, el programa de simulación trabaja bastante bien.

#### 4.1.2 Ejemplo 2. Ecuación de Van der Pol.

La ecuación diferencial de Van der Pol (1) tiene la siguiente configuración

$$d^2y/dt^2 + u(y^2 - 1)dy/dt + y = 0 \quad (4.4)$$

Se requiere calcular el valor de  $y(t)$  para  $0 < t < 7.5$ ,  $u = 1.9$  y las condiciones iniciales  $y(0) = -2$  e  $\dot{y}(0) = 0$ .

La simulación requerida para encontrar la solución de 4.4 nos permitirá observar el comportamiento de los módulos multiplicador y cuadrador. El diagrama de simulación se muestra en la Fig. 4.5.

Escogiendo un intervalo de integración de 0.1 y un intervalo entre impresos de 0.2 encontramos la solución cuyos valores numéricos se muestran en la tabla 4.3.

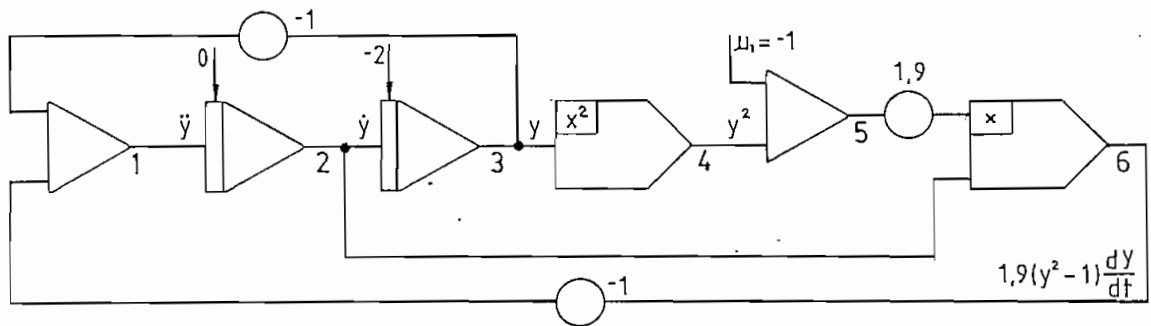


Fig. 4.5 Diagrama de simulación para resolver la ec. de Van der Pol.

| TIEMPO | SALIDAS:  |           |           |          |           |          |
|--------|-----------|-----------|-----------|----------|-----------|----------|
|        | 1         | 2         | 3         | 4        | 5         | 6        |
| 0.0    | 2.000000  | 0.000000  | -2.000000 | 4.000000 | 3.000000  | 0.000000 |
| 0.2    | 0.657729  | 0.239499  | -1.971670 | 3.887481 | 2.887481  | 1.31394  |
| 0.4    | 0.272233  | 0.324424  | -1.914026 | 3.663494 | 2.663494  | 1.64179  |
| 0.6    | 0.171756  | 0.366458  | -1.844612 | 3.402593 | 2.402593  | 1.67286  |
| 0.8    | 0.157811  | 0.398629  | -1.768059 | 3.126033 | 2.126033  | 1.61025  |
| 1.0    | 0.176350  | 0.431672  | -1.685091 | 2.839533 | 1.839533  | 1.50874  |
| 1.2    | 0.214798  | 0.470470  | -1.595006 | 2.544043 | 1.544043  | 1.38021  |
| 1.4    | 0.274683  | 0.519003  | -1.496257 | 2.238786 | 1.238786  | 1.22157  |
| 1.6    | 0.365972  | 0.582414  | -1.386419 | 1.922158 | 0.922158  | 1.02045  |
| 1.8    | 0.509392  | 0.668826  | -1.261771 | 1.592065 | 0.592065  | 0.75238  |
| 2.0    | 0.745466  | 0.792265  | -1.116444 | 1.246447 | 0.246447  | 0.37098  |
| 2.2    | 1.154544  | 0.978379  | -0.940735 | 0.884982 | -0.115018 | -0.21381 |
| 2.4    | 1.892946  | 1.275783  | -0.717765 | 0.515187 | -0.484813 | -1.17518 |
| 2.6    | 3.200997  | 1.773865  | -0.417155 | 0.174019 | -0.825981 | -2.78384 |
| 2.8    | 4.984688  | 2.588817  | 0.013277  | 0.000176 | -0.999824 | -4.91789 |
| 3.0    | 3.421932  | 3.541895  | 0.638786  | 0.397789 | -0.602211 | -4.05264 |
| 3.2    | -6.530251 | 3.309681  | 1.350448  | 1.823709 | 0.823709  | 5.17980  |
| 3.4    | -8.762085 | 1.524716  | 1.840945  | 3.389877 | 2.389877  | 6.92186  |
| 3.6    | -3.687872 | 0.293089  | 2.005239  | 4.028983 | 3.028983  | 1.68183  |
| 3.8    | -1.162767 | -0.146644 | 2.011577  | 4.046442 | 3.046442  | -0.84881 |
| 4.0    | -0.402225 | -0.287319 | 1.965710  | 3.864016 | 2.864016  | -1.56349 |
| 4.2    | -0.197209 | -0.342768 | 1.902037  | 3.617743 | 2.617743  | -1.70483 |
| 4.4    | -0.150875 | -0.376245 | 1.829986  | 3.348848 | 2.348848  | -1.67911 |
| 4.6    | -0.154765 | -0.406320 | 1.751743  | 3.068605 | 2.068605  | -1.59698 |
| 4.8    | -0.188822 | -0.439585 | 1.667240  | 2.779689 | 1.779689  | -1.48642 |
| 5.0    | -0.224401 | -0.479799 | 1.575447  | 2.482032 | 1.482032  | -1.35105 |
| 5.2    | -0.290125 | -0.530806 | 1.474605  | 2.174460 | 1.174459  | -1.18448 |
| 5.4    | -0.390218 | -0.598115 | 1.362045  | 1.855166 | 0.855166  | -0.97183 |
| 5.6    | -0.540642 | -0.690735 | 1.233685  | 1.521980 | 0.521980  | -0.68504 |
| 5.8    | -0.812079 | -0.824476 | 1.083038  | 1.172970 | 0.172970  | -0.27096 |
| 6.0    | -1.273165 | -1.028554 | 0.899262  | 0.808672 | -0.191328 | 0.37390  |
| 6.2    | -2.108719 | -1.358464 | 0.663330  | 0.440006 | -0.559994 | 1.44539  |
| 6.4    | -3.553947 | -1.913495 | 0.340958  | 0.116252 | -0.883748 | 3.21299  |
| 6.6    | -5.105925 | -2.796335 | -0.124611 | 0.015528 | -0.984472 | 5.23054  |
| 6.8    | -1.942198 | -3.654588 | -0.779743 | 0.607999 | -0.392081 | 2.72194  |
| 7.0    | 0.289634  | -3.601155 | -1.481216 | 2.194000 | 1.194000  | -6.80842 |
| 7.2    | 7.735189  | -1.182842 | -1.896798 | 3.597844 | 2.597844  | -5.83839 |
| 7.4    | 2.925267  | -0.156766 | -2.014437 | 4.057955 | 3.057955  | -0.91083 |
| 7.5    | 1.636172  | 0.865405  | -2.017935 | 4.072860 | 3.072860  | 0.38176  |

Tabla 4.3 Resultados del Ejemplo 2.



Fig. 4.6 ECUACION DE VAN DER POL

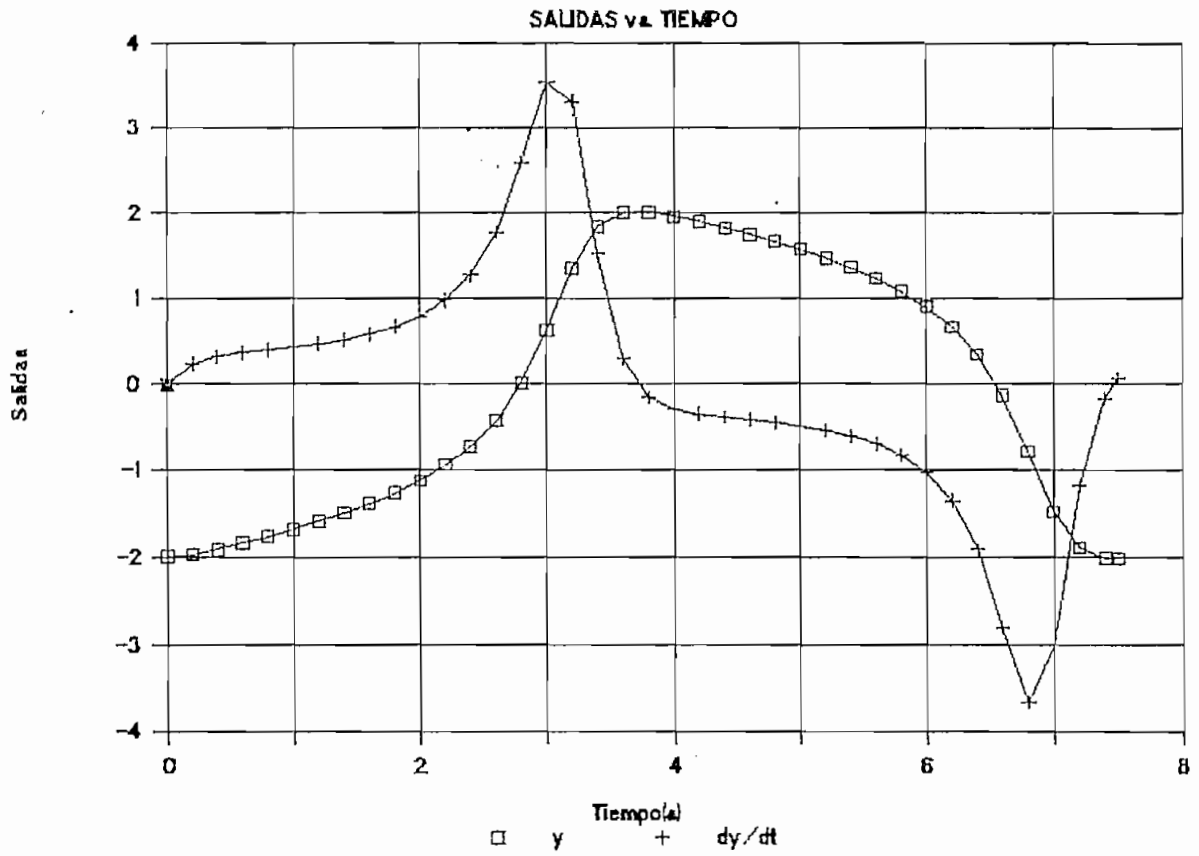
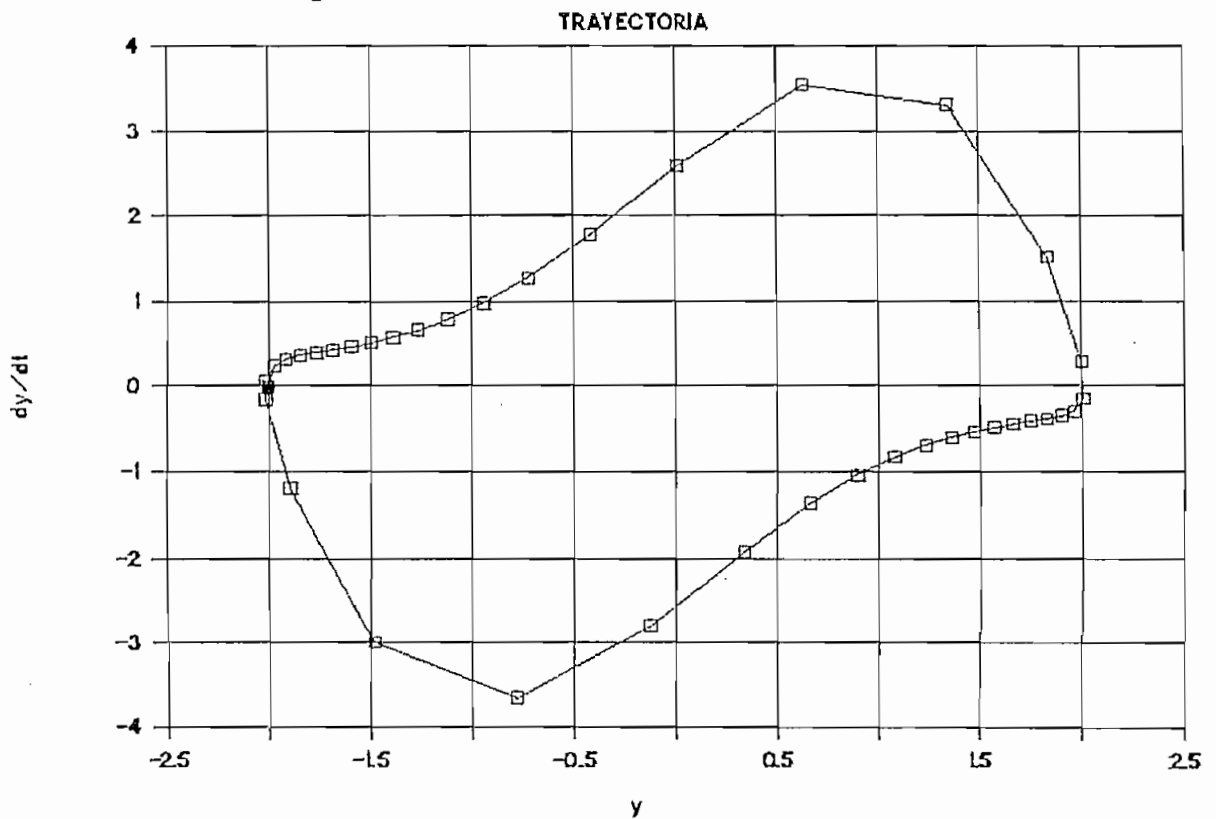


Fig. 4.7 ECUACION DE VAN DER POL



Los gráficos de  $y$ ,  $dy/dt$  se muestran en la Fig. 4.6. El gráfico de  $dy/dt$  vs.  $y$  se muestra en la Fig. 4.7.

Observando las respuestas de la tabla 3.3 vemos que se están cumpliendo las siguientes relaciones

$$\begin{aligned}y_1(t) &= -y_5(t) - y_6(t) \\y_4(t) &= (y_5(t))^2 \\y_5(t) &= y_4(t) - 1 \\y_6(t) &= 1.9y_5(t)y_2(t)\end{aligned}$$

Esto demuestra que los diferentes módulos funcionan correctamente. El resultado final coincide con el resultado obtenido por simulación analógica (Ref. 1).

#### 4.1.3 Ejemplo 3. Sistema de control Si-No con histéresis.

Se requiere investigar el comportamiento del sistema de control que se muestra en la Fig. 4.8 frente a una perturbación escalón unitario en la entrada.

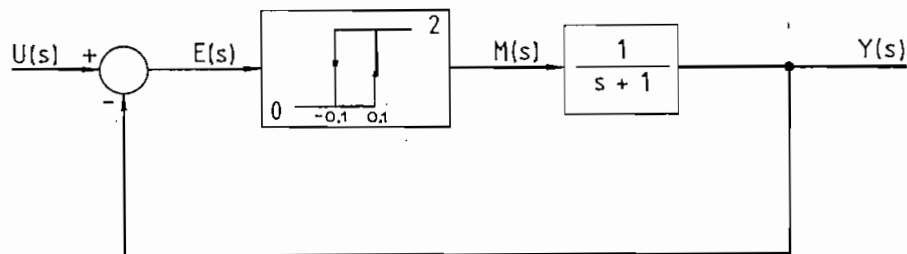


Fig. 4.8 Sistema de control Si-No con histéresis.

La relación de transferencia del comparador con histéresis del diagrama no es igual a la relación de transferencia del módulo disponible en el programa. Para adaptar esta relación podemos utilizar la configuración que se muestra en la Fig. 4.9.

El diagrama de simulación para el problema se muestra en la Fig. 4.10. Este diagrama permite observar las respuestas de  $e(t)$ ,  $m(t)$  y la salida.

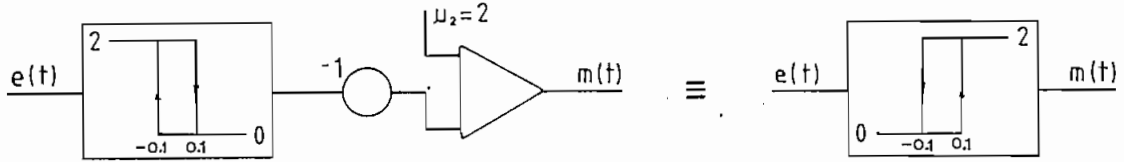


Fig. 4.9 Diagrama de simulación para el comparador con histéresis.

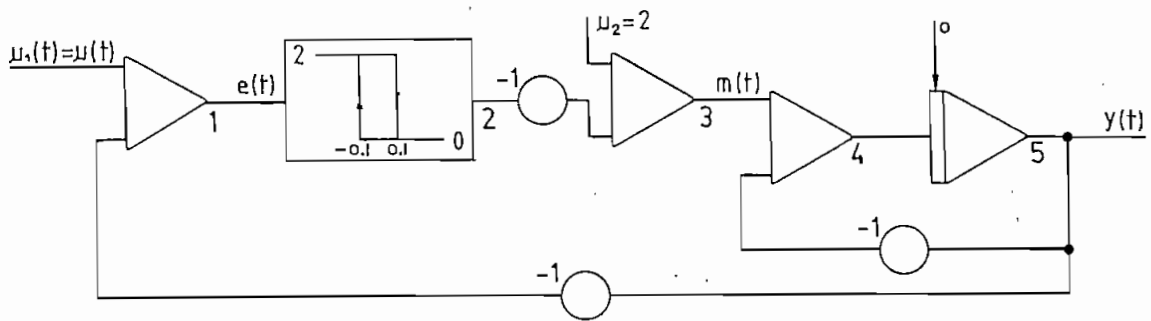


Fig. 4.10 Diagrama de simulación para el sistema de control de la Fig. 4.8.

Los resultados numéricos obtenidos para  $t_0=0$ ,  $t_r=1.6$  y  $h=0.1$  se muestran en la tabla 4.4. En la Fig. 4.11 se presenta la gráfica de la salida vs. tiempo y en la Fig. 4.12 el diagrama en el plano de fase.

| TIEMPO | SALIDAS:  |   |   |           |          |
|--------|-----------|---|---|-----------|----------|
|        | 1         | 2 | 3 | 4         | 5        |
| 0.0    | 1.000000  | 0 | 2 | 2.000000  | 0.000000 |
| 0.1    | 0.809675  | 0 | 2 | 1.809675  | 0.190325 |
| 0.2    | 0.637462  | 0 | 2 | 1.637462  | 0.362538 |
| 0.3    | 0.481637  | 0 | 2 | 1.481637  | 0.518363 |
| 0.4    | 0.340640  | 0 | 2 | 1.340640  | 0.659360 |
| 0.5    | 0.213061  | 0 | 2 | 1.213061  | 0.786939 |
| 0.6    | 0.097623  | 0 | 2 | 1.097623  | 0.902377 |
| 0.7    | -0.006829 | 0 | 2 | 0.993171  | 1.006829 |
| 0.8    | -0.098009 | 2 | 0 | -1.098009 | 1.098009 |
| 0.9    | -0.006481 | 2 | 0 | -0.993519 | 0.993519 |
| 1.0    | 0.097693  | 0 | 2 | 1.097693  | 0.902307 |
| 1.1    | -0.006766 | 0 | 2 | 0.993234  | 1.006766 |
| 1.2    | -0.097951 | 2 | 0 | -1.097951 | 1.097951 |
| 1.3    | 0.006532  | 2 | 0 | -0.993468 | 0.993468 |
| 1.4    | 0.097740  | 0 | 2 | 1.097740  | 0.902260 |
| 1.5    | -0.006724 | 0 | 2 | 0.993276  | 1.006724 |
| 1.6    | -0.097913 | 2 | 0 | -1.097913 | 1.097913 |

Tabla 4.4 Resultados del Ejemplo 3.

Fig. 4.11 SISTEMA SI-NO CON HISTERESIS

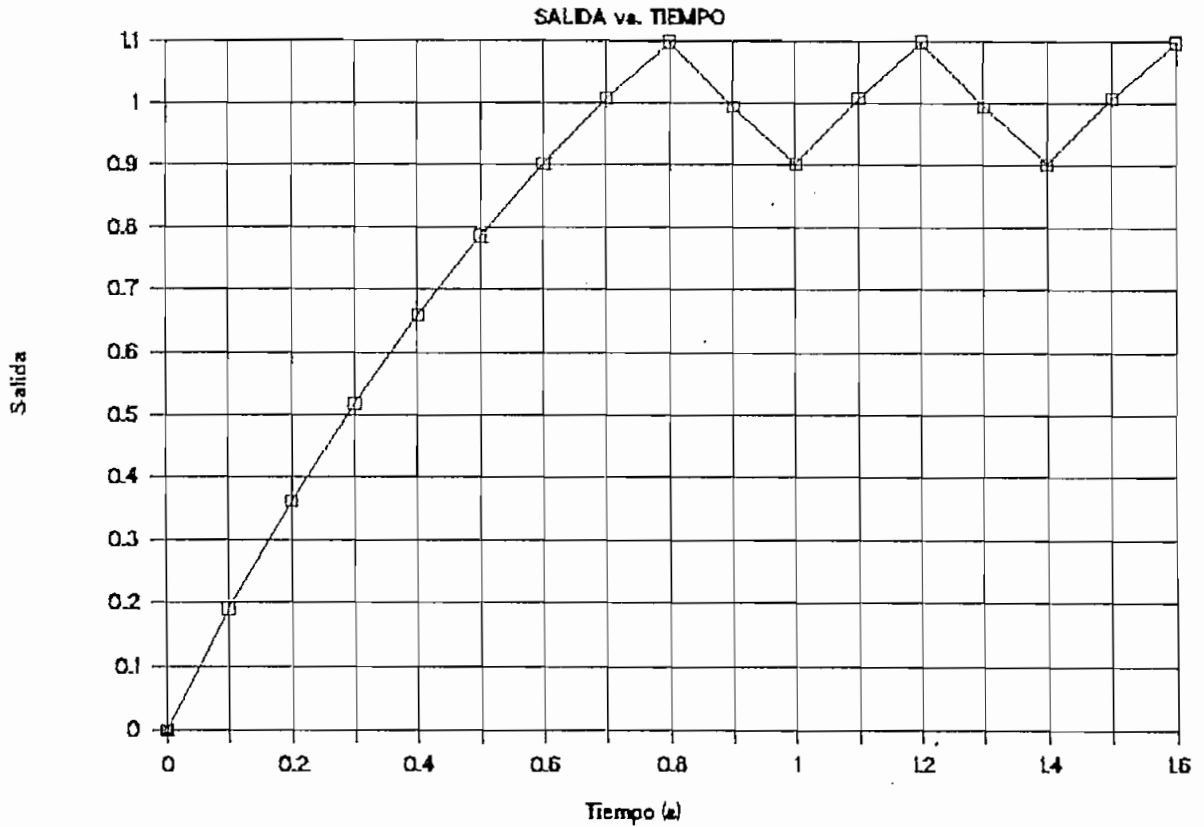
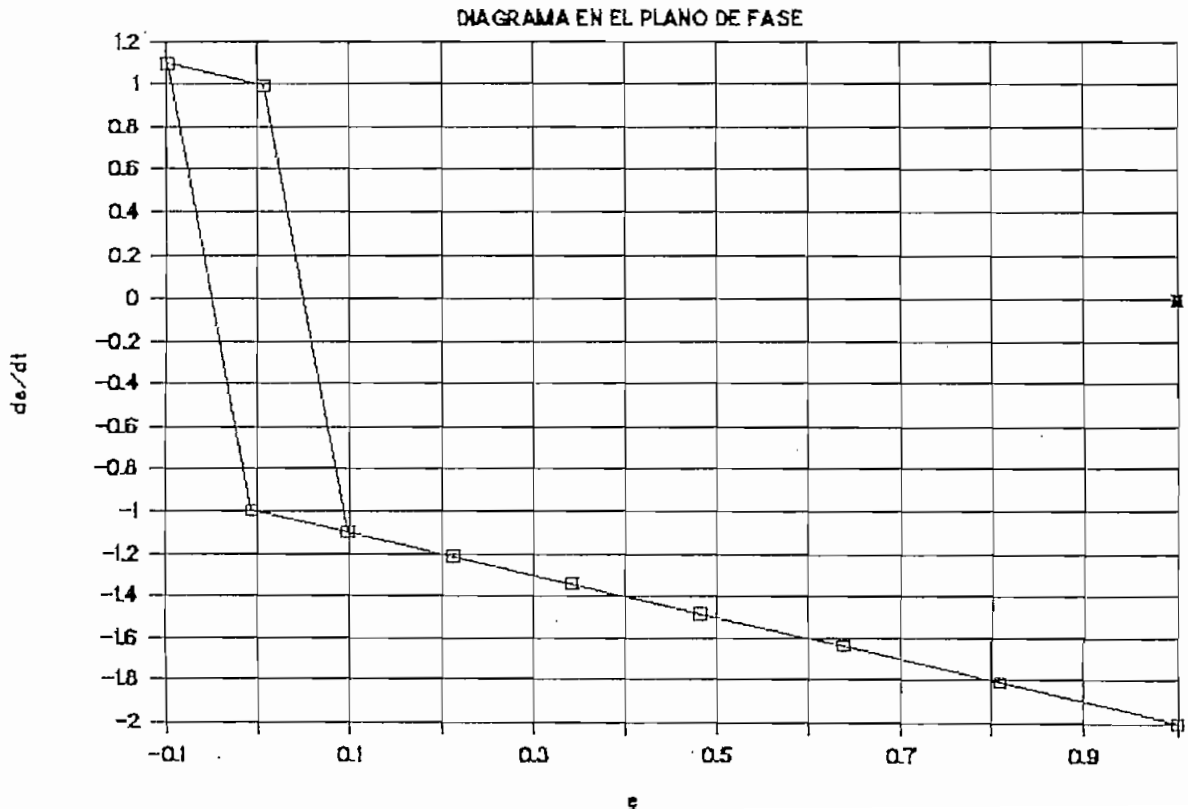


Fig. 4.12 SISTEMA SI-NO CON HISTERESIS



Para comprobar los resultados obtenidos, presentamos la solución analítica del problema

$$\begin{aligned}
 y(t) &= 2(1-e^{-t}) && \text{para } 0.0 < t < 0.8 \\
 y(t) &= 1.1e^{-(t-0.8)} && \text{para } 0.8 < t < 1.0 \\
 y(t) &= 2-1.1e^{-(t-1)} && \text{para } 1.0 < t < 1.2 \\
 y(t) &= 1.1e^{-(t-1.2)} && \text{para } 1.2 < t < 1.4 \\
 y(t) &= 2-1.1e^{-(t-1.4)} && \text{para } 1.4 < t < 1.6
 \end{aligned} \tag{4.5}$$

Los resultados analíticos coinciden plenamente con los resultados del programa como se puede ver en la tabla 4.5.

| TIEMPO | PROGRAMA | ANALITICO | ERROR% |
|--------|----------|-----------|--------|
| 0.0    | 0.000000 | 0.000000  | 0.00%  |
| 0.1    | 0.190325 | 0.190325  | 0.00%  |
| 0.2    | 0.362538 | 0.362538  | 0.00%  |
| 0.3    | 0.518363 | 0.518364  | 0.00%  |
| 0.4    | 0.659360 | 0.659360  | 0.00%  |
| 0.5    | 0.786939 | 0.786939  | 0.00%  |
| 0.6    | 0.902377 | 0.902377  | 0.00%  |
| 0.7    | 1.006829 | 1.006829  | 0.00%  |
| 0.8    | 1.098009 | 1.101342  | -0.33% |
| 0.9    | 0.993519 | 0.995321  | -0.18% |
| 1.0    | 0.902307 | 0.900604  | 0.17%  |
| 1.1    | 1.006766 | 1.004679  | 0.21%  |
| 1.2    | 1.097951 | 1.099396  | -0.14% |
| 1.3    | 0.993468 | 0.995321  | -0.19% |
| 1.4    | 0.902260 | 0.900604  | 0.17%  |
| 1.5    | 1.006724 | 1.004679  | 0.20%  |
| 1.6    | 1.097913 | 1.099396  | -0.15% |

Tabla 4.5 Comparación entre resultados del Ejemplo 3.

#### 4.1.4 Ejemplo 4. Sistema de control con ganancia no lineal.

Consideremos el sistema no lineal que se ve en la Fig. 4.13(a). El bloque indicado como  $G_N$  es el elemento de ganancia no lineal. En la Fig. 4.13(b) se ve la función de transferencia de este elemento. La ganancia del elemento es la unidad o  $k$ , toda vez que el valor absoluto de la señal de error es mayor o menor que  $e_0$ , respectivamente. Es decir,

$$\begin{aligned}
 m &= e && \text{para } |e| > e_0 \\
 m &= ke && \text{para } |e| < e_0
 \end{aligned}$$

El sistema tiene elevada ganancia para señales de error

elevadas y pequeña ganancia para señales de error pequeñas. Se puede efectuar la conmutación de ganancia utilizando un dispositivo conmutador, que modifica abruptamente la ganancia del amplificador de un valor a otro. Para valores pequeños de  $e$  el sistema presenta una respuesta lenta y para valores elevados de error, respuesta rápida. Esta característica puede ser deseable en sistemas sujetos a ruidos de baja amplitud y alta frecuencia, ya que reducen sustancialmente las señales de ruido indeseables, mientras se transmiten satisfactoriamente las señales de mando.

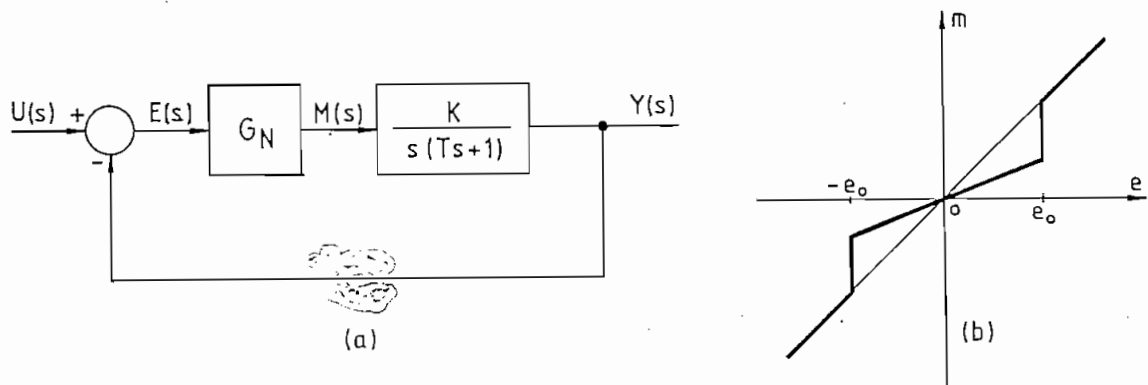


Fig. 4.13 Sistema de control con ganancia no lineal.

Se requiere averiguar el comportamiento del sistema para  $k=0.2$ ,  $K=10$  y  $T=2$ .

La función de transferencia del elemento de ganancia no lineal no está disponible como un solo módulo en el programa, por tanto, debemos generar esta función con la ayuda de los módulos disponibles. El diagrama de simulación que se muestra en la Fig. 4.14 presenta una forma de hacerlo.

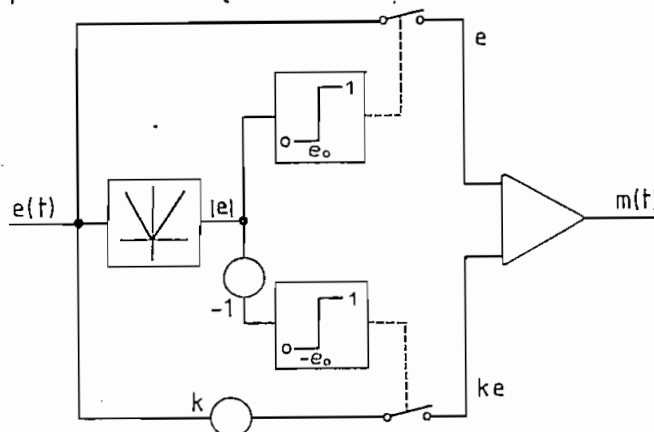


Fig. 4.14 Diagrama de simulación para obtener la función de transferencia de la Fig. 4.13(b)

El diagrama de la Fig. 4.14 permite además observar el comportamiento de los módulos valor absoluto, comparador y switch análogo.

El diagrama de simulación completo que permite obtener los valores de las señales de error y la señal  $m$  se muestra en la Fig. 4.15.

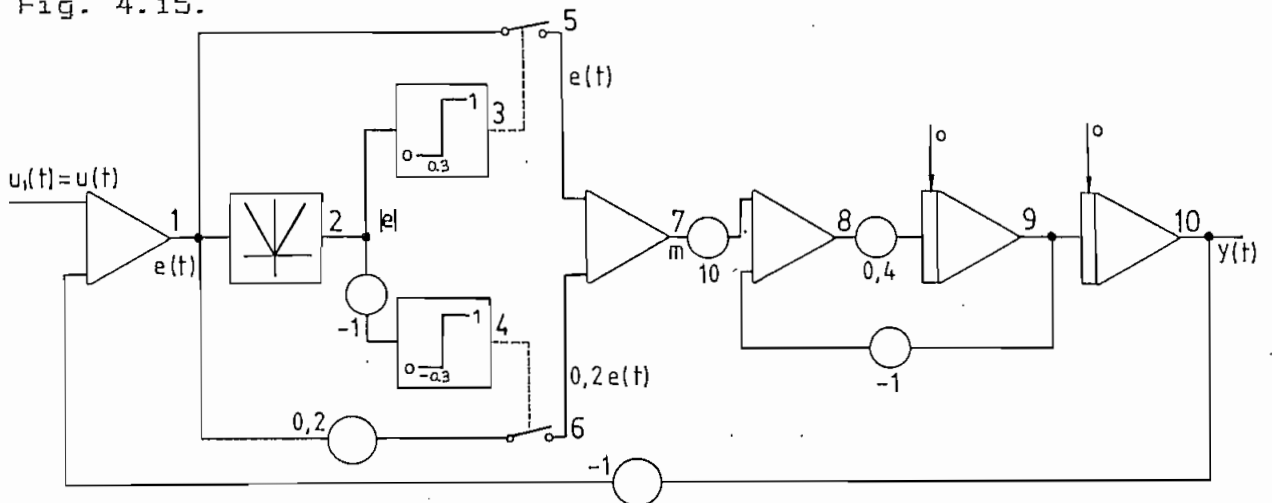


Fig. 4.15 Diagrama de simulación del Ejemplo 4.

Los resultados numéricos obtenidos para  $t_0=0$ ,  $t_f=12$  y  $h=0.01$  se muestran en la tabla 4.6. En las Figs. 4.16 y 4.17 se exhiben la forma de las señales de salida vs. tiempo y el diagrama en el plano de fase respectivamente. Es claro el efecto de la ganancia no lineal sobre las salidas del sistema. La comprobación de los resultados puede hacerse considerando al sistema de segundo orden con frecuencia natural de 2 y relación de amortiguación de 0.1 para valores absolutos de error mayores a 0.3 y un sistema de segundo orden con frecuencia natural de 0.8944 y relación de amortiguamiento de 0.2236 para valores absolutos de error menores a 0.3. Los resultados obtenidos de esta forma coinciden plenamente con los obtenidos en el programa.

De acuerdo a los resultados de la tabla 4.6, es fácil observar el correcto funcionamiento de los módulos valor absoluto, comparador y switch análogo. Otra forma de comprobar el funcionamiento de estos módulos es mediante el resultado final de la simulación del módulo de ganancia no lineal Fig.

| SALIDAS: |          |         |   |   |          |          |          |          |          |         |
|----------|----------|---------|---|---|----------|----------|----------|----------|----------|---------|
| TIEMPO   | 1        | 2       | 3 | 4 | 5        | 6        | 7        | 8        | 9        | 10      |
| 0.00     | 1.00000  | 1.00000 | 1 | 0 | 1.00000  | 0.00000  | 1.00000  | 10.00000 | 0.00000  | 0.00000 |
| 0.25     | 0.88155  | 0.88155 | 1 | 0 | 0.88155  | 0.00000  | 0.88155  | 7.90299  | 0.91247  | 0.11845 |
| 0.50     | 0.56897  | 0.56897 | 1 | 0 | 0.56897  | 0.00000  | 0.56897  | 4.16420  | 1.52552  | 0.43103 |
| 0.75     | 0.15632  | 0.15632 | 0 | 1 | 0.00000  | 0.04690  | 0.04690  | -1.20022 | 1.66919  | 0.84368 |
| 1.00     | -0.24148 | 0.24148 | 0 | 1 | 0.00000  | -0.07244 | -0.07244 | -2.22065 | 1.49621  | 1.24148 |
| 1.25     | -0.56034 | 0.56034 | 1 | 0 | -0.56034 | 0.00000  | -0.56034 | -6.50904 | 0.90569  | 1.56034 |
| 1.50     | -0.71802 | 0.71802 | 1 | 0 | -0.71802 | 0.00000  | -0.71802 | -7.45505 | 0.26770  | 1.71802 |
| 1.75     | -0.69474 | 0.69474 | 1 | 0 | -0.69474 | 0.00000  | -0.69474 | -6.50302 | -0.44434 | 1.69474 |
| 2.00     | -0.51100 | 0.51100 | 1 | 0 | -0.51100 | 0.00000  | -0.51100 | -4.12571 | -0.98509 | 1.51100 |
| 2.25     | -0.22720 | 0.22720 | 0 | 1 | 0.00000  | -0.06018 | -0.06018 | 0.51000  | -1.19992 | 1.22720 |
| 2.50     | 0.06203  | 0.06203 | 0 | 1 | 0.00000  | 0.01805  | 0.01805  | 1.29645  | -1.10797 | 0.93717 |
| 2.75     | 0.32067  | 0.32067 | 1 | 0 | 0.32067  | 0.00000  | 0.32067  | 4.13415  | -0.92742 | 0.67933 |
| 3.00     | 0.49425  | 0.49425 | 1 | 0 | 0.49425  | 0.00000  | 0.49425  | 5.30204  | -0.44034 | 0.50575 |
| 3.25     | 0.53615  | 0.53615 | 1 | 0 | 0.53615  | 0.00000  | 0.53615  | 5.25054  | 0.10299  | 0.46385 |
| 3.50     | 0.44915  | 0.44915 | 1 | 0 | 0.44915  | 0.00000  | 0.44915  | 3.92000  | 0.57062  | 0.55005 |
| 3.75     | 0.26647  | 0.26647 | 0 | 1 | 0.00000  | 0.07994  | 0.07994  | -0.02060 | 0.02002  | 0.73353 |
| 4.00     | 0.06233  | 0.06233 | 0 | 1 | 0.00000  | 0.01070  | 0.01070  | -0.60049 | 0.79540  | 0.93767 |
| 4.25     | -0.12601 | 0.12601 | 0 | 1 | 0.00000  | -0.03004 | -0.03004 | -1.09005 | 0.70961  | 1.12601 |
| 4.50     | -0.20096 | 0.20096 | 0 | 1 | 0.00000  | -0.00669 | -0.00669 | -1.44046 | 0.50150  | 1.20096 |
| 4.75     | -0.39074 | 0.39074 | 1 | 0 | -0.39074 | 0.00000  | -0.39074 | -4.11542 | 0.20006  | 1.39074 |
| 5.00     | -0.39191 | 0.39191 | 1 | 0 | -0.39191 | 0.00000  | -0.39191 | -3.72704 | -0.19211 | 1.39192 |
| 5.25     | -0.30167 | 0.30167 | 1 | 0 | -0.30167 | 0.00000  | -0.30167 | -2.50725 | -0.50943 | 1.30167 |
| 5.50     | -0.17077 | 0.17077 | 0 | 1 | 0.00000  | -0.05123 | -0.05123 | 0.01715  | -0.52946 | 1.17077 |
| 5.75     | -0.04022 | 0.04022 | 0 | 1 | 0.00000  | -0.01207 | -0.01207 | 0.30010  | -0.50076 | 1.04022 |
| 6.00     | 0.00076  | 0.00076 | 0 | 1 | 0.00000  | 0.02423  | 0.02423  | 0.69623  | -0.45395 | 0.91924 |
| 6.25     | 0.10450  | 0.10450 | 0 | 1 | 0.00000  | 0.05535  | 0.05535  | 0.92565  | -0.37215 | 0.81550 |
| 6.50     | 0.26520  | 0.26520 | 0 | 1 | 0.00000  | 0.07950  | 0.07950  | 1.06759  | -0.27174 | 0.73472 |
| 6.75     | 0.31462  | 0.31462 | 1 | 0 | 0.31462  | 0.00000  | 0.31462  | 3.21645  | -0.07029 | 0.60530 |
| 7.00     | 0.29300  | 0.29300 | 0 | 1 | 0.00000  | 0.00014  | 0.00014  | 0.67602  | 0.20530  | 0.70620 |
| 7.25     | 0.23495  | 0.23495 | 0 | 1 | 0.00000  | 0.07049  | 0.07049  | 0.44335  | 0.26150  | 0.76505 |
| 7.50     | 0.16504  | 0.16504 | 0 | 1 | 0.00000  | 0.04951  | 0.04951  | 0.20139  | 0.29374  | 0.83496 |
| 7.75     | 0.09000  | 0.09000 | 0 | 1 | 0.00000  | 0.02702  | 0.02702  | -0.03104 | 0.30207  | 0.90992 |
| 8.00     | 0.01506  | 0.01506 | 0 | 1 | 0.00000  | 0.00476  | 0.00476  | -0.24061 | 0.20019  | 0.90014 |
| 8.25     | -0.05242 | 0.05242 | 0 | 1 | 0.00000  | -0.01573 | -0.01573 | -0.41245 | 0.25519  | 1.05242 |
| 8.50     | -0.11049 | 0.11049 | 0 | 1 | 0.00000  | -0.03315 | -0.03315 | -0.53060 | 0.20723  | 1.11049 |
| 8.75     | -0.15519 | 0.15519 | 0 | 1 | 0.00000  | -0.04656 | -0.04656 | -0.61470 | 0.14913  | 1.15519 |
| 9.00     | -0.10463 | 0.10463 | 0 | 1 | 0.00000  | -0.05539 | -0.05539 | -0.63900 | 0.00599  | 1.10463 |
| 9.25     | -0.19017 | 0.19017 | 0 | 1 | 0.00000  | -0.05945 | -0.05945 | -0.61720 | 0.02275  | 1.19010 |
| 9.50     | -0.19637 | 0.19637 | 0 | 1 | 0.00000  | -0.05091 | -0.05091 | -0.55304 | -0.03600 | 1.19637 |
| 9.75     | -0.10002 | 0.10002 | 0 | 1 | 0.00000  | -0.05425 | -0.05425 | -0.45570 | -0.00675 | 1.10002 |
| 10.00    | -0.15392 | 0.15392 | 0 | 1 | 0.00000  | -0.04617 | -0.04617 | -0.33530 | -0.12645 | 1.15392 |
| 10.25    | -0.11066 | 0.11066 | 0 | 1 | 0.00000  | -0.03560 | -0.03560 | -0.20257 | -0.15340 | 1.11066 |
| 10.50    | -0.07034 | 0.07034 | 0 | 1 | 0.00000  | -0.02350 | -0.02350 | -0.06011 | -0.16691 | 1.07034 |
| 10.75    | -0.03630 | 0.03630 | 0 | 1 | 0.00000  | -0.01009 | -0.01009 | 0.05039  | -0.16729 | 1.03630 |
| 11.00    | 0.00431  | 0.00431 | 0 | 1 | 0.00000  | 0.00129  | 0.00129  | 0.16071  | -0.15577 | 0.99569 |
| 11.25    | 0.04075  | 0.04075 | 0 | 1 | 0.00000  | 0.01223  | 0.01223  | 0.25656  | -0.13430 | 0.95925 |
| 11.50    | 0.07004  | 0.07004 | 0 | 1 | 0.00000  | 0.02125  | 0.02125  | 0.31706  | -0.10535 | 0.92916 |
| 11.75    | 0.09303  | 0.09303 | 0 | 1 | 0.00000  | 0.02791  | 0.02791  | 0.35070  | -0.07160 | 0.90697 |
| 12.00    | 0.10652  | 0.10652 | 0 | 1 | 0.00000  | 0.03196  | 0.03196  | 0.35569  | -0.03613 | 0.89340 |

Tabla 4.6 Resultados del ejemplo 4.



Fig. 4.16 SISTEMA DE GANANCIA NO LINEAL

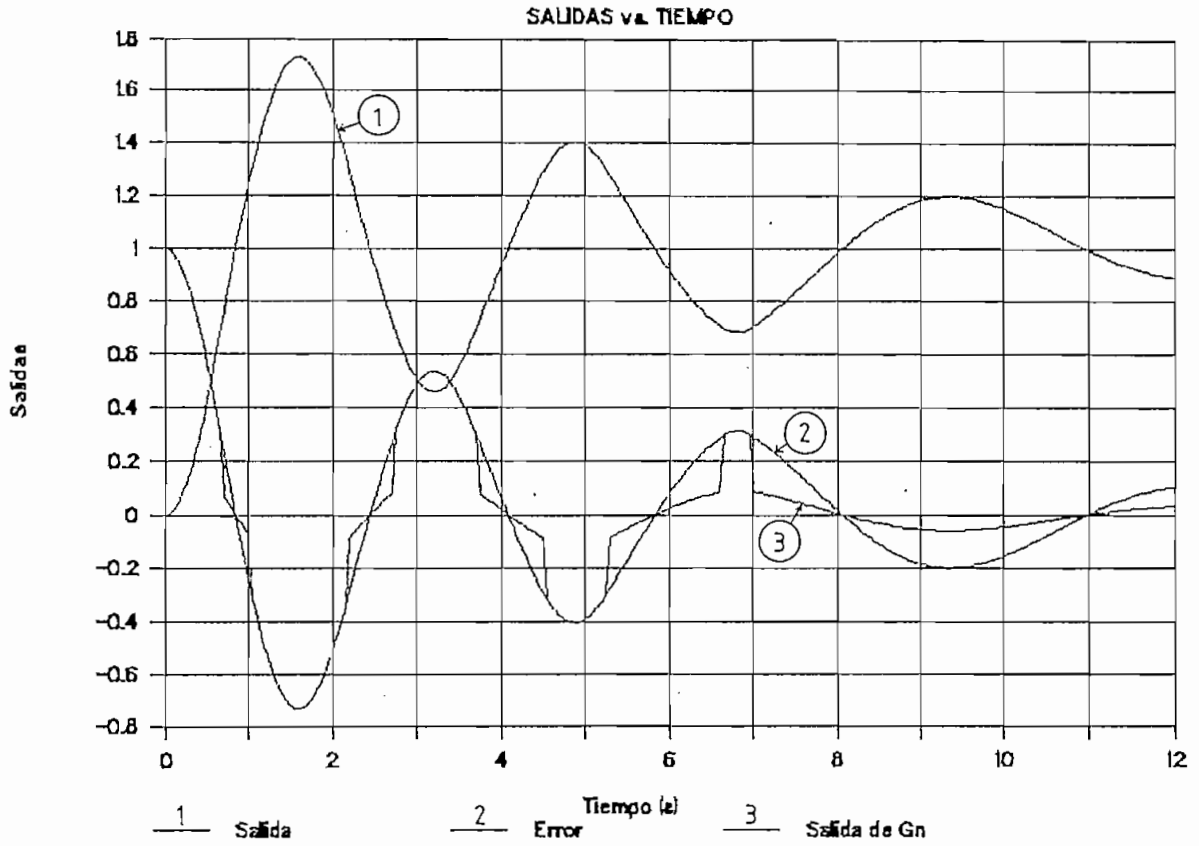
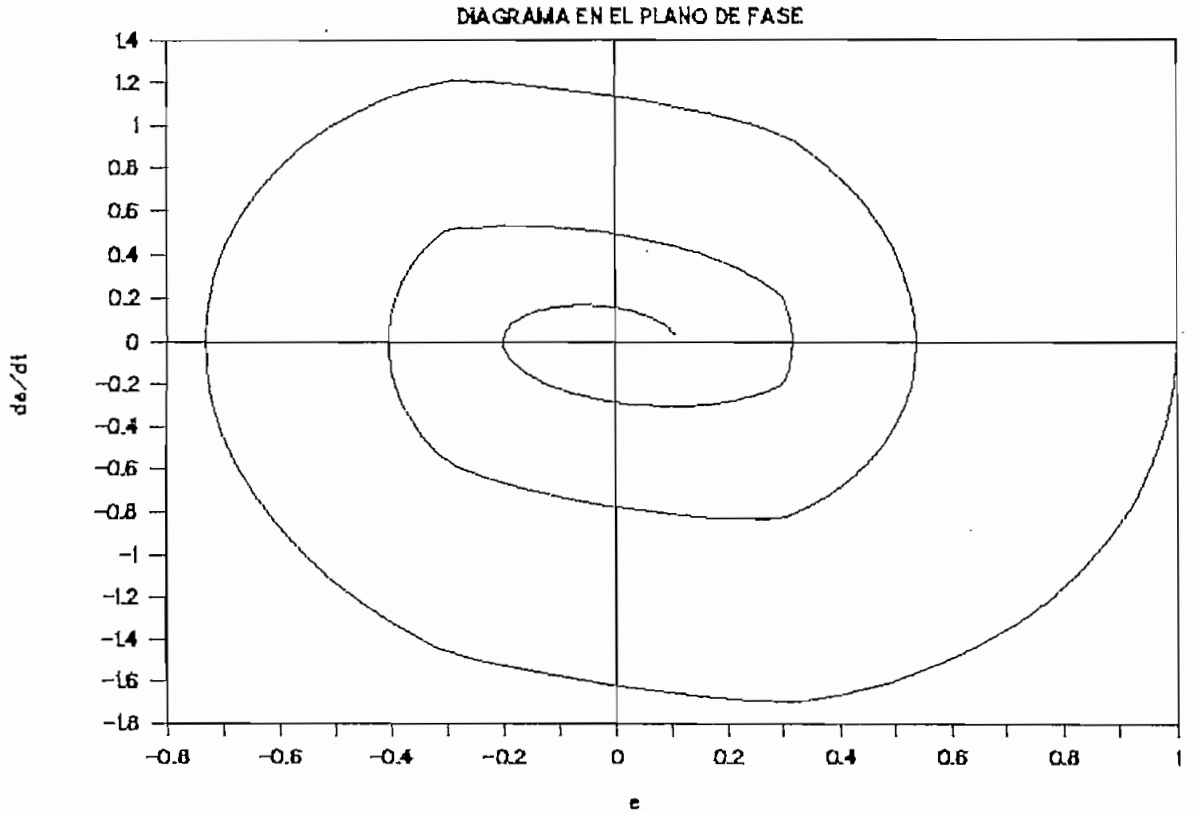


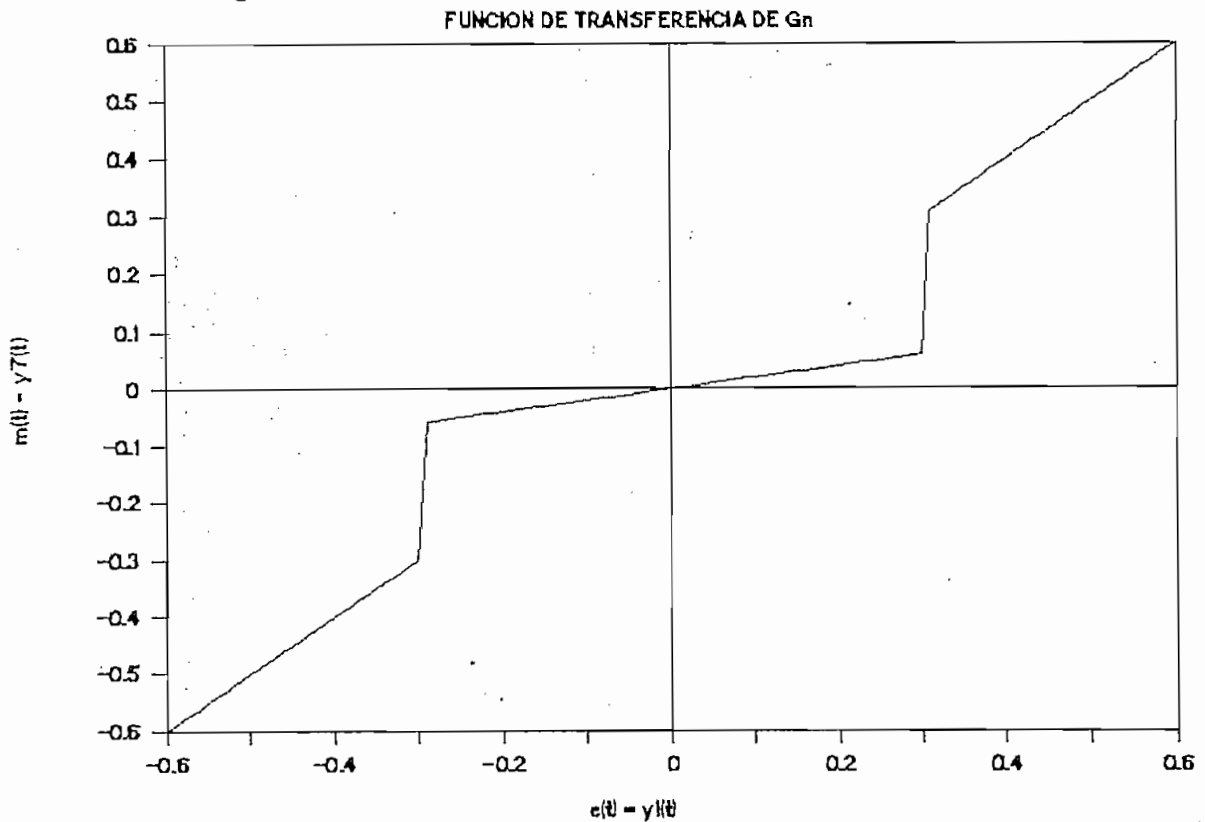
Fig. 4.17 SISTEMA DE GANANCIA NO LINEAL



4.14, si graficamos  $m(t)$  en función de  $e(t)$  tendremos como resultado la función de transferencia de la Fig. 4.13(b). En la Fig. 4.18 se observa el gráfico.

Este ejemplo permite demostrar la versatilidad del programa en lo referente a las interconexiones serie-paralelo que se pueden realizar entre los diferentes módulos disponibles.

**Fig. 4.18 SISTEMA DE GANANCIA NO LINEAL**



4.1.5 Ejemplo 5... Sistema lineal con perturbación.

En la Fig. 4.19 se muestra un modelo muy sencillo del sistema de retroalimentación que usa un estudiante para controlar sus calificaciones. Este modelo ha sido sugerido por el profesor O. J. M. Smith de la Universidad de California en Berkeley (2). Algunos valores típicos de los parámetros serían

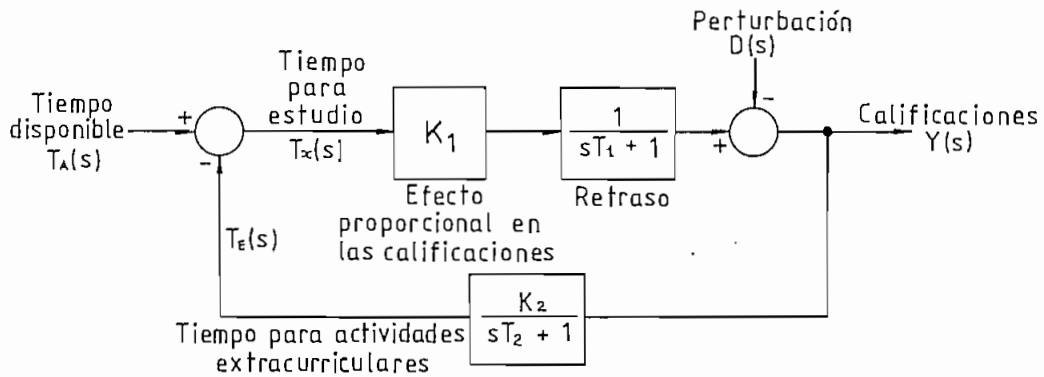


Fig. 4.19 Sistema de control de calificaciones.

$K_1=1$ ,  $T_1=1$  mes y  $T_2=0.5$  mes. El esfuerzo por eliminar actividades extracurriculares se refleja en la ganancia  $K_2$ , para la cual un estudiante demasiado confiado podría tener  $K_2=0.5$ . Se requiere el valor de la respuesta para un incremento de escalón en el tiempo disponible. Una perturbación de escalón  $D(s)=0.2/s$ , se presenta debido a materias más difíciles y de mayor competencia. Queremos determinar el efecto de la perturbación sobre las calificaciones.

Vamos a obtener un diagrama de simulación que permita observar simultáneamente las respuestas del sistema frente a la perturbación en el tiempo disponible como a la perturbación en las calificaciones. Además, se podrá observar las señales de tiempo de estudio y tiempo para actividades extracurriculares. El diagrama de simulación se muestra en la Fig. 4.20.

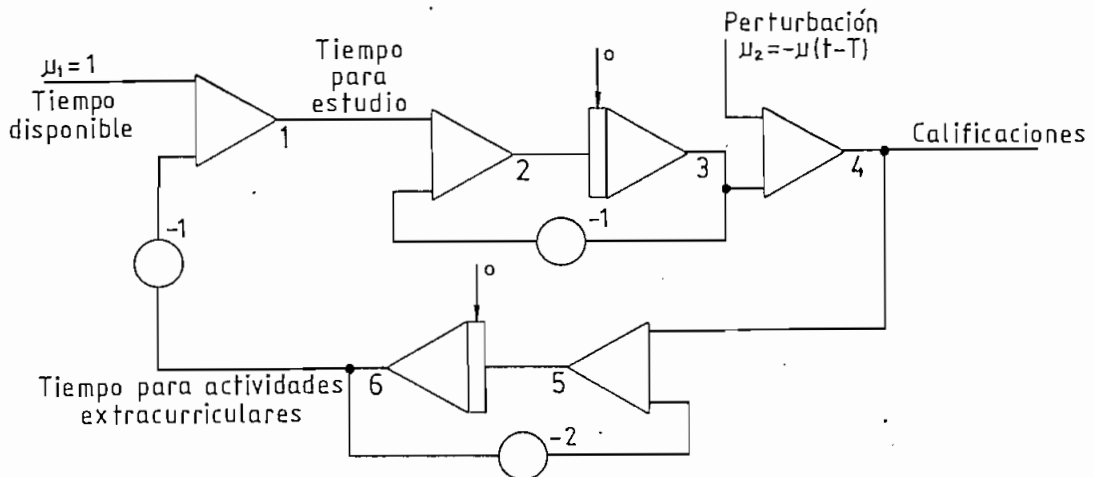


Fig. 4.20 Diagrama de simulación del ejemplo 5.

En la Fig. 4.20  $u_1(t)=u(t)$  es la señal de entrada que representa el incremento en tiempo disponible,  $u_2(t)=-u(t-a)$  corresponde a la perturbación en las calificaciones. Una primera simulación considerando a  $u_2(t)=0$  en todo el intervalo permite determinar el tiempo requerido para la estabilización de las calificaciones por efecto del incremento en el tiempo disponible. Este se produce a partir del cuarto mes. Con este dato, podemos determinar un valor adecuado de  $a$ , es decir, un valor de  $a$  igual o superior a 4. El resultado final de la simulación considerando  $t_0=0$ ,  $t_f=10$  (meses),  $h=0.1$  y  $a=6$  se muestra en la tabla 4.7.

La Fig. 4.21 muestra cual es la evolución de las calificaciones en el tiempo, en la Fig. 4.22 se exhiben los tiempos utilizados en estudio y en actividades extracurriculares. Podemos observar que gracias a un mayor tiempo dedicado al estudio las calificaciones crecen rápidamente en el primer mes y se estabilizan. Desde que se produce la perturbación en el sexto mes, las calificaciones son afectadas adversamente y se alcanza una pequeña recuperación gracias a una disminución del tiempo de actividades extracurriculares y al incremento en actividades de estudio. En todo caso, luego de la perturbación nunca se vuelve a alcanzar el valor de estado estable que se tenía antes de esta.

La comprobación de los resultados obtenidos encontrando la solución analítica del problema es algo compleja, sin embargo, podemos hacer una sencilla comprobación basada en el valor de régimen del sistema antes y después de la perturbación en las calificaciones. La función de transferencia del sistema que relaciona las calificaciones respecto del tiempo disponible es

$$T(s) = Y(s)/T_A(s) = (s+2)/(s^2+3s+3) \quad (4.6)$$

aplicando el teorema del valor final a 4.6 para  $T_A(s)=1/s$  obtenemos un valor final de  $y(t)$  igual a 0.66666. Este valor concuerda con el obtenido por simulación. La función de transferencia que relaciona las calificaciones respecto de la per-

| TIEMPO | SALIDAS: |           |          |          |           |          |
|--------|----------|-----------|----------|----------|-----------|----------|
|        | 1        | 2         | 3        | 4        | 5         | 6        |
| 0.0    | 1.000000 | 1.000000  | 0.000000 | 0.000000 | 0.000000  | 0.000000 |
| 0.2    | 0.983622 | 0.803443  | 0.160179 | 0.160179 | 0.147423  | 0.016378 |
| 0.4    | 0.946315 | 0.623788  | 0.322528 | 0.322528 | 0.215158  | 0.053685 |
| 0.6    | 0.900056 | 0.469459  | 0.431397 | 0.431397 | 0.233108  | 0.099144 |
| 0.8    | 0.854984 | 0.342818  | 0.512166 | 0.512166 | 0.222133  | 0.145017 |
| 1.0    | 0.812983 | 0.242688  | 0.570296 | 0.570296 | 0.196263  | 0.187017 |
| 1.2    | 0.776861 | 0.166052  | 0.610809 | 0.610809 | 0.164530  | 0.223139 |
| 1.4    | 0.747191 | 0.109158  | 0.638033 | 0.638033 | 0.132415  | 0.252809 |
| 1.6    | 0.723714 | 0.068181  | 0.655532 | 0.655532 | 0.102959  | 0.276287 |
| 1.8    | 0.705732 | 0.039601  | 0.666131 | 0.666131 | 0.077595  | 0.294268 |
| 2.0    | 0.692373 | 0.020378  | 0.671995 | 0.671995 | 0.056742  | 0.307627 |
| 2.2    | 0.682745 | 0.008008  | 0.674738 | 0.674738 | 0.040228  | 0.317255 |
| 2.4    | 0.676025 | 0.000503  | 0.675522 | 0.675522 | 0.027571  | 0.323975 |
| 2.6    | 0.671500 | -0.003668 | 0.675161 | 0.675161 | 0.018161  | 0.328500 |
| 2.8    | 0.668585 | -0.005618 | 0.674204 | 0.674204 | 0.011375  | 0.331415 |
| 3.0    | 0.666814 | -0.006191 | 0.673005 | 0.673005 | 0.006634  | 0.333186 |
| 3.2    | 0.665829 | -0.005952 | 0.671781 | 0.671781 | 0.003439  | 0.334171 |
| 3.4    | 0.665363 | -0.005289 | 0.670652 | 0.670652 | 0.001379  | 0.334637 |
| 3.6    | 0.665224 | -0.004453 | 0.669677 | 0.669677 | 0.000125  | 0.334776 |
| 3.8    | 0.665277 | -0.003596 | 0.668872 | 0.668872 | -0.000575 | 0.334724 |
| 4.0    | 0.665430 | -0.002804 | 0.668234 | 0.668234 | -0.000907 | 0.334571 |
| 4.2    | 0.665624 | -0.002119 | 0.667744 | 0.667744 | -0.001008 | 0.334376 |
| 4.4    | 0.665824 | -0.001554 | 0.667378 | 0.667378 | -0.000974 | 0.334176 |
| 4.6    | 0.666009 | -0.001105 | 0.667114 | 0.667114 | -0.000868 | 0.333991 |
| 4.8    | 0.666169 | -0.000760 | 0.666929 | 0.666929 | -0.000732 | 0.333831 |
| 5.0    | 0.666302 | -0.000503 | 0.666804 | 0.666804 | -0.000592 | 0.333698 |
| 5.2    | 0.666407 | -0.000317 | 0.666723 | 0.666723 | -0.000463 | 0.333593 |
| 5.4    | 0.666488 | -0.000186 | 0.666674 | 0.666674 | -0.000350 | 0.333512 |
| 5.6    | 0.666548 | -0.000098 | 0.666646 | 0.666646 | -0.000257 | 0.333452 |
| 5.8    | 0.666592 | -0.000041 | 0.666633 | 0.666633 | -0.000183 | 0.333408 |
| 6.0    | 0.666956 | 0.000328  | 0.666628 | 0.666628 | -0.193460 | 0.330044 |
| 6.2    | 0.701590 | 0.031194  | 0.670397 | 0.670397 | -0.126423 | 0.298410 |
| 6.4    | 0.721788 | 0.043700  | 0.678088 | 0.678088 | -0.078337 | 0.276212 |
| 6.6    | 0.733904 | 0.046659  | 0.687244 | 0.687244 | -0.044949 | 0.266096 |
| 6.8    | 0.740502 | 0.044114  | 0.696388 | 0.696388 | -0.022609 | 0.259498 |
| 7.0    | 0.743483 | 0.039776  | 0.704706 | 0.704706 | -0.008328 | 0.256517 |
| 7.2    | 0.744212 | 0.032382  | 0.711829 | 0.711829 | 0.000253  | 0.255788 |
| 7.4    | 0.743640 | 0.025980  | 0.717660 | 0.717660 | 0.004940  | 0.256360 |
| 7.6    | 0.742405 | 0.020145  | 0.722260 | 0.722260 | 0.007071  | 0.257595 |
| 7.8    | 0.740917 | 0.015143  | 0.725774 | 0.725774 | 0.007608  | 0.259083 |
| 8.0    | 0.739423 | 0.011045  | 0.728378 | 0.728378 | 0.007224  | 0.260577 |
| 8.2    | 0.738059 | 0.007809  | 0.730250 | 0.730250 | 0.006367  | 0.261941 |
| 8.4    | 0.736888 | 0.005336  | 0.731552 | 0.731552 | 0.005328  | 0.263112 |
| 8.6    | 0.735928 | 0.003581  | 0.732427 | 0.732427 | 0.004262  | 0.264072 |
| 8.8    | 0.735169 | 0.002182  | 0.732987 | 0.732987 | 0.003325  | 0.264831 |
| 9.0    | 0.734589 | 0.001263  | 0.733326 | 0.733326 | 0.002503  | 0.265411 |
| 9.2    | 0.734158 | 0.000646  | 0.733512 | 0.733512 | 0.001820  | 0.265842 |
| 9.4    | 0.733848 | 0.000249  | 0.733599 | 0.733599 | 0.001295  | 0.266152 |
| 9.6    | 0.733632 | 0.000009  | 0.733622 | 0.733622 | 0.000886  | 0.266368 |
| 9.8    | 0.733487 | -0.000123 | 0.733610 | 0.733610 | 0.000583  | 0.266514 |
| 10.0   | 0.733393 | -0.000185 | 0.733578 | 0.733578 | 0.000364  | 0.266607 |

Tabla 4.7 Resultados del ejemplo 5.

Fig. 4.21 CONTROL DE CALIFICACIONES

CALIFICACIONES vs. TIEMPO

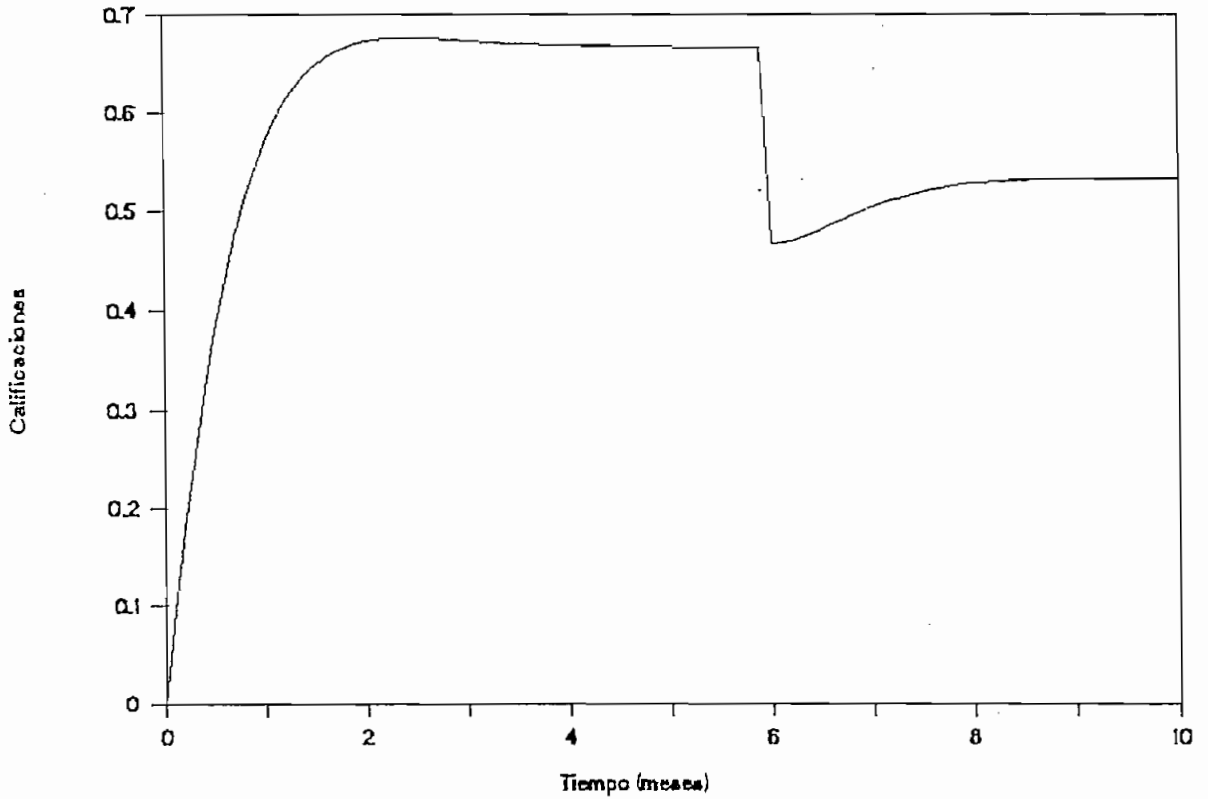
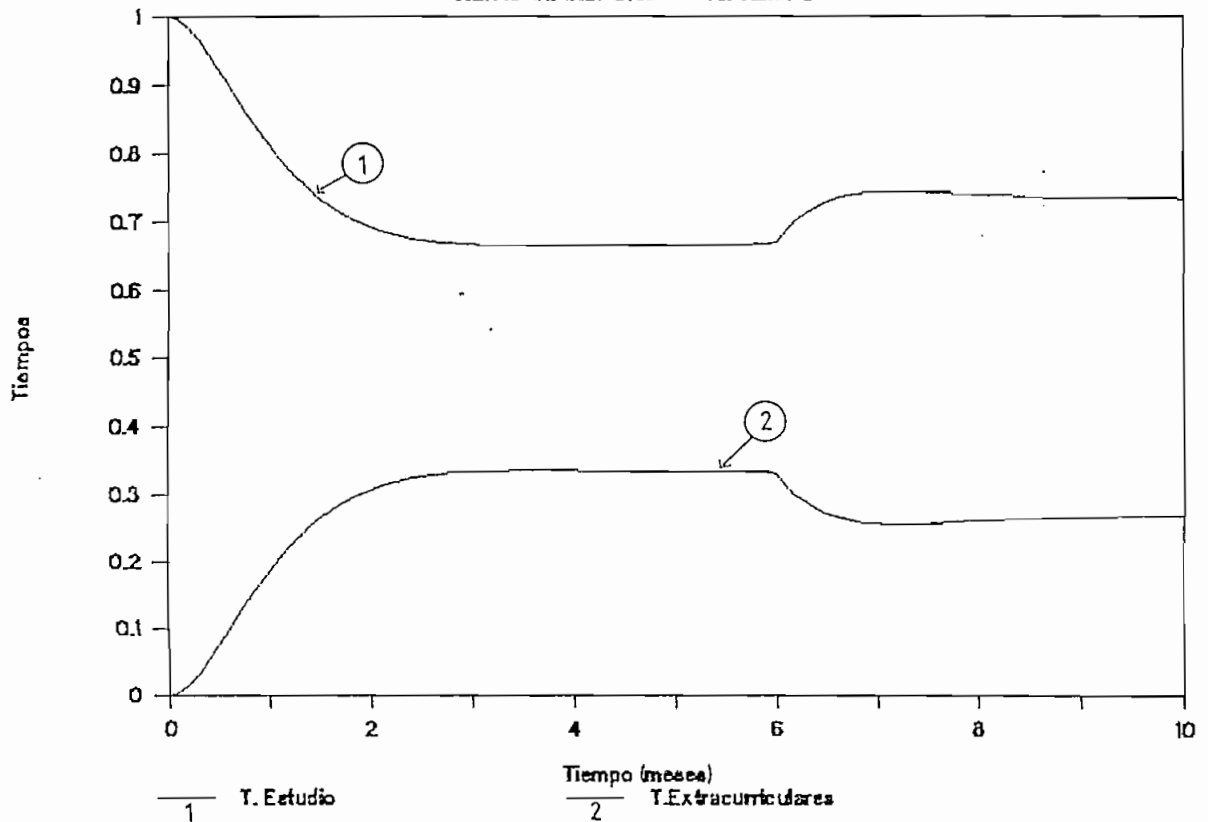


Fig. 4.22 CONTROL DE CALIFICACIONES

TIEMPOS DISPONIBLES vs. TIEMPO



turbación es

$$T_D(s) = Y(s)/D(s) = -(s^2+3s+2)/(s^2+3s+3) \quad (4.7)$$

para  $D(s)=0.2/s$  el teorema del valor final da  $-0.13333$ , es decir se espera que luego de la perturbación el valor final de las calificaciones sea de  $0.66666-0.13333=0.53333$ . Nuevamente el valor analítico concuerda con el obtenido en el programa.

El propósito fundamental del ejemplo es demostrar la capacidad que tiene el programa para generar funciones de entrada. Siendo introducidas como líneas de programa BASIC, prácticamente no existen limitaciones en cuanto a la generación de estas funciones. En el Anexo A se presentan algunos ejemplos que demuestran lo dicho. Para el caso del ejemplo, las líneas de programa BASIC introducidas para generar las funciones de entrada son

```
5000 U(1)=1
5010 IF T<6 THEN U(2)=0 ELSE U(2)=-0.2
```

#### 4.1.6 Ejemplo 6. Circuito generador de funciones.

El circuito de la Fig. 4.23 es utilizado para el disparo de tiristores. Consiste de un rectificador de onda completa formado por los diodos  $D_1$ , un recortador formado por  $R_1$  y  $D_2$ , la fuente  $V_{cc}$  se consigue con  $D_3$  y  $C_1$  y un timer 555 conectado como monoestable.

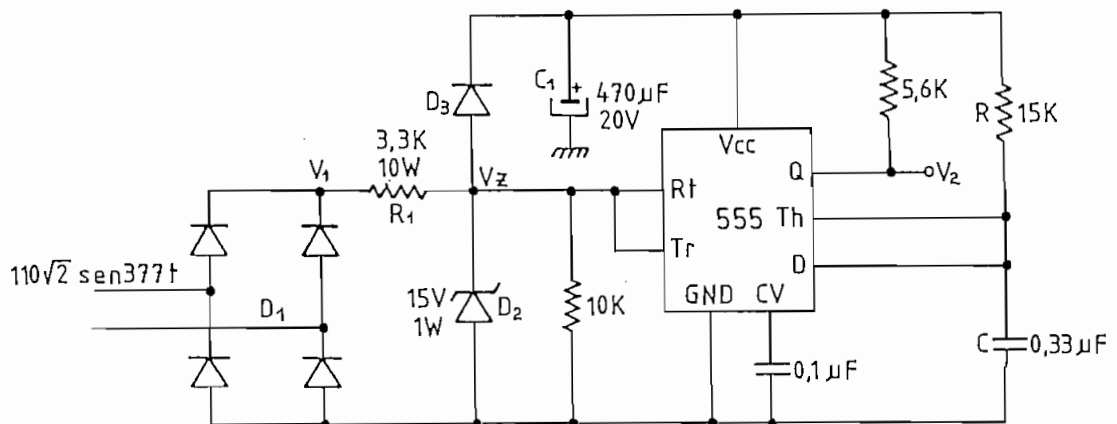


Fig. 4.23 Circuito para disparo de tiristores.

Los valores de los parámetros son típicos para el control de fase en corriente alterna de  $110V_{RMS}$  y  $60Hz$ . De acuerdo a los valores de  $R$  y  $C$ , el tiempo de pulso del monoestable es  $5.44$  ms. En la Fig. 4.24 se muestran las formas de onda del circuito.

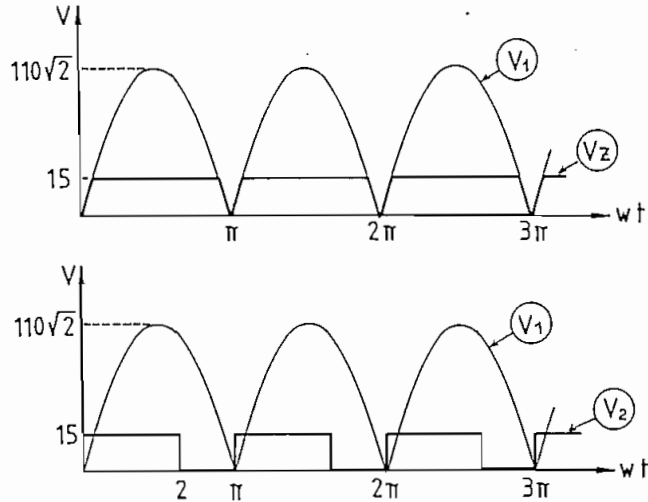


Fig. 4.24 Formas de onda del circuito de la Fig. 4.23.

Con el fin de demostrar el funcionamiento de los módulos limitador y monoestable, vamos a obtener por simulación las formas de onda de la Fig. 4.24. Debido a que los tiempos involucrados en el problema son muy pequeños y las magnitudes muy grandes, lo que puede ocasionar errores en los cálculos, vamos a escalar en magnitud y en tiempo. Llamamos  $t_1=wt=377t$  y a la magnitud la escalamos en un factor  $k$  de  $0.06428$ , de forma que la magnitud máxima de la señal de entrada es de  $10$ . Con este escalamiento, el tiempo de pulso de monoestable será de  $2$  y la amplitud de  $V_2$  será  $0.964$ . El diagrama de simulación para el problema se muestra en la Fig. 4.25.

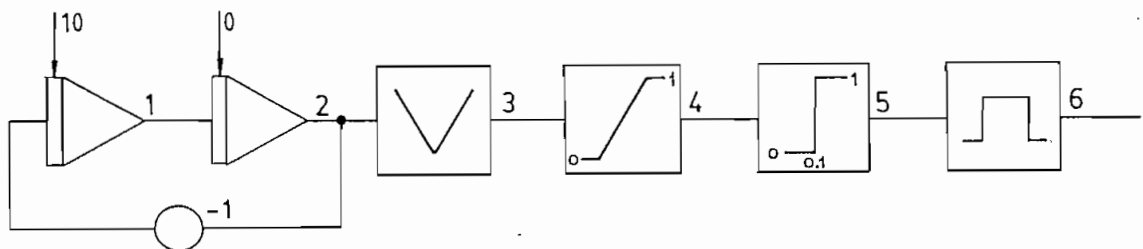


Fig. 4.25 Diagrama de simulación del ejemplo 6.



En el diagrama de simulación, los integradores de salidas 1 y 2 permiten generar la función de entrada  $10\text{sen}(t_1)$ , el módulo de salida 3 simula el comportamiento de los rectificadores  $D_1$ , el limitador de salida 4 simula el efecto del zener  $D_2$ , el comparador de salida 5 ajusta los valores del cruce por cero a valores lógicos que son aceptados en la señal de disparo del monoestable y el módulo de salida 6 simula el trabajo del monoestable.

El programa se corre para un tiempo inicial de 0, un tiempo final de 10 y un intervalo de integración de 0.1.

En la Fig. 4.26 se muestra el gráfico de las señales  $V_1$  ( $y_3$  en el diagrama de simulación) y  $V_z$  ( $y_4$ ), en función del tiempo. La Fig. 4.27 muestra las señales de  $V_1$  y  $V_2$  ( $y_6$  en el diagrama de simulación). Como se puede observar, las señales obtenidas concuerdan plenamente con los mostrados en la Fig. 4.24.

Este ejemplo, corrido para los valores sin escalamiento, requiere de un intervalo de integración bastante pequeño para evitar errores considerables en los cálculos, lo que significa un gran tiempo de ejecución. Esta es una buena razón para recurrir a los escalamientos en tiempo y magnitud.

#### 4.1.7 Ejemplo 7. Detector de secuencia.

El circuito digital mostrado en la Fig. 4.28(a) es utilizado como detector de secuencia en sistemas trifásicos. Las señales  $x_1$ ,  $x_2$  y  $x_3$  detectan el cruce por cero de cada una de las fases y generan un pulso al acerlo. Si los pulsos son aplicados al sistema en el orden  $x_1-x_2-x_3$ , en la salida Q se obtiene un tren de pulsos que indica que la secuencia de fases es positiva; si el orden de los pulsos es  $x_1-x_3-x_2$ , la salida Q permanece en cero indicando que la secuencia de fases es negativa. En las Figs. 4.28(b) y 4.28(c) se muestran la forma de las señales de entrada y salida del sistema para los dos casos.

Fig. 4.26 GENERADOR DE FUNCIONES

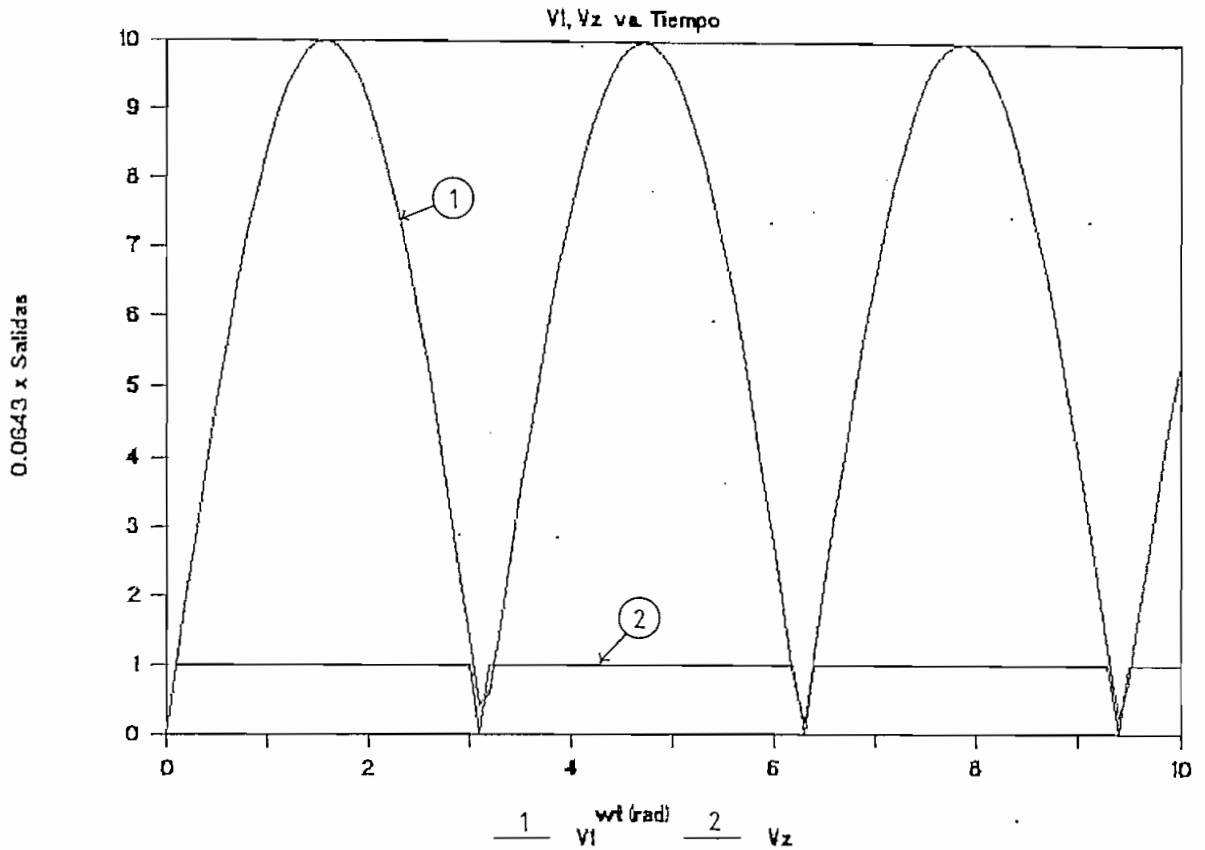
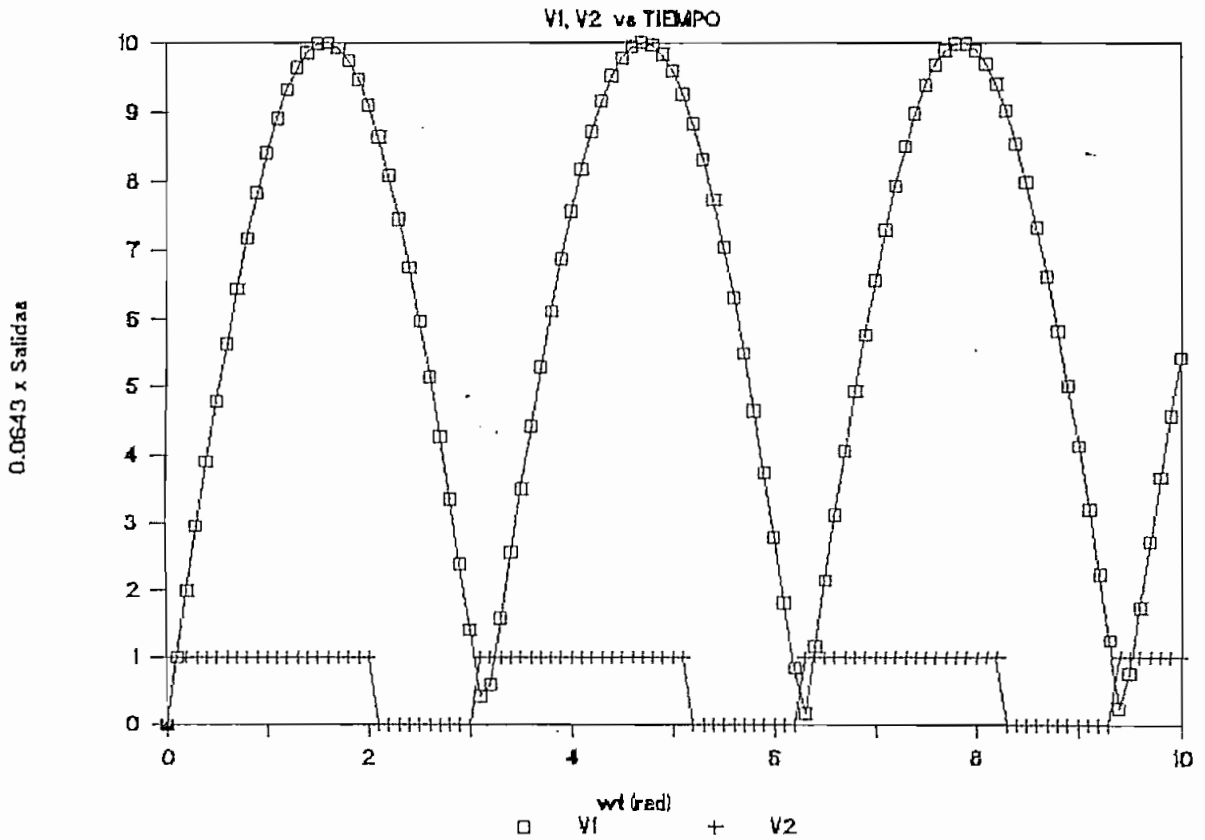


Fig. 4.27 GENERADOR DE FUNCIONES



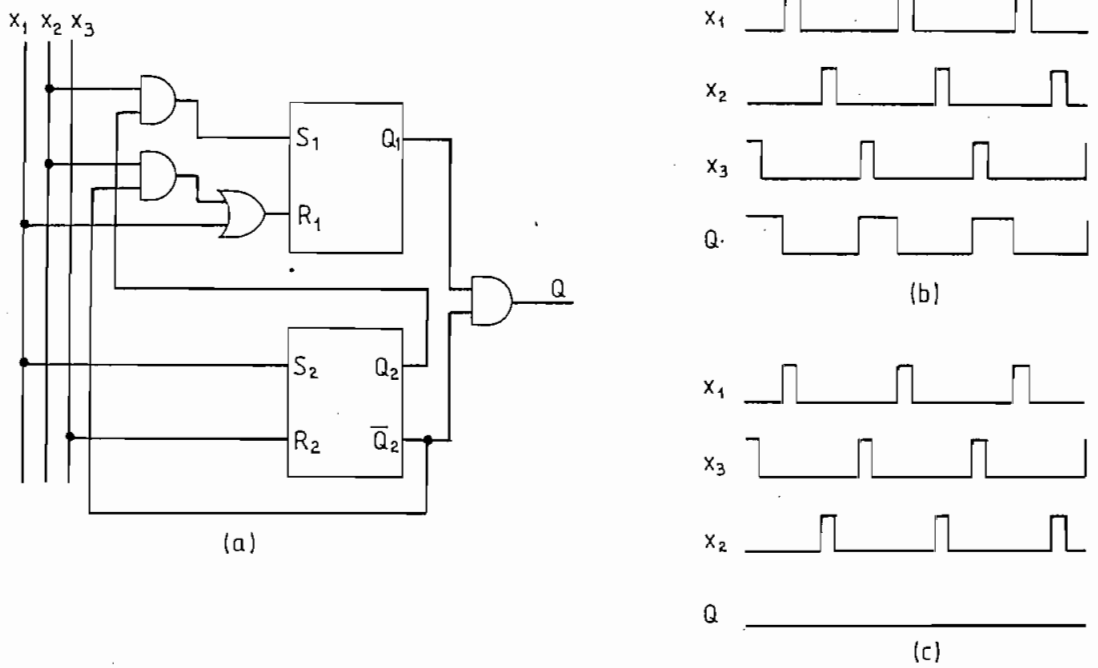


Fig. 4.28 a) Circuito detector de secuencia.  
 b) Formas de onda para secuencia positiva.  
 c) Formas de onda para secuencia negativa.

El obtener por simulación las formas de onda de la Fig. 4.28, nos permitirá observar el comportamiento de los módulos digitales disponibles en el programa. El diagrama de simulación del sistema se muestra en la Fig. 4.29.

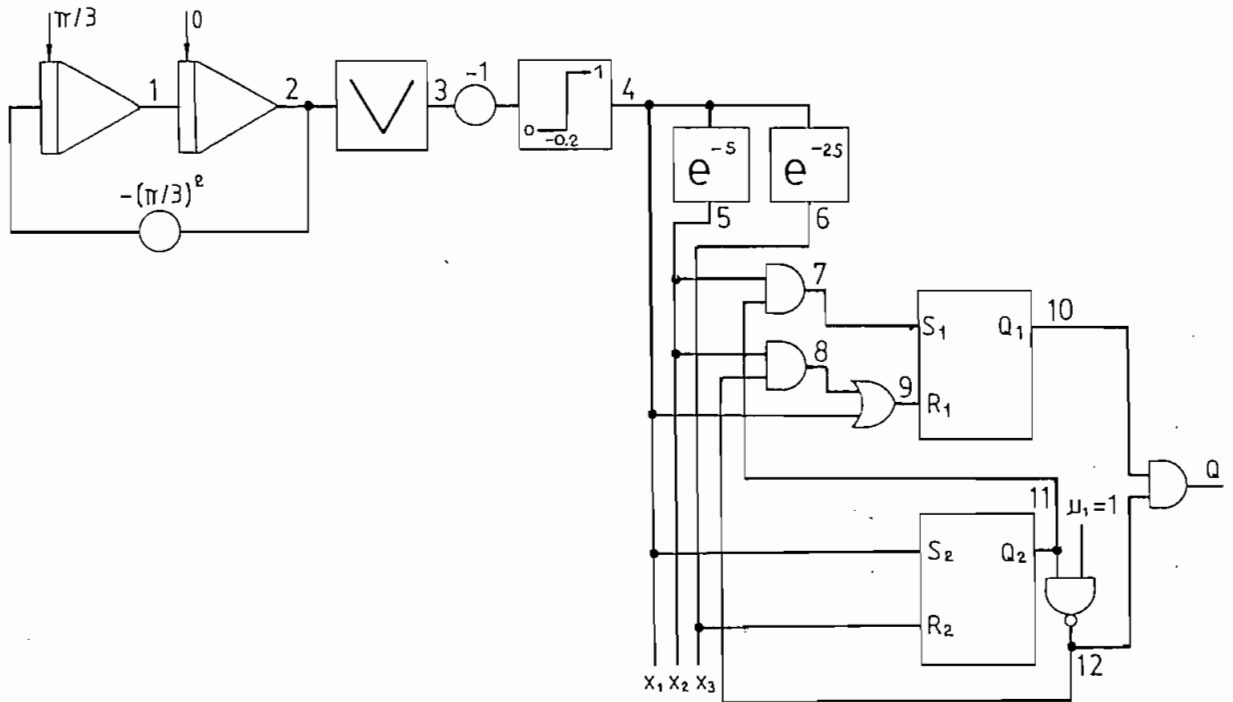


Fig. 4.29 Diagrama de simulación del circuito de la Fig. 4.28.

|      |           |           |          |   |   |   |   |   |   |   |   |   |   |
|------|-----------|-----------|----------|---|---|---|---|---|---|---|---|---|---|
| 10.0 | -0.523613 | -0.866022 | 0.866022 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
|------|-----------|-----------|----------|---|---|---|---|---|---|---|---|---|---|

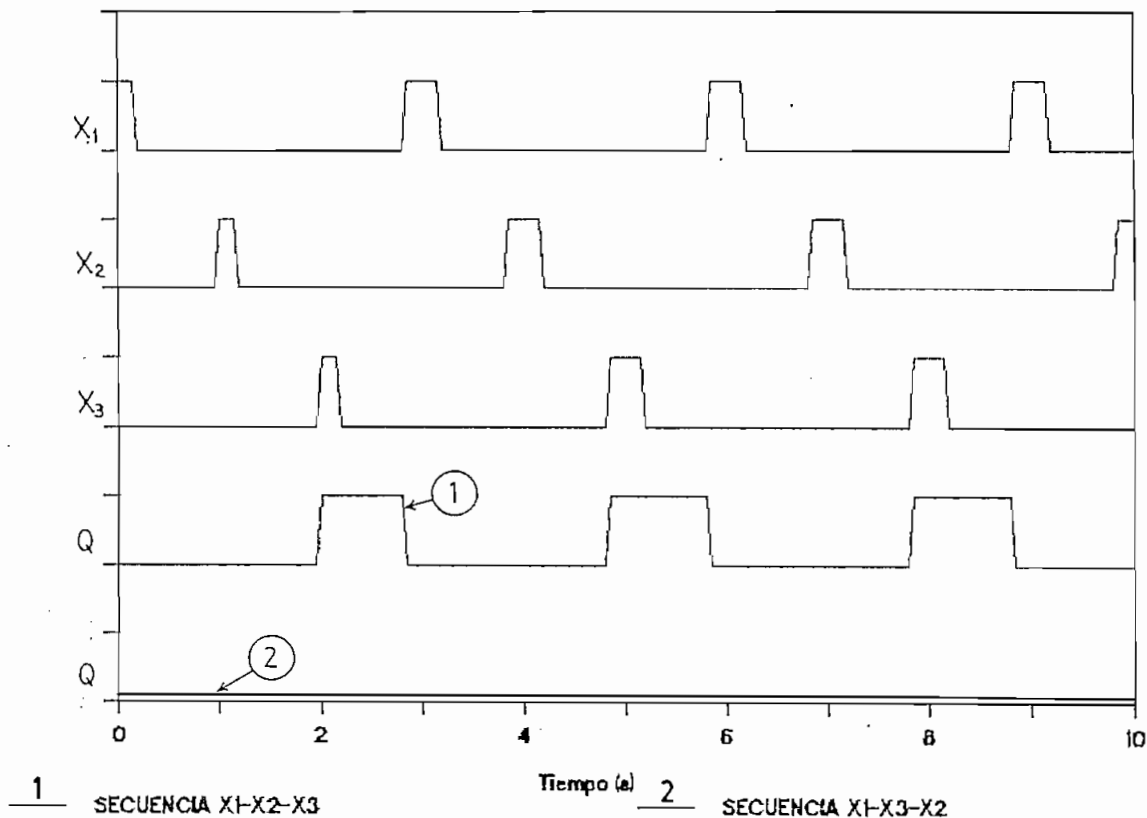
Tabla 4.8 Resultados del ejemplo 7 para secuencia positiva.

| TIEMPO | SALIDAS:  |           |          |   |   |   |   |   |   |    |    |    |    |
|--------|-----------|-----------|----------|---|---|---|---|---|---|----|----|----|----|
|        | 1         | 2         | 3        | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 0.0    | 1.047200  | 0.000000  | 0.000000 | 1 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 0  |
| 0.2    | 1.024316  | 0.207912  | 0.207912 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 0.4    | 0.956665  | 0.406730  | 0.406730 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 0.6    | 0.847203  | 0.587787  | 0.587787 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 0.8    | 0.700714  | 0.743147  | 0.743147 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 1.0    | 0.523601  | 0.866028  | 0.866028 | 0 | 0 | 1 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 1.2    | 0.323604  | 0.951059  | 0.951059 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 1.4    | 0.109464  | 0.994525  | 0.994525 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 1.6    | -0.109460 | 0.994526  | 0.994526 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 1.8    | -0.323600 | 0.951061  | 0.951061 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 2.0    | -0.523598 | 0.866030  | 0.866030 | 0 | 1 | 0 | 0 | 1 | 1 | 0  | 0  | 1  | 0  |
| 2.2    | -0.700711 | 0.743149  | 0.743149 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 2.4    | -0.847201 | 0.587790  | 0.587790 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 2.6    | -0.956663 | 0.406741  | 0.406741 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 2.8    | -1.024315 | 0.207916  | 0.207916 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 3.0    | -1.047200 | 0.000004  | 0.000004 | 1 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 0  |
| 3.2    | -1.024317 | -0.207909 | 0.207909 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 3.4    | -0.956667 | -0.406734 | 0.406734 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 3.6    | -0.847206 | -0.587783 | 0.587783 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 3.8    | -0.700718 | -0.743144 | 0.743144 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 4.0    | -0.523605 | -0.866026 | 0.866026 | 0 | 0 | 1 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 4.2    | -0.323608 | -0.951058 | 0.951058 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 4.4    | -0.109469 | -0.994525 | 0.994525 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 4.6    | 0.109456  | -0.994526 | 0.994526 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 4.8    | 0.323596  | -0.951062 | 0.951062 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 5.0    | 0.523594  | -0.866032 | 0.866032 | 0 | 1 | 0 | 0 | 1 | 1 | 0  | 0  | 1  | 0  |
| 5.2    | 0.700708  | -0.743152 | 0.743152 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 5.4    | 0.847198  | -0.587793 | 0.587793 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 5.6    | 0.956662  | -0.406745 | 0.406745 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 5.8    | 1.024314  | -0.207920 | 0.207920 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 6.0    | 1.047200  | -0.000008 | 0.000008 | 1 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 0  |
| 6.2    | 1.024318  | 0.207904  | 0.207904 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 6.4    | 0.956669  | 0.406730  | 0.406730 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 6.6    | 0.847208  | 0.587780  | 0.587780 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 6.8    | 0.700721  | 0.743141  | 0.743141 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 7.0    | 0.523609  | 0.866024  | 0.866024 | 0 | 0 | 1 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 7.2    | 0.323613  | 0.951057  | 0.951057 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 7.4    | 0.109473  | 0.994524  | 0.994524 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 7.6    | -0.109451 | 0.994527  | 0.994527 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 7.8    | -0.323592 | 0.951063  | 0.951063 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 8.0    | -0.523590 | 0.866034  | 0.866034 | 0 | 1 | 0 | 0 | 1 | 1 | 0  | 0  | 1  | 0  |
| 8.2    | -0.700705 | 0.743155  | 0.743155 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 8.4    | -0.847196 | 0.587797  | 0.587797 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 8.6    | -0.956660 | 0.406749  | 0.406749 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 8.8    | -1.024313 | 0.207924  | 0.207924 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |
| 9.0    | -1.047200 | 0.000012  | 0.000012 | 1 | 0 | 0 | 0 | 0 | 1 | 0  | 1  | 0  | 0  |
| 9.2    | -1.024319 | -0.207900 | 0.207900 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 9.4    | -0.956670 | -0.406726 | 0.406726 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 9.6    | -0.847211 | -0.587777 | 0.587777 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 9.8    | -0.700724 | -0.743139 | 0.743139 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  |
| 10.0   | -0.523613 | -0.866022 | 0.866022 | 0 | 0 | 1 | 0 | 0 | 0 | 0  | 0  | 1  | 0  |

Tabla 4.9 Resultados del ejemplo 7 para secuencia negativa.

Fig. 4.30 DETECTOR DE SECUENCIA

ENTRADAS, SALIDAS vs. TIEMPO



#### 4.2 CONCLUSIONES

El programa implementado permite la simulación de gran cantidad de sistemas sean estos lineales, no lineales, de una o múltiples entradas. En general, simula la gran mayoría de las bondades de un computador análogo de gran capacidad, sofisticado equipo de medida y alta precisión. La principal desventaja frente a este es el mayor tiempo requerido para la ejecución.

El diagrama de simulación puede tener cualquier configuración serie-paralelo entre los módulos disponibles, salvo por la limitación de que la estructura del diagrama no lleve a sistemas de ecuaciones lineales, no lineales o trascendentes al relacionar las diferentes salidas.

Los módulos disponibles en el programa cubren gran cantidad de funciones lineales, no lineales y digitales necesarias

... lenguaje GW-BASIC COMPILADO, utilizado para implementar el programa da magníficos resultados en lo referente a presentación y velocidad de ejecución.

En un computador análogo, mediante el modo de operación HOLD, se puede interrumpir el proceso de simulación y acceder a información intermedia. El programa digital implementado no permite esta facilidad.

En resumen, el programa SDSC se convierte en una poderosa herramienta para la solución de ecuaciones diferenciales, en el cual, la entrada de datos se generaliza gracias a que se fundamenta en un diagrama de simulación. Adicionalmente, se

ecuaciones diferenciales considerándolas como un conjunto de "módulos o bloques matemáticos" debidamente interconectados.

#### 4.3 RECOMENDACIONES

Es recomendable que el usuario tenga conocimientos básicos sobre el sistema operativo, programación en lenguaje BASIC y el manejo de la hoja electrónica LOTUS 123 antes de utilizar el programa.

Lo fundamental dentro de la entrada de datos de un problema, es el diagrama de simulación y la matriz de incidencia, por tanto, se debe poner especial cuidado en la delimitación de estos requerimientos.

Antes de ejecutar un problema de simulación se debe prever posibles errores de cálculo como divisiones por cero, raíces de números negativos, señales digitales fuera de rango, etc. ya que de presentarse estas fallas durante la ejecución el proceso se interrumpe con la posibilidad de que se pierdan los datos, y tengan que ser introducidos nuevamente.

En algunos problemas es difícil estimar el intervalo de integración adecuado, para estos casos, se sugiere evaluar el diagrama para dos intervalos de integración diferentes y comparar los resultados. Si los resultados de una y otra evaluación concuerdan o tienen una diferencia relativa baja, se asume que los intervalos asumidos son correctos. Si los resultados varían sustancialmente, será necesario escoger un intervalo más pequeño para la integración.

Cuando se trabaja en equipos con dos unidades de discos flexibles, el disco del programa deberá estar desprotegido para escritura, de forma que, errores de operación podrían ocasionar el daño del mismo; por esta razón, se recomienda tener un respaldo del programa.

Un trabajo adicional podría consistir en dotar al programa

de subrutinas capaces de evaluar sistemas de ecuaciones. Esto puede realizarse buscando un procedimiento que a partir del diagrama de simulación genere las ecuaciones del sistema y las evalúe para cada instante de tiempo. El proceso necesariamente incrementará en forma considerable el tiempo de ejecución.

Es importante desarrollar un paquete de programas afines que permitan el análisis y diseño de sistemas con la ayuda de otras técnicas como lugar geométrico de las raíces, análisis en el dominio de la frecuencia, análisis de estabilidad absoluta, etc.

#### REFERENCIAS

1. ARTHUR HAUSNER, *Analog and Analog/Hybrid Computer Programming*. Prentice-Hall International, 1971; pp. 72.
2. RICHARD C. DORF, *Sistemas Automáticos de Control*. Fondo Educativo Interamericano S.A., 1976, pp. 122-123.

ANEXO A



```
BASIC.EXE
BASRUN.EXE
PRES.BAS
ENT.EXE
SOL.EXE
RESULTAD.BAT
KEY-FAKE.COM
MENU.WK1
LOTUS.COM
123.EXE
123.SET
123.CMP
123.CNF
```

Todos los archivos corresponden al programa SDSC con excepción de los tres últimos que se copiarán del programa principal de LOTUS instalado para el computador.

Con la ayuda del programa EDLIN.COM del sistema operativo modificar el archivo de proceso por lotes RESULTAD.BAT de forma que tenga la siguiente configuración:

```
ECHO OFF
KEY-FAKE 0 13 0 13 0 13 0 13 0 13 "/FR" 0 13 "MENU.WK1" 13
LOTUS
```

En la raíz del disco C, crear un archivo de proceso por lotes llamado SDSC.BAT cuyo listado es:

```
ECHO OFF
CD SDSC
BASIC PRES
CD\
```

Se recomienda arrancar el programa desde la unidad C, con el disco de datos en la unidad A del sistema.

A.1.3 Equipo con una unidad de disco rígido en la cual se dispone del programa LOTUS 123.

Supongamos que el directorio que contiene el programa LOTUS 123 se llama LOTUS.

Crear un directorio de nombre SDSC en la unidad de disco rígido C. En este directorio se copian los archivos:

```
BASIC.EXE
BASRUN.EXE
PRES.BAS
ENT.EXE
SOL.EXE
RESULTAD.BAT
```

El archivo RESULTAD.BAT tiene ahora la siguiente configuración:

```
ECHO OFF
PATH C:\LOTUS
KEY-FAKE 0 13 0 13 0 13 0 13 0 13 "/FRC:\LOTUS\MENU.WK1"13
LOTUS
```

Copiar en el directorio LOTUS los archivos MENU.WK1 y KEY-FAKE.COM.

En la raíz del disco C, crear el archivo SDSC.BAT con la siguiente estructura:

```
ECHO OFF
CD SDSC
BASIC PRES
CD\
```

## A.2 UTILIZACION

El programa arranca desde la unidad A, con el disco de datos en la unidad B, en el caso de que se tenga un sistema sin disco fijo y desde la unidad C con el disco de datos en la unidad A si es que se dispone de disco rígido.

El acceso al programa se lo hace desde la unidad que lo contiene, digitando la secuencia SDSC [Enter]. Luego de unos pocos segundos asoma una pantalla de presentación con la indicación de que se presione cualquier tecla para continuar. Presionada una tecla, se muestra en la pantalla el menú principal del programa con las siguientes opciones:

1. Problema nuevo
2. Problema con datos cambiados
3. Revisar un archivo de datos
4. Terminar

La opción 1. permite evaluar un diagrama de simulación cuya estructura no ha sido analizada anteriormente. La opción 2. permite evaluar un diagrama de simulación cuyos datos ya han sido introducidos pero en el cual se ha cambiado uno o más parámetros. La opción tres permite revisar los datos y resultados de un problema resuelto y la opción 4. se la utiliza para abandonar el proceso y regresar el control al sistema operativo.

#### A.2.1 Entrada de datos.

La entrada de datos de un problema por analizar se basa en la conformación de un diagrama de simulación analógica cuyas características más importantes son:

- ningún módulo presente en el diagrama invierte la señal,
- las constantes involucradas pueden tener cualquier valor numérico,
- no se debe exceder el número de entradas permisibles para cada uno de los módulos,
- los módulos disponibles son:

#### FUNCIONES LINEALES

| MODULO               | N. DE ENTRADAS | FIGURA |
|----------------------|----------------|--------|
| Sumador              | 2 o más        | 2.1b   |
| Integrador           | 1              | 2.1a   |
| Retardo en el tiempo | 1              | 2.22   |

#### FUNCIONES NO LINEALES

| MODULO                     | N. DE ENTRADAS | FIGURA |
|----------------------------|----------------|--------|
| Limitador                  | 1              | 2.2    |
| Comparador                 | 1              | 2.3    |
| Comparador con Histéresis  | 1              | 2.4    |
| Rectificador de Media Onda | 1              | 2.6    |
| Valor Absoluto             | 1              | 2.7    |
| Zona Muerta                | 1              | 2.8    |
| Alinealidad de Juego       | 1              | 2.9    |
| Cuadrador                  | 1              | 2.10   |
| Raíz Cuadrada              | 1              | 2.11   |
| Logaritmo Natural          | 1              | 2.12   |
| Exponencial                | 1              | 2.13   |
| Seno                       | 1              | 2.14   |
| Coseno                     | 1              | 2.15   |

FUNCIONES LOGICAS

| MODULO         | N. DE ENTRADAS | FIGURA |
|----------------|----------------|--------|
| Compuerta AND  | 2 o más        | 2.18   |
| Compuerta NAND | 2 o más        | 2.18   |
| Compuerta OR   | 2 o más        | 2.18   |
| Compuerta NOR  | 2 o más        | 2.18   |
| Biestable SR   | 2              | 2.19   |
| Monoestable    | 1              | 2.20   |

OTRAS FUNCIONES

| MODULO         | N. DE ENTRADAS | FIGURA |
|----------------|----------------|--------|
| Multiplicador  | 2 o más        | 2.16   |
| Divisor        | 2              | 2.17   |
| Switch Análogo | 2              | 2.21   |

- todas las entradas son afectadas por un potenciómetro antes de llegar al módulo correspondiente
- las entradas externas del sistema pueden ser aplicadas solamente a módulos que acepten más de una entrada
- la salida de cada uno de los módulos del diagrama se considera una salida del sistema y deben ir numeradas en forma continua
- el diagrama de simulación no debe tener configuraciones que lleven a sistemas de ecuaciones lineales o trascendentes cuando se relacionan las diferentes salidas y debe incluir al menos un integrador (sistema dinámico). Salvando esta excepción, el diagrama puede tener cualquier combinación serie-paralelo entre los diferentes módulos
- las entradas externas del diagrama son llamadas  $u_i(t)$ ; para  $i=1....l$ , donde  $l$  es el número total de entradas
- las características de cada uno de los módulos disponibles se las puede observar en el literal 2.1 del presente trabajo, en la figura correspondiente
- las señales lógicas aceptan como  $1L$  valores comprendidos entre  $0.8u$  y  $1.0u$  y como  $0L$  valores comprendidos entre  $0.0u$  y  $0.2u$ .

Para ilustrar la forma de introducir los datos de un problema nuevo por analizar, consideremos el sistema no lineal de segundo orden cuyo diagrama de bloques se muestra en la Fig. A.1.

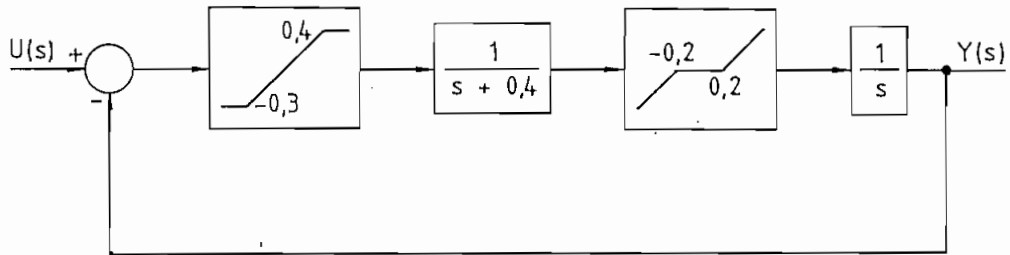


Fig. A.1. Diagrama de bloques de un sistema no lineal de segundo orden.

Para simular el comportamiento del sistema ante una perturbación escalón unitaria en la entrada, debemos obtener el diagrama de simulación del mismo. Asumiremos que las condiciones iniciales del sistema son nulas. El diagrama de simulación adecuado para la introducción de los datos en el computador digital se muestra en la Fig. A.2.

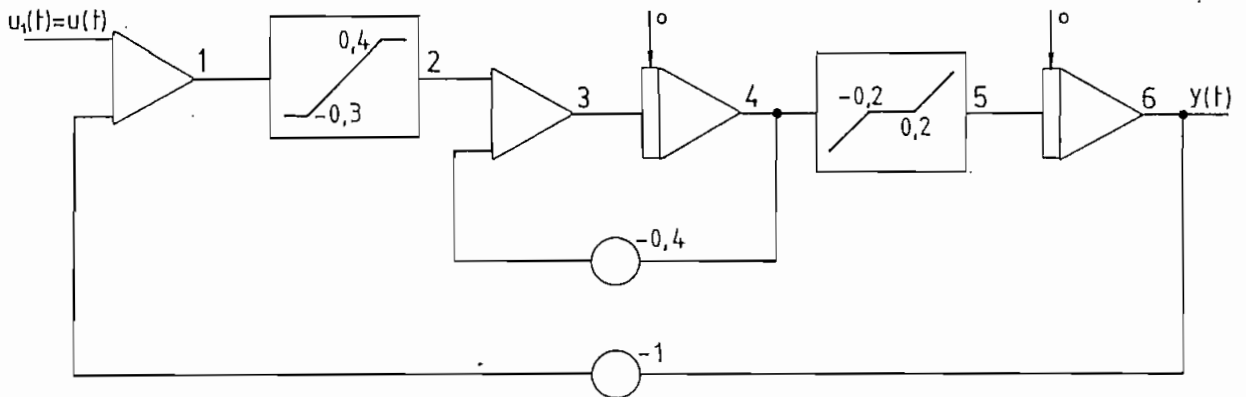


Fig. A.2. Diagrama de simulación del sistema.

Notemos que ningún módulo invierte la señal, los valores de las constantes pueden ser negativas, los módulos zona muerta y limitador pueden ser simulados directamente ya que están disponibles en el programa, hemos omitido el dibujo de los potenciómetros que se encuentran asociados a la entrada de los módulos ya que tienen un valor unitario, la salida de cada uno de los módulos presentes ha sido numerada, el orden que se



En la parte superior izquierda de la pantalla se presenta la pregunta "Existe ramal desde 1 hasta 2 (1=S 0=N)". Si es que existe algún módulo entre los puntos 1 y 2 siendo 1 la entrada del módulo y 2 su salida la respuesta será afirmativa, es decir, se debe digitar un 1. En cualquier otro caso la respuesta es negativa, es decir, se digitará un 0. En el ejemplo los ramales existentes son:

```
desde 1 hasta 2
desde 2 hasta 3
desde 3 hasta 4
desde 4 hasta 3
desde 4 hasta 5
desde 5 hasta 6
desde 6 hasta 1
```

notamos que desde 2 hasta 3 y desde 4 hasta 3 existe un mismo módulo (sumador). La última fila de la matriz de incidencia lleva los datos sobre la incidencia de las entradas externas, es decir, si existe una entrada que es aplicada al módulo cuya salida es  $j$  el término de la última fila y de la columna  $j$  llevará un 1. Para el caso del ejemplo, existe una entrada aplicada al sumador de salida 1 lo que indica que ante la pregunta "Existe ramal desde 7 hasta 1 (1=S 0=N)" se debe responder digitando 1. Al terminar la introducción de los datos de la matriz, el programa pregunta "Es correcta la definición (S/N)", si la respuesta es afirmativa, se pasa al siguiente punto, en caso contrario, se puede cambiar cualquier elemento de la matriz digitando los datos de la fila y columna del valor por cambiar. El proceso de corrección termina cuando se digita 0,0. La forma final de la matriz de incidencia es

```

      0 1 0 0 0 0
      0 0 1 0 0 0
      0 0 0 1 0 0
      0 0 1 0 1 0
      0 0 0 0 0 1
      1 0 0 0 0 0
Entradas --> 1 0 0 0 0 0
```

Terminada la introducción de los datos de la matriz de incidencia, el programa analiza la configuración de la misma y determina cuales son los módulos que tienen más de una entrada

y cuales los que tienen una sola entrada, adicionalmente si es que existe una entrada externa aplicada a un módulo que acepta una sola entrada, el programa pide nuevamente la definición dando la indicación de error respectiva.

A partir de este momento, todos los datos correspondientes a los módulos existentes en el diagrama se introducen al computador con la ayuda de menús y consultas sobre la pantalla lo cual facilita mucho la introducción de los mismos. Por tanto, lo fundamental dentro de la entrada de datos es obtener un buen diagrama de simulación y definir correctamente la matriz de incidencia.

El proceso continúa con la definición de los módulos de que aceptan más de una entrada, para la cual se presenta en la pantalla el menú correspondiente con las siguientes opciones:

1. Sumador
2. Multiplicador
3. Divisor
4. Sw. Análogo
5. Compuerta AND
6. Compuerta NAND
7. Compuerta OR
8. Compuerta NOR
9. Biestable SR

Con el menú en la pantalla se realizan las consultas que detallamos a continuación.

Código del módulo. Digitar el número asignado al módulo correspondiente dentro del menú.

Valores de los potenciómetros que afectan a las entradas. Digitar los valores numéricos correspondientes según el diagrama de simulación.

Datos adicionales. En el caso de que el módulo seleccionado sea un Sw. Análogo (Cod. 4), se pregunta cual de las entradas es la que controla la acción del interruptor, esta entrada es de tipo lógico. En el caso del Biestable SR (Cód. 9), se pregunta cual de las entradas está aplicada a la entrada S (Set).

Terminada la introducción de datos de un módulo, se

presenta en la pantalla la indicación sobre la definición dada junto con la consulta "Es correcta la definición (S/N)?" De ser afirmativa la respuesta, se continúa con la siguiente definición, en caso contrario se procede a introducir nuevamente todos los datos sobre el módulo.

Las preguntas y respuestas para el caso del ejemplo son:

Código del módulo cuya salida es 1? 1  
Valor del potenciómetro desde 6 hasta el módulo? -1



presenta en la pantalla la indicación sobre la definición dada junto con la consulta "Es correcta la definición (S/N)?". De ser afirmativa la respuesta, se continúa con la siguiente definición, en caso contrario se procede a introducir nuevamente todos los datos sobre el módulo.

Las preguntas y respuestas para el caso del ejemplo son:

Código del módulo cuya salida es 1? 1  
Valor del potenciómetro desde 6 hasta el módulo? -1

Definición del módulo 1

El módulo es un sumador  
Desde Valor del potenciómetro  
6 -1

Es correcta la definición (S/N)? s

Código del módulo cuya salida es 3? 1  
Valor del potenciómetro desde 2 hasta el módulo? 1  
Valor del potenciómetro desde 4 hasta el módulo? -.4

Definición del módulo 3

El módulo es un sumador  
Desde Valor del potenciómetro  
2 1  
4 -.4

Es correcta la definición (S/N)? s

Para la definición de los módulos de una sola entrada se muestra en la pantalla el siguiente menú

- |                               |                          |
|-------------------------------|--------------------------|
| 1. Integrador                 | 9. Cuadrador             |
| 2. Limitador                  | 10. Raíz Cuadrada        |
| 3. Comparador                 | 11. Logaritmo Natural    |
| 4. Comparador con Histéresis  | 12. Exponencial          |
| 5. Rectificador de media onda | 13. Seno                 |
| 6. Valor Absoluto             | 14. Coseno               |
| 7. Zona muerta                | 15. Retardo en el tiempo |
| 8. Alinealidad de Juego       | 16. Monogestable         |

Con el menú en la pantalla se realizan las siguientes consultas.

Código del módulo.      Digitar el número correspondiente según el menú.

Valor del potenciómetro que afecta a la entrada.      Digitar

el valor numérico que consta en el diagrama de simulación.

Datos adicionales. Los datos adicionales corresponden a los valores referenciales que definen la función o relación de transferencia del módulo seleccionado. La introducción de estos datos es bastante sencilla ya que se la realiza en base a consultas en la pantalla. Las características de cada uno de los módulos disponibles se muestran claramente en las Figuras 2.1 a 2.22.

Al igual que en el caso anterior, luego de definir un módulo se pregunta si es correcta la definición, si la respuesta es afirmativa se continúa con el siguiente módulo, en caso contrario se pide nuevamente por los datos.

En el caso del ejemplo, las preguntas y respuestas serán:

Código del módulo existente desde 1 hasta 2? 2  
Valor del potenciómetro en la entrada del limitador? 1  
Valor máximo del limitador? .4  
Valor mínimo del limitador? -.3  
Es correcta la definición (S/N)? s

Código del módulo existente desde 3 hasta 4? 1  
Valor del potenciómetro en la entrada del integrador? 1  
Condición inicial del integrador? 0  
Es correcta la definición (S/N)? s

Código del módulo existente desde 4 hasta 5? 7  
Valor del potenciómetro en la entrada de la zona muerta? 1  
Amplitud de la zona muerta? 0.4  
Es correcta la definición (S/N)? s

Código del módulo existente desde 5 hasta 6? 1  
Valor del potenciómetro en la entrada del integrador? 1  
Condición inicial del integrador? 0  
Es correcta la definición (S/N)? s

El siguiente paso es la definición de los subíndices de las entradas externas del diagrama. El programa pregunta sobre el valor del subíndice de una entrada que es aplicada a un módulo. Con el diagrama de simulación es fácil introducir estos datos. El programa no acepta que dos entradas diferentes tengan el mismo subíndice ni subíndices cuyo valor supere el número máximo de entradas externas existentes. Terminada la introducción de los datos, es posible editar valores erróneos introducidos respondiendo en forma negativa ante la consulta de

que si es correcta la definición de los subíndices. Para el ejemplo la consulta es:

Subíndice de la entrada que va al módulo 1? 1

La definición dada es

U1 va al módulo 1

Es correcta la definición (S/N)? s

Finalmente se pide los datos referidos a las magnitudes de tiempo para la evaluación del diagrama. Los datos solicitados son:

Tiempo inicial de ejecución. Se debe digitar el valor del tiempo desde el cual arranca la evaluación del diagrama.

Tiempo final de ejecución. Corresponde al dato de tiempo al cual termina la evaluación del diagrama.

Intervalo de tiempo entre cálculos. Corresponde al paso de integración  $h$ . El valor introducido dependerá de la precisión que se requiera en los resultados. Algunas reglas cualitativas para escoger un valor adecuado son:

- El error de paso estimado es del orden de  $h^4$  para el método de integración utilizado, para minimizar el error se debe escoger un  $h$  suficientemente pequeño pero en contraposición se tiene que mientras más pequeño sea  $h$  mayor es el tiempo que se tarda el computador en evaluar el diagrama,
- $h$  debe ser menor que la menor constante de tiempo involucrada en el problema y
- $h$  debe ser submúltiplo de los tiempos relacionados con el monoestable y el retardo en el tiempo si es que uno de estos módulos o ambos forman parte del diagrama de simulación.

Intervalo de tiempo entre resultados impresos. Se debe digitar el intervalo de tiempo que transcurre entre una impresión de resultados y la siguiente. Este dato debe ser múltiplo de  $h$ .

Hasta este punto, hemos definido toda la estructura del diagrama de simulación y los valores de los parámetros involucrados, faltaría por definir las entradas externas en función del tiempo. Para el efecto, el programa se detiene y

pide la definición de las mismas como líneas de programa BASIC. Esto se hace de forma que no exista limitación en cuanto a la "generación" de señales externas de prueba. Así, cualquier función del tiempo puede ser generada utilizando las funciones de biblioteca y las diferentes instrucciones disponibles en el lenguaje de alto nivel. La definición se hace desde la línea 5000 y puede extenderse como máximo hasta la línea 6000. Cada una de las funciones de entrada es definida como una variable subíndiceada  $U(i)$ , donde  $i$  es el subíndice correspondiente introducido como dato en la definición de los subíndices de las entradas. Al terminar de programar las funciones de entrada el programa debe ser arrancado nuevamente desde la línea 1200, para lo cual se debe digitar la secuencia RUN 1200 [Enter]. El programa presenta en la pantalla las indicaciones respectivas para una correcta definición de las entradas. Vamos a ver algunos ejemplos de definición de entradas para ilustrar el método de generarlas.

Supongamos que en un sistema existen 3 entradas, la primera es un escalón unitario, la segunda tiene una amplitud máxima de 5 y varía con el  $\sin(10t)$  y la última corresponde a un escalón unitario retardado en el tiempo en 4 seg. La definición sería:

```
5000 U(1)=1
5010 U(2)=5*SIN(10*T)
5020 IF T<4 THEN U(3)=0 ELSE U(3)=1
```

Para generar un pulso de amplitud unitaria que inicie en  $t=3$  y termine en  $t=5$  haríamos:

```
5000 IF T<3 OR T>5 THEN U(1)=0: GOTO 5020
5010 U(1)=1
5020 REM
```

Para generar ruido, cuya amplitud máxima sea 2 con una distribución de probabilidad constante podemos utilizar la función RND de la siguiente manera:

```
5000 SIGNO=RND
5010 U(1)=2*RND
5020 IF SIGNO<0.5 THEN U(1)=-U(1)
```

Una función rampa y una función impulso unitario en cero pueden ser generadas de la siguiente manera:

```
5000 REM ***** FUNCION RAMPA *****
5010 U(1)=T
5020 REM ***** FUNCION IMPULSO *****
5030 IF T=0 THEN U(2)=1 ELSE U(2)=0
```

En estos ejemplos se puede observar la versatilidad que se tiene para generar funciones de entrada. En el caso del ejemplo, queremos analizar el comportamiento del sistema ante una entrada escalón unitaria por tanto lo que debemos introducir es:

```
5000 U(1)=1
RUN 1200 [Enter]
```

Cuando existe un error de sintaxis en la definición de las entradas o cuando no se ha digitado las líneas de programa requeridas, el programa pide nuevamente por la definición dando la indicación respectiva sobre el error.

Terminada la introducción de los datos, el programa inicia la evaluación del diagrama tomando para el propósito un tiempo proporcional a la complejidad del problema y al intervalo de tiempo  $h$  asumido. Los resultados finales se guardan en el disco de datos en un archivo tipo PRN que luego puede ser revisado con la opción 3. del menú principal.

#### A.2.2 Revisión de resultados.

Cuando un problema ha sido resuelto, para revisar los resultados, se escoge la opción 3. del menú principal. Luego de escoger esta opción, se carga automáticamente la hoja electrónica LOTUS 123 y sobre esta el programa MENU.WK1, este programa contiene MACRO instrucciones y menús propios que facilitan la revisión de resultados. En este momento, los datos y resultados pueden ser tratados como una hoja construida sobre LOTUS, por tanto, es indispensable que el usuario tenga conocimientos básicos sobre el manejo de esta hoja electrónica.

El análisis de los resultados incluye opciones como gráfico en pantalla, gráficos en impresora, impresión de resultados numéricos, etc. Las opciones de mayor utilidad se han incluido en un menú propio de presentación de resultados que se exhiben en la parte superior de la pantalla. En cualquier momento se puede salir del menú de resultados presionando la tecla ESC, el retorno al mismo se lo consigue presionando las teclas Alt y A simultáneamente. Cualquier error que se cometa dentro de la hoja, puede ser eliminado presionando las teclas Ctrl y Break simultáneamente y luego la tecla Esc. Los comandos dentro del menú de resultados son:

CHDIR. Permite cambiar el directorio corriente. Esta opción se la utiliza para indicar al programa cual es el directorio donde se encuentra el archivo de datos y resultados. En el caso del ejemplo, este se corrió en un equipo con dos unidades de disco, con el disco del programa en la unidad A y el de datos en la B; por tanto, debemos escoger esta opción para cambiar el directorio al B. Inmediatamente después de escoger este comando, en la línea de control de la hoja (parte superior de la pantalla), se muestra la consulta:

Enter current directory:

para el caso del ejemplo la respuesta será B:\

REVISAR. Carga en la hoja el archivo de datos y resultados. Supongamos que el nombre del archivo de datos introducido al inicio del programa fue B:EJEMPLO, entonces, al escoger esta opción la consulta y respuesta será:

Enter name of file to import: EJEMPLO

Pasarán algunos segundos mientras se cargan y se etiquetan las tablas de valores, al final del proceso la estructura de la hoja es la siguiente:

DATOS DEL SISTEMA:

| REFERENCIAS: | CODIGO | MODULO                  | REF1 | REF2      | REF3   | REF4   |
|--------------|--------|-------------------------|------|-----------|--------|--------|
|              | 0      | Sumador                 |      |           |        |        |
|              | 1      | Integrad. C.Inicial     |      |           |        |        |
|              | 2      | Limitad. Máximo         |      | Mínimo    |        |        |
|              | 3      | Comparad. Referenc.     |      | Máximo    | Mínimo |        |
|              | 4      | Cmp.Hist. Ref.Min.      |      | Ref. Máx. | Máximo | Mínimo |
|              | 5      | Rec.M.Ond.              |      |           |        |        |
|              | 6      | Val. Abs.               |      |           |        |        |
|              | 7      | Zon.Muerta Amplt. ZM    |      |           |        |        |
|              | 8      | Ali.Juego Amplt.Jueg.   |      | Máximo    | Mínimo |        |
|              | 9      | Cuadrador               |      |           |        |        |
|              | 10     | Raiz.Cuad.              |      |           |        |        |
|              | 11     | Logaritmo               |      |           |        |        |
|              | 12     | Exponenc.               |      |           |        |        |
|              | 13     | Seno                    |      |           |        |        |
|              | 14     | Coseno                  |      |           |        |        |
|              | 15     | Retardo · Tiempo Retar. |      |           |        |        |
|              | 16     | Monostab. Tiempo Pulso  |      |           |        |        |
|              | 17     | Multiplic.              |      |           |        |        |
|              | 18     | Divisor l=Dividendo     |      |           |        |        |
|              | 19     | Sw.Analogo l=Control    |      |           |        |        |
|              | 20     | AND                     |      |           |        |        |
|              | 21     | NAND                    |      |           |        |        |
|              | 22     | OR                      |      |           |        |        |
|              | 23     | NOR                     |      |           |        |        |
|              | 24     | Biestable l=Set         |      |           |        |        |

NUMERO DE ENTRADAS: 1  
 NUMERO DE SALIDAS: 6  
 TIEMPO INICIAL: 0  
 TIEMPO FINAL: 16  
 INTERVALO ENTRE CALCULOS: 0.1  
 ERROR DE PASO ESTIMADO: 0.0001

| DATOS: | DESDE | HASTA | CODIGO POT.ENTRADA | REF1 | REF2 | REF3 | REF4 |
|--------|-------|-------|--------------------|------|------|------|------|
|        | 6     | 1     | 0                  | -1   | 0    | 0    | 0    |
|        | 2     | 3     | 0                  | 1    | 0    | 0    | 0    |
|        | 4     | 3     | 0                  | -0.4 | 0    | 0    | 0    |
|        | 1     | 2     | 2                  | 1    | 0.4  | -0.3 | 0    |
|        | 3     | 4     | 1                  | 1    | 0    | 0    | 0    |
|        | 4     | 5     | 7                  | 1    | 0.4  | 0    | 0    |
|        | 5     | 6     | 1                  | 1    | 0    | 0    | 0    |

| ENTRADAS: | SUBINDICE | MODULO |
|-----------|-----------|--------|
|           | 1         | 1      |

RESULTADOS:

| TIEMPO | ENTRADAS: |  | SALIDAS:    |             |          |          |          |          |
|--------|-----------|--|-------------|-------------|----------|----------|----------|----------|
|        | 1         |  | 1           | 2           | 3        | 4        | 5        | 6        |
| 0      | 1         |  | 1           | 0.4         | 0.4      | 0        | 0        | 0        |
| 0.3    | 1         |  | 1           | 0.4         | 0.354760 | 0.113079 | 0        | 0        |
| 0.6    | 1         |  | 0.9997771   | 0.4         | 0.314651 | 0.213372 | 0.013372 | 0.000222 |
| 0.9    | 1         |  | 0.9821559   | 0.4         | 0.279070 | 0.302323 | 0.102323 | 0.017044 |
| 1.2    | 1         |  | 0.9393002   | 0.4         | 0.247513 | 0.301216 | 0.101216 | 0.006611 |
| 1.5    | 1         |  | 0.8743176   | 0.4         | 0.219524 | 0.451100 | 0.251100 | 0.125602 |
| 1.8    | 1         |  | 0.789466    | 0.4         | 0.194700 | 0.513247 | 0.313247 | 0.210534 |
| 2.1    | 1         |  | 0.6870704   | 0.4         | 0.172604 | 0.560209 | 0.360209 | 0.312929 |
| 2.4    | 1         |  | 0.5691145   | 0.4         | 0.153157 | 0.617107 | 0.417107 | 0.430085 |
| 2.7    | 1         |  | 0.4373570   | 0.4         | 0.135030 | 0.660404 | 0.460404 | 0.562642 |
| 3      | 1         |  | 0.2941909   | 0.2941909   | 0.019192 | 0.607516 | 0.407516 | 0.705001 |
| 3.3    | 1         |  | 0.1492403   | 0.1492403   | -0.11953 | 0.671930 | 0.471930 | 0.850759 |
| 3.6    | 1         |  | 0.01404507  | 0.01404507  | -0.23250 | 0.610303 | 0.410303 | 0.905154 |
| 3.9    | 1         |  | -0.09806025 | -0.09806025 | -0.31313 | 0.535672 | 0.335672 | 1.090060 |
| 4.2    | 1         |  | -0.1046660  | -0.1046660  | -0.35029 | 0.434059 | 0.234059 | 1.104667 |
| 4.5    | 1         |  | -0.2304047  | -0.2304047  | -0.36017 | 0.324234 | 0.124234 | 1.230405 |
| 4.8    | 1         |  | -0.2594004  | -0.2594004  | -0.34595 | 0.216371 | 0.016371 | 1.259400 |
| 5.1    | 1         |  | -0.2596012  | -0.2596012  | -0.30707 | 0.110492 | 0        | 1.259601 |
| 5.4    | 1         |  | -0.2596012  | -0.2596012  | -0.27235 | 0.031601 | 0        | 1.259601 |
| 5.7    | 1         |  | -0.2596012  | -0.2596012  | -0.24155 | -0.04531 | 0        | 1.259601 |
| 6      | 1         |  | -0.2596012  | -0.2596012  | -0.21424 | -0.11136 | 0        | 1.259601 |
| 6.3    | 1         |  | -0.2596012  | -0.2596012  | -0.19001 | -0.17416 | 0        | 1.259601 |
| 6.6    | 1         |  | -0.2574515  | -0.2574515  | -0.16634 | -0.22776 | -0.02776 | 1.257452 |
| 6.9    | 1         |  | -0.2421000  | -0.2421000  | -0.13294 | -0.27209 | -0.07209 | 1.242101 |
| 7.2    | 1         |  | -0.2140411  | -0.2140411  | -0.09212 | -0.30670 | -0.10670 | 1.214041 |
| 7.5    | 1         |  | -0.1793135  | -0.1793135  | -0.04016 | -0.32705 | -0.12705 | 1.179314 |
| 7.8    | 1         |  | -0.1394436  | -0.1394436  | -0.00512 | -0.33570 | -0.13570 | 1.139444 |
| 8.1    | 1         |  | -0.09907042 | -0.09907042 | 0.033475 | -0.33130 | -0.13130 | 1.099070 |
| 8.4    | 1         |  | -0.06167105 | -0.06167105 | 0.064096 | -0.31642 | -0.11642 | 1.061672 |
| 8.7    | 1         |  | -0.030030   | -0.030030   | 0.007302 | -0.29335 | -0.09335 | 1.030030 |
| 9      | 1         |  | -0.00610604 | -0.00610604 | 0.099025 | -0.26503 | -0.06503 | 1.006107 |
| 9.3    | 1         |  | 0.000754015 | 0.000754015 | 0.10253  | -0.23443 | -0.03443 | 0.991246 |
| 9.6    | 1         |  | 0.01453441  | 0.01453441  | 0.095290 | -0.20440 | -0.00440 | 0.985465 |
| 9.9    | 1         |  | 0.01460791  | 0.01460791  | 0.005473 | -0.17716 | 0        | 0.985392 |
| 10.2   | 1         |  | 0.01460791  | 0.01460791  | 0.075000 | -0.15300 | 0        | 0.985392 |
| 10.5   | 1         |  | 0.01460791  | 0.01460791  | 0.067236 | -0.13157 | 0        | 0.985392 |
| 10.8   | 1         |  | 0.01460791  | 0.01460791  | 0.059633 | -0.11256 | 0        | 0.985392 |
| 11.1   | 1         |  | 0.01460791  | 0.01460791  | 0.052009 | -0.09570 | 0        | 0.985392 |
| 11.4   | 1         |  | 0.01460791  | 0.01460791  | 0.046909 | -0.08075 | 0        | 0.985392 |
| 11.7   | 1         |  | 0.01460791  | 0.01460791  | 0.041604 | -0.06749 | 0        | 0.985392 |
| 12     | 1         |  | 0.01460791  | 0.01460791  | 0.036099 | -0.05573 | 0        | 0.985392 |
| 12.3   | 1         |  | 0.01460791  | 0.01460791  | 0.032727 | -0.04529 | 0        | 0.985392 |
| 12.6   | 1         |  | 0.01460791  | 0.01460791  | 0.029026 | -0.03604 | 0        | 0.985392 |
| 12.9   | 1         |  | 0.01460791  | 0.01460791  | 0.025744 | -0.02704 | 0        | 0.985392 |
| 13.2   | 1         |  | 0.01460791  | 0.01460791  | 0.022033 | -0.02056 | 0        | 0.985392 |
| 13.5   | 1         |  | 0.01460791  | 0.01460791  | 0.020251 | -0.01410 | 0        | 0.985392 |
| 13.8   | 1         |  | 0.01460791  | 0.01460791  | 0.017961 | -0.00830 | 0        | 0.985392 |
| 14.1   | 1         |  | 0.01460791  | 0.01460791  | 0.015930 | -0.00330 | 0        | 0.985392 |
| 14.4   | 1         |  | 0.01460791  | 0.01460791  | 0.014120 | 0.001197 | 0        | 0.985392 |
| 14.7   | 1         |  | 0.01460791  | 0.01460791  | 0.012531 | 0.005192 | 0        | 0.985392 |
| 15     | 1         |  | 0.01460791  | 0.01460791  | 0.011114 | 0.000734 | 0        | 0.985392 |
| 15.3   | 1         |  | 0.01460791  | 0.01460791  | 0.009057 | 0.011076 | 0        | 0.985392 |
| 15.6   | 1         |  | 0.01460791  | 0.01460791  | 0.000742 | 0.014663 | 0        | 0.985392 |



La forma en la cual se codifican los resultados es bastante sencilla de interpretar. En la parte superior de la hoja se tiene un cuadro de referencias las cuales indican el significado de cada una de las referencias para un determinado módulo. Un poco por debajo se indica los datos generales del sistema como son: número de entradas externas, número de salidas, tiempo inicial de compilación, tiempo final, intervalo de integración y el error de paso estimado. La tabla de datos que se encuentra a continuación indica cual es la estructura del diagrama analizado y los diferentes parámetros que lo afectan; así, interpretemos la cuarta fila de esta tabla

- esta fila indica cual es el módulo existente entre los puntos 1 y 2 del diagrama (desde 1 hasta 2);
- la tabla nos dice que existe un módulo cuyo código es 2, por tanto, es un limitador según el cuadro de referencias;
- el potenciómetro que afecta a la entrada del limitador tiene un valor unitario;
- la referencia 1, que según el cuadro de referencias corresponde al valor máximo de limitación tiene, un valor de 0.4;
- la referencia 2 correspondiente al valor mínimo al cual se limita tiene un valor de -0.3 y por último,
- las referencias 3 y 4 no tienen ningún sentido en esta fila.

Como se puede ver, el cuadro de datos define completamente el diagrama de simulación de la Fig. A.2. A continuación se describe los subíndices de las entradas. En el ejemplo, existe una entrada cuyo subíndice es 1 que está aplicada al módulo de salida es 1.

Finalmente se tiene el cuadro de resultados numéricos tanto de las entradas externas como de las salidas para cada instante de tiempo.

IMPRIMIR. Se lo utiliza para imprimir en papel una parte de la hoja. Las consultas que realiza el programa cuando se escoge esta opción son:

Enter print range:  
Enter Left Margin (0..240):

```
Enter Right Margin (0..240):  
Enter Top Margin (0..32):  
Enter Botton Margin (0..100):  
Enter Lines per Page (1..100):
```

Los datos respectivos se ingresarán de acuerdo al formato de impresión que se requiera.

GRAFICAR. Permite graficar resultados en la pantalla. Se puede graficar hasta seis rangos de valores simultaneamente. Las consultas y respuestas para graficar todas las salidas del ejemplo en función del tiempo serían:

```
Enter X axis range: B57..B111  
Enter first data range: F57..F111  
Enter second data range: G57..G111  
Enter third data range: H57..H111  
Enter fourth data range: I57..I111  
Enter fifth data range: J57..J111  
Enter sixth data range: K57..K111  
Enter graph title, top line: Fig. A.3. SALIDAS DEL EJEMPLO  
Enter graph title, second line: SSO CON ALINEALIDADES  
Enter X axis title: Tiempo (s)  
Enter Y axis title: Salidas  
Enter legend for A range: Y1  
Enter legend for B range: Y2  
Enter legend for C range: Y3  
Enter legend for D range: Y4  
Enter legend for E range: Y5  
Enter legend for F range: Y6
```

El gráfico obtenido es del tipo XY, con reticulado y ajuste automático de escalas, tal como lo muestra la Fig. A.3. Es posible editar este gráfico si utilizamos los comandos adicionales de LOTUS, es decir, se puede salir al menú de LOTUS, entrar a la modalidad de gráficos y editarlo. Adicionalmente, hemos obtenido los gráficos de la Fig. A.4 y A.5. con el fin de ilustrar las características del programa en este sentido.

PGRAFIC. Este comando se lo utiliza luego de creado un gráfico en la pantalla con el fin de almacenarlo para su impresión. Al escoger esta opción el programa pregunta por el nombre del archivo .PIC bajo el siguiente formato:

# Fig. A.3 SALIDAS DEL EJEMPLO

SSO CON ALINEALIDADES

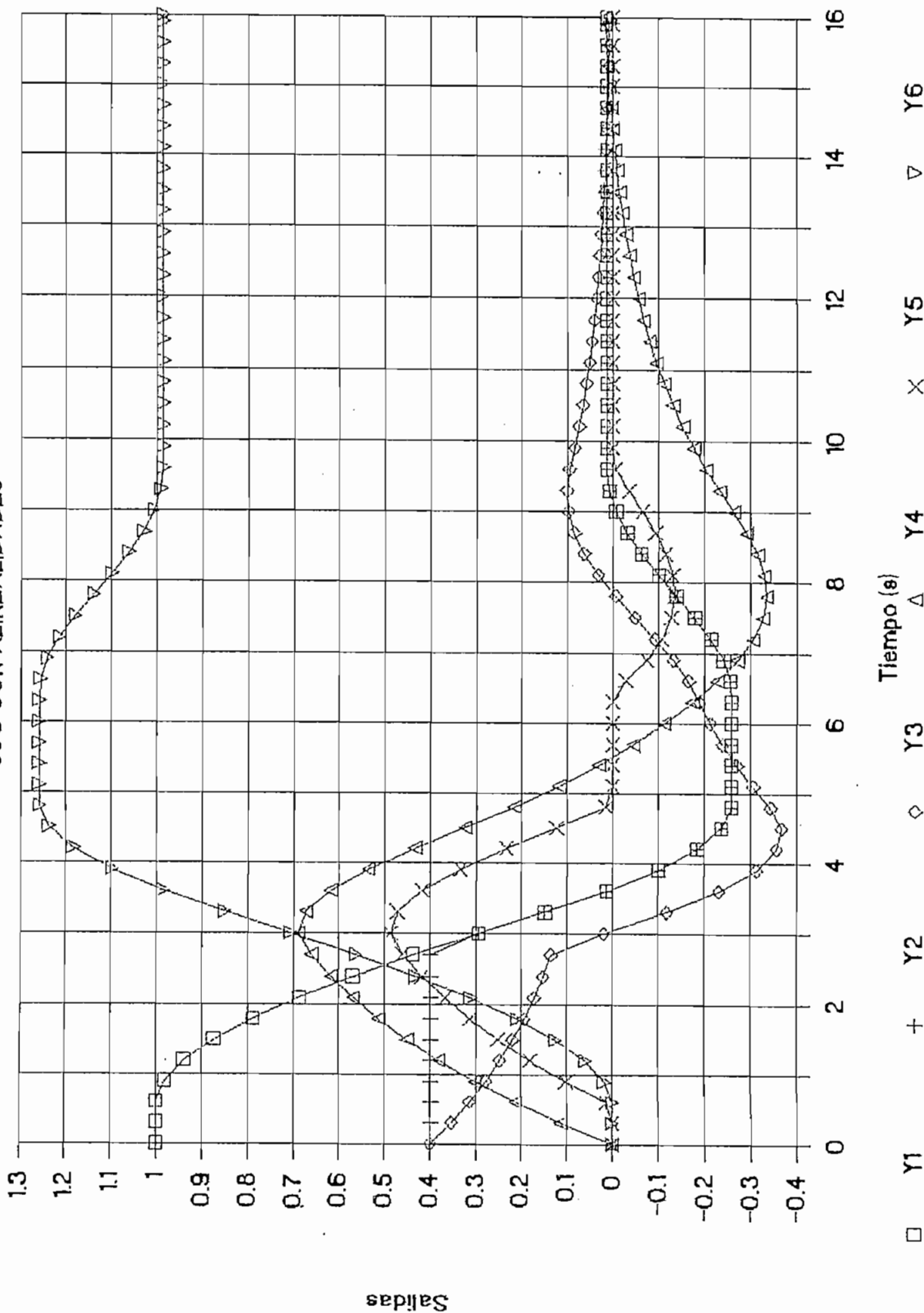


Fig. A.4 GRAFICO EN EL PLANO DE FASE

SSO CON ALINEALIDADES

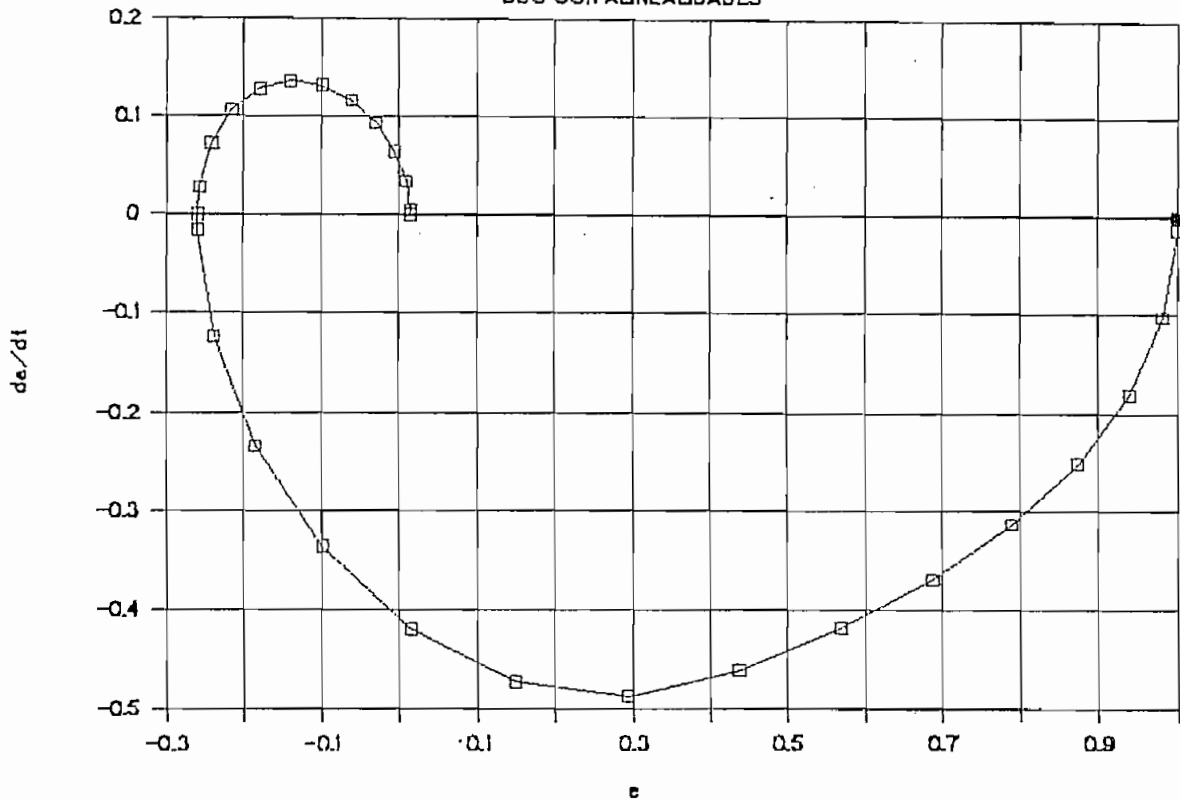
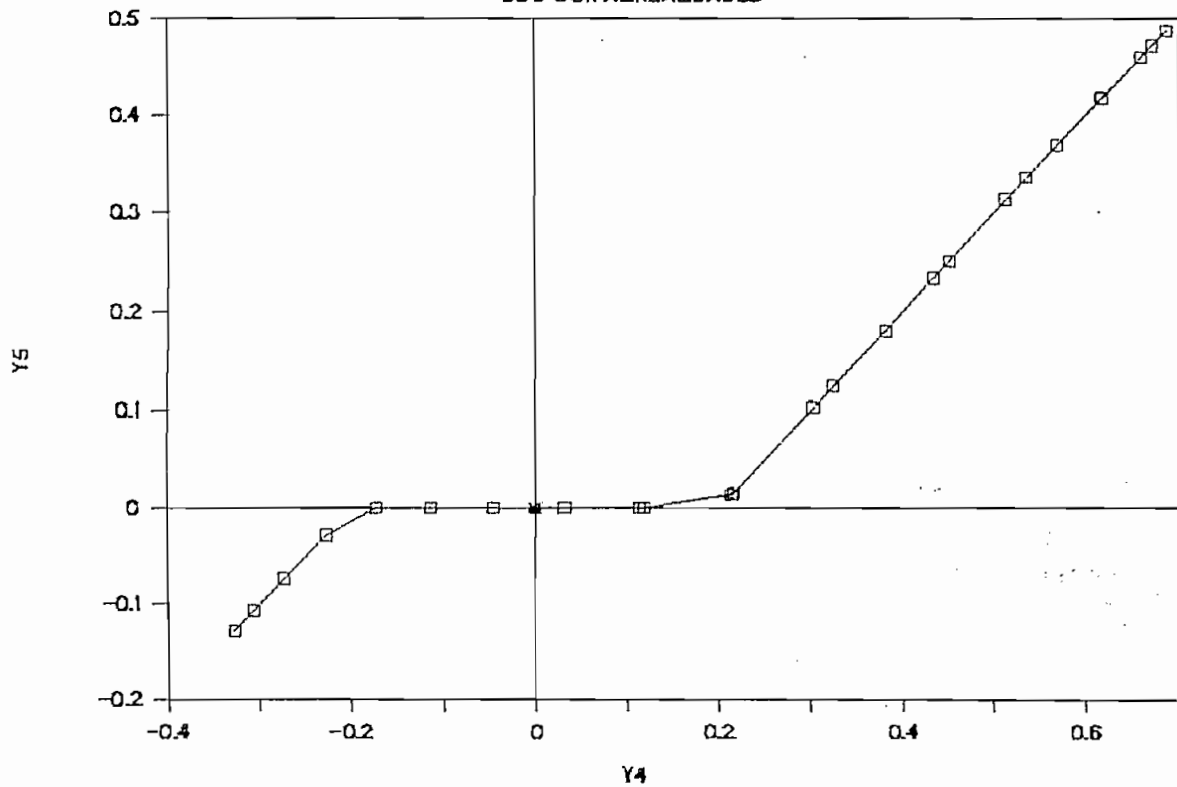


Fig. A.5 ALINEALIDAD ZONA MUERTA

SSO CON ALINEALIDADES



Enter graph file name:

ALMACENAR. Permite almacenar la hoja como un archivo de LOTUS. Un archivo almacenado de esta manera, puede ser revisado en lo futuro independientemente del programa SDSC. El nombre del archivo de LOTUS es requerido con la siguiente consulta:

Enter save file name:

OTROS. Permite el acceso a otros comandos del menú

OTROS/ CAMBIAR. Se escoge esta opción cuando se requiera cambiar uno o más parámetros del modelo de simulación. Los nuevos datos serán analizados mediante la opción 2. del menú principal. Los datos factibles de cambiar son:

Tiempo inicial  
Tiempo final  
Intervalo de tiempo entre cálculos  
Potenciómetros  
Valores referenciales.

Al escoger esta opción, el cursor se localiza y puede moverse solamente por las casillas que contienen los valores de las variables antes citadas. El cambio se realiza ubicándose en la casilla correspondiente, digitando el nuevo valor y luego [Enter]. Al terminar los cambios, basta tipear [Enter] para salir al menú de resultados. En el ejemplo, escogemos esta opción para cambiar el valor del potenciómetro incluido en el ramal que va desde 4 hasta 3 a -0.2.

OTROS/ ALMACENAR. Almacena los datos cambiados en el disco de datos para el futuro análisis. Para el caso del ejemplo, si queremos un intervalo de tiempo entre impresos de 0.3 y trabajar con el mismo archivo de datos responderemos:

Intervalo de tiempo entre impresos? 0.3  
Enter print file name: EJEMPLO

OTROS/ MENU. Se lo utiliza para retornar al menú inicial.

OTROS/ SALIR. Con esta opción se sale del menú de resultados al menú de LOTUS.

OTROS/ FIN. Permite abandonar el trabajo de revisión de resultados para salir a imprimir gráficos o al menú principal.

Cuando salimos de la hoja electrónica, se presenta en la pantalla un nuevo menú. Las opciones ahora disponibles son:

1-2-3  
PrintGraph  
Exit

1-2-3. Permite el retorno a la hoja de trabajo. Cuando entremos a la hoja por este camino, para tener nuevamente el menú de resultados disponible es necesario cargar el programa MENU.WK1 desde el directorio corriente.

PrintGraph. Se lo utiliza para obtener copias impresas de los gráficos creados en la hoja. Los diferentes parámetros requeridos por el programa PrintGraph, tales como: directorio de gráficos, tipo de impresora, dimensiones del papel, dimensiones del gráfico, tipo de letra, etc. son de fácil introducción ya que se los solicita con la ayuda de menús.

Exit. Permite el retorno al menú principal.

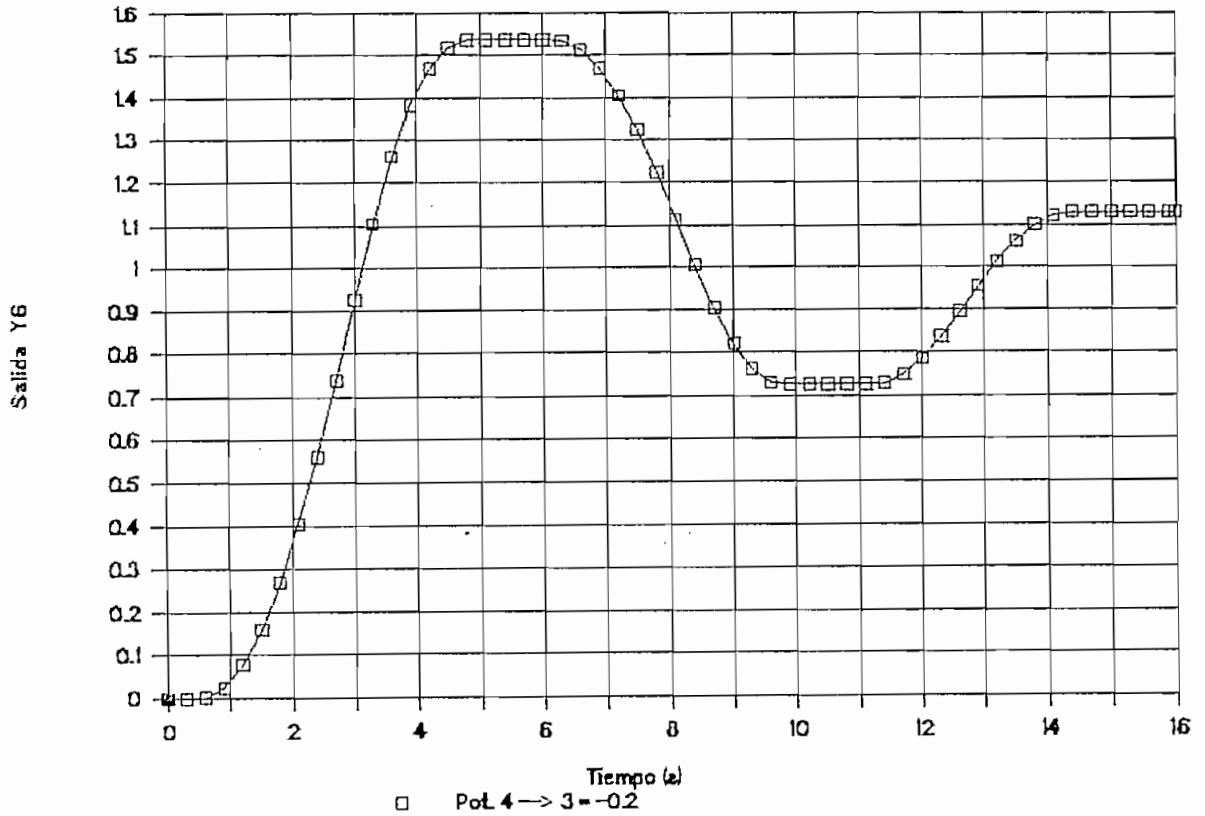
#### A.2.3 Problema con datos cambiados.

Ya hemos cambiado uno de los datos del ejemplo y queremos averiguar el comportamiento del sistema con el nuevo dato para lo cual, escogemos la opción 2. del menú principal. Inmediatamente después de seleccionar esta opción, el programa pide por el nombre del archivo tipo PRN que contiene los datos cambiados. Para el ejemplo, el nombre con el cual grabamos los nuevos datos desde la hoja es EJEMPLO.PRN, por tanto, se debe digitar este nombre. Como último paso, se pide la generación de las funciones de entrada y se procede en forma similar a como se lo hizo al escoger la opción 1. del menú principal.

En la Fig. A.6 se muestra el gráfico de la salida en función del tiempo para el nuevo sistema.

Fig. A.6 SALIDA DEL EJEMPLO

SSO CON ALINEALIDADES .



ANEXO B



## ANEXO B

### OPCIONES ADICIONALES DE LOTUS 123

Al presentar los resultados en la hoja LOTUS 123 se consigue tener gran versatilidad en el tratamiento de las tablas de datos. Muchas operaciones adicionales a las indicadas en el menú de presentación de resultados pueden ser ejecutadas mediante la utilización de los comandos propios de LOTUS. Algunas operaciones generalmente requeridas son:

- cambiar el etiquetado de las tablas de datos,
- borrar rangos de valores,
- formateo de números en lo referente a decimales,
- análisis estadístico de los resultados,
- formatos adicionales para impresión,
- edición de gráficos,
- etc.

No vamos a detallar la forma de tratar cada una de estas opciones ya que son de fácil ejecución si es que el usuario tiene conocimientos básicos sobre el manejo de los comandos de la hoja.

Una operación que es de mucha importancia y que involucra cierta complejidad en su ejecución, consiste en obtener un gráfico que muestre el comportamiento de una salida frente a la variación de un parámetro del diagrama. La operación requiere varias evaluaciones del diagrama de simulación y juntar los resultados de la salida de interés en una sola hoja, desde la cual se puede obtener el gráfico deseado.

Para ilustrar el método de realizar esta operación, consideremos el sistema de segundo orden que se muestra en la Fig. B.1. Se requiere en un solo gráfico las señales de salida para

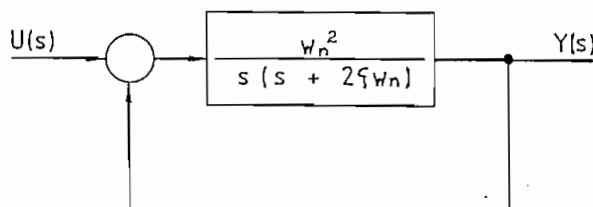


Fig. B.1 Sistema de segundo orden.

$W_n=1$  y valores de  $\zeta$  iguales a 0.1, 0.2, 0.4, 0.7, 1.0 y 2.0.

El diagrama de simulación del problema se muestra en la Fig. B.2.

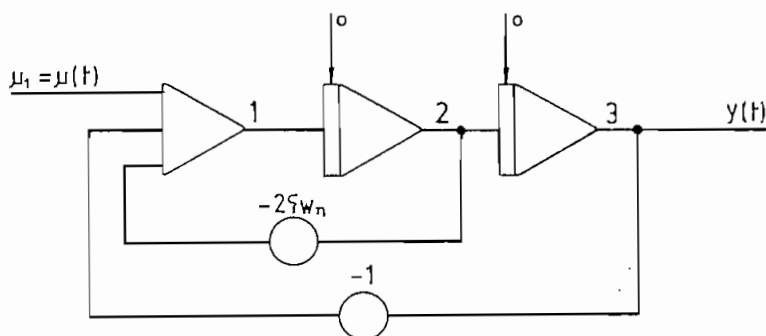


Fig. B.2 Diagrama de simulación del sistema de segundo orden.

En primer lugar corremos el programa para  $\zeta=2$  (valor del potenciómetro desde 2 hasta 1 = -4),  $t_0=0$ ,  $t_f=16$ ,  $h=0.1$ , intervalo entre impresos de 0.4 y SEGUNDO como nombre del archivo de datos; y revisamos los resultados. Estando en la hoja que contiene los resultados, con la tecla Esc salimos del menú de resultados. Ahora podemos movernos con el cursor hasta encontrar la columna de valores que corresponde a la salida del sistema ( $y_3$ ), en la parte superior de esta columna ponemos la etiqueta 2, correspondiente a la relación de amortiguamiento para la cual está evaluada la salida. Esta columna, incluida su etiqueta puede ser almacenada como una hoja de LOTUS mediante el comando / File Xtract Values. Para el efecto LOTUS pide por el nombre del archivo donde se almacenará la columna y el rango de la misma. La secuencia de operaciones será:

```
/FXV
Enter xtract file name: SEG1
Enter xtract range: H52..H94
```

El nombre que hemos dado a la hoja es SEG1.WK1 y la porción de columna almacenada está comprendida entre H52 y H94, para el caso del ejemplo. Ahora presionamos simultaneamente las teclas Alt . y A para entrar nuevamente al menú de resultados. Utilizamos la opción OTROS CAMBIAR para cambiar el valor del potenciómetro desde 2 hasta 1 a -2 de forma que,  $\zeta=1$ . Con la opción OTROS ALMACENAR, almacenamos los nuevos datos en el archivo SEGUNDO.PRN, con el mismo intervalo entre impresos (0.4). Salimos de la hoja al menú principal y con la opción 2. de este evaluamos el diagrama para el nuevo valor de  $\zeta$ .

Una vez evaluado el diagrama procedemos en forma idéntica para crear las hojas que contengan los resultados de la salida para valores de  $\zeta$  iguales a 1.0, 0.7, 0.4 y 0.2. Los nombres dados a los diferentes archivos son:

| $\zeta$ | Archivo  |
|---------|----------|
| 1.0     | SEG2.WK1 |
| 0.7     | SEG3.WK1 |
| 0.4     | SEG4.WK1 |
| 0.2     | SEG5.WK1 |

Al realizar la última corrida para  $\zeta=0.1$  utilizamos la hoja de resultados para combinar los diferentes valores. Así, evaluamos el diagrama para  $\zeta=0.1$  y revisamos los resultados, con Esc salimos del menú de resultados, etiquetamos la salida  $y_3$  con el valor de 0.1 y nos ubicamos con el cursor en la columna contigua a la que lleva los resultados de  $y_3$ , a la altura de la etiqueta. La secuencia de comandos /File Combien Copy Entire-File de LOTUS, permite cargar los archivos de tipo WK1 anteriormente creados. La operación para cargar la hoja SEG5.WK1 que contiene los valores de la salida para  $\zeta=0.2$  es:

```
/FCCE
Name of file to combine: SEG5
```

El archivo SEG5.WK1 se carga a partir de la posición del cursor

hacia abajo de forma que, para cargar los archivos restantes es necesario repetir la operación descrita ubicándose cada vez con el cursor en la columna contigua.

Terminada la operación de carga, tenemos una hoja con los valores de la salida del sistema para cada uno de los parámetros. La tabla B.1 muestra el resultado final. Ahora podemos hacer uso de la opción GRAFICAR del menú de resultados para obtener el gráfico deseado, este gráfico se muestra en la Fig. B.3.

La operación descrita es sin duda de gran importancia en el proceso de diseño de sistemas, razón por la cual la hemos detallado.

FACTOR DE AMORTIGUAMIENTO

| TIEMPO | 0.1       | 0.2       | 0.4       | 0.7       | 1         | 2         |
|--------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0.0    | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| 0.4    | 0.0768887 | 0.0749827 | 0.0711721 | 0.0666981 | 0.0615528 | 0.0495363 |
| 0.8    | 0.2879136 | 0.2736985 | 0.2483581 | 0.2167468 | 0.1912889 | 0.1344248 |
| 1.2    | 0.5987650 | 0.5490657 | 0.4785652 | 0.3974950 | 0.3373737 | 0.2197651 |
| 1.6    | 0.9324246 | 0.8495878 | 0.7161882 | 0.5735881 | 0.4758698 | 0.2984729 |
| 2.0    | 1.2588690 | 1.1274848 | 0.9278846 | 0.7257137 | 0.5939947 | 0.3696481 |
| 2.4    | 1.5195218 | 1.3457680 | 1.0990790 | 0.8457737 | 0.6915593 | 0.4336773 |
| 2.8    | 1.6828468 | 1.4813188 | 1.1962668 | 0.9329984 | 0.7689219 | 0.4912292 |
| 3.2    | 1.7285898 | 1.5266110 | 1.2468568 | 0.9989228 | 0.8287988 | 0.5429378 |
| 3.6    | 1.6618878 | 1.4883768 | 1.2582398 | 1.0251868 | 0.8743188 | 0.5893988 |
| 4.0    | 1.4983288 | 1.3845438 | 1.2188848 | 1.0415978 | 0.9084218 | 0.6311231 |
| 4.4    | 1.2725458 | 1.2399588 | 1.1665428 | 1.0459888 | 0.9337823 | 0.6686141 |
| 4.8    | 1.0227698 | 1.0817438 | 1.1068328 | 1.0429368 | 0.9522675 | 0.7022947 |
| 5.2    | 0.7887428 | 0.9349943 | 1.0477678 | 1.0368828 | 0.9657973 | 0.7325522 |
| 5.6    | 0.6847612 | 0.8194275 | 0.9998834 | 1.0276898 | 0.9755941 | 0.7597345 |
| 6.0    | 0.4948944 | 0.7472927 | 0.9637285 | 1.0195938 | 0.9826487 | 0.7841541 |
| 6.4    | 0.4781131 | 0.7226924 | 0.9438144 | 1.0126888 | 0.9877844 | 0.8068918 |
| 6.8    | 0.5276358 | 0.7421876 | 0.9356723 | 1.0078828 | 0.9913124 | 0.8257998 |
| 7.2    | 0.6524868 | 0.7964148 | 0.9398323 | 1.0038748 | 0.9938779 | 0.8435849 |
| 7.6    | 0.8283245 | 0.8723258 | 0.9497697 | 1.0004138 | 0.9956968 | 0.8594184 |
| 8.0    | 1.0825948 | 0.9556394 | 0.9642579 | 0.9988429 | 0.9969887 | 0.8736993 |
| 8.4    | 1.1784568 | 1.0331878 | 0.9796513 | 0.9988868 | 0.9978861 | 0.8865368 |
| 8.8    | 1.2995588 | 1.0942918 | 0.9935786 | 0.9978851 | 0.9985226 | 0.8988888 |
| 9.2    | 1.3732688 | 1.1326728 | 1.0045688 | 0.9988268 | 0.9989693 | 0.9084279 |
| 9.6    | 1.3846838 | 1.1468218 | 1.0119178 | 0.9983451 | 0.9992819 | 0.9177349 |
| 10.0   | 1.3368548 | 1.1368938 | 1.0156588 | 0.9987274 | 0.9995885 | 0.9268968 |
| 10.4   | 1.2415918 | 1.1077778 | 1.0162468 | 0.9998997 | 0.9996538 | 0.9336873 |
| 10.8   | 1.1169978 | 1.0679248 | 1.0144848 | 0.9994211 | 0.9997593 | 0.9403551 |
| 11.2   | 0.9842299 | 1.0248538 | 1.0112478 | 0.9996747 | 0.9998332 | 0.9464171 |
| 11.6   | 0.8648633 | 0.9831599 | 1.0073778 | 0.9998589 | 0.9998846 | 0.9518631 |
| 12.0   | 0.7737618 | 0.9587697 | 1.0035788 | 0.9999812 | 0.9999282 | 0.9567555 |
| 12.4   | 0.7247282 | 0.9383518 | 1.0003248 | 1.0000538 | 0.9999449 | 0.9611587 |
| 12.8   | 0.7212316 | 0.9231129 | 0.9979222 | 1.0000888 | 0.9999628 | 0.9658992 |
| 13.2   | 0.7683888 | 0.9281626 | 0.9964592 | 1.0000978 | 0.9999738 | 0.9686463 |
| 13.6   | 0.8327353 | 0.9429464 | 0.9958752 | 1.0000918 | 0.9999828 | 0.9718329 |
| 14.0   | 0.9249796 | 0.9638882 | 0.9968885 | 1.0000768 | 0.9999876 | 0.9746957 |
| 14.4   | 1.0215158 | 0.9869681 | 0.9966446 | 1.0000598 | 0.9999915 | 0.9772675 |
| 14.8   | 1.1873638 | 1.0085548 | 0.9975599 | 1.0000418 | 0.9999942 | 0.9795788 |
| 15.2   | 1.1783218 | 1.0257888 | 0.9985542 | 1.0000278 | 0.9999961 | 0.9816536 |
| 15.6   | 1.2825978 | 1.0365618 | 0.9994716 | 1.0000158 | 0.9999973 | 0.9835182 |
| 16.0   | 1.2816518 | 1.0484838 | 1.0002898 | 1.0000068 | 0.9999982 | 0.9851933 |

Tabla B.1 Salidas de un sistema de segundo orden para diferentes valores de  $\zeta$ .

Fig. B3 SISTEMA DE SEGUNDO ORDEN

